



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE TECNOLOGÍAS

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CARRERA:

ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y AVIÓNICA.

TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

TECNÓLOGO EN ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y AVIÓNICA.

TEMA:

“IMPLEMENTACIÓN DE UN CONTROL DE UN ARDUINO ROBOT MEDIANTE MÓDULOS DE RADIO FRECUENCIA Y LABVIEW PARA EL MONITOREO DE SENSORES DE TEMPERATURA Y HUMEDAD.”

AUTOR: TACO PAUCAR JORGE LUIS

DIRECTORA: ING. MORETA CHANGOLUIZA JANNETH ELIZABETH

LATACUNGA

2019



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE TECNOLOGÍAS

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

CERTIFICADO

Yo, **ING. MORETA CHANGOLUIZA JANNETH ELIZABETH** certifico que el trabajo titulado TEMA: “IMPLEMENTACIÓN DE UN CONTROL DE UN ARDUINO ROBOT MEDIANTE MÓDULOS DE RADIO FRECUENCIA Y LABVIEW PARA EL MONITOREO DE SENSORES DE TEMPERATURA Y HUMEDAD.” Realizado por Jorge Luis Taco Paucar, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la Universidad de las Fuerzas Armadas – ESPE.

Debido a que se trata de un trabajo de investigación recomiendo su publicación, el mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf).

ING. MORETA CHAMGOLUIZA JANNETH ELIZABETH
DIRECTORA DEL PROYECTO

Latacunga, Febrero 2019.



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE TECNOLOGÍAS

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

AUTORÍA DE RESPONSABILIDAD

Yo, TACO PAUCAR JORGE LUIS, con cedula N° 1723014633 declaro bajo juramento que el trabajo de titulación “IMPLEMENTACIÓN DE UN CONTROL DE UN ARDUINO ROBOT MEDIANTE MÓDULOS DE RADIO FRECUENCIA Y LABVIEW PARA EL MONITOREO DE SENSORES DE TEMPERATURA Y HUMEDAD.” Ha sido previamente desarrollado considerando métodos de investigación de igual manera se ha respetado derechos de autor considerándose en citas bibliográficas.

Seguidamente declaro que este trabajo es de mi autoría, por ello me declaro responsable del contenido de la investigación realizada.

JORGE LUIS TACO PAUCAR
AUTOR DEL PROYECTO
172301463-3

Latacunga, Febrero 2019.



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

UNIDAD DE GESTIÓN DE TECNOLOGÍAS

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

AUTORIZACIÓN

Yo, **TACO PAUCAR JORGE LUIS** autorizo A la Unidad de Gestión de Tecnologías sustentada a la Universidad de las Fuerzas Armadas – ESPE, la publicación en la biblioteca virtual y física de la Institución el trabajo “IMPLEMENTACIÓN DE UN CONTROL DE UN ARDUINO ROBOT MEDIANTE MÓDULOS DE RADIO FRECUENCIA Y LABVIEW PARA EL MONITOREO DE SENSORES DE TEMPERATURA Y HUMEDAD.” cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

JORGE LUIS TACO PAUCAR
AUTOR DEL PROYECTO
172301463-3

Latacunga, Febrero 2019.

DEDICATORIA

El presente trabajo de investigación va dedicado a toda mi familia por haber sido un pilar fundamental en mi vida la cual estuvo llena de grandes esfuerzos y sacrificios por ese motivo quiero dedicarles a mis padres y mis hermanos que día a día estuvieron apoyándome constantemente sin importar la distancia.

A mis padres Luis Taco y Liliana Paucar quienes a pesar de todo lo bueno y malo siempre estuvieron a mi lado dándome consejos y palabras de aliento que siempre las llevo presentes, gracias a ellos me he convertido en un profesional a la confianza que siempre han tenido en mí, ahora he logrado cumplir un sueño anhelado.

A mis hermanos Karen Taco y Sebastián Taco, quienes a pesar de la distancia siempre me apoyaron y me dieron ánimos para seguir adelante.

A mis abuelitos Cecilio Paucar y Soledad Jácome, quienes me vieron crecer desde muy pequeño y nunca me dejaron descarriar por un mal camino gracias a los consejos que me brindaron y siempre me han guiado para seguir adelante.

Jorge Luis Taco Paucar

AGRADECIMIENTO

En primer lugar agradezco a Dios y a la virgen del Quinche por haberme guiado en este camino de buenos y malos momentos, y permitirme lograr una meta más en mi vida para seguir adelante en mi futuro.

A mis padres por haberme apoyado y ser una pieza clave en mi vida ya que son un ejemplo a seguir por su constancia y dedicación, en ayudarme a cumplir esta meta.

A todos los ingenieros de la institución por la dedicación y el tiempo empleado para transmitirme sus conocimientos permitiéndome así finalizar con éxito mi carrera.

Jorge Luis Taco Paucar

ÍNDICE DE CONTENIDO

| | |
|---|-------------|
| CERTIFICADO | ii |
| AUTORÍA DE RESPONSABILIDAD | iii |
| AUTORIZACIÓN | iv |
| DEDICATORIA..... | v |
| AGRADECIMIENTO..... | vi |
| ÍNDICE DE CONTENIDO | vii |
| ÍNDICE DE FIGURAS | x |
| ÍNDICE DE TABLAS | xii |
| RESUMEN..... | xiii |
| ABSTRACT | xiv |
| CAPÍTULO I | 1 |
| 1.1 TEMA | 1 |
| 1.2 ANTECEDENTES..... | 1 |
| 1.3 PLANTEAMIENTO DEL PROBLEMA | 3 |
| 1.4 JUSTIFICACIÓN | 3 |
| 1.5 OBJETIVOS | 4 |
| 1.5.1 OBJETIVO GENERAL..... | 4 |
| 1.5.2 OBJETIVOS ESPECÍFICOS | 4 |
| 1.6 ALCANCE: | 4 |
| CAPÍTULO II | 6 |
| MARCO TEÓRICO | 6 |
| 2.1 TELEMETRÍA..... | 6 |
| 2.1.1 TELEMANDO O TELE-COMANDO | 6 |
| 2.1.2 APLICACIONES..... | 7 |
| 2.1.3 FUNCIONAMIENTO..... | 8 |
| 2.2 ARDUINO ROBOT | 8 |

| | |
|--|-----------|
| 2.2.1 INTRODUCCIÓN | 9 |
| 2.2.2 CARACTERÍSTICAS..... | 11 |
| 2.3 LABVIEW | 12 |
| 2.3.1 VENTAJAS..... | 14 |
| 2.3.2 NI VISA | 15 |
| 2.3.2.1 APLICACIONES VISA..... | 15 |
| 2.4 ARDUINO..... | 16 |
| 2.4.1 ARDUINO NANO | 17 |
| 2.4.1.1 CARACTERÍSTICAS | 17 |
| 2.4.1.2 COMUNICACIÓN..... | 19 |
| 2.4.1.3 DIAGRAMA DE PINES ARDUINO NANO | 20 |
| 2.5 MÓDULOS DE RADIO FRECUENCIA | 20 |
| 2.5.1 APLICACIONES..... | 21 |
| 2.5.2 ESPECTRO ELECTROMAGNÉTICO | 22 |
| 2.5.3 ONDAS DE RADIO | 23 |
| 2.6 SENSORES | 25 |
| 2.6.1 DTH..... | 25 |
| 2.6.1.1 CARACTERÍSTICAS..... | 26 |
| CAPÍTULO III | 28 |
| DESARROLLO DEL TEMA | 28 |
| 3.1 INSTALACIÓN DE SOFTWARE..... | 28 |
| 3.1.1 INSTALACIÓN DE ARDUINO | 28 |
| 3.1.2 INSTALACIÓN DE LABVIEW | 29 |
| 3.2 PROGRAMACIÓN..... | 32 |
| 3.2.1 ARDUINO NANO | 32 |
| 3.2.2 PROGRAMACIÓN ARDUINO ROBOT (CONTROL) | 34 |
| 3.2.3 PROGRAMACIÓN LABVIEW | 37 |
| 3.2.3.1 PANEL FRONTAL..... | 37 |
| 3.2.3.2 DIAGRAMA DE BLOQUES..... | 37 |
| 3.3 MONTAJE | 39 |
| 3.3.1 DISEÑO EN PROTEUS | 39 |

| | |
|--|-----------|
| 3.3.2 CONEXIÓN | 39 |
| 3.3.2.1 CONEXIÓN DEL MÓDULO..... | 39 |
| 3.3.2.2 CONEXIÓN DEL SENSOR DTH11 | 40 |
| 3.3.3.3 CONEXIÓN DEL ARDUINO ROBOT | 40 |
| 3.4 AJUSTE..... | 41 |
| 3.4.1 AJUSTE DEL CONTROL | 41 |
| 3.5 PUESTA EN FUNCIONAMIENTO | 42 |
| 3.5.1 FUNCIONAMIENTO DEL ARDUINO ROBOT | 42 |
| CAPÍTULO IV..... | 45 |
| CONCLUSIONES Y RECOMENDACIONES..... | 45 |
| 4.1 CONCLUSIONES | 45 |
| 4.2 RECOMENDACIONES..... | 45 |
| BIBLIOGRAFÍA..... | 46 |
| ANEXOS | 49 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1 Control de dispositivos | 7 |
| Figura 2 Aplicaciones de telemetría | 8 |
| Figura 3 Arduino Robot | 9 |
| Figura 4 Arduino Robot Motor Control - Vista Inferior..... | 10 |
| Figura 5 Arduino Robot Motor Control -Vista Superior | 10 |
| Figura 6 Arduino Robot - Tableros de control y motor | 11 |
| Figura 7 Comunicación Visa | 15 |
| Figura 8 Programación de Visa..... | 16 |
| Figura 9 Plataforma de creación | 17 |
| Figura 10 Arduino Nano | 19 |
| Figura 11 Nano Pinout | 20 |
| Figura 12 Subrangos de Frecuencia | 23 |
| Figura 13 Sensores..... | 25 |
| Figura 14. Sensores DTH | 26 |
| Figura 15 Arduino 1.8.8..... | 28 |
| Figura 16 Drivers Arduino | 29 |
| Figura 17 Pantalla Principal de Arduino | 29 |
| Figura 18 Labview 2018..... | 30 |
| Figura 19 Archivo de Instalación. | 30 |
| Figura 20 Pantallas Principales..... | 31 |
| Figura 21 NI-VISA 18.0 | 31 |
| Figura 22 Archivo de Instalación NI-VISA. | 32 |
| Figura 23 Programación Arduino Nano. | 34 |
| Figura 24 Programación del Arduino Robot | 36 |
| Figura 25 Panel Frontal..... | 37 |
| Figura 26 Comunicación VISA | 38 |
| Figura 27 Casos para las funciones | 38 |
| Figura 28 Indicadores | 38 |
| Figura 29 Esquema de Conexiones | 39 |
| Figura 30 Conexiones del Módulo 3DR..... | 40 |
| Figura 31 Conexión del Sensor DTH11..... | 40 |
| Figura 32 Conexión del Arduino Robot | 41 |

| | |
|---|----|
| Figura 33 Ajuste de velocidad | 41 |
| Figura 34 Ajuste de Giro | 42 |
| Figura 35 Lectura de Humedad y Temperatura..... | 42 |
| Figura 36 Prueba de Lectura de Humedad y Temperatura | 43 |
| Figura 37 Monitoreo del Arduino Robot hacia adelante..... | 43 |
| Figura 38 Monitoreo del Arduino Robot hacia atrás | 44 |
| Figura 39 Giro del Arduino Robot..... | 44 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1 Bandas de Frecuencia..... | 24 |
| Tabla 2 Características DTH11 | 27 |
| Tabla 3 Características DTH22 | 27 |

RESUMEN

El presente proyecto de titulación fue desarrollado en la Unidad de Gestión y Tecnologías-ESPE, mediante el mismo se implementó un control de un robot arduino para poder monitorear las magnitudes físicas, en este caso se va a monitorear la humedad y la temperatura.

El proyecto consiste en el diseño e implementación de un sistema de control. El objetivo principal con el que se realizó fue, supervisar las magnitudes de lugares con muy poco espacio, en donde se pueda controlar al robot y monitorearlo hasta llegar al punto de visualizar en el control los valores de humedad y temperatura, utilizando la comunicación inalámbrica mediante los módulos 3DR (módulos de radio frecuencia) ya que con los mismos obtendremos señales que son recibidas por el computador, que nos permitirá realizar una interfaz entre hombre y máquina (HMI).

Se estableció una serie de investigaciones para la realización de la programación del robot arduino mediante las cuales realizamos la comunicación entre un Arduino Nano y los Módulos 3dr.

Se diseñó un panel de control en el software de Labview el cual utilizaremos como comunicación bidireccional ya que mediante el software podemos enviar señales al robot arduino para su movimiento, y de igual manera recibimos señales del robot arduino por medio de indicadores en Labview.

Palabras Claves:

- Arduino Robot
- Comunicación Bidireccional.
- Módulos 3DR
- Telemetría.

ABSTRACT

The present research project was developed in the Technology management Unit-ESPE through a control system of an arduino robot that was implemented to monitor physical magnitudes, in this case the robot is going to monitors the humidity and temperature.

The project consists in the design and the implementation of a control system. The main objective was made with the purpose of supervising the magnitudes of places with very little space, where is possible to control the robot, and to monitor it until arriving at the point to visualize in the indicator system the values of humidity and temperature using the wireless communication by means of the modules 3DR (modules of radio frequency) signals will be obtained and will be acquired by the computer, and at the end it will be possible to have an interface between man and machine (HMI).

It was established a series of investigations for the accomplishment of the programming of the arduino robot by the communication between an Arduino Nano and the Modules 3dr.

A control panel was designed in the Labview software which researchers will use as bidirectional communication since through the software it's possible to send signals to the arduino robot for its movement and in the same way it is possible to receive signals from the arduino robot by means of indicators in Labview.

Keywords:

- Arduino Robot
- Bidirectional Comunicación.
- 3DR Módulos
Telemetry.

Checked by:

Lcdo. Flavio Hurtado
DOCENTE

CAPÍTULO I

1.1 TEMA

Implementación de un control de un arduino robot mediante módulos de radio frecuencia y labview para el monitoreo de sensores de temperatura y humedad.

1.2 ANTECEDENTES

A medida que van pasando los años la tecnología crece el desarrollo de herramientas y productos tecnológicos, estos permitieron innovar nuevos sistemas de control, sistemas automatizados y una gran variedad de aplicaciones.

La telemetría es una tecnología que permite la medición remota de magnitudes físicas y el envío de la información hacia el operador del sistema.

Mediante la telemetría se han realizado trabajos como los que se exponen a continuación:

El primer trabajo revisado en revisado en la Universidad de las Fuerzas Armadas ESPE pertenece al autor (Reisancho Casco, 2016) "Diseño e implementación del sistema de telemetría para la tricicleta solar que participará en la carrera Atacama Solar 2014. Carrera de Ingeniería en Electrónica e Instrumentación".

El propósito del proyecto, es diseñar e implementar el sistema de telemetría para la tricicleta solar, en base a las especificaciones de la competencia Atacama Solar Challenger categoría "La Ruta Solar". La principal misión del proyecto es brindar el monitoreo de las magnitudes físicas de la tricicleta y tomar acciones correctivas a tiempo en caso de ser necesario. El documento inicia con una introducción a la reseña histórica de la carrera sus

categorías y versiones además de una redacción de los parámetros de diseño en cuanto a la estructura, sistema de transmisión y frenos, sistema fotovoltaico y el sistema eléctrico. El segundo capítulo realiza un estudio a profundidad sobre los elementos del sistema telemétrico como son: tipos de variables, sensores, microcontroladores, medios de comunicación y la interfaz de visualización de las variables. El tercer capítulo se describe detalladamente los parámetros de selección de los componentes a utilizar, con cuadros comparativos para las siguientes variables: voltaje de los paneles solares, el estado de las baterías, el consumo de corriente del motor, la velocidad del vehículo, y la temperatura de trabajo de las celdas fotovoltaicas. Seguidamente en el capítulo cuatro se procede a realizar las pruebas de funcionamiento del sistema telemétrico tomando datos en pleno movimiento de la tricicleta y comparándolas con sensores de referencia llegando a concluir que los elementos seleccionados han sido los adecuados para la construcción de la "Tricicleta Solar" la cual tiene como destino competir en la carrera ya mencionada representando a la Universidad de las Fuerzas Armadas ESPE Extensión Latacunga.

El segundo trabajo revisado en la Universidad Israel pertenece a los autores (Andrade, Parco, & Gustavo, 2013). "El estudio, diseño e implementación de un Sistema de Modulación Digital de Banda Ancha de Telemetría para el monitoreo y control de variables físicas". Eléctrico, temperatura, voz, video y detección de presencia dentro de la caseta de transmisión de la empresa Watching Soluciones Tecnológicas en la ciudad de Quito, se realizó con el fin de garantizar la continuidad del servicio de radio comunicaciones a los usuarios del sistema (cooperativa de taxis) y brindan mayor seguridad a los equipos de transmisión alojados en la caseta de Lumbisi (Cumbaya).

El monitoreo de las variables físicas monitoreadas, requiere la utilización de sistemas electrónicos que garanticen el funcionamiento de los equipos en tiempo real, para lo cual las comunicaciones inalámbricas juegan un papel importante en el diseño e implementación un sistema de monitoreo y control remoto.

1.3 PLANTEAMIENTO DEL PROBLEMA

Actualmente en la Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas “ESPE”. Se oferta la Carrera de Electrónica mención Instrumentación y Aviónica cuenta con laboratorios de equipos y módulos de últimas generaciones tecnológicas.

Sin embargo, se ha evidenciado que existe una cantidad mínima de equipos de telemetría los cuales no son suficientes para el aprendizaje de los estudiantes, para formar profesionales es necesario que los estudiantes conozcan y manipulen las nuevas tecnologías que existen en el mercado con un fin de mejorar y obtener una mayor cantidad de conocimientos y habilidades.

La implementación de un control de un arduino robot permitirá a los estudiantes de la carrera fortalecer el aprendizaje en el área de electrónica, robótica e instrumentación en los campos de aplicación según la necesidad.

1.4 JUSTIFICACIÓN

El desarrollo de este proyecto tiene como propósito alcanzar conocimientos de acuerdo a la implementación y el control de un arduino robot de una manera prioritaria dentro del desarrollo tecnológico, la robótica aplica conocimientos del área de electrónica e instrumentación ampliando el campo de aplicación de sistemas y prototipos. El arduino robot contará con una interfaz por módulos 3DR de radio frecuencia los cuales permitirán monitorear los sensores de temperatura y humedad mediante LabVIEW.

Al implementar el “Arduino Robot” se beneficiara la carrera de Electrónica Mención Instrumentación y Aviónica ya que sus estudiantes obtendrán nuevos conocimientos teóricos y prácticos de una nueva tecnología la cual van a poder manipular y así puedan desempeñarse de mejor manera en el campo laboral como excelentes profesionales.

Es factible implementar el proyecto del Arduino Robot ya que si es cierto se retiró del mercado de Arduino pero se lo puede encontrar en Ecuador en la ciudad de Cuenca, el mismo se lo utiliza como una computadora con ruedas.

1.5 OBJETIVOS

1.5.1 OBJETIVO GENERAL

Implementar un control de un arduino robot mediante módulos de radio frecuencia y labview para el monitoreo de sensores de temperatura y humedad.

1.5.2 OBJETIVOS ESPECÍFICOS

- Investigar las características técnicas, funcionalidad y la conexión del arduino robot y el módulo 3DR, para establecer los modos de comunicación.
- Establecer la comunicación del Arduino Robot y el usuario a través de un módulo 3DR de radio frecuencia para el monitoreo del robot y sus sensores.
- Implementar un HMI en labview para la fácil e intuitiva interacción con el usuario.

1.6 ALCANCE:

Este proyecto tiene como finalidad la implementación del telecontrol de un arduino robot, va dirigido a la Unidad de Gestión de Tecnologías especialmente a los alumnos y docentes de la Carrera de Electrónica mención

Instrumentación y Aviónica, con fines académicos para adquirir una mayor cantidad de conocimientos de robótica, programación e instrumentación así como también la manipulación de los diferentes componentes y materiales electrónicos que se utilizaran en la realización de este proyecto.

CAPÍTULO II

MARCO TEÓRICO

2.1 TELEMETRÍA

La telemetría generalmente significa “medición a distancia”, la cual nos permite medir, monitorear, rastrear magnitudes físicas y químicas de forma automática a través de sensores los cuales transmiten datos tanto de temperatura, humedad, presión, etc. Estos datos llegan a un punto de control de manera inalámbrica (redes de celulares, redes de computadoras, ondas de radio) para su respectiva visualización y análisis para comprobar un óptimo funcionamiento.

Según (Escalera, 2015) La Telemetría es una técnica de las comunicaciones cuya principal funcionalidad es aportar en las mediciones y recopilación de datos que se realizan en lugares alejados, para luego transmitir los datos para la vigilancia y análisis de los mismos. La técnica se caracteriza por ser de transmisión inalámbrica, aunque existen sistemas de transmisión utilizados por cable

2.1.1 TELEMANDO O TELE-COMANDO

El telemando es el dispositivo de control donde se realiza el monitoreo de comandos que son enviados desde el punto de control estos son de acorde a su respectivo funcionamiento dando órdenes de activar o desactivar equipos las mismas que provienen de los datos de la telemetría .

Se realiza una estructura (control) para manipulaciones a distancia Fig1. De algunos elementos del canal como instrumentos que nos reflejen 1 o 0 (abierto o cerrado). Para cumplir con este proceso se necesita de elementos o dispositivos como relés, pulsadores y contactos.

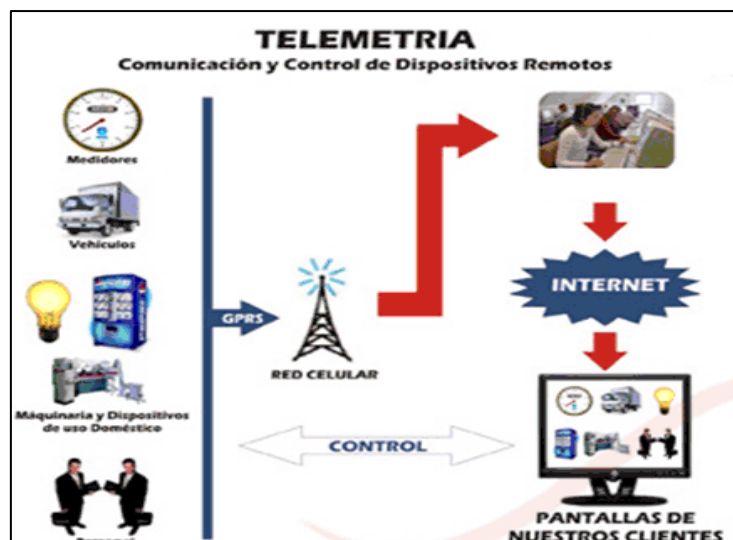


Figura 1 Control de dispositivos

Fuente: (Castro, 2014)

2.1.2 APLICACIONES

Según (Escalera, 2015) Los inicios del estudio y aplicación de la telemetría empezaron en el siglo 20 para ser utilizado de supervisión de la naturaleza, supervisando la distribución de energía eléctrica. La evolución continuó en Chicago en 1912, aplicando conocimientos de telemetría en un centro de vigilancia de líneas telefónicas para recibir los datos operativos de plantas de energía que se encontraban a distancias alejadas. La aplicación de la telemetría en la aeronáutica y recae en la década de 1930, cuando se recopila datos sobre las condiciones atmosféricas utilizando un globo con equipamiento electrónico. Ésta tecnología evolucionó encontrando una gran aplicación en la manipulación satelital para varias aplicaciones que incluye el registro por las condiciones meteorológicas, la observación de fenómenos espaciales y teledetección. Es así que de este modo la telemetría continuó evolucionando hasta entrega su aporte a la biomedicina, en la que los datos importantes sobre los órganos internos de un paciente es transmitida por los dispositivos que se implantan quirúrgicamente dentro de ese órgano.

En la Figura 2. Se muestra como se obtiene registro de los datos para la supervisión.

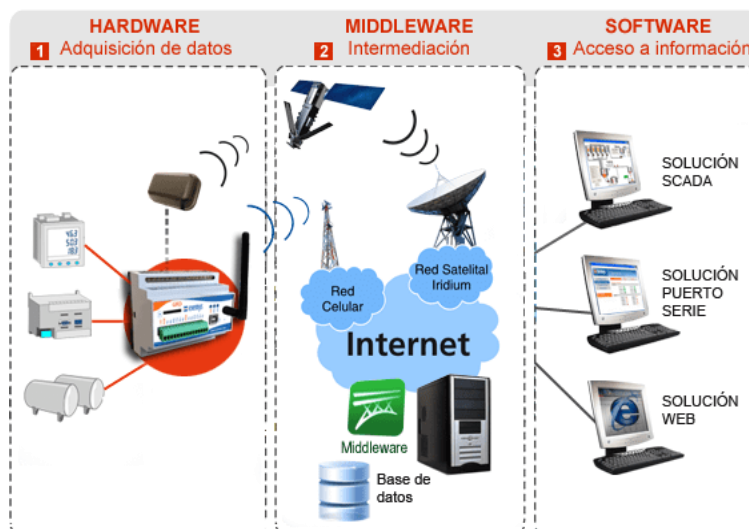


Figura 2 Aplicaciones de telemetría

Fuente: (Escalera, 2015)

2.1.3 FUNCIONAMIENTO

Según (Escalera, 2015) Un sistema de telemetría generalmente está formado por un transductor como un dispositivo de adquisición de datos, un medio de transmisión en forma de líneas de cable o las ondas de radio, dispositivos que procesan las señales, y de visualización de datos. El transductor convierte una magnitud física como la temperatura, presión o vibraciones en una señal eléctrica correspondiente, que es transmitida a una distancia a efectos de medición y registro.

2.2 ARDUINO ROBOT

Según (Pastor, 2013) El Arduino Robot ha sido diseñado en colaboración con Complubot, la Asociación de Robótica Educativa con sede en Madrid que ha logrado proclamarse cuatro veces ganadora de la competición de fútbol robótico Robocup Junior. El resultado del proyecto llamado Lottie Lemon es un sistema perfecto para iniciarse en el mundo de la robótica. Por un lado

tenemos la placa de control, que recibe la respuesta de los sensores y decide cómo operar, y por otro la placa motora, que controla el motor que imprime el movimiento a las ruedas. Cada placa es un sistema Arduino completo en sí misma, y ambas se pueden programar usando el IDE de Arduino.

En la Figura 3. Observaremos la estructura del arduino robot.

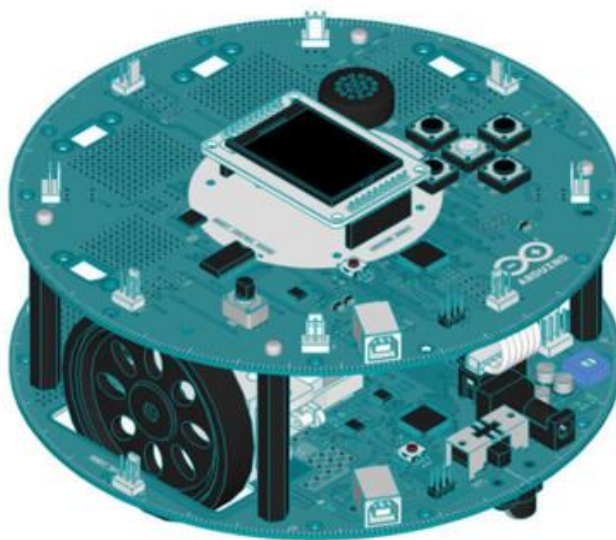


Figura 3 Arduino Robot

Fuente: (Pastor, 2013)

2.2.1 INTRODUCCIÓN

Según (Costachioiu, 2015) Si muchos de mis propios robots construidos no superan el tamaño de un CD, los Arduino Robots son como un elefante Figura 6. Los robots PCB también actúan como chasis, con la placa inferior que aloja los motores, las baterías, los sensores de línea y un microcontrolador ATMEGA32U4 que actúa como control del motor Figura 5. Además de esto, también hay un cargador de batería, el interruptor de encendido y apagado y cuatro conectores TinkerKit. Esa es una elección extraña, ya que el rango TinkerKit no es excelente, y particularmente no hay sensores de distancia disponibles.

El conector USB está cubierto por un enchufe negro. Normalmente, el

robot funciona programando solo el tablero de control, dejando que el tablero del motor ejecute su programa original. También puede optar por utilizar solo el tablero de control del motor, usar su microcontrolador como cerebro maestro y desconectar el tablero superior. Ya estoy pensando en un proyecto de seguidor de líneas que utiliza este enfoque.

En la parte posterior de la tabla inferior Figura 4. Hay un cartón protector, que ya sufrí algunos daños menores mientras realizaba las pruebas iniciales. Con esta cubierta protectora instalada, la distancia al suelo es de 1 cm, menos de 0.4 ". Sí, funcionará bien en superficies lisas, pero no manejará ningún terreno accidentado. La extracción de esta cubierta protectora ganará otros 3 mm de distancia al suelo. Sin embargo, descubrí que esto también aumenta la posibilidad de que el robot sufra descargas estáticas, especialmente cuando se ejecuta en una alfombra sintética.



Figura 4 Arduino Robot Motor Control - Vista Inferior

Fuente: (Costachioiu, 2015)



Figura 5 Arduino Robot Motor Control -Vista Superior

Fuente: (Costachioiu, 2015)



Figura 6 Arduino Robot - Tableros de control y motor

Fuente: (Costachioiu, 2015)

2.2.2 CARACTERÍSTICAS

El robot arduino posee las siguientes características según (Coretronica Cursos y Proyectos, 2016)

- Microcontrolador: ATmega32u4
- Voltaje de funcionamiento: 5 V
- Pines I/O digitales: 5
- Canales PWM: 6
- Canales de entradas análogas: 4 (de los pines digitales I/O)
- Canales (multiplexados) de entradas análogas: 8
- Corriente DC por cada pin I/O: 40 mA
- Memoria Flash: 32 KB (ATmega32u4) de los cuales 4 KB son utilizados por el bootloader
- SRAM: 2 KB (ATmega32u4)
- EEPROM (interno): 1 KB (ATmega32u4)
- EEPROM (externo): 512 KB (I2C)
- Velocidad de reloj: 16 MHz
- Teclado: 5 teclas
- Perilla: Potenciómetro conectado a un pin análogo

- LCD a color: Comunicación SPI
- Lector de tarjetas SD: Para tarjetas formateadas FAT16
- Altavoz: 8 Ohms
- Compás digital: Proporciona la desviación desde el norte geográfico
- Áreas de prototipado: 4

(Coretronica Cursos y Proyectos, 2016)

2.3 LABVIEW

Según (Delgado, 2014) “LabVIEW (Laboratory Virtual Instrument Engineering Workbench) es un entorno de desarrollo basado en programación gráfica”. Utiliza símbolos gráficos en lugar de lenguaje textual para describir acciones de programación. Está totalmente integrado para la comunicación con hardware GPIB, VXI, RS-232, RS-485 y tarjetas de adquisición de datos plug-in. Además incorpora librerías para estándares de software como TCP/IP y ActiveX.

Según (National Instruments, 2015) Por muchos años, “LabVIEW ha sido conocido como una herramienta de desarrollo fácil de usar para construir de sistemas de adquisición de datos y control de instrumentos de manera muy fácil y rápida”. El lenguaje de programación gráfica LabVIEW, con su interfaz de usuario gráfica integrada, está hecho especialmente para crear rápidamente prototipos de sistemas. Esto resulta particularmente evidente con el desarrollo de Express VIs en LabVIEW 7, lo cual lo hizo mucho más fácil desarrollar sistemas de adquisición de datos con LabVIEW.

Sin embargo, en alguna forma, la simplicidad y velocidad para desarrollar un sistema con LabVIEW ha escondido el hecho de que LabVIEW es de hecho, un lenguaje de programación completo, hecho para manejar las aplicaciones más grandes y complejas que ingenieros y científicos se enfrentan hoy en día. En particular, programadores creando aplicaciones críticas – control embebido, aplicaciones de monitoreo industrial, sistemas de prueba de alto desempeño – no pueden darse el lujo de introducir errores o

incertidumbre al sistema. Para este tipo de aplicaciones, debe seguirse un proceso de programación muy estructurada y en algunos casos, certificado por externos para asegurar la calidad y repetición de los códigos desarrollados. Específicamente, muchos ingenieros de prueba se enfrentan al reto de construir más y más sistemas de prueba complejos para mantenerse al día en cuanto a la creciente complejidad de los productos que están probando. Las situaciones donde un proceso de programación estructurado puede requerirse se pueden generalizar de la manera siguiente:

- Cuando se construye una aplicación particularmente grande.
- Cuando se trabaja como equipo en el código de prueba, requiriendo un acercamiento estructurado para asegurar que los miembros del equipo puedan trabajar con seguridad y eficiencia en diferentes áreas del proyecto sin conflicto alguno.
- Cuando se trabaja en una industria o área de aplicación que requiere de un proceso de codificación a certificarse por una agencia gubernamental (FDA, FAA, etc.) o por un cliente (manufacturero automotriz trabajando con proveedores).

En todos estos casos, los ingenieros pueden fácilmente adecuar la herramienta o lenguaje al proceso. Por ejemplo, muchos ingenieros pueden pensar que estas situaciones requieren de un lenguaje de programación basado en texto par desarrollar el código de prueba. En realidad, un proceso estructurado para desarrollar códigos – en cualquier lenguaje – es independiente del lenguaje o herramienta utilizada. Este documento enfatiza en algunas prácticas comunes para el seguimiento de procesos de programación estructurada y muestra cómo LabVIEW puede ser utilizado en estas situaciones. Con la introducción de LabVIEW 8, LabVIEW incorpora un número de características nuevas que lo hacen una herramienta ideal para que científicos e ingenieros puedan construir sistemas a gran escala ó críticos que requieren de acercamientos de programación estructurada.

2.3.1 VENTAJAS

Según (Scientec, 2016) “LabVIEW constituye un revolucionario sistema de programación gráfica para aplicaciones que involucren adquisición, control, análisis y presentación de datos”. Las ventajas que proporciona el empleo de LabVIEW se resumen en las siguientes:

- Se reduce el tiempo de desarrollo de las aplicaciones al menos de 4 a 10 veces, ya que es muy intuitivo y fácil de aprender.
- Dota de gran flexibilidad al sistema, permitiendo cambios y actualizaciones tanto del hardware como del software.
- Da la posibilidad a los usuarios de crear soluciones completas y complejas.
- Con un único sistema de desarrollo se integran las funciones de adquisición, análisis y presentación de datos.
- El sistema está dotado de un compilador gráfico para lograr la máxima velocidad de ejecución posible.
- Tiene la posibilidad de incorporar aplicaciones escritas en otros lenguajes.

LabVIEW es un entorno de programación destinado al desarrollo de aplicaciones, similar a los sistemas de desarrollo comerciales que utilizan el lenguaje C o BASIC. Sin embargo, LabVIEW se diferencia de dichos programas en un importante aspecto: los citados lenguajes de programación se basan en líneas de texto para crear el código fuente del programa, mientras que LabVIEW emplea la programación gráfica o lenguaje G para crear programas basados en diagramas de bloques.

Para el empleo de LabVIEW no se requiere gran experiencia en programación, ya que se emplean iconos, términos e ideas familiares a científicos e ingenieros, y se apoya sobre símbolos gráficos en lugar de lenguaje escrito para construir las aplicaciones. Por ello resulta mucho más intuitivo que el resto de lenguajes de programación convencionales.

2.3.2 NI VISA

Según (Stanton, 2017) Puede usar NI-VISA para comunicarse con la mayoría de los buses de instrumentación, incluidos GPIB, USB, Serial y Ethernet. Proporciona un conjunto de comandos consistente y fácil de usar para comunicarse con una variedad de instrumentos como muestra la siguiente imagen Figura 7.

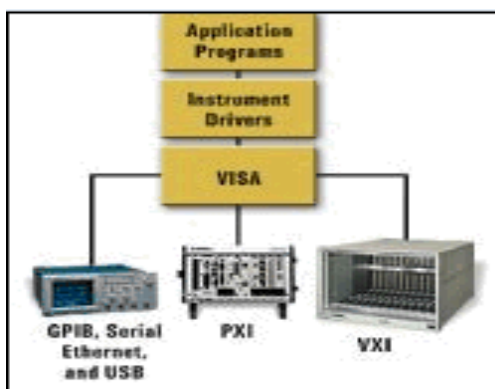


Figura 7 Comunicación Visa

Fuente: (Stanton, 2017)

2.3.2.1 APLICACIONES VISA

Según (Stanton, 2017) Una aplicación típica de VISA pasaría por los siguientes pasos:

- 1) Abrir una sesión a un recurso dado.
- 2) Realice cualquier configuración en el recurso dado (configuración de velocidades en baudios, carácter de terminación, etc.).
- 3) Realizar escrituras y lecturas al dispositivo.
- 4) Cierre la sesión al recurso.
- 5) Manejar cualquier error que pueda haber ocurrido.

NI-VISA hace que la programación de instrumentos en serie sea rápida y

fácil. VISA Open, VISA Read, VISA Write y VISA Close son las funciones básicas de VISA que se usan en el ejemplo de LabVIEW VI a continuación se muestra en la Figura 8.

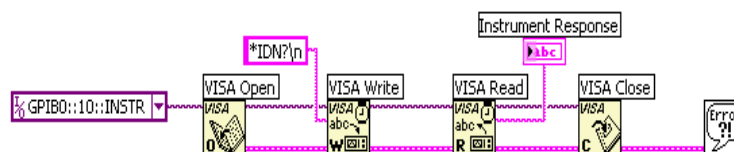


Figura 8 Programación de Visa

Fuente: (Stanton, 2017)

NI-VISA es un software API que reduce considerablemente el tiempo de desarrollo de los sistemas de prueba y medición. Brinda a los desarrolladores la capacidad de crear fácilmente un código para comunicarse con cualquier instrumento, a través de cualquier bus, en la mayoría de los sistemas operativos en uso hoy en día. También permite la generación de código que se puede mover de una plataforma a otra o de un tipo de bus a otro con poca o ninguna sobrecarga en el cambio

2.4 ARDUINO

Según (XAKATA, 2018) Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa Figura 9. A los que la comunidad de creadores puede darles diferentes tipos de uso.

Para poder entender este concepto, primero vas a tener que entender los conceptos de hardware libre y el software libre. El hardware libre son los dispositivos cuyas especificaciones y diagramas son de acceso público, de manera que cualquiera puede replicarlos. Esto quiere decir que Arduino ofrece las bases para que cualquier otra persona o empresa pueda crear sus propias placas, pudiendo ser diferentes entre ellas pero igualmente funcionales al

partir de la misma base.

El software libre son los programas informáticos cuyo código es accesible por cualquiera para que quien quiera pueda utilizarlo y modificarlo. Arduino ofrece la plataforma Arduino IDE (Entorno de Desarrollo Integrado), que es un entorno de programación con el que cualquiera puede crear aplicaciones para las placas Arduino, de manera que se les puede dar todo tipo de utilidades.

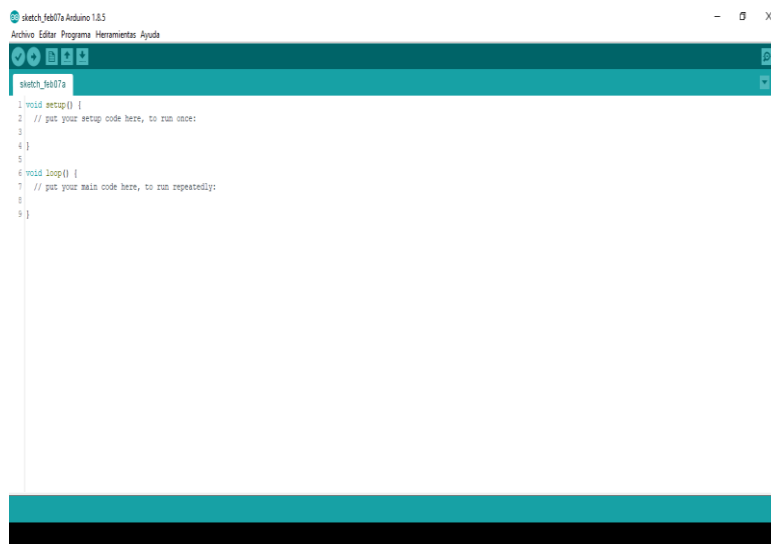


Figura 9 Plataforma de creación

2.4.1 ARDUINO NANO

2.4.1.1 CARACTERÍSTICAS

- Según (Aqeel, 2018) **Arduino Nano** es una placa para microcontroladores pequeña Figura 10. Compatible, flexible y apta para el uso del tablero, desarrollada por Arduino.cc en Italia, basada en ATmega328p (Arduino Nano V3.x) / Atmega168 (Arduino Nano V3.x).
- Viene con exactamente la misma funcionalidad que en Arduino UNO pero en tamaño pequeño.

- Viene con un voltaje de operación de 5V, sin embargo, el voltaje de entrada puede variar de 7 a 12V.
- **Arduino Nano Pinout** contiene 14 pines digitales, 8 pines analógicos, 2 pines de restablecimiento y 6 pines de alimentación.
- Cada uno de estos pines digitales y analógicos tiene asignadas múltiples funciones, pero su función principal debe configurarse como entrada o salida.
- Se actúan como pines de entrada cuando están interconectados con sensores, pero si está conduciendo algo de carga, entonces utilícelos como salida.
- Las funciones como `pinMode ()` y `digitalWrite ()` se usan para controlar las operaciones de los pines digitales, mientras que la analógica (`analogWrite ()`) se usa para controlar los pines analógicos.
- Los pines analógicos vienen con una resolución total de 10 bits que miden el valor de cero a 5V.
- Arduino Nano viene con un oscilador de cristal de frecuencia de 16 MHz. Se utiliza para producir un reloj de frecuencia precisa utilizando voltaje constante.
- Hay una limitación en el uso de Arduino Nano, ya que no viene con un conector de alimentación de CC, lo que significa que no puede suministrar una fuente de alimentación externa a través de una batería.
- Esta placa no usa USB estándar para la conexión con una computadora, en cambio, viene con soporte Mini USB.
- El tamaño pequeño y la naturaleza apta para el uso del tablero hacen de este dispositivo una opción ideal para la mayoría de las aplicaciones en las que el tamaño de los componentes electrónicos es una gran preocupación.

- La memoria flash es de 16KB o 32KB, todo depende de la placa Atmega, es decir, Atmega168 viene con 16KB de memoria flash, mientras que Atmega328 viene con una memoria flash de 32KB. La memoria flash se utiliza para almacenar el código. El 2KB de memoria de la memoria flash total se utiliza para un cargador de arranque.



Figura 10 Arduino Nano

Fuente: (Aqeel, 2018)

2.4.1.2 COMUNICACIÓN

Según (NewBloggerThemes, 2014) El Arduino Nano tiene una serie de instalaciones para la comunicación con un ordenador, otro Arduino, u otros microcontroladores. Los ATmega168 y ATmega328 proporcionan UART TTL (5V) de comunicación en serie, que está disponible en los pines digitales 0 (RX) y 1 (TX). Un FTDI FT232RL en los canales de mesa esta comunicación en serie a través de USB y los drivers FTDI (incluido con el software de Arduino) proporcionan un puerto com virtual para el software en el ordenador. El software de Arduino incluye un monitor de serie que permite a los datos de texto simples para ser enviados hacia y desde la placa Arduino. Las RX y TX LED en el tablero parpadean cuando se están transmitiendo datos a través de la conexión USB FTDI chip y al ordenador (pero no para la comunicación en serie en los pines 0 y 1).

El alcance depende del voltaje con el que alimentemos el módulo y la antena que usemos. A 5V y con la antena del módulo, el alcance difícilmente excederá de los 2 metros. Alimentando a 12V y con una antena de cobre de 16.5cm el rango en exteriores puede alcanzar 300 metros.

La comunicación es simplex (canal único y unidireccional) y tienen baja velocidad de transmisión (típicamente 2400bps). Se realiza, básicamente, por modulación ASK (amplitude shift keying). No disponen de filtro ni ID por hardware, por lo que si queremos una comunicación robusta tendremos que implementarlo por software

Este tipo de comunicación RF 433MHz es muy frecuente en proyectos caseros de electrónica y robótica, debido a su bajo precio y medio-largo alcance en condiciones adecuadas, aunque tenemos que tener en cuenta que la robustez y calidad de la comunicación puede ser limitada.

Ejemplos de uso son recepción remota de sensores como temperatura presión, aplicaciones de domótica y edificios inteligentes, activación remota de dispositivos como iluminación, alarmas, o control y monitorización de robots y vehículos.

2.5.1 APLICACIONES

Según (Alsina, 2015) Los productos DNT proporcionan mayores alcances para las aplicaciones en exteriores, de hasta 400 millas (643,7 kilómetros) para la banda de los 900 MHz, y 10 millas (16 kilómetros) para la de los 2,4 GHz en la línea de visión. Esto convierte a estos módulos de radiofrecuencia en ideales para aplicaciones de alumbrado para calles, SCADA, drones comerciales o dedicadas al sector de la agricultura.

Murata también cita las aplicaciones industriales y las médicas como posibles campos de aplicación de estos módulos, aunque el límite no existe, y allí donde sea necesaria una comunicación de gran alcance, existe la posibilidad de utilizar estos componentes.

Configurables con altas ratios de datos por radiofrecuencia de hasta 500 kbps tanto en la banda de los 900 MHz como en la de los 2,4 GHz, es especialmente útil en los casos de aplicaciones que requieran de un uso intensivo de datos con poca latencia.

Adicionalmente, un gateway DNT y una red local (LAN) segura y separada con encriptación AES de 128 bits, ayudan su integración en las redes corporativas locales.

Finalmente, mencionar que toda la serie de módulos de radiofrecuencia a 900 MHz incorpora una topología de mesh que reduce el coste total de propiedad al limitar el número de repetidores y gateways necesarios para su implementación.

2.5.2 ESPECTRO ELECTROMAGNÉTICO

Según (López, 2016) La bobina de inducción de Hertz producía radiación electromagnética con un longitud de onda de 1 m aproximadamente. Esto viene a ser un millón de veces la longitud de onda de la luz visible. Experimentos posteriores mostraron que era posible obtener un rango muy amplio y continuo de longitudes (y frecuencias) de ondas electromagnéticas. El rango completo se conoce como espectro electromagnético.

No debe confundirse espectro electromagnético con el espectro visible, que incluye solo las frecuencias visibles por el ojo humano. En principio, el espectro electromagnético abarca frecuencias desde 0 hertz (la unidad de medida de la frecuencia se denomina así en honor de Hertz, obviamente, y su símbolo es Hz; corresponde a un ciclo por segundo, en este caso un hertz corresponde a una oscilación completa de los campos eléctrico y magnético por segundo) a infinito, pero en la práctica se ha estudiado el rango de frecuencias que va desde 1 Hz a los 10²⁶ Hz, lo que corresponde a longitudes de onda entre los 10⁸ y los 10⁻¹⁸ m. Como resultado de estos estudios se han encontrado aplicaciones prácticas para muchos subrangos de frecuencias como la siguiente imagen Fig12:

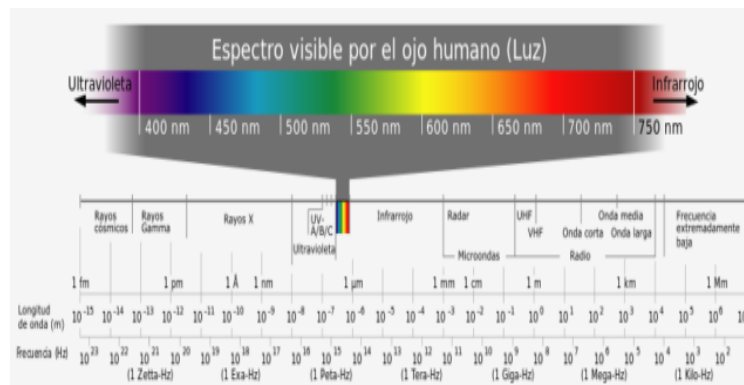


Figura 12 Subrangos de Frecuencia

Fuente: (López, 2016)

2.5.3 ONDAS DE RADIO

Según (Cuaderno de Cultura Científica, 2017) Longitudes de onda entre 10 m a 10.000 m; frecuencias entre 10^4 Hz a 10^7 Hz. Las ondas electromagnéticas en esta parte del espectro se reflejan muy bien en las capas de iones, cargadas eléctricamente, que existen en las capas altas de la atmósfera (ionosfera). Esta reflexión hace que sea posible detectar ondas de radio a grandes distancias de la fuente. Como las señales de radio tienen longitudes de onda desde decenas a miles de metros, pueden difractarse alrededor de pequeños obstáculos, como árboles o edificios, que no impiden su transmisión, pero colinas de cierta consideración y cadenas montañosas generan “zonas de sombra” donde la señal no llega.

Las ondas de radio que pueden viajar largas distancias, bien directamente o usando repetidores, son muy útiles para transportar información. Se emplean no solo para las transmisiones de radio convencional sino también en telefonía móvil mediante “células” geográficas (de aquí, telefonía celular) cuyo núcleo es un solo transmisor de radio (antena de telefonía). Un teléfono móvil no deja de ser un transceptor de radio celular, puesto que recibe y envía señales de radio.

En la Tabla 1 se muestra las bandas de frecuencias de las ondas de radio

Tabla 1

Bandas de Frecuencia

| NOMBRE | ABREVIATURA | FRECUENCIA | LONGITUD DE ONDA |
|-----------------------|-------------|-------------|------------------|
| Extra baja frecuencia | ELF | 3-30Hz | 100000–10000km |
| Súper baja frecuencia | SLF | 30-300Hz | 10000–1000km |
| Ultra baja frecuencia | ULF | 300–3000Hz | 1000–100km |
| Muy baja frecuencia | VLf | 3–30kHz | 100–10km |
| Baja frecuencia | LF | 30–300kHz | 10–1km |
| Media frecuencia | MF | 300–3000kHz | 1km–100m |
| Alta frecuencia | HF | 3–30MHz | 100–10m |
| Muy alta frecuencia | VHF | 30–300MHz | 10–1m |
| Ultra alta frecuencia | UHF | 300–3000MHz | 1m–100mm |
| Súper alta frecuencia | SHF | 3-30GHz | 100-10mm |
| Extra alta frecuencia | EHF | 30-300GHz | 10–1mm |

2.6 SENSORES

Según (Mecafenix, 2017), Los sensores son aparatos que son capaz de transformar magnitudes físicas o químicas, llamadas variables de instrumentación, en magnitudes eléctricas. Las variables de instrumentación dependen del tipo de sensor y pueden ser por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, pH, etc. Una magnitud eléctrica obtenida puede ser una resistencia eléctrica (como en una RTD), una capacidad eléctrica (como en un sensor de humedad), una tensión eléctrica (como en un termopar), una corriente eléctrica (como un fototransistor), etc.

A continuación en la Fig 13 se muestra los tipos de sensores:

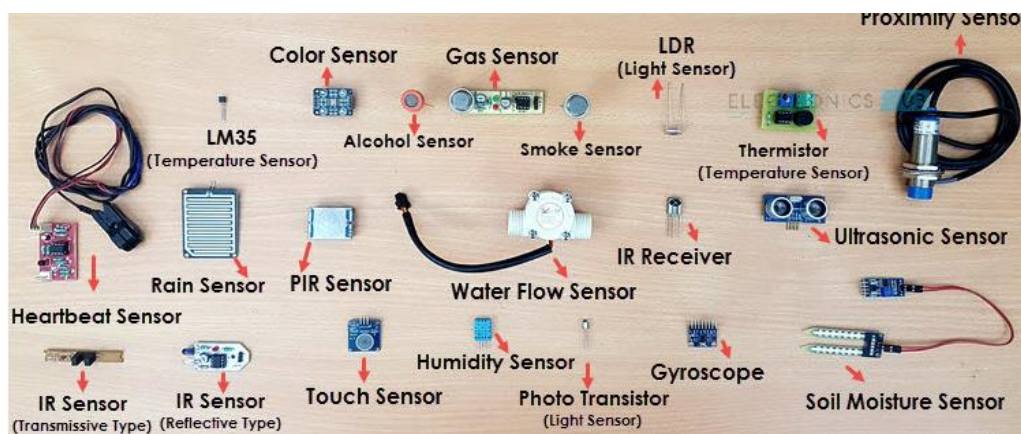


Figura 13 Sensores

Fuente: (prototipadoLAB, 2018)

2.6.1 DTH

Según (LLamas, 2016) El DHT11 y el DHT22 son dos modelos de una misma familia de sensores, que permiten realizar la medición simultánea de temperatura y humedad.

Estos sensores disponen de un procesador interno que realiza el proceso de medición, mediante una señal digital, por lo que resulta muy sencillo

obtener la medición desde un microprocesador como Arduino.

Ambos sensores presentan un encapsulado de plástico similar. Se puede distinguir ambos modelos por el color del mismo. El DHT11 presenta una carcasa azul, mientras que en el caso del sensor DHT22 el exterior es blanco.

De ambos modelos se muestran en la Fig 14, el DHT11 es el hermano pequeño de la familia, y cuenta peores características técnicas. El DHT22 es el modelo superior pero, por contra, tiene un precio superior.

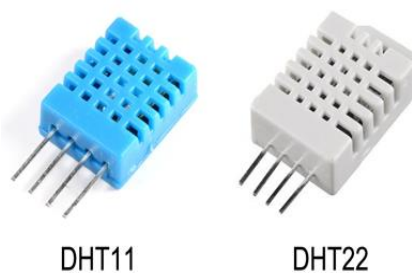


Figura 14 Sensores DTH

Fuente: (LLamas, 2016)

2.6.1.1 CARACTERÍSTICAS

A continuación en la Tabla 2 se muestra las características del DTH11 y en la Tabla 3 se muestra las características del DTH22.

Tabla 2

Características DTH11

| Alimentación | $3Vdc \leq Vcc \leq 5Vdc$ |
|--------------------------------------|---------------------------|
| Rango de medición de Temperatura | 0 a 50 °C |
| Precisión de medición de temperatura | ± 2.0 °C . |
| Resolución Temperatura | 0.1°C |
| Rango de medición de humedad | 20% a 90% RH. |
| Precisión de medición de humedad | 4% RH. |
| Resolución Humedad | 1% RH |
| Tiempo de sensado | 1 seg |

Tabla 3

Características DTH22

| Alimentación | $3Vdc \leq Vcc \leq 6Vdc$ |
|--------------------------------------|---------------------------|
| Rango de medición de temperatura | -40 a 80 °C |
| Precisión de medición de temperatura | ± 0.5 °C . |
| Resolución Temperatura | 0.1°C |
| Rango de medición de humedad | 0% a 100% RH. |
| Precisión de medición de humedad | 2% RH. |
| Resolución Humedad | 0.1% RH |
| Tiempo de sensado | 2 seg |

CAPÍTULO III

DESARROLLO DEL TEMA

3.1 INSTALACIÓN DE SOFTWARE

3.1.1 INSTALACIÓN DE ARDUINO

Instalar el software de Arduino para nuestro dispositivo (computador) este se lo descarga de la página oficial de arduino Fig. 15, a continuación el link: www.arduino.cc.

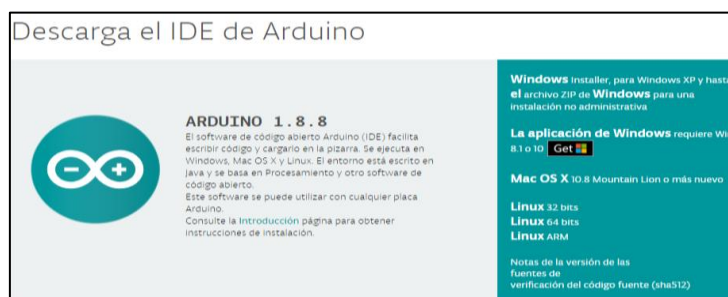


Figura 15 Arduino 1.8.8

Fuente: (Página Oficial Arduino)

- En el momento que se termina de descargar se procede a descomprimir el archivo el cual está en formato .rar o .zip.
- Al terminar de descomprimir se busca la carpeta en donde se encuentran todos los archivos se descomprime y se localiza el instalador de arduino el cual suele estar como “**arduino_debug**”, y lo ejecutaremos como administrador.

A continuación se muestra en la Fig. 16.

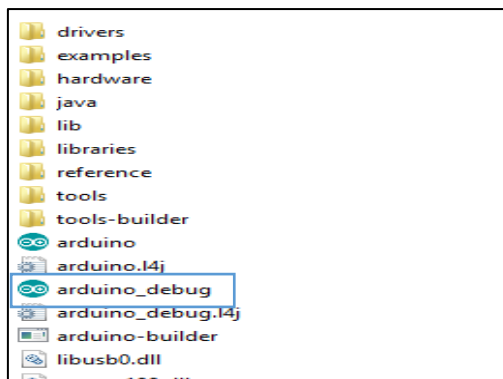


Figura 16 Drivers Arduino

- Una vez ejecutado el archivo se iniciara la instalación de Arduino.
- Ya instalado se procede abrir el programa en el cual vamos a tener la siguiente pantalla de la Fig. 17.



Figura 17 Pantalla Principal de Arduino

3.1.2 INSTALACIÓN DE LABVIEW

Instalar el software de Labview para nuestro dispositivo (computador) este se lo descarga de la página oficial de National Instruments Fig.18, a continuación el link: <http://www.ni.com/download/labview-development-system-2018/7558/en/>

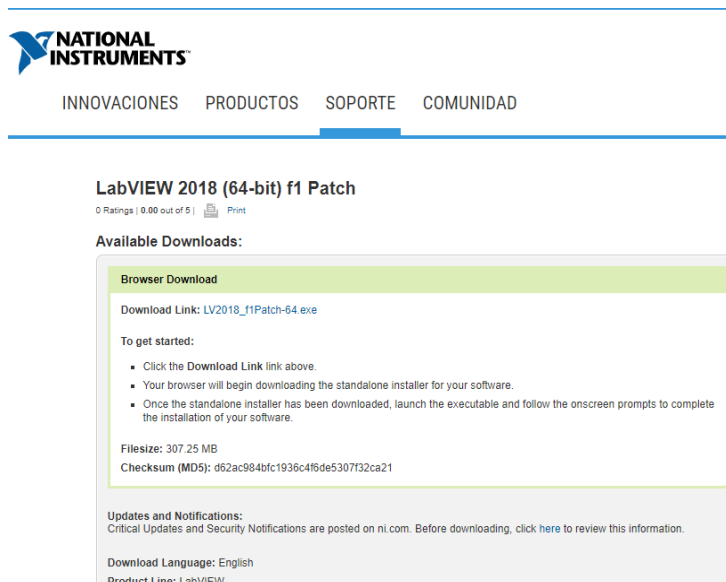


Figura 18 Labview 2018

Fuente: (Página Oficial National Instruments)

- En el momento que se termina de descargar se procede a descomprimir el archivo el cual está en formato .rar o .zip.
- Al terminar de descomprimir se busca la carpeta en donde se encuentran todos los archivos como se muestra en la Fig.19 y se procede con la instalación.

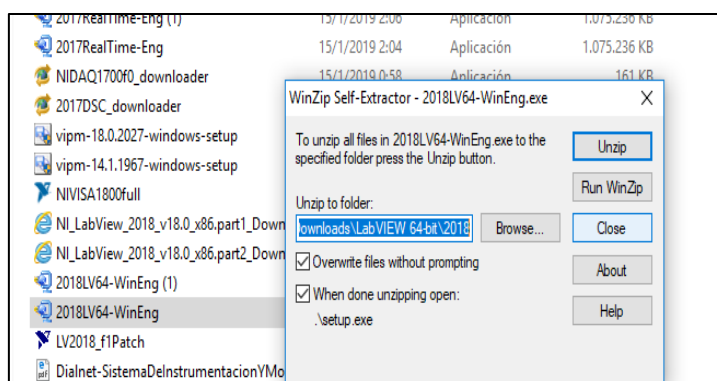


Figura 19 Archivo de Instalación.

- Una vez ejecutado el archivo se iniciara la instalación de Arduino.
- Ya instalado se procede a abrir el programa en el cual vamos a tener la siguiente pantalla como se muestra en la Fig.20.

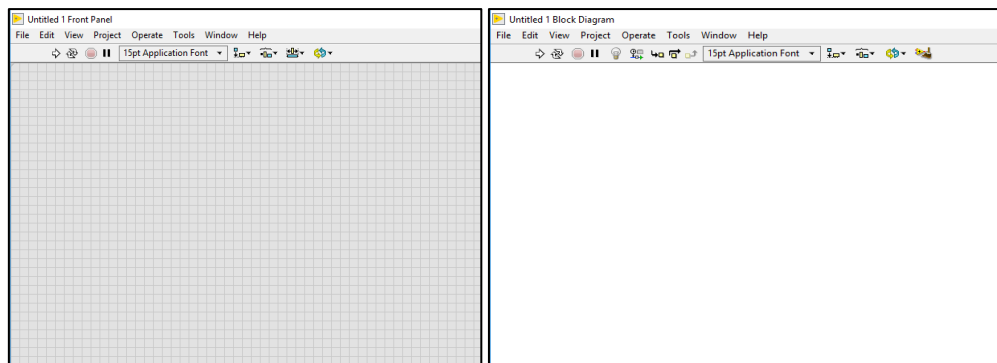


Figura 20 Pantallas Principales

3.1.2.1 INSTALACIÓN NI VISA

Instalar el software de NI VISA dentro de Labview para nuestro dispositivo (computador) este se lo descarga de la página oficial de National Instruments a continuación el link: <http://www.ni.com/download/ni-visa-18.0/7597/en/>

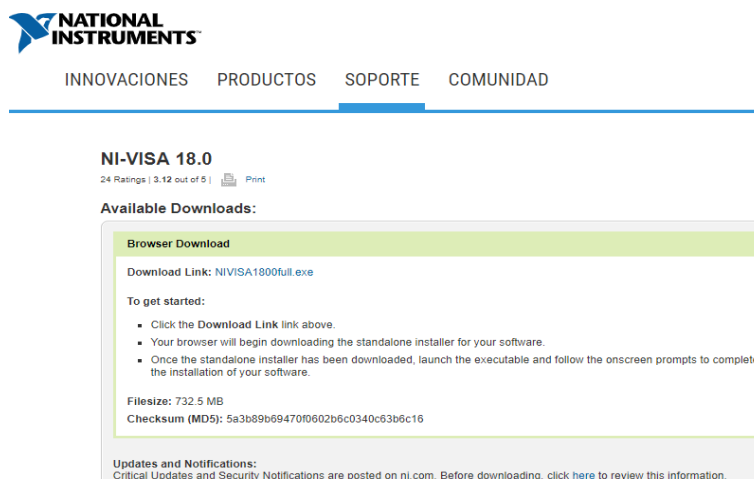


Figura 21 NI-VISA 18.0

Fuente: (Página Oficial National Instruments)

- En el momento que se termina de descargar se procede a descomprimir el archivo el cual está en formato .rar o .zip.
- Al terminar de descomprimir ubicamos la carpeta en donde se encuentran todos los archivos y se procede con la instalación.

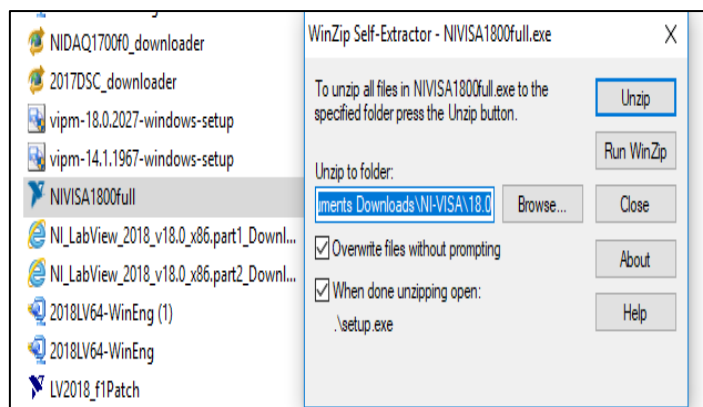


Figura 22 Archivo de Instalación NI-VISA.

3.2 PROGRAMACIÓN

3.2.1 ARDUINO NANO

- **#include "DHT.h"** se colocó la librería del sensor con el que se va a trabajar que en este caso es el DHT.
- **#define DHTPIN 2** puerto en el que vamos a utilizar
- **#define DHTTYPE DHT11** el sensor que se va a utilizar que en este caso es el sensor dht11.
- **Serial.begin(57600);** a este valor va a trabajar el Puerto de los módulos el valor puede llegar hasta el (96000) pero el valor debe ser el mismo en la programación de Arduino y en Labview.
- **pinMode 4** está configurado como salida
- **pinMode 5** está configurado como salida
- **pinMode 6** está configurado como salida
- **pinMode 7** está configurado como salida
- **float h = dht.readHumidity();** leer el valor de la humedad.
- **float t = dht.readTemperature();** leer el valor de la temperatura.
- **float f = dht.readTemperature(true);** leer el valor de la temperatura.
- **if (isnan(h) || isnan(t) || isnan(f))** condición
- **Serial.println("Failed to read from DHT sensor!");** no lee el sensor.
- **return;** retorno.

- **Serial.print(h)**; imprimir el valor de la humedad.
- **Serial.print(" ");** el espacio donde va imprimir el valor.
- **Serial.println(t)**; imprimir el valor de la temperatura.
- **while (Serial.available() > 0)**
- **char rxd=Serial.read()**; leer el carácter de la comunicación.

Si recibe la letra A se escribe en binario 100 y va hacia adelante.

- **if(rxd=='A'){//adelante**
- **digitalWrite(4,HIGH);digitalWrite(5,LOW);digitalWrite(6,LOW);**

Si recibe la letra B se escribe en binario 010 y va hacia atrás.

- **if(rxd=='B'){//atras**
- **digitalWrite(4,LOW);digitalWrite(5,HIGH);digitalWrite(6,LOW);**

Si recibe la letra D se escribe en binario 110 y 000 va hacia la izquierda.

- **if(rxd=='D'){//izquierda**
- **digitalWrite(4,HIGH);digitalWrite(5,HIGH);digitalWrite(6,LOW);**
- **delay(300);** Tiempo de giro
- **digitalWrite(4,LOW);digitalWrite(5,LOW);digitalWrite(6,LOW);**

Si recibe la letra C se escribe en binario 001 y 000 va hacia la derecha.

- **if(rxd=='C'){//derecha**
- **digitalWrite(4,LOW);digitalWrite(5,LOW);digitalWrite(6,HIGH);**
- **delay(300);** Tiempo de giro
- **digitalWrite(4,LOW);digitalWrite(5,LOW);digitalWrite(6,LOW);**

Si recibe la letra E se escribe en binario 000 y se detiene.

- **if(rxd=='E'){//OFF**
- **digitalWrite(4,LOW);digitalWrite(5,LOW);digitalWrite(6,LOW);**

A continuación en la Fig.23 se muestra la programación realizada en la plataforma de Arduino para el Arduino Nano.

```

1 #include "DHT.h"
2 #define DHTPIN 2 // what pin we're connected to
3 #define DHTTYPE DHT11 // DHT 11
4 DHT dht(DHTPIN, DHTTYPE);
5 void setup() {
6 // put your setup code here, to run once:
7 Serial.begin(57600);
8 dht.begin();
9 pinMode(4, OUTPUT);
10 pinMode(5, OUTPUT);
11 pinMode(6, OUTPUT);
12 pinMode(7, OUTPUT);
13 }
14
15 void loop() {
16 float h = dht.readHumidity();
17 float t = dht.readTemperature();
18 float f = dht.readTemperature(true);
19
20 // Check if any reads failed and exit early (to try again).
21 if (isnan(h) || isnan(t) || isnan(f)) {
22 Serial.println("Failed to read from DHT sensor!");
23 return;
24 }
25
26 // Compute heat index
27 // Must send in temp in Fahrenheit!
28 float hi = dht.computeHeatIndex(f, h);
29
30 Serial.print(h);
31 Serial.print(" ");
32 //Serial.print("Temperature: ");
33 Serial.println(t);}
34
35 while (Serial.available() > 0) {
36 char rxd=Serial.read();
37 if(rxd=='A'){//adelante
38 digitalWrite(4,HIGH);digitalWrite(5,LOW);digitalWrite(6,LOW);
39 }
40 if(rxd=='B'){//atras
41 digitalWrite(4,LOW);digitalWrite(5,HIGH);digitalWrite(6,LOW);
42 }
43 if(rxd=='D'){//izquierda
44 digitalWrite(4,HIGH);digitalWrite(5,HIGH);digitalWrite(6,LOW);
45 delay(300);
46 digitalWrite(4,LOW);digitalWrite(5,LOW);digitalWrite(6,LOW);
47 }
48 if(rxd=='C'){//derecha
49 digitalWrite(4,LOW);digitalWrite(5,LOW);digitalWrite(6,HIGH);
50 delay(300);
51 digitalWrite(4,LOW);digitalWrite(5,LOW);digitalWrite(6,LOW);
52 }
53 if(rxd=='E'){//OFF
54 digitalWrite(4,LOW);digitalWrite(5,LOW);digitalWrite(6,LOW);
55 }
56 }
57 delay(50);
58 }

```

Figura 23 Programación Arduino Nano.

3.2.2 PROGRAMACIÓN ARDUINO ROBOT (CONTROL)

- **#include <ArduinoRobot.h>** incluir librería Arduino Robot.
- **#include <Wire.h>** incluir librería de cable.
- **int t=200;** tiempo 200ms
- **void setup() {**
- **Robot.begin();** iniciar el robot.
- **pinMode(TK0,INPUT);** TK0 está configurado como entrada.
- **pinMode(TK1,INPUT);** TK1 está configurado como entrada.
- **pinMode(TK2,INPUT);** TK2 está configurado como entrada.

- **pinMode(TK3,INPUT);** TK3 está configurado como entrada.
- **Robot.motorsStop();** detener el robot.

Si lee el binario 000 que envía el Arduino Nano se detiene.

- **if((Robot.digitalRead(TK0)==LOW)&&(Robot.digitalRead(TK1)==LOW)&&(Robot.digitalRead(TK2)==LOW))**
- **Robot.motorsStop();**

Si lee el binario 100 que envía el Arduino Nano se va adelante.

- **if((Robot.digitalRead(TK0)==HIGH)&&(Robot.digitalRead(TK1)==LOW)&&(Robot.digitalRead(TK2)==LOW))**
- **adelante();**

Si lee el binario 010 que envía el Arduino Nano se va atras.

- **if((Robot.digitalRead(TK0)==LOW)&&(Robot.digitalRead(TK1)==HIGH)&&(Robot.digitalRead(TK2)==LOW))**
- **atras();**

Si lee el binario 110 que envía el Arduino Nano se va a la izquierda.

- **if((Robot.digitalRead(TK0)==HIGH)&&(Robot.digitalRead(TK1)==HIGH)&&(Robot.digitalRead(TK2)==LOW))**
- **izquierda();**

Si lee el binario 001 que envía el Arduino Nano se va a la derecha.

- **if((Robot.digitalRead(TK0)==LOW)&&(Robot.digitalRead(TK1)==LOW)&&(Robot.digitalRead(TK2)==HIGH)){**
- **derecha();**

Se mueve adelante este valor puede variar de -255 a 255

- **void adelante()**
- **Robot.motorsWrite(190, 190);**
- **delay(t);**
- **Robot.motorsStop();**

Se mueve hacia atrás este valor puede variar de -255 a 255

- **void atras()**
- **Robot.motorsWrite(-190, -190);**
- **delay(t);**
- **Robot.motorsStop();**

Se mueve hacia la izquierda este valor puede variar de -255 a 255

- **void izquierda()**
 - **Robot.motorsWrite(-190, 190);**
 - **delay(t);**
 - **Robot.motorsStop();**
- Se mueve hacia la derecha este valor puede variar de -255 a 255
- **void derecha()**
 - **Robot.motorsWrite(190, -190);**
 - **delay(t);**
 - **Robot.motorsStop();**

A continuación en la Fig.24 se muestra la programación realizada en la plataforma de Arduino para el Control del Arduino Robot.

```

1 #include <ArduinoRobot.h>
2 #include <Wire.h>
3 int t=200;
4
5 void setup() {
6   // initialize the robot
7   Robot.begin();
8   pinMode(TK0, INPUT);
9   pinMode(TK1, INPUT);
10  pinMode(TK2, INPUT);
11  pinMode(TK3, INPUT);
12  Robot.motorsStop();
13 }
14
15 void loop() {
16  if ((Robot.digitalRead(TK0)==LOW) && (Robot.digitalRead(TK1)==LOW) && (Robot.digitalRead(TK2)==LOW)) {
17    Robot.motorsStop();
18  }
19  if ((Robot.digitalRead(TK0)==HIGH) && (Robot.digitalRead(TK1)==LOW) && (Robot.digitalRead(TK2)==LOW)) {
20    adelante();
21  }
22  if ((Robot.digitalRead(TK0)==LOW) && (Robot.digitalRead(TK1)==HIGH) && (Robot.digitalRead(TK2)==LOW)) {
23    atras();
24  }
25  if ((Robot.digitalRead(TK0)==HIGH) && (Robot.digitalRead(TK1)==HIGH) && (Robot.digitalRead(TK2)==LOW)) {
26    izquierda();
27  }
28  if ((Robot.digitalRead(TK0)==LOW) && (Robot.digitalRead(TK1)==LOW) && (Robot.digitalRead(TK2)==HIGH)) {
29    derecha();

```

```

30  }
31  }
32 }
33
34 void adelante() {
35   Robot.motorsWrite(190, 190); // move forward
36   delay(t);
37   Robot.motorsStop(); // fast stop
38 }
39
40 void atras() {
41   Robot.motorsWrite(-190, -190); // move backward
42   delay(t);
43   Robot.motorsStop(); // fast stop
44 }
45
46 void izquierda() {
47   Robot.motorsWrite(-190, 190); // move left
48   delay(t);
49   Robot.motorsStop(); // fast stop
50 }
51
52 void derecha() {
53   Robot.motorsWrite(190, -190); // move left
54   delay(t);
55   Robot.motorsStop(); // fast stop
56 }

```

Figura 24 Programación del Arduino Robot

3.2.3 PROGRAMACIÓN LABVIEW

3.2.3.1 PANEL FRONTAL

En la programación de Labview utilizamos dos partes el panel frontal donde tendremos controles e indicadores y el diagrama de bloque donde realizamos las funciones.

En el Panel Frontal que se muestra en la Fig.25 utilizaremos:

- Controles
- Indicadores
- Puerto VISA

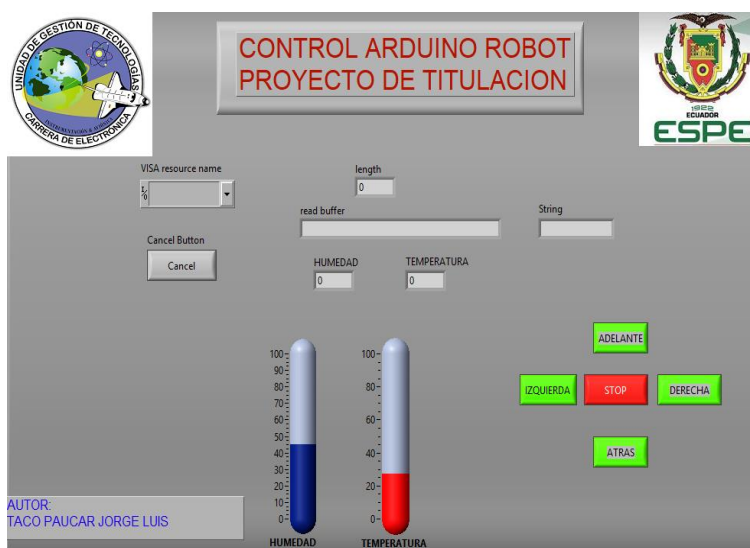


Figura 25 Panel Frontal

3.2.3.2 DIAGRAMA DE BLOQUES

En el diagrama de bloque se realizó las respectivas conexiones para obtener los resultados a continuación se explicara los pasos para realizar la programación.

1. Comunicación de VISA desde el puerto serial como se uestra en la Fig.26.

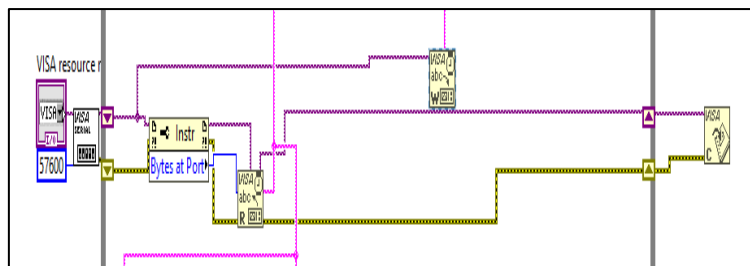


Figura 26 Comunicación VISA

- Realizar los controladores por medio de un botón para poder enviar una letra del caso que se visualizara y realizar así su función principal como se muestra en la Fig.27.

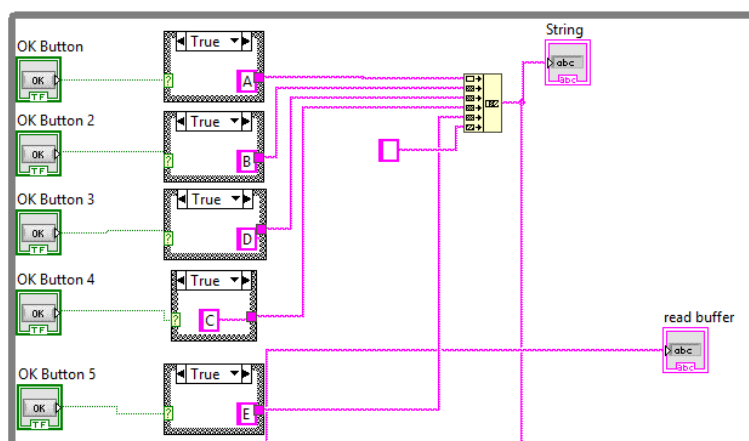


Figura 27 Casos para las funciones

- Realizar los indicadores para poder visualizar las magnitudes que estamos midiendo en este caso es la temperatura y humedad.

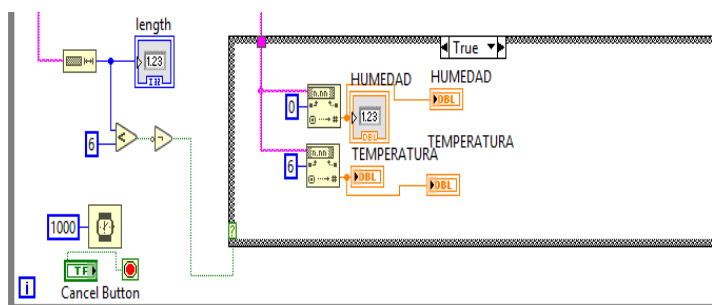


Figura 28 Indicadores

3.3 MONTAJE

3.3.1 DISEÑO EN PROTEUS

El diseño en proteus se lo realizó para verificar las conexiones correctas que se deben realizar para el funcionamiento del arduino robot como se muestra en la Fig.29.

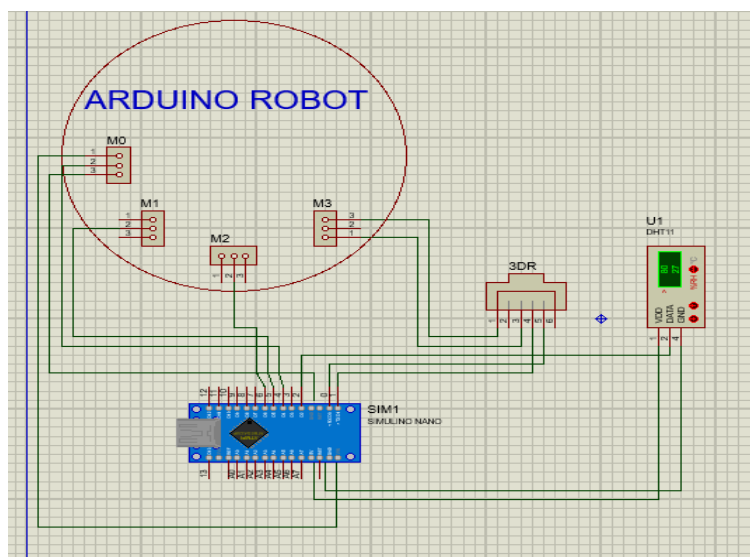


Figura 29 Esquema de Conexiones

3.3.2 CONEXIÓN

3.3.2.1 CONEXIÓN DEL MÓDULO

El módulo 3DR realiza la conexión con el arduino nano para realizar la comunicación RX y TX y se alimenta del conector M3 del Arduino Robot como se muestra en la Fig.30.

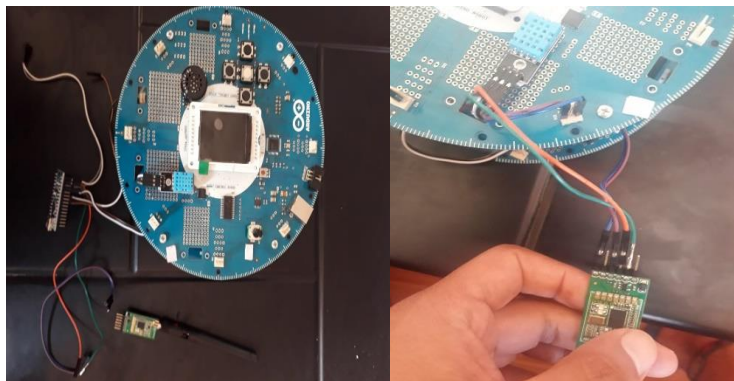


Figura 30 Conexiones del Módulo 3DR

3.3.2.2 CONEXIÓN DEL SENSOR DTH11

La conexión del DTH11 se realizó tomando la alimentación del arduino nano y la salida de señal va conectado al pin D2 del arduino nano como se muestra en la Fig.31.

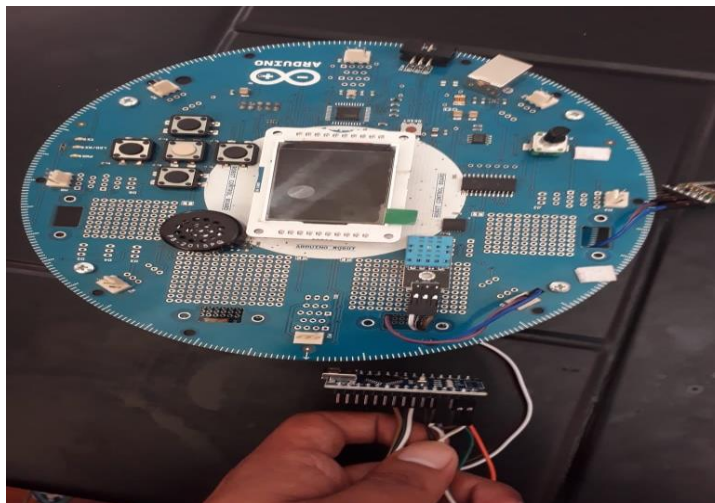


Figura 31 Conexión del Sensor DTH11

3.3.3.3 CONEXIÓN DEL ARDUINO ROBOT

El Arduino Robot se conecta con el Arduino Nano como se muestra en la Fig.32, desde los pines D4, D5, D6 del Arduino Nano que envían datos binarios hacia los conectores M0, M1, M2 del Arduino Robot que es el que los recibe y realiza sus funciones.

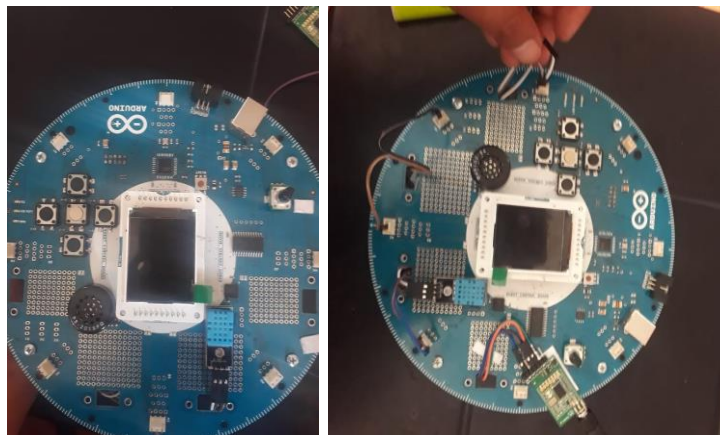


Figura 32 Conexión del Arduino Robot

3.4 AJUSTE

3.4.1 AJUSTE DEL CONTROL

El Arduino Robot se puede ajustar de la siguiente manera; en la programación que se encuentra en el Arduino Robot Control como se muestra en la Fig.33 para poder aumentar o disminuir la velocidad el valor puede variar de 0 a 255.

```

35
36 void adelante() {
37   Robot.motorsWrite(190, 190); // move forward
38   delay(t);
39   Robot.motorsStop(); // fast stop
40 }
41
42 void atras() {
43   Robot.motorsWrite(-190, -190); // move backward
44   delay(t);
45   Robot.motorsStop(); // fast stop
46 }
47
48 void izquierda() {
49   Robot.motorsWrite(-190, 190); // move left
50   delay(t);
51   Robot.motorsStop(); // fast stop
52 }
53
54 void derecha() {
55   Robot.motorsWrite(190, -190); // move left
56   delay(t);
57   Robot.motorsStop(); // fast stop
58 }

```

Figura 33 Ajuste de velocidad

El Arduino Robot se puede ajustar de la siguiente manera como se muestra en la Fig.34 en la programación que se encuentra en el Arduino Nano para poder aumentar o disminuir el tiempo de giro del Robot.

```

44 }
45 if(rxd=='D'){//izquierda
46 digitalWrite(4,HIGH);digitalWrite(5,HIGH);digitalWrite(6,LOW);
47 delay(300);
48 digitalWrite(4,LOW);digitalWrite(5,LOW);digitalWrite(6,LOW);
49 }
50 if(rxd=='C'){//derecha
51 digitalWrite(4,LOW);digitalWrite(5,LOW);digitalWrite(6,HIGH);
52 delay(300);
53 digitalWrite(4,LOW);digitalWrite(5,LOW);digitalWrite(6,LOW);
54 }

```

Figura 34 Ajuste de Giro

3.5 PUESTA EN FUNCIONAMIENTO

Se puso en funcionamiento el Arduino Robot luego de cargar los programas en las placas correspondientes y realizar sus respectivas conexiones.

3.5.1 FUNCIONAMIENTO DEL ARDUINO ROBOT

A continuación se muestra en la Fig.35 la lectura de magnitudes.

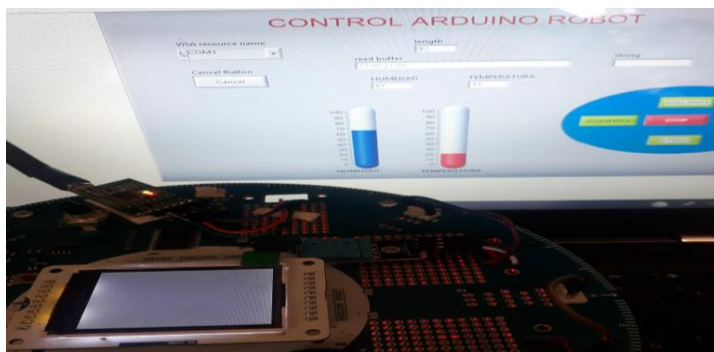


Figura 35 Lectura de Humedad y Temperatura

A continuación se muestra en la Fig.36 la prueba de lectura de magnitudes.

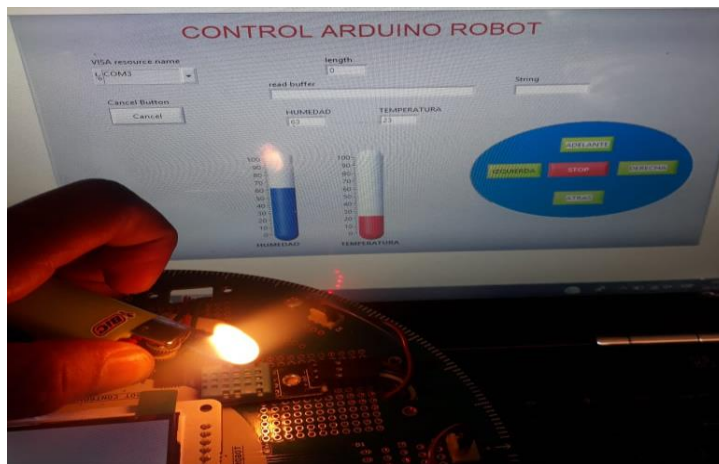


Figura 36 Prueba de Lectura de Humedad y Temperatura

A continuación se muestra en la Fig.37 la prueba de movimiento hacia adelante.



Figura 37 Monitoreo del Arduino Robot hacia adelante

A continuación se muestra en la Fig.38 la prueba de movimiento hacia atrás.

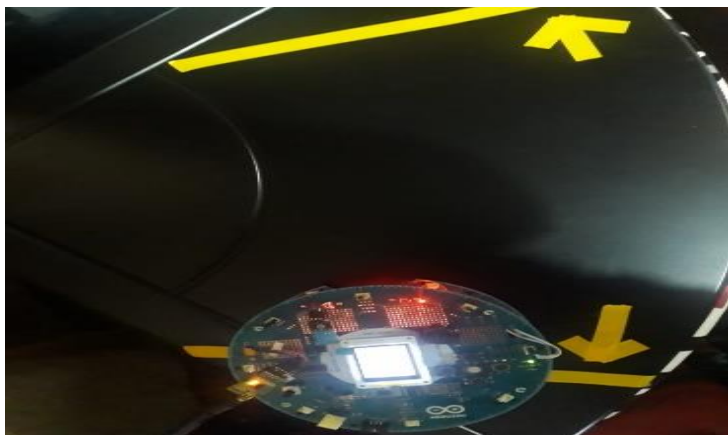


Figura 38 Monitoreo del Arduino Robot hacia atrás

A continuación se muestra en la Fig.39 la prueba de giro del robot.

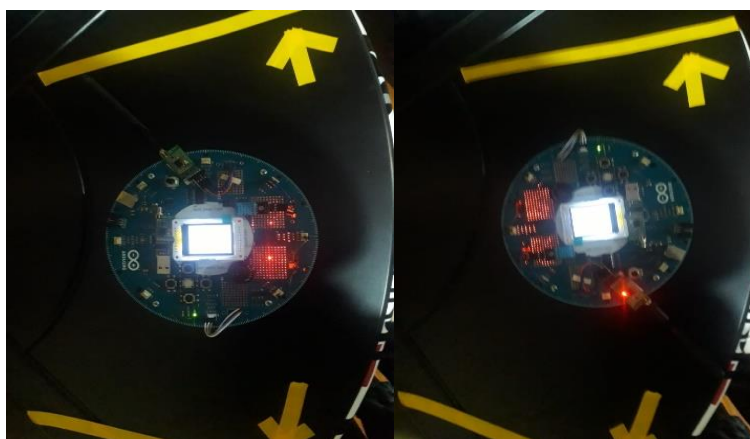


Figura 39 Giro del Arduino Robot.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- Al momento de investigar en las fichas técnicas del Arduino Robot se observó que no tenía puertos para comunicación serial (RX, TX), por lo cual se procedió a utilizar una placa de Arduino para poder realizar la comunicación entre Labview y Arduino Robot.
- Se estableció la comunicación con el Arduino Robot a través de los módulos 3DR de radio frecuencia se lo realizó mediante un Arduino Nano y mediante el mismo podemos monitorear al Arduino Robot y recibir los valores de Temperatura y Humedad.
- Se implementó una interfaz de Humano-Maquina (HMI) en Labview mediante el cual podemos controlar y al mismo tiempo supervisar la funcionalidad que está haciendo el Arduino Robot.

4.2 RECOMENDACIONES

- Cuando se vaya a realizar cualquier aplicación con el Arduino Robot primero revisar toda la ficha técnica y como están conectados sus puertos.
- Para realizar comunicación serial utilizar una placa Arduino adicional pero la misma debe tener (RX, TX) ya que así será más sencillo realizar los trabajos para una comunicación bidireccional también se puede utilizar el mismo modulo para la conexión de sensores.
- Al momento de implementar un HMI en Labview primeramente se debe instalar el programa NI VISA ya que sin él es imposible la conexión inalámbrica.

BIBLIOGRAFÍA

- Alsina, G. (26 de 09 de 2015). comunicaciones inalambricashoy. Obtenido de <https://www.comunicacionesinalambricashoy.com/modulos-de-radiofrecuencia-a-900-mhz/>
- Andrade, V., Parco, F. L., & Gustavo, O. (2013). Obtenido de <http://repositorio.uisrael.edu.ec/handle/47000/447>
- Aqeel, A. (25 de 06 de 2018). Obtenido de <https://www.theengineeringprojects.com/2018/06/introduction-to-arduino-nano.html>
- Castro, R. A. (11 de 02 de 2014). Blogspot. Obtenido de Electricidad: <http://elec3k.blogspot.com/2014/02/que-es-la-telemetria.html>
- Contreras, Á. (01 de 11 de 2014). Obtenido de <http://eduteka.icesi.edu.co/articulos/arduino1>
- Coretronica Cursos y Proyectos. (09 de 05 de 2016). Obtenido de <https://www.youtube.com/watch?v=Bo81kwiJyOE>
- Costachioiu, T. (25 de 07 de 2015). ELECTRONZA. Obtenido de <https://electronza.com/arduino-robot-review/>
- Cuaderno de Cultura Científica. (23 de 08 de 2017). Obtenido de <https://culturacientifica.com/2016/08/23/ondas-de-radio/>
- Delgado, P. S. (12 de 06 de 2014). Obtenido de <https://es.scribd.com/document/229317148/Que-Es-Labview>
- Escalera, E. M. (13 de 03 de 2015). Compumundo Hipermegaredes. Obtenido de <https://compumundohipermegaredes.wordpress.com/2015/03/13/que-es-la-telemetria/>

- Geek Factory. (04 de 05 de 2016). Obtenido de <https://www.geekfactory.mx/tutoriales/tutoriales-arduino/diagrama-de-pines-arduino-pinout-arduino/>
- Llamas, L. (06 de 12 de 2016). Obtenido de <https://www.luisllamas.es/comunicacion-inalambrica-en-arduino-con-modulos-rf-433mhz/>
- LLamas, L. (29 de 03 de 2016). Obtenido de <https://www.luisllamas.es/arduino-dht11-dht22/>
- López, C. T. (16 de 08 de 2016). Cuaderno de Cultura Científica. Obtenido de <https://culturacientifica.com/2016/08/16/el-espectro-electromagnetico/>
- Mecafenix, F. (11 de 04 de 2017). Ingeniería Mecafenix. Obtenido de <http://www.ingmecafenix.com/automatizacion/sensores/>
- National Instruments. (07 de 10 de 2015). Obtenido de <http://www.ni.com/white-paper/3046/es/>
- NewBloggerThemes. (09 de 06 de 2014). Aprendiendo Arduino. Obtenido de <http://aprendiendoarduino27.blogspot.com/2016/08/arduino-nano.html#comment-form>
- Pastor, J. (20 de 05 de 2013). xataka. Obtenido de <https://www.xataka.com/robotica-e-ia/llega-el-robot-arduino-nueva-tentacion-para-aficionados-a-la-robotica>
- prototipadoLAB. (05 de 05 de 2018). Obtenido de <http://paolaguimerans.com/openeart/?p=1372>
- Reisancho Casco, S. P. (09 de 05 de 2016). Repositorio Institucional de la Universidad de las Fuerzas Armadas ESPE. Obtenido de <http://repositorio.espe.edu.ec/handle/21000/12403>
- Ruesca, P. (25 de 09 de 2016). Radio Comunicaciones. Obtenido de <http://www.radiocomunicaciones.net/radio/telemetria/>

Scientec. (24 de 09 de 2016). Obtenido de <https://www.scientec.com.mx/labview/>

Stanton, A. (27 de 04 de 2017). Obtenido de <https://blog.digilentinc.com/using-ni-visa-in-labview/>

XAKATA. (03 de 08 de 2018). XAKATA. Obtenido de <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>

ANEXOS

ÍNDICE DE ANEXOS

ANEXO A.- PROGRAMACIÓN

ANEXO B.- FOTOGRAFÍAS

ANEXO A

PROGRAMACIÓN ARDUINO NANO

```
nano $
1 #include "DHT.h"
2 #define DHTPIN 2
3 #define DHTTYPE DHT11
4 DHT dht(DHTPIN, DHTTYPE);
5 void setup() {
6
7   Serial.begin(57600);
8   dht.begin();
9   pinMode(4,OUTPUT);
10  pinMode(5,OUTPUT);
11  pinMode(6,OUTPUT);
12  pinMode(7,OUTPUT);
13 }
14
15 void loop() {
16 float h = dht.readHumidity();
17 float t = dht.readTemperature();
18 float f = dht.readTemperature(true);
19
20
21 if (isnan(h) || isnan(t) || isnan(f)) {
22   Serial.println("Failed to read from DHT sensor!");
23   return;
24 }
25 float hi = dht.computeHeatIndex(f, h);
26
27 // Serial.print("Humidity: ");
28 Serial.print(h);
29 Serial.print(" ");
30 //Serial.print("Temperature: ");
31 Serial.println(t);
32 //Serial.print(" *C ");
33
34 while (Serial.available() > 0) {
35 char rxd=Serial.read();
36 if (rxd=='A'){//adelante
37 digitalWrite(4,HIGH);digitalWrite(5,LOW);digitalWrite(6,LOW);
38 }
39 if (rxd=='B'){//atras
40 digitalWrite(4,LOW);digitalWrite(5,HIGH);digitalWrite(6,LOW);
41 }
42 if (rxd=='D'){//izquierda
43 digitalWrite(4,HIGH);digitalWrite(5,HIGH);digitalWrite(6,LOW);
44 delay(300);
45 digitalWrite(4,LOW);digitalWrite(5,LOW);digitalWrite(6,LOW);
46 }
47 if (rxd=='C'){//derecha
48 digitalWrite(4,LOW);digitalWrite(5,LOW);digitalWrite(6,HIGH);
49 delay(300);
50 digitalWrite(4,LOW);digitalWrite(5,LOW);digitalWrite(6,LOW);
51 }
52 if (rxd=='E'){//OFF
53 digitalWrite(4,LOW);digitalWrite(5,LOW);digitalWrite(6,LOW);
54 }
55 }
56 delay(50);
57 }
```

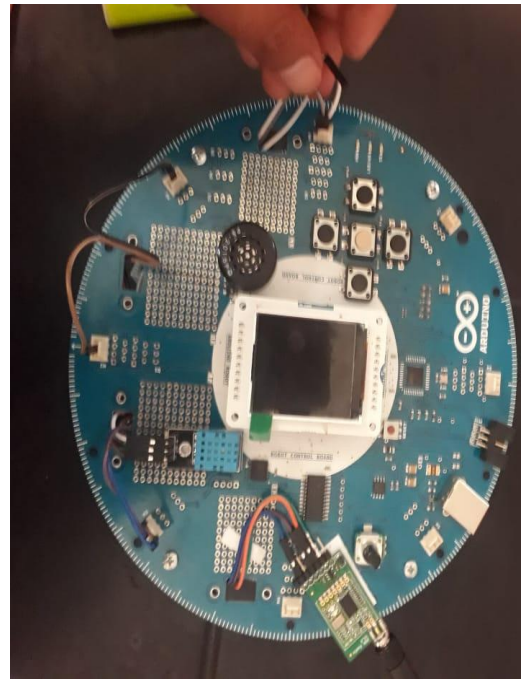
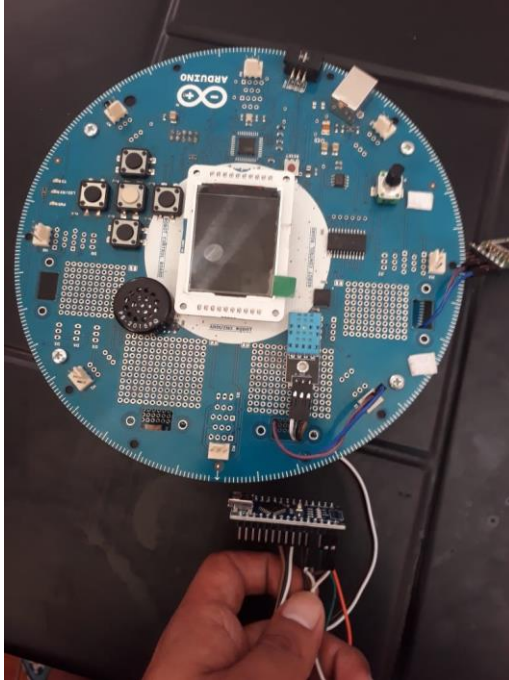
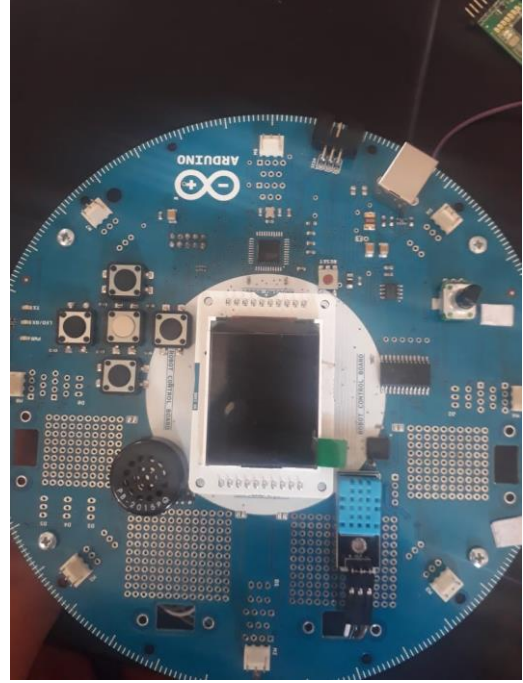
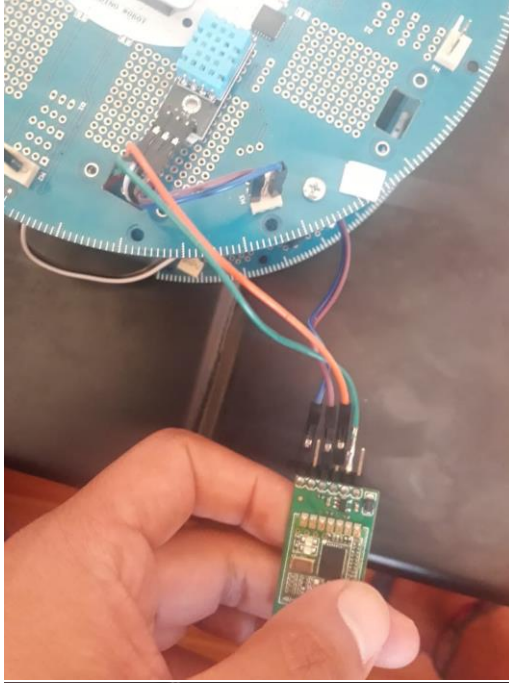
PROGRAMACIÓN ARDUINO ROBOT

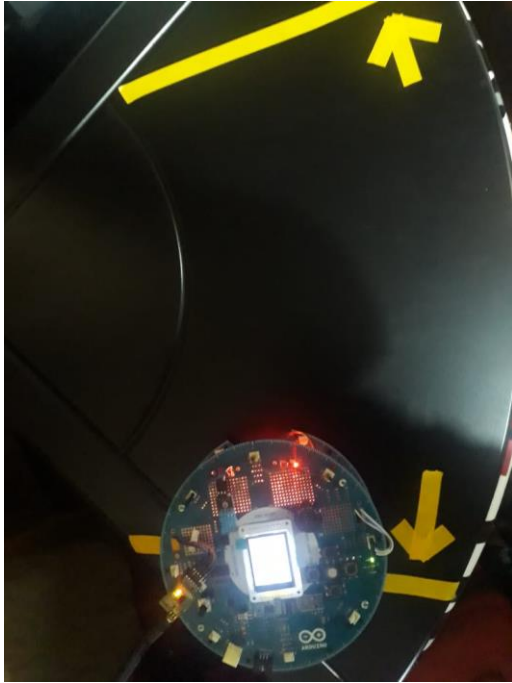
```
MotorTest_nano $
1 #include <ArduinoRobot.h>
2 #include <Wire.h>
3 int t=200;
4
5 void setup() {
6   // initialize the robot
7   Robot.begin();
8   pinMode(TK0, INPUT);
9   pinMode(TK1, INPUT);
10  pinMode(TK2, INPUT);
11  pinMode(TK3, INPUT);
12  Robot.motorsStop();
13 }
14
15 void loop() {
16  if((Robot.digitalRead(TK0)==LOW)&&(Robot.digitalRead(TK1)==LOW)&&(Robot.digitalRead(TK2)==LOW)) {
17    Robot.motorsStop();
18  }
19  if((Robot.digitalRead(TK0)==HIGH)&&(Robot.digitalRead(TK1)==LOW)&&(Robot.digitalRead(TK2)==LOW)) {
20    adelante();
21  }
22  if((Robot.digitalRead(TK0)==LOW)&&(Robot.digitalRead(TK1)==HIGH)&&(Robot.digitalRead(TK2)==LOW)) {
23    atras();
24  }
25  if((Robot.digitalRead(TK0)==HIGH)&&(Robot.digitalRead(TK1)==HIGH)&&(Robot.digitalRead(TK2)==LOW)) {
26    izquierda();
27  }
28  if((Robot.digitalRead(TK0)==LOW)&&(Robot.digitalRead(TK1)==LOW)&&(Robot.digitalRead(TK2)==HIGH)) {
29    derecha();
```

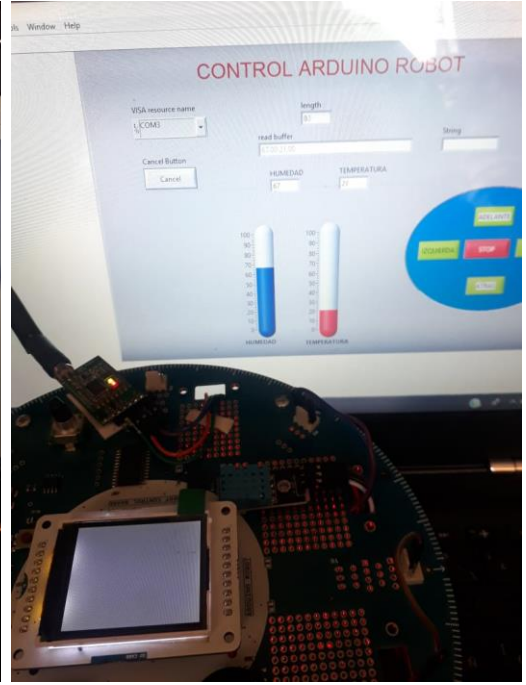
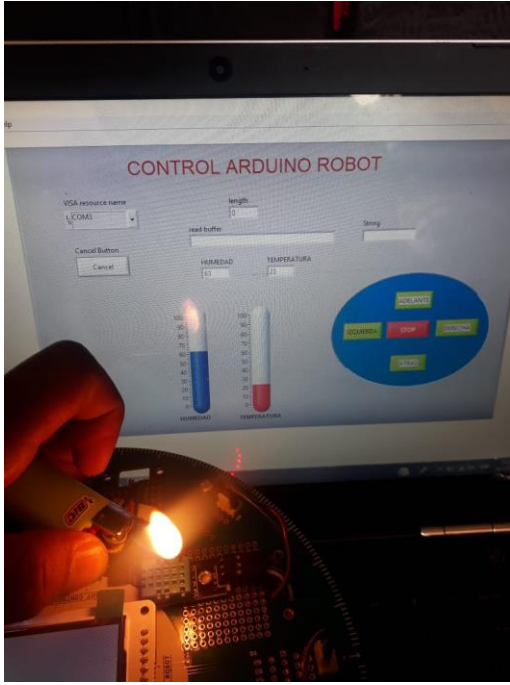
```
30  }
31 }
32 }
33
34 void adelante() {
35   Robot.motorsWrite(190, 190); // move forward
36   delay(t);
37   Robot.motorsStop(); // fast stop
38 }
39
40 void atras() {
41   Robot.motorsWrite(-190, -190); // move backward
42   delay(t);
43   Robot.motorsStop(); // fast stop
44 }
45
46 void izquierda() {
47   Robot.motorsWrite(-190, 190); // move left
48   delay(t);
49   Robot.motorsStop(); // fast stop
50 }
51
52 void derecha() {
53   Robot.motorsWrite(190, -190); // move left
54   delay(t);
55   Robot.motorsStop(); // fast stop
56 }
```


ANEXO B

FOTOGRAFÍAS







HOJA DE VIDA



DATOS PERSONALES:

Nombres y Apellidos: Jorge Luis Taco Paucar

Lugar y Fecha de Nacimiento: Quito, 20 de septiembre de 1996

Estado civil: Soltero

Dirección personal: Quito, Sector Guajalo

Número de teléfono de teléfono: 0958834288

Dirección de correo electrónico: luistaco1996@gmail.com

FORMACIÓN ACADÉMICA:

- **Secundaria y Bachillerato:**

Periodo 2007 / 2014, Colegio Técnico Industrial “Miguel de Santiago”

Título:

“ELECTRÓNICA DE CONSUMO”

- **Universidad:**

Periodo 2015 / 2019, Universidad de las Fuerzas Armadas “ESPE”-UGT

Título:

TEGNÓLOGO EN ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y AVIÓNICA.

EXPERIENCIA PROFESIONAL:

Sony Car Audio: Servicio técnico - mantenimiento de equipos electrónicos y máquinas eléctricas. 5 años.

Saurus Ecuador: Supervisor de Proyectos – 5 meses actualmente

Induce del Ecuador: Instalaciones Eléctricas – Pasantías Pre Profesionales. (120hrs)

Saurus Ecuador: Mantenimiento de máquinas eléctricas – Pasantías Pre Profesionales. (120hrs)

CERTIFICADOS:

- Jornadas de Electromecánica. Duración 30hrs
- Ingles nivel B1

**UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
UNIDAD DE GESTIÓN DE TECNOLOGÍAS**

HOJA DE LEGALIZACION DE FIRMAS

**DEL CONTENIDO DE LA PRESENTE INVESTIGACIÓN SE
RESPONSABILIZA EL AUTOR**

JORGE LUIS TACO PAUCAR

CC. 1723014633

**DIRECTOR DE LA CARRERA DE ELECTRÓNICA MENCIÓN
INSTRUMENTACIÓN Y AVIÓNICA**

ING. PABLO PILATASIG

Latacunga, Febrero Del 2019