UNIDAD DE GESTIÓN DE TECNOLOGÍAS

DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN

CARRERA DE ELECTRÓNICA MENCIÓN

INSTRUMENTACIÓN & AVIÓNICA

TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN & AVIÓNICA

TEMA: "CONTROL DE MOVIMIENTO DE UN ARDUINO ROBOT EMPLEANDO UNA BRÚJULA DIGITAL PARA PRÁCTICAS DE MICROCONTROLADORES"

AUTOR: MORALES CHUGCHILÁN JONATHAN XAVIER

DIRECTOR: ING. PARDO IBARRA JORGE

LATACUNGA

2016

DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN

CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN & AVIÓNICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, "**CONTROL DE MOVIMIENTO DE UN ARDUINO ROBOT EMPLEANDO UNA BRÚJULA DIGITAL PARA PRÁCTICAS DE MICROCONTROLADORES"** realizado por el señor **MORALES CHUGCHILÁN JONATHAN XAVIER,** ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señor **MORALES CHUGCHILÁN JONATHAN XAVIER** para que lo sustente públicamente.

Latacunga, diciembre 2016

ING. PARDO IBARRA JORGE

DIRECTOR

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE TECNOLOGÍAS

DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN

CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN & AVIÓNICA

AUTORÍA DE RESPONSABILIDAD

Yo, **MORALES CHUGCHILÁN JONATHAN XAVIER**, con cédula de identidad **N° 172535905-1** declaro que este trabajo de titulación **"CONTROL DE MOVIMIENTO DE UN ARDUINO ROBOT EMPLEANDO UNA BRÚJULA DIGITAL PARA PRÁCTICAS DE MICROCONTROLADORES"**, ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Latacunga, diciembre 2016

_____

JONATHAN XAVIER MORALES CHUGCHILÁN

CC.1725359051

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE TECNOLOGÍAS

DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN

CARRERA DE ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN & AVIÓNICA

AUTORIZACIÓN

Yo, **MORALES CHUGCHILÁN JONATHAN XAVIER**, autorizo a la Universidad de Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación **"CONTROL DE MOVIMIENTO DE UN ARDUINO ROBOT EMPLEANDO UNA BRÚJULA DIGITAL PARA PRÁCTICAS DE MICROCONTROLADORES"** cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Latacunga, diciembre 2016

_____

JONATHAN XAVIER MORALES CHUGCHILÁN

CC.1725359051

DEDICATORIA

A mi madre María del Rosario.

Por apoyarme en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien; pero más que nada, por su amor.

A mis familiares.

A mi abuelita Evita, a mi hijo Nigel, hermano, tíos, primos y a todos aquellos que participaron directa o indirectamente en la elaboración de esta tesis.

¡Gracias a ustedes!

JONATHAN MORALES

AGRADECIMIENTO

A Dios y A la Virgen Dolorosa por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

A la Institución Unidad de Gestión de Tecnologías-ESPE por darme la oportunidad de estudiar y ser un profesional.

A mi director de tesis, Ing. Jorge Pardo por su esfuerzo y dedicación, quien, con sus conocimientos, su experiencia, su paciencia y su motivación ha logrado en mí que pueda terminar mis estudios con éxito.

También me gustaría agradecer a mis profesores durante toda mi carrera profesional porque todos han aportado con un granito de arena a mi formación.

JONATHAN MORALES

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El presente trabajo de titulación, tuvo como objetivo general la implementación de un sistema de control de movimiento de un Arduino Robot desde un Smartphone con sistema Android, empleando una brújula digital para realizar prácticas en la asignatura de microcontroladores. El sistema está fundamentado en el diseño de una aplicación Android que mediante una interfaz gráfica permite al usuario ingresar los movimientos a ser ejecutados, mediante tres tiempos y tres coordenadas geográficas diferentes; el proceso de control de movimiento consiste en que el Smartphone envía vía Bluetooth los datos ingresados en la aplicación a un Arduino Nano que se encarga de validar y enviar los datos vía protocolo de comunicación I2C al Arduino Robot. Al mismo tiempo, éste procesa la información y empleando una brújula digital ejecuta los tiempos y movimientos ingresados por el usuario en la aplicación. En la parte experimental del presente proyecto se ha llevado a cabo un análisis de resultados de cada una de las etapas del sistema, de modo que se pueda evaluar con puntos favorables y no favorables. Este tipo de proyectos, consolida los contenidos teóricos y prácticos que orientan el estudio de sensores y accesorios del Arduino Robot, beneficiando a estudiantes de 5to y 6to nivel de la carrera de Electrónica mención e Instrumentación y Aviónica.

PALABRAS CLAVE:

- ARDUINO ROBOT
- BLUETOOTH
- I2C
- APLICAIONES ANDROID
- BRÚJULA DIGITAL

ABSTRACT

The following academic project had as general objective the implementation of a motion control system of an Arduino Robot from a Smartphone with Android system using a digital compass in order to do a practical work in the subject of microcontrollers. The system is based on the design of an Android application that allows the user to enter the movements to be executed in three times and three different geographical coordinates; the smartphone sends via Bluetooth the entered data in the application to an Arduino Nano that is responsible for sending data via I2C communication protocol to the Arduino Robot. At the same time, it processes the information and executes the times and movements entered by the user in the application using a digital compass. For the experimental part, an analysis of the results of each of the stages of the system was carried out, so that the system can be evaluated with favorable and unfavorable points. This type of project consolidates the theoretical and practical contents that guide the study of sensors and accessories of an Arduino Robot benefiting students of 5th and 6th level of the career of Electronics mention and Instrumentation and Avionics.

KEY WORDS:

- ARDUINO ROBOT
- BLUETOOTH
- I2C
- ANDROID APPLICATIONS
- DIGITAL COMPASS

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1. Antecedentes

Actualmente el campo de la electrónica evoluciona a grandes pasos por lo cual el Arduino Robot es una plataforma física moderna que servirá de gran ayuda para todos los estudiantes en las prácticas e innovaciones en la asignatura de Microcontroladores de la Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas U.G.T. – ESPE. Después de navegar por los distintos repositorios de las diferentes instituciones educativas estatales e internacionales, que ofertan en el medio las tecnologías e ingenierías afines al presente tema de investigación, se evidenció que existen grandes investigaciones relacionadas con el tema. A continuación, se presenta las siguientes investigaciones:

El ingeniero Quezada en el año 2014, diseñó y construyó un robot todo terreno utilizando el sistema de suspensión rocker- bogie y teleoperado inalámbricamente con la programación de una placa Arduino la cual es el cerebro del robot. El mismo que está basado en tres sistemas principales: sistema mecánico, sistema electrónico y sistema de control que ayudan a la movilidad y control del robot en terrenos.

El ingeniero Sánchez en el año 2014, diseñó y construyó un robot para inspección visual de tuberías operado remotamente para la empresa FSB, el cual está diseñado mecánicamente para ingresar dentro de tuberías teniendo un avance de 36 metros dentro de la misma para la búsqueda y detección de errores. Los datos como fotos, videos e información se almacenan dentro de una memoria MicroSD, El robot cuenta con sensores los cuales se encargan del monitoreo de temperatura, humedad, presencia de gases inflamables e informar la distancia recorrida por el robot en el interior de la tubería. El sistema control y programación está basado en tecnología Arduino y la interfaz maquina usruario se programó enteramente en el software

“Labview” utilizando el “Arduino Toolkit” para “Labview”. Universidad de las Fuerzas Armadas.

Los ingenieros Cabrera y Delgado en el año 2014, diseñaron y construyeron un robot para mapeo y exploración de minas subterráneas mediante un prototipo capaz de movilizarse dentro de un túnel, el cual mediante los sensores colocados dentro del mismo realice un mapa que describa los caminos extensión y variables ambientales dentro de la mina. Universidad de la Azuay.

Los trabajos anteriormente mencionados y realizados por los señores ingenieros de las diferentes entidades educativas del país fueron realizados de manera exitosa en su totalidad, debido que, los proyectos de investigación relacionados con robots obtuvieron resultados positivos, constituyendo una guía y un referente para el desarrollo del presente trabajo de titulación.

1.2. Planteamiento del problema de investigación

En la Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas – ESPE, en la asignatura de microcontroladores existe la necesidad de implementar una placa de hardware libre completa y moderna, como es Arduino Robot, para que los estudiantes tengan conocimientos y desarrollen habilidades sobre el manejo de este dispositivo y puedan utilizarlo en diferentes aplicaciones como, por ejemplo, la automatización del Arduino Robot.

Esta es una placa de hardware libre, la misma que dentro de todos sus sensores incorpora una brújula digital y dos procesadores uno en cada una de sus dos tableros, donde la junta de motor controla los dos motores y, la otra junta lee la información de los sensores y según su programación decide como operar, siendo así el primer Arduino oficial en ruedas.

La implementación de dicha placa permitirá a los estudiantes poner en práctica la teoría, la programación y el uso de los diversos componentes y sensores que ésta placa ofrece; además, porque brinda oportunidades para desenvolverse de forma

satisfactoria en el ámbito laboral de la robótica. Una consecuencia futura, se puede reflejar en la actualización y el uso de estas nuevas plataformas con las cuales se trabaja actualmente en el mundo de la robótica. Adquisición, que será de gran ayuda para estudiantes y docentes.

1.3. Justificación

La implementación del sistema de control de movimiento de un Arduino Robot desde un Smartphone con sistema Android, empleando una brújula digital para realizar prácticas en la asignatura de microcontroladores de la UGT– ESPE, es de suma importancia porque beneficiará a los estudiantes en el uso y el estudio del funcionamiento de dispositivos electrónicos innovadores como una brújula digital y la interacción práctica del Arduino Robot, de la misma manera, facilitará el estudio de sensores y accesorios que esta moderna placa ofrece.

Es factible la realización de este proyecto ya que se cuenta con la información necesaria, su implementación no generará un alto costo económico, ni demandará de un largo tiempo para su diseño e implantación. Por tal razón, es necesario implementar un control de movimiento usando una placa Arduino Robot con una brújula digital para prácticas en la asignatura de microcontroladores de la UGT - ESPE.

1.4. Objetivos

1.4.1. Objetivo general

Implementar un sistema de control de movimiento de un Arduino Robot desde un Smartphone con sistema Android, empleando una brújula digital para realizar

prácticas en la asignatura de microcontroladores de la Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas – ESPE.

1.4.2. Objetivos específicos

- Indagar las características de un Arduino Robot que permita el estudio de una brújula digital para el control de sus movimientos
- Desarrollar una aplicación Android que permita controlar los movimientos de un Arduino Robot, mediante un módulo Bluetooth para la comunicación inalámbrica entre el Arduino Robot y un Smartphone con sistema Android.
- Realizar pruebas del sistema de control de movimiento para verificar el funcionamiento del Arduino Robot
- Desarrollar una guía de práctica de laboratorio utilizando el Arduino robot y la brújula digital para la signatura de microcontroladores

1.5. Alcance

Este proyecto va dirigido a los estudiantes de 5to y 6to nivel de la carrera de Electrónica mención e Instrumentación y Aviónica de la UGT-ESPE, para la realización de prácticas con el control de movimiento de un Arduino robot, empleando una brújula digital en la asignatura de microcontroladores y para futuras prácticas de laboratorio utilizando el Arduino Robot y la brújula digital.

CAPÍTULO II

MARCO TEÓRICO

2.1. Arduino

Arduino es una plataforma de prototipos de código abierto basado en hardware y software fácil de usar. Las Tarjetas de Arduino son capaces de leer las entradas de luz de un sensor, una pulsación en un botón, pulsador o interruptor, o una entrada analógica de datos y lo convierten en una salida, activando de un motor, encendiendo un LED o mostrando información en pantalla. Se puede programar al Arduino para que envíe un set de instrucciones para el microcontrolador Atmega interno en la placa; para ello se utiliza el lenguaje de programación de Arduino (basado en Lenguaje C) a través de la plataforma de software de Arduino (IDE). Arduino ha sido el cerebro de miles de proyectos, desde objetos cotidianos a instrumentos científicos complejos.

Figura 1 Productos oficiales de Arduino

Fuente: (Arduino, 2015)

2.1.1. Características generales de las placas Arduino:

Las principales características generales de las placas Arduino son:

- Microprocesador ATmega328
- 1 kbyte memoria RAM
- Pines de entrada: analógicas y digitales
- Pines de salidas: analógicas y digitales.
- Voltaje de funcionamiento 5V
- Voltaje límite de funcionamiento 6-20 V
- DC corriente I/O Pin 40 mA
- DC corriente 3.3V Pin 50 mA
- Memoria Flash 32 KB (2 KB para el bootloader)
- EEPROM 512 byte
- Velocidad de reloj 16 MHz
- Fasil manejo
- Menos consumo de corriente

2.1.2. Ventajas:

Entre las ventajas del Arduino robot tenemos las siguientes:

- Arduino es una plataforma múltiple, eso quiere decir que su software es compactible y funciona con todos los sistemas operativos como son: Windows, Macintosh OSX y Linux
- Usa un medio de programación que es simple y directo
- A su software se lo puede ampliar ya que es de código abierto
- Es fundamentado en microcontroladores ATMEGA, lo único que cambia son sus características

2.2. Robot Arduino

El Arduino robot es la primera placa oficial que viene ensamblada con: dos ruedas, un kit de sensores, pantalla LCD, etc., es por eso que no requiere de la incorporación de algún sensor, chip adicional para su comunicación y funcionamiento, al menos que su estudio o proyecto lo requieran. El robot incorpora dos placas cada una con su propio microcontrolador, la primera es una placa motora la que se encarga de controlar el movimiento de los motores y la segunda placa de control que es la que lee los sensores y decide como operar.

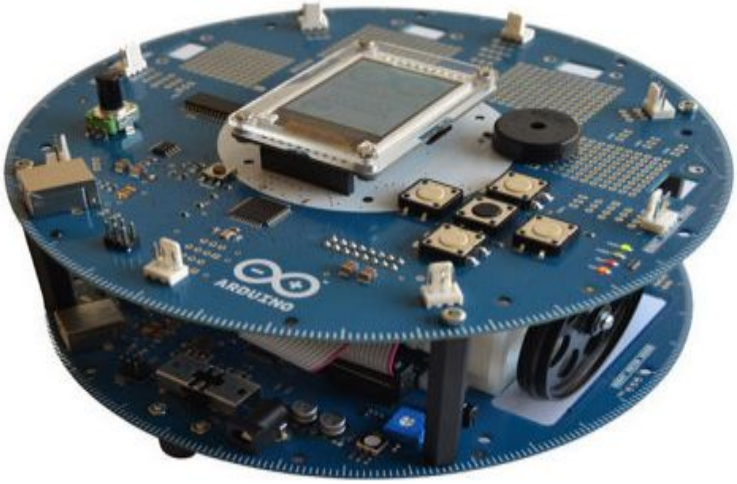

Figura 2 Arduino Robot

Fuente: (Robot, 2016)

Cada placa es autónoma, eso quiere decir que su programación se la realiza por separado, mismas que deben ser programadas con el IDE de Arduino, las dos placas llevan un microcontrolador ATmega32u4. La programación del Arduino robot es similar a la del Arduino Leonardo ya que estas dos placas incorporan comunicación USB, así se elimina a un segundo procesador secundario para la comunicación con el computador, esto permite que la placa aparezca como virtual (CDC) de puerto serie/COM.

2.2.1. Placa de control

El Arduino Robot es “el primer Arduino oficial sobre ruedas. El robot tiene dos procesadores, uno en cada uno de sus dos tableros. El Motor Board controla los motores, y la Junta de Control lee los sensores y decide cómo operar” (Robot, 2016, pág. 1)

La placa de control del Arduino robot trae con ella incorporado un microcontrolador ATmega32u, básicamente el funcionamiento de la placa es recibir las señales de los sensores y decidir cómo actuar dependiendo de la programación diseñada por el estudiante o docente. La placa, trae incorporada un sin número de pines incrustados en un chasis circular, mismos que fueron asignados para el uso de sensores o actuadores.

Según, la práctica que se vaya a realiza o el trabajo que el robot vaya a ejecutar todas sus líneas son multiplexadas y conectadas directamente al microcontrolador. La placa de control cuenta con los siguientes accesorios para sus principales ejercicios: sensores IR, brújula digital, altavoz, botones, potenciómetros, lector de tarjetas SD, tarjeta SD 2gb, display TFT, entras y salidas analógicas y digitales.

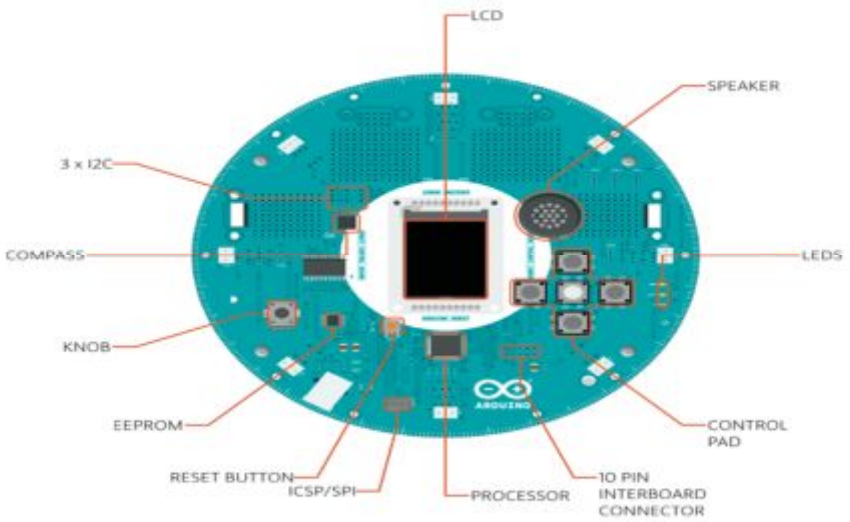

Figura 3 Placa de Control Arduino Robot
Fuente: (Robot, 2016)

2.2.2. Características técnicas placa de control

Las principales características de la placa de control son las siguientes:

- Microcontrolador: ATMega32u4 32 bits
- Voltaje de operación 5V
- Tinkerkit IN/OUT digitales: 5
- Canales PWM: 6
- Entradas Tinkerkit analógicas: 4 (de 5 pines IN/OUT)
- Entradas Tinkerkit analógicas Multiplexadas: 8
- Intensidad CC por IN o OUT: 40 mA
- Memoria Flash: 32 KB; SRAM: 2.5 KB y EEPROM: (en ATMega32u4)
- EEPROM (externa): 512 Kbit (I2C)
- Reloj 16 MHz
- 5 teclas
- LCD a color con comunicación SPI
- Lector tarjetas SD para tarjetas con formato FAT16
- Altavoz: 8 Ohm
- Brújula digital proporciona los grados con sentido al norte geográfico
- Puertos I2C soldables: 3

2.2.3. Placa de motores

La segunda placa del Arduino robot es una placa motora, la cual consta de su propio microcontrolador que es un ATmega32u4, puerto I2C, sensores de piso IR, cuatro pilas AA recargables, puerto de carga, dos motores, dos llantas, entre algunas entradas, salidas analógicas y digitales para futuros proyectos y estudios relacionados con el Arduino Robot.

Esta placa motora es la encargada de controlar la potencia y el movimiento de los motores, misma que es encargada de mover las llantas logrando que el robot tome dirección en diferentes sentidos. Además, es autónoma la cual puede ser programada

con el IDE de Arduino, el fabricante recomienda el uso de la misma siempre y cuando se tenga conocimientos en la materia, ya que esta placa fue diseñada para realizar proyectos complejos.

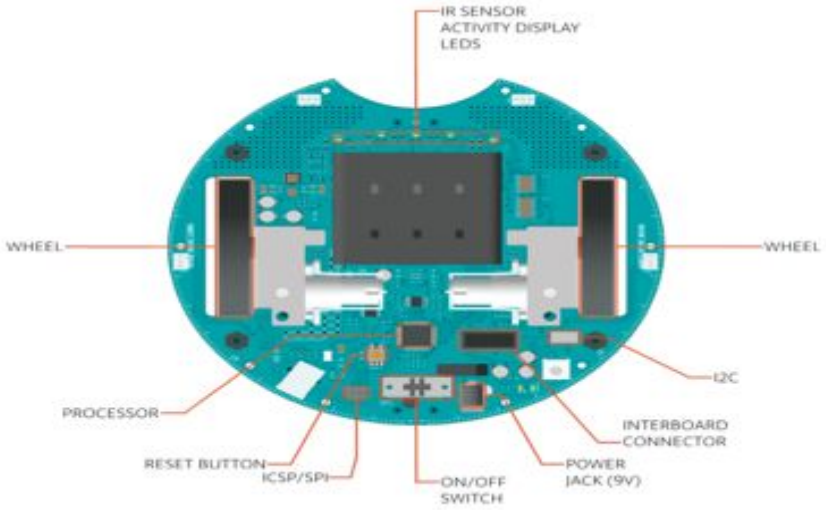

Figura 4 Placa superior de Motores Arduino Robot

Fuente: (Robot, 2016)

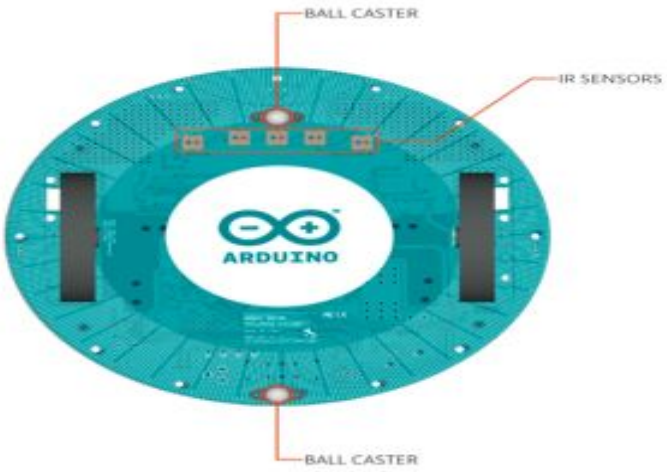

Figura 5 Placa inferior de Motores Arduino Robot

Fuente: (Robot, 2016)

2.2.4. Características técnicas placa de motores

A continuación, se detalla las características de la placa de motores:

- UN microcontrolador ATMega32u4 32 bits
- Tensión de operación: 5V
- Tensión de Entrada: 9V del cargador de batería
- Porta pilas para 4 pilas AA alkalinas o recargables de Ni-Mh
- Puertos Tinkerkit IN y OUT digitales: 4
- Entradas Tinkerkit sensores analógicos: 4
- Canales PWM (Modulación por pulsos): 1
- Puerto convertidor CC de 5V para alimentar el robot
- Memoria Flash: 32 KB
- SRAM: 2.5 KB
- EEPROM: 1 KB en ATMega32u4
- Reloj: 16 MHz
- Trimmer para calibrar el movimiento
- Sensores IR seguidores de líneas: 5
- Puertos I2C soldables: 1

2.2.5. Motores DC

Su funcionamiento se logra al poner a circular una corriente por uno de los conductores que está fijado en un campo magnético, mismos que son sometidos a una fuerza mecánica que se la conoce como electromotriz, que es usada como el principio básico de funcionamiento de un motor eléctrico. Por ejemplo, si se pone a circular una corriente por un conducto situado en entre dos imanes, se crea una fuerza mecánica que se opondrá a los cambios de atracción de los imanes, generando un movimiento entre los polos de cada imán.

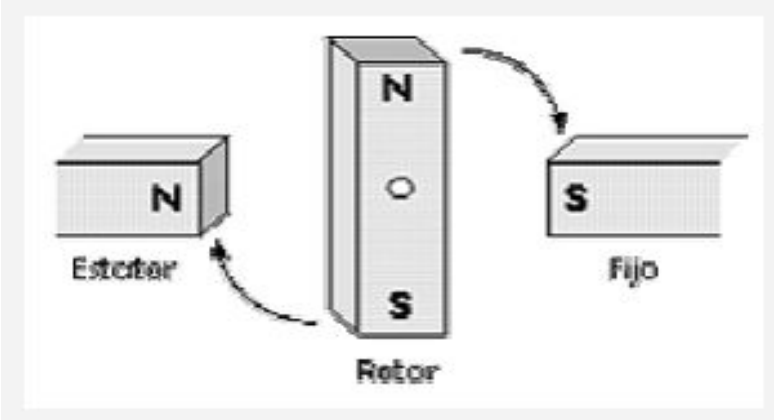

Figura 6 Movimiento de un motor DC

Fuente: (Arduino O. , 2015)

En resumen, se podría decir que la fuerza que surja será directamente proporcional a la intensidad de todo el campo magnético y también al número de hilos de cobre que serán recorridos por dicha corriente, es por eso que los motores llevan hilos de cobre alrededor del núcleo, todo esto dependerá de la fuerza que vayamos a dar el motor. A diferencia de los motores DC mucho más grandes y con más fuerza de tracción todo dependerá del su uso y su aplicación.

2.2.6. Alimentación

La alimentación del Arduino robot puede ser mediante la conexión USB directamente de un computador o por cuatro pilas AA, el Arduino robot selecciona automáticamente la fuente de alimentación, todo esto por razones de seguridad. El robot trae incorporado un cargador de baterías que puede ser conectado a una fuente eléctrica de 120/220 AC, el mismo que proporcionara de 9v dc para cargar las cuatro baterías AA, por temas de seguridad el cargador no funcionara si el robot se encuentra conectado al cable usb.

2.2.7. Memoria

El estudio de la página oficial de Arduino robot de 2016 presenta lo siguiente:

> El ATmega32u4 tiene 32 KB (con 4 KB utilizado para el gestor de arranque). También cuenta con 2,5 KB de SRAM y 1 KB de EEPROM (que se pueden leer y escribir con la librería EEPROM). La tarjeta de control tiene una EEPROM extra de 512 kbit que se puede acceder a través de I2C. Hay un lector de tarjetas SD externo conectado a la pantalla GTFT a la que puede acceder el procesador de la tarjeta de control para un almacenamiento adicional. (p. 4)

En relación a lo expuesto anteriormente, el Arduino robot cuenta con un consejo regulador al cual se puede acceder mediante el puerto I2C y puede ser usado como almacenamiento adicional. Además, el arduino robot cuenta con un lector de tarjetas SD externo conectado a la pantalla GTFT y puede ser utilizado para el gestor de arranque.

2.2.8. Entrada y salida

El Arduino robot trae algunos conectores pre soldado, para que puedan ser usados como puntos adicionales para la incorporación de sensores o algún chip necesario para algún proyecto futuro. Todos los conectores tanto en la tarjeta de control como en la tarjeta motora están señalados y se puede asignan algún puerto mediante la biblioteca del robot, misma que permite el acceso a las funciones estándar de Arduino, tomando en cuenta que cada pin puede proporcionar o recibir un máximo de 40 mA a 5V.

A continuación, de detalla los pines con funciones especializadas:

- TK0 a TK7: Estos pines viene multiplexados a un solo pin analógico en el microcontrolador de la placa de control, estos pueden ser usados como como entradas analógicas para sensores, por ejemplo: sensores de distancia, ultrasónicos, analógicos.

- TKD0 a TKD5: Estos son pines digitales y vienen conectados directamente al procesador, estos también pueden ser usados como entradas analógicas.
- TK1 a TK4: A estos pines se los nombra en el software como B_TK1 a B_TK4 y pueden ser pines de entrada analógica o digital.

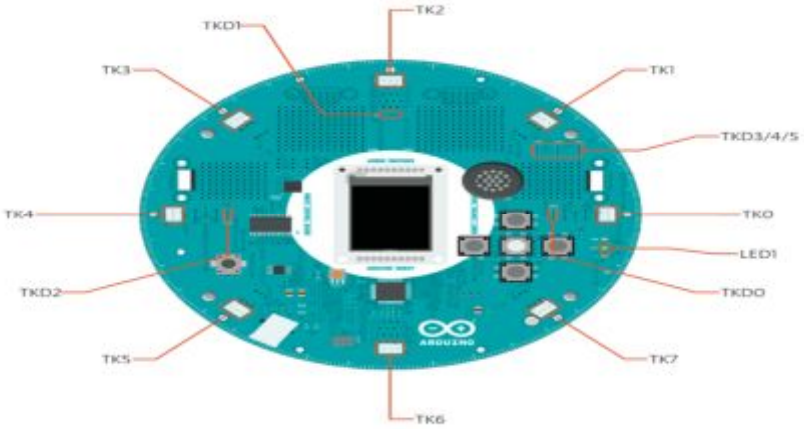

Figura 7 Entradas y Salidas de la tarjeta control

Fuente: (Robot, 2016)

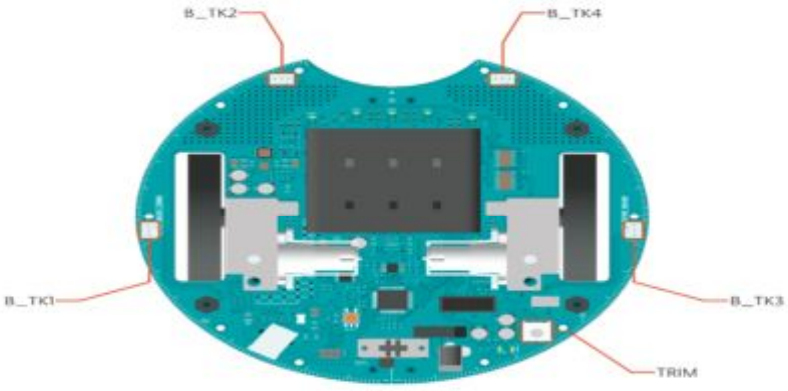

Figura 8 Entradas y Salidas de la tarjeta motor

Fuente: (Robot, 2016)

2.2.9. Comunicación

Para la comunicación el Arduino robot cuenta con un sin número de instalaciones o puertos para la comunicación con un ordenador o con otro Arduino y otros microcontroladores. El microcontrolador ATmega32U4 ofrece UART TTL (5V) de comunicación en serie, misma que está disponible en un formato digital que se puede usar entre el conector 10-pin- entre placa y placa, entre ellos los siguientes:

- **Comunicación en serie:** Las placas del Arduino robot se comunican entre sí mediante el puerto serie de los transformadores que están conectados una con la otra. Se conectan gracias al uso de un conector de 10 pines llevado acabo la comunicación en serie, así también como la alimentación y toda la información adicional como la batería restante del robot.
- **Junta de control de LEDs:** La placa de control tiene tres LEDs situadas en su chasis. El primer led indica que el robot está encendido (PWR) y los dos leds restantes indican la comunicación mediante del puerto USB (LED1 / RX y TX). Se puede acceder al led número 1, mediante el software Arduino.
- **Comunicación I2C**: Las dos placas del Arduino robot incorporan conectores I2C, la placa de control tiene tres puertos y la placa motora tiene solamente un puerto I2C.

El microcontrolador 32U4 también permite comunicación la serie (CDC) a través de los puertos USB que cada una de las placas trae por separado, el mismo que aparece como un puerto COM virtual para el software del computador, el chip suele actuar como un dispositivo de máxima velocidad USB 2.0 usando el controlador USB COM estándar, el mismo que viene con su propio cable de alimentación.

Los leds RX y TX de la placa de control se mantienen parpadeando cuando se está transmitiendo datos a través de la conexión USB con el computador, cosa que no pasa en la comunicación serial entre las dos tablas. Cada una de las placas cuenta con un identificador independiente de puerto USB, que se mostrará como puertos diferentes en el IDE. Es por eso que hay que asegúrese de elegir la placa más adecuada para nuestra programación y proyecto.

2.2.10. Programación

Al Arduino robot se lo puede programar con el software de Arduino, nada más hay que seleccionar la placa con la cual se vaya a trabajar ya sea la junta de Control o la junta Motor en el menú de Herramientas del IDE. Cada procesador ATmega32U4 en cada una de las placas del Arduino Robot vienen precargados con un gestor de arranque, el mismo que permite cargar una nueva programación sin presionar el botón de reset externo, Se comunica utilizando el protocolo AVR109. También, se puede ignorar el gestor de arranque y proceder con la programación en el microcontrolador a través del ICSP (In-Circuit Serial Programming).

2.2.11. Reset Automático (software)

El Arduino robot cuenta con un reset automático vía software, en vez de presionar el botón reset antes de que se vaya a cargar una programación nueva, el robot cuenta con una función que permite restablecer la memoria mediante software con tan solo estar conectada al computador, esto es posible cuando se activa el virtual (CDC) del puerto serie / COM del robot, ya que este se abre en 1200 baudios y luego se cierra.

Cuando esto suceda, automáticamente el procesador se reiniciará rompiendo la conexión USB con el ordenador, esto significa que el puerto serie / COM virtual desaparecerá. Posterior a esto el procesador se restablezca, el gestor de arranque se iniciará y permanecerá activo alrededor de 8 segundos. El gestor de arranque también puede ser iniciado presionando dos veces el botón de reinicio en la placa de control del Arduino Robot.

2.2.12. Protección de sobre corriente USB

Para la protección contra alta corriente está incorporado en las dos placas del robot tienen un polyfuse que es reajustable mediante software que protege a los puertos USB de nuestro computador de cortos circuitos y sobre corrientes, tomando en cuenta que cada computar trae sus propias protecciones. El fusible logra proporcionar una protección adicional en el caso que se aplique más de 500 mA al puerto USB, básicamente el fusible corta automáticamente la corriente y la conexión hasta que el cortocircuito desaparezca o su vez este sea eliminado.

2.3. Microcontrolador atmega32u4

El ATmega32U4 es “un microcontrolador CMOS de 8 bits de bajo consumo de potencia basado en la arquitectura AVR RISC mejorada. Ejecuta potentes instrucciones en un solo ciclo de reloj, llegando a un desempeño cercano a 1 MIPS por MHz” (Datasheet, 2014, pág. 1)

Es decir, el ATmega es un microcontrolador con un circuito integrado, en cuyo interior posee toda la arquitectura de un computador, esto es CPU, memorias RAM, EEPROM, y circuitos para la interfaz análoga / digital de entrada y salida. El microcontrolador Atmel ATmega32U4 es un dispositivo moderno que es muy usando por los estudiantes en proyectos, este es un mega AVR de 8 bits basado en la arquitectura RISC y mejorada de los AVR., realizando instrucciones en un solo ciclo de reloj.

Como ya es habitual os fabricantes en este tipo de microcontrolador ATmega32U4 incorpora en él un puerto USB 2.0, esta cualidad lo convierte en el microcontrolador más usado y recomendado por el docente y estudiantes para las diferentes aplicaciones que necesitan conectividad vía puerto USB. Además, este micro controlar integra una interfaz JTAG para lograr depuraciones en el chip, convirtiéndolo en el microcontrolador más usado en su especie.

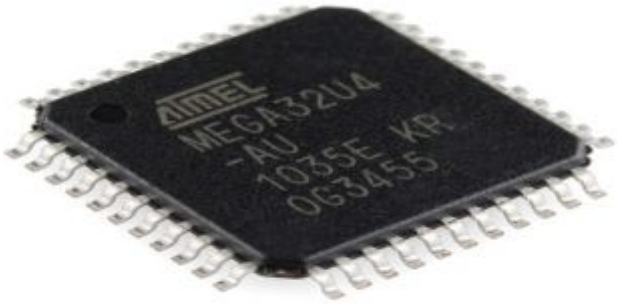

Figura 9 Microcontrolador ATmega32U4

Fuente: (Datasheet, 2014)

2.3.1. Características del microcontrolador:

A continuación, se detalla características del microcontrolador:

- Funcionamiento de frecuencia máxima de 16 MHz
- Flash de 32 KB
- EEPROM de 1024 B
- SRAM de 2,5 KB
- 135 instrucciones: la mayoría de ejecución en un solo ciclo de reloj
- 32 registros de uso general
- Reset de encendido
- Oscilador interno 8 MHz
- Fusibles de interrupción internos y externos
- Seis modos automáticos de hibernación para ahorro de energía
- Interrupciones internas y externas

2.4. Brújula digital HMC6352

Antes de hablar de la brújula digital, hablaremos de ostros sistemas que funcionan mediante orientación, con el norte magnético o coordenadas geográficas, El sistema GPS (Sistema de Posicionamiento Global), el cual entrega datos de alguna ubicación en cualquier lugar del mundo con un pequeño margen de error, el mismo sistema puede entregar información sobre velocidad, altura, coordenadas geográficas.

Como un punto en contra podemos decir que es un sistema más complejo en su funcionamiento y por ende necesita de más líneas de programación, esto genera más espacio en la memoria del microcontrolador tomando en cuenta que la información enviada por el sistema solo aportaría un dato adicional a lo requerido. Por lo que, es muy factible el uso de una brújula digital si la información requerida es en grados con sentido al norte magnético.

La Honeywell HMC6352 es “una brújula totalmente integrada Módulo que combina sensores magneto-resistivos de 2 ejes con los circuitos de soporte analógico y digital requeridos, y Algoritmos para el cálculo de encabezamientos” (HMC6352., 2015, pág. 1)

Figura 10 Brújula digital HMC6352

Fuente: (HMC6352., 2015)

Los datos censados que envía el sensor de la brújula digital HMC6352, son datos en valores de décimas de grados que encuentran entre un margen de 0 a 3599, los mismos que se encuentran previstos en el formato de los dos bytes. Se realiza la siguiente operación matemática y de esta manera se lograr mostrar la impresión del ángulo censado por la brújula digital.

$$Grados = \frac{Valor_{sensor}}{100}$$

Figura 11 Ecuación Matemática

Fuente: (HMC6352., 2015)

2.4.1. Descripción de los pines

Tabla1
Descripción de pines HMC6352

PIN	NOMBRE	DESCRIPCION	TIPO DE PIN
1	SDA	Datos Serie I2C	Entrada/Salida
2	SDA	Datos Serie I2C	Entrada/Salida
3	GND	Tierra	Potencia
4	SCL	Reloj I2C	Salida
5	NC	Sin conectar	No Aplica
6	VDD	Voltage de Entrada	Potencia

Fuente: (HMC6352., 2015)

2.4.2. Modos de funcionamiento

En realidad, este sensor es un magnetómetro, su función es medir la fuerza de los campos magnéticos, también es conocido como compás o brújula digital. Generalmente los magnetómetros permiten censar o saber la dirección hacia de donde está situado el Norte magnético más fuerte. Estos suelen ser usados para obtener una orientación basada en el Norte magnético terrestre y es muy usada en robots, proyectos, etc.

Para lograr la navegación autónoma de un artefacto o para censar y devolver los datos de orientación a distancia, se podría decir que este censor es uno de los más sencillos que se encuentra en el mercado, tomando en cuenta que únicamente puede detectar un campo magnético en un eje. Pese a eso nos bridara una alta precisión y cuenta con una interfaz I2C, lo que facilita el envió de datos a un Arduino únicamente con dos pines.

2.4.3. Características del HMC6352

A continuación, se detalla las características del HMC6352:

- Interfaz I2C
- 2 pines para obtener los datos
- Alimentación 2.7, 5.2V
- Resolución 0.5 grados
- Consumo: 1mA (3V)
- Dimensiones: 15x15mm
- Permite ser calibrado
- Mínimo margen de error
- Fácil uso
- Compartibles con librerías arduino

2.4.4. Ventajas:

Se puede mencionar las siguientes ventajas:

- Algoritmo de calibración interna
- Funciona como esclavo al procesador maestro de cliente (100 kHz)
- El sensor puede ser usado en entornos de fuerte campo magnético

2.5. Arduino Nano

El Arduino Nano es “un tablero pequeño, completo y basado en la placa ATmega328 (Arduino Nano 3.x) o ATmega168 (Arduino Nano 2.x). Tiene más o menos la misma funcionalidad del Arduino Duemilanove, pero en un paquete diferente” (Arduino N. , 2015, pág. 1)

La placa Arduino Nano es una pequeña o mini placa completa que incorpora en ella un ATmega328, tiene un funcionamiento similar a los demás Arduino, pero este a diferencia de los demás está basado en un módulo DIP. En sí, este Arduino es usado para proyectos muy pequeños o que no requieran de mayores prestaciones. Una de las pequeñas desventajas podría ser que no tiene un puerto de alimentación DC y que para su programación funciona con un cable Mini-B USB a comparación de los demás que usan uno estándar.

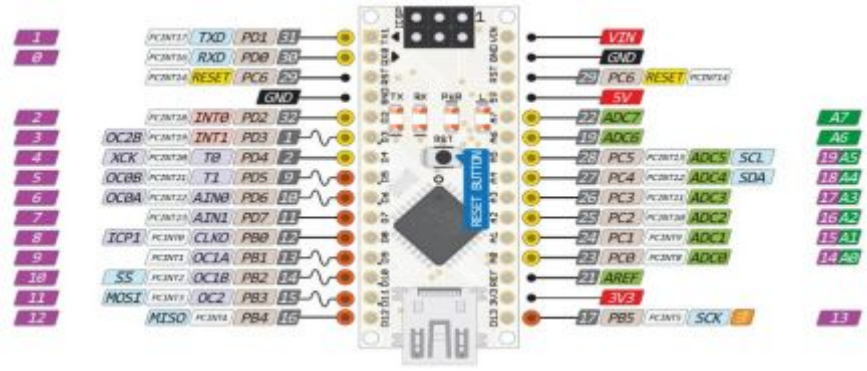

Figura 12 Numeración de puertos Arduino Nano

Fuente: (Arduino N. , 2015)

2.5.1. Especificaciones:

Las especificaciones del Arduino Nano son las siguientes:

- Microcontrolador ATmega328
- Tensión de entrada recomendada: +7 a + 12 V
- Pines de entrada analógica/ digital
- Corriente DC por pin de E/S: 40 mA
- Memoria Flash de 32 KB 2 KB para cargador de inicio
- Admite comunicación serie I2C
- Frecuencia de reloj: 16 MHZ
- Dimensiones: 0,73″ x 1,7″

2.5.2. Alimentación del Arduino Nano

El Arduino Nano puede ser alimentado mediante la conexión mini USB misma que no es regulada a 6-20V, o a su vez se puede usar una fuente de alimentación externa conectada en los pines 30, o usando 5V de algunas pilas o baterías conectadas en el pin 27. El Arduino escoge la fuente de alimentación automáticamente dependiendo del voltaje más alto.

2.5.3. Memoria

La memoria del Arduino nano usa un ATmega168 que incorpora 16 KB en su memoria flash para el almacenamiento de un código, de los cuales solo se utiliza 2 KB para el gestor de arranque, a diferencia de sus hermanos que usan un ATmega328 teniendo un total de 32 KB, y 2 KB que serán usado para el gestor de arranque. El ATmega168 tiene 1 KB de SRAM y 512 bytes de EEPROM los cuales pueden ser grabados o escritos mediando software con la librería EEPROM, el ATmega328 tiene 2 KB de SRAM y 1 KB de EEPROM.

2.5.4. Entrada y salida

Todos los 14 pines digitales del Arduino nano se pueden utilizar como una entrada o salida, según como el usuario lo requiera, estos pines operan a 5 voltios. Es de mucha importancia saber que cada pin entrega o recibe un máximo de 40 mA mismo que tienen resistencia de pull-up de 20 y 50 kOhms para realizar comunicación I2C se usan solo los pines A4 (SDA) y A5 (SCL).

Los pines que tienen funciones especializadas son los siguientes:

- Los pines: 0 (RX) y 1 (TX). Son usados para recibir (RX) y transmitir datos en forma serial (TX) TTL. Estos pines están conectados directamente al microprocesador de serie FTDI USB - a - TTL
- Los pines 2 y 3. Estos pines pueden ser usados y configurados para alguna interrupción en un valor bajo, para un flanco ascendente o para un descendente
- PWM: 3, 5, 6, 9, 10, y 11. proporcionar una salida de PWM de 8 bits con la función analogWrite
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines soportan la comunicación SPI
- LEDs: 13. Existe un led conectado al pin digital 13. Este identifica cuando el ALTO, el LED se encenderá y cuando este en BAJO el led permanecerá apagado

2.5.5. Comunicación

El Arduino Nano tiene una serie de instalaciones y puertos para lograr la comunicación, ya sea con un ordenador a otro Arduino o entre microcontroladores. Los procesadores ATmega168 y ATmega328 proporcionan una comunicación en serie, la misma que está disponible en los pines digitales 0 (RX) y 1 (TX), esta comunicación es en serie y logra través de un puerto USB y los drivers FTDI

proporcionan un puerto como virtual para el software refleje en el computador. El ATmega168 y ATmega328 también incorporan un puesto de comunicación I2C (TWI). Este Arduino nano mediante software oficial de Arduino puede simplificar el uso de la comunicación I2C.

2.5.6. Programación

El Arduino Nano se puede programar con el software de Arduino, solo hay que seleccionar al Arduino Nano en las Herramientas del menú de acuerdo con el microcontrolador que tiene su Arduino. Los ATmega168 o ATmega328 en el Arduino Nano como es habitual estos vienen precargados con un cargador de arranque, el mismo que permite cargar una nueva programación sin el uso de algún programador de hardware externo. Esto se logra mediante el uso de la comunican y el protocolo STK500.

Como es habitual entre los usuarios de Arduino, el estudiante presiona el botón físico de reinicio antes de cargar una programación, el Arduino Nano está diseñado de una manera automática de reinicio, esto se logra mediante el software que se ejecuta en un ordenador conectado. Este sistema deja como alternativa el uso del botón de reinicio, ya que todo este proceso lo realiza el Arduino nano mediante software.

2.6. Bluetooth HC-06

El modulo Bluetooth HC-06 es “un tipo de comunicación inalámbrica que logra la trasferencia de datos a través de radiofrecuencia usando la banda de 2,4 GHz. En nuestro campo existen 2 tipos de Bluetooth el HC-05 y el HC-06” (Bluetooth-Serial-HC-06, 2016, pág. 1)

El modulo Bluetooth HC-06 es el más usado en los proyectos, existe una simple diferencia entre estos dos módulos y es que el módulo HC-06 solamente funciona como esclavo, y el módulo HC-05 funciona como maestro y esclavo; lo cual sería una desventaja ya que nos impediría su uso en algún proyecto o desarrollo, sin embargo, el módulo HC-06 sirve tanto para enviar y recibir, todo dependerá de la configuración y el tipo de conexión que se realice.

Figura 13 Módulo Bluetooth HC-06

Fuente: (Bluetooth-Serial-HC-06, 2016)

2.6.1. Características del módulo Bluetooth HC-06

Entre las ventajas principales del módulo HC-06, podemos encontrar su costo, su pequeño tamaño, sus características de transmisión y recepción de datos los cuales brindan un alcance muy amplio al ser un sistema Bluetooth, además este módulo es el bajo consumo de corriente tanto en su funcionamiento, como en el modo de espera, es decir, que mientras el modulo este alimentado con energía, pero sin conexión o enlace a otro dispositivo no consumirá energía.

Otra de sus ventajas es que puede recordad a un dispositivo enlazada anteriormente, ya que cuenta con una memoria no volátil, y no vuelve a solicita el

código de validación que por defecto es 1234, en el caso que se requiera eliminar la lista de dispositivos almacenados, se debe aplicar tensión en el pin 26 (KEY) así el módulo HC-06 solicitará nuevamente la validación del enlace.

2.6.2. Pines del módulo Bluetooth HC-06

Físicamente los pines que encontraremos en el módulo son los siguientes:

- Vcc: Alimentación del módulo: 3,6V y 6V
- GND: Masa del módulo
- TXD: Transmisión de datos
- RXD: Recepción de datos a un voltaje de 3,3V
- KEY: Poner a nivel alto para entrar en modo configuración del módulo (esto solo pasa en el modelo HC-05)
- STATE: Se puede conectar un led en este pin para visualizar cuando se transfieran datos

2.6.3. Prueba de comandos AT

Los comandos AT son comandos que permiten al usuario configurar el módulo Bluetooth a través de un microcontrolador, un ordenador, un Arduino o con cualquier chip o dispositivo que permita realizar una comunicación serie (Tx/Rx); básicamente son instrucciones que permiten al usuario cambiar algunas configuraciones es como son: los baudios del módulo, el PIN, el nombre, etc.

Para hacer uso de los comandos AT se tiene que cumplir con las siguientes condiciones: el módulo Bluetooth no tiene que estar vinculado con ningún otro dispositivo Bluetooth, según las especificaciones del módulo el tiempo que tarda en el envío de un comando AT y otro tiene que ser de 1 segundo. En el caso que envié un comando AT y en menos de un segundo no se guardaran los cambios.

El usuario puede configurar mediante los siguientes comandos ATTC:

- Sin paridad (predeterminado) AT + PN
- paridad impar AT + PO
- Paridad par AT + PE

Tabla2
Comandos ATT

COMANDOS AT	DESCRIPCION	RESPUESTA
AT	Test de comunicación.	Responde con un OK
AT+VERSION	Retorna la versión del módulo.	OKlinvor 1.8
AT+BAUDx	1 configura 1200bps 2 configura 2400bps 3 configura 4800bps 4 configura 9600bps (Default) 5 configura 19200bps 6 configura 38400bps	AT+BAUDA4 Configura la velocidad a 9600 baud rate responde con OK9600
AT+NAMEx	Configurar el nombre con el que se visualiza el modulo, soporta hasta 20 caracteres	**AT+NAMEDITYMarkers** Configura el nombre del módulo a **DIYMarkers** responde con **OKsetname**
AT+PINxxxx	Configurar en Pin de Acceso al módulo (pasword).1234 por defecto.	AT+PIN1122 Configurar el PIN a 1122 Responde con **OKsetPIN**

Fuente: (Bluetooth-Serial-HC-06, 2016)

2.7. Comunicación I2C

La comunicación mediante protocolo I2C, tiene la peculiaridad de facilita la comunicación entre, dispositivos virtuales, microcontroladores, memorias, Arduino y

otros dispositivos, esta comunicación solo requiere de dos líneas de señal y un común o masa. Esta comunicación fue desarrollada por la gran compañía Philips, que básicamente permite el intercambio de información entre algunos dispositivos y a una velocidad considerable.

Además, se logra con esta comunicación alrededor de unos 100 Kbits por segundo, aunque se han logrado en casos especiales que el reloj llega hasta los 3,4 MHz. El sistema de comunicación y trasferencia de los datos del bus I2C es en serie y sincrónica. Por lo tanto, una de las líneas del bus marca el tiempo de pulsos del reloj y la otra línea se encarga de intercambiar datos.

A continuación, se detalla la descripción de las siguientes señales:

- **SCL** (System Clock) esta es una línea de pulsos de reloj que sincronizan el sistema
- **SDA** (System Data) esta es la línea por la que se tramitaran los datos entre los dispositivos
- **GND** (Masa) esta línea es el común de la interconexión entre todos los dispositivos enganchados en el bus I2C

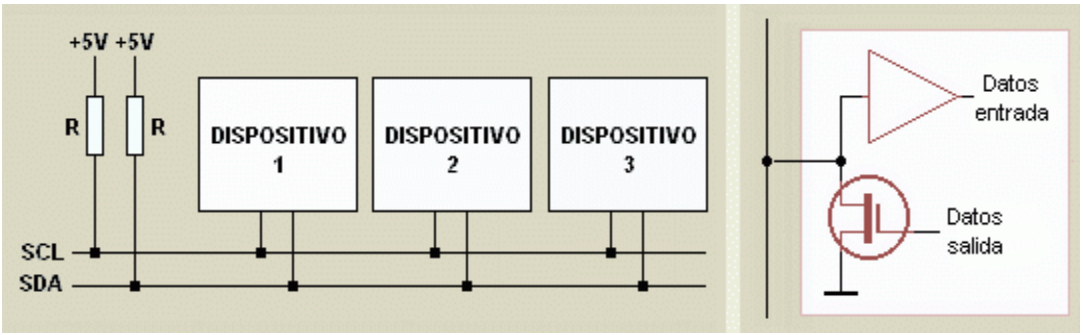

Figura 14 Comunicación I2C

Fuente: (argentina, 2014)

Tendremos que poner mucha atención en la imagen se puede apreciar que las dos líneas del bus I2C están en nivel lógico alto cuando estas no están activas. También es de mucha importancia hacer mención que no existe un límite de comunicación

entre dispositivos en todo el bus, tomando en cuenta que la suma de todos dispositivos en el proyecto no debe exceder los 400pF.

2.7.1. Funcionamiento de la comunicación I2C.

Datos importantes que tienen que ser tomados es que es una comunicación serial, utilizando un conductor para manejar el timming (SCL) (los pulsos del reloj) y el otro conducto para el intercambio de datos (SDA), el cual trasportara toda la información entre todos los dispositivos que se encuentren conectados en el bus. Las dos líneas del bus SDA (Serial Data) y SCL (Serial Clock) están conectadas a la fuente de alimentación a través de resistencias pull-up, cuando el bus esté libre estas dos líneas estarán en nivel alto.

El dispositivo puede ser considerado como Master (Master) o como esclavo (Slave), tomando en cuenta que el maestro es el dispositivo que inicia la transferencia de datos en el bus, es el que genera una señal de Clock y el esclavo (Slave) es el dispositivo direccionado. Entre unas de las características más importantes es que cada dispositivo es reconocido por una dirección única ya sea un microcontrolador, una pantalla LCD, una memoria o un teclado, cualquiera de los dispositivos puede operar como transmisor o receptor de datos, esto dependerá de la función del dispositivo.

Transmisión de bit:

A continuación, se enlista las características de trasmisión:

- Los bits de datos van por el canal SDA
- Por cada bit de información es necesario un pulso del SCL
- Los datos sólo pueden cambiar y transitar cuando SCL está en nivel bajo
- El bit más significativo es el que se enviará primero
- El maestro siempre es el que genera el reloj
- Los datos y direcciones que se transmiten por SDA son de 8 bits
- Tras cada bloque debe recibirse una señal de reconocimiento

- El bus permite la conexión de varios Masters, ya que incluye un detector de colisiones
- El bus serial I2C ha sido extendido para soportar velocidades de hasta 3.4 Mbits/s

La transferencia de datos del bus I2C tiene los siguientes modos:

- Modo Estándar a aproximadamente a 100kBits/Seg
- Modo Rápido aproximadamente a 400kbits/Seg
- Modo Alta velocidad a más de 3,4Mbits/Seg

2.7.2. Definición de términos en I2C

- **Maestro:** Es el único dispositivo que decreta los tiempos y la dirección de la información en el bus, es el encargado en activar los pulsos del reloj en la línea SCL.
- **Esclavo:** Son todos los dispositivos conectados en el bus, estos dispositivos reciben la señal de comando y reloj del maestro.
- **Bus libre:** Cuando el bus está libre quiere decir que tiene las dos líneas (SDA y SCL) inactivas este es el único momento en que el maestro puede usar del bus.
- **Comienzo:** El maestro ocupa el bus, crea la condición de inicio y la línea (SDA) toma un estado bajo a lo contrario de la línea de reloj (SCL) que permanece en alta.
- **Parada:** El maestro genera esta condición dejando libre el bus y dejando las dos líneas de datos y de reloj en estado lógico alto.
- **Dirección:** Todos los dispositivos están diseñado para funcionar en el bus I2C, se debe designar una dirección diferente a cada esclavo para que el maestro los puede reconocer.
- **Lectura/Escritura:** Cada uno de los esclavos tiene una dirección de 7 bits. Siendo el octavo el menos significativo, este indica que operación se va a

cumplir. En donde si el bit es alto el maestro lee información del esclavo, si el bit es bajo el maestro escribe información en el esclavo.

- **Dato válido:** Un dato valido se logra cuando el dato esté presente en la línea SDA y este sea estable al tiempo en que la línea SCL este en nivel lógico alto.
- **Formato de Datos:** Los datos y direcciones que se transmiten por SDA son de 8 bits de dato 1 byte, posteriormente se crea un noveno pulso de reloj en el cual el dispositivo receptor de información generara un pulso de reconocimiento.
- **Reconocimiento:** El reconocimiento se logra colocando la línea de datos a un nivel lógico bajo durante la transmisión del noveno pulso de reloj.

2.8. App inventor

App Inventor es “un entorno de desarrollo de aplicaciones para dispositivos Android, para desarrollar aplicaciones con App Inventor sólo necesitas un navegador web y un télefono o tablet Android” (Appinventor, 2016, pág. 1)

App Inventor usa un lenguaje de programación parecido a los bloques de un rompe cabezas que están orientados o diseñados a cumplir eventos de programación, así podremos indicarle al procesador de un dispositivo Android como actuar o desarrollar alguna aplicación. Además, se ha convertido en una web para desarrolladores que quieres introducirse en la programación, ya que con App inventor se pude desarrollar aplicaciones ya sean simples o complejas.

A continuación, se detalla los accesorios para desarrollar una aplicación con App Inventor:

- Un navegador web
- Un dispositivo Android
- Un emulador de aplicaciones en el caso de no tener un dispositivo Android.

Todo el funcionamiento del App Inventor en mediante un servidor web, el cual permite grabar todas nuestras apps desarrolladas o trabajos a ser realizados, ya en la interfaz de programación se trabajará dos herramientas: App Inventor Designer y App Inventor Blocks Editor. En el App Inventor Designer es donde se puede construir la Interfaz que el usuario la cual verá en la pantalla de su dispositivo, elijaremos los elementos y componentes con los que el usuario interactuará en la aplicación.

En el Blocks Editor se diseñará la interfaz con la cual mantendremos internación, y toda la programación que vendrá hacer el cerebro de todos los componentes, comandos, botones y funciones de nuestra aplicación. A continuación, se detalla tres pasos fundamentales para entender el diseño y el funcionamiento del software App Inventor:

- El gestor de proyectos
- El diseñador
- EL editor y programación de bloques

Entre las principales características del App Inventor tenemos las siguientes:

- El software es libre y no necesita de licencias para su uso
- Es una multiplataforma que solo necesita de un navegador y la máquina virtual de Java instalada en el computador
- La programación está diseñada solo para dispositivos móviles
- Hay que tomar en cuenta que los Smartphone y Tablet han tenido una buena acogida en todo el mundo, siendo estos los dispositivos más usados a nivel mundial

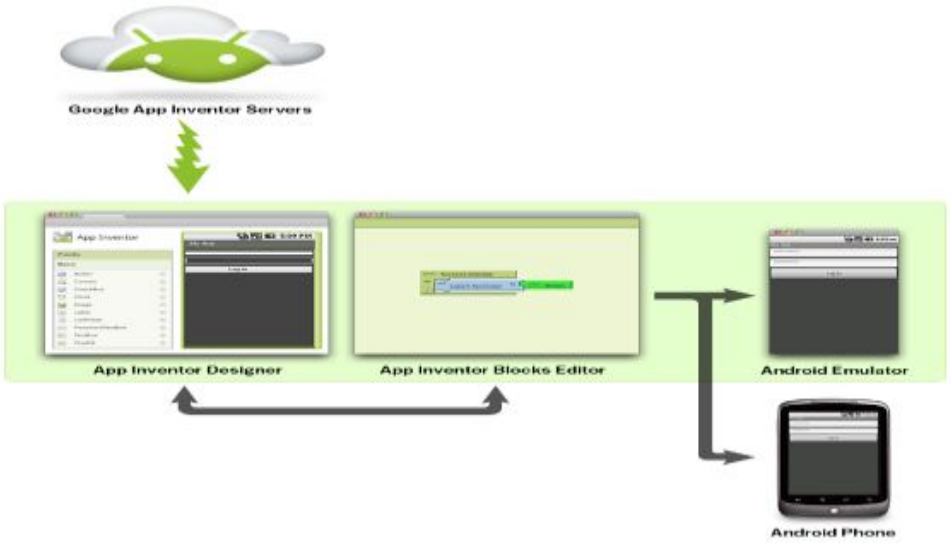

Figura 15 Protocolo App inventor

Fuente: (Appinventor, 2016)

CAPÍTULO III

DESARROLLO DEL PROYECTO

3.1. Preliminares

Para la implementación del presente trabajo de titulación cuyo tema es "**CONTROL DE MOVIMIENTO DE UN ARDUINO ROBOT EMPLEANDO UNA BRÚJULA DIGITAL PARA PRÁCTICAS DE MICROCONTROLADORES",** fue necesario el uso y la adquisición las siguientes placas electrónicas, software, elementos electrónicos y materiales.

- Arduino Robot
- Brújula Digital HMC6352
- Arduino Nano
- Módulo Bluethooth
- Smartphone Android
- Software IDE Arduino

3.2. Introducción del proyecto.

El presente proyecto, tiene como objetivo, implementar un sistema de control de movimiento de un Arduino Robot, desde un Smartphone con una aplicación en sistema Android, empleando una brújula digital para realizar prácticas en la asignatura de microcontroladores de la Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas – ESPE. Para ello, se estableció las siguientes orientaciones: qué es lo que se va hacer, qué vamos a utilizar, cuántos recursos se necesita y qué disponibilidad de tiempo se le va a dar al proyecto para poderlo concluir.

La estructura del presente proyecto es la siguiente:

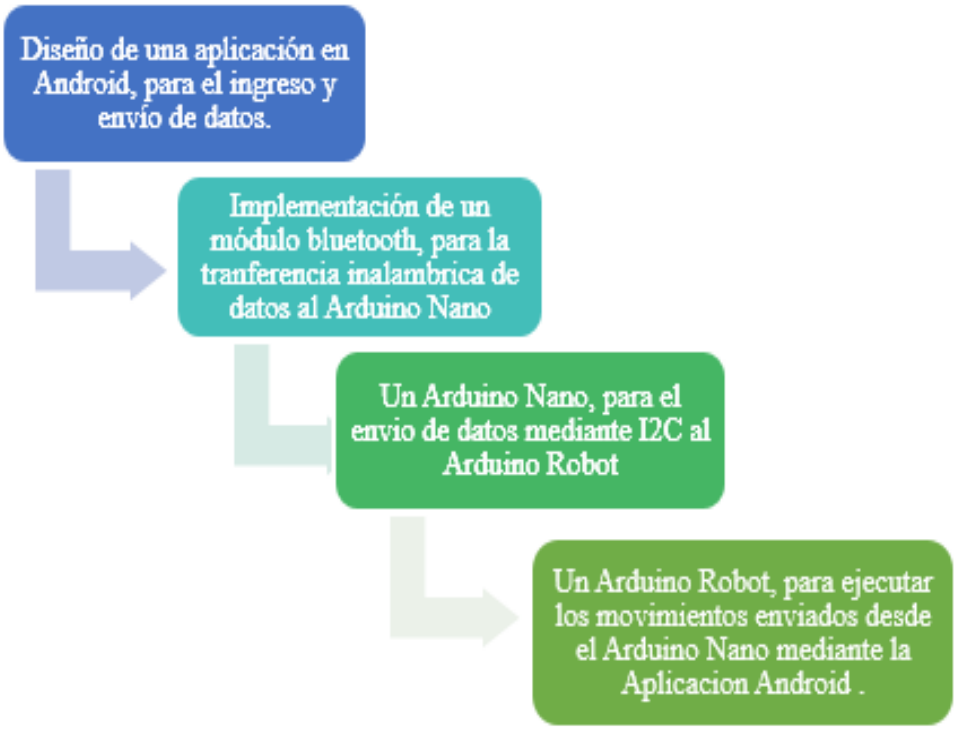

Figura 16 Estructura del Proyecto

Para lograr el funcionamiento del sistema de control de movimiento de un Arduino Robot, se utilizaron los siguientes elementos:

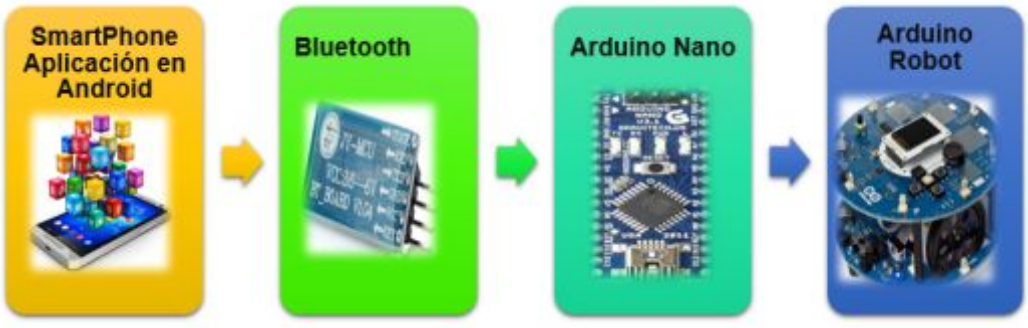

Figura 17 Diseño de Proyecto de Control de Movimiento

Para la selección de los componentes y elementos, estos fueron analizados previamente, tanto en la parte del hardware y el software, para verificar que cumplan con las características requeridas para el presente proyecto, los mismos que serán usados en todo el desarrollo del proyecto, permitiendo la programación y comunicación entre ellos.

- **Smartphone:** Tiene como función enviar vía Bluetooth los datos de las coordenadas y los tiempos ingresados por el usuario mediante una aplicación Android.
- **Modulo Bluetooth:** Tiene como funciones la vinculación entre el modulo y el Smartphone. Él envió inalámbrico de los datos ingresados en la aplicación al Arduino nano.
- **Arduino Nano:** Tiene como función recibir la trama de datos enviada por la aplicación Android, validad la información y posteriormente enviar mediante protocolo de comunicación I2C al Arduino Robot.
- **Arduino Robot:** Tiene como función validar la información enviada por el Arduino nano. Ejecutar los movimientos en las coordenadas y los tiempos ingresados por el usuario en la aplicación.

El sistema de control de movimiento del Arduino Robot mediante una brújula digital, requiere la siguiente estructura y se implementó de la siguiente manera:

- El usuario debe verificar que el Bluetooth de su Smartphone este encendido y vinculado con el Bluetooth del Arduino Robot.
- Al presionar el botón Conectar, la aplicación iniciara una conexión entre el Bluetooth del Smartphone y el Bluetooth del Arduino Robot.
- Al presionar el botón Desconectar, la aplicación desconectará de la vinculación Bluetooth del Smartphone con el Arduino Robot.
- Al presionar el botón Enviar, la aplicación enviará las coordenadas y tiempos ingresados al Arduino Robot.
- Al presionar el botón instrucciones, automáticamente se desplegará una lista con todas las instrucciones de uso de la aplicación. Al presionar el botón “Volver”: La aplicación le permitirá regresar a la página principal.

En los campos COORDENADA 1, 2,3, Se debe ingresar la coordenada a ser ejecutada por el Arduino Robot. Se debe ingresar una variable de máximo 4 dígitos, por ejemplo: N45E, n45e. Es importante recalcar que, solo se podrá ingresar datos dentro de los siguientes parámetros basados en el formato de coordenadas geografías, ya sea con letras mayúsculas o minúsculas ejm: N<90E, N <90O, S<90O, S<90E.

En los campos TIEMPO1, 2,3, el usuario debe ingresar el tiempo en segundos, en una escala de 1 a 99 segundos como máximo, los cuales el Arduino Robot pondrá en funcionamiento los dos motores y de esta manera estará en movimiento en la coordenada ingresada en la App. Ingresaremos una variable numérica de máximo 2 dígitos, por ejemplo: 15seg.

3.3. Diseño de la aplicación Android

A continuación, se detallarán el diseño de la aplicación, funcionamiento de cada bloque de programación y las diferentes pruebas de rendimiento realizadas:

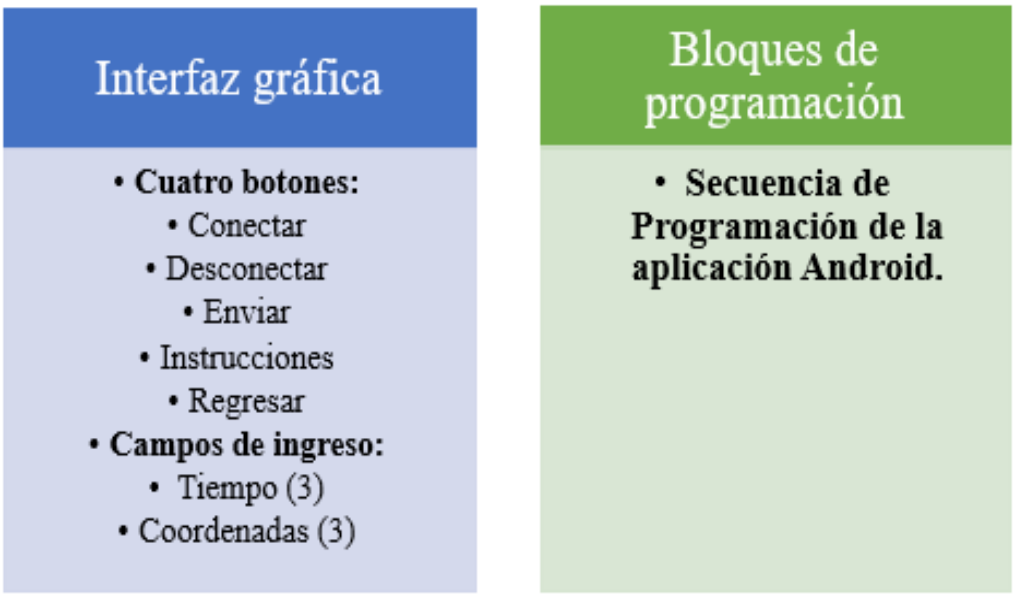

Figura 18 Interfaz y Bloques de programación

3.3.1. Interfaz gráfica

La interfaz es la capa que hay entre el usuario y el corazón funcional de la app, es el lugar donde nacen las interacciones. La aplicación está compuesta por botones, imágenes e íconos, los cuales pueden tener una apariencia visual diferente en cada dispositivo Android, el diseño está basado en que sea una interfaz que el usuario la puede entender, pero a su vez que sea atractiva ante los ojos de los diferentes usuarios.

La pantalla principal de la aplicación está diseñada con imágenes de fondo y una gama de colores, cuenta de cuatro botones que son: conectar, desconectar, enviar e instrucciones, también, cuenta con seis campos de ingreso de textos en los cuales el usuario ingresará: tres coordenadas geográficas y tres tiempos diferentes los cuales al final del proceso serán ejecutados por el Arduino Robot.

Figura 19 Interfaz App

Conectar: Este botón es de mucha importancia, ya que permite al usuario desplegar una ventana en la cual se imprimirá una lista con todos los dispositivos Bluetooth existentes en el entorno. En esta opción, es necesaria la vinculación entre el Bluetooth del Smartphone y el módulo Bluetooth 20:15:06:10:03:61 HC-06, ya que de esta manera se podrá mantener una conexión entre dispositivos para el envío y recepción de datos ingresados en la aplicación Android.

Es de suma importancia mencionar que, es de manera obligatoria la vinculación entre el modulo Bluetooth 20:15:06: 10:03:61 HC-06 con el dispositivo celular del usuario, ya que de caso contrario no se podrá establecer una conexión entre estos dos dispositivos, de tal manera se impedirá en ingreso y el envió de coordenadas geográficas y tiempos al arduino robot.

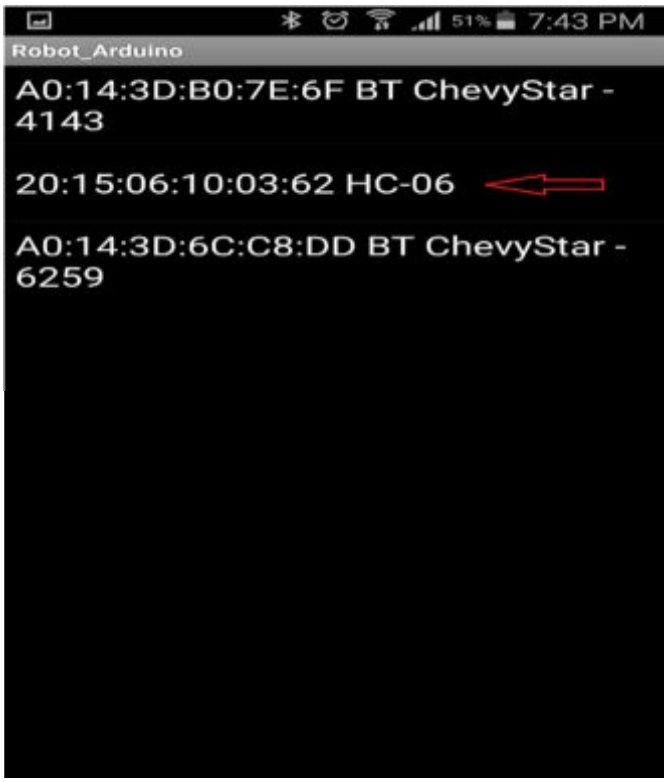

Figura 20 Interfaz App Bluetooth

Posterior a la vinculación con el módulo Bluetooth HC-06, la aplicación enviará en la pantalla del Smartphone un mensaje de “Dispositivo Conectado”, esto quiere decir que la vinculación entre los dispositivos Bluetooth se ha logrado con éxito. A continuación, el usuario podrá interactuar con la aplicación Android, donde podrá ingresar las tres coordenadas y los tres tiempos diferentes a ser ejecutados por el Arduino Robot.

Figura 21 Interfaz App Bluetooth 2

Coordenada 1, 2, 3: En estos campos se debe ingresar las tres coordenadas a ser ejecutadas por el Arduino Robot, mismas que tendrán que cumplir las siguientes condiciones: no tendrán que ser mayores a cuatro caracteres, no se tomaran en cuenta el ingreso de mayúsculas o minúsculas, por ejemplo: N45O, n45o, datos que serán basados en el formato de las coordenadas geográficas estandarizadas entre N90O, N90E, S90O, S90E.

Es decir, no se podrá ingresar datos numéricos mayores a 90 grados, ya que caso contrario la aplicación nos despliega un mensaje en la pantalla de “Datos no Permitidos”, de esta manera está obligando al usuario a ingresar datos dentro de los parámetros de las coordenadas geográficas y tiempos establecidos en las instrucciones y en la programación.

Tiempo 1, 2, 3: En estos tres campos de texto el usuario debe ingresar dos variables numéricas en cada campo de texto, que puede ser entre los números: 1 y 99 como máximo, los mismo que representaran el tiempo en segundos en que el arduino robot se mantendrá en movimiento siguiendo la trayectoria de la coordenada geográfica ingresada en la aplicación. En el caso de que el usuario no ingrese los tres tiempos en el formato y parámetros antes mencionados, nuevamente se desplegará un mensaje en la pantalla de la aplicación de “Datos no Permitidos” obligando al usuario a ingresar datos dentro de los parámetros establecidos en la programación.

Figura 22 Interfaz App Validación de Datos

Desconectar: Este botón permitirá al usuario desconectar la vinculación entre el Bluetooth del Smartphone y el módulo Bluetooth HC-06 incorporado en el Arduino Robot, de esta manera se finalizará toda comunicación entre dispositivos, en el caso que ocurra algún problema con la vinculación, se recomienda cerrar la aplicación y realizar nuevamente el proceso.

Enviar: Mediante la comunicación inalámbrica Bluetooth entre los dos dispositivos, este botón permite enviar los datos que el usuario ingresó en los tres campos de coordenadas y los tres campos de tiempos en la aplicación, al Arduino Nano, para posteriormente serán enviados al Arduino Robot mediante protocolo de comunicación I2C.

Figura 23 Interfaz App Datos Enviados

Instrucciones: Este botón permite desplegar una nueva interfaz gráfica donde se detallarán las instrucciones del funcionamiento de la Aplicación, como, por ejemplo: el funcionamiento de cada botón, parámetros máximos que podrán ser ingresados en los campos de coordenadas geografías y tiempos a ser ejecutados por el Arduino Robot.

Regresar: Este botón le permitirá al usuario salir de la interfaz de instrucciones, posteriormente ingresará automáticamente a la interfaz principal, la cual ha sido detallada anteriormente y donde el usuario podrá empezar continuar con el proceso de control de movimiento, ingresar los movimientos y tiempos a ser ejecutados por el Arduino Robot.

Figura 24 Interfaz botón regresar

3.3.2. Programación de la aplicación Android

La programación de la aplicación está programada para el envío de mensajes al usuario, estas están basadas en distintas validaciones dependiendo del estado de conexión Bluetooth; es decir, su diseño está pensado en asegurar la conexión y vinculación entre el Smartphone y el modulo Bluetooth, es por eso que el Bluetooth del Smartphone debe estar activado, de no estarlo la aplicación envía un mensaje de "Bluetooth no activado".

En el caso que el Bluetooth este encendido, pero no se haya realizado la vinculación con el modulo, la aplicación enviara un mensaje de "desconectado", cuando el usuario haya ingresado incorrectamente las coordenadas y tiempos la aplicación enviara un mensaje de "datos no permitidos" y cuando el usuario ingrese datos dentro de los parámetros establecidos se enviara un mensaje de "datos enviados".

Se ha creado un bloque Bnt conectar, este bloque validad que el dispositivo Bluetooth del Smartphone este encendido y automáticamente imprime una lista con todos los dispositivos Bluetooth existentes en el entorno, posteriormente le permitirá la vinculación con algunos de los dispositivos, al final enviará un mensaje de "dispositivo conectado".

De la misma manera, se trabajó en finalizar la conexión inalámbrica, es por eso que se creó un bloque Bnt desconectar, el cual tiene como función llamar al bloque Bluetooth y posteriormente desactivar la vinculación y la conexión, de manera inmediata la aplicación imprimirá en la pantalla de la app un mensaje de "desconectado".

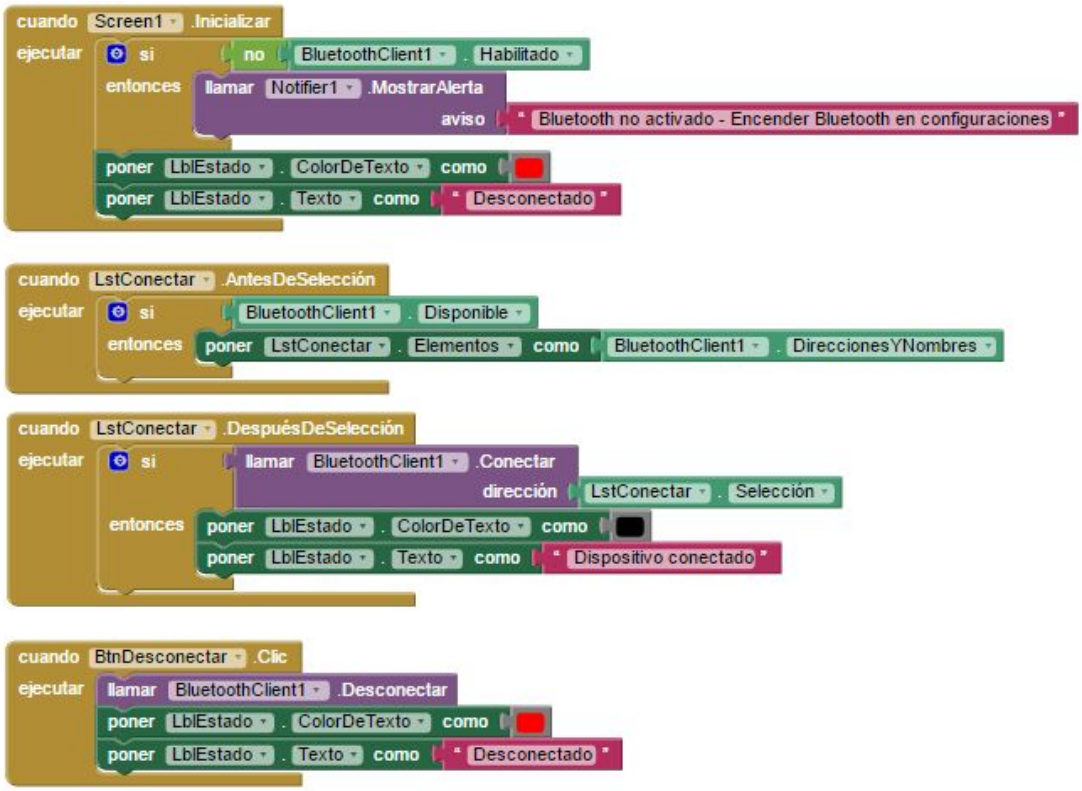

Figura 25 Programación en App Inventor2

Para el envió de datos se diseñó un Bloque Bnt enviar, el mismo que debe cumplir con las siguientes condiciones:

Con un bloque comparativo la aplicación valida la longitud ingresada en coordenadas 1, 2 y 3, la misma que tendrán que cumplir con el formato de las coordenadas geográficas, sus longitudes no deben ser mayor a cuatro caracteres. Si el usuario no ingresa los datos dentro de los parámetros antes mencionados la aplicación no permitirá enviar las coordenadas y tiempos ingresados enviando un mensaje de datos “no permitidos”.

De la misma manera se validará la longitud ingresada en los campos de texto de tiempo 1,2 y 3, estos datos no deben ser mayores a dos caracteres. Si el usuario no ha ingresado datos dentro de los parámetros establecidos, la aplicación no permitirá enviar las coordenadas y tiempos ingresados enviando un mensaje de datos “no permitidos”.

Al final luego de que la aplicación haya realizado todas las validaciones antes mencionadas, se podrán enviar los datos vía Bluetooth a un Arduino nano. Se diseñó una lógica para lograr organizar la toda la información a ser enviada que consiste en enviar un “+” al inicio de cada dato ingresado ya sean coordenadas o tiempos, al final de la trama de datos se enviará un “*” seguido de un signo de “$”, manteniendo la siguiente secuencia: +C1+C2+C3+T1+T2+T3+*$.

El signo “*” se enviará al final de toda la trama de datos para que el Arduino nano sepa cuando esta termina, el signo “$” se enviara cuando todos los datos se hayan validado y únicamente realizará operaciones que serán de mucha importancia para el envío de datos al Arduino robot vía proctólogo de comunicación I2C, mismas que se detallaran en cada uno de los bloques de programación.

Figura 26 Programación en App Inventor2 Validación de Datos

Es de mucha importancia mencionar que la aplicación solo valida la longitud de los datos ingresados, mas no si el usuario realizo el ingreso de coordenadas correctas basadas en el formato de coordenadas geográficas, es por eso que se ha creado un temporizador que actúa cada mil milisegundos, de esta manera se podrá saber cuándo la aplicación estará enviando datos, de la misma manera se habilita la recepción de una “V” y una “F” que serán enviados por el Arduino Nano.

El Arduino Nano enviará a la aplicación una “F” cuando no se ha cumplido con la validación, obligando al usuario que ingrese nuevamente los datos cumplimento el formato antes mencionado. De la misma manera, enviará una “V” cuando todos los datos cumplan con las validaciones, posteriormente la aplicación enviará un mensaje de “datos enviados” finalizando el proceso de programación.

3.4. Implementación del módulo Bluetooth HC-06

Figura 27 Implementación módulo Bluetooth

La implementación del módulo Bluetooth HC-06 cumple la función de enviar inalámbricamente los datos ingresados por el usuario en la aplicación Android a un Arduino Nano, esto se lo realiza utilizando comunicación serial, comunicación que no se pudo realizar directamente con el Arduino Robot, es por eso que se implementó un Arduino Nano para establecer una combinación con el Arduino robot mediante I2C.

Para la comunicación Bluetooth se cumplen las siguientes condiciones:

- La comunicación es entre un maestro, el cual es el Smartphone y el esclavo que es el prototipo en este caso el módulo HC-06
- La contraseña para la vinculación es la que viene por defecto de fábrica "1234"
- Por defecto la velocidad de comunicación es de 9600 baudios

Asimismo, se adquirió un módulo Bluetooth HC-06, no se realizaron ningún cambio en sus configuraciones de fábrica, el módulo HC-06 fue conectado en una placa protoboard la cual fue fijada en la tarjeta de control del Arduino Robot, conectado junto al Arduino Nano. Las configuraciones de fábrica pueden ser cambiadas de acuerdo a las necesidades del usuario, estos cambios se pueden lograr haciendo uso de los comandos ATT.

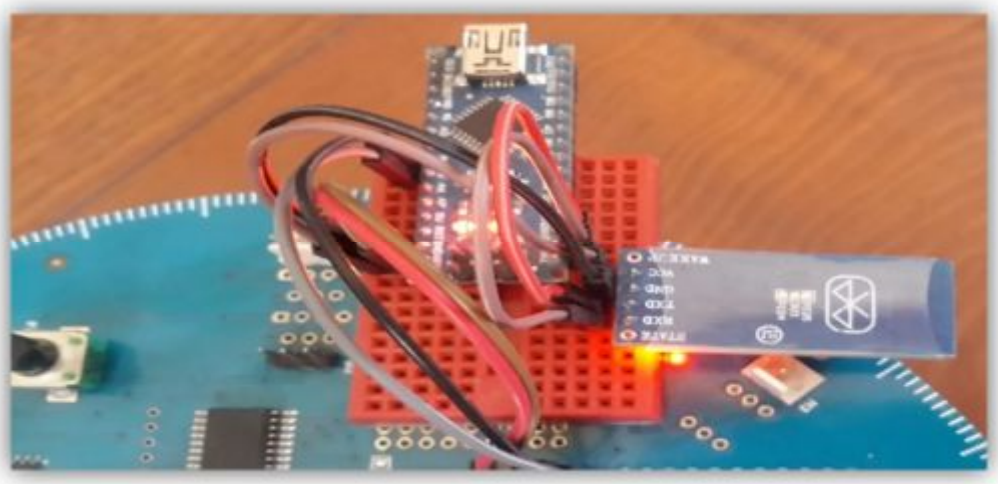

Figura 28 Implementación del Módulo Bluetooth HC-06.

Los siguientes dispositivos: El modulo Bluetooth HC-06 y el Arduino Nano son alimentados de energía por el Arduino Robot, se realizaron las conexiones en los pines VCC y GND del módulo Bluetooth en los pines VCC y GND de la placa de

control del Arduino Nano, estos pines son alimentados de 5v por el Arduino Robot mediante el puerto de comunicación I2C

Es importante resaltar que, el módulo HC-06 funciona a 3, 3 V, hay mucha información en las páginas de internet que hablan sobre como alimentar al módulo o si es factible poner divisores de tensión para acceder a los pines del módulo con un Arduino. Por los resultados que se ha podido validar, me atrevo a decir que no es necesario el uso de divisores de voltaje ya que el modulo no fue afectado en su funcionamiento, por lo que concluyo que sí es recomendable conectar directamente los pines del Arduino Nano.

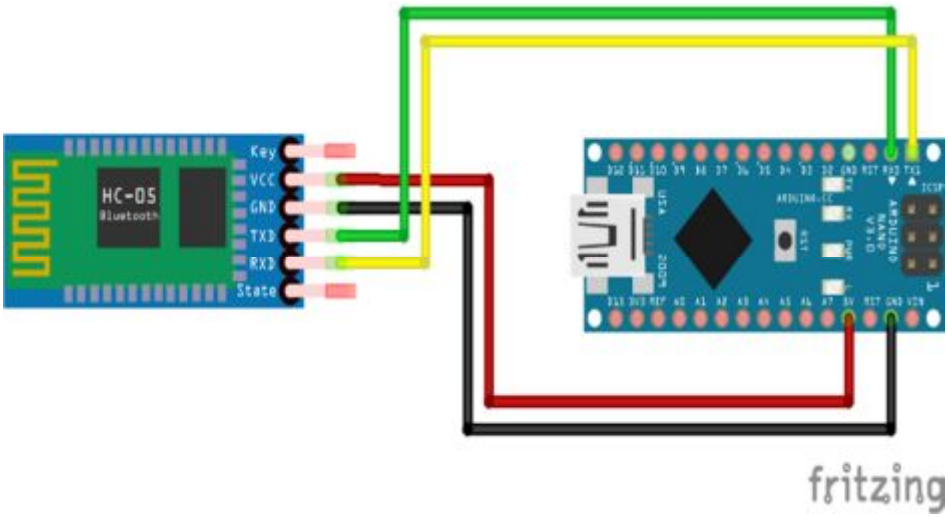

Figura 29 Alimentación de Módulo Bluetooth HC-06

Fuente: (Bluetooth-Serial-HC-06, 2016)

El módulo HC-06 no necesita de ninguna programación extra para entrar en modo de aceptar comandos AT, en este caso no se realizó ningún cambió en la configuración que viene de fábrica, por lo tanto, se trabajó con las configuraciones que viene por de defecto, por ejemplo: La velocidad de comunicación entre el modulo y el Arduino Nano es de 9600bps, el código para acceder a la vinculación

sigue siendo el mismo que viene por defecto que es el “1234” así como su nombre que es HC-06.

Al iniciar la conexión entre el modulo Bluetooth y el Smartphone el LED indicador del módulo Bluetooth pasará de parpadeante a estático lo que asegura que el prototipo ha realizado exitosamente la conexión con el Smartphone. Hay que asegurar que el Bluetooth del Smartphone esté activo y por último vincularse desde la App instalada en el dispositivo Android.

3.5. Programación del Arduino Nano.

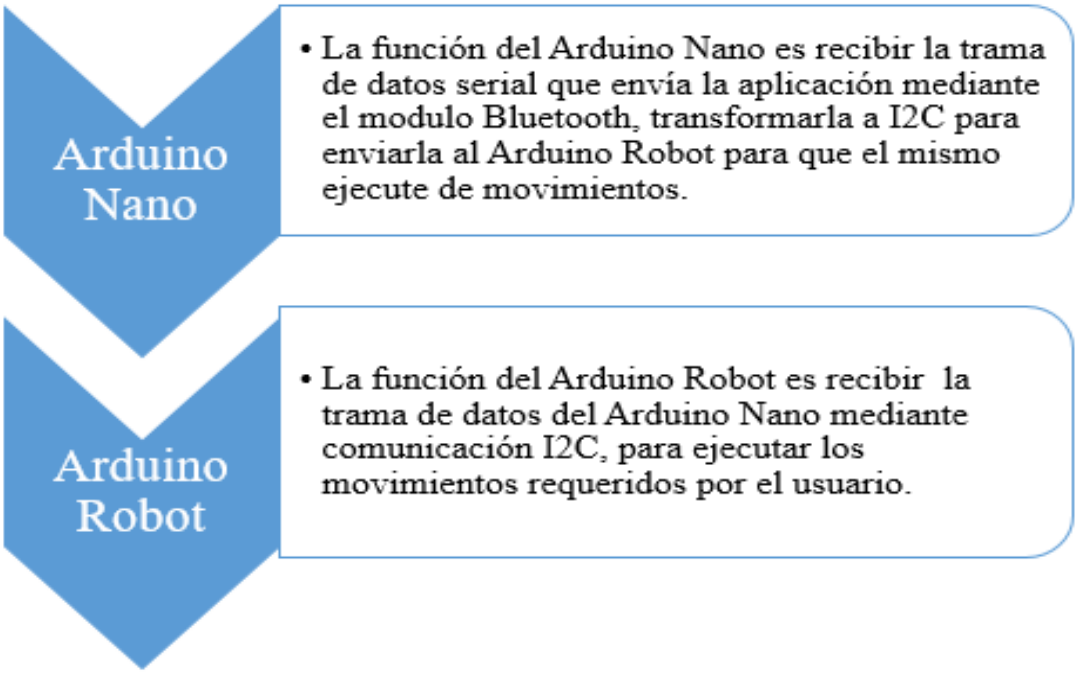

Figura 30 Programación del Arduino Nano

El Arduino Robot tiene pines analógicos y digitales, tanto de entrada como de salida, la desventaja es que todos sus pines son multiplexados es decir todos los pines están conectados en una misma línea, y su puerto serial es usado para la comunicación entre la placa de control y la placa motora, esto causa que no se pueda

establecer una conexión serial directa entre el modulo Bluetooth y el Arduino Robot, por lo tanto, se realizaron pruebas implementando librerías para usar otros pines como entradas seriales sin tener buenos resultados.

Como solución se implementó un Arduino Nano para recibir la comunicación serial del módulo Bluetooth, procesar la información, transformar y posteriormente enviar al Arduino Robot mediante comunicación I2C para ejecutar la información de coordenadas y movimientos. Básicamente, la función del Arduino Nano es recibir la trama de datos serial que envía la aplicación Android mediante el modulo Bluetooth, validar los datos recibido, transformar la trama de datos validada en protocolo de comunicación I2C para posteriormente enviar al Arduino Robot para que ejecute los movimientos.

Entre los puntos importantes de la programación se puede mencionar los siguientes:

- Importar librerías para el uso de comunicación I2C
- Recepción de datos vía Bluetooth
- Recepción del vector con coordenadas
- Recepción del vector con tiempos
- Validación de coordenadas
- Validación de tiempos
- Asignación de dirección al Arduino esclavo
- Bandera indicadora de datos recibidos vía Bluetooth
- Trama de datos enviada vía protocolo de comunicación I2C

```
#include <Wire.h>

char blueToothVal;
String datosR = String();
String datos = String();
String coord[3];
int index[7];
int tiempo[3];
char datosE[30];
int val1 = 0;
int val2 = 0;
int val3 = 0;
int flag = 0;
```

Figura 31 Importación de librerías

Recepción de datos vía Bluetooth

El Arduino Nano recepta los datos de forma serial vía Bluetooth, cada dato ingresado por la aplicación Android en coordenada 1,2,3 y tiempo 1,2,3, es separado por una ",", para finalizar la trama de datos se envía el carácter "*" y para saber que la trama de datos terminó y así dejar de almacenar los datos se envió un signo de "$", posteriormente será almacenada en la variable blueToothVal, al final se creó una variable flag que es una bandera que se pondrá en 1 cuando todos los datos se hayan recibido.

```
if(Serial.available())
{
  blueToothVal=Serial.read();
  if(blueToothVal!='$'){
    datosR += blueToothVal;
  }
```

Figura 32 Recepción de datos vía Bluetooth

Codificación de datos recibidos

Para la codificación de datos se cumplirá con las siguientes condiciones:

- Se creó una variable indexOf, que será el índice del lugar donde se almacena cada dato en la trama de datos final, con la función lastdatos asignamos una posición a cada carácter separado por un "+" en la trama de datos recibidos.
- Una vez que se obtiene el índice de los datos de la variable datosR, se extraerá las coordenadas de la trama de datos recibidos y se almacenará en la variable coord. El mismo proceso se realizó con los datos de los tiempos, se extraerán los tiempos de la trama de datos recibidos y se almacenará en la variable tiempo.
- Existen tres variables val1, val2, val3, cada una de ellas se pondrán en 1 lógico cuando estén validadas, de esta manera se seguirá el resto del proceso de la programación.

```
  if (blueToothVal=='*')
  {
      flag = 1;
      // Busca la posicion del caracter '+' en la trama de datos recibidos
      for(int i = 1; i < 7; i++){
        index[i] = datosR.indexOf('+',last+1);
        last = index[i];
      }
      // Extraigo las coordenadas de la trama de datos recibidos
      for(int j = 1; j < 4; j++){
        coord[j-1] = datosR.substring(index[j-1]+1,index[j]);
        //Serial.println(coord[j-1]);
      }
      // Extraigo el tiempo de la trama de datos recibidos
      for(int k = 4; k < 7; k++){
        tiempo[k-4] = datosR.substring(index[k-1]+1,index[k]).toInt();
        //Serial.println(tiempo[k-4]);
      }
    val1 = Validation(coord[0]);
    val2 = Validation(coord[1]);
    val3 = Validation(coord[2]);
    datosR = String();
    delay(10);
  }
}
```

Figura 33 Codificación de datos recibidos

Validación de coordenadas

En la aplicación Android se valida la longitud de los datos ingresados, mas no la información de coordenadas y tiempos ingresados por el usuario. Además, se tendrá diferentes casos de validación basados en el formato de coordenadas geográficas y basadas en las dimensiones como, por ejemplo, se podrá ingresar una coordenada de tres caracteres como N3E, y de cuatro caracteres N45E.

El primer caso es de dimensión tres, consiste en validar que la primera coordenada este entre norte o sur ya sea mayúsculas o minúsculas, la segunda coordenada este entre oeste o este ya sea mayúsculas o minúsculas, y el número que representa el ángulo este entre 1 y 9. Al final se retornara un 1 si es verdadero o de lo contrario se retornará un 0 si es falso.

El segundo caso de dimensión cuatro, consiste en validar que la primera coordenada este entre norte o sur, la segunda coordenada este entre oeste o este ya sean mayúsculas o minúsculas, Que el primer número este entre 1 y 8, y que el segundo número este entre 0 y 9. Al final se retornara un 1 si es verdadero o de lo contrario me retorne un 0 si es falso. Este proceso se repite con las tres coordenadas, y se retorna un 1 en tres variables val1, val2, val3.

```
int Validation(String coord){
  int dim = coord.length();
  String coord1;
  String coord2;
  int num1;
  int num2;
  int theta;
  if (dim == 3){
      coord1 = coord.substring(0,1);
      num1 = coord.substring(1,2).toInt();
      coord2 = coord.substring(2);
      if (coord1 == "N" || coord1 == "n" || coord1 == "S" || coord1 == "s"){
        if (coord2 == "O" || coord2 == "o" || coord2 == "E" || coord2 == "e"){
          if (num1 > 0 && num1 < 10){
            return 1;
          } else return 0;
        } else return 0;
      } else return 0;
  } else
  if (dim == 4){
      coord1 = coord.substring(0,1);
      num1 = coord.substring(1,2).toInt();
      num2 = coord.substring(2,3).toInt();
      coord2 = coord.substring(3);
      if (coord1 == "N" || coord1 == "n" || coord1 == "S" || coord1 == "s"){
        if (coord2 == "O" || coord2 == "o" || coord2 == "E" || coord2 == "e"){
          if (num1 > 0 && num1 < 9 && num2 >= 0 && num2 < 10){
            return 1;
          } else return 0;
```

Figura 34 Validación de coordenadas

Comunicación I2C

La comunicación I2C trabaja con un Master y un Esclavo en nuestro caso el esclavo es el Arduino Nano al cual se asignó el número 2 para su identificación, posteriormente iniciamos la comunicación I2C a 9600 baudios. Se creó un evento (requestEvent) el cual va a ser ejecutado cada vez que el maestro pida información del esclavo.

```
void setup()
{
  Wire.begin(2);
  Serial.begin(9600);
  Wire.onRequest(requestEvent);
}
```

Figura 35 Comunicación I2C

Envió de trama de datos vía I2C

Para él envió de datos vía protocolo de comunicación I2C se creó un evento requestEvent el cual tiene el mismo principio de las programaciones anteriores que consiste en separar cada dato con el signo de “+”, para saber cuándo se está enviando datos un “;” y para saber cuándo se envió toda la trama de datos una “,” para que se realice el envío de datos vía protocolo de comunicación I2C se debe cumplir con las siguientes condiciones:

Cuando la variable flag este en 1 y la validación de las tres coordenadas sean las correctas, se concatenará todos los datos en la variable datos con la siguiente secuencia:
init0+init+sep+coord[0]+sep+coord[1]+sep+coord[2]+sep+tiempo[0]+sep+tiempo[1]+sep+tiempo[2]+sep+fin.

```
void requestEvent(){
  String sep = "+";
  String init = ";";
  String fin = ",";
  String init0 = ".";

  if (flag == 1){
    if (val1 == 1 && val2 == 1 && val3 == 1){
      datos = init0+init+sep+coord[0]+sep+coord[1]+sep+coord[2]+sep+tiempo[0]+sep+tiempo[1]+sep+tiempo[2]+sep+fin;
      datos.toCharArray(datosE,30);
      Wire.write(datosE);
      Serial.write("V");
    }else Serial.write ("F");
  }
  flag = 0;
  val1 = 0;
  val2 = 0;
  val3 = 0;
}
```

Figura 36 Envío de trama de datos via I2C

La comunicación I2C solo acepta datos de tipo chart, y nuestros datos almacenados en la variable datos son de tipo string, es por eso que, se realiza una conversión de datos tipo string a tipo chart, para el envío de datos I2C, para finalizar con las validaciones se realiza comunicación vía Bluetooth con la aplicación Android, donde se envía una “V” si todo esta validado correctamente mostrando un mensaje de “datos enviados” en la interfaz de la aplicación y una “F” si los datos no

son válidos mostrando un mensaje de “datos incorrectos”. Al final del envió de datos mediante I2C, se resetea las variables bandera y las validaciones de las coordenadas dejándolas en 0.

3.6. Secuencia de bloques de programación del Arduino Robot

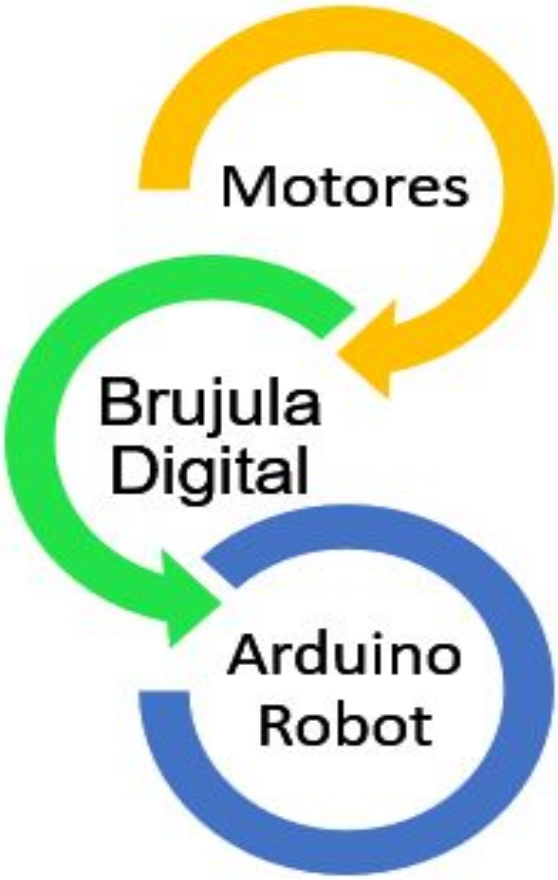

Figura 37 Secuencia de bloques de programación del Arduino Robot

Antes de proceder con la programación del Arduino Robot se realizaron algunas calibraciones de los diferentes componentes que se trabajó como son: los motores dc y la brújula digital HMC6352. Al trabajar con el Arduino Robot siguiendo pasos básicos del uso de cualquier dispositivo Arduino, se conectó la tarjeta en cualquier puerto USB del computador, luego el Arduino Robot permanecerá encendido el led

indicador ON y, posteriormente se realizó las programaciones de cada uno de los componentes usados en el control de movimiento del Arduino Robot.

3.7. Calibración de los motores de Arduino Robot

Se realizaron pruebas con los motores del Arduino Robot, donde se pudo evidenciar que los motores estaban des calibrados, ya que al girar las llantas no podían sincronizarse entre sí, danto diferente velocidad y distintas direcciones. Además, una investigación del uso y funcionamiento de los motores, y se tomó como ejemplo un tutorial oficial de Arduino en YouTube, donde explica los pasos a seguir para calibrar la dirección y velocidad de los motores del Arduino Robot.

Figura 38 Programación motores Arduino Robot

Fuente: (Youtube, 2016)

Para la calibración de los motores se cargó la siguiente programación en la placa de control R06 Wheel Calibration: La calibración del Arduino Robot se logra con la manipulación de los dos potenciómetros, cada uno situado en cada una de las placas del Arduino Robot, uno situado en la placa motor y el otro potenciómetro en la placa

de control. Y la dirección de los motores dependerá de la manipulación del potenciómetro situado en la placa motor, los motores normalmente están des calibrados, el Arduino Robot tomaba su derecha o su izquierda, es por eso que se necesita de la manipulación de del potenciómetro hasta lograr que los motores caminen en línea reta.

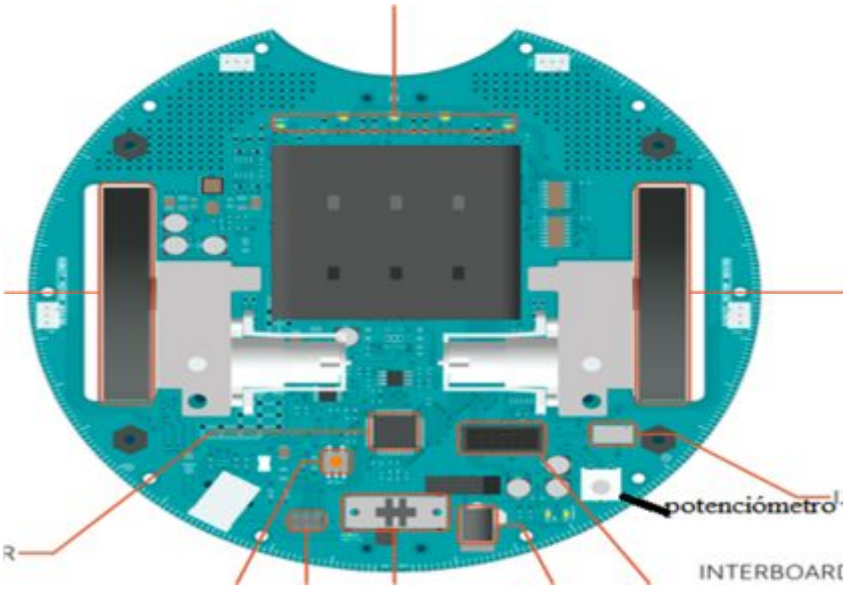

Figura 39 Potenciómetro de Dirección de Motores

Fuente: (Robot, 2016)

Para la calibración de la velocidad de los motores, se debe manipular el segundo potenciómetro, el mismo que se encuentra en la placa control del Arduino Robot, este potenciómetro calibra la velocidad de los motores en proporciones de 0 al 100%, siendo así el 0% la velocidad nula, el 50% la velocidad media y el 100% la velocidad máxima de cada uno de los motores.

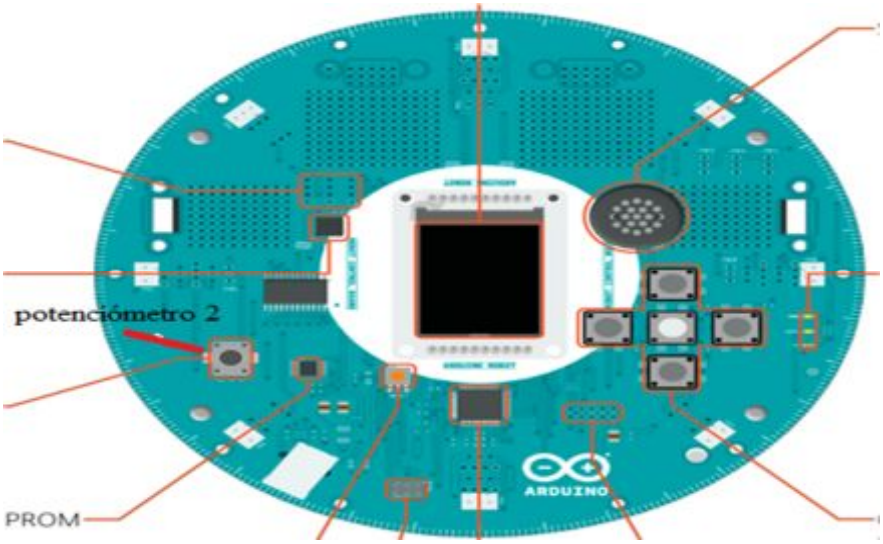

Figura 40 Potenciómetro de Velocidad de Motores

Fuente: (Robot, 2016)

Al final del proceso se realizó la calibración de la dirección y la velocidad de los motores, dejando una velocidad apta para el movimiento del Arduino Robot y una dirección en línea recta. Cabe mencionar que, es de suma importancia calibrar los motores de Arduino Robot, antes de ser usado para evitar problemas en los proyectos futuros.

3.8. Calibración de la brújula digital del Arduino Robot

Para lograr la obtención de datos más exactos censados por la brújula digital del Arduino Robot, Fue de mucha importancia que la brújula sea calibrada, realizando un numero de pasos que a continuación se detallan, todo este proceso se logró mediante una programación de calibración, la misma que se encuentra disponible en la librería de Arduino Oficial.

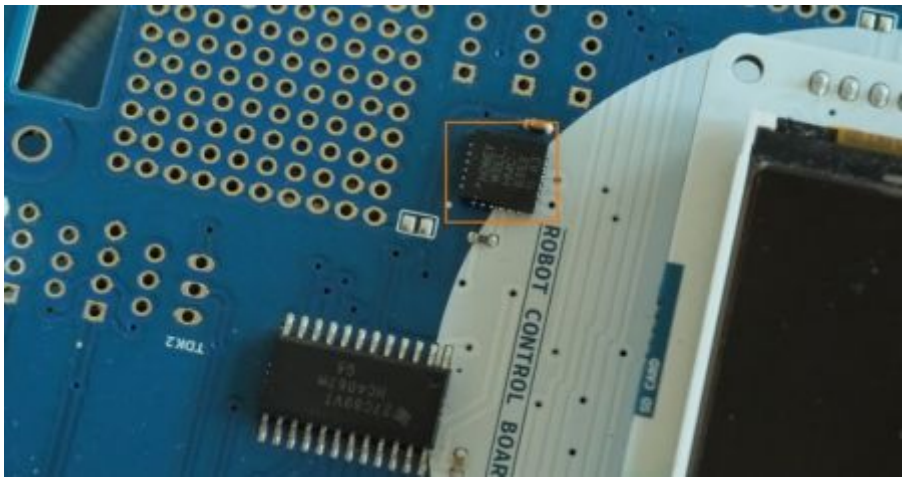

Figura 41 Tarjeta Brújula Digital del Arduino Robot

Fuente: (Arduino B. , 2015)

Para la calibración básicamente se trata de establecer I2C con la brújula, donde se enviará datos a través de una señal de calibración, y luego se envía una señal de fin de calibración. Así mismo, se cargó el programa "Robot_Control > aprendizaje> brújula" al Arduino Robot, posterior a este paso se verá en el display un mensaje de "modo de calibración", después se verá un mensaje "Inicio". Cuando se llegue a este punto, el sensor leerá la posición actual de del Arduino robot y segundos después empezará la auto-calibración para los próximos 15 segundos.

A continuación, se giró lentamente el robot realizando varios círculos alrededor de nuestro perímetro durante 15 segundos que verán, en el dislay del Arduino Robot se verá el tiempo transcurrido durante la calibración, y al finalizar la calibración de la brújula digital, se desplegará un mensaje de "hecho" el display, dejando el módulo de la brújula digital totalmente calibrado y listo para ser usado en proyectos futuros. Posterior al proceso de la calibración de la brújula digital, se recomienda cargar el ejemplo de la librería IDE Arduino, con nombre de "Robot_Control > aprendizaje> brújula" y verificar si las lecturas de la brújula digital son reales y están entre los 360 grados.

3.9. Programación del Arduino Robot

Para el uso y programación de las placas del Arduino Robot se debe ejecutar la aplicación IDE de arduino, que anteriormente fue instalada y descargada de su página oficial, ya en el IDE de arduino al conectar la placa a continuación se presenta una lista de dispositivos de los cuales se debe seleccionar la placa en Herramientas/Tarjetas Arduino Robot.

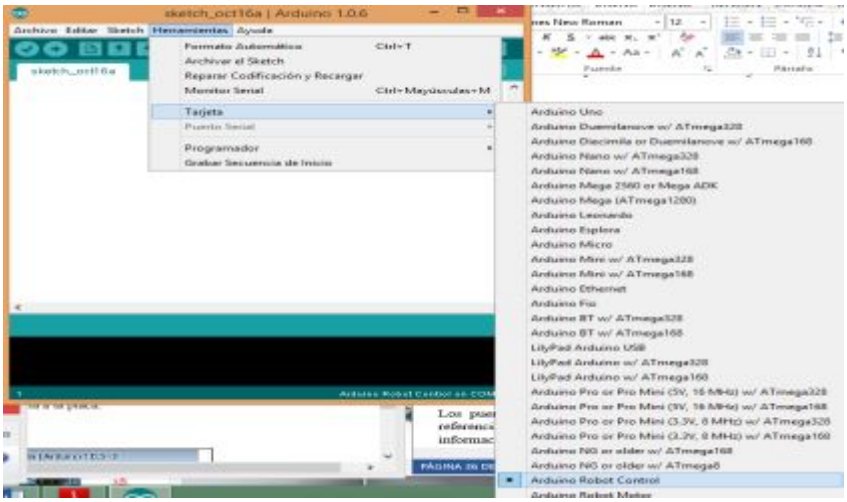

Figura 42 Selección de la tarjeta Arduino Robot

Entre los puntos más importantes de la programación están los siguientes:

- Dato recibido vía I2C
- Ubicación del robot en coordenadas geográficas que se imprime en pantalla
- Trama de datos recibidos vía I2C
- Vector con las coordenadas recibidas
- Vector con los tiempos recibidos
- Lee el valor del potenciómetro
- Bandera que me indica cuando se han recibido los datos vía I2C

Importación de librerías

Para iniciar con la programación e inicialización del robot en necesario incluir las siguientes librerías: #include <ArduinoRobot.h>, #include <Wire.h>, #include <SPI.h>, ya que son base fundamental y obligatoria para lograr un óptimo trabajo, caso contrario sin las librerías mencionadas el Arduino Robot enviará un error de complicación en el sistema.

```
#include <ArduinoRobot.h>
#include <Wire.h>
#include <SPI.h>

char i2cVal;
char direccion[6];
String datosR = "";
String lastDatos = "";
String coordenadas[3];
int index[7];
int tiempo[3];
int cont = 0;
int vel;
int flag = 0;
```

Figura 43 Importación de librerías Arduino Robot

Declaramos variables a ser ejecutadas

En este bloque de programación iniciamos con los siguientes dispositivos a ser usados:

- **Inicio del Robot**: Se usó la variable Robot.begin(), esta función inicia el Arduino Robot preparando todos sus dispositivos para que estos ejecuten la programación almacena internamente en su memoria.
- **Inicio del Display:** Se usó la variable Robot.beginTFT(), esta función inicia el display del Arduino Robot el cual se mostrarán en pantalla las coordenadas, el tiempo y un mensaje para que el usuario edifique que acción que está ejecutando.

- **Inicio del altavoz:** Se usó la variable Robot.beginSpeaker(), esta función permite que el Arduino robot hago uso del altavoz realizando dos pitazos al final de la ejecución de los movimientos enviados por el usuario.
- **Inicio de comunicación I2C:** Se usó la variable Wire.begin(), esta función permite que el robot active el puerto de entrada I2C esperando la comunicación con el Arduino nano para posteriormente recibir la trama de datos.

```
void setup() {
  // Inicializa el robot
  Robot.begin();
  // Inicializa la pantalla del robot
  Robot.beginTFT();
  // Inicializa el sonido del altavoz
  Robot.beginSpeaker();
  // Inicializa la comunicacion I2C
  Wire.begin();
}
```

Figura 44 Inicio de comunicación I2C

El maestro Arduino Robot solicita información del esclavo

Para iniciar la comunicación I2C entre los dispositivos y así solicitar información del esclavo, se creó el siguiente evento “Wire.requestFrom(2,30)”, evento en el cual se asigna un nombre al esclavo, en este caso su nombre es el número 2 y también se solicita el tamaño de 30 bit como dimensión para la capacidad de recepción de toda la trama de datos I2C.

```
Wire.requestFrom(2,30);

while (Wire.available()){
  int last = 0;
```

Figura 45 Solicitud de información maestro, esclavo

A continuación, se detalla una imagen que muestra la impresión de tres mensajes en el display del arduino robot, donde se elige el color de texto usando el código RGB en nuestro caso se asignará el color negro que corresponde a Robot.stroke(0,0,0); a continuación, se imprimirán en el display los siguientes mensajes “Estoy listo”, “Velocidad:”, “Coordenadas”.

```
Robot.stroke(0,0,0);
Robot.text("Estoy listo!",5,5);
Robot.text("Velocidad:",5,25);
Robot.text("Coordenadas:",5,55);
```

Figura 46 Impresión de mensajes en display Arduino Robot.

Recepción de datos vía I2C

En la comunicación I2C por defecto siempre se está recibiendo datos, es por eso que nuevamente se creó artificios para identificar cuando empieza y cuando termina la trama de datos recibida ya que se recibió datos basura como por ejemplo “y”. La programación está diseñada de la siguiente forma, para identificar cuando empieza la trama de datos se recibe una “.”, que separa cada dato recibido, mientras no se reciba una “,” cada dato se ira concatenando en la variable datosR.

```
i2cVal = Wire.read();
if (i2cVal == '.'){
  while(i2cVal != ','){
    i2cVal = Wire.read();
    datosR += i2cVal;
  }
```

Figura 47 Concatenación de datos recibidos

Impresión de velocidad y dirección en pantalla

La impresión en la pantalla del Arduino Robot se logra con la función PrintDir donde el robot por defecto se ubicará en la posición N80E y se imprimirán las siguientes variables:

- Void PrintDir: Esta variable cumple la función de imprimir la velocidad que será leída del potenciómetro en la tarjeta de control y la posición del Arduino Robot en coordenadas geográficas mediante el uso de la brújula digital.
- Vel = map Robot.knobRead: Esta variable se usó para la lectura del potenciómetro que varía de 0 a 1023 y se codifico para que varié de 0 a 100 obteniendo así el 0% velocidad nula, el 50% velocidad media y 100% velocidad máxima, y con la fusión Robot.debugPrint(vel,5,35) se imprime la velocidad leída por el potenciómetro.
- Robot.compassRead: Esta variable se usó para la lectura del ángulo que emite la brújula digital. La información de la brújula es tipo Dir en Arduino Robot solo imprime información de tipo Char, es por ese q se realiza una conversión de tipo Dir a tipo Chart.

Figura 48 Display Arduino Robot con mensajes de programación

Los ángulos censados e impresos por la brújula varían cada milisegundo y constante mente sobrescribiendo la impresión en el display es por eso que gracias a la función Robot.fill(255,255,255) creamos un rectángulo de color blanco que se imprime cada 500 milisegundos, para evitar que se sobrescriba el ángulo emitido por la brújula y logran una impresión fija de ángulos.

```
void PrintDir(){
  String dir = AngToDir(Robot.compassRead());
  vel = map(Robot.knobRead(),0,1023,0,100);
  dir.toCharArray(direccion,6);
  Robot.fill(255,255,255);
  Robot.stroke(255,255,255);
  Robot.rect(0,65,128,10);
  Robot.stroke(0,0,0);
  Robot.text(direccion,5,65);
  Robot.debugPrint(vel,5,35);
  delay(500);
}
```

Figura 49 Impresión de velocidad y dirección en pantalla

Transformación del ángulo a coordenadas geográficas

Para la codificación e impresión de las coordenada geográficas en la pantalla del Arduino Robot se diseñó una función que a partir de la lectura del ángulo que emite la brújula digital, se trasforme a coordenadas geográficas, todo este proceso se logra usando la función String AngToDir(int angle), a continuación de detallará el proceso de trasformación e impresión:

Manteniendo la siguiente lógica se creó cuatro String: String Str1="N", String Str2="S", String Str3="E", String Str4="O", que representa a cada uno de los puntos cardinales. La impresión de las coordenadas en la pantalla del Arduino robot dependerá del ángulo emitido por la brújula, si el ángulo leído por la brújula es igual a 0 se imprimirá la palabra “Norte”, si el ángulo esta entre 0 y 90, se imprimirá N (el ángulo) E.

```
String AngToDir(int angle){
  String Str1="N";
  String Str2="S";
  String Str3="E";
  String Str4="O";
  if (angle == 0){
    return "Norte";
  } else
  if (angle > 0 && angle < 90){
    return Str1 + angle + Str3;
  } else
  if (angle == 90){
    return "Este";
  } else
  if (angle > 90 && angle < 180){
    return Str2+(180-angle)+Str3;
  } else
  if (angle == 180){
    return "Sur";
  } else
  if (angle > 180 && angle < 270){
    return Str2+(angle-180)+Str4;
  } else
  if (angle == 270){
    return "Oeste";
  } else
  if (angle > 270 && angle < 360){
    return Str1+(360-angle)+Str4;
  }
```

Figura 50 Trasformación del ángulo a coordenada geográfica

A continuación, se detalla la lógica de impresión de coordenadas geográficas:

- Si el ángulo leído por la brújula es igual a 90 se imprimirá la palabra “Este”, si el ángulo esta entre 90 y 180, se imprimirá S (180- ángulo) E.
- Si el ángulo leído de la brújula es igual a 180 se retorna la palabra “Sur”, si el ángulo esta entre 180 y 270, se imprimirá S (ángulo-180) O.
- Si el ángulo leído de la brújula es igual a 270 se retorna la palabra “Oeste”, si el ángulo esta entre 270 y 360, se imprimirá N (360- ángulo) O.

Codificación de datos recibidos

La codificación de datos es básicamente el mismo principio de la extracción de datos de las anteriores programaciones, donde se identifica el comienzo y el final de la trama de datos reciba, un “;” para saber dónde inicia y una “,” para saber dónde termina la trama de datos:

- Con la función index [i] se logra que la programación busque la posición del carácter “+” en el índice del vector de la trama de datos recibidos.
- Con la función coordenadas [j-1] se logra extraer las coordenadas de la trama de datos recibidos y postreramente se almacena en la variable datos.substring.
- Con la función tiempo [k-4] se logra extraer los tiempos de la trama de datos recibida y posteriormente se los almacena en la variable datos.substring.
- Al final de todo el proceso de extracción y codificación de datos, se pone la bandera en 1y se llama a la función void Execute la cual se encargará de ejecutar los movimientos y tiempos del robot.

```
int init = datosR.lastIndexOf(';');
int fin = datosR.lastIndexOf(',');
String datos = datosR.substring(init,fin);

if (lastDatos == datos){
  cont = cont + 1;
} else cont = 0;
if (cont == 1){

  for(int i = 0; i < 7; i++){
    index[i] = datos.indexOf('+',last+1);
    last = index[i];
  }
  for(int j = 1; j < 4; j++){
    coordenadas[j-1] = datos.substring(index[j-1]+1,index[j]);
  }
  for(int k = 4; k < 7; k++){
    tiempo[k-4] = datos.substring(index[k-1]+1,index[k]).toInt();
  }
   flag = 1;
}
```

Figura 51 Codificación de datos recibidos

Para lograr que el Arduino Robot ejecute los movimientos se deben realizar las siguientes funciones:

- Se llama a la función position para saber hacia dónde debe girar el robot según el ángulo leído por la brújula, los movimientos del Arduino Robot se basan en la función alpha que es el ángulo ingresado por el usuario que me envía la aplicación Android.
- Se imprimirá en la pantalla las nuevas coordenadas y los nuevos tiempos donde se encuentre el Arduino Robot, se indica al motor la velocidad con la cual debe activar los motores dependiendo el valor leído por el potenciómetro que se encuentra en la placa de control.
- Con la función Robot.motorsWrite se activa los motores durante el tiempo ingresado en la aplicación Android, tiempo que es trasformado de segundos a milisegundos.
- Al final de la ejecución de todos los movimientos, con la función Robot.beep se crea dos sonidos, posteriormente se imprime la palabra “Hecho”, se realiza un delay de 500 milisegundos y se resetea los valores en la memoria dejando impresos los nuevos datos en pantalla.

```
void Execute(){
    Robot.text("Ejecutando...",5,85);
    for(int i = 0; i < 3; i++){
      PrintDir();
      int alpha = CoordToAng(coordenadas[i]);
      int beta = Position(Robot.compassRead(),alpha);
      Robot.turn(beta);
      PrintDir();
      int velocidad = map(vel,0,100,0,255);
      Robot.motorsWrite(velocidad,velocidad);
      delay(1000*tiempo[i]);
      Robot.motorsStop();
      PrintDir();
      if(i == 2)
        break;
      delay(1000);
    }
    Robot.beep(BEEP_DOUBLE);
    Robot.stroke(0,0,0);
    Robot.text("Hecho!",5,105);
    delay(1500);
    Robot.fill(255,255,255);
    Robot.stroke(255,255,255);
    Robot.rect(0,0,128,120);
```

Figura 52 Arduino Robot ejecutando las acciones

CAPÍTULO IV

ANÁLISIS DE RESULTADOS

4.1. Análisis de resultados

A continuación, se presenta el análisis de resultados, con el objetivo de comprobar el funcionamiento del sistema de control de movimiento de un Arduino robot desde un Smartphone con sistema Android empleando una brújula digital, mediante la ejecución de pruebas para determinar la eficiencia del sistema. Para ello, se utilizó los siguientes criterios cualitativos (Tabla 3) para lograr realizar una evaluación al sistema de control de movimiento de un Arduino Robot desde un Smartphone con sistema Android, empleando una brújula.

Tabla3
Criterios cualitativos

Ponderación	Parámetros
3	**Excelente al cumplir el objetivo**
2	**Bueno al efectuar el objetivo**
1	**Deficiente desempeño del objetivo**
0	**No verifica el objetivo**

Las pruebas que fueron evaluadas son los procesos realizados en el trascurso de proyecto, siendo estos siete bloques fundamentales, los cuales se resumen en la siguiente tabla con sus respectivas ponderaciones:

Tabla4
Funciones evaluadas

Numero	Funciones evaluadas	Ponderación
1	**Funcionamiento y diseño de la aplicación Android en AppInventor.**	**3**
2	**La implementación del módulo Bluetooth HC-06 para la trasferencia de datos vía serial al arduino nano.**	**3**
3	**La implementación de un arduino nano para el envió de datos mediante I2C al arduino robot.**	**3**
4	**Funcionamiento de los motores del arduino robot.**	**3**
5	**Funcionamiento de las ruedas del arduino robot**	**2**
6	**Funcionamiento de la brújula digital HMC 6352 del arduino robot.**	**2.8**
7	**Duración de las baterías del arduino robot**	**2.5**
8	**Sistema de control de movimiento de un Arduino Robot desde un Smartphone con sistema Android, empleando una brújula.**	**2.8**

4.2. Análisis funcional de la App Android

Mediante las pruebas realizadas se logró constatar que la aplicación fluye con normalidad en cualquier Smartphone y puede ser instalada sin importar la versión Android del mismo, el diseño de botones, campos de ingreso de datos, instrucciones en la interfaz pudo ser interpretada fácilmente por el usuario. Concluyendo que el funcionamiento de la aplicación cumple satisfactoriamente con lo requerido para él envió de datos desde el Smartphone.

4.3. Análisis módulo Bluetooth HC-06

El módulo bluetooth HC-06 cumplió satisfactoriamente con la función de vinculación con el Smartphone, recibir los datos del mismo y enviarlos al arduino nano, el modulo bluetooth pudo grabar en su memoria la vinculación de los diferentes dispositivos celulares, pudo validad la contraseña y trabajar con normalidad durante todo el proceso de control.

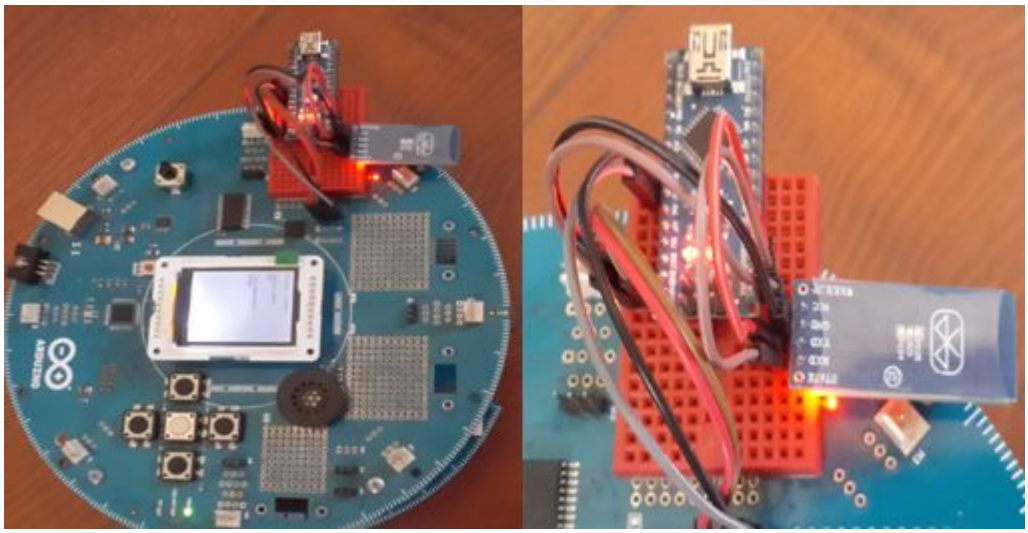

Figura 53 Análisis del módulo Bluetooth HC-06

4.4. Análisis del Arduino nano

El Arduino robot no pudo recibir la comunicación serial del módulo bluetooth debido a que todos sus pines son multiplexados, es decir todos están conectados a una misma línea. A su vez el arduino robot no cuenta con pines asignados para la trasmisión y recepción de datos, debido que, estos pines están siendo usados en la comunicación entre la placa de control y placa motora.

Es por eso que se tuvo la necesidad y se implementó un arduino nano el cual cumplió satisfactoriamente la función de validad la información de coordenadas geográficas y tiempos enviados por la aplicación mediante el modulo bluetooth y posteriormente realizar una trasformación de comunicación serial a protocolo I2C para ser enviada al arduino robot.

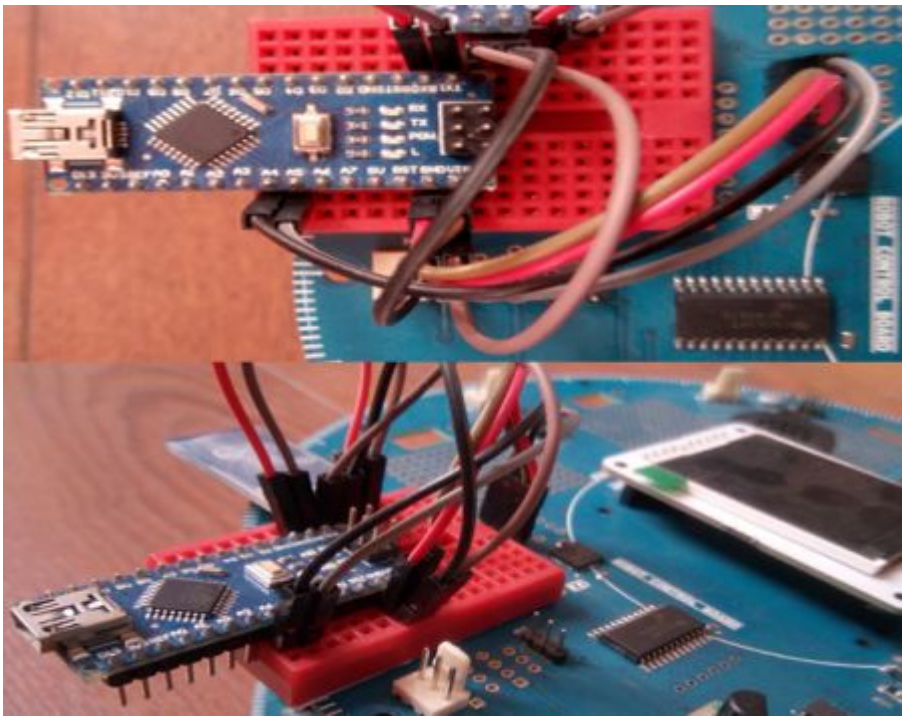

Figura 54 Análisis del Arduino nano

4.5. Análisis motores arduino robot

Se realizaron las distintas pruebas y se pudo evidenciar que los motores del arduino robot cumplen con el funcionamiento en los tiempos establecidos por el usuario en la aplicación; además, se comprobó el funcionamiento de los dos potenciómetros que regulan la velocidad y la dirección de los motores requeridos por el usuario, teniendo así a los dos motores aptos para la movilización del Arduino robot.

4.6. Análisis ruedas arduino robot

Mediante varias pruebas realizadas se constató que las llantas del arduino robot no se adhieren a los diferentes suelos del entorno, dando un porcentaje de error de 8 a 12 grados en su posición final, es por eso que se implementó una lija con pegatinas en el contorno de las llantas, de esta manera se disminuyó el margen de error a 2 o 3 grados en su posición final.

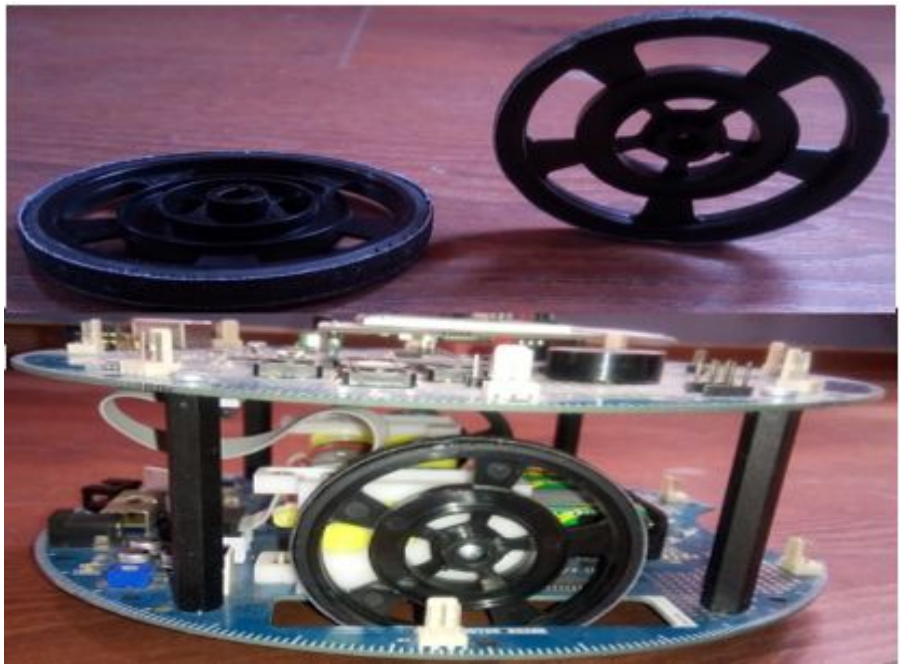

Figura 55 Análisis de las ruedas del arduino robot

4.7. Análisis brújula digital HMC 6352

Las pruebas realizadas mediante el funcionamiento de la brújula digital cumplen con parámetros óptimos para el desarrollo del presente proyecto, ya que se obtuvo una buena precisión al censar el norte magnético, se puso a prueba usando una brújula digital donde se obtuvo los mismos grados resultantes en ambas brújulas. Es factible mencionar que, los grados censados por la brújula digital pueden estar sujetos a pequeños cambios, esto dependerá del estado climático y el campo magnético del lugar.

4.8. Análisis pilas AAA arduino robot

Por medio del proceso de pruebas y funcionamiento las pilas AAA fueron sometidas a un régimen de operación continuo donde se pudo constatar que no brindan un tiempo prologando de duración, ya que las pilas alimentan de energía a

diferentes accesorios como son; el arduino robot, arduino nano y modulo bluetooth limitando a dos horas de funcionamiento.

4.9. Análisis del control de movimiento

Se realizó un análisis de funcionamiento al proyecto final, donde se demostró que el proyecto cumple con el objetivo planteado en su estructura, a continuación, se realizó el respectivo uso del proyecto donde se pondrá en práctica cada uno de sus elementos y bloques de programación.

Para la siguiente prueba de funcionamiento enviaremos los siguientes datos: tres coordenadas diferentes tomando en cuenta la orientación geográfica norte, sur, este y oeste; así mismo, se enviará tres tiempos diferentes para que el robot este en movimiento:

- **Coordenadas geográficas:** N45E tiempo: 3seg, S5E tiempo: 6seg, N45O tiempo: 2seg,
- **Velocidad de motores:** 35%

Para concluir la prueba se demostró que el del sistema de control de movimiento de un Arduino Robot desde un Smartphone con sistema Android, empleando una brújula cumple satisfactoriamente las órdenes del usuario como son; coordenadas geográficas, tiempos de funcionamiento de motores y tomando en cuenta el mínimo error en su posición final de 2 a 3 grados.

Figura 56 Análisis general del sistema de control de movimiento

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

Al finalizar la investigación se formulan las siguientes conclusiones:

- Se implementó un control de movimiento de un Arduino Robot empleando una brújula digital, para el análisis del funcionamiento de un módulo Bluetooth, un Arduino Nano y el diseño de una aplicación Android.
- El Arduino Robot no admitió la comunicación serial del módulo Bluetooth; por tal razón, fue necesario usar uno de sus puertos de comunicación I2C, empleando un Arduino Nano.
- El Arduino Robot posterior a la ejecución de los movimientos, está sujeto a un margen de error de 2 a 3 grados en su posicion final, que dependerá de sus llantas, campo electromagnético y condiciones climáticas del lugar.
- El proyecto permitió poner en práctica todos los conocimientos adquiridos en el trascurso de la carrera como: electrónica, sistemas digitales, microcontroladores, programación, redes y telecomunicaciones.

5.2. RECOMENDACIONES

Como resultado de todo el proceso investigativo, es necesario hacer las siguientes recomendaciones:

- Considerar el estudio de la comunicación, funcionamiento y compatibilidad de los sensores antes de ser usados en el Arduino Robot.
- Continuar con las investigaciones referentes al tema para el desarrollo de nuevas y diversas aplicaciones con el Arduino Robot.
- Implementar un nuevo sistema de alimentacion para el Arduino Robot, ya que sus pilas AA limitan su uso a dos horas de funcionamiento continuo.
- Aplicar los voltajes adecuados para cada uno de los dispositivos asegurando la fiabilidad y vida útil de los mismos.

REFERENCIAS BIBLIOGRÁFICAS

Appinventor. (2016). *http://appinventorcaracterisiticas.blogspot.com*. Obtenido de http://appinventorcaracterisiticas.blogspot.com/2016/05/que-es-appinventor-inventor-parte-de.html

Arduino. (2015). *Productos de Arduino*. Obtenido de http://www.arduino.cc/en/Main/Products

Arduino, B. (2015). *Blogs Arduino*. Obtenido de http://arduinoteco.blogspot.com/2013_12_12_archive.html

Arduino, N. (2015). *www.arduino.cc*. Obtenido de https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf

Arduino, O. (2015). *Oficial Arduino movieminto cosas.* Obtenido de http://playground.arduino.cc/Es/OSW042

Argentina, R. (2014). *www.robots-argentina.com.ar*. Obtenido de http://robots-argentina.com.ar/Comunicacion_busI2C.htm

ATMEGA32U4. (2016). *ATMEGA32U4*. Obtenido de ATMEGA32U4: http://es.rs-online.com/web/p/microcontroladores/7153805P/

Bluetooth-Serial-HC-06. (2016). *https://www.olimex.com*. Obtenido de https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf

Datasheet, A.-3. (2014). *www.atmel.com*. Obtenido de http://www.atmel.com/Images/Atmel-7766-8-bit-AVR-ATmega16U4-32U4_Datasheet.pdf

HMC6352., d. (2015). *www.sparkfun.com.* Obtenido de https://www.sparkfun.com/datasheets/Components/HMC6352.pdf

Robot, A. (2016). *Productos de Arduino*. Obtenido de https://www.arduino.cc/en/Main/Robot

ANEXOS