



Grupo de Rayos C3smicos de la Ciudad de M3xico en colaboraci3n con la Universidad de Nagoya.

Pruebas, resultados y conclusiones

En este capítulo nos enfocaremos en exponer la evaluación del prototipo final del proyecto. Para esto nos apoyaremos en los resultados obtenidos a través de nuestra experimentación con el nuevo sistema de adquisición de datos.

Antes de realizar pruebas con el sistema completo, se hicieron pruebas de forma individual a cada uno de los módulos para encontrar posibles fallas en su funcionamiento. El primer paso consistió en comprobar la operación de los módulos por medio de simulación. El HDL que utilizamos para describir los módulos fue VHDL —Very High Speed Integrated Circuit HDL— y se capturó el diseño mediante la plataforma de desarrollo Quartus II 9.1 de Altera. De igual modo, se utilizaron las herramientas de simulación, compilación y configuración de dispositivos que ofrece este software.

Después de que se obtuvieron resultados aceptables en la simulación, se evaluó el desempeño sintetizando los módulos con un PLD. Tras evaluar la capacidad necesaria para sintetizar cada módulo, se eligió un PLD con suficientes bloques lógicos para la realizar las pruebas.

El primer PLD que se ocupó fue un CPLD —PLD con capacidad de síntesis menor a la de un FPGA— de Altera modelo *EPM240T100C5*. Con este dispositivo se realizaron pruebas de los *módulos canales de neutrones y canal de presión*. Posteriormente se utilizó un CPLD *EPM570T100C5* de mayor capacidad. El objetivo fue probar en conjunto la parte de control del *módulo de comunicación I²C* y el *módulo canal de neutrones*. Asimismo se probó el *módulo de presión* con la parte de control.

Luego de verificar los módulos en el CPLD y antes de probar la comunicación con la PC, se examinó la respuesta de los tres módulos trabajando en grupo.

Como los dispositivos que se emplearon en un principio no cuentan con una capacidad de síntesis grande, para realizar las siguientes pruebas se recurrió a una tarjeta de desarrollo de Altera modelo DE1; proporcionada por el departamento de Ingeniería Electrónica de la Facultad de Ingeniería de la UNAM. Dicha tarjeta cuenta con FPGA modelo *EP2C20F484* —de capacidad de síntesis 40 veces mayor a la de un CPLD— y diversos periféricos que facilitan el diseño de muchas aplicaciones. Tiempo después se adquirió por parte del Instituto de Geofísica una tarjeta de desarrollo DE2. Esta tarjeta tiene un FPGA de mayor capacidad y con ella se realizaron todas las pruebas restantes al sistema.

Como parte final, desarrollamos el *software de adquisición* y al mismo tiempo implementamos la comunicación entre el FPGA y la computadora personal. Se comprobó el funcionamiento de todo el sistema haciendo pruebas durante varias semanas. El resultado esperado de las pruebas es la adquisición de datos de forma continua y automática. Además, se aspira a tener un error absoluto distribuido normalmente, con media cero y desviación estándar diez; tomando como referencia los datos del sistema de adquisición de datos actual.

5.1. Depuración del sistema

A continuación mencionaremos algunos problemas que se presentaron durante el desarrollo de las pruebas e indicaremos su solución.

En la figura anterior se ilustra el proceso de habilitación, copia y reestablecimiento que lleva el *módulo de control* sobre el *módulo canal de neutrones*; proceso que se explicó en el capítulo cuatro.

Problema 1

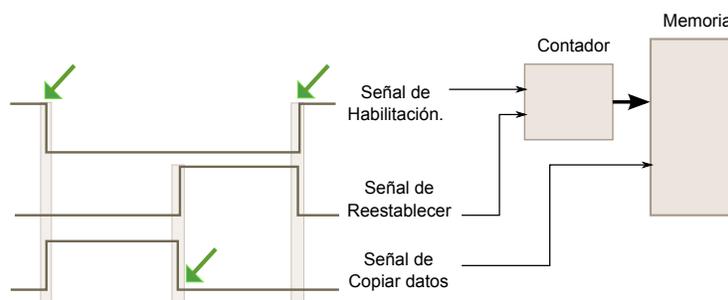


Figura 5.1: Señales que controlan la adquisición de datos.

En las regiones encerradas se observa que las señales de habilitación, copia y reestablecer cambian de valor en el mismo instante. Esto implica un problema ya que mientras se detiene el contador, el registro copia datos y mientras los datos son almacenados, el contador es reestablecido; en ambos casos los datos

están cambiando sin que se respete un tiempo para que se estabilicen.

Si los tiempos de estabilidad no se respetan, los datos registrados pueden ser incorrectos. A este fenómeno se le conoce como *Metaestabilidad*. Para resolver esta situación se agregaron pequeños periodos de tiempo entre cada cambio de señal, para dar oportunidad a que los datos se estabilicen. Este método se aplicó en todas las demás máquinas de estado que generan señales de forma secuencial.

Problema 2

En la imagen se pueden observar las señales provenientes del sensor de presión. Estas señales deben viajar por un cable de aproximadamente 8m de longitud y en consecuencia, se encuentran contaminadas con ruido y distorsionadas por la línea de transmisión.

Si deseamos procesar estas señales, primero debemos restaurarlas, ya que disminuir la longitud del cable no es una opción.

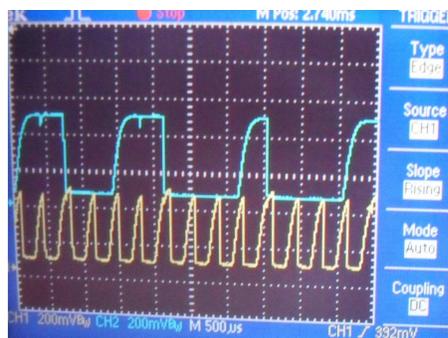


Figura 5.2: Señales del sensor de presión antes de la etapa de acondicionamiento.

Fue entonces necesario añadir la etapa de acoplamiento que se mencionó en el capítulo cuatro. El Schmitt Trigger se encarga de eliminar el ruido de las señales además de conformarlas.

Por otro lado, originalmente se hacía el cambio de nivel de 5V a 3,3V mediante un buffer colector abierto; sin embargo, este circuito deformaba en un mayor grado las señales. Por esta razón reemplazamos este circuito con un cambiador de nivel *MC14504BCP*.

Problema 3

Cuando se comenzó a trabajar con la tarjeta DE2 se tuvieron que realizar algunos cambios al diseño. El FPGA que tiene la tarjeta, además de ser de mayor capacidad, tiene un tiempo de respuesta menor. Esto lo hace más susceptible al ruido o perturbaciones que se encuentren en las señales que entran al dispositivo.

Como resultado de esta problemática se añadió a las líneas del bus I^2C un circuito Schmitt Trigger para eliminar el ruido de las señales del bus.

Problema 4

Otro módulo que requirió modificaciones es el módulo que genera las señales de reloj. En un principio las señales de reloj se generaban a partir de la *división de frecuencia* de una señal de $27MHz$. El inconveniente que este método presenta es que todas las transiciones de nivel en las señales se realizan de manera simultánea, como se puede observar en la parte superior de la siguiente figura.

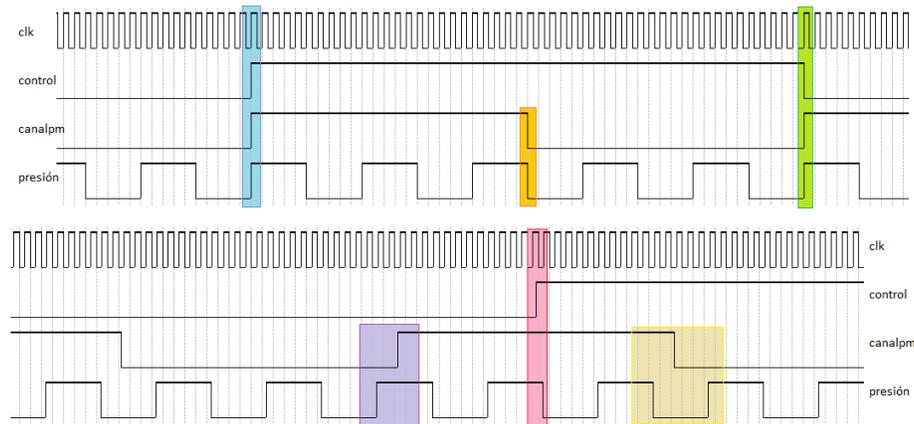


Figura 5.3: Señales de reloj que alimentan el sistema de adquisición de datos. En la parte superior se muestran las señales obtenidas por medio de un divisor de frecuencia. En la parte inferior se agregaron retardos para evitar que las señales cambien al mismo tiempo.

El problema que origina operar el sistema con estas señales es la metaestabilidad. La metaestabilidad se encuentra en la evaluación que realiza el circuito cuando sus entradas no son estables, cuando el circuito interpreta de manera incorrecta el valor de sus entradas. Esto se puede dar cuando dos circuitos se encuentran trabajando a diferentes frecuencias y transfieren información entre ellos. Si las señales que envía el circuito A al circuito B cambian de nivel al mismo tiempo que cambia la señal de reloj del circuito B, se presenta la metaestabilidad.

La solución de este problema es agregar retardos a las señales de reloj que se generan. Esto evita que las transiciones se efectuen en el mismo instante.

El circuito que genera las señales de reloj, expuesto en el capítulo anterior, cuenta con circuitos de retardo que resuelven el problema de metaestabilidad. Sus señales de salida se pueden observar en la parte inferior de la imagen anterior.

Problema 5

Cuando transferimos una señal entre dos circuitos que tienen señales de reloj independientes —no se generan a partir de la misma señal— es necesario sincronizar la señal que se transmite con la señal de reloj del receptor para poder utilizarla. Para sincronizar la señal se utiliza una cadena de registros conocida como *sincronizador*. Estos registros retardan la señal para que ésta se resuelva a un valor conocido antes de ser utilizada.

En la siguiente imagen se observan el resultado de utilizar un *sincronizador*:

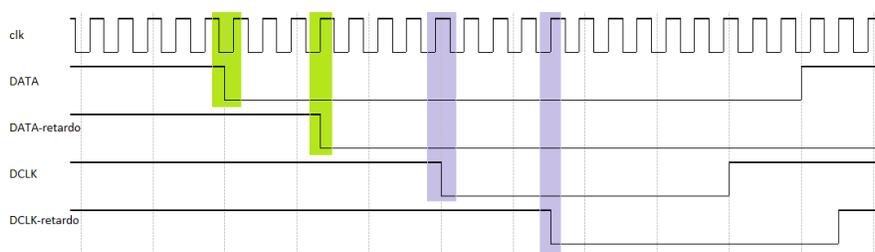


Figura 5.4: Resultado de pasar dos señales a través de una cadena de retardos. Se observa que en ambos casos la señal se sincroniza con la nueva señal de reloj.

En nuestro caso fue necesario utilizar este circuito en el *módulo de presión* para poder recibir las señales provenientes del sensor. Previo al uso de este circuito, los datos de presión adquiridos eran en ocasiones incorrectos.

Problema 6

Para desarrollar el *software de adquisición* el primer lenguaje de programación que se utilizó fue JAVA, por ser capaz de ejecutarse en cualquier plataforma que cuente con su máquina virtual. Sin embargo, JAVA nos representó un problema, debido al extinto soporte que ofrece Sun Microsystem a las API's para el manejo del puerto serie.

En JAVA se intentó desarrollar un software que enviara y recibiera información por el puerto serie. A pesar de esto, nunca se alcanzó una comunicación correcta.

En vista de tal situación se optó por buscar un lenguaje de programación que nos ofreciera soporte para nuestra aplicación y que su sintaxis fuera más sencilla que la de JAVA.

En respuesta a esto se utilizó como lenguaje de programación Python.

Como ejemplo ponemos dos fragmentos de código que realizan la misma función; uno escrito en JAVA y el otro en Python. En ambos casos se abre un

puerto serie y se configura. Se puede observar que el código hecho en Python es más sencillo y compacto que el de JAVA. Esto nos ayudó a reducir el tiempo de desarrollo del programa.

```

listapuertos=CommPortIdentifier.getPortIdentifiers();
while listapuertos.hasMoreElements() do
    puertoId = (CommPortIdentifier) listapuertos.nextElement();
    if puertoId.getPortType() == CommPortIdentifier.PORT_SERIAL
    then
        if puertoId.getName().equals(usb2c) then
            System.out.println(Conexión lista:-usb2c);
            encuentra=true;
            puertoserial=(SerialPort) puertoId.open("prueba",2000);
            salida=puertoserial.getOutputStream();
            puertoserial.setSerialPortParams(19200,DATABITS
            8,STOPBITS 2,PARITY_NONE);
        end
    end
end
end
if !encuentra then
    System.out.println("No se encontró interfaz USB-I2C");
end

```

Código en JAVA.

```

try;
    ser=serial.Serial('/dev/ttyUSB0',19200,timeout=1,stopbits=2);
except;
    print 'No se puede encontrar la interfaz USB-I2C';
    sys.exit(-1)

```

Código en Python.

Problema 7

Al realizar las pruebas finales con todo el sistema en funcionamiento, se detectaron errores en los datos de hora y fecha. El problema consistía en un *retraso* de un segundo cada 100 datos almacenados. El retraso se debía a la forma en que el *software de adquisición* llevaba el tiempo de adquisición por medio de la instrucción *time.sleep()*. Esta instrucción detiene la ejecución del programa por un periodo determinado y la reanuda al terminar este periodo —para nuestras pruebas 60s. El sistema operativo se encarga de gestionar los procesos que se ejecutan en una PC y les asigna una prioridad. Ya que *time.sleep()* no es un proceso prioritario, el tiempo que permanece detenido el programa no es siempre el mismo.

El primer paso para solucionar el problema fue evitar el uso de la instrucción

ya mencionada. En lugar de este método se utilizó `time.time()`. La instrucción `time.time()` nos devuelve el tiempo de la PC con una alta precisión. Por medio de un ciclo se puede determinar si ya transcurrió el *tiempo de adquisición*. Este proceso ayuda a mejorar la precisión con que se miden los intervalos de tiempo, sin embargo no resuelve del todo el problema del retraso. Para este caso, el retraso se presenta cada 1000 datos.

El segundo paso es detectar la ocurrencia del retraso en los datos de tiempo. Una vez detectado, se debe ajustar el *tiempo de adquisición* para compensar el retraso. De esta forma se logra eliminar la acumulación del retraso.

En la imagen se muestra una serie de datos y la ocurrencia de un retraso. También se puede observar que, en el siguiente periodo de adquisición, se logra compensar este retraso.

```

182134431 2238 14 2394 31 2494 15 2437 11 2287 20 2437 13 4596 4868 4690 7732 0
182134531 2253 12 2456 35 2512 22 2423 14 2123 18 2396 11 4678 4902 4476 7732 0
182134631 2292 12 2388 27 2507 14 2437 14 2148 14 2414 12 4642 4905 4522 7732 0
182134731 2192 11 2314 19 2385 12 2475 13 2110 17 2411 13 4483 4819 4473 7732 0
182134831 2266 15 2394 37 2519 14 2472 11 2192 29 2544 12 4631 4954 4688 7732 0
182134931 2126 14 2336 27 2354 16 2380 22 2337 27 2454 17 4440 4683 4734 7732 0
182135031 2300 12 2453 41 2487 19 2500 17 2141 20 2405 9 4719 4947 4503 7733 0
182135131 2180 8 2442 24 2569 12 2525 15 2324 12 2460 19 4588 5038 4741 7733 0
182135231 2258 9 2477 37 2410 13 2492 14 2057 21 2265 14 4699 4857 4285 7733 0
182135331 2188 16 2407 25 2555 15 2497 12 2271 24 2550 14 4565 5007 4769 7733 0
182135431 2132 12 2349 21 2381 18 2427 9 2181 22 2388 13 4442 4759 4518 7733 0
182135531 2396 15 2384 36 2524 13 2380 13 2157 19 2421 26 4738 4865 4532 7733 0
182135632 2173 15 2319 28 2520 10 2473 19 2205 19 2389 17 4460 4958 4557 7733 1
182135731 2153 12 2282 17 2389 8 2362 17 1974 20 2312 13 4404 4704 4243 7734 0
182135831 2322 8 2435 35 2503 17 2486 10 2189 18 2427 13 4715 4929 4564 7734 0
182135931 2271 7 2436 35 2424 14 2479 16 2170 15 2455 10 4670 4852 4571 7734 0
182140031 2223 11 2438 37 2467 14 2334 8 2151 23 2376 15 4631 4765 4473 7734 0
182140131 2128 14 2334 28 2421 6 2354 14 2299 29 2389 13 4443 4730 4638 7734 0
182140231 2207 12 2465 38 2515 13 2442 17 2219 25 2404 18 4655 4917 4588 7734 0
182140331 2268 10 2433 34 2378 15 2327 9 2142 17 2405 9 4676 4671 4504 7734 0

```

Figura 5.5: En el recuadro rojo se muestra un conjunto de datos que tienen un atraso de 1s. En la parte izquierda de la imagen se pueden ver los datos de fecha y hora. También se observa que al ocurrir el retraso, en el siguiente periodo éste se compensa.

5.2. Resultados

El prototipo final del sistema se constituye de las siguientes unidades:

- Seis *módulos canal de neutrones* para cada uno de los contadores proporcionales del monitor NM64: A1, A2, B1, B2, C1 y C2.
- Un *módulo canal de presión*.
- Tres módulos adicionales que llevan el conteo de la suma lógica entre cada pareja de contadores: $A1 + A2$, $B1 + B2$ y $C1 + C2$.
- Además de los módulos necesarios para la comunicación con la PC, sincronización del sistema y envío de datos.

Estas unidades se sintetizaron en el FPGA y tan sólo ocupan el 25% de la capacidad de éste.

Se realizaron pruebas de todo el sistema funcionando durante varios días para garantizar que el sistema podía operar de forma ininterrumpida y automática. El principal problema que se encontró en este punto fue con respecto a la sincronía en la recepción y envío de datos. Para solucionar esto, tuvimos que ajustar el diseño del *módulo de comunicación I²C* hasta llegar al diseño que se mostró en el capítulo anterior. Lograr superar este paso requirió de aproximadamente tres meses.

Una vez que conseguimos que el sistema funcionará por varios días, se puso como objetivo realizar una prueba de dos semanas y comparar los datos entre ambos sistemas de adquisición de datos; el adquisidor actual y el nuevo sistema de adquisición. Para efectuar correctamente esta prueba se sincronizaron ambos sistemas.

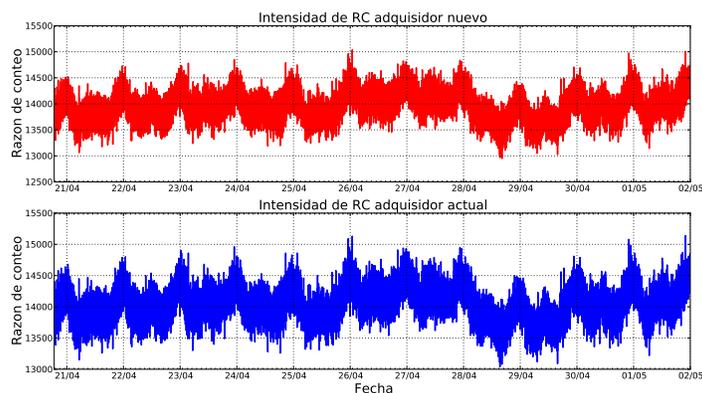


Figura 5.6: Intensidad de RC registrada por ambos sistemas de adquisición.

En la figura 5.6 se muestran los datos registrados por ambos sistemas de adquisición en el periodo del 4 de abril al 2 de mayo de 2011. Para estimar la intensidad de la radiación cósmica se sumaron las cuentas por minuto de todos los contadores proporcionales. Posteriormente se deben corregir los datos por presión atmosférica. En nuestro caso este ajuste no es necesario ya que ambos sistemas se encuentran sometidos a la mismas condiciones climáticas.

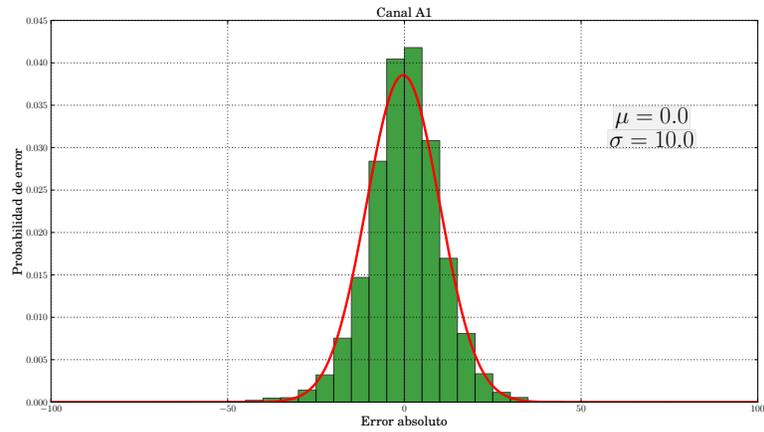
A simple vista podemos observar que las dos gráficas siguen la misma tendencia. Sin embargo, para obtener una comparación más acertada, obtendremos el error absoluto $—V_{SDAQnuevo} - V_{SDAQactual}—$ entre ambas mediciones. Como se tiene una gran cantidad de datos y todos distintos, consideraremos para nuestro análisis la distribución de probabilidad del error absoluto. Esta distribución se espera sea del tipo *normal* con μ igual a cero —error absoluto mínimo— y σ igual a diez. La razón de que se espere una desviación estándar de diez se debe a que los datos del sistema actual están escalados entre diez.

A continuación se muestran las distribuciones de *probabilidad de error* de cada uno de los seis detectores del NM64. La línea roja representa la distribución normal teórica. Ésta se genera con los valores de media y desviación estándar obtenidos para cada canal. Las barras verdes representan la distribución real, para cada uno de los canales.

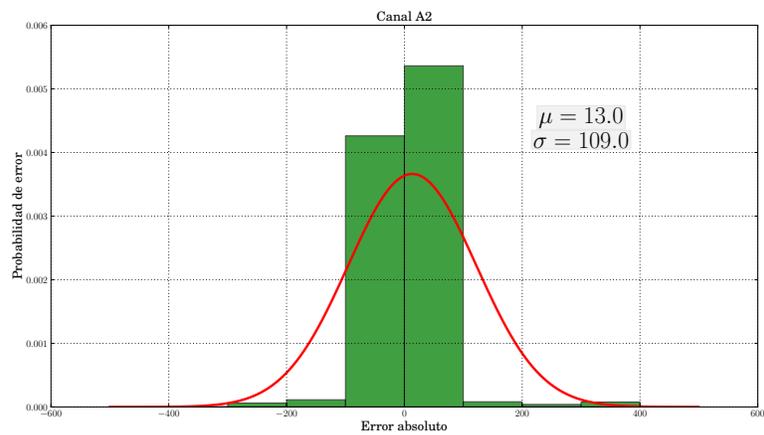
Se puede observar en las figuras, que las distribuciones de los canales A1, B2 y C1 corresponden a lo esperado, con lo que se infiere que no existe ningún problema con los datos.

Con respecto al canal C2, se observa un comportamiento del tipo *normal*; sin embargo su media está desplazada hacia la derecha. Esto puede solucionarse ajustando el nivel de discriminación en la etapa de acondicionamiento del canal en cuestión, en otras palabras, *calibrarlo*.

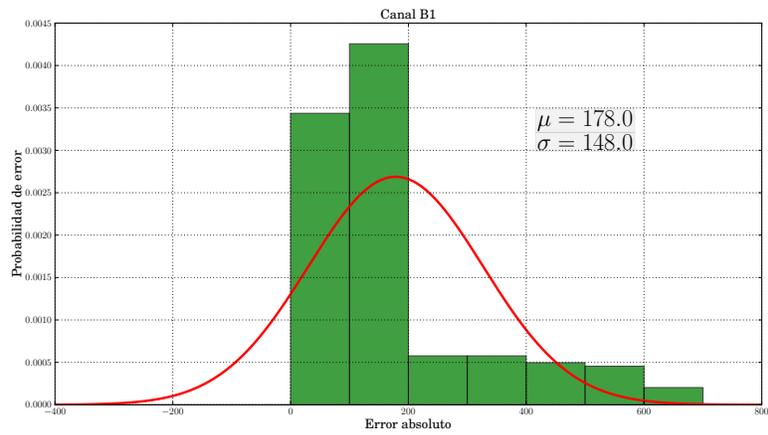
Sobresalen de entre las gráficas las de los canales A2 y B1, las cuales no tienen una distribución normal y sus parámetros μ y σ son de valores muy diferentes a los esperados. Para determinar el origen de este problema realizaremos otro tipo de análisis estadístico.



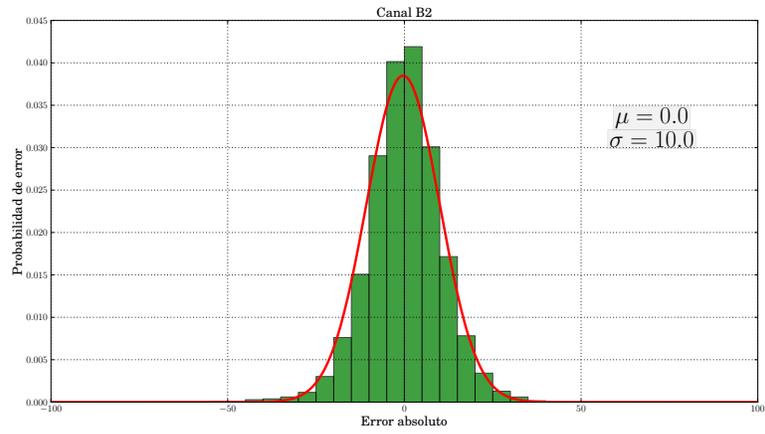
(a) Distribución canal A1



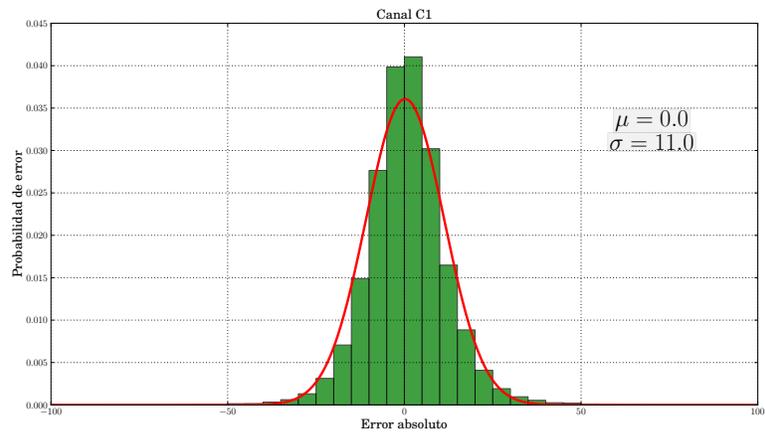
(b) Distribución canal A2



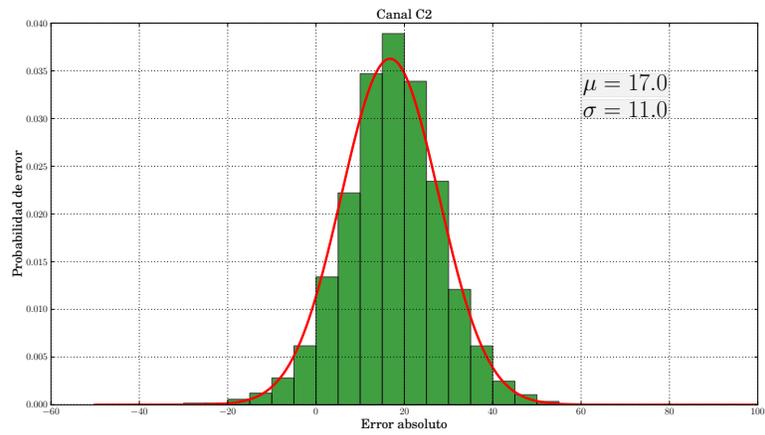
(c) Distribución canal B1



(a) Distribución canal B2



(b) Distribución canal C1

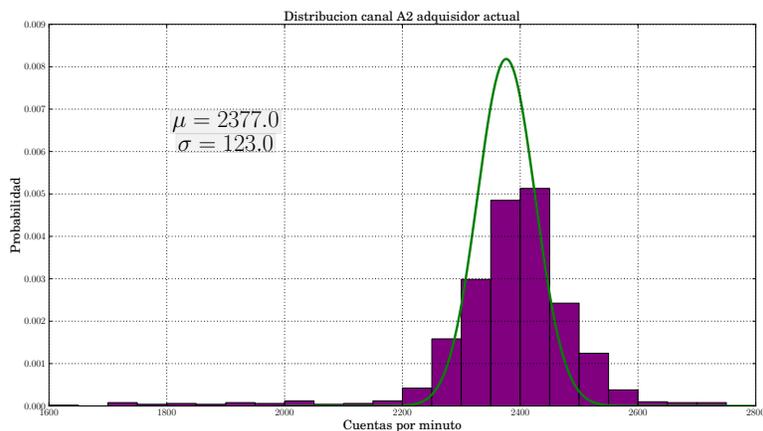


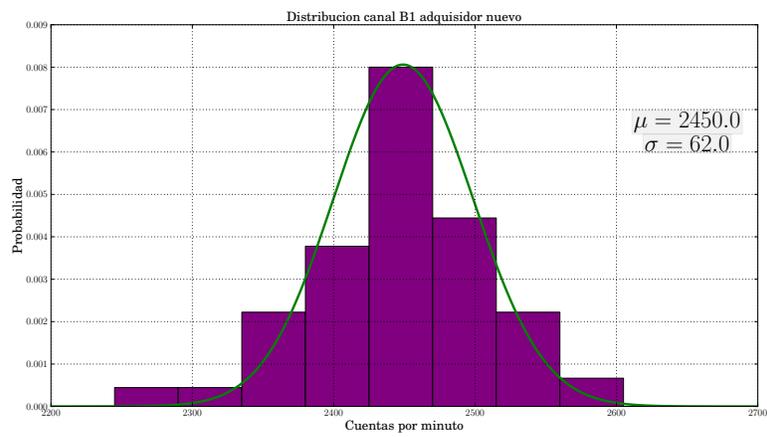
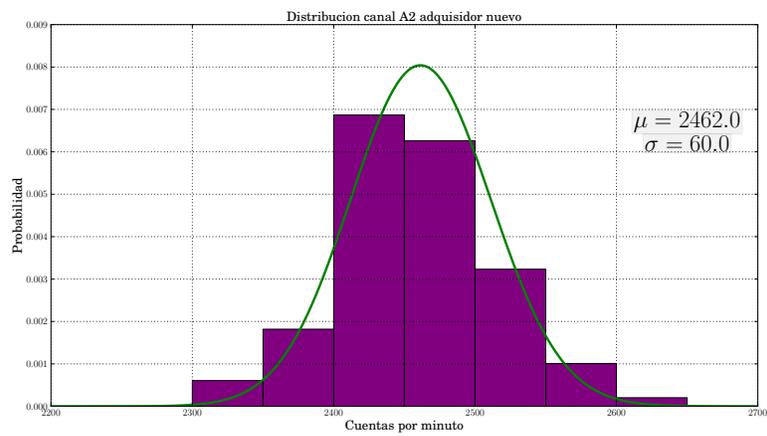
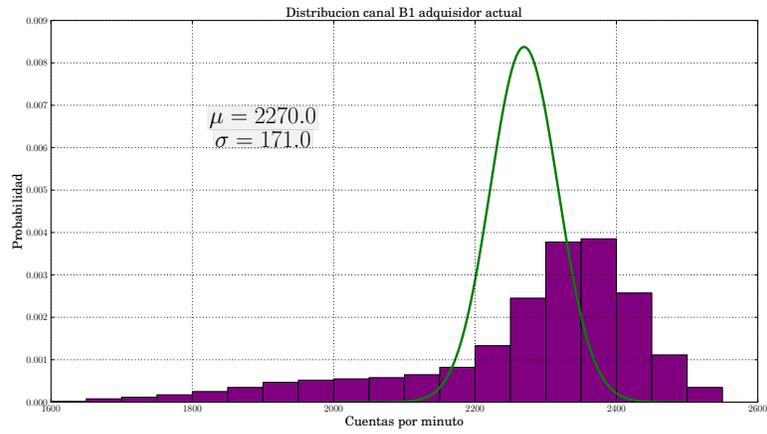
(c) Distribución canal B2

La *distribución de Poisson* es la *distribución de probabilidad* que expresa la *probabilidad de ocurrencia* de un cierto número de eventos en un intervalo de tiempo. En un sistema de conteo de partículas, se espera que la distribución de probabilidad de las cuentas se ajuste a una *poissoniana*. Tomando esto en cuenta, comprobaremos si los canales A2 y B1 de ambos sistemas se comportan de dicha forma.

En las siguientes gráficas se puede ver que existe una desviación en la distribución de las cuentas del equipo actual con respecto a la *distribución de Poisson* teórica. La línea verde representa la distribución teórica, mientras que las barras moradas, la distribución real de las cuentas. Entre los factores que pueden ocasionar que se alteren las cuentas del sistema, se encuentran: ruido eléctrico, humedad en el ambiente o desgaste en las tarjetas de circuito impreso, entre otras.

Por otro lado, las distribuciones del nuevo adquisidor se ajustan de manera satisfactoria al modelo poissoniano.





En conclusión, se infiere que el problema no se encuentra en los detectores, ni en el sistema nuevo, sino que proviene del sistema de adquisición de datos actual. Con esto se comprueba que es necesario el cambio hacia el nuevo sistema de adquisición de datos.

Cabe señalar que los datos de las cuentas individuales no son publicados por el observatorio; estos datos son útiles para estudios que se realizan internamente en la estación. Los datos publicados son la suma total de las tres secciones del monitor: $A1 + A2$, $B1 + B2$ y $C1 + C2$. Este proceso es independiente del proceso que lleva las cuentas individuales.

5.3. Conclusiones

Tras haber realizado todas las pruebas y analizado los datos arrojados por el sistema; concluimos que los resultados obtenidos son satisfactorios. Retomando los objetivos que planteamos al principio de esta tesis, podemos decir que se cumplió el objetivo principal de diseñar e implementar un sistema de adquisición de datos que cubra las necesidades del observatorio. En otras palabras, el nuevo sistema de adquisición de datos es un sistema orientado a la detección de radiación cósmica y en específico para el monitor NM64. El sistema se compone de seis canales para cada uno de los contadores proporcionales, tres canales para cada sección del monitor y un canal para el sensor MeteolaborAG.

Además, al basar nuestro diseño en tecnología moderna, se consiguieron superar limitaciones técnicas que afectan al sistema actual. Dentro de estas mejoras, el ajuste del tiempo de adquisición por medio del servidor *ntp*, contribuye en gran manera a la operación automática del sistema y mejora la precisión con la que se obtienen los datos.

De igual forma, el realizar el ajuste de tiempo mediante software contribuyó a incorporar el uso de la computadora en otras tareas, tales como: el control de adquisición, el almacenamiento de datos, el despliegue de información y el establecimiento de herramientas de diagnóstico. Todo esto se logra sin la necesidad de software comercial o tarjetas electrónicas especializadas y sin demandar una gran cantidad de recursos a la PC.

Por otra parte, se añadió al equipo un nuevo módulo de adquisición que discrimina entre dos tipos de pulsos. La información que arroja este nuevo canal permitirá caracterizar la multiplicidad del detector y de esta forma estimar con mayor precisión la intensidad de la radiación.

Finalmente, al haber utilizado un dispositivo lógico programable en el diseño, se concluye que es factible añadir nuevas funciones al sistema de adquisición de datos cuando este lo requiera. En un futuro, el Observatorio pretende incorporar mayor número de detectores y posiblemente nuevas funciones que requieran agregar más canales de adquisición. El FPGA que utilizó en el sistema, tiene

capacidad de albergar hasta cincuenta canales de adquisición.

A través de las pruebas que se realizaron, hemos podido constatar que el sistema puede operar de forma automática e ininterrumpida por un largo periodo de tiempo. Además, como resultado de este trabajo de investigación se ha generado la documentación necesaria para el mantenimiento y operación del sistema. Ambos factores, nos indican que el sistema está listo para su construcción.

Como parte final, mencionaremos algunas mejoras que se pueden hacer al sistema y temas de investigación que podrían desprenderse a partir de esta.

5.3.1. ¿Qué falta por hacer?

Agregar canales de muones. El nuevo adquisidor de datos captura datos del monitor de neutrones NM64; sin embargo el Observatorio de Rayos Cósmicos de la ciudad de México ubicado en Ciudad Universitaria cuenta con un telescopio de Muones. El nuevo equipo no cubre por el momento la adquisición de datos de este telescopio, no obstante que la capacidad del sistema soporta la incorporación de estos detectores.

Agregar sensor de temperatura. Como se mencionó en la sección 1.2, los rayos cósmicos se ven afectados por ciertas manifestaciones atmosféricas; entre ellas la presión y la temperatura. Añadir un sensor de temperatura al sistema de adquisición de datos nos brindaría mayor información sobre dichos efectos.

Software amigable. Aunque el software de adquisición no es complicado para su operación, ya que sólo necesita de algunas instrucciones —comandos—, se entiende por software amigable aquel cuya interacción con el usuario es de uso fácil, recurriendo a: gráficos intuitivos, guías, punteros, etc.

Tiempo de adquisición automático. El tiempo de captura de datos del nuevo sistema es de 60s y puede ser modificado según las necesidades del operador. Contar con un ajuste de tiempo automático es útil cuando se presenta un evento solar y se incrementa el número de cuentas del monitor. Si el sistema de adquisición logra detectar este incremento, puede disminuir el tiempo de adquisición para recoger un mayor número de datos y tener un mejor seguimiento del fenómeno.

Sensar fuentes de voltaje. La experiencia en el mantenimiento del equipo trabajando en el Observatorio de Radiación Cósmica en Ciudad Universitaria indica que la mayoría de las fallas provienen de las fuentes de alto voltaje que suministran energía a los contadores proporcionales. Sensar las fuentes ayudaría a prevenir posibles fallas.

Información sobre la energía que deposita cada partícula. Actualmente no se mide la cantidad de energía que cada partícula deposita en el monitor. El diseño e implementación de un *super módulo* encargado de registrar

esta información y procesarla, impulsaría nuevos estudios, además de reforzar la investigación que actualmente se desarrolla en el Observatorio.

Información sobre la trayectoria de arribo de cada partícula. Como se vio en secciones anteriores, el monitor NM64 tiene algunas limitaciones, una de estas es que no proporciona de manera inherente la trayectoria de las partículas. Sin embargo, se puede diseñar un *super módulo* que capte las señales de cada uno de los contadores proporcionales y por medio de alguna descripción lógica identifique cada partícula y la rastree durante su paso en el monitor. Esta información es sumamente importante para el estudio de la radiación cósmica.