



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**  
**FACULTAD DE INGENIERÍA**

***Diseño de un Sistema de Comando y  
Manejo de Información***

***TESIS***

*Que para obtener el título de  
Ingeniero Eléctrico y Electrónico*

***PRESENTA:***

*Christo Aldair Lara Tenorio*

***DIRECTOR DE TESIS:***

*Dr. Saúl de la Rosa Nieves*



*Ciudad Universitaria, Cd. Mx, 2021*

## *Agradecimientos*

---

En primer lugar, siempre estaré agradecido con mi tutor, el doctor Saúl de la Rosa Nieves, por creer en mí y darme la oportunidad de crecer personal y profesionalmente. Le agradezco por todo el tiempo, conocimiento, experiencia, apoyo y la guía que me ha brindado para desempeñarme de mejor manera y elaborar cada una de las etapas necesarias para este proyecto.

A mi hermana Daniela, por ser mi compañera de vida, quien me da ánimos para seguir adelante y ser un motivo para esforzarme en todas mis actividades.

A mi madre Gloria, por apoyarme en todo momento, por las enseñanzas, valores y todo lo que me ha aportado para salir adelante. Gracias por ser mi soporte, mi motivación para no rendirme, para seguir adelante en mis estudios y dar comienzo a mi vida profesional.

A mi padre Martín, quien me ha orientado y aconsejado para lograr mis objetivos. Tus enseñanzas y apoyo me han convertido en una persona de bien. Gracias por tu apoyo, paciencia, enseñanzas y las fuerzas que me das para continuar creciendo.

A mis padres en general, por confiar en mí y en mis decisiones, por estar en los momentos difíciles, por enseñarme, orientarme y apoyarme. Gracias por brindarme de todas las herramientas que he necesitado durante mis estudios y mi vida. Gracias por ser mis dos mayores ejemplos de vida.

A Rebeca, por darme nuevas motivaciones para crecer como persona y como ingeniero. Gracias por estar a mi lado en todo momento, por ayudarme a mejorar y mejorar junto a mí. Me has enseñado a disfrutar del área a la que pertenecemos y establecer nuevos y mejores objetivos para el futuro.

A mis grandes amigos de la carrera, Diego, Venancio, Mariana, Alberto, Estefany, Noemi, Pepe, Osvaldo y a todos aquellos con quienes pasé momentos increíbles. Me enseñaron muchas cosas y le dieron un brillo especial a la facultad.

A todos mis compañeros y grandes amigos del LIESE, a Diego, Sandra, Alejandro, Eduardo, Erika, Daniel, Lars, Jorge y Guillermo, de quienes he aprendido bastante y juntos hemos disfrutado de excelentes momentos durante el trabajo.

A mi familia y amigos en general, por acompañarme y el apoyo que me han dado durante mi vida estudiantil.

Y, por último, a aquellos profesores que, además de enseñarme, generaron un interés especial para continuar desarrollándome en el área. Principalmente, le agradezco al doctor Cuevas, al maestro Mota, y al doctor Saúl.

Aldair

# Índice general

---

Agradecimientos .....	ii
Índice general .....	iii
Índice de figuras .....	vi
Índice de tablas .....	ix
Acrónimos .....	x
Resumen .....	xii
<b>Sección I Planteamiento del proyecto .....</b>	<b>1</b>
<b>1 Introducción .....</b>	<b>2</b>
1.1 Presentación del problema .....	2
1.2 Hipótesis .....	3
1.3 Objetivo .....	3
1.3.1 Objetivos particulares .....	3
1.4 Alcances .....	3
1.5 Justificación .....	3
<b>2 Metodología de diseño .....</b>	<b>4</b>
<b>Sección II Estado del arte .....</b>	<b>7</b>
<b>3 Estado del arte de los SCMI en nanosatélites bajo el estándar CubeSat .....</b>	<b>8</b>
3.1 Áreas de oportunidad .....	8
3.2 Estado del arte de los SCMI comerciales .....	12
3.2.1 ISIS On-Board Computer (IOBC) [4] .....	12
3.2.2 NanoMind A3200 [5] .....	14
3.2.3 On-Board Computer (C3S) [6] .....	16
3.2.4 KRYTEN-M3 [7] .....	17
3.2.5 CubeComputer Nanosatellite OBC [8] .....	19
3.2.6 OBC EnduroSat [9] .....	20
3.2.7 ABACUS [10] .....	22
3.2.8 HERCULES [11] .....	24
3.2.9 N-ART OBCOMMS V2 [12] .....	25
3.2.10 IMT CubeSat OBC [13] .....	26
3.2.11 SatBus 3C2 [14] .....	27
3.2.12 NANOobc Gen2 [15] .....	29
3.2.13 Análisis de los SCMI comerciales .....	31
3.3 Estado del arte de los SCMI de investigación .....	40
3.3.1 Computadora dentro del presupuesto tolerante a fallas para nanosatélites [16] .....	40
3.3.2 Mitigación a nivel de sistema de SEFI en arquitecturas de manejo de datos, una solución para satélites pequeños [18] .....	41
3.3.3 Implementación de un RTOS de código abierto para una computadora de a bordo (OBC) en STUDSAT-2 [19] .....	42
3.3.4 Encuesta sobre la implementación y fiabilidad de las interfaces eléctricas de bus en CubeSat [20] .....	43
3.3.5 Estandarización de la interfaz eléctrica de CubeSat para una entrega más rápida y más éxito en la misión [23] .....	46
<b>Sección III Marco teórico .....</b>	<b>48</b>
<b>4 Satélites artificiales .....</b>	<b>49</b>
4.1 Antecedentes históricos .....	49
4.2 Tipos de satélites artificiales .....	50
4.3 Satélites pequeños .....	51

4.4	Los nanosatélites .....	52
4.5	El estándar CubeSat .....	53
4.5.1	Sistemas de un CubeSat .....	53
4.5.2	Estructuras de lanzamiento .....	54
4.6	Sistema de comando y manejo de información (SCMI) .....	55
4.6.1	Arquitectura de la computadora .....	56
4.6.2	Factor de forma .....	58
4.7	Nivel de Madurez Tecnológica (TRL) .....	59
<b>5</b>	<b>Procesador de a bordo.....</b>	<b>61</b>
5.1	Unidad de procesamiento central (CPU).....	61
5.1.1	Microcontrolador .....	61
5.1.2	Arreglo de compuertas programables en campo (FPGA).....	62
5.1.3	Sistema en chip (SoC).....	62
5.2	Programación y depuración.....	62
5.3	Sistemas operativos en tiempo real (RTOS) .....	63
5.3.1	RTOS disponibles .....	64
<b>6</b>	<b>Unidades de almacenamiento .....</b>	<b>65</b>
6.1	Memoria .....	65
6.2	Bancos de memoria .....	66
6.3	Tecnologías de las memorias .....	67
6.3.1	Memoria dinámica de acceso aleatorio (DRAM) .....	67
6.3.2	Memoria estática de acceso aleatorio (SRAM).....	68
6.3.3	Memoria ferroeléctrica de acceso aleatorio (FRAM) .....	68
6.3.4	Memoria magnética de acceso aleatorio (MRAM).....	69
6.3.5	Memoria de solo lectura eléctricamente programable y borrrable (EEPROM).....	70
6.3.6	Flash .....	70
6.4	Protección de las memorias.....	72
6.4.1	Memorias con código de corrección de errores (ECC).....	72
<b>7</b>	<b>Interfaces eléctricas y periféricos.....</b>	<b>73</b>
7.1	Interfaces de entrada/salida paralelas .....	73
7.2	Interfaces de entrada/salida seriales .....	73
7.3	Interfaces de entrada/salida analógicas .....	79
7.4	Interfaces de entrada/salida de tiempo .....	80
7.5	Dispositivos externos .....	80
<b>8</b>	<b>Técnicas de mitigación y tolerancia a fallas .....</b>	<b>81</b>
8.1	Ambiente de radiación espacial.....	81
8.1.1	Fuentes de la radiación.....	81
8.2	Efectos de la radiación en los dispositivos electrónicos .....	83
8.2.1	Dosis total de ionización (TID).....	83
8.2.2	Efectos de desplazamiento .....	83
8.2.3	Efectos de evento único (SEE).....	84
8.3	Confiabilidad.....	85
8.3.1	Amenazas de la confiabilidad .....	85
8.3.2	Atributos.....	86
8.3.3	Medios de la confiabilidad .....	86
8.4	Tolerancia a fallas .....	87
8.4.1	Redundancia en hardware .....	87
8.4.2	Redundancia en software .....	89
8.4.3	Redundancia en información.....	90
8.4.4	Redundancia en tiempo .....	92

<i>Sección IV</i>	<i>Diseño y Construcción</i> .....	<i>94</i>
<b>9</b>	<b>Diseño conceptual</b> .....	<b>95</b>
9.1	Definición del concepto de diseño .....	95
9.1.1	Medio ambiente de radiación espacial .....	96
9.2	Definición de bloques funcionales .....	105
9.3	Planteamiento de los requisitos de sistema .....	106
9.4	Diseño conceptual .....	107
9.4.1	Diseño conceptual del procesador de a bordo .....	108
9.4.2	Diseño conceptual de los dispositivos de almacenamiento .....	108
9.4.3	Diseño conceptual de los puertos de entrada/salida .....	110
9.4.4	Diseño conceptual de los dispositivos externos .....	110
9.4.5	Diseño conceptual del factor de forma .....	111
<b>10</b>	<b>Diseño de detalle</b> .....	<b>113</b>
10.1	Planteamiento de las especificaciones objetivo de diseño .....	113
10.1.1	Especificaciones objetivo del SCMI .....	113
10.1.2	Especificaciones objetivo del procesador de a bordo .....	114
10.1.3	Especificaciones objetivo de los dispositivos de almacenamiento .....	115
10.1.4	Especificaciones objetivo de los puertos de entrada/salida .....	116
10.1.5	Especificaciones objetivo de los dispositivos externos .....	117
10.1.6	Especificaciones objetivo del factor de forma .....	118
10.2	Diseño de detalle .....	118
10.2.1	Diseño de detalle del procesador de a bordo .....	118
10.2.2	Diseño de detalle de los dispositivos de almacenamiento .....	123
10.2.3	Diseño de detalle de los puertos de entrada/salida .....	127
10.2.4	Diseño de detalle de los dispositivos externos .....	128
10.2.5	Diseño de detalle del factor de forma .....	132
10.3	Integración .....	133
10.3.1	Asignación de terminales .....	134
10.4	Estimación de las necesidades cuantitativas .....	145
<b>11</b>	<b>Construcción</b> .....	<b>146</b>
11.1	Diseño asistido por computadora (CAD) .....	146
11.1.1	Diseño del diagrama esquemático .....	146
11.1.2	Diseño de la tarjeta de circuito impreso (PCB) .....	148
	<i>Conclusiones</i> .....	<i>153</i>
	<i>Trabajo a futuro</i> .....	<i>154</i>
	<i>Referencias</i> .....	<i>155</i>

## Índice de figuras

---

<b>Figura 1</b>	Metodología de diseño .....	6
<b>Figura 2</b>	Nanosatélites lanzados y la predicción de los próximos 4 años .....	9
<b>Figura 3</b>	Tipos de nanosatélites lanzados por año .....	9
<b>Figura 4</b>	Tipos de nanosatélites desarrollados .....	10
<b>Figura 5</b>	Nanosatélites lanzados por institución al año .....	10
<b>Figura 6</b>	Nanosatélites desarrollados por institución .....	11
<b>Figura 7</b>	Empresas activas de nanosatélites fundadas por año .....	11
<b>Figura 8</b>	Número de nanosatélites lanzados por país .....	12
<b>Figura 9</b>	Diagrama de bloques del IOBC [4].....	13
<b>Figura 10</b>	IOBC, vistas superior, inferior e integrada a una daughterboard [4] .....	14
<b>Figura 11</b>	Diagrama de bloques del NanoMind A3200 [5] .....	15
<b>Figura 12</b>	NanoMind A3200, vistas exterior, interior e integrada en una NanoDock [5] .....	16
<b>Figura 13</b>	Diagrama de bloques del KRYTEN-M3 [7] .....	17
<b>Figura 14</b>	KRYTEN-M3 [7] .....	18
<b>Figura 15</b>	Diagrama de bloques del CubeComputer Nanosatellite OBC [8].....	19
<b>Figura 16</b>	CubeComputer Nanosatellite OBC [8] .....	20
<b>Figura 17</b>	Diagrama de bloques del OBC de EnduroSat [9] .....	21
<b>Figura 18</b>	OBC EnduroSat [9] .....	22
<b>Figura 19</b>	Diagrama de bloques de ABACUS [10] .....	23
<b>Figura 20</b>	ABACUS [10] .....	23
<b>Figura 21</b>	Diagrama de bloques de HERCULES [11].....	24
<b>Figura 22</b>	HERCULES [11].....	25
<b>Figura 23</b>	N-ART OBCOMMS V2 [12].....	26
<b>Figura 24</b>	IMT CubeSat OBC [13] .....	27
<b>Figura 25</b>	Diagrama de bloques del SatBus 3C2 [14] .....	28
<b>Figura 26</b>	Vistas superior e inferior del SatBus 3C2 [14] .....	29
<b>Figura 27</b>	Diagrama de bloques del NANOobc Gen2 [15] .....	30
<b>Figura 28</b>	Vista superior e inferior del NANOobc Gen2 [15] .....	30
<b>Figura 29</b>	Diagrama a nivel de componente de la arquitectura de OBC [16].....	41
<b>Figura 30</b>	Ejemplo de arquitectura del sistema de manejo de datos de a bordo [18] .....	42
<b>Figura 31</b>	Estructura general de la OBC [19] .....	43
<b>Figura 32</b>	Buses de datos implementados en CubeSat [20].....	44
<b>Figura 33</b>	Técnicas utilizadas por desarrolladores en los buses de datos I2C, SPI y RS232 [20].....	45
<b>Figura 34</b>	Problemas en órbita reportados en los buses de datos I <sup>2</sup> C, SPI y RS232 [20].....	45
<b>Figura 35</b>	Problemas reportados en el conector PC/104 (n=36) [20] .....	46
<b>Figura 36</b>	Buses de comunicación serial utilizados en CubeSat, según desarrolladores [23] .....	47
<b>Figura 37</b>	Satélite artificial .....	49
<b>Figura 38</b>	Satélite "Sputnik I", primer satélite artificial lanzado por la Unión Soviética.....	49
<b>Figura 39</b>	La Tierra rodeada de miles de satélites artificiales [27].....	50
<b>Figura 40</b>	Estación Espacial Internacional .....	51
<b>Figura 41</b>	Constelación de satélites artificiales .....	52
<b>Figura 42</b>	CubeSat .....	53
<b>Figura 43</b>	Estructuras CubeSat desde 1U hasta 16U [3].....	53
<b>Figura 44</b>	Contenedor P-POD y el resorte de despliegue [31] .....	54
<b>Figura 45</b>	Estructura de un SCMI.....	55
<b>Figura 46</b>	Arquitectura centralizada de un SCMI.....	56
<b>Figura 47</b>	Arquitectura distribuida de un SCMI .....	56
<b>Figura 48</b>	Arquitectura de un SCMI .....	57
<b>Figura 49</b>	Módulo PC/104 de 16 bits [35] .....	59

<b>Figura 50</b>	Nivel de madurez tecnológica (TRL) (adaptado de [36]) .....	60
<b>Figura 51</b>	Celda de memoria de la DRAM .....	67
<b>Figura 52</b>	Celda de memoria de la SRAM.....	68
<b>Figura 53</b>	Celda de memoria de la FRAM.....	69
<b>Figura 54</b>	Estructura cristalina del condensador ferroeléctrico .....	69
<b>Figura 55</b>	Celda de memoria de la MRAM .....	70
<b>Figura 56</b>	Celda de memoria de las NAND Flash y NOR Flash [47] .....	71
<b>Figura 57</b>	Topología "bus" [49].....	74
<b>Figura 58</b>	Palabra de datos del bus UART .....	74
<b>Figura 59</b>	Conector DE-9/DB-9 [49].....	75
<b>Figura 60</b>	Configuración de la conexión RS485 [49].....	75
<b>Figura 61</b>	Bus I2C con dos nodos [49] .....	76
<b>Figura 62</b>	Interfaz Periférico Serial (SPI) con tres esclavos [49] .....	77
<b>Figura 63</b>	Nodos y transceivers del bus CAN [49].....	78
<b>Figura 64</b>	Niveles de tensión recesivos y dominantes del bus CAN [49] .....	78
<b>Figura 65</b>	Diagrama de los cinturones de radiación de Van Allen terrestres [53].....	82
<b>Figura 66</b>	Representación de todos los tipos de radiación que experimenta una nave espacial [52] .....	82
<b>Figura 67</b>	Concepto básico de la TMR .....	88
<b>Figura 68</b>	Repuesto de reserva.....	89
<b>Figura 69</b>	Menú inicial de SPENVIS.....	97
<b>Figura 70</b>	Generador de órbita en SPENVIS.....	97
<b>Figura 71</b>	Parámetros de la órbita en SPENVIS .....	98
<b>Figura 72</b>	Resumen de la misión en SPENVIS .....	98
<b>Figura 73</b>	Generador de órbita en SPENVIS.....	98
<b>Figura 74</b>	Mapa terrestre de la órbita en SPENVIS.....	99
<b>Figura 75</b>	Fuentes y efectos de radiación en SPENVIS .....	100
<b>Figura 76</b>	Parámetros del modelo de radiación atrapada en SPENVIS .....	100
<b>Figura 77</b>	Resultados del modelo de radiación atrapada en SPENVIS .....	101
<b>Figura 78</b>	Mapa terrestre del flujo de protones atrapados en SPENVIS .....	101
<b>Figura 79</b>	Espectro promedio de electrones atrapados en SPENVIS .....	102
<b>Figura 80</b>	Parámetros del modelo para TID en SPENVIS .....	102
<b>Figura 81</b>	Resultados del modelo de TID en SPENVIS .....	103
<b>Figura 82</b>	Dosis en función del espesor de blindaje en SPENVIS .....	103
<b>Figura 83</b>	Parámetros de entrada del dispositivo para la estimación de SEU en SPENVIS .....	104
<b>Figura 84</b>	Resultados de la tasa de SEU a corto plazo en SPENVIS .....	104
<b>Figura 85</b>	Espectro LET de naves espaciales blindadas en SPENVIS .....	105
<b>Figura 86</b>	Diseño conceptual del SCMI, con el mayor nivel de abstracción.....	107
<b>Figura 87</b>	Diseño conceptual del procesador de a bordo .....	108
<b>Figura 88</b>	Diseño conceptual de los dispositivos de almacenamiento.....	109
<b>Figura 89</b>	Diseño conceptual de los puertos de entrada/salida .....	110
<b>Figura 90</b>	Diseño conceptual de los dispositivos externos .....	111
<b>Figura 91</b>	Diseño conceptual del SCMI.....	112
<b>Figura 92</b>	Diagrama del diseño de detalle del procesador de a bordo .....	123
<b>Figura 93</b>	Diagrama del diseño de detalle de los dispositivos de almacenamiento.....	127
<b>Figura 94</b>	Diagrama del diseño de detalle de los dispositivos externos .....	132
<b>Figura 95</b>	Diseño de la PCB del factor de forma PC/104 en EAGLE .....	133
<b>Figura 96</b>	Diagrama de diseño de detalle del SCMI.....	134
<b>Figura 97</b>	Diagrama de conexiones del FSMC.....	137
<b>Figura 98</b>	Diagrama de conexiones para configuración y control (general) .....	138
<b>Figura 99</b>	Diagrama de conexiones para configuración y control (modo terrestre – MSP430) .....	139

<b>Figura 100</b>	Diagrama de conexiones para configuración y control (modo terrestre – STM32).....	139
<b>Figura 101</b>	Diagrama de conexiones para configuración y control (modo de vuelo) .....	140
<b>Figura 102</b>	Diagrama de conexiones del bus SPI interno.....	142
<b>Figura 103</b>	Diagrama de conexiones del bus I2C interno.....	143
<b>Figura 104</b>	Asignación de terminales en el conector PC/104.....	145
<b>Figura 105</b>	Diagrama esquemático del procesador de a bordo en EAGLE .....	146
<b>Figura 106</b>	Diagrama esquemático de los dispositivos de almacenamiento en EAGLE.....	147
<b>Figura 107</b>	Diagrama esquemático de los dispositivos externos en EAGLE .....	147
<b>Figura 108</b>	Diagrama esquemático de los puertos de entrada/salida en EAGLE .....	148
<b>Figura 109</b>	Diseño de la PCB del SCMI en EAGLE – Vista general.....	149
<b>Figura 110</b>	Diseño de la PCB del SCMI en EAGLE – Capa superior (Top layer) .....	149
<b>Figura 111</b>	Diseño de la PCB del SCMI en EAGLE – Primera capa interior .....	150
<b>Figura 112</b>	Diseño de la PCB del SCMI en EAGLE – Segunda capa interior .....	150
<b>Figura 113</b>	Diseño de la PCB del SCMI en EAGLE – Capa inferior (Bottom layer).....	151
<b>Figura 114</b>	Diseño de la PCB del SCMI en EAGLE – Vista de componentes .....	151
<b>Figura 115</b>	Diseño de la PCB del SCMI en EAGLE – Vista de manufactura (cara superior) .....	152
<b>Figura 116</b>	Diseño de la PCB del SCMI en EAGLE – Vista de manufactura (cara inferior) .....	152



## Índice de tablas

---

<b>Tabla 1</b>	Procesador de a bordo de los SCMI comerciales .....	31
<b>Tabla 2</b>	Unidades de almacenamiento de los SCMI comerciales .....	33
<b>Tabla 3</b>	Interfaces eléctricas y periféricos de los SCMI comerciales.....	35
<b>Tabla 4</b>	Interfaces de comunicación serial de los SCMI comerciales .....	37
<b>Tabla 5</b>	Propiedades físicas y eléctricas de los SCMI comerciales.....	38
<b>Tabla 6</b>	Técnicas de mitigación y tolerancia a fallas de los SCMI comerciales .....	39
<b>Tabla 7</b>	Clasificación de los satélites según su masa y sus costos .....	52
<b>Tabla 8</b>	Comparación de las arquitecturas centralizada, distribuida y bus .....	57
<b>Tabla 9</b>	Comparación de los tipos de memorias .....	71
<b>Tabla 10</b>	Efectos de evento único (SEE) (adaptado de [52]) .....	84
<b>Tabla 11</b>	Requisitos a nivel de sistema .....	106
<b>Tabla 12</b>	Especificaciones objetivo generales del SCMI .....	113
<b>Tabla 13</b>	Especificaciones objetivo del procesador de a bordo.....	114
<b>Tabla 14</b>	Especificaciones objetivo de los dispositivos de memoria .....	115
<b>Tabla 15</b>	Puertos de entrada/salida disponibles en el microcontrolador principal [65] .....	116
<b>Tabla 16</b>	Especificaciones objetivo de los dispositivos externos.....	118
<b>Tabla 17</b>	Especificaciones finales del microcontrolador principal a $T_A = 25^\circ C$ [65] .....	120
<b>Tabla 18</b>	Especificaciones finales del microcontrolador supervisor a $T_A = 25^\circ C$ [67].....	121
<b>Tabla 19</b>	Especificaciones finales de la memoria de trabajo .....	124
<b>Tabla 20</b>	Especificaciones finales de la memoria de almacenamiento masivo de datos (Flash) [69].....	124
<b>Tabla 21</b>	Especificaciones finales de la memoria de almacenamiento crítico de datos [67] .....	125
<b>Tabla 22</b>	Especificaciones finales de la memoria de almacenamiento de código [65] .....	126
<b>Tabla 23</b>	Especificaciones finales de los puertos de E/S [65].....	127
<b>Tabla 24</b>	Especificaciones finales del RTC [67] .....	129
<b>Tabla 25</b>	Especificaciones finales del sensor de temperatura [70].....	129
<b>Tabla 26</b>	Especificaciones finales del acelerómetro [71].....	130
<b>Tabla 27</b>	Especificaciones finales del giroscopio [72].....	131
<b>Tabla 28</b>	Especificaciones finales del magnetómetro [73].....	131
<b>Tabla 29</b>	Asignación de terminales del FSMC [65] [68] .....	134
<b>Tabla 30</b>	Asignación de terminales para configuración y control [65] [67] .....	137
<b>Tabla 31</b>	Asignación de terminales del bus SPI interno [65] [67] [69].....	140
<b>Tabla 32</b>	Asignación de terminales del bus I <sup>2</sup> C interno [65] [70] [71] [72] [73].....	143
<b>Tabla 33</b>	Determinación de periféricos disponibles en el SCMI [65].....	144
<b>Tabla 34</b>	Estimación de los costos de construcción de los componentes principales .....	145

## *Acrónimos*

---

<b>ADC</b>	Analog-to-digital converter	<b>FET</b>	Field-effect transistor
<b>ADCS</b>	Attitude determination and control system	<b>FPGA</b>	Field-programmable gate array
<b>ALU</b>	Arithmetic logic unit	<b>FPU</b>	Floating-point unit
<b>ASIC</b>	Application-specific integrated circuit	<b>FRAM</b>	Ferroelectric random-access memory
<b>BIOS</b>	Basic input/output system	<b>FSMC</b>	Flexible static memory controller
<b>C3S</b>	Complex Systems & Small Satellites	<b>GAUSS</b>	Group of Astrodynamics for the Use of Space Systems
<b>CAD</b>	Computer-aided design	<b>GCR</b>	Galactic cosmic rays
<b>CAN</b>	Controller area network	<b>GEO</b>	Geosynchronous equatorial orbit
<b>CCD</b>	Charge-coupled device	<b>GNC</b>	Guidance, navigation & control
<b>CCM</b>	Core coupled memory	<b>GNSS</b>	Global navigation satellite system
<b>CCSDS</b>	Consultative Committee for Space Data Systems	<b>GPIO</b>	General purpose input/output
<b>CDS</b>	CubeSat design specification	<b>GPS</b>	Global positioning system
<b>CISC</b>	Complex instruction set computer	<b>GSSB</b>	GOMspace sensor bus
<b>COTS</b>	Commercial off-the-shelf	<b>I<sup>2</sup>C</b>	Inter-integrated circuit
<b>CPU</b>	Central process unit	<b>IMT</b>	Ingeniería Marketing Tecnología
<b>CRC</b>	Cyclic redundancy check	<b>IMU</b>	Inertial measurement unit
<b>CSP</b>	CubeSat space protocol	<b>IOBC</b>	ISIS on-board computer
<b>C&amp;DH</b>	Command and data handling	<b>ISA</b>	Industry standard architecture
<b>DAC</b>	Digital-to-analog converter	<b>ISIS</b>	Innovative Solutions In Space
<b>DCMI</b>	Digital camera interface	<b>ISS</b>	International space station
<b>DD</b>	Displacement damage	<b>ISSI</b>	Integrated Silicon Solution Inc
<b>DDR</b>	Double data rate	<b>JTAG</b>	Joint test action group
<b>DMA</b>	Direct memory access	<b>LCL</b>	Latch-up current limiter
<b>DRAM</b>	Dynamic random-access memory	<b>LEO</b>	Low Earth orbit
<b>DTMF</b>	Dual-tone multi-frequency	<b>LET</b>	Linear energy transfer
<b>ECC</b>	Error-correcting code	<b>LIESE</b>	Laboratorio de Instrumentación Electrónica de Sistemas Espaciales
<b>EDAC</b>	Error detection and correction	<b>LVDS</b>	Low-voltage differential signaling
<b>EEPROM</b>	Electrically erasable programmable read-only memory	<b>LVTTL</b>	Low-voltage transistor-transistor logic
<b>EPS</b>	Electrical power system	<b>M-LVDS</b>	Multipoint low-voltage differential signaling
<b>ESA</b>	European space agency	<b>MIT</b>	Massachusetts Institute of Technology
<b>ETM</b>	Embedded trace macrocell	<b>MMC</b>	Multimedia card
<b>FDIR</b>	Fault detection, isolation, and recovery		

<b>MOSFET</b>	Metal-oxide-semiconductor field-effect transistor	<b>SEGR</b>	Single-event gate rupture
<b>MPSoC</b>	Multiprocessor system on a chip	<b>SEIDC</b>	Single-event induce dark current
<b>MRAM</b>	Magneto resistive random-access memory	<b>SEL</b>	Single-event latch-up
<b>N2HET</b>	Enhanced high-end timer	<b>SET</b>	Single-event transient
<b>NASA</b>	National Aeronautics and Space Administration	<b>SEU</b>	Single-event upset
<b>OBC</b>	On-board computer	<b>SoC</b>	System on chip
<b>OS</b>	Operating system	<b>SPENVIS</b>	Space Environment Information System
<b>PCB</b>	Printed circuit board	<b>SPI</b>	Serial peripheral interface
<b>PCM</b>	Phase-change memory	<b>SQI</b>	Serial quad interface
<b>PIC</b>	Programmable integrated circuit	<b>SRAM</b>	Static random-access memory
<b>P-POD</b>	Poly picosatellite orbital deployer	<b>SWD</b>	Serial wire debug
<b>PPS</b>	Pulse per second	<b>TCXO</b>	Temperature compensated crystal oscillator
<b>PWM</b>	Pulse width modulation	<b>TI</b>	Texas Instruments
<b>QSPI</b>	Quad serial peripheral interface	<b>TID</b>	Total ionizing dose
<b>RAM</b>	Random-access memory	<b>TMR</b>	Triple modular redundancy
<b>RISC</b>	Reduced instruction set computer	<b>TRL</b>	Technology readiness level
<b>ROM</b>	Read-only memory	<b>TTL</b>	Transistor-transistor logic
<b>RS232</b>	Recommended standard 232	<b>UART</b>	Universal asynchronous receiver-transmitter
<b>RS422</b>	Recommended standard 422	<b>UHF</b>	Ultra high frequency
<b>RS485</b>	Recommended standard 485	<b>UNAM</b>	Universidad Nacional Autónoma de México
<b>RTC</b>	Real-time clock	<b>UNOOSA</b>	United Nations Office for Outer Space Affairs
<b>RTEMS</b>	Real-time executive for multiprocessors systems	<b>USART</b>	Universal synchronous/asynchronous receiver-transmitter
<b>RTOS</b>	Real-time operating system	<b>USB</b>	Universal serial bus
<b>SAA</b>	South Atlantic anomaly	<b>VLSI</b>	Very-large scale integration
<b>SCMI</b>	Sistema de comando y manejo de información		
<b>SD</b>	Secure digital		
<b>SDRAM</b>	Synchronous dynamic random-access memory		
<b>SEB</b>	Single-event burnout		
<b>SEE</b>	Single-event effects		
<b>SEFI</b>	Single-event functional interrupt		

## *Resumen*

---

Tesis profesional presentada por Christo Aldair Lara Tenorio como requisito para obtener el título de ingeniero eléctrico y electrónico, en la Facultad de Ingeniería de la Universidad Nacional Autónoma de México (UNAM), en donde presenta una propuesta de diseño de un sistema de comando y manejo de información (SCMI) para nanosatélites bajo el estándar CubeSat 3U, basado en gran medida en una investigación del estado del arte.

El presente trabajo surge de la necesidad de un SCMI diseñado totalmente dentro de un laboratorio de investigación, con la finalidad de construir un satélite completo. El SCMI tiene como requisito construirse empleando principalmente componentes de bajo costo y alta disponibilidad en el mercado, también llamados componentes COTS, implementando técnicas de mitigación y tolerancia a fallas con las que el sistema alcance niveles de confiabilidad adecuados para desempeñar misiones espaciales en órbitas bajas.

El SCMI desarrollado pretende alcanzar un nivel de madurez tecnológica (TRL) adecuado para obtener un sistema completamente funcional y, con este, incentivar la participación de estudiantes de la facultad en proyectos espaciales.

## *Sección I*

---

# *Planteamiento del proyecto*

# 1 Introducción

---

El éxito de la industria espacial ha impulsado, desde sus comienzos a finales de la década de los 50, un crecimiento en el mercado internacional de sistemas espaciales gracias a las funcionalidades que brindan los satélites artificiales a la humanidad. Los satélites ofrecen un sinfín de conocimiento sobre nuestro planeta y el espacio exterior aportando avances científicos, además de fomentar el desarrollo de nuevas tecnologías y métodos de ingeniería en áreas relacionadas con sistemas satelitales.

Actualmente, se desarrollan satélites en gran parte del mundo gracias a que el aprendizaje obtenido con el paso de los años ha llevado a la comunidad científica a adentrarse en la miniaturización de satélites, posibilitando la integración de satélites convencionales de gran tamaño en sistemas cada vez más pequeños, hasta alcanzar unidades con masas menores a 10 kg conocidas como nanosatélites, las cuales se han convertido en todo un atractivo de inversión, en donde participan universidades y empresas de todo el mundo. El éxito de los nanosatélites radica en que su diminuto tamaño reduce significativamente los costos y tiempos de desarrollo, aumentando las oportunidades de lanzamiento y, de este modo, permitiendo la participación de países en vías de desarrollo en proyectos espaciales.

En el presente trabajo se muestra el desarrollo del hardware de un prototipo de sistema de comando y manejo de información (SCMI) para nanosatélites bajo el estándar CubeSat 3U, diseñado con base en el análisis del estado del arte de los SCMI comerciales y de investigación, dividido en cuatro secciones.

En la *Sección I Planteamiento del proyecto*, se definen los puntos fundamentales del trabajo, que vienen siendo la presentación del problema, la hipótesis planteada, el objetivo general propuesto y los objetivos particulares, así como los alcances, la justificación y la metodología de diseño establecida para el desarrollo del proyecto.

En la *Sección II Estado del arte*, se presentan los resultados de la investigación del estado del arte, comenzando con las áreas de oportunidad del desarrollo de SCMI bajo el estándar CubeSat 3U, seguido de la descripción de las características de los SCMI investigados, dividida en dos tipos: los SCMI comerciales y los SCMI de investigación.

En la *Sección III Marco teórico*, se definen los conocimientos necesarios para entender el trabajo, prestando principal atención en los conceptos identificados en el estado del arte.

En la *Sección IV Diseño y Construcción*, se realiza el diseño del prototipo de hardware del SCMI dividido en tres pasos: el diseño conceptual, el diseño de detalle y la construcción, tomando como base los resultados obtenidos del estado del arte y los conocimientos adquiridos del marco teórico.

Al final del documento se presentan las conclusiones obtenidas a partir de los resultados observados en las etapas de diseño, definiendo el trabajo a futuro del presenta trabajo.

## 1.1 Presentación del problema

Los sistemas de un satélite son diseñados principalmente tomando como base la integración de tecnologías probadas y catalogadas como componentes con clasificación espacial, elevando los costos hasta un punto en donde disminuye la posibilidad de que universidades y empresas con bajo presupuesto puedan adquirir estos sistemas, minimizando su participación en misiones espaciales y los aportes que estas puedan dar a la industria.

El buen desempeño de los satélites bajo el estándar CubeSat incrementó su popularidad e incorporación en un mayor número de misiones en orbitas bajas terrestres<sup>1</sup> (LEO, por sus siglas en inglés) de corta duración. En el caso del SCMI, existen varios fabricantes que ofrecen sus diseños como soluciones rápidas, pero a unos costos elevados, en donde algunos de ellos presentan compatibilidad únicamente con subsistemas del mismo fabricante, dificultando las etapas de integración. Además, el éxito del CubeSat lo ha llevado a ser propuesto como una alternativa para misiones de cada vez mayor duración, por lo que surge la necesidad de que el SCMI sea más

---

<sup>1</sup> Órbita terrestre localizada entre la atmósfera y el cinturón de radiación de Van Allen (típicamente entre los 200 y 2000 km de altura sobre la superficie terrestre).

resistente al impacto de la radiación espacial. Para esto, actualmente existe una gran variedad de componentes con calificación espacial y buena resistencia a la radiación que pueden usarse en el SCMI, la problemática consiste en que al implementar estas tecnologías los costos aumentan significativamente comparándolos con el uso de componentes comercialmente disponibles, conocidos como COTS<sup>2</sup> (del inglés *Commercial Off-The-Shelf*). Sin embargo, los componentes COTS no presentan niveles de confiabilidad<sup>3</sup> suficientes para integrarse en sistemas espaciales, por lo que es necesario implementar técnicas de mitigación y tolerancia a fallas para garantizar que el sistema desempeñe correctamente sus tareas asignadas. [1]

## 1.2 Hipótesis

El uso mayoritario de componentes COTS, en conjunto con técnicas de mitigación y tolerancia a fallas en el diseño de un SCMI para nanosatélites bajo el estándar CubeSat 3U, es una alternativa significativamente más económica que puede presentar niveles de confiabilidad que permitan realizar misiones espaciales en LEO, con valores de confiabilidad bien determinados.

## 1.3 Objetivo

Desarrollar un SCMI para nanosatélites bajo el estándar CubeSat 3U, empleando principalmente componentes COTS e implementando técnicas de mitigación y tolerancia a fallas, presentando niveles de confiabilidad suficientes para realizar misiones en LEO.

### 1.3.1 Objetivos particulares

- ✓ Diseñar un SCMI para nanosatélites bajo el estándar CubeSat 3U que cumpla con las especificaciones mecánicas y eléctricas establecidas en la especificación de diseño CubeSat (CDS, por sus siglas en inglés).
- ✓ Diseñar e integrar el hardware de un SCMI para nanosatélites bajo el estándar CubeSat 3U empleando principalmente componentes COTS y tomando como base el estado del arte de los SCMI.
- ✓ Diseñar un SCMI para nanosatélites bajo el estándar CubeSat 3U implementando técnicas de mitigación y tolerancia a fallas, que proporcionen niveles de confiabilidad adecuados para cumplir misiones en LEO.

## 1.4 Alcances

Los alcances del presente trabajo contemplan el diseño electrónico y construcción de un prototipo de SCMI para nanosatélites bajo el estándar CubeSat 3U, utilizando principalmente componentes COTS e implementando técnicas de mitigación y tolerancia a fallas para que posea niveles de confiabilidad adecuados para desempeñar misiones en LEO.

## 1.5 Justificación

La tendencia actual de diseñar nanosatélites bajo el estándar CubeSat ha propiciado que un mayor número de fabricantes ofrezcan sus sistemas como soluciones rápidas, sin embargo, se están desarrollando métodos para diseñar con base en componentes COTS con el objetivo de reducir los costos y aumentar la confiabilidad de los sistemas para desempeñar misiones en LEO.

Actualmente, en el Laboratorio de Instrumentación Electrónica de Sistemas Espaciales (LIESE) de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México (UNAM) se fomenta el desarrollo de nanosatélites bajo el estándar CubeSat integrando tarjetas de desarrollo de microcontroladores como núcleo de computadora de a bordo (OBC, por sus siglas en inglés), por lo que se requiere diseñar un SCMI que pueda integrar los otros sistemas pertenecientes a un CubeSat, con el propósito de desarrollar un satélite funcional completamente diseñado en el laboratorio.

---

<sup>2</sup> Componentes que no poseen algún tipo de clasificación restrictiva, fácilmente localizables en el mercado, para uso común.

<sup>3</sup> Propiedad de un sistema para conservar, en un límite de tiempo establecido, los valores de todos los parámetros que caracterizan su capacidad para desempeñar correctamente sus funciones.

## 2 Metodología de diseño

---

La metodología de diseño es el conjunto de procedimientos y técnicas de rigor científico aplicadas sistemáticamente durante un proceso de investigación para alcanzar un resultado teóricamente válido. En otras palabras, es el soporte conceptual que rige la manera en que se aplican los procedimientos de un proyecto. Definir una buena metodología es fundamental para desarrollar un proyecto cumpliendo con cada uno de los aspectos necesarios para obtener el resultado deseado. En este capítulo se presenta a detalle la metodología aplicada para desarrollar un SCMI para nanosatélites bajo el estándar CubeSat 3U.

Para este proyecto, se siguió el modelo de proceso iterativo<sup>4</sup> de diseño (el proceso completo con sus iteraciones se presenta en la *Figura 1*) de ingeniería para misiones espaciales establecido en [2] adaptándolo a los objetivos y alcances planteados en un principio. La metodología seguida se dividió en 18 pasos organizados en 5 etapas, los cuales se describen a continuación:

### **ETAPA I** *Planteamiento del proyecto*

**Paso 1** **Identificación de la necesidad**

Consiste en la definición de un objetivo general y su justificación.

**Paso 2** **Investigación preliminar**

Se desarrolla el estudio del estado del arte y la investigación del marco teórico.

**Paso 3** **Definición del problema**

Se realiza la presentación del problema y el planteamiento de una hipótesis.

**Paso 4** **Definición de los objetivos y restricciones**

Consiste en el planteamiento más certero del objetivo (objetivos específicos) y los alcances del proyecto.

**Paso 5** **Estimación de los tiempos de desarrollo**

Se presenta una estimación de los tiempos en los que se desarrollará cada etapa.

### **ETAPA II** *Diseño conceptual*

**Paso 6** **Definición del concepto de diseño**

Se define el planteamiento de la arquitectura de diseño, los manejadores del sistema<sup>5</sup>, los requisitos críticos<sup>6</sup> y el concepto de la misión.

**Paso 7** **Definición de bloques funcionales**

Se realiza la división conceptual del sistema en bloques funcionales para agrupar los requisitos y especificaciones en sistemas más pequeños con funciones específicas.

**Paso 8** **Planteamiento de los requisitos de sistema**

Se plantean los requisitos a nivel bloque funcional.

**Paso 9** **Diseño conceptual**

Se desarrolla el diseño conceptual de cada bloque funcional.

### **ETAPA III** *Diseño de detalle*

**Paso 10** **Planteamiento de las especificaciones de diseño**

Consiste en el planteamiento de las especificaciones de diseño de detalle.

---

<sup>4</sup> Es un proceso cíclico en donde se repiten los pasos el número de veces necesarias para alcanzar un objetivo partiendo de los resultados obtenidos en la iteración anterior.

<sup>5</sup> Son los requisitos que afectan de manera general el diseño.

<sup>6</sup> Son los requisitos que tienen mayor impacto en los costos, riesgos y tiempos de desarrollo.



**Paso 11 Diseño de detalle**

Se realiza el desarrollo del diseño de detalle a nivel bloque funcional.

**Paso 12 Integración**

Se diseña el sistema y la interfaz necesaria para integrar cada bloque funcional en un único sistema.

**Paso 13 Estimación de las necesidades cuantitativas**

Se hace la definición de los componentes del sistema y la estimación de los costos requeridos para construirlo.

**Paso 14 Análisis y evaluación**

Consiste en el análisis del sistema y la evaluación de su comportamiento partiendo de los requisitos y especificaciones.

**ETAPA IV Construcción**

**Paso 15 Diseño Asistido por Computadora (CAD)**

Consiste en el diseño asistido por computadora (CAD, por sus siglas en inglés) del sistema.

**Paso 16 Construcción física e integración**

Consiste en la construcción e integración física del sistema.

**ETAPA V Documentación**

**Paso 17 Análisis y evaluación de desempeño**

Se realiza el análisis, las pruebas y la evaluación de desempeño del sistema en funcionamiento.

**Paso 18 Presentación de resultados y conclusiones**

Consiste en la discusión de los resultados obtenidos y la presentación de las conclusiones.

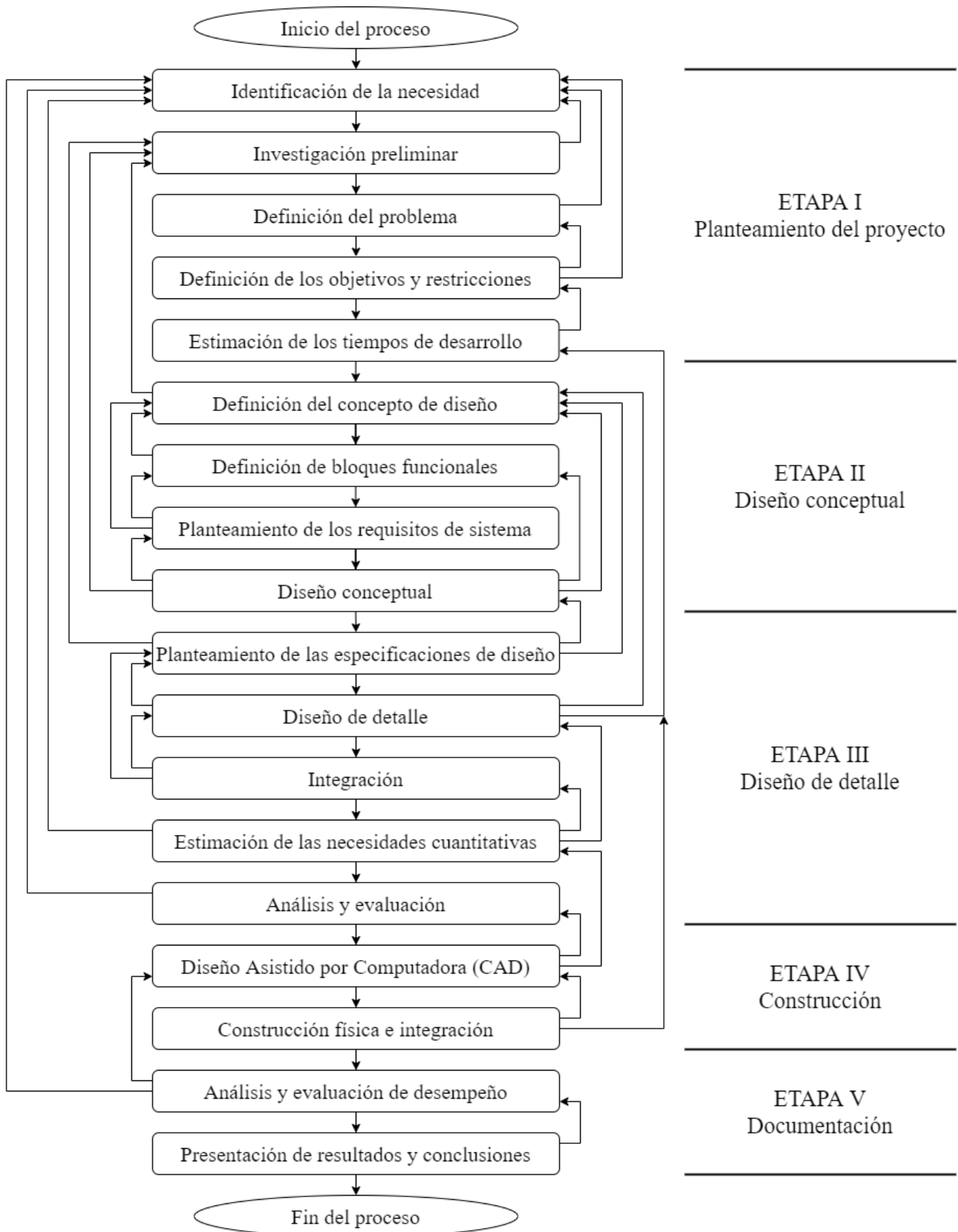


Figura 1. Metodología de diseño

## *Sección II*

---

# *Estado del arte*

### 3 Estado del arte de los SCMI en nanosatélites bajo el estándar CubeSat

---

El estado del arte se define como una compilación de los resultados de las investigaciones que se han realizado sobre un tema en particular, con la finalidad de establecer lo que se ha hecho recientemente o el estado actual en el que se encuentra el tema seleccionado. En el presente trabajo, el estado del arte se orienta a la investigación de dos tipos de SCMI para nanosatélites bajo el estándar CubeSat: los SCMI comerciales, encontrados en diferentes empresas dedicadas a la industria espacial, en donde se presta especial atención en sus características estructuradas en módulos, tales como el procesador de a bordo, las unidades de almacenamiento, las interfaces eléctricas y periféricos, las interfaces de comunicación serial, las propiedades físicas y eléctricas y las técnicas de mitigación y tolerancia a fallas implementadas en cada sistema; y los SCMI de investigación, siendo aquellos sistemas identificados en artículos, publicaciones, conferencias y sitios de investigación, en donde se presta principal interés en las técnicas de diseño, tales como las arquitecturas propuestas, las tecnologías seleccionadas, los resultados alcanzados y las conclusiones obtenidas de investigadores y equipos universitarios. El objetivo es brindar un acercamiento a lo que se ofrece en el mercado y lo que se está desarrollando actualmente en cuanto al diseño de SCMI, cuya investigación sirva como punto de partida para el diseño de un sistema propio, obteniendo de esta manera características de operación y técnicas de diseño reales que planteen los requisitos del proyecto.

Este capítulo presenta los resultados de la investigación del estado del arte, dividido en tres partes: las áreas de oportunidad, que justifican el diseño de un SCMI para nanosatélites bajo el estándar CubeSat 3U; los SCMI comerciales, en donde se presentan las características identificadas en diferentes sistemas comercialmente disponibles, mostrando los resultados en tablas comparativas en donde se exponen las características fundamentales agrupadas en módulos con tareas específicas para facilitar el análisis y la comparación, y así llegar a conclusiones más certeras; y los SCMI de investigación, en donde se describen el funcionamiento general, sus características y las estrategias de diseño utilizadas, además de publicaciones cuyos resultados fundamenten la toma de decisiones para el desarrollo del proyecto. Es importante mencionar que dentro del estado del arte solo se presentan las características y los resultados obtenidos de cada SCMI. Las definiciones y conceptos necesarios que involucra el estado del arte se presentan en la *Sección III Marco teórico*.

#### 3.1 Áreas de oportunidad

El aumento en el desarrollo de nanosatélites se ha dado gracias a la tendencia de la comunidad científica por diseñarlos para desempeñar misiones espaciales, tendencia impulsada por los bajos costos y tiempos de desarrollo que requieren, además del alto índice de misiones exitosas y los resultados obtenidos en sus misiones. Desde 1998 hasta enero de 2020 se han lanzado 1307 nanosatélites, de los cuales 1200 han sido construidos bajo el estándar CubeSat [3]. En la *Figura 2* se puede apreciar el número de nanosatélites lanzados exitosamente por año desde 1998, además de una predicción para los próximos cuatro años incluyendo los que ya han sido anunciados, en donde se puede observar claramente el aumento que los nanosatélites han presentado en los últimos años y las grandes expectativas que se tienen para los siguientes.

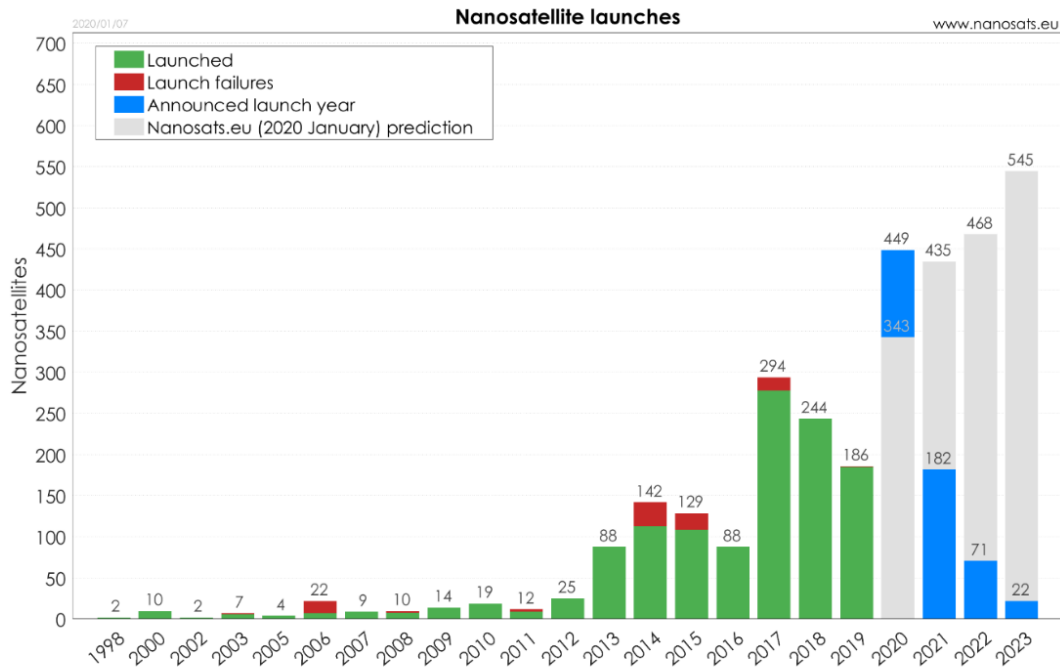


Figura 2. Nanosatélites lanzados y la predicción de los próximos 4 años [3]

Dentro de los satélites pequeños, los CubeSat son el tipo de nanosatélite más destacado. Como se puede observar en la *Figura 3*, los nanosatélites bajo el estándar CubeSat presentan un número mayor de lanzamientos con respecto a otros nanosatélites. En la *Figura 4* igualmente se aprecia esta tendencia en los tipos de nanosatélites desarrollados, en donde únicamente alrededor del 5% no son CubeSat. Además, en ambas figuras se observa claramente que el tipo de CubeSat favorito para desarrollar es el 3U, seguido del 1U, por lo que diseñar un SCMI para nanosatélites bajo el estándar CubeSat 3U representa una gran área de oportunidad desde hace 5 años aproximadamente y todo indica que esta preferencia continuará en los próximos 3 años como mínimo.

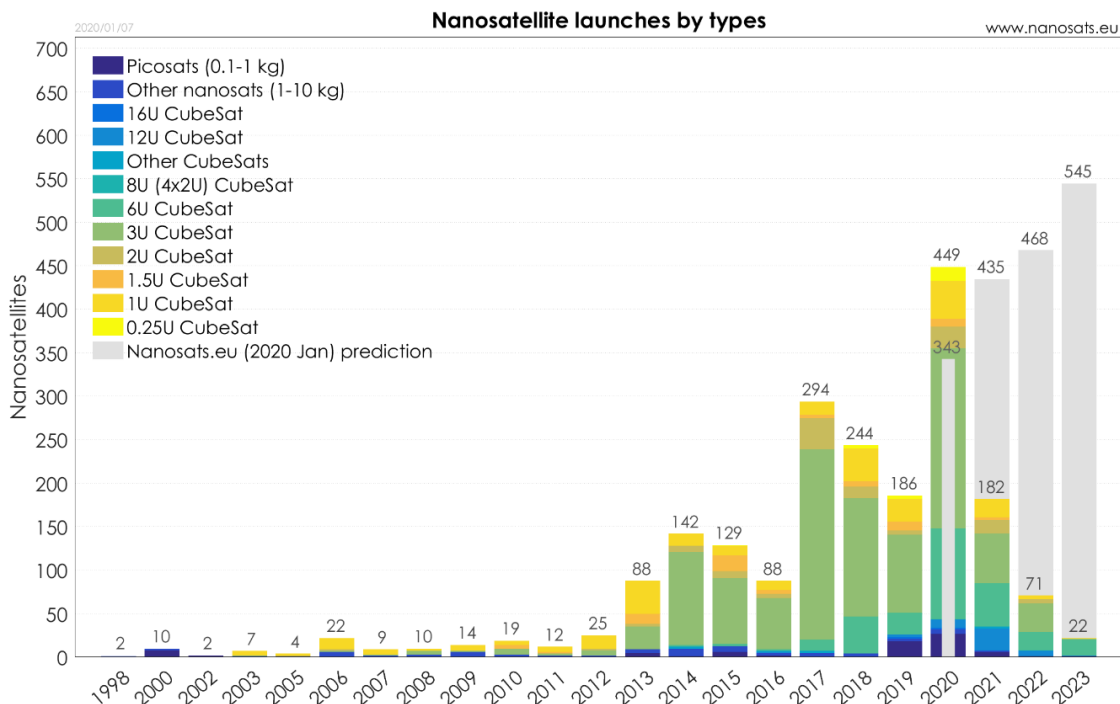


Figura 3. Tipos de nanosatélites lanzados por año [3]

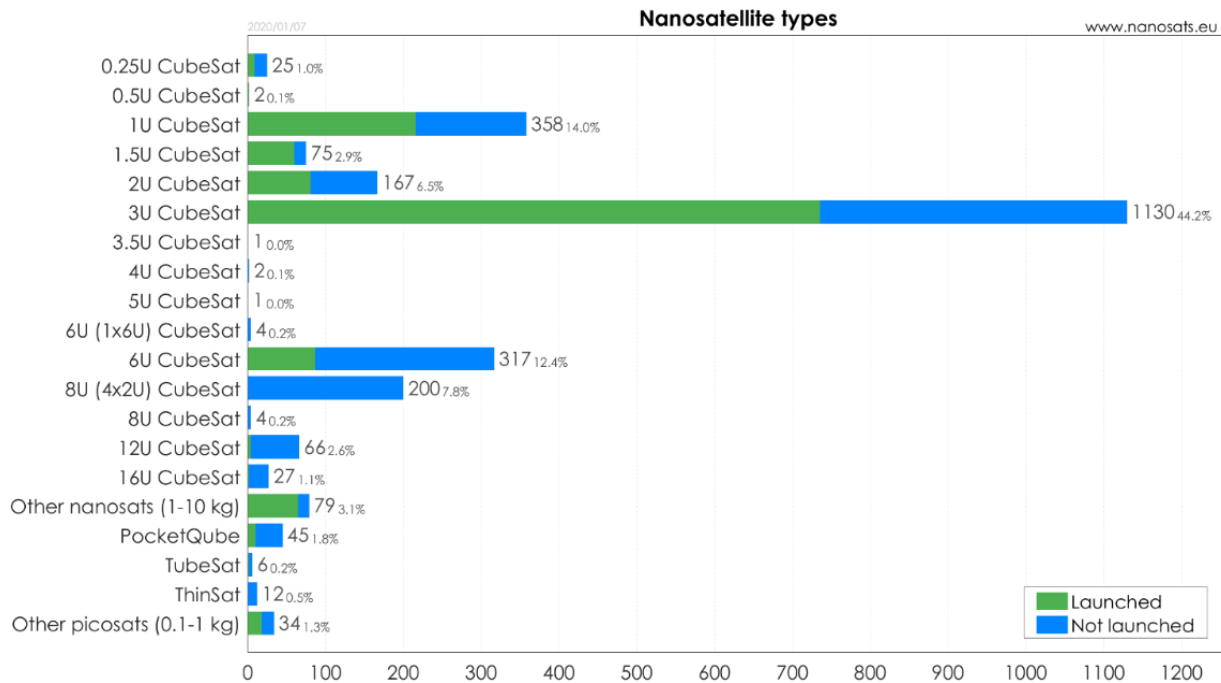


Figura 4. Tipos de nanosatélites desarrollados [3]

En adición a lo anterior, los nanosatélites han aumentado su popularidad año tras año gracias a instituciones en donde, como se indica en la *Figura 5*, destacan las empresas y universidades, las cuales realizan una gran cantidad de proyectos espaciales al año empleando nanosatélites (principalmente el estándar CubeSat). En la *Figura 6* se aprecia más claramente el dominio que tienen las empresas y universidades en el desarrollo de nanosatélites frente a otras instituciones.

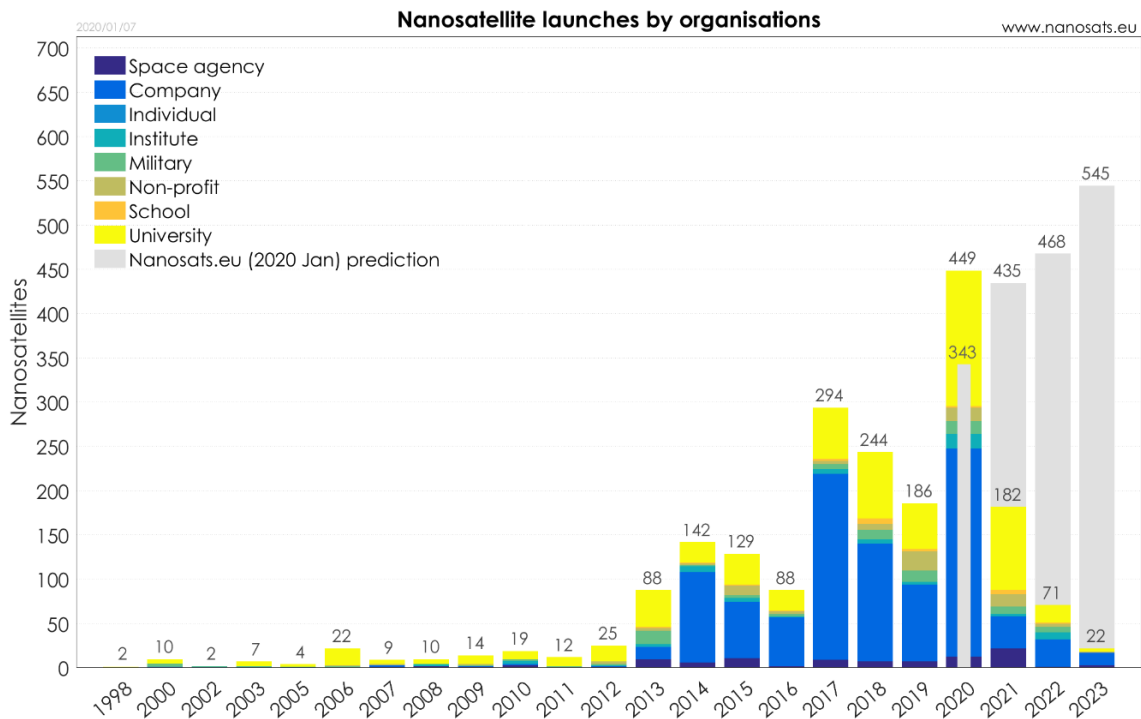


Figura 5. Nanosatélites lanzados por institución al año [3]

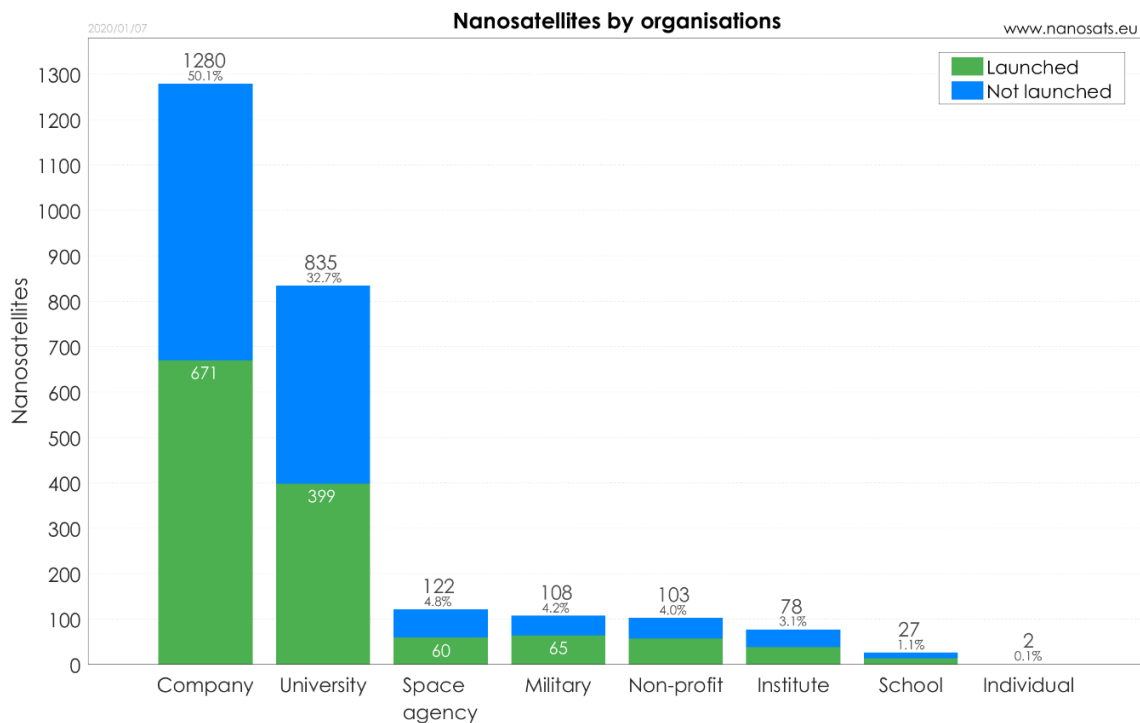


Figura 6. Nanosatélites desarrollados por institución [3]

Como parte del crecimiento de la industria espacial y gracias a las oportunidades que ofrecen los nanosatélites, desde 1990 se ha presentado un surgimiento elevado de nuevas empresas dedicadas al desarrollo de nanosatélites (véase la Figura 7), lo cual demuestra las grandes oportunidades que se presentan al adentrarse en esta industria. Otro factor que es importante mencionar para demostrar la popularidad que han alcanzado los nanosatélites se puede observar en la Figura 8, en donde se presenta el número de nanosatélites lanzados con los que cuenta cada país. El desarrollo de nanosatélites lo encabeza claramente Estados Unidos y es fácil concluir que México se encuentra demasiado atrás en esta área, por lo que el invertir en la industria espacial puede fomentar la participación de más instituciones nacionales.

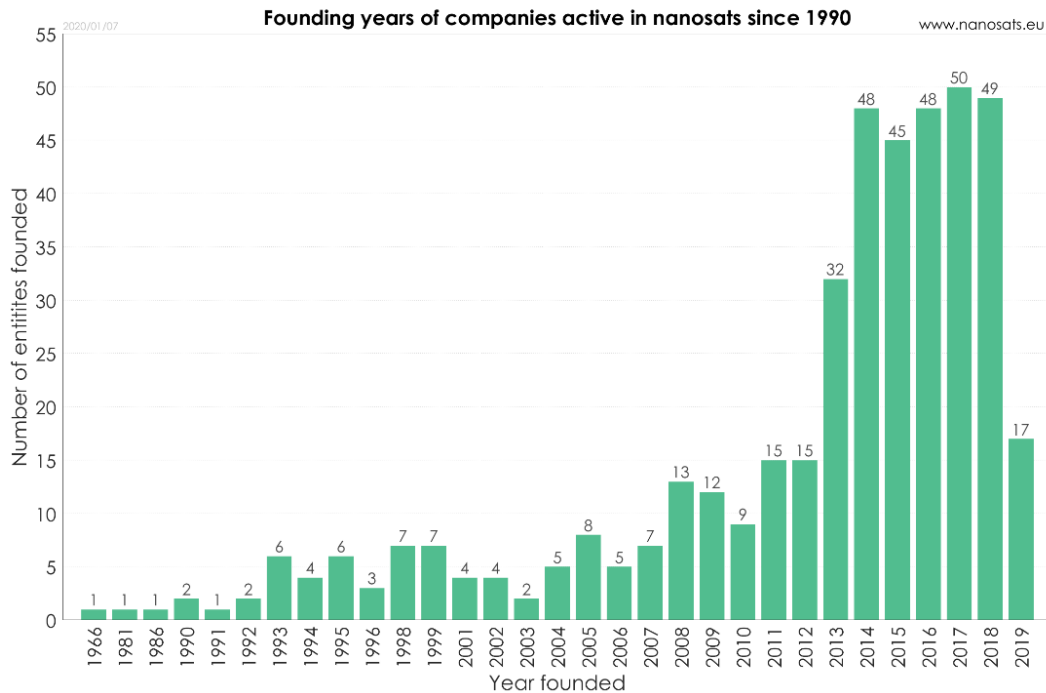
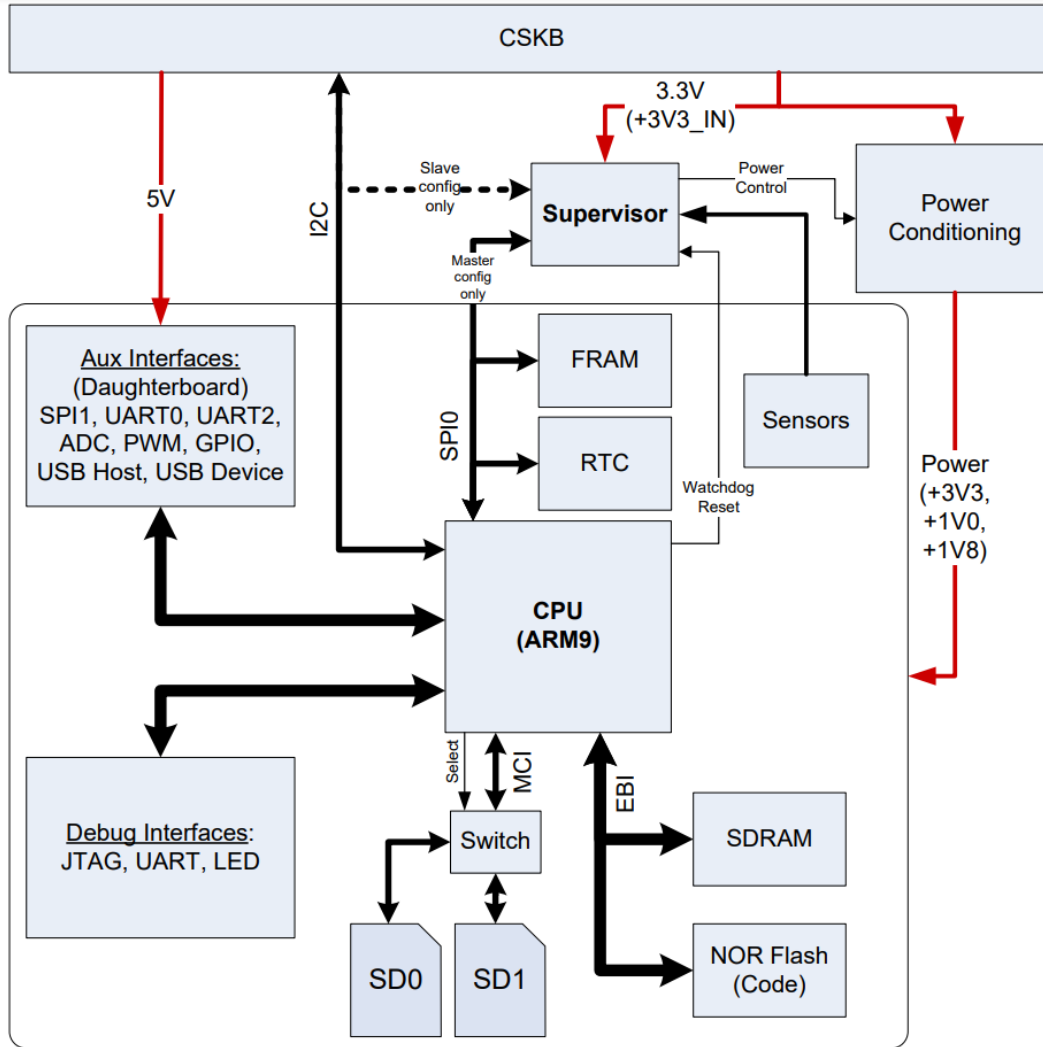


Figura 7. Empresas activas de nanosatélites fundadas por año [3]





El diagrama de bloques del IOBC se presenta en la *Figura 9*.



*Figura 9. Diagrama de bloques del IOBC [4]*

### **Unidades de almacenamiento**

La CPU opera con 32 MB de memoria dinámica de acceso aleatorio síncrona (SDRAM, por sus siglas en inglés) como memoria de trabajo. El SCMI cuenta con dos ranuras para tarjetas digitales seguras (SD, por sus siglas en inglés) de 2 hasta 32 GB para el almacenamiento masivo de datos, en donde ISIS emplea tarjetas SD de alta calidad industrial con memorias Flash. La computadora emplea 256 kB de memoria ferroeléctrica de acceso aleatorio (FRAM, por sus siglas en inglés) destinados al almacenamiento crítico de datos. Para el almacenamiento de código, la IOBC posee 1 MB de memoria NOR Flash, en donde el código es copiado a la memoria volátil para ejecutarse más rápido durante el arranque del sistema.

### **Interfaces eléctricas y periféricos**

El SCMI contiene 27 terminales de entrada/salida de propósito general (GPIO, por sus siglas en inglés), un convertidor analógico-digital (ADC, por sus siglas en inglés) de 8 canales y 8/10 bits, modulación por ancho de pulso (PWM, por sus siglas en inglés) de 6 canales. Para controlar el tiempo, el SCMI posee dos relojes de tiempo real (RTC, por sus siglas en inglés), uno interno a la CPU y otro externo capaz de compensar la alteración del reloj por variaciones de temperatura ( $\pm 3.5$  ppm) y  $\pm 5$  ppm por más de 10 años de uso. Además, algunas de las terminales GPIO pueden ser utilizadas de manera alterna como una interfaz para un sensor de imagen y sensores de temperatura, corriente y tensión embebidos con protección a sobre corriente.

### Interfaces de comunicación serial

Las interfaces de comunicación serial que forman parte del SCMI son un bus de circuito inter-integrado (I<sup>2</sup>C, por sus siglas en inglés) en modo maestro y esclavo con una tasa de transmisión máxima de 400 kbit/s; dos buses de interfaz periférico serial (SPI, por sus siglas en inglés), de las cuales una está destinada únicamente para la comunicación interna de la CPU con el RTC, la FRAM y el supervisor, mientras que la otra cuenta con los modos maestro y esclavo con capacidad de hasta 8 esclavos y una tasa de transmisión máxima de 10 Mbit/s destinada para la comunicación externa; dos buses UART, de los cuales uno puede ser utilizado con lógica transistor-transistor<sup>8</sup> (TTL, por sus siglas en inglés) de 3.3 V o el estándar recomendado 232 (RS232, por sus siglas en inglés) dependiendo de la configuración de hardware solicitada en la compra, mientras que el otro bus UART puede configurarse a través de software para soportar el RS232 con control de flujo o los buses diferenciales full-duplex RS422/485; y también contiene el bus serial universal (USB, por sus siglas en inglés) tanto como anfitrión (host) y dispositivo (device) en dos buses separados de hasta 12 Mbit/s.

### Propiedades físicas y eléctricas

Como se observa en la *Figura 10*, la IOBC cuenta con el factor de forma PC/104 y posee la interfaz necesaria para integrarse a daughterboards<sup>9</sup>, también ofrecidas por ISIS, que aumentan las capacidades de la OBC. El sistema posee las dimensiones de 96 x 90 x 12.4 mm y una masa de 106 g incluyendo una daughterboard, opera en un rango de temperatura de -25 a +65 °C y a una temperatura de almacenamiento de -40 a +80 °C con una tensión de alimentación de 3.3 V. El sistema consume un promedio de 380 mW de energía.

### Técnicas de mitigación y tolerancia a fallas

Como métodos de tolerancia a fallas, el SCMI posee un temporizador watchdog externo y un circuito integrado programable (PIC, por sus siglas en inglés) como supervisor (véase la *Figura 9*) de energía encargado de controlar la distribución de energía en el sistema apagando y encendiendo individualmente la CPU y el RTC dependiendo del comportamiento del watchdog, además de que cuenta con sensores de temperatura, tensión y corriente que pueden ser leídos directamente desde la CPU con el bus SPI cuando esta se encuentra configurada como maestro o desde I<sup>2</sup>C cuando está configurada como esclavo. El SCMI cuenta con dos ranuras para tarjetas SD como redundancia, además de que mientras una tarjeta SD se encuentra en uso, la otra permanece apagada para mantenerla a salvo de perturbaciones de evento único (SEU, por sus siglas en inglés) y obtener una mayor eficiencia de energía. Sumado a lo anterior, la memoria FRAM empleada para el almacenamiento crítico de datos no es susceptible a SEU debidos a la radiación.



Figura 10. IOBC, vistas superior, inferior e integrada a una daughterboard [4]

### 3.2.2 NanoMind A3200 [5]

**Fabricante:** GOMspace.

**Precio:** €6,000.00 (USD 6,783.06).

**Disponible en:** <https://gomspace.com/shop/subsystems/command-and-data-handling/nanomind-a3200.aspx>

<sup>8</sup> Familia lógica en donde los elementos de entrada y salida de un dispositivo son transistores bipolares. Los circuitos TTL operan de 0 a 0.8 V como cero lógico y entre 2 y 5 V como uno lógico.

<sup>9</sup> Circuito impreso que puede ser insertado en una tarjeta madre (motherboard) de una computadora para añadir funcionalidades al sistema a través de un bus de expansión.

### Procesador de a bordo

La CPU de la NanoMind A3200 está basado en el microcontrolador Atmel AT32UC3C de 32 bits AVR tipo RISC con arquitectura Harvard que opera con un rango de frecuencia de 8 a 64 MHz y cuenta con el soporte de una unidad de punto flotante (FPU, por sus siglas en inglés) operando bajo el estándar IEEE 574 para aplicaciones como sistema de control y determinación de posición (ADCS, por sus siglas en inglés). La programación del SCMI se realiza mediante el bus JTAG, mientras que su depuración a través del bus transmisor-receptor síncrono/asíncrono universal (USART, por sus siglas en inglés) y opera con FreeRTOS. El sistema, como todo sistema comercial, cuenta con TRL 9.

El diagrama de bloques del NanoMind A3200 se presenta en la *Figura 11*.

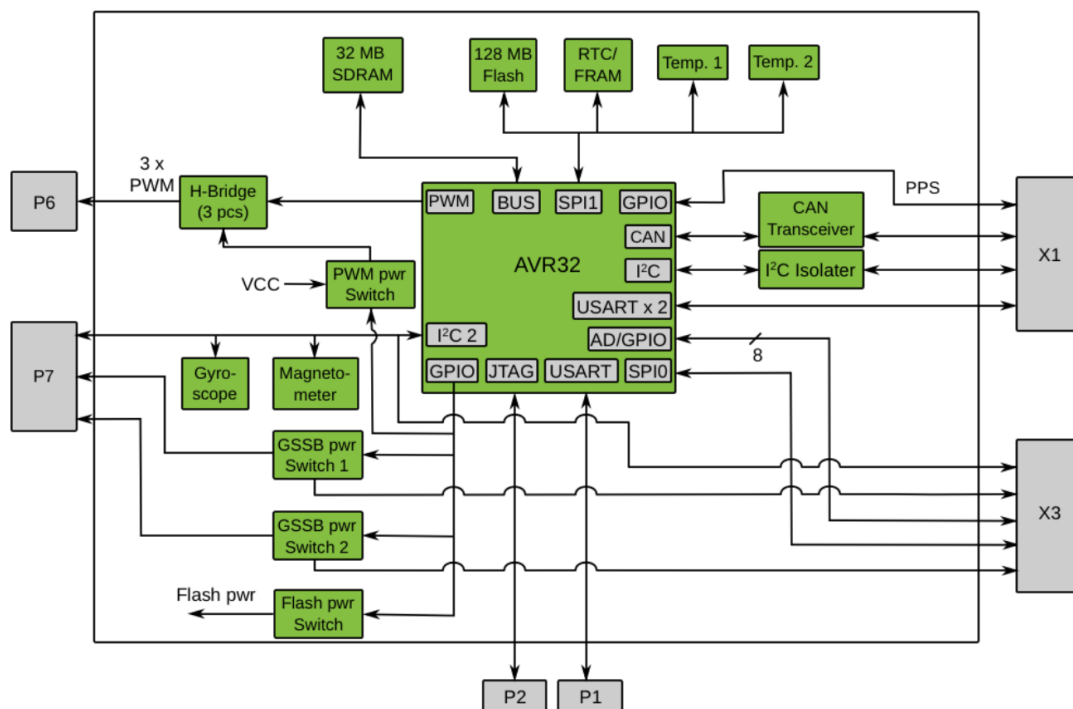


Figura 11. Diagrama de bloques del NanoMind A3200 [5]

### Unidades de almacenamiento

La NanoMind A3200 posee 32 MB de SDRAM como memoria volátil, 128 MB de memoria NOR Flash (divididos en dos memorias de 64 MB) para el almacenamiento masivo de datos, 32 kB de FRAM por parte de un procesador FM33256B de Cypress integrado al SCMI para proporcionar almacenamiento crítico de datos y 512 kB de memoria Flash para el almacenamiento de código.

### Interfaces eléctricas y periféricos

El SCMI posee un ADC de 8 canales que también pueden ser ocupados como GPIO, un RTC por parte del procesador FM33256B, un sistema de estabilización de posición integrado por tres salidas PWM bidireccionales con medición de corriente, un magnetómetro de 3 ejes HMC5843 de Honeywell y un giroscopio de 3 ejes MPU-3300 de InvenSense. Además, el sistema cuenta con una señal de pulso por segundo (PPS, por sus siglas en inglés), cuya terminal también puede configurarse como GPIO y dos sensores de temperatura integrados cerca de la CPU.

### Interfaces de comunicación serial

El SCMI cuenta con el bus controlador de red de área (CAN, por sus siglas en inglés) con una tasa de transmisión máxima de 1 Mbit/s empleando el transceiver SN65HVD230; dos buses I²C configurables como maestro y esclavo y permiten el arbitraje para emplear un bus multimaestro y un tercero disponible como conector que puede ser utilizado para el magnetómetro, giroscopio, un sensor solar y usado como bus de sensores de

GOMspace (GSSB, por sus siglas en inglés); una interfaz UART para consola de laboratorio; dos buses USART y dos buses SPI, de los cuales uno es interno para comunicarse con los sensores de temperatura, la memoria Flash y el RTC y uno externo con capacidad para tres esclavos.

### ***Propiedades físicas y eléctricas***

La NanoMind A3200, vista en la *Figura 12*, está diseñada con un factor de forma tipo daughterboard más compacto que el estándar empleado en el CubeSat, sin embargo, es necesario también adquirir una motherboard disponible en GOMspace, conocidas como NanoDocks, las cuales poseen el factor de forma PC/104; cuenta con dimensiones de 65 x 40 x 7.1 mm y una masa de 24 g; tanto su temperatura de operación como de almacenamiento se encuentran en el rango de -30 a +85 °C y su consumo de energía promedio es de 170 mW con un máximo de 900 mW con una tensión de alimentación de 3.3 V.

### ***Técnicas de mitigación y tolerancia a fallas***

El SCMI cuenta con un temporizador watchdog integrado en el procesador FM33256B.



*Figura 12. NanoMind A3200, vistas exterior, interior e integrada en una NanoDock [5]*

## **3.2.3 On-Board Computer (C3S) [6]**

**Fabricante:** Complex Systems & Small Satellites (C3S).

**Precio:** €9,500.00 - €14,500.00 (USD 10,128.09 – USD 15,458.67).

**Disponible en:** <https://www.c3s.hu/portfolio-item/on-board-computer/>

### ***Procesador de a bordo***

La CPU del sistema ofrecido por C3S está integrada por un procesador ARM Cortex-M7 de 32 bits tipo RISC con arquitectura Harvard que puede operar hasta 300 MHz; cuenta con dos interfaces de programación y depuración, el bus USB y el bus de depuración de cable serial (SWD, por sus siglas en inglés) como interfaz de depuración estándar. El SCMI es compatible con FreeRTOS y, al igual que todo sistema comercial, cuenta con TRL 9.

### ***Unidades de almacenamiento***

El SCMI cuenta con 384 kB de memoria de acceso aleatorio (RAM, por sus siglas en inglés) interna, 16 GB de tarjeta multimedia de memoria (MMC, por sus siglas en inglés) embebida y 16 MB de memoria magnética de acceso aleatorio (MRAM, por sus siglas en inglés) con una tarjeta daughterboard y 2 MB de memoria Flash interna para el almacenamiento de código.

### ***Interfaces de comunicación serial***

El SCMI cuenta principalmente con dos buses CAN y cuatro sistemas de señal diferencial multipunto de baja tensión (M-LVDS, por sus siglas en inglés).

### ***Propiedades físicas y eléctricas***

La OBC de C3S no cuenta con el factor de forma PC/104, sin embargo, está diseñada para poder integrarle daughterboards que pueden ser adquiridas mediante el mismo proveedor. Sin una daughterboard cuenta con unas dimensiones de 92 x 89.5 x 12.3 mm y una masa de 65 g, con daughterboard sus dimensiones cambian a 92 x 89.5 x 13.6 mm y su masa a 100 g. El sistema consume un promedio de 460 mW.

### Técnicas de mitigación y tolerancia a fallas

El SCMI cuenta con un sistema watchdog multinivel interno y externo, además de que posee un diseño tolerante a fallas de un solo punto y posee un par de buses CAN y dos pares de M-LVDS como métodos de redundancia en frío.

#### 3.2.4 KRYTEN-M3 [7]

**Fabricante:** AAC Clyde Space.

**Precio:** USD 8,500.00 / USD 23,100.00 con sistema de posicionamiento global (GPS, por sus siglas en inglés).

**Disponible en:** <https://www.aac-clyde.space/satellite-bits/command-data-handling>

El sistema de Clyde Space es una computadora de mayordomo (housekeeping) para misiones espaciales bajo el estándar CubeSat, la cual puede ser adquirida únicamente como el SCMI o añadiéndole un GPS.

#### Procesador de a bordo

La CPU está conformada por el sistema en chip (SoC, por sus siglas en inglés) SmartFusion 2 de Microsemi, el cual incluye un arreglo de compuertas programables en campo (FPGA, por sus siglas en inglés) y un procesador ARM Cortex-M3 de 32 bits tipo RISC y arquitectura Harvard que opera a una frecuencia de 50 MHz. Utiliza el bus JTAG con soporte de macrocelda de seguimiento embebida<sup>10</sup> (ETM, por sus siglas en inglés) para la programación y depuración y un bus UART con lógica transistor-transistor de baja tensión<sup>11</sup> (LVTTL, por sus siglas en inglés) para depuración. El sistema es compatible con FreeRTOS y sistema ejecutivo en tiempo real para sistemas multiprocesador (RTEMS, por sus siglas en inglés) y cuenta con un TLR 9.

El diagrama de bloques del KRYTEN-M3 se presenta en la *Figura 13*.

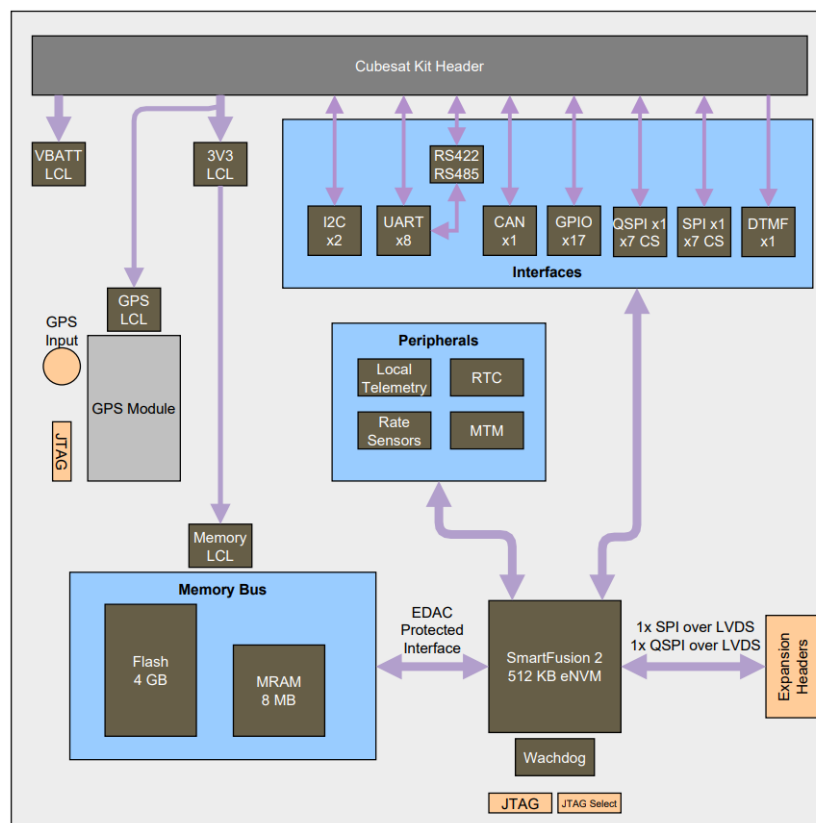


Figura 13. Diagrama de bloques del KRYTEN-M3 [7]

<sup>10</sup> Módulo de seguimiento en tiempo real que proporciona instrucciones y seguimiento de datos de un procesador.

<sup>11</sup> Familia lógica perteneciente a la TTL en donde el uno lógico se define entre 2 y 3.3 V.

### **Unidades de almacenamiento**

El SmartFusion 2 cuenta con 512 kB de memoria no volátil embebida y 64 kB de memoria estática de acceso aleatorio (SRAM, por sus siglas en inglés), además de que la OBC proporciona 8 MB de MRAM para el almacenamiento de código y su ejecución. A su vez, el SCMI cuenta con 4 GB de memoria Flash para el almacenamiento masivo de datos y una ranura para tarjetas MicroSD.

### **Interfaces eléctricas y periféricos**

El SCMI cuenta con 30 terminales GPIO de 3.3 V, un RTC con capacidad para funcionar sin energía por al menos 40 min, dos señales PPS, un magnetómetro de 3 ejes, un acelerómetro criptográfico de 3 ejes, un acelerómetro bajo el estándar 131.0-B-3 del Comité Constitutivo para Datos de Sistemas Espaciales<sup>12</sup> (CCSDS, por sus siglas en inglés) de 3 ejes, sensores de tensión y corriente internas embebidos, tres sensores de temperatura interna embebidos distribuidos en diferentes ubicaciones del sistema, un módulo GPS opcional y un sistema de marcación por tonos o sistema multifrecuencial (DTMF, por sus siglas en inglés).

### **Interfaces de comunicación serial**

El sistema cuenta con un bus CAN, dos buses I<sup>2</sup>C, ocho buses UART LVTTTL, dos buses RS485 configurables como uno RS422, un bus SPI con capacidad de hasta 7 esclavos y una interfaz periférico serial cuádruple (QSPI, por sus siglas en inglés) con capacidad de hasta 7 esclavos. Además, posee unas terminales de expansión que contienen un bus SPI y un bus QSPI, ambos con capacidad de hasta 2 esclavos sobre señal diferencial de baja tensión (LVDS, por sus siglas en inglés).

### **Propiedades físicas y eléctricas**

El KRYTEN-M3 cuenta con el factor de forma PC/104, como se puede observar en la *Figura 14*, con dimensiones de 90.2 x 95.9 x 10.2 mm, una masa de 68 g, un rango de temperatura de operación de -40 a +80 °C y de temperatura de almacenamiento de -40 a +85 °C. El SCMI opera con una tensión de 3.3 V y tiene un consumo de energía promedio de 610 mW.



*Figura 14. KRYTEN-M3 [7]*

### **Técnicas de mitigación y tolerancia a fallas**

El SmartFusion 2 está basado en tecnología flash, por lo que es tolerante a SEU, además de que las memorias Flash y MRAM están protegidas por código de detección y corrección de errores (EDAC, por sus siglas en inglés) y posee registros con redundancia modular triple (TMR, por sus siglas en inglés). El sistema además posee protección contra latch-up de evento único (SEL, por sus siglas en inglés) en sus entradas de alimentación mediante un limitador de corriente de latch-up (LCL, por sus siglas en inglés) y detección de caída de tensión en la fuente de alimentación (brownout detection). El SCMI también cuenta con hardware dedicado para un temporizador watchdog interno del SmartFusion 2.

<sup>12</sup> Comité designado a desarrollar y establecer estándares de datos y sistemas espaciales.

### 3.2.5 CubeComputer Nanosatellite OBC [8]

**Fabricante:** CubeSpace.

**Precio:** USD 4,500.00

**Disponible en:** <https://cubespace.co.za/cubecomputer>

#### *Procesador de a bordo*

La CPU de la OBC de CubeSpace está compuesta por un microcontrolador EFM32GG280F1024 basado en la arquitectura ARM Cortex-M3 de 32 bits tipo RISC con arquitectura Harvard que opera con un rango de frecuencia de 4 hasta 48 MHz, el cual utiliza el bus SWD como interfaz de programación y depuración y el bus UART para depuración, opera con FreeRTOS y posee un TRL de 9.

#### *Unidades de almacenamiento*

El SCMI utiliza dos SRAM de 1 MB cada una como memoria volátil, una ranura para tarjetas MicroSD de hasta 2 GB destinada para el almacenamiento masivo de datos, una memoria Flash externa de 4 MB para el almacenamiento de código y una memoria de solo lectura eléctricamente programable y borrable (EEPROM, por sus siglas en inglés) externa de 32 kB para el almacenamiento crítico de código, como un programa para operación en modo seguro de los efectos de la radiación o errores de software.

#### *Interfaces eléctricas y periféricos*

El microcontrolador cuenta con un RTC y ADC internos, sin embargo, el sistema posee un conector que puede ser utilizado para expandir las capacidades de la tarjeta agregando cuatro PWM, cuatro ADC y múltiples GPIO conectados directamente al procesador, este conector permite aplicaciones conectando la OBC con un ADCS o una carga útil.

El diagrama de bloques del CubeComputer Nanosatellite OBC se presenta en la *Figura 15*.

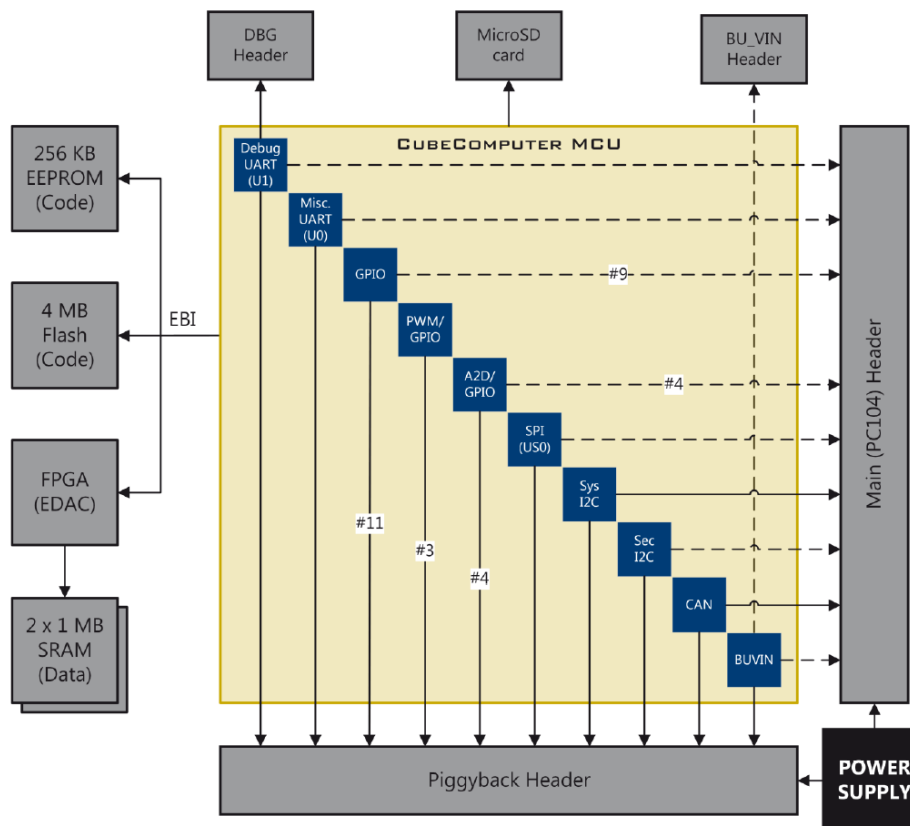


Figura 15. Diagrama de bloques del CubeComputer Nanosatellite OBC [8]

### ***Interfaces de comunicación serial***

El sistema cuenta con un bus CAN, dos buses I<sup>2</sup>C con capacidad multimaestro, en donde uno se encuentra conectado al conector PC/104 y puede ser utilizado como esclavo o maestro, mientras que el otro bus considerado como secundario puede ser empleado para comunicarse por separado con otro esclavo, como un dispositivo o sistema del satélite. La OBC también posee dos buses UART y un bus SPI, además de que su conector de expansión puede ser utilizado para agregar interfaces como I<sup>2</sup>C, UART y SPI.

### ***Propiedades físicas y eléctricas***

Como se puede ver en la *Figura 16*, el sistema está diseñado bajo el estándar PC/104, el cual cuenta con dimensiones de 96 x 90 x 10 mm y una masa de 56 g. Opera en un rango de temperatura de -10 a +70 °C, se alimenta con una tensión de 3.3 V y su consumo de energía promedio es de 200 mW. Además, está diseñado para misiones de más de 10 años de duración en vuelo con una resistencia a la dosis de ionización total (TID, por sus siglas en inglés) de 20 krad.



*Figura 16. CubeComputer Nanosatellite OBC [8]*

### ***Técnicas de mitigación y tolerancia a fallas***

El sistema cuenta con dos temporizadores watchdog, uno interno y uno externo para evitar el latch-up. Además, la memoria SRAM cuenta con protección contra SEL y SEU mediante un FPGA con EDAC.

### **3.2.6 OBC EnduroSat [9]**

**Fabricante:** EnduroSat.

**Precio:** €2,900.00 (USD 3,121.41) bloqueado (para uso exclusivo con sistemas de EnduroSat) / €7,500.00 (USD 8,072.62) incluyendo el kit de desarrollo completo de software.

**Disponible en:** <https://www.endurosat.com/cubesat-store/cubesat-obc/onboard-computer-obc/>

### ***Procesador de a bordo***

La CPU está conformada por un procesador de la familia ARM Cortex, permitiendo escoger entre el M4 y M7, ambos de 32 bits tipo RISC y arquitectura Harvard, los cuales pueden correr hasta una frecuencia de 180 MHz y 216 MHz, respectivamente. Posee el puerto JTAG para la programación y depuración y tiene FreeRTOS instalado.

El diagrama de bloques del OBC de EnduroSat se presenta en la *Figura 17*.



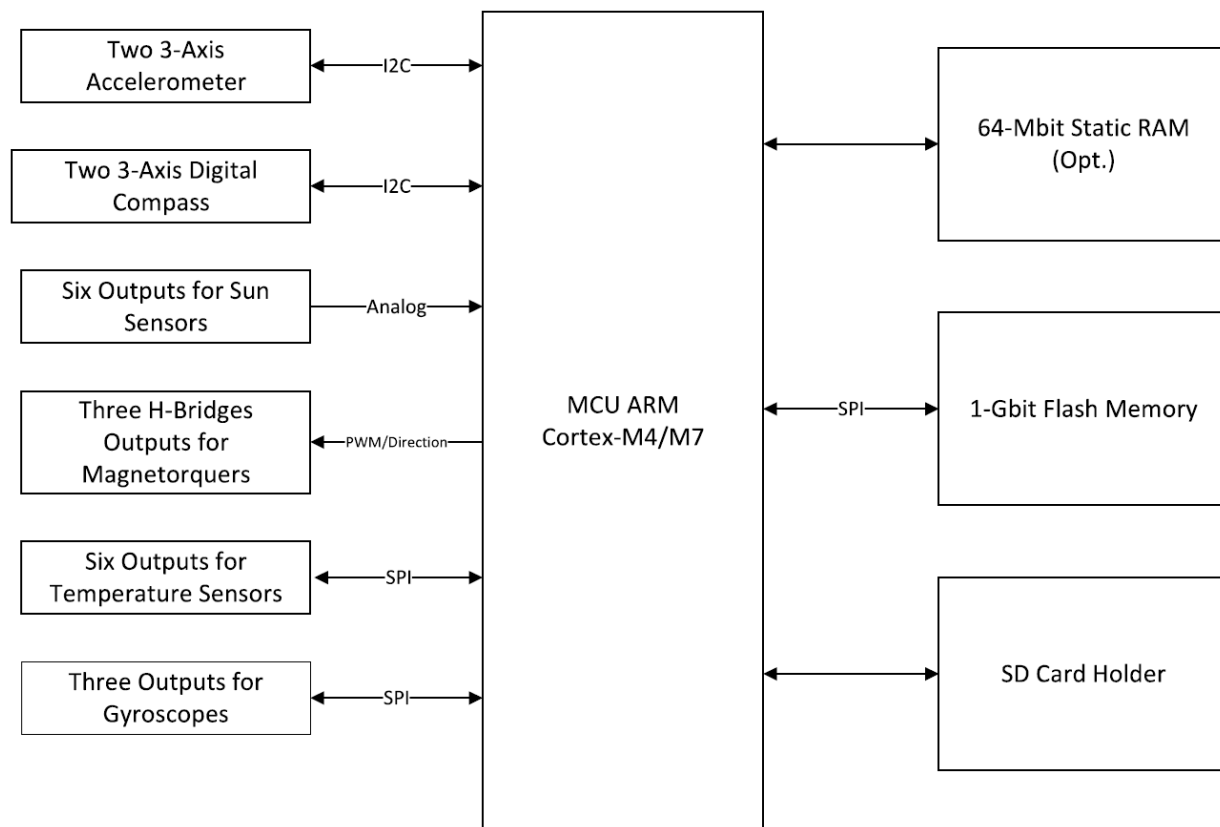


Figura 17. Diagrama de bloques del OBC de EnduroSat [9]

### Unidades de almacenamiento

El sistema cuenta con 256 kB (para el M4) y 2 MB de RAM (para el M7) y una SRAM de 8 MB opcional como memoria volátil, ambos procesadores poseen 2 MB de memoria Flash para el almacenamiento de código y el SCMI posee 32 MB de memoria NOR Flash y una ranura para tarjetas MicroSD para el almacenamiento masivo de datos.

### Interfaces eléctricas y periféricos

La OBC posee seis terminales GPIO con diodos de protección y resistencias de pull-down<sup>13</sup>, seis ADC para sensores solares, tres salidas PWM para torques magnéticos y un RTC. Además, el SCMI puede ser utilizado como ADCS, por lo que posee dos acelerómetros y dos magnetómetros, ambos pares de 3 ejes, redundantes y ubicados en la misma posición, pero del lado contrario de la OBC conectados mediante un bus I<sup>2</sup>C, tres terminales para giroscopios y seis para sensores de temperatura mediante SPI, un conector para el despliegue de antena y como se puede ver en la *Figura 18*, un área tipo protoboard para conectar de manera sencilla la carga útil y que esta pueda acceder a la alimentación y los buses de comunicación serial.

### Interfaces de comunicación serial

El sistema cuenta con el bus CAN, dos buses I<sup>2</sup>C: el primero para la comunicación con todos los sistemas del satélite y el segundo para la carga útil y el control del despliegue de una antena para ultra alta frecuencia<sup>14</sup> (UHF, por sus siglas en inglés) de EnduroSat en el área tipo protoboard, un bus UART libre para la carga útil, dos buses USART: el primero utilizado para un transceiver UHF de EnduroSat y el segundo se encuentra libre para la carga útil, que puede ser utilizado como puerto virtual<sup>15</sup> cuando el puerto USB está montado, dos buses

<sup>13</sup> Resistencia conectada a tierra para establecer un estado lógico bajo cuando el circuito se encuentra en estado de reposo.

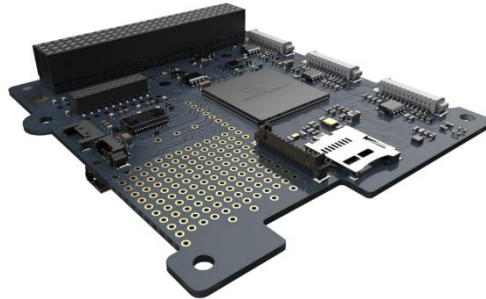
<sup>14</sup> Banda del espectro electromagnético que utiliza el rango de frecuencias de 300 MHz a 3 GHz.

<sup>15</sup> Paquete de software que emula un puerto de hardware.

SPI: en donde el primero es utilizado para un transceiver de banda S<sup>16</sup> de EnduroSat, mientras que el segundo se encuentra libre para uso de la carga útil y el área tipo protoboard, también posee un puerto USB conectado al área tipo protoboard y al conector PC/104.

### ***Propiedades físicas y eléctricas***

El SCMI está diseñado con el factor de forma PC/104 (véase la *Figura 18*) con unas dimensiones de 95 x 89 x 23.2 mm y una masa de 58 g. Su temperatura de operación se encuentra en el rango de -30 a +85 °C y su temperatura de almacenamiento entre los +15 y +27 °C y necesita de 3.3 V de tensión de alimentación.



*Figura 18. OBC EnduroSat [9]*

### **3.2.7 ABACUS [10]**

**Fabricante:** Group of Astrodynamics for the Use of Space Systems (GAUSS Srl).

**Disponible en:** <https://www.gaussteam.com/products/onboard-computer/abacus-2/>

#### ***Procesador de a bordo***

La OBC ABACUS cuenta con una CPU compuesta por dos núcleos independientes: un microcontrolador MSP430F5438A-EP de Texas Instruments (TI) de 16 bits tipo RISC con arquitectura Von Neumann cuya frecuencia alcanza los 24 MHz y un FPGA Xilinx Spartan-3E con frecuencias de 25 o 100 MHz, conectados entre sí mediante un bus de 24 líneas. Utiliza el bus JTAG para la programación y depuración y es compatible con el sistema FreeRTOS.

#### ***Unidades de almacenamiento***

Además de la RAM que posee el FPGA de 500k de compuertas, el sistema cuenta con 2 MB de SRAM dedicados al FPGA y dos memorias NOR Flash embebidas de 16 MB para el almacenamiento de datos.

#### ***Interfaces eléctricas y periféricos***

El microcontrolador cuenta con 45 terminales GPIO con 16 niveles de tensión configurables mediante interrupciones y diez ADC de 3.3 V. En cambio, el FPGA posee 34 terminales GPIO y ocho canales de un controlador integral proporcional generalizado. Además, ambos núcleos comparten la conexión de un magnetómetro, giroscopio y acelerómetro embebidos, cada uno de 3 ejes, tres sensores de temperatura, un monitor de corriente y un RTC embebidos.

<sup>16</sup> Estándar de ondas de radio de IEEE que utiliza el rango de frecuencias de 2 a 4 GHz.

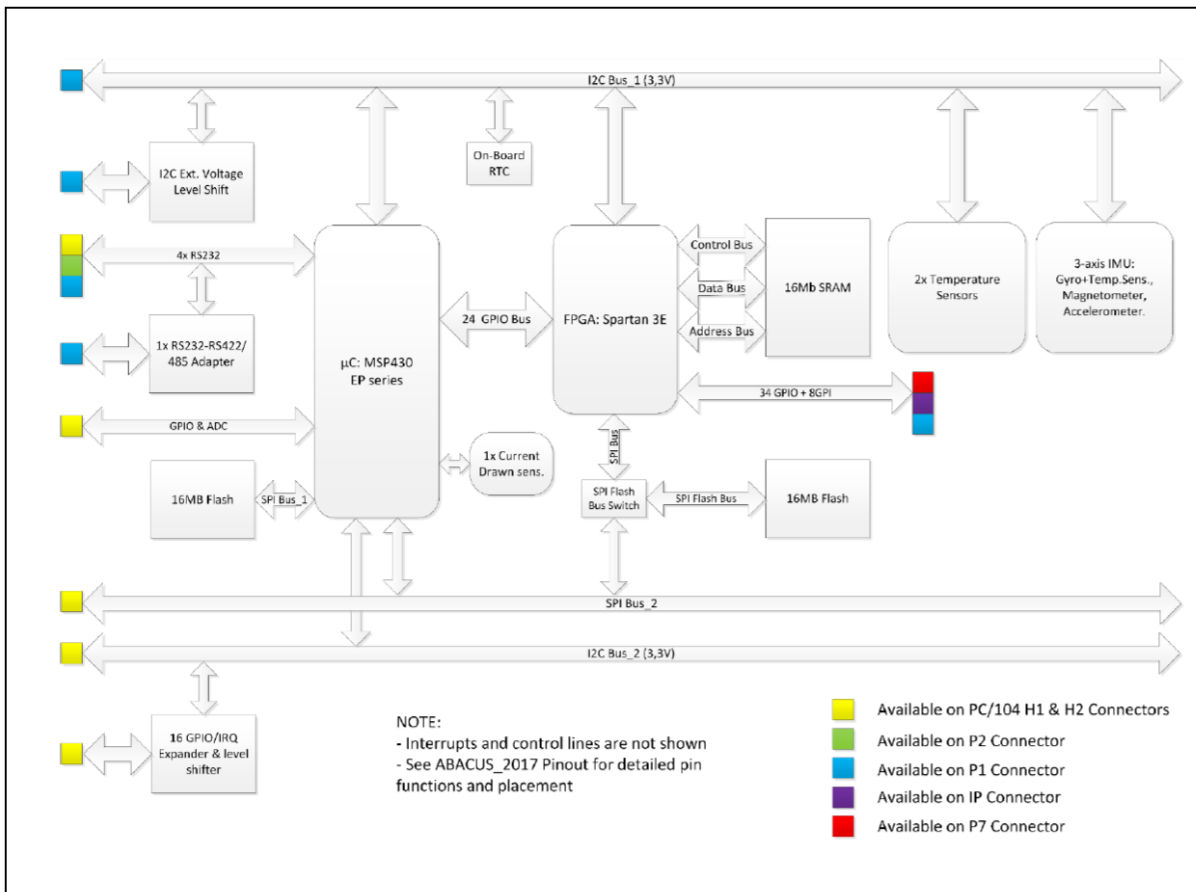


Figura 19. Diagrama de bloques de ABACUS [10]

### Interfaces de comunicación serial

El sistema posee dos buses I<sup>2</sup>C, un SPI, un LVDS (mediante las terminales GPIO del FPGA) y cuatro puertos de comunicación RS232, de los cuales uno funciona bajo el estándar RS232 – RS422/485 full o half duplex.

### Propiedades físicas y eléctricas

El SCMI está diseñado con el factor de forma PC/104 (véase la Figura 20) con dimensiones de 95.86 x 90.14 x 23.24 mm y una masa de 62 g. Su rango de temperatura de operación es de -40 a +85 °C y de almacenamiento de -55 a +105 °C y necesita 5 V de tensión de alimentación y tiene un consumo de energía de 50 a 800 mW.



Figura 20. ABACUS [10]

### Técnicas de mitigación y tolerancia a fallas

Al poseer un microcontrolador y un FPGA organizados como dos núcleos diferentes que trabajan en conjunto el SCMI cuenta con redundancia de hardware y tolerancia a fallas en modo común, además de que el FPGA puede alcanzar una alta confiabilidad utilizando códigos de configuración TMR.

### 3.2.8 HERCULES [11]

**Fabricante:** GAUSS Srl.

**Disponible en:** <https://www.gaussteam.com/products/onboard-computer/hercules/>

#### Procesador de a bordo

La CPU del sistema HERCULES cuenta con un procesador dual ARM Cortex R4F (RM46) de 32 bits tipo RISC, el cual cuenta con dos núcleos trabajando en paralelo (lockstep) y opera hasta 220 MHz.

El diagrama de bloques de HERCULES se presenta en la *Figura 21*.

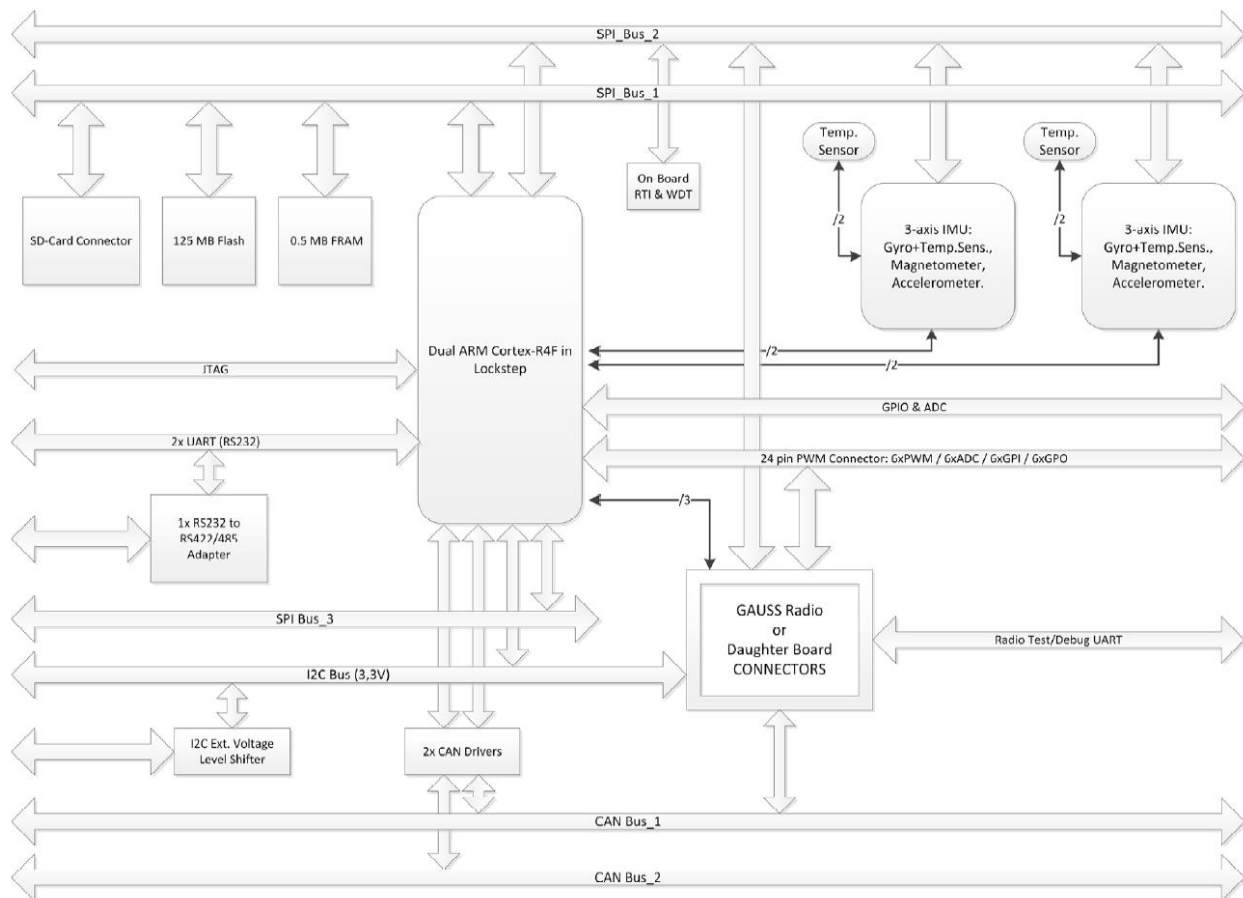


Figura 21. Diagrama de bloques de HERCULES [11]

#### Unidades de almacenamiento

El SCMI cuenta con tecnología SRAM como memoria volátil, 128 MB de memoria NOR Flash y una ranura para tarjetas SD destinadas al almacenamiento masivo de datos y 512 kB de FRAM para el almacenamiento crítico de datos.

### ***Interfaces eléctricas y periféricos***

HERCULES cuenta con más de veinte terminales GPIO, dieciséis ADC, un RTC embebido con oscilador de cristal compensado por temperatura<sup>17</sup> (TCXO, por sus siglas en inglés) para obtener mayor exactitud, un conector con doce terminales GPIO, seis ADC y seis PWM extras destinado para integrar tres motores y tres bobinas. Además, posee de manera embebida acelerómetros, giroscopios y magnetómetros duales de 3 ejes, tres sensores de temperatura, dos temporizadores de gama alta mejorados<sup>18</sup> (N2HET, por sus siglas en inglés) con 32 canales que pueden ser programados y ejecutados en paralelo e independientemente del CPU y un conector para daughterboards compatibles con una antena UHF de GAUSS Srl (véase en la *Figura 22*).

### ***Interfaces de comunicación serial***

El sistema posee dos buses CAN 2.0B, una interfaz I<sup>2</sup>C, una SPI y dos UART, de los cuales uno es RS232 y el otro posee un adaptador a RS422/485 half y full duplex.

### ***Propiedades físicas y eléctricas***

HERCULES está diseñado con el factor de forma PC/104, posee una masa de 65 g y opera en un rango de temperatura de -40 a +85 °C, alimentándose con 3.3 o 5 V de tensión.



*Figura 22. HERCULES [11]*

### ***Técnicas de mitigación y tolerancia a fallas***

La OBC está diseñada de manera que posee redundancia sin pérdidas de energía, en donde ambos núcleos del CPU operan en paralelo y sus salidas son comparadas para obtener redundancia y mayor confiabilidad, además de que, al igual que las memorias de cada uno, están físicamente orientados de manera distinta para reducir las fallas en modo común. El CPU usa códigos de corrección de errores (ECC, por sus siglas en inglés) en las memorias Flash y SRAM brindando protección contra SEU y también realiza revisiones de bit de paridad en los periféricos de las memorias.

### **3.2.9 N-ART OBCOMMS V2 [12]**

**Fabricante:** GUMUSH.

**Disponible en:** <https://gumush.com.tr/product/n-art-obcomms-v2/>

#### ***Procesador de a bordo***

La CPU del sistema consta de dos procesadores ARM M4 (F429 & F407), ambos de 32 bits tipo RISC, los cuales operan a una frecuencia de hasta 180 MHz. El sistema es compatible con FreeRTOS.

<sup>17</sup> Oscilador de cristal diseñado especialmente para operar en ambientes de alta temperatura.

<sup>18</sup> Temporizador inteligente avanzado sofisticado para aplicaciones de tiempo real.

### **Unidades de almacenamiento**

Las unidades de almacenamiento que posee el sistema son 256 kB de RAM integrada y 64 kB de SDRAM externa como memorias volátiles, 2 MB de memoria Flash para el almacenamiento de código y dos ranuras para tarjetas SD destinadas para el almacenamiento masivo de datos.

### **Interfaces eléctricas y periféricos**

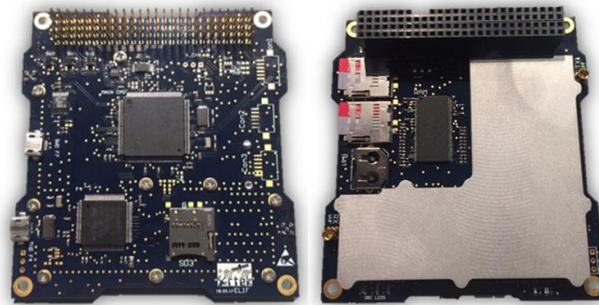
El sistema cuenta con un RTC y puede implementarse como módulo de comunicaciones, para el cual posee las modulaciones 2FSK, 4FSK, CW y GMSK.

### **Interfaces de comunicación serial**

La OBC posee las interfaces CAN, I<sup>2</sup>C, UART y SPI.

### **Propiedades físicas y eléctricas**

Como se aprecia en la *Figura 23*, el sistema está diseñado bajo el estándar PC/104, con dimensiones de 93 x 86 x 15 mm y una masa de 88 g.



*Figura 23. N-ART OBCOMMS V2 [12]*

### **Técnicas de mitigación y tolerancia a fallas**

Como medidas de protección, el sistema cuenta con un temporizador watchdog y protección contra picos de tensión y corriente.

## **3.2.10 IMT CubeSat OBC [13]**

**Fabricante:** Ingegneria Marketing Tecnologia (IMT)

**Disponible en:** <http://www.imtsrl.it/obc-cubesat.html>

El SCMI de IMT está diseñado en base de componentes COTS y ofrece un sistema con vida útil mayor a tres años en órbita.

### **Procesador de a bordo**

La CPU del sistema está constituida por el PIC32MZ M14K de 32 bits tipo RISC y arquitectura Harvard, el cual opera hasta una frecuencia de 200 MHz. El SCMI utiliza el bus JTAG para la depuración y programación. El sistema es compatible con FreeRTOS y Micrium uC/OS-II.

### **Unidades de almacenamiento**

La OBC posee 16 MB expandibles hasta 64 MB de memoria dinámica de acceso aleatorio (DRAM, por sus siglas en inglés) como memoria volátil, 64 MB expandibles hasta 128 MB de memoria NOR Flash para el almacenamiento de código de mayordomo y 8 GB de memoria NAND Flash destinados al almacenamiento masivo de datos.

### ***Interfaces eléctricas y periféricos***

Las interfaces con las cuenta el sistema son 32 canales de ADC de 12 bits (de los cuales dieciséis se encuentran disponibles para el usuario y dieciséis se utilizan para la transferencia de telemetría interna de la computadora, cuatro canales PWM bidireccionales, un RTC dual con batería de estado sólido con capacidad para operar durante 5 días sin alimentación externa, una interfaz de cámara de a bordo, un circuito comparador analógico auxiliar, un circuito de muestreo y retención (sample & hold) dual de 32 canales multiplexados, un comparador front-end analógico, un bus front-end dual analógico independiente, una unidad de micro procesamiento integrada y sensores de temperatura de a bordo embebidos.

### ***Interfaces de comunicación serial***

El sistema posee dos interfaces CAN 2.0B de hasta 1 Mb/s, dos buses I<sup>2</sup>C, dos buses UART, dos buses SPI con capacidad de hasta 16 esclavos, un puerto USB 2.0 tipo A de alta velocidad, una interfaz de NAND Flash abierta para el banco de memorias NAND Flash y una interfaz cuádruple en serie (SQI, por sus siglas en inglés) para el banco de memorias NOR Flash.

### ***Propiedades físicas y eléctricas***

Como se aprecia en la *Figura 24*, la IMT OBC está diseñada con el factor de forma PC/104, sus dimensiones son de 96 x 90 x 10 mm con una masa de 38 g y rangos de temperatura de operación de -20 a +70 °C y de almacenamiento de -40 a +85 °C. El sistema opera con 3.3 V de tensión de alimentación y tiene un consumo de energía de 300 mW en operación normal. Además, cuenta con resistencia de exposición a la radiación menor a 15 krad, una resistencia a la vibración aleatoria de 10 grms (en todos los ejes) y emplea un conector Molex para integrar otras interfaces.



*Figura 24. IMT CubeSat OBC [13]*

### ***Técnicas de mitigación y tolerancia a fallas***

El sistema cuenta con un sistema dual de temporizadores watchdog y deadman, TMR y un sistema de votación mayoritaria en los bancos de memorias NOR y NAND Flash, circuitos resistentes al latch-up independientes para cada banco de memorias y monitores de potencia para reiniciar el sistema. Además de que el circuito de muestreo y retención cuenta con redundancia y los 32 canales del ADC poseen redundancia en caliente en cada canal.

#### **3.2.11 SatBus 3C2 [14]**

**Fabricante:** NanoAvionics.

**Costo:** €10,000.00 (USD 10,791.70) con soporte técnico.

**Disponible en:** <https://nanoavionics.com/cubesat-components/cubesat-on-board-computer-main-bus-unit-sat-bus-3c2/>

## Procesador de a bordo

La arquitectura del SCMI está basada en la serie de microcontrolador STM32 H7, con un núcleo ARM Cortex-M7 de 32 bits tipo RISC de arquitectura Harvard y frecuencia de operación de 200 a 400 MHz, el cual desempeña funciones tanto de OBC como ADCS. El sistema, de bajo consumo, posee una FPU con doble precisión, se puede depurar y programar mediante las interfaces USB y SWD y posee soporte para el Protocolo Espacial CubeSat<sup>19</sup> (CSP, por sus siglas en inglés), lo que significa que es compatible con FreeRTOS, Linux, Mac OS X y Microsoft Windows.

El diagrama de bloques del SatBus 3C2 se presenta en la *Figura 25*.

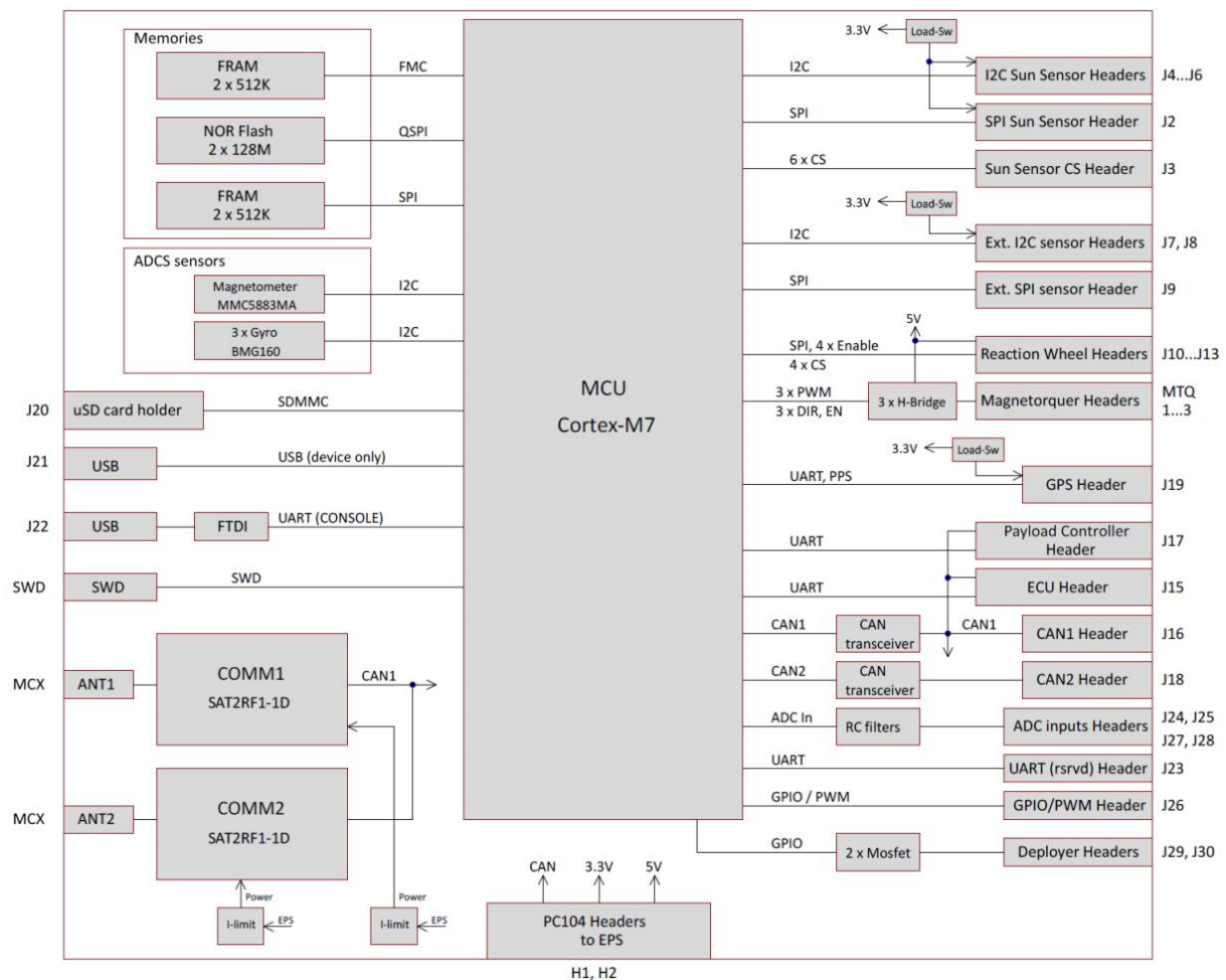


Figura 25. Diagrama de bloques del SatBus 3C2 [14]

## Unidades de almacenamiento

El sistema cuenta con tres tipos de memorias externas: dos FRAM de 512 kB para el almacenamiento crítico de datos, dos memorias NOR Flash SCL de 128 MB y una ranura para tarjeta MicroSD destinadas al almacenamiento masivo de datos. Además, posee integrados 1 MB de RAM y 2 MB de memoria Flash para el almacenamiento de código.

## Interfaces eléctricas y periféricos

La OBC cuenta con cinco terminales GPIO, ocho entradas de ADC y tres salidas PWM. Además, posee una señal PPS, controladores para torques magnéticos de a bordo (puentes H), sensores de tensión y corriente para los torques magnéticos, tres giroscopios de 3 ejes (BMG160) ubicados en diferentes orientaciones y un magnetómetro

<sup>19</sup> Protocolo de comunicación de red por capas diseñado para el estándar CubeSat.



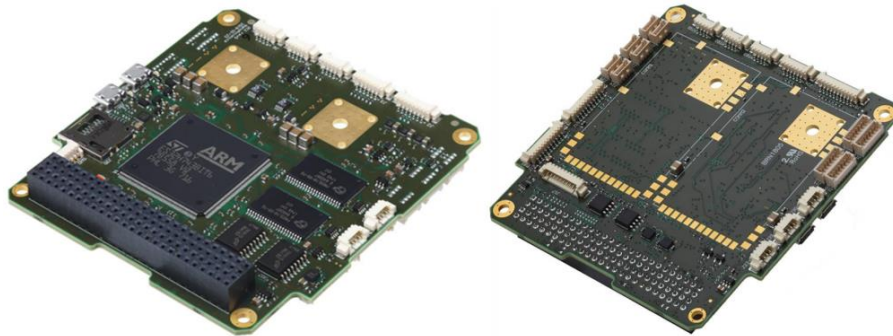
de 3 ejes (MMC5883MA) de a bordo, ambos conectados por I<sup>2</sup>C y un radio digital UHF dual (redundante) SAT2RF1-1D.

### ***Interfaces de comunicación serial***

El sistema posee dos módulos CAN (uno para el módulo UHF y otro disponible para el uso de sensores auxiliares), cuatro buses I<sup>2</sup>C (uno para los giroscopios y el magnetómetro embebidos, dos auxiliares para sensores externos al ADCS y uno para sensores solares), un bus UART destinado a un módulo GPS, un bus USART, tres interfaces SPI (una para la FRAM, uno para sensores solares y externos al ADCS y uno para ruedas de reacción), un QSPI para la memoria NOR Flash y el bus USB.

### ***Propiedades físicas y eléctricas***

Como se puede observar en la *Figura 26*, el sistema está diseñado bajo el estándar PC/104 con dimensiones de 95.5 x 90.17 x 15.12 mm y masa de 62 g. La OBC opera en un rango de temperatura y almacenamiento de -40 a +85 °C, se alimenta con 3.3 V y consume un promedio de 330 mW. Tiene una resistencia de exposición a la radiación de 20 krad y una resistencia a la vibración de 14 grms.



*Figura 26. Vistas superior e inferior del SatBus 3C2 [14]*

### ***Técnicas de mitigación y tolerancia a fallas***

El sistema cuenta con redundancia de hardware en algunos de sus módulos, resistencia a la radiación en la FRAM y posee interruptores de carga para el ADCS externo, los cuales incluyen protección de corriente.

## **3.2.12 NANOobc Gen2 [15]**

**Fabricante:** SkyLabs

**Disponible en:** <https://www.skylabs.si/>

NANOobc Gen2 es una plataforma escalable desde CubeSat 2U hasta 20U basada en un enfoque totalmente distribuido para reducir el nivel de complejidad del sistema e incrementar su versatilidad. El SCMI se encuentra actualmente en pruebas y está impulsando cuatro misiones de constelaciones futuras, de donde la primera es estima ser lanzada a finales de 2020.

### ***Procesador de a bordo***

La CPU del sistema está conformada por el SoC PicoSkyFT con arquitectura de 8/16 bits tipo RISC de 16 MHz que posee un FPGA IGLOO2, posee el bus JTAG para la programación y depuración, y un conector de depuración orientado a PicoSkyLINK para la depuración local. El sistema es compatible con FreeRTOS y se encuentra con un TRL 6.

### ***Unidades de almacenamiento***

El sistema integra un banco de 2 MB conformado por dos SRAM paralelas como memoria volátil, un banco de dos memorias NAND Flash de 2 GB en total para el almacenamiento masivo de datos, 1 MB de MRAM para el almacenamiento crítico de datos y telemetría adicional y 4 MB proporcionados por dos MRAM para el almacenamiento de código y telemetría.

### Interfaces eléctricas y periféricos

El SCMI posee ocho terminales GPIO, un ADC, un PWM, tres entradas de reloj disponibles incluyendo un RTC, sensores embebidos para medir telemetría del ADCS (corriente, tensión y temperatura) y un receptor GPS con una PPS.

### Interfaces de comunicación serial

El sistema posee el bus CAN redundante para telemetría y telecomando, dos LVDS para transferencia de datos de alta velocidad y sus terminales GPIO poseen interfaz con capacidad para I<sup>2</sup>C, SPI y UART.

El diagrama de bloques del NANOobc Gen2 se presenta en la *Figura 27*.

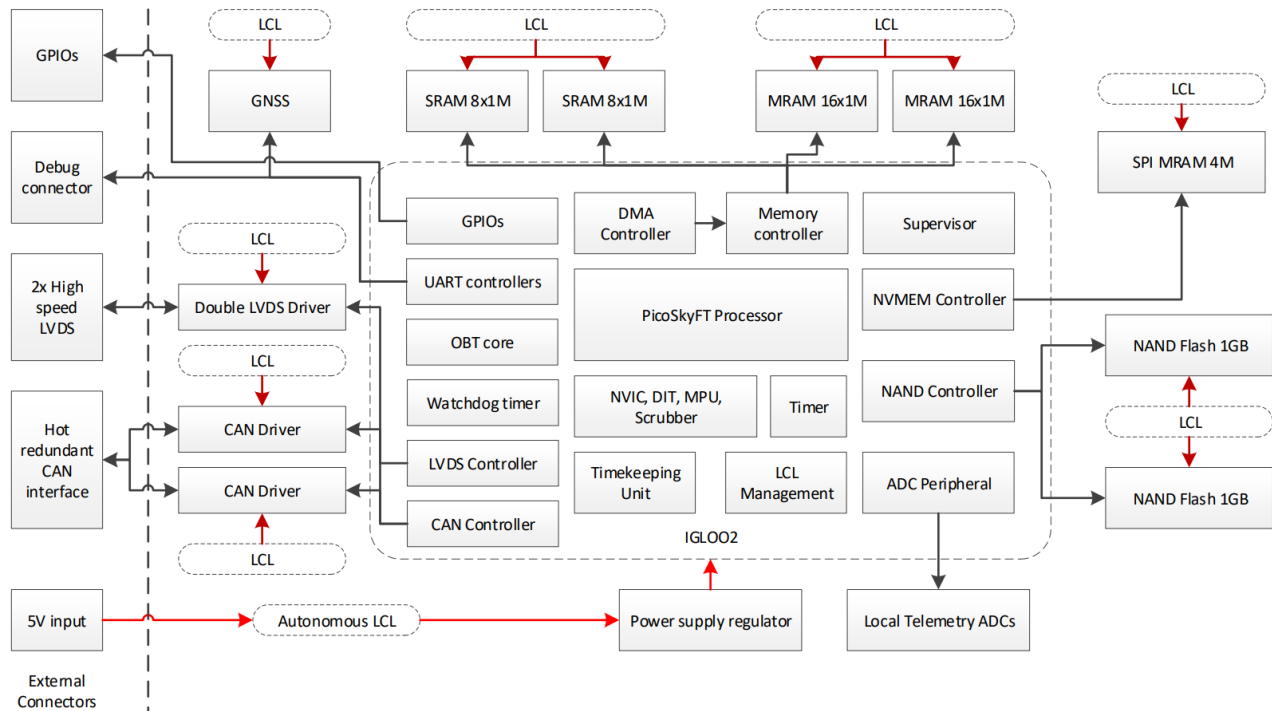


Figura 27. Diagrama de bloques del NANOobc Gen2 [15]

### Propiedades físicas y eléctricas

Como se puede apreciar en la *Figura 28*, la NANOobc Gen2 es compatible con el factor de forma PC/104 pero carece de su conector característico sustituyéndolo por otro con menor número de terminales, sus dimensiones son de 94.9 x 90.2 x 11.1 mm y su masa es de 63 g. Soporta un rango de temperatura de operación de -10 a +60 °C y de almacenamiento de -20 a +65 °C, se alimenta con 5 V, consume un pico de 1.5 W y menos de 1 W en modo desocupado y soporta una exposición a la radiación de 10 krad.

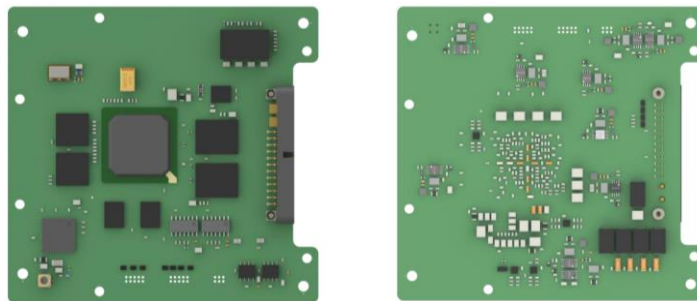


Figura 28. Vista superior e inferior del NANOobc Gen2 [15]

### Técnicas de mitigación y tolerancia a fallas

La NANOobc Gen2 se caracteriza por implementar diversas técnicas avanzadas de detección, aislamiento y recuperación de fallas<sup>20</sup> (FDIR, por sus siglas en inglés), posee un temporizador watchdog, TMR en todos sus registros, EDAC protegiendo todos sus bancos de memorias y registros. Su diseño asegura tolerancia a fallas mediante redundancia de sus componentes seleccionados por su herencia de vuelo, protección contra latch-up mediante monitoreo constante de corriente y fiabilidad y robustez contra efectos de evento único (SEE, por sus siglas en inglés). Para las memorias posee redundancia en caliente de las NAND Flash, MRAM inmune a SEE, almacenamiento masivo de datos redundante, controlador de acceso directo a memoria<sup>21</sup> (DMA, por sus siglas en inglés), dos sistemas de verificación de redundancia cíclica (CRC, por sus siglas en inglés) protegiendo el banco de memorias del programa y una arquitectura de dos bancos de memorias permitiendo que el programa en ejecución realice actualizaciones al firmware en el otro banco. También verifica con CRC que se esté ejecutando el código correctamente y posee un bus CAN con redundancia en caliente y fría.

### 3.2.13 Análisis de los SCMI comerciales

Como se mencionó anteriormente, la investigación del estado del arte se dirigió hacia las características con las que fueron diseñados los SCMI de propósito general de diferentes fabricantes organizadas en módulos con funciones específicas para facilitar el análisis y llegar a conclusiones más certeras que sirvan como punto de partida para un diseño propio. A continuación, se presentan los módulos definidos con sus tablas comparativas correspondientes, presentando los resultados obtenidos de cada sistema seguido de las conclusiones a las que se llegó en cada caso.

#### Procesador de a bordo

El procesador de a bordo representa el núcleo del sistema, en donde se ejecutan las instrucciones definidas por software, así como la CPU y sus principales características, su arquitectura y frecuencia de operación. Además, se incluyen las interfaces de comunicación que emplean para la programación y depuración de la CPU, los RTOS compatibles con el sistema y su TRL. Estos resultados se presentan en la *Tabla 1*.

Tabla 1. Procesador de a bordo de los SCMI comerciales

SCMI	Fabricante	CPU	Arquitectura	Frecuencia de operación MHz	Programación / Depuración	RTOS	TRL	Costo USD
<i>IOBC</i>	ISIS	ARM9 AT91SAM9G20 32 bits RISC	Harvard	400	JTAG UART (interfaz de usuario)	FreeRTOS KubOS Linux	9	4,982.23 – 6,963.80
<i>NanoMind A3200</i>	GOMspace	Atmel AVR AT32UC3C 32 bits RISC	Harvard	8 – 64	JTAG (programación) USART (depuración)	FreeRTOS	9	6,783.06
<i>OBC C3S</i>	C3S	ARM7 32 bits RISC	Harvard	300	USB SWD (depuración)	FreeRTOS	9	10,128.09 – 15,458.67

<sup>20</sup> Módulo de ingeniería de control utilizado para monitorear sistemas y detectar el tipo y ubicación de fallas.

<sup>21</sup> Interfaz para leer o escribir en memoria independientemente de la CPU.

<i>KRYTEN-M3</i>	AAC Clyde Space	Microsemi SmartFusion 2 FPGA ARM Cortex-M3 32 bits RISC	Harvard	50	JTAG con ETM UART LVTTTL (depuración)	FreeRTOS RTEMS	9	8,500.00 / 23,100.00
<i>CubeComputer Nanosatellite OBC</i>	CubeSpace	ARM Cortex-M3 EFM32GG280F1024 32 bits RISC	Harvard	4 – 48	SWD UART (depuración)	FreeRTOS	9	4,500.00
<i>OBC EnduroSat</i>	EnduroSat	ARM Cortex-M4 32 bits RISC ARM Cortex-M7 32 bits RISC	Harvard	180 216	JTAG	FreeRTOS	9	3,121.41 – 8,072.62
<i>ABACUS</i>	GAUSS Srl	MSP430F5438A 16 bits RISC FPGA Xilinx Spartan-3E	Von Neumann	24 25 / 100 (FPGA)	JTAG	FreeRTOS	9	
<i>HERCULES</i>	GAUSS Srl	ARM Cortex-R4F (RM46) 32 bits RISC		220				
<i>N-ART OBCOMMS V2</i>	GUMUSH	ARM Cortex-M4 (F429 & F407) 32 bits RISC		180		FreeRTOS	9	
<i>IMT CubeSat OBC</i>	IMT	PIC32MZ M14K 32 bits RISC	Harvard	200	JTAG	FreeRTOS Micrium uC/OS-II	9	
<i>SatBus 3C2</i>	Nano Avionics	STM32 H7 ARM Cortex-M7 32 bits RISC	Harvard	200 – 400	USB SWD	FreeRTOS Linux Mac OS X Microsoft Windows	9	10,791.70
<i>NANOobc Gen2</i>	SkyLabs	SoC PicoSkyFT 8/16 bits RISC FPGA IGLOO2		16	JTAG PicoSkyLINK (depuración)	FreeRTOS	6	

Conclusiones: En su totalidad, los SCMI poseen un microcontrolador como CPU, sin embargo, algunos sistemas cuentan con dos núcleos para brindar redundancia integrando un segundo microcontrolador o un FPGA para brindar la posibilidad de describir hardware y aumentar las capacidades del sistema. En la *Tabla 1* se puede

apreciar claramente que en los SCMI comerciales destacan los procesadores con arquitectura ARM y sus diferentes familias (principalmente la Cortex-M7) y de la misma forma, destacan los procesadores de 32 bits tipo RISC con arquitectura Harvard. El rango de frecuencias de operación es muy variado en cada sistema, siendo 4 MHz la mínima y subiendo hasta 400 MHz en algunos casos.

En cuanto a las interfaces empleadas para la programación y depuración del SCMI, el puerto JTAG es el más ocupado para estos fines seguido del UART y el SWD. Por parte del software, claramente se observa que el sistema más empleado y presente en cada SCMI es el FreeRTOS.

En estos sistemas, al ser comerciales, la mayoría cuenta con un TRL 9 (nivel máximo).

### **Unidades de almacenamiento**

Los diseñadores han optado por implementar principalmente tres bancos de memoria en sus sistemas: memoria de trabajo, memoria de almacenamiento masivo de datos y la memoria de almacenamiento de código, las cuales son consideradas como esenciales para el funcionamiento del sistema. Sumado a ellos, algunos fabricantes integran un cuarto banco de memorias destinado al almacenamiento crítico de datos, el cual proporciona menor capacidad que el almacenamiento masivo, pero emplea una tecnología que brinda mayor protección a la radiación, por lo que, como su nombre lo indica, se emplea para almacenar la información más importante (crítica) de la misión y se asegura de que sea menos probable que estos datos se pierdan a la información “no crítica”.

Existe un quinto banco de memorias poco utilizado conocido como memoria de almacenamiento de código de mayordomo o housekeeping, el cual es una herramienta de software que se encarga de “vigilar” que el código principal se encuentre operando correctamente.

Los resultados obtenidos en cuanto a las tecnologías implementadas para almacenar información en los SCMI y sus capacidades se presentan en la *Tabla 2*.

*Tabla 2. Unidades de almacenamiento de los SCMI comerciales*

<i>SCMI</i>	<b>Memoria volátil</b>	<b>Almacenamiento masivo de datos</b>	<b>Almacenamiento crítico de datos</b>	<b>Almacenamiento de código</b>	<b>Almacenamiento de código de mayordomo</b>
<i>IOBC</i>	32 MB SDRAM	(2x) ranuras para tarjetas SD	256 kB FRAM	1 MB NOR Flash	
<i>NanoMind A3200</i>	32 MB SDRAM	(2x) 64 MB NOR Flash	32 kB FRAM	512 kB Flash	
<i>OBC C3S</i>	384 kB RAM	16 GB MMC 16 MB (daughterboard)		2 MB Flash	
<i>KRYTEN-M3</i>		4 GB Flash (1x) ranura para tarjeta MicroSD		512 kB no volátil 64 kB SRAM 8 MB MRAM	

<i>CubeComputer Nanosatellite OBC</i>	(2x) 1 MB SRAM	(1x) ranura para tarjeta MicroSD		4 MB Flash 32 kB EEPROM	
<i>OBC EnduroSat</i>	256 kB RAM (M4) 2 MB RAM (M7) 8 MB SRAM (opcional)	32 MB NOR Flash (1x) ranura para tarjeta MicroSD		2 MB Flash	
<i>ABACUS</i>	2 MB SRAM (FPGA)	(2x) 16 MB NOR Flash			
<i>HERCULES</i>	SRAM	128 MB NOR Flash (1x) ranura para tarjeta SD	512 kB FRAM		
<i>N-ART OBCOMMS V2</i>	256 kB RAM 64 kB SDRAM (externa)	(2x) ranuras para tarjetas SD		2 MB Flash	
<i>IMT CubeSat OBC</i>	16 – 64 MB DRAM	8 GB NAND Flash			64 – 128 MB NOR Flash
<i>SatBus 3C2</i>	1 MB RAM	(2x) 128 MB NOR Flash (1x) ranura para tarjetas microSD	(2x) 512 kB FRAM	2 MB Flash	
<i>NANOobc Gen2</i>	2 MB SRAM	2 GB NAND Flash	1 MB MRAM	4 MB MRAM	

Conclusiones: La tecnologías que destacan son: la SRAM para la memoria de trabajo, las memorias Flash para el almacenamiento masivo de datos, de código y de código de mayordomo y las FRAM para el almacenamiento crítico de datos. Además, muchos sistemas cuentan con ranuras para la integración de tarjetas SD y MicroSD, brindándole mayor capacidad de almacenamiento al sistema.

### ***Interfaces eléctricas y periféricos***

En cuanto a las interfaces eléctricas y periféricos, se consideran dentro de esta clasificación las terminales de entrada y/o salida que cumplen con un objetivo en particular y aquellos circuitos/módulos embebidos en la tarjeta pero que no pertenecen directamente al CPU, los cuales se implementan en la OBC con el objetivo de

facilitar la integración de tareas típicas que realizan otros sistemas del satélite, además de que los fabricantes diseñan sus sistemas con la finalidad de que sean compatibles con otros subsistemas que también ofrecen y permiten al interesado adquirir un sistema cuyas funciones dependan de sus necesidades. Los resultados obtenidos en este módulo se presentan en la *Tabla 3*.

*Tabla 3. Interfaces eléctricas y periféricos de los SCMI comerciales*

<i>SCMI</i>	<b>GPIO</b>	<b>ADC</b>	<b>PWM</b>	<b>RTC</b>	<b>Otros</b>
<i>IOBC</i>	(27x)	(1x) 8 canales 8/10 bits	(1x) 6 canales	(2x)	Interfaz para sensor de imagen. Sensor de temperatura embebido. Sensor de corriente embebido. Sensor de tensión embebido. PIC supervisor.
<i>NanoMind A3200</i>	(9x)	(1x) 8 canales	(3x)	(1x)	Magnetómetro de 3 ejes (HMC5843). Giroscopio de 3 ejes (MPU-3300). Señal PPS. (2x) sensores de temperatura. Procesador FM33256B.
<i>OBC C3S</i>					
<i>KRYTEN-M3</i>	(30x)			(1x)	(2x) señales PPS. Magnetómetro de 3 ejes. Acelerómetro criptográfico de 3 ejes. Acelerómetro CCSDS de 3 ejes. Sensor de tensión interna embebido. Sensor de corriente interna embebido. (3x) sensores de temperatura embebidos. Modulo GPS (opcional). Sistema DTMF.
<i>CubeComputer Nanosatellite OBC</i>	-	(4x)	(4x)	(1x)	Conector externo para ADCS y carga útil.
<i>OBC EnduroSat</i>	(6x)	(6x)	(3x)	(1x)	(2x) acelerómetros de 3 ejes embebidos. (2x) magnetómetros de 3 ejes embebidos. (3x) terminales SPI para giroscopios. (6x) terminales SPI para sensores de temperatura. Conector para despliegue de antena. Área tipo protoboard.

<i>ABACUS</i>	(79x)	(10x)		(1x)	(8x) canales de controlador integral proporcional generalizado. Magnetómetro de 3 ejes embebido. Acelerómetro de 3 ejes embebido. Giroscopio de 3 ejes embebido. (3x) sensores de temperatura embebidos. Monitor de corriente embebido.
<i>HERCULES</i>	(32x)	(22x)	(6x)	(1x)	(2x) acelerómetros de 3 ejes embebidos. (2x) giroscopios de 3 ejes embebidos. (2x) magnetómetros de 3 ejes embebidos. (3x) sensores de temperatura embebidos. (2x) N2HET de 32 canales. Conector para daughterboards.
<i>N-ART OBCOMMS V2</i>				(1x)	Modulaciones 2FSK, 4FSK, CW y GMSK.
<i>IMT CubeSat OBC</i>		(32x)	(4x)	(1x)	Interfaz de cámara de a bordo. Circuito comparador analógico. Circuito de muestreo y retención dual. Comparador front-end dual analógico. Unidad de micro procesamiento integrada. Sensores de temperatura embebidos.
<i>SatBus 3C2</i>	(5x)	(8x)	(3x)		Señal PPS. Controladores para torques magnéticos. Sensores de tensión y corriente. (3x) giroscopios de 3 ejes (BMG160). Magnetómetro de 3 ejes (MMC5883MA). (2x) Radio digital UHF SAT2RF1-1D.
<i>NANOobc Gen2</i>	(8x)	(1x)	(1x)	(1x)	(3x) entradas de reloj. Sensores de corriente. Sensores de tensión. Sensores de temperatura. Receptor GPS con PPS.

Conclusiones: Las interfaces eléctricas principales que pueden encontrarse en prácticamente cualquier sistema son las terminales GPIO, los ADC, las salidas PWM y un RTC. Además, muchos de los fabricantes han optado por implementar unidades de medición inercial (IMU, por sus siglas en inglés) como acelerómetros, giroscopios y magnetómetros para cumplir con tareas fundamentales del ADCS, así como también sensores de temperatura, corriente y tensión que sirvan para obtener telemetría e implementar técnicas de protección y tolerancia a fallas.

### **Interfaces de comunicación serial**

Similar a las interfaces eléctricos y periféricos, las interfaces de comunicación serial puertos de entrada/salida que permiten la transferencia serial de datos a través de topologías tipo bus. Las principales interfaces de comunicación serial que se localizaron en los SCMI comerciales de propósito general se presentan en la *Tabla 4*.



Tabla 4. Interfaces de comunicación serial de los SCMI comerciales

SCMI	CAN	I <sup>2</sup> C	UART	USART	RS232	RS422	RS485	SPI	QSPI	USB	LVDS	M-LVDS
<i>IOBC</i>		(1x)	(2x)		(2x)	(1x)	(1x)	(2x)		(2x)		
<i>NanoMind A3200</i>	(1x)	(3x)	(1x)	(2x)				(2x)				
<i>OBC C3S</i>	(2x)											(4x)
<i>KRYTEN-M3</i>	(1x)	(2x)	(8x)			(1x)	(2x)	(2x)	(2x)		(2x)	
<i>CubeComputer Nanosatellite OBC</i>	(1x)	(2x)	(2x)					(1x)				
<i>OBC EnduroSat</i>	(1x)	(2x)	(1x)	(2x)				(2x)		(1x)		
<i>ABACUS</i>		(2x)			(4x)	(1x)	(1x)	(1x)			(1x)	
<i>HERCULES</i>	(2x)	(1x)	(2x)		(2x)	(1x)	(1x)	(1x)				
<i>N-ART OBCOMMS V2</i>	(1x)	(1x)	(1x)					(1x)				
<i>IMT CubeSat OBC</i>	(2x)	(2x)	(2x)					(2x)		(1x)		
<i>SatBus 3C2</i>	(2x)	(4x)	(1x)	(1x)				(3x)	(1x)	(1x)		
<i>NANOobc Gen2</i>	(2x)	(1x)	(1x)					(1x)			(2x)	

Conclusiones: Los buses presentes en prácticamente todo SCMI son CAN, I<sup>2</sup>C, UART y SPI, seguidos del USB, aunque su integración no es tan frecuente, mientras que los demás dependen de las características y aplicaciones adicionales que agregue cada fabricante a su diseño.

### Propiedades físicas y eléctricas

Dentro de las propiedades físicas se consideran el factor de forma, las dimensiones, masa, temperatura de operación y la resistencia a TID, mientras que las propiedades eléctricas se obtienen principalmente la tensión de alimentación y la energía promedio y máxima (si se especifica) que consume el sistema en operación. Los resultados obtenidos se presentan en la *Tabla 5*.

Tabla 5. Propiedades físicas y eléctricas de los SCMI comerciales

<i>SCMI</i>	Factor de forma	Dimensiones mm	Masa g	Temperatura de operación °C	Tensión de alimentación V	Consumo de energía promedio mW	Resistencia a la TID krad
<i>IOBC</i>	PC/104	96 x 90 x 12.4	106	-25 a +65	3.3	380	
<i>NanoMind A3200</i>	GOMspace Daughter-board	65 x 40 x 7.1	24	-30 a +85	3.3	170 – 900	
<i>OBC C3S</i>	-	92 x 89.5 x 12.3	65			460	
<i>KRYTEN-M3</i>	PC/104	95.9 x 90.2 x 10.2	68	-40 a +80	3.3	610	
<i>CubeComputer Nanosatellite OBC</i>	PC/104	96 x 90 x 10	56	-10 a +70	3.3	200	20
<i>OBC EnduroSat</i>	PC/104	95 x 89 x 23.2	58	-30 a +85	3.3		
<i>ABACUS</i>	PC/104	95.86 x 90.14 x 23.24	62	-40 a +85	5	50 – 800	
<i>HERCULES</i>	PC/104		65	-40 a +85	5 / 3.3		
<i>N-ART OBCOMMS V2</i>	PC/104	93 x 86 x 15	88				
<i>IMT CubeSat OBC</i>	PC/104	96 x 90 x 10	38	-20 a +70	3.3	300	15
<i>SatBus 3C2</i>	PC/104	95.5 x 90.17 x 15.12	62	-40 a +85	3.3	330	20
<i>NANOobc Gen2</i>	PC/104 sin conector	94.9 x 90.2 x 11.1	63	-10 a +60	5	<1000 – 1500	10

Conclusiones: De acuerdo con los resultados obtenidos, el factor de forma presente en la mayoría de los SCMI es el PC/104. Además, algo destacable de los sistemas es que su consumo de energía no sobrepasa los 1000 mW, esto debido a que deben de ser sistemas muy eficientes en términos de energía porque el suministro de ella es muy limitado en el espacio, lo que lo hace un factor fundamental para implementar un CubeSat.

#### **Técnicas de mitigación y tolerancia a fallas**

Las técnicas de mitigación y tolerancia a fallas son aquellos métodos o estrategias implementadas para aumentar el nivel de confiabilidad de un sistema, aun con la presencia de fallas. Los resultados obtenidos se presentan en la *Tabla 6*.

Tabla 6. Técnicas de mitigación y tolerancia a fallas de los SCMI comerciales

<i>SCMI</i>	<b>Temporizador Watchdog</b>	<b>TMR</b>	<b>EDAC</b>	<b>Otros</b>
<i>IOBC</i>	(1x)			PIC supervisor de energía. (2x) ranuras para tarjetas SD redundantes, operando alternadamente.
<i>NanoMind A3200</i>	(1x)			Procesador adicional con tareas críticas específicas.
<i>OBC C3S</i>	(2x)			CAN redundante. M-LVDS redundante.
<i>KRYTEN-M3</i>	(1x)	✓	✓	EDAC en memorias Flash y MRAM. Limitador LCL
<i>CubeComputer Nanosatellite OBC</i>	(2x)		✓	
<i>OBC EnduroSat</i>				
<i>ABACUS</i>				Redundancia en CPU con núcleos diferentes.
<i>HERCULES</i>			✓	Repuesto caliente de reserva en CPU. 2x CPU con diferente orientación. ECC en memorias Flash y SRAM. Bit de paridad en memorias.
<i>N-ART OBCOMMS V2</i>	(1x)			Protección contra picos de tensión y corriente.
<i>IMT CubeSat OBC</i>	(2x)		✓	TMR en memorias Flash. Redundancia en hardware.
<i>SatBus 3C2</i>				Redundancia en hardware.
<i>NANOobc Gen2</i>	(1x)	✓	✓	Técnicas FDIR. Redundancia en hardware. Redundancia en información. (2x) CRC en memoria del programa. CAN de repuesto caliente y frío de reserva.

Conclusiones: En la *Tabla 6* se clasifican de manera superficial las técnicas de mitigación y tolerancia a fallas localizadas en el estado del arte, claro está que los fabricantes no comparten la información completa ni detallada sobre su diseño, pero los resultados obtenidos servirán como punto de partida para desarrollar un diseño propio. Entre las técnicas más utilizadas se encuentran los temporizadores watchdog, diferentes esquemas de

redundancia destacando la TMR y la aplicación de software EDAC. Todo lo anterior, combinado con la selección de componentes tolerantes a la radiación, abre muchas posibilidades y variantes a la hora de diseñar un sistema tolerante a fallas.

### 3.3 Estado del arte de los SCMI de investigación

Los SCMI anteriores se presentaron con el objetivo de identificar el tipo de tecnología, dispositivos y características presentes en los diseños comerciales de manera general, ya que, al ser sistemas comerciales, el fabricante no revela la descripción a detalle del proceso con el que se desarrollaron, así como las técnicas con las que logran el sistema sea tolerante a fallas, requisito indispensable para el diseño de un sistema satelital.

Con el fin de investigar a detalle los procesos de diseño, en esta parte se documenta el estado del arte del diseño de SCMI desarrollados por centros de investigación y presentados en algunas publicaciones que describen el proceso de diseño de diferentes sistemas. Para este caso, se toma principal atención en las tecnologías, componentes, arquitecturas, técnicas de tolerancia a fallas y metodologías aplicadas en sus sistemas, de modo que sirvan como base para que el SCMI diseñado en este trabajo se acerque a los niveles con confiabilidad aceptables para desempeñar misiones espaciales. De estas publicaciones se describe brevemente hacia donde van dirigidos los sistemas, los tipos de misiones para los cuales fueron diseñados y, principalmente, las técnicas de diseño empleadas para su desarrollo.

#### 3.3.1 Computadora dentro del presupuesto tolerante a fallas para nanosatélites [16]

**Título original:** Fault Nanosatellite Computing on a Budget.

**Autores:** Christian M. Fuchs, Nadie M. Murillo, Pai Chou, Jing-Jia Liou, Yu-Min Cheng, Xiaoqing Wen, Stefan Holst, Antonios Tavoularis, Gianluca Furano, Giorgio Magistrati, Kostas Marinis, Shyue-Kung Lu, Aske Plaat.

**Lugar de publicación:** 33th Annual AIAA/USU Conference on Small Satellites, Session VIII: Advanced Concepts II.

**Año de publicación:** 2019.

**Disponible en:** <https://digitalcommons.usu.edu/smallsat/2019/all2019/135/>

En esta publicación se presenta el diseño de una arquitectura de OBC compatible con el estándar CubeSat tolerante a fallas para misiones críticas y de larga duración, desarrollada únicamente con componentes COTS, IP de bibliotecas de proveedores estándar y la ejecución de software y sistema operativo (OS, por sus siglas en inglés) estándar. La principal característica del sistema es que no introduce las limitaciones de diseño de las soluciones tradicionales de redundancia en hardware, como la TMR, sino que alcanza la tolerancia a fallas a través de software.

La OBC utiliza un FPGA como plataforma de procesamiento principal, en donde se implementa un sistema de multiprocesador en chip (MPSoC, por sus siglas en inglés), y un microcontrolador adicional que actúa como watchdog, realizando tareas de depuración y diagnóstico, conocido como supervisor. El supervisor no participa en las operaciones normales de procesamiento de datos, por lo que su consumo de potencia es mínimo, y únicamente se encuentra conectado al FPGA mediante una interfaz SPI, con la que realiza diagnósticos de bajo nivel en diferentes partes del MPSoC y a los componentes conectados al FPGA; y mediante GPIO para controlar la interfaz JTAG del FPGA y así poder restablecerlo en caso de que pierda funcionalidad ante la presencia de fallas, además de compartir el acceso a la memoria de configuración del FPGA con una interfaz SPI multimaestro para reconfigurar al FPGA de forma independiente. Como supervisor, utilizan un microcontrolador robusto de bajo rendimiento MSP430FR5969, cuya familia es fabricada con FRAM inherentemente tolerante a la radiación en lugar de tecnología Flash, por lo que se ha vuelto popular en CubeSats diseñados a través de COTS de bajo rendimiento gracias a su buen desempeño bajo un ambiente de radiación [17].

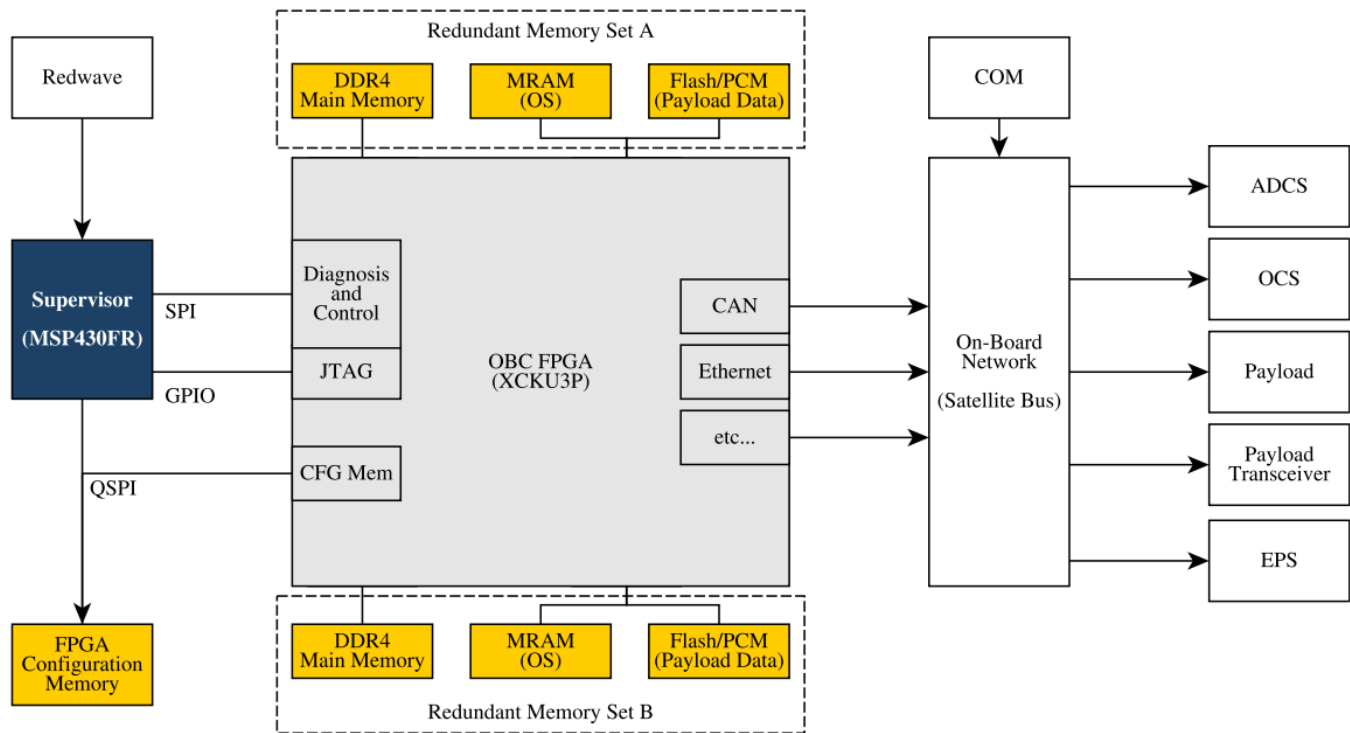


Figura 29. Diagrama a nivel de componente de la arquitectura de OBC [16]

Además de la memoria de configuración del FPGA, se implementaron dos conjuntos redundantes de memorias para el uso del MPSoC. Cada conjunto incluye una memoria de doble tasa de datos (DDR, por sus siglas en inglés) SDRAM como memoria volátil principal del MPSoC, MRAM para almacenar el OS y el software de vuelo y una memoria de cambio de fase (PCM, por sus siglas en inglés) para el almacenamiento masivo de datos de la carga útil. Sin embargo, por limitaciones de hardware se sustituyeron la MRAM y la PCM por NAND Flash, además de que no se implementó la conmutación por error en el hardware. En la *Figura 29* se presenta el diagrama a nivel de componentes de la arquitectura de la OBC.

### 3.3.2 Mitigación a nivel de sistema de SEFI en arquitecturas de manejo de datos, una solución para satélites pequeños [18]

**Título original:** System-Level Mitigation of SEFIs in Data Handling Architectures, A Solution for Small Satellites.

**Autores:** Shazia Maqbool.

**Lugar de publicación:** AIAA/USU Conference on Small Satellites, Technical Session VII: 13th Annual Frank J. Redd Student Scholarship Competition.

**Año de publicación:** 2005.

**Disponible en:** <https://digitalcommons.usu.edu/smallsat/2005/all2005/39/>

En este trabajo se presenta una arquitectura tolerante a fallas para sistemas espaciales dirigida principalmente a satélites pequeños basados en componentes COTS, en donde se diseña un sistema para el manejo de datos de a bordo.

El diseño propone la implementación de un sistema de supervisión inteligente de a bordo conectado a una red de manejo de datos, el cual detecta fallas, determina su naturaleza y aplica el procedimiento de recuperación apropiado para recuperar el sistema a un estado funcional sin necesidad de una intervención inmediata de una estación terrestre. Para lograr esto, el supervisor debe ser un dispositivo endurecido a la radiación, lo cual no es una limitante debido a que no realizará las tareas que requieren de un alto rendimiento y desempeño como el

procesador principal, además de que permitirá que el resto de los subsistemas puedan estar basados en componentes COTS.

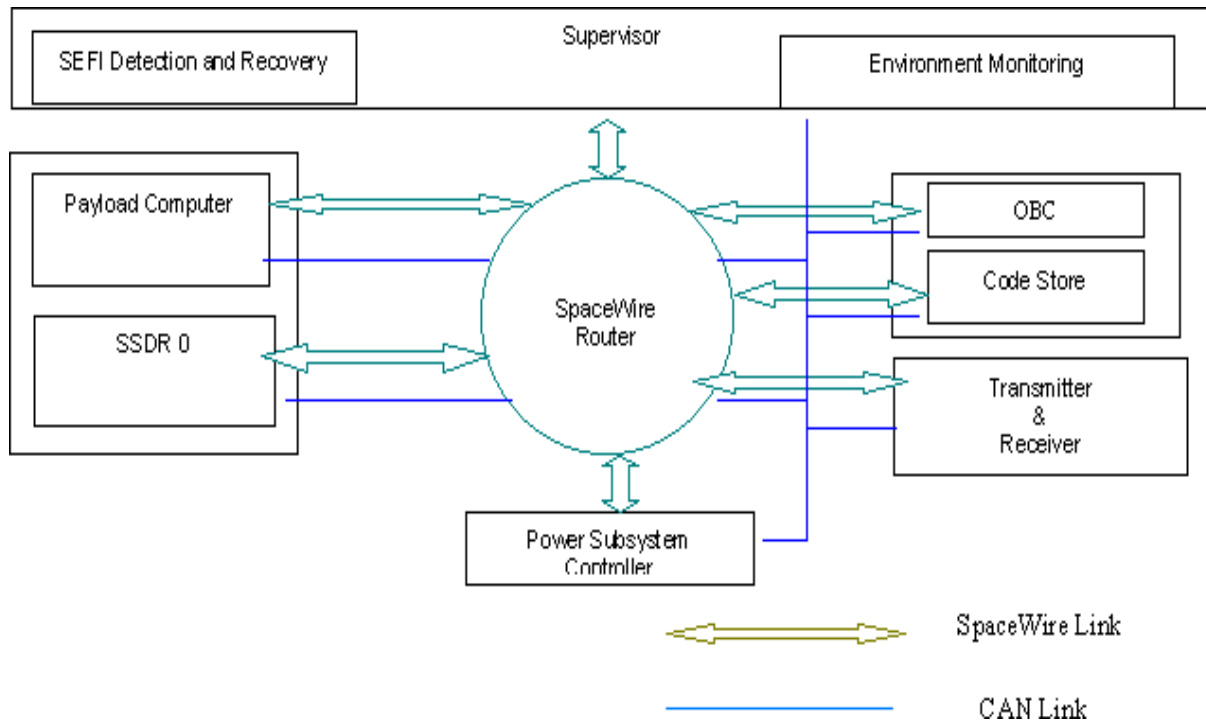


Figura 30. Ejemplo de arquitectura del sistema de manejo de datos de a bordo [18]

El supervisor se encarga de monitorear las tareas, el estado y el comportamiento de los otros dispositivos, lo que genera la necesidad de una comunicación rápida y confiable. Para esto, se utiliza una red de manejo de datos SpaceWire, que proporciona una transferencia de datos rápida a bajo consumo de energía, escalabilidad en el número de nodos y un alto grado de aislamiento entre los sistemas. El estándar SpaceWire describe el hardware y software necesarios para la implementación del protocolo, incluyendo funciones de tolerancia a fallas y es utilizado para enviar mensajes de estado, comandos de diagnóstico y bloques de datos entre los subsistemas. Además, se utiliza un bus CAN independiente para proporcionar un comando redundante y una función de control de configuración, como se muestra en *Figura 30*.

### 3.3.3 Implementación de un RTOS de código abierto para una computadora de a bordo (OBC) en STUDSAT-2 [19]

**Título original:** Open Source RTOS Implementation for On-Board Computer (OBC) in STUDSAT-2.

**Autores:** Bheema Rajulu, Sankar Dasiga & Naveen R. Iyer.

**Lugar de publicación:** 2014 IEEE Aerospace Conference.

**Año de publicación:** 2014.

**Disponible en:** <https://ieeexplore-ieee-org.pbidi.unam.mx:2443/document/6836377>

En este trabajo se presenta el diseño del SCMI para el satélite indio STUDSAT-2, que consiste en dos nanosatélites, cada uno siendo un CubeSat con dimensiones de 30\*30\*20 cm y una masa menor a 10 kg. En el trabajo destaca la selección del procesador de a bordo y los dispositivos de almacenamiento, además de presentar la justificación técnica del RTOS y su implementación en el procesador.

Se seleccionó el procesador de la familia ARM Cortex-M4 STM32F407VGT6 como procesador principal, esto debido a sus características de bajo consumo de energía y alta capacidad de rendimiento. Este procesador cumple con los requisitos de funcionalidad y operación, además de contar con las interfaces y protocolos necesarios para el STUDSAT-2, como lo son GPIO, PWM, USART, SPI e I<sup>2</sup>C, y posee con una amplia variedad de

características de procesamiento de señales altamente eficiente aplicables en el mercado de control digital de señales y las necesarias para el satélite.

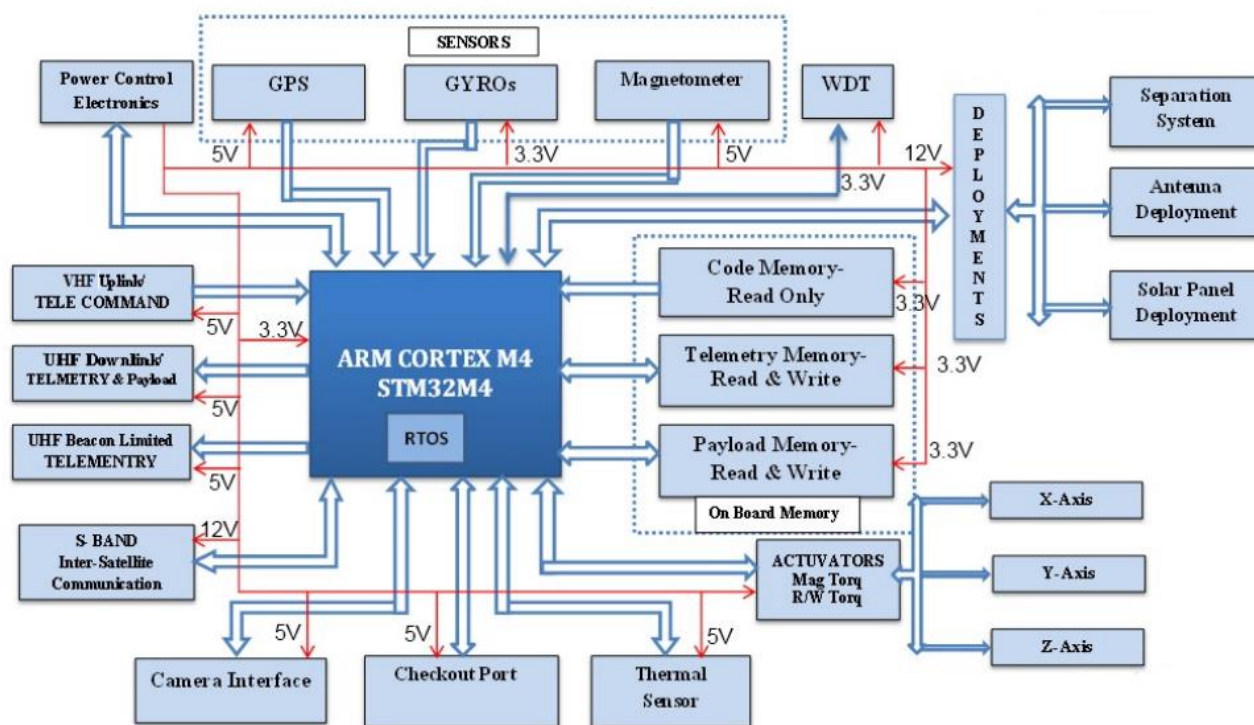


Figura 31. Estructura general de la OBC [19]

Por parte de los dispositivos de almacenamiento, el SCMI cuenta con tres memorias para el almacenamiento del programa: EEPROM, Flash y SRAM. La memoria Flash es utilizada para almacenar el software principal de operación y otras subrutinas, respaldando una copia de este código en la EEPROM (24LC02) que, además, cuenta con el software de arranque para reiniciar al procesador cuando el temporizador watchdog detecte un error en su operación. Es importante mencionar que la memoria Flash es susceptible a errores de alteración de bits inducidos por la radiación. Por su parte, los 512k de SRAM se utiliza como memoria de trabajo, corriendo el software y almacenando los valores obtenidos por los sensores de la carga útil. En cuanto a la memoria de datos, se seleccionó una memoria NOR Flash (AT45DB161D). La estructura general de la OBC se presenta en la *Figura 31*.

En cuanto al software se refiere, se implementa FreeRTOS como sistema operativo en tiempo real. La selección fue basada en uno de los requisitos fundamentales del trabajo: el bajo costo. FreeRTOS es un RTOS de código abierto que ofrece todas las características necesarias adecuadas para el proyecto con un tamaño accesible para el procesador, que requiere de un RTOS pequeño, además de presentar un nivel aceptable de rendimiento en comparación con otros RTOS, sin embargo, la disponibilidad de su documentación para distintas arquitecturas de microcontrolador, su alto soporte técnico y la gran comunidad de usuarios que lo utilizan, convierten a FreeRTOS en una opción más atractiva, sin mencionar que al ser de código abierto, es totalmente gratuito para el usuario.

### 3.3.4 Encuesta sobre la implementación y fiabilidad de las interfaces eléctricas de bus en CubeSat [20]

**Título original:** Survey on the implementation and reliability of CubeSat electrical bus interfaces.

**Autores:** Jasper Bouwmeester, Martin Langer & Eberhard Gill.

**Lugar de publicación:** CEAS Space Journal (2017) 9:163-173

**Año de publicación:** 2016

**Disponible en:** <https://link.springer.com/article/10.1007/s12567-016-0138-0>

Esta publicación proporciona los resultados y conclusiones de una encuesta en los aspectos de implementación y fiabilidad de las interfaces de bus CubeSat, dando énfasis a los buses de datos, la distribución de la potencia y el factor de forma. La encuesta está basada en el estudio de la literatura y un cuestionario representando 60 CubeSat lanzados y 44 por lanzarse.

De acuerdo con un estudio de 129 satélites de todas las clases [21], al menos 45% presenta daños por fallas eléctricas, de los cuales 72% son por parte del sistema eléctrico de potencia (EPS, por sus siglas en inglés) y 15% del SCMI. Ambas cantidades juntas tienen un gran impacto en la fiabilidad del satélite.

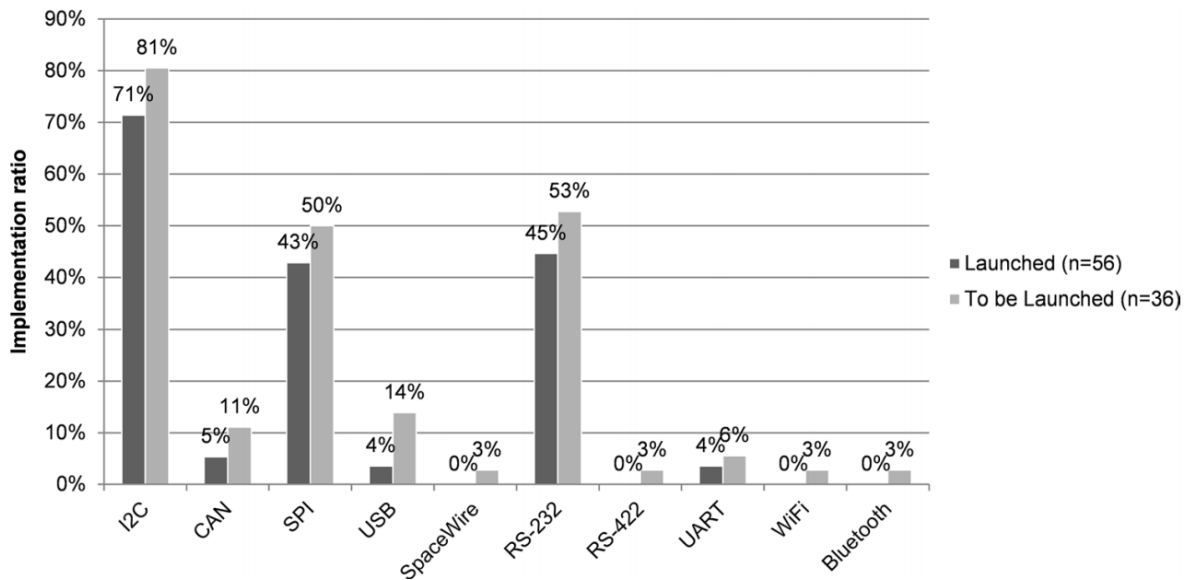


Figura 32. Buses de datos implementados en CubeSat [20]

Por parte de los buses de datos, en la *Figura 32* se observan los tipos de buses implementados en CubeSat, de donde destaca el uso de I<sup>2</sup>C. Esto se explica por el hecho de que muchos circuitos integrados COTS tienen un controlador I<sup>2</sup>C interno, por lo que se utiliza en la mayoría de los subsistemas CubeSat basados en componentes COTS. Además, también se utilizan frecuentemente SPI y RS232, agregando que el bus CAN se implementa en pocos CubeSat, pero se estima que aumentará su popularidad en el futuro.

De los buses de datos implementados, SpaceWire es el único diseñado para aplicaciones espaciales. Esto indica que para garantizar un buen nivel de fiabilidad en los buses más utilizados es necesario aplicar técnicas de tolerancia a fallas. La *Figura 33* y *Figura 34* presentan, respectivamente, las técnicas más utilizadas por desarrolladores de CubeSat y los problemas en órbita reportados en I<sup>2</sup>C, SPI y RS232.



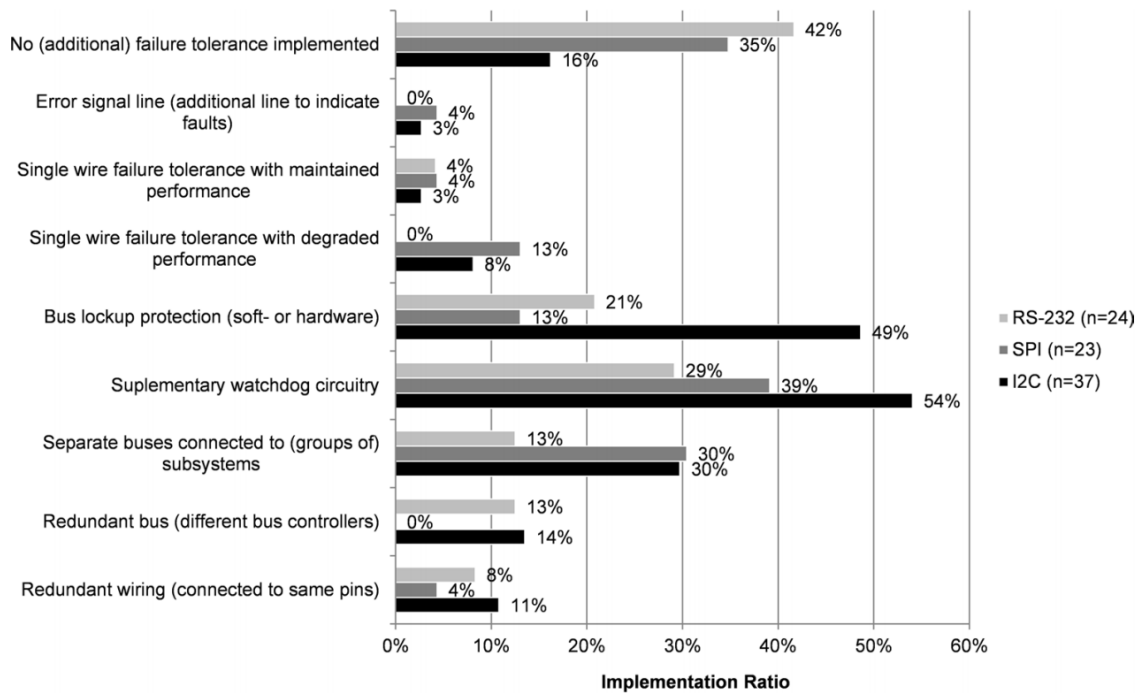


Figura 33. Técnicas utilizadas por desarrolladores en los buses de datos I<sup>2</sup>C, SPI y RS232 [20]

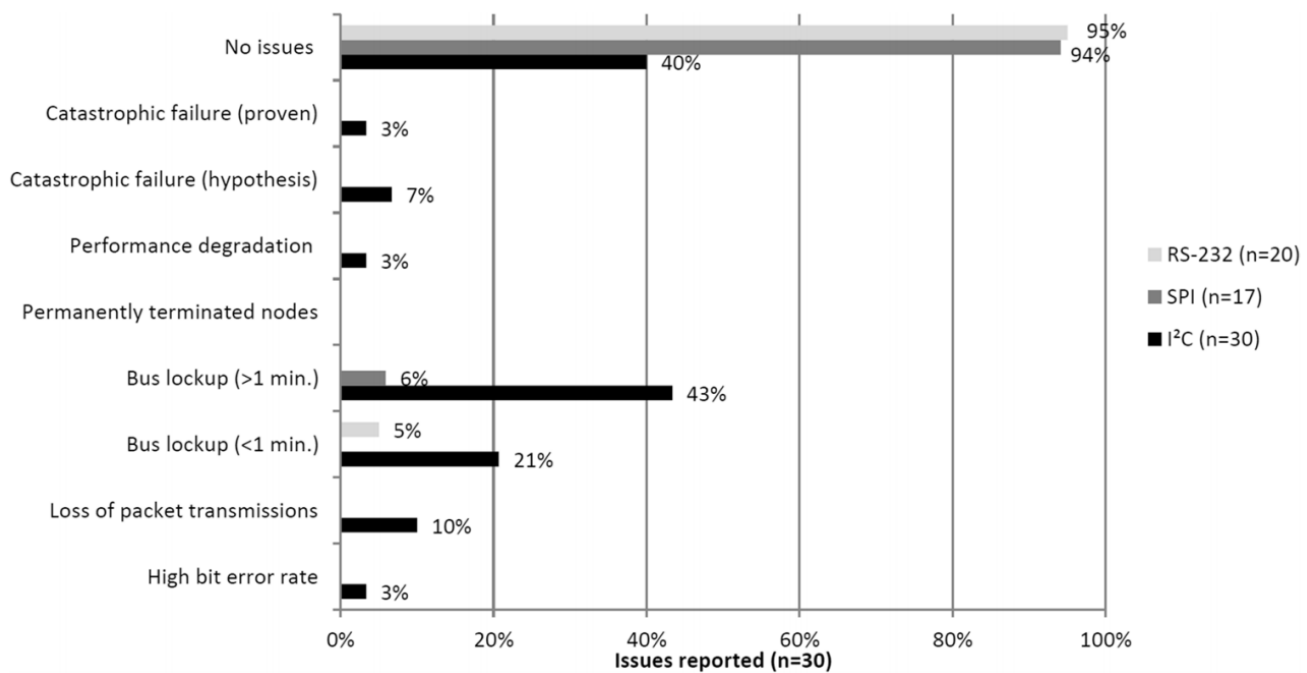


Figura 34. Problemas en órbita reportados en los buses de datos I<sup>2</sup>C, SPI y RS232 [20]

De lo anterior se observa claramente que RS232 presenta la menor cantidad de fallas y técnicas de tolerancia a fallas implementadas, pero únicamente permite conectar dos nodos en su bus, por lo que comúnmente es complementado con uno o más de las otras interfaces.

Por otra parte, el conector PC/104 se ha convertido en el factor de forma más empleado por desarrolladores CubeSat. En muchos aspectos, el estándar PC104 no es correctamente implementado. Los proveedores de subsistemas CubeSat típicamente implementan I<sup>2</sup>C, cuando el bus de datos definido en el estándar es el bus de industria estándar de arquitectura (ISA, por sus siglas en inglés) [35]. Además, la topología de asignación y distribución de energía no se asume ni se estandariza para CubeSats, lo que genera problemas de compatibilidad [22]. Por lo

tanto, cuando se menciona PC/104 como estándar en relación con CubeSats, esto se refiere al conector físico y al diseño mecánico y no al bus de datos o la asignación de pines.

En la *Figura 35*, se observan los problemas reportados por desarrolladores con el conector PC/104, de donde se concluye que es recomendable buscar un conector más pequeño y con menor cantidad de terminales que el descrito en el estándar PC/104.

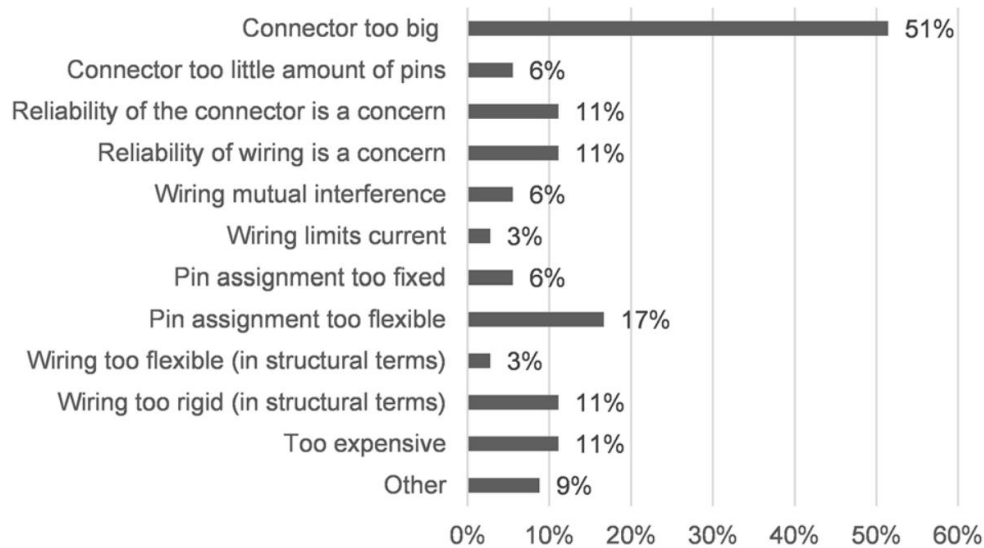


Figura 35. Problemas reportados en el conector PC/104 (n=36) [20]

### 3.3.5 Estandarización de la interfaz eléctrica de CubeSat para una entrega más rápida y más éxito en la misión [23]

**Título original:** CubeSat Electrical Interface Standardization for Faster Delivery and More Mission Success.

**Autores:** Mengu Cho, Takashi Yamauchi, Marloun Sejera, Yukihiisa Ohtani, Sangkyun Kim, Hirokazu Masui.

**Lugar de publicación:** AIAA/USU Conference on Small Satellites, Pre-Conference Workshop Session I: Advanced Concepts I.

**Año de publicación:** 2020

**Disponible en:** <https://digitalcommons.usu.edu/smallsat/2020/all2020/3/>

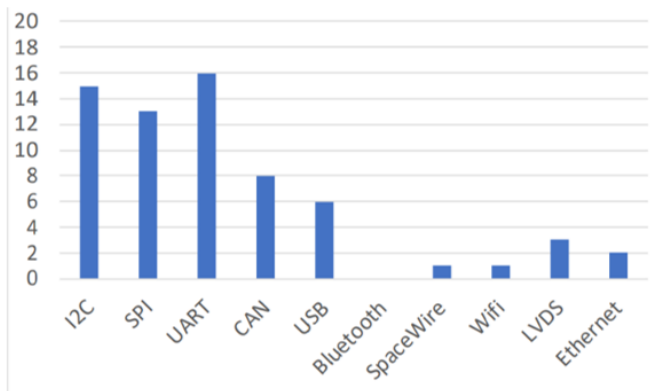
El trabajo consiste en un proyecto para estandarizar la interfaz eléctrica de CubeSat basándose en la experiencia adquirida por los desarrolladores y proveedores de CubeSat, planeado para publicarse en 2024. Para esto, se realizaron encuestas aplicadas en diferentes reuniones de la comunidad CubeSat, las cuales recopilan los comentarios de la comunidad con respecto a la interfaz. Más específicamente, el objetivo de esta publicación es proporcionar una descripción general y el estado del proyecto presentando los resultados de encuestas aplicadas a la comunidad de desarrolladores y proveedores de CubeSat, los cuales son de utilidad para conocer el estado del arte de los deseos de la comunidad y dar mayor fundamento en las decisiones que se tomen en la etapa de diseño.

Las principales ventajas del estándar CubeSat son los cortos tiempos de desarrollo a bajos costos, sin embargo, las interfaces eléctricas de los diferentes proveedores CubeSat no siempre son compatibles, incluso si utilizan el estándar PC/104. La incompatibilidad incrementa el tiempo de desarrollo, ensamble e integración del satélite, además de que puede requerir una tarjeta adicional de interfaz o un mecanismo que facilite su construcción, a cambio de una mayor complejidad en el diseño y complicaciones con los cumplimientos de volumen y masa limitados del CubeSat. Para combatir la incompatibilidad se necesita una buena interfaz. La mayoría de los componentes CubeSat disponibles en el mercado utilizan el factor de forma PC/104, del cual, más del 50% de

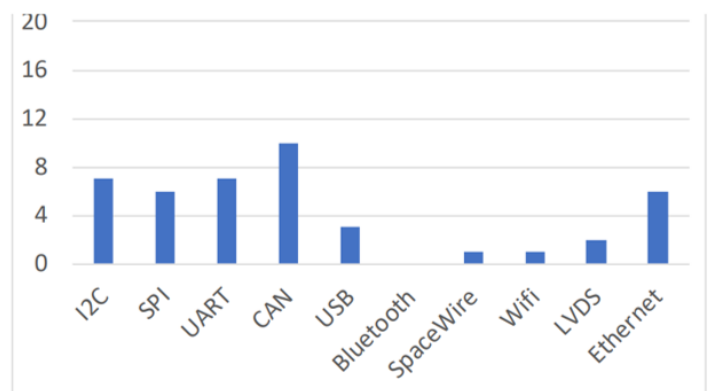
integradores CubeSat creen que el conector es demasiado grande y el 17% explica que la asignación de pines es demasiado flexible [20].

Como plataforma, el proyecto utiliza la comunidad de satélites ajustados (lean satellite), que se refiere al desarrollo de satélites pequeños utilizando enfoques de desarrollo y gestión no tradicionales y asumiendo riesgos con el objetivo de proporcionar algún tipo de valor al cliente a bajo costo y tiempo de desarrollo (más información disponible en: <https://lean-sat.org/>).

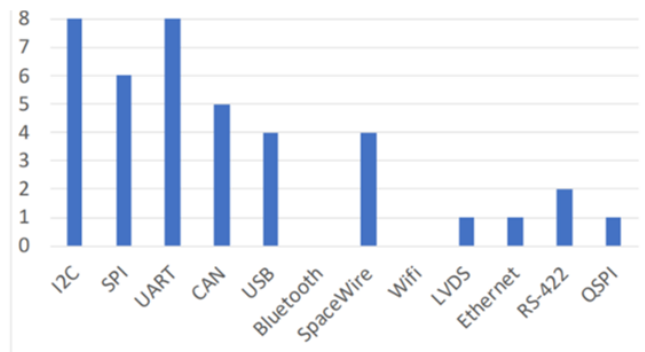
Además del factor de forma, se realizó una encuesta en la International Workshop on Lean Satellite 2019 (IWLS-2019), donde 50 encuestados (21 desarrolladores, 9 proveedores y otros 20) respondieron qué buses de comunicación serial son los más utilizados y cuales se estima que aumentará su uso en un futuro. Estos resultados se muestran en la *Figura 36*, en donde se observa que I2C, SPI y UART son altamente utilizadas, mientras que se espera un aumento del uso de CAN y Ethernet por parte de los desarrolladores y de SpaceWire para los proveedores.



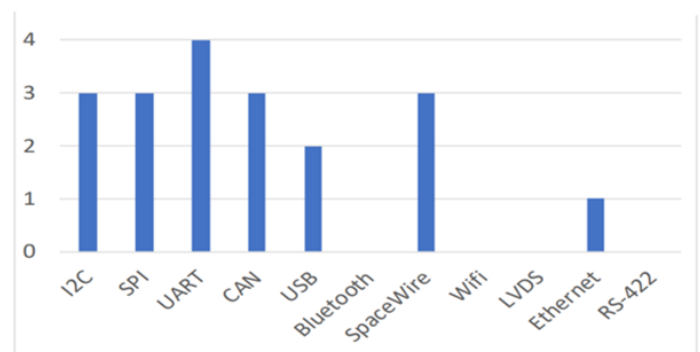
a) Buses de comunicación serial más utilizados por desarrolladores



b) Predicción de los buses de comunicación serial más utilizados a futuro por desarrolladores



c) Buses de comunicación serial más utilizados por proveedores



d) Predicción de los buses de comunicación serial más utilizados por proveedores

Figura 36. Buses de comunicación serial utilizados en CubeSat, según desarrolladores [23]

## *Sección III*

---

# *Marco teórico*

## 4 Satélites artificiales

Se le conoce como satélite artificial a cualquier objeto construido por seres humanos que orbita un planeta o cuerpo celeste con un objetivo en específico denominado misión. Estos artefactos son construidos en la Tierra y lanzados al espacio a través de un medio de transporte, como lo son misiles, naves espaciales y aeronaves. Al terminar su vida útil, los satélites artificiales pueden permanecer en órbita como basura espacial o programarse para regresar a tierra y, en algunos casos, desintegrarse al reingresar a la atmósfera terrestre.

Los satélites artificiales han sido un factor fundamental para el progreso de las telecomunicaciones, el conocimiento del planeta y la exploración espacial. Actualmente, un gran número de satélites se encuentran orbitando la Tierra desempeñando tareas, experimentos y servicios para la humanidad, además de permitirle a la comunidad científica el desarrollo de nuevas tecnologías y áreas de aplicación en la carrera espacial. Algunas de las aplicaciones más conocidas que realizan los satélites son la televisión y la radio satelital, los sistemas de comunicación, geolocalización y navegación por satélite y las mediciones del clima y medio ambiente terrestre para monitorear y predecir el comportamiento del planeta, entre otras.



Figura 37. Satélite artificial

### 4.1 Antecedentes históricos

La era espacial comenzó durante la guerra fría, en donde los Estados Unidos y la Unión Soviética se encontraban en un enfrentamiento político, económico, social, militar y científico en donde ambas naciones pretendían convertirse en la primera potencia del mundo. Durante este enfrentamiento, naturalmente, surgió el interés de llegar al espacio exterior, impulsando una nueva etapa llena de avances científicos y tecnológicos.

Antes de la guerra, únicamente se estudiaba la ionósfera<sup>22</sup> a través de ondas de radio y se realizaban mediciones a una altura máxima de 30 km sobre la superficie terrestre con globos estratosféricos<sup>23</sup>. En 1946, con los cohetes alemanes V-2 capturados como resultado de la segunda guerra mundial, se alcanzaron los 200 km de altura realizando mediciones de temperatura, densidad y presión.

El primer viaje exitoso al espacio lo realizó la Unión Soviética el 4 de octubre de 1957 al lanzar el *Sputnik I*, satélite artificial de 83.6 kg que logró posicionarse correctamente en una órbita elíptica de 938 km de apogeo<sup>24</sup> y 214 km de perigeo<sup>25</sup>, enviando telemetría a tierra durante los siguientes 22 días hasta que se desintegró al reingresar



Figura 38. Satélite "Sputnik I", primer satélite artificial lanzado por la Unión Soviética

<sup>22</sup> Parte de la atmósfera terrestre ionizada permanentemente debido a la fotoionización que provoca la radiación solar (situada aproximadamente entre los 80 y 400 km de altura sobre la superficie terrestre).

<sup>23</sup> Es todo globo que puede volar de forma estable en la estratosfera (situada aproximadamente entre los 10 y 50 km de altura sobre la superficie terrestre).

<sup>24</sup> Punto más lejano de una órbita elíptica de un satélite artificial con respecto al centro de la Tierra.

<sup>25</sup> Punto más cercano de una órbita elíptica de un satélite artificial con respecto al centro de la Tierra.



Figura 39. La Tierra rodeada de miles de satélites artificiales [27]

a la atmósfera terrestre 92 días después de su lanzamiento [24] [25]. Tres años después del programa *Sputnik*, la Administración Nacional de la Aeronáutica y del Espacio (NASA, por sus siglas en inglés) lanzó el primer satélite de comunicaciones pasivo, el *Echo I*, cuyo propósito fue reflejar las señales provenientes de tierra y no fue hasta 1962 que se colocó en órbita el primer satélite de comunicaciones activo, el *Telstar I*, que realizó la primera transmisión de televisión internacional en vivo. Con estos logros dio inicio una era de avances y desarrollo científico y tecnológico que con el pasar de los años generó un aumento en el número de satélites orbitando la Tierra. Desde el lanzamiento del *Sputnik I* hasta finales de marzo de 2020, según la Oficina de Naciones Unidas para Asuntos del Espacio Exterior (UNOOSA, por sus siglas en inglés), se han registrado un total de 9386 lanzamientos de objetos al espacio, de los cuales 4462 se encuentran actualmente

operando en órbita [26], además de que estima que, hasta la fecha, se han registrado un 87% de todos los objetos lanzados a la órbita terrestre o más allá [27], representando claramente el crecimiento de la industria espacial.

## 4.2 Tipos de satélites artificiales

Desde el comienzo de la era espacial, con el propósito de cumplir con los objetivos de diversas misiones se han desarrollado diferentes tipos de satélites, cada uno integrado con distintas tecnologías y sistemas orientados a obtener un buen desempeño en sus tareas correspondientes.

### *Satélites científicos*

Son aquellos satélites cuya misión consiste en el estudio de aspectos de la Tierra y otros objetos astronómicos con el objetivo de obtener información de ellos y entender mejor su comportamiento. Dentro de este tipo se encuentran satélites atmosféricos, geodésicos, biológicos, astronómicos y astrofísicos. Un ejemplo de sus aplicaciones es el monitoreo de la Tierra para predecir su actividad climática.

### *Satélites de comunicaciones*

Son utilizados para establecer comunicación entre dos o más puntos del planeta reflejando señales de radio, telefonía y televisión mediante arreglos de antenas hacia otras zonas de la Tierra. Son implementados comúnmente en una órbita geoestacionaria<sup>26</sup> (GEO, por sus siglas en inglés) para que no se requiera de modificar la orientación de las antenas y mantener una comunicación ininterrumpida.

### *Satélites de navegación*

Son los satélites que permiten conocer la localización de un dispositivo en casi cualquier punto de la Tierra. Esto se logra implementando una constelación de satélites de forma que puedan cubrir la mayor parte del planeta en todo momento. El ejemplo más común es el sistema global de navegación por satélite (GNSS, por sus siglas en inglés) estadounidense GPS, con 24 satélites distribuidos en 6 órbitas.

<sup>26</sup> Órbita circular localizada en el plano ecuatorial a 35 786 km de altura sobre la superficie terrestre en donde el periodo orbital es igual al periodo de rotación del planeta.

### *Satélites de inteligencia y defensa*

Son satélites desarrollados para fines militares que brindan una gran variedad de servicios, que van desde comunicaciones, recolección de información a través de imágenes, navegación y pronóstico del tiempo hasta aplicaciones de emergencia, ataque y defensa de un territorio contra amenazas externas.

### *Satélites de percepción remota*

También llamados satélites de observación terrestre son utilizados para observar en tiempo real un área determinada. Están equipados con una gran variedad de sensores e instrumentos que ayudan a monitorear el planeta desde el espacio con fines no militares.

## 4.3 Satélites pequeños

Los primeros satélites artificiales construidos fueron pequeños debido a que no se contaba con la infraestructura necesaria para desarrollar un sistema tan demandante, sin embargo, con el pasar de los años se requirió la construcción de satélites cada vez más grandes hasta llegar a gigantescas naves como la Estación Espacial Internacional (ISS, por sus siglas en inglés) con dimensiones de 73 x 109 m (véase la *Figura 40*) y una masa cercana a los 420 000 kg lanzada a finales de 1998, proyecto en donde participaron los Estados Unidos, Rusia, Europa, Canadá, Japón, Italia y Brasil [28]. El gran tamaño de los satélites dificultaba su lanzamiento y su construcción requería de grandes cantidades de recursos, por lo que se comenzaron a buscar alternativas que, acompañadas del progreso tecnológico de las últimas décadas que llevó a la integración de dispositivos electrónicos a muy gran escala<sup>27</sup> (VLSI, por sus siglas en inglés), dieron lugar a la miniaturización de satélites, que consiste en el máximo aprovechamiento posible de las dimensiones de un satélite implementando sistemas con mayor nivel de integración para reducir en gran porcentaje los recursos y tiempos necesarios para su construcción y lanzamiento.



*Figura 40. Estación Espacial Internacional*

Los avances en el desarrollo de satélites pequeños posibilitan la implementación de funciones sofisticadas en espacios muy reducidos disminuyendo también la masa del sistema, además de que al utilizar tecnología VLSI se minimizan los requerimientos de potencia. Estos aspectos generaron una enorme reducción en los tiempos de desarrollo y los recursos necesarios para la construcción y lanzamiento de un satélite pequeño en comparación con los satélites convencionales, abriendo las puertas a la participación de universidades y empresas de países en vías de desarrollo en proyectos espaciales.

La popularidad que ganaron los satélites pequeños y el aumento de su frecuencia de lanzamiento generó nuevas ideas para aumentar sus capacidades reduciendo cada vez más sus dimensiones hasta introducir el concepto de microsátélite, cuyo desarrollo se vio impulsado por los avances en la microelectrónica rebasando por mucho la tecnología espacial convencional, por lo que se adoptaron los componentes COTS, cuya disponibilidad en el mercado disminuyó todavía más los costos y tiempos de desarrollo de satélites artificiales.

El enorme éxito obtenido del desarrollo de satélites pequeños en todas sus variantes generó una nueva clasificación de satélites según su masa. En la *Tabla 7*, se pueden observar las enormes diferencias de masa y costo alcanzadas por el desarrollo de satélites pequeños, los cuales en la actualidad han alcanzado un tamaño más pequeño que una caja de zapatos.

<sup>27</sup> Proceso de crear un circuito integrado compuesto por cientos de miles de transistores en un único chip.

Tabla 7. Clasificación de los satélites según su masa y sus costos [29]

Clase	Masa kg	Costo MUSD
Satélite grande	>1000	>122.08
Satélite mediano	500 – 1000	30.52 – 122.08
Minisatélite	100 – 500	8.54 – 30.52
Microsatélite	10 – 100	1.22 – 8.54
Nanosatélite	1 – 10	0.12 – 1.22
Picosatélite	<1	<0.12

En donde los mini, micro, nano y picosatélites se encuentran dentro de la clasificación de satélites pequeños (satélites con masa menor a 500 kg), convirtiéndose en todo un atractivo para que gobiernos, empresas y universidades de todo el mundo inviertan en el desarrollo de proyectos espaciales.

#### 4.4 Los nanosatélites

Como se observa en la *Tabla 7*, el término de nanosatélite hace referencia a un satélite artificial con una masa localizada entre 1 y 10 kg. Para construir un nanosatélite es necesario aprovechar al máximo el pequeño espacio disponible para implementar los sistemas mecánicos y electrónicos. Los nanosatélites han adquirido alta popularidad gracias a sus bajos costos y han vuelto realidad los lanzamientos múltiples, en donde con un solo vehículo de lanzamiento se colocan decenas de unidades en órbita. Sin embargo, a pesar de las ventajas que ofrece su diminuto tamaño surgen algunas complicaciones en su diseño, como por ejemplo el sistema de propulsión, del cual carecen muchos nanosatélites por la dificultad de almacenar combustible dentro del sistema, por lo que se han aplicado alternativas como velas solares, sistemas desplegados y propulsores iónicos.



Figura 41. Constelación de satélites artificiales

Los nanosatélites pueden llevar a cabo misiones individuales no tan demandantes sin ningún problema, sin embargo, sus capacidades no pueden compararse con un satélite de mayor tamaño, por lo que se ha optado por colocar en órbita un cierto número de ellos trabajando en conjunto formando constelaciones de satélites que pueden cubrir una región o hasta la mayor parte del planeta, como se puede ver en la *Figura 41*. Este método mantiene la diferencia de costos entre un satélite grande con decenas de pequeños y, de igual manera que la tecnología desechable, es más factible sustituir una unidad dañada a reparar o cambiar un sistema de grandes dimensiones.

Dentro de los nanosatélites existen algunas plataformas con diferentes características, como lo son el TubeSat<sup>28</sup>, la Sphere<sup>29</sup> y el CubeSat, siendo este último el más destacado y con el que se trabajará en el presente proyecto describiéndose a detalle a continuación.

<sup>28</sup> Nanosatélite con forma de tubo popular en proyectos académicos por su bajo costo y sencilla construcción, pero energéticamente ineficiente.

<sup>29</sup> Nanosatélite con forma similar a una esfera desarrollado por el Instituto Tecnológico de Massachusetts (MIT, por sus siglas en inglés) presente en el interior de la ISS con el objetivo de monitorear su estado [30].



## 4.5 El estándar CubeSat

La Especificación de diseño CubeSat (CDS) es un estándar de diseño de nanosatélites propuesto en 1999 por los profesores Jordi Puig-Suari y Robert Twiggs de la Universidad Politécnica Estatal de California (Cal Poly) y Stanford respectivamente con el propósito de brindar un acercamiento a estudiantes al área espacial permitiéndoles diseñar, construir, probar y operar un satélite en el espacio con capacidades similares al del primer satélite (*Sputnik*). La propuesta surgió de la necesidad de diseñar un nanosatélite que facilite su construcción aprovechando eficientemente sus dimensiones, fijándose el reto de cumplir una misión al menor costo [2]. Con el paso de los años, el buen desempeño de la propuesta del CubeSat lo convirtió en un estándar.

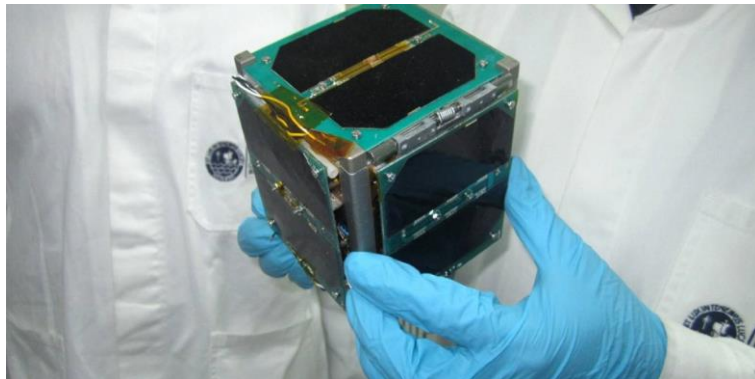


Figura 42. CubeSat

Una unidad CubeSat (U) es una plataforma satelital en forma de cubo con aristas de 10 cm y una masa máxima de 1.33 kg [31], representado en la *Figura 42*, que puede escalarse apilando varias de estas unidades formando sistemas de mayores dimensiones cuyos nombres están dados por el número de unidades que lo conforman (véase la *Figura 43*). El éxito del CubeSat se dio gracias a su forma cúbica, que es perfecta para distribuir eficientemente los sistemas en su interior y colocar paneles solares en sus caras externas para captar energía aprovechando la radiación solar. Las plataformas CubeSat que se han desarrollado van desde 0.25U hasta 27U con masas desde 0.2 hasta los 40 kg [3], pero como se observó en el *Estado del arte* la estructura más utilizada, y con la que se trabajará en el presente proyecto, es el CubeSat 3U.

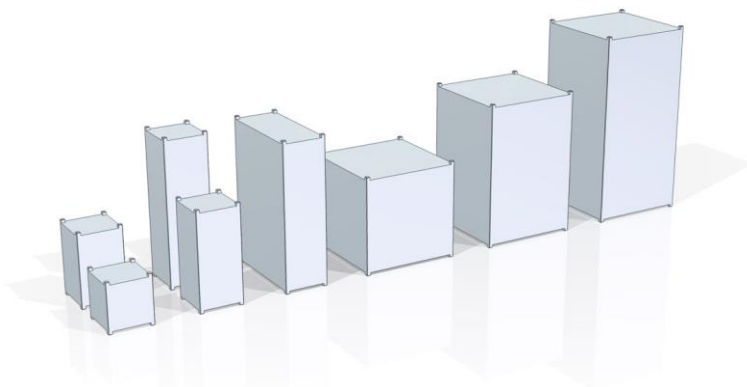


Figura 43. Estructuras CubeSat desde 1U hasta 16U [3]

### 4.5.1 Sistemas de un CubeSat

Un satélite artificial, hablando más específicamente del CubeSat, está compuesto por diferentes sistemas que cumplen tareas específicas. Los sistemas se describen a continuación.

#### *Sistema de Energía Eléctrica (EPS)*

El EPS es el sistema encargado de la generación, almacenamiento y distribución de la energía eléctrica a través de celdas fotovoltaicas, baterías y sistemas de gestión y distribución de energía, respectivamente, lo que lo convierte en el sistema fundamental del satélite.

#### *Sistema de propulsión*

Es el sistema encargado de dar el impulso o empuje necesario a un satélite para moverse en el espacio, puede emplearse para controlar la posición, orientación y órbita del sistema. Existen diferentes tipos de propulsión dependiendo de su fuente de energía y su aplicación.

#### *Sistema de Guía, Navegación y Control (GNC)*

El Sistema de Guía, Navegación y Control (GNC, por sus siglas en inglés) se encarga de determinar y controlar la posición y orientación del satélite, involucra tanto los componentes utilizados para la determinación la posición como los empleados en el ADCS.

### **Sistema de estructura, mecanismos y materiales**

Es el sistema que sostiene todos los sistemas brindándoles soporte y rigidez mecánica necesaria para soportar las fuerzas a las que se somete el satélite desde que despegue y abandona la superficie terrestre.

### **Sistema de regulación térmica**

El sistema de regulación térmica está encargado de mantener la temperatura interna de cada sistema dentro de un rango aceptable para que pueda operar correctamente.

### **Sistema de comando y manejo de información (SCMI)**

El SCMI, OBC o C&DH (por sus siglas en inglés) es el encargado de interpretar, ejecutar y distribuir los comandos descritos por software hacia el resto de los sistemas, además de recopilar, almacenar, administrar y procesar los datos obtenidos de cada uno de ellos. El presente trabajo consiste en el diseño de un SCMI, por lo que será descrito a detalle más adelante.

### **Sistema de comunicaciones**

Es el sistema encargado de establecer comunicación y flujo de información entre el satélite y la estación terrestre o con otro satélite, puede transmitir telemetría recopilada durante la misión o recibir nuevos comandos para modificar y/o actualizar la operación del satélite.

### **Carga útil (payload)**

La carga útil es el punto de partida del diseño de un satélite, consiste en el sistema que llevará a cabo la misión y el que establecerá los requerimientos de diseño para todos los sistemas y el tipo de satélite que podrá satisfacer las necesidades.

### **Sistemas pasivos de desorbita**

Un sistema de desorbita es aquel que permite a un satélite reingresar a la atmosfera terrestre para posteriormente ser destruido por la fricción una vez terminada su vida útil, con el propósito de disminuir el número de escombros en órbita que ha dejado el incremento de lanzamientos al espacio. Se les denomina sistemas pasivos porque no requieren de energía suministrada por el satélite para desempeñar su función como, por ejemplo, velas solares.

## **4.5.2 Estructuras de lanzamiento**

La estandarización del CubeSat facilitó su puesta en órbita mediante una estructura de lanzamiento común para sus diferentes combinaciones de 1 a 3U [2]. La estructura de lanzamiento es el sistema que se implementa en el vehículo que transportará al satélite y se encarga de desplegarlo y colocarlo en su órbita correspondiente. La estructura más utilizada es el sistema de despliegue orbital de picosatélites del Poly (P-POD, por sus siglas en inglés), cuyas características también se describen en el CDS ([31]).

El sistema P-POD consiste en una plataforma con forma de caja rectangular con una compuerta y un mecanismo de resorte en uno de sus extremos (véase la *Figura 44*) sirviendo como interfaz entre el satélite y el vehículo de lanzamiento. El P-POD tiene capacidad para tres unidades estándar CubeSat 1U o una unidad CubeSat 3U y el lanzamiento se habilita mediante una señal del vehículo en donde una serie de resortes de torsión fuerzan el abierto de la compuerta desplegando la unidad CubeSat [31].

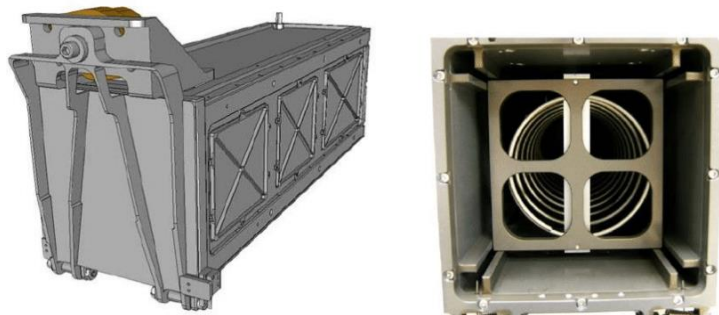


Figura 44. Contenedor P-POD y el resorte de despliegue [31]

La importancia de definir un estándar para la estructura de lanzamiento consiste en reducir los recursos y tiempos que requeriría un diseñador de CubeSat y el equipo de lanzamiento para desarrollar y verificar el funcionamiento de cada propuesta. El

contenedor P-POD es una estructura simple, sencilla de implementar, eficiente y está diseñado para la plataforma de CubeSat más utilizada (3U).

## 4.6 Sistema de comando y manejo de información (SCMI)

El SCMI es en pocas palabras el cerebro de un CubeSat, es considerado el sistema central del satélite porque es el encargado de interactuar con los operadores de la unidad satelital, así como mantener una constante comunicación con cada uno de los sistemas de a bordo. Algunas de las funciones principales del SCMI son las siguientes: [32] [33]

- ✓ Permitir el flujo de datos entre todos los sistemas.
- ✓ Proporcionar la recolección y almacenamiento de datos para todos los sistemas.
- ✓ Recepción, validación, decodificación y distribución de comandos a todos los sistemas.
- ✓ Administración y coordinación de los recursos de cada sistema y las cargas útiles.
- ✓ Ejecución de los comandos, horarios e instrucciones.
- ✓ Realización de protocolos de telemetría y telecomando.
- ✓ Distribución y coordinación del tiempo en el satélite.
- ✓ Monitoreo del estado del satélite y todos sus sistemas.
- ✓ Toma de decisiones autónomas.
- ✓ Detección, aislamiento y corrección de fallas.

Además, el SCMI debe cumplir una serie de requerimientos que le permitirán, al contrario de las computadoras convencionales encontradas en tierra, poder implementarse en entornos espaciales. Algunos de los criterios que debe cumplir son los siguientes: [32]

- ✓ Principalmente, debe tener el rendimiento necesario para poder cumplir el propósito de la misión, como el procesamiento de datos, control del vehículo, manejo de los datos de la carga útil y otras funciones.
- ✓ Debe ser mecánicamente robusto para soportar cargas inducidas por el lanzados, como vibraciones y descargas.
- ✓ En órbita debe de soportar las condiciones electromagnéticas y los cambios térmicos correspondientes.
- ✓ Debe soportar una alta TID, considerando que se acumula durante la vida útil del sistema.
- ✓ Debe contar con protección contra SEE.
- ✓ Debe ser resistente a sustancias químicas agresivas.
- ✓ Su consumo de energía debe ser mínimo considerando las restricciones impuestas por el EPS.
- ✓ Debe cumplir con los estándares de seguridad y redundancia.

El SCMI se puede dividir a su vez en subsistemas, los cuales se presentan con poco detalle en la *Figura 45* pero se describirán más adelante.

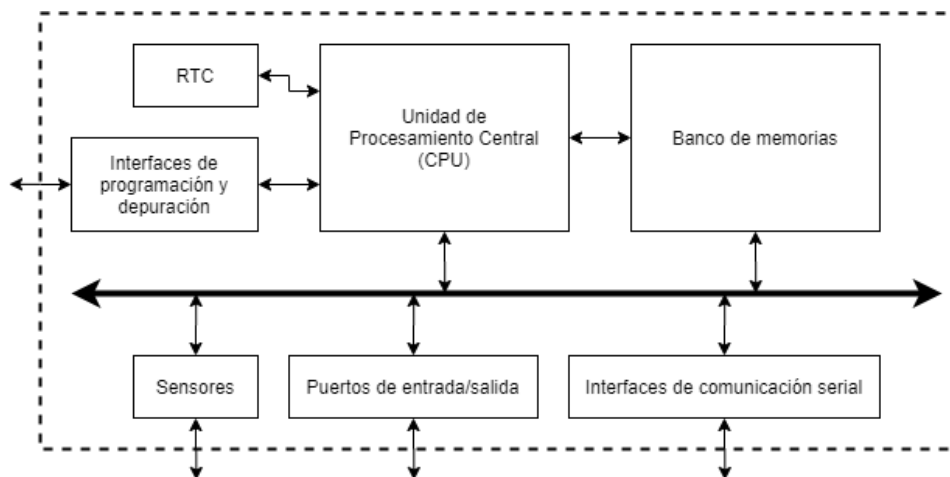


Figura 45. Estructura de un SCMI

### 4.6.1 Arquitectura de la computadora

La arquitectura o topología de la computadora se refiere al diseño conceptual de la distribución con la que operará la computadora. Existen tres arquitecturas muy utilizadas en sistemas espaciales cuya selección se realizará en base a la aplicación y requerimientos del sistema.

#### Arquitectura centralizada

Una computadora con arquitectura centralizada se localiza como la única unidad de procesamiento (véase la *Figura 46*) y principal ruteador que realiza la transmisión y recepción de telemetría, telecomando e instrucciones con cada uno de los sistemas del satélite mediante interfaces en paralelo.

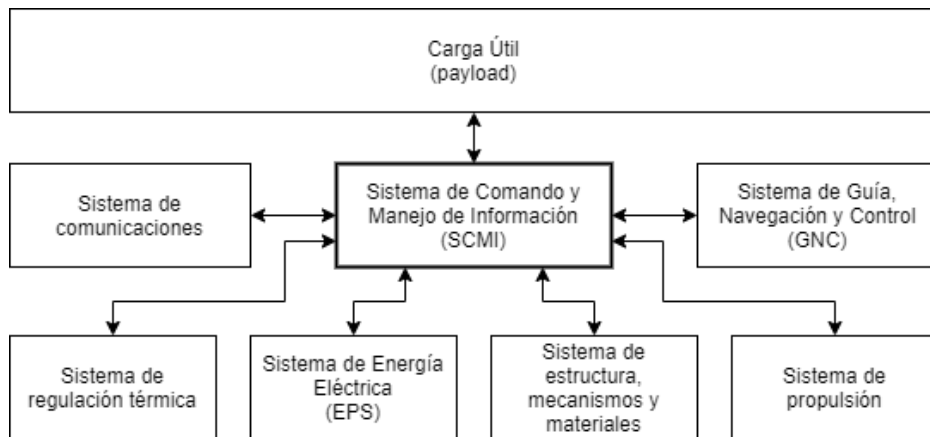


Figura 46. Arquitectura centralizada de un SCMI

#### Arquitectura distribuida

La arquitectura distribuida (también llamada arquitectura anillo) se refiere a un conjunto de unidades de procesamiento conectadas entre sí distribuidas en distintos puntos del sistema controlando de manera independiente un bloque en particular. En otras palabras, cada sistema del satélite o algunos de ellos cuentan con su propia computadora encargada estrictamente de las tareas correspondientes a dicho sistema. En esta distribución cada computadora puede reemplazar a otra en caso de que ocurra una falla. Esta arquitectura se representa en la *Figura 47*, en donde algunos sistemas del satélite cuentan con su propia computadora y cada una de estas es seleccionada con base en los requerimientos de cada sistema. Por ejemplo, pensando en un satélite de percepción remota, en donde la carga útil consiste en la operación de una cámara, la captura de imágenes, el procesamiento digital de estas imágenes y su envío hacia el sistema de comunicaciones para transmitir la información a tierra, la computadora que requiere el sistema de la carga útil debe tener una muy alta velocidad y gran capacidad de memoria a comparación de la computadora principal del satélite, mientras que la computadora del sistema de comunicaciones requiere de una velocidad más baja y poca capacidad de memoria [34].

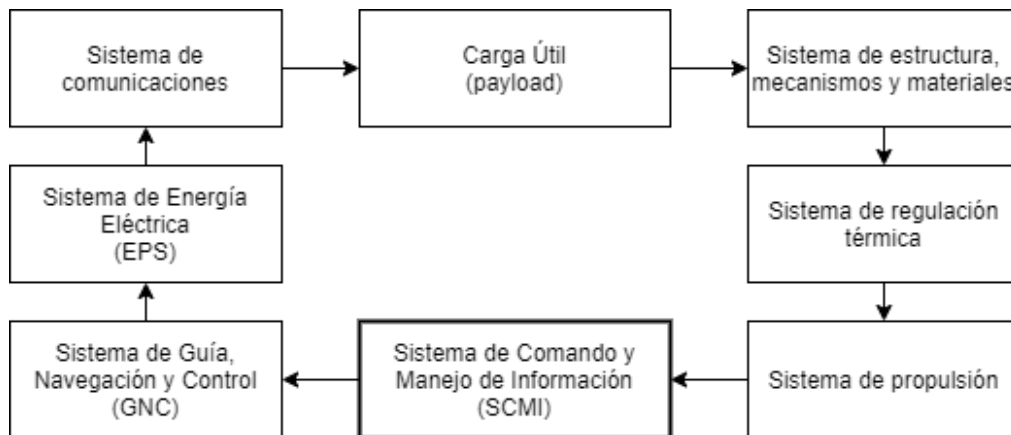


Figura 47. Arquitectura distribuida de un SCMI

**Arquitectura bus**

Un sistema con arquitectura bus se refiere a que todos los sistemas están interconectados entre sí mediante un único canal de comunicaciones bidireccional, siendo este un bus en donde se transmite información bit a bit a través de un protocolo de comunicación serial a cada uno de los dispositivos conectados o nodos, pero con la capacidad de seleccionar cuál o cuáles de ellos atienden la solicitud. Normalmente el SCMI actúa como el dispositivo maestro que administra todos los recursos del bus, pero dependiendo de la interfaz serial que se utilice se pueden implementar más de un dispositivo maestro dividiendo el control del sistema similar a una arquitectura distribuida, pero con la diferencia de que todos los dispositivos comparten el mismo bus. La arquitectura bus se representa en la *Figura 48*.

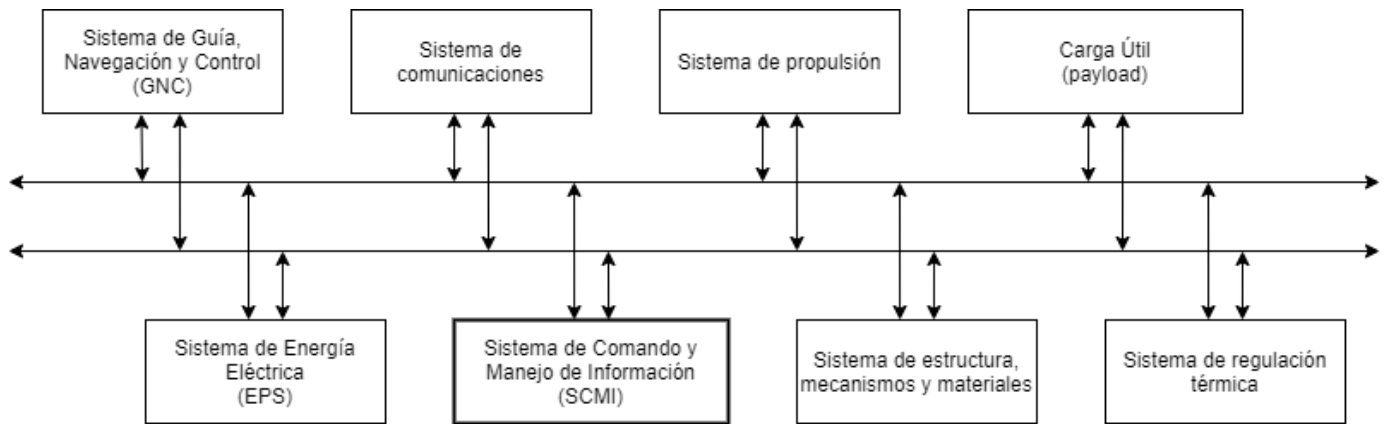


Figura 48. Arquitectura de un SCMI

Cada arquitectura posee ventajas y desventajas con respecto a otra (véase la *Tabla 8*), su selección depende de los requisitos y características del sistema.

Tabla 8. Comparación de las arquitecturas centralizada, distribuida y bus

Arquitectura	Ventajas	Desventajas
		Velocidad de procesamiento reducida.
	La información de todos los sistemas se encuentra en un solo lugar.	Tecnología de procesamiento de mayor nivel.
<i>Arquitectura centralizada</i>	Existe una sola unidad capaz de tomar decisiones.	Alto consumo de energía.
	Cada sistema cuenta con una comunicación directa con la computadora.	Alto riesgo de falla catastrófica.
		Necesidad de una unidad de respaldo externa.
		Límite de funcionalidad.

<i>Arquitectura distribuida</i>	Repartición de la toma de decisiones.	
	Procesamiento de mayor velocidad con computadoras menos robustas.	
	Distribución de hardware.	
	Distribución de la información.	Mayor complejidad de software.
	Redundancia de hardware.	Necesidad de mayor coordinación y distribución de datos.
	Distribución del software.	Buena sincronización entre las unidades de procesamiento.
	Consumo de energía usualmente menor.	
	Mayor facilidad de integración de software de todo el satélite (escrito por múltiples personas).	
	Menor costo.	
<i>Arquitectura bus</i>	Permite que la información llegue a todos los dispositivos.	
	Posibilidad de integrar nuevos dispositivos fácilmente.	El desempeño de la comunicación disminuye a medida que aumenta el número de dispositivos.
	Simplicidad en la arquitectura.	
	Requiere menos hardware.	Mayor probabilidad de falla en todo el bus.
	Compatible con diferentes protocolos de comunicación.	Mayor complejidad en técnicas de tolerancia a fallas.
	Menor costo.	
	Requiere poco espacio.	

La arquitectura bus es altamente utilizada gracias a sus características, además de que hoy en día existe una gran variedad de interfaces de comunicación serial con alta fiabilidad en la transmisión de datos que están presentes en la mayoría de los dispositivos electrónicos.

#### 4.6.2 Factor de forma

El factor de forma es el diseño físico del hardware de un dispositivo electrónico. Define las especificaciones físicas del sistema, como la forma y sus dimensiones. En la industria, la estandarización de un factor de forma para es necesario para permitir la compatibilidad de dispositivos diseñados por diferentes fabricantes. La definición de un factor de forma ayuda a distribuir los componentes aprovechando eficientemente el espacio disponible, facilita el diseño, construcción e integración en sistemas más grandes, facilita la transportación, permite la escalabilidad y brinda un diseño físico eficiente y resistente para obtener el mejor desempeño del sistema.

##### *PC/104*

En el área espacial, específicamente hablando de los nanosatélites, existe un factor de forma con el que se diseñan prácticamente todos los satélites CubeSat. Además de la estructura externa en forma de cubo, que es un ejemplo de factor de forma, el PC/104 es un estándar que define las especificaciones mecánicas y eléctricas de la tarjeta de circuito impreso (PCB, por sus siglas en inglés), así como el tipo de conector para apilar varias tarjetas entre sí aprovechando al máximo el reducido espacio del que se dispone en un CubeSat.



orientados hacia desarrollos futuros, y el 9 el más alto, en donde la tecnología ha sido probada en vuelo a través de una misión exitosa. Para que una tecnología aumente su TRL es necesario que pase por procesos de pruebas correspondientes a cada nivel, con lo que se facilita la toma más acertada de decisiones con respecto a la integración de tecnología a un proyecto espacial. En la *Figura 50*, se presentan los estados que debe poseer una tecnología para adquirir cada nivel.

**TRL 9**

- Sistema real “probado en vuelo” a través de operaciones de misión exitosas.

**TRL 8**

- Sistema real completado y “calificado para vuelo” mediante prueba y demostración (en tierra o en el espacio).

**TRL 7**

- Demostración de un prototipo de sistema en un entorno espacial.

**TRL 6**

- Demostración de modelo o prototipo de sistema/subsistema en un entorno relevante (en tierra o en el espacio).

**TRL 5**

- Validación de componentes y/o protoboard en el entorno relevante.

**TRL 4**

- Validación de componentes y/o protoboard en entorno de laboratorio.

**TRL 3**

- Función crítica analítica y experimental y/o prueba de concepto característica.

**TRL 2**

- Concepto de tecnología y/o aplicación formulada.

**TRL 1**

- Principios básicos observados e informados.

Figura 50. Nivel de madurez tecnológica (TRL) (adaptado de [36])



## 5 *Procesador de a bordo*

---

Como se mencionó con anterioridad, el SCMI se encarga de ejecutar las instrucciones definidas por software<sup>30</sup>, así como procesar toda la información e interconectar las interfaces de comunicación y control. En el presente trabajo, se define como el subsistema de procesador de a bordo a la unidad encargada de realizar las operaciones lógicas y aritméticas con el fin de ejercer control y administración de los datos a través del software, que, a su vez, requiere de interfaces de hardware<sup>31</sup> para comunicarse con el exterior. Específicamente, se detallan los tipos de dispositivos de procesamiento que se implementan típicamente en los SCMI, el medio que utilizan para configurarse y parte del software que requieren para operar.

### 5.1 **Unidad de procesamiento central (CPU)**

La CPU es la unidad destinada a interpretar y ejecutar las instrucciones definidas por software a través de las operaciones definidas en la unidad lógica-aritmética (ALU, por sus siglas en inglés), además de realizar el procesamiento de la información interna y proveniente de los otros sistemas del satélite. Dentro del SCMI se considera como CPU al o los dispositivos programables que desempeñan estas tareas, dando la posibilidad de seleccionar entre diferentes tipos de circuitos integrados según sean los requerimientos de la aplicación.

#### 5.1.1 **Microcontrolador**

Un microcontrolador es un circuito integrado VLSI que tiene embebido un procesador (CPU), unidades de almacenamiento y puertos de entrada/salida, capaz de ejecutar secuencialmente las instrucciones definidas por software almacenadas en su memoria. Los microcontroladores se caracterizan por ser pequeños, económicos y de bajo consumo de energía, por lo que son utilizados comúnmente en sistemas embebidos.

#### *Arquitecturas de computadoras*

Existen diferentes criterios de clasificación para las arquitecturas de computadoras. En este caso, se refiere a la forma en la que realizan sus cálculos, que puede ser de manera secuencial o paralela. Las computadoras secuenciales cuentan con un único CPU que ejecuta una instrucción a la vez, mientras que las computadoras paralelas poseen elementos repetidos para ejecutar varias instrucciones al mismo tiempo.

Dentro de las computadoras secuenciales destaca la arquitectura Von Neumann, la cual emplea una única memoria para almacenar las instrucciones y los datos, utilizando el mismo formato y requiriendo de un bus de datos compartido para la comunicación. Las computadoras Von Neumann solo pueden buscar o almacenar un elemento de datos a la vez. Por otra parte, una forma de obtener paralelismo sin repetir componentes se consigue a través de la segmentación encauzada (pipeline), en donde la computadora se divide en diferentes etapas, de modo que cada una efectúa operaciones sobre instrucciones diferentes, permitiendo comenzar con el proceso de una nueva instrucción (también llamado ciclo de Fetch) en el momento en que la instrucción anterior pase a la siguiente etapa, aumentando enormemente la velocidad de la computadora. Este tipo de arquitectura se conoce como computadora tipo Harvard, en donde se leen y escriben instrucciones y datos simultáneamente en la memoria, por lo que se emplean dos dispositivos separados con buses independientes. [37]

Actualmente, el modelo más utilizado es la arquitectura Harvard gracias a que presenta una eficiencia muy superior a la arquitectura Von Neumann.

#### *Conjunto de instrucciones*

El conjunto de instrucciones de un microcontrolador define las operaciones que puede realizar. Existen dos tipos de microprocesadores, la computadora de conjunto de instrucciones reducido (RISC) y la computadora de conjunto de instrucciones complejo (CISC, por sus siglas en inglés).

---

<sup>30</sup> Secuencia ordenada de instrucciones altamente específicas almacenadas en memoria que definen exactamente las tareas que realizará una computadora y cuándo efectuarlas.

<sup>31</sup> Conjunto de los componentes físicos que conforman una computadora.

Las computadoras RISC, correspondientes a la arquitectura Von Neumann, poseen instrucciones de tamaño fijo y presentadas en un reducido número de formatos en donde sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos, reduciendo el acceso a la memoria. Este conjunto posibilita la segmentación<sup>32</sup> y paralelismo<sup>33</sup> en la ejecución de instrucciones. Algunas de sus ventajas son:

- ✓ Mayor velocidad, la CPU utiliza menos ciclos de reloj para ejecutar instrucciones (puede utilizar solo un ciclo de reloj).
- ✓ Utiliza un sistema de direcciones no destructivas en la memoria RAM, lo que significa que conserva en memoria los dos operandos y su resultado después de realizar una operación, reduciendo la ejecución de nuevas operaciones.

En cambio, las computadoras CISC, correspondientes a la arquitectura Harvard, poseen un conjunto de instrucciones amplio permitiendo operaciones complejas entre operandos situados en la memoria o en los registros internos. Su diseño dificulta el paralelismo entre instrucciones y requiere de mayor número de ciclos de reloj para ejecutar una instrucción.

Hoy en día, los procesadores modernos emplean arquitecturas RISC, mientras que los CISC implementan un sistemas de conversión de CISC a RISC.

### 5.1.2 Arreglo de compuertas programables en campo (FPGA)

Un FPGA es un dispositivo programable formado por unidades lógicas cuyas conexiones pueden configurarse mediante un lenguaje de descripción de hardware. Es un dispositivo VLSI con el que se pueden implementar circuitos electrónicos desde combinacionales básicos hasta microcontroladores.

Los FPGA se utilizan en aplicaciones similares a los circuitos integrados de aplicación específica<sup>34</sup> (ASIC, por sus siglas en inglés) añadiendo la ventaja de que son dispositivos reprogramables. Sin embargo, los costos y tiempos de desarrollo son considerablemente más elevados, por lo que es común la inclinación hacia el uso de otros dispositivos programables como los microcontroladores. Actualmente, se desarrollan FPGA lo suficientemente robustos para implementar sistemas complejos con un bajo consumo de energía. Los FPGA se implementan en los SCMI de propósito general operando en conjunto con un microcontrolador con el propósito de integrar sistemas embebidos de procesamiento paralelo, además de aumentar la capacidad de integración de hardware para todo tipo de misiones, convirtiéndose además en un dispositivo con el que se puede distribuir el procesamiento e implementar técnicas de tolerancia a fallas.

### 5.1.3 Sistema en chip (SoC)

Un SoC hace referencia a un único circuito integrado que se distingue de otras tecnologías por la cantidad de módulos IP<sup>35</sup> del tipo hard<sup>36</sup> y diferentes etapas analógicas, como convertidores, comparadores, amplificadores operacionales, filtros, etc., lo que permite su implementación en diversas aplicaciones. Además, cuentan con zonas configurables similares a los FPGA. La construcción de un SoC representa en gran medida a la tendencia de diseño VLSI.

## 5.2 Programación y depuración

Programación se refiere a la implementación de un algoritmo empleando las instrucciones que ofrece un procesador para desarrollar un conjunto de tareas específicas orientadas a cumplir objetivos determinados. El

<sup>32</sup> También conocida como pipeline, se basa en dividir las instrucciones en etapas para que puedan ejecutarse simultáneamente.

<sup>33</sup> Tipo de micro arquitectura capaz de ejecutar más de una instrucción por cada ciclo de reloj.

<sup>34</sup> Circuito integrado diseñado a la medida para una aplicación en específico.

<sup>35</sup> Núcleo de propiedad intelectual de semiconductores. Se refiere a un circuito integrado o chip licenciado como propiedad intelectual de una de las partes. Suelen ser usados como bloques de construcción en sistemas ASIC o FPGA.

<sup>36</sup> Tipo de núcleo IP diseñado para no permitir modificaciones de hardware y software, por lo que implementarlos solo requiere de su habilitación y configuración.

conjunto de instrucciones es conocido como programa o software, el cual se almacena en una memoria que es leída y posteriormente ejecutado por el procesador.

Un depurador (debugger) es todo instrumento de hardware o software empleado para probar y modificar un programa permitiéndole al desarrollador localizar y corregir sus errores (bugs). Un depurador tiende a ser lo menos intrusivo posible para que el programa opere con normalidad como si el depurador no existiera [38]. Además de que añade herramientas que facilitan examinar el código, como puntos de ruptura (breakpoints), información sobre los elementos del programa, ejecución del programa instrucción a instrucción, edición del programa, entre otras. Entre sus tareas más importantes, además de la corrección del código, destacan el control y la observación, verificando la funcionalidad del programa internamente en el dispositivo y comprobando el comportamiento de las entradas y salidas del sistema.

El presente trabajo se centrará en el funcionamiento de las interfaces de programación y depuración, siendo el medio por el cual se comunicará la computadora con un dispositivo externo para configurarse y llevar a cabo estas funciones. Por facilidad de comprensión, todas las interfaces de comunicación se ven a detalle en el capítulo *Interfaces eléctricas y periféricos*.

### 5.3 Sistemas operativos en tiempo real (RTOS)

El presente trabajo no involucra la implementación de un RTOS, sin embargo, el diseño está visualizado con la posibilidad de escalar hasta obtener un sistema completo que posea las características necesarias para volar cumpliendo ciertos estándares de confiabilidad, por lo que se introduce brevemente al tema con el objetivo de dar las bases para el trabajo a futuro que esto representa. Para entender lo que representa un RTOS se presentan algunos conceptos anteriores a este.

Como primera parte, un sistema es una colección de componentes que son combinados para desempeñar una tarea simple y complicada de manera coordinada [38]. Este se define como sistema de tiempo real si existe un retraso determinado entre el momento en que una tarea debe completarse y cuando realmente se completa. Teniendo claro esto, un sistema operativo (OS) es la capa principal de software que gestiona todos los recursos de hardware y software. En otras palabras, es un administrador de los recursos de una computadora (memoria, puertos de entrada/salida, datos y el procesador) proporcionado por su arquitectura fundamental cuyo objetivo es facilitar el traslado del sistema de una plataforma a otra. En un sistema básico en donde el software es escrito para un microcontrolador específico no es necesario implementar un OS, pero conforme aumenta la complejidad del sistema, su implementación facilita la integración del software en múltiples plataformas. Entre mayor complejidad presente un sistema, las tareas que desempeña el OS adquieren importancia al proveer mecanismos para que los recursos puedan comunicarse entre ellos administrando su rendimiento en términos de tiempo de respuesta. Ahora, de la misma forma que un sistema se convierte en un sistema de tiempo real, un RTOS es un sistema operativo que gestiona los recursos de una computadora garantizando que se cumplan estrictamente todas las restricciones de tiempo definiendo límites superiores e inferiores en el ancho de banda<sup>37</sup> con el fin de proporcionar una operación confiable con una latencia<sup>38</sup> acotada. Se encarga de soportar la sincronización y comunicación entre tareas, planificar tareas periódicas y, a medida que aumenta la complejidad del sistema, gestionar el cambio e integración de componentes permitiendo que los cambios de componentes ocurran con mucha menor frecuencia. Para esto, es necesario que el RTOS sea predecible, lo que significa que debe conocer cuánto tiempo necesita cada tarea para ejecutarse y cuál es la prioridad relativa entre ellas. También puede emplear un planificador (scheduler) para ejecutar múltiples tareas en primer plano para satisfacer los contratiempos del sistema. La tolerancia de un RTOS al incumplimiento de los requisitos de tiempo determina si se clasifica como tiempo real duro, tiempo real firme o tiempo real suave [39].

Un sistema embebido es un sistema diseñado para desempeñar tareas específicas que típicamente cumple seis restricciones: pequeño tamaño, baja masa, bajo consumo de energía, alta resistencia a ambientes hostiles, son utilizados en sistemas críticos de seguridad y son extremadamente sensibles al costo [39]. En un sistema

<sup>37</sup> Número de bytes por segundo de información que pueden ser transferidos o procesados.

<sup>38</sup> También llamada tiempo de respuesta, es el retraso desde una solicitud hasta el comienzo del servicio de dicha solicitud.

embebido, el tiempo real significa que el sistema responde a eventos críticos dentro de un tiempo estrictamente definido, denominado tiempo límite (deadline). En ellos, el usuario usualmente no puede hacer cambios en el hardware, actualizar el software ni modificar su objetivo, a menos que posea un sistema operativo que pueda gestionar estas modificaciones, con la finalidad de mejorar su rendimiento, características y confiabilidad.

### 5.3.1 RTOS disponibles

Existe una gran variedad de RTOS disponibles, sin embargo, en este trabajo se les da prioridad a aquellos identificados en los SCMI comerciales según el *Estado del arte*, entre los que destacan los siguientes (para más información, visitar la página correspondiente de cada sistema operativo).

#### ***FreeRTOS***

FreeRTOS es un sistema operativo en tiempo real para microcontroladores, distribuido libremente bajo la licencia de código abierto del MIT, diseñado para obtener fiabilidad mediante un uso sencillo, destaca siendo el RTOS con mayor compatibilidad con los SCMI comerciales. Incluye bibliotecas para la conectividad, la seguridad y las actualizaciones inalámbricas. El núcleo o kernel<sup>39</sup> de FreeRTOS admite más de 40 arquitecturas de procesador e incluye un programador multitareas, opciones de asignación de memoria y primitivos<sup>40</sup> de coordinación entre tareas (notificaciones, colas de mensajes, varios tipos de semáforo y búferes de transmisión y mensajes) [40].

#### ***KubOS Linux***

KubOS Linux es un paquete de framework<sup>41</sup> que se ejecuta directamente en plataformas satelitales combinando una distribución de Linux personalizada, interfaz de programación de aplicaciones para los subsistemas y servicios básicos. Brinda servicios como base de datos de telemetría, comunicación, protección, monitoreo, transferencia de archivos, aplicación de la misión y protocolos de archivos y protección. Se puede implementar en los diferentes sistemas del satélite y es compatible con productos de las empresas dedicadas al desarrollo y distribución de sistemas espaciales [41].

#### ***Ejecutivo en tiempo real para sistemas multiprocesador (RTEMS)***

RTEMS es un sistema operativo en tiempo real de código abierto que soporta interfaces de programación de aplicaciones estándar abiertas utilizado en aplicaciones espaciales, medicas, de redes y sistemas embebidos para diferentes arquitecturas de procesador. Tiene un soporte comercial disponible para las compañías estadounidenses y europeas y gratuito de manera global [42].

#### ***Micrium uC/OS-II***

El software Micrium, parte de Silicon Labs, es una familia de RTOS para desarrolladores que promete alta confiabilidad y todas las funcionalidades para construir sistemas embebidos, en donde destaca el kernel de tiempo real uC/OS. Micrium ofrece el código fuente completo en C, la documentación, funciones de depuración y el soporte para un gran número de arquitecturas de procesador [43].

---

<sup>39</sup> Software fundamental de un OS, responsable de facilitar al software el hardware de la computadora, gestionando qué programa puede utilizarlo y durante cuánto tiempo.

<sup>40</sup> Segmento de código básico que puede ser utilizado para construir elementos o interfaces de software más sofisticadas.

<sup>41</sup> Estructura o esquema conceptual que sirve de base para el desarrollo de software.

## **6** *Unidades de almacenamiento*

---

Los dispositivos de almacenamiento o memorias son uno de los componentes fundamentales de la computadora, su propósito es almacenar todo tipo de información con la que opera el SCMI, tanto las instrucciones que realiza el sistema como los datos recopilados durante la misión. En este capítulo se dan algunos conceptos que permitan conocer qué es una memoria, sus características, sus funciones y se describe de manera breve los tipos de memoria y las tecnologías con las que se implementan en sistemas espaciales, con el fin de fundamentar la selección del tipo de dispositivos de almacenamiento que se ocuparán en el diseño del SCMI. También, se describen los tipos de almacenamiento o bancos de memoria que se establecieron para la computadora.

### **6.1 Memoria**

La memoria es la colección de elementos de hardware en una computadora en donde se almacena la información [38]. Un dispositivo de memoria está compuesto por una serie de celdas de memoria, término que hace referencia a la unidad básica y fundamental de cualquier memoria capaz de almacenar un bit de información, organizadas y ubicadas físicamente por una dirección secuencial única.

Existen algunos conceptos o características con los cuales se pueden clasificar los diferentes tipos de memorias, los cuales son los siguientes:

#### ***Volatilidad de la información***

Es la capacidad de una memoria de mantener o perder la información almacenada si esta pierde su alimentación. Se dice que una memoria es volátil cuando requiere un suministro constante de energía para mantener almacenada la información, mientras que una memoria no volátil puede retenerla incluso cuando esta no recibe energía. Además, se conoce como memoria dinámica a aquella memoria volátil cuya información requiere refrescarse periódicamente, refiriéndose a que para mantener almacenado un dato, es necesario revisar el mismo y recargarlo cada cierto periodo de tiempo en un ciclo de refresco.

#### ***Tiempo de acceso***

Similar a la latencia, el tiempo de acceso es el tiempo que transcurre entre el instante en que se ordena una operación de lectura y el instante en el que se dispone de la información.

#### ***Tiempo de ciclo***

Se trata del tiempo transcurrido desde que se solicita un dato a la memoria hasta que ésta se encuentre disponible para efectuar una nueva operación de lectura o escritura.

#### ***Modo de direccionamiento***

Establece el tipo de acceso a la información. En una memoria de acceso aleatorio el tiempo de acceso no depende de la ubicación de los datos, lo que significa que se puede acceder a cualquier localización de la memoria en el mismo intervalo de tiempo. De manera contraria, en una memoria con acceso secuencial el tiempo de acceso depende de la localización del elemento de memoria a utilizar, lo que significa que acceder a una unidad de información tomará un intervalo de tiempo variable, dependiendo de la unidad de información leída anteriormente.

#### ***Capacidad***

Es la cantidad máxima de bits de información que pueden almacenarse en la memoria. Esta capacidad se define por el tipo de aplicación que desempeña la computadora.

#### ***Tipos de memoria***

Históricamente, toda computadora utiliza dos tipos de memorias: la RAM y la ROM.

### *Memoria de acceso aleatorio (RAM)*

La RAM es la memoria operativa de la computadora utilizada para el almacenamiento temporal de las instrucciones, datos y procesos temporales, como el sistema operativo. En otras palabras, es la memoria principal de trabajo en donde se almacena la mayor parte del software e información con los que opera el procesador y requiere acceder de manera inmediata. Se trata de una memoria volátil de escritura y lectura y, como su nombre lo indica, de acceso aleatorio, por lo que requiere de una cantidad de tiempo constante para ingresar a cualquier localidad de memoria.

### *Memoria de solo lectura (ROM)*

La memoria de solo lectura (ROM, por sus siglas en inglés) es un tipo de memoria no volátil cuyo contenido se almacena desde su creación y no puede modificarse, aunque actualmente existen memorias ROM cuyo contenido puede modificarse a través de distintos métodos. Entre sus principales funciones se encuentran el almacenamiento de software, de datos permanentes y el firmware<sup>42</sup>. La principal diferencia con la RAM es que se trata de una memoria permanente.

## **6.2 Bancos de memoria**

Un SCMI está equipado con diversos tipos de memoria orientados a cumplir diferentes propósitos. En este proyecto se definieron seis tipos de memoria, cuyas funciones se describen a continuación.

### *Memoria de trabajo*

Es la memoria de trabajo o RAM, en donde se almacenan temporalmente las instrucciones, datos y el sistema operativo con los que opera el procesador.

### *Memoria de almacenamiento masivo de datos*

Es el banco de memoria con mayor densidad dentro del sistema en donde se almacena toda la información obtenida de la misión para la que fue diseñado el SCMI.

### *Memoria de almacenamiento crítico de datos*

Su propósito es almacenar la información crítica, refiriéndose a aquellos datos indispensables para el correcto desempeño de la misión, por lo que su densidad es mucho menor que la memoria de almacenamiento masivo de datos, pero se emplea un tipo de tecnología que brinde menor probabilidad de que ocurra alguna falla que pueda dañar su información.

### *Memoria de almacenamiento de código*

Como su nombre lo indica, es la memoria destinada a el almacenamiento del código desarrollado para que el SCMI cumpla su misión correspondiente.

### *Memoria de almacenamiento de código de mayordomo (housekeeping)*

El código de mayordomo o housekeeping es el software agregado por el usuario o el proceso manual o automatizado que se encargará de limpiar la computadora liberando sus recursos y eliminando los archivos temporales que ya no tienen propósito dentro del sistema garantizando que el SCMI opere con normalidad y con la mayor eficiencia posible.

### *Memoria de arranque*

La memoria de arranque se refiere al dispositivo que almacena el sistema básico de entrada/salida (BIOS, por sus siglas en inglés), el cual corresponde al firmware de mayor prioridad ejecutado al momento de encender la computadora, encargado de iniciar y probar el hardware del sistema y gestionar las instrucciones de arranque

---

<sup>42</sup> Software que controla el hardware a más bajo nivel.

de la computadora, como el sistema operativo. En un microcontrolador, la memoria de arranque corresponde a una zona de la ROM embebida en el dispositivo.

## 6.3 Tecnologías de las memorias

Como se mencionó con anterioridad, la celda de memoria es la unidad básica y fundamental de cualquier memoria que, a su vez, define su modo de operación con base en su implementación. Sabiendo esto, actualmente se construyen memorias mediante una cierta variedad de tecnologías, lo que les otorga diferentes características y propiedades. A continuación, se describe el funcionamiento y las características de las tecnologías más ocupadas en los SCMI con la finalidad de seleccionar el tipo de memoria más apropiado para cada propósito.

### 6.3.1 Memoria dinámica de acceso aleatorio (DRAM)

La memoria DRAM es un tipo de RAM basada en condensadores, siendo una memoria volátil, dinámica y asíncrona. Su celda de memoria está constituida por un condensador para almacenar la carga que, al perderla progresivamente, requiere de un circuito de refresco que cada determinado periodo de tiempo revise y reponga el dato, y de un transistor de efecto de campo (FET, por sus siglas en inglés) operando como un interruptor que conecta y desconecta al condensador, este circuito se puede observar en la *Figura 51*. Al ser estos dos elementos los únicos para implementar una celda de memoria, la DRAM permite tener una gran densidad de celdas de memoria en un solo integrado, por lo que es muy utilizada como memoria de trabajo de una computadora.

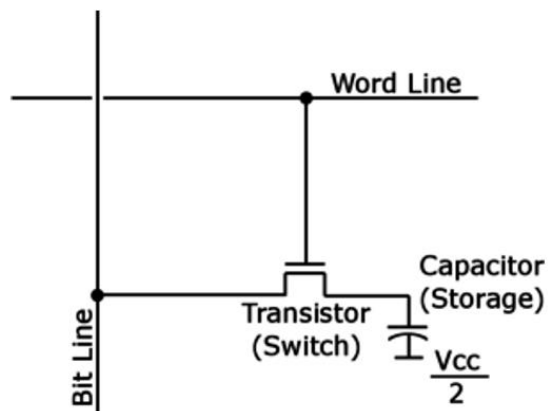


Figura 51. Celda de memoria de la DRAM

La DRAM se caracteriza por ser una memoria económica, en donde el costo de paquete se redujo al multiplexar las líneas de dirección, considerando que la primera mitad de la dirección indica la fila y la segunda la columna, lo que quiere decir que la estructura interna de la DRAM está organizada como una matriz rectangular direccionada por filas y columnas [44]. Sin embargo, también es una memoria de poca velocidad, por ser dinámica, en comparación a su competencia.

La lectura en una DRAM es destructiva, lo que quiere decir que al leer un bit de información este es destruido, por lo que debe ser restaurado. Debido a esto, el tiempo de ciclo de la DRAM es tradicionalmente mayor al tiempo de acceso [44].

### Memoria dinámica de acceso aleatorio síncrona (SDRAM)

Como su nombre lo indica, se trata de una DRAM síncrona, que puede controlarse de manera segmentada, de modo que la memoria puede recibir una nueva instrucción antes de que haya terminado de procesar la anterior, lo que la convierte en una memoria más eficiente que la DRAM. Su desventaja es su limitada capacidad de almacenamiento comparada con la memoria Flash [33].

### Memoria dinámica de acceso aleatorio síncrona de doble tasa de transmisión (DDR SDRAM)

Es un tipo de SDRAM caracterizada por permitir mayores tasas de transmisión de datos utilizando ambas transiciones de la señal de reloj para transferir datos y, de este modo, obtener una velocidad hasta dos veces más rápida que la SDRAM estándar.

### 6.3.2 Memoria estática de acceso aleatorio (SRAM)

La SRAM es un tipo de RAM volátil basada en semiconductores. La principal diferencia que tiene con la DRAM es que es estática, lo que quiere decir que no necesita de un ciclo de refresco para mantener almacenada la información, por lo que el tiempo de acceso es muy cercano al tiempo de ciclo [44]. Conservando la comparación, es una memoria de mayores costos y dimensiones, ya que su celda de memoria está construida mediante un mayor número de componentes, siendo normalmente seis transistores de efecto de campo metal-óxido-semiconductor (MOSFET, por sus siglas en inglés) para evitar que la información sea perturbada cuando se lee [44]. Siguiendo el ejemplo de la *Figura 52*, cuatro de los transistores ( $M_1$ ,  $M_2$ ,  $M_3$  y  $M_4$ ) forman un circuito biestable<sup>43</sup> en donde se almacena el bit de información, mientras que los transistores  $M_5$  y  $M_6$  controlan el acceso al circuito biestable para las operaciones de lectura y escritura mediante el bus WL.

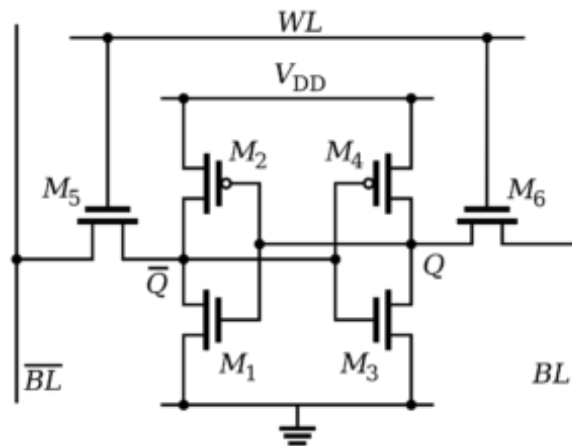


Figura 52. Celda de memoria de la SRAM

En comparación con la DRAM, la SRAM posee una mayor velocidad (menor latencia) y un menor consumo de energía, aunque ofrece menores densidades de memoria a un mayor costo. En aplicaciones espaciales se utiliza típicamente la SRAM debido a su precio y disponibilidad. [1]

### 6.3.3 Memoria ferroeléctrica de acceso aleatorio (FRAM)

Es un tipo de memoria de estado sólido cuya celda de memoria es construida, similar a la DRAM, mediante un transistor y un condensador, como se observa en la *Figura 53*, pero con la diferencia de que la FRAM utiliza una capa de material ferroeléctrico en lugar de un dieléctrico, este material se deposita de forma cristalina entre dos placas formando un condensador. Su funcionamiento se basa en aprovechar el movimiento del átomo, localizado en el centro de la estructura cristalina del ferroeléctrico, que se presenta al aplicarle un campo eléctrico, provocando que se mueva en la misma dirección y sentido del campo (véase la *Figura 54*) y que permanecerá estable en la ausencia de este, por lo que la FRAM es una memoria no volátil y estática, por lo que no necesita de un ciclo de refresco para retener la información.

Sus procesos de escritura y lectura son similares, ambos consisten en forzar el movimiento de los átomos aplicándose un campo eléctrico. Para la escritura, se aplica un campo eléctrico cuya polarización eléctrica determina el valor del estado, el cual se mantiene de forma estable. Para el caso de la lectura, el campo eléctrico provoca que los átomos adquieran un estado en particular considerando que aquellos que ya se encontraban en aquel estado no presentarán cambio alguno, mientras que los que se encontraban en el estado opuesto cambiarían

<sup>43</sup> Circuito capaz de permanecer en uno de los dos estados lógicos durante un tiempo indefinido en ausencia de perturbaciones.



de dirección generando un pequeño pulso de corriente a la salida, dando a entender que esas celdas almacenaban el estado opuesto. Lo anterior significa que el proceso de lectura de la FRAM es destructivo, por lo que el dispositivo se encargará de recuperar la información perdida tras cada lectura.

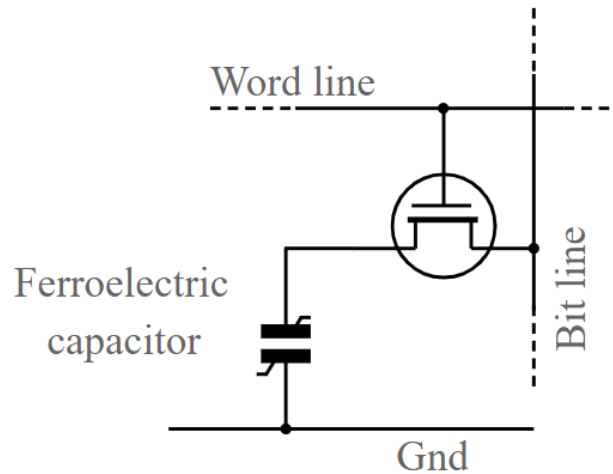


Figura 53. Celda de memoria de la FRAM

Entre sus principales ventajas frente a las memorias Flash y EEPROM destacan su mayor velocidad en el proceso de escritura, su bajo consumo de energía al requerir de un nivel de tensión bajo para operar y poca corriente para escribir, su mayor fiabilidad en los datos siendo más resistente a la corrupción de su información por campos eléctricos externos, también es inmune a los campos magnéticos y muy resistente a la radiación. Las celdas de la FRAM se bloquean utilizando una estructura de película PZT<sup>44</sup>, la cual tiene mayor probabilidad de mantener su estado durante un SEE que las estructuras capacitivas tradicionales encontrados en RAM [45] [46]. La FRAM es una memoria que soporta un alto rendimiento a un bajo consumo de energía, presentando un tiempo de acceso bajo, pero a un costo mayor a pesar de poseer una densidad menor.

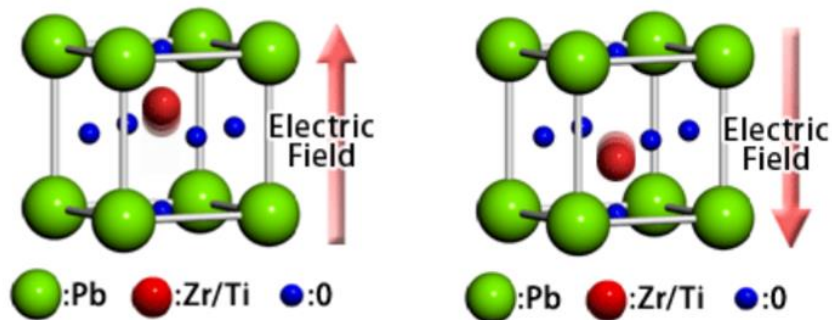


Figura 54. Estructura cristalina del condensador ferroeléctrico

#### 6.3.4 Memoria magnética de acceso aleatorio (MRAM)

Es un tipo de memoria no volátil que, a diferencia de la RAM, almacena información mediante elementos de almacenamiento magnético. Su celda de memoria está construida por dos discos ferromagnéticos capaces de mantener un campo magnético separados por una capa de material aislante. Un disco se coloca en un imán permanente con una polaridad establecida, mientras que el otro variará dependiendo de un campo externo. La estructura de la celda de la MRAM se presenta en la *Figura 55*.

El proceso de lectura se realiza midiendo la corriente para calcular la resistencia eléctrica de la celda que variará según la orientación de los campos magnéticos de ambos discos. Para el caso de la escritura, comúnmente se induce un campo magnético en la unión generado por una corriente eléctrica, lo que provoca que la densidad

<sup>44</sup> Cerámica piezoeléctrica de titanato de plomo-circonio.

de la memoria sea limitada por la interferencia de los campos magnéticos entre diferentes celdas requiriéndose un tamaño mínimo para evitarlo, actualmente se encuentran en desarrollo otros métodos para corregir esta limitante.

Las principales ventajas que ofrece frente a otras tecnologías son su velocidad cercana a la SRAM, siendo mucho más rápida que la Flash, una densidad comparable con la DRAM, pero un consumo de energía menor y que no presenta ningún tipo de degradación como en el caso de la Flash. Sin embargo, requiere de una corriente alta para generar el campo magnético. Entre sus principales aplicaciones destacan su integración en sistemas espaciales y militares.

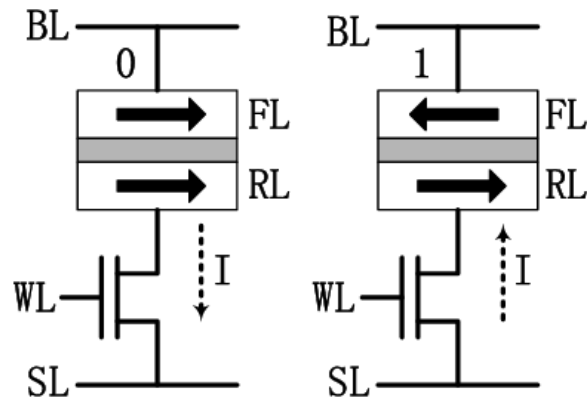


Figura 55. Celda de memoria de la MRAM

### 6.3.5 Memoria de solo lectura eléctricamente programable y borrable (EEPROM)

La EEPROM es un tipo de memoria ROM no volátil que, a diferencia de sus versiones anteriores, puede ser programada, borrada y reprogramada eléctricamente. Es la versión anterior a la memoria Flash. Sus celdas están organizadas como un arreglo de transistores de compuerta flotante con transistores MOSFET y utiliza una interfaz serial, como I<sup>2</sup>C o SPI, o paralela para la entrada y salida de datos.

El proceso de programación se realiza aplicando un nivel de tensión alto (de 21 V) a la compuerta flotante del transistor para que se acumule una carga eléctrica suficiente para encender al transistor y que este almacene el dato, mientras que para el borrado se aplica una tensión negativa para liberar esta carga. Parte de sus ventajas son que permite el borrado y reprogramación de bytes individuales, se programa y borra a través del mismo circuito eléctrico con el que opera y no con uno especial como sus versiones anteriores.

### 6.3.6 Flash

La memoria Flash es una memoria estática no volátil. Es la evolución de la EEPROM, permitiendo la lectura y escritura de múltiples celdas de memoria en una misma operación, por lo que poseen una mayor velocidad. Su celda de memoria está construida por dos transistores MOSFET, en donde la compuerta de uno está eléctricamente aislada, lo que permite mantener almacenada la información por años. El otro transistor es utilizado para leer el bit detectando si el primer transistor ha almacenado o no la carga [38]. Opera colocando electrones a través de una barrera de material aislante obstruyendo la terminal del transistor, lo que requiere de altos niveles de tensión que, naturalmente, deterioran las celdas de memoria limitando el número de ciclos de escritura y borrado entre 100 000 y 1 millón de veces. Parte de las ventajas que ofrecen son su bajo tiempo de acceso y su bajo consumo de energía.

Existen dos tipos de memorias Flash diferentes: la NAND Flash y la NOR Flash, su diferencia es la forma por la que se accede a ellas [33].

#### *NOR Flash*

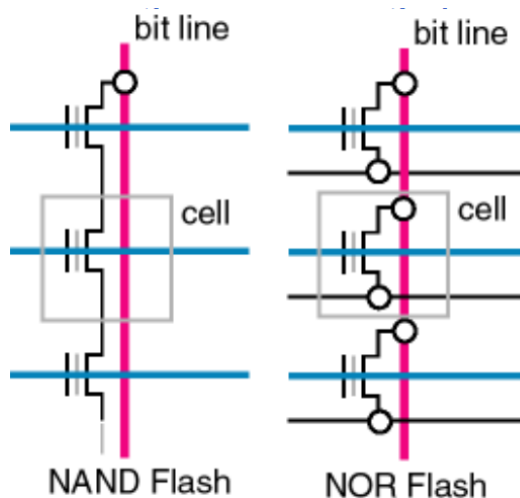
La NOR Flash tiene un acceso del tipo aleatorio, es mucho más veloz que la NAND Flash, pero ofrece una densidad considerablemente menor a un mayor precio. Suele ocuparse típicamente para el almacenamiento de

código del programa [33]. A diferencia de la NAND Flash, permite modificar un solo bit y presenta una mayor fiabilidad de la información, siendo relativamente inmune a la corrupción de datos.

### NAND Flash

La NAND Flash tiene una lectura de acceso aleatorio a un menor costo por bit y procesos de escritura y borrado a través de operaciones de bloque [47], por lo que es más rápida que la NOR Flash, ofrece una densidad de memoria mucho mayor a un precio menor, siendo más sencillo su proceso de borrado, pero menor fiabilidad en sus datos. Son utilizadas comúnmente como dispositivos de almacenamiento masivo de datos [33].

Las estructuras de las celdas de memoria de las NOR y NAND Flash se presenta en la *Figura 56*.



*Figura 56. Celda de memoria de las NAND Flash y NOR Flash [47]*

Como parte del progreso que tuvo la memoria Flash, específicamente la NAND Flash, se estandarizaron algunos formatos de tarjetas retirables como la tarjeta multimedia de memoria (MMC) y la memoria digital segura (SD), entre otros, siendo la SD la más utilizada en todo tipo de dispositivos hoy en día gracias a su gran velocidad y densidad de memoria. A su vez, existen diferentes clases de tarjetas SD dependiendo de su capacidad, velocidad de escritura y aplicación y se pueden encontrar en tres factores de forma: SD estándar, miniSD y microSD. La tarjeta SD emplea cuatro buses de comunicación, destacando el bus SPI.

Parte de la tarea del diseño de un SCMI consiste en la selección de los dispositivos, en la *Tabla 9* se presenta una comparativa de algunas características de las memorias importantes a considerar para definir el tipo de memoria a utilizar para cada función.

*Tabla 9. Comparación de los tipos de memorias [1] [47]*

<i>Característica</i>	<b>DRAM</b>	<b>SRAM</b>	<b>FRAM</b>	<b>MRAM</b>	<b>Flash</b>
<i>No volátil</i>	No	No	Sí	Sí	Sí
<i>Tensión de operación ±10%</i>	3.3 V	3.3 – 5 V	3.3 V	3.3 V	3.3 / 5 V
<i>Retención de la información (a 70 °C)</i>	-	-	10 años	10 años	10 años
<i>Vida útil (ciclos de borrado/escritura)</i>	Ilimitado	Ilimitado	10 <sup>13</sup>	10 <sup>13</sup>	10 <sup>6</sup>

<i>Tiempo de acceso</i>	25 ns	10 ns	300 ns	300 ns	50 ns (después de la página lista) 200 $\mu$ s (escritura) 2 ms (borrar)
<i>Radiación (TID)</i>	50 krad	1 Mrad	1 Mrad	1 Mrad	30 krad
<i>Tasa de SEU (relativa)</i>	Alta	Baja-nula	Nula	Nula	Nula (celdas) Baja (dispositivos electrónicos)
<i>Rango de temperatura</i>	Industrial	Militar-estándar	Militar-estándar	Militar-estándar	Comercial
<i>Consumo de energía</i>	Alto	Medio			Bajo
<i>Capacidad</i>	Media	Baja			Alta

## 6.4 Protección de las memorias

Además de la selección de tecnologías, en aplicaciones espaciales se utilizan técnicas de protección que aumentan la fiabilidad de los datos almacenados en memoria, como los ECC y el software EDAC.

### 6.4.1 Memorias con código de corrección de errores (ECC)

Los ECC son técnicas EDAC de mitigación y tolerancia a fallas (véase el capítulo 8.4. *Tolerancia a fallas*), utilizados para el control de posibles errores en la información. Su operación consiste en enviar datos adicionales en la información, de modo que el dispositivo receptor sea capaz de detectar y corregir un número determinado de errores. De manera similar, las memorias con ECC son capaces de detectar y corregir  $n$  cantidad de bits erróneos en memorias RAM y Flash. En general, las memorias con ECC operan almacenando sumas de verificación en una parte de la memoria, las cuales se utilizan para detectar zonas inestables de la memoria. Sumado a esto, se utiliza procesamiento adicional para corregir, a través de las sumas de verificación, errores de uno o varios bits. El controlador de memoria es responsable de administrar la memoria con ECC durante las operaciones de lectura y escritura. [48]

## 7 Interfaces eléctricas y periféricos

---

Como se ha visto con anterioridad, el SCMI se encarga de reunir, procesar y almacenar la información obtenida durante una misión espacial, pero ¿de dónde viene la información? En principio, la información es adquirida por otros subsistemas del satélite, como la carga útil, y es transferida a la computadora a través de puertos de entrada y salida, con los cuales se mantiene una comunicación constante entre el SCMI y el resto de los subsistemas.

Un puerto de entrada/salida es la conexión física que permite la transferencia de información entre la computadora y los dispositivos externos. Ahora, una interfaz se define como la colección de los puertos de entrada/salida, la electrónica externa y el software que, combinados, permiten a la computadora comunicarse con el mundo exterior. En general, las interfaces de entrada/salida se pueden clasificar en cuatro categorías: paralelas, seriales, analógicas y de tiempo [38].

### 7.1 Interfaces de entrada/salida paralelas

Una interfaz de entrada/salida paralela permite la transferencia simultánea de datos binarios a través de un grupo de líneas de conexión [38], también conocido como bus de datos. Las interfaces paralelas son rápidas, pero más costosas que las seriales porque requieren de más hardware.

#### *Entrada/salida de propósito general (GPIO)*

Los GPIO son el tipo de puerto más básico en un dispositivo de procesamiento, pueden ser controlados individualmente o como un bus y, como su nombre lo indica, tienen una función genérica tanto de entrada como de salida. Suelen ser utilizados como líneas digitales adicionales para controlar los dispositivos externos a la computadora y algunos pueden tener una o varias funciones alternas, como otros buses, temporizadores o convertidores, permitiendo programar cuál de estas desempeñará. Además, algunos dispositivos permiten configurar los GPIO para leer y escribir en diferentes niveles de tensión lógica.

### 7.2 Interfaces de entrada/salida seriales

Una interfaz de entrada/salida serial permite la transferencia secuencial de datos binarios un bit a la vez a través de una sola línea de conexión [38]. A diferencia de las interfaces paralelas, un bus serial sacrifica velocidad a un menor costo al requerir mucho menos hardware. Sin embargo, para algunas aplicaciones, la falta de velocidad no es una desventaja cuando la confiabilidad es esencial [49]. La arquitectura de la conexión física o topología más utilizada comúnmente en las interfaces de comunicación serial es la de tipo bus, representada en la *Figura 57*, en la cual todos los dispositivos están interconectados mediante un único bus bidireccional, siendo el maestro quien realiza el control principal de todos los nodos o esclavos coordinando en qué momento transmiten información y reciben instrucciones.

Una interfaz serial puede comunicarse de tres modos diferentes, siendo el modo simplex el más sencillo, se refiere a cuando solo se puede transmitir o recibir información. El modo half duplex es cuando se puede transmitir y recibir información, pero una sola a la vez, mientras que el modo full duplex permite ambas acciones simultáneamente. Además, existen dos tipos de transmisión: la transmisión asíncrona, en donde el dato es enviado como una palabra delimitada por secuencias de inicio y final sin un flujo de datos determinado por una señal de reloj común. En cambio, la transmisión síncrona es más rápida al enviar las palabras una tras otra en forma de bloque con inicio y final definido por palabras o bits específicos, además de contar con una señal de reloj para sincronizar el envío de datos entre el emisor y el receptor [49].

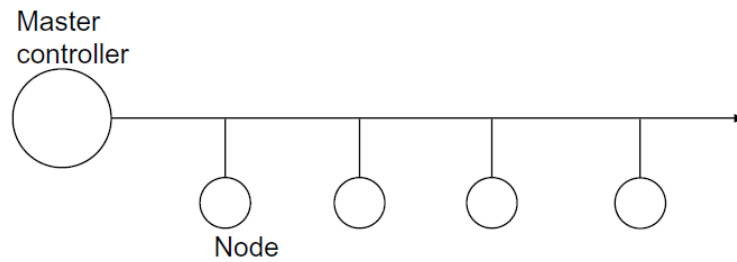


Figura 57. Topología "bus" [49]

### Transmisor-receptor asíncrono universal (UART)

El UART es un bus de comunicación serial asíncrono ampliamente utilizado en procesos básicos y en muchas otras interfaces que puede operar en modo simplex, half duplex y full duplex. El protocolo consiste en el envío de los datos al puerto UART a través de un puerto paralelo almacenándolos temporalmente en una memoria para ser enviados a un buffer mediante el bus interno del UART. Posteriormente, la palabra se carga en un registro de corrimiento donde se añaden los bits de inicio y parada y opcionalmente el bit de paridad<sup>45</sup> para transmitirla en forma serial en formato NRZ<sup>46</sup>. Una vez recibida, la palabra se transfiere en paralelo a un buffer y después al dispositivo de procesamiento receptor [49]. Las palabras están estructuradas por un bit de inicio (estado lógico bajo), de cinco a nueve bits de datos, un bit de paridad opcional y uno o dos bits de parada (estado lógico alto). Cuando el bus se encuentra inactivo se mantiene en estado lógico alto, como se muestra en la *Figura 58*.

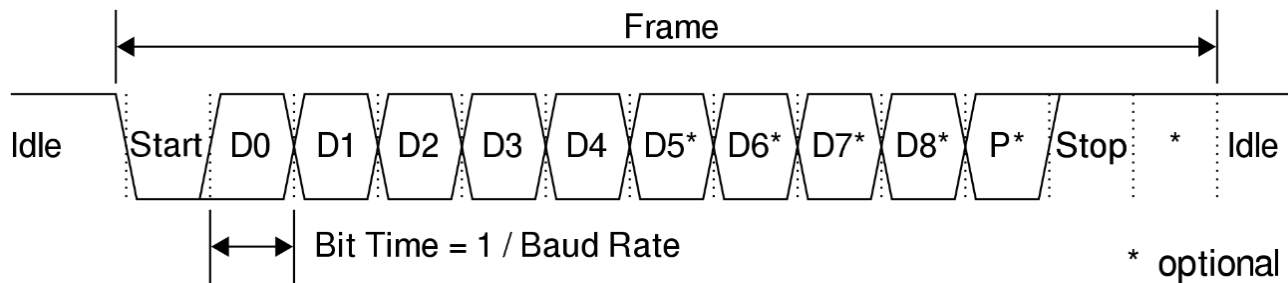


Figura 58. Palabra de datos del bus UART

### Transmisor-receptor síncrono/asíncrono universal (USART)

En pocas palabras, es un UART que puede configurarse para trabajar en modo síncrono añadiendo una señal de reloj. Un USART en modo asíncrono tiene las mismas capacidades que un UART. En modo síncrono, el USART únicamente opera en modo half duplex, puede transmitir a una velocidad fija mayor y transmite los datos en forma de bloques, aunque presenta un funcionalidad más compleja que el UART, también puede generar datos correspondientes a diferentes protocolos.

### Estándar recomendado 232 (RS232)

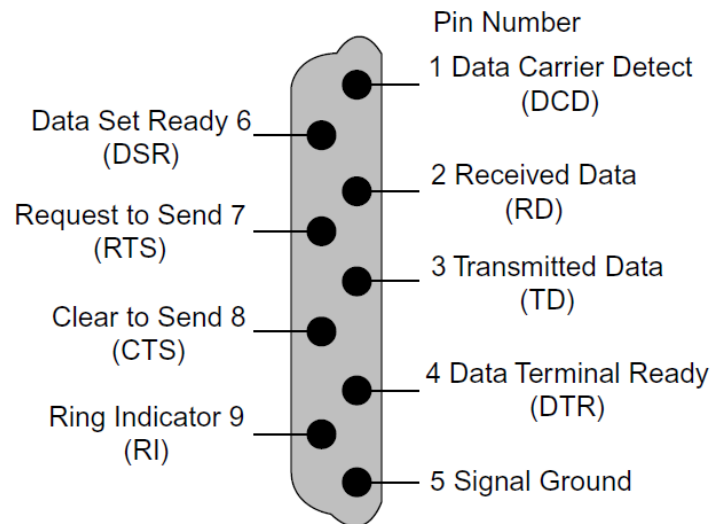
También conocido como EIA/TIA RS232, es un estándar que utiliza la comunicación UART operando tanto en forma asíncrona como síncrona (en su modo modem) y de modo simplex, half duplex y full duplex con una tasa de transmisión de hasta 115.2 kb/s.

El tamaño de palabra consiste en cinco a ocho bits de datos y bits de inicio, parada y paridad. Se implementa mediante el conector DB-25 de 25 terminales, pero en la mayoría de las aplicaciones no se utilizan todas las líneas, por lo que es más común emplearse una versión recortada de nueve terminales conocida como DE-9 o DB-

<sup>45</sup> Es una técnica de detección de errores sencilla. Consiste en un bit que indica si el número de bits con valor lógico alto (1) en una palabra es par o impar.

<sup>46</sup> Non-Return-to-Zero, es una forma sencilla de transmisión de señales digitales en donde se utiliza un nivel de tensión para definir cada estado lógico y este no vuelve a cero entre bits consecutivos de un mismo valor lógico.

9, representada en la *Figura 59*. RS-232 opera con una señal bipolar definiendo el 1 lógico entre los 3 y 15 V y el 0 lógico entre -3 y -15 V.



*Figura 59. Conector DE-9/DB-9 [49]*

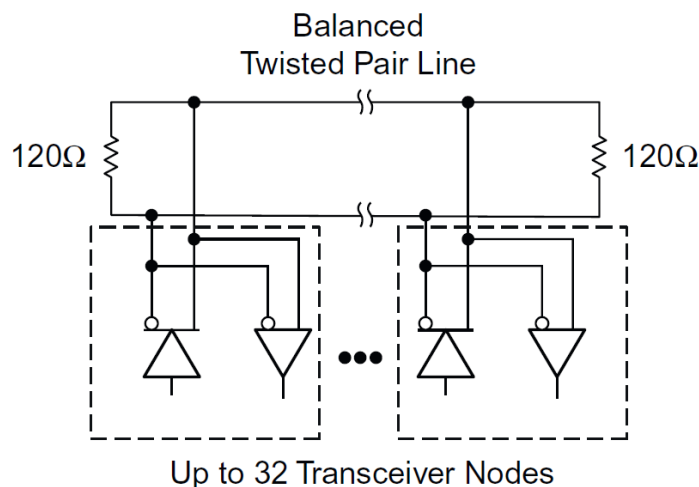
### **Estándar recomendado 422 (RS422)**

El RS422 o TIA/EIA-422 es un estándar que, similar al RS232, utiliza el estándar de transmisión asíncrona del UART y la misma estructura de palabra, está definido principalmente por su señal diferencial, localizando el 0 lógico entre 0.2 y 7 V y el 1 lógico de -2 a -6 V alcanzando una mayor tasa de transmisión de hasta 10 Mb/s. Sus líneas de señal diferencial equilibradas minimizan en gran medida el ruido.

El RS422 suele utilizarse como mecanismo repetidor o extensión para el RS232, transformando las señales de ambos extremos para que sean compatibles con el protocolo correspondiente. No existe un conector definido para el RS422, pero suele implementarse con el conector DE-9.

### **Estándar recomendado 485 (RS485)**

Utiliza el estándar de transmisión asíncrono del UART y la misma estructura de palabra con una tasa de transmisión máxima de 10 Mb/s. La norma RS485 define una interfaz eléctrica compuesta por una señal diferencial, en donde el estado lógico se determina por la polaridad entre ambas señales (A y B), siendo un estado lógico alto cuando la señal A es negativa con respecto a B, y un estado lógico bajo en la situación contraria, con los rangos de tensión de  $\pm 1.5$  a  $\pm 6$  V. RS485 permite la conexión de hasta 32 nodos (véase la *Figura 60*) utilizando un cable de par trenzado, lo que minimiza la interferencia electromagnética.



*Figura 60. Configuración de la conexión RS485 [49]*

**Circuito inter-integrado (I<sup>2</sup>C)**

I<sup>2</sup>C es una interfaz síncrona tipo half duplex sencilla de bajo costo altamente utilizada en sistemas embebidos, se trata de un bus multipunto maestro-esclavo de dos líneas de comunicación de colector abierto o drenaje abierto: la línea de datos (SDA) bidireccional y la línea para la señal de reloj (SCL) únicamente controlada por el maestro, ambas conectadas a resistencias de pull-up (véase la *Figura 61*), en donde el número de esclavos depende de la capacitancia del medio de transmisión, siendo usualmente no mayor a veinte o treinta esclavos por bus y cada uno diferenciado por una dirección de esclavo única. I<sup>2</sup>C puede implementarse como un bus multimaestro alternando el control del bus (mediante arbitraje) para que únicamente opere un dispositivo como maestro a la vez.

La estructura de la comunicación del bus I<sup>2</sup>C consiste en el envío de un bit de inicio, seguido de siete o diez bits de dirección y un último bit de lectura/escritura por parte del maestro. Si existe un esclavo con la dirección enviada este responderá con un bit de acknowledge (ACK). Posteriormente, se enviará en cualquier dirección, dependiendo el caso, un byte de datos seguido de un bit de ACK en repetidas ocasiones hasta que se dé por terminada la comunicación con un bit de parada. El bus utiliza lógica positiva, siendo 0 V para definir un 0 lógico, mientras que el 1 lógico estará definido por la tensión de alimentación del dispositivo.

Una de las principales ventajas del bus I<sup>2</sup>C es su bajo costo, permitiendo controlar una gran cantidad de dispositivos únicamente con dos líneas de conexión, a pesar de que posee menor velocidad y mayor vulnerabilidad a interferencias. La tasa de transmisión máxima del I<sup>2</sup>C es de 3.4 Mb/s en modo bidireccional y de 5 Mb/s en modo unidireccional.

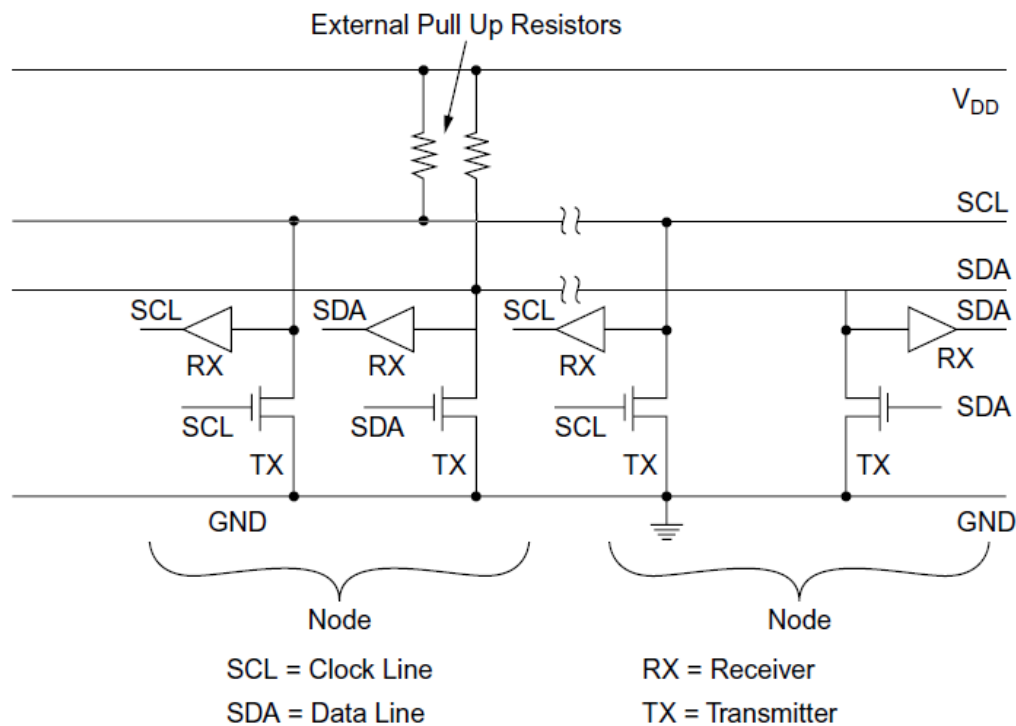


Figura 61. Bus I<sup>2</sup>C con dos nodos [49]

**Interfaz periférico serial (SPI)**

SPI es una interfaz de comunicación serial síncrona del tipo full duplex de alta velocidad con tasa de transmisión de 20 Mb/s a 100 Mb/s que consiste en un bus de tres líneas: master output slave input (MOSI), master input slave output (MISO) y la señal de reloj (SCLK) más una línea de selección (SS) por cada esclavo (véase la *Figura 62*). El protocolo de comunicación del bus SPI consiste en enviar un 0 lógico al esclavo con el que se desea comunicar mediante la línea SS del esclavo correspondiente y manteniendo los otros esclavos en un estado



alto. Posteriormente se pasa al envío de datos bidireccional entre el maestro y el esclavo seleccionado con un tamaño de palabra de 8, 12 o 16 bits. El bus SPI es compatible con los niveles de tensión TTL y CMOS<sup>47</sup>.

Las principales ventajas del bus SPI frente a I<sup>2</sup>C es su mayor velocidad, su simplificado circuito y un menor consumo de energía, aunque necesite de más líneas de conexión.

Cuando el bus SPI no está utilizándose como full duplex, permite transformar las líneas MOSI y MISO en dos señales de entrada/salida seriales (SIO0 y SIO1) como dos interfaces half duplex enviando dos bits por ciclo de reloj, a esta interfaz se le conoce como SPI Dual. Además, también se pueden añadir otras dos líneas de entrada/salida seriales (SIO2 y SIO3) para enviar cuatro bits por cada ciclo de reloj, conocido como SPI cuádruple (QSPI).

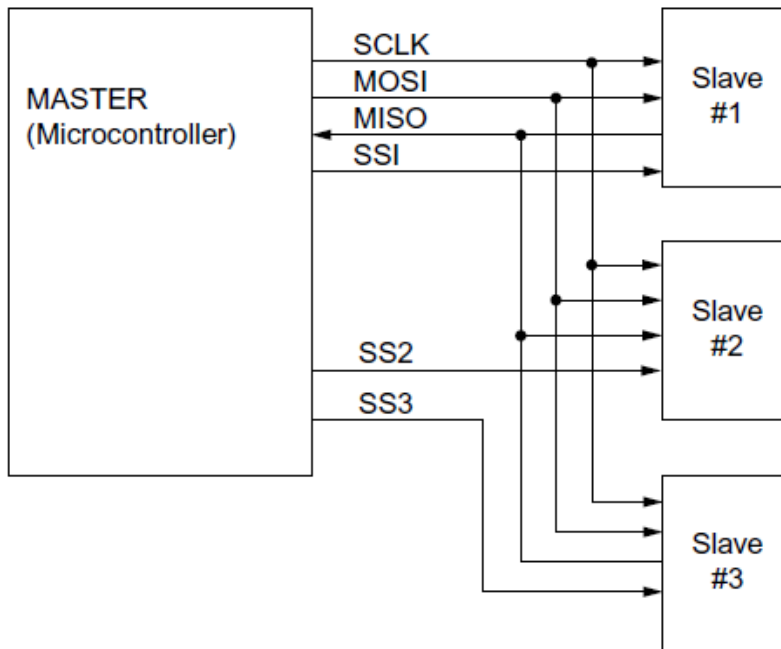


Figura 62. Interfaz Periférico Serial (SPI) con tres esclavos [49]

### Controlador de red de área (CAN)

El bus CAN es una interfaz multipunto de comunicación serial asíncrona multimaestro de bajo costo con una excelente inmunidad al ruido, manejo de datos en tiempo real, manejo de niveles de prioridad y métodos de revisión y corrección de errores, por lo que se trata de un bus de muy alta confiabilidad que alcanza una tasa de transmisión máxima de 1 Mb/s y distancias de hasta 1 km. Consiste en un cable diferencial balanceado con impedancia de 120  $\Omega$ <sup>48</sup> que puede conectar hasta 127 nodos necesitando de un transceiver en cada uno (véase la Figura 63). Utiliza el conector DE-9.

<sup>47</sup> Familia lógica cuyo funcionamiento está basado en transistores de efecto de campo (MOSFET).

<sup>48</sup> Las resistencias de 120  $\Omega$  se utilizan para acoplar las impedancias en el bus de datos, evitando pérdidas por ondas reflejadas.

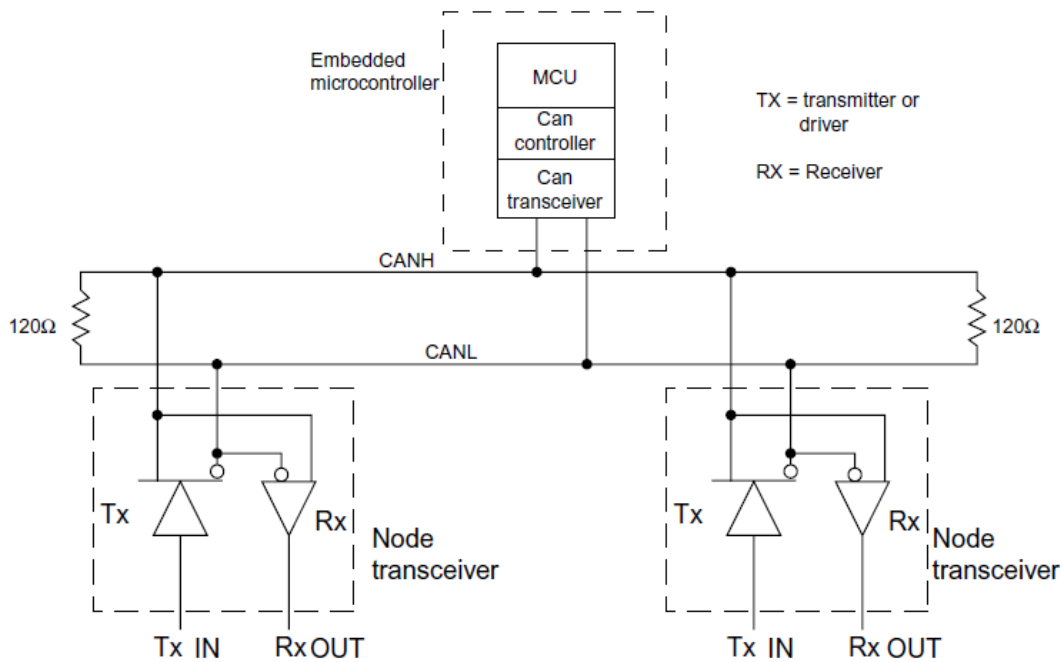


Figura 63. Nodos y transceivers del bus CAN [49]

Como se puede ver en la *Figura 64*, los niveles de tensión que utiliza CAN se conocen como estados recesivo y dominante. Normalmente, el bus se encuentra en modo inactivo en el estado recesivo que suele ser la mitad de la tensión de alimentación, denominado como CANH (estado lógico alto), mientras que el estado dominante o CANL (estado lógico bajo) se localiza a una diferencia de tensión mayor a 1.5 V del estado recesivo.

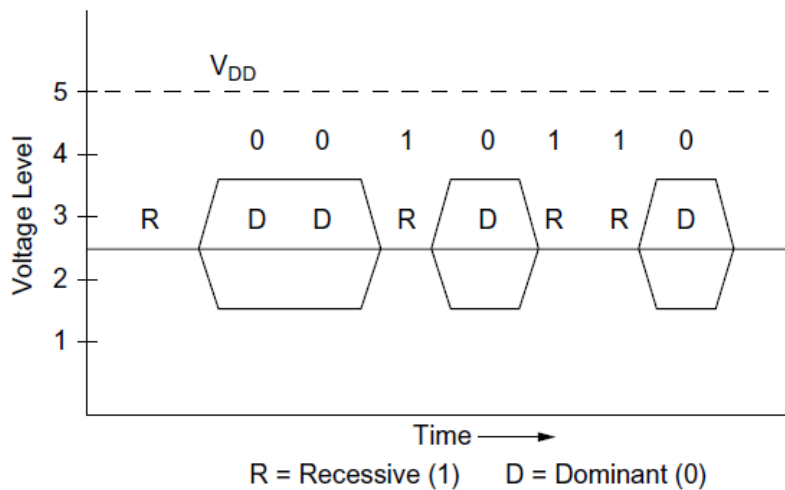


Figura 64. Niveles de tensión recesivos y dominantes del bus CAN [49]

El protocolo de comunicación que utiliza CAN se puede implementar de dos formas, conocidas como CAN 2.0A y CAN 2.0B. La transmisión comienza con un bit de inicio en estado dominante, seguido de 11 bits (para el caso de CAN 2.0A) o 29 bits (para el caso de CAN 2.0B) para el campo de arbitraje, el cual es el identificador (ID). Posteriormente, se envía un bit en estado dominante para definir el envío de una trama de datos o en estado recesivo para definir el envío de tramas de peticiones remotas, seguido de un bit en estado dominante y 4 bits de control en donde se especificará el número de bytes de datos que se enviarán. Después, se envían de 0 a 8 bytes de datos seguidos de 16 bits de código de verificación por redundancia cíclica<sup>49</sup> (CRC) en donde el último bit

<sup>49</sup> Método de detección de errores en donde se calcula el CRC de un mensaje antes de ser enviado, cuyo valor se adjunta a la trama de datos que, al ser recibida, se obtiene de nuevo su CRC, comparándose con el CRC generado previamente. Si ambos CRC son diferentes, significa que el mensaje sufrió alguna perturbación. El bit de paridad es la implementación de un solo bit de un CRC. [1]

debe ser en estado recesivo. Finalmente, se envían 2 bits de ACK (el transmisor envía un bit en estado recesivo y el receptor un bit en estado dominante) y 7 bits en estado recesivo para terminar la comunicación.

El manejo de prioridades ayuda a que, si más de un nodo intenta transmitir al mismo tiempo, se seleccione aquel que tenga la prioridad más alta (valor más bajo en el campo de arbitraje) y posteriormente se atiendan las transmisiones de los nodos pendientes.

### ***Bus serial universal (USB)***

USB es una interfaz implementada en la mayoría de los dispositivos electrónicos que rápidamente ha ido sustituyendo a RS232. Existen diferentes versiones del USB. USB 2.0 consiste en un bus diferencial con topología centralizada con capacidad para conectar hasta 127 dispositivos, siendo estos controlados por un anfitrión, el cual regulará la comunicación en el bus. El protocolo de comunicación está dividido en tres paquetes: el paquete de fichas, en donde se describirá el tipo de transferencia, la dirección en la que viajará la información, siete bits de dirección del dispositivo y cuatro bits para definir el punto final a donde llegará la información; el paquete de datos, que enviará la información seguida de 16 bits de código CRC; y el paquete de estado, que transmitirá los bits de ACK [50].

### ***Señal diferencial de baja tensión (LVDS)***

LVDS, o también conocido como TIS/EIA-644, es únicamente la especificación de la capa física que utilizan otras interfaces seriales, por lo que no especifica un esquema de codificación de bits. Consiste en la especificación de las características eléctricas de una señal diferencial serial de bajo consumo, bajo ruido electromagnético y alta velocidad, alcanzando tasas de transmisión de 1 a 3 Gb/s.

LVDS puede operar de modo half duplex y full duplex utilizando dos cables, además de que también puede emplearse como una interfaz paralela. Maneja un nivel de tensión diferencial de únicamente  $\pm 350$  mV, por lo que tiene un muy bajo consumo de energía. Su estado lógico bajo se establece con un nivel de tensión de 1 V y el estado lógico alto con 1.4 V. El estándar LVDS no especifica una topología de bus, por lo que se generó el estándar LVDS multipunto (M-LVDS), que permite conectar hasta 32 nodos en el mismo bus.

### ***Grupo de acción conjunta de prueba (JTAG)***

JTAG es una interfaz diseñada para probar y verificar diseños de circuito impreso y circuitos integrados de gran escala en un ambiente de producción, también es muy común utilizarlo para la depuración del código porque permite acceder al módulo de depuración del circuito para modificar su código, ingresar programas de prueba y extraer datos de prueba. Consiste en un bus serial de cuatro o cinco líneas diseñado para conectar varios dispositivos en daisy chain<sup>50</sup>. Su tasa de transmisión se encuentra normalmente entre 10 a 50 Mb/s.

Algunos dispositivos tienen la capacidad de utilizar un bus de únicamente dos líneas de conexión alternativo al JTAG, conocido como bus de depuración de cable serial (SWD), el cual utiliza el mismo protocolo.

## **7.3 Interfaces de entrada/salida analógicas**

Una computadora maneja exclusivamente señales digitales, pero sabemos que en el mundo real todas las señales que se perciben son analógicas, por lo que se necesita de un dispositivo capaz de convertir una señal analógica a su forma digital y viceversa. Una interfaz de entrada/salida analógica es aquella que permite la transferencia de información codificada en magnitudes eléctricas, como la tensión, corriente o potencia [38].

### ***Convertidor analógico-digital (ADC)***

Es un dispositivo que cuantifica una señal analógica asignando un valor binario a un valor correspondiente a la magnitud de la señal, siendo normalmente el nivel de tensión o de corriente. Es utilizado como un sistema de adquisición de datos, permitiéndole a la computadora recolectar información del mundo exterior a través de la

<sup>50</sup> Es un esquema de alambrado en donde múltiples dispositivos están conectados en serie de forma lineal o de anillo.

magnitud de la señal. Para convertir la señal digital de regreso a su forma analógica se utiliza un convertidor digital-analógico (DAC, por sus siglas en inglés).

## 7.4 Interfaces de entrada/salida de tiempo

Una interfaz de entrada/salida de tiempo permite la transferencia de datos codificados como un periodo, frecuencia, ancho de pulso o cambio de fase [38].

### *Temporizadores*

Los temporizadores o contadores son circuitos síncronos cuya señal de reloj puede ser el oscilador interno del microcontrolador o provenir de una fuente externa. Entre sus principales funciones se encuentran realizar cuentas determinadas para habilitar interrupciones, generar retardos precisos e implementar señales de reloj.

### *Señal de pulso por segundo (PPS)*

Es una sola línea de salida que manda un pulso una vez por segundo con mucha precisión que individualmente permite a dispositivos externos sincronizarse al tiempo de la computadora y combinada con una fuente de tiempo ayuda a proporcionar la fecha y hora con gran precisión.

### *Modulación por ancho de pulso (PWM)*

Es una señal cuadrada periódica que, al variar su ciclo de trabajo<sup>51</sup>, modifica su tensión promedio permitiendo enviar información y controlar la cantidad de energía que se envía a una carga.

## 7.5 Dispositivos externos

El diseño del SCMI no involucra necesariamente dispositivos externos. Sin embargo, en prácticamente todo satélite se puede encontrar algunos dispositivos que desempeñan funciones necesarias para el sistema, como lo vienen siendo algunos sensores que verifican el estado eléctrico de la computadora y las unidades de medición inercial (IMU) que son parte del ADCS, por lo que se suelen considerar como parte del diseño de un SCMI de propósito general.

### *Reloj de tiempo real (RTC)*

Es el dispositivo encargado de mantener el registro de la hora y fecha reales en el SCMI que, a diferencia de un temporizador interno, cuenta con mayor precisión y menor consumo de energía. A pesar de que la mayoría de los microcontroladores cuentan con un RTC interno se suele integrar uno externo con mejores características, como una mayor tolerancia a fallas y resistencia a la radiación, mayor precisión en misiones de larga duración y diferentes modos de operación. Además de las ventajas ya mencionadas, el utilizar un RTC eficientiza el procesamiento, ya que la computadora no tiene que hacerse cargo de llevar el registro del tiempo real en el satélite.

### *Unidades de medición inercial (IMU)*

Como se mencionó con anterioridad, en el diseño de un SCMI de propósito general se pueden considerar dispositivos que no pertenecen al sistema, pero cuyas funciones son necesarias para cualquier tipo de misión, por lo que se encuentran en todo satélite. Este es el caso de las IMU, término que corresponde a aquellos dispositivos electrónicos que se encargan de medir las fuerzas de un cuerpo para conocer su posición y orientación, como acelerómetros, giroscopios y magnetómetros. Los IMU son los componentes principales del ADCS y su integración al SCMI disminuye los costos y tiempos necesarios para su diseño, aunque no limitan al diseñador a no utilizar otras alternativas que cumplan mejor las especificaciones de su misión.

---

<sup>51</sup> Relación del tiempo en que una señal periódica permanece en estado alto y el periodo de la señal.

## 8 Técnicas de mitigación y tolerancia a fallas

---

Una parte esencial para el diseño de un SCMI es la implementación de técnicas de mitigación y tolerancia a fallas, que tienen el objetivo de asegurar el correcto funcionamiento del sistema cuando este se encuentre en un ambiente espacial o, como mínimo, disminuir las probabilidades de que este sufra alguna condición que le imposibilite desempeñar las tareas para las cuales fue diseñado. Estas técnicas se convierten en la parte fundamental del diseño, ya que en el ambiente espacial se presentan condiciones que pueden alterar el funcionamiento de la computadora, provocando desde errores simples hasta fallas catastróficas que conlleven a un daño en los dispositivos electrónicos, con el cual no se alcancen los objetivos planteados para la misión.

El presente capítulo servirá como base para diseñar un SCMI tolerante a fallas, en donde primero se deben conocer las condiciones a las que se expondrá la computadora una vez que haya sido lanzada al espacio, en donde destaca el ambiente de radiación espacial. En la segunda parte se describen los principales efectos que se presentan en los dispositivos electrónicos al encontrarse dentro de estas condiciones. En la tercera parte se define el concepto de confiabilidad, el cual será importante para determinar la calidad del funcionamiento que proporciona el sistema y, por último, se describen las técnicas de mitigación y tolerancia a fallas observadas en el *Estado del arte* y los conceptos básicos para implementarlas en el diseño de la computadora.

### 8.1 Ambiente de radiación espacial

El ambiente de radiación espacial contiene una gran densidad de partículas cargadas energéticamente, refiriéndose principalmente a electrones, protones y iones pesados, que pueden tener su origen dentro o fuera del sistema solar. Estas partículas de alta energía pueden atravesar el blindaje de una nave espacial induciendo radiación en los componentes electrónicos, lo que conlleva a cambios en su funcionamiento. [51]

#### 8.1.1 Fuentes de la radiación

El ambiente de radiación espacial puede ser dividido en dos diferentes tipos de radiación: la radiación atrapada y la radiación transitoria, en las que se clasifican distintas fuentes de partículas radiactivas.

##### *Ambiente de radiación atrapada*

Son aquellas partículas cargadas, que quedaron atrapadas dentro del campo magnético de la Tierra, confinándose en ciertas regiones del espacio que se extienden aproximadamente desde los 500 hasta los 76 000 km. [52]

##### *Cinturones de Van Allen*

Formados principalmente por electrones y protones, son las zonas en donde el campo magnético de la Tierra ha atrapado la radiación, concentrando una gran cantidad de partículas cargadas altamente energéticas que una vez atrapadas viajan alrededor del planeta formando una superficie toroidal, como se puede apreciar en la *Figura 65*. Existen dos cinturones de Van Allen: el cinturón interior, que va desde los 500 hasta más de 5000 km de altura, en donde se localizan la anomalía del Atlántico Sur<sup>52</sup> (SAA, por sus siglas en inglés) y partículas con una energía menor a 5 MeV; y el cinturón externo, que va desde los 15 000 hasta los 58 000 km, con partículas de hasta 7 MeV y mayor flujo de electrones [51]. La *Figura 65* muestra una representación de los cinturones de Van Allen.

---

<sup>52</sup> Región de alta intensidad radiactiva en donde los cinturones de Van Allen se encuentran a menor distancia de la superficie terrestre, alrededor de los 200 km, producida por una desviación entre el centro del campo magnético y el centro geográfico terrestre.

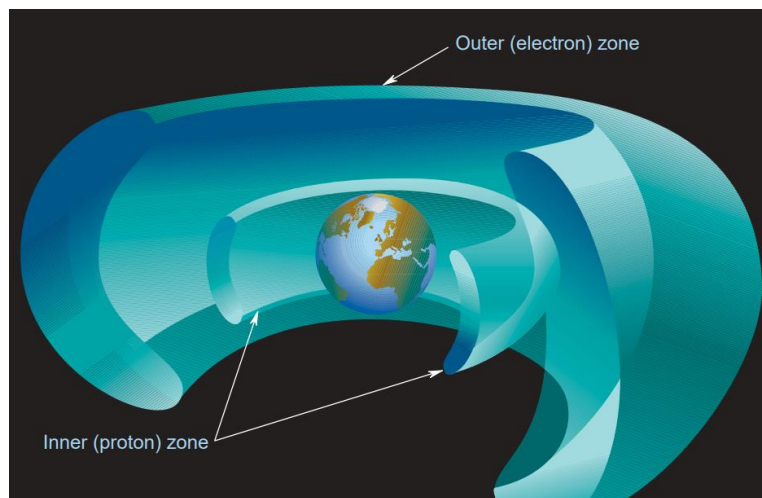


Figura 65. Diagrama de los cinturones de radiación de Van Allen terrestres [53]

### **Ambiente de radiación transitoria**

Se debe a los efectos del Sol y a los rayos cósmicos galácticos (GCR, por sus siglas en inglés), sus principales fuentes son las siguientes:

#### *Viento solar*

Constituido principalmente por electrones y protones de baja energía, es un flujo de partículas que escapan del Sol y afectan principalmente a los componentes montados en el exterior de la nave espacial. [52]

#### *Llamaradas solares*

Son liberaciones de electrones, protones y iones pesados de alto nivel energético derivadas de la actividad solar que inducen diversos efectos en los dispositivos electrónicos.

#### *Radiación cósmica galáctica (GCR)*

Son electrones, protones y iones pesados de muy alto nivel energético provenientes desde el exterior del sistema solar que representan una fuente de radiación significativa en el ambiente de radiación cercano a la Tierra. [52]

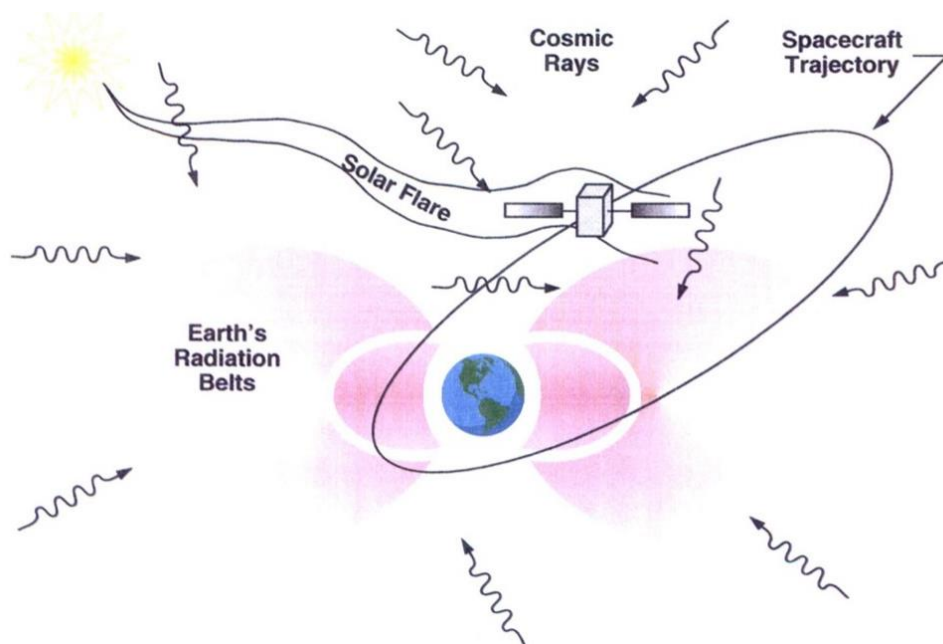


Figura 66. Representación de todos los tipos de radiación que experimenta una nave espacial [52]

## 8.2 Efectos de la radiación en los dispositivos electrónicos

Como se mencionó con anterioridad, el ambiente de radiación espacial presenta condiciones que inducen una variedad de efectos en los dispositivos electrónicos. Estos daños se han convertido en uno de los problemas más desafiantes para los sistemas electrónicos modernos, hablando más específicamente de aquellos diseñados para desempeñar aplicaciones espaciales. [51]

La radiación puede interactuar con los componentes de manera electrónica o nuclear. En las interacciones electrónicas la radiación incidente interactúa con la carga circundante del átomo transfiriéndole energía de radiación, mientras que las interacciones nucleares, con una probabilidad de ocurrencia menor, tienen mayor impacto debido a que la radiación interactúa directamente con el núcleo del átomo. [52]

Los efectos de la radiación se dan de muchas maneras y afectan a los componentes electrónicos de diversas formas. A continuación, se presentan los principales efectos que tiene la radiación en los dispositivos electrónicos.

### 8.2.1 Dosis total de ionización (TID)

Es la ionización acumulada en un material durante un cierto periodo de tiempo, inducida por la exposición homogénea y continua a protones y electrones dentro de un ambiente de radiación, acumulando carga progresivamente hasta alcanzar niveles que producirán efectos negativos y comportamientos erróneos en los dispositivos electrónicos.

Los iones energéticos pueden causar daños a los materiales modificando su estructura al romper y/o reorganizar sus enlaces atómicos [54] alterando las características de los dispositivos electrónicos, resultando en una degradación paramétrica gradual y cambios en su funcionalidad [55]. Produce una variedad de efectos, entre los que destacan:

- ✓ Cambios en los niveles de tensión de umbral de los estados lógicos.
- ✓ Aumento del consumo de energía.
- ✓ Degradación del rendimiento y los materiales.
- ✓ Aumento de sensibilidad a los SEE.
- ✓ Daños permanentes en semiconductores.
- ✓ Pérdida de funcionalidad en los circuitos.
- ✓ Aumento de la sensibilidad de los estados lógicos en las celdas de memoria.

El nivel de TID absorbida que puede soportar un dispositivo electrónico depende de su tecnología, siendo todos los tipos de electrónica susceptibles a la ionización [52]. Depende del material y se mide en rad, que es la unidad de dosis absorbida igual a 100 erg/g, 0.01 Gy o 0.01 J/kg.

### 8.2.2 Efectos de desplazamiento

Es un efecto de la radiación acumulada que, a diferencia de la TID, conduce a daños en los dispositivos electrónicos a través de la pérdida de energía inducida por la transferencia de energía mecánica [56]. Son inducidos por la transferencia de energía no ionizante [51], siendo una interacción de la radiación con el material, tanto electrónicamente como por medio de interacción nuclear directa, donde partículas de alto nivel energético impactan los componentes y la energía se imparte al átomo provocando que, si es lo suficientemente alta como para superar su energía de unión, sufra un desplazamiento de su ubicación normal dentro de la red cristalina hacia diferentes posiciones, perturbando el orden regular de los materiales semiconductores y causando cambios en la operación de los dispositivos electrónicos [52].

Los efectos de la TID y de desplazamiento son causados por la pérdida de energía de una partícula, pero mientras la mayor parte ocurre por ionización, solo el 0.1% se destina a producir desplazamientos [55]. Sus efectos pueden ser similares a los producidos por la TID, pero afectan a una variedad menos extensa de componentes optoelectrónicos [56]. Sus principales efectos son los siguientes:

- ✓ Degradación paramétrica de los dispositivos electrónicos.

- ✓ Pérdida de funcionalidad y rendimiento de los componentes.
- ✓ Cambios en las propiedades ópticas, mecánicas y en la estructura electrónica de los dispositivos [57].
- ✓ Perturbaciones en la estructura de la red cristalina de los materiales, conocidos como daños de desplazamiento (DD, por sus siglas en inglés) [52].

### 8.2.3 Efectos de evento único (SEE)

Son efectos transitorios resultantes del impacto de partículas individuales de alto nivel energético, como rayos cósmicos, protones y iones pesados, en los dispositivos electrónicos [56], en donde la partícula ionizante atraviesa la estructura y viaja a través del material semiconductor depositando un camino de iones [58] que, dependiendo de la energía liberada y la ubicación del impacto, puede producir una variedad de efectos eléctricos agrupados bajo el nombre de SEE.

Un SEE es provocado por el impacto de una sola partícula ionizante, proveniente de un flujo omnidireccional, lo que significa que puede golpear al dispositivo en cualquier momento y ubicación, con dirección y ángulo de incidencia aleatorios, por lo que se trata de un efecto probabilístico en donde el blindaje deja de ser una opción efectiva [54]. Los SEE son agrupados en efectos no destructivos y efectos destructivos, además de clasificarse dependiendo del efecto que producen (véase la *Tabla 10*).

*Tabla 10. Efectos de evento único (SEE) (adaptado de [52])*

<i>Tipo de efecto</i>	<b>Acrónimo</b>	<b>Efecto</b>	<b>Descripción</b>	<b>Mitigación</b>
<i>No destructivo</i>	SEU	Upset	Perturbación o transición del estado lógico de uno o varios bits en zonas de memoria.	Detección y corrección de errores.
	SEFI	Interrupción funcional	Perturbación o transición del estado lógico de uno o varios bits en registros de configuración.	Depende de identificar el efecto.
	SET	Transitorio	Pico de tensión temporal y corriente transitoria que causan un malfuncionamiento.	Diseño de circuitos cronometrados (digital) y añadir cargas RC (analógico).
	SEIDC	Incidencia de corriente oscura	Aumento de la corriente de fuga en un arreglo de dispositivo de carga acoplada <sup>53</sup> (CCD, por sus siglas en inglés).	
<i>Destructivo</i>	SEL	Latch-Up	Impacto de una partícula en una celda parásita (usualmente equivalente a la compuerta de un tiristor), que perturba el funcionamiento incrementando el consumo de corriente.	Interrupción de la alimentación del dispositivo (reinicio).
	SEGR	Ruptura de compuerta	Daño catastrófico de un transistor de potencia.	Reducción del nivel de tensión en el drenaje y la compuerta.

<sup>53</sup> Circuito integrado que contiene un arreglo de condensadores acoplados.



SEB

Burnout

Daño catastrófico de un transistor de potencia.

Reducción del nivel de tensión drenaje-colector o colector-emisor.

En donde los SEU y SEL son los efectos más comunes [52]. La sensibilidad de un dispositivo a los SEE es determinada por dos parámetros: el umbral de transferencia lineal de energía (LET, por sus siglas en inglés) y la sección transversal.

#### *Transferencia lineal de energía (LET)*

A medida que un ion energético viaja a través de un material semiconductor deja una pista de material ionizado [54]. La LET es la cantidad de energía perdida por unidad de longitud de una partícula ionizante que atraviesa un material, medida en  $\text{MeVcm}^2/\text{mg}$ , siendo directamente proporcional al número de pares de electrones generados por unidad de longitud [59]. El umbral LET ( $\text{LET}_{\text{th}}$ ) es un parámetro del material que indica la cantidad de energía que debe sobrepasar la partícula para que esta produzca un SEE en el dispositivo. Si no se alcanza esta cantidad, la partícula no tendrá un efecto inmediato, pero se acumulará en el material para que a largo plazo se produzcan los efectos relacionados a la TID.

#### *Sección transversal ( $\sigma$ )*

Es un parámetro que representa la probabilidad de que el impacto de una partícula provoque un efecto único. Indica la susceptibilidad de un dispositivo a la radiación bajo la fluencia de partículas ionizantes, medida en  $\text{cm}^2$  [51]. Cuanto mayor sea la sección transversal, mayor será la sensibilidad del dispositivo [57].

## 8.3 Confiabilidad

Confiabilidad es, en pocas palabras, la calidad del servicio proporcionada por un sistema en particular [60]. Es la propiedad de un sistema para conservar, en un límite de tiempo establecido, los valores de todos los parámetros que caracterizan su capacidad para desempeñar correctamente sus funciones con base en regímenes y condiciones determinadas. La confiabilidad encapsula y es cuantificada probabilísticamente en términos de seis atributos: fiabilidad, disponibilidad, seguridad, mantenimiento, confidencialidad e integridad, enfatizando en mayor o menor medida cada uno de los atributos dependiendo de la aplicación del sistema.

### 8.3.1 Amenazas de la confiabilidad

Existen tres conceptos que amenazan a la confiabilidad.

#### *Falla*

Una falla es una condición irregular producida por un defecto físico, imperfección, problema de manufactura, error en el diseño o perturbaciones externas, entre otras, que ocurre dentro de algún componente de hardware o software que puede conducir a un daño en el sistema. Una falla se puede describir en cuatro atributos: [60]

- a) *Naturaleza*. Específica el tipo de falla.
- b) *Duración*. Específica el periodo de tiempo en el que una falla se encuentra activa.
  - a. *Falla permanente*. Permanecerá indefinidamente si no se toman acciones correctivas.
  - b. *Falla transitoria*. Falla de un componente que ocurre dentro de un periodo de tiempo determinado.
  - c. *Falla intermitente*. Falla recurrente en un componente que en ocasiones aparece y desaparece.
- c) *Grado*. Especifica si la falla está localizada en un módulo de hardware o software dado o si afecta globalmente el hardware, software o ambos.
- d) *Valor*. Puede ser determinado o indeterminado.
  - a. *Determinado*. Falla cuyo estado permanece sin cambios a lo largo del tiempo a menos que se actúe externamente.
  - b. *Indeterminado*. Falla cuyo estado en algún momento puede ser diferente en algún incremento de tiempo.

**Error**

Un error es la consecuencia y manifestación de una falla [56]. Representado como una acción incorrecta o una desviación de la precisión. La presencia de un error puede ocasionar un daño en el sistema.

**Daño**

Un daño puede ser ocasionado por un error. Es el incumplimiento de alguna acción esperada del sistema. Un daño también es el desempeño de alguna función en una cantidad o calidad subnormal. [60]

**8.3.2 Atributos**

Como se mencionó con anterioridad, la confiabilidad puede determinarse evaluando ciertas cualidades de un sistema, enfatizando cada uno en mayor o menor medida dependiendo de la aplicación.

**Fiabilidad –  $R(t)$** 

Es la probabilidad condicional de que el sistema funcione correctamente durante un intervalo de tiempo  $[t_0, t]$ , dado que el sistema estaba funcionando correctamente en el tiempo  $t_0$ . El nivel de fiabilidad depende de la implementación de tecnología tolerante a fallas. [56]

**Disponibilidad –  $A(t)$** 

Es la probabilidad de que un sistema esté funcionando correctamente y esté disponible para realizar sus funciones en un instante de tiempo  $t$ . [60]

**Seguridad –  $S(t)$** 

Es la probabilidad de que un sistema realice sus funciones correctamente o suspenda sus funciones de manera que no interrumpa el funcionamiento de otros sistemas o comprometa la seguridad de las personas asociadas con el sistema. [60]

**Mantenimiento –  $M(t)$** 

Es la probabilidad de que un sistema fallido pueda someterse a reparaciones para reestablecer su estado operativo en un intervalo de tiempo  $t$ .

**Confidencialidad**

Es el acceso limitado o la restricción a la divulgación no autorizada de la información.

**Integridad**

Es la ausencia de alteraciones inapropiadas de la información y el estado del sistema.

**8.3.3 Medios de la confiabilidad**

Una vez entendido el concepto de confiabilidad, sobra decir que un alto grado de confiabilidad es esencial para que un sistema desempeñe correctamente sus funciones. En el caso de un SCMI, el mantenimiento y/o manipulación que se le puede proporcionar es nulo una vez puesto en órbita, por lo que parte de su diseño es implementar técnicas que le brinden un nivel de confiabilidad aceptable. Para esto, existen cuatro medios con los que se pueden alcanzar mayores niveles de confiabilidad.

- a) *Prevención a fallas*. Técnicas utilizadas para prevenir la ocurrencia de fallas.
- b) *Predicción a fallas*. Técnicas utilizadas para predecir la incidencia futura de fallas y sus consecuencias.
- c) *Eliminación a fallas*. Técnicas utilizadas para reducir la presencia de fallas.
- d) *Tolerancia a fallas*. Técnicas utilizadas para evitar los daños en el sistema ante la presencia de fallas.

En el presente trabajo se diseña un SCMI aplicando los medios de la confiabilidad. Para comenzar, a través del estudio del *Estado del arte* y la investigación del *Marco teórico* se obtienen los criterios de diseño necesarios para prevenir y eliminar fallas mediante técnicas de selección de componentes y técnicas de diseño.

Posteriormente, la predicción a fallas se alcanza evaluando el medio ambiente espacial al que se expondrá el SCMI. Por último, se presentan de manera particular técnicas de tolerancia a fallas.

## 8.4 Tolerancia a fallas

Una vez introducido el ambiente de radiación espacial, que presenta una serie de características que pueden perjudicar el funcionamiento correcto del SCMI, así como sus efectos en los dispositivos electrónicos y el concepto de confiabilidad, es sencillo comprender la necesidad de alguna propiedad que aumente el nivel de confiabilidad del sistema para que pueda operar correctamente dentro de un ambiente espacial, aquí es en donde entra el termino de tolerancia a fallas.

El concepto de tolerancia a fallas se refiere a la habilidad de un sistema para funcionar apropiadamente en el caso de que uno o más de sus componentes fallen, por lo que la presencia de fallas no resulta en un daño completo del sistema [56]. El concepto fundamental de la tolerancia a fallas es la redundancia, que consiste en la replicación de componentes o funciones del sistema orientada a incrementar su confiabilidad, de modo que los elementos duplicados sirvan de soporte para garantizar una operación correcta, aun con la presencia de fallas en uno o varios de ellos, o bien, sirvan de sustituto para continuar desempeñando las tareas que un elemento dañado ya no puede cumplir. Lo anterior se resume en implementar un sistema con componentes redundantes que no le sean necesarios si este estuviera dentro de un ambiente libre de fallas. La idea de aplicar redundancia es diseñar un sistema altamente fiable usando componentes que poseen una fiabilidad estándar [61], que sea capaz de continuar desempeñando sus tareas asignadas aun con la presencia de fallas de hardware y/o software.

Para que un sistema sea tolerante a fallas es necesario implementar técnicas de mitigación y tolerancia a fallas basadas en redundancia aplicando los siguientes conceptos: enmascaramiento de fallas, detección de fallas, localización de fallas, contención de fallas y reconfiguración. El enmascaramiento de fallas es cualquier proceso que previene la introducción de errores en el sistema por parte de una falla. Por su parte, la detección de fallas se logra al reconocer que ha ocurrido una falla para, posteriormente, determinar su ubicación a través de la localización de fallas y, de esta manera, contenerla o aislarla previniendo que sus efectos no se propaguen a otra parte del sistema. Por último, la reconfiguración consiste en un proceso capaz de eliminar una entidad fallida de un sistema y restaurarlo a su estado operacional [60].

En un sistema se pueden distinguir cuatro tipos de redundancia:

- a) *Redundancia en hardware*. Añade elementos de hardware adicionales.
- b) *Redundancia en software*. Añade funciones de software adicionales.
- c) *Redundancia en información*. Añade información adicional.
- d) *Redundancia en tiempo*. Añade tiempo adicional para ejecutar funciones y comparar los resultados.

A continuación, se presentan con mayor detalle los cuatro tipos de redundancia, así como una introducción a las distintas técnicas aplicadas para alcanzar cada una de ellas, destacándose aquellas localizadas en el *Estado del arte*.

### 8.4.1 Redundancia en hardware

Como su nombre lo indica, disminuye el impacto de fallas añadiendo elementos de hardware adicionales. Existen tres maneras de implementar redundancia en hardware en un sistema:

- a) *Redundancia pasiva o estática*. En donde la replicación de elementos va dirigida a que el sistema continúe desempeñando sus tareas asignadas enmascarando las fallas, aunque sufra una disminución en su rendimiento. La redundancia pasiva utiliza el concepto del enmascaramiento de fallas para ocultar la ocurrencia de fallas y prevenir errores, está diseñada para alcanzar tolerancia de fallas sin requerir ninguna acción por parte del sistema. [60]
- b) *Redundancia activa o dinámica*. Evita la reducción de fiabilidad detectando y aislando las fallas tomando acción para reconfigurar al sistema automáticamente y recuperar su estado operacional correcto.

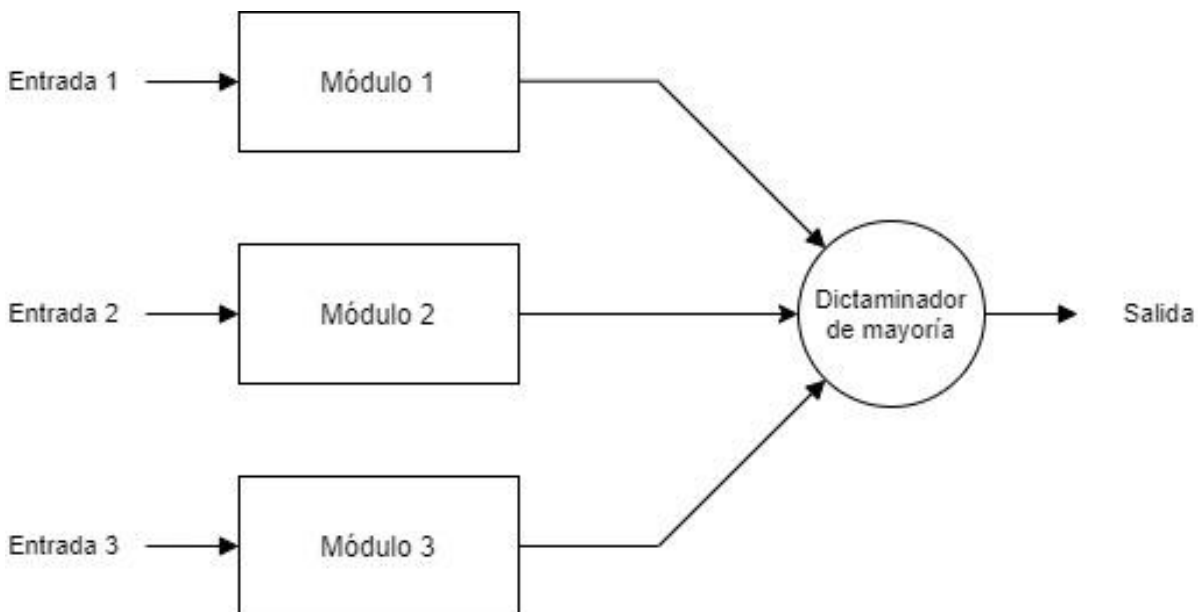
- c) *Redundancia híbrida*. Utiliza enmascaramiento de fallas para prevenir resultados erróneos y detección, ubicación y recuperación de fallas para mejorar la tolerancia a fallas al remover hardware fallido y reemplazarlo con repuestos. [60]

### **Redundancia modular-N**

Es un tipo de redundancia pasiva que utiliza lógica de votación para determinar cuál es el resultado correcto en un sistema, consiste en implementar un número  $N$  de elementos idénticos trabajando en paralelo cuyos datos de salida se entregan a un circuito dictaminador de mayoría, el cual identifica la existencia de alguna falla, la enmascara y decide cual es el resultado correcto después de comparar todos los resultados. Claramente, si  $N=2$ , el dictaminador de mayoría puede identificar la presencia de una falla cuando los resultados no coinciden, pero carece de la capacidad de decidir cuál de los dos módulos se encuentra operando correctamente, por lo que se requieren un mínimo de tres elementos idénticos ( $N \geq 3$ ) para que el dictaminador mayoritario pueda identificar cuál de ellos se encuentra perturbado por una falla. Aplicando este concepto, se pueden implementar un número mucho mayor a dos elementos idénticos para asegurar proceso libre de fallas, pero al aumentar el número de componentes también se perjudica significativamente los costos y requerimientos de potencia del sistema, por lo que el esquema de redundancia modular más utilizado es la redundancia modular triple (TMR).

### **Redundancia modular triple (TMR)**

Como se explica con anterioridad, la TMR es una forma de redundancia en donde tres sistemas o módulos realizan un proceso idéntico, cuyos resultados los recibe un circuito dictaminador de mayoría con una única salida. En el caso de que alguno de los elementos presente una falla que perturbe su resultado, el dictaminador de mayoría detecta una discrepancia en los datos del componente fallido con respecto a los otros dos componentes, por lo que asume que, por mayoría, el resultado correcto es el entregado por estos dos componentes con resultado idéntico. El concepto básico de la TMR se puede observar en la *Figura 67*.



*Figura 67. Concepto básico de la TMR*

Al implementar TMR, se asume que, al presentarse una falla, esta perturba el resultado de un solo componente, ya que es muy baja la probabilidad de que los tres elementos fallen de la misma manera por la presencia de la misma falla. De esta manera, el dictaminador de mayoría presenta un modo de falla único y una falla de este disminuye la fiabilidad de todo el sistema. Para resolver esto, el dictaminador de mayoría debe construirse con un nivel de fiabilidad superior a los módulos votantes, además de poder aplicar TMR con réplica del dictaminador de mayoría para disminuir el riesgo de que falle y TMR múltiple, en donde las salidas de los dictaminadores de mayoría se envían a otros módulos que, a su vez, pasen por dictaminadores de mayoría.

### Repuesto de reserva

Un sistema con repuesto de reserva es un tipo de redundancia activa que consiste en la replicación de un componente  $n$  veces, siendo todas las réplicas idénticas y conectadas en paralelo, como se puede observar en la *Figura 68*. Los repuestos de reserva pueden implementarse mediante dos esquemas: repuesto frío de reserva y repuesto caliente de reserva. En un repuesto frío de reserva, únicamente el componente principal se encuentra operando normalmente hasta que se detecte una falla. Una vez se detecte, el componente se sustituye por el de reserva, para que este continúe desempeñando sus tareas, asumiendo que mientras se encontraba fuera de operación no sufrió ningún tipo de falla. De diferente manera, un repuesto caliente de reserva opera en paralelo con el componente principal y está preparado para asumir sus funciones en cualquier momento.

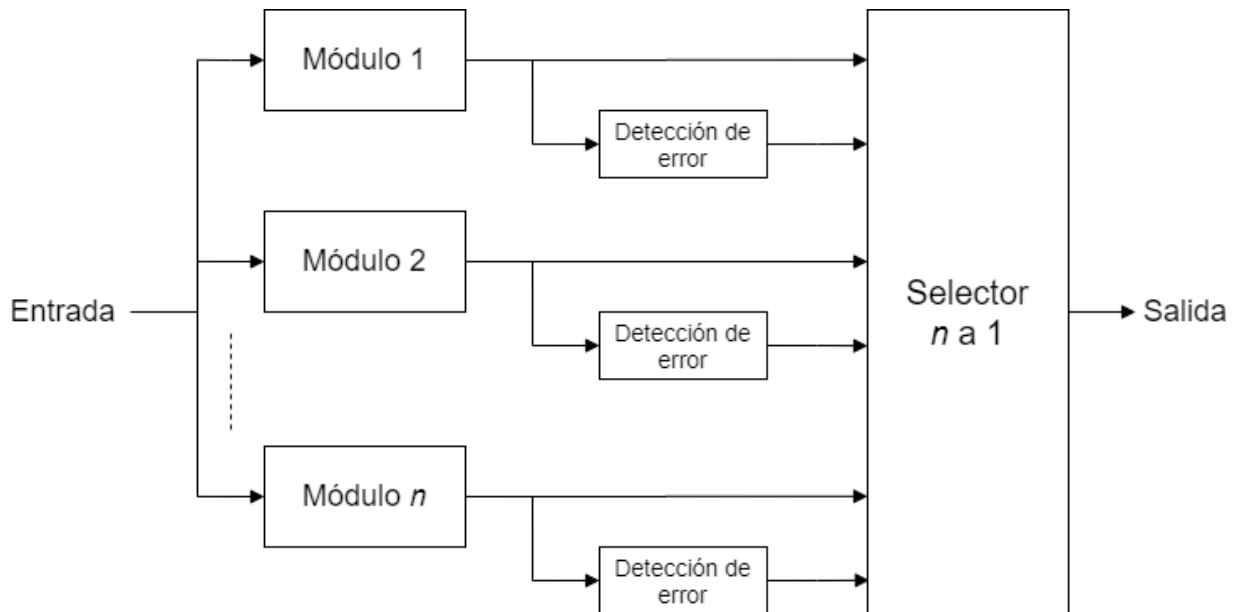


Figura 68. Repuesto de reserva

Haciendo una comparación, los sistemas con repuesto de reserva son superiores a los sistemas en paralelo, sin embargo, su superioridad depende totalmente del circuito acoplador, que debe tener la capacidad de detectar inmediatamente cualquier tipo de falla en el componente primario para intercambiarlo con el componente de reserva, de esta forma este se convierte en el elemento activo. [62]

### Temporizador watchdog

En un satélite, la computadora realiza todos los procesos y, en caso de alguna falla, debe tener la capacidad de reaccionar automáticamente de modo que pueda continuar con su operación, ya sea enmascarando una falla o reparando el sistema de un estado fallido. El temporizador watchdog es, como su nombre lo indica, un vigilante, el cual realiza una cuenta específica que la computadora reinicia constantemente para evitar que pase un tiempo determinado. Mientras la computadora funcione correctamente, el temporizador watchdog nunca llegará a este tiempo, pero en el caso de que, por una falla de hardware o un error de software, la computadora falle o entre en un estado de reposo incorrecto y no se reinicie el contador, el temporizador watchdog genera una señal de tiempo de espera, con la cual da inicio a tomar acciones correctivas para recuperar el estado operacional normal de la computadora, que pueden ser desde iniciar un proceso de recuperación hasta reiniciar la computadora.

## 8.4.2 Redundancia en software

Un sistema redundante en software es aquel que desempeña las mismas funciones a través de diferentes elementos de software para incrementar la fiabilidad de los procesos. A diferencia de la redundancia en hardware, no es factible integrar dispositivos con replicas idénticas del mismo programa, debido a que, si alguno de ellos presenta un error de software, los otros dispositivos también lo harán. Por lo tanto, para proporcionar mayor fiabilidad en software, se deben escribir diferentes versiones independientes del software. [62]

### ***Versión N de programación***

Una vez explicada la redundancia modular-N de hardware, es sencillo entender que se trata de un grupo de versiones del mismo programa cuyos resultados son comparados para determinar cuál es el resultado correcto. Como se explica con anterioridad, la diferencia radica en que todas las versiones de software son totalmente independientes para evitar errores comunes entre ellos. La *Figura 67* sirve para representar este esquema, en donde cada módulo pasa a ser una versión del programa, cuyos resultados también pasarán a un sistema dictaminador de mayoría. Para diseñar diferentes versiones del mismo software, se necesitan grupos independientes de desarrollo, distintos procesos y probablemente diferentes lenguajes de programación. [62]

### **8.4.3 Redundancia en información**

En el capítulo 8.2. *Efectos de la radiación en los dispositivos electrónicos*, se explican los principales efectos que el ambiente de radiación espacial induce en los dispositivos electrónicos, los cuales perturban su funcionamiento y provocan errores en la información transmitida de un componente a otro. Con esto en mente, ¿cómo saber si la información obtenida es correcta o no?, tratándose de un sistema colocado en un ambiente hostil en donde los humanos no pueden estar presentes y la ocurrencia de fallas disminuiría la utilidad de la misión, el sistema debe estar preparado para identificar cualquier tipo de falla y, en caso de la presencia de algún error, corregirlo. Para aumentar la confiabilidad de la información, se utilizan técnicas de redundancia presentadas como códigos de detección y corrección de errores (EDAC), permitiendo así la entrega confiable de la información.

Para adentrarse en las técnicas de detección y corrección de errores, primero es importante entender el concepto de código. Un código es un conjunto de  $n$  elementos que satisfacen algún grupo de reglas específicas, en donde  $n$  representa el tamaño del código. Estos elementos reciben el nombre de palabras de código, que a su vez están construidas por una cadena de símbolos con un significado bien definido. Las palabras de código pueden ser validas o no validas dependiendo de si cumplen o no con las reglas del código, de modo que la presencia de una palabra no valida en una transmisión de datos significa que ha ocurrido algún error y los datos no son confiables. Ahora bien, un código sirve para representar información, por ejemplo, la computadora se comunica y ejecuta sus instrucciones con código binario, en donde se almacena y envía información con cadenas de un tamaño determinado utilizando los símbolos 0 y 1. Otros dos conceptos importantes son la codificación, que consiste en la transformación de datos en una palabra de código, con el objetivo de transmitir y/o almacenar su información; y la decodificación, siendo el proceso contrario que transforma de regreso las palabras de código para recuperar sus datos.

Entrando en materia, las técnicas de detección y corrección de errores son esquemas que añaden redundancia en información en un mensaje, de modo que, al recibirlo, el dispositivo receptor sea capaz de verificar su consistencia y reconstruir el mensaje original [62]. Inicialmente, se puede observar el proceso de detección de errores como una verificación de cada palabra recibida para determinar si pertenecen al grupo de palabras ilegales [62] y, en caso de que así sea, proceder a corregir sus errores a través de un código de corrección de errores (ECC). Lo anterior se puede realizar de manera sistemática, en donde se envían los mensajes originales adicionados con bits de verificación derivados de los datos del mensaje con un algoritmo determinístico; y no sistemática, en donde el mensaje original se envía codificado. [62]

La efectividad de un código de detección y corrección de errores está dada por la distancia de código. Para entender esto, es más sencillo comenzar con la distancia de Hamming ( $d$ ), que es el número de símbolos por los cuales dos secuencias son diferentes. En otras palabras, sirve como métrica para comparar dos secuencias de datos binarios del mismo tamaño, indicando el número de bits en que estas difieren entre sí. Ahora bien, la distancia de código es la distancia mínima de Hamming presente entre palabras validas distintas del mismo código, lo que significa que entre mayor sea la distancia de código, menor es la probabilidad de que un grupo de errores convierta una palabra de código valida en otra también valida, dificultando la detección de errores. Visto desde otro modo, en un código con distancia de uno, no hay forma de detectar un error, ya sea simple o múltiple, si cualquier cambio en la palabra de código la transforma en otra palabra de código valida. Para evitar esto, se pueden implementar esquemas de detección de errores en un código al añadir, por ejemplo, bits de paridad. [62]

### **Bit de paridad**

Es el esquema de código de detección de errores más sencillo. Se trata de añadir un bit redundante en un mensaje para indicar si el número de bits con valor de 1 es par o impar, dependiendo de su configuración. Una vez recibido el mensaje se verifica que el número de bits en 1 sea el correcto según el bit de paridad, en caso contrario, se asume la presencia de un error. Su limitante principal es que, si se presenta un número par de errores en una misma palabra, el número de bits en 1 seguirá siendo del mismo tipo, ya sea par o impar, por lo que no se detecta el error. En otras palabras, la ocurrencia de un número par de errores en una misma palabra de código la transforma en otra palabra de código válida, por lo que no se detecta ningún error.

### **Códigos de duplicación**

Se trata de un esquema bastante sencillo encontrado comúnmente en sistemas de memoria [60], en donde las palabras de código se forman al duplicar parcial o completamente la información original del mensaje. Claramente, si un bit duplicado difiere en su valor, se ha presentado un error. La principal desventaja de los códigos de duplicación es la reducción de la tasa de información.

### **Códigos cíclicos**

Los códigos cíclicos son esquemas de bloques que utilizan propiedades algebraicas para detectar y corregir errores. Su característica fundamental es que cualquier cambio en el final de una palabra de código produce otra palabra de código. Un código cíclico opera con un polinomio generador,  $G(X)$ , de grado  $(n-k)$ , donde  $n$  es el número de bits de la palabra de código completa producida por  $G(X)$ , y  $k$  es el número de bits del mensaje con la información original. [60]

Los códigos cíclicos operan en función de la siguiente ecuación:

$$R(X) = D(X)G(X) + S(X)$$

En donde:

$R(X)$	=	palabra polinomio de código válida.
$D(X)$	=	mensaje polinomio original.
$G(X)$	=	polinomio generador.
$S(X)$	=	residuo.

Siguiendo la ecuación anterior, una palabra de código cíclico válida  $R(X)$  se obtiene multiplicando el mensaje original  $D(X)$  por el polinomio generador  $G(X)$ , lo que significa que  $R(X)$  debe ser un múltiplo exacto de  $G(X)$  y que el residuo  $S(X)$  debe ser igual a 0. De esta manera, para regresar a  $D(X)$  solo hay que dividir  $R(X)$  entre  $G(X)$ . En el caso de que, al hacer la división, de como resultado un  $S(X)$  diferente de 0, significa que ha ocurrido un error.

Un código cíclico de detección de errores utilizado muy frecuentemente es la verificación de redundancia cíclica (CRC), que es una función hash<sup>54</sup> que agrega un valor de verificación redundante a los bloques de datos en función del residuo de la división de polinomios.

### **Códigos de Hamming**

Los códigos de Hamming son una familia de EDAC utilizados frecuentemente en aplicaciones espaciales. Se tratan de códigos que añaden una cantidad  $c$  de bits de verificación para detectar más de un único error en una palabra codificada y, si se utilizan suficientes bits de verificación, corregir algunos de esos errores [62]. Además de la distancia de Hamming, es necesario entender las capacidades de detección y corrección de errores de un código basado en la distancia de Hamming, aplicadas a códigos lineales<sup>55</sup>.

<sup>54</sup> Función que tiene como entrada una cadena de elementos de cualquier longitud y como salida devuelve una cadena de longitud fija.

<sup>55</sup> Códigos en donde la diferencia y adición entre dos palabras de código da como resultado otra palabra de código [62].

Como se ha mencionado, el proceso de detección de errores comienza en la codificación, en donde se añaden  $c$  bits de verificación en un mensaje para convertirlo en una palabra de código, de manera que al decodificarse se detecten posibles errores y, en caso de contar con bits de verificación suficientes, corregir algunos errores [62]. La condición para que uno o varios errores puedan detectarse en el proceso de decodificación es que dichos errores conviertan la palabra de código legal en una ilegal. Para detectar una cantidad  $D$  de errores, la distancia de Hamming ( $d$ ) debe ser mayor que  $D$  por al menos uno, como se muestra en la siguiente ecuación:

$$d \geq D + 1$$

En donde:

- $d$  = distancia de Hamming de un código.  
 $D$  = Número de errores que un código puede detectar.

Esta relación indica que el código de detección más simple, capaz de detectar un único error, debe contar con una distancia de Hamming de 2. En otras palabras, si  $D = 1$ , todo error transforma una palabra de código válida en otra palabra de código válida, imposibilitando su detección. Una vez detectado un, el proceso de corrección de errores se generaliza seleccionando como palabra correcta a aquella palabra de código legal más cercana al mensaje codificado, de modo que si se tiene un código con  $D = 1$ , no se puede saber qué palabra es la correcta dado que un único error entregaría una palabra de código ilegal con una distancia de 1 entre dos o más palabras de código válidas. De este modo, si se tiene un código con  $D = 2$ , un único error resultaría en una palabra de código ilegal, localizada a una distancia de 1 entre la palabra correcta y de 2 entre el resto de las palabras válidas, por lo que se asume que la palabra legal más cercana corresponde al mensaje correcto. [62]

Lo anterior indica que la relación entre el número de errores que pueden ser corregidos por un código ( $C$ ) con una distancia de Hamming ( $d$ ) determinada, está dada por la siguiente ecuación:

$$d \geq 2C + 1$$

En donde:

- $d$  = distancia de Hamming de un código.  
 $C$  = Número de errores que un código puede corregir.

El algoritmo general de los códigos de Hamming consiste en añadir  $c$  bits de verificación, cada uno obtenido de la partición de los bits de datos en grupos de paridad para determinar un bit de verificación para cada grupo. El algoritmo más utilizado es el código de Hamming(7,4), que corresponde a palabras con 4 bits de datos ( $m = 4$ ) y 3 bits de verificación ( $c = 3$ ), dando como resultado palabras de código de 7 bits ( $n = 7$ ) capaz de detectar hasta dos errores y corregir un único error.

#### 8.4.4 Redundancia en tiempo

Una cuarta forma de redundancia es la que se obtiene a través del tiempo, dando un tiempo adicional para ejecutar tareas de tolerancia a fallas. A diferencia de la redundancia en hardware y en información, la redundancia en tiempo ahorra recursos reduciendo la cantidad de elementos de hardware adicionales al añadir tiempo de ejecución en sus tareas, reflejándose en la repetición de procesos determinados para detectar y corregir errores. En muchas aplicaciones, el tiempo es mucho menos importante que el hardware porque este último es una entidad física que impacta en el tamaño, masa, consumo de potencia y costos, mientras que el tiempo, por su parte, puede estar fácilmente disponible en algunas aplicaciones, recordando que la selección del tipo de redundancia se realiza en función de la aplicación [60].

##### *Detección de fallas transitorias y permanentes*

Como se mencionó anteriormente, el concepto básico de la redundancia en tiempo es la repetición de procesos de software de manera que permita la detección de fallas, estos pueden ser idénticos con la finalidad de localizar la presencia de alguna falla. Claramente, si los resultados son idénticos en todas las repeticiones, se



asume que los resultados son correctos, pero si existe alguna discrepancia entre ellos, el proceso de cálculo se repite de nuevo para observar si la falla continua o desaparece. En caso de que se trate de una falla transitoria, esta desaparece después de un cierto tiempo, por lo que los resultados vuelven a coincidir y el proceso continúa trabajando normalmente, pero si la falla persiste, se detecta la ocurrencia de una falla permanente, por lo que se procede a reconfigurar o remover la entidad fallida.

## *Sección IV*

---

# *Diseño y construcción*

Una vez identificado el *Estado del arte* de los SCMI, visto desde un enfoque comercial y de investigación, seguido de la investigación preliminar para introducir los conceptos fundamentales necesarios para el diseño de un sistema propio, se procede a entrar en la segunda etapa de la *Metodología de diseño*, el diseño conceptual. En este capítulo se define el concepto de diseño, que consiste en el planteamiento de la arquitectura, los manejadores del sistema, los requisitos críticos de diseño y el concepto de la misión; la definición de bloques funcionales, refiriéndose a la división conceptual del sistema en bloques funcionales agrupando los requisitos y especificaciones en sistemas más pequeños con funciones específicas; el planteamiento de los requisitos del sistema, tratándose de la definición de los requisitos a nivel de bloque funcional; y por último, el diseño conceptual, en donde se desarrolla el diseño conceptual de cada bloque funcional.

Para comenzar, ¿qué es diseño? El diseño es, en esencia, el arte de saber aplicar los conocimientos científicos para buscar y localizar la mejor solución a un determinado problema. Durante el proceso de diseño, el diseñador debe aplicar su creatividad, conocimiento y experiencia para identificar el problema que pretende resolver, además de las características y necesidades que debe satisfacer con su propuesta. De esta forma, el diseñador tomará las decisiones pertinentes fundamentadas en el *Estado del arte* y la investigación preliminar para determinar la mejor manera con la que solucionará el problema. La primera etapa del proceso de diseño es el diseño conceptual, en donde se sientan las bases de cómo se va a diseñar considerando los requisitos y especificaciones del sistema, correspondiendo a la etapa más creativa del desarrollo.

### 9.1 **Definición del concepto de diseño**

Entrando en el diseño conceptual, el primer paso es definir el concepto de diseño. En esta parte se comenzará con la toma de decisiones, las cuales se harán más específicas y detalladas conforme se avance en el proceso de diseño, pero siempre fundamentando con hechos el por qué se llegó a cada una de ellas. En el concepto de diseño se plantea la arquitectura de diseño, los manejadores del sistema, los requisitos críticos y el concepto de la misión. Para esto, es esencial comenzar con los objetivos generales y específicos para entender qué es lo que se está desarrollando y qué es lo que se busca con ello.

Empezando por el objetivo general, planteado desde el principio del trabajo, se busca desarrollar un SCMI de propósito general para nanosatélites bajo el estándar CubeSat 3U. Esta decisión se fundamenta en el *Estado del arte*, en donde se observa con claridad que el estándar CubeSat 3U destaca entre todos los tipos de satélites lanzados en los últimos años y que, al parecer, esta tendencia continuará en los próximos tres años como mínimo, indicando que diseñar bajo el estándar CubeSat garantiza que el sistema cumpla con las necesidades del factor de forma encontrados actualmente en la industria. A su vez, parte del objetivo es diseñar el SCMI empleando principalmente componentes COTS, siendo estos la base del diseño moderno de nanosatélites gracias a la combinación de su bajo consumo de energía, costo mínimo y amplia disponibilidad [16]. Sin embargo, estos componentes no están diseñados para tener un nivel de confiabilidad adecuado para cumplir misiones espaciales, por lo que, para mantener un funcionamiento correcto, es necesario implementar técnicas de mitigación y tolerancia a fallas disminuyendo el riesgo de perder el satélite al exponerse al ambiente de radiación espacial en LEO. Con esta descripción, queda definida la idea general de hacia dónde se quiere llegar, por lo que procedemos a especificar la arquitectura de diseño.

Antes de definir el concepto de arquitectura de diseño es importante saber que un sistema puede considerarse en términos de elementos funcionales y físicos. Los elementos funcionales son las operaciones individuales que contribuyen al rendimiento general del sistema, mientras que los elementos físicos son los componentes que realizan las funciones del sistema. Tanto los elementos funcionales como los físicos se describen en un principio en términos generales o como esquemático, definiéndose con mayor detalle a medida que se avanza en el proceso de diseño. De esta manera, la arquitectura de diseño es el esquema modular<sup>56</sup> en donde los elementos funcionales

---

<sup>56</sup> Característica de un sistema que le permite ser diseñado u observado como la integración de subsistemas con tareas específicas diferentes que interactúan entre ellos para alcanzar un objetivo común.

se asignan a los elementos físicos, por lo que el sistema se estructura en conjuntos de elementos físicos organizados en términos de lo que hacen y lo que representan para el resto del sistema. [63] La división en conjuntos de elementos físicos y funcionales se realizó al definir bloques funcionales, presentados en el *Estado del arte*, que se explicarán con mayor profundidad más adelante.

Como siguiente paso del concepto de diseño, se encuentran los manejadores del sistema. Estos se refieren a los requisitos que afectan de manera general al diseño. En otras palabras, son los requisitos que brindan un primer acercamiento al proceso de diseño basándose en las necesidades, controlando hacia dónde va dirigido el desarrollo, aun sin la necesidad de elaborar una lista de métricas ni especificaciones. ¿A qué se refiere lo anterior? Es normal confundir el concepto de requisitos con especificaciones. Explicándolo brevemente, los requisitos son los parámetros que el sistema debe satisfacer, mientras que las especificaciones son los parámetros que describen cómo el sistema debe trabajar [38]. En otras palabras, los requisitos parten de las necesidades del cliente en términos de calidad subjetiva. Contrario a ellos, las especificaciones detallan en términos precisos y cuantitativos lo que el sistema debe hacer, definiendo las variables clave de diseño [63]. Las especificaciones se dividen en dos grupos: las especificaciones objetivo, establecidas después de identificar los requisitos sin considerar las restricciones tecnológicas; y las especificaciones finales, que se definen más adelante después de evaluar las restricciones técnicas reales [63]. En este caso, el requisito parte del objetivo general, como una computadora de vuelo que pueda desempeñar una gran variedad de misiones en LEO a bajo costo. De manera similar, las especificaciones objetivo corresponden a los objetivos particulares:

- ✓ Diseñar un SCMI para nanosatélites bajo el estándar CubeSat 3U que cumpla con las especificaciones mecánicas y eléctricas establecidas en la especificación de diseño CubeSat (CDS).
- ✓ Diseñar e integrar el hardware de un SCMI para nanosatélites bajo el estándar CubeSat 3U empleando principalmente componentes COTS y tomando como base el estado del arte de los SCMI.
- ✓ Diseñar un SCMI para nanosatélites bajo el estándar CubeSat 3U implementando técnicas de mitigación y tolerancia a fallas, que proporcionen niveles de confiabilidad adecuados para cumplir misiones en LEO.

De lo anterior, se puede definir como requisito crítico el diseño del sistema utilizando principal o totalmente componentes COTS. Esta decisión se fundamenta con los bajos costos y alta disponibilidad de estos componentes que, a pesar de no ser diseñados para aplicaciones espaciales, su excelente rendimiento a bajo consumo de energía combinado con técnicas rudimentarias de tolerancia a fallas brinda niveles de confiabilidad adecuados para misiones espaciales de corta duración en LEO. [16]

### 9.1.1 Medio ambiente de radiación espacial

Todo sistema espacial es diseñado con base en los requisitos y especificaciones de la misión para la cual se están desarrollando. En este caso se pretende diseñar un SCMI de propósito general, por lo que no existe una misión específica definida. Al referirse al sistema como uno de propósito general significa que sus características y desempeño cumplen con las necesidades de una gran variedad de misiones realizadas en el estándar CubeSat 3U. De esta forma, las especificaciones que debe de satisfacer están dadas principalmente por los SCMI identificados en el *Estado del arte* y se presentan más adelante. Sin embargo, una parte integral del proceso de diseño es la predicción del entorno para la electrónica [52]. El definir una misión espacial no solo se refiere a su propósito, sino también a las condiciones a las que se expondrá el sistema, con las cuales, se determina la lista de requisitos relacionados con su rendimiento y nivel de confiabilidad bajo la presencia de partículas cargadas energéticamente en el ambiente de radiación espacial.

Para determinar las condiciones del ambiente de radiación espacial, en el presente trabajo se utiliza el Sistema de Información del Entorno Espacial (SPENVIS, por sus siglas en inglés) de la Agencia Espacial Europea (ESA, por sus siglas en inglés). SPENVIS es una interfaz en línea para modelos del entorno espacial y sus efectos; incluyendo rayos cósmicos, cinturones de radiación natural, partículas de energía solar, plasmas, gases y “micro-partículas”. [64] A continuación, se presenta el procedimiento para la elaboración de la simulación en la plataforma SPENVIS y los resultados obtenidos.

Presentando el inicio de la plataforma SPENVIS en la *Figura 69*, es necesario comenzar generando la órbita con la que volará el sistema. Para esto, dentro del menú de **generador de coordenadas**, se selecciona la opción de **trayectoria de la nave espacial**.

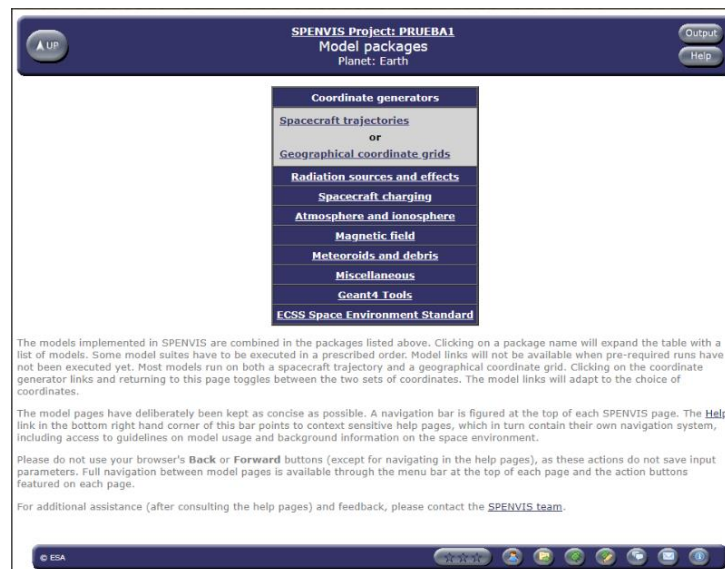


Figura 69. Menú inicial de SPENVIS

Como se aprecia en la *Figura 70*, se selecciona un año para la duración de la misión, que servirá para realizar las primeras pruebas del prototipo del SCMI, recordando que se está diseñando bajo un estándar CubeSat, que son satélites fabricados para cumplir misiones de corta duración. Es importante mencionar que la versión de SPENVIS utilizada en esta simulación no permite la habilitación de la **presión de la radiación solar** y del **arrastre atmosférico**, opciones que a largo plazo modificarán la trayectoria de la órbita. En otras palabras, al no contar con estas opciones se obtendrán los resultados de una misión ideal.

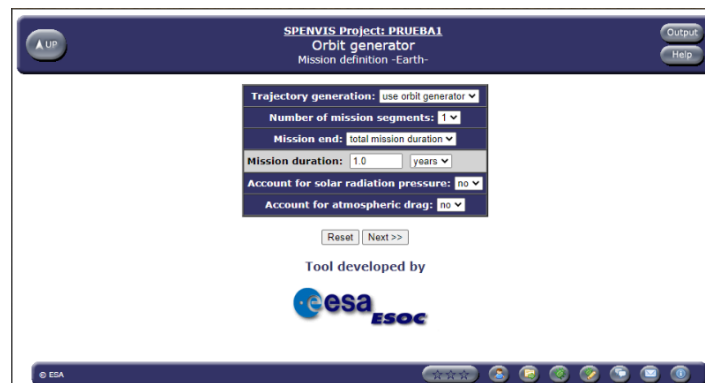


Figura 70. Generador de órbita en SPENVIS

A continuación, como se observa en la *Figura 71*, se pueden generar diferentes tipos de orbitas. En este caso, se selecciona una órbita solar sincrona o helio sincrona (heliosynchronous, en inglés), debido a que es el tipo de órbita típico utilizado en satélites pequeños en órbitas bajas. Una órbita solar sincrona es una órbita con una inclinación cercana a la de una órbita polar (inclinación cercana a  $90^\circ$ ) que se caracteriza porque el satélite siempre cruza el ecuador a una hora determinada, conocida como hora local del nodo ascendente<sup>57</sup>. El siguiente parámetro importante se encuentra en la altura de la órbita, definida en 800 km. Actualmente se vuelan muchos satélites pequeños a una altura similar a la ISS (400 km). De esta manera, una órbita de 800 km es considerada muy alta para satélites pequeños, por lo que seleccionar esta altura asegura que el sistema pueda volar en las órbitas típicas de los satélites pequeños (menores a 800 km), además de que los vehículos de lanzamiento sí

<sup>57</sup> Es el punto donde el satélite cruza el plano del ecuador desde el hemisferio sur hacia el hemisferio norte.

pueden llegar a dicha altura, así que, en dado caso, es posible colocar un satélite en una órbita similar. Además de la altura, se define la hora local del nodo ascendente en las 12 horas. Un detalle importante es la fecha de lanzamiento y la duración de la misión, ya que, a partir de los modelos de radiación solar, se obtiene una estimación de la cantidad de radiación que se tendrá para esas fechas, a la cual se expondrá el sistema, realizando un pronóstico de la actividad del medio ambiente espacial presente durante la misión.

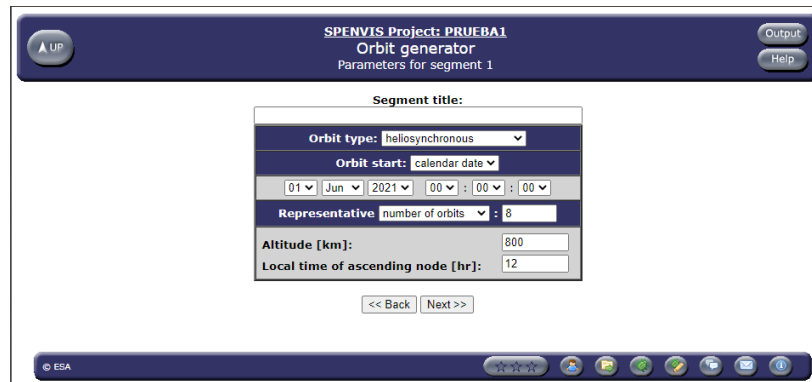


Figura 71. Parámetros de la órbita en SPENVIS

En resumen, se definió una órbita solar síncrona de 800 km de altura con una hora local del nodo ascendente de las 12 horas. En la *Figura 72* se observa el tipo de órbita, la fecha de inicio y el número de órbitas representadas en la simulación, siendo un total de ocho.

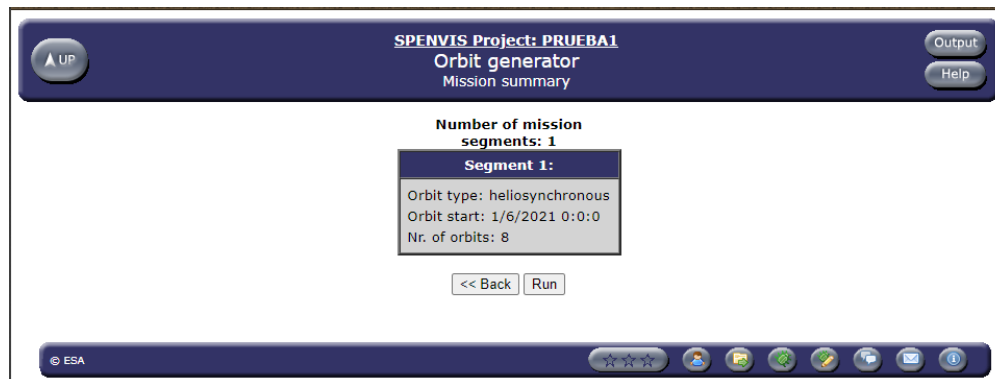


Figura 72. Resumen de la misión en SPENVIS

Al correr la simulación de la órbita aparece la ventana mostrada en la *Figura 73*. En esta parte, SPENVIS permite obtener distintas proyecciones. Por ejemplo, para este trabajo se necesita conocer la proyección de la órbita sobre un mapa terrestre.

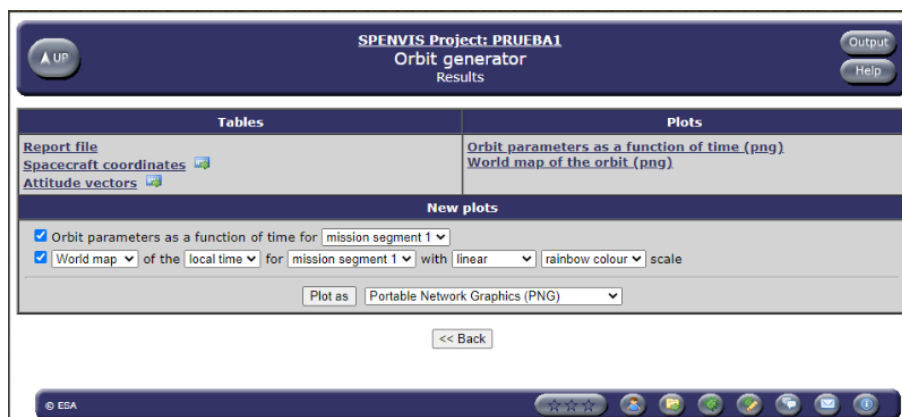
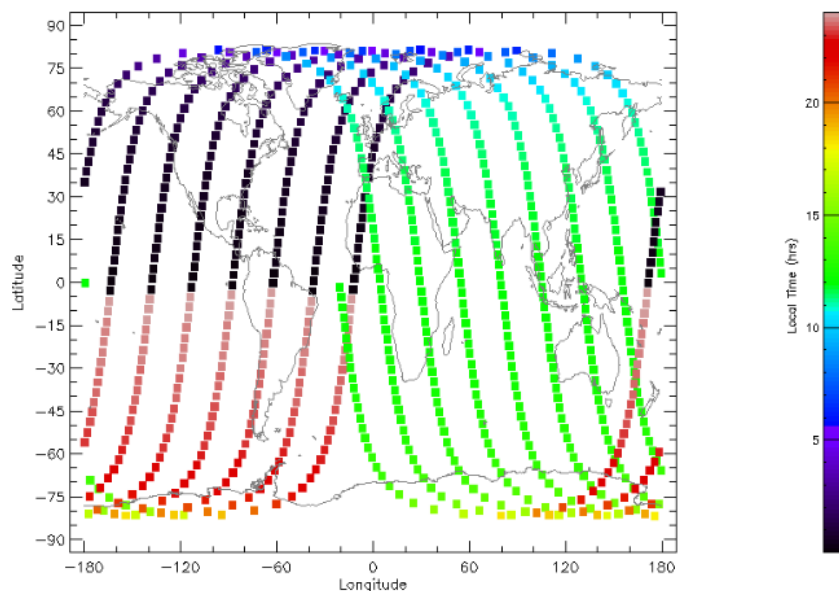


Figura 73. Generador de órbita en SPENVIS

Graficando el **mapa terrestre de la órbita**, representado en la *Figura 74*, se observa que a las 12 horas el satélite pasa por el nodo ascendente. Además, al definirse una órbita solar sincrónica se obtiene una órbita con inclinación ligeramente mayor a los 90 grados, por lo que se trata de una órbita retrógrada<sup>58</sup>. La *Figura 74* representa el escenario al que se expondrá el sistema a lo largo del día en función de la latitud y longitud.



*Figura 74. Mapa terrestre de la órbita en SPENVIS*

Una vez definida la órbita, se selecciona “up” para regresar a la pantalla de inicio de SPENVIS guardando la representación de la órbita deseada.

En esta etapa del proyecto, lo más interesante es determinar los efectos y fuentes de radiación a los que se enfrentará el sistema una vez puesto en órbita, por lo que, regresando a la pantalla de inicio de SPENVIS, se selecciona el menú de **fuentes y efectos de radiación**.

En la *Figura 75* se encuentran diferentes modelos para representar el medio ambiente de radiación espacial, que se muestran en cuatro grupos: fuentes de radiación, daño por radiación en celdas solares, las dosis de radiación a largo plazo y los SEE.

Dentro de las **fuentes de radiación** se pueden correr distintos modelos, con los cuales se realizarán pronósticos para generar estimaciones de la radiación proveniente de las distintas fuentes que experimentará el dispositivo en la órbita definida anteriormente. Se pueden agrupar en tres tipos: la radiación atrapada de los cinturones de radiación, correspondiendo al **flujo de protones y electrones atrapados** y la anisotropía del **flujo de protones atrapados**; la actividad solar, modelada con el **flujo máximo de partículas solares** y la **fluencia de misión de partículas solares**; y el **flujo de GCR**.

La segunda parte, daño por **radiación en celdas solares**, consiste básicamente los efectos de la actividad solar en los dispositivos.

Lo siguiente, siendo las **dosis de radiación a largo plazo**, corresponde a la **dosis ionizante para geometrías simples**, modelando la TID; y la **pérdida de energía no ionizante para geometrías simples**, o también conocida como los efectos de desplazamiento.

La cuarta y última parte corresponde a los **SEE**, determinado por la **tasa de SEU a corto plazo y espectros LET** y la **tasa de SEU a largo plazo y espectros**.

<sup>58</sup> Órbita en donde el objeto se traslada en dirección contraria al movimiento de rotación del cuerpo que orbita.

En principio, se pueden correr todos los modelos independientemente o de manera conjunta. De manera que, si se desea saber cuál es la aproximación más cercana a la realidad se deberán correr todos los modelos. Conforme se simulen los modelos ya disponibles se desbloquearán otros para observar cual es el efecto posible a partir del medio ambiente simulado.

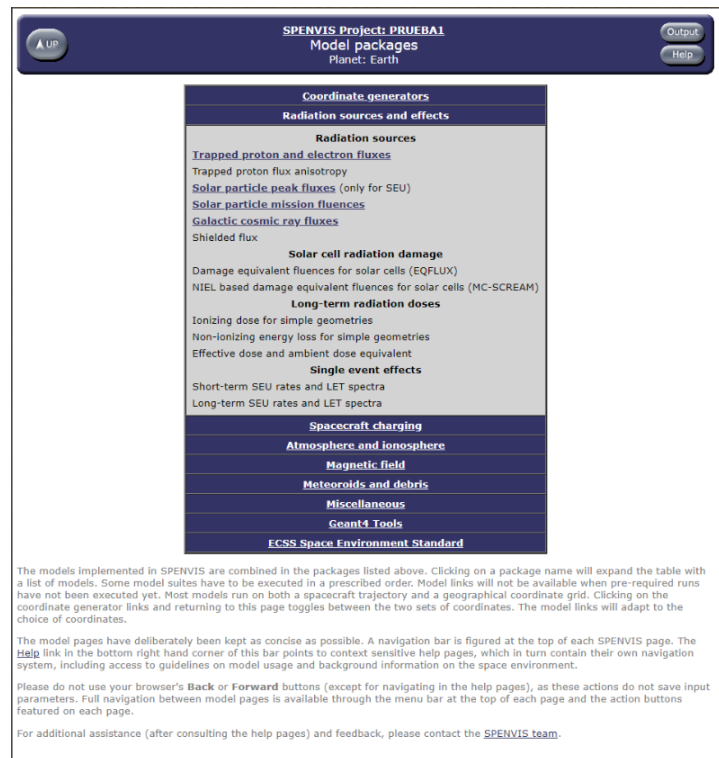


Figura 75. Fuentes y efectos de radiación en SPENVIS

Comenzando con las fuentes de radiación, se procede a simular el modelo de flujos de protones y electrones atrapados. A continuación, se definen los parámetros del modelo de radiación atrapada, teniendo en mente que SPENVIS sirve de integrador de diferentes modelos de simulación, cada uno de ellos presentado por un laboratorio o instituto, por lo que si se desea investigar alguno de ellos y realizar una simulación con mayor detalle se debe verificar en la documentación de SPENVIS qué institución lo realiza y que esta, probablemente, ofrezca su propia plataforma. Por ahora, se dejarán los valores predeterminados, mostrados en la Figura 76.

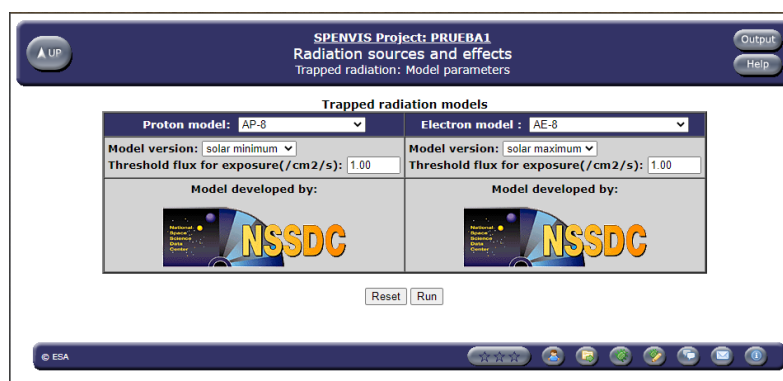


Figura 76. Parámetros del modelo de radiación atrapada en SPENVIS

Una vez definidos los parámetros del modelo de radiación atrapada de los cinturones de radiación, se pasa a obtener los resultados del modelo (Figura 77), en donde se observa que los **espectros promedio de protones y electrones atrapados** se muestran independientemente en un mapa terrestre en escala logarítmica, representando la órbita de la Figura 74 con el nivel de radiación que se experimentaría un dispositivo en cada instante de tiempo.



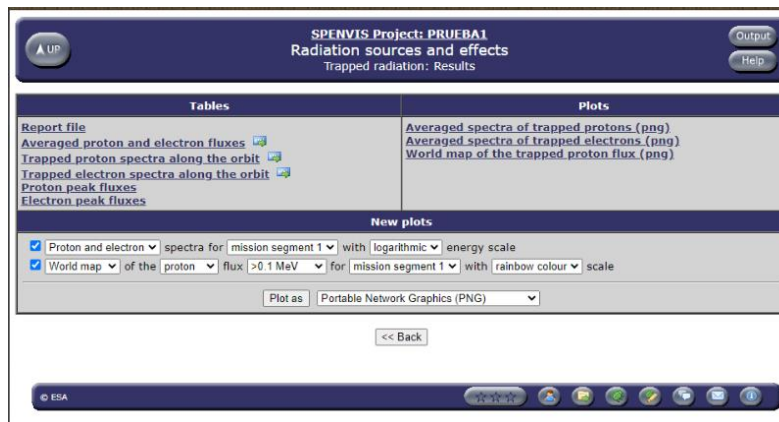


Figura 77. Resultados del modelo de radiación atrapada en SPENVIS

En la Figura 78 se muestra el mapa terrestre del flujo de protones atrapados que, al tratarse de una órbita muy baja, presenta un nivel bajo de radiación. En general, la mayor parte de la radiación se localiza en pequeños periodos de tiempo cuando se acerca a los polos. Además, se presenta la SAA, por la cual se observa un aumento de radiación en la zona sur del océano Atlántico (zona verde y roja). Tomando en cuenta lo anterior, la cantidad de radiación es muy pequeña, considerando que el dispositivo únicamente se expondrá a esta situación cuando pase por la SAA. Sin embargo, se concluye que sí existe un nivel de radiación presente por un flujo de protones en la órbita deseada y hay que estimarlo.

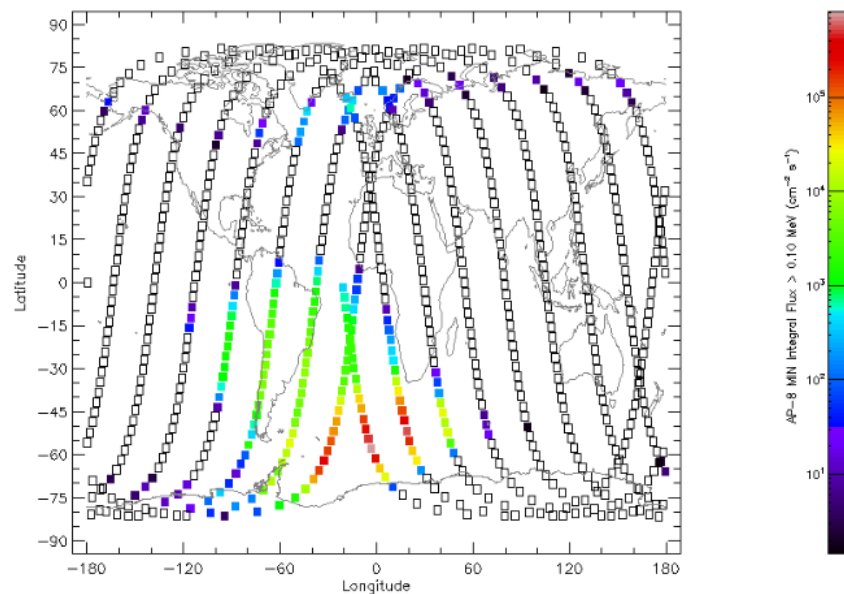


Figura 78. Mapa terrestre del flujo de protones atrapados en SPENVIS

Por su parte, el flujo de electrones observado en la Figura 79 presenta un comportamiento similar al flujo de protones, pero con mayor intensidad.

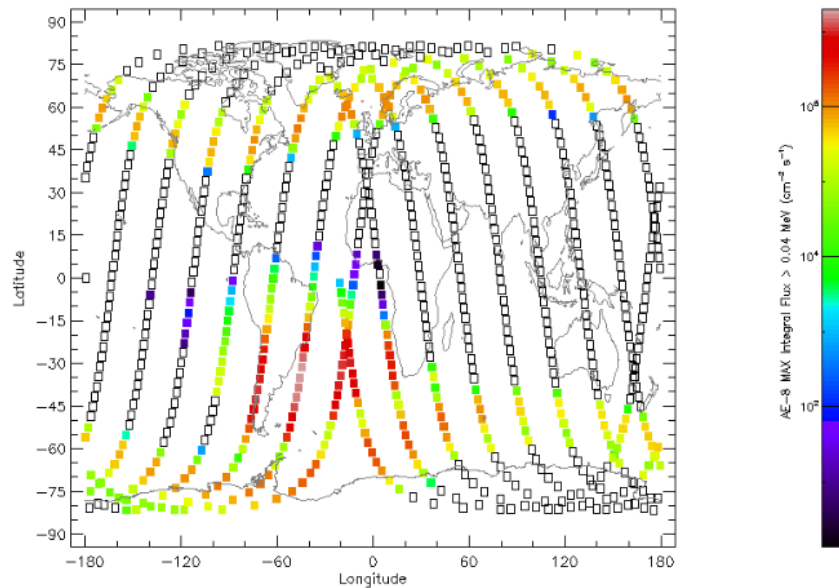


Figura 79. Espectro promedio de electrones atrapados en SPENVIS

Ahora, el primer efecto de interés para este trabajo es la TID, recordando que es un efecto que causa daños a largo plazo gracias a la acumulación de radiación por el impacto de partículas de baja energía que no son lo suficientemente energéticas para producir un efecto instantáneo. Para este punto, en la *Figura 75* se habilitan la mayor parte de los modelos, seleccionando la **dosis ionizante para geometrías simples**, llevándonos a la ventana de parámetros mostrada en la *Figura 80*.

El modelo de TID en SPENVIS sirve para conocer el nivel de acumulación de radiación en un material, en este caso, silicio. Es importante recordar que la cantidad de radiación que recibe un material depende de si este está protegido por un blindaje, normalmente hecho de aluminio. Se selecciona la configuración del blindaje más simple, en donde el material semiconductor se posiciona en el centro de una esfera de aluminio.



Figura 80. Parámetros del modelo para TID en SPENVIS

Una vez definidos los parámetros, se pasa a la obtención de los resultados del modelo de TID, en donde únicamente se presenta la opción de determinar la TID o curvas de profundidad de dosis de radiación en función del espesor del blindaje. Esto se aprecia en la *Figura 81*.

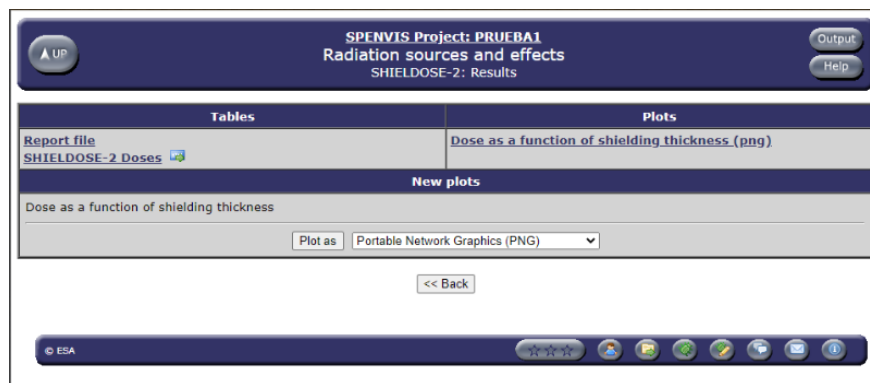


Figura 81. Resultados del modelo de TID en SPENVIS

Los resultados del modelo de TID se presentan en la gráfica de la *Figura 82*. De este resultado, el principal interés recae en el espesor del blindaje de aluminio. Recordando que se definió una duración de un año, el modelo realiza la estimación o pronóstico de la TID a la que se expondrá el sistema después de un año en órbita en función del espesor del blindaje. Pero ¿qué grosor seleccionamos y por qué? Con base en el *Estado del arte*, en donde los SCMI están diseñados en su mayoría por componentes COTS, la mayor parte de estos especifican que sus sistemas presentan un endurecimiento a la TID de 20 krad, así que, considerando un factor de seguridad de 2 para brindarle mayor protección a los dispositivos electrónicos, se selecciona el grosor del blindaje para un endurecimiento a la TID de 10 krad ( $10^4$  rad).

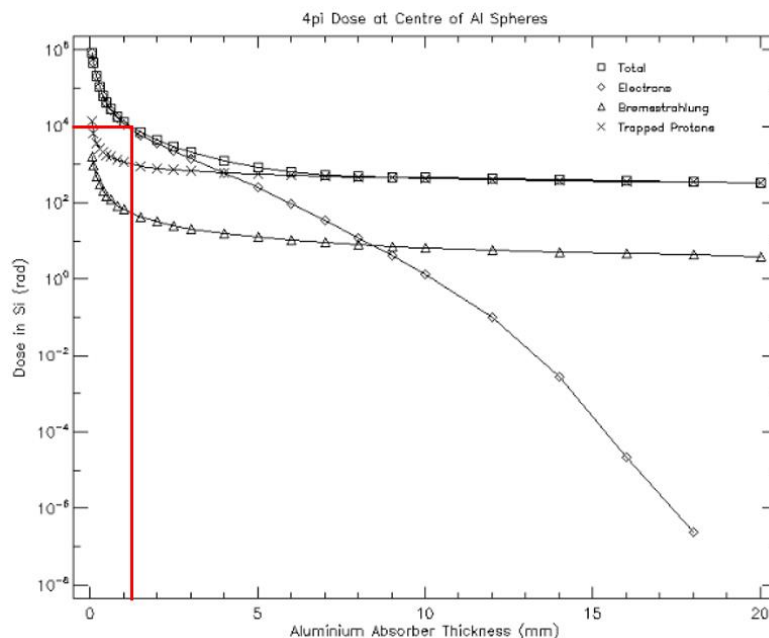


Figura 82. Dosis en función del espesor de blindaje en SPENVIS

De esta manera, la gráfica con mayor peso es la que representa la radiación total, por lo que, se observa que un espesor de aproximadamente 1.25 mm (marcado con una línea roja) es suficiente para que la dosis total acumulada no sobrepase los 10 krad en un año. Este dato es importante porque, a pesar de ser una labor mecánica, el diseñar cualquier dispositivo electrónico involucra determinar las condiciones con las que el sistema funcionará correctamente. Por lo que, como requisito del sistema, se necesita de un blindaje de aluminio con un espesor mínimo de 1.5 mm para que funcione correctamente durante un año en una órbita solar síncrona de 800 km de altura.

Regresando a la *Figura 75*, se procede a estimar los SEE. Para un diseño real, se realiza la estimación de probabilidad de ocurrencia de upset, ya que es el SEE más común seguido del latch-up, que debido a que presenta una probabilidad de ocurrencia menor no se ofrece en SPENVIS un modelo para estimarlo. En caso del upset,

conocer su probabilidad de ocurrencia sirve para determinar qué tipos de EDAC utilizar, qué técnicas de actualización de datos implementar o qué tipo de redundancia específica utilizar.

Para estimar los SEU existen dos casos: la estimación a corto y a largo plazo que, a diferencia del modelo de TID, utilizan el espectro LET. Empezando por el modelo a corto plazo se obtiene lo siguiente.

Como primer paso en la determinación de la **tasa de SEU a corto plazo y espectros LET**, se especifica el espesor del blindaje de aluminio, recordando que, para los SEE, el blindaje no es una solución, pero sí sirve como medida de mitigación, es decir, ayuda a disminuir los niveles de energía de las partículas incidentes, pero la ocurrencia de alguna partícula altamente cargada llevará a que la partícula atraviese el blindaje causando un efecto inmediato en el semiconductor.

En los modelos de SEE se realiza una estimación probabilística, en donde, al poseer grandes cantidades de dispositivos (transistores) dentro de los circuitos integrados del sistema, estos representan un área sensible a partículas de altos niveles energéticos. Por ahora, los parámetros del dispositivo de la *Figura 83* se mantendrán con sus valores predeterminados para obtener un primer acercamiento. Una vez que se seleccionen los dispositivos y se avance en el diseño se podrá hacer una estimación más real de la probabilidad de ocurrencia de SEU.

SPENVIS Project: PRUEBA1  
Radiation sources and effects  
Short-term SEU rates: Input parameters for device

Shielding thickness (Al equivalent): 0.2 cm

Device material: Si (SRIM2008)

Device source: user defined

Device name: SCMI

Shape Sensitive Volume: rectangular parallelepiped

Dimensions: 38.7 x 38.7 x 2.0 [µm]  
1450 x 2.0 [µm]

Direct ionisation upset rates  
Cross-section method: critical charge  
Qc [pC]: 1.13e-2  
Lth [MeV·cm<sup>2</sup>/mg]: 54.85e-2  
Algorithm: constant LET (CREME)

Proton induced upset rates  
Cross-section method: Bendel function  
A [MeV]: 4.88  
B [MeV]: 7.09

Reset Run

Model developed by

ESA BIRA-IASB ESA ESTEC

Figura 83. Parámetros de entrada del dispositivo para la estimación de SEU en SPENVIS

En la *Figura 84* se observan los resultados que se pueden obtener de la tasa de SEU a corto plazo, en donde se define el material semiconductor y el espectro que se desea obtener, que por lo general se trata del espectro LET.

SPENVIS Project: PRUEBA1  
Radiation sources and effects  
Short-term SEU Rates: Results

Tables

Report file  
Average LET (Si, SRIM), proton and ion fluxes  
LET (Si, SRIM) averaged over all particle species

Plots

Spacecraft shielded LET spectra (png)

New plots

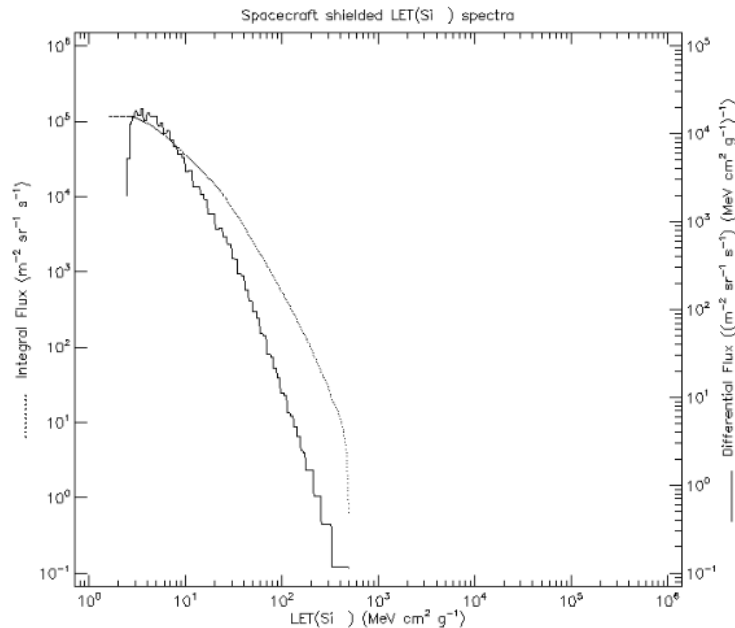
Spacecraft shielded Si (SRIM) LET spectra for mission segment 1

Plot as Portable Network Graphics (PNG)

<< Back

Figura 84. Resultados de la tasa de SEU a corto plazo en SPENVIS

A continuación, en la *Figura 85* se presenta la gráfica del espectro LET, en donde se tiene un área de cuantas partículas hay por unidad de tiempo y de cuál es su valor LET. Se interpreta que después de un año en la órbita seleccionada se presentan muchas partículas con un valor LET pequeño que impactan al sistema cada vez que pase por los cinturones de radiación y que las partículas de alto LET se presentan en pequeñas cantidades. En este caso, el principal interés radica en las partículas de alto nivel energético, de manera que se tiene una cantidad de partículas posibles, cuyo valor toma importancia al seleccionar los componentes buscando que estén caracterizados en su espectro LET para determinar la probabilidad de ocurrencia de un SEE. Por ejemplo, si un componente presenta un LET de 20, significa que, según la *Figura 85*, aproximadamente  $10^4$  partículas pueden ser dañinas para el dispositivo. A partir de esto, se puede determinar una probabilidad de ocurrencia de un evento, que servirá como fundamento para establecer el tipo de tolerancia a fallas necesario para evitar errores, por ejemplo, la frecuencia de restablecimiento de una memoria.



*Figura 85. Espectro LET de naves espaciales blindadas en SPENVIS*

La determinación de la probabilidad de ocurrencia de un evento se realizará más adelante, una vez se tenga más avanzado el diseño del sistema.

## 9.2 Definición de bloques funcionales

Como parte de la *Metodología de diseño*, este paso consiste en la división conceptual del sistema en bloques funcionales con el objetivo de agrupar los requisitos en sistemas más pequeños con funciones específicas. De manera similar, parte de la arquitectura de diseño consiste en hacer la división del sistema en conjuntos de elementos físicos. La división del sistema en módulos más pequeños que se presentó en el *Estado del arte*, se realizó con el objetivo de agrupar las características identificadas en los SCMI comerciales para una mejor comprensión, además de que sirvieron como base para estructurar el *Marco teórico*.

Para el diseño, el SCMI se divide en los tres elementos básicos de la arquitectura de una computadora: la CPU (cuyo bloque funcional se le da el nombre de procesador de a bordo), los dispositivos de almacenamiento y los puertos de entrada/salida. Además, se añaden los bloques funcionales de los dispositivos externos y el factor de forma, explicados a continuación.

Recordando brevemente, la CPU es el dispositivo principal de la computadora que, entre sus principales funciones, se encarga de ejecutar las instrucciones definidas por software, procesar la información recopilada a través de los elementos de hardware y realizar la toma de decisiones. De este modo, al servir como elemento central, interconecta todos los elementos del SCMI, gestionando y distribuyendo tareas específicas a cada uno de

ellos para que, en conjunto, cumplan con procesos más complejos. Con lo anterior, es sencillo deducir que el bloque funcional del **procesador de a bordo** se refiere a la definición del dispositivo o los dispositivos encargados de desempeñar las tareas de la CPU.

Por otra parte, los **dispositivos de almacenamiento** se encargan de retener información. En toda computadora se emplean diferentes memorias, encargadas de almacenar distintos tipos de información. Para el SCMI, se definen cinco tipos de memorias: la memoria volátil, encargada de almacenar temporalmente las instrucciones, datos y el sistema operativo con los que trabaja la computadora; la memoria de almacenamiento masivo de datos, siendo la de mayor densidad para almacenar la mayor parte de la información durante la misión; la memoria de almacenamiento crítico de datos, de menor capacidad pero mayor confiabilidad para almacenar los datos indispensables para el cumplimiento de la misión; la memoria de almacenamiento de código, que es el software con el que opera el SCMI; y la memoria de almacenamiento de código de mayordomo (housekeeping), que guarda el software dedicado a dar mantenimiento al sistema.

El bloque funcional de los **puertos de entrada/salida** agrupa todos los periféricos a través de los cuales el SCMI se comunica con otros sistemas. En otras palabras, establecen un canal de comunicación bidireccional permitiendo el flujo de datos entre el SCMI y el resto de los subsistemas de un satélite, con el objetivo de controlar, gestionar los recursos del satélite, obtener datos de la carga útil y realizar diagnósticos de todos los sistemas.

El bloque funcional de los **dispositivos externos** hace referencia a aquellos elementos ajenos al procesador de a bordo, pero que son parte del SCMI. Se divide en dos tipos de dispositivos: los que brindan servicio a la computadora con funciones específicas, y los de medición, destinados a interactuar con el mundo exterior, a través de los cuales el SCMI obtiene información que le ayude a desempeñar sus tareas.

Por último, el bloque funcional del **factor de forma** se refiere a la presentación del SCMI, con la que se podrá integrar como subsistema en sistemas más complejos. El factor de forma engloba todos los elementos físicos del SCMI, detallando las especificaciones eléctricas y mecánicas que debe cumplir la construcción del sistema.

### 9.3 Planteamiento de los requisitos de sistema

Como se menciona anteriormente, los requisitos son los parámetros que el sistema debe satisfacer. Todo requisito parte de una necesidad, de modo que en este paso se identifican las necesidades del sistema, lo que lleva a la definición de los requisitos y que estos, a su vez, marquen el camino hacia la determinación de las especificaciones.

Hasta ahora, se ha determinado el concepto de diseño, por lo que se conoce hacia dónde va dirigido el proceso de diseño. De manera similar, se definieron los bloques funcionales, con los que se divide al SCMI en sistemas más pequeños para facilitar la tarea de diseño, pero ¿Qué necesidades satisface el sistema? Como se menciona anteriormente, los requisitos son los parámetros que el sistema debe satisfacer. Todo requisito parte de una necesidad, de modo que en este paso se identifican las necesidades del sistema, lo que lleva a la definición de los requisitos y su justificación, de manera que marquen el camino hacia la determinación de las especificaciones.

En la *Tabla 11* se presentan los requisitos a nivel de sistema, concentrando los requisitos globales que el SCMI debe satisfacer, en donde es importante mencionar que para cada requisito se presenta su justificación y el nivel de importancia que presenta su cumplimiento en el diseño, en donde el nivel 5 se entiende como el máximo nivel. Además, en el diseño de detalle se determinan diferentes niveles de importancia dependiendo del bloque funcional y los elementos correspondientes.

*Tabla 11. Requisitos a nivel de sistema*

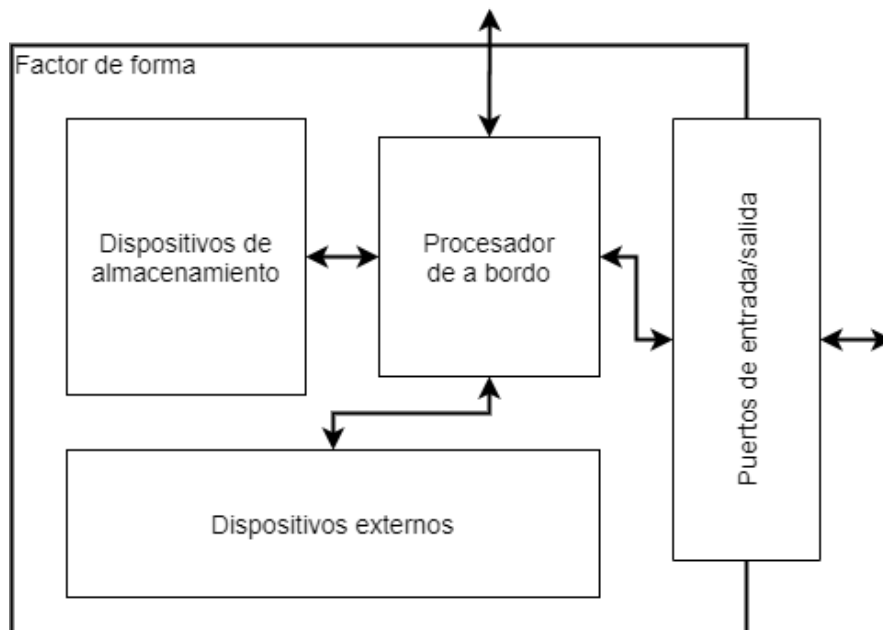
No.	Requisito	Justificación	Imp.
1	Desempeño	Tener la capacidad de realizar sus operaciones en un determinado tiempo de forma autónoma.	5

2	Recursos	Contar con los necesarios para realizar sus tareas.	4
3	Confiabilidad	Presentar un nivel adecuado para conservar los valores de sus parámetros dentro de un límite de tiempo establecido.	5
4	Resistencia al medio ambiente espacial	Adecuada para cumplir misiones espaciales en LEO.	5
5	Consumo de energía	Cumplir con los estándares de nanosatélites CubeSat.	4
6	Precio y disponibilidad	Debe construirse con componentes COTS.	4
7	Compatibilidad	Capacidad de integrarse como subsistema en nanosatélites CubeSat.	3
8	Capacidad de almacenamiento	Suficiente para desempeñar misiones de un año de duración y almacenar todos los datos de la carga útil.	4

## 9.4 Diseño conceptual

Como último paso de esta etapa se encuentra el diseño conceptual o diseño de alto nivel. Durante esta fase, se construye un modelo conceptual del sistema de hardware y software. En este modelo se aprovecha tanta abstracción como sea apropiado. El diseño conceptual es un proceso modular en donde, a través de diagramas de bloques y gráficas de flujo de datos, se construyen cada uno de los bloques funcionales, denominados como módulos a partir de ahora, describiendo su operación general, sin considerar los detalles de cómo trabajan. [38]

Comenzando con el diagrama de bloques del SCMI con el máximo nivel de abstracción, observado en la *Figura 86*, se coloca el procesador de a bordo como el módulo central. Conectados a él, se encuentran los dispositivos de almacenamiento, los dispositivos externos y los puertos de entrada/salida, todos integrados dentro del factor de forma.



*Figura 86. Diseño conceptual del SCMI, con el mayor nivel de abstracción*

A continuación, se presenta el diseño conceptual de cada uno de los módulos para que, al final, se integren en un único sistema y se conozca con mayor certeza lo que se está diseñando.

### 9.4.1 Diseño conceptual del procesador de a bordo

Para el procesador de a bordo, se selecciona como dispositivo de procesamiento central un microcontrolador COTS, acompañado de un segundo microcontrolador de menor rendimiento como supervisor (ver la *Figura 87*). El microcontrolador principal se encarga de todo el procesamiento de datos, ejecución de instrucciones, control de los puertos de entrada/salida, comunicación con los dispositivos externos y, en general, todas las tareas que desempeña el SCMI. Mientras que el supervisor se encarga exclusivamente de monitorear el comportamiento del microcontrolador principal, detectando posibles fallas en su funcionamiento y, si es necesario, realizar el proceso de recuperación correspondiente.

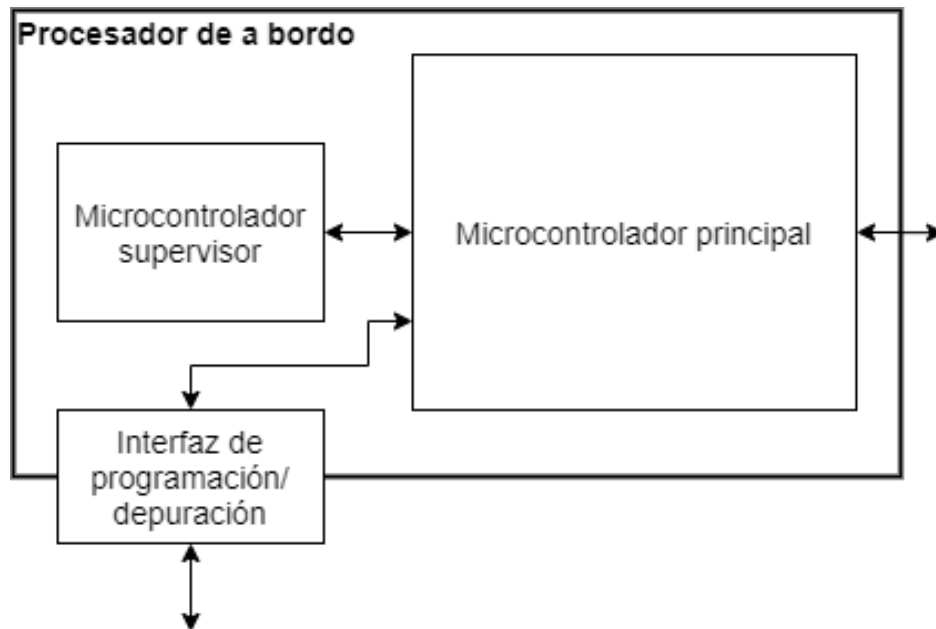


Figura 87. Diseño conceptual del procesador de a bordo

Con lo descrito anteriormente, las especificaciones del microcontrolador principal vienen dadas en su mayoría por los requisitos de rendimiento, los recursos que posee y su precio y disponibilidad en el mercado. En cambio, al desempeñar únicamente las tareas de diagnóstico y recuperación, el supervisor debe cumplir principalmente con los requisitos de confiabilidad y tolerancia a la radiación, garantizando que si el microcontrolador principal falla, el supervisor seguirá funcionando correctamente y será capaz de regresar al principal a un estado funcional. Además, es importante mencionar que, de ambos dispositivos, la mayor parte de la energía la consume el microcontrolador principal, ya que, a diferencia del supervisor, es el dispositivo que se encuentra operando en un modo de alto rendimiento y empleando un número mayor de recursos.

### 9.4.2 Diseño conceptual de los dispositivos de almacenamiento

El módulo de los dispositivos de almacenamiento corresponde a la definición de las memorias utilizadas para almacenar la información de la computadora. Para esto, se identifican cinco tipos de almacenamiento: información volátil o de trabajo, almacenamiento masivo de datos, almacenamiento crítico de datos, almacenamiento de código y almacenamiento de código de mayordomo, para los cuales se utilizarán distintos tipos de memorias dependiendo de los requisitos para el almacenamiento de cada tipo de información.



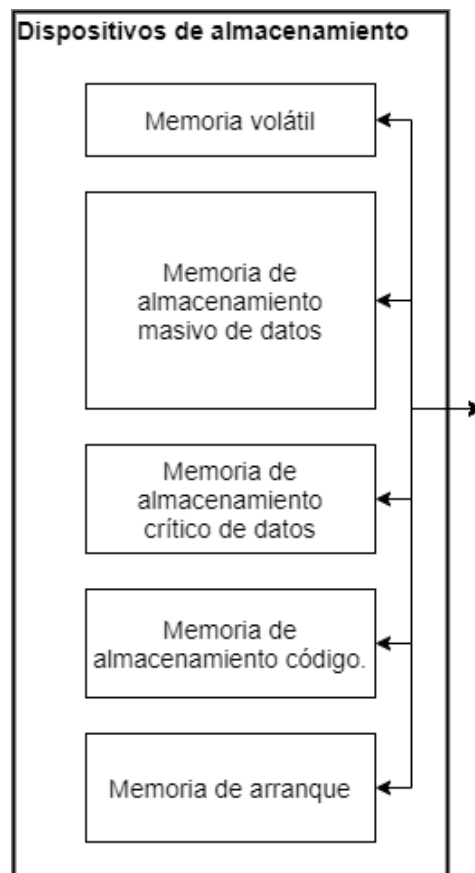


Figura 88. Diseño conceptual de los dispositivos de almacenamiento

Como se observa en la *Figura 88*, se plantean cuatro dispositivos de almacenamiento, en donde se seleccionan los tipos de memorias en función de sus requisitos, que a su vez dependen de su propósito o del tipo de información que almacenen. A continuación, se presenta la aplicación de cada tipo de memoria y con esto, sus requisitos.

#### ***Requisitos de la memoria volátil***

Es la memoria de trabajo de la computadora, encargada de almacenar temporalmente las instrucciones que ejecuta la CPU, el software, los datos que se están operando en distintos procesos, el sistema operativo y toda la información temporal necesaria para que la computadora desempeñe sus tareas correctamente. Para esto, se necesita seleccionar una memoria volátil, confiable, altamente tolerante a la radiación, con larga vida útil y, principalmente, una alta velocidad, ya que el buen desempeño de la computadora depende en gran medida de la velocidad de su memoria de trabajo.

#### ***Requisitos de la memoria de almacenamiento masivo de datos***

Es el dispositivo en donde se almacena casi en su totalidad la información obtenida durante la misión. Sabiendo esto, se debe tener una memoria no volátil, confiable, resistente a la radiación, rápida, de bajo consumo de energía y con la capacidad de almacenamiento necesaria para retener toda la información del SCMI.

#### ***Requisitos de la memoria de almacenamiento crítico de datos***

El almacenamiento crítico de datos se refiere a la retención de la información de mayor importancia o considerada como esencial para el cumplimiento de la misión. Debido a esto, se trata de un dispositivo que debe asegurar el almacenamiento correcto y seguro de su información, por lo que, contraria a la memoria de almacenamiento masivo de datos, su selección se hace tomando como prioridad su confiabilidad y alta tolerancia a la radiación frente a su capacidad de almacenamiento.

### Requisitos de la memoria de almacenamiento de código

Como su nombre lo indica, la memoria de almacenamiento de código se encarga de guardar el conjunto de instrucciones que ejecuta el SCMI. Con esto, sus requisitos son similares al almacenamiento masivo de datos, pero con la tolerancia de utilizar una memoria con poca capacidad de almacenamiento. Además, se utiliza la misma memoria para almacenar el código de mayordomo, que como se dijo anteriormente, es el software encargado de limpiar la computadora, liberando sus recursos y eliminando sus archivos temporales obsoletos dentro del sistema, con el fin de optimizar la operación del SCMI. En pocos sistemas del *Estado del arte* se observa una memoria destinada específicamente a este propósito, por lo que, para economizar, se utilizará la memoria de almacenamiento de código para este mismo fin.

### Requisitos de la memoria de arranque

Debido a que se utilizan microcontroladores en el SCMI, los cuales ya cuentan con los elementos necesarios para desempeñar sus funciones, la memoria de arranque de ambos microcontroladores se encuentra embebida en cada uno de ellos, correspondiendo a una zona de la ROM correspondiente, por lo que no es necesario implementar una memoria adicional para este fin.

### 9.4.3 Diseño conceptual de los puertos de entrada/salida

El diseño de los puertos de entrada/salida se realiza principalmente seleccionando aquellos que presentan mayor utilidad en los nanosatélites CubeSat. Para esto, a través del *Estado del arte*, se identifican aquellos puertos presentes en la mayoría de los sistemas, garantizando que son altamente utilizados y su funcionamiento es conocido por la comunidad científica que desarrolla sistemas del mismo tipo.

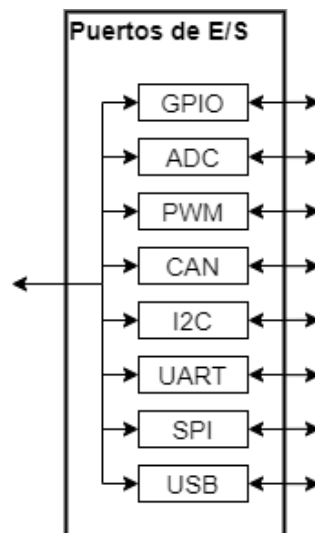


Figura 89. Diseño conceptual de los puertos de entrada/salida

Como se puede observar en la *Figura 89*, se seleccionaron los puertos GPIO, ADC, PWM, CAN, I<sup>2</sup>C, UART, SPI y USB para implementarlos en el SCMI. La selección de los puertos de entrada/salida afecta directamente a la selección del dispositivo de procesador de a bordo, ya que se debe buscar un dispositivo compatible o que posea todos estos puertos de comunicación.

### 9.4.4 Diseño conceptual de los dispositivos externos

Para el caso de los dispositivos externos, anteriormente se mencionó que están divididos en dos tipos: de servicio y de medición. Para el caso de los dispositivos de medición, se seleccionan aquellos que cumplan con funciones específicas que sirvan de apoyo a la computadora para operar correctamente. En cambio, los dispositivos de medición son aquellos que obtendrán información del mundo real para la computadora. Al tratarse de un SCMI de propósito general, estos últimos deben seleccionarse los dispositivos cuya función es comúnmente necesitada en los sistemas espaciales.

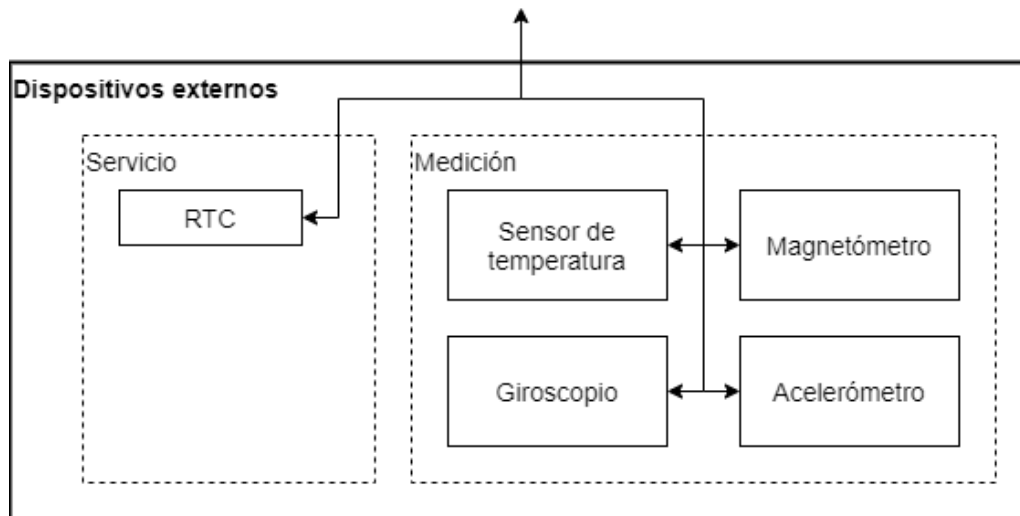


Figura 90. Diseño conceptual de los dispositivos externos

En la *Figura 90*, se puede observar que se definió dispositivo de servicio al RTC. Este realiza la función específica de brindar el registro del tiempo real del SCMI con alta precisión, por lo que debe seleccionarse un dispositivo confiable, de bajo consumo de energía y principalmente con alta tolerancia a la radiación.

En caso contrario, los dispositivos de medición se seleccionaron con ayuda del *Estado del arte*, observando que los sensores de temperatura y las Unidades de Medición Inercial (IMU) son altamente utilizadas en el diseño de SCMI de propósito general. De esta manera, el SCMI puede realizar tareas básicas del ADCS, apoyando en la determinación de la orientación y permitiéndole al diseñador la opción de no integrar un ADCS completo, dependiendo de la complejidad de su misión.

#### 9.4.5 Diseño conceptual del factor de forma

Para el factor de forma, es sencillo determinarlo tomando como base la investigación del *Estado del arte*, en donde claramente destaca la utilización del PC/104. Para este proyecto, igualmente se utilizará el factor de forma PC/104 para diseñar el SCMI, con la finalidad de obtener un sistema compatible con el estándar CubeSat.

Con todo lo anterior, se obtiene el diagrama de bloques general del SCMI, integrando cada uno de los módulos en un único sistema, observado en la *Figura 91*.

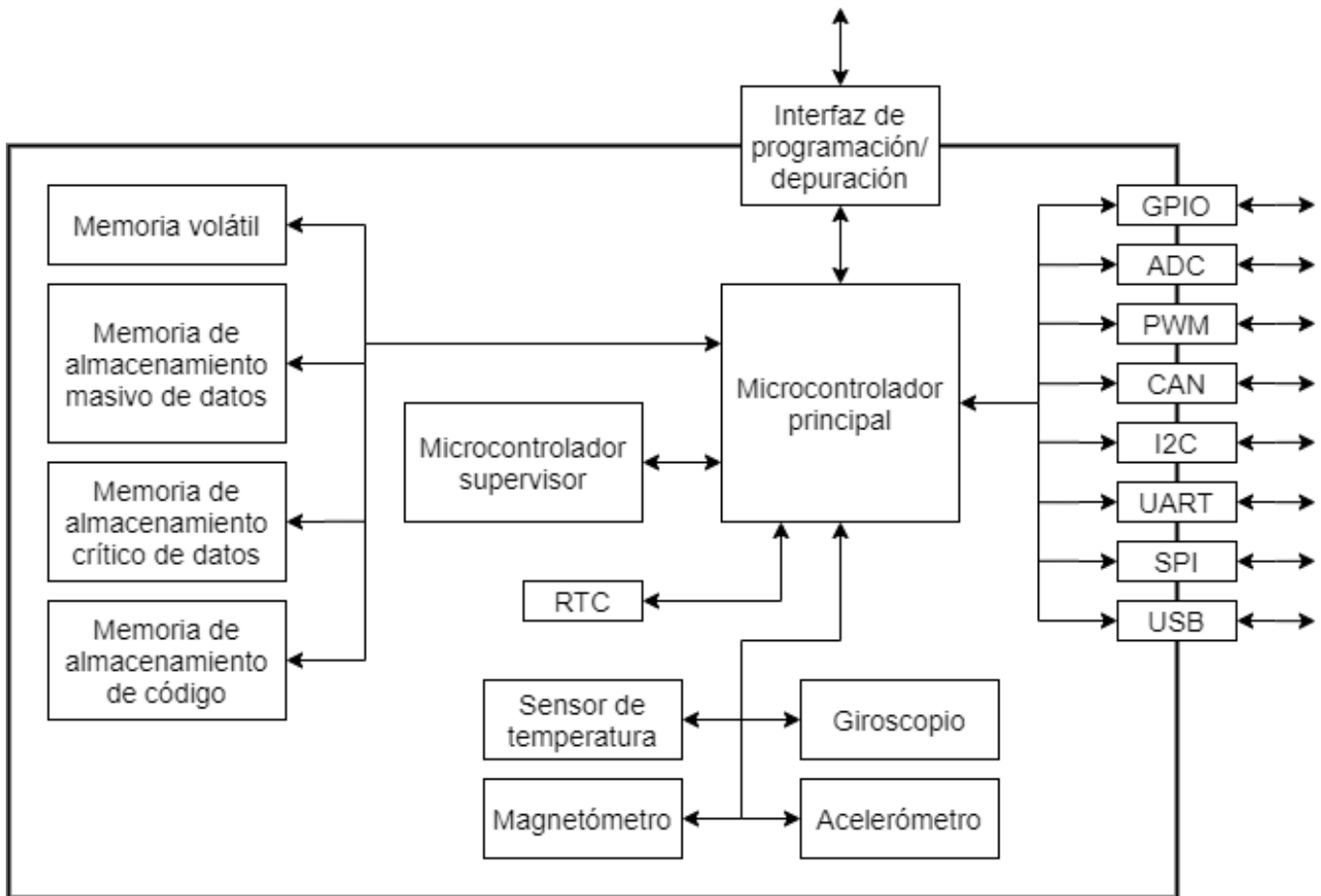


Figura 91. Diseño conceptual del SCMI

Al principio del proceso de diseño únicamente se contaba con una ligera idea de hacia dónde se quería llegar, observando al SCMI como una “caja negra”. Siguiendo la *Metodología de diseño*, se realizó la investigación preliminar del *Estado del arte* y del *Marco teórico*, obteniendo así los conocimientos necesarios para determinar qué es lo que se pretende desarrollar. Posteriormente, se introdujo el diseño conceptual, dando comienzo a la toma de decisiones justificadas en función de los requisitos del sistema, aterrizando con mayor claridad los conocimientos adquiridos anteriormente para visualizar progresivamente el contenido de la caja negra. Este proceso de desarrollo se conoce como esquema top-down, en donde, conforme se avanza en el proceso de diseño, el sistema que, en un principio, carecía de claridad, se redefine en diferentes partes cada vez más detalladas hasta alcanzar un sistema construido por módulos con especificaciones suficientemente certeras y características altamente detalladas para validar el diseño.

En el diseño de detalle, o diseño de ingeniería, se comienza definiendo el diseño preliminar. El proceso consiste en desarrollar con mayor profundidad cada una de las partes del sistema hasta determinar las especificaciones finales, de tal manera que se alcance un diseño definitivo. En esta parte del desarrollo se plantean las especificaciones de diseño a nivel de módulo y sistema, se realiza el diseño de detalle a nivel de módulo para posteriormente hacer la integración de estos en un único sistema. Después de obtener un sistema completo, se realiza la estimación de las necesidades cuantitativas, determinando los componentes definitivos del sistema para poder estimar los costos requeridos para su construcción y, finalmente, realizar el análisis del sistema y la evaluación de su desempeño en función del cumplimiento de los requisitos y las especificaciones. En esta etapa se consideran principalmente tres problemas: la selección de componentes, el costo de producción y el desempeño robusto del sistema [63].

## 10.1 Planteamiento de las especificaciones objetivo de diseño

Como primer paso, se determinan las especificaciones de diseño a nivel de sistema y de módulo. Es importante mencionar que el planteamiento de las especificaciones objetivo de diseño y el diseño de detalle en sí se realizan simultáneamente. En otras palabras, la toma de decisiones depende del uno y del otro. En este caso, por facilidad, se documentan primero las especificaciones respondiendo a la pregunta ¿Qué hace el sistema? para, posteriormente, presentar el diseño resolviendo ¿Cómo lo hace el sistema?.

### 10.1.1 Especificaciones objetivo del SCMI

Uno de los objetivos generales al comienzo del trabajo fue diseñar un SCMI lo suficientemente robusto para competir funcionalmente con los sistemas comerciales identificados en el *Estado del arte*. Partiendo de esto, el diseño de detalle empieza con un panorama general del sistema, determinando las especificaciones objetivo generales con base en el *Estado del arte*, presentadas en la *Tabla 12*.

Tabla 12. Especificaciones objetivo generales del SCMI

No.	Métrica	No. de requisito	Imp.	Parámetro	Min.	Tip.	Max.	Unid.
1	Tensión de operación	5	2	$V_{DD}$	3	3.3	5	V
2	Consumo de energía	5	4	$P_D$	50	-	1500	mW
3	Frecuencia de operación	1	3	$f_{CLK}$	4	-	400	MHz
4	Temperatura de operación	4	5	$T_A$	-40	-	85	°C
5	Temperatura de almacenamiento	4	4	$T_{STG}$	-55	-	105	°C

6	Resistencia a la radiación acumulada	3, 4	5	-	10	-	20	Krad
7	Masa	7	4	-	24	-	106	g

En esta parte, se definen de nuevo los niveles de importancia de cada métrica y se especifican los requisitos de la *Tabla 11* a los que hacen alusión.

Recordando que el diseño a detalle se realiza a nivel de módulos para posteriormente integrarse en el último paso del proceso, se procede a adentrarse detalladamente en cada módulo determinando los elementos que lo conforman e identificando sus especificaciones correspondientes. Es importante mencionar que este paso se realiza en paralelo con la selección de componentes (documentada en el *Diseño de detalle*).

### 10.1.2 Especificaciones objetivo del procesador de a bordo

El procesador de a bordo contempla dos elementos fundamentales: el microcontrolador principal y el microcontrolador supervisor, cuyas especificaciones objetivo se presentan en la *Tabla 13*.

*Tabla 13. Especificaciones objetivo del procesador de a bordo*

No.	Métrica	No. de requisito	Imp.	Parámetro	Min.	Tip.	Max.	Unid.
<b>Microcontrolador principal</b>								
1	Tensión de operación	5	3	$V_{DD}$	3	3.3	5	V
2	Consumo de energía	5	5	$P_D$	50	-	1500	mW
3	Frecuencia de operación	1	4	$f_{CLK}$	4	-	400	MHz
4	Temperatura de operación	4	5	$T_A$	-40	-	85	°C
5	Temperatura de almacenamiento	4	4	$T_{STG}$	-55	-	105	°C
6	Resistencia a la radiación acumulada	3, 4	2	-	10	-	20	krad
<b>Microcontrolador supervisor</b>								
7	Tensión de operación	5	5	$V_{DD}$	3	3.3	5	V
8	Consumo de energía	5	5	$P_D$	50	-	1000	mW
9	Frecuencia de operación	1	3	$f_{CLK}$	4	-	400	MHz
10	Temperatura de operación	4	5	$T_A$	-40	-	85	°C
11	Temperatura de almacenamiento	4	5	$T_{STG}$	-55	-	105	°C
12	Resistencia a la radiación acumulada	3, 4	5	-	20	-	-	krad

Además de los microcontroladores empleados en el procesador de a bordo, es necesario considerar las características del RTOS, cuya implementación no abarca el presente trabajo, pero debe considerarse en la elección del microcontrolador principal para que a futuro no se requiera un cambio radical en el diseño. Dicho esto, los requisitos definidos para el RTOS son: buen desempeño (requisito 1, imp. 5), un bajo precio y alta disponibilidad (requisito 6, imp. 5), que cuente con los recursos necesarios (requisito 2, imp. 5) y que necesite poca memoria (requisito 8, imp 3).

### 10.1.3 Especificaciones objetivo de los dispositivos de almacenamiento

El SCMI tiene cuatro tipos de almacenamiento, por lo que se seleccionan las tecnologías de memoria dependiendo de sus funciones y sus requisitos, presentados en la *Tabla 14*.

*Tabla 14. Especificaciones objetivo de los dispositivos de memoria*

No.	Métrica	No. de requisito	Imp.	Parámetro	Min.	Tip.	Max.	Unid.
Memoria de trabajo								
1	Tensión de operación	5	4	$V_{DD}$	-	-	5	V
2	Capacidad de almacenamiento	8	5	-	256k	-	64M	B
3	Temperatura de operación	4	4	$T_A$	-40	-	85	°C
4	Temperatura de almacenamiento	4	4	$T_{STG}$	-55	-	105	°C
5	Resistencia a la radiación acumulada	4	4	-	10	-	20	krad
Memoria de almacenamiento masivo de datos								
6	Tensión de operación	5	3	$V_{DD}$	-	-	5	V
7	Capacidad de almacenamiento	8	5	-	16M	-	-	B
8	Temperatura de operación	4	4	$T_A$	-40	-	85	°C
9	Temperatura de almacenamiento	4	4	$T_{STG}$	-55	-	105	°C
10	Resistencia a la radiación acumulada	4	4	-	10	-	20	krad
Memoria de almacenamiento crítico de datos								
11	Tensión de operación	5	2	$V_{DD}$	-	-	5	V
12	Capacidad de almacenamiento	8	2	-	32k	-	1M	B
13	Temperatura de operación	4	5	$T_A$	-40	-	85	°C
14	Temperatura de almacenamiento	4	5	$T_{STG}$	-55	-	105	°C
15	Resistencia a la radiación acumulada	4	5	-	20	-	-	krad

Memoria de almacenamiento de código								
16	Tensión de operación	5	2	$V_{DD}$	-	-	5	V
17	Capacidad de almacenamiento	8	2	-	32k	-	4M	kB
18	Temperatura de operación	4	5	$T_A$	-40	-	85	°C
19	Temperatura de almacenamiento	4	5	$T_{STG}$	-55	-	105	°C
20	Resistencia a la radiación acumulada	4	5	-	20	-	-	krad

Algunas especificaciones no se enlistan porque no se tiene rango de valores concreto, como el tiempo de acceso de la memoria de trabajo (requisito 1, imp. 5).

#### 10.1.4 Especificaciones objetivo de los puertos de entrada/salida

Los puertos de entrada/salida del SCMI están definidos por los puertos ofrecidos por el microcontrolador principal, por lo que los datos de interés para el planteamiento de las especificaciones objetivo de los puertos de entrada/salida son la cantidad y tipos de puertos disponibles en el microcontrolador. Estos se presentan en la *Tabla 15* añadiendo algunas de sus características, considerando el encapsulado LQFP176.

*Tabla 15. Puertos de entrada/salida disponibles en el microcontrolador principal [65]*

Puerto/interfaz de E/S	Características	Cantidad máxima*
I <sup>2</sup> C	Modo multimaestro/esclavo. Modo de velocidad estándar/rápido. Modo de direccionamiento de 7/10 bits. Modo de direccionamiento dual de 7 bits (modo esclavo). Hardware de generación/verificación CRC embebido. Soporte de SMBus 2.0 <sup>59</sup> /PMBus <sup>60</sup> . Puede ser atendida por el controlador DMA.	3
USART	Modo de comunicación multiprocesador. Modo de comunicación single-wire half-duplex. Capacidad de maestro/esclavo LIN <sup>61</sup> . Gestión de hardware de señales CTS/RTS <sup>62</sup> . Modo Smart Card (compatible con el estándar ISO 7816). Modo maestro SPI. Soporte irDA <sup>63</sup> . Puede ser atendida por el controlador DMA.	4

<sup>59</sup> Del inglés “System management bus”, es un bus de comunicación simple de dos alambres derivado de I<sup>2</sup>C, utilizado principalmente para la comunicación entre la computadora y la fuente de alimentación.

<sup>60</sup> Del inglés “Power management bus”, es una variante del SMBus que permite mayor velocidad de transferencia de datos.

<sup>61</sup> Del inglés “Local interconnect network”, es un protocolo de comunicación serial que presenta una buena alternativa para el bus CAN en aplicaciones de bajos costos, pero sacrificando velocidad y ancho de banda.

<sup>62</sup> Del inglés “Request to send/Clear to send”, son señales que permiten el control de flujo de datos entre dos nodos de un bus serial.

<sup>63</sup> Del inglés “Infrared data association”, define las especificaciones para establecer comunicación infrarroja.



UART	<p>Modo de comunicación multiprocesador.  Modo de comunicación single-wire half-duplex.  Capacidad de maestro/esclavo LIN.  Soporte IrDA.  Puede ser atendida por el controlador DMA.</p>	2
SPI	<p>Modos maestro/esclavo.  Modos de comunicación simplex y full-duplex.  Preescalador de 3 bits (8 frecuencias de operación como maestro).  Frame configurable de 8 o 16 bits.  Hardware de generación/verificación CRC que soporta memoria SD y MMC.  Puede ser atendida por el controlador DMA.</p>	3
CAN	<p>Especificación CAN 2.0A/B.  Tramas estándar con identificadores de 11 bits y extendidas de 29 bits.  3 mailboxes, 2 receptores FIFO<sup>64</sup> con 3 etapas y 28 bancos de filtros escalables compartidos para cada CAN.  256 B de SRAM para cada CAN.</p>	2
DCMI <sup>65</sup>	<p>Interfaz paralela de 8/10/12/14 bits para conectar un módulo de cámara y sensores CMOS.  Soporta video progresivo monocromo de 8 bits o formato Bayer sin procesar, video progresivo YCbCr 4:2:2, video progresivo RGB 565 o datos comprimidos (como JPEG).  Soporta modo continuo o snapshot.  Capacidad para recortar la imagen automáticamente.</p>	1
GPIO	<p>Configurables como salida (push-pull, open-drain, con/sin pull-up/down).  Configurables como entrada (flotante, con/sin pull-up/down).  Configurables como función alterna.  Resistentes a alta corriente.</p>	140
ADC	<p>Convertidores de 12 bits.  Hasta 16 canales externos en cada ADC.  Conversiones de modo single-shot o scan.  Funciones lógicas de muestro simple e intercalado.  Puede ser atendido por el controlador DMA.  Función de watchdog analógico para el monitoreo de la conversión.</p>	3

\* La cantidad depende de la configuración del microcontrolador, siendo imposible utilizar todos los puentes/interfaces de entrada/salida al mismo tiempo.

### 10.1.5 Especificaciones objetivo de los dispositivos externos

Similar al punto anterior, la primera determinación de las especificaciones objetivo del módulo de los dispositivos externos consiste en la decisión de qué dispositivos se utilizarán en el SCMI. Según el *Estado del arte*, es común identificar los sensores principales del ADCS integrados en el SCMI. De esta manera, además de los

<sup>64</sup> Del inglés "First in, first out", es un búfer de datos en donde el primer dato en entrar será el primer dato en salir y procesarse.

<sup>65</sup> Interfaz de cámara digital que permite conectar eficientemente módulos de cámara paralelos.

dispositivos de servicio para la computadora (RTC), se integran sensores de temperatura y sensores IMU (acelerómetro, giroscopio y magnetómetro).

Los dispositivos externos deben cumplir con las especificaciones objetivo del SCMI para poder integrarse sin problema. Las especificaciones objetivo con mayor prioridad se presentan en la *Tabla 16*.

*Tabla 16. Especificaciones objetivo de los dispositivos externos*

No.	Métrica	No. de requisito	Imp.	Parámetro	Min.	Tip.	Max.	Unid.
1	Tensión de alimentación	5	4	$V_{DD}$	3	-	5	V
2	Frecuencia de operación	1	3	$f_{CLK}$	4	-	400	MHz
3	Temperatura de operación	4	4	$T_A$	-40	-	85	°C
4	Temperatura de almacenamiento	4	4	$T_{STG}$	-55	-	105	°C

### 10.1.6 Especificaciones objetivo del factor de forma

La especificación objetivo del factor de forma recae principalmente en la elección del estándar que se utilizará para el SCMI. Según el *Estado del arte*, a pesar de los comentarios negativos de la comunidad de desarrollo CubeSat, el estándar PC/104 se utiliza en la mayoría de los satélites CubeSat. Actualmente, se desarrollan diferentes factores de forma, algunos de ellos compatibles con módulos específicos (del mismo proveedor) y otros basados en el PC/104, variando únicamente el tamaño del conector o ciertas características.

Para el diseño del SCMI se utilizará el factor de forma PC/104, respetando principalmente sus especificaciones mecánicas, como el tamaño y forma de la PCB, pero variando el tamaño del conector y la asignación de señales a sus terminales.

## 10.2 Diseño de detalle

Alcanzado este paso del proceso de diseño, se realiza la selección de componentes de modo que se cumplan los requisitos a nivel de sistema, justificando todas las decisiones con apoyo del *Estado del arte*. Una vez seleccionados los componentes, las especificaciones objetivo pasarán a convertirse en las especificaciones finales, dependiendo de las características de cada componente seleccionado y estas a su vez, definirán en conjunto las especificaciones finales del SCMI al final del proceso de diseño.

### 10.2.1 Diseño de detalle del procesador de a bordo

#### *Selección de componentes*

##### *Microcontrolador principal*

Como se ha mencionado anteriormente, el microcontrolador principal es el encargado de ejecutar las instrucciones, procesar los datos, administrar los recursos y gestionar las tareas, entre otras cosas. En otras palabras, se trata del CPU de la computadora, por lo que se debe seleccionar un dispositivo con las características adecuadas.

Tomando como base el *Estado del arte*, se selecciona el microcontrolador STM32F407VG de STMicroelectronics, con núcleo ARM Cortex-M4, identificado en SCMI de proveedores y desarrolladores. Los procesadores ARM son altamente utilizados en satélites CubeSat, tanto por proveedores como por desarrolladores. Estos

procesadores utilizan una arquitectura RISC de 32 bits y son diseñados para cumplir con una gran variedad de aplicaciones con un bajo consumo de energía. Entre los procesadores ARM, se encuentra la familia Cortex-M, procesadores optimizados para dispositivos de bajo costo y alta eficiencia energética, implementados en aplicaciones de internet de las cosas y dispositivos industriales y de consumo diario [66]. Esta familia se basa en la arquitectura M-Profile, que proporciona una baja latencia y una operación altamente determinista, para sistemas profundamente integrados. Dentro de esta familia, se localiza el procesador Cortex-M4 de alto rendimiento, que cuenta con una FPU de precisión simple, un conjunto completo de instrucciones DSP y una unidad de protección de memoria, diseñado para el control y rendimiento para dispositivos de señal mixta, siendo el procesador de la familia Cortex-M más utilizado [66].

El microcontrolador STM32F407IGT6 cuenta con las siguientes características, especificando los requisitos de diseño que cumplen y el nivel de importancia que presentan: [65]

- ✓ Precio: USD 10.51 **(Requisito 6, imp. 2)**
- ✓ Núcleo: ARM Cortex-M4 de 32 bits. **(Requisito 2, imp. 5)**
  - FPU.
  - Unidad de protección de memoria.
  - Conjunto de instrucciones DSP.
  - Frecuencia de operación de hasta 168 MHz.
  - 210 DMIPS/1.25 DMIPS/MHz.
- ✓ Memorias. **(Requisito 2, imp. 5)**
  - Hasta 1 MB de memoria Flash.
  - Hasta 192+4 KB de SRAM, incluyendo 64 KB de RAM de datos de memoria de núcleo acoplado<sup>66</sup> (CCM, por sus siglas en inglés).
  - Controlador flexible de memoria estática que admite Flash, SRAM, Pseudo SRAM<sup>67</sup>, NOR y NAND.
- ✓ Gestión de reloj, reinicio y suministro. **(Requisito 1, imp. 4)**
  - Suministro de aplicación y E/S de 1.8 V a 3.6 V.
  - Oscilador de cristal de 4 a 26 MHz.
  - Oscilador de 32 kHz para RTC con calibración.
- ✓ Consumo de energía. **(Requisito 5, imp. 5)**
  - Modos de sleep, stop y standby.
  - Suministro  $V_{BAT}$  para RTC, registros de respaldo de 20x32 bits + SRAM de respaldo opcional de 4 KB.
- ✓ Recursos. **(Requisito 2, imp. 5)**
  - 3x12 bits, 2.4 Msps ADC: hasta 24 canales y 7.2 Msps en modo triple intercalado.
  - 2x12 bits DAC.
  - DMA de propósito general: Controlador DMA de 16 flujos con registro FIFO y soporte de ráfagas.
  - 12x temporizadores de 16 bits de propósito general, incluyendo 2x temporizadores PWM, 2x temporizadores de 32 bits de propósito general.
  - Hasta 17 temporizadores: hasta doce de 16 bits y dos de 32 bits hasta 168 MHz, cada uno con hasta cuatro PWM.
  - Modo de depuración.
    - Interfaces SWD y JTAG.
    - Macrocelda de Seguimiento Embebida (ETM) Cortex-M4.
  - Hasta 140 puertos de E/S con capacidad de interrupción.
    - Hasta 136 rápidos de hasta 84 MHz.

<sup>66</sup> Memoria interna del STM32 que ofrece mayor velocidad para ejecutar instrucciones en una fracción del tiempo que necesitan las instrucciones almacenadas en la memoria Flash

<sup>67</sup> DRAM con un circuito integrado de actualización y control de direcciones de modo que se comporte de una manera similar a la SRAM.

- Hasta 138 tolerantes a E/S de 5 V.
- Hasta 15 interfaces de comunicación serial.
  - Hasta 3x interfaces I<sup>2</sup>C (SMBus/PMBus).
  - Hasta 4x USART/2x UART (10.5 Mbit/s, interfaz ISO 7816).
  - Hasta 3x SPI (42 Mbit/s), 2 con multiplexado I<sup>2</sup>S full-duplex.
  - 2x interfaces CAN (2.0B).
- Interfaz de 8 a 14 bits para cámara de hasta 54 MB/s.
- RTC.
- ✓ Técnicas de mitigación y tolerancia a fallas.
  - Unidad de cálculo CRC.
  - 2x temporizadores watchdog.

**(Requisito 3, imp. 2)**

De esta forma, las especificaciones finales del microcontrolador principal vienen descritas en la *Tabla 17*, en donde únicamente se presentan las características principales. En [65] se observan con mayor detalle las especificaciones del dispositivo.

*Tabla 17. Especificaciones finales del microcontrolador principal a  $T_A = 25^\circ\text{C}$  [65]*

No.	Métrica	No. de requisito	Imp.	Parámetro	Min.	Tip.	Max.	Unid.
1	Tensión de operación	5	4	$V_{DD}$	1.8	-	3.6	V
2	Corriente de operación	5	3	$I_{DD}$	-	-	240	mA
3	Consumo de energía	5	4	$P_D$	-	-	526	mW
4	Frecuencia de operación	1	4	$f_{CLK}$	0	-	168	MHz
5	Temperatura de operación	4	4	$T_A$	-40	-	105	°C
6	Temperatura de almacenamiento	4	4	$T_{STG}$	-65	-	150	°C
7	Precio	6	4	-		12.87		USD
8	Encapsulado	6, 7	1	-			LQFP (176)	

### *Microcontrolador supervisor*

Como se ha observado en el *Estado del arte*, una buena estrategia para que el SCMI adquiera tolerancia a fallas es utilizar un microcontrolador adicional que desempeñe las tareas de diagnóstico y recuperación, llamado supervisor. En otras palabras, mientras el microcontrolador principal realiza todas las tareas de procesamiento del SCMI, el supervisor se encarga de asegurar que el microcontrolador principal desempeñe sus funciones correctamente, aún con la presencia de fallas provocadas por la exposición al ambiente de radiación espacial. El uso de un supervisor aumenta el nivel de fiabilidad del SCMI sin la necesidad de diseñar un sistema redundante que dificulte el cumplimiento de las limitadas especificaciones del estándar CubeSat, además de abrir la posibilidad de que el SCMI sea diseñado principalmente con componentes COTS. Esto indica que la selección del supervisor, de manera inversa al microcontrolador principal, debe hacerse dándole prioridad a la resistencia a la radiación del dispositivo frente a su nivel de robustez, además de que su consumo de energía debe ser mínimo.

El supervisor seleccionado para el diseño es el microcontrolador de señal mixta con clasificación espacial MSP430FR5969-SP de TI, cuya familia es fabricada con FRAM inherentemente tolerante a la radiación, por lo que se ha vuelto popular en CubeSat diseñados con componentes COTS y ha presentado buen desempeño bajo un ambiente de radiación espacial [17].

La documentación completa del MSP430FR5969-SP se puede observar en [67]. A continuación, se presentan sus características principales:

- ✓ Precio: 2,376 USD (paquete de 100 unidades) - 23.76 (precio unitario) **(Requisito 6, imp. 2)**
- ✓ Núcleo: RISC de 16 bits. **(Requisito 2, imp. 3)**
  - Frecuencia de operación de hasta 16 MHz.
- ✓ Memorias. **(Requisito 2, imp. 3)**
  - FRAM de ultra bajo consumo de energía.
    - 64 KB.
    - Resistente a la radiación y no magnética.
- ✓ Gestión de reloj, reinicio y suministro. **(Requisito 1, imp. 3)**
  - Suministro de tensión de 1.8 a 3.6 V.
- ✓ Consumo de energía. **(Requisito 5, imp. 5)**
  - Modos de operación optimizados para ultra bajo consumo de energía.
    - Modo activo: aproximadamente 100  $\mu\text{A}/\text{MHz}$ .
    - Standby: Típicamente 0.4  $\mu\text{A}$ .
    - RTC: Típicamente 0.25  $\mu\text{A}$ .
    - Shutdown: Típicamente 0.02  $\mu\text{A}$ .
- ✓ Recursos. **(Requisito 2, imp. 3)**
  - DMA de 3 canales.
  - RTC con calendario y función de alarmas.
  - Analógico.
    - 16x canales ADC de 12 bits
    - 16x canales de comparador analógico.
  - 40x GPIO.
  - Interfaces de comunicación serial.
    - 1x I<sup>2</sup>C.
    - 3x SPI.
    - 2x UART.
    - Codificador y decodificador IrDA.
  - 5x temporizadores de 16 bits.
- ✓ Técnicas de mitigación y tolerancia a fallas. **(Requisito 3, imp. 5)**
  - CRC de 16 bits.
  - Temporizador watchdog.

Las especificaciones finales del microcontrolador supervisor se presentan en la *Tabla 18*.

*Tabla 18. Especificaciones finales del microcontrolador supervisor a  $T_A = 25^\circ\text{C}$  [67]*

No.	Métrica	No. de requisito	Imp.	Parámetro	Min.	Tip.	Max.	Unid.
1	Tensión de operación	5	3	$V_{DD}$	1.8	-	3.6	V
2	Corriente de operación ( $V_{DD} = 3.0\text{V}, f_{CLK} = 16\text{MHz}$ )	5	2	$I_{DD}$	-	-	260	$\mu\text{A}$

3	Frecuencia de operación	1	2	$f_{CLK}$	0	-	16	MHz
4	Temperatura de operación	4	5	$T_A$	-55	-	105	°C
5	Temperatura de almacenamiento	4	5	$T_{STG}$	-55	-	125	°C
6	Resistencia a la radiación acumulada	3, 4	5	-	-	50	-	krad
7	Inmunidad a SEL ( $T_A = 125$ °C)	4	5	-	-	72	-	MeVcm <sup>2</sup> /mg
8	Precio	6	2	-	23.76			USD
9	Encapsulado	6, 7	1	-	HTQFP (48)			

### Sistema Operativo en Tiempo Real (RTOS)

Como se menciona anteriormente, la implementación del RTOS no está contemplada para este trabajo, pero es importante considerarse para seleccionar el microcontrolador principal y que como trabajo a futuro se realicen las tareas correspondientes para completar el diseño del SCMI sin la necesidad modificar considerablemente el primer diseño.

El *Estado del arte* indica que el procesador ARM Cortex-M4 es altamente utilizado en SCMI para CubeSat, por lo que es de esperar que cumpla con las características necesarias para implementar un RTOS. Además, en [19] se justifica la selección del microcontrolador STM32F407IGT6, en donde se implementa FreeRTOS, sistema operativo que cumple con los requisitos de bajo costo, disponibilidad, uso de memoria y desempeño.

FreeRTOS es un sistema operativo pequeño con un nivel de desempeño aceptable en comparación con otros RTOS, pero que, al ser de código abierto, presenta gran disponibilidad de su documentación para distintas arquitecturas de microcontrolador a un bajo costo, un alto soporte técnico por la gran comunidad que lo utiliza. Adicionalmente, FreeRTOS se utiliza frecuentemente en los SCMI de proveedores, indicando que el microcontrolador STM32F407IGT6 es una buena elección para que se implemente FreeRTOS como trabajo futuro.

### Integración

El microcontrolador principal (STM32F407IGT6) se encarga de realizar todas las tareas de procesamiento del SCMI, por lo que requiere establecer comunicación con el resto de los módulos a través de sus interfaces de comunicación serial. En cambio, el microcontrolador supervisor (MSP430FR5969-SP) únicamente requiere comunicarse con el principal para realizar tareas de diagnóstico y recuperación.

La comunicación entre ambos microcontroladores se realiza mediante un bus SPI y líneas GPIO. SPI se utiliza para establecer una comunicación simple y rápida de únicamente dos nodos, a través de la cual se transmiten datos entre ambos microcontroladores, correspondiendo el microcontrolador principal al maestro y el supervisor al esclavo. Las líneas GPIO del supervisor se utilizan para transmitir señales de estado y control entre ambos microcontroladores y algunas se conectan a la interfaz JTAG del microcontrolador principal para funciones de depuración, además de que el supervisor puede reconfigurarlo si se detecta algún mal comportamiento. Dicho de otra forma, el microcontrolador supervisor se encarga de las funciones de diagnóstico del microcontrolador principal, verificando que funcione correctamente. Además, también sirve como temporizador watchdog, en donde el microcontrolador principal envía constantemente una señal que reinicie un contador en el supervisor, de modo que si este llega a un valor de cuenta determinado significa que el microcontrolador principal tuvo un problema y quedó atrapado en un bucle infinito, por lo que el supervisor obliga un reinicio para que el microcontrolador principal se reestablezca a un estado operativo funcional.

Ambos microcontroladores se programan a través de un puerto JTAG, por lo que para ahorrar recursos se utiliza un interruptor para seleccionar que microcontrolador se desea programar en tierra y, una vez en vuelo, la interfaz JTAG del microcontrolador principal es controlada por las líneas de control GPIO del supervisor.

La integración de los dispositivos del procesador de a bordo se presenta en la *Figura 92*.

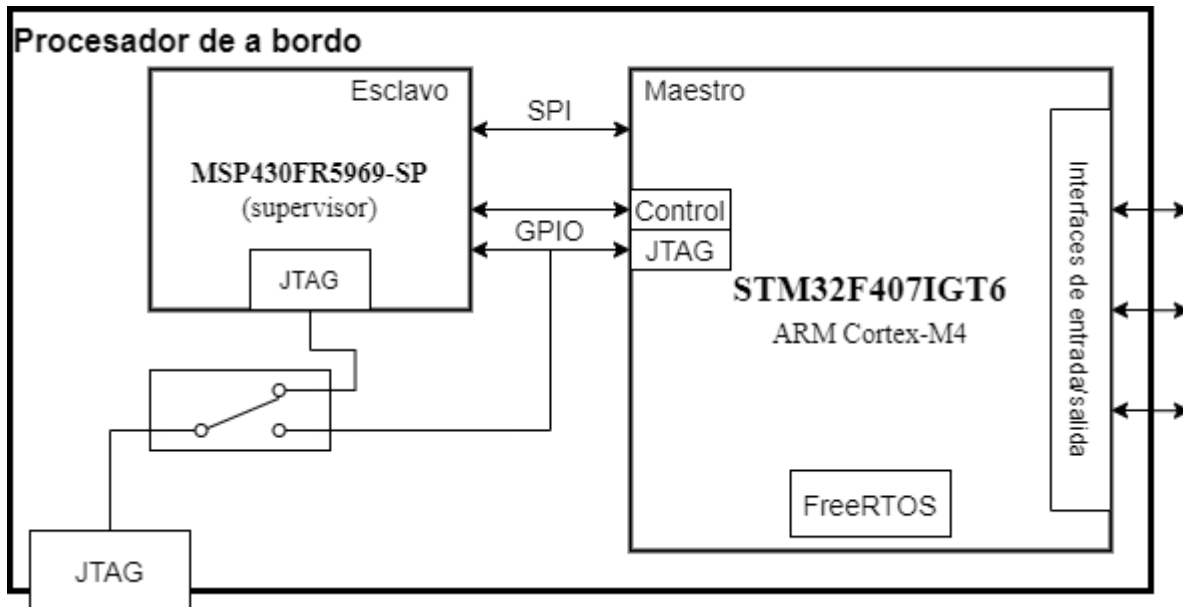


Figura 92. Diagrama del diseño de detalle del procesador de a bordo

## 10.2.2 Diseño de detalle de los dispositivos de almacenamiento

### Selección de componentes

Para seleccionar los dispositivos de memoria es esencial que su capacidad de almacenamiento sea la suficiente para retener toda la información hasta que se termine la misión o se transmitan los datos a una estación terrestre. En este trabajo, el SCMI no se diseña para desempeñar una misión en particular, por lo que no se cuenta con los datos suficientes para hacer la estimación de la capacidad de almacenamiento mínimo. Debido a esto, la selección de memorias se centra en la determinación del tipo de tecnología, definiendo por ahora una capacidad de almacenamiento intermedia en el rango identificado en el *Estado del arte*, con el objetivo de realizar pruebas con el primer prototipo.

Para seleccionar la tecnología es de gran ayuda utilizar la *Tabla 9*, en donde se realiza la comparación de los diferentes tipos de memoria.

### Memoria de trabajo

Las tecnologías utilizadas para la memoria de trabajo son la DRAM, SRAM y SDRAM. Las celdas de la DRAM están constituidas únicamente por un condensador y un transistor FET, lo que la lleva a poseer una gran capacidad de almacenamiento a bajo costo, pero cuenta con poca velocidad a comparación de la competencia. En cambio, la SRAM posee mayor velocidad a un menor consumo de energía a cambio de un mayor precio y tamaño porque sus celdas de memoria se construyen con un mayor número de componentes. Por última parte, la SDRAM que, en esencia, se trata de una DRAM síncrona que permite controlar su segmentación, lo que la convierte en una opción más rápida, pero con una menor densidad de almacenamiento.

Con base en lo anterior y en el *Estado del arte*, se opta por una tecnología SRAM como memoria de trabajo. Se puede seleccionar una memoria SRAM serial o paralela, las cuales tienen la diferencia fundamental que presentan los modos de comunicación serie y paralelo: velocidad contra cantidad de hardware. Para este proyecto se selecciona un banco de dos SRAM asíncronas paralelas IS64WV512EDBLL de 512 kB cada una (1 MB en

total) con ECC independiente con códigos de Hamming para cada byte, de Integrated Silicon Solution Inc (ISSI), cuyas especificaciones se presentan en la *Tabla 19*.

*Tabla 19. Especificaciones finales de la memoria de trabajo [68]*

No.	Métrica	No. de requisito	Imp.	Parámetro	Min.	Tip.	Max.	Unid.
1	Tensión de operación	5	4	$V_{DD}$	2.4	3.3	3.6	V
2	Corriente de operación	5	4	$I_{DD}$			35	mA
3	Consumo de energía	5	4	$P_D$		85 7	1000	mW
4	Capacidad de almacenamiento	8	5	-		1M		B
5	Organización	2, 8	2	-			512k*8	
6	Tiempo de acceso	1	5			8 10		ns
7	Temperatura de operación	4	5	$T_A$	-40	-	125	°C
8	Temperatura de almacenamiento	4	5	$T_{STG}$	-65	-	150	°C
9	Precio	6	3	-		7.91		USD
10	Encapsulado	6, 7	1	-				TSOP2 (44) / BGA (36) / SOJ (36)

#### *Memoria de almacenamiento masivo de datos*

Para la memoria de almacenamiento masivo se seleccionan dos tecnologías: Flash y SD. La memoria Flash consiste en un banco de 512 MB (cuatro memorias de 128 MB cada una) de NOR Flash serial (IS25LE01G) con ECC, fabricada por ISSI. Además, tanto para obtener un sistema redundante como para aumentar la capacidad de almacenamiento, se añade un adaptador para tarjeta microSD. En la *Tabla 20* se presentan las especificaciones del banco de memoria Flash.

*Tabla 20. Especificaciones finales de la memoria de almacenamiento masivo de datos (Flash) [69]*

No.	Métrica	No. de requisito	Imp.	Parámetro	Min.	Tip.	Max.	Unid.
1	Tensión de operación	5	4	$V_{DD}$	2.3	3.0	3.6	V
2	Frecuencia de operación	1	3	$f_{CLK}$	-	50	133	MHz
3	Capacidad de almacenamiento	8	5	-		512M		B
4	Organización	2, 8	2	-			128M*8	



5	Ciclos de lectura y escritura	3	4	$N_{END}$	$10^{14}$	-	-	ciclos
6	Retención de datos ( $T_A = 55^\circ C$ )	3	5	$t_{RET}$	-	120	-	años
	Retención de datos ( $T_A = 75^\circ C$ )				-	12.8	-	
7	Resistencia a Latch-Up	3, 4	4	-	-100	-	100	mA
8	Tiempo de ciclo de escritura	1	3	-	-	2	-	ms
9	Temperatura de operación	4	5	$T_A$	-40	-	105	$^\circ C$
10	Temperatura de almacenamiento	4	5	$T_{STG}$	-60	-	150	$^\circ C$
11	Precio	6	3	-	15.10			USD
12	Encapsulado	6, 7	1	-	LFBGA (24)			

### Memoria de almacenamiento crítico de datos

La memoria de almacenamiento crítico únicamente guardará los datos con mayor relevancia para que el satélite complete satisfactoriamente la misión. Esto indica que no es necesario que la memoria cuente con una alta capacidad como la memoria masiva, pero sí un nivel de confiabilidad mayor que asegure que los datos almacenados en ella no sufran ninguna perturbación. En otras palabras, requiere tener una alta tolerancia a la radiación.

Las tecnologías utilizadas para este fin son la FRAM y la MRAM. La FRAM es un tipo de memoria de estado sólido que utiliza una capa de material ferroeléctrico en lugar de un dieléctrico, que aprovecha el movimiento del átomo dentro de la estructura cristalina al aplicársele un campo eléctrico. Sus principales ventajas son su gran velocidad, bajo consumo de energía (opera con un bajo nivel de tensión y requiere de poca corriente para escribir), su alta resistencia a los campos eléctricos externos, su inmunidad a los campos magnéticos y su elevada resistencia a la radiación. Sin embargo, se trata de una memoria que ofrece poca capacidad de almacenamiento a un precio elevado. Por otra parte, la MRAM está construida por dos discos ferromagnéticos que pueden mantener campos magnéticos para almacenar los datos. Se utiliza normalmente en aplicaciones espaciales y militares gracias a que posee una velocidad cercana a la SRAM, capacidad de almacenamiento comparable con la DRAM a un consumo menor de energía. Sin embargo, para la generación del campo magnético en el proceso de escritura requiere de una alta corriente.

Con base en la comparación de ambas tecnologías, para fines de este proyecto y aprovechando los recursos ya considerados para el diseño, el microcontrolador supervisor MSP430FR5969-SP contiene una FRAM de 64 kB con ECC para almacenar el programa y los datos. Considerando que el microcontrolador principal mantiene una comunicación constante con el supervisor y que este último únicamente realiza tareas pequeñas de diagnóstico y recuperación no requiere de mucha memoria, se puede aprovechar el espacio no utilizado de la FRAM para este fin.

Tabla 21. Especificaciones finales de la memoria de almacenamiento crítico de datos [67]

No.	Métrica	No. de requisito	Imp.	Parámetro	Min.	Tip.	Max.	Unid.
1	Tensión de operación	5	4	$V_{DD}$	1.8	-	3.6	V

2	Capacidad de almacenamiento*	8	5	-	-	-	64k	B
3	Ciclos de lectura y escritura	3	3	$N_{END}$	$10^{14}$	-	-	ciclos
	Retención de datos ( $T_A = 25^\circ C$ )				100	-	-	
4	Retención de datos ( $T_A = 70^\circ C$ )	3	5	$t_{RET}$	40	-	-	años
	Retención de datos ( $T_A = 105^\circ C$ )				2.5	-	-	

\* La capacidad de almacenamiento va a depender de la cantidad de software y datos almacenados en el supervisor.

#### Memoria de almacenamiento de código

Para almacenar el código es muy común utilizar memoria Flash. Para este trabajo, el microcontrolador principal posee 1 MB de memoria Flash embebida, por lo que no se considera necesario implementar una memoria externa.

Tabla 22. Especificaciones finales de la memoria de almacenamiento de código [65]

No.	Métrica	No. de requisito	Imp.	Parámetro	Min.	Tip.	Max.	Unid.
1	Tensión de operación	5	4	$V_{DD}$	1.8	-	3.6	V
	Corriente de operación – escritura / lectura de 8 bits ( $V_{DD} = 1.8V$ )				-	5	-	
2	Corriente de operación – escritura / lectura de 16 bits ( $V_{DD} = 2.1V$ )	5	3	$I_{DD}$	-	8	-	mA
	Corriente de operación – escritura / lectura de 32 bits ( $V_{DD} = 3.3V$ )				-	12	-	
3	Capacidad de almacenamiento	8	5	-	-	-	1M	B
4	Ciclos de lectura y escritura	3	3	$N_{END}$	10k	-	-	ciclos
	Retención de datos ( $N_{END} = 1kciclo; T_A = 85^\circ C$ )				30	-	-	
5	Retención de datos ( $N_{END} = 1kciclo; T_A = 105^\circ C$ )	3	5	$t_{RET}$	10	-	-	años
	Retención de datos ( $N_{END} = 10kciclos; T_A = 55^\circ C$ )				20	-	-	
6	Tiempo de programación de palabra	1	2	$t_{prog}$	-	16	100	$\mu s$

**Integración**

La integración de los dispositivos de almacenamiento se observa en la *Figura 93*, en donde la memoria de trabajo (SRAM) se conecta a través del controlador flexible de memoria estática (FSMC, por sus siglas en inglés) del microcontrolador principal. Por otra parte, se establece un bus SPI con el microcontrolador principal como maestro, conectado a tres esclavos: el supervisor, utilizando su FRAM embebida como unidad de almacenamiento crítico; el banco de memorias Flash y la ranura para memoria microSD, ambas para el almacenamiento masivo de datos. Por último, se presenta la memoria de almacenamiento de código (Flash) embebida en el microcontrolador principal.

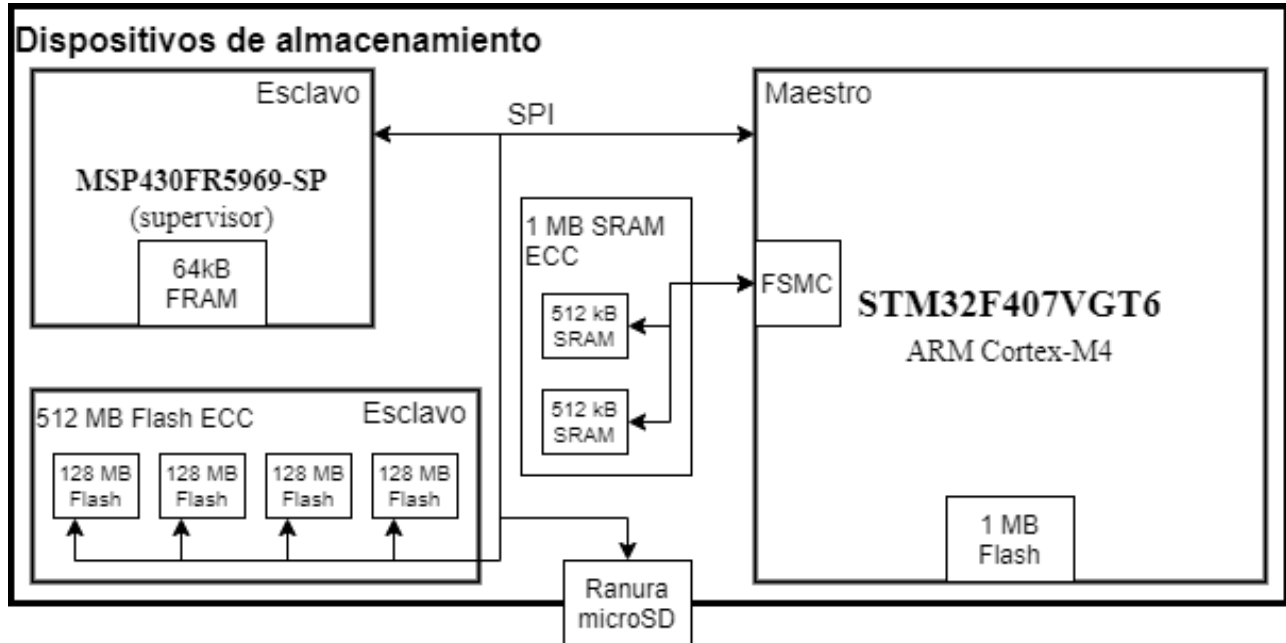


Figura 93. Diagrama del diseño de detalle de los dispositivos de almacenamiento

**10.2.3 Diseño de detalle de los puertos de entrada/salida0**

En la *Tabla 15* se enlistan los tipos, características y la cantidad máxima de puertos de entrada/salida disponibles en el microcontrolador principal. Con base en ella, en la *Tabla 23* se presentan las especificaciones finales de cada uno de estos puertos.

Tabla 23. Especificaciones finales de los puertos de E/S [65]

No.	Métrica	Parámetros	Parámetro	Min.	Tip.	Max.	Unid.
<b>I<sup>2</sup>C</b>							
1	Tasa de transferencia	Modo estándar	-	-	-	100	kbit/s
		Modo rápido	-	-	-	400	
<b>USART</b>							
2	Frecuencia de operación	USART1, USART6	-	-	-	84	MHz
		USART2, USART3	-	-	-	42	
3	Tasa de transferencia	USART1, USART6	-	-	-	10.5	Mbit/s

		USART2, USART3		-	-	5.25	
<b>UART</b>							
4	Frecuencia de operación	UART4, UART5	-	-	-	42	MHz
5	Tasa de transferencia	UART4, UART5	-	-	-	5.25	Mbit/s
<b>SPI</b>							
6	Tasa de transferencia	SPI1	-	-	-	42	Mbit/s
		SPI2, SPI3	-	-	-	21	
<b>CAN</b>							
7	Tasa de transferencia	CAN 2.0A/B	-	-	-	1	Mbit/s
<b>DCMI</b>							
8	Tasa de transferencia	Frecuencia = 54 MHz	-	-	-	54	MByte/s
<b>GPIO</b>							
9	Velocidad de cambio de estado	Entrada/salida	-	-	-	84	MHz
<b>ADC</b>							
10	Tensión de alimentación	-	$V_{DD}$	1.8	-	3.6	V
11	Frecuencia de operación	$V_{DD} = 1.8 - 2.4 V$	$f_{ADC}$	0.6	15	18	MHz
		$V_{DD} = 2.4 - 3.6 V$		0.6	30	36	

Es importante mencionar que la definición de los puertos de entrada/salida seleccionados para estar disponibles en el SCMI se realiza tomando como base la investigación del *Estado del arte*, dándole prioridad a aquellos puertos identificados con mayor frecuencia en satélites CubeSat. Además, la cantidad de puertos disponibles en el SCMI depende del diseño del factor de forma, documentado más adelante.

## 10.2.4 Diseño de detalle de los dispositivos externos

### Selección de componentes

#### Reloj de tiempo real (RTC)

El microcontrolador supervisor no se utiliza para tareas complejas de procesamiento, por lo que se decide aprovechar su alta resistencia a la radiación para reducir la cantidad de dispositivos empleando su RTC embebido, el cual está disponible en el modo de bajo consumo de energía. Sus especificaciones se presentan en la *Tabla 24*.

Tabla 24. Especificaciones finales del RTC [67]

No.	Métrica	No. de requisito	Imp.	Parámetro	Min.	Tip.	Max.	Unid.
1	Tensión de operación	5	4	$V_{DD}$	2.2	-	3.6	V
2	Corriente de operación ( $T_A = 25^\circ\text{C}$ )	5	3	$I_{DD}$	-	250	-	nA
3	Frecuencia de operación	1	3	$f_{CLK}$	-	32	50	kHz

*Sensor de temperatura*

Se define utilizar tres sensores ubicados en diferentes posiciones de la PCB para permitir la medición de temperatura de tres áreas distintas y contar con un sistema redundante. Para esto, se selecciona el sensor de temperatura digital TMP461-SP de TI, con clasificación de grado espacial, cuyas especificaciones se presentan en la *Tabla 25*.

Tabla 25. Especificaciones finales del sensor de temperatura [70]

No.	Métrica	No. de requisito	Imp.	Parámetro	Min.	Tip.	Max.	Unid.
1	Tensión de operación	5	4	$V_{DD}$	1.7	3.3	3.6	V
2	Corriente de operación	5	3	$I_{DD}$	-	240	375	$\mu\text{A}$
3	Frecuencia de operación	1	2	$f_{CLK}$	-	-	2.17	MHz
4	Precisión de temperatura local	3	4	-	-2	$\pm 0.5$	2	$^\circ\text{C}$
	Precisión de temperatura remota				-1.5	$\pm 0.12$	1.5	
5	Resolución	2	4	-	0.0625			$^\circ\text{C}$
6	Rango de medición	2	3	-	-64	-	191	$^\circ\text{C}$
7	Temperatura de operación	4	5	$T_A$	-55	-	125	$^\circ\text{C}$
8	Temperatura de almacenamiento	4	5	$T_{STG}$	-60	-	160	$^\circ\text{C}$
9	Precio	6	3	-	1541.728 (100 unidades)			USD
10	Encapsulado	6, 7	1	-	CFP (10)			

*Acelerómetro*

Se decide utilizar dos acelerómetros digitales de 3 ejes ADXL313 de Analog Devices por redundancia, los cuales estarán ubicados en distintas zonas de la PCB para disminuir la probabilidad de fallas comunes en los dos dispositivos. En la *Tabla 26* se presentan sus especificaciones.

*Tabla 26. Especificaciones finales del acelerómetro [71]*

No.	Métrica	No. de requisito	Imp.	Parámetro	Min.	Tip.	Max.	Unid.
1	Tensión de operación	5	4	$V_{DD}$	2.0	3.3	3.6	V
2	Corriente de operación (tasa de datos > 100 Hz)	5	3	$I_{DD}$	100	170	300	$\mu\text{A}$
	30				55	110		
3	Resolución	2	3	-	10, 11, 12, 13			bit
4	Tasa de datos		3	-	6.25	-	3200	Mdps/digit
5	Rango de medición	2	5	-	$\pm 0.5, \pm 1, \pm 2, \pm 4$			g
6	Sensibilidad	2	4	-	921	1024	1126	LSB/g
					460	512	563	
					230	256	282	
					115	128	141	
7	Aceleración	2	3	-	10,000			g
8	Temperatura de operación	4	5	$T_A$	-40	-	105	$^{\circ}\text{C}$
9	Temperatura de almacenamiento	4	5	$T_{STG}$	-40	-	125	$^{\circ}\text{C}$
10	Precio	6	2	-	4.44			USD
11	Encapsulado	6, 7	1	-	LFCSP (32)			

*Giroscopio*

Se selecciona el giroscopio digital de 3 ejes L3GD20HTR de STMicroelectronics, cuyas especificaciones se presentan en la *Tabla 27*. Similar al acelerómetro, se utilizan dos sensores por redundancia y ubicados en distintas zonas de la PCB para disminuir la probabilidad de fallas comunes.

Tabla 27. Especificaciones finales del giroscopio [72]

No.	Métrica	No. de requisito	Imp.	Parámetro	Min.	Tip.	Max.	Unid.
1	Tensión de operación	5	4	$V_{DD}$	2.2	3	3.6	V
2	Corriente de operación	5	3	$I_{DD}$	-	5	-	mA
3	Resolución	2	3	-		16		bit
4	Rango de medición	2	5	-	$\pm 245, \pm 500, \pm 2000$			g
5	Sensibilidad	2	4	$S_O$	-	8.75 17.50 70.00	-	mdps/digit
6	Aceleración	2	4	-	-	1000	-	g
7	Temperatura de operación	4	5	$T_A$	-40	-	85	°C
8	Temperatura de almacenamiento	4	5	$T_{STG}$	-40	-	125	°C
9	Precio	6	2	-		3.25		USD
10	Encapsulado	6, 7	1	-			LGA (16)	

### Magnetómetro

Se selecciona el magnetómetro de 3 ejes digital HMC6343 de Honeywell Aerospace, cuyas especificaciones se presentan en la *Tabla 28*. Similar al acelerómetro y giroscopio, se utilizan dos magnetómetros redundantes ubicados en distintas zonas de la PCB, con el objetivo de disminuir la probabilidad de fallas comunes.

Tabla 28. Especificaciones finales del magnetómetro [73]

No.	Métrica	No. de requisito	Imp.	Parámetro	Min.	Tip.	Max.	Unid.
1	Tensión de operación	5	4	$V_{DD}$	2.7	3.3	3.6	V
2	Corriente de operación	5	3	$I_{DD}$	-	4.5	-	mA
3	Resolución	2	3	-		0.1		°
4	Rango de campo	2	4	-	-1500	-	1500	mgauss
5	Rango de medición	2	5	-	-	$\pm 2$	-	G
6	Temperatura de operación	4	5	$T_A$	-40	-	85	°C

7	Temperatura de almacenamiento	4	5	$T_{STG}$	-55	-	125	°C
8	Precio	6	2	-			100.7	USD
9	Encapsulado	6, 7	1	-			LCC (36)	

### Integración

Todos los dispositivos externos se comunican a través de I<sup>2</sup>C, por lo que se asigna un bus I<sup>2</sup>C del microcontrolador principal para este fin, como se observa en la *Figura 94*.

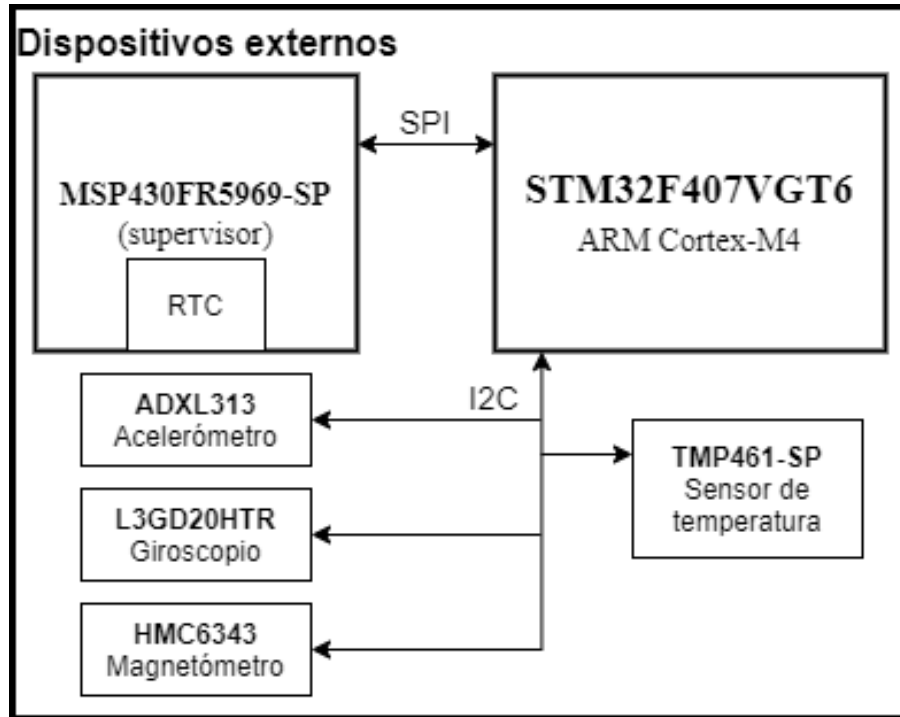


Figura 94. Diagrama del diseño de detalle de los dispositivos externos

### 10.2.5 Diseño de detalle del factor de forma

Como se mencionó anteriormente, se opta por implementar el factor de forma PC/104, respetando principalmente las especificaciones mecánicas, debido a que el estándar PC/104 utiliza el bus ISA. La *Figura 95* presenta el diseño mecánico (de acuerdo con las especificaciones planteadas en [35]) de la PCB del factor de forma PC/104 en el software EAGLE, en donde las áreas A, B y C representan el espacio disponible para conectores de entrada/salida integrados en la PCB. Además, debido a que se está considerando modificar el tamaño del conector 104, únicamente se coloca como referencia el área de su ubicación, considerando las 104 terminales.



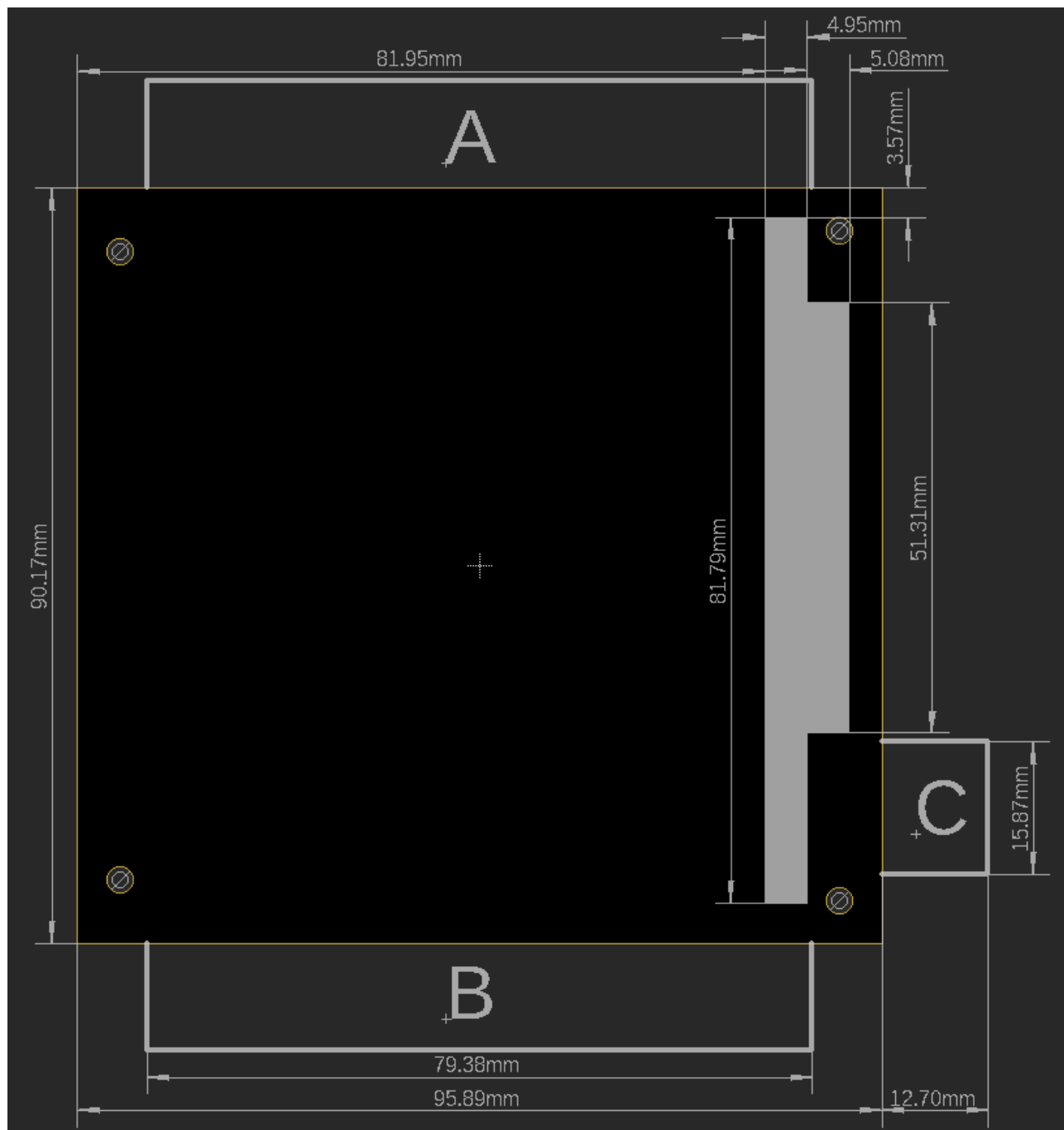


Figura 95. Diseño de la PCB del factor de forma PC/104 en EAGLE

### 10.3 Integración

El paso de integración se refiere al diseño del sistema y la interfaz para integrar cada módulo en un único sistema. Para comenzar, en la *Figura 96* se presenta el diagrama de diseño de detalle con todos los módulos descritos anteriormente.

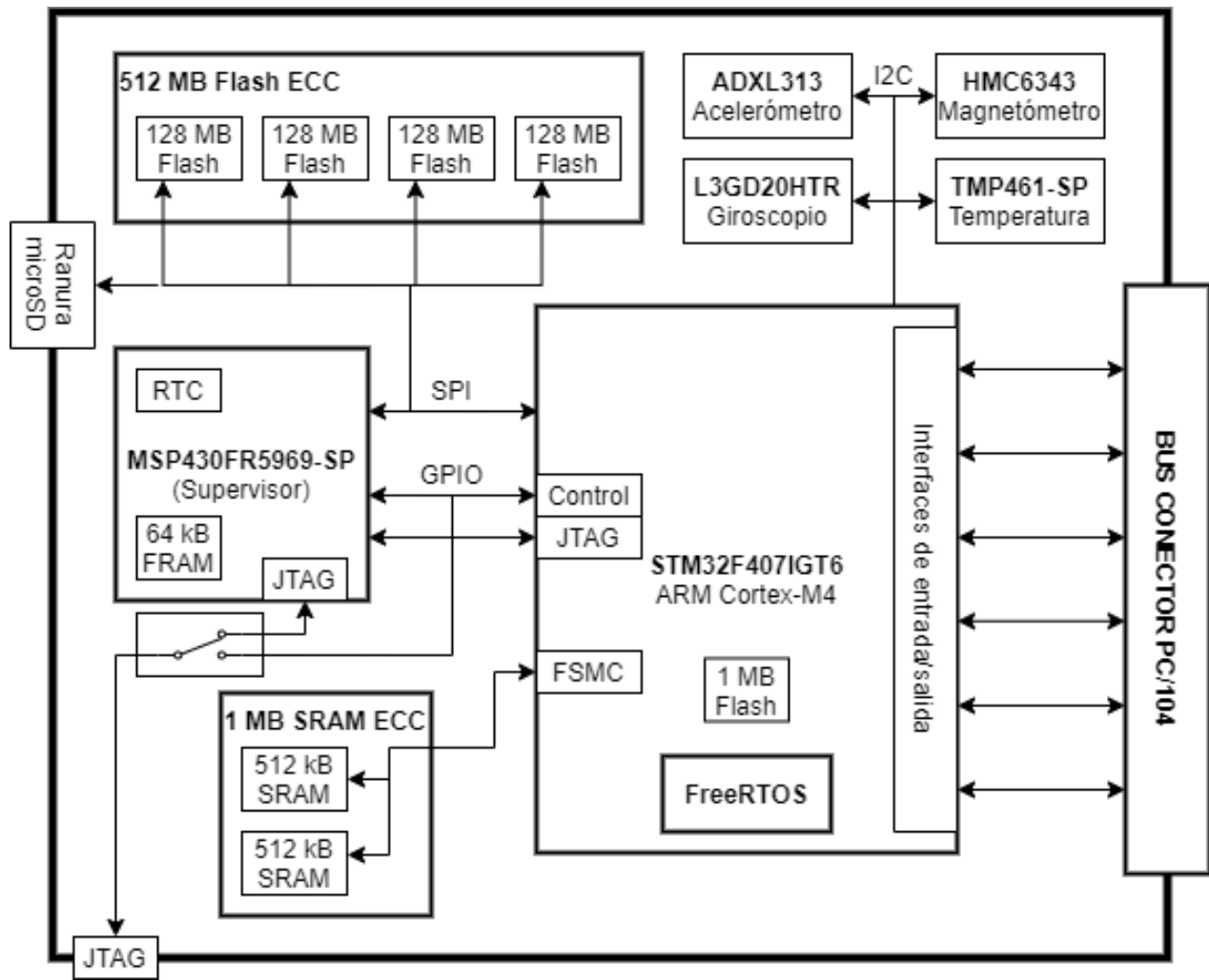


Figura 96. Diagrama de diseño de detalle del SCMI

### 10.3.1 Asignación de terminales

A continuación, se realiza la asignación y descripción de terminales de ambos microcontroladores, agrupándose para elementos conectados internamente en el SCMI.

#### Controlador flexible de memoria estática (FSMC)

La comunicación entre el microcontrolador principal y la SRAM se realiza a través del controlador de memoria (FSMC), indicándose las conexiones en la *Tabla 29*.

Tabla 29. Asignación de terminales del FSMC [65] [68]

Puerto (STM32)	Número de terminal (STM32)	Función	Conexión
PD4	146	FSMC_ $\overline{\text{OE}}$	SRAM0_ $\overline{\text{OE}}$ SRAM1_ $\overline{\text{OE}}$
PD5	147	FSMC_ $\overline{\text{WE}}$	SRAM0_ $\overline{\text{WE}}$ SRAM1_ $\overline{\text{WE}}$

PF0	16	FSMC_A0	SRAM0_A0 SRAM1_A0
PF1	17	FSMC_A1	SRAM0_A1 SRAM1_A1
PF2	18	FSMC_A2	SRAM0_A2 SRAM1_A2
PF3	19	FSMC_A3	SRAM0_A3 SRAM1_A3
PF4	20	FSMC_A4	SRAM0_A4 SRAM1_A4
PF5	21	FSMC_A5	SRAM0_A5 SRAM1_A5
PF12	60	FSMC_A6	SRAM0_A6 SRAM1_A6
PF13	63	FSMC_A7	SRAM0_A7 SRAM1_A7
PF14	64	FSMC_A8	SRAM0_A8 SRAM1_A8
PF15	65	FSMC_A9	SRAM0_A9 SRAM1_A9
PG0	66	FSMC_A10	SRAM0_A10 SRAM1_A10
PG1	67	FSMC_A11	SRAM0_A11 SRAM1_A11
PG2	106	FSMC_A12	SRAM0_A12 SRAM1_A12
PG3	107	FSMC_A13	SRAM0_A13 SRAM1_A13
PG4	108	FSMC_A14	SRAM0_A14 SRAM1_A14
PG5	109	FSMC_A15	SRAM0_A15 SRAM1_A15
PD11	99	FSMC_A16	SRAM0_A16 SRAM1_A16

PD12	100	FSMC_A17	SRAM0_A17 SRAM1_A17
PD13	101	FSMC_A18	SRAM0_A18 SRAM1_A18
PE3	2	FSMC_A19*	SRAM0_ $\overline{\text{CE}}$ SRAM1_ $\overline{\text{CE}}$
PD14	104	FSMC_D0	SRAM0_I/O0 SRAM1_I/O0
PD15	105	FSMC_D1	SRAM0_I/O1 SRAM1_I/O1
PD0	142	FSMC_D2	SRAM0_I/O2 SRAM1_I/O2
PD1	143	FSMC_D3	SRAM0_I/O3 SRAM1_I/O3
PE7	68	FSMC_D4	SRAM0_I/O4 SRAM1_I/O4
PE8	69	FSMC_D5	SRAM0_I/O5 SRAM1_I/O5
PE9	70	FSMC_D6	SRAM0_I/O6 SRAM1_I/O6
PE10	73	FSMC_D7	SRAM0_I/O7 SRAM1_I/O7

\* La SRAM únicamente utiliza 19 bits de dirección (A0-A18). Sin embargo, al utilizar dos memorias, se añade otro bit de dirección (A19) para expandir la memoria, conectada a la línea de habilitación de ambas memorias (una de ellas negada), multiplexando las líneas de dirección, datos y control en ambas memorias. Lo explicado anteriormente se observa en la *Figura 97*.

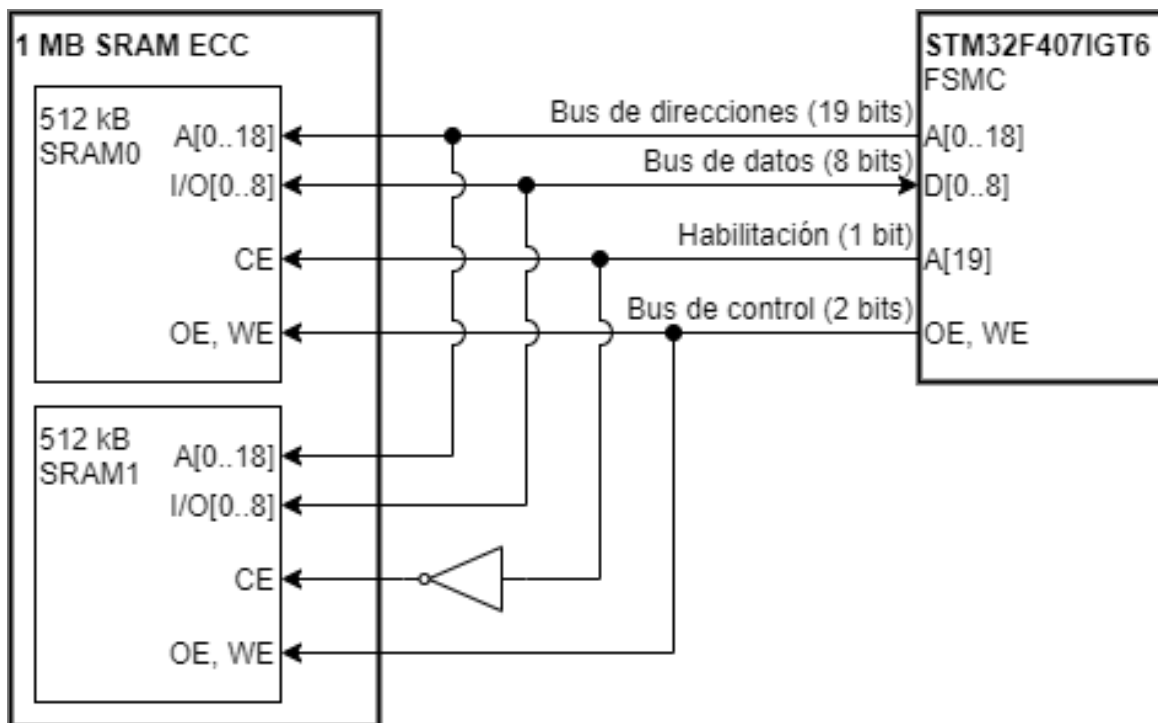


Figura 97. Diagrama de conexiones del FSMC

### Configuración y control

Para la programación de ambos microcontroladores se establecen tres modos de configuración y control: el modo terrestre – MSP430, el modo terrestre – STM32 y el modo vuelo, los cuales se seleccionan a través de un arreglo de jumpers de configuración, cuyas conexiones se enlistan en la *Tabla 30* y el diagrama de conexiones se presenta en la *Figura 98*.

Tabla 30. Asignación de terminales para configuración y control [65] [67]

Fila Función	JP1 (MSP430) Puerto (# de terminal)	JP2 (puerto JTAG) (# de terminal)	JP3 (STM32) Puerto - # de terminal	JP4 (MSP430) Puerto - # de terminal
F1 PUSH RST	-	PUSH RST	NRST (31)	P2.7 (38)
F2 TMS	PJ.2 (14)	TMS (2)	PA13 (124)	P2.2 (26)
F3 TCK	PJ.3 (15)	TCK (4)	PA14 (137)	P3.4 (27)
F4 TDO	PJ.0 (12)	TDO (6)	PB3 (161)	P3.5 (28)
F5 TDI	PJ.1 (13)	TDI (8)	PA15 (138)	P3.6 (29)
F6 JTAG RST	$\overline{\text{RST}}$ (23)	RST (10)	PB4 (162)	P3.7 (30)

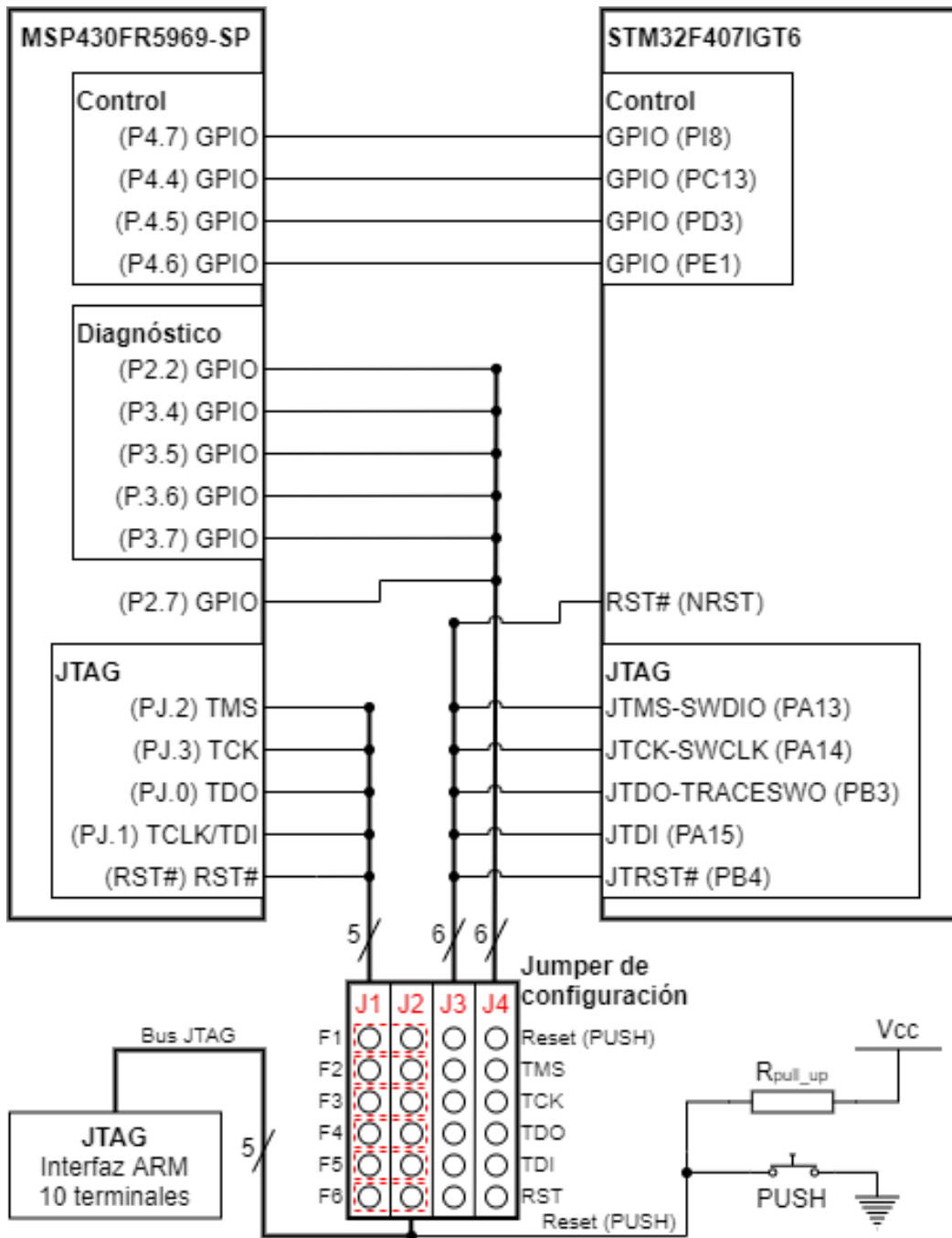


Figura 98. Diagrama de conexiones para configuración y control (general)

Modo terrestre – MSP430

Este modo se selecciona conectando las terminales de J1 con las terminales de J2, obteniendo el circuito correspondiente para la programación y depuración del MSP430 (supervisor) a través del puerto JTAG. Las conexiones correspondientes se pueden observar en la *Tabla 30* y el diagrama de conexiones se presenta en la *Figura 99*.

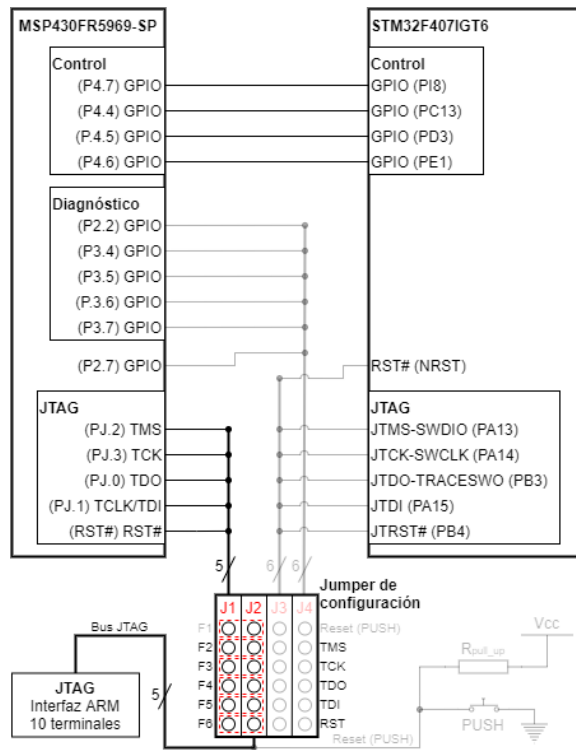


Figura 99. Diagrama de conexiones para configuración y control (modo terrestre – MSP430)

Modo terrestre – STM32

Al seleccionar este modo se obtiene el circuito correspondiente para la programación y depuración del STM32 (microcontrolador principal) a través del puerto JTAG, además de conectar un PUSH para el reinicio manual del STM32. Para seleccionar este modo, se deben conectar las terminales de J2 con las terminales de J3, obteniendo el diagrama de conexiones de la *Figura 100*. En la *Tabla 30* se pueden observar las conexiones correspondientes.

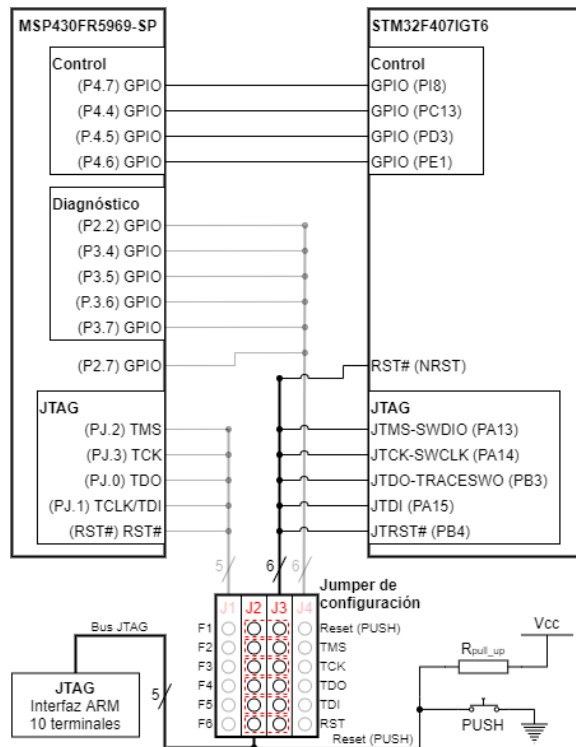


Figura 100. Diagrama de conexiones para configuración y control (modo terrestre – STM32)

**Modo de vuelo**

El modo de vuelo define el circuito correspondiente para que el supervisor controle las líneas JTAG y reset del microcontrolador principal a través de terminales GPIO, permitiendo realizar el diagnóstico y reconfiguración del STM32 en caso de la presencia de fallas. En otras palabras, el modo de vuelo se selecciona después de haber configurado en tierra (a través de los dos modos anteriores) ambos microcontroladores, preparando al sistema para realizar su misión correspondiente. Las conexiones realizadas en este modo se pueden observar en la *Tabla 30*, mientras que su diagrama de conexiones se presenta en la *Figura 101*.

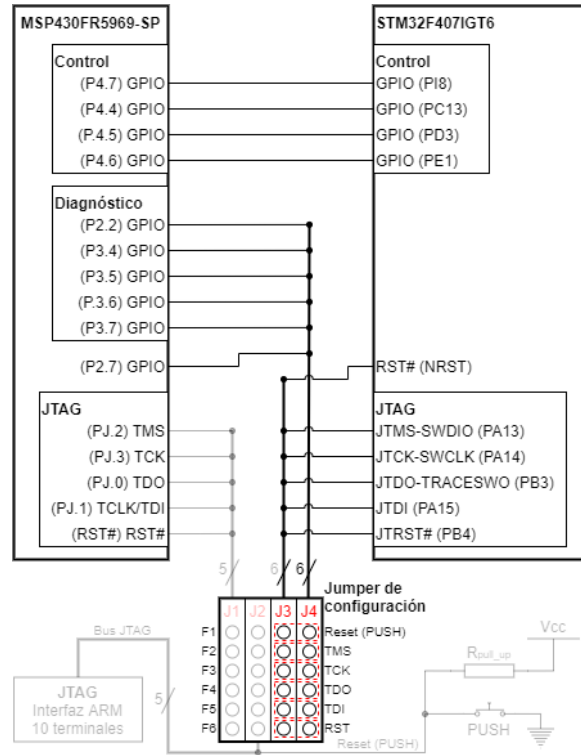


Figura 101. Diagrama de conexiones para configuración y control (modo de vuelo)

**Interfaz periférico serial (SPI) interno**

El bus SPI interno comunica el microcontrolador principal con el supervisor y las memorias de almacenamiento masivo Flash y SD. Para esto, el microcontrolador STM32 cuenta con tres buses SPI, de los cuales se analizan las funciones alternas de sus terminales para determinar aquel cuyo uso disminuya lo menos posible la cantidad y tipos de periféricos que ofrece el microcontrolador. Después de este análisis, se selecciona al bus SPI2 y sus terminales correspondientes (véase la *Tabla 31*), con el cual no se reduce nada la cantidad de periféricos del STM32.

Tabla 31. Asignación de terminales del bus SPI interno [65] [67] [69]

Puerto (STM32)	Número de terminal (STM32)	Función	Conexión
PI1	132	SPI2_SCK	MSP430_eUSCI_A0_UCLK Flash0_SCK Flash1_SCK Flash2_SCK Flash3_SCK MicroSD_SCLK



PI2	133	SPI2_MISO	MSP430_eUSCI_A0_SOMI Flash0_SO Flash1_SO Flash2_SO Flash3_SO MicroSD_DO
PI3	134	SPI2_MOSI	MSP430_eUSCI_A0_SIMO Flash0_SI Flash1_SI Flash2_SI Flash3_SI MicroSD_DI
PG6	110	SPI2_CS0*	MSP430_eUSCI_A0_STE
PE11	74	SPI2_CS1*	Flash0_ $\overline{\text{CE}}$
PE12	75	SPI2_CS2*	Flash1_ $\overline{\text{CE}}$
PE13	76	SPI2_CS3*	Flash2_ $\overline{\text{CE}}$
PE14	77	SPI2_CS4*	Flash3_ $\overline{\text{CE}}$
PE15	78	SPI2_CS5*	MicroSD_ $\overline{\text{CS}}$

\* El microcontrolador no posee estas terminales de selección de esclavo, ya que sus buses SPI únicamente cuentan con una línea de selección, soportando hasta un solo esclavo por cada bus. Debido a esto, se destinan cinco GPIO (PG6, PE11, PE12, PE13, PE14 y PE15) como terminales de selección de esclavo para utilizar un solo bus SPI. El diagrama de conexiones se presenta en la *Figura 102*.

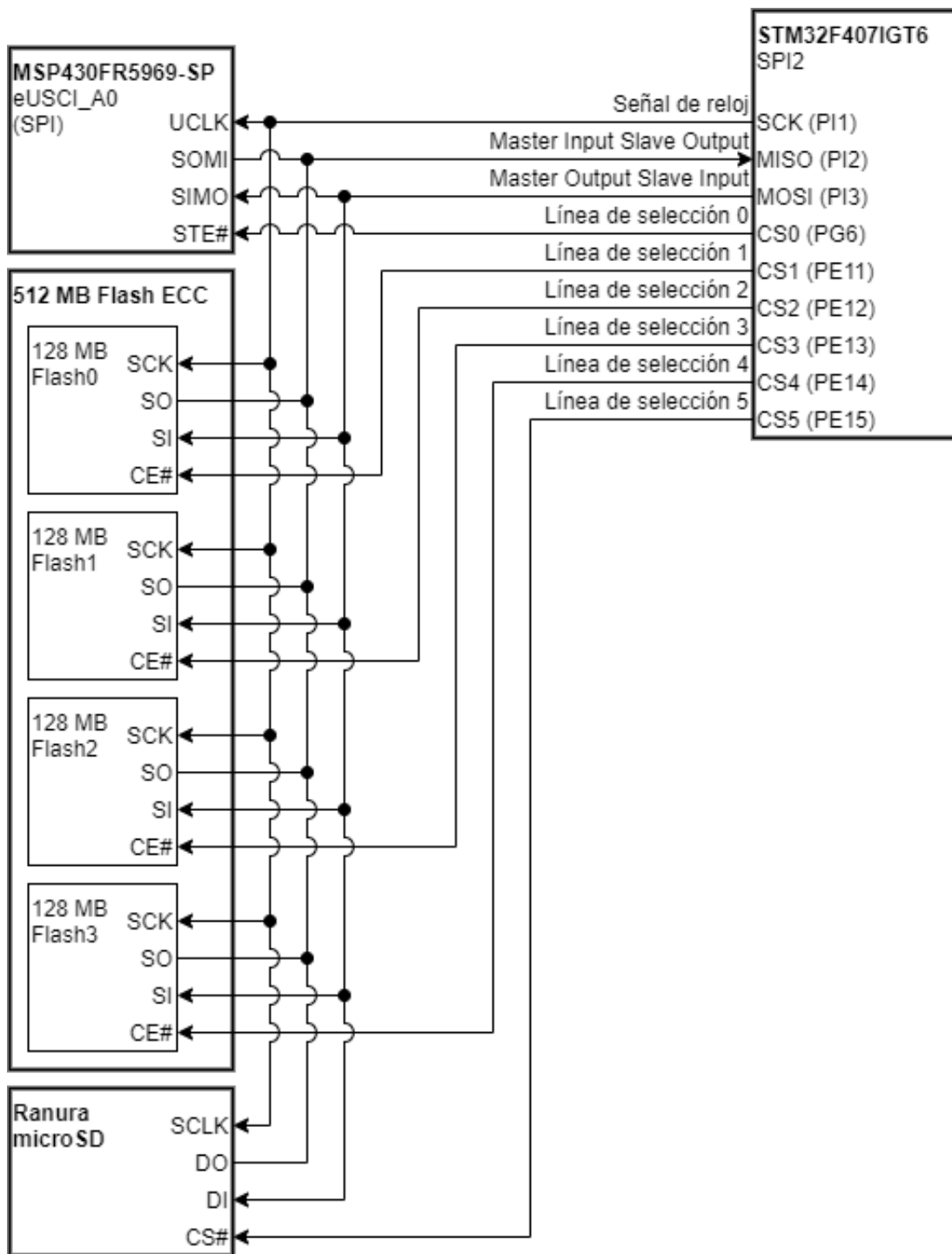


Figura 102. Diagrama de conexiones del bus SPI interno

### Circuito inter-integrado (I<sup>2</sup>C) interno

El microcontrolador STM32 cuenta con tres buses I<sup>2</sup>C, de los cuales se utiliza uno para establecer comunicación con los dispositivos externos: tres sensores de temperatura, dos acelerómetros, dos giroscopios y dos magnetómetros. De la misma manera que el bus SPI interno, se realiza un análisis de las terminales del microcontrolador y sus funciones alternas, de modo que se seleccione el bus I<sup>2</sup>C que disminuya la menor cantidad de periféricos disponibles en el STM32, en donde el bus I2C2 presenta esta característica. La selección y asignación de terminales se presenta en la *Tabla 32*.

Tabla 32. Asignación de terminales del bus I<sup>2</sup>C interno [65] [70] [71] [72] [73]

Puerto (STM32)	Número de terminal (STM32)	Función	Conexión
PH4	45	I2C2_SCL	TMP0_SCL
			TMP1_SCL
			TMP2_SCL
PH5	46	I2C2_SDA	ADXL0_SCL
			ADXL1_SCL
			L3GD0_SCL
			L3GD1_SCL
			HMC0_SCL
			HMC1_SCL
PH4	45	I2C2_SCL	TMP0_SDA
			TMP1_SDA
			TMP2_SDA
PH5	46	I2C2_SDA	ADXL0_SDA
			ADXL1_SDA
			L3GD0_SDA
			L3GD1_SDA
			HMC0_SDA
			HMC1_SDA

En la *Figura 103* se presenta el diagrama de conexiones del bus I<sup>2</sup>C interno.

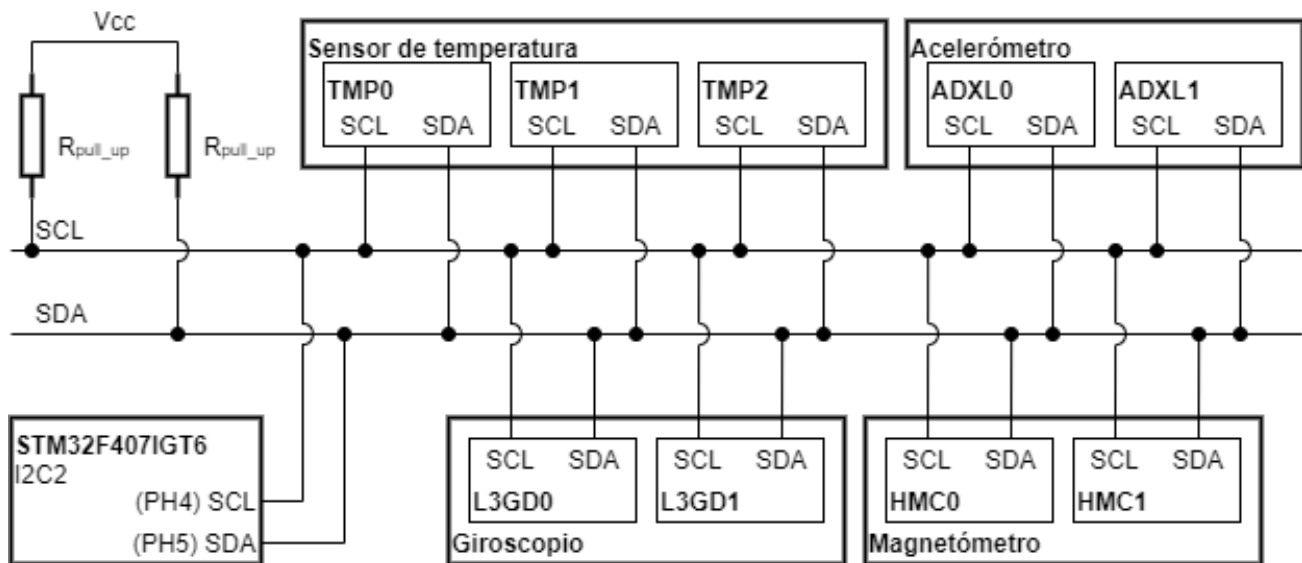


Figura 103. Diagrama de conexiones del bus I<sup>2</sup>C interno

### Conector PC/104

Como último paso del proceso de integración, se realiza la asignación de terminales en el conector descrito en el estándar PC/104, recordando que no es necesario respetar las especificaciones eléctricas. Como primer paso, se contabilizan las terminales disponibles del microcontrolador principal después de realizar las conexiones

internas del SCMI, dando como resultado 90 terminales libres. Como segundo paso, se observa en el estándar PC/104 que el conector cuenta con 14 líneas de alimentación ( $V_{CC}$  y GND), por lo que sin ningún problema se puede respetar el tamaño del conector y la ubicación de las terminales de alimentación.

Considerando las conexiones y dispositivos internos del SCMI, descritas anteriormente, el SCMI cuenta con la cantidad de puertos descritos en la *Tabla 33*.

*Tabla 33. Determinación de periféricos disponibles en el SCMI [65]*

<b>Interfaz</b>	<b>Módulo (Puertos)</b>	<b>Cantidad máxima*</b>
I <sup>2</sup> C	I2C1 (PB 5-9)	2
	I2C3 (PA 8-9 / PC 9 / PH 7-9)	
USART	USART1 (PA 8-12 / PB 6-7)	4
	USART2 (PA 0-4 / PD 6-7)	
	USART3 (PB 10-14 / PC 10-12 / PD 8-10)	
	USART6 (PC 6-8, / PG 7-9, 12-15)	
UART	UART4 (PA 0-1 / PC 10-11)	2
	UART5 (PC 12 / PD 2)	
SPI	SPI1 (PA 4-7 / PB 5)	2
	SPI3 (PA 4 / PB 5 / PC 10-12)	
CAN	CAN1 (PA 11-12 / PB 8-9 / PH 13 / PI 9)	2
	CAN2 (PB 5-6, 12-13)	
DCMI	DCMI (PA 4, 6, 9-10 / PB 5-9 / PC 6-9, 10-12 / PD 2 / PE 0, 4-6 / PF 11 / PG 15 / PH 8-12, 14-15 / PI 0, 4-7)	1
GPIO	GPIO (PA 0-12 / PB 0-2, 5-15 / PC 0-12, 14-15 / PD 2, 6-10 / PE 0, 2, 4-6 / PF 6-11 / PG 7-15 / PH 0-3, 6-15 / PI 0, 4-7, 9-11)	90
ADC	ADC3 (PF 6-10)	3
	ADC12 (PA 4-7 / PB 0-1 / PC 4-5)	
	ADC123 (PA 0-3 / PC 0-3)	

\* La cantidad depende de la configuración del microcontrolador, siendo imposible utilizar todos los puertos/interfaces de entrada/salida al mismo tiempo.

Por último, se realiza la asignación de terminales al conector PC/104, realizándose a manera de facilitar el diseño de la PCB. La asignación de terminales se presenta en la *Figura 104*.

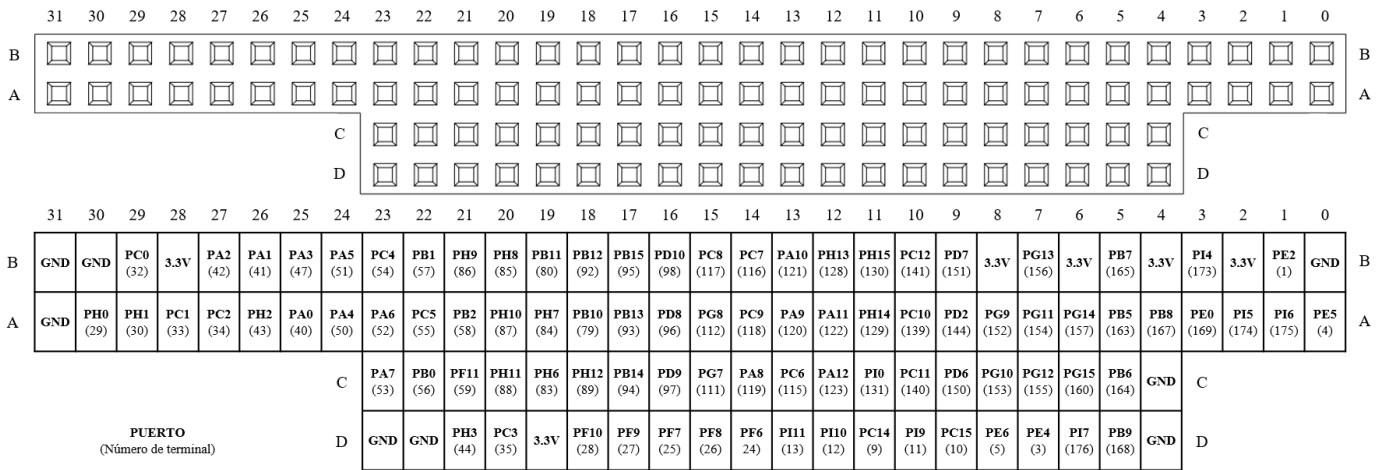


Figura 104. Asignación de terminales en el conector PC/104

## 10.4 Estimación de las necesidades cuantitativas

Como último punto antes de comenzar con la construcción del sistema, es necesario enlistar los componentes principales y realizar una estimación de los costos que requiere su construcción, presentada en la *Tabla 34*.

Tabla 34. Estimación de los costos de construcción de los componentes principales

Componente	Descripción	Cantidad	Proveedor	Costo unitario USD	Costo total USD
STM32F407IGT6	Microcontrolador principal	1	STMicro-electronics	12.87	12.87
MSP430FR5969-SP	Microcontrolador supervisor	1	TI	23.76	23.76
IS64WV5128EDBLL	Memoria de trabajo (SRAM)	2	ISSI	7.91	15.82
IS25LE01G	Memoria de almacenamiento masivo de datos (Flash)	4	ISSI	15.10	60.4
TMP461-SP	Sensor de temperatura	3	TI	15.42	46.26
ADXL313	Acelerómetro	2	Analog Devices	4.44	8.88
L3GD20HTR	Giroscopio	2	STMicro-electronics	3.25	6.50
HMC6343	Magnetómetro	2	Honeywell Aerospace	100.70	201.40
Total					375.89

Es importante aclarar que el costo total obtenido en la *Tabla 34* únicamente considera la adquisición de los componentes principales. En otras palabras, consiste en un aproximado de los costos que conlleva la construcción del SCMI, sin considerar los componentes pasivos, los costos de fabricación, los costos de los procesos de prueba y verificación y los costos del proceso de diseño.

## 11.1 Diseño asistido por computadora (CAD)

Entrando al último paso del proceso de la *Metodología de diseño*, se realiza el diseño del prototipo de la PCB en el software EAGLE, comenzando con el diagrama esquemático.

### 11.1.1 Diseño del diagrama esquemático

A continuación, se presenta el diagrama esquemático del SCMI, de acuerdo con las conexiones descritas anteriormente, dividido en cuatro páginas de esquemático: la primera página corresponde al módulo del procesador de a bordo (*Figura 105*), la segunda página a los dispositivos de almacenamiento (*Figura 106*), la tercera página a los dispositivos externos (*Figura 107*) y la cuarta página al conector PC/104, como el módulo de los puertos de entrada/salida (*Figura 108*).

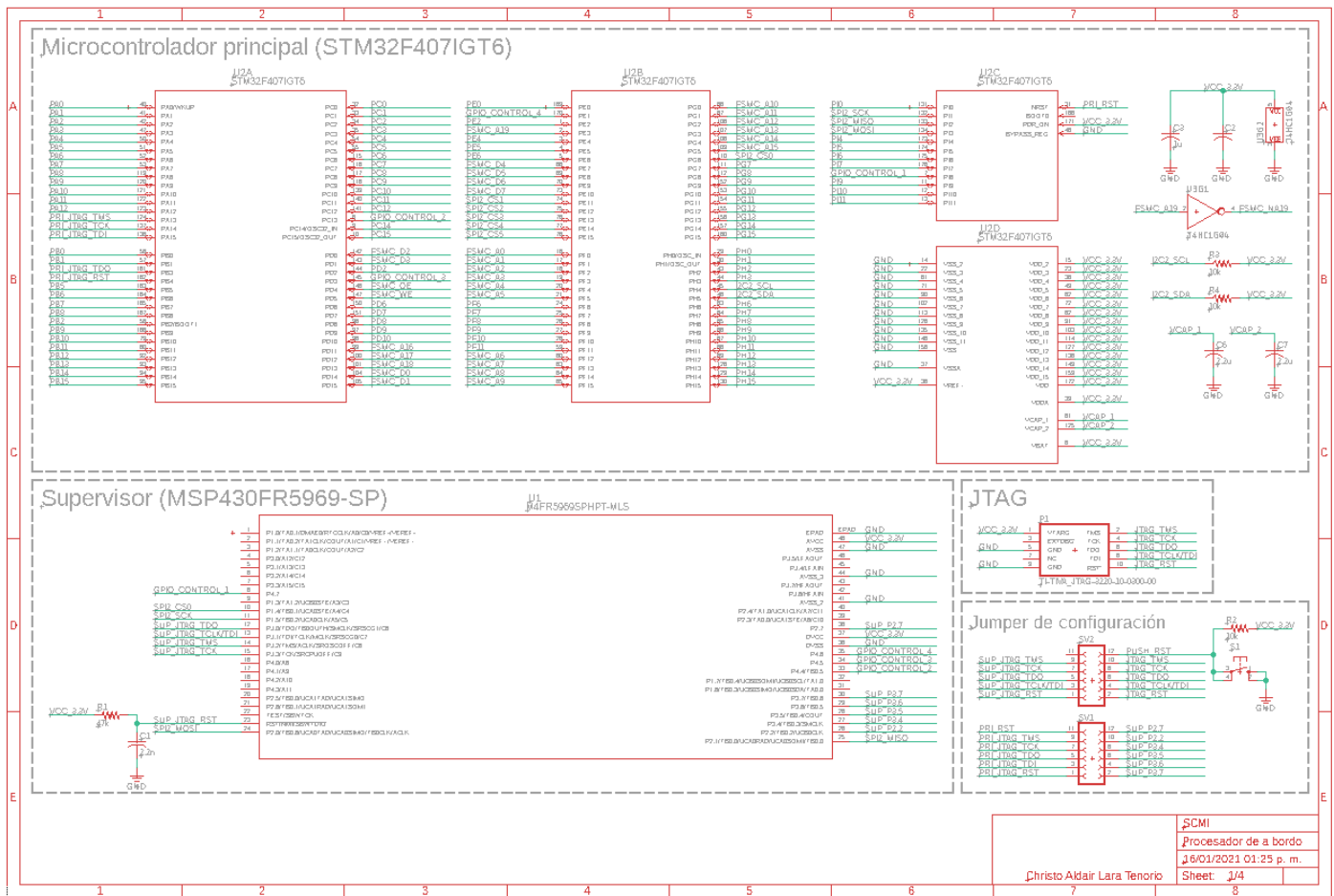


Figura 105. Diagrama esquemático del supervisor de a bordo en EAGLE

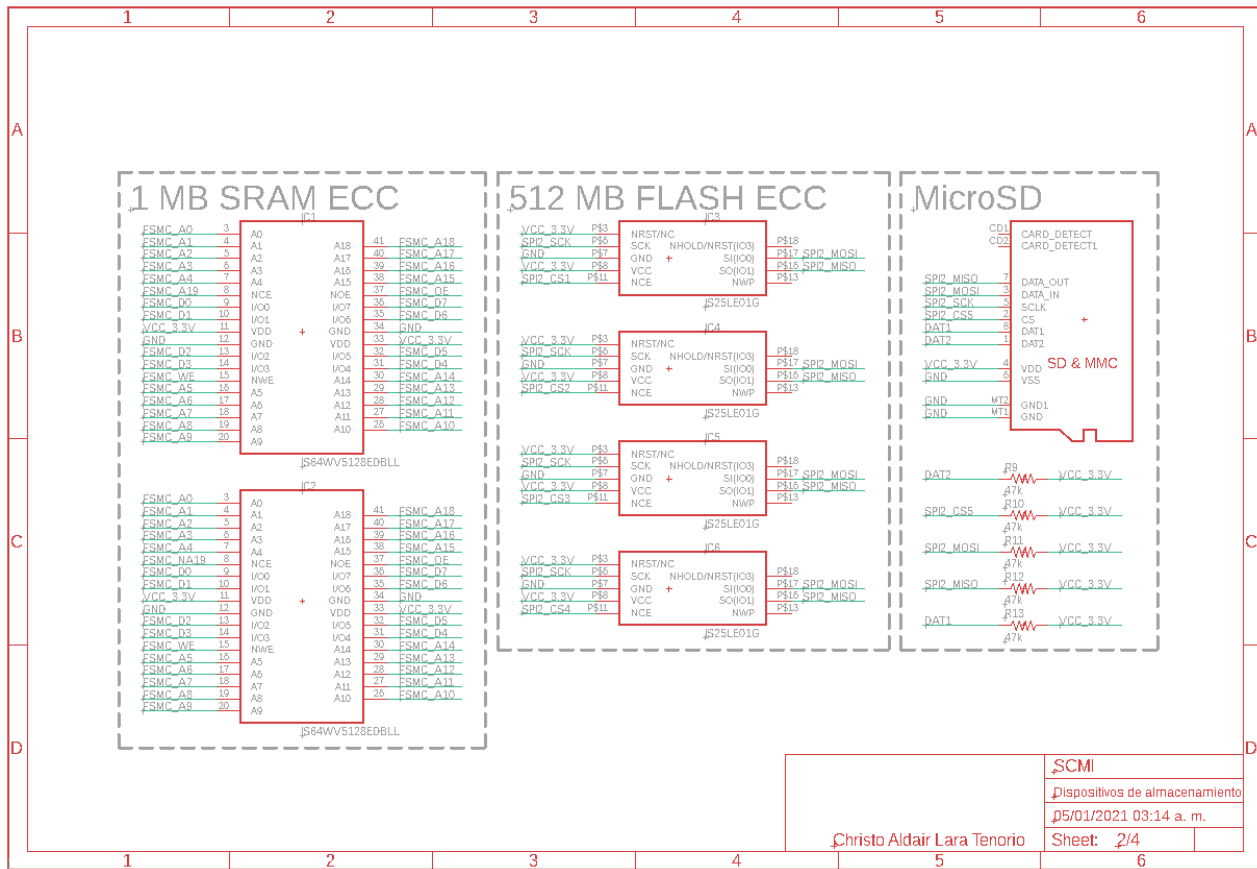


Figura 106. Diagrama esquemático de los dispositivos de almacenamiento en EAGLE

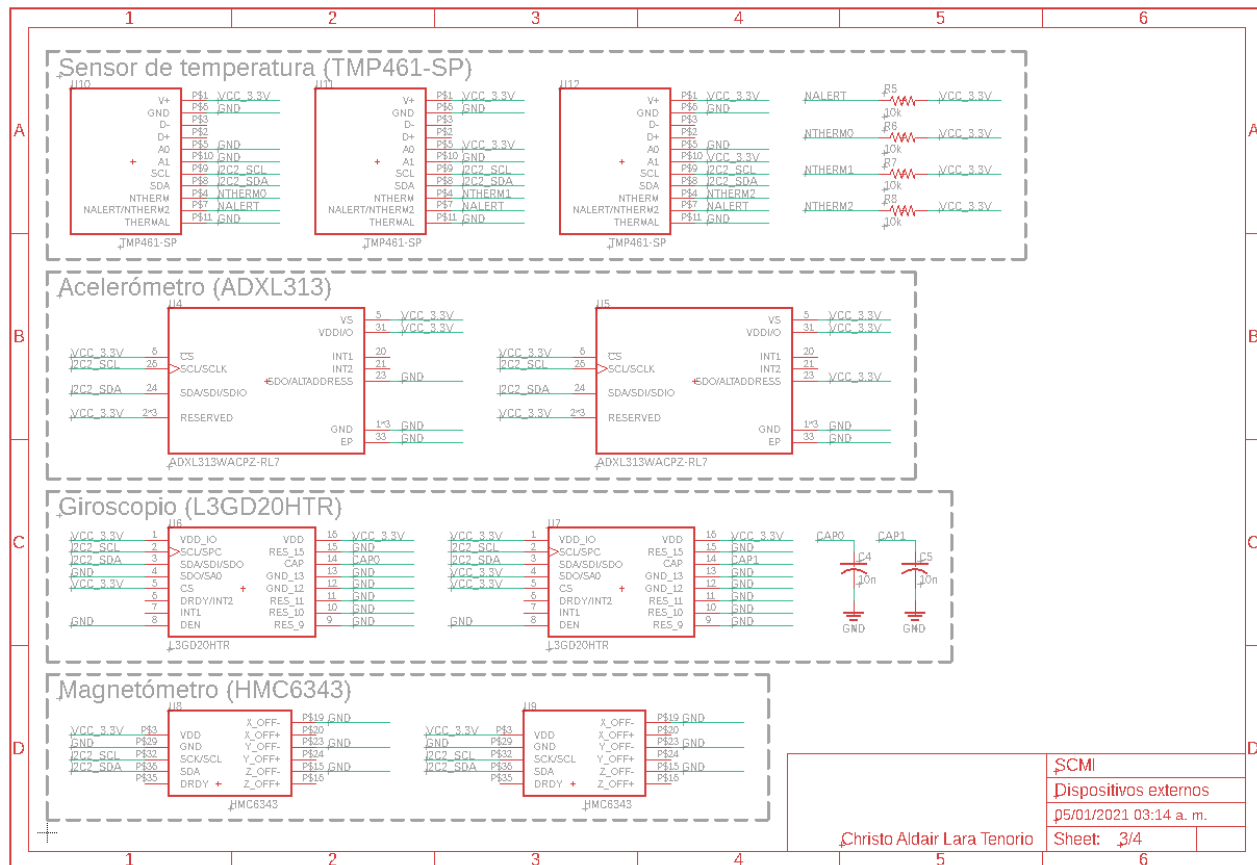


Figura 107. Diagrama esquemático de los dispositivos externos en EAGLE

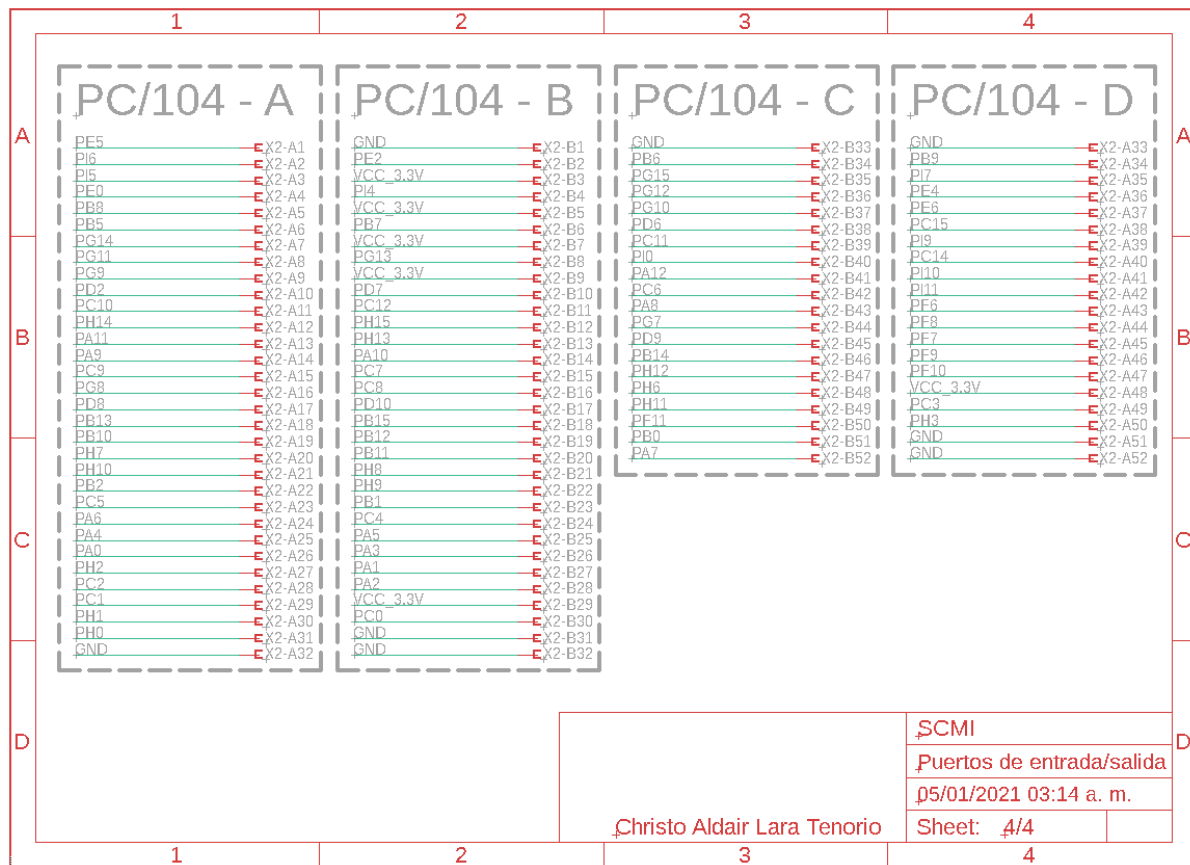


Figura 108. Diagrama esquemático de los puertos de entrada/salida en EAGLE

### 11.1.2 Diseño de la tarjeta de circuito impreso (PCB)

Una vez terminado el diagrama esquemático, se realiza el diseño físico de la PCB, obteniendo los resultados descritos a continuación.

La Figura 109 presenta el diseño completo de la PCB.

La Figura 110, Figura 111, Figura 112 y Figura 113 corresponden a una sola capa en el siguiente orden correspondiente: capa superior (top layer) en color rojo, la primera capa interna en color amarillo, la segunda capa interna en color naranja y la capa inferior (bottom layer) en color azul. Se aclara que en ambas capas externas (roja y azul) se colocó un plano de tierra (GND), mientras que las capas internas se utilizan principalmente como planos de alimentación, correspondiendo la primera (amarilla) a Vcc y la segunda (naranja) a GND.

En la Figura 114 se presenta la vista y ubicación de los componentes de ambas capas (superior e inferior).

Por último, en la Figura 115 y Figura 116 se presenta la vista de manufactura final del diseño de la PCB.



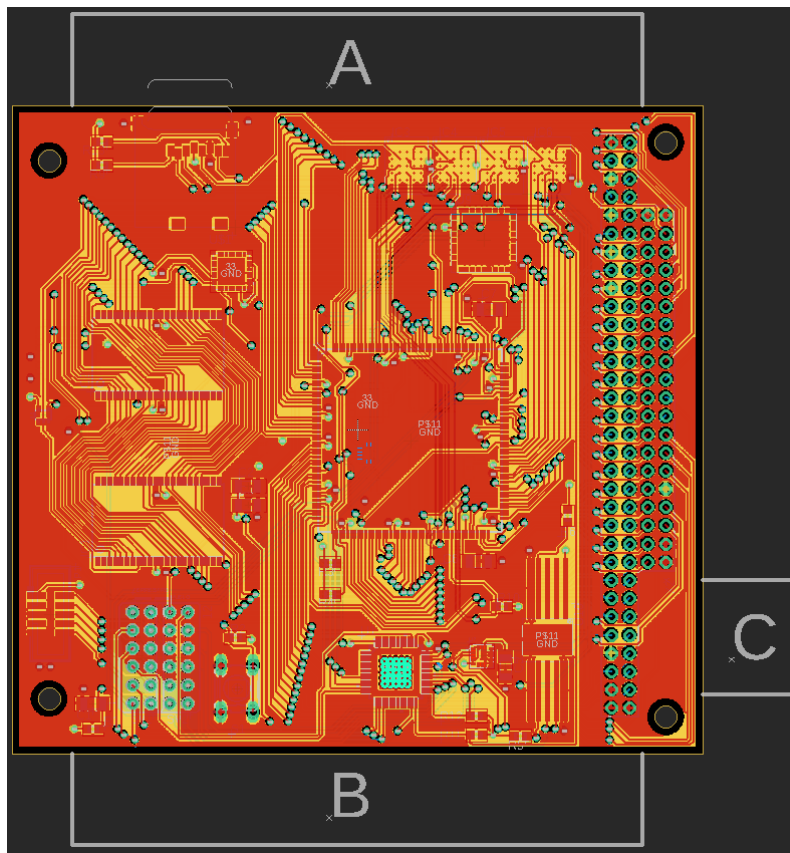


Figura 109. Diseño de la PCB del SCMI en EAGLE – Vista general

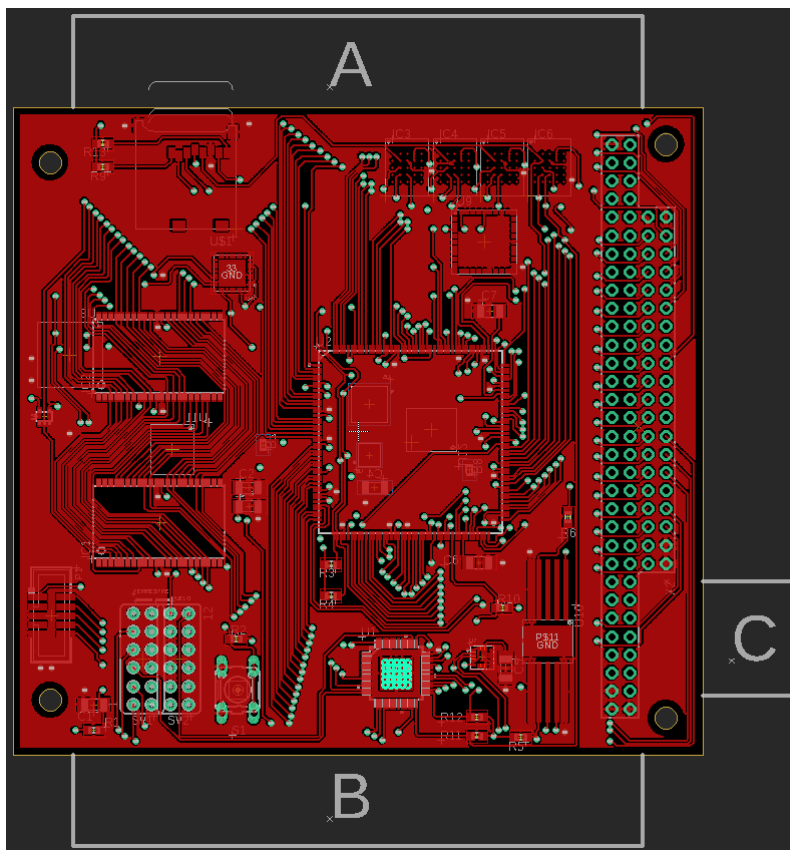


Figura 110. Diseño de la PCB del SCMI en EAGLE – Capa superior (Top layer)

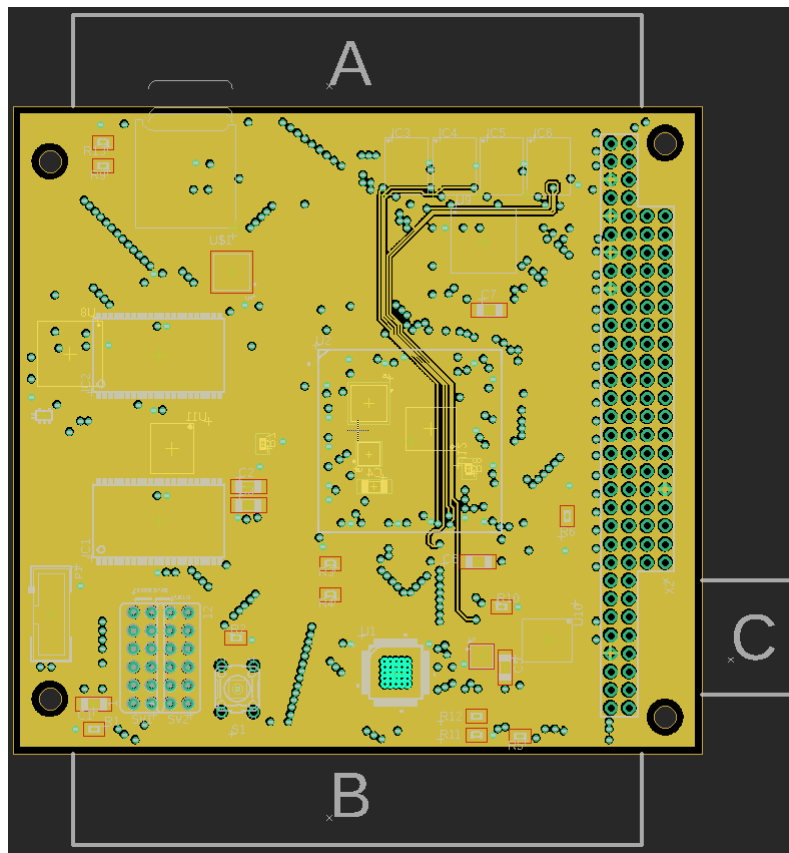


Figura 111. Diseño de la PCB del SCMI en EAGLE – Primera capa interior

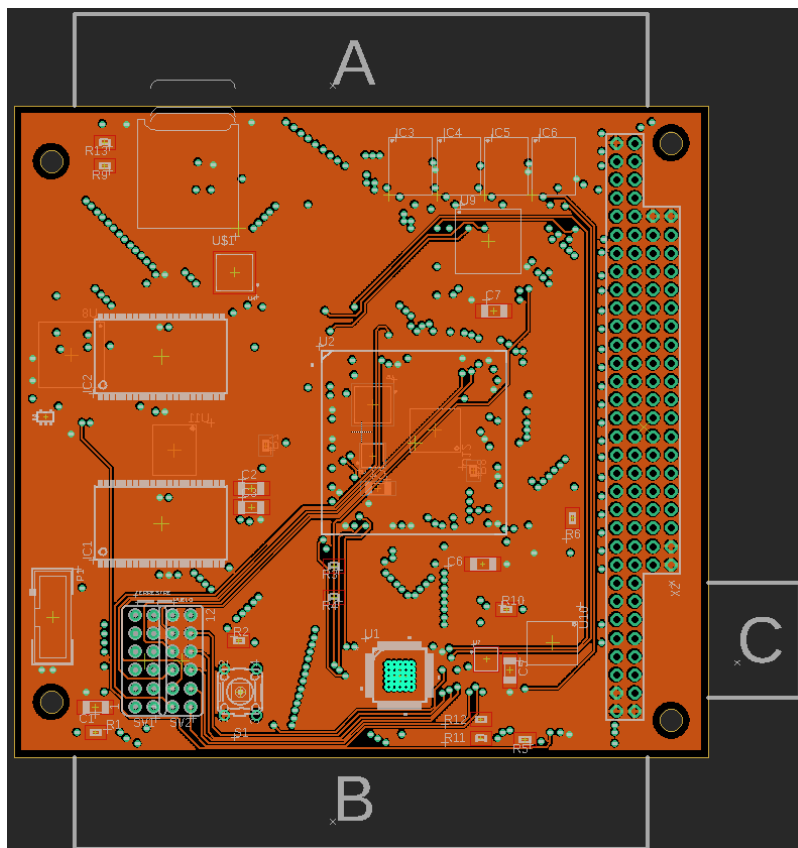


Figura 112. Diseño de la PCB del SCMI en EAGLE – Segunda capa interior

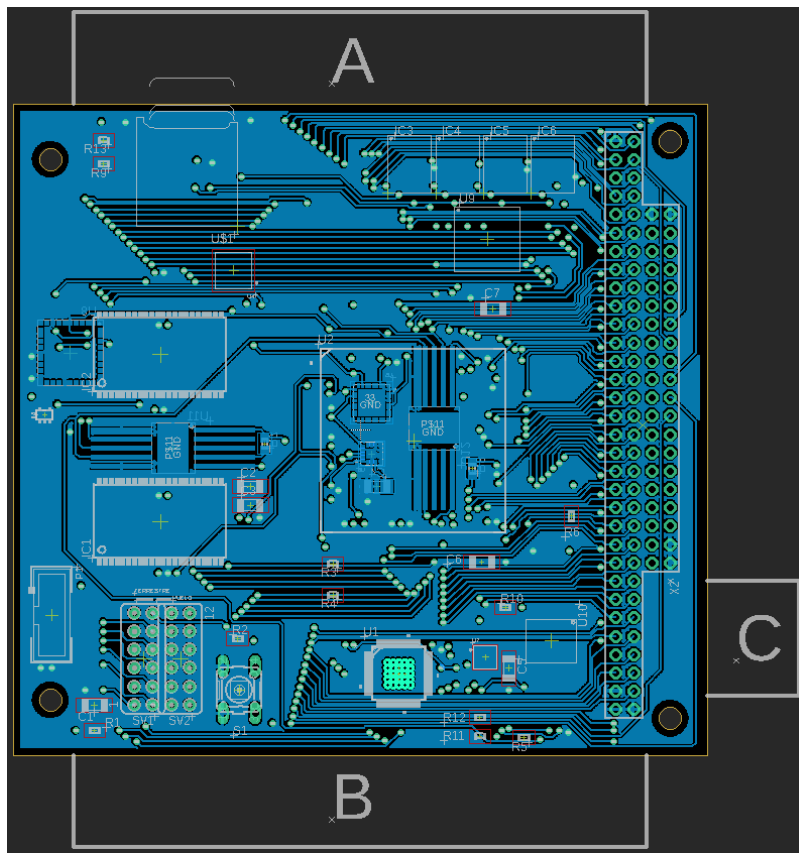


Figura 113. Diseño de la PCB del SCMI en EAGLE – Capa inferior (Bottom layer)

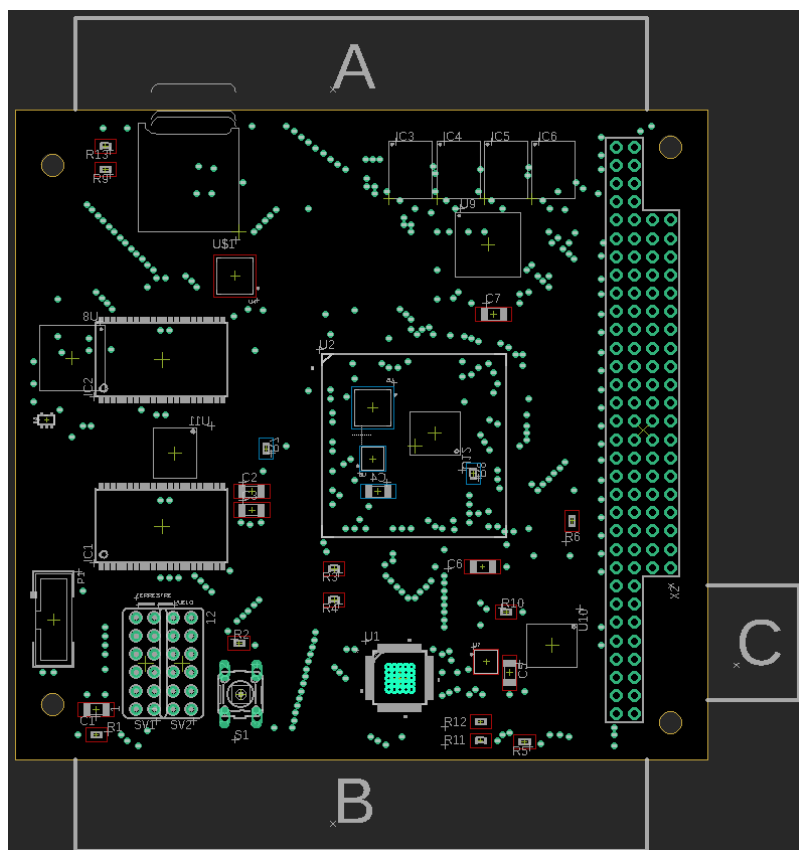


Figura 114. Diseño de la PCB del SCMI en EAGLE – Vista de componentes

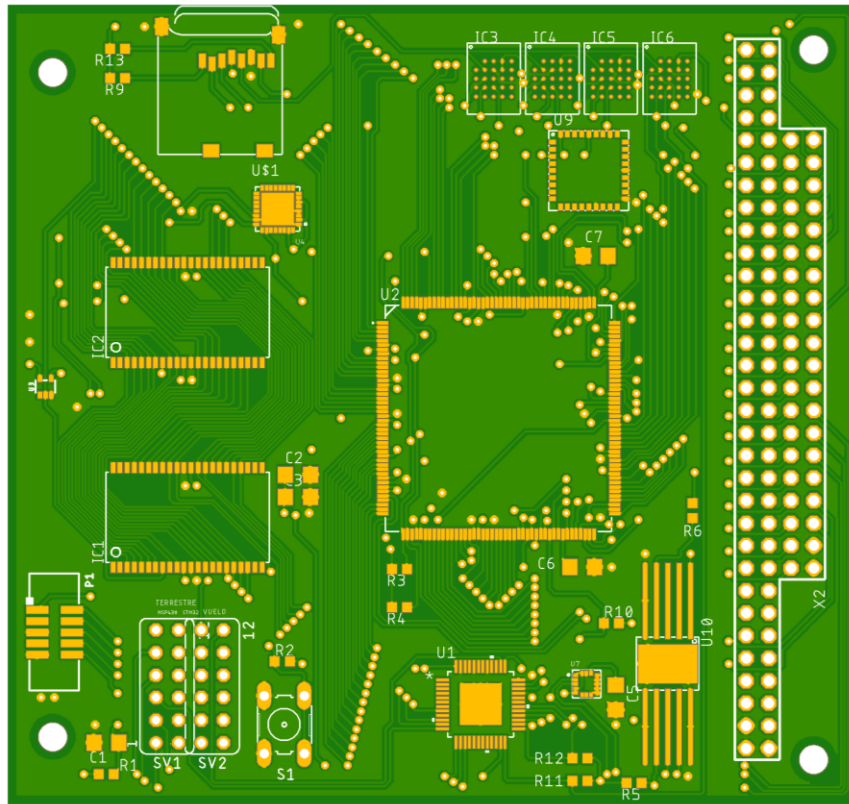


Figura 115. Diseño de la PCB del SCMI en EAGLE – Vista de manufactura (cara superior)

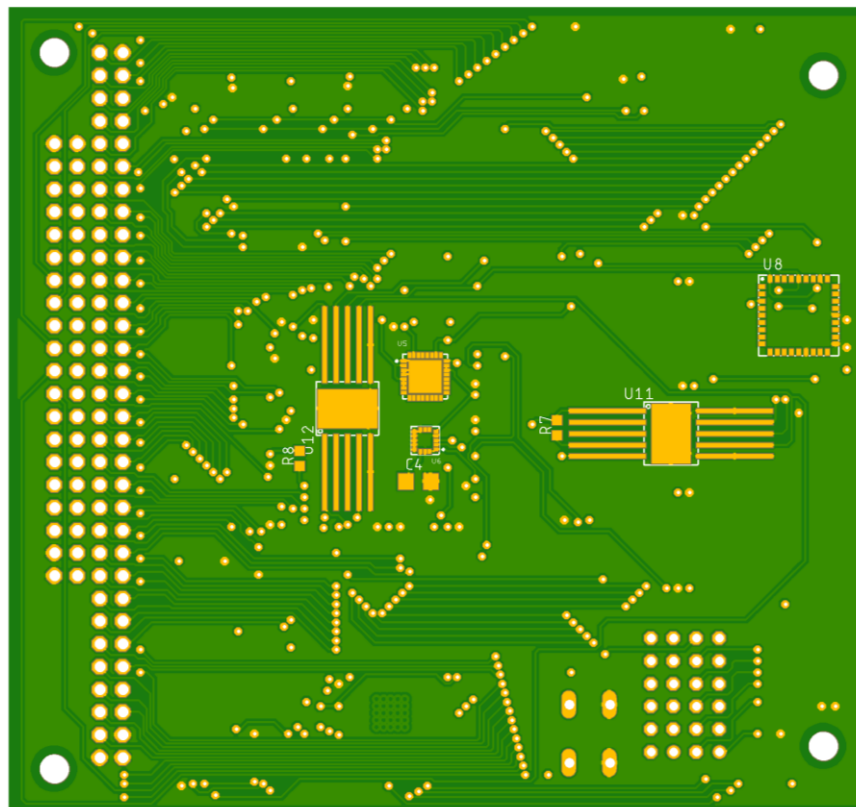


Figura 116. Diseño de la PCB del SCMI en EAGLE – Vista de manufactura (cara inferior)

## *Conclusiones*

---

En un principio, la era espacial fue motivada por fines políticos y militares, pero, con el paso de los años y los cambios históricos, diferentes naciones se fueron uniendo para impulsar la llamada exploración espacial, en donde se han detectado grandes oportunidades para obtener beneficios económicos, científicos y tecnológicos, a través del desarrollo de sistemas espaciales.

La era espacial ha supuesto grandes retos a la comunidad científica, dado que, para diseñar un sistema que desempeñe sus funciones correctamente en el espacio es necesario considerar factores que aumentan la probabilidad de que este falle, como el medio ambiente de radiación espacial, considerando que si el sistema falla, difícilmente se podrán realizar tareas de mantenimiento para recuperarlo. Considerando esto, se construían sistemas lo suficientemente grandes para obtener un sistema robusto y resistente, sin embargo, los avances tecnológicos recientes han abierto las puertas a la integración de sistemas más pequeños con niveles de confiabilidad adecuados para desempeñar misiones espaciales.

La miniaturización de satélites presenta una excelente área de oportunidad gracias a sus bajos costos y tiempos de desarrollo, en donde organizaciones, institutos y universidades de todo el mundo tienen la oportunidad de participar en proyectos satelitales. Dentro de los satélites pequeños, el CubeSat es el que presenta mayores oportunidades de desarrollo, en donde su factor de forma permite integrar y ubicar los subsistemas de manera eficiente y sencilla, facilitando las tareas de diseño, además de ser un sistema escalable en varias unidades apiladas para construir satélites de mayor tamaño.

El presente trabajo muestra la propuesta de diseño de un primer prototipo de SCMI, basado principalmente en una investigación profunda del estado del arte de los SCMI comerciales y de investigación. El diseño de este prototipo inicia el desarrollo e integración de los subsistemas necesarios para construir un satélite completo, diseñado totalmente por estudiantes de la Facultad de Ingeniería.

Al inicio de este proyecto fue necesario plantear una buena metodología, en donde destaca la importancia de conocer adecuadamente el estado del arte, a través del cual se observan los conocimientos teóricos necesarios para diseñar. Además, el estado del arte sirvió como base para la toma de decisiones, comenzando con el planteamiento más certero de los objetivos y alcances del proyecto.

El diseño de un SCMI contempla campos extensos, como el diseño del hardware y del software y la implementación de un RTOS y técnicas de mitigación y tolerancia a fallas. En este trabajo se realizó el concepto de la tecnología diseñando el hardware e implementando algunas técnicas de mitigación y tolerancia a fallas, además de proponer la estructura general del software adecuado para el sistema, basada en los resultados obtenidos de la investigación del estado del arte. De este modo, se concluye que el SCMI diseñado presenta un nivel de madurez tecnológica (TRL) 2.

## *Trabajo a futuro*

---

Como se ha mencionado anteriormente, el presente trabajo contempla la propuesta de un prototipo de SCMI para nanosatélites bajo el estándar CubeSat. Sin embargo, la idea de este proyecto surgió de la necesidad de diseñar una computadora capaz de integrarse al resto de los subsistemas de un CubeSat, con la finalidad de plasmar las bases para el desarrollo de un satélite completo. Para esto, es necesario realizar el siguiente trabajo a futuro:

- ✓ Construir físicamente el sistema, con la finalidad de realizar las pruebas necesarias para verificar su funcionamiento correcto y, en caso de ser necesario, realizar las modificaciones pertinentes.
- ✓ Diseñar el software necesario para las tareas del supervisor, contemplando el diagnóstico, reconfiguración y recuperación del estado funcional del microcontrolador principal.
- ✓ Implementar el sistema operativo FreeRTOS.
- ✓ Realizar las pruebas experimentales para alcanzar un TRL 3.
- ✓ Validar el desempeño de los componentes en un entorno de laboratorio para alcanzar un TRL 4.
- ✓ Validar el desempeño de los componentes en un entorno relevante para alcanzar un TRL 5.
- ✓ Validar el desempeño del sistema en un entorno relevante para alcanzar un TRL 6.

## Referencias

---

- [1] NASA, *Small Spacecraft Technology State of the Art*. California, 2018.
- [2] J. Wertz, D. Everett, J. Puschell, *Space Mission Engineering: The New SMAD*. California: Space Technology Library, 2011.
- [3] <https://www.nanosats.eu>
- [4] <https://www.isispace.nl/product/on-board-computer/>
- [5] <https://gomspace.com/shop/subsystems/command-and-data-handling/nanomind-a3200.aspx>
- [6] <https://www.c3s.hu/portfolio-item/on-board-computer/>
- [7] <https://www.aac-clyde.space/satellite-bits/command-data-handling>
- [8] <https://cubespace.co.za/cubecomputer>
- [9] <https://www.endurosat.com/cubesat-store/cubesat-obc/onboard-computer-obc/>
- [10] <https://www.gaussteam.com/products/onboard-computer/abacus-2/>
- [11] <https://www.gaussteam.com/products/onboard-computer/hercules/>
- [12] <https://gumush.com.tr/product/n-art-obcomms-v2/>
- [13] <http://www.imtsrl.it/obc-cubesat.html>
- [14] <https://nanoavionics.com/cubesat-components/cubesat-on-board-computer-main-bus-unit-satbus-3c2/>
- [15] <https://www.skylabs.si/>
- [16] C. M. Fuchs, N. M. Murillo, P. Chou, et al. 2019. "Fault Tolerant Nanosatellite Computing on a Budget" *Proceedings of the AIAA/USU Conference on Small Satellites*, Advanced Concepts II, SSC19-WKVIII-09. <https://digitalcommons.usu.edu/smallsat/2019/all2019/135/>
- [17] S. M. Guertin, "CubeSat and mobile processors", in *NASA Electronics Technology Workshop*, 2015, pp. 23-26.
- [18] S. Maqbool. 2005. "System-Level Mitigation of SEFIs in Data Handling Architectures, A Solution for Small Satellites" *Proceedings of the AIAA/USU Conference on Small Satellites*, 13<sup>th</sup> Annual Frank J. Redd Student Scholarship Competition. <https://digitalcommons.usu.edu/smallsat/2005/all2005/39/>
- [19] B. Rajulu, S. Dasiga and N. R. Iyer, "Open source RTOS implementation for on-board computer (OBC) in STUDSAT-2," 2014 IEEE Aerospace Conference, Big Sky, MT, 2014, pp. 1-13, doi: 10.1109/AERO.2014.6836377.
- [20] J. Bouwmeester, M. Langer, & E. Gill. "Survey on the implementation and reliability of CubeSat electrical bus interfaces". *CEAS Space Journal* 9, 163–173 (2017). doi: 10.1007/s12567-016-0138-0. <https://doi.org/10.1007/s12567-016-0138-0>
- [21] M. Tafazoli. "A study of on-orbit spacecraft failures". *Acta Astronaut.* 64(2-3), 195-205 (2009).
- [22] J. Bouwmeester, N. Santos. "Analysis of the distribution of electrical power in CubeSats". *Proceedings of the 4S Symposium*, Mallorca (2014).
- [23] M. Cho, T. Yamauchi, M. Sejera, et al. 2020. "CubeSat Electrical Interface Standardization for Faster Delivery and More Mission Success" *Proceedings of the AIAA/USU Conference on Small Satellites*, Advanced Concepts I, SSC20-WKI-03. <https://digitalcommons.usu.edu/smallsat/2020/all2020/3/>
- [24] <https://history.nasa.gov/sputnik/>
- [25] [http://www.russianspaceweb.com/sputnik\\_mission.html](http://www.russianspaceweb.com/sputnik_mission.html)
- [26] [https://www.unoosa.org/oosa/osoindex/search-ng.jsp?lf\\_id=](https://www.unoosa.org/oosa/osoindex/search-ng.jsp?lf_id=)
- [27] <https://www.unoosa.org/oosa/en/spaceobjectregister/index.html>
- [28] [https://www.nasa.gov/mission\\_pages/station/main/index.html](https://www.nasa.gov/mission_pages/station/main/index.html)
- [29] P. Fortescue, G. Swinerd & J. Stark, *Spacecraft Systems Engineering*. 2011.
- [30] <https://ssl.scripts.mit.edu/www/portfolio/spheres/>
- [31] Cal Poly, *CubeSat Design Specification (CDS)*. REV 13. 2014
- [32] J. Eickhoff, *Onboard Computers, Onboard Software and Satellite Operations*. 2011.
- [33] M. Macdonald & V. Badescu, *The International Handbook of Space Technology*. Springer, 2014.
- [34] G. Sebestyen, S. Fujikawa, N. Galassi & A. Chuchra, *Low Earth Orbit Satellite Design*. USA: Springer, 2018.
- [35] *PC/104 Specification*. Version 2.6. 2008.

- [36] [https://www.nasa.gov/directorates/heo/scan/engineering/technology/txt\\_accordion1.html](https://www.nasa.gov/directorates/heo/scan/engineering/technology/txt_accordion1.html)
- [37] J. Savage, G. Vázquez & N. E. Chávez, *Diseño de microprocesadores*. Facultad de Ingeniería, UNAM, 2015.
- [38] J. W. Valvano, *Embedded Systems: Introduction to ARM Cortex-M Microcontrollers*. Volume 1, 5<sup>th</sup> edition. 2015.
- [39] J. W. Valvano, *Embedded Systems: Real-Time Operating Systems for ARM Cortex-M Microcontrollers*. Volume 3, 4<sup>th</sup> edition. 2017.
- [40] <https://www.freertos.org/>
- [41] <https://www.kubos.com/kubos/>
- [42] <https://www.rtems.org/>
- [43] <https://www.micrium.com/>
- [44] J. L. Hennessy & D. A. Patterson, *Computer Architecture: A Quantitative Approach*. 5<sup>th</sup> edition. USA: Morgan Kaufmann, 2012.
- [45] Ball Aerospace & Technologies Corp. “*The User Interface for Command and Control of Embedded Systems*.” 2015.
- [46] H. Henkel. “*Total Dose Radiation Tests at FRAM Non-Volatile Memories*.” 1996.
- [47] I. Troxel. “*Memory Technology for Space*”. Disponible en:  
[https://nepp.nasa.gov/mapld\\_2009/talks/083109\\_Monday/06\\_Troxel\\_Ian\\_mapld09\\_pres\\_2.pdf](https://nepp.nasa.gov/mapld_2009/talks/083109_Monday/06_Troxel_Ian_mapld09_pres_2.pdf)
- [48] K. A. LaBel et al., “*Commercial microelectronics technologies for applications in the satellite radiation environment*,” 1996 IEEE Aerospace Applications Conference. Proceedings, Aspen, CO, USA, 1996, pp. 375-390 vol.1, doi: 10.1109/AERO.1996.495897.
- [49] L. E. Frenzel Jr, *Handbook of Serial Communications Interfaces: A comprehensive compendium of serial digital input/output (I/O) standards*. 2016.
- [50] J. W. Valvano, *Embedded Microcomputer Systems: Real Time Interfacing*. Volume 2, 3<sup>rd</sup> edition. 2011.
- [51] ESA, *Space product assurance: Techniques for radiation effects mitigation in ASICs and FPGAs handbook*. 2016.
- [52] J. W. Howard Jr. & D. M. Hardage, *Spacecraft Environments Interactions: Space Radiation and Its Effects on Electronic Systems*. NASA. 1999.
- [53] J. E. Mazur, “An Overview of the Space Radiation Environment”, CROSSLINK Radiation in the Space Environment, vol. 4, no. 2, pp. 10-14, Summer 2003.
- [54] J. Scarpulla & A. Yarbrough, “What Could Go Wrong? The Effects of Ionizing Radiation on Space Electronics”, CROSSLINK Radiation in the Space Environment, vol. 4, no. 2, pp. 15-19, Summer 2003.
- [55] R. D. Schrimpf & D. M. Fleetwood, *Radiation Effects and Soft Errors in Integrated Circuits and Electronic Devices*. Vanderbilt University, USA. 2004.
- [56] M. Yang, G. Hua, Y. Feng & J. Gong, *Fault-Tolerance Techniques for Spacecraft Control Computers*. WILEY, 2017.
- [57] R. Velazco, P. Fouillat & R. Reis, *Radiation Effects on Embedded Systems*. Netherlands: Springer, 2007.
- [58] N. Battezzati, L. Sterpone & M. Violante, *Reconfigurable Field Programmable Gate Arrays for Mission-Critical Applications*. Springer. 2011.
- [59] K. Iniewski, *Radiation Effects in Semiconductors*. USA: CRC Press, 2011.
- [60] B. W. Johnson, *An Introduction to the Design and Analysis of Fault-Tolerant Systems*. University of Virginia.
- [61] Von Neumann J (1956). Probabilistic logics and the synthesis of reliable organisms from unreliable components. In: Shannon C E and McCarthy J (eds). *Automata Studies*. Princeton: Princeton University Press.
- [62] M. L. Shooman, *Reliability of computer systems and networks: Fault Tolerance, Analysis, and Design*. John Wiley & Sons, 2002.
- [63] K. T. Ulrich, S. D. Eppinger, *Product Design and Development*. 5<sup>th</sup> edition. USA: McGraw-Hill, 2012.
- [64] <https://www.spnvis.oma.be/>
- [65] <https://estore.st.com/en/stm32f407igt6-cpn.html>
- [66] <https://developer.arm.com/ip-products/processors/cortex-m>



- [67] <https://www.ti.com/product/MSP430FR5969-SP>
- [68] <http://www.issi.com/WW/pdf/61-64WV5128EDBLL.pdf>
- [69] <http://www.issi.com/WW/pdf/25LE-WE01G.pdf>
- [70] <https://www.ti.com/product/TMP461-SP>
- [71] <https://www.analog.com/en/products/adx1313.html#product-samplebuy>
- [72] [https://www.st.com/content/st\\_com/en/products/mems-and-sensors/gyroscopes/l3gd20h.html](https://www.st.com/content/st_com/en/products/mems-and-sensors/gyroscopes/l3gd20h.html)
- [73] <https://www.digikey.com.mx/product-detail/es/honeywell-aerospace/HMC6343/342-1056-ND/1692480>