



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**
FACULTAD DE EDUCACIÓN TÉCNICA
PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TEMA:

**Implementación de un robot de batalla de 1 lb y desarrollo de
una aplicación en App Inventor para controlarlo utilizando
comunicación inalámbrica bluetooth**

AUTOR:

Vera Arzube, Leandro Roberto

Componente práctico del examen complejo previo a la
obtención del grado de **INGENIERO EN
TELECOMUNICACIONES**

REVISOR

Palacios Meléndez, Edwin Fernando

Guayaquil, Ecuador

02 de Septiembre del 2016



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

CERTIFICACIÓN

Certificamos que el presente **componente práctico del examen complejo**, fue realizado en su totalidad por **Vera Arzube, Leandro Roberto**, como requerimiento para la obtención del título de **INGENIERO EN TELECOMUNICACIONES**.

REVISOR

Palacios Meléndez, Edwin Fernando

DIRECTOR DE CARRERA

Heras Sánchez, Miguel Armando

Guayaquil, a los 02 del mes de Septiembre del año 2016



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, Vera Arzube, Leandro Roberto

DECLARÓ QUE:

El componente práctico del examen complejo, Implementación de un robot de batalla de 1 lb y desarrollo de una aplicación en App Inventor para controlarlo utilizando comunicación inalámbrica bluetooth previo a la obtención del Título de **Ingeniero en Telecomunicaciones**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 02 del mes de Septiembre del año 2016

EL AUTOR

VERA ARZUBE, LEANDRO ROBERTO



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **Vera Arzube, Leandro Roberto**

Autorizó a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución el **componente teórico del examen complejo, Implementación de un robot de batalla de 1 lb y desarrollo de una aplicación en App Inventor para controlarlo utilizando comunicación inalámbrica bluetooth**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 02 del mes de Septiembre del año 2016

EL AUTOR

VERA ARZUBE, LEANDRO ROBERTO

REPORTE URKUND

URKUND

Documento	Complexivo-Leandro-Vera-2016.docx (D21854355)
Presentado	2016-09-19 14:12 (-05:00)
Presentado por	fernandopm23@hotmail.com
Recibido	edwin.palacios.ucsg@analysis.orkund.com
Mensaje	Revisión EC Vera Arz ube Leandro Mostrar el mensaje completo 1% de esta aprox. 15 páginas de documentos largos se componen de texto presente en 1 fuentes.

Lista de fuentes Bloques

+	Categoría	Enlace/nombre de archivo	▣
+	>	TT-DANIEL-SOLIS-NN-1-2016-FINAL-...	▣
Fuentes alternativas			
+		TT-EDWIN-PINARGOTE-NNN-1-2016-...	▣
+		TT-EDWIN-PINARGOTE-NNN-1-2016-...	▣
+		TT-EDWIN-PINARGOTE-NNN-1-2016-...	▣

Reiniciar Exportar Compartir

0 Advertencias



UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO CARRERA DE INGENIERÍA EN TELECOMUNICACIONES TEMA:

Implementación de un robot de batalla de 1 lb y desarrollo de una aplicación en App Inventor para controlarlo utilizando comunicación inalámbrica bluetooth AUTOR: Vera Arzube, Leandro Roberto



Componente práctico del examen complejo previo a la obtención del grado de INGENIERO EN TELECOMUNICACIONES REVISOR Palacios Meléndez, Edwin Fernando Guayaquil, Ecuador 02 de Septiembre del 2016

UNIVERSIDAD CATÓLICA DE SANTIAGO DE GUAYAQUIL FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO CARRERA DE INGENIERÍA EN

DEDICATORIA

A Dios ya que sin su ayuda y apoyo espiritual no hubiese sido posible finalizar la carrera de Ingeniero en Telecomunicaciones.

A mi esposa Yarima Miranda e hijo Denzel Ezequiel Vera Miranda, porque ellos son mis pilares.

A mi mamá Kelly Arzube Mendoza, tía Erika Arzube Mendoza y en especial a la persona que siempre está ahí para brindar su ayuda y consejos, mi mamá **MARGARITA MENDOZA**.

EL AUTOR

VERA ARZUBE, LEANDRO ROBERTO

AGRADECIMIENTO

Nuevamente mencionaré a mi familia, en especial a mi mamá **MARGARITA MENDOZA.**

A la Universidad Católica de Santiago de Guayaquil, en especial a todos los docentes de la Carrera de Ingeniería en Telecomunicaciones de la Facultad de Educación Técnica para el Desarrollo.

Al Decano, M. Sc. Manuel Romero Paz, por su apoyo brindado, y al Coordinador de Titulación, M. Sc. Fernando Palacios Meléndez.

EL AUTOR

VERA ARZUBE, LEANDRO ROBERTO

ÍNDICE GENERAL

ÍNDICE DE FIGURAS.....	X
ÍNDICE DE TABLAS.....	XI
RESUMEN.....	XII
CAPÍTULO 1: Descripción del Componente Práctico.....	13
1.1. Introducción.....	13
1.2. Objetivo General.	15
1.3. Objetivos Específicos.	15
CAPÍTULO 2: Fundamentación Teórica.....	16
2.1. Fundamentos de microcontroladores.	16
2.1.1. Familia de los microcontroladores PIC.....	17
2.1.2. Familia de los microcontroladores AVR.	24
2.2. Sistema de desarrollo Arduino.....	26
2.3. Desarrollo de aplicaciones móviles.	31
2.3.1. Sistema iOS – Private Source.....	31
2.4. Programación en App Inventor.	35
2.4.1. Desarrollo de aplicaciones para uso personal.....	36
2.4.2. Desarrollar aplicaciones completas.....	37
CAPÍTULO 3: Desarrollo del programa.....	38
3.1. Hardware	38
3.1.1. Motores.....	38
3.1.2. Placa de desarrollo Arduino Nano.	39
3.1.3. Placa de comunicación serial UART Hc-05.....	40
3.1.4. Driver de motor TB6612NF (Puente H).....	40
3.1.5. Control electrónico de velocidad.	41
3.1.6. Motor Brushless.....	42
3.1.7. Placa de control (PCB).	42
3.1.8. Robot de batalla de 1 lb.	45
3.2. Entorno de programación IDE Arduino.....	45

3.2.1. Codificación en Arduino.....	46
3.3. Entorno de desarrollo para creaciones móviles App Inventor.....	48
3.3.1. Codificación App Inventor.....	49
Conclusiones.....	51
Referencias Bibliográficas.....	52

ÍNDICE DE FIGURAS

Capítulo 2:

Figura 2. 1: Diagrama de conexiones del PIC12C de gama baja.	18
Figura 2. 2: Tarjeta Arduino UNO modelo R3.	30
Figura 2. 3: Capas de las tecnologías con las que trabaja iOS.	34
Figura 2. 4: Bloques de App Inventor que especifican la funcionalidad de su aplicación.	35

Capítulo 3:

Figura 3. 1: Micro Motor DC 30:1 a 12 V de Pololu.	38
Figura 3. 2: Tarjeta Arduino Nano V3	39
Figura 3. 3: Modulo Hc05.....	40
Figura 3. 4: Puente H tb6612nf	41
Figura 3. 5: Diagrama interno de un puente h(general).....	41
Figura 3. 6: Control electrónico de velocidad.	42
Figura 3. 7: Motor Brushless (Sin escobilla).....	42
Figura 3. 8: Diagrama de PCB.	43
Figura 3. 9: Diseño Esquemático.	44
Figura 3. 9: Ide arduino.	46

ÍNDICE DE TABLAS

Capítulo 2:

Tabla 2. 1: Características más importantes de la gama baja.....19

Tabla 2. 2: Características más importantes del PIC16X8X de la gama media.
.....22

Tabla 2. 3: Características física e intrínsecas más importantes del
PIC17CXXX de la gama alta.....24

Capítulo 3:

Tabla 3. 1: Relación Velocidad y torque de micro-motores DC de Pololu. ...39

RESUMEN

Durante el desarrollo del componente teórico, se pensó en desarrollar un robot de batalla de 1 lb, ya que el Club de Robótica "ROBOFET" no disponía de este tipo de robot en la categoría de batalla. Primero se realizó una búsqueda bibliográfica de trabajos relacionados en la implementación de robots de batalla, y que en nuestro país este tipo de concursos de robótica inicia en el año 2005 organizado por ESPOL. El objetivo principal de este trabajo de titulación modalidad examen complejo componente práctico fue diseñar un robot de batalla de 1 lb, que tiene incorporado mecanismos de defensa y ataque ofensivo (arma cortadora). Se realiza el código de programación de IDE Arduino para que la tarjeta controladora reciba información mediante Bluetooth y transmita señales a los motores y el arma del robot. También se creó una aplicación para controlar el robot mediante un celular inteligente, el mismo que fue desarrollado en el software App inventor.

Palabras clave: MICROCONTROLADORES, PIC, ATMEL, BLUETOOTH, ARDUINO, APP INVENTOR

CAPÍTULO 1: Descripción del Componente Práctico.

1.1. Introducción.

Las participaciones en concursos o torneos de robótica a nivel local, nacional e internacional por parte de los integrantes del Club de Robótica “ROBOFET”, han ido mejorando en cada una de las categorías en donde son inscritos nuestros robots. Una de las categorías de mayor impacto y atracción, es la batalla de robots de 120 lb, 60 lb, 30 lb, 12 lb, 3 lb y 1 lb. Aunque, ROBOFET no cuenta con robots en esta categoría, el motivo es que implementar un robot de 120 lb sería imposible, debido al elevado costo de los materiales, que la inversión podría superar los \$ 5.000,00.

Por ejemplo, el trabajo de titulación realizado por Quinde Llerena & Ulloa Patiño (2012) muestran el diseño y construcción de dos robots tipo Warbot (batalla o guerra de robots), el mismo que el trabajo fue dividido en dos fases, la parte electrónica y la mecánica. En el proyecto de Ahnn (2007) establecen la comunicación inalámbrica entre el robot móvil y una estación base remota, y también la comunicación serial entre una estación base remota y la aplicación GUI.

Existen otros trabajos, en la que implementan robots de batalla utilizando comunicación RC, por ejemplo, Yoo C. (2007) realiza en la ESPOL el diseño e implementación de un robot tele-manipulado y que fue utilizado en el concurso ecuatoriano de robótica – CER 2005. De manera similar, otro

estudiante de la ESPOL, Ortega N. (2015) realiza el diseño e implementación de un robot de batalla tele-manipulado mediante comunicación RC utilizando la tarjeta controladora inalámbrica Sabertooth 2x60.

Justamente, la ESPOL en el 2005 da iniciativa a los concursos de robótica, ya para el año 2015 la Universidad Católica de Santiago de Guayaquil fue la sede del XI Concurso Ecuatoriano de Robótica (XI CER 2015), y que este año la organización está a cargo de la Universidad Tecnológica Equinoccial (UTE) de Quito a realizarse entre el 30 de noviembre y 2 de diciembre del 2016.

Los robots de batalla utilizan también la comunicación inalámbrica Bluetooth y su control es a través de un teléfono inteligente, por ejemplo, en la ESPOCH el estudiante Torres G. & Lozano G. (2010) implementa un robot de batalla controlado mediante la tecnología Bluetooth. Este trabajo se asemeja mucho al realizado en este componente práctico.

Para diseñar y construir un robot de batalla de 1 lb, debemos considerar el prototipo de chasis liviano, de tal forma que se desplacen sobre dos ruedas, cumpliendo con las normativas y directrices proporcionadas por la categoría. Mientras que este robot, tendrá que ser capaz de atacar y vencer al oponente durante el tiempo asignado.

La mayoría de participantes en torneos de robótica, prefieren construir robots pequeños, como el de 1 lb, ya que, la inversión es mínima, aproximadamente unos \$ 350,00. Durante los tres años de creado el Club de Robótica "ROBOFET" se ha considerado la posibilidad de invertir en el diseño y construcción de un robot de batalla, pero debido a la inversión, nos decidimos por el robot de 1 lb.

1.2. Objetivo General.

Implementar un robot de batalla de 1 lb y desarrollar una aplicación en App Inventor para controlarlo mediante comunicación inalámbrica bluetooth.

1.3. Objetivos Específicos.

- Describir la fundamentación teórica de los microcontroladores, dispositivos Arduino, aplicaciones móviles y App Inventor.
- Desarrollar la aplicación para controlar el robot de batalla de 1 lb utilizando el programa App Inventor.
- Implementar un robot de batalla de 1 lb controlado por un dispositivo electrónico usando tecnología Bluetooth.

CAPÍTULO 2: Fundamentación Teórica.

En este capítulo se describen las bases fundamentales de los microcontroladores, de aplicaciones en dispositivos móviles y del programa App Inventor.

2.1. Fundamentos de microcontroladores.

Se conoce actualmente a un microcontrolador como aquel circuito integrado de tipo digital estructurado con un material monolítico en el cual están situados en su interior el mismo tipo de elementos que se encuentran dentro de un procesador digital secuencial de tipo síncrono programable. Poseen arquitecturas conocidas actualmente como la Harvard o la Princeton. En algunos libros de texto académicos se nos nombran como microcomputador integrado donde lo enfocan principalmente a funciones de control o comunicaciones.

Con respecto a sus dimensiones de tamaño, poseen una escala muy pequeña, la cual es muy útil para varias aplicaciones o implementaciones a empresas o industrias que se enfocan en elaboración de mecanismos automáticos o de control autónomos. Debido a sus considerables bajos costos comerciales también son muy considerados en la implementación de esos sistemas. Según (Pérez, 2007) los microcontroladores poseen características importantes como su bajo consumo de energía y velocidades de procesamiento ajustable o adaptable dependiendo de la necesidad. Poseen

propiedades de seguridad de funcionamiento y mecanismos de protección electrónica o de edición de programa sin autorización.

Para dar ejemplos de donde se utilizan comúnmente los microcontroladores de manera comercial, tenemos los sistemas eléctricos, mecánicos, por ejemplo, electrodomésticos en los cuales están las lavadoras, televisores, microondas, aires acondicionados, y equipos similares. En los sistemas informáticos tenemos los usados comúnmente en oficina como impresoras, monitores, ratones, faxes, parlantes, proyectores, etc. Y en los sistemas de telecomunicaciones actualmente utilizados están los circuitos de conmutación, Nodos, dispositivos de modulación, dispositivos de transmisión, amplificación, teléfonos móviles, etc. (Aguayo, 2014)

2.1.1. Familia de los microcontroladores PIC.

Cuando de un microcontrolador se refiere, la familia PIC es la más conocida entre todas, no solo por su facilidad que tiene en el uso, sino también se considera su tipo de programación y la integración que se tiene en cuenta en las aplicaciones usuales que se requieren tanto en el aprendizaje y el ámbito laboral.

Con respecto a la familia PIC se puede decir que ha sido desarrollada por Microchip, donde son divididas en cuatro subcategorías como la gama enana, la baja, la gama media y alta (Pérez, 2007). Con respecto a estas subcategorías una de las diferencias más importantes a considerar serian la

cantidad de funciones e instrucciones que se pueden establecer en el microcontrolador; también es importante nombrar las cantidades de puertos, es decir las entradas o salidas posibles de usar.

Con respecto a los voltajes de funcionamiento de la familia PIC operan a un rango de 2.5 a 5 Voltios y con respecto al amperaje consumido está alrededor de 0.02 amperios usando los 5 voltios máximos y a una frecuencia de operación de 4 MHz, determinados en la hoja de especificaciones (Aguayo, 2014). Correspondiente a la gama baja poseen una característica reconocible física de 8 pines y de dimensiones muy pequeñas, y un tamaño muy reducido en comparación a las demás gamas. Este tipo de microcontrolador puede trabajar de 12 o 14 bits dependiendo del formato que tengan las instrucciones establecidas, es decir que puede aceptar hasta un máximo de 33 o 35 instrucciones durante la operación del mismo. En la Figura 1 se puede observar las configuraciones de los pines de este tipo de PIC. (Pérez, 2007)

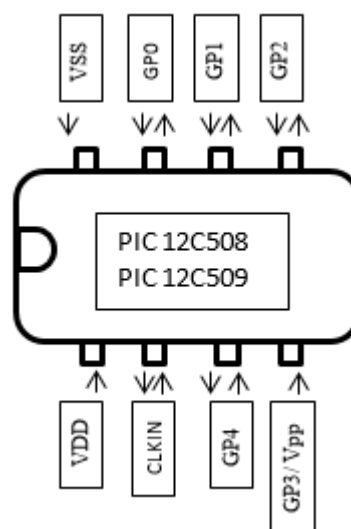


Figura 2. 1: Diagrama de conexiones del PIC12C de gama baja.
Fuente: (Aguayo, 2014)

Los PIC de la gama baja tienen varias cualidades, pero la más importante, es la de disponer dentro un oscilador de tipo R-C, lo cual le permite establecer de los 8 pines, hasta 6 pines de entrada o salida para los demás periféricos. En la tabla 2.1 se quiere mostrar los modelos más usados de esta gama, podemos observar también las características de hardware y software que poseen.

Tabla 2. 1: Características más importantes de la gama baja

Modelo PIC	Memoria de Programa	Memoria de Datos	Frecuencia de operación	E/S	Temporizadores	Pines
12C508	512X12	25X8	4 MHz	6	Tmr0+WDT	8
12C509	1024X12	41X8	4 MHz	6	Tmr0+WDT	8
12C670	512X12	80X8	4 MHz	6	Tmr0+WDT	8
12C671	1024X14	128X8	4 MHz	6	Tmr0+WDT	8
12C672	2048X14	128X8	4 MHz	6	Tmr0+WDT	8
12C680	512X12 FLASH	80X8 EEPROM	4 MHz	6	Tmr0+WDT	8
12C681	1024X14 FLASH	80X8 EEPROM	4 MHz	6	Tmr0+WDT	8

Fuente: (Aguayo, 2014)

Como se aprecia en la tabla 2.1, con respecto a la gama baja podemos concluir que los PIC12C5XX pertenecen a esta gama, por lo que poseen un máximo de 12 bits de instrucciones.

Por otro lado, tenemos la gama media donde convergen los modelos PIC12C6XX, estos modelos poseen una mayor capacidad con respecto a la gama baja, pues asimilan hasta un máximo de 14 bits en sus instrucciones (Aguayo, 2014). En esta gama también encontramos a los PIC12F6XX lo

cuales incluyen una memoria de tipo flash para la ejecución de los programas y una memoria de tipo EEPROM la cual es asignada para el desarrollo de datos en el PIC.

Hablemos entonces de la gama media, esta gama de PIC se enfoca en los modelos de PIC16CXXX los cuales en su mayoría poseen una cantidad de instrucciones de 14 bits, en sus características físicas la más común es su cantidad de pines, un total de 18 hasta 68, es decir que está enfocado en utilizar varios periféricos conectados al PIC. El modelo más conocido de esta gama son los tipos PIC16X84. (Pérez, 2007)

Esta gama fue creada con el fin de mejorar la gama anterior, es decir conservar las características más usadas e importantes de la gama baja en una gama más alta, para mejorar el funcionamiento y potenciar el procesamiento de operaciones un poco más complejas. Pueden aceptar interrupciones y a su vez cuentan con comparadores de cantidades tanto analógicas como los convertidores de analógico a digital, los puertos serie y varios tipos de temporizadores en su estructura. Posee un repertorio de 35 instrucciones como máximo con un tamaño de 14 bits para cada instrucción (Aguayo, 2014). Tienen la compatibilidad de trabajar también con la gama baja. En sus modelos la gama media tiene los elementos necesarios para trabajar en aplicaciones de microcontroladores que trabajan en 8 bits. La gama media es capaz de realizar sub rutinas.

Con respecto a las características más importantes están el temporizador TMR1 con el que trabajan, el cual posee un circuito de tipo oscilador con el cual se puede trabajar de forma asíncrona y a su vez dar la posibilidad de aumentar o incrementarse sin importa que el microcontrolador se encuentre en un estado de Reposo o Sleep.

Esta propiedad posibilita que el microcontrolador pueda trabajar con un control de reloj de tiempo real. Con respecto a la alimentación, la gama media trabaja con baterías de Li-Ion, NiCd, Zinc, Ph o NiMH (Aguayo, 2014). Lo cual facilita mucho en el momento de realizar alguna implementación estudiantil o trabajo técnico, debido a que la mayoría de estas baterías poseen una buena cantidad de almacenamiento y de descarga; considerando también su tamaño y peso.

Para terminar con las características de la gama media, esta posee un rango de alimentación de 2 a 6 Voltios, con una cantidad de 4 interrupciones y de 13 entradas o salidas respectivamente. Posee también una memoria volátil de datos de 36 o 68 bytes. O a su vez una memoria EEPROM también asignada a datos de 64 bytes en todos los diseños. (Pérez, 2007)

A continuación, se dispone la tabla 2.2 con los modelos restantes de PIC16XXX y sus características. (Pérez, 2007)

Tabla 2. 2: Características más importantes del PIC16X8X de la gama media.

Modelo	Memoria de Programa	Memoria de datos	Registros específicos	Temporizadores	Interrupciones	E/S	Pines
PIC16C84	1kx14	64	11	TMR0 + WDT	4	13	18
PIC16F84	1kx14	64	11	TMR0 + WDT	4	13	18
PIC16F83	512x14	64	11	TMR0 + WDT	4	13	18
PIC16CR84	1kx14	64	11	TMR0 + WDT	4	13	18
PIC16CR83	512x14	64	11	TMR0 + WDT	4	13	18

Fuente: (Aguayo, 2014)

Continuemos con la última gama restante, la gama alta. En esta gama encontramos a los PIC17CXXX los cuales tienen una característica muy notable debido a que poseen un rendimiento superior a las dos gamas anteriores. Llegan a una cantidad muy elevada de 58 instrucciones cuyos tamaños son de 16 bits. Como podemos apreciar es un nivel de procesamiento muy alto y potente, a parte de esta característica también tienen un manejo de interrupciones de forma de vectores bastante eficiente. (Pérez, 2007)

Esta gama también tiene estructurado controladores de distintos tipos de periféricos acoplados, en la comunicación es posible realizar una en serie o en paralelo con la posibilidad de emplear elementos externos para enlazar. Contiene un hardware de tipo multiplicador con una velocidad elevada. En la memoria asignada a datos o a programas, su capacidad es mucho mayor

respectivamente. Con una capacidad de 8000 palabras asignadas a la memoria a instrucciones del PIC y también una cantidad de 454 bytes establecida a la memoria de datos. (Aguayo, 2014)

Una de las características más importantes de esta gama se encuentra su tipo de arquitectura de tipo abierta, en la cual existe la posibilidad de una extensión de un microcontrolador incluyendo los elementos externos para el ensamblaje de alguna circuitería. Aquí es donde la cantidad de pines que posee la gama alta entran en función, en la cual se pueden apreciar las pistas de los buses en los que se transmiten datos, controles o direcciones.

En este tipo de buses van unidas a memorias o periféricos de control. Esta propiedad nos indica que la cantidad de pines que debería tener el PIC de gama alta deberían ser un número algo elevado en comparación a los otros PIC ya revisados. Los PIC pertenecientes a esta gama poseen una cantidad razonable de 40 a 44 pines.

Entendemos entonces que las capacidades de arquitectura de un microcontrolador de este tipo son muy considerables iguales a las de un microprocesador, pues tienen un procesamiento muy parecido con respecto a su sistema. Para dar a conocer las características más importantes de los PIC que se encuentran en la gama alta observemos la tabla 2.3.

Tabla 2. 3: Características física e intrínsecas más importantes del PIC17CXXX de la gama alta.

Modelo	Memoria de Programa	Memoria de datos	Registros específicos	PWM	Temporizadores	E/S	Pines
17C42A	2KX16	232	48	2	4 + WDT	33	40/44
17C43	4KX16	454	48	2	4 + WDT	33	40/44
17C41	8KX16	454	48	2	4 + WDT	33	40/44
17C752	8KX16	454	76	3	4 + WDT	50	64/68
17C756	16KX16	902	76	3	4 + WDT	50	64/68

Fuente: (Aguayo, 2014)

2.1.2. Familia de los microcontroladores AVR.

En esta sección se describirá brevemente los microcontroladores AVR, aunque hay muy poca información técnica, sin embargo, se muestra lo más importante. Este microcontrolador ha sido desarrollado para poder ser programado utilizando el lenguaje C. El lenguaje más común en el mundo de la programación y de fácil comprensión. El microcontrolador AVR abarca múltiples modelos, pero todos comparten el mismo modelo de núcleo y estructura. Aunque cada uno posee distintos periféricos y también sus propios niveles de memoria con respecto a la RAM o la ROM.

Los AVR poseen una alta compatibilidad con los periféricos que se encuentran hoy en día en el mercado, también dan la posibilidad de realizar ejecuciones de varias instrucciones con un par de etapas que se denominan cargar – ejecutar. Esto le permite al microcontrolador dar ejecución de una

muy importante cantidad de instrucciones en un mismo ciclo de reloj. Eso quiere decir que en comparación a los microcontroladores 8 bits, los AVR son mucho más rápidos. (Aguayo, 2014)

Con respecto a la comunicación que tienen los AVR, están las interfaces de tipo SPI, USB, I2C, etc. Y al igual que los PIC, también vienen integrados con los muy útiles conversores del analógico a digital (A/D), oscilador de tipo RC y la memoria de clase EEPROM de almacenamiento interno. Además, con respecto a sus categorías están separados en Gamas (baja, media y alta). (Pérez, 2007)

En la gama baja correspondiente a los AVR podemos encontrar a los modelos Tiny AVR. Estos microcontroladores poseen una memoria de almacenamiento Flash de capacidad de 2KBytes y de 128 Bytes en medias de tipo EEPROM. Son enfocados en un uso general, para implementaciones educativas o talleres prácticos. (Pérez, 2007)

En la gama media tenemos al modelo LCD AVR. Este microcontrolador de uso general de igual manera posee una memoria flash de un máximo de 8 Kbytes y de memoria de tipo EEPROM de 512 bytes algo un poco superior a la gama baja con respecto a la cantidad de memoria disponible. Con respecto a los de la gama Alta tenemos a los Mega AVR. Los cuales tienen un encapsulado con un máximo de 100 pines y un mínimo de 28 pines. Se los

puede energizar con un voltaje de 1.8 hasta 5.5. En la memoria, posee una Flash de hasta 256 KBytes. (Aguayo, 2014)

2.2. Sistema de desarrollo Arduino.

Al referirnos a una placa impresa estamos enfocándonos directamente a una PCB. Un PCB es una tarjeta que la mayoría de veces son de dimensión pequeña y son de material de plástico, cerámica o común mente utilizado con resinas de fibras de vidrio. Esta tarjeta tiene en una superficie el diseño plasmado de pistas de un circuito con todos sus elementos distribuidos en forma de mapa. La mayoría de casos las pistas pertenecen a un material conductor de cobre.

La PCB es muy parecida a una protoboard debido a que se utiliza para realizar el montaje de circuitos eléctricos la única diferencia es que una PCB no se puede modificar una vez impresa la tarjeta, pero la protoboard puede ser modificada la cantidad de veces necesarias debido a que es una tarjeta perforada. Conociendo esto podemos entender porque la tarjeta arduino se enfoca en usar una PCB en vez de usar una protoboard.

Con respecto a la tarjeta arduino existen muchos modelos en el mercado tanto en tamaño como en procesamiento. Entre las características más importantes están las dimensiones físicas la cantidad de puertos disponibles, y la clase de microcontrolador con el que trabajará arduino. Es importante también entre las características físicas establecer cuanta memoria puede ser

utilizada. Todas estas características desplegadas anteriormente nos ayudaran a reconocer que tarjeta arduino es la que vamos a necesitar adquirir. (Artero, 2013)

La segunda característica de la que vamos hablar es el software con el que se va a trabajar en el arduino, este software de tipo OP-SOURCE y también posee una característica de ser multiplataforma, debido a que puede ejecutarse en varios sistemas operativos tales como Windows, Linux, MacOC (Artero, 2013).

Al tener este software instalado en nuestro sistema operativo no va a dar la posibilidad de poder crear editar o corregir el programa que se vaya a subir a la memoria del arduino. Es decir, esto nos va permitir programarlo las veces que queramos, y nos facilitara al momento de hacer pruebas con nuestro circuito ya ensamblado.

La conexión de este software al arduino es por vía USB, los trabajos realizados en arduino pueden trabajar de manera autónoma o a través de una transmisión de información (módulos bluetooth, infrarrojo, GPS, etc.). Una vez que la tarjeta asido cargada con la programación que nosotros hemos creado, la tarjeta puede ser desconectada del computador y trabajara de manera independiente utilizando una alimentación de hasta 9 voltios. Esta tarjeta puede trabajar de manera serial conectada permanentemente a una computadora en la cual va haber una transmisión y recepción de datos

mediante los periféricos que se vayan a utilizar en la circuitería, esta forma podemos trabajar con la tarjeta y ver los resultados en nuestra pantalla del ordenador. (Artero, 2013)

La tercera característica de la que cuenta arduino es su lenguaje de programación el cual es de tipo libre. Lo que se quiere dar a entender es que emplea un idioma de programación de clase artificial el cual ha sido elaborado para el fácil entendimiento del usuario y la fácil comprensión de la maquina al momento de compilarlo.

El lenguaje de arduino contiene funciones en su programación las cuales son muy similares a otros tipos de lenguajes que sirven para programar tales como lazos repetitivos, comparadores lógicos, interrupciones y establecimiento de variables muy parecido al lenguaje C. Esto ayuda al usuario a programar de una manera más rápida y fácil evitando errores durante las instrucciones bridadas al microcontrolador que posee el arduino.

Con respecto a las características que posee el microcontrolador que trabaja en la placa arduino, existen en el mercado distintos modelos de placas, y todas ellas con características enfocadas al tamaño, al peso, a la cantidad de puertos. Dependiendo de qué condición le convenga más al usuario. Aun así, se puede definir el modelo básico de la tarjeta que es el más adquirido entre los usuarios que utiliza arduino y que es el que vamos a describir a continuación.

La tarjeta arduino tiene como nombre Arduino UNO. Es una tarjeta que surgido en un tiempo relativamente actual siendo su origen en el 2010 se han elaborado hasta la fecha tres cambios en su estructura, provocando la creación de varios modelos en la cual se los pueden encontrar en el mercado como Arduino UNO R3.

Ahora, con respecto al modelo del microcontrolador que va trabajar la tarjeta Arduino UNO, será la serie ATmega328P de la marca ATMEL. La p que se encuentra al final del modelo del microcontrolador va ser referencia al uso del tipo de tecnología Picopower, el cual es una autoría de ATML con lo que se puede disminuir la cantidad de corriente con el que va a trabajar en comparación al ATmega328. Es decir que este microcontrolador va a necesitar una menor alimentación de voltaje y por ende va requerir de menos cantidad de corriente, los dos modelos funcionan de igual forma. (Artero, 2013)

El microcontrolador ATmega328P con respecto a su estructura de su diseño entran en la categoría AVR, quiere decir que fue construida en la familia ATML, por lo que entra en una categoría de microcontroladores más vendidos en el mercado por ende le hace competencia a los demás microcontroladores existentes que tenemos por ejemplo el microcontrolador Pic que fue ensamblado en la empresa micro chip, al igual que los otros microcontroladores AVR se divide en varias subcategorías, donde estamos hablando del ATmega328P el cual le pertenece en este caso al arduino y con

respecto a la familia ATML entra en la sección MegaAVR. Existen varios modelos de AVR muy utilices disponibles, pero nos enfocaremos en el ATmea328p debido que arduino trabaja únicamente con este modelo.

Una vez enfocándonos en el ATmega328P sería conveniente revisar las características físicas en las cuales ha sido ensamblado este microcontrolador. Entre esas características están las cantidades de pines disponibles, tanto en puertos como en alimentación, como también en dimensiones físicas y pesos. Es necesario entonces, tener un diagrama de las ubicaciones de los pines en relación con el tipo de entrada o salida que es aceptado por el microcontrolador, es decir, analógica o digital, de transmisión o de recepción, reset, etc.

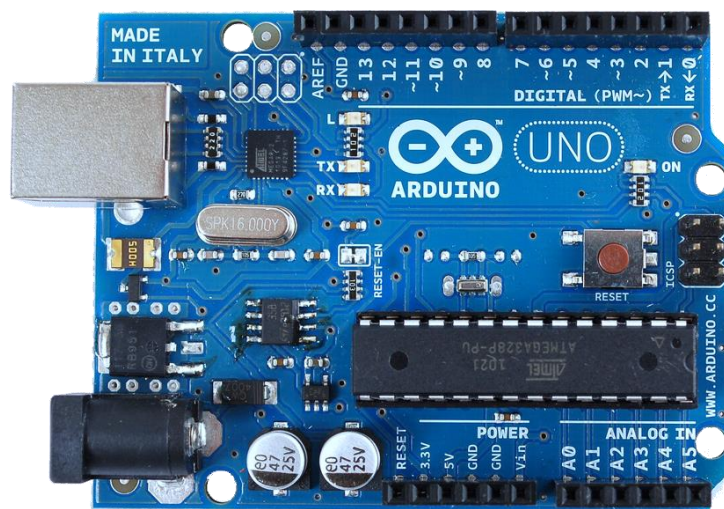


Figura 2. 2: Tarjeta Arduino UNO modelo R3.
Fuente: (Artero, 2013)

Con respecto a la figura 2.2 podemos darnos cuenta que los pines siete y ocho se enfocan en alimentación de energía al microcontrolador, posee un punto en común en ambos lados del chip conocido como GND o tierra, un pin

dedicado exclusivamente al reset, el cual va a detener todo proceso en ejecución y reiniciara la toda la programación desde el principio. Con respecto a los pines de entrada y salida (véase figura 2.2) se puede ver expresado con las siglas PB, PC y PD, existen dos pines que sirven para alimentar el convertidor de analógico a digital que se encuentra incorporado en el interior del microcontrolador y los dos últimos pines (dos y tres) pertenecen a la recepción y trasmisión de datos respectivamente que muy útiles cuando se utilizan para comunicaciones inalámbricas en la implementación del circuito enfocado en algún proyecto estudiantil o laboral. (Artero, 2013)

2.3. Desarrollo de aplicaciones móviles.

En esta sección se describe

2.3.1. Sistema iOS – Private Source.

El sistema operativo iOS fue presentado el 29 de junio del 2007 en los celulares iPhone, cuando en una venta lograron vender alrededor de doscientos setenta mil celulares en tan solo dos días. Y una cantidad asombrosa de alrededor de un millón de unidades en un mes y medio. Cuando el sistema operativo fue presentado a los usuarios no existían desarrolladores de software pertenecientes a Apple, pero la empresa anuncio que no habría necesidad de crear uno debido a que se podrían utilizar desarrolladores ya existentes en el mercado. Desarrolladores basados en HTML, en CSS o también en JavaScript, algo muy parecido a lo que utiliza hoy en día el sistema operativo Android (open source).

Este sistema esta creado para que comunique el hardware con hardware de manera directa, es decir de la aplicación que aparece en la pantalla táctil, a la función que quiere establecer en el teléfono, sea la utilización de cámaras, reproducción de música, navegar por internet, etc. Lo que lleva a entender que las aplicaciones que se instalan en el sistema operativo rara vez interactúan con el hardware directamente. Lo que conlleva a pensar que utilizará un sistema de interfaces entre hardware, lo que, en cierta manera, servirá de protección a aplicaciones como malware que realizan cambios sin autorización al hardware o memorias del software.

Con respecto a las aplicaciones que se utilizan en el sistema operativo iOS, podemos decir que son desarrolladas con un lenguaje de programación orientado a objetos, la misma lógica de programación que se utiliza en Visual Basic al crear formularios. A esta lógica de programación se la conoce con el nombre de Objetivo C. Con respecto a esta programación, se puede decir que utiliza toda su estructura del lenguaje como un objeto, es decir partes y sub partes. En pocas palabras, La programación Objetivo C incluye una comunicación de tipo Small Talk, al mismo lenguaje C. (Eero Maasalmi, 2011)

Al poder observar las características con las que está estructurado el sistema operativo iOS, se puede llegar a entender que puede ser un sistema operativo algo inestable y de uso en ordenadores computacionales, pero no es así. iOS fue creado con el fin de utilizarse en telefonía y dispositivos

móviles. Por lo que comparte un núcleo igual al modelo básico de Kernel (MACH KERNEL).

Un núcleo parecido es utilizado en las Mac OS X. Con respecto al Kernel, trabaja con capas las cuales se dividen en 4 tecnologías (véase la figura 2.3). Nombrando a las tecnologías tenemos el Core OS el cual va a trabajar al nivel del núcleo del sistema, es decir en el núcleo interno del Kernel. Tenemos Core Services, el cual va a enfocarse en el nivel de la comunicación entre el núcleo y las aplicaciones o funciones ejecutadas en ese momento.

Luego tenemos el Media, se enfoca directamente en los procesos posibles que son brindados por las aplicaciones instaladas en el dispositivo móvil. Y por último esta Cocoa Touch, enfocado directamente a las ubicaciones y logaritmos que son establecidos para la ubicación de las acciones introducidas por el usuario a través del táctil. Es decir, posee las características y herramientas para que el usuario pueda utilizar las aplicaciones a través de pulsos.

A continuación, tenemos la figura 2.3 donde se muestran las capas del modelo iOS (Eero Maasalmi, 2011). Con lo visto en referencia al modelo del sistema operativo iOS, podemos concluir que la manera de interactuar con el usuario, será igual que en cualquier dispositivo táctil, el cual emplea algún desliz, un pellizco de equina a esquina, o simples toques. Con respecto a la interactividad, este sistema utiliza acelerómetros, los cuales sirven en el

momento de que el usuario necesite realizar cualquier movimiento al dispositivo como sacudir el teléfono, aunque de manera tradicional, esto se enfoca en el sentido de orientación del teléfono, al cambiar la pantalla de una forma vertical a horizontal. (Eero Maasalmi, 2011)

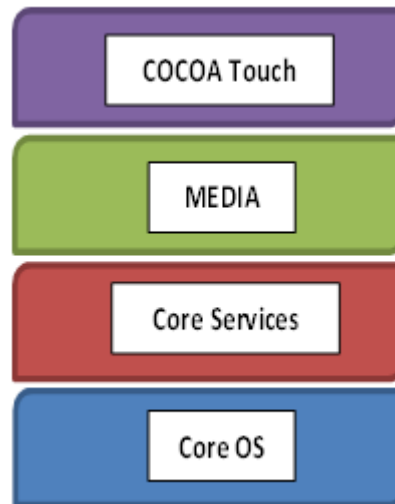


Figura 2. 3: Capas de las tecnologías con las que trabaja iOS.
Fuente: (Eero Maasalmi, 2011)

Con respecto a la seguridad, el sistema operativo iOS, posee hasta la actualidad el sistema de seguridad más seguro disponible en el mercado. Y esto fue generado debido a la cantidad muy elevada de robos que se producían cuando el dispositivo salió a la venta. Por ende, la necesidad de hacerlo un sistema de seguridad más robusto fue más que evidente. Se logra este nivel de seguridad debido al registro único y privado de datos en el iCloud, solicitando los datos que brindan acceso al dispositivo que se encuentran en la cuenta original que le pertenece a cada dueño. En caso contrario, el teléfono queda completamente inservible.

2.4. Programación en App Inventor.

App Inventor es una herramienta muy parecido a una plataforma de tipo visual, que se enfoca en arrastrar y soltar para la creación de nuevas aplicaciones para dispositivos móviles en la plataforma Android. Aquí diseñas la interfaz de usuario (el aspecto visual) de una aplicación usando un constructor basado en web que trabaja en la interfaz gráfica de usuario, a continuación, se especifica el comportamiento de la aplicación juntando "bloques", como si estuviera trabajando en un rompecabezas. (David Wolber, 2011)

En la figura 2.4, se muestran los bloques de una primera versión de una aplicación creada por Daniel Finnegan, un estudiante de la Universidad que nunca habían programado antes. Podemos observar que la función de este programa es simple de entender.

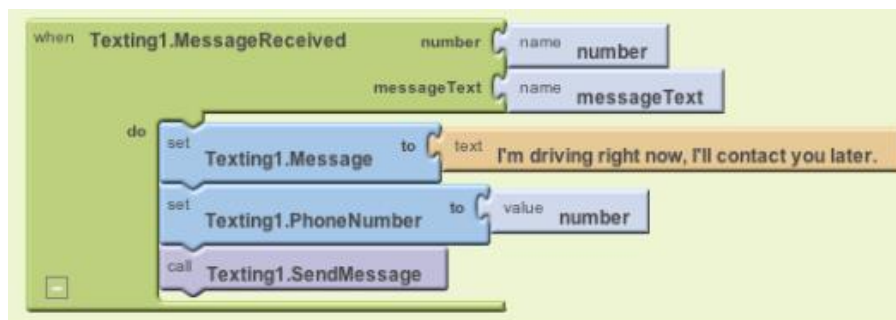


Figura 2. 4: Bloques de App Inventor que especifican la funcionalidad de su aplicación.

Fuente: (David Wolber, 2011)

La aplicación es un texto "contestador automático". Que lo ejecute cuando se está conduciendo y se auto responde a los textos que recibe. Debido a que los bloques son más comprensibles que el código de

programación tradicional, el usuario del App Inventor lo puede entender inmediatamente. Esto nos abre un mundo entero de posibilidades a personas que no han programado nunca antes en su vida.

Debido a que los seres humanos nos gustan aprender cosas de manera fácil, y a base de esto, poder ir mejorando lo que ya ha sido creado. Por ejemplo, utilizar sensores de sonido para poder responder el mensaje de texto que se ha recibido, o realizar una llamada sin necesidad de tocar el dispositivo móvil. Entre las cosas que se pueden hacer en App Inventor, es desarrollar juegos para teléfonos inteligentes, y que son entretenidos.

Sólo tiene que abrir la aplicación Inventor en un navegador web, conectar el teléfono y empezar a poner juntos los bloques como los de la figura 2.4. Inmediatamente se puede ver e interactuar con la aplicación que está construyendo en el teléfono. Así que usted está programando, pero también estás enviando un correo electrónico a su amigo que le envíe un texto para probar su aplicación, o que está controlando un robot LEGO NXT™ con la aplicación que acaba de construir, o si buscas desconectar el teléfono y caminar afuera para ver si su aplicación está utilizando el sensor de localización correctamente. (David Wolber, 2011)

2.4.1. Desarrollo de aplicaciones para uso personal.

En el estado actual del mundo de aplicaciones móviles, estamos atascados con las aplicaciones que nos dan. ¿Quién no se ha quejado de una

aplicación y deseado que esta pueda ser personalizada o ajustarse de alguna manera? Con App Inventor, puede crear una aplicación exactamente como usted lo desee. Por ejemplo, se puede construir un juego Mole Mash que le permite ganar puntos por tocar un lunar en movimiento al azar (David Wolber, 2011). Pero en lugar de utilizar la imagen del topo en el tutorial, se puede personalizar para que aplaste una foto de su hermano o hermana, algo que quien hace el programa puede hacer sin mucha dificultad.

2.4.2. Desarrollar aplicaciones completas.

App Inventor es no sólo un sistema de prototipo o una interfaz de diseño, se puede construir programas completos, y aplicaciones de uso general. El lenguaje proporciona todos los bloques de construcción fundamental de programación como bucles y condicional, pero en forma de bloques.

CAPÍTULO 3: Desarrollo del programa.

En este capítulo se muestra la parte del aporte práctico interdisciplinario, en la que se indica el diseño del robot de batalla de 1 lb, así como el diseño de la aplicación en App Inventor que permitirá controlar al robot mediante comunicación inalámbrica bluetooth.

3.1. Hardware

3.1.1. Motores.

Para el diseño del robot de batalla de 1 libra es necesario el uso de micro-motores. Estos funcionan a corriente directa (DC) y tienen una relación de engranajes, que van desde 5: 1 hasta 1000:1, pero para el presente trabajo utilizaremos el micro-motor DC a 12 V con relación 30:1, cuya velocidad es de 650 rpm y torque 1.2 kg.cm, las dimensiones son 10 x 12 mm (0.39 "x 0,47"), tal como se muestra en la figura 2.1.



Figura 3. 1: Micro Motor DC 30:1 a 12 V de Pololu.

Fuente: (Pololu, 2016)

En la tabla 3.1 se muestran las relaciones de engranajes de los micro-motores DC de alto rendimiento (potencia) con escobillas de carbón de larga duración (HPCB).

Tabla 3. 1: Relación Velocidad y torque de micro-motores DC de Pololu.

Rated Voltage	Motor Type	Stall Current @ Rated Voltage	No-Load Speed @ Rated Voltage	Approximate Stall Torque @ Rated Voltage
12 V	high-power, carbon brushes (HPCB)	800 mA	6000 RPM	2 oz-in
			3000 RPM	4 oz-in
			1000 RPM	9 oz-in
			625 RPM	15 oz-in
			400 RPM	22 oz-in
			320 RPM	30 oz-in
			200 RPM	40 oz-in
			140 RPM	50 oz-in
			120 RPM	60 oz-in
			100 RPM	70 oz-in
			32 RPM	125 oz-in

Fuente: (Pololu, 2016)

3.1.2. Placa de desarrollo Arduino Nano.

En la figura 3.3 se muestra la tarjeta Arduino Nano. Es una placa de desarrollo de hardware y software que contiene un microcontrolador Atmel (Atmega328p) y poseen su propio IDE, donde se puede programar en lenguaje de alto nivel (Lenguaje C) de acuerdo a las especificaciones del usuario. Este dispositivo, contiene tanto entradas, como salidas digitales y entradas analógicas.



Figura 3. 2: Tarjeta Arduino Nano V3
Elaborado por: arduino.cc

3.1.3. Placa de comunicación serial UART Hc-05.

El módulo de comunicación UART (véase la figura 3.3) permite el enlace entre 2 periféricos usando como protocolo la comunicación Bluetooth teniendo un transmisor y un receptor de datos. Este módulo de comunicación de punto a punto poseen 2 modos de trabajo, los cuales son maestro y esclavo.



Figura 3. 3: Modulo Hc05.
Elaborado por: Autor

3.1.4. Driver de motor TB6612NF (Puente H).

Este driver (véase la figura 3.4) controla los motores eléctricos DC, que permiten el giro del motor en giros horarios y anti horarios, así como la velocidad del mismo. Son ampliamente usados en robótica en la etapa de potencia para poder usar elementos con un amperaje considerable para la aplicación que sea pertinente. Internamente posee cuatro transistores (véase la figura 3.5) y cuatro diodos npn, los cuales permiten habilitar el control de giro y velocidad.



Figura 3. 4: Puente H tb6612nf
Elaborado por: Autor

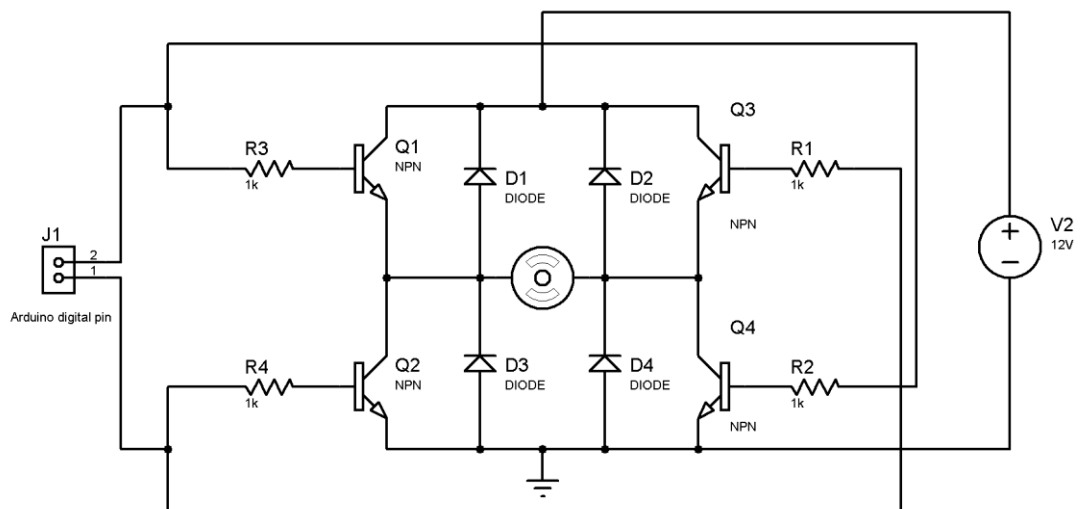


Figura 3. 5: Diagrama interno de un puente h(general).
Elaborado por: Autor

3.1.5. Control electrónico de velocidad.

Este periférico actúa como un controlador electrónico de velocidad, que es capaz de definir la velocidad de giro de un motor brushless (muy utilizado en robots voladores) mediante la generación de anchos de pulsos (PWM). En la figura 3.6 se muestra el controlador para velocidad de motores DC.



Figura 3. 6: Control electrónico de velocidad.

Elaborado por: Autor.

3.1.6. Motor Brushless.

En la figura 3.7 se muestra un motor brushless (sin escobilla) es capaz de moverse gracias a sus múltiples rotores, que permiten al motor llegar a una alta velocidad para mover sus hélices. Una de las ventajas de los motores eléctricos, es que estos producen un desgaste mucho menor a los motores normales.



Figura 3. 7: Motor Brushless (Sin escobilla).

Elaborado por: mobus.es

3.1.7. Placa de control (PCB).

Esta placa de control es una plaqueta de circuito impreso, están compuestas de una superficie constituida por vías, pistas o buses de material conductor laminadas sobre una base no conductora. La placa

controladora consta con 4 periféricos que se muestran en la figura 3.8, y que son: tarjeta control, hc 05, motor tb6612nf, ESC.

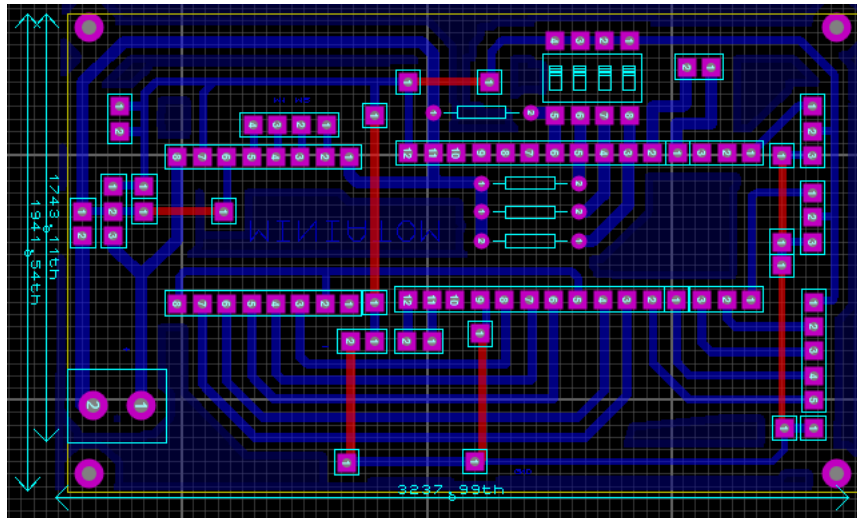


Figura 3. 8: Diagrama de PCB.

Elaborado por: Autor

Para la implementación es necesario conectar el puente h a los pines PWM del controlador y enviar la velocidad, sentido del motor. Para la utilización de la comunicación es necesario conectar a los puertos de transmisión y recepción serial en la placa de desarrollo arduino nano.

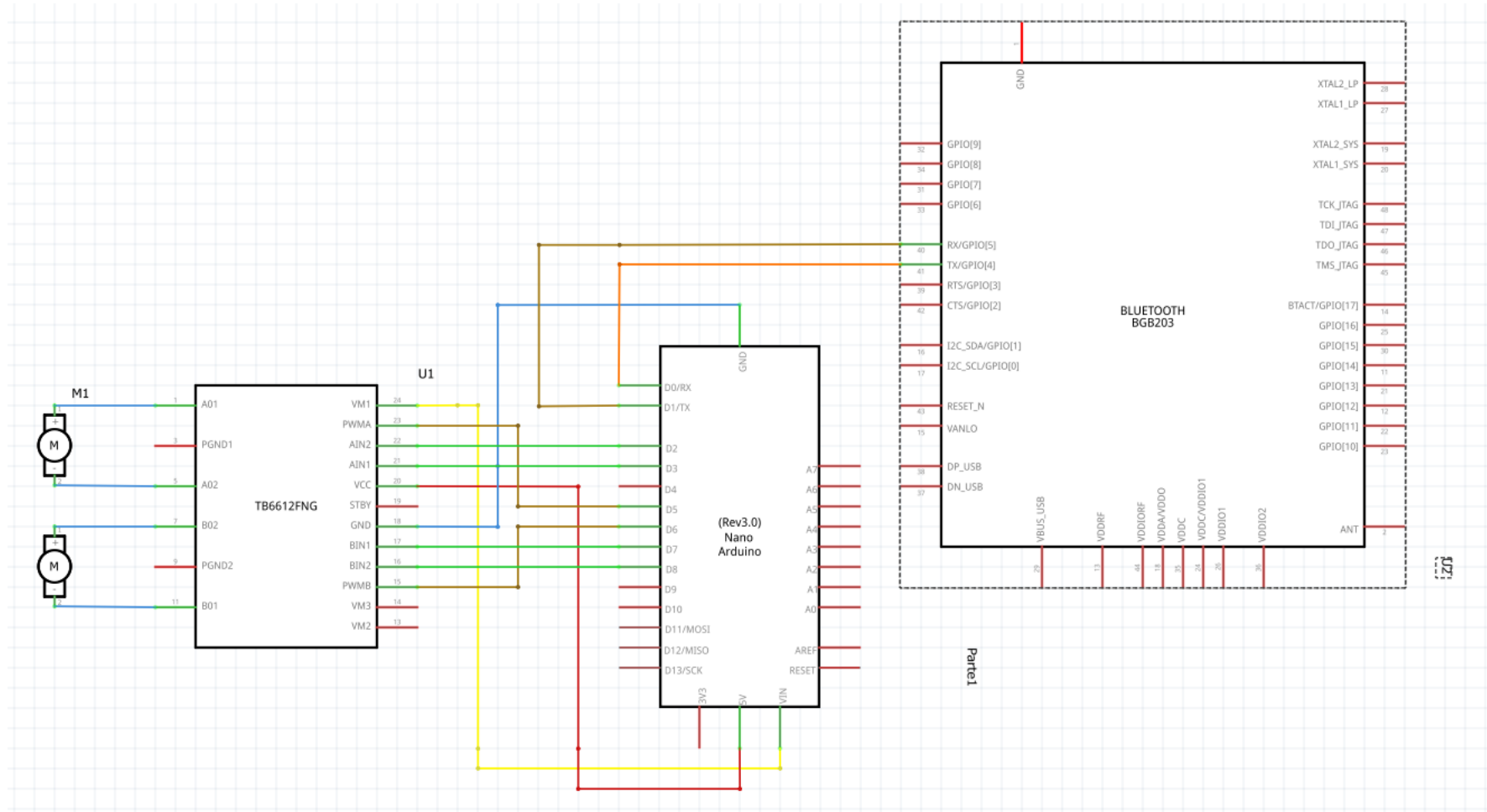


Figura 3. 9: Diseño Esquemático.
Elaborado por: Autor (Programa Fritzing)

3.1.8. Robot de batalla de 1 lb.

Una vez que se describieron los dispositivos a utilizar para el robot de batalla de 1 lb, se procedió a realizar el diseño de la estructura que va tener dicho robot. La estructura debe ser lo menos pesada, ya que el límite de peso es 1 lb. El diseño de la estructura del robot fue realizado en AutoCAD, la figura 3.10 muestra el resultado final del diseño. El chasis donde va montados motores y la tarjeta controladora es de plástico acrílico, mientras, que la carrocería es hecha en aluminio. Estos materiales tienen masas pequeñas, lo que permite llegar al límite del peso en esta categoría.

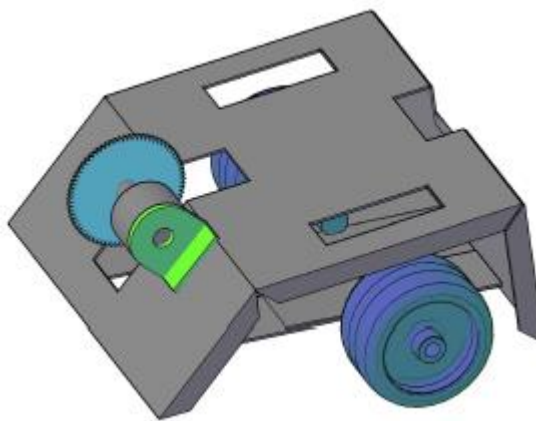


Figura 3. 10: Diseño en AutoCAD del robot de batalla de 1 lb.
Elaborado por: Autor

3.2. Entorno de programación IDE Arduino.

En esta sección se realiza la programación de la tarjeta controladora que permita controlar el robot de batalla de 1 lb usando comunicación Bluetooth mediante un teléfono inteligente. Este programa se llama IDE, que significa "Integrated Development Environment" ("Entorno de Desarrollo Integrado"). La figura 3.11 muestra la ventana principal del software Sketch de Arduino.

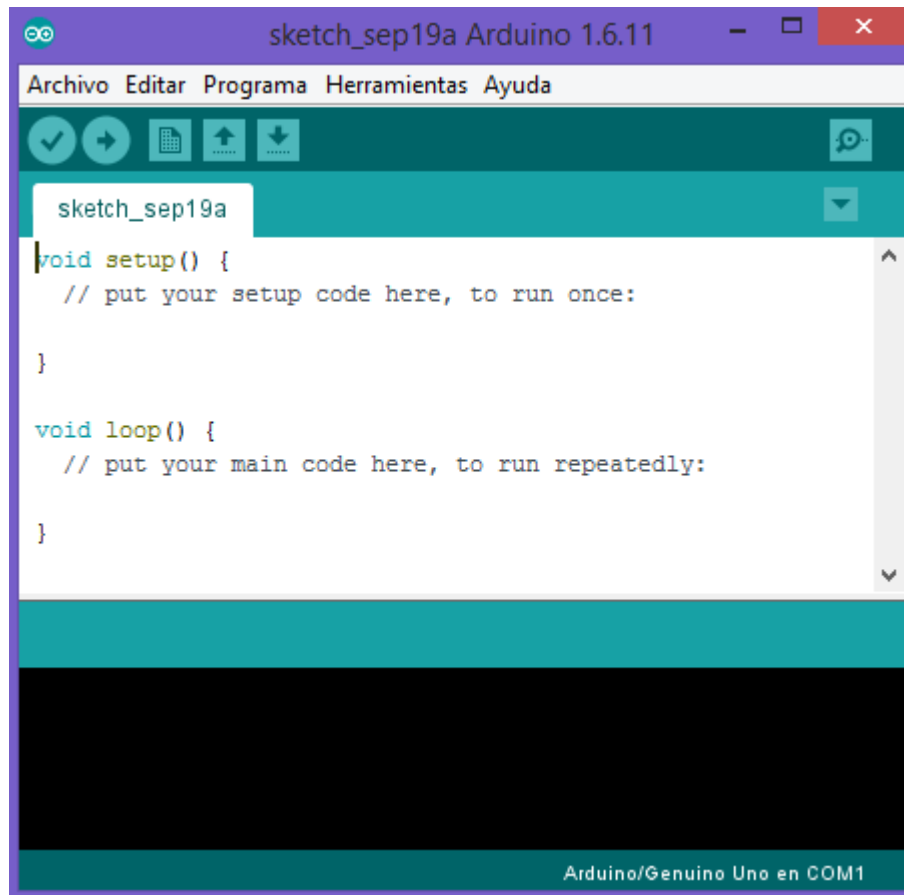


Figura 3. 11: Ide arduino.
Elaborado por: Autor

3.2.1. Codificación en Arduino.

En esta sección se configura y utiliza las clases que se agregarán para controlar el motor brushless, y la comunicación mediante serial.

```
#include <SoftwareSerial.h>
#define DELAY 27
#include <AFMotor.h>
#include <Servo.h>
AF_DCMotor motor1(1);
AF_DCMotor motor2(2);
```

Declaración de variables como velocidad de motores y arma, como el bit que se recibirá al momento de la comunicación.

```

Servo servol;
int vmotores = 120;
int vdisparador = 200;
char dato = 0;
char dataIn = 'S'; //Carácter o datos procedente del teléfono
char determinant; //Se utiliza en la función check, almacena el carácter recibido desde el teléfono.
char det; //Se utiliza en la función loop, almacena el carácter recibido desde el teléfono..
int velocity = 0; //Almacena la velocidad basada en el carácter enviado por el teléfono.

```

Configuración previa para poder usar el serial y el puerto en donde se conectará el arma.

```

void setup() {
  Serial.begin(9600); char det; servol.attach(9);
}

```

La sub-clase llamada Check, que va preguntar si hay dato en el buffer y a la vez realizará una comparación con la sentencia if de que acción a ejecutar y configurar la velocidad.

```

void check() {
  if (Serial.available())
  {
    dataIn = (char)Serial.read();
    if (dataIn == 'F') {adelante();}
    else if (dataIn == 'B') {atras();}
    else if (dataIn == 'L') {izquierda();}
    else if (dataIn == 'R') {derecha();}
    else if (dataIn == 'S') {detener();}
    else if (dataIn == '0') {velocity = 25;vdisparador=25;}
    else if (dataIn == '1') {velocity = 45;vdisparador=45;}
    else if (dataIn == '2') {velocity = 60;vdisparador=60;}
    else if (dataIn == '3') {velocity = 85;vdisparador=85;}
    else if (dataIn == '4') {velocity = 110;vdisparador=110;}
    else if (dataIn == '5') {velocity = 135;vdisparador=135;}
    else if (dataIn == '6') {velocity = 140;vdisparador=140;}
    else if (dataIn == '7') {velocity = 165;vdisparador=165;}
    else if (dataIn == '8') {velocity = 180;vdisparador=180;}
    else if (dataIn == '9') {velocity = 200;vdisparador=200;}
    else if (dataIn == 'q') {velocity = 250;vdisparador=255;}
    else if (dataIn == 'W') {disparadora();}
    else if (dataIn == 'w') {disparadora();}
    else if (dataIn == 'V') {disparadorb();}
    else if (dataIn == 'v') {disparadorb();}
  }
}

```

Acciones las cuales realizara al momento de ir hacia adelante, atrás, derecha, izquierda, detener, disparadora o activación del arma y disparadorb o desactivación de arma.

```
void adelante() {
    motor1.run(FORWARD);    motor1.setSpeed(velocity);
    motor2.run(FORWARD);    motor2.setSpeed(velocity);
}
void atras() {
    motor1.run(BACKWARD);   motor1.setSpeed(velocity);
    motor2.run(BACKWARD);   motor2.setSpeed(velocity);
}
void derecha() {
    motor1.run(FORWARD);    motor1.setSpeed(velocity);
    motor2.run(BACKWARD);   motor2.setSpeed(velocity);
}
void izquierda() {
    motor1.run(BACKWARD);   motor1.setSpeed(velocity);
    motor2.run(FORWARD);    motor2.setSpeed(velocity);
}
void detener() {
    motor1.run(RELEASE);    motor1.setSpeed(velocity);
    motor2.run(RELEASE);    motor2.setSpeed(velocity);
}
void disparadora() {
    servol.write(vdisparador);
}
void disparadorb() {
    servol.write(vdisparador);
}
```

Secuencia o clase principal que se ejecutara al momento de iniciar.

```
void loop()
{
    while (true) {check(); }
}
```

3.3. Entorno de desarrollo para creaciones móviles App Inventor.

Es un entorno el cual permite la creación de aplicaciones con formato apk las cuales sirve para la comunicación con otros periféricos.

3.3.1. Codificación App Inventor.

En la figura 3.12 se muestra el icono para la declaración de variables que permiten la configuración de conexión con el dispositivo Bluetooth. Después, en la figura 3.13 se muestra la búsqueda de dispositivos bluetooth cercanos y la conexión con el mismo.



Figura 3. 12: Diagrama de bloque para comunicación Bluetooth.
Elaborado por: Autor

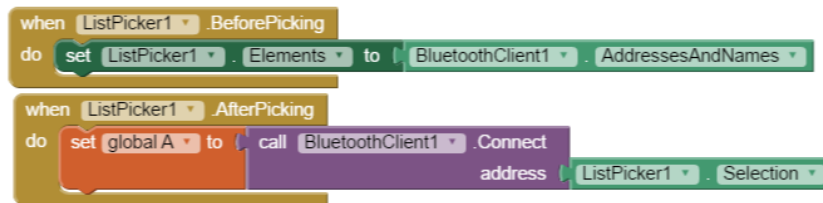


Figura 3. 13: Diagrama de bloque para búsqueda de dispositivos bluetooth.
Elaborado por: Autor

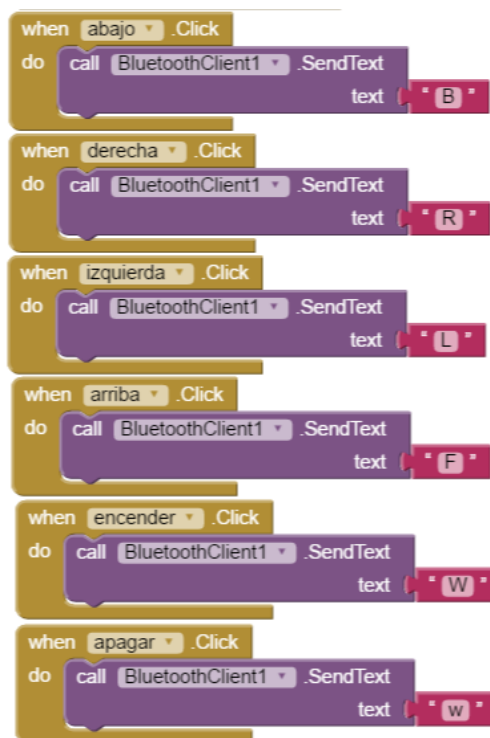


Figura 3. 14: Configuración para definir el estado de posiciones del robot.
Elaborado por: Autor

La figura 3.14 muestra, como se realiza el chequeo del estado de posiciones para él envío de caracteres, tales como, el movimiento y bit para activar el disparo.

Para enviar la velocidad de los motores, se necesita verificar siempre y si existiese un cambio de variables, este permita enviar el dato pertinente por Bluetooth, tal como se muestra en la figura 3.15.

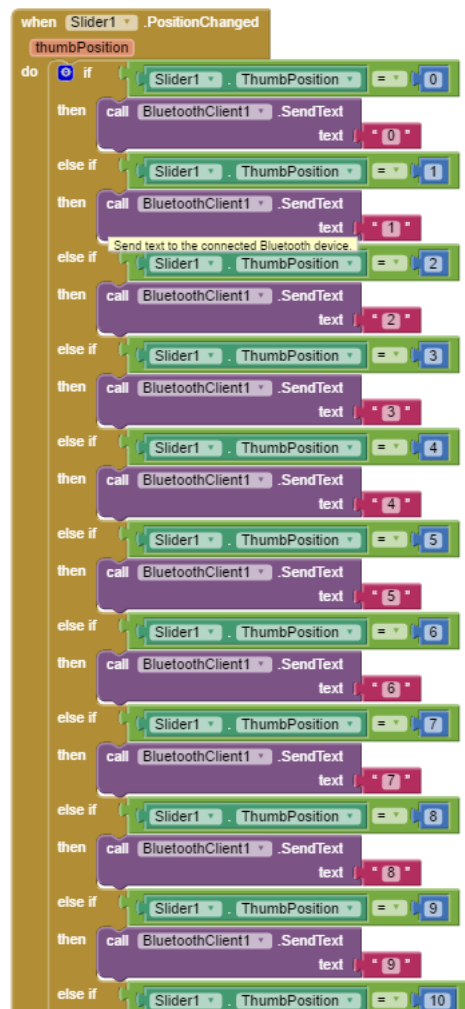


Figura 3. 15: Diagrama de bloque para configurar velocidades del robot.
Elaborado por: Autor

Conclusiones

Referencias Bibliográficas.

Aguayo, P. (10 de Noviembre de 2014). *Introduccion Al Microcontrolador*. Recuperado el 14 de Julio de 2016, de <http://s3.amazonaws.com/academia.edu.documents/39407044/micro.pdf?AWSAccessKeyId=AKIAJ56TQJRTWSMTNPEA&Expires=1472783748&Signature=PfCGZb06Q78mF%2F%2FsZ0hllc6h0KY%3D&response-content-disposition=inline%3B%20filename%3DMicro.pdf>

Ahnn, J. H. (2007). *The Robot control using the wireless communication and the serial communication* (Master). Cornell University. Recuperado a partir de <http://www.cs.cornell.edu/~ja275/nasa/nasa%20final%20report.pdf>

Artero, Ó. T. (2013). *Arduino. Curso práctico de formación*. México: Alfaomega Grupo Editor.

David Wolber, H. A. (2011). *App Inventor*. Abril: O'Reilly Media.

Eero Maasalmi, P. P. (27 de Noviembre de 2011). *publications.theseus.fi*. Recuperado el 23 de Agosto de 2016, de HAAGA-HELIA: https://publications.theseus.fi/bitstream/handle/10024/36104/Maasalmi_Eero_Pitkanen_Panu.pdf

Ortega N., G. M. . (2015). Diseño y construcción de un robot de batalla telemanipulado. Recuperado a partir de <http://www.dspace.espol.edu.ec/handle/123456789/30245>

Pololu. (2016). Pololu - Micro Metal Gearmotors. Recuperado el 19 de agosto de 2016, a partir de <https://www.pololu.com/category/60/micro-metal-garmotors>

Quinde Llerena, J. L., & Ulloa Patiño, L. D. (2012). Diseño y construcción de dos Robot tipo Warbot. Recuperado a partir de

<http://localhost:8080/xmlui/handle/123456789/6849>

Torres G., & Lozano G. (2010). Diseño y Construcción del un Robot de Batalla Controlado Mediante Dispositivo Bluetooth. Recuperado a

partir de <http://dspace.espol.edu.ec/handle/123456789/384>

Yoo C. (2007). *Diseño e implementación de un robot telemanipulado puesto a prueba en el campeonato ecuatoriano de robots 2005*. ESPOL.

Recuperado a partir de

<http://www.dspace.espol.edu.ec/handle/123456789/5312>

Pérez, E. M. (2007). *Microcontroladores PIC: sistema integrado para el autoaprendizaje*. Barcelona: Marcombo.



Presidencia
de la República
del Ecuador



Plan Nacional
de Ciencia, Tecnología,
Innovación y Saberes



SENESCYT
Secretaría Nacional de Educación Superior,
Ciencia, Tecnología e Innovación

<DECLARACIÓN Y AUTORIZACIÓN

Yo, **VERA ARZUBE, LEANDRO ROBERTO** con C.C: # 0928258706 autor del Trabajo de Titulación: **IMPLEMENTACIÓN DE UN ROBOT DE BATALLA DE 1 LB Y DESARROLLO DE UNA APLICACIÓN EN APP INVENTOR PARA CONTROLARLO UTILIZANDO COMUNICACIÓN INALÁMBRICA BLUETOOTH** previo a la obtención del título de **INGENIERO EN TELECOMUNICACIONES** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 09 de Septiembre de 2016

f. _____

Nombre: VERA ARZUBE, LEANDRO ROBERTO

C.C: 0928258706

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TÍTULO Y SUBTÍTULO:	IMPLEMENTACIÓN DE UN ROBOT DE BATALLA DE 1 LB Y DESARROLLO DE UNA APLICACIÓN EN APP INVENTOR PARA CONTROLARLO UTILIZANDO COMUNICACIÓN INALÁMBRICA BLUETOOTH		
AUTOR(ES)	VERA ARZUBE, LEANDRO ROBERTO		
REVISOR(ES)/TUTOR(ES)	M. Sc. EDWIN F. PALACIOS MELÉNDEZ		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Facultad de Educación Técnica para el Desarrollo		
CARRERA:	Ingeniería en Telecomunicaciones		
TÍTULO OBTENIDO:	Ingeniero en Telecomunicaciones		
FECHA DE PUBLICACIÓN:	09 de Septiembre de 2016	No. DE PÁGINAS:	53
ÁREAS TEMÁTICAS:	Microcontroladores, Sistemas de Microprocesadores, Diseño electrónico digital.		
PALABRAS CLAVES/ KEYWORDS:	MICROCONTROLADORES, PIC, ATMEL, BLUETOOTH, ARDUINO, APP INVENTOR		
RESUMEN/ABSTRACT (150-250 palabras):			
<p>Durante el desarrollo del componente teórico, se pensó en desarrollar un robot de batalla de 1 lb, ya que el Club de Robótica "ROBOFET" no disponía de este tipo de robot en la categoría de batalla. Primero se realizó una búsqueda bibliográfica de trabajos relacionados en la implementación de robots de batalla, y que en nuestro país este tipo de concursos de robótica inicia en el año 2005 organizado por ESPOL. El objetivo principal de este trabajo de titulación modalidad examen complejo componente práctico fue diseñar un robot de batalla de 1 lb, que tiene incorporado mecanismos de defensa y ataque ofensivo (arma cortadora). Se realiza el código de programación de IDE Arduino para que la tarjeta controladora reciba información mediante Bluetooth y transmita señales a los motores y el arma del robot. También se creó una aplicación para controlar el robot mediante un celular inteligente, el mismo que fue desarrollado en el software App inventor.</p>			
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR/ES:	Teléfono: +593-4-3849937 +593-9-89846552	E-mail: leandro.vera@hotmail.es	
CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO DE UTE	Nombre: Palacios Meléndez Edwin Fernando		
	Teléfono: +593-9-68366762		
	E-mail: edwin.palacios@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			