



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Diseño y construcción de un robot
didáctico para experiencias en el
área de ciencias básicas**

TESIS

Que para obtener el título de
Ingeniero Eléctrico - Electrónico

P R E S E N T A

Eduardo Hernández Romero

DIRECTOR DE TESIS

Ing. Daniel Martínez Gutiérrez



Ciudad Universitaria, Cd. Mx., 2016

AGRADECIMIENTOS

A mi padre:

Por ser el pilar fundamental en mi vida, y por su apoyo incondicional en todo momento.

A mi madre:

Por ser mi guía espiritual.

A los técnicos en fabricación:

Blas Sánchez y Jorge Hernández, no solo por su ayuda en el trazado de planos, fabricación de piezas, confianza y paciencia al enseñarme el uso de herramientas y maquinaria de fabricación; sino también por las pláticas relacionadas con temas de vida y los momentos de convivencia.

Al ingeniero:

Eduardo Bernal, por su asesoría y consejos en la fabricación de las piezas mecánicas.

A mi amigo y compañero:

Sergio Coronado, por escucharme, aconsejarme y apoyarme para durante todo el proceso de elaboración de esta tesis.

A mis profesores y compañeros:

Que en algún punto del proceso de la tesis, me brindaron consejo y orientación.

CONTENIDO	Pág.
INTRODUCCIÓN	1
1. MARCO TEÓRICO	
1.1. Clasificación de robots según la ISO	3
1.2. Sistemas embebidos basados en microcontroladores	14
1.3. Control y programación de un robot	19
2. DESARROLLO	
2.1. Diseño conceptual del robot	27
2.2. Selección de componentes para el prototipo del robot	30
2.3. Elementos mecánicos	58
2.4. Maquinado y construcción de los elementos mecánicos	63
2.5. Elementos de programación	64
3. IMPLEMENTACIÓN	
3.1. Integración de los componentes electrónicos y mecánicos	68
3.2. Programación de la interfaz gráfica de usuario	85
4. VERIFICACIÓN DE OPERACIÓN	
4.1. Pruebas de comunicación y programación	94
4.2. Comprobación de las funciones del robot	98
5. CONCLUSIONES Y TRABAJO A FUTURO	107
REFERENCIAS	108

ÍNDICE DE FIGURAS	Pág.
<i>1. Robots con alimentación DC y AC</i>	1
<i>2. Robot neumático para fabricación y empaquetado</i>	5
<i>3. Geometrías de los espacios de trabajo, formados por los robots de la Clasificación</i>	7
<i>4. Fotografías de robots que forman algunos de los espacios de trabajo</i>	8
<i>5. Ejemplos de la clasificación basada en los medios de transmisión</i>	9
<i>6. Ejemplos de la clasificación ISO, para los robots de servicio</i>	12
<i>7. Robot quirúrgico DaVinci</i>	12
<i>8. Aplicaciones de los sistemas embebidos</i>	14
<i>9. Componentes de un microcontrolador</i>	15
<i>10. Diagrama a bloques de una placa de desarrollo basada en microcontrolador</i>	16
<i>11. Esquema simplificado de programación textual de un robot industria</i>	18
<i>12. Proceso y diferentes procesos de programación</i>	20
<i>13. Capas del Modelo OSI</i>	21
<i>14. Descripción general de las capas y el modelo OSI en operación</i>	24
<i>15. Tabla comparativa de los diferentes medios de comunicación inalámbrica de la IEEE 802</i>	25
<i>16. Ejemplo de un seguidor de línea, su operación y programación vía bluetooth</i>	28
<i>17. Esquema general de los elementos del robot prototipo</i>	29
<i>18. Sistema de desarrollo Raspberry Pi</i>	30
<i>19 Sistema de desarrollo UDOO</i>	31
<i>20. Sistema de desarrollo para el microcontrolador MPS430</i>	31
<i>21. Sistema de desarrollo Arduino Nano para microcontrolador ATMEGA 328</i>	32
<i>22. Shield Xbee Arduino Bluetooth</i>	35
<i>23. Modulo bluetooth HC-05</i>	36
<i>24. Modulo Xbee de radio frecuencia para Arduino</i>	37
<i>25. Diferentes módulos de radio frecuencia, tanto receptor como transmisor</i>	39
<i>26. Modulo universal para radio frecuencia APC-220</i>	40
<i>27. Módulo universal para radio frecuencia APC-220</i>	41
<i>28. Utilización de los módulos</i>	43

<i>29. Aditamentos necesarios, para el uso coordinado con PC, de los módulos de radio frecuencia</i>	44
<i>30. Librería del módulo APC-220 para LabVIEW</i>	45
<i>31. Forma de funcionamiento de los sensores ultrasónicos</i>	45
<i>32. Secuencia de pulsos ultrasónicos para calcular la distancia</i>	46
<i>33. Sensores ultrasónicos para la medición de distancia en robótica</i>	47
<i>34. Acelerómetro MMA7361L</i>	48
<i>35. Esquema genérico de un servomotor</i>	49
<i>36. Control de un servomotor, por medio de modulación de ancho de pulso</i>	51
<i>37. Micro servomotores para el desplazamiento del brazo</i>	51
<i>38. Esquema general del funcionamiento del lanzador de proyectiles</i>	53
<i>39. Servomotor para robótica que incluye rueda</i>	54
<i>40. Servomotores más utilizados en robótica que incluye rueda</i>	54
<i>41. Distribución de los servomotores, para la movilidad del robot</i>	56
<i>42. Combinaciones de giro en los servomotores, para movilidad del robot</i>	58
<i>43. Bosquejo en 3D del robot prototipo</i>	59
<i>44. Bosquejo de la base general del robot y de la placa de conexiones</i>	60
<i>45. Bosquejo de la base para el brazo mecánico</i>	61
<i>46. Bosquejo del primer eslabón del brazo mecánico</i>	62
<i>47. Taladro fresador y prensa que se usará en el proceso de fabricación de las piezas</i>	63
<i>48. Torno que se usará para el proceso de fabricación de las piezas</i>	64
<i>49. Entorno de Desarrollo Integrado (IDE) para Arduino</i>	65
<i>50. Entorno de desarrollo Eclipse para realizar la aplicación de dispositivos móviles</i>	65
<i>51. Entorno de desarrollo de NetBeans para realizar la aplicación de equipo de cómputo</i>	66
<i>52. Programa Terminal.exe que se usará en las pruebas de comunicación serial, con los módulos HC-05 y RF APC-220</i>	68
<i>53. Modificaciones realizadas a los servomotores de tracción</i>	69
<i>54. Conexión de servomotores con Arduino</i>	70

55. <i>Instalación automática de los controladores del convertidor USB/TTL</i>	71
56. <i>Instalación correcta de los controladores del APC-220</i>	72
57. <i>Configuración de los módulos emisores/receptores del APC-220 por medio de software</i>	73
58. <i>Conexión de los módulos APC-220 con Arduino y PC</i>	74
59. <i>Esquema del MAX232 para programar el módulo HC-05 y el circuito armado</i>	75
60. <i>Aplicación ALDBluetoothATConfig para el módulo HC-0</i>	76
61. <i>Conexión para prueba de transmisión de datos del módulo HC-05</i>	77
62. <i>Conexión para prueba del sensor de distancia HC-SR04 con Arduino</i>	77
63. <i>Conexión para realizar pruebas con el acelerómetro y Arduino</i>	78
64. <i>Maquilado de la base general del robot</i>	79
65. <i>Maquilado de la placa de conexiones</i>	80
66. <i>Maquilado de las piezas restantes</i>	80
67. <i>Piezas ya terminadas del robot</i>	81
68. <i>Diagrama del sistema electrónico del robot</i>	82
69. <i>Elaboración e integración del sistema electrónico del robot</i>	83
70. <i>Presentación final del robot prototipo</i>	84
71. <i>Bosquejo de la interfaz gráfica de usuario para PC</i>	86
72. <i>Software ArduBlock, para la programación gráfica del Arduino Nano del robot</i>	87
73. <i>Proceso de elaboración y presentación final de la interfaz gráfica de usuario para PC</i>	88
74. <i>Programación de la interfaz gráfica de usuario para dispositivos móviles en Eclipse</i>	89
75. <i>Interfaz gráfica de usuario terminada e instala en tableta</i>	90
76. <i>Pruebas de programación de Arduino con IDE y ArduBlock.jar</i>	93
77. <i>Pruebas de comunicación y control inalámbrica por medio de Terminal.exe y SerialTerminal.apk</i>	95
78. <i>Programación de un detector de obstáculo por medio de Ardublock.jar</i>	97
79. <i>Adquisición de datos de un sensor Vernier. (Colorímetro)</i>	99
80. <i>Adquisición de datos de sensores Vernier de presión y temperatura</i>	100

<i>81. Los datos procesados por la interfaz gráfica de los sensores de presión y temperatura</i>	100
<i>82. Comprobación al problema utilizando el robot didáctico</i>	105

INTRODUCCIÓN

El objetivo de este trabajo es el diseño y la construcción de un prototipo de robot, que puede emplearse como una herramienta didáctica en la enseñanza en el bachillerato en las asignaturas de ciencias: física, química y biología. En la actualidad la mayoría de los robots didácticos se emplean para enseñanza de robótica básica y muy poco en la enseñanza de ciencias; es decir, su diseño está enfocado en introducir al alumno en aspectos básicos de la robótica: construcción, programación, control, electrónica e instrumentación; temas que no están incluidos en los programas de estudio del bachillerato de la UNAM.

El integrar la robótica como herramienta en aulas y laboratorios, requiere de seleccionar los robots más adecuados para realizar el diseño de prácticas adecuadas a los objetivos que se tengan, así como un rediseño conceptual de lo que se llama práctica. Esto implica capacitación constante al personal docente y de laboratorio en el manejo de dichos robots.

En este trabajo se presenta una propuesta de un robot prototipo, y un ejemplo de su uso en una práctica escolar, para ejemplificar su aplicación. El robot fue diseñado mecánicamente para construir una interfaz gráfica de usuario de fácil uso para los docentes y alumnos. Se pretende que los cursos de capacitación sean enfocados al uso del robot en prácticas y no al diseño del mismo.

1.- MARCO TEÓRICO

Capítulo 1.- Marco Teórico

1.1.- Clasificación de robots según la ISO

Para empezar con el robot prototipo de este trabajo; debemos conocer las formas de clasificar los robots, para saber cómo diseñar e integrar el prototipo con los robots existentes mediante dicha clasificación. El campo actual de la robótica es tan extenso, que surge la necesidad de una clasificación; esto con la finalidad de agrupar robots semejantes y facilitar la toma de decisiones sobre que robot seleccionar para determinadas tareas; así como la transmisión de la información sobre el sistema tanto para fabricantes como usuarios.

La forma para clasificar los robots es tan variada como el campo de la robótica en sí; algunos de los factores que se pueden tener en cuenta para realizar una clasificación de los robots son: funcionamiento, programas que utilizan, sistemas internos, la tarea que realizan, etc. Como podemos apreciar la clasificación es muy diversa; por tal motivo, fabricantes y usuarios llegaron a un consenso sobre las formas más comunes de clasificar los robots, la cuales están basadas en su:

A.- Fuente de alimentación. Esta clasificación es la más fácil y obvia, pues los robots en este grupo utilizan el mismo tipo de energía para realizar sus funciones, es una de las más utilizadas en los grandes robots y herramientas; dentro de esta categoría existen sub categorías genéricas para agrupar los robots. Cada una de ellas hace referencia a la fuente de alimentación que utilizan las partes del robot para llevar a cabo su funcionamiento.

- **Eléctrica.** Utiliza energía eléctrica para hacer funcionar diferentes tipos de motores eléctricos y darle movilidad al robot; así como a sus diversos actuadores. Cuando se está, se tiene la opción de utilizar corriente alterna (AC) o corriente directa (DC). Las grandes industrias y fábricas, selecciona robots que para su movilidad y acción requieran de motores y actuadores de corriente alterna debido a su fácil disponibilidad en la red eléctrica, bajo mantenimiento y es viable usarle en motores paso a paso y servomotores.

Los motores de corriente directa son empleados en los robots enfocados en tareas de trabajo pesado, pues estos ofrecen gran par, tanto al arranque como a tensión reducida. Una variante de los motores de corriente directa son los motores de pasos, muy utilizados en aplicaciones que requieren un control preciso en la posición del motor. En la figura 1 muestran un robot de la marca FANUC, empleado para apilar paquetes en una fábrica de chocolate, emplea servomotores y cableado de corriente alterna (izquierda); un par de robots que utilizan servomotores de corriente directa, para impulsar a los robots en su avance y accionamiento (derecha).

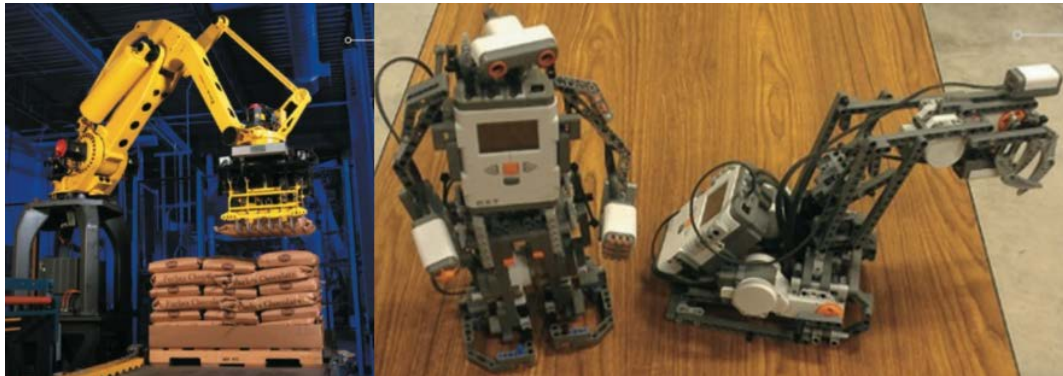


Figura 1.A la izquierda, un robot que utiliza AC y a la derecha un robot que utiliza DC

Fuente:(Basic Robotics, 2016)

- **Hidráulica:** Es utilizada por aquellos robots que necesiten aplicar una gran cantidad de fuerza; que se obtiene de presionar un líquido incompresible. Los robots que utilizan esta fuente de alimentación solo son usados exclusivamente en el ámbito industrial debido al riesgo de fugas hidráulicas, costos en aceites, alto mantenimiento, equipos de refrigeración y bombas. Un ejemplo de estos robots, son las trituradoras industriales.
- **Neumática:** Funciona de la misma manera que la hidráulica, solo que se sustituye el líquido por un gas. Presenta los problemas de mantener la presión del gas constante y posicionar los actuadores. Sin embargo estos robots son baratos, ya que muchas industrias tienen un suministro de aire comprimido para su uso, además de que su mantenimiento es más limpio y barato que el de un sistema hidráulico.

Un área donde la neumática ha tomado auge es en el área de automatización de procesos, en la figura 2 se ilustra un robot de la marca FANUC que utiliza la neumática para la fabricación de piezas.



Figura 2. Robot neumático para fabricación y empaquetado.

Fuente: (Basic Robotics, 2016)

B.- Geometría del espacio de trabajo. Otra forma, según [1]; ampliamente difundida de agrupar los robots es con base en su “espacio de trabajo”; o dicho de otras maneras, por el área o puntos que pueden abarcar. Esta forma de clasificación es la más utilizada en la industria ya que le da al usuario una mejor idea de cómo se mueve el robot al interactuar con el espacio que lo rodea. Dentro de esta categoría existen varias sub-categorías para poder agrupar los robots, las cuales son:

- **Robots Cartesianos.** Se incluyen en esta sub-categoría aquellos robots cuyo movimiento sea cúbico o rectangular, o cuyo movimiento sea lineal. Los robots de este grupo tienen a menudo 2 o 3 ejes principales para su movimiento:

X-adelante/atrás, Y-derecha/izquierda y Z-arriba/abajo. Son utilizados para carga, descarga y transporte de piezas, algunas máquinas maquiladoras y de control numérico entran dentro de este grupo.

- **Robots Cilíndricos.** Su espacio de trabajo describe un cilindro. Estos robots tienen 1 eje giratorio y 2 ejes lineales para mover la herramienta además de dos ejes menores para la orientación de la misma. Este tipo de robots son buenos para ahorrar espacio y pueden con grandes cargas, su inconveniente es la pérdida de movimiento del eje Y, es compensado con la adición de una base móvil.
- **Robots Esféricos.** El espacio de trabajo teórico formado por los robots en esta agrupación, es el de una esfera con el robot en el centro de la misma; aunque en la práctica dicha esfera es restringida por la construcción física del robot. Esta clase de robots tienen un barrido y alcance de un robot cilíndrico más el posicionamiento angular para las principales funciones del efector final.
- **Robots Articulado o antropomórficos.** Los robots en este grupo cuentan con movimientos de rotación en diferentes ejes, su espacio de trabajo se parece a la de un robot esférico con algunos salientes en la esfera. Dado que su espacio de trabajo no tiene una forma geométrica definida; también se les conoce como robots articulados cuando poseen un brazo articulado con efector final; ó como antropomórficos, porque en algunos casos sus movimientos imitan los de animales y el ser humano, en este conjunto entrarían los robots humanoides; aunque la desventaja de todos estos robots es que requieren controladores más complejos.
- **Robots SCARA.** *Selective Compliance Articulated Robot Arm*, son aquellos robots que combinan el movimiento lineal de un robot cartesiano, con la rotación de un sistema articulado, creándose un nuevo tipo de movimiento, 2 ejes rotan y 1 eje se traslada linealmente, por esta flexibilidad en su movimiento son utilizados en industrias cuyas tareas requieren de precisión, como por ejemplo el ensamblado de

placas electrónicas. Una variante de este tipo de robots, tienen 1 eje que rota y 2 ejes lineales; se les da el nombre de robots DELTA.

En la figura 3 se ilustran los espacios teóricos de trabajo formados por los robots de cada una de las sub-categorías; espacio de un robot cartesiano en el inciso (a), espacio de un robot cilíndrico en el inciso (b), espacio de un robot esférico en el inciso c), espacio de un robot articulado o antropomórfico en el inciso (d) y el espacio de un robot SCARA en el inciso (e).

Geometrías formadas por los espacios de trabajo de cada uno de los robots de la clasificación

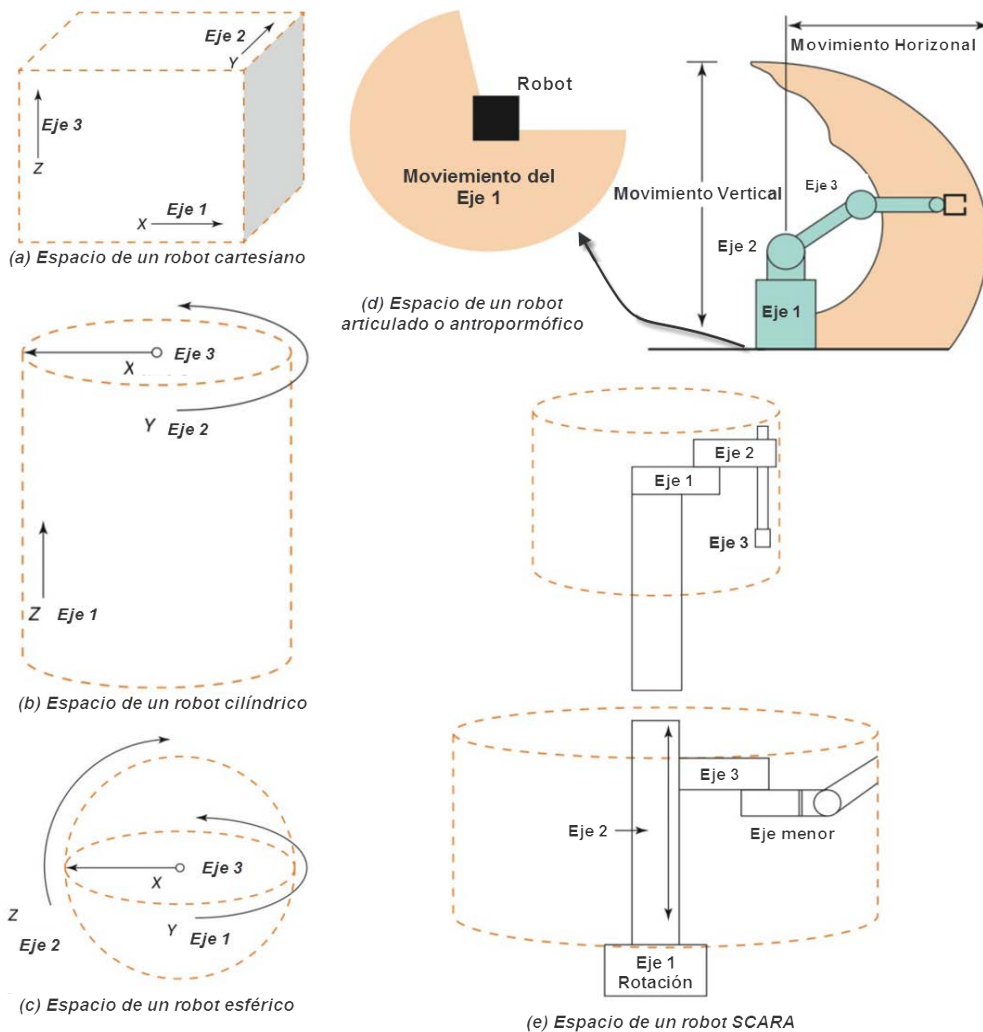


Figura 3. Geometrías de los espacios de trabajo, formados por los robots de la clasificación.

Fuente: (Basic Robotics, 2016)

En la figura 4, aparecen ejemplos de robots que forman algunas de las geometrías de los espacios de trabajo descritos, en el inciso (a) aparece un robot cartesiano utilizado para la soldadura de tanques, en el inciso (b) observamos un robot articulado, cuya base fue montada en un tipo riel, es empleado para la fabricación de piezas, el inciso (c) muestra un robot SCARA, el inciso (d) muestra una fotografía de un robot DELTA de la marca ABB con aplicaciones en el maquinado y la pintura de piezas.

Fotografías de robots reales, que forman las geometrías de la clasificación por espacio de trabajo



(a) Robot cartesiano, utilizado en la soldadura de tanques



(b) Robot articulado, empleado en la fabricación de piezas



(c) Robot SCARA



(d) Robot DELTA, usado para la pintura

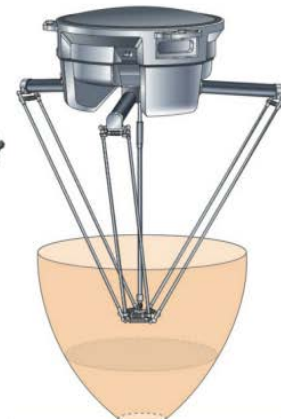


Figura 4. Fotografías de robots que forman algunos de los espacios de trabajo.

Fuente: (Basic Robotics, 2016)

C.- Sistemas de potencia o sistemas actuadores. Esta clasificación se basa en la forma en cómo se conectan los motores a las partes móviles, o el modo de transmisión de los robots. En general dentro de esta clasificación se pueden tener cuatro sub-categorías:

- **Transmisión directa del movimiento.** Agrupa aquellos robots que tienen el eje de rotación del motor conectado directamente a la parte del robot que se mueve. El acople es directo con una relación 1:1, por tanto no existe limitación en el par.
- **Transmisión reducida.** En esta agrupación se encuentran aquellos robots cuyos sistemas de accionamiento alternan la salida del eje del motor a través de medios mecánicos; para reducir velocidad, aumentar el par del sistema; y cambiar la dirección de la rotación.
- **Transmisión por correa y/o cadenas.** La forma en que estos robots transmiten la fuerza del motor al sistema de movimiento, es por medio de un sistema de poleas y correas (planas o dentadas) o en su debido defecto con cadenas.
- **Transmisión por engranes.** La transmisión entre los motores y las articulaciones, se realiza por medio de engranes, con ellos se pueden conseguir reducciones o aumento de la fuerza motriz; así como cambio en la dirección del giro del motor.



Figura 5. Ejemplos de la clasificación basada en los medios de transmisión.

Fuente: (Basic Robotics, 2016)

En la figura 5, podemos apreciar algunos de estos medios de la transmisión de movimiento, la mayoría son utilizados sobre todo en maquinaria de todo tipo; como por ejemplo compresores de aire, y clasificar un robot con este criterio resulta un poco abstracto; es decir, no es una clasificación muy utilizada en la robótica.

De las formas de clasificar los robots expuestas, podemos decir, que algunas dejan al fabricante y usuario un amplio criterio para su clasificación, otras no se adecuan al campo de la robótica de una manera satisfactoria; en otras palabras, aunque existen clasificaciones consensuadas por fabricantes y usuarios, no existe un estándar para la clasificación

Por lo anterior, en los últimos 10 a 15 años, la Organización Internacional de Normalización (ISO), ha empezado a fijar los métodos de clasificación más comunes en un solo Estándar.

La clasificación ISO agrupa a los robots en 3 principales categorías:

D. Robots Industriales.

E. Robots de Servicios.

F. Robots Médicos.

D.- Robots Industriales. Esta clasificación es la más trabajada por la ISO, y fue la primera categoría que se dio a conocer oficialmente. Para estar en esta clasificación, un robot industrial deberá cumplir con la siguiente definición de la ISO: “Un robot industrial es aquel mecanismo programable accionado en 2 o más ejes, con un grado de autonomía moviéndose dentro de su entorno, para realizar tareas previstas” [1]. Pero con el crecimiento actual de la electrónica industrial y su aplicación en la robótica, provocó que la norma ISO redefiniera esta categoría; basándose en la estructura mecánica de los robots, de la siguiente manera:

- **Robots Lineales**
- **Robots Articulados**
- **Robots Cilíndricos**
- **Robots Paralelos**
- **Otros**

Al observar las sub categorías que se incluyen en la clasificación ISO para un robot industrial, podemos ver que es parecida a la clasificación basada en el espacio de trabajo de los robots; de hecho, la ISO contempla en el grupo de robots lineales a los robots cartesianos y de tipo pórtico, en el grupo de robots paralelos a los robots tipo delta, los robots articulados de la ISO incluyen a los robots esféricos, los sub grupos de robots cilíndricos coinciden en ambos casos y por último la categoría de otros de la ISO, agrupa a los robots SCARA, antropomórficos, etc.

E.- Robots de Servicio. Para que un robot pueda ser catalogado como de servicio, debe cumplir con la siguiente definición de la ISO: " Un robot que realiza tareas útiles para el ser humano o el equipo con exclusión de las aplicaciones de automatización industrial." [1].

La definición excluye específicamente a los robots industriales, y se compone de 3 sub categorías que son:

- **Robots de cuidado personal.**
- **Robots sirvientes móviles.**
- **Robots asistentes físicos.**
- **Robots transportador personal.**

Podemos resumir estos robots, como aquellos robots que pueden entrar en contacto con la gente para ayudar a realizar acciones que mejoren la calidad de vida. Los robots de esta clasificación son sistemas de gama alta que pueden ayudar con el día a día de la vida personal, particularmente con dificultades en la movilidad humana. Por las sub categorías que propone la ISO para este grupo podemos notar que están diseñados para moverse alrededor de un espacio, llevar objetos e interactuar con el medio ambiente, o incrementar las capacidades físicas de la gente o sustituir aquellas que han perdido. Algunos ejemplos de estos robots se pueden apreciar en la figura 6, algunos de ellos son comunes como las sillas de ruedas motorizadas y exoesqueletos robóticos.

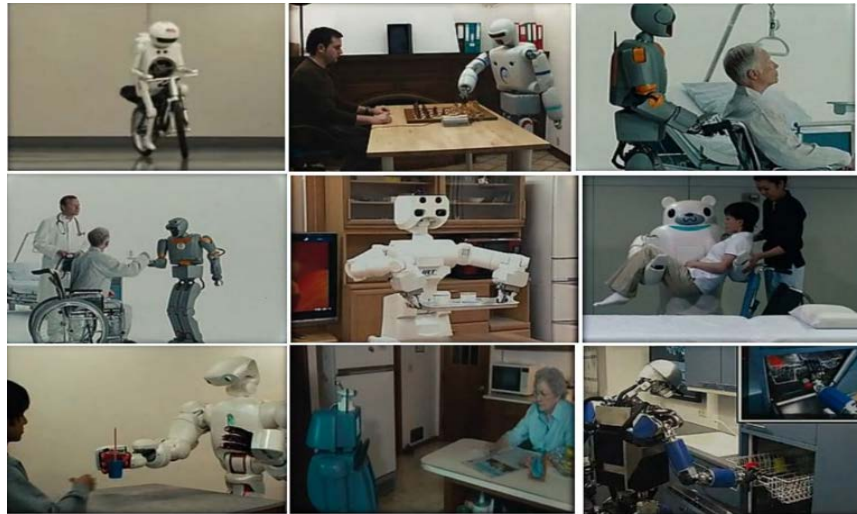


Figura 6. Ejemplos de la clasificación ISO, para los robots de servicio.

Fuente: <http://recuerdame2013.blogspot.mx/2013/06/alzheimer-robotica-y-un-futuro-no-muy.html>

F.- Robots Médicos. Se define como: “Un robot o dispositivo robótico, es aquel destinado a ser utilizado como equipo electro-médico.” [1]. Esta clasificación es la más reciente dada por la ISO, puesto que anteriormente los robots médicos, estaban en la categoría de robots de servicio. Esta clasificación se dio a raíz del desarrollo de robots de apoyo para el cirujano, tales como el robot da vinci mostrado en la figura 7.



Figura 7. Robot quirúrgico Da Vinci.

Fuente: <http://www.jornada.unam.mx/ultimas/2015/01/31/avanza-cirugia-robotica-en-mexico-especialistas-9110.html>

Una vez estudiadas las clasificaciones disponibles para los robots, basaremos la clasificación del robot prototipo de este trabajo en la norma ISO. Dado que el robot prototipo, es un robot enfocado a la enseñanza de ciencias básicas del bachillerato UNAM, no podemos clasificarlo de acuerdo con la tarea que realizará, pues no entraría en alguna de las clasificaciones de norma ISO; por la misma razón no puede ser catalogado como robot de servicio y mucho menos como un robot médico.

El robot prototipo deberá tener facilidad en su desplazamiento, servirá para el lanzamiento de proyectiles y el estudio de tiro parabólico; en otras palabras, el brazo contará con 2 ejes con movimientos giratorios y su base tendrá una combinación de movimientos lineales y circulares; si bien no es un robot que será usado en la industria sino en la educación, basaremos su clasificación de acuerdo con su espacio de trabajo; por lo tanto, este robot será un robot articulado perteneciente a la clasificación de robot industrial.

1.2.- Sistemas embebidos basados en microcontrolador

Los sistemas embebidos se encuentran inmersos en nuestra vida cotidiana, es un término poco utilizado y prácticamente cualquier ser humano ha interactuado con un dispositivo basado en ellos. Ejemplos de estos dispositivos se encuentran en aparatos domésticos como, lavadoras, relojes, juguetes, equipo de audio y video, teléfonos hasta en dispositivos médicos y militares. En la figura 8, se ilustran algunas de las aplicaciones mencionadas de los sistemas embebidos.

Un sistema electrónico embebido básicamente se compone de una unidad central de procesamiento (o varias) conectada a diferentes dispositivos mecánicos, químicos y eléctricos, programada para un propósito dedicado y empacada como un sistema completo. El término embebido viene de la palabra inglesa “*embed*” que significa, en este contexto, oculto en el interior. Es decir, se refiere a sistemas electrónicos formados por una unidad central de procesamiento y otros dispositivos que; en su mayoría, se encuentran “ocultos en el interior” de un mismo gabinete.

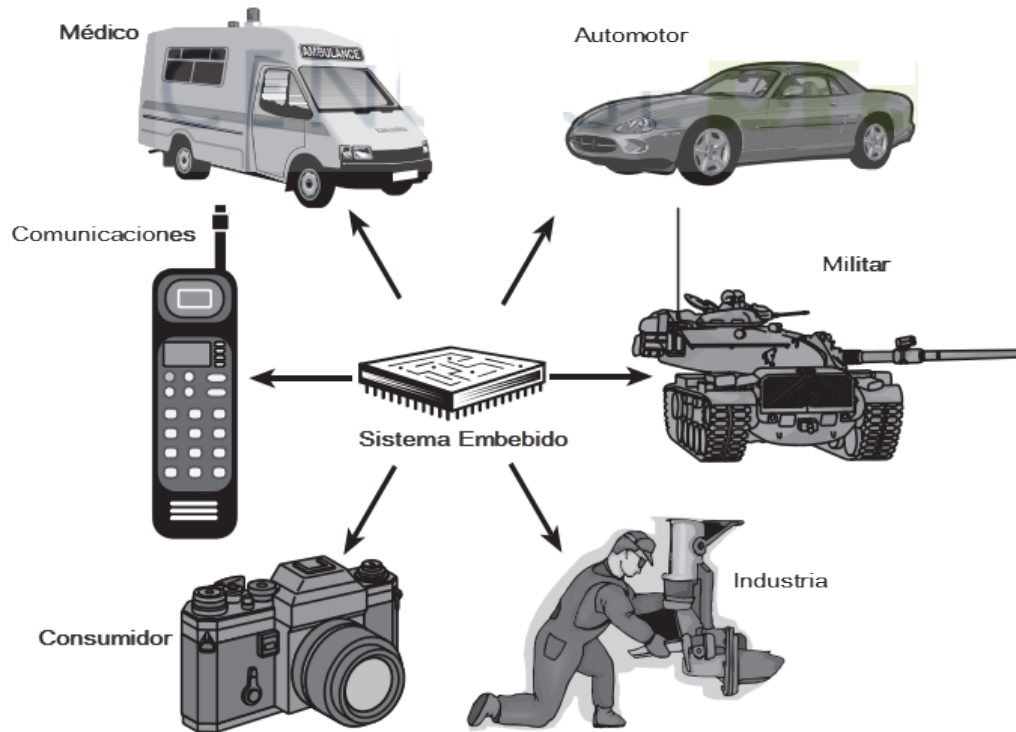


Figura 8. Aplicaciones de los sistemas embebidos.

Fuente: (*Embedded Microcomputer Systems: Real Time Interfacing*, 2012)

Un sistema embebido está conformado por circuitos integrados programables que constituyen la unidad central de procesamiento (CPU), memoria RAM, memoria flash o ROM, el correspondiente circuito impreso y el software embebido como parte esencial del mismo sistema, conocido en inglés como “*embedded software*”, utilizado para controlar, monitorizar y/o programar tanto los productos electrónicos, como a la CPU.

Para la CPU del sistema embebido, se escoge alguno de los siguientes dispositivos:

- Un microprocesador interno
- Un microcontrolador
- Un procesador digital de señal (DSP)
- Una compuerta programable en campo (FPGA)
- Un controlador lógico programable (PLC)
- Una PC de propósito general adaptada para un fin específico.

En [3]; en cuanto al software para el sistema embebido se pueden tener varias categorías:

- El software de base indispensable para el funcionamiento de la CPU y el cual conforma el sistema operativo; algunos de estos lenguajes de programación son ASSEMBLER, C/C++ o VHDL.
- Otro tipo es el dedicado a la supervisión y control de sistemas complejos cuyo funcionamiento implica la interrelación de gran cantidad de equipos o aparatos, por ejemplo el funcionamiento y control de los distintos procesos en la industria manufacturera; este es el caso del software inherente al funcionamiento de los PLC's.

De los dispositivos anteriores empleados para el desarrollo de sistemas embebidos, los microcontroladores (abreviado μ C, UC o MCU) son los más utilizados para dicho fin; debido a que en su encapsulado integran todos los componentes funcionales como el microprocesador. En la figura 9, se ilustra el esquema básico de un sistema básico de microcontrolador.

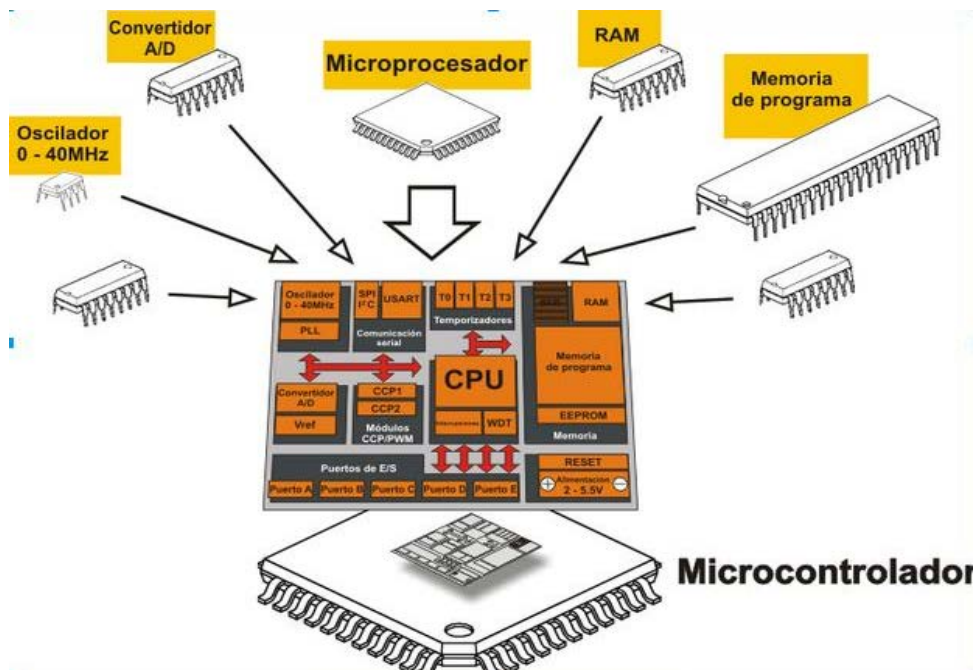


Figura 9. Componentes de un microcontrolador.

Fuente: <http://learn.mikroe.com/ebooks/microcontroladorespic/chapter/introduccion-al-mundo-de-los-microcontroladores/>

En la actualidad existen varios fabricantes de microcontroladores; cuyos microprocesadores están basados en algunas de las arquitecturas presentadas. Ofrecen al usuario una placa o sistema de desarrollo con algún modelo de sus microcontroladores, junto con todos los componentes necesarios para su funcionamiento; todo en una sola PCB acondicionada para su uso. En este sistema de desarrollo el usuario puede conectar todos los aditamentos eléctricos-electrónicos necesarios para la realización de sus proyectos. Además el fabricante otorga; generalmente de manera gratuita, el software requerido para la utilización de sus sistemas de desarrollo. Una estructura de estos sistemas de desarrollo, se ilustra en la figura 10.

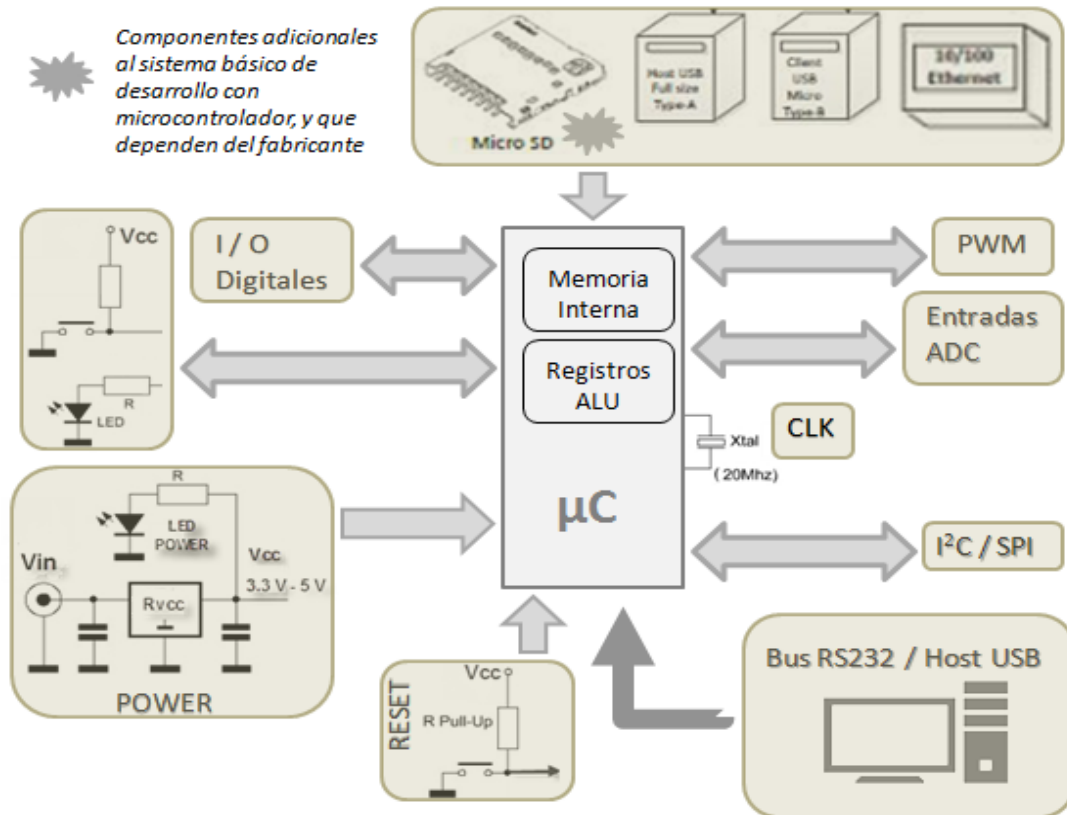


Figura 10. Diagrama a bloques de una placa de desarrollo basada en un microcontrolador

Cabe señalar que el diagrama es muy sencillo, puesto que cada fabricante escoge qué componentes estarán en su sistema de desarrollo, esto con la finalidad de ofrecer al usuario tanto placas de bajo costo con los componentes mínimos para su funcionamiento; como placas de un alto costo, pero con altas prestaciones y funcionalidades.

1.3.- Control y programación de un robot

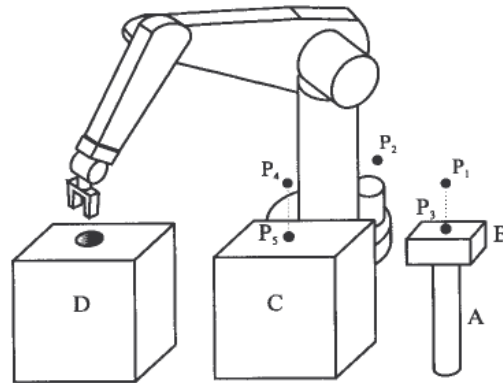
En los apartados anteriores, categorizamos al robot prototipo como un robot articulado perteneciente a la clasificación de robot industrial de acuerdo con la norma ISO; también estudiamos la estructura genérica de un sistema embebido basado en microcontrolador, puesto que emplearemos dichos sistemas para el control y programación del robot. El siguiente paso es saber cómo podemos programar, controlar y por consiguiente comunicarnos con un robot.

A un robot de tipo industrial se le demandan por lo general elevadas prestaciones en velocidad y precisión. Este objetivo ha marcado las investigaciones y avances tecnológicos actuales de la robótica, tales como el uso de materiales más ligeros, sistemas de transmisión con baja inercia, actuadores más rápidos y precisos, sensores de elevada resolución, etc. Del mismo modo, los algoritmos de control del robot tienen por objetivo mejorar al máximo las características y funcionalidades del mismo. La adaptación rápida y económica a diferentes aplicaciones de un robot industrial, proviene de la capacidad de reprogramación. “La programación de un robot se puede definir como el proceso mediante el cual se le indica a éste la secuencia de acciones que deberá llevar a cabo durante la realización de su tarea.” [4]

Durante la ejecución de un programa, interactúan la memoria del sistema y el control del mismo leyendo y actualizando las variables del programa encargadas de dar la señal de mando a los actuadores del robot; por tanto, el sistema de programación es una herramienta con que cuenta el usuario para acceder a las diversas prestaciones del robot. Otra definición de programar un robot es: “programar un robot consiste en indicar paso por paso las diferentes acciones que éste deberá realizar durante su funcionamiento”. [4]

La programación consiste en hacer realizar al robot la tarea (llevándolo manualmente por ejemplo) al tiempo que se registran todos los movimientos, variables y configuraciones, para su posterior repetición de manera automática; o bien, indicarle al robot la tarea a realizar mediante comandos específicos; en un entorno de programación. En la figura 11, se muestra un ejemplo genérico de la programación.

Ejemplo de tarea de programación.



```

Mover_a P1 via P2          ; Situarse en un punto sobre la pieza B
Vel = 0.2 * VELMAX        ; Reducir la velocidad
Pinza = ABRIR             ; Abrir la pinza
Prec = ALTA               ; Aumentar la precisión
Mover_recta_a P3          ; Descender verticalmente en línea recta
Pinza = CERRAR           ; Cerrar la pinza para coger la pieza B
Espera= 0.5              ; Esperar para garantizar cierre de pinza
Mover_recta_a P1          ; Ascender verticalmente en línea recta
Prec = MEDIA             ; Decrementar la precisión
Vel = VELMAX             ; Aumentar la velocidad
Mover_a P4 via P2         ; Situarse sobre la pieza C
Prec = ALTA              ; Aumentar la precisión
Vel = 0.2 * VELMAX       ; Reducir velocidad
Mover_recta_a P5          ; Descender verticalmente en línea recta
Pinza = ABRIR            ; Abrir pinza

```

Figura 11. Esquema simplificado de programación de un robot industrial.

Fuente: (Fundamentos de Robótica, 2010)

Para el caso concreto de nuestro robot prototipo, no se diseñará un lenguaje y entorno de programación en particular sino más bien nos basaremos en lenguajes ya existentes para lograr dicho cometido. Podremos verlo en dos bloques; el bloque del sistema embebido encargado del control del robot, y el bloque del entorno de programación; cada uno con su lenguaje propio. Para este último, se disponen de los siguientes lenguajes:

- C/C++
- Java
- Python
- Visual Basic
- Ajax
- Ada
- Php

- Ruby
- Perl
- Pascal
- Cobol
- Fortran

Seleccionaremos el entorno Java para dicho propósito, más adelante se explicará él por qué de esta selección. En lo correspondiente al sistema embebido, el lenguaje de programación, será el lenguaje que el fabricante haya escogido para su sistema de desarrollo de microcontrolador; generalmente el lenguaje para microcontroladores es uno de bajo nivel o lenguaje ensamblador. El lenguaje ensamblador a diferencia del lenguaje de uno de alto nivel, permite al usuario tener un control preciso del flujo de programa, espacio en memoria ocupado entre otras, pero requiere un conocimiento avanzado de este tipo de lenguaje; en cambio, el lenguaje de alto nivel es el más cercano al lenguaje humano, por lo que resulta más sencillo su entendimiento. En términos generales, los fabricantes de microcontroladores que ofrecen un sistema de desarrollo, otorgan al usuario un entorno de programación, donde este podrá realizar sus proyectos en un lenguaje de alto nivel, y este entorno se encargará de traducirlos a un lenguaje ensamblador o lenguaje máquina, para posteriormente cargarlos al microcontrolador y este pueda ejecutarlos. En la figura 12 se ilustra un ejemplo de este proceso, donde se puede apreciar la diferencia entre dichos lenguajes.

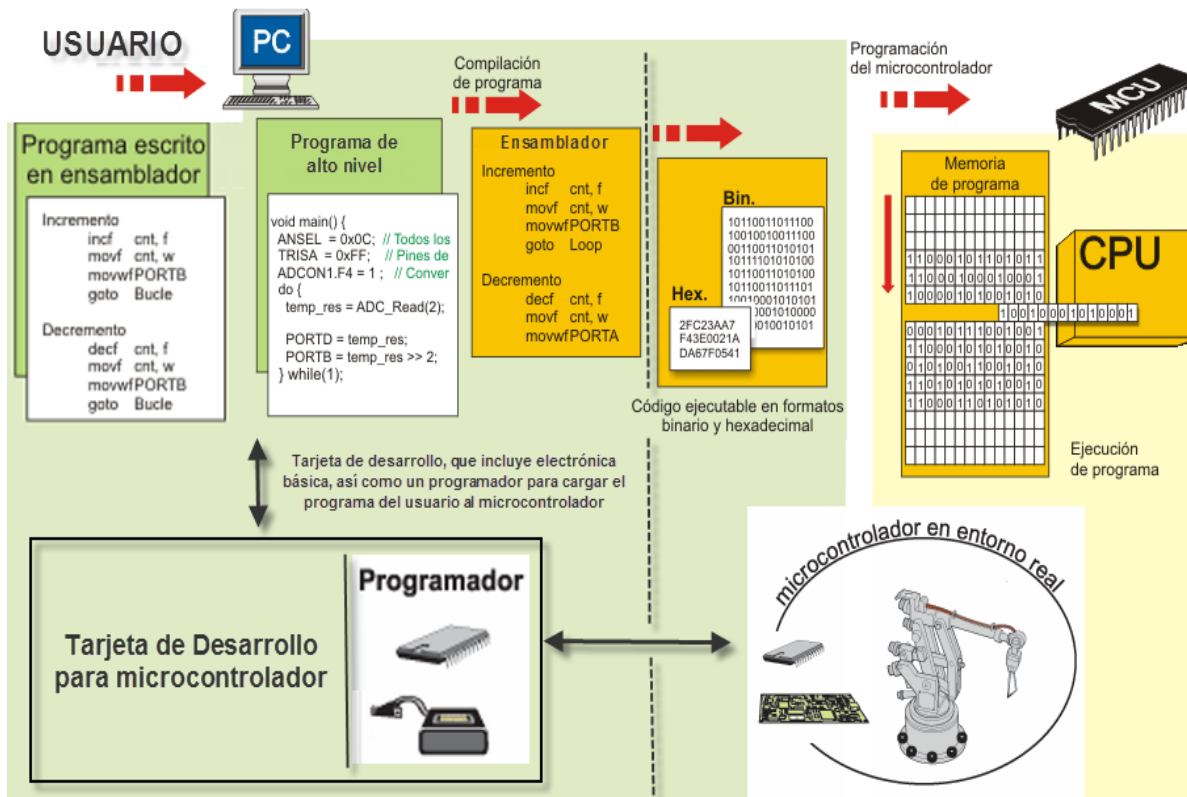


Figura 12. Proceso y diferentes lenguajes de programación.

Fuente: <http://learn.mikroe.com/ebooks/microcontroladorespicc/chapter/lenguajes-de-programacion/>

En la robótica industrial moderna, según [7], las comunicaciones contemplan tres principales características que son:

- Velocidad de datos. Cantidad de datos que viajan por la red en cada envío (Paquete de datos)
- Velocidad de transmisión. Velocidad a la que viajan los datos por la red (Baudaje)
- Velocidad de respuesta. Tiempo de respuesta que hay entre el momento de dar la orden y la respuesta del dispositivo (Timeout).

Estas características, junto con características eléctricas, mecánicas, computacionales, las formas de envío/recepción de información, los medios de transmisión, etc. Están comprendidas dentro de un protocolo de comunicación; por lo tanto podemos definir un

protocolo de comunicación como el conjunto de reglas y formatos, para la transmisión de información entre dos puntos. Estas reglas definen la forma como debe efectuarse la comunicación, interpretación de los comandos, formato de los mensajes, la sincronización y revisión de datos; así como la corrección de errores. Con base en [9] todo esto se organiza en sub-tareas y funciones denominados como capas.

De las reglas y tareas para la transmisión de información, surgen diferentes criterios para definir las mismas, por lo mismo, existe una gran variedad de protocolos de comunicación, cada uno con sus propias reglas. Con el propósito de unificar la forma comunicación entre emisor y receptor, la Organización Internacional de Normalización (ISO) en 1984 desarrolló un modelo para la conexión denominado *Open Systems Interconnection Reference Model* o Modelo de Referencia de Interconexión de Sistemas Abiertos, comúnmente conocido como modelo OSI. Este consiste en dividir en 7 capas encargadas de ejecutar una determinada parte del proceso general. El propio modelo ofrece los mecanismos y reglas que permiten resolver todas las cuestiones que se mencionaron. La figura 13 ilustra el modelo OSI con sus respectivas capas.

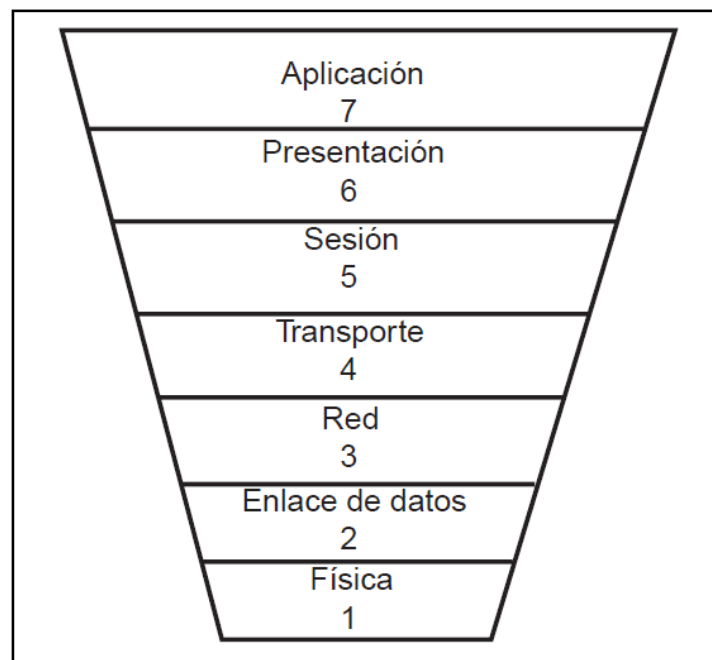


Figura 13. Capas del Modelo OSI

http://www.info-ab.uclm.es/labelec/Solar/Comunicacion/Redes/index_files/Modelos.htm

A continuación se describe en términos generales cada una de las capas:

- **Capa 7. La capa de aplicación** es la capa del modelo OSI más cercana al usuario; puesto que suministra servicios de red a sus aplicaciones. Difiere de las demás capas debido a que no proporciona servicios a ninguna otra capa, sino solamente a aplicaciones que se encuentran fuera del modelo. Algunos ejemplos de aplicaciones son los programas de hojas de cálculo, de procesamiento de texto y los de las terminales bancarias.
- **Capa 6. La capa de presentación** garantiza que la información que envía la capa de aplicación de un sistema pueda ser leída por la capa de aplicación de otro. De ser necesario, esta capa traduce entre varios formatos de datos utilizando un formato común. Se encarga también de la compresión y encriptado de datos.
- **Capa 5. La capa de sesión** establece, administra y finaliza las sesiones entre dos *hosts* (equipos) que se están comunicando. Proporciona servicios a la capa de presentación, sincroniza el diálogo y administra su intercambio de datos. Además regula la sesión, la transferencia de datos, controlar el diálogo entre aplicaciones en sistemas y registra de excepciones acerca de los problemas de la capa de sesión, presentación y aplicación.
- **Capa 4. La capa de transporte** la capa de transporte segmenta los datos originados en el *host* emisor y los re-ensambla en una corriente de datos dentro del sistema del *host* receptor. Mientras que las capas de aplicación, presentación y sesión están relacionadas con asuntos de aplicaciones, las cuatro capas inferiores se encargan del transporte de datos. La capa de transporte intenta suministrar un servicio de transporte de datos que aísla las capas superiores de los detalles de implementación del transporte. Específicamente, temas como la confiabilidad del transporte entre dos *hosts* es responsabilidad de la capa de transporte. Al proporcionar un servicio de comunicaciones, la capa de transporte establece, mantiene y termina adecuadamente los circuitos virtuales.

- **Capa 3. La capa de red** la capa de red es una capa que proporciona conectividad y selección de ruta entre dos sistemas de *hosts* que pueden estar ubicados en redes geográficamente distintas. Se encarga de que los datos atraviesen distintas redes interconectadas (ruteo).
- **Capa 2. La capa de enlace de datos** la capa de enlace de datos proporciona tránsito de datos confiable a través de un enlace físico. Al hacerlo, la capa de enlace de datos se ocupa del direccionamiento físico, la topología y acceso a la red, la notificación de errores, entrega ordenada de tramas y control de flujo. En esta capa los protocolos realizan control de errores, de secuencia y de flujo.
- **Capa 1. La capa física** define las especificaciones eléctricas, mecánicas, de procedimiento y funcionales para activar, mantener y desactivar el enlace físico entre sistemas finales. Las características tales como niveles de voltaje, temporización de cambios de voltaje, velocidad de datos físicos, distancias de transmisión máximas, conectores físicos y otros atributos similares son definidos por las especificaciones de la capa física.

La figura 14, muestra como es la interacción de las capas del modelo OSI entre dos equipos; así mismo muestra como se agrupan las capas para formar el hardware, software y sistema operativo con los que trabaja el usuario. También muestra cables UTP y de fibra óptica, como ejemplos de los medios de transmisión para la capa física.

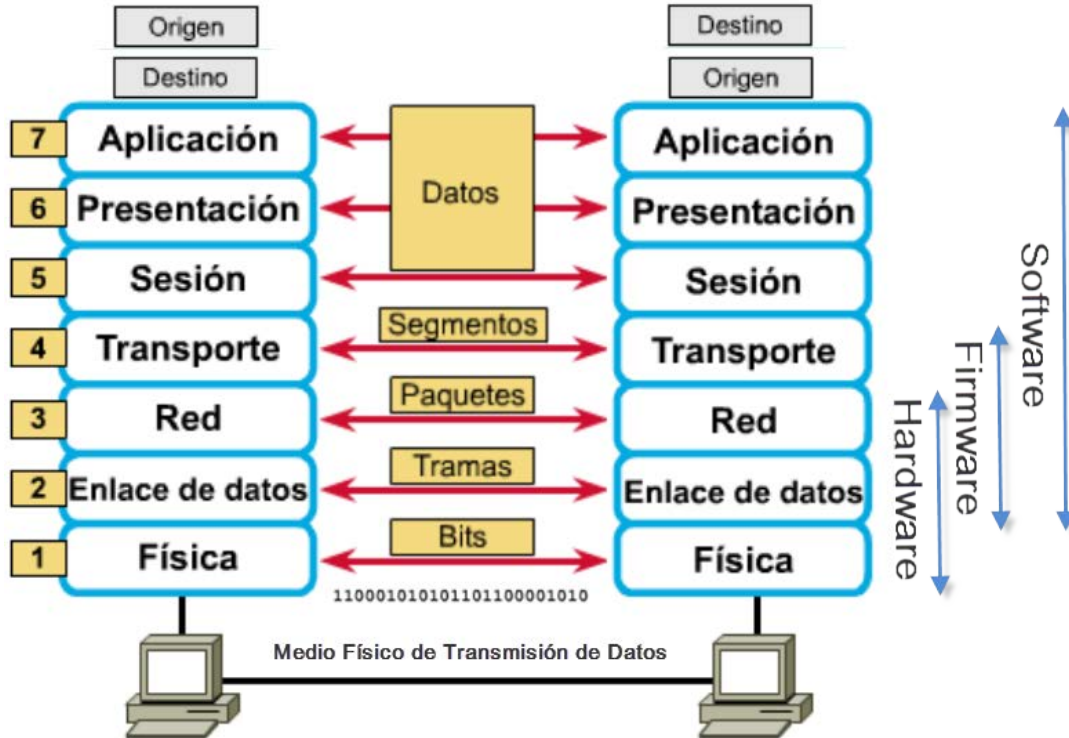


Figura 14. Descripción general de las capas y el modelo OSI en operación
<http://www.exa.unicen.edu.ar/catedras/comdat1/material/ElmodeloOSI.pdf>

Es justamente en el medio de transmisión de la capa física, el que seleccionaremos para incorporarlo a nuestro robot prototipo. En cuanto a las demás capas, escogeremos el protocolo RS – 232 para el envío/recepción y no profundizaremos más que en los aspectos necesarios para su utilización. Dentro del modelo OSI, la IEEE, ofrece diferentes protocolos para la comunicación inalámbrica; pero los más usados, incluso en el ámbito de la robótica aparecen en la tabla de la figura 15.

Medio / Características	GPRS / GSM 1xRTT / CDMA	IEEE 802.11 b/g	IEEE 802. 15.1	IEEE 802 .15.4
Nombre comercial para el estándar	2.5 G 3G 4G	Wi - Fi	Bluetooth	ZigBee
Objetivo de la red	WAN / MAN	WiFi y punto de acceso	PAN & DAN (zona de recepción n/w)	WSN
Aplicaciones destinadas	Amplia área de voz y datos	Aplicaciones empresariales (datos y VoIP)	Sustitución del cable	Monitorización control
Ancho de Banda (Mbps)	0.064 – 0.128 +	11 - 54	0.7	0.020 – 0.25
Rango de Transmisión (ft)	3000+	300+	30+	300+
Consideraciones para el diseño y uso	Gran alcance y transmisión	Apoyo y uso en empresas, costo elevado	Costo bajo, facilidad de uso	Fiabilidad, bajo consumo energético

Figura 15. Tabla comparativa de los diferentes medios de comunicación inalámbrica de la IEEE 802

Fuente: <http://www.slideshare.net/Jaddu44/basics-of-wireless-sensor-networks>

Estos protocolos de ser los más utilizados, también son los de más fácil adquisición en el mercado; por tal motivo, buscaremos algún módulo inalámbrico que se comunique con alguno de estos protocolos de la IEEE 802, y que para la transmisión de datos trabaje con el protocolo RS – 232; esto con la finalidad de que sea compatible con el sistema de microcontrolador seleccionado para el control del robot. Otra razón para lo anterior, es que la mayoría de los fabricantes de tarjetas de desarrollo de microcontroladores, incorporan de fábrica un módulo para comunicación serial RS – 232. Con esto haremos que el robot prototipo sea totalmente inalámbrico.

2.- DESARROLLO

Capítulo 2.- Desarrollo

2.1.- Diseño conceptual del robot

En el capítulo anterior se presentó la clasificación para robots, los lenguajes de programación, el control de los mismos a través de sistemas embebidos basados en microcontroladores, así como los diferentes protocolos para la comunicación y los medios para la transmisión de información. Por lo expuesto, el esquema más adecuado para diseñar el robot didáctico es el de un robot articulado, según la clasificación de la ISO. Un microcontrolador será su unidad central de procesamiento; y tendrá un módulo de comunicación inalámbrica bajo el protocolo RS – 232, para que el controlador, el módulo de comunicación y la programación se realicen bajo este mismo protocolo. Las funciones a implementar son:

- En las prácticas del bachillerato de la UNAM, se requiere adquirir datos experimentales a través de sensores, con la finalidad de que el alumno cuente con información suficiente que le permita analizar el fenómeno que observa. Por lo anterior el robot contará con un sistema de adquisición de datos con interfaz gráfica de usuario para captura y análisis de los mismos.
- El brazo del robot tendrá dos grados de libertad.
- El robot tendrá un lanzador de balines accionado electromecánicamente.
- El robot, será usado como herramienta para experiencias básicas en robótica; es decir, podrá ser reprogramado por el usuario para llevar a cabo tareas específicas (por ejemplo programar un seguidor de línea.
- En la robótica actual; una tendencia es el control y programación de robots por medio de dispositivos móviles; tales como *smartphones* y *tablets*; bajo el sistema operativo Android, a través de protocolos de comunicación inalámbrica (bluetooth, wi.fi, etc.). En consecuencia, el prototipo contará con comunicación vía bluetooth y una aplicación para dispositivos móviles Android.

En la figura 16, se ilustran las funciones para el prototipo, mencionadas en los dos últimos puntos; en el inciso (a) un robot seguidor de línea operado mediante un teléfono inteligente, y en el inciso (b) la programación inicial y pruebas de la tarea personalizada para el robot seguidor.



(a) Operación remota con bluetooth y aplicación en dispositivo móvil de un robot.



(b) Programación y pruebas iniciales del robot

Figura 16. Ejemplo de un seguidor de línea, su operación y programación vía bluetooth.

Fuente: <http://itszapopan.edu.mx/robotica/>

Con las funciones del prototipo definidas, se propone el esquema general del mismo mostrado en la figura 17.

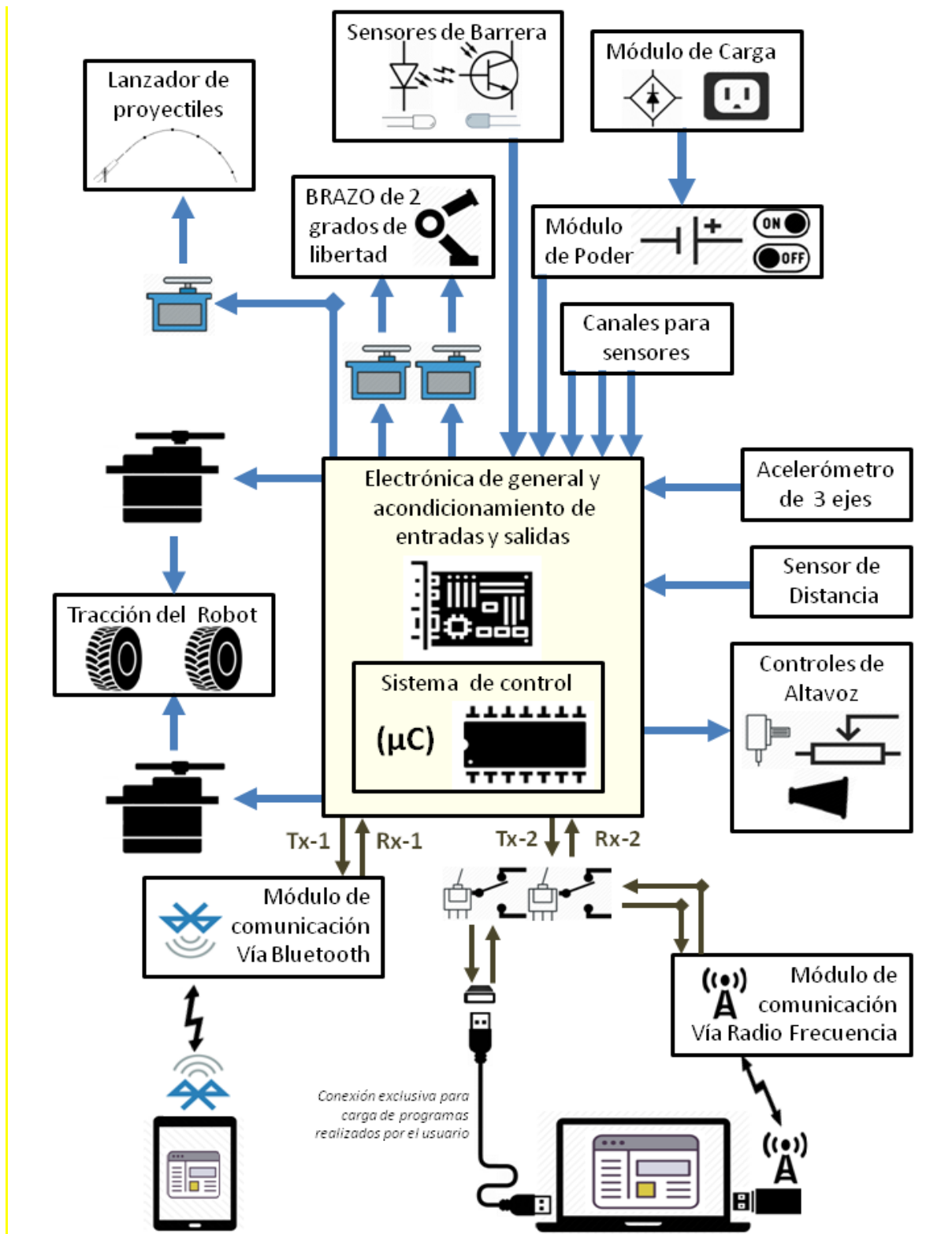


Figura 17. Esquema general de los elementos del robot prototipo

2.2.- Selección de componentes para el prototipo del robot

Ya planteado un esquema general del robot prototipo, lo siguiente es seleccionar los componentes para implementar cada elemento del robot prototipo.

SISTEMA DE CONTROL (μ C)

Este módulo, se encargará de gestionar la transmisión de la información entre las diferentes interfaces gráficas de usuario y los módulos de comunicación inalámbrica. También recibirá, transmitirá datos y ejecutará los comandos correspondientes a cada elemento del robot, provenientes de dichas interfaces. Para el sistema controlador, disponemos de varios sistemas embebidos, pero los que se muestran a continuación, son los de más fácil adquisición en el mercado actual y uso en el ámbito de la robótica.

- **Raspberry Pi.** Según [13], fue una de las primeras placas de desarrollo que soportaban una distribución de Linux sobre un micro ARM y con salida HDMI a 1080p; es una placa enfocada al manejo multimedia o tareas de gestión de alto nivel. Dado que solo cuenta con 10 puertos digitales y 1 PWM, el control de servomotores es complicado; por lo cual su uso en robótica es limitado.



Figura 18. Sistema de desarrollo Raspberry Pi.

Fuente: <http://beembedded.blogspot.mx/2013/07/un-mundo-mas-alla-de-raspberry-pi.html>

- **UDOO.** Esta placa de desarrollo está enfocada en el desarrollo de proyectos de robótica de bajo costo, ya que combina un sistema operativo con las características de un microcontrolador compatible con la programación de alto nivel. Además, en la versión QuadCore podemos encontrar un puerto SATA, memorias RAM DDR3 de 1Gb y módulo WiFi.



Figura 19 Sistema de desarrollo UDOO

Fuente: <http://beembedded.blogspot.mx/2013/07/un-mundo-mas-alla-de-raspberry-pi.html>

- **Tarjeta MSP430 Launchpad.** Es una herramienta de desarrollo y de evaluación para los microcontroladores MSP-430 de Texas Instruments (*value line*). La tarjeta dispone de un socket de 20 pines, comunicación serial (UART), I²C y SPI, memoria flash y RAM de 16 kB y 512Mb respectivamente.

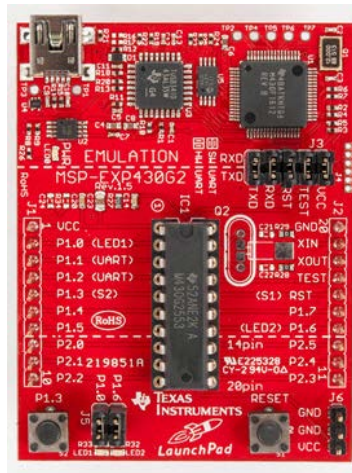


Figura 20. Sistema de desarrollo para el microcontrolador MPS430

Fuente: <http://electronilab.co/tienda/msp430-launchpad-programador-de-microcontroladores-texas-instruments/>

- **Tarjeta Arduino NANO.** Es una placa electrónica basada en el microcontrolador ATmega328P. Cuenta con 14 pines digitales de entrada / salida (de los cuales 6 se podrán utilizar como salidas PWM), 8 entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, pines para I²C y SPI. Además cuenta con:
 - 14 pines digitales I / O (de los cuales 6 proporcionan salida PWM)
 - 8 Pines de entrada analógica
 - Corriente continua para Pin I / O 20 mA
 - Memoria flash 32 kB (ATmega328P)
 - EEPROM 1 kB (ATmega328P)

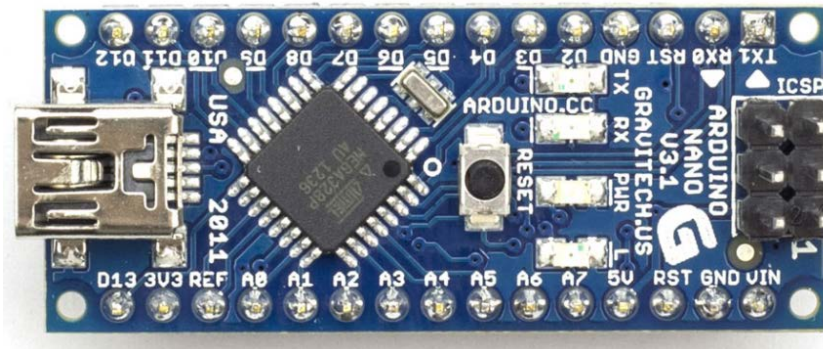


Figura 21. Sistema de desarrollo Arduino Nano para microcontrolador ATMEGA 328

Fuente: <https://www.arduino.cc/en/Main/ArduinoBoardNano>

De los sistemas basados en microcontroladores, para el robot prototipo usaremos el sistema embebido de la plataforma Arduino uno rev 3, basada en el microcontrolador AVR, para el control y programación del robot prototipo, por los siguientes puntos:

- El tamaño reducido permitirá que el sistema electrónico del robot, ocupe un placa pequeña y haya un mayor espacio para realizar las conexiones guiar los cables de los dispositivos del robot.
- En el robot prototipo no se requiere un control preciso del flujo de programa, direccionamiento, espacio de memoria, y mucho menos conocer el lenguaje propio del microcontrolador; es decir, es transparente para el usuario.

- El software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux. La mayoría de los entornos para microcontroladores están limitados a Windows.
- Es una plataforma de hardware y software libre; por lo que resulta de fácil implementación, y los componentes se pueden conseguir a un bajo costo.
- Al ser una plataforma libre, pueden ahorrarse recursos en el procesamiento y en el hardware, pues únicamente se utilizan los recursos necesarios para cada aplicación.
- El sistema embebido arduino, cuenta con varios modelos que se adaptan a las necesidades de cada proyecto, en lo referente al procesamiento, tamaño físico de la placa, cantidad de recursos que se utilizarán en conjunto con la placa, etc.
- Su entorno de desarrollo y programación son muy básicos e intuitivos en su uso, por lo que el usuario, solo requerirá para su manejo, conocimientos básicos de computación.
- Su programación se realiza en lenguaje C, que es un lenguaje de alto nivel. Por consecuencia el usuario deberá tener conocimientos básicos de programación en general, sin meterse al lenguaje máquina o ensamblador para su programación.
- Es el sistema más utilizado para enseñar el uso de microcontroladores y desarrollo de aplicaciones de varias ramas.
- En la robótica aplicada a la enseñanza de ciencias básicas, e introducción al mundo de la robótica, es el sistema embebido más popular, tanto a nivel principiante como a nivel intermedio.
- Al ser el sistema más popular para la enseñanza de ciencias básicas, cuenta con basta información, impresa y electrónica sobre su utilización y funcionamiento.

- Amplia aceptación en el desarrollo de proyectos independientes de control y electrónica; así como para su uso en la robótica, se han desarrollado e implementado diversos dispositivos de comunicación inalámbrica, control remoto, monitorización satelital, mensajería, así como varios sensores y actuadores para trabajar en conjunto con esta plataforma en específico.
- En el desarrollo de proyectos de electrónica que son vigilados y/o controladores por medio de aplicaciones móviles para tabletas y teléfonos inteligentes; suelen integrar el sistema embebido arduino en el desarrollo de dichos proyectos.

Por los puntos anteriores, el sistema embebido arduino uno es el más utilizado en el mundo de la robótica educativa y es el que usaremos para este robot prototipo. Una vez seleccionado el sistema Arduino UNO como sistema de control para el prototipo, escogeremos los componentes restantes con base en la compatibilidad con dicho sistema y la disponibilidad de información de los mismos.

MÓDULO DE COMUNICACIÓN POR BLUETOOTH

El módulo de comunicaciones, tendrá la tarea de transmitir la información entre la aplicación en sistema Android instalada en tabletas o teléfonos inteligentes, y el sistema de control arduino. A través del módulo, el usuario enviará los comandos para accionar el robot y simultáneamente, recibirá los datos obtenidos por los sensores del robot, visualizándolos en la aplicación. En el mercado actual, existen dos componentes compatibles con el sistema arduino uno, que pueden trabajar por medio de bluetooth. El primero de ellos se muestra en la figura 22. Se trata de un *shield* específicamente para conectar en la tarjeta arduino y poder trabajar con bluetooth.

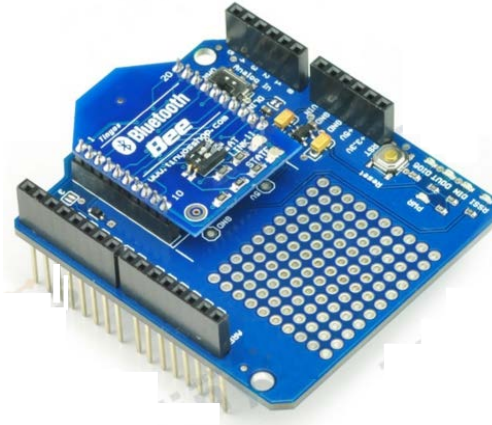


Figura 22. Shield Xbee Arduino Bluetooth

Fuente: http://www.tinyosshop.com/index.php?route=product/product&product_id=339

El *shield* tiene un módulo Bluetooth que puede funcionar como maestro /esclavo; y puede ser aplicado directamente con arduino. Además del bluetooth dispone de un puerto UART para comunicación y de las siguientes características:

- Transmisión a distancia: 20 ~ 30 metros en espacio libre.
- Velocidad de transferencia: 2.1Mbps (máx.) / 160 kbps (típico)
- Características de seguridad: autenticación y cifrado
- 12. Configuración del puerto serie: 1200 ~ 1382400 / N / 8 / 1
- Voltaje de entrada: 3,3 DC / 50 mA
- Tamaño 17: 32 × 24 × 9 mm

El Segundo componente; aunque no fue diseñado para trabajar específicamente con la tarjeta del arduino uno, es empleado en este sistema para poder comunicar por medio de bluetooth, esto debido a que trabaja directamente con el protocolo RS-232, se trata del sistema HC – 05, el cual se ilustra en la figura 24 [16]. Algunas de sus características son:

- Tensión de alimentación: 3.3V a 5V
- Frecuencia: 2.4GHz
- Seguridad: Autenticación y encriptación.
- Velocidad de transferencia: 2.1Mbps (Máx) / 160 kbps (típico).

- Soporta comandos AT para configuración a través de un puerto serie.
- Configuración por defecto para el puerto COM: 9600, N, 8,1
- Dimensiones: 26.9mm x 13mm x 2.2 mm



Figura 23. Módulo bluetooth HC-05

Fuente: <http://tallerarduino.com/2011/12/06/modulo-bluetooth-hc-06-o-gp-gc021-y-arduino/>

De los dos componentes presentados, seleccionaremos el módulo HC-05 para la comunicación bluetooth de nuestro robot, por las siguientes razones:

- El módulo HC-05 al ser muy utilizado en proyectos de electrónica y de robótica, es de bajo costo y alta disponibilidad en el mercado.
- A diferencia del *shield* de bluetooth que debe ser configurado necesariamente en el entorno de desarrollo del propio arduino, el modulo HC-05 puede ser configurado por medios de comandos AT; con cualquier *hyper-terminal* o software que soporte el protocolo RS-232, incluyendo el propio entorno de arduino.

- Con base en la información disponible en varios *blogs* de usuarios de la comunidad arduino, el *shield* de bluetooth presenta problemas en la comunicación y transmisión de datos con la placa arduino; problemática que en el módulo HC-05, generalmente no se presenta.
- Para utilizar el módulo HC-05, la configuración básica solo requiere asignarle un nombre al módulo, un *baudrate*, una paridad y un pin de identificación (código de emparejamiento). Con esto el módulo ya se puede trabajar, como una comunicación serial alámbrica; de tal modo, que este módulo puede ser empleado de manera universal en cualquier aplicación implementada con cualquier microcontrolador que disponga de un módulo de comunicación serial.

MÓDULO DE COMUNICACIÓN POR RADIO FRECUENCIA

El módulo de radio frecuencia, desempeñará la misma tarea que el módulo de bluetooth; con la diferencia que la comunicación será entre el robot y la interfaz gráfica de usuario instalada en un equipo de cómputo. Del mismo modo que con el bluetooth, existe un *shield* de radio comunicación específicamente para arduino, y módulos universales para trabajar con radiofrecuencia bajo el protocolo RS-232. En la figura 24, se muestra dicho *shield*.

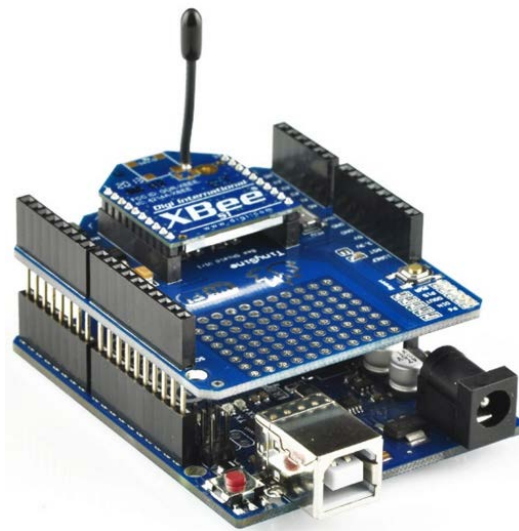


Figura 24. Modulo Xbee de radio frecuencia para Arduino

Fuente: http://www.tinyosshop.com/index.php?route=product/product&product_id=681

El *Shield Xbee* para Arduino es un accesorio que permite conectar de manera rápida los módulos de radiofrecuencia compatibles con Xbee. Algunas de sus características son:

- Alimentación eléctrica de 500mA a 3.3V para funcionamiento.
- Pines soldables con espacio de 0.1" o 2.54 mm para diversos componentes.
- Comunicación UART en hardware o emulación de software.
- Indicadores leds: encendido, RSSI (Intensidad de señal).
- Selección de los pines para la comunicación.
- Su ventaja principal es poder programar el arduino de manera inalámbrica a través de cualquier módulo Xbee; lo cual es especialmente útil cuando se utiliza en conjunto con una plataforma móvil

Las alternativas al *shield* de arduino, son trabajar con módulos universales que trabajan con el protocolo RS-232, y que se pueden usar con el puerto UART del propio microcontrolador; además se caracterizan por trabajar a un frecuencia unificada de 433MHz.

Algunos ejemplos de estos módulos se ilustran en la figura 25; existen diversos fabricantes que ofrecen algunas o todas las versiones de los módulos, y aunque sus diferencias están en el uso de los componentes, alcance y potencia de la señal, todos estos módulos comparten las siguientes características básicas:

- Bajo costo y alta disponibilidad en el mercado
- Fácil uso e implementación para los proyectos electrónicos
- Voltaje de operación típico 5V y una corriente de 45mA
- Frecuencia de operación 433MHz
- Rango de transmisión: 1000 metros en línea libre de visión.
- Ancho de banda: 1.5MHz
- Velocidad de transmisión: < 9.6 kbps

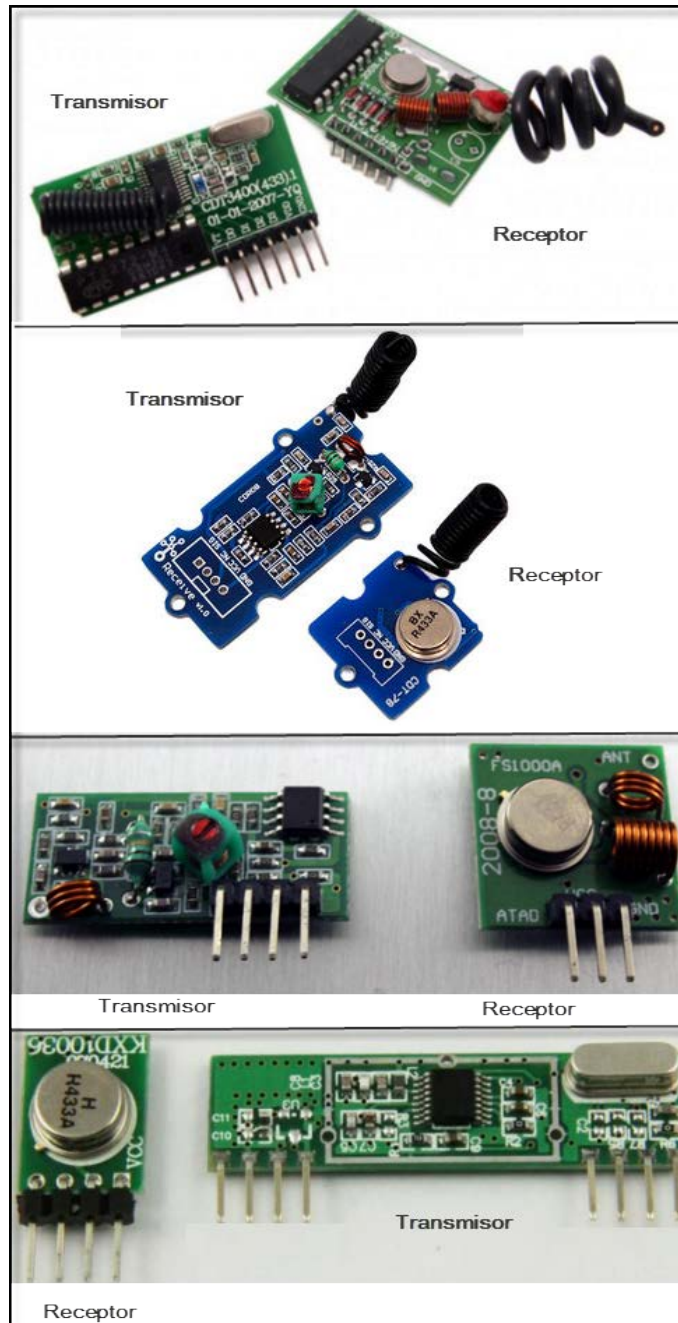


Figura 25. Diferentes módulos de radio frecuencia, tanto receptor como transmisor.

<http://www.electronicaestudio.com/rfestudio.htm>

Otra opción para la comunicación de radio frecuencia, es usar módulos más preparados y eficientes para esta tarea; y el módulo que cada vez es más utilizado, es el APC-220 que se muestra en la figura 26.



Figura 26. Módulo universal para radio frecuencia APC-220

Fuente: http://www.dfrobot.com/index.php?route=product/product&product_id=57#.VtnjSPnhDIU

El módulo de radio APC220 ofrece una solución simple y económica para comunicaciones de datos inalámbricas. Integra un microprocesador de alta velocidad y rendimiento para crear una interfaz UART / TTL transparente al usuario en el empaquetamiento y codificación de datos. Es la solución de bajo costo con mejor rendimiento y alcance en la transmisión [17]. Algunas de sus características son las siguientes:

- Distancia de transmisión de hasta 1000 metros (línea de visión) a 9600 bps
- Interfaz UART / TTL
- Tamaño: 37 x 17 x 6,5 mm (1.46x 0.67X 0.26 in)
- Además el módulo incluye un convertidor de USB a TTL, se puede utilizar para configurar el módulo inalámbrico RF APC220. y para descargar programas

De los módulos de radio frecuencia presentados, seleccionaremos el módulo APC-220 de radio frecuencia por los siguientes puntos:

- La mayoría de los módulos de bajo costo mostrados en la figura 25, tienen la particularidad de poder diseñar e implementar la antena; por tanto las propiedades

en la transmisión se ven modificadas. En el caso respectivo al robot prototipo esto representa una desventaja en tiempo, al ser que existen módulos como el APC-220 que tienen la antena integrada y es totalmente transparente el usuario.

- De experiencias publicadas, las características de estos módulos suelen tener fallas en la comunicación, tales como: pérdida de la señal al atravesar por varios medios de transmisión, ineficacia al recuperar la señal posterior al haberse perdido el contacto o interferencia mutua. En el módulo APC-220 estos problemas suelen no presentarse; es decir, es muy estable y eficaz en la comunicación.
- A diferencia de los módulos de bajo costo, con el módulo APC-220 se pueden manipular las propiedades de la transmisión y la comunicación serial por medio de software; por lo tanto, su uso de hace más eficiente, lo anterior queda ilustrado en la figura 27.

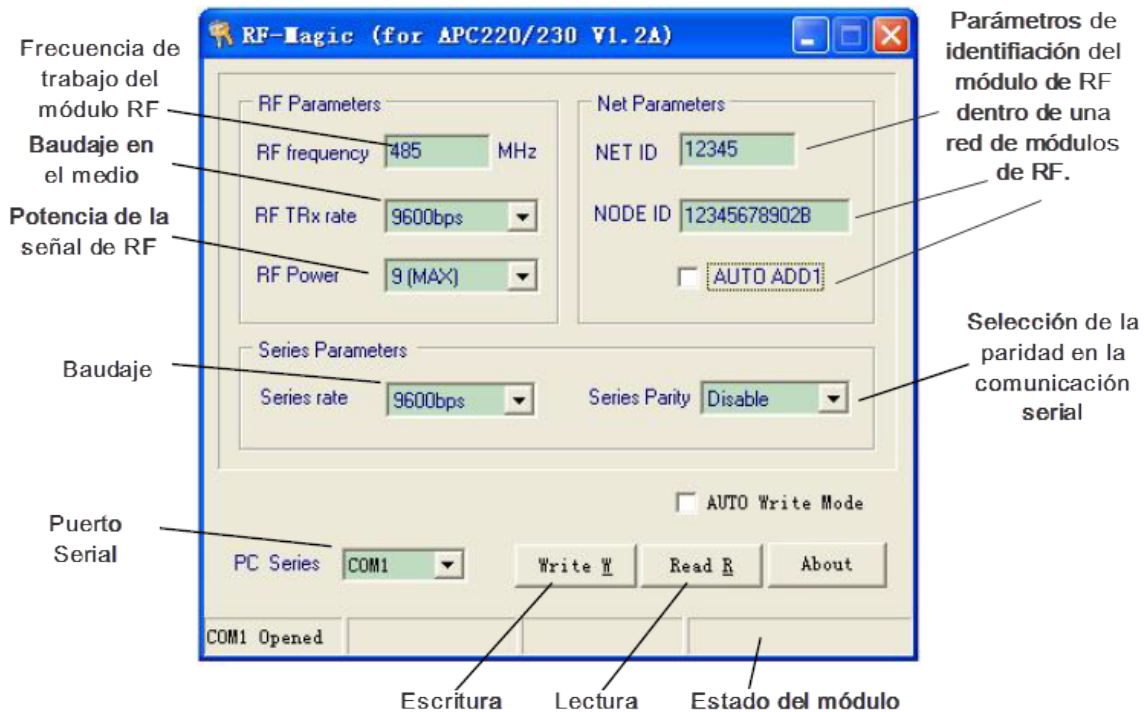
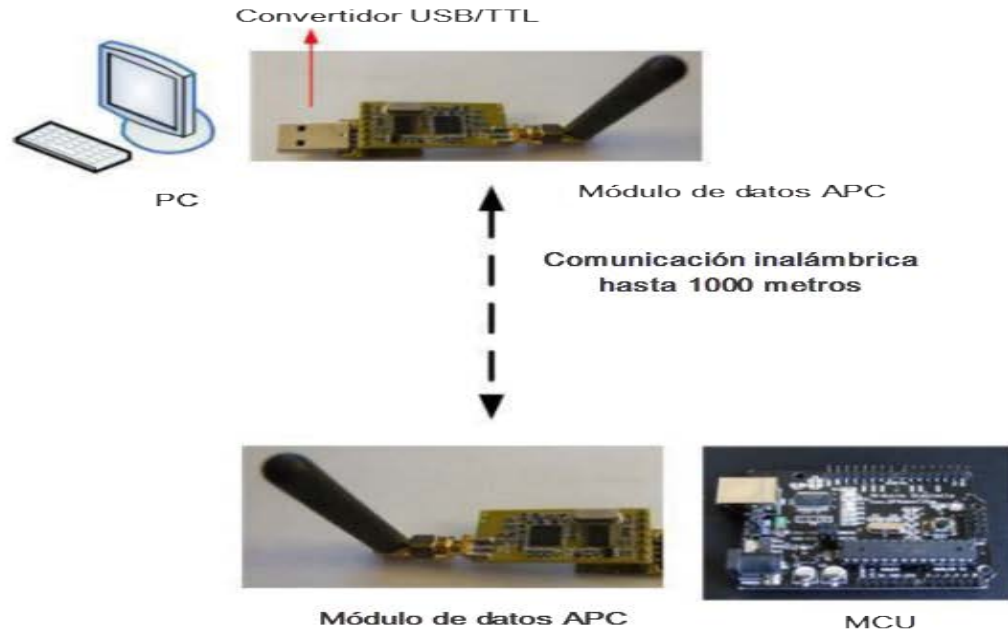


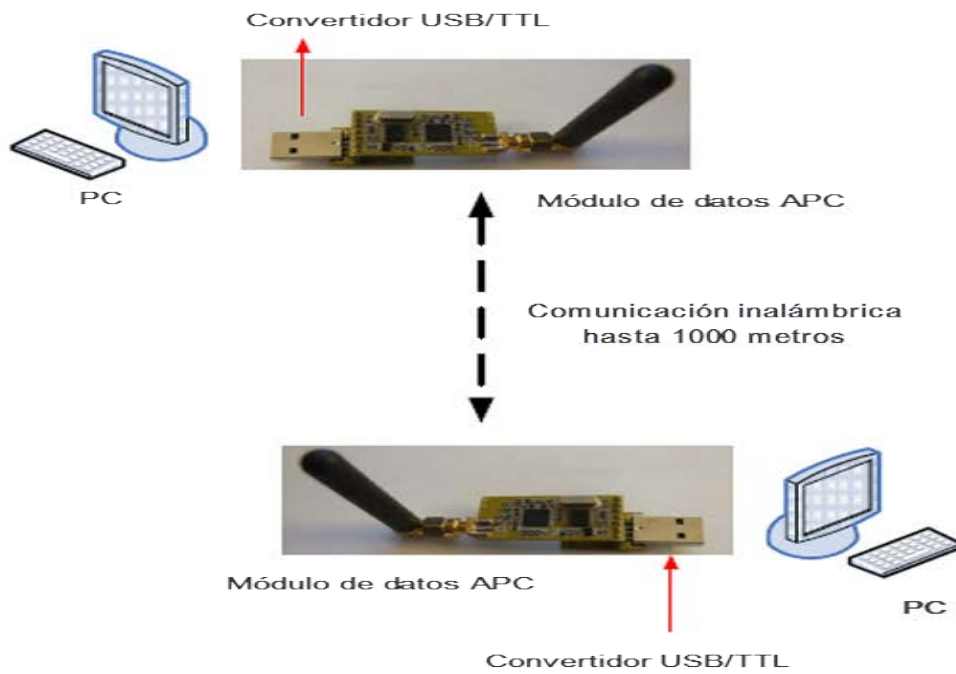
Figura 27. Módulo universal para radio frecuencia APC-220

Fuente: (APC_220 Datasheet, 2008)

- Otra ventaja adicional de usar los módulos APC-220; sobre los módulos universales de la figura 25, es que estos módulos son del tipo *transceiver*; esto quiere decir, que cada pieza que conforma el módulo APC-220, puede fungir como transmisor o receptor de manera automática y totalmente transparente al usuario; este solo tendrá que hacer coincidir los parámetros que se muestran en la figura 27 en ambos módulos, para lograr la comunicación. La ventaja de esto, es que el usuario puede realizar diferentes conexiones: MCU - MCU, PC – PC y PC - MCU, que es la que nos interesa para el robot prototipo y su interacción con la interfaz gráfica.



(a) Conexión entre PC y MCU

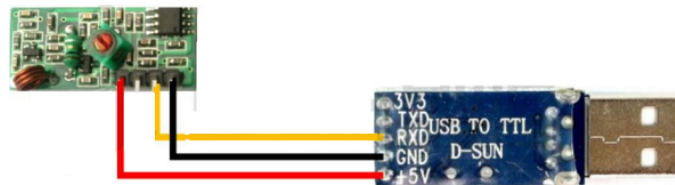


(b) Conexión entre PC y PC

Figura 28. Utilización de los módulos

Fuente: (APC_220 Datasheet, 2008)

- El uso con microcontroladores tanto de los módulos de bajo costo, como del APC-220 se reduce a la conexión básica de los pines de alimentación (5V, GND) y de comunicación serial (Rx, Tx). Pero, para el uso de dichos módulos en un equipo de cómputo, requieren del uso de un convertidor USB/TTL, para poder establecer la comunicación. En el caso del módulo APC-220 dicho convertidor ya viene integrado con sus drivers debidamente configurados. Por lo anterior, con este módulo se transparenta más el uso de la comunicación serial limitándola al envío/recepción de datos; esto queda ilustrado en la figura 28.



(a) Convertidor genérico TTL/USB para los módulos de bajo costo, requiere de la búsqueda y prueba de los respectivos drivers, lo que puede llevar al fallo de comunicación entre ambos.



(b) El módulo APC-220, incluye su propio convertidor TTL/USB con driver, lo que garantiza que el conjunto trabaje correctamente, y facilite la tarea al usuario.



Figura 29. Aditamentos necesarios, para el uso coordinado con PC, de los módulos de radio frecuencia.

Fuente: http://www.naylampmechatronics.com/blog/32_Comunicaci%C3%B3n-Inal%C3%A1mbrica-con-m%C3%B3dulos-de-RF-de.html

- Los módulos de bajo costo solo cuentan; en su debido caso, con las librerías necesarias para su uso conjunto con microcontroladores PIC o Arduino, pero el módulo APC; a consecuencia de su sencillo uso y eficiencia en la transmisión de datos; se han elaborado diversas librerías para poder utilizar el módulo junto a un software de procesamiento de datos, control e instrumentación; por ejemplo se desarrollaron las librerías para el LabVIEW, software que también es utilizado en robótica. El uso de estas librerías se muestra en la figura 30.

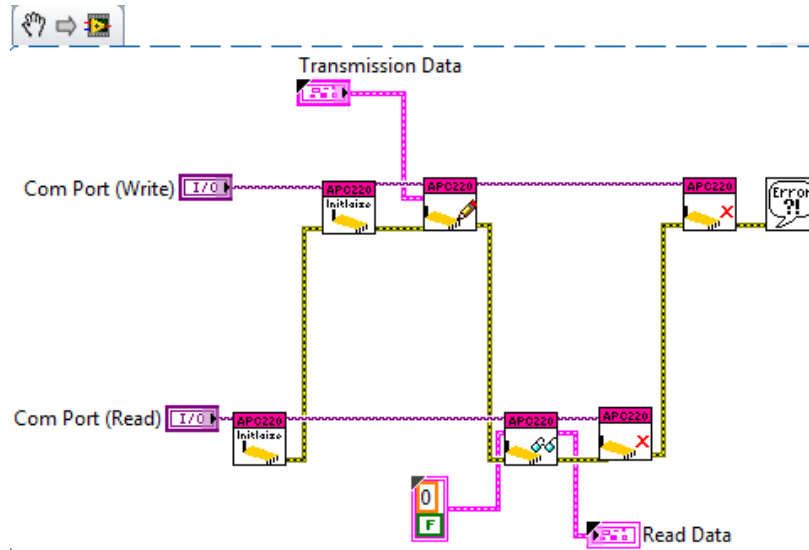


Figura 30. Librería del módulo APC-220 para LabVIEW.

Fuente: <https://decibel.ni.com/content/docs/DOC-32640>

SENSOR DE DISTANCIA

Este elemento, enviará constantemente el valor de la distancia comprendida entre el 1er eslabón del brazo mecánico y el objeto en línea recta más cercano, ubicado dentro de un radio no mayor a 4 metros. Los sensores de distancia disponibles en el mercado para dicho propósito trabajan del mismo modo, el cual se ilustra en la figura 31.

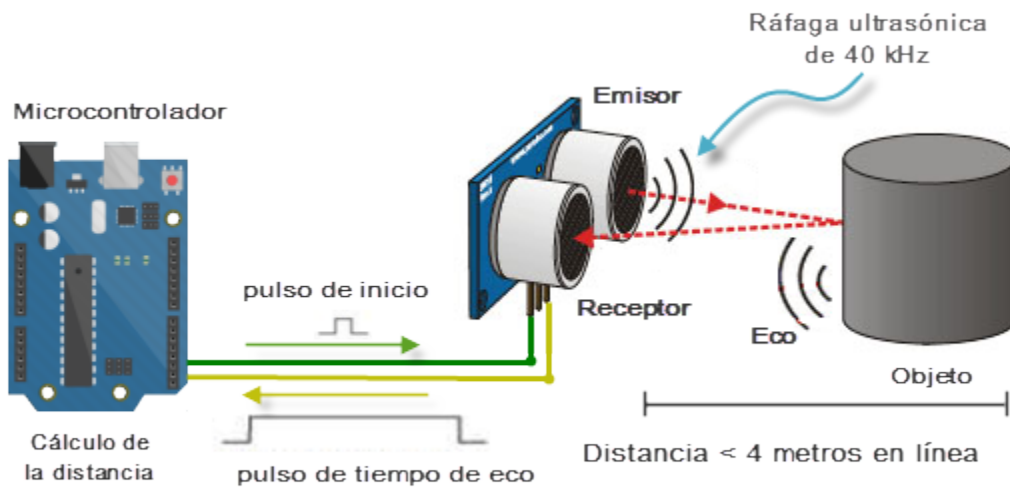


Figura 31. Forma de funcionamiento de los sensores ultrasónicos.

Fuente: <http://www.luisllamas.es/2015/06/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/>

El sensor emite impulsos de ultrasonido casi inaudibles para el oído humano. Estos impulsos emitidos viajan a la velocidad del sonido hasta alcanzar un objeto, entonces el sonido es reflejado y captado de nuevo por el receptor de ultrasonidos. Lo que hace el controlador incorporado es emitir una ráfaga de impulsos, y a continuación empieza a contar el tiempo que tarda en llegar el eco. Este tiempo se traduce en un pulso de eco de anchura proporcional a la distancia a la que se encuentra el objeto. En la figura 32 se muestra este proceso de enviar y contar el tiempo entre los pulsos.

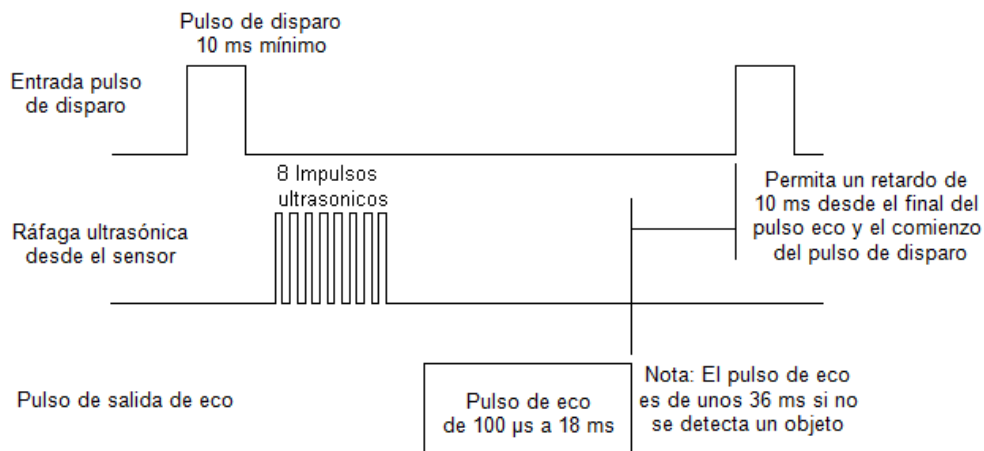


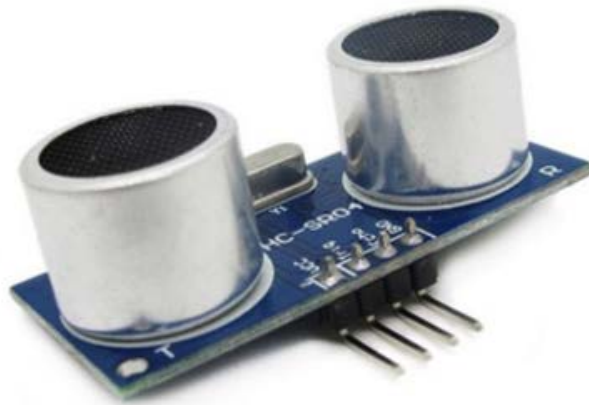
Figura 32. Secuencia de pulsos ultrasónicos para calcular la distancia

Fuente: <http://www.superrobotica.com/S320110.htm>

En el mercado actual, son dos los sensores ultrasónicos que se utilizan junto al arduino para el cálculo de distancia; estos se muestran en la figura 33 y de ellos el más popular en el ámbito de la robótica es el sensor ultrasónico HC-SR04 que se muestra en el inciso (a). El sensor HC-SR04 es el más utilizado en el mundo de la robótica debido a su bajo costo, disponibilidad en el mercado, facilidad de conexiones con el microcontrolador y un programa de funcionamiento sencillo [20]. Algunas de sus características son:

- Tensión de operación: 5V
- Corriente 30 mA Típico. 50mA Máx
- Frecuencia: 40 kHz
- Distancia Mínima: 5 cm
- Distancia Máxima: 400 cm

- Tamaño: 43mm x 20mm x 17 mm
- Peso: 10 g
- Ángulo efectivo <math><15^\circ</math>



(a) Sensor ultrasónico HC-SR04



(b) Sensor ultrasónico Sharp

Figura 33. Sensores ultrasónicos para la medición de distancia en robótica.

Fuente: <http://www.electronicaestudio.com/sensores.htm>

ACELEROMETRO DE 3 EJES

En robótica, el acelerómetro es utilizado para diversas aplicaciones como: equilibrio de un robot, detectar impactos con objetos cercanos, auto-pilotaje en aerodelismo, censar la fuerza G experimentada por el robot, medir la velocidad y aceleración del mismo. El funcionamiento de un acelerómetro se basa en variar la capacitancia de dos placas metálicas, para lograr la medición. Dos de las formas más comunes para lograr esto son:

En el mercado actual, existen diversos acelerómetros que utilizan alguna de estas formas para medir aceleración, y la mayoría de ellos son compatibles con el sistema arduino y están basados en el *chip* MMA7361L. Algunas de sus características son las siguientes:

- Con base en [21], este sensor posee salidas de voltaje análogas y sensibilidad ajustable (± 1.5 g o ± 6 g), también tiene una salida digital que indica cuando la tarjeta está en caída libre. Contiene reguladores de voltaje lineales de 3.3V que permiten que el voltaje de entrada esté entre 2.2V y 16V. Además de detección de 0g.
- El robot prototipo no se desplazará a grandes velocidades, ni experimentará grandes fuerzas; por lo tanto, el rango de medida de este acelerómetro es suficiente para los experimentos donde se empleará el prototipo; además un rango de medida, garantiza una mayor sensibilidad en las lecturas.
- Al ser de salida analógica el procesamiento y la programación resultan más sencillas, puesto que no hay que programar alguno protocolo de comunicación, para interpretar los datos del sensor.
- Aunque el costo del acelerómetro es bajo, para la construcción del robot prototipo no se tuvo que adquirir; dado que fue prestado de manera provisional para la realización de esta tesis.

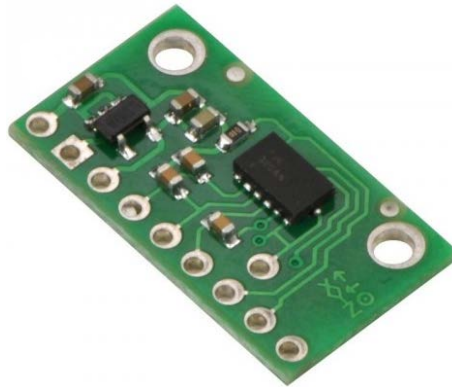


Figura 34. Acelerómetro MMA7361L

Fuente: <http://www.picaxe.biz/tienda/producto/po1251/1/acelerometro-de-3-ejes-1-5-6g-p2-c68->

BRAZO ROBÓTICO Y LANZADOR DE PROYECTILES

El sistema del brazo robótico se compone de 2 eslabones, en consecuencia, requerirá de 2 actuadores mecánicos para su movimiento. En robótica suelen emplearse servomotores para lograr este movimiento. Lo que normalmente se denomina servo, también se conoce como servomotor, motor servo controlado o incluso servo rc, y en cualquier caso hacen referencia a un sistema compuesto por un motor eléctrico, un sistema de regulación que

actúa sobre el motor y un sistema de sensor que controla el movimiento del motor. Es precisamente este sensor el que marca la diferencia con respecto a un motor controlado electrónicamente, ya que gracias a la información obtenida del sensor, se puede saber en tiempo real lo que está haciendo un motor. La figura 35 muestra un esquema básico de un servo motor.

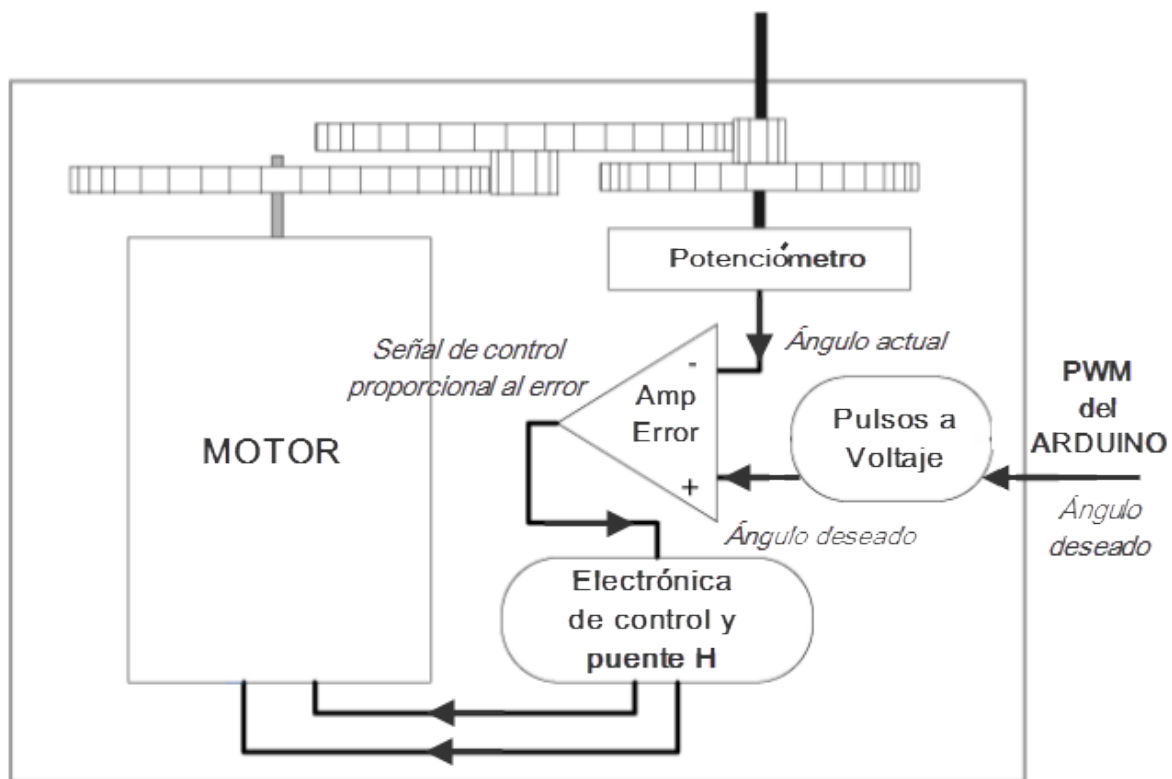


Figura 35. Esquema genérico de un servomotor.

Fuente: <https://alonsodub.wordpress.com/2012/06/>

Supongamos que en una plataforma giratoria movida por un motor, deseamos controlar su giro de forma que esta se pare en determinadas posiciones. Si utilizamos un motor y un circuito de control, la única forma que tenemos de hacerlo sería alimentando el motor durante un tiempo, en función del número de grados que queremos que gire. Este sistema además de ser muy poco preciso; carece de una realimentación al circuito de control que le indique el giro de la plataforma. Otra desventaja, son las variaciones de voltaje y carga que experimenta el sistema con respecto al tiempo, hacen que el sistema sea impráctico.

Lo anterior se puede resolver con posiciones definidas y con algún tipo de sensor como un interruptor final de carrera, o bien un sensor óptico que nos indique cuando la plataforma alcanza la posición correcta. En este caso el sistema funcionaría correctamente, pero siempre limitado a las posiciones preestablecidas. Para solventar esta problemática, a los servomotores se les añade un sensor (generalmente un potenciómetro) que proporcione información sobre el giro y velocidad de la plataforma. De esta forma el sistema de control cuenta con la información necesaria para actuar sobre el motor, con independencia de la carga que tenga el servomotor.

Por todo lo anterior, los servomotores son los más empleados en la robótica; no requieren de electrónica adicional y solo se tiene que modular el ancho de pulso con el microcontrolador; como se aprecia en la figura 38; también se ilustra el rango de movilidad del servo solo es de 180° o media vuelta, pero en el caso de nuestro robot prototipo, dicho rango de giro es suficiente dado que en los experimentos básicos de tiro parabólico no se necesitan cubrir los 360° , en los giros de cintura y codo.

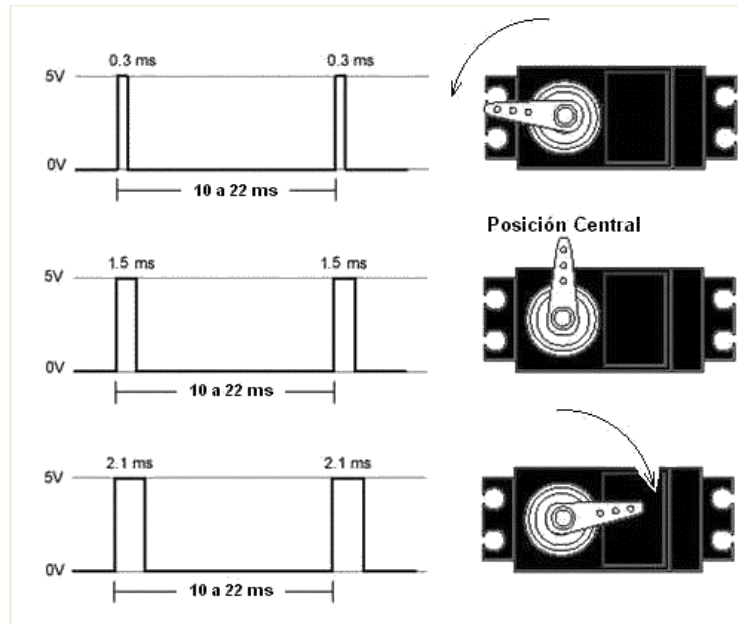


Figura 36. Control de un servomotor, por medio de modulación de ancho de pulsos

Fuente: http://robots-argentina.com.ar/MotorServo_basico.htm

En el diseño conceptual del robot, no se contempla que tenga un efector final; es decir, los servomotores solo tendrán la finalidad de mover al brazo mecánico y sistema lanzador de proyectiles. Por lo mismo, los servomotores que utilizaremos en el brazo serán de bajo par y dimensiones para lograr que todo el conjunto de lanzador y brazo sea pequeño y ligero. Dichos servos los podemos apreciar en la figura 37.



Figura 37. Micro servomotores para el desplazamiento del brazo

Fuente: <http://sonrobots.com/arduinos/motores-servomotores-y-motores-paso-a-paso/>

Este tipo de servomotores cuentan con un par de salida de: 1.5 kgf·cm (0.1471 Ncm) y tienen un paso de aproximadamente 2.5° - 3° por cada pulso de control que se manda; resolución suficiente para el ámbito de aplicación del robot. Además el servo, solo requiere de una sola línea de control para lograr dicho paso.

Para el caso del lanzador de proyectiles, también se empleará un servomotor como el mostrado en la figura 37. Su función será la de retener y liberar el resorte que empujará al balón fuera del lanzador. Un bosquejo de este sistema lo podemos ver en la figura 38.

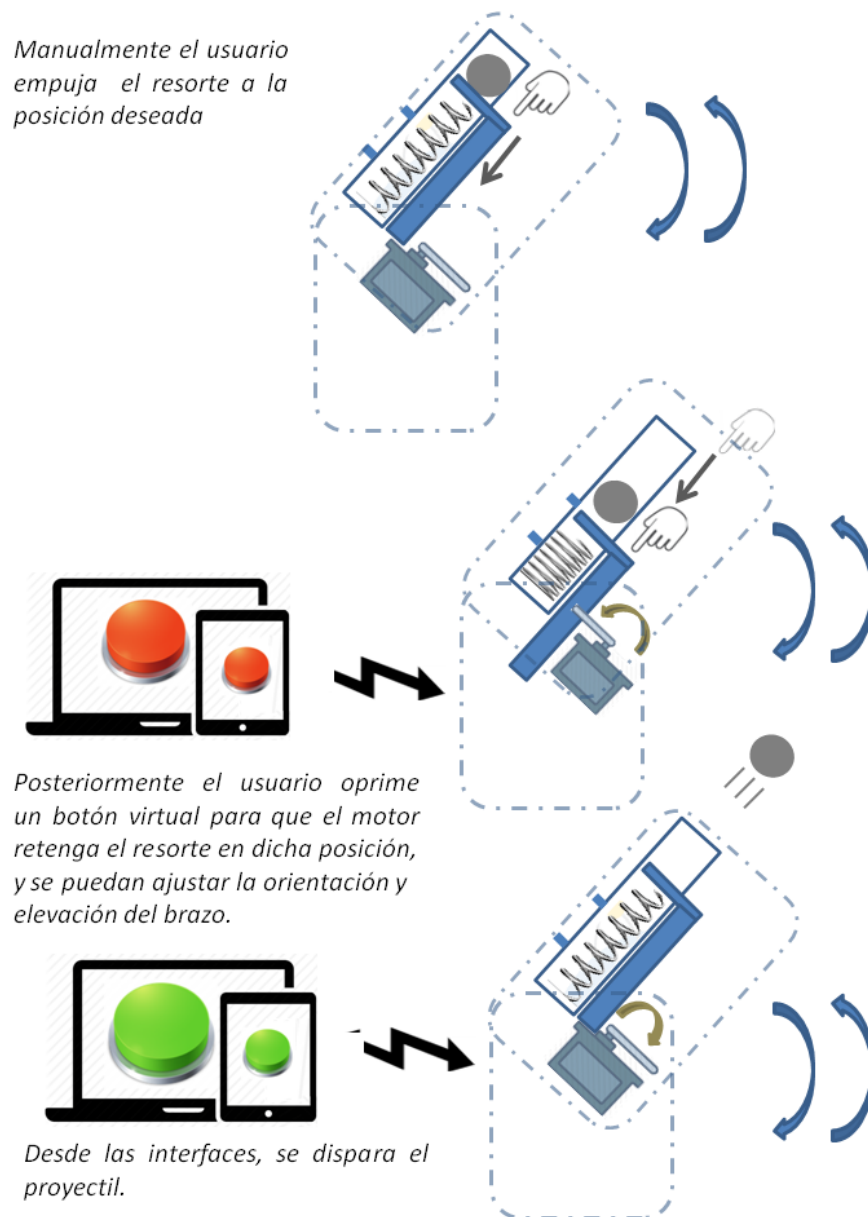


Figura 38. Esquema general del funcionamiento del lanzador de proyectiles.

En primera instancia, el resorte se comprimirá a una de las 3 posiciones marcadas y retenido por el servomotor cuando el usuario; desde alguna de las interfaces gráficas disponibles (en PC o dispositivo móvil Android), envíe el comando. Posteriormente desde la misma interfaz gráfica, el usuario manipulará la orientación y elevación del brazo mecánico; así como el disparo del balón de forma manual o automática.

TRACCIÓN DEL ROBOT

Con respecto a la tracción de un robot, suelen emplearse alguno de los servomotores mostrados en las figuras 39 y 40.

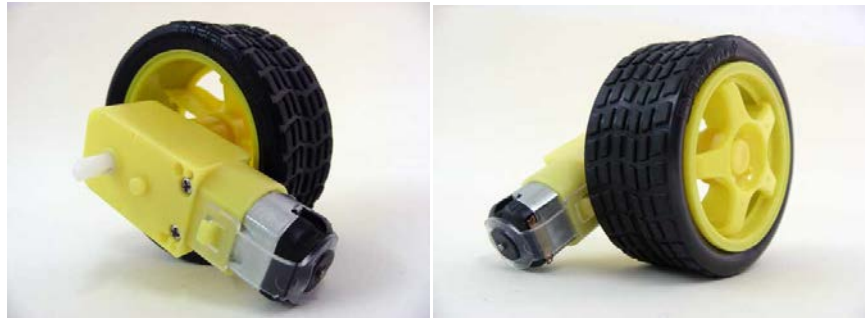


Figura 39. Servomotor para robótica que incluye rueda.

Fuente: <http://www.electronicaestudio.com/>

Algunas de las características de estos servomotores son las siguientes:

- Tensión de alimentación: de 3 a 6 V CD.
- Par de salida: 1.0 kgf·cm (0.0981 N·m)
- Peso: 50 g



Figura 40. Servomotores más utilizados en robótica que incluye rueda.

Fuente: <http://www.electronicaestudio.com/>

Algunas de sus características son:

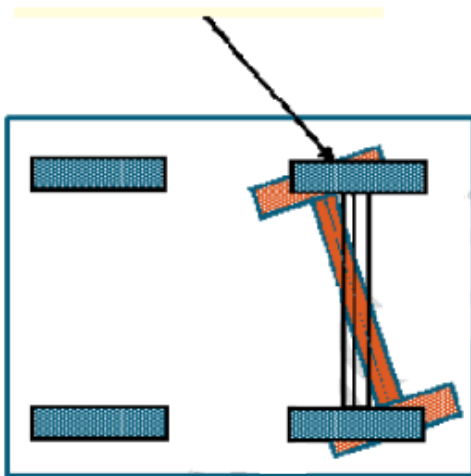
- Peso: 38 g
- Par de salida a 6V: 4.1 kgf·cm (0.4021N·m)
- Par de salida a 4.8V: 3.2 kgf·cm (0.3138 N·m)
- Medidas: 1.6" (40mm) X 0.8" (20mm) X 1.4" (44mm)

Nosotros usaremos el servomotor de más alto par, puesto que solo tenemos estimados del peso total del robot; en consecuencia, el diseño y las piezas serán lo más ligeros posible, para no sobrepasar el par que el servomotor nos puede otorgar. A diferencia de los servos de la figura 39 que son de rotación continua, los servos de la figura 40 solo giran 180°; por lo tanto tendremos que modificarlos para poder obtener la rotación continua, sin descuidar el par de salida.

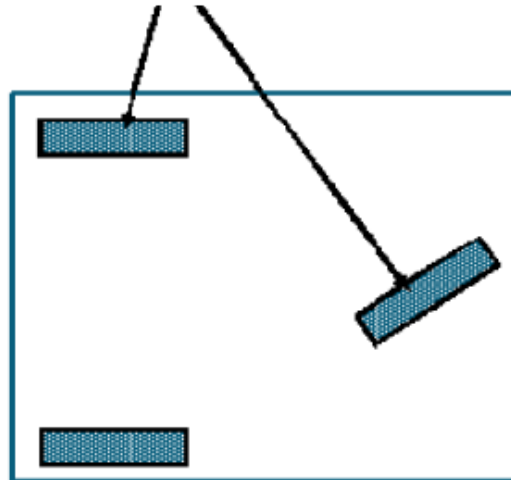
Otro punto a considerar en la tracción del robot, es el número de servo motores y la distribución de los mismos; con la finalidad de conocer cómo será la dirección y tracción de los mismos. En la figura 41 se ilustran las distribuciones más comunes de los motores, para nuestro robot prototipo usaremos la configuración del inciso (c), puesto que es la más equilibrada y nos brindará un mejor soporte en la distribución de los componentes del prototipo.

La figura 42 muestra como se llevará a cabo el control de cada servo de la configuración del inciso (c), para lograr que el robot tenga ciertos movimientos. Dichos movimientos se logran a partir de las diferentes combinaciones en el giro de los servomotores. La desventaja estaría en los movimientos limitados que tendría el robot, sin embargo las ventajas serían el uso de solo 2 motores y dos ruedas locas; por consecuencia el ahorro económico en la compra de los mismos y la ausencia de modificaciones mecánicas en las piezas del robot.

(a) Tracción y dirección en el mismo eje



(b) Tracción y dirección independientes



(c) Dirección diferencial con una o 2 ruedas locas

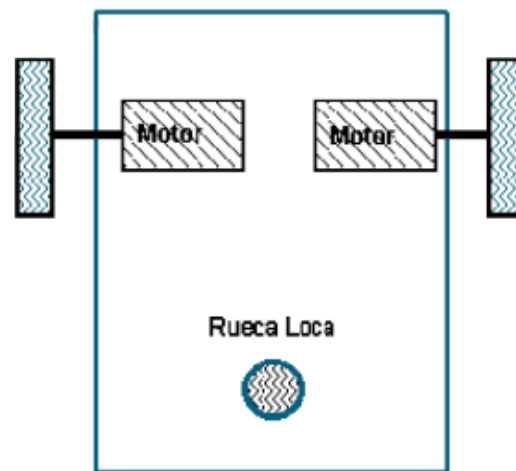
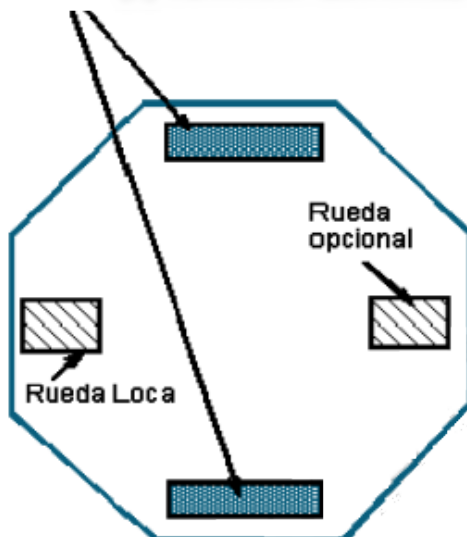
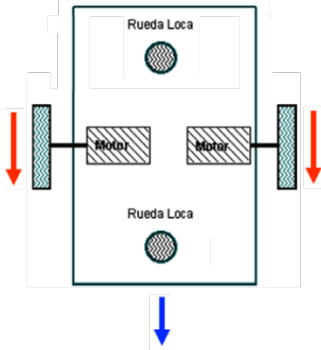
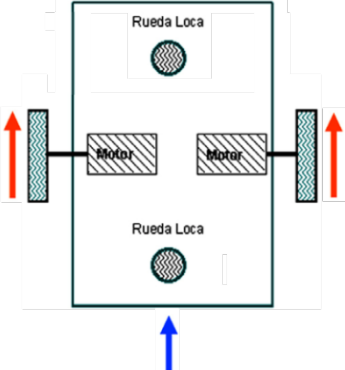
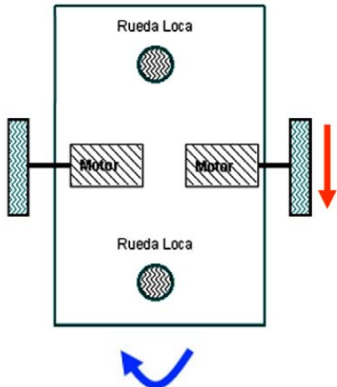
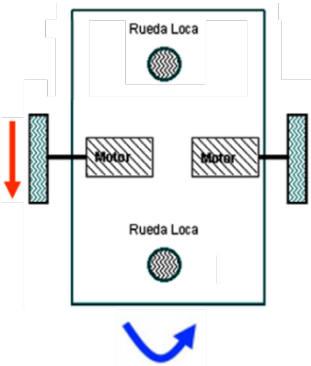
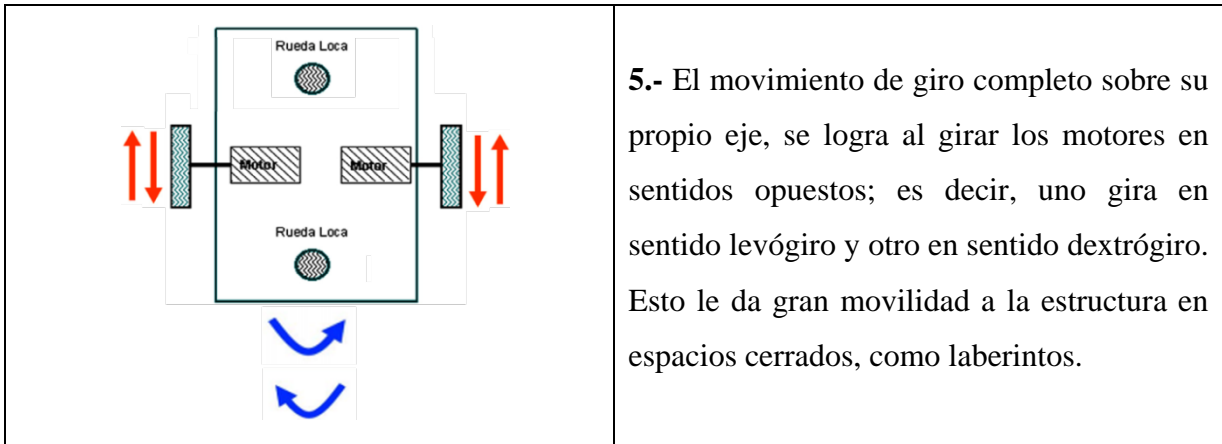


Figura 41. Distribución de los servomotores, para la movilidad del robot.

Fuente: http://platea.pntic.mec.es/~mhidalgo/docEducaBot/04_EducaBot_ServomotoresMovilidad02.pdf

	<p>1.- Movimiento hacia adelante, se rotan los dos motores en sentido dextrógiro. Esto provoca un movimiento rectilíneo suponiendo que los dos motores son exactamente iguales.</p>
	<p>2.- Movimiento hacia atrás, se rotan los dos motores en sentido levógiro. Esto provoca un movimiento rectilíneo suponiendo que los dos motores son exactamente iguales.</p>
	<p>3.- Movimiento de giro a la derecha, se rota el motor izquierdo en sentido dextrógiro y el motor de la derecha en sentido levógiro. Esto provoca un movimiento de giro a la derecha de la estructura</p>
	<p>4.- Movimiento de giro a la izquierda, se rota el motor izquierdo en sentido levógiro o parado y el motor de la derecha en sentido dextrógiro. Esto provoca un movimiento de giro a la izquierda de la estructura.</p>



5.- El movimiento de giro completo sobre su propio eje, se logra al girar los motores en sentidos opuestos; es decir, uno gira en sentido levógiro y otro en sentido dextrógiro. Esto le da gran movilidad a la estructura en espacios cerrados, como laberintos.

Figura 42. Combinaciones de giro en los servomotores, para movilidad del robot.

Fuente: http://platea.pntic.mec.es/~mhidalgo/docEducaBot/04_EducaBot_ServomotoresMovilidad02.pdf

2.3.- Elementos mecánicos

Con todo lo anterior, ya tenemos definidos los elementos con los cuales implementar las funciones para nuestro robot prototipo y la manera como se interconectarán los mismos, por lo tanto, el siguiente paso será la construcción de las piezas mecánicas del mismo. En el apartado 2.1 se mencionó que una de las aplicaciones de la robótica es en el estudio del tiro parabólico y las leyes del movimiento; por consecuencia el diseño del robot será el brazo mecánico montado sobre una base con servomotores para que todo el conjunto se pueda desplazar. También se incluirán las adaptaciones mecánicas necesarias, para poder integrar todos los elementos electrónicos mencionados. El bosquejo del robot, se aprecia en la figura 43, en esta se puede ver la base general con los 2 servomotores, las 2 ruedas locas, la base para el robot prototipo; así como el propio brazo mecánico y el espacio para la electrónica del robot.

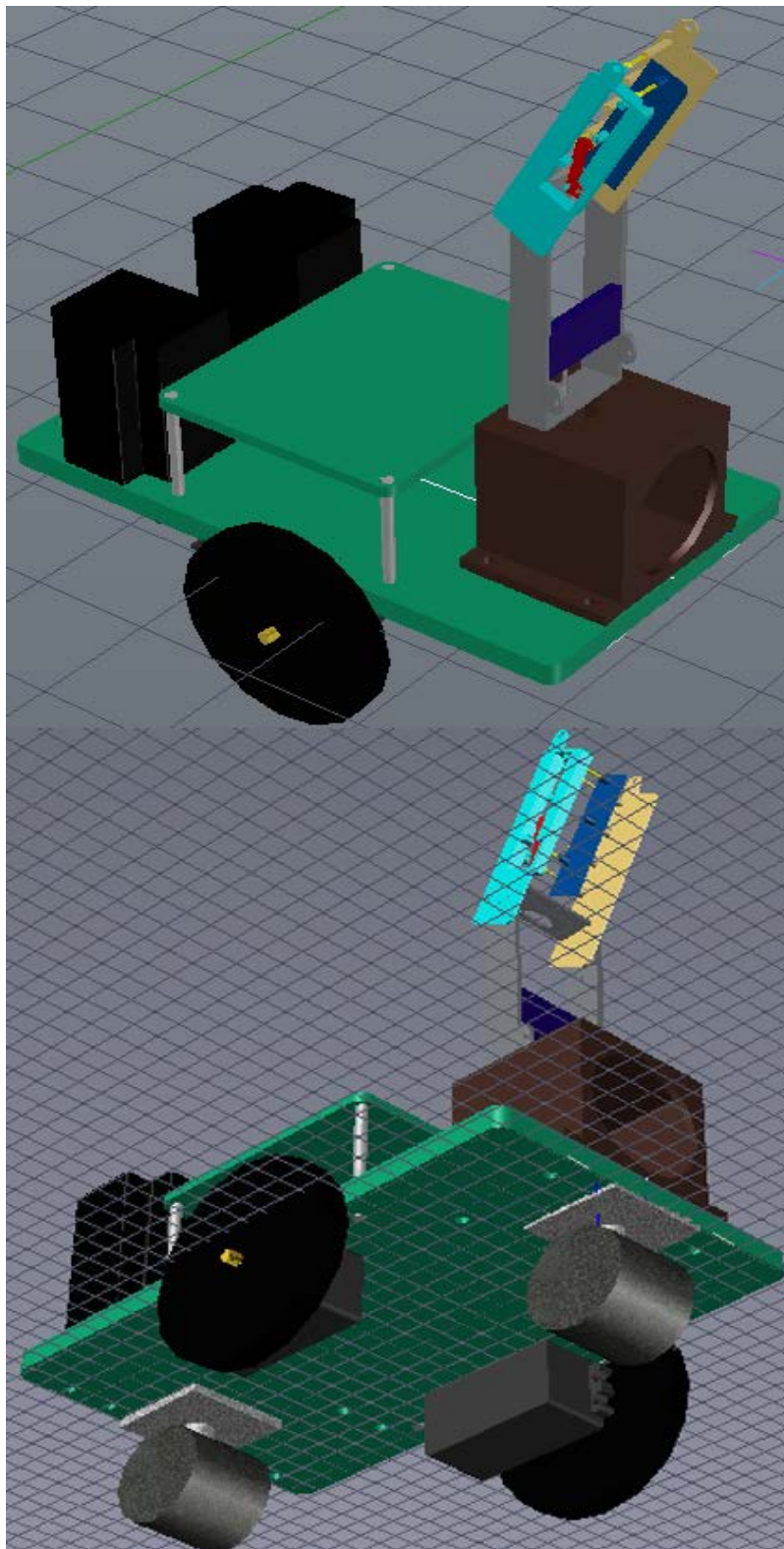
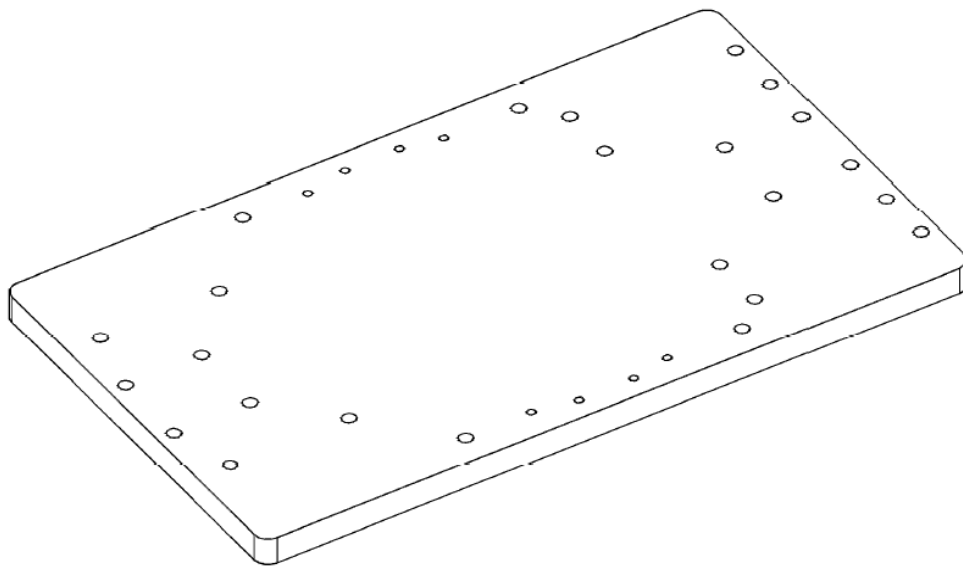
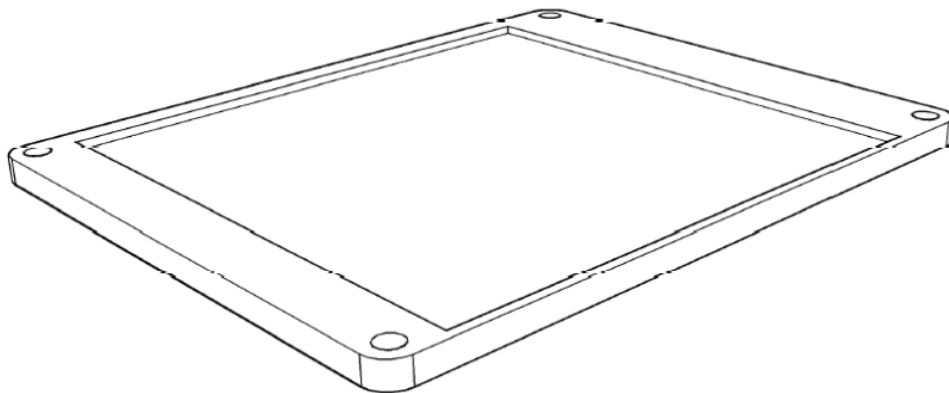


Figura 43. Bosquejo en 3D del robot prototipo.

La figura 44 ilustra las dos primeras piezas que componen el robot. La primera de ellas en el inciso (a), es la base general del robot, en ella se colocará los servomotores junto a las ruedas locas para el desplazamiento del robot; además de contener la base para el brazo móvil con su lanzador, la placa con electrónica, y la fuente de alimentación. La segunda pieza en el inciso (b) es la placa de conectores; esta contendrá los aditamentos necesarios para que el usuario realice las conexiones de los sensores, control de la comunicación, observe los led's indicadores y manipule otros controles del robot; así mismo servirá como placa de resguardo a la electrónica general del robot, buses de comunicación y energía.



(a) Placa Base del robot prototipo



(a) Placa para conexiones

Figura 44. Bosquejo de la base general del robot y de la placa de conexiones

La tercera pieza a diseñar es la base del robot; será una caja que albergará el servo motor encargado de girar todo el brazo mecánico. También por esta pieza pasarán todos los cables de los servomotores y sensor de distancia. Se le añadirá una bocina para futuras prácticas de acústica.

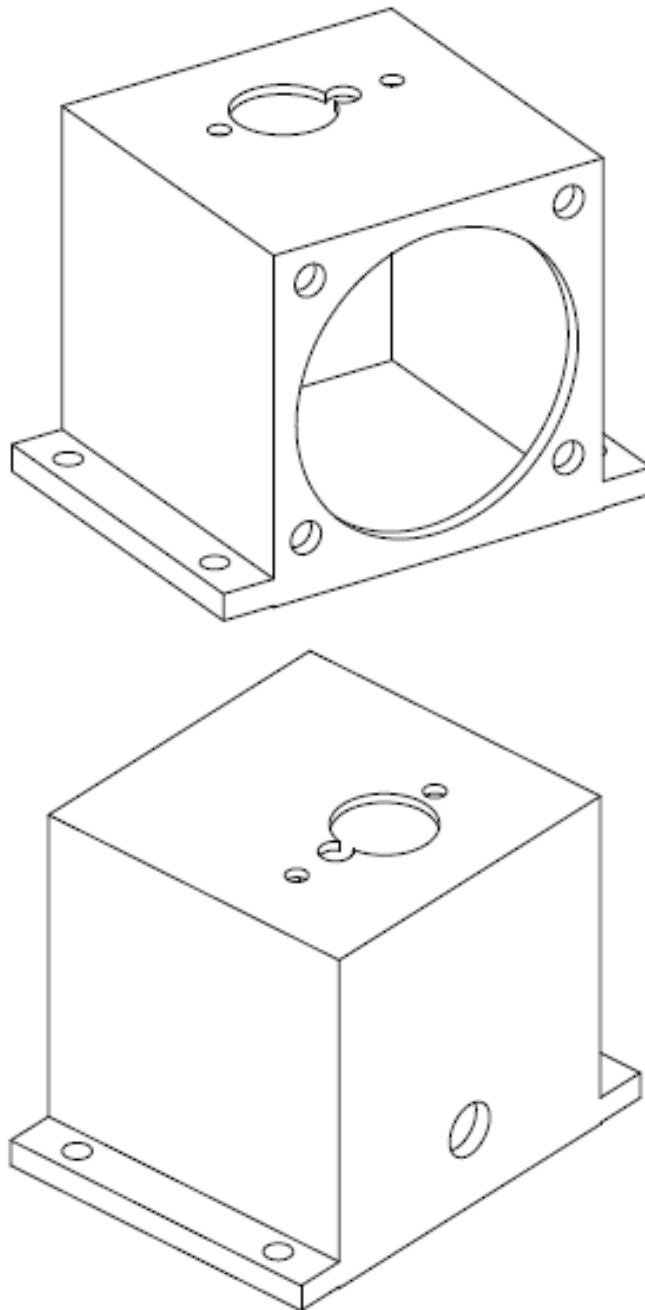


Figura 45. Bosquejo de la base para el brazo mecánico.

La siguiente pieza a diseñar, es el 1er eslabón del brazo mecánico. Este tendrá la pieza que sostendrá al sensor de distancia y el servomotor que le dará movilidad al segundo eslabón del brazo mecánico. Se diseña de tal manera que el material empleado sea mínimo con la finalidad de aligerar el peso del brazo mecánico. El bosquejo del 1er eslabón lo podemos apreciar en la figura 46.

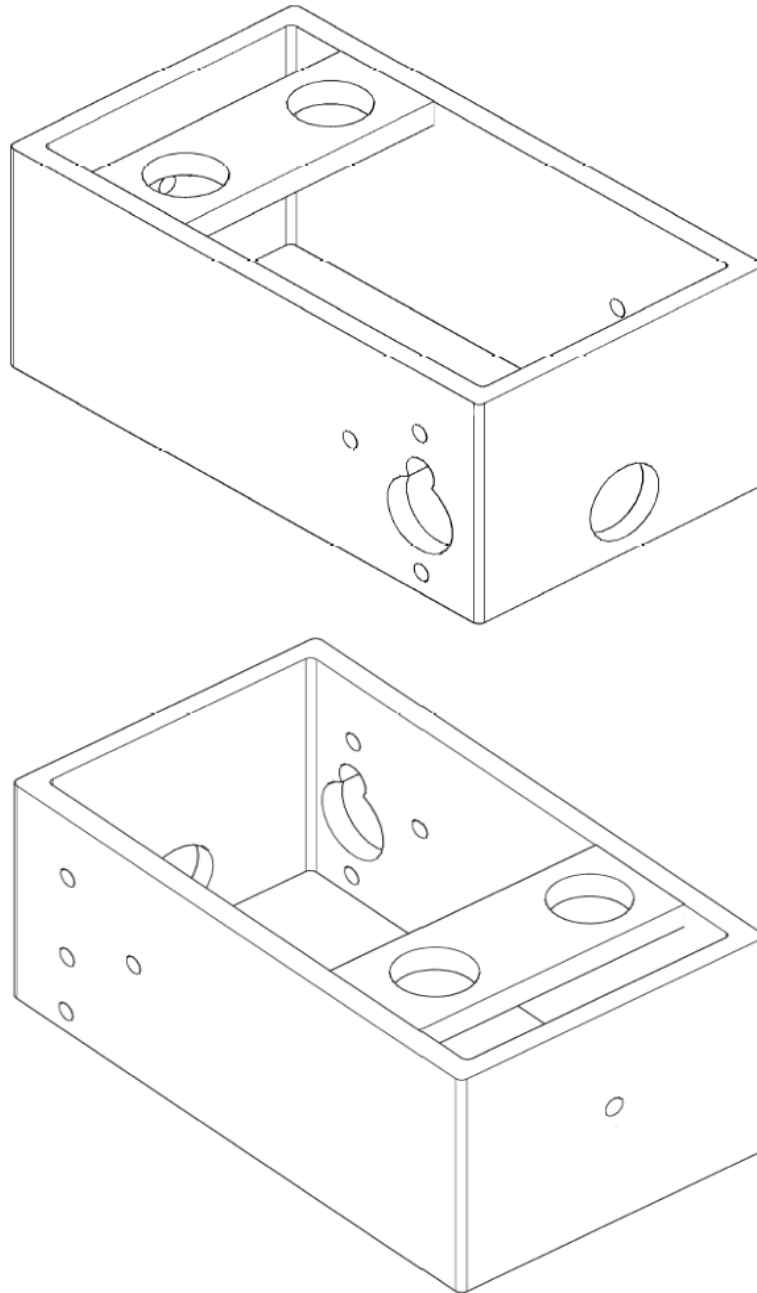


Figura 46. Bosquejo del primer eslabón del brazo mecánico.

2.4.- Maquinado y construcción de los elementos mecánicos

Ya con las piezas definidas, lo siguiente será el maquinado de las mismas. Para la fabricación de nuestro prototipo, usaremos una placa de PVC de $\frac{1}{2}$ " (12.7 mm) para fabricar la base general del robot y la placa de conexiones. Mientras que para el resto de las piezas utilizaremos una placa de PVC $\frac{1}{4}$ " (6.35mm). Se escogió este material debido a que; su precio es barato. El proceso de maquilado se hará con taladro fresador, torno, prensa y la herramienta necesaria para el proceso. El taladro fresador se ilustra en la figura 47 y el torno en la figura 48. Se seleccionó este método de fabricación, puesto que se tenía la facilidad de acceso a las máquinas y herramientas.



Figura 47. Taladro fresador y prensa que se usará en el proceso de fabricación de las piezas



Figura 48. Torno que se usará para el proceso de fabricación de las piezas.

2.5.- Elementos de programación

Los elementos de programación; serán el software que el robot prototipo, usará para su funcionamiento y que el usuario utilizará para la interacción con el mismo.

1.- Entorno de desarrollo integrado de Arduino (IDE): Este entorno mostrado en la figura 49, es el software que se necesita, para programar y monitorizar al microcontrolador arduino. El usuario, usará este software para crear, cargar y probar sus programas personalizados con el robot prototipo.

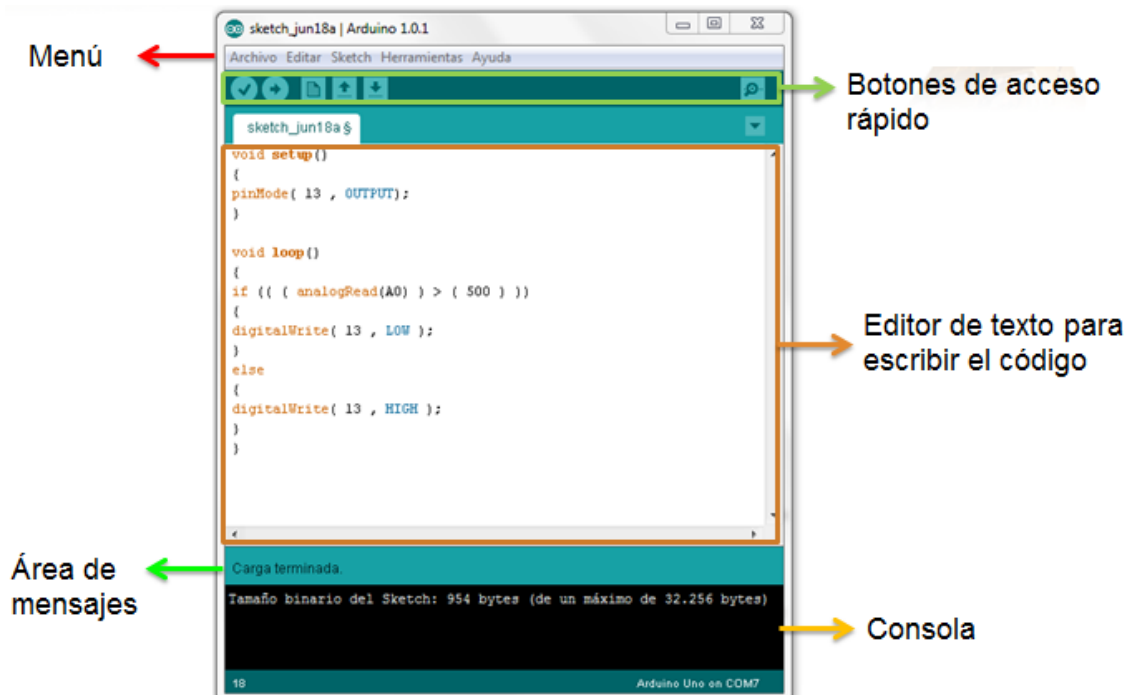


Figura 49. Entorno de Desarrollo Integrado (IDE) para Arduino

Fuente: <http://solorobotica.blogspot.mx/2012/07/programacion-de-arduino-elide-de.html>

2.- Se utilizará el entorno de desarrollo eclipse, puesto que el mismo software ofrece los *plugs* necesarios para poder crear las aplicaciones para sistema Android.



Figura 50. Entorno de desarrollo Eclipse para realizar la aplicación de dispositivos móviles

Fuente: <http://androideity.com/2011/07/06/%C2%BFque-necesito-para-programar-en-android/>

3.- Interfaz gráfica de usuario para equipo de cómputo: Para esta interfaz, se utilizará el entorno de desarrollo de NetBeans 8.0.

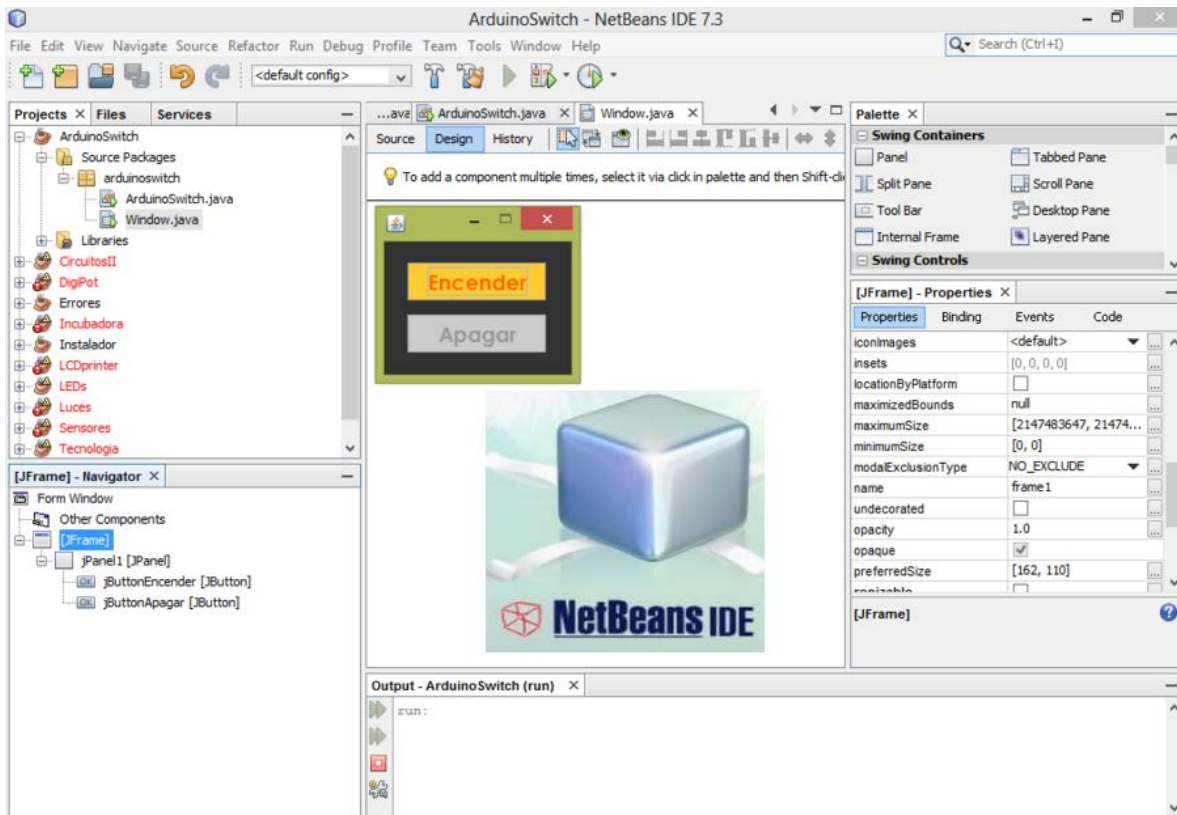


Figura 51. Entorno de desarrollo de NetBeans para realizar la aplicación de equipo de cómputo.

Fuente: <http://panamahitek.com/tag/netbeans/>

En general se seleccionó JAVA como lenguaje de programación; puesto que es el lenguaje más utilizado en la creación de interfaces de dispositivos móviles; además de que existe una gran documentación, para realizar interfaces gráficas con arduino. Lo anterior facilita que la interfaz gráfica de la PC y de dispositivos móviles, tengan la misma estructura; con lo que se ahorra tiempo en la programación. Para uso futuro de podría utilizar JAVASCRIPT, para crear una aplicación web del robot; o bien, integrar *applets* a las funciones del robot. Todo el software utilizado es de código abierto y no requiere ningún tipo de licencia para su uso.

3.- IMPLEMENTACIÓN

Capítulo 3.- Implementación

3.1.- Integración del Hardware

En este capítulo se describirán las pruebas o modificaciones realizadas a los componentes electrónicos, la realización de las interfaces gráficas de usuario; así como la fabricación y ensamblado de las piezas mecánicas del robot. También se mostrará la integración final de los componentes electrónicos y mecánicos.

- **Pruebas individuales y modificaciones al sistema electrónico**

Estas pruebas consistieron en verificar el funcionamiento correcto de cada dispositivo; para ello, se creó un código en la IDE de Arduino y con ayuda del programa “terminal.exe” mostrado en la figura 52, se monitorea el envío/recepción de datos. Este último fue de mayor utilidad en la prueba de los módulos inalámbricos; debido a que el monitor serial del Arduino queda deshabilitado, cuando no se utiliza el cable USB.

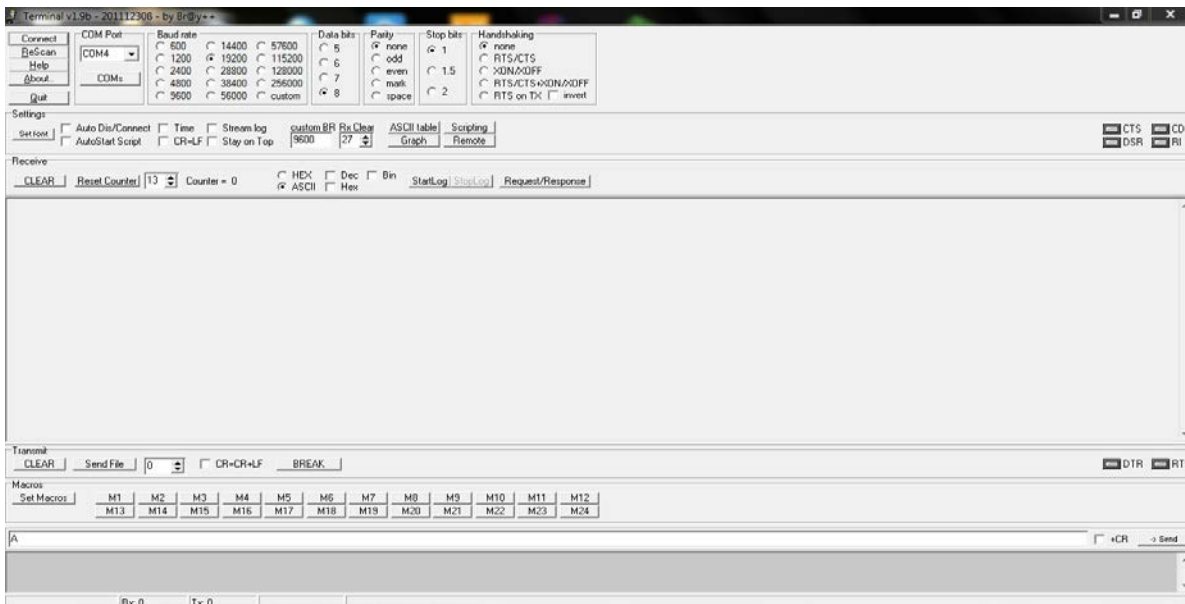


Figura 52. Programa Terminal.exe que se usará en las pruebas de comunicación serial, con los módulos de Bluetooth HC-05 y RF APC-220.

- **Servomotores de tracción y del brazo mecánico:** Los servomotores de tracción fueron los únicos que requirieron de modificaciones. Estas fueron la remoción del vástago y sensor de posición que limitaban mecánicamente el giro continuo del servomotor. Se sustituyó dicho sensor por dos resistencias de precisión. En la figura 53 se pueden apreciar estas modificaciones.

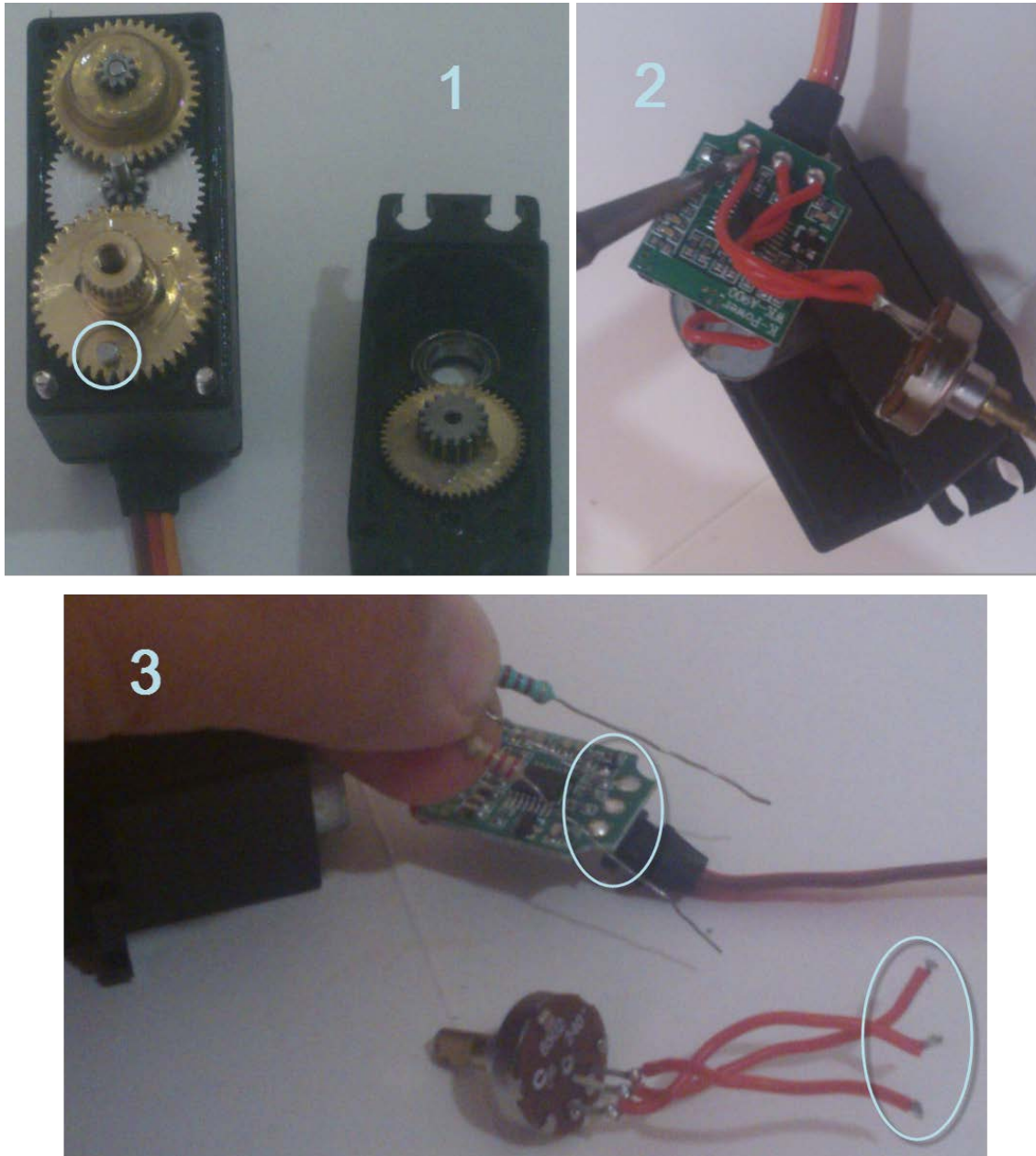


Figura 53. Modificaciones realizadas a los servomotores de tracción.

Con la conexión mostrada en la figura 54, probé que los servomotores giraran en sentido dextrógiro, levógiro y se detuvieran al variar el ancho de pulso de las señales de control.

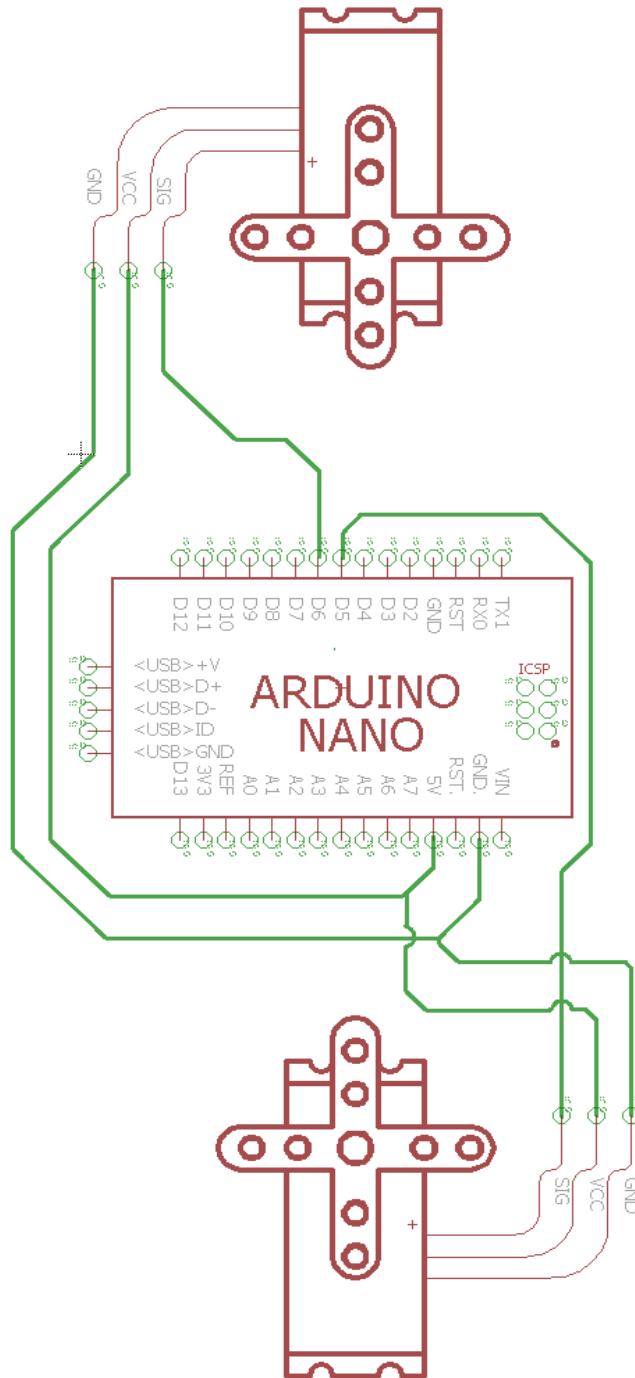


Figura 54. Conexión de servomotores con Arduino.

A diferencia de los servomotores de tracción; los servomotores del brazo no requirieron modificaciones. Se probó de forma similar a los servomotores de tracción; con modificaciones al programa de Arduino para conocer la posición en la que se ubicaba el servomotor.

- **Módulo de radiofrecuencia APC-220.** Las pruebas que se le realizaron fueron:
 1. Instalación de controladores.
 2. Configuración de comunicación entre los módulos emisores/receptores.
 3. Transmisión de datos entre los dispositivos comunicados con dichos módulos y con el programa Terminal.exe

Al conectar el convertidor USB/TTL al equipo de cómputo se detecta el dispositivo como se muestra en la figura 55.

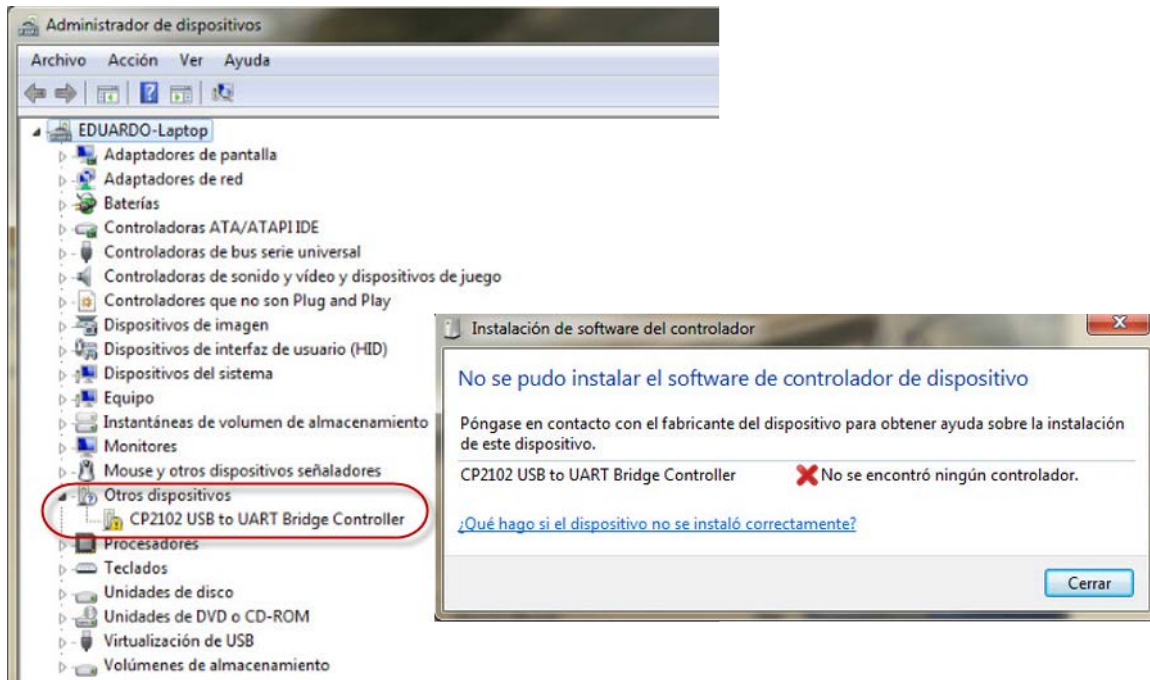


Figura 55. Instalación automática de los controladores del convertidor USB/TTL

Para instalar correctamente el dispositivo, se ejecuta el instalador *CP210xVCPInstaller_x64.exe* proporcionado por el fabricante. Con la instalación, se crean controladores virtuales necesarios para el funcionamiento de los módulos; así como un puerto serial (puerto COM para la comunicación con de la PC).

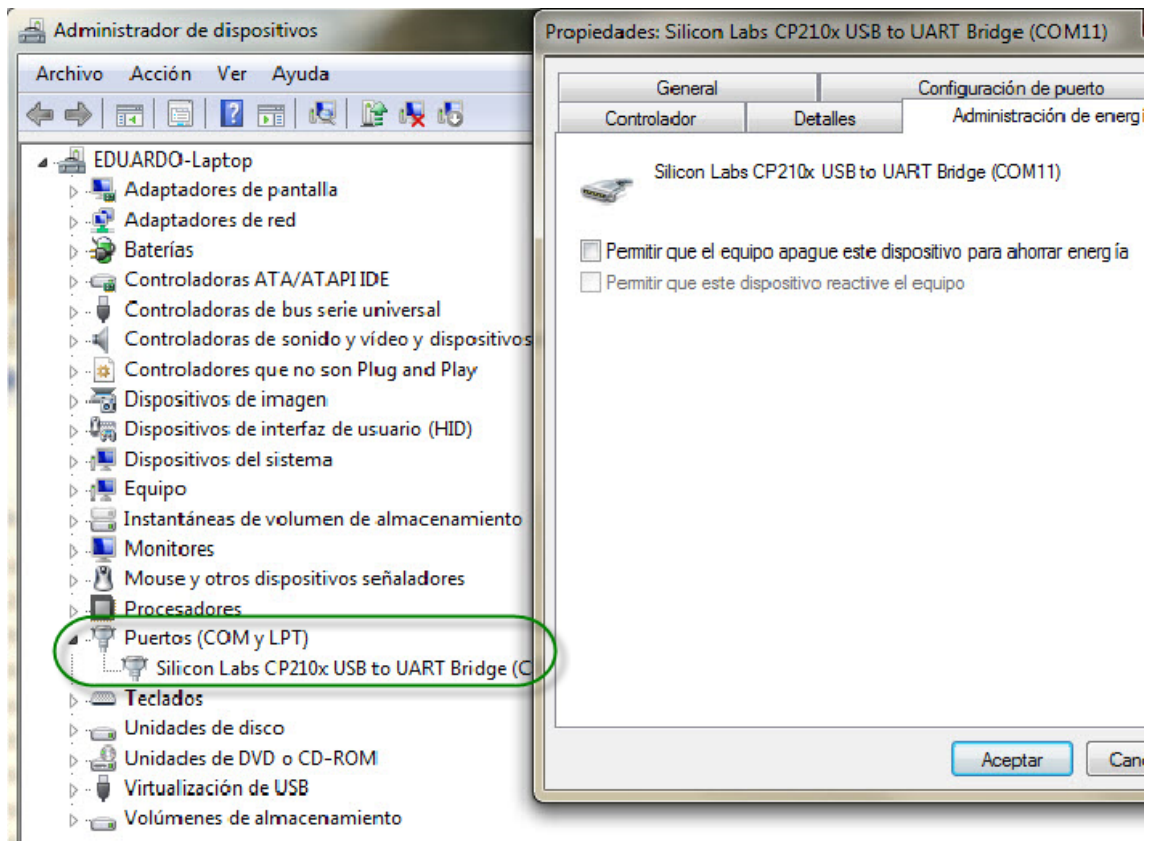


Figura 56. Instalación correcta de los controladores del APC -220

También se instala el programa RF- Magic (*APC22X_V12A.exe*) que sirve para configurar los módulos emisores/receptores del APC-220. Sólo el controlador *CP2102 USB to UART Bridge Controller* se deberá instalar en los equipos de cómputo donde se desee utilizar el robot.

Al ejecutar el software APC22X_V12A.exe se abre la ventana mostrada en la figura 59. Este software permite controlar los parámetros de radio y de transmisión serial; además muestra el puerto COM virtual donde está conectado el convertidor USB/TTL. En la parte inferior de la ventana solo aparecen las leyendas: “COM2 *Opened*”, “*Found device*” y “APC22X series” y un recuadro en blanco junto a dos botones que dicen: “ Write W” y “Read R”. El recuadro en blanco se debe a que los módulos tienen la configuración de fábrica.

Para configurar los módulos emisores/receptores, se conectan los módulos en el convertidor USB/TTL, se ajustan los parámetros de la comunicación y se presiona el botón “Write W”.

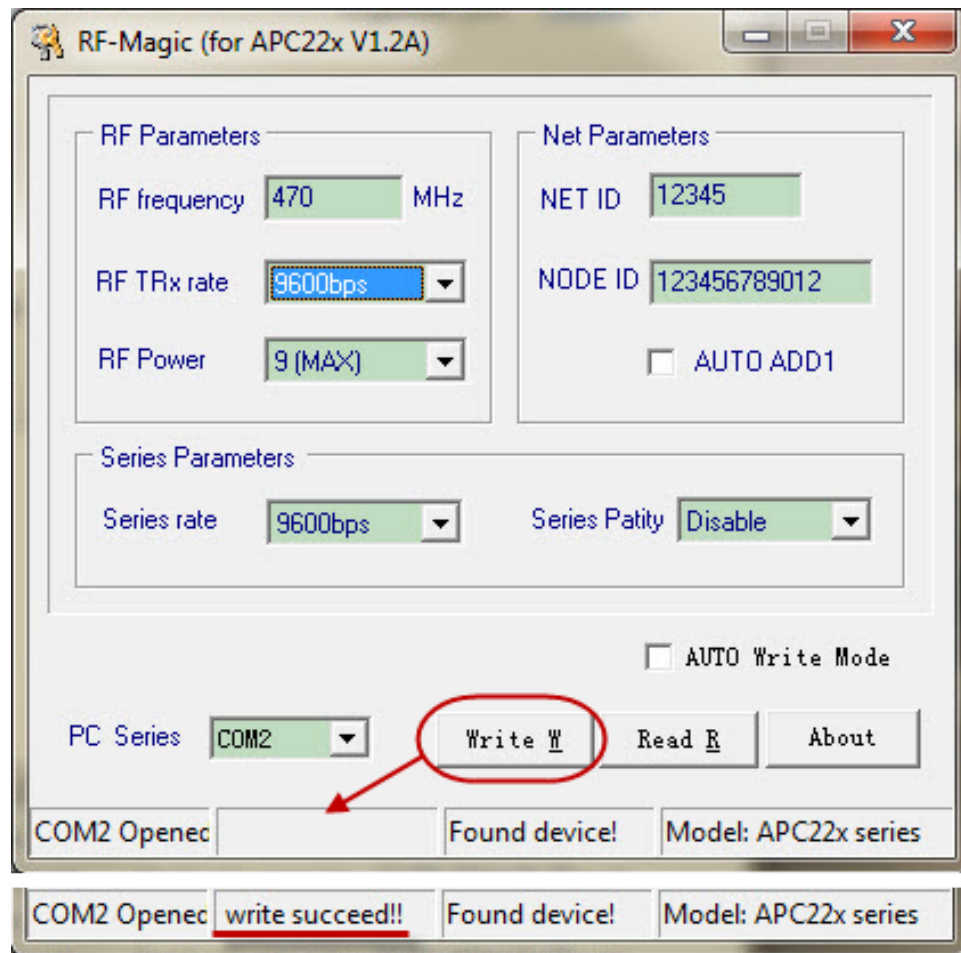


Figura 57. Configuración de los módulos emisores/receptores del APC-220 por medio de software.

Para comprobar la transmisión de datos; se conectaron los módulos RF y el Arduino como se muestra la figura 58.

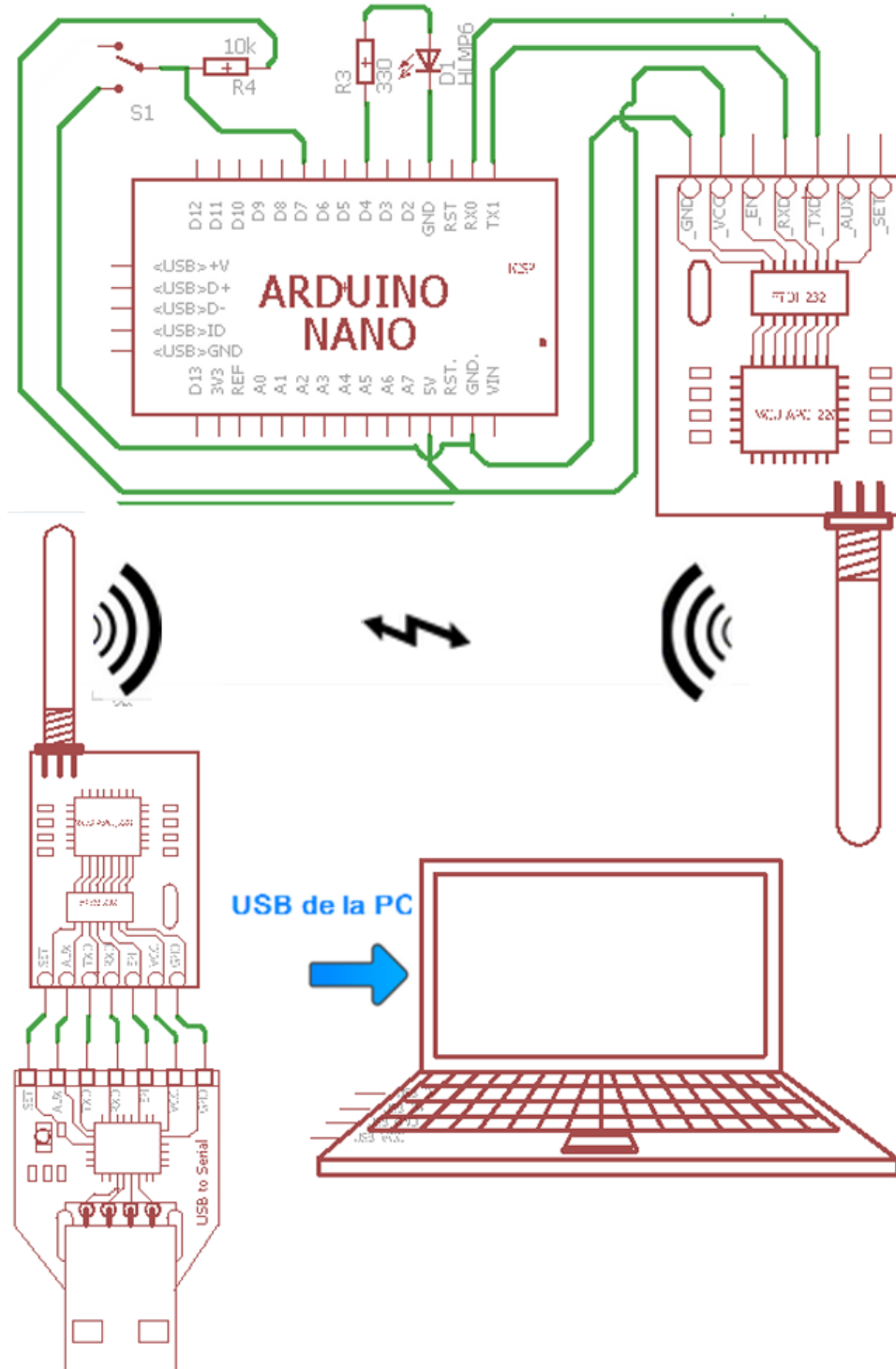


Figura 58. Conexión de los módulos APC-220 con arduino y PC.

El led sirvió de indicador y el interruptor de controlador. El programa de Arduino envió el estado del interruptor (accionado o no accionado) y encendía el led al recibir el comando (carácter “A” o “B”). En el programa terminal.exe se monitorizan los datos enviados por Arduino y se envía el comando.

- **Módulo Bluetooth HC-05:** Se configuraron con comandos AT los parámetros básicos del dispositivo: tasa de transmisión, código de emparejamiento, paridad y nombre. Se conectó el módulo bluetooth como se muestra en la figura 59.

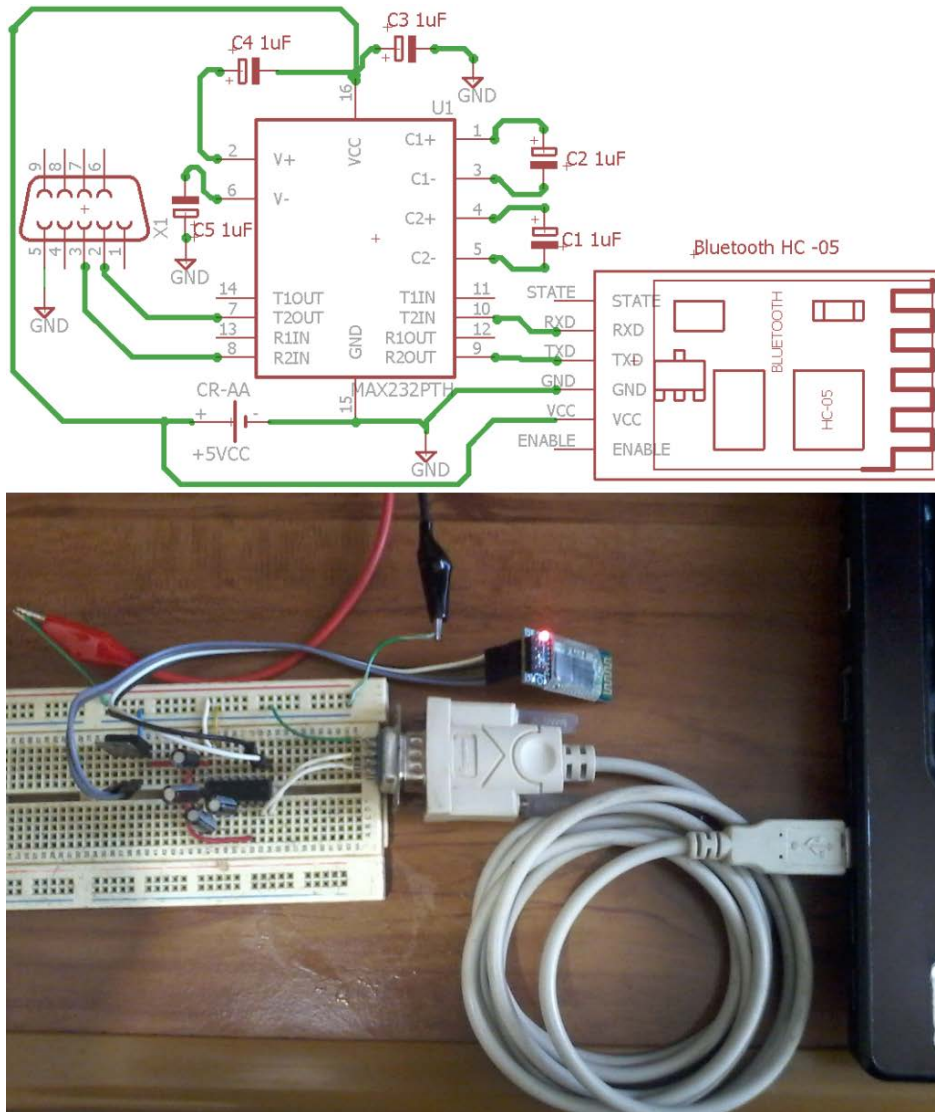


Figura 59. Esquema del MAX232 para programar el módulo HC-05 y el circuito armado

Con la aplicación ALDBluetoothATConfig.exe se programan los elementos básicos del HC-05 sin ejecutar el programa terminal.exe y escribir los comandos AT individualmente. Una pantalla de la aplicación se muestra en la figura 60.

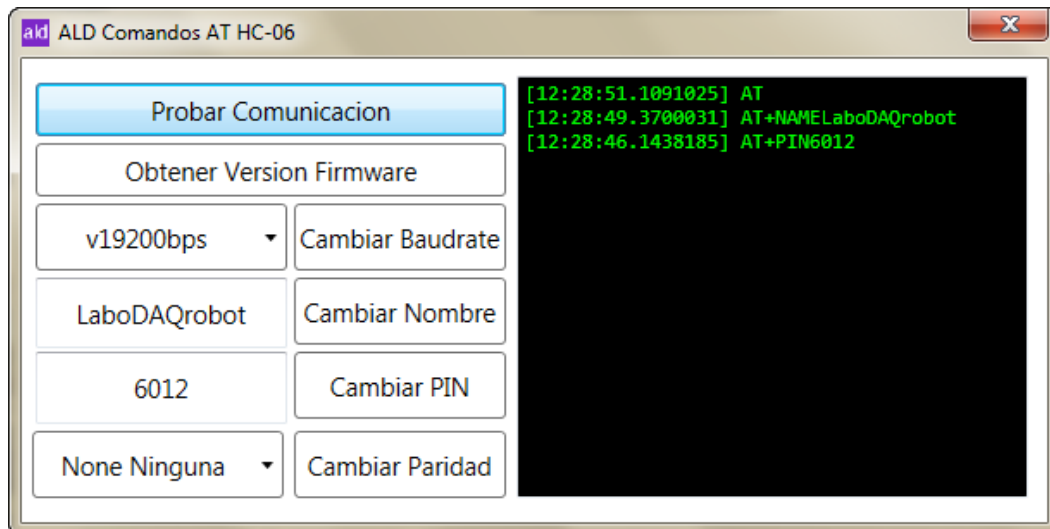


Figura 60. Aplicación ALDBluetoothATConfig para el módulo HC-05

Si los parámetros son correctos y el HC – 05 los aceptó, este enviará un “OK” seguido del nuevo valor de parámetro configurado. Para probar el envío/recepción de datos se hace la conexión mostrada en la figura 60. Arduino se programó para devolver el carácter alfanumérico que recibe por el puerto serie. Con la aplicación “Bluetooth serial” se realizó la conexión, emparejamiento y envío/recepción de los caracteres a través de la consola de dicha aplicación.

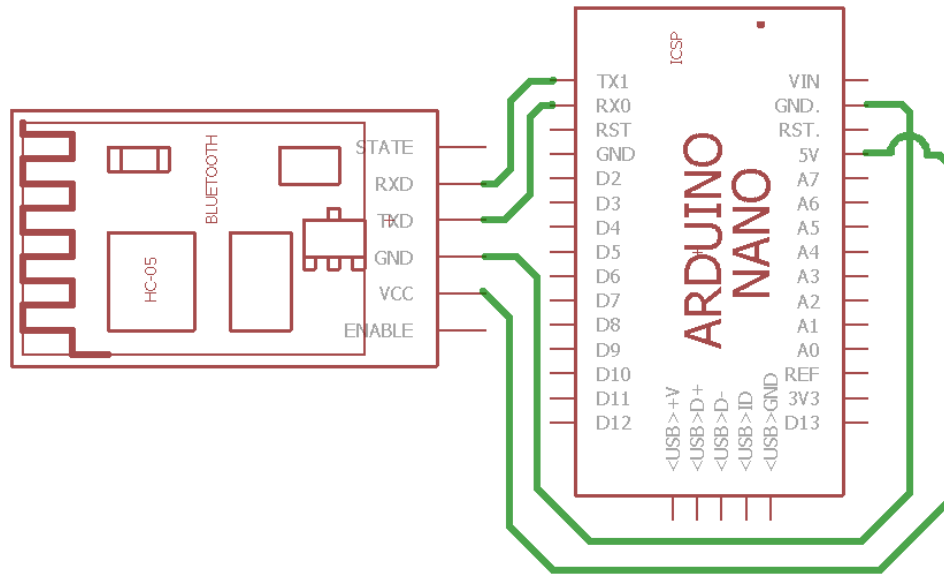


Figura 61. Conexión para prueba de transmisión de datos del módulo HC-05.

- **Sensor ultrasónico y acelerómetro:** Se realizó la conexión mostrada en la figura 62. El programa de Arduino envía la señal de inicio al sensor ultrasónico, espera por el pulso reflejado y cuenta el lapso entre ambos, para realizar el cálculo de la distancia. Con el monitor serial de Arduino se monitorizan las lecturas de distancia.

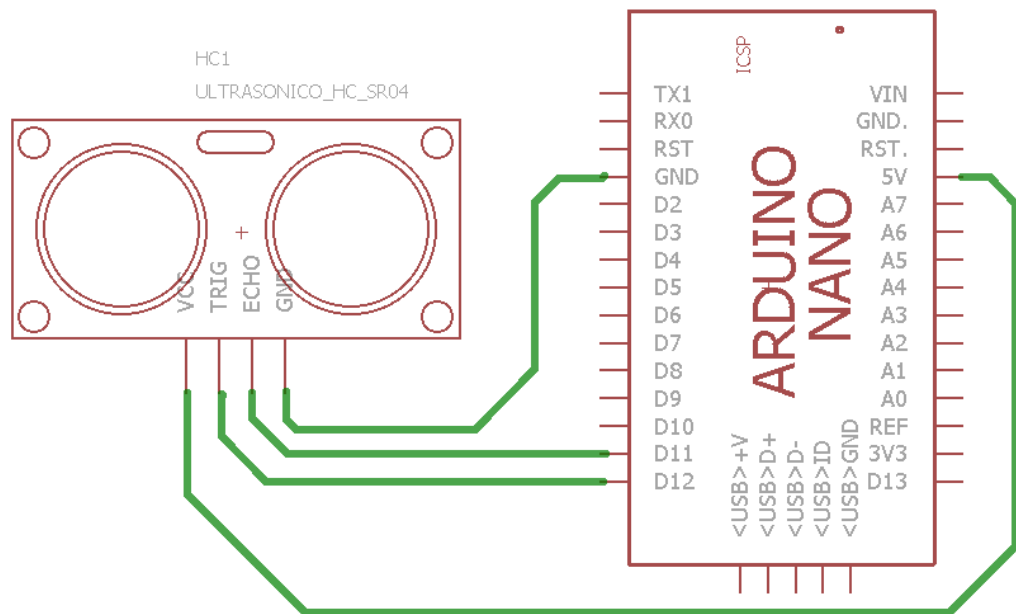


Figura 62. Conexión para prueba del sensor de distancia HC-SR04 con Arduino.

- Para el acelerómetro, se siguió la conexión mostrada en la figura 63. Dado que las salidas del acelerómetro analógicas, el programa de Arduino consistió en leer dichas entradas, acondicionarlas y por último enviar los valores de aceleración por puerto serie.

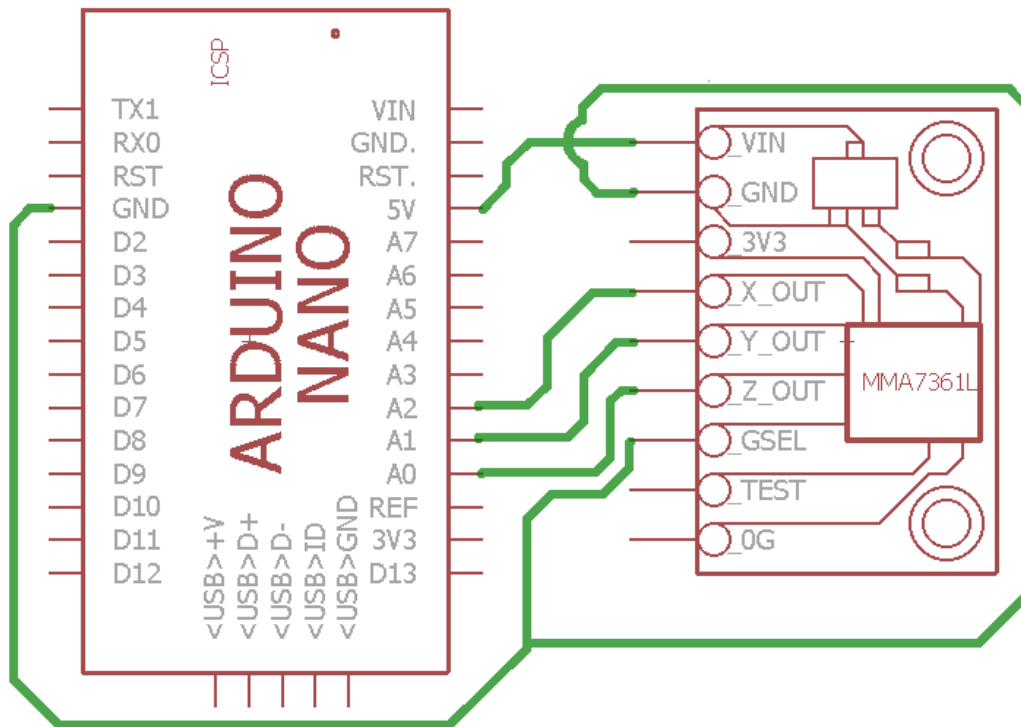


Figura 63. Conexión para realizar pruebas con el acelerómetro y el Arduino.

- **Fabricación de las piezas mecánicas, y ensamblado del robot.**

El maquinado de la base general del robot y la placa de conexiones se realizó en PVC de $\frac{1}{2}$ " (12.7mm). El resto de las piezas se maquinó en PVC de $\frac{1}{4}$ " (6.35mm). La sujeción de las piezas se hace mediante tornillos tipo Allen de $\frac{1}{8}$ " (3.175mm) y $\frac{1}{4}$ " (6.35mm) y pijas de un $\frac{1}{4}$ " (6.35mm) para los servomotores del brazo. El acabado de las piezas se obtuvo al lijar, resanar y pintar cada parte antes de ensamblar el robot.

En la figura 64 se aprecia la base general y se muestran los espacios destinados para los servomotores de tracción y ruedas locas.



Figura 64. Maquilado de la base general del robot.

La figura 65 ilustra la fabricación de la placa donde el usuario puede realizar sus conexiones y que sirve como cubierta para el sistema electrónico; y en la figura 66, se muestra un paso del maquinado del resto de las piezas; en específico la imagen corresponde a la fabricación de la base del brazo.

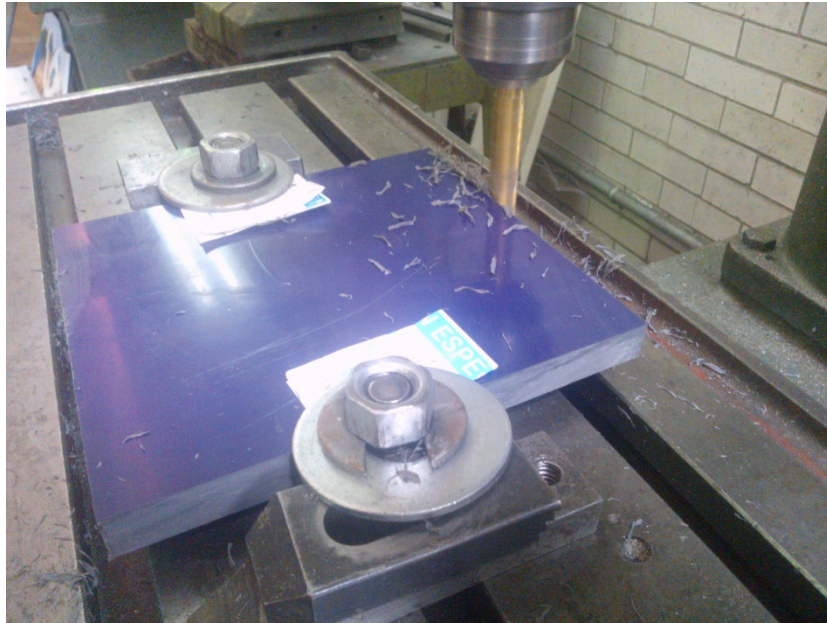


Figura 65. Maquilado de la placa de conexiones.

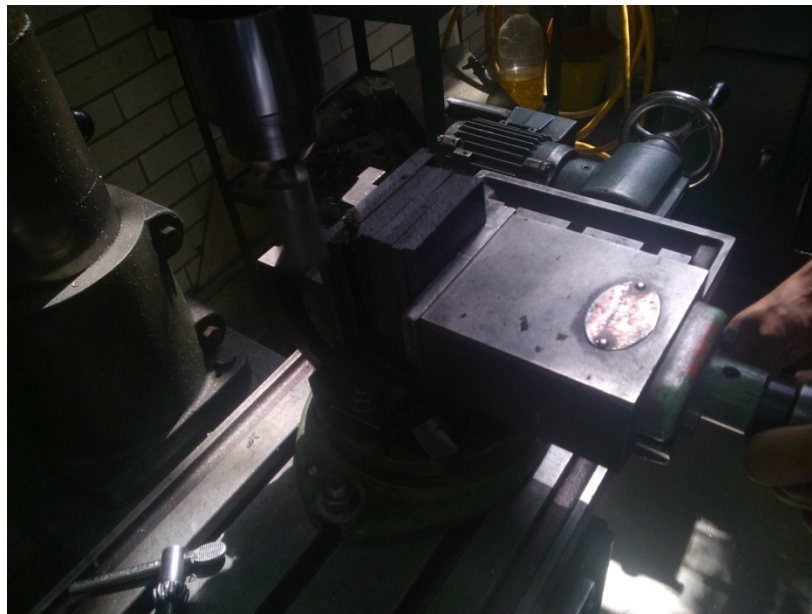


Figura 65. Maquilado de las piezas restantes.

La figura 67 muestra la totalidad de las piezas ya maquinadas, y parte del ensamblado del robot.

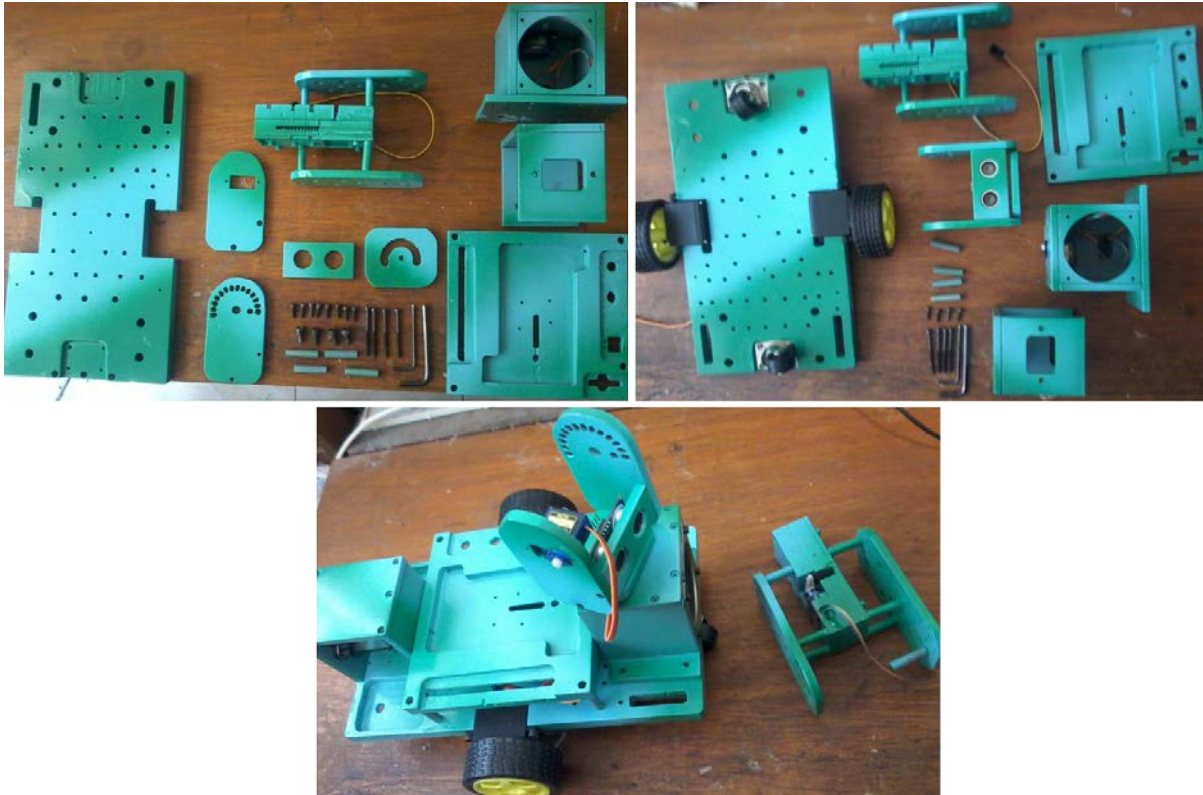


Figura 67. Piezas ya terminadas del robot.

- **Integración de los componentes electrónicos en el ensamblado final del robot.**

En el apartado 3.1.1. se describieron las pruebas individuales a los componentes del sistema electrónico. Lo siguiente es integrar cada uno de los componentes junto con el Arduino Nano, en una placa fenólica para proyectos (placa perforada), soldar y montar en las piezas mecánicas ya terminadas del robot (Figura 66); con base en el diagrama electrónico que se muestra en la figura 67. La figura 68 ilustra algunos pasos de la elaboración de la placa del sistema electrónico y su integración en las piezas mecánicas; mientras que la figura 70 muestra algunas vistas del ensamblado final del robot prototipo; por lo anterior, lo relacionado a la parte mecánica y de electrónica del robot queda concluida.

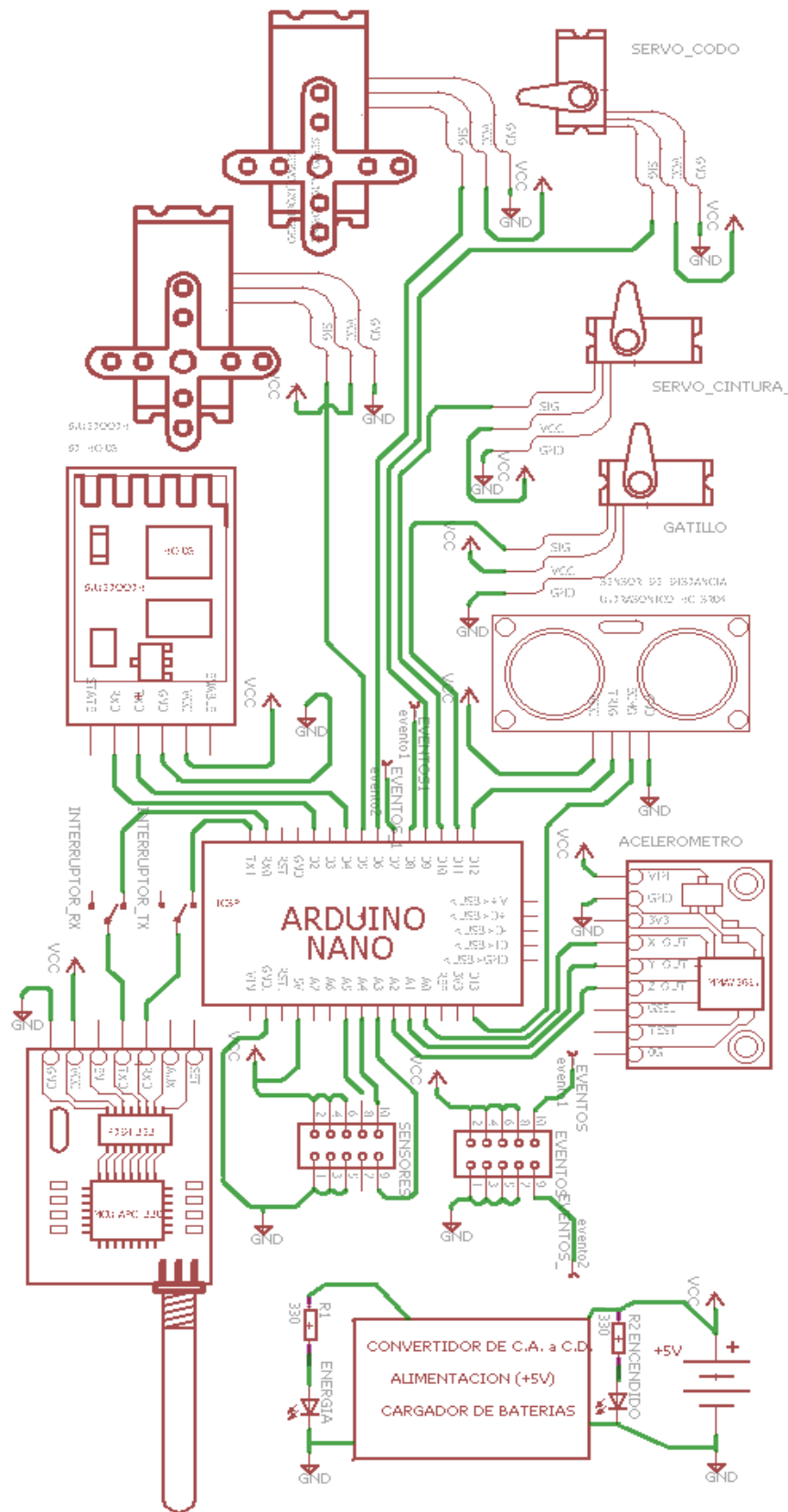


Figura 68. Diagrama del sistema electrónico del robot.

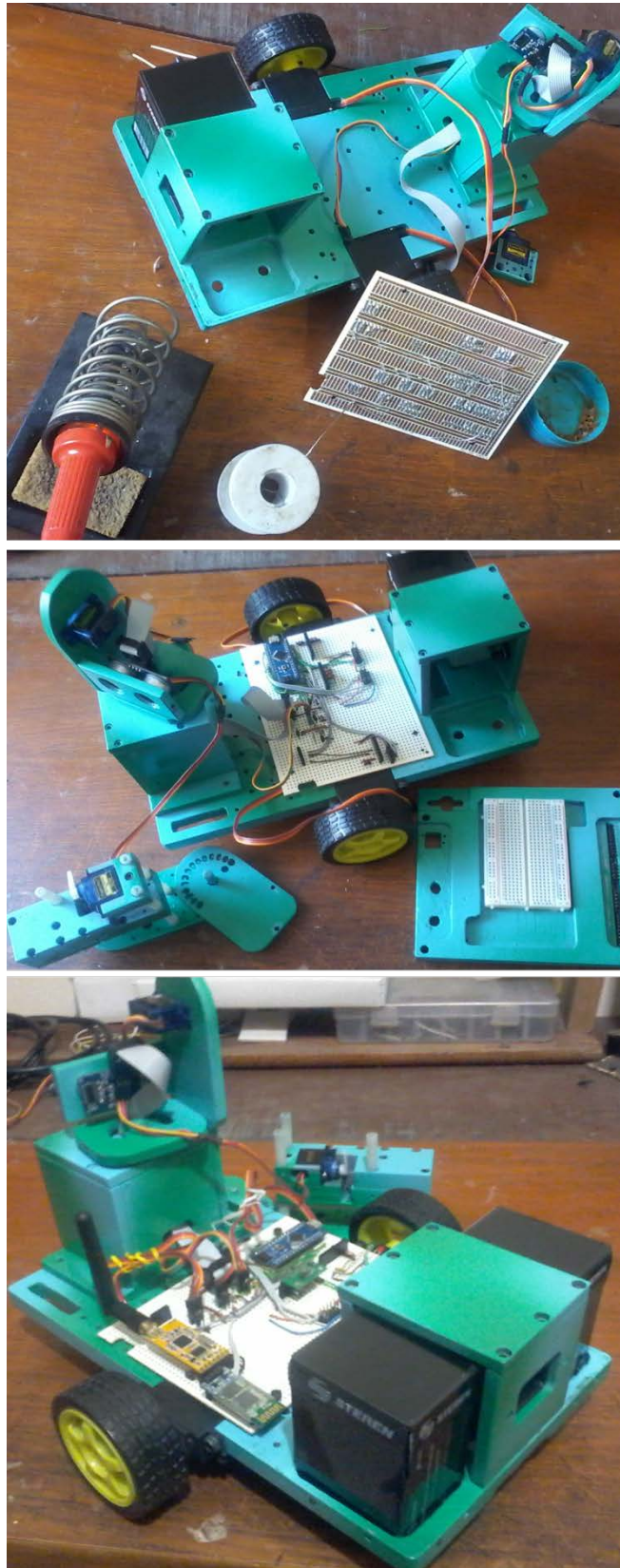


Figura 69. Elaboración e integración del sistema electrónico del robot.



Figura 70. Presentación final del robot prototipo.

3.2.- Programación de las interfaces gráficas de usuario.

El siguiente paso es la creación de las interfaces gráficas de usuario para PC y dispositivos móviles en plataforma JAVA; así como la programación del Arduino Nano. La programación del arduino Nano llevará a cabo las siguientes tareas:

- Control de giro de los servos de tracción.
- Posicionamiento de los servos que conforman el brazo mecánico y el disparador.
- Lectura del acelerómetro y sensor de distancia ultrasónico.
- Adquisición de datos de sensores conectados por el usuario.
- Lectura de eventos dados por sensores del tipo todo o nada.
- Gestionar las comunicaciones inalámbricas de RF y Bluetooth.
- Envío de datos a las interfaces gráficas de usuario y recepción de comandos por parte de las mismas.

Una de las características del robot es que simultáneamente podrá ser manipulado mediante Radio Frecuencia desde la PC y Bluetooth por medio de dispositivos móviles.

- **Interfaz gráfica para PC**

La interfaz gráfica de usuario en PC, será la que constantemente esté en comunicación con el robot; por lo tanto las tareas que tendrá que realizar son:

- Gestión de los parámetros de la comunicación serial en el equipo de cómputo.
- Procesar las instrucciones de los diferentes controles de la interfaz en un solo comando; para luego enviarlo hacia el Arduino Nano.
- Recepción, interpretación, visualización en pantalla y utilización por las diferentes funciones de la interfaz, de los datos enviados por el Arduino Nano

- Creación y carga de programas personalizados del usuario en el robot, mediante la aplicación Ardublock.jar
- Se podrán trazar gráficas con los datos experimentales tomados de los sensores conectados.
- Exportación de los datos en archivos .xls y .csv.
- Análisis matemáticos básicos, a los datos adquiridos de los sensores.

Un bosquejo de la interfaz gráfica se puede apreciar en la figura 71; la cual contendrá las características mencionadas.

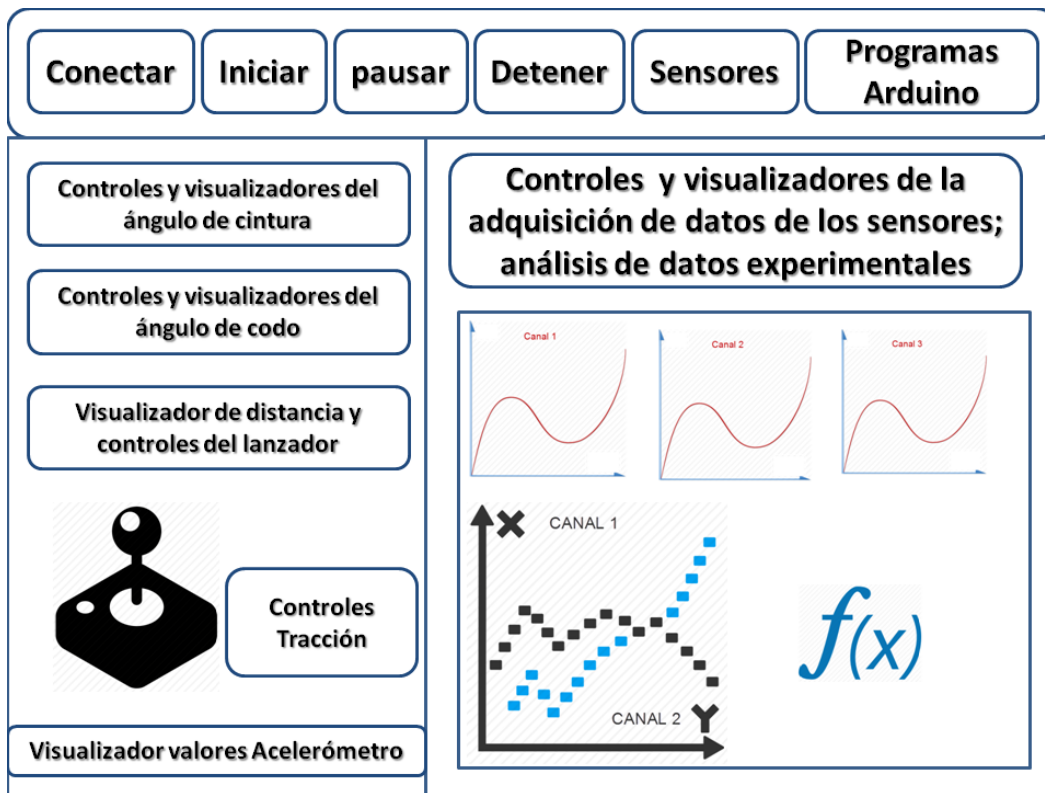


Figura 71. Bosquejo de la interfaz gráfica de usuario para PC.

Para cumplir la característica de programación personalizada, se recurrió al software Ardublock.jar mostrado en la figura 72. Dicho software está elaborado bajo la misma plataforma JAVA y es de código abierto, lo que permite la incorporación a la interfaz del robot. Este software permite programar los comandos de arduino de un modo gráfico.

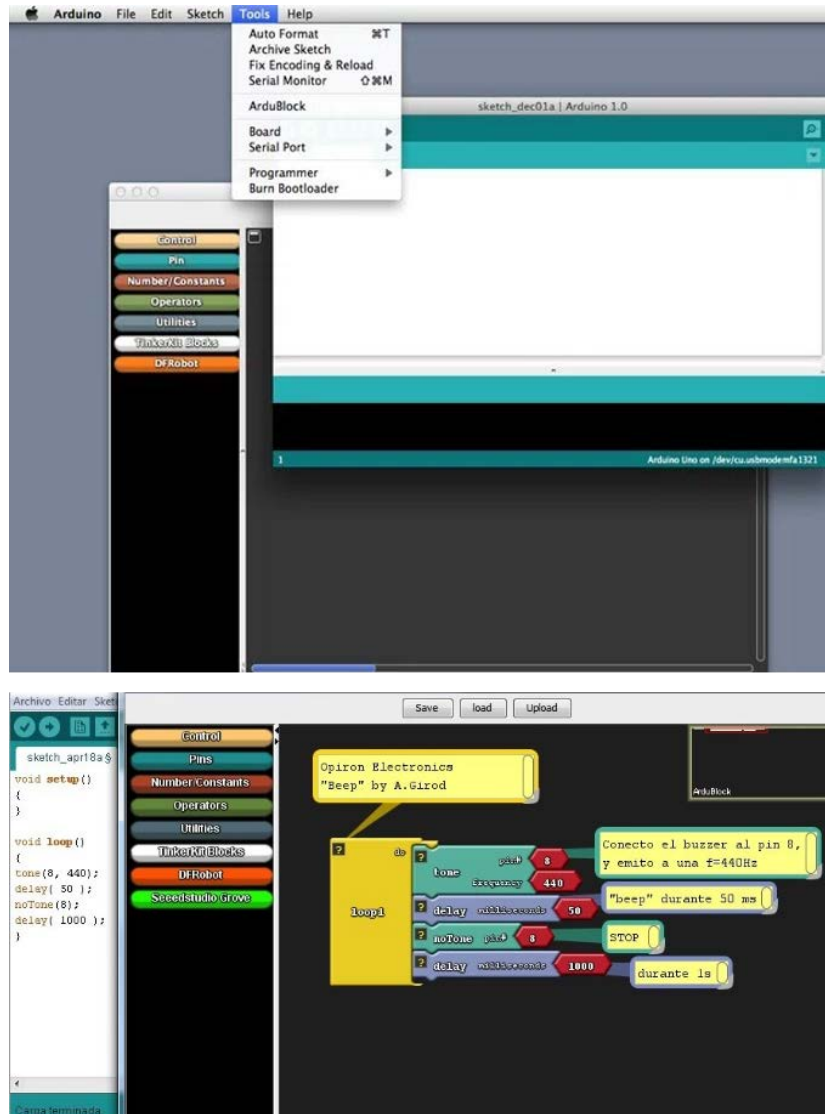


Figura 72. Software ArduBlock, para la programación gráfica del Arduino Nano del robot.

Fuente: <http://blog.ardublock.com/>

En la figura 71 se pueden apreciar algunos pasos del proceso de elaboración de la interfaz gráfica de PC. Este se inicio por definir los controles y gráficos, su distribución y características visuales, con base en el bosquejo presentado en la figura 72. Posteriormente a los controles con algún comando asociado con las funcionalidades del robot o una tarea del Arduino Nano, se les programó dicha instrucción. Por último se programaron todas las clases, rutinas y librerías relacionadas con el funcionamiento de la interfaz; este proceso y la interfaz gráfica de usuario finalizada, se pueden apreciar en la figura 73.

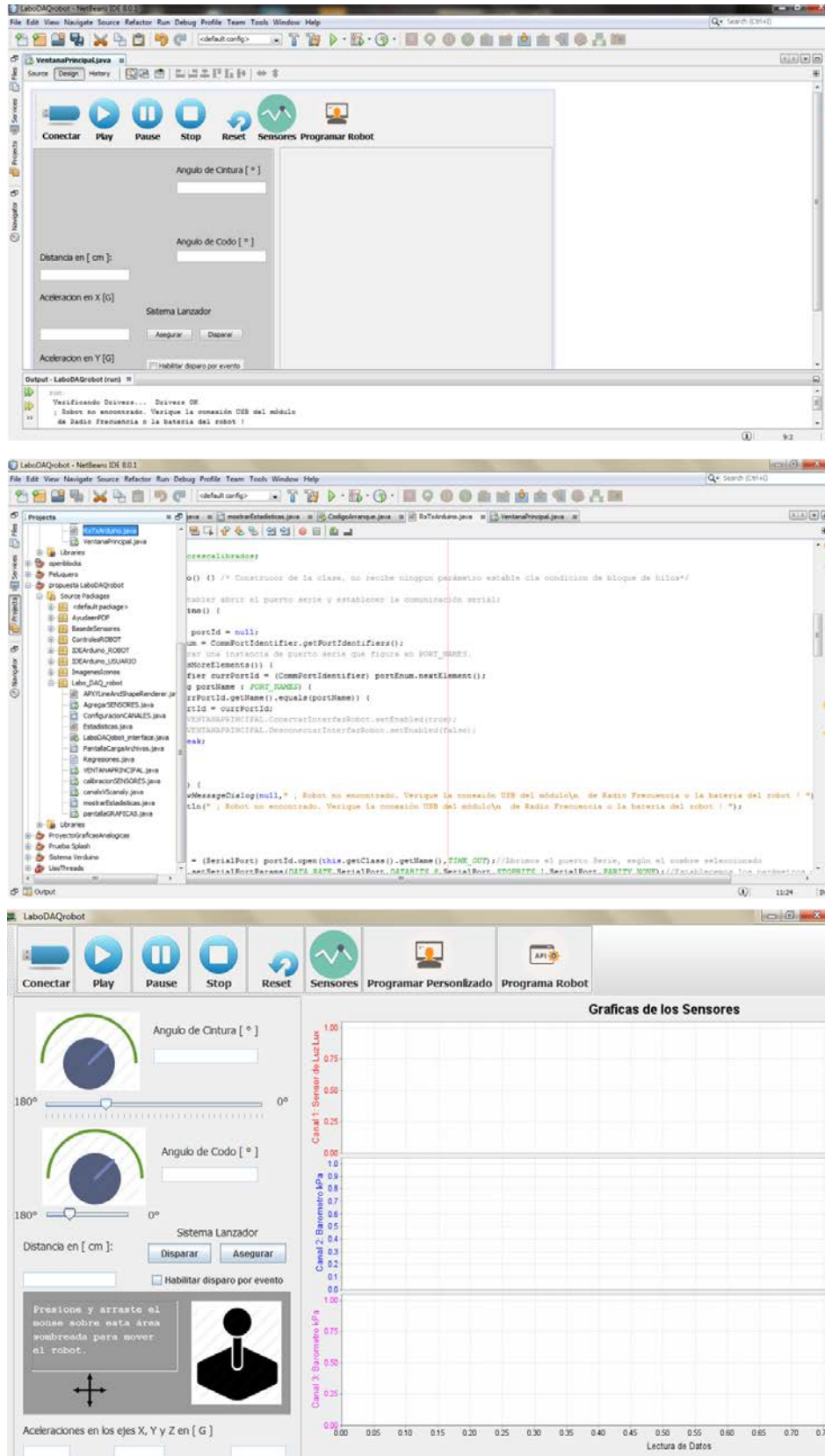


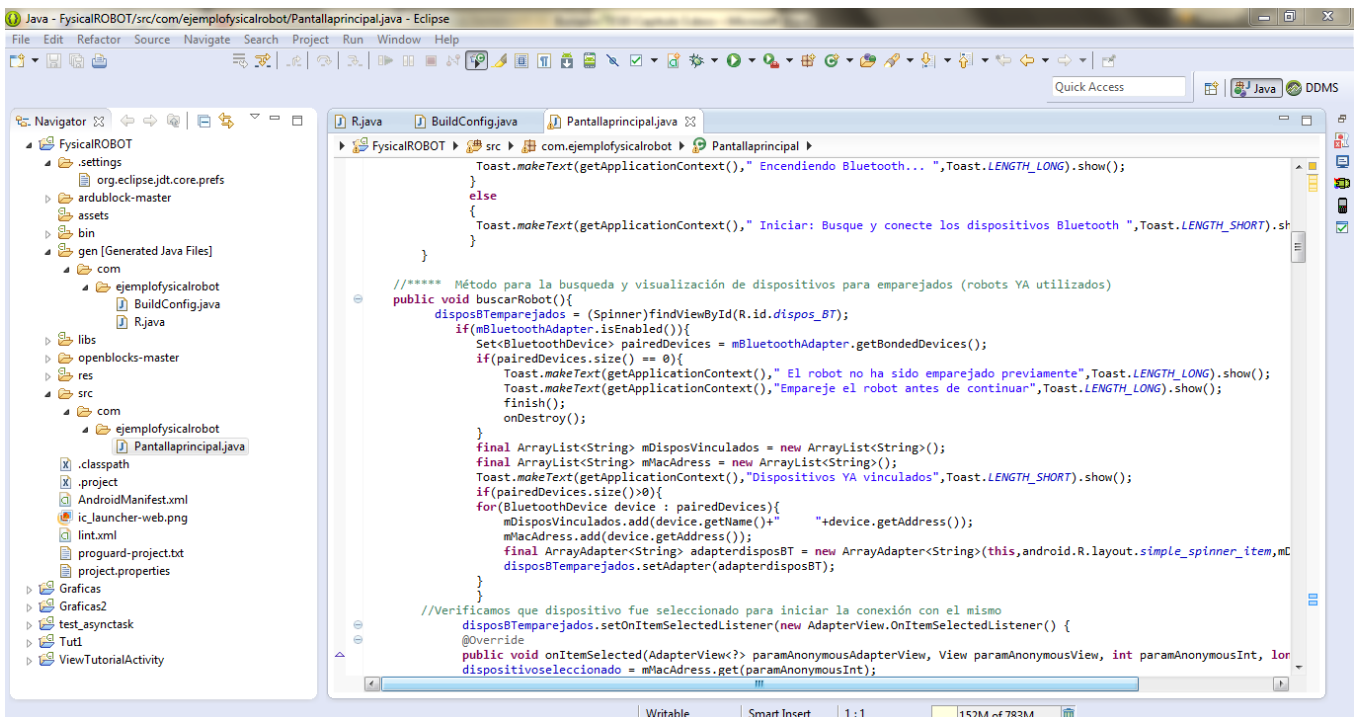
Figura 73. Proceso de elaboración y presentación final de la interfaz gráfica de usuario para PC.

3.2.1.- Interfaz gráfica para dispositivos móviles

La interfaz para dispositivos móviles tendrá las siguientes funciones:

- Búsqueda y emparejamiento del módulo de bluetooth HC-05 del robot.
- Gestionar la conexión del módulo bluetooth y de los parámetros del puerto serial virtual necesarios para el bluetooth del robot.
- Control de las funciones básicas del robot.
- Lectura del acelerómetro y el sensor de distancia del prototipo.
- Visualización de datos de los canales de adquisición del robot.

En la figura 74 se puede apreciar la programación de la interfaz para dispositivos móviles con sistema Android, mientras que en la figura 75 se ilustra la interfaz de usuario ya terminada.



```
Java - PhysicalROBOT/src/com/ejemplofysicalrobot/PantallaPrincipal.java - Eclipse
File Edit Refactor Source Navigate Search Project Run Window Help
Quick Access Java DDMS

Navigator
PhysicalROBOT
  .settings
  org.eclipse.jdt.core.preferences
  ardublock-master
  assets
  bin
  gen [Generated Java Files]
  com
    ejemplofysicalrobot
      BuildConfig.java
      R.java
  libs
  openblocks-master
  res
  src
    com
      ejemplofysicalrobot
        PantallaPrincipal.java
  .classpath
  .project
  AndroidManifest.xml
  ic_launcher-web.png
  lint.xml
  proguard-project.txt
  project.properties
  Graficas
  Graficas2
  test_asynctask
  Tut1
  ViewTutorialActivity

R.java BuildConfig.java PantallaPrincipal.java
PhysicalROBOT src com.ejemplofysicalrobot PantallaPrincipal
  Toast.makeText(getApplicationContext(), " Encendiendo Bluetooth... ", Toast.LENGTH_LONG).show();
  }
  else
  {
    Toast.makeText(getApplicationContext(), " Iniciar: Busque y conecte los dispositivos Bluetooth ", Toast.LENGTH_SHORT).sh
  }
}

//***** Método para la búsqueda y visualización de dispositivos para emparejados (robots YA utilizados)
public void buscarRobot(){
    disposBTemparejados = (Spinner)findViewById(R.id.dispos_BT);
    if(mBluetoothAdapter.isEnabled()){
        Set<BluetoothDevice> pairedDevices = mBluetoothAdapter.getBondedDevices();
        if(pairedDevices.size() == 0){
            Toast.makeText(getApplicationContext(), " El robot no ha sido emparejado previamente", Toast.LENGTH_LONG).show();
            Toast.makeText(getApplicationContext(), "Empareje el robot antes de continuar", Toast.LENGTH_LONG).show();
            finish();
        }
        onDestroy();
        final ArrayList<String> mDisposVinculados = new ArrayList<String>();
        final ArrayList<String> mMacAddress = new ArrayList<String>();
        Toast.makeText(getApplicationContext(), "Dispositivos YA vinculados", Toast.LENGTH_SHORT).show();
        if(pairedDevices.size() > 0){
            for(BluetoothDevice device : pairedDevices){
                mDisposVinculados.add(device.getName()+" "+device.getAddress());
                mMacAddress.add(device.getAddress());
            }
            final ArrayAdapter<String> adapterdisposBT = new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item, mC
            disposBTemparejados.setAdapter(adapterdisposBT);
        }
    }

//Verificamos que dispositivo fue seleccionado para iniciar la conexión con el mismo
disposBTemparejados.setOnItemClickListener(new AdapterView.OnItemClickListener() {
@Override
public void onItemClick(AdapterView? paramAnonymousAdapterView, View paramAnonymousView, int paramAnonymousInt, lon
dispositivoseleccionado = mMacAddress.get(paramAnonymousInt);
}
})
}
}

Writable Smart Insert 1:1 152M of 783M
```

Figura 74. Programación de la interfaz gráfica de usuario para dispositivos móviles en Eclipse.



Figura 75. Interfaz gráfica de usuario terminada e instala en tableta.

4.- VERIFICACIÓN DE OPERACIÓN

Capítulo 4.- Verificación de operación

4.1.- Pruebas de comunicación y programación

En esta prueba se utilizaron la IDE de Arduino, la aplicación SerialTerminal.apk, ArduBlock.jar y el programa Terminal.exe. Se programó el Arduino para que accionara cada uno de los dispositivos del robot de manera individual y repetitiva. A los servomotores de tracción los detenía y giraba en diferentes direcciones, a los servomotores del brazo los giraba de 0° a 180° para que las articulaciones del brazo se movieran y el mecanismo del lanzador funcionaran. También tomaba y enviaba las lecturas del sensor de distancia, acelerómetro y sensores analógicos conectados en los canales del robot. Posteriormente este esperaba un comando proveniente de la IDE y del programa Terminal.exe para accionar el dispositivo asociado a cada comando; o bien para enviar información de los parámetros del robot.

Dicha prueba la podemos apreciar en las figuras 76 y 77. En la figura 78 se aprecia la programación del robot por medio del ArduBlock.jar y la IDE de Arduino; para ello se desconectó el módulo de Radio Frecuencia APC-220 y cambiaron los interruptores a la posición de “Comunicación por PC”. Con un cable USB – micro USB se realizó la programación.

En la figura 79, podemos apreciar la transmisión de los parámetros del robot en el programa Terminal.exe y la aplicación SerialTerminal.apk; dichos parámetros son enviados de manera consecutiva separados por comas “,” en una sola cadena que luego será interpretada por las interfaces gráficas de usuario. En este punto se probó la parte de la interfaz de PC encargada de controlar los parámetros de la comunicación serial y recibir los parámetros enviados por el Arduino.

Con estas pruebas se verificaron las funciones del robot prototipo así como las comunicaciones inalámbricas.

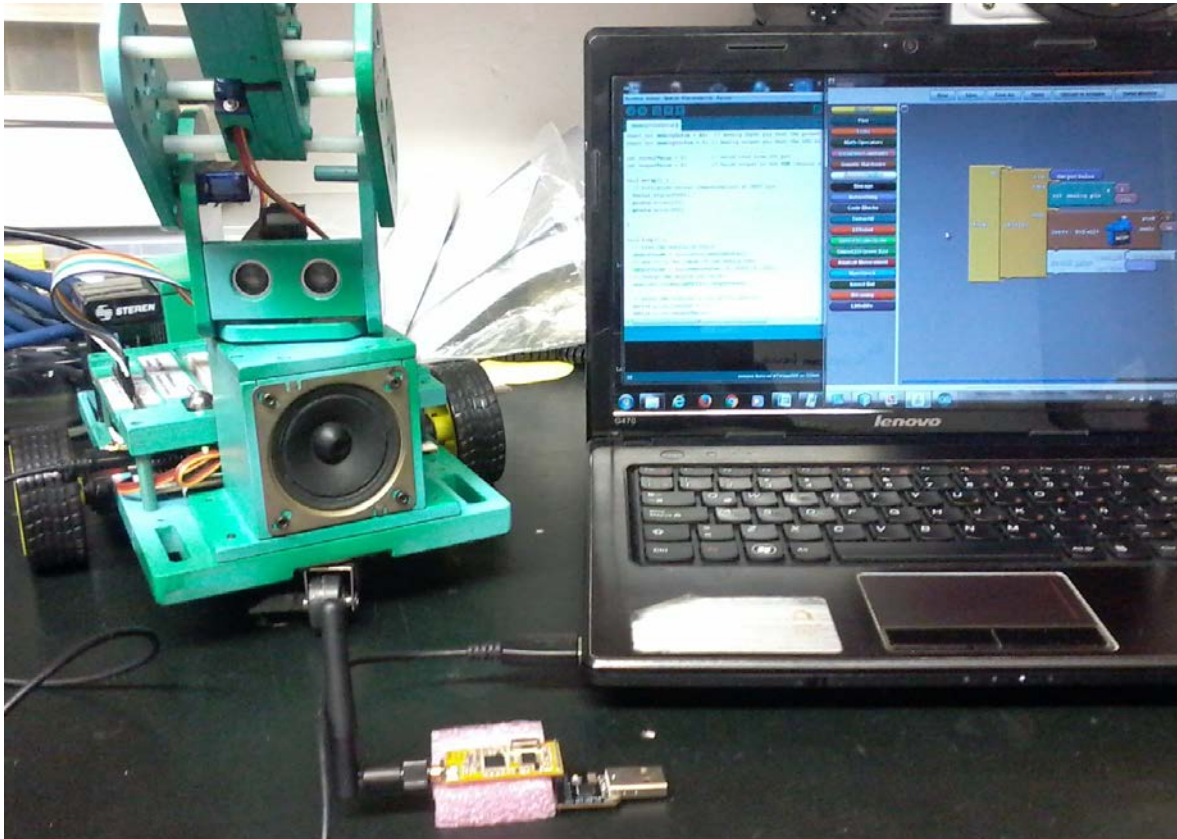


Figura 76. Pruebas de programación de Arduino con IDE y ArduBlock.jar.

En la prueba de la figura 81:

- Se retiró el módulo de Radio Frecuencia APC-220 y conectó el robot a la PC por medio de un cable micro USB – USB tipo A.
- En ArduBlock.jar se creó un programa básico para que de forma repetitiva el Arduino del robot, accionar el servomotor en la base del brazo y este girara en un rango de 0° - 180° al mismo tiempo que enviaba el dato de posición por puerto serie.
- Una vez depurado el programa se cargó al Arduino Nano del robot para verificar la operación del mismo. El programa ArduBlock.jar generó el código equivalente para el Arduino.
- Una vez terminada la carga del programa, el brazo giraba tanto en sentido levógiro como dextrógiro, abarcando un rango de entre 0° - 180° .

- En el monitor serial de la IDE de Arduino se visualizaba el dato de ángulo.
- Al programa creado en ArduBlock.jar, se le cambió el número de pin definido para el servomotor de la cintura, por el definido para el servomotor que aseguraba el proyectil del lanzador. Con esto se lograba la acción de soltar y asegurar el disparador del brazo.
- Sin colocar ningún proyectil ni contraer el resorte se cargó el programa anteriormente creado, y se observó como el servomotor libera y aseguraba el mecanismo lanzador.
- Nuevamente se modificó el número del pin en el programa de ArduBlock.jar, pero en esta ocasión se colocaron los números de los pines correspondientes a los servomotores de tracción del robot de manera individual.
- Al cargar y ejecutar el programa, se observó cómo cada uno de los servomotores de tracción giraba en un sentido, detenía paulatinamente su marcha y posteriormente giraba en sentido de manera repetitiva.

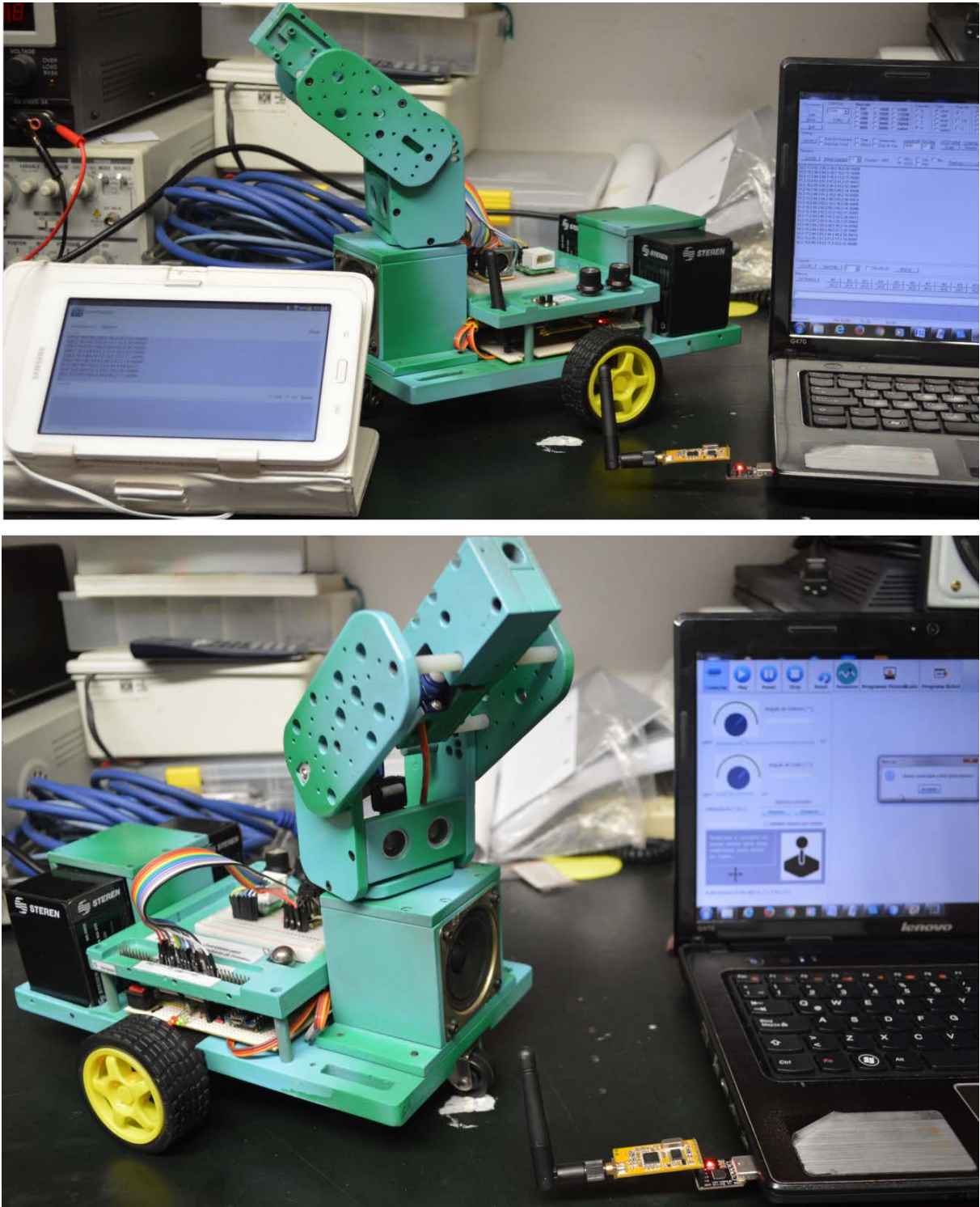


Figura 77. Pruebas de comunicación y control inalámbrica por medio de Terminal.exe y SerialTerminal.apk

En la figura 77:

- Se utilizaron los programas Terminal.exe y SerialTerminal.apk para comprobar la transmisión de datos entre el robot y los dispositivos. El programa Terminal.exe se ejecutó en PC, para emular la comunicación serial con la interfaz gráfica en JAVA; mientras que la aplicación SerialTerminal.apk emulaba la comunicación con dispositivos móviles Android.
- Al encender el robot, éste empezaba a enviar sus parámetros. En el programa Terminal.exe se seleccionó el puerto COM correspondiente al convertidor USB / TTL que contenía a uno de los módulos del APC-220 de inmediato reflejaba los datos en consola. Posteriormente en la aplicación SerialTerminal.apk se busca el dispositivo Bluetooth HC-05 del robot (anteriormente ya conectado) y se establecía la comunicación; de forma inmediata comenzaba a recibir los datos que el robot empezaba a enviar.
- En ambos programas se envió el comando correspondiente para el accionamiento de los servomotores del brazo y el lanzador de proyectiles. Se observó que ambos se accionaban. Con esto se comprobó la característica de transmisión simultánea entre PC y dispositivos móviles.

Con las pruebas anteriores, las funciones del robot así como los componentes del mismo, fueron verificados. Lo siguiente fue la realización de algunas prácticas prototipo para comprobar el resto de las características del prototipo.

4.2.- Comprobación de las funciones del robot.

1.- Programación de códigos personalizados.

Esta práctica consistirá en realizar con el programa de ArduBlock.jar un detector de obstáculo que posteriormente pueda ser empleado dentro de un programa para que el robot pueda resolver un laberinto. El robot avanzará en línea recta siempre y cuando no haya un

obstáculo a menos de 10cm. Al detectar un obstáculo este se detendrá y girará hasta encontrar una dirección donde no haya obstáculos; es decir, girará hasta que no detecte objeto alguno en su rango de medida; o bien en un rango de medida especificado por el usuario, en ese instante avanzará en esa dirección; lo anterior de manera repetitiva.

La figura 79 ilustra parte de la programación en ArduBlock.jar; dicha programación fue realizada por una estudiante recién egresada de bachillerato, con la finalidad de probar la facilidad en el uso de la interfaz y del robot, por parte de un usuario promedio. Al terminar la programación, el comportamiento resultante no fue el adecuado; debido a fallos en el código del estudiante, pero no hubo inconvenientes al entender el uso del robot y la interfaz gráfica.

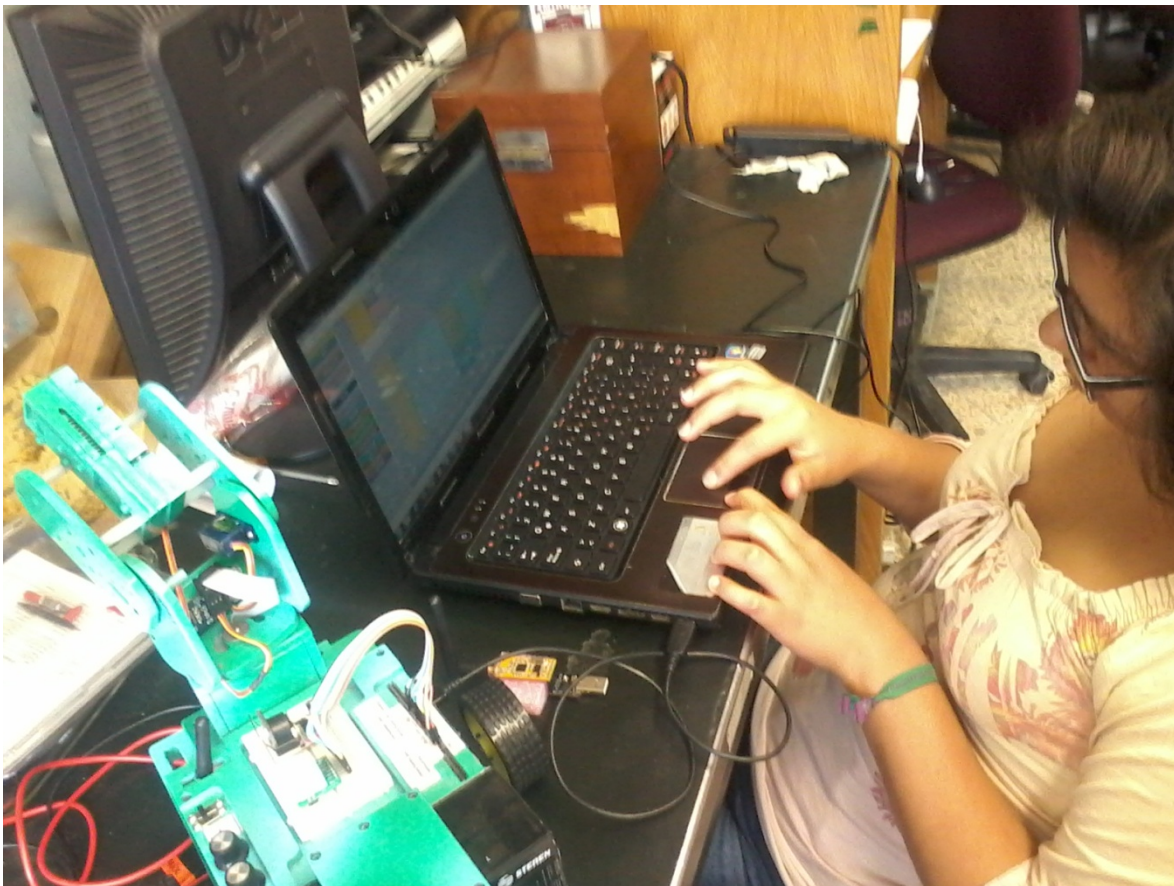


Figura 78. Programación de un detector de obstáculo por medio de ArduBlock.jar

2.- Adquisición de datos experimentales:

Para esta práctica se utilizaron los sensores de marca Vernier:

- Colorímetro.
- Sensor de temperatura.
- Sensor de presión.

Todos estos sensores entregan una señal de voltaje proporcional a la variable que censan. En primer lugar se utilizó el colorímetro, este entrega una señal variable en voltaje proporcional a la longitud de onda y a la intensidad de luz registrada por el sensor. La interfaz gráfica de usuario interpretará la adquisición realizada por el Arduino y con base en esto, se podrá medir la concentración de un soluto o el cultivo de bacterias biológicas de la muestra que se metió en el colorímetro.

La figura 79 muestra la conexión del adaptador y colorímetro a la tablilla de pruebas incorporada en el robot, por parte de una alumna recién egresada de bachillerato y se comprobó que la distribución de los pines de conexión y la tablilla de pruebas, facilitaba el uso del robot. En la interfaz gráfica se adquirían datos, mientras se manipulaba el colorímetro. Las figuras 79 y 80 ilustran la conexión de los sensores Vernier de temperatura y presión al robot; así como la adquisición de datos en la interfaz gráfica del robot. En la tablilla de pruebas se colocan los adaptadores de *protoboard* y posteriormente con *headers*, se realizó la conexión al robot. Los datos registrados y procesados para los sensores, corresponden a valores ambiente de presión y temperatura; por lo cual la adquisición, quedó verificada.

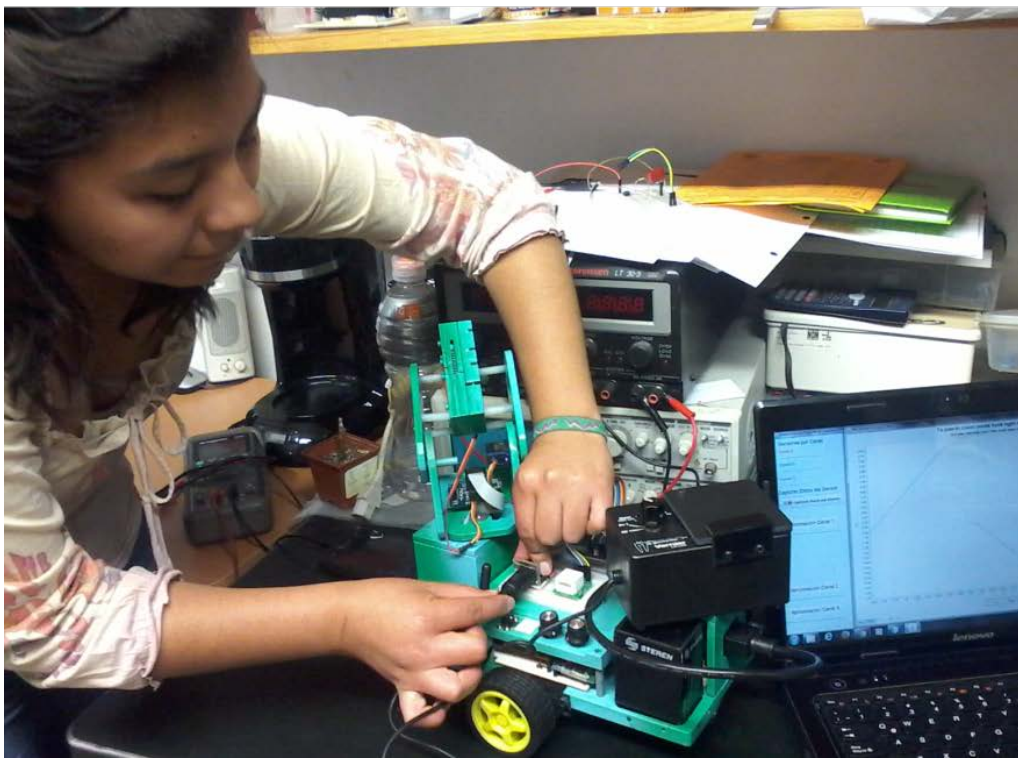
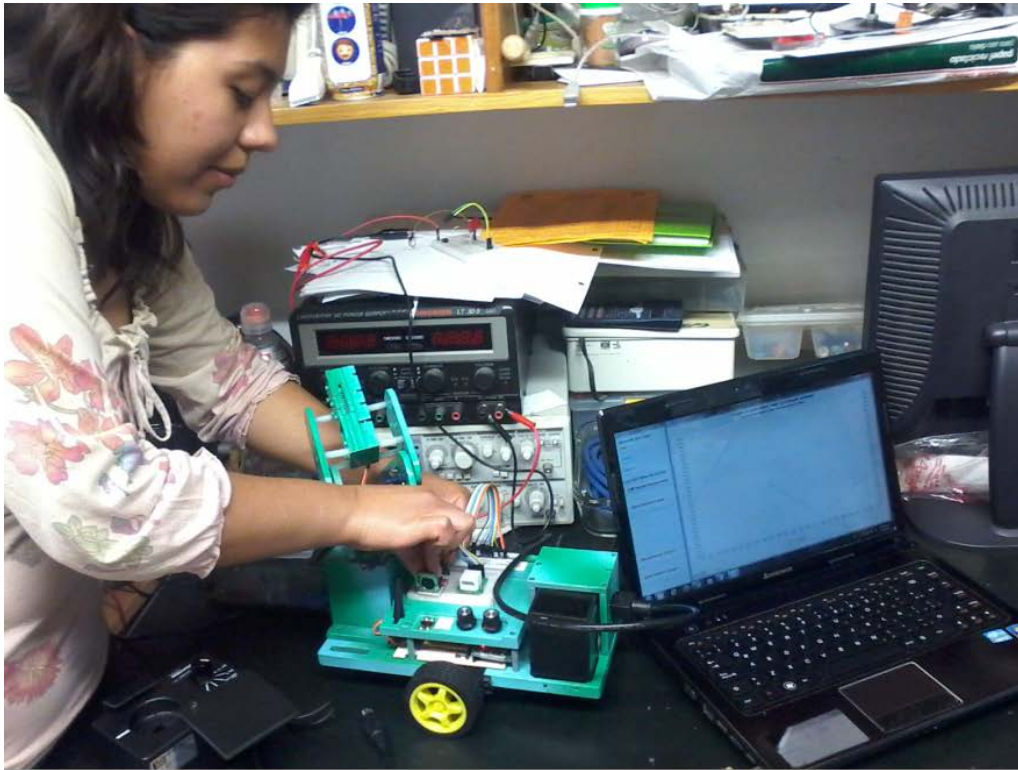


Figura 79. Adquisición de datos de un sensor Vernier.(Colorímetro).

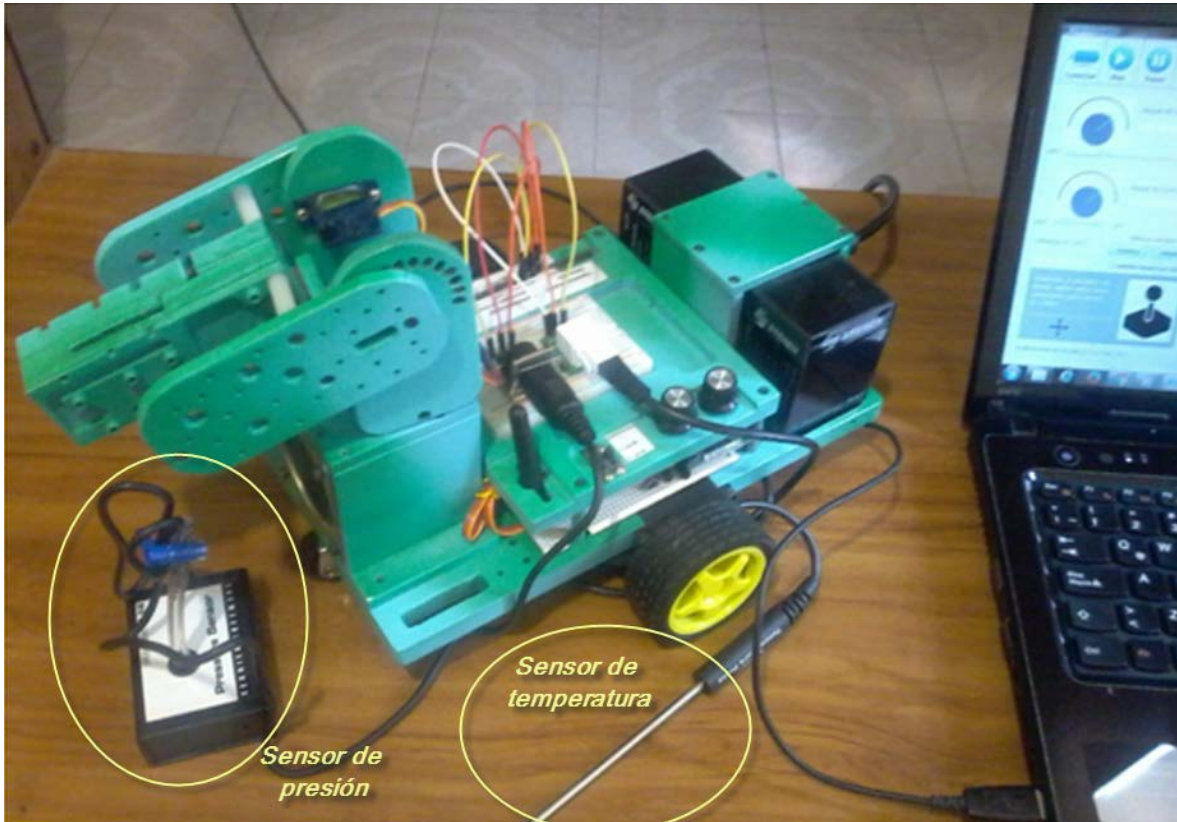


Figura 80. Adquisición de datos de sensores Vernier de presión y temperatura.

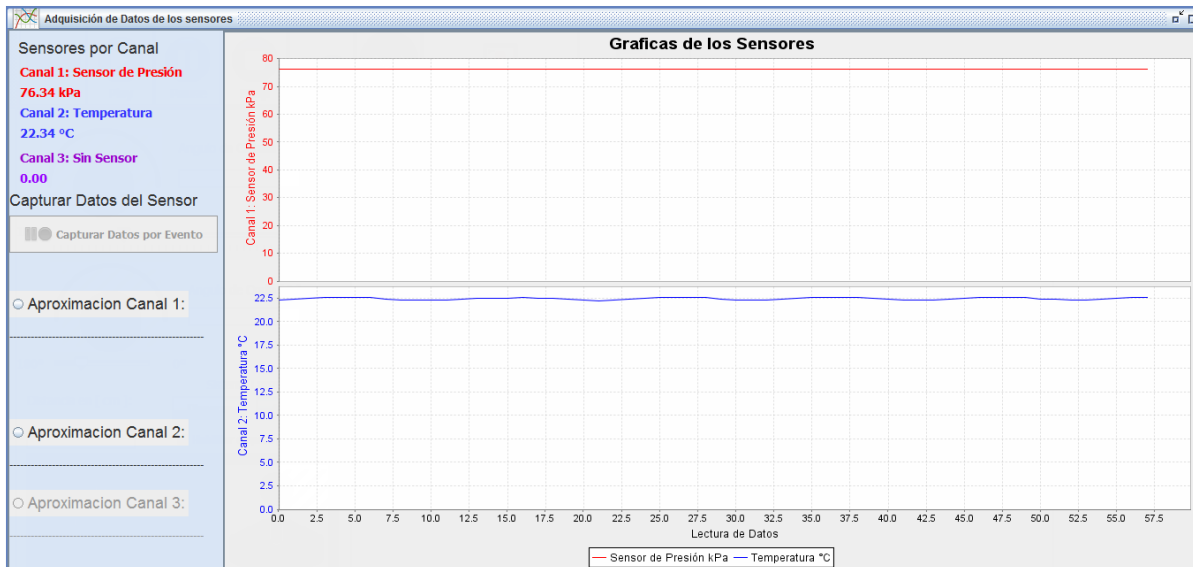


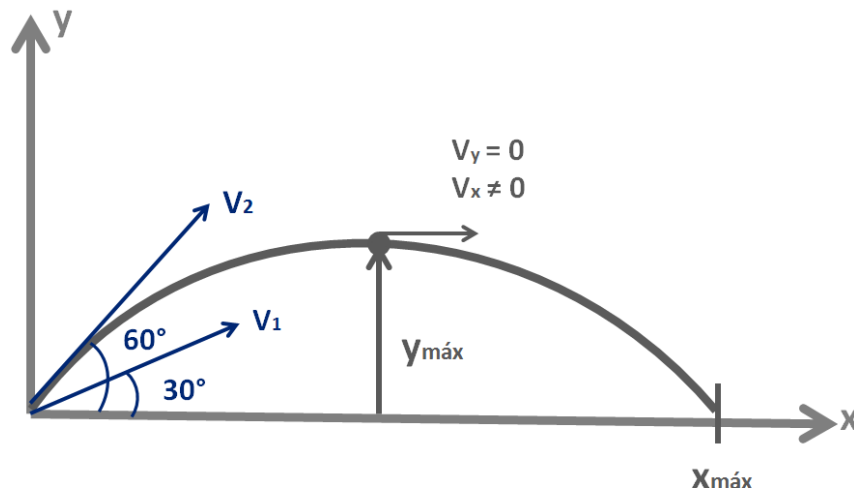
Figura 81. Los datos procesados por la interfaz gráfica de los sensores de presión y temperatura,

3.- Tiro parabólico:

En práctica, se usará al robot, para comprobar la demostración al siguiente problema de física del bachillerato:

“2.43 [II]. Demuestre que el disparo de una pistola puede alcanzar el triple de altura cuando tiene un ángulo de elevación de 60° que cuando su ángulo es de 30° , pero que tendrá el mismo alcance horizontal” [44].

Solución:



Tomamos las componentes de las velocidades iniciales

$$V_{1x} = V_1 \cos 30^\circ = \frac{\sqrt{3}}{2} V_1$$

$$V_{1y} = V_1 \sin 30^\circ = \frac{1}{2} V_1$$

$$V_{2x} = V_2 \cos 60^\circ = \frac{1}{2} V_2$$

$$V_{2y} = V_2 \sin 60^\circ = \frac{\sqrt{3}}{2} V_2$$

El alcance está dado por: $x = V_x t$

La altura está dada por: $y = \frac{-gt^2}{2} + V_y t$

Para conocer el alcance máximo, se multiplica la componente en x de la velocidad inicial, por el tiempo, este cuando la altura es cero; es decir, $y = 0$. Para encontrar este tiempo, de la fórmula de altura, sustituimos la condición anterior:

$$\begin{aligned} \frac{-gt^2}{2} + V_y t &= 0 \\ \frac{-gt^2}{2} &= -V_y t \end{aligned}$$

Cancelamos términos y despejamos la variable tiempo t:

$$\begin{aligned} \frac{-gt^2}{2} &= -V_y t \\ \cancel{+gt^2} &= \cancel{+V_y t} \\ \frac{gt}{2} &= V_y \\ t &= \frac{2V_y}{g} \end{aligned}$$

Al sustituir el tiempo encontrado, en la fórmula del alcance, obtendremos el valor máximo de alcance. Entonces:

Para 30°

Para 60°

$$t = \frac{2V_{1y}}{g} = \frac{2 \left(\frac{1}{2} V_1 \right)}{g} = \frac{V_1}{g}$$

$$t = \frac{2V_{2y}}{g} = \frac{2 \left(\frac{\sqrt{3}}{2} V_2 \right)}{g} = \frac{\sqrt{3}V_2}{g}$$

$$x_{\max 1} = V_{1x} t$$

$$x_{\max 2} = V_{2x} t$$

$$x_{\max 1} = V_{1x} t = \frac{\sqrt{3}}{2} V_1 \left(\frac{V_1}{g} \right)$$

$$x_{\max 2} = V_{2x} t = \left(\frac{1}{2} V_2 \right) \left(\frac{\sqrt{3}V_2}{g} \right)$$

$$x_{\max 1} = \frac{\sqrt{3}}{2g} V_1^2$$

$$x_{\max 2} = \frac{\sqrt{3}}{2g} V_2^2$$

$Si V_1 = V_2 \Rightarrow x_{\max 1} = x_{\max 2}$

El siguiente punto es demostrar que la altura a 60° es el triple que la de 30° . Esta altura se da en el punto tangente a la parábola; entonces podemos derivar la fórmula de altura:

$$y = \frac{-gt^2}{2} + V_y t$$

$$\frac{dy}{dt} \left(\frac{-gt^2}{2} + V_y t \right) = -gt + V_y$$

Dado que el valor de la derivada en el punto de altura máxima es cero, podemos saber el tiempo: $0 = -gt + V_y$ de la esta fórmula. Posteriormente sustituimos dicho tiempo en la fórmula de altura, para conocer la altura máxima:

Para 30° :

$$0 = -gt + V_{1y}$$

$$gt = \frac{1}{2} V_1$$

$$t = \frac{V_1}{2g}$$

$$y_{\max 1} = -\frac{g}{2} \left(\frac{V_1}{2g} \right)^2 + \frac{1}{2} V_1 \left(\frac{V_1}{2g} \right)$$

$$y_{\max 1} = -\frac{V_1^2}{8g} + \frac{V_1^2}{4g} = \frac{V_1^2}{4g}$$

$$y_{\max 1} = \frac{V_1^2}{4g}$$

Para 60° :

$$0 = -gt + V_{2y}$$

$$gt = \frac{\sqrt{3}}{2} V_2$$

$$t = \frac{\sqrt{3} V_2}{2g}$$

$$y_{\max 2} = -\frac{g}{2} \left(\frac{\sqrt{3} V_2}{2g} \right)^2 + \frac{\sqrt{3}}{2} V_2 \left(\frac{\sqrt{3} V_2}{2g} \right)$$

$$y_{\max 2} = -\frac{3V_2^2}{8g} + \frac{3V_2^2}{4g} = \frac{3V_2^2}{4g}$$

$$y_{\max 2} = 3 \left(\frac{V_2^2}{4g} \right)$$

$$\text{Si } V_1 = V_2 \Rightarrow y_{\max 2} = 3y_{\max 1}$$

Las anteriores soluciones se cumplen si las velocidades iniciales a 30° y 60° son iguales $V_1 = V_2$. Por el principio de Conservación de la energía :

$$\Delta EC_i + \Delta EP_i = \Delta EC_f + \Delta EP_f$$

$$\frac{1}{2} kx^2 = \frac{1}{2} mv^2$$

$$\cancel{\frac{1}{2}} kx^2 = \cancel{\frac{1}{2}} mv^2$$

$$V = \sqrt{\frac{k}{m}} x$$

Dado que la velocidad depende de la masa del balón, la constante del resorte y el desplazamiento del mismo, la velocidad inicial con la que salga el balón será constante y dado que los parámetros no cambian, V_1 y V_2 son iguales y las anteriores soluciones serán validas.

Para la comprobar la demostración del problema, primero se ubicó el disparador a un ángulo de 30° , se realizó un disparo y con el sensor ultrasónico se obtuvo la distancia a la cual llegó el proyectil, posteriormente se elevó el brazo a ángulo de 60° y realizó el disparo nuevamente; a lo cual el proyectil llegó al mismo punto, en consecuencia la demostración al problema quedó comprobada. La figura 88 muestra un bosquejo de como se llevó a cabo la demostración con el robot prototipo y parte de la comprobación llevada a cabo con el robot, por parte de una alumna egresada de bachillerato.

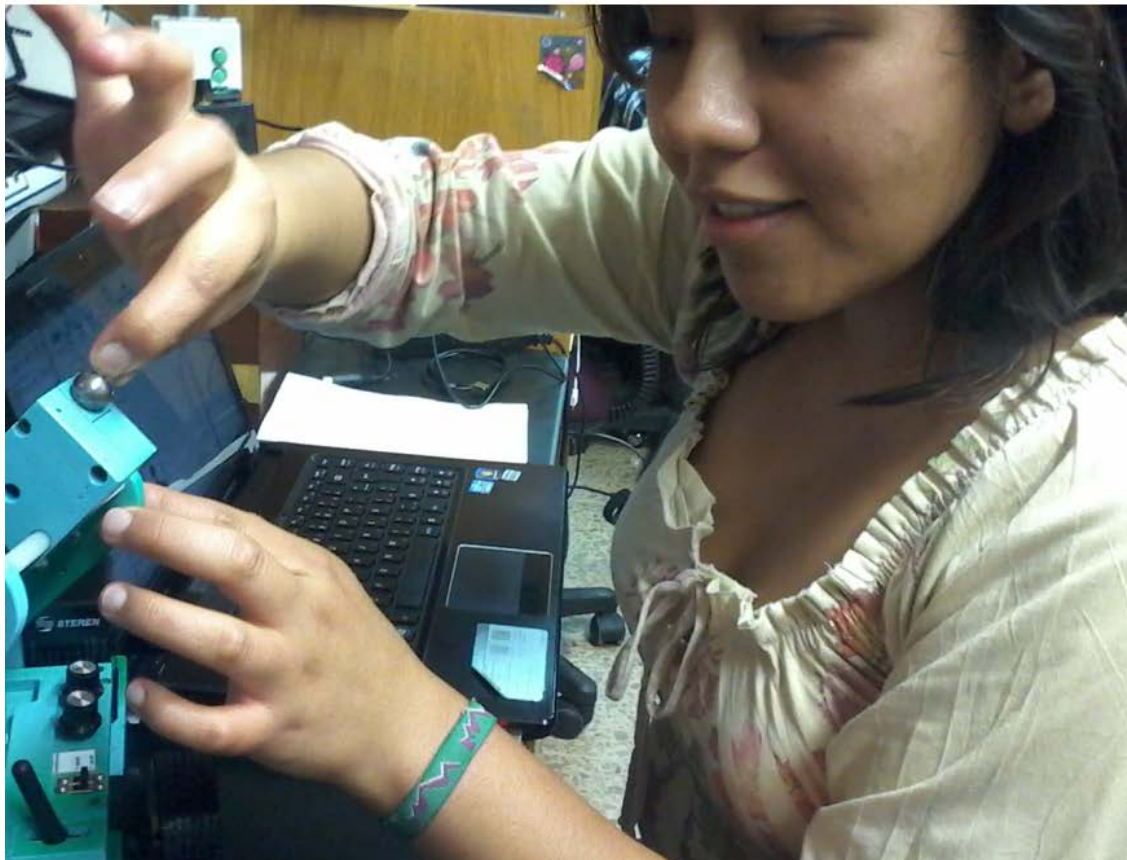
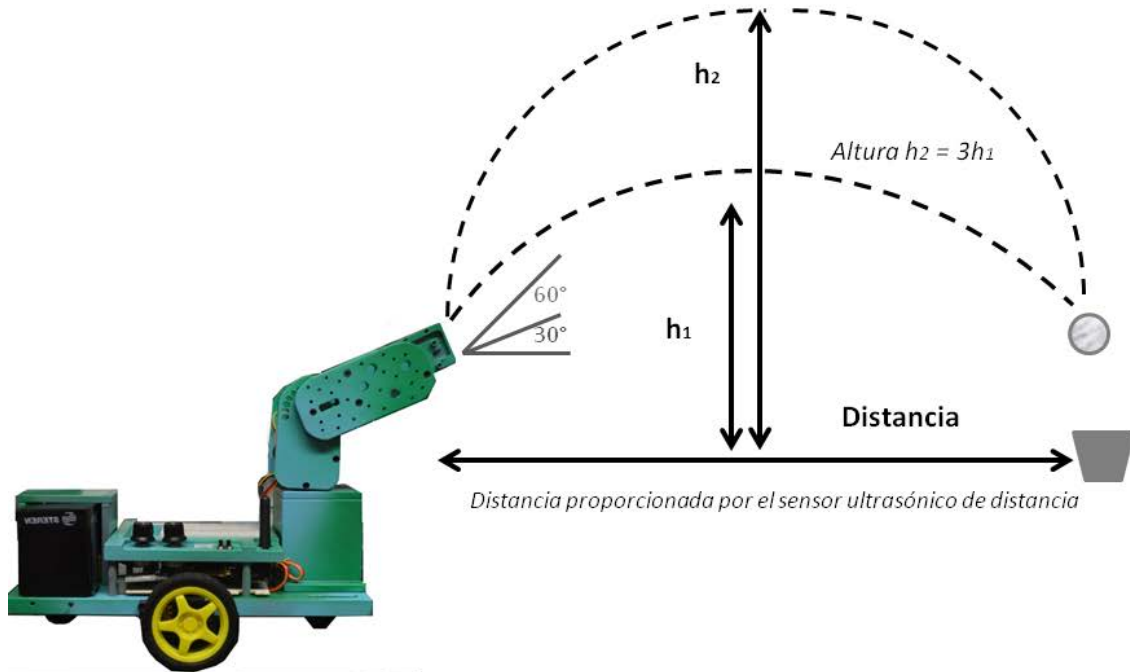


Figura 82. Comprobación al problema utilizando el robot didáctico

5.- CONCLUSIONES Y TRABAJO A FUTURO

Capítulo 5.- Conclusiones y trabajo a futuro

Con el robot prototipo se logró:

- Que el docente pueda tener una herramienta para poder instrumentar sus experimentos; ya sean demostrativos o bien que sean elaborados por él mismo, esto con la funcionalidad del robot de poder adquirir datos de sensores, exportar los datos en diferentes formatos y llevar a cabo análisis matemático básico a los datos con experimento.
- Se logró que el alumno pudiera probar con el robot códigos personalizados al hacer que robot pudiera ejecutar las tareas que el alumno había programado; en este caso que el robot pudiera detectar obstáculos y “decidir” qué dirección poder tomar para seguir con su avance.
- Se pudo utilizar el robot para la comprobación de un problema de tiro parabólico de física experimental del bachillerato, proporcionándole al docente una herramienta didáctica con la cual poder comprobar sus experimentos.

Con esto el prototipo, puede ser utilizado como herramienta didáctica en la enseñanza de las ciencias básicas del bachillerato.

El trabajo a futuro es:

- Mejoramiento del mecanismo de lanzamiento al utilizar materiales más ligeros y reducir el tamaño del mismo. También se podría mejorar la funcionalidad del mismo, al hacer que el mecanismo regule automáticamente la compresión del resorte; todo controlado desde las interfaces de usuario.
- La programación puede ser optimizada si se elabora entre varios participantes; puesto que, al realizarla de manera individual se crearon errores que posteriormente intervinieron en el funcionamiento del prototipo.

REFERENCIAS

- [1] Dinwiddie K. *Basic Robotics*, 1st ed USA: Cengage Learning, 2016
- [2] González V. & López A, Cabero E, *Control y Robótica curso en línea*, “ 5.4 Robots Industriales” Marzo 2016: http://platea.pntic.mec.es/vgonzale/cyr_0708
- [3] Valvano J. W. *Introducción a los sistemas de microcomputadores embebidos*, 2da ed McGraw-Hill, 2004, pp 1-7
- [4] Valvano J. W. *Microcomputer-Based Systems. Embedded Microcomputer Systems: Real time interfacing*, 3rd ed. USA: Cengage Learning, 2007, pp 1- 59.
- [5] Barrientos A. & Balaguer C, *Fundamentos de Robótica*, 2da ed. McGraw-Hill, 2012.
- [6] Galeano G. *Programación de Sistemas Embebidos en C*, 4ta ed. Alfaomega, 2009
- [7] Cayssials R. *Sistemas Embebidos en FPGA*, 4ta ed. Alfaomega, 2014, pp 1-36.
- [8] Muhammad A. & Rolin D. & Danny C. *PIC Microcontroller and Embedded Systems*, 3rd ed. Prentice Hall, 2008, pp 1-30
- [9] Martínez L & Guerrero V. & Yuste R. *Comunicaciones Industriales*, 1a ed. Alfaomega, 2009, pp 1-50
- [10] Aquilino R. *Comunicaciones Industriales*, 1a ed. Marcombo, 2008, pp 1- 55
- [11] UNAM, Dirección General de la Escuela Nacional Preparatoria, “*Planes de estudio*” Septiembre 2015: <http://dgenp.unam.mx/planesdeestudio/sexta.html>
- [12] Zabala G. *Robótica: Guía Teórica y Práctica*, 1a ed. Colección Red Users, 2007
- [13] Tojeiro G. *Raspberry Pi2 para electrónicos*, 1a ed. Marcombo, 2016, pp 1-8, pp 58-107
- [14] Texas Instruments, “*MSP430 LaunchPad Value Line Development Kit*”, Febrero 2016: <http://www.ti.com/tool/msp-exp430g2>
- [15] Arduino Italy, “*Arduino Shields*”, Febrero: 2016: <https://www.arduino.cc/en/Main/ArduinoShields>
- [16] Guangzhou HC Information Technology Co., Ltd. “*Product Data Sheet, Module Data Sheet*”, HC Serial Bluetooth Products, 2015.

- [17] Shenzhen Appcon Technologies Co. Ltd, “*APC Series Transparent Transceiver Module APC220-43*”, Appcon Technologies, 2008.
- [18] DFRobot, “*APC220 Radio Communication Module*”, Marzo 2016: http://www.dfrobot.com/index.php?route=product/product&product_id=57#.V6JPkfnhDIV
- [19] National Instruments Corporation. “*NI Community APC220 Example*”, Marzo 2016: <https://decibel.ni.com/content/docs/DOC-32640>
- [20] ELECFreaks, “*Ultrasonic Ranging Module HC-SR04*”, elecfreaks, 2013.
- [21] Freescale Semiconductor, “*Technical Data, Micromachined Accelerometer MMA7361L*”, NXP , 2008
- [22] Reyes F. & Cid J. *Arduino: Aplicaciones en Robótica, Mecatrónica e Ingenierías*, 1a ed. Alfaomega, 2015, pp 51-243, pp 333-344.
- [23] Cogollor J.L. *AutoCAD: Dibujo y Modelado*, 3a ed. RC Libros, 2011.
- [24] AutoCAD 2011, *Curso Online AutoCAD 2011*, “Cursos ONLINE | e-Learning”, Abril 2016: <http://autocad2011libro.blogspot.mx/>
- [25] Rubio E. & Sebastián P. *Ejercicios y problemas de mecanizado*, 3a ed. PEARSON EDUCACIÓN, 2011
- [26] Arduino Italy, *Download the Arduino Software*, “Download”, Abril 2016: <https://www.arduino.cc/en/Main/Software>
- [27] The Eclipse Foundation, *Download Eclipse Technology*, “Download”, Abril 2016: <https://eclipse.org/downloads/>
- [28] Oracle Corporation and/or its affiliates, *NetBeans IDE 8.1 Download*, “Download”, Abril 2016: <https://netbeans.org/downloads/>
- [29] González V, *Materiales didácticos de Control y Robótica . “Servomotores”* Mayo 2016: http://platea.pntic.mec.es/vgonzale/cyr_0204/
- [30] AUTODESK, *Eagle PCB Design*, “Download Eagle”, Mayo 2016: <https://cadsoft.io/>
- [31] ArduBlock, *Getting Started with ArduBlock*, “Download and tutorials of ArduBlock”, Mayo 2016: <http://blog.ardublock.com/en/getting-started-ardublockzhardublock/#>

- [32] Arduino Italy, *Arduino and Java*, Genuino, Mayo 2016: <http://playground.arduino.cc/Interfacing/Java#>
- [33] Pérez M. *Tutorial Java + Arduino – JAVADUINO*, GeekyTheory, Mayo 2016: <http://geekytheory.com/tutorial-java-arduino-javarduino/>
- [34] García G. A. *Arduino*, PanamaHitek, Mayo 2016: <http://panamahitek.com/arduino/>
- [35] Andreas V. *JFreeChart*, Object Refinery limited, Mayo 2016: <http://blog.ardublock.com/engetting-started-ardublockzhardublock/#>
- [36] Monk S. *12 Proyectos Arduino + Android: Controle Arduino con su teléfono inteligente o su tableta*, 1a ed. McGrawHill/Estribor, 2012.
- [37] Monk S. *Arduino + Android Projects for the Evil Genius*, 1a ed. McGrawHill / TAB Electronics, 2012.
- [38] Gironés J. T. *El gran libro de Android*, 2a ed. Alfaomega, 2012
- [39] Vernier Software & Technology, *Arduino and Vernier Sensors*, “Connecting Vernier Sensors to Arduino”, Junio 2016: <http://www.vernier.com/engineering/arduino/connect/>
- [40] SparkFun Electronics, *Vernier Shield Hookup Guide*, “Start a project”, Junio 2016: <https://learn.sparkfun.com/tutorials/vernier-shield-hookup-guide>
- [41] Vernier Software Technology, *Colorimeter*, Vernier 2000.
- [42] Vernier Software Technology, *Stainless Steel Temperature Probe*, Vernier 2016.
- [43] Vernier Software Technology, *Pressure Sensor*, Vernier 2000.
- [44] Hecht E., Frederick J. B., *Física General*, 10a ed. McGraw-Hill, 2007, p-24.