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

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

TEMA: "IMPLEMENTACION DE UN MONITOREO INALAMBRICO DE TRANSMISORES DE SALIDA ANALOGICA, EMPLEANDO TECNOLOGIA XBEE PARA PRACTICAS DE REDES INDUSTRIALES"

AUTOR: VALLEJO SANTIANA LUIS CRISTOBAL

DIRECTOR: ING. PABLO PILATAXI

LATACUNGA

2017

DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN

CARRERA ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y AVIÓNICA

CERTIFICACIÓN

Certifico que el trabajo de Titulación,**"IMPLEMENTACION DE UN MONITOREO INALAMBRICO DE TRANSMISORES DE SALIDA ANALOGICA, EMPLEANDO TECNOLOGIA XBEE PARA PRACTICAS DE REDES INDUSTRIALES"** realizado el Sr. **VALLEJO SANTIANA LUIS CRISTOBAL,** 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 las Fuerzas Armadas ESPE, por lo tanto me permito acreditarlo y autorizar al señor **VALLEJO SANTIANA LUIS CRISTOBAL** para que lo sustente públicamente.

Latacunga, 22 de Febrero del 2017

Atentamente,

Ing.Pablo Pilataxi

DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN

CARRERA ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y AVIÓNICA

AUTORÍA DE RESPONSABILIDAD

Yo, **VALLEJO SANTIANA LUIS CRISTOBAL,** con cédula de identidad Nº 1719088872, declaro que este trabajo de titulación **"IMPLEMENTACION DE UN MONITOREO INALAMBRICO DE TRANSMISORES DE SALIDA ANALOGICA, EMPLEANDO TECNOLOGIA XBEE PARA PRACTICAS DE REDES INDUSTRIALES"** 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.

Consecuentemente 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, 22 de Febrero 2017.

_____

Vallejo Santiana Luis Cristóbal

C.I: 1711655132

DEPARTAMENTO DE ELECTRÓNICA Y COMPUTACIÓN

CARRERA ELECTRÓNICA MENCIÓN INSTRUMENTACIÓN Y AVIÓNICA

AUTORIZACIÓN (PUBLICACIÓN BIBLIOTECA VIRTUAL)

Yo, **VALLEJO SANTIANA LUIS CRISTOBAL,** autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación **"IMPLEMENTACION DE UN MONITOREO INALAMBRICO DE TRANSMISORES DE SALIDA ANALOGICA, EMPLEANDO TECNOLOGIA XBEE PARA PRACTICAS DE REDES INDUSTRIALES"** cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Latacunga, 22 de Febrero 2017.

_____

Vallejo Santiana Luis Cristóbal

C.I: 1711655132

DEDICATORIA

A Dios.

Por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

A Carina

Bella esposa que me ha apoyado siempre para hacer los sueños realidad, gracias por tu cariño y apoyo factores fundamentales que me brindan equilibrio.

A José David, Luis Alejandro, Luis David

Si algo me ha motivado a concluir esta carrera es el amor incondicional que ustedes me han dado, gracias a cada momento en familia sacrificado para lograr este peldaño, gracias por entender que el éxito demanda muchos sacrificios.

AGRADECIMIENTO

Agradezco a Dios, artífice de todos los pasos que en mi vida he logrado

A mi esposa, que me apoyado constantemente

Mis padres Favian y Guadalupe, quienes siempre estuvieron latentes de mi esfuerzo y dedicación por poder alcanzar esta meta, los gestores de los valores y enseñanzas que me ha encaminado por senda de la responsabilidad y el respeto

Gracias a todas las personas que ayudaron directa e indirectamente en la realización de este proyecto.

ÍNDICE DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El presente Proyecto Técnico trata de la implementación de un monitoreo inalámbrico de transmisores de salida analógica, empleando tecnología Xbee para prácticas de redes industriales se ha desarrollado con comunicación punto a punto mediante la tarjeta Arduino Uno, el código de programación para la recepción de datos de los transmisores y el envío de datos inalámbrico se ha desarrollado en la plataforma Arduino IDE y para visualizar los valores de las señales analógicas enviadas por el Arduino Uno en forma inalámbrica se ha desarrollado un interfaz y tratamiento de datos en el software Labwiev. Las señales analógicas empleadas para el proyecto pertenecen a los transmisores de nivel ultrasónico Banner S18UIS y al transmisor de temperatura Th-200 monitoreo, las dos señales analógicas son conectadas directamente a las entradas de la placa Arduino la misma que se reenvía por el pin1 Tx hacia el módulo de transmisión Xbee S2 denominado “coordinador”, y en la parte de recepción el otro módulo Xbee S2 denominado “end device” que se encarga de transmitir la información mediante el protocolo 232 al computador donde se procesan los datos para que mediante una interface desarrollada en LabView se monitoree en línea los datos de nivel y temperatura. La comunicación que se ejecuta por el puerto serial, se procesa dentro de los bloques de programación con el fin de realizar el escalamiento de la respuesta de los transmisores y tener una respuesta real del proceso, de manera que la interface sea amigable con el usuario.

Palabras Claves

XBEE

ARDUINO UNO

COMUNICACION

SEÑAL ANALOGICA

LABWIEV

ABSTRACT

The present Technical Project deals with the implementation of wireless monitoring of analog output transmitters using Xbee technology for industrial networking practices has been developed with point-to-point communication using the Arduino Uno card, the programming code for receiving data from Transmitters and wireless data sending has been developed in the Arduino IDE platform and to visualize the values of the analog signals sent by the Arduino One wirelessly has developed an interface and data processing in the Labwiev software. The analog signals used for the project belong to the Banner S18UIS ultrasonic transmitters and the Th-200 temperature transmitter monitoring, the two analog signals are connected directly to the inputs of the Arduino board which is forwarded by the pin1 Tx to The Xbee S2 transmission module called "coordinator", and in the receiving part the other module Xbee S2 called "end device" which is responsible for transmitting the information through the protocol 232 to the computer where the data is processed so that through an interface Developed in LabView to monitor the level and temperature data online. The communication that is executed by the serial port is processed within the programming blocks in order to carry out the scaling of the response of the transmitters and to have a real response of the process, so that the interface is friendly to the user.

Keywords

XBEE

ARDUINOUNO

COMMUNICATION

ANALOG SIGNAL

LABWIEV

Lic. Wilson Vilalviciencio

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1 ANTECEDENTES

En la ciudad de Latacunga provincia de Cotopaxi se encuentra ubicada la Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas - ESPE, institución que está encauzada a la enseñanza de materias enfiladas de logística, seguridad aérea, electrónica, y mantenimiento de aeronaves, apoyados en material técnico de sus múltiples laboratorios promoviendo el aprendizaje a través de las prácticas.

La institución acrecienta sus esfuerzos a través de investigaciones generando así una cultura de búsqueda de conocimientos, es imperativo señalar que todo este trabajo servirá para futuras generaciones.

Es por ello que en la institución se han realizado algunas investigaciones, entre las cuales se puede citar, y tienen relación con el objeto de estudio.

(Molina, 2012) realizó un trabajo de investigación para monitorear de manera inalámbrica sensores de temperatura y humedad utilizando módulos Xbee Pro, el cual mediante los módulos mencionados se puedo obtener datos y realizar el respectivo monitoreo de los sensores, que sirvió de uso práctico para los estudiantes en el laboratorio de control industrial de la UGT- ESPE.

(Jácome, 2013) diseñó e implementó un tipo de red inalámbrica tipo Mesh utilizando módulos Zigbee, el cual mediante este tipo de topología logró determinar que para este tipo de red la incidencia de otros dispositivos inalámbricos y electrónicos no afectan el funcionamiento de esta configuración.

El presente proyecto de investigación es importante para adentrarnos en la tecnología inalámbrica, para explotar este campo de comunicación

1.2 PLANTEAMIENTO DEL PROBLEMA

En la Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas – ESPE, en la asignatura de Automatización y Control de Procesos existe la necesidad de implementar la tecnología XBee, debido a que este tipo de tecnología cuenta con características robustas, y su utilización se está haciendo cada vez más indispensable en el campo industrial, se ha propuesto esta investigación para que los estudiantes tengan conocimientos y desarrollen su habilidad sobre el manejo de este dispositivo y puedan utilizarlo en diferentes aplicaciones como por ejemplo en Redes Industriales.

Esta tecnología XBee contribuye con soluciones integradas que brindan un medio inalámbrico para la interconexión y comunicación entre dispositivos

Esta tecnología crece a pasos agigantados, no involucrarse significaría quedar relegados a la tecnología inalámbrica y a sus comunicaciones

1.3 JUSTIFICACIÓN

La implementación de un monitoreo inalámbrico de transmisores de salida analógica, empleando tecnología Xbee, permitirá a las futuros tecnólogos de la UGT-ESPE estar a la par del desarrollo de la industria e indagar el funcionamiento de este tipo de tecnologías, de este modo podrán realizar prácticas en la asignatura de Automatización y Control de Procesos. Es importante porque beneficiará a los estudiantes en la interacción y práctica con la placa mencionada, además, facilita el estudio de comunicación inalámbrica 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, de igual manera no demandará de un largo tiempo para su implementación y ejecución, por tal razón es necesario implementar un monitoreo inalámbrico de transmisores de salida analógica, empleando tecnología XBee para prácticas en la asignatura de Automatización y Control de Procesos de la UGT -ESPE.

1.4 OBJETIVOS

1.4.1 Objetivo General

Implementar el monitoreo inalámbrico de transmisores de salida analógica, empleando tecnología XBee, para prácticas en la asignatura Redes Industriales de la Unidad de Gestión de Tecnologías de la Universidad de las Fuerzas Armadas – ESPE

1.4.2 Objetivos Específicos

- Indagar las características de la tecnología Xbee mediante la revisión bibliográfica de manuales, sitios web.
- Establecer los requerimientos mínimos de hardware y software para la conexión inalámbrica de los transmisores de salida.
- Establecer la comunicación inalámbrica para monitorear los transmisores de salida analógica.
- Verificar el funcionamiento, atreves de pruebas en laboratorio de Instrumentación Virtual

1.5 ALCANCE

El proyecto está dirigido a la Carrera de Electrónica Mención Instrumentación y Aviónica de la Unidad de Gestión de Tecnologías como una herramienta muy importante para el proceso de aprendizaje teórico-práctico de los estudiantes y docentes de esta carrera. Para la realización de prácticas, con la tecnología Xbee

CAPÍTULO II

MARCO TEÓRICO

2.1 **Xbee**

XBee son módulos de soluciones integradas de radio frecuencia que brindan un medio inalámbrico para la interconexión y comunicación entre dispositivos, los módulos Prearmados (XBEE_XBEE ZB) implementan internamente la comunicación Zigbee con lo cual reducen la complejidad a simples oraciones a través de un puerto de serie. (DIGI, 2016)

Los módulos XBee son ideales para aplicaciones enfocadas a sectores energéticos y control donde la eficiencia de los equipos es crítica. Su SPI (Serial Peripheral Interface) provee una interfaz de alta velocidad e integración mediante micro controladores (DIGI, 2016).

Figura 1 Xbee S2

Fuente: (DIGI, 2016)

2.1.1 Xbee S2

El módulo Xbee S2 posibilita la opción de innovar redes de malla compleja basadas en el firmware de malla XBee ZB ZigBee, y funcionan mediante un simple protocolo serie TTL. Permiten una comunicación bidireccional entre micro controlador, computador o prácticamente cualquier cosa que disponga de un puerto serie o Serie/USB a nivel TTL. Para su

funcionamiento nos basaremos en las siguientes características. (DIGI, 2016)

Tabla 1

Características técnicas

CARACTERISTICAS TECNICAS
Alimentación: 3.3V
Velocidad de transferencia: 250kbps Max
Potencia de salida: 1mW o 60mW (+18dBm)
Alcance: 90metros o 1500 metros aprox.
Certificado FCC
6 pines ADC de 10-bit
8 pines digitales IO
Encriptación 128-bit
Configuración local o de forma inalámbrica
Comandos AT o API

Fuente: (Colección Manuales Users)

2.1.2 Funcionamiento Xbee S2

Para que el Xbee entre en funcionamiento es necesario suministrar al mismo con el voltaje indicado en la Tabla 1. Para lo cual existen shields de alimentación de voltaje, comunicación y configuración varían de acuerdo a las necesidades reales de operación, alimentar con el voltaje apropiado y mantener comunicación son las prioridades que las vamos a revisar en los siguientes shields. (DIGI, 2016)

2.1.2.1 Xbee Spark Fun Explorer

Lo más destacado de esta placa es un convertidor FT231X de USB a serie. Eso es lo que traduce los datos entre el ordenador y el XBee. También hay un botón de reinicio, y un regulador de voltaje para suministrar el XBee con mucha potencia. Además, hay cuatro LEDs que ayudarán si alguna vez tiene que depurar el XBee: RX, TX, RSSI (indicador de intensidad de señal), y el indicador de alimentación. (DIGI, 2016)

Esta también se desata cada uno de los pines de E / S de la XBee a un par de colectores compatible con la placa. (DIGI, 2016)

Figura 2 Xbee Spark Fun Explorer

Fuente: (DIGI, 2016)

2.1.2.2 Xbee Explorer regulado

Se encarga de la regulación de 3.3V, el acondicionamiento de señal, y los indicadores de actividad básicas (alimentación, RSSI y DIN / DOUT LED de actividad). Se traduce las señales de serie de 5 V a 3,3 V para que pueda conectar un 5V (hasta 3,3 V) del sistema a cualquier módulo XBee. La placa fue convenientemente diseñada para acoplarse directamente con la serie Pro Sparkfun Arduino de tablas parabootloading inalámbrica y la configuración basada en USB. (DIGI, 2016)

Esta placa está totalmente llena con regulador de 3.3V (5V de entrada máx), zócalo XBee, cuatro LEDs de estado, y el cambio de nivel.

Figura 3 Xbee explorer regulado

Fuente: (DIGI, 2016)

2.1.3 Modos de operación

Los módulos Xbee pueden operar de cinco formas:

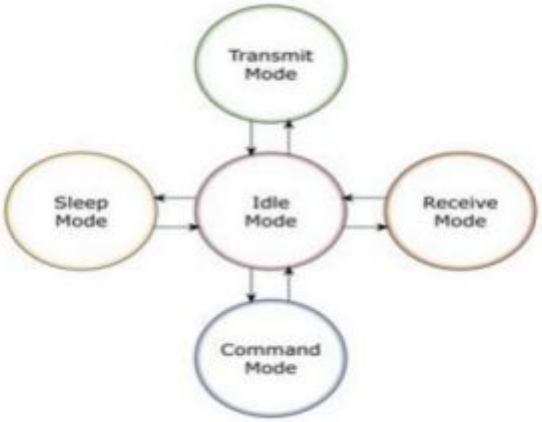


Figura 4 Modos de operación módulo Xbee

Fuente: (Colección Manuales Users)

2.1.3.1 Modo recibir-transmitir

Se encuentra en estos modos cuando al módulo le llega algún paquete RF a través de la antena(modo Receive) o cuando se manda información serial al buffer del pin 3 (UART Data in) que luego será transmitida (modo Transmit). (Colección Manuales Users)

2.1.3.2 Modos de sueños cíclicos

Sueño cíclico remoto

El modo de sueño cíclico remoto permite que el módulo revise la data por la interfaz RF periódicamente.

Sueño cíclico remoto y Pin para despertar

Este modo se utiliza para despertar un módulo remoto, ya sea por la interfaz RF o por poner en estado bajo el pin Sleep_RQ utilizado para comunicación orientada a eventos. (DIGI, 2016)

Coordinador de sueño

Este modo configura al módulo para funcionar como coordinador de sueño.

2.1.3.3 Modo de comando

Este modo permite ingresar comandos AT al módulo Xbee, para configurar, ajustar o modificar parámetros. Permite ajustar parámetros como la dirección propia o la de destino, así como su modo de operación entre otras cosas. (DIGI, 2016)

2.1.3.4 Modo transparente

El modo Transparente viene por defecto en los módulos Xbee. Este modo está destinado principalmente a la comunicación punto a punto.

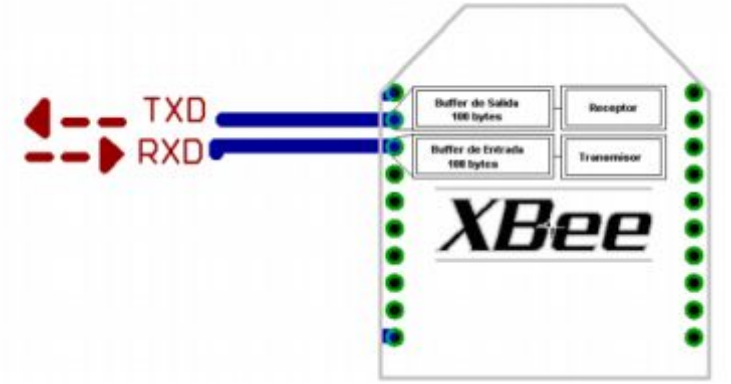


Figura 5 Transmisión de datos

Fuente: (Colección Manuales Users)

2.1.3.5 Modo de operación API

Cuando el módulo XBEE se encuentra en este modo, toda la información que entra y sale, es empaquetada en frames.

2.1.4 Red Malla ZigBee

La red malla ZigBee se basa en la interconexión de varios dispositivos los cuales se comunican mediante saltos hasta alcanzar su destino, si encuentra algún tipo de obstáculo o llega a fallar algún dispositivo dentro de la red, el tráfico de la red se re direcciona hacia otro dispositivo disponible, en una red malla ZigBee siempre se envía el tráfico por medio de la mejor ruta disponible según los algoritmos de ruteo del protocolo (Gislason, 2008) .

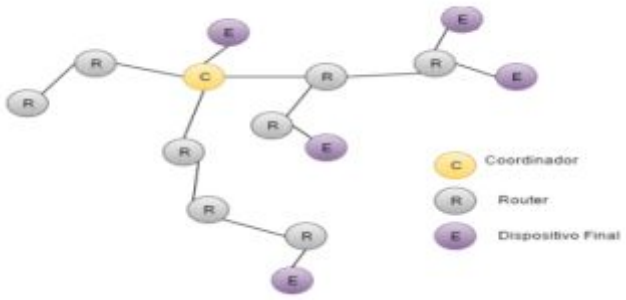


Figura 6 Estructura red malla ZigBee

Fuente: (Gislason, 2008)

2.1.5 Entorno de Configuración

2.1.5.1 XCTU

X-CTU es una aplicación basada en Windows que provee una interfaz gráfica para la configuración e interacción con los módulos de radiofrecuencia del fabricante DIGI

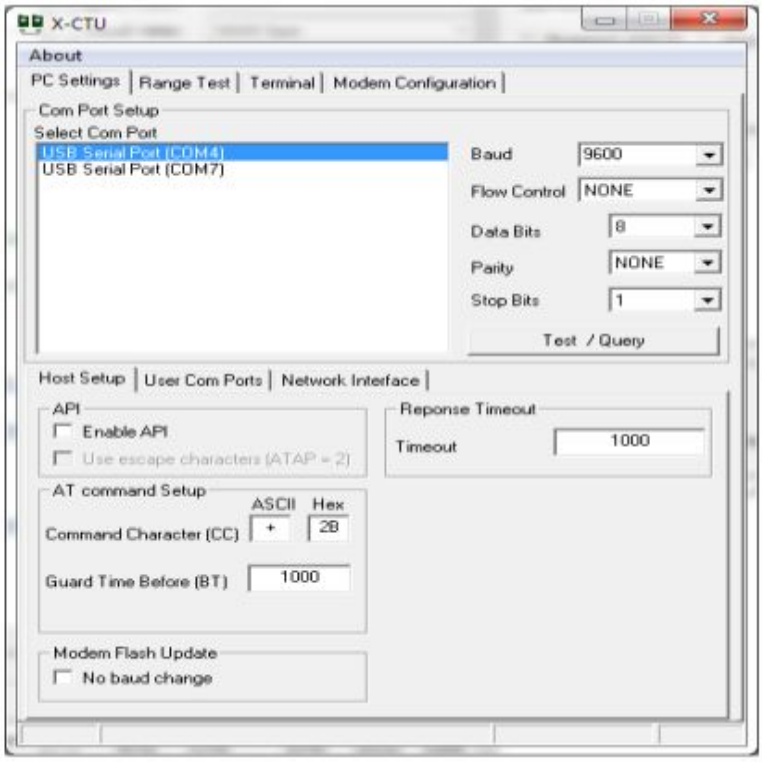


Figura 7 Software XCTU

Fuente: (DIGI, 2016)

En la pantalla principal se presentan los parámetros de configuración inicial que se los describe en la tabla3.

Tabla 2

Parámetros de Configuración Inicial

Parámetros	Descripción
PC Settings	Permite seleccionar el puerto y la configuración del mismo
Range Test	Permite desarrollar un test de rango entre dos dispositivos
Terminal	Permite acceso al puerto COM del computador con un programa de emulación terminal. También desde esta pantalla se puede acceder al firmware de los módulos mediante comandos AT. La lista de comandos AT se encuentra en el manual del producto
Modem Configuration	Permite programar los parámetros de los radios mediante una interfaz gráfica de usuario. Además permite cambiar la versión del firmware del radio

Fuente: (Colección Manuales Users)

2.1.5.2 Configuración de red punto a punto

Configura ración básica una red Xbee, La tabla3 muestra la configuración que involucra dos módems.

Tabla 3

Configuración de red punto a punto

XBee A Valores	XBee B Valores
DH 13A200	DH 13A200
DL 4076E267	DL 4076E26E
MY AAAA	MY AAAA
SH 13A200 (viene por defecto)	SH 13A200 (viene por defecto)
SL 4076E26E (viene por defecto)	SL 4076E267 (viene por defecto)
CE 1 -Coordinator	CE 0 -End Device Serie 1 Pro

Fuente: (Colección Manuales Users)

2.2 Arduino

Arduino es una plataforma computacional física que se fundamenta en código abierto, su arquitectura está desarrollada por circuitos impresos que integran un micro controlador y una interface de desarrollo de programación, su diseño está considerado para facilitar el uso de la electrónica y la programación en proyectos multidisciplinares. (Arduino, Arduino Uno, 2016)

Actualmente Arduino ha desarrollado varios productos oficiales destinados a satisfacer las necesidades tecnológicas del usuario como se muestra en la Figura 1 (Arduino, Arduino Uno, 2016)

Figura 8 Productos Oficiales de Arduino

Fuente: (Arduino, Productos de Arduino, 2016)

2.2.1 Arduino Uno

El Arduino Uno cuenta con un micro controlador ATmega328. La placa posee 14 pines de (E/S) entrada/salida digital (6 pueden ser usados como salidas PWM), 6 entradas analógicas, un oscilador de cuarzo a 16MHz, una conexión USB controlada por el micro controlador ATmega16U2 para transferencia de datos con mayor velocidad y extender la capacidad de me memoria (Gislason, 2008)

El ATmega328 tiene 32KB (0,5KB son usados por el bootloader). Además tiene 2KB de SRAM y 1KB EEPROM (que puede ser leída y escrita con la librería EEPROM). (Gislason, 2008)

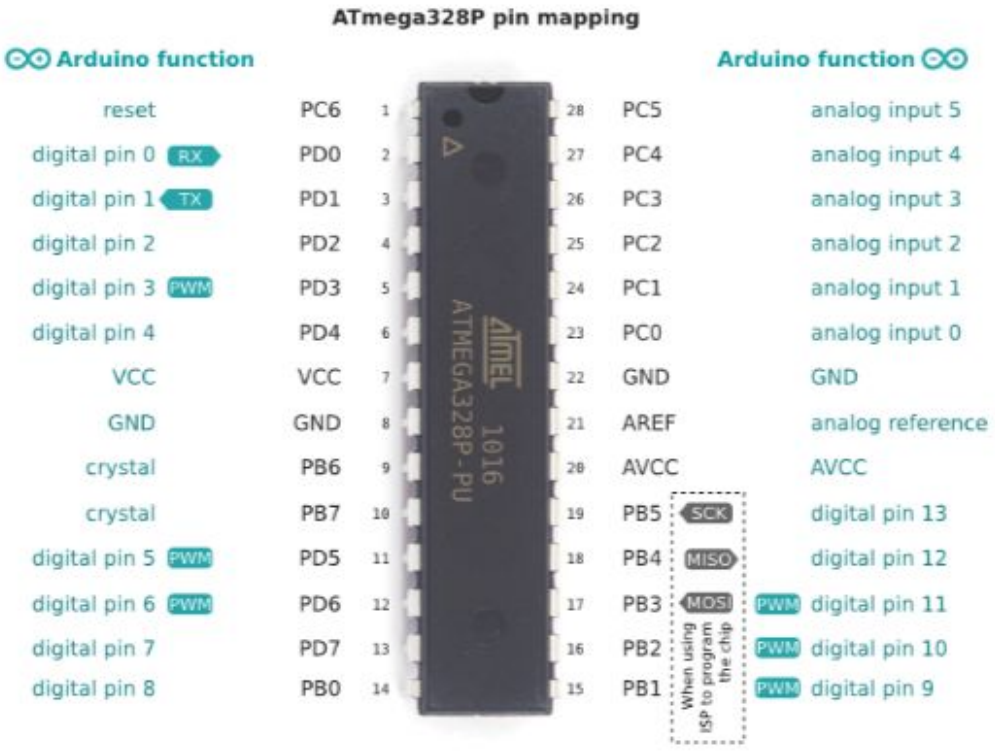


Figura 9 Distribución de pines Atmega 328

Fuente: (Arduino, Arduino Uno, 2016)

Otras de las características de Arduino Uno se describen en la siguiente tabla.

Tabla 4

Microcontrolador Atmega

Micro controlador	ATmega328P
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límite)	6-20V
E / S digitales prendedores	14 (de los cuales 6 proporcionan salida PWM)
Memoria flash	32 KB (ATmega328P) de los cuales 0,5 KB utilizado por el gestor de arranque
PWM digital pines I / O	6
Pines de entrada analógica	6
Corriente continua para Pin I / O	20 mA
Corriente CC para Pin 3.3V	50 mA
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Velocidad de reloj	16 MHz
LED_BUILTIN	13

Fuente: (Colección Manuales Users)

2.2.2 Alimentación

Puede operar con una alimentación externa de 6 a 20 voltios. Si la tensión suministrada es menor a 7V, sin embargo, el pin de 5V puede suministrar menos de cinco voltios y la placa podría ser inestable. Si se usa

más de 12V, el regulador de voltaje podría sobrecalentarse y dañar la placa. El rango recomendado es de 7 a 12 voltios (Evans & Ruiz Gutierrez, 2011)

2.2.3 Entradas y Salidas

Salida y entrada digital: los valores de salida pueden ser o 0 V (LOW) o 5 V (HIGH), y se interpretará una entrada de entre 0 y 2 V como LOW y de entre 3 y 5 V como HIGH. (Evans & Ruiz Gutierrez, 2011)

Salida analógica: los valores de salida van desde 0 V a 5 V en un rango de 0 a 255 (precisión de 8 bits) valores intermedios. Entrada analógica: los valores de entrada van desde 0 V a 5 V en un rango de 0 a 1023 (precisión de 10 bits) valores intermedios. La intensidad máxima de todos estos pines es de 40 mA. (Evans & Ruiz Gutierrez, 2011)

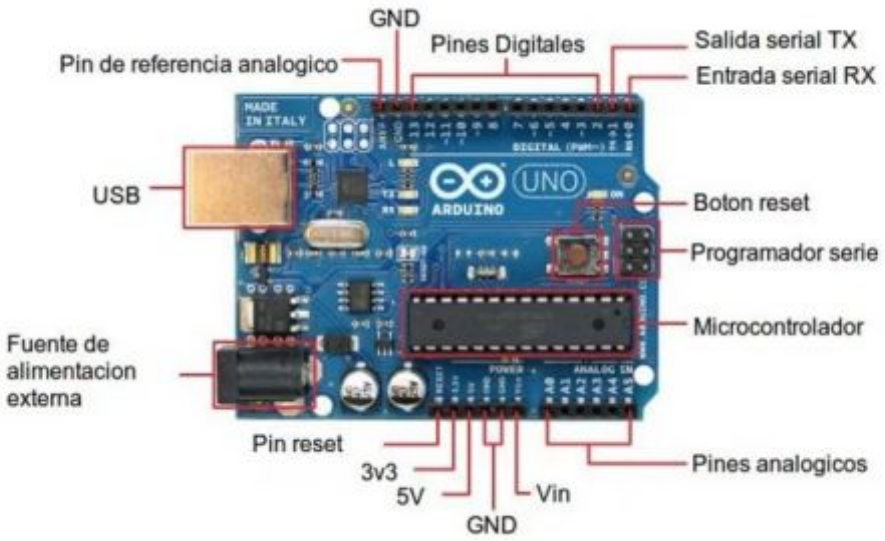


Figura 10 Placa Arduino Uno

Fuente: (Artero, 2013)

***Serial*:** Se utilizan los pines **0 (RX)** recibir **y 1 (TX)** transmitir datos serie de señales TTL.

SPI: Arduino permite trasladar información full dúplex en una comunicación Maestro/Esclavo. Los pines del 10 al 13 para esta comunicación

***Interrupciones externas*:** se puede definir dos interrupciones por hardware llamadas 0 y 1, conectadas a los pines 2 y 3

***PWM*:** Arduino proporciona 6 salidas de 8 bits destinadas a la generación de señales PWM

***LED 13*:** La placa integra en su hardware un led de estado conectado al pin 13. Se encenderá cuando dicho pin se configura como

AREF. Voltaje de referencia para entradas analógicas. Se usa con la función analogReference(). (Arduino, Arduino Uno, 2016)

***Entradas analógicas*:** 6 entradas analógicas, nombradas desde A0 hasta A5, las cuales proporcionan 10 bits de resolución (, 1024 valores diferentes)

***Serial*:** Se utilizan los pines **0 (RX)** recibir **y 1 (TX)** transmitir datos serie de señales TTL. (Arduino, Arduino Uno, 2016)

SPI: Arduino permite trasladar información full dúplex en una comunicación Maestro/Esclavo. Los pines del 10 al 13 para esta comunicación (Arduino, Arduino Uno, 2016)

***Interrupciones externas*:** se puede definir dos interrupciones por hardware llamadas 0 y 1, conectadas a los pines 2 y 3

***PWM*:** Arduino proporciona 6 salidas de 8 bits destinadas a la generación de señales PWM (Arduino, Arduino Uno, 2016)

***LED 13*:** La placa integra en su hardware un led de estado conectado al pin 13. Se encenderá cuando dicho pin se configura como

AREF. Voltaje de referencia para entradas analógicas. Se usa con la función analogReference(). (Arduino, Arduino Uno, 2016)

***Entradas analógicas*:** 6 entradas analógicas, nombradas desde A0 hasta A5, las cuales proporcionan 10 bits de resolución (1024 valores diferentes)

2.2.4 Comunicación

Comunicación serie UART TTL (5V), la cual está disponible en los pines digitales 0 (RX) y 1 (TX). Un chip ATmega16U2 en la placa canaliza esta

comunicación serie al USB y aparece como un puerto COM virtual en el software del ordenador (Arduino, Arduino Uno, 2016)

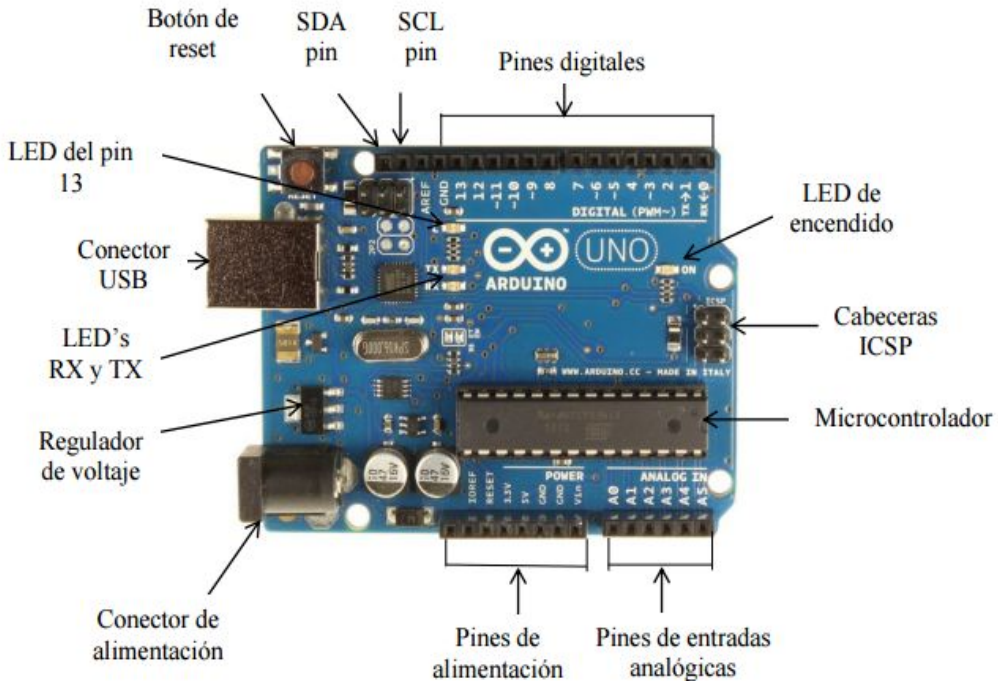


Figura 11 Placa Arduino Uno

Fuente: (Artero, 2013)

2.2.5 Estructura de programación

La estructura básica del lenguaje de programación de Arduino se compone de tres secciones necesarias, la primera sección la función de declaraciones de variables globales, la segunda sección llamada "Void setup" delimitada por llaves de apertura y cierre; la tercera la sección "Void loop" delimitada por llaves de apertura y cierre (Evans & Ruiz Gutierrez, 2011)

La función de configuración debe contener la declaración de las variables. Es la primera función a ejecutar en el programa, se ejecuta solo una vez, y se utiliza para configurar o inicializar pinMode (modo de trabajo de las E/S), configuración de la comunicación en serie y otras. (Arduino, Productos de Arduino, 2016)

En donde setup( ) es la parte encargada de recoger la configuración y loop( ) es la que contienen el programa que se ejecutara cíclicamente (de ahí el termino loop –bucle-). Ambas funciones son necesarias para que el programa trabaje. (Arduino, Arduino Uno, 2016)

La función bucle (loop) contiene el código que se ejecutara continuamente (lectura de entradas, activación de salidas, etc) Esta función es el núcleo de todos los programas de Arduino y la que realiza la mayor parte del trabajo. (Arduino, Productos de Arduino, 2016)

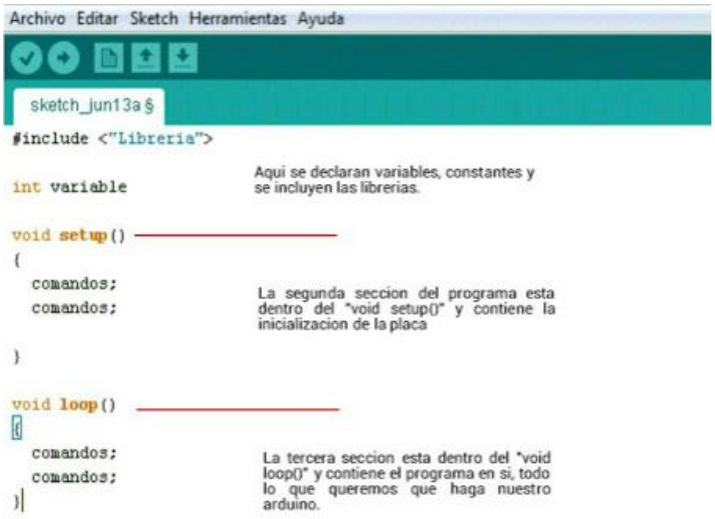


Figura 12 Estructura de programación

Fuente: (Evans & Ruiz Gutierrez, 2011)

2.2.6 Declaración de variables

Las variables tienen que definirse antes de ser utilizadas. Para declarar una variable se comienza por definir su tipo, asignándoles siempre un nombre, y opcionalmente un valor inicial, pero puede cambiar el mismo mientras se ejecuta el programa usando aritmética y reasignaciones diversas. (Colección Manuales Users)

2.2.7 Variables

Una variable es una manera de nombrar y almacenar un valor numérico para su uso posterior por el programa. Como su nombre indica, las variables son números que se pueden variar continuamente en contra de lo que ocurre con las constantes cuyo valor nunca cambia. Una variable debe ser declarada y, opcionalmente, asignarle un valor. (Arduino, Arduino Uno, 2016)

Ejemplo de uso de variables en un sketch:

Veamos el típico sketch que hace parpadear un LED activado a través de un pin:

```
int LEDpin = 6;   // la variable LEDpin se inicializa a 6, es decir vamos
                     activar el pin 6

void setup()
{
   pinMode(LEDpin,OUTPUT);

  }
void loop()

{
        digitalWrite(LEDpin,HIGH);
        delay (1000);
        digitalWrite(pinLED,LOW);
        delay (1000);
  }
```

2.2.8 Tipos de datos

En la siguiente tabla se determina que tipos de datos podemos usar en la la estructura de programación

Tabla 5

Tipos de datos

VARIABLES DE TIPOS PRIMITIVOS.					
Nombre	**Tipo**	**Tamaño**	**Valor por defecto**	**Forma de inicializar**	**Rango**
Boolean	Lógico	1 bit	False	Boolean a=true	True-false
Char	Carácter	16 bits	Null	Char a='Z'	Unicode
Byte	Numero entero	8 bits	0	Byte a =0	-128 a 127
Short	Numero entero	16 bits	0	Short a =12	-32.768 a 32.767
Int	Numero entero	32 bit	0	Int a= 1250	-2.147.483.648 a 2.147.483.649
Long	Numero entero	64 bits	0	Long a= 125000	-9*10^18 a 9*10^18
Float	Numero real	32 bits	0	Float a =3.1	-3,4*10^38 a 3,4*10^38
Double	Numero real	64 bits	0	Double a = 125.2333	-1,79*10^308 a 1,79*10^308

Fuente: (Arduino, Arduino Uno, 2016)

2.2.9 Constantes

El lenguaje de programación de Arduino palabras resevadas que se entiende como variables que no cambian de valor en todo el programa las cuales son:

INPUT/OUTPUT (Entrada/Salida).

Los pins digitales pueden ser configurados de ambos modos: como entrada (INPUT) o como salida

(OUTPUT)

Mediante la función pinMode(): pinMode(13, OUTPUT); // Configura el pin 13 como salida digital.

INPUT_PULLUP

Reservado como parámetro de la función pinMode() para el uso de resistencias pull-up integradas en el chip Atmega del Arduino.

LED_BUILTIN

Para el uso del Led de serie con el que viene equipado el Arduino (generalmente conectado al pin digital 13).

TRUE/FALSE (Verdadero/Falso).

Para el Arduino True (Verdadero) es cualquier valor que no es 0. False (Falso) es el valor 0.

HIGH/LOW (Alto/Bajo).

Es el valor lógico en una puerta digital: LOW es el valor 0 -0 Voltios- y HIGH es el valor 1 -5 Voltios-

2.2.10 Estructura setup( )

La estructura setup( ) se invoca una sola vez cuando el programa empieza. Se utiliza para inicializar los modos de trabajo de los pins, o el puerto serie. Debe ser incluido en un programa aunque no haya declaración que ejecutar. Así mismo se puede utilizar para establecer el estado inicial de las salidas de la placa. (Evans & Ruiz Gutierrez, 2011)

```
void setup()
{
pinMode(pin, OUTPUT);             // configura el 'pin' como salida
}
```

2.2.11 Estructura loop( )

Después de llamar a setup( ), la estructura loop( ) hace precisamente lo que sugiere su nombre, se ejecuta de forma cíclica, lo que posibilita que el programa este respondiendo continuamente ante los eventos que se produzcan en la placa. (Evans & Ruiz Gutierrez, 2011)

```
void loop()
{
```

```
digitalWrite(pin, HIGH);       // pone en uno (on, 5v) el´pin´
delay(1000);            // espera un segundo (1000 ms)
digitalWrite(pin, LOW);       // pone en cero (off, 0v.) el
delay(1000);             // ´pin´
}
```

2.2.12 Funciones

Una función es un bloque de código que tiene un nombre y un conjunto de instrucciones que son ejecutadas cuando se llama a la función. Son funciones setup( ) y loop( ) de las que ya se ha hablado. (Evans & Ruiz Gutierrez, 2011)

Las funciones de usuario pueden ser escritas para realizar tareas repetitivas y para reducir el tamaño de un programa. Las funciones se declaran asociadas a un tipo de valor. Este valor será el que devolverá la función, por ejemplo 'int' se utilizará cuando la función devuelva un dato numérico de tipo entero. Si la función no devuelve ningún valor entonces se colocará delante la palabra “void”, que significa “función vacía”. Después de declarar el tipo de dato que devuelve la función se debe escribir el nombre de la función y entre paréntesis se escribirán, si es necesario, los parámetros que se deben pasar a la función para que se ejecute. (Arduino, Arduino Uno, 2016)

```
tipo nombreFunción(parámetros)
{
instrucciones;
}
```

La función siguiente devuelve un numero entero, delayVal( ) se utiliza para poner un valor de retraso en un programa que lee una variable analógica de un potenciómetro conectado a una entrada de Arduino. (Arduino, Arduino Uno, 2016)

Al principio se declara como una variable local, ´v´ recoge el valor leído del potenciómetro que estará comprendido entre 0 y 1023, luego se divide el valor por 4 para ajustarlo a un margen comprendido entre 0 y 255, finalmente se devuelve el valor ´v´ y se retornaría al programa principal.

```
int delayVal()
{
int v;            // crea una variable temporal 'v'
v= analogRead(pot);         // lee el valor del potenciómetro
v /= 4;            // convierte 0-1023 a 0-255
return v;         // devuelve el valor final
}
```

2.2.13 Entradas/Salidas digitales

pinMode(pin, mode)

Esta instrucción es utilizada en la parte de configuración setup () y sirve para configurar el modo de trabajo de un PIN pudiendo ser INPUT (entrada) u OUTPUT (salida). (Arduino, Arduino Uno, 2016)

```
pinMode(pin, OUTPUT); // configura ‘pin’ como salida
```

Los terminales de Arduino, por defecto, estan configurados como entradas, por lo tanto no es necesario definirlos en el caso de que vayan a trabajar como entradas. Los pines configurados como entrada quedan, bajo el punto de vista electrico, como entradas en estado de alta impedancia. (Arduino, Productos de Arduino, 2016)

Estos pines tienen a nivel interno una resistencia de 20 KΩ a las que se puede acceder mediante software. Estas resistencias se acceden de la siguiente manera:

```
pinMode(pin, INPUT);            // configura el ‘pin’ como entrada
digitalWrite(pin, HIGH);        // activa las resistencias internas
```

Las resistencias internas normalmente se utilizan para conectar las entradas a interruptores. En el ejemplo anterior no se trata de convertir un pin en salida, es simplemente un método para activar las resistencias interiores. (Artero, 2013)

Los pines configurados como OUTPUT (salida) se dice que están en un estado de baja impedancia y pueden proporcionar 40 mA (miliamperios) de corriente a otros dispositivos y circuitos. Esta corriente es suficiente para alimentar un diodo LED (no olvidando poner una resistencia en serie), pero no es lo suficiente grande como para alimentar cargas de mayor consumo como relés, solenoides, o motores. (Artero, 2013)

Un cortocircuito en las patillas Arduino provocara una corriente elevada que puede dañar o destruir el chip Atmega. A menudo es una buena idea conectar en la OUTUPT (salida) una resistencia externa de 470 o de 1000 Ω. (Artero, 2013)

digitalRead(pin)

Lee el valor de un pin (definido como digital) dando un resultado HIGH (alto) o LOW (bajo). El pin se puede especificar ya sea como una variable o una constante (0-13).

```
valor = digitalRead(Pin);    // hace que 'valor sea igual al estado leído en ´Pin´
```

digitalWrite(pin, value)

Envia al ´pin´ definido previamente como OUTPUT el valor HIGH o LOW (poniendo en 1 o 0 la salida). El pin se puede especificar ya sea como una variable o como una constante (0-13).

```
digitalWrite(pin, HIGH);      // deposita en el 'pin' un valor HIGH (alto o 1)
```

2.2.14 Entradas/Salidas analógicas

analogRead(pin)

Lee el valor de un determinado pin definido como entrada analógica con una resolución de 10 bits. Esta instrucción sólo funciona en los pines (0-5). El rango de valor que podemos leer oscila de 0 a 1023.

valor = analogRead(pin); // asigna a valor lo que lee en la entrada ´pin'

analogWrite(pin, value)

A uno de los pines de Arduino marcados como “pin PWM”. El valor que se puede enviar a estos pines de salida analógica puede darse en forma de variable o constante, pero siempre con un margen de 0-255. (Artero, 2013)

analogWrite(pin, valor); // escribe 'valor' en el 'pin' definido como analógico (Artero, 2013)

Si se envía el valor 0 genera una salida de 0 voltios en el pin especificado; un valor de 255 genera una salida de 5 voltios de salida en el pin especificado. (Artero, 2013)

2.3 Transmisores

Los transmisores captan la variable de proceso a través del elemento primario y la transmiten a distancia en forma de señal neumática, electrónica, digital, óptica, hidráulica o por radio. (Solé, 2011)

2.3.1 Transmisor de temperatura TH-200

Los transmisores de temperatura SITRANS TH200 son aparatos con conexión a 2 hilos que se montan en perfiles. Gracias a sus entradas de sensor universales y al tipo de montaje, se pueden utilizar en cualquier sector. En su nivel de entrada se pueden conectar las siguientes fuentes de señales y sensor: termo resistencias, termopares, Emisores de resistencia / potenciómetros, fuentes de tensión continua. (Artero, 2013)

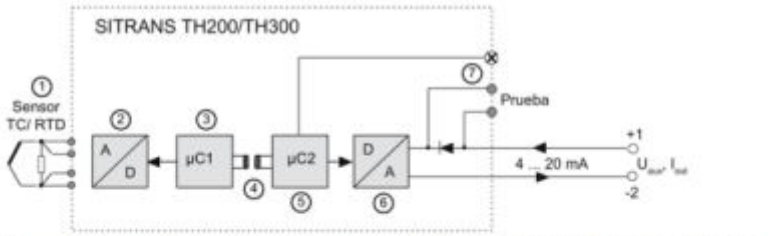
La señal de salida es una corriente de salida correspondiente a la curva característica de sensor, de entre 4 y 20 mA. (Siemens, 1995-01)

Figura 13 Transmisor de temperatura TH-200

Fuente: (Siemens, 1995-01)

2.3.2 Funcionamiento

El sensor emite una señal eléctrica. Esta señal es convertida por un convertidor analógico-digital en una señal digital. La señal digital es evaluada por un microcontrolador situado en el lado secundario y corregido en función de la curva característica del sensor. A través de la separación galvánica, la señal es transmitida al microcontrolador situado en el lado primario. En el microcontrolador del lado primario, se calcula el valor de salida analógico. El estado de funcionamiento se determina mediante LED y los datos de comunicación se procesan. Después, la señal es convertida en la corriente de salida de 4 a 20 mA por el convertidor digital-analógico. La fuente de energía auxiliar se encuentra en el circuito de la señal de salida. (Siemens, 1995-01)



① Sensores de tipo termorresistencia, termopar, emisor de resistencia, transmisor de milivoltios
② Convertidor analógico-digital
③ Microcontrolador secundario
④ Aislamiento galvánico
⑤ Microcontrolador primario
⑥ Convertidor digital-analógico
⑦ LED
U_{aux} Alimentación auxiliar
I_{out} Corriente de salida
Prueba Terminales de prueba para la conexión breve de un amperímetro

Figura 3-2 Diagrama de funciones del SITRANS TH200/TH300

Figura 14 Funcionamiento Transmisor TH-200

Fuente: (Siemens, 1995-01)

2.3.3 Transmisor Ultrasónico Banner S18UIA

Este Transmisor tiene el mismo funcionamiento que un radar ya que envía pulsos a alta frecuencia en este caso ultrasónicos. Está compuesto por dos piezoeléctricos: un transmisor y receptor ambos en forma de cilindro. El transmisor se encarga de emitir la señal ultrasónica para luego ser rebotada a través de un objeto llegando al cilindro receptor. Debido a que la señal le tardará un tiempo en regresar desde que se emite, es obvio pensar que la velocidad con la cual llegará al receptor está íntimamente relacionada con la distancia de transmisión y rebote. (Banner, 2016)

Figura 15 Transmisor ultrasónico Banner S18UIA

Fuente: *(Banner, 2016)*

2.4 Softaware Labwiev

LabVIEW es un entorno de desarrollo integrado y diseñado específicamente para ingenieros y científicos que desarrollan sistemas de medidas y control. Con un lenguaje de programación gráfica nativo, IP integrado para análisis de datos y procesamiento de señales y una arquitectura abierta que permite la integración de cualquier dispositivo de hardware y cualquier enfoque de software (Instruments, 2017)

2.4.1 Características

Su entorno de programación a diferencia de los lenguajes comunes es totalmente gráfico facilitando su comprensión, al tener prediseñados bloques

de programa su desarrollo se hace más amigable para la creación de programas,

Su entorno facilita y reduce el tiempo que invierte un desarrollador en realizar un programa, optimizando así más tiempo para la interfaz gráfica de interacción con el usuario. Cada VI consta de dos paneles de interacción diferenciadas consideradas de la siguiente manera: (Instruments, 2017)

2.4.1.1 Panel Frontal

Cuando se abre un VI nuevo o existente, aparece la ventana del panel frontal del VI. La ventana del panel frontal es la interfaz de usuario para el VI. La **Figura 16** muestra un ejemplo de una ventana del panel frontal.

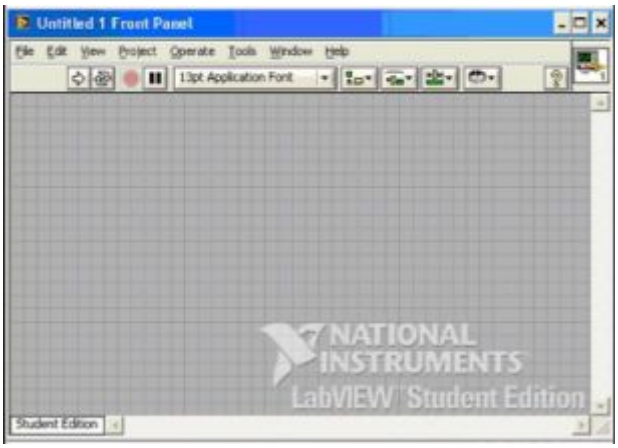


Figura 16 Panel Frontal VI

Fuente: (Instruments, 2017)

2.4.1.2 Diagrama de bloques

Este es el programa propiamente dicho donde se define su funcionalidad, aquí se crea bloques de programa en conjunto con funciones y operaciones aritméticas que se crean para un determinado objetivo.

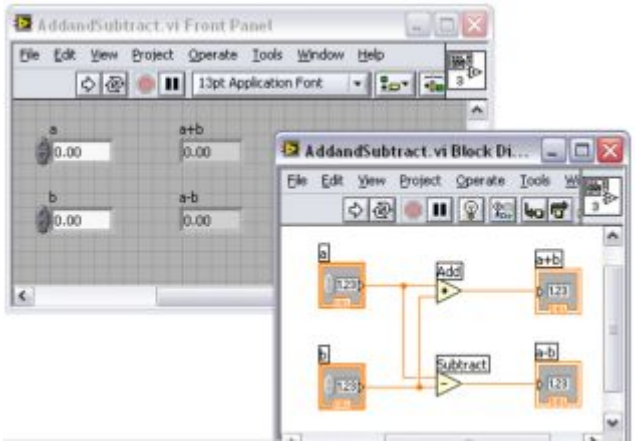


Figura 17 Bloque de Programa

Fuente: (Instruments, 2017)

2.4.1.3 Aplicaciones

NI LabVIEW lo último en software de diseño de sistemas usado por ingenieros y científicos para diseñar, generar prototipos y desplegar aplicaciones embebidas de control y monitoreo de manera eficiente. Combina cientos de bibliotecas pre escritas, estrecha integración con hardware comercial y una variedad de enfoques de programación incluyendo desarrollo gráfico, scripts de archivos .m y conectividad a código ANSI C y HDL existente. Ya sea para diseñar dispositivos médicos o robots complejos, usted puede reducir el tiempo al mercado y los costos totales del diseño monitoreo y control embebidos con LabVIEW. (Instruments, 2017)

CAPÍTULO III

DESARROLLO DEL PROYECTO

3.1 Preliminares

Para la implementación del proyecto técnico cuyo tema es: **IMPLEMENTACION DE UN MONITOREO INALAMBRICO DE TRANSMISORES DE SALIDA ANALOGICA, EMPLEANDO TECNOLOGIA XBEE PARA PRACTICAS DE REDES INDUSTRIALES,** fue necesario lo siguiente:

- Módulos Xbee SB2
- Módulo Xbee Explorer Spark fun
- Modulo Explorer regulado
- Arduino UNO
- Computador o laptop
- Software XCTU
- Software Aruduino IDE
- Software LabWiev
- Transmisor de Temperatura TH200
- Transmisor de Banner S18UIA
- Resistencias
- Fuente de alimentación de 9 VDC
- Fuente de alimentación de 12 VDC
- Protoboard

3.2 Requerimiento de Hardware

Los requerimientos de hardware que se utilizaron para el proyecto son dos módulos Xbee S2, puesto que permiten comunicarse con el arduino Uno por tener un puerto serie, uno en la transmisión y el otro en la recepción de datos marcados en los pines Rx y Tx de las placas.

En vista que se requiere que uno de los Xbee S2 se comunique con la Pc, se hizo necesario utilizar el shield Xbee Spark Fun Explorer que es un

convertidor de USB a serie, además tiene un regulador de voltaje a fin de regular la tensión del puerto usb de 5V a 3.3 V, esto en la etapa de recepción.

En la etapa de Transmisión se utilizó el Shield Xbee Explorer regulado, el cual fue diseñado para acoplarse en el protoboard, se utilizó como regulador de voltaje de 5V a 3.3V, para el acondicionamiento de señal, así como los indicadores de actividad básica.

3.3 Requerimiento de Software

3.3.1 Configuración de los Módulos Xbee

Una vez que está instalado el software XCTU, conecte en el módulo Xbee S2 sobre la placa del módulo Xbee explorer como se muestra en la figura 18, y realice la conexión al puerto USB del computador. Después de algunos segundos se encenderá el led indicador del Módulo Xbee Explorer el mismo que indicará que está listo para configuración.

Figura 18 Conexión Shield Xbee y Xbee S2

De inmediato se debe escoger en la opción de búsqueda de dispositivos para que se actualice el puerto que asigna el software.

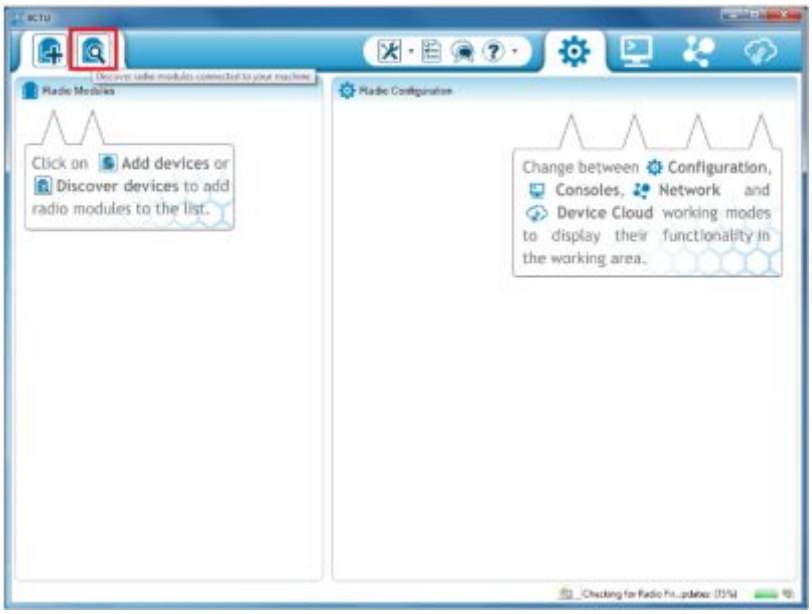


Figura 19 Configuración inicial Xbee

Una vez que reconoce el puerto de enlace aparecerá un cuadro de dialogo mostrando los dispositivos encontrados, seleccionar los puertos encontrados y escoger la opción NEXT como se muestra en la figura 20 de inmediato se aparecerán las opciones de configuración de los dispositivos.

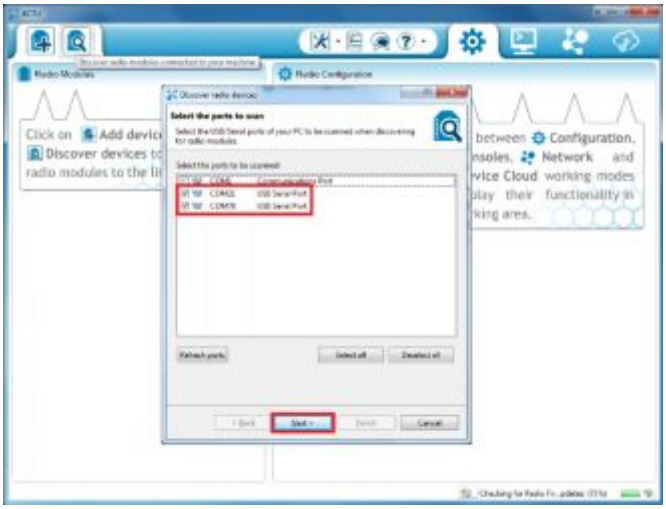


Figura 20 Reconocimiento de dispositivos

Al dar clic en el botón NEXT, aparece una ventana donde están especificados los parámetros que se necesitan para la configuración.

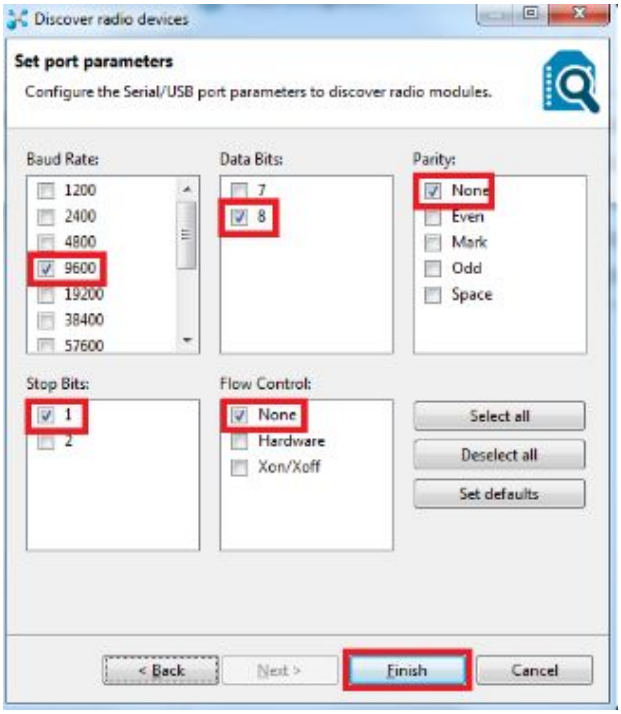


Figura 21 Parámetros de configuración

Una vez realizada la configuración seleccionar la opción FINISH.

3.3.2 Comprobación de comunicación

Después de la configuración seleccionar el icono de consola ubicado en la parte superior derecha de la pantalla, un vez en ese cuadro de dialogo seleccionamos la opción Open que inicializa la transmisión de datos los mismo que se visualizan de manera Hexadecimal como se muestra en la Figura 22.

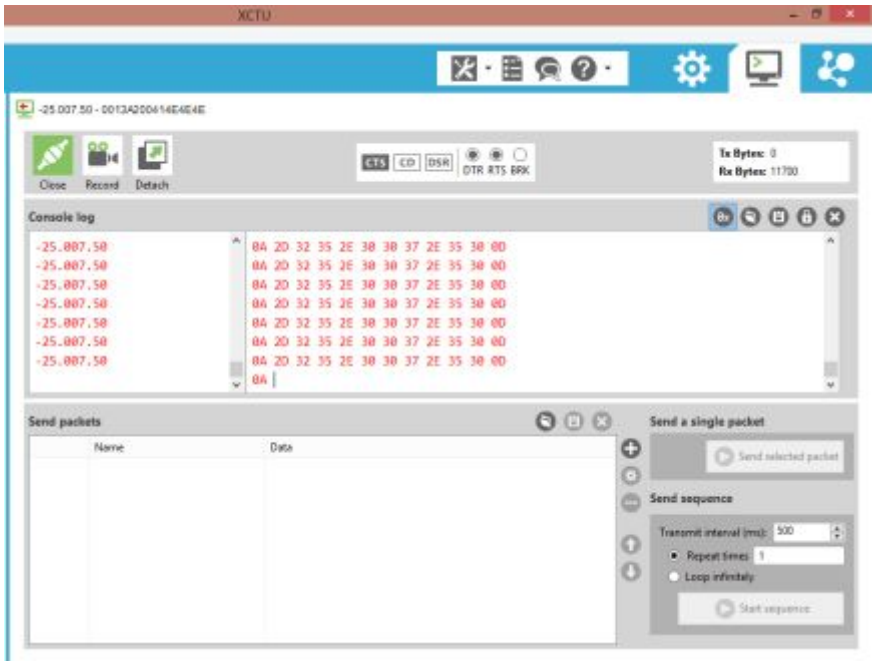


Figura 22 Comunicación entre dispositivos

3.3.3 Programación del Arduino Uno

Instala el IDE de Arduino, abre un nuevo Scketch

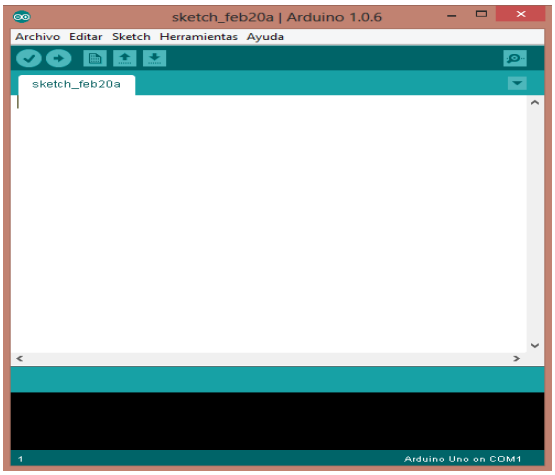


Figura 23 Scektch Arduino IDE

En la barra de Opciones selecciona la opción Herramientas, se desplaza hacia la opción tarjetas y se escoge Arduino UNO

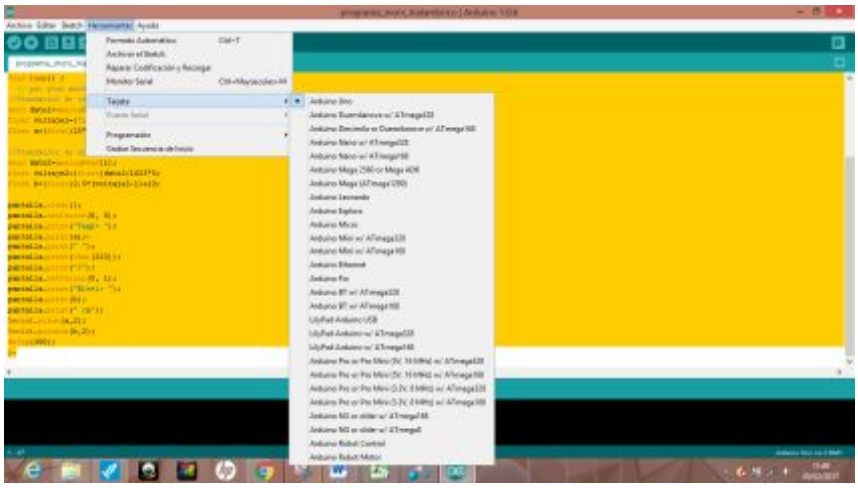


Figura 24 Selección de la tarjeta Arduino UNO

El programa que se carga al Arduino UNO es el siguiente:

Se escribe el siguiente código:

```
#include <LiquidCrystal.h>
LiquidCrystal pantalla(12, 11, 5, 4, 3, 2);
                    //        (RS,E,D4,D5,D6,D7)


void setup() {
  // Se Ejecuta una vez
pantalla.begin(16,2);
Serial.begin(9600);
}


void loop() {
  // se hace el bucle repetitivo:
//Transmisor de temperatura TH 200
word dato1=analogRead(0);
```

```
float voltaje1=(float)dato1/1023*5;
float a=(float)25*(voltaje1-1);

//Transmisor de nivel S18UA
word dato2=analogRead(1);
float voltaje2=(float)dato2/1023*5;
float b=(float)2.5*(voltaje2-1)+10;

pantalla.clear();
pantalla.setCursor(0, 0);
pantalla.print("Temp= ");
pantalla.print(a);-
pantalla.print(" ");
pantalla.print(char(223));
pantalla.print("C");
pantalla.setCursor(0, 1);
pantalla.print("Nivel= ");
pantalla.print(b);
pantalla.print(" cm");
Serial.print(a,2);
Serial.println(b,2);
delay(300);
}-
```

Crear una instancia llamada lcd, de la clase LiquidCrystal y pasar como parámetros los pines que se ha usado:

LiquidCrystal pantalla(12, 11, 5, 4, 3, 2);

// (RS,E,D4,D5,D6,D7)

En esta sentencia se asignan los pines donde se debe conectar físicamente el LCD al Arduino como se muestra en la figura 26

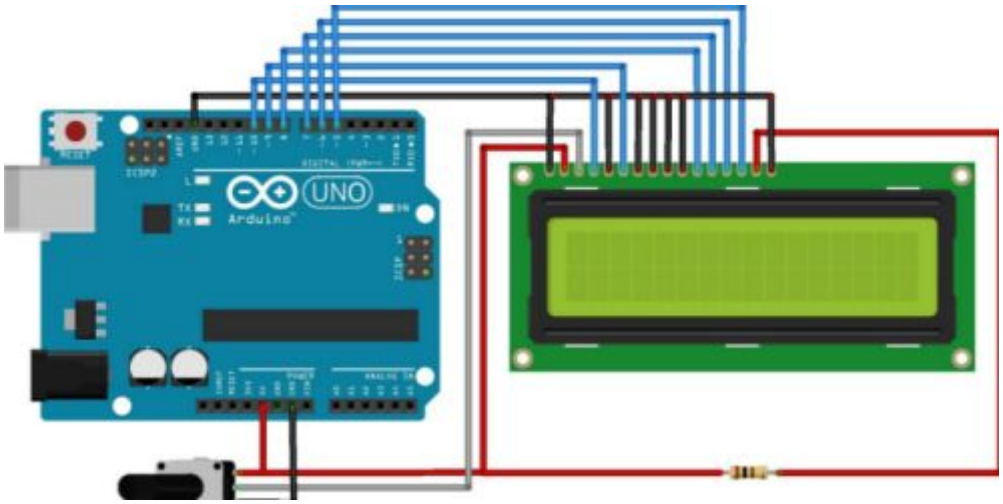


Figura 25 Conexión LCD- Arduino

A continuación en la línea de código se declara el formato al que debe enviar los datos para imprimir, en este caso se utiliza el LCD 16x2

pantalla.begin(16,2)

Luego se declara la velocidad de transmisión con la cual serán enviados los datos en la siguiente línea de código.

Serial.begin (9600);

En las líneas siguientes se almacena en la variable word dato1 los datos que recibe a través de los pines se señal analógica y lo convierte en dato flotante en este caso se realiza todo el acondicionamiento de la entrada en el mismo lugar que es dividir el valor la entrada analógica por 5V que es el fondo de escala y multiplicarlo por los 1023 pasos de resolución que dispone el microcontrolador Atmega 328 del arduino, así releja un valor de tensión de 0 a 5V se debe tomar en cuenta que el mismo proceso se hace en los dos transmisores de nivel.

El Arduino posee 6 entradas analógicas, etiquetadas desde la A0 a A5, cada una tiene 10 bits de resolución (es decir, 1024 estados). Por defecto, se tiene una tensión de 5V, las entradas analógicas leen un valor entre 0 y 1023 cuando en los terminales de la tarjeta existe un voltaje cuyo rango es de 0 a 5 VDC.

Figura 26 Entradas analógicas tarjeta Arduino UNO

```
//Transmisor de temperatura TH 200
word dato1=analogRead(0);
float voltaje1=(float)dato1/1023*5;
float a=(float)25*(voltaje1-1);

//Transmisor de nivel S18UA
word dato2=analogRead(1);
float voltaje2=(float)dato2/1023*5;
float b=(float)2.5*(voltaje2-1)+10}
```

y por ultimo de realiza la impresión en el LCD, de acuerdo al siguiente código se imprime la visualización directa de los transmisores de nivel y de temperatura, todo esto se encuentra dentro del Void Loop como su nombre indica, ésta parte se ejecuta una y otra vez. En un programa de Arduino todo el código se ejecuta línea a línea. Después de ejecutar la void setup() en el arranque continua con la void loop() con una repetición cíclica de 300milisegundos como se muestra en la línea de programa delay(300)

```
pantalla.clear();
pantalla.setCursor(0, 0);
pantalla.print("Temp= ");
pantalla.print(a);-
pantalla.print(" ");
pantalla.print(char(223));
pantalla.print("C");
pantalla.setCursor(0, 1);
pantalla.print("Nivel= ");
pantalla.print(b);
pantalla.print(" cm");
Serial.print(a,2);
Serial.println(b,2);
delay(300);
}
```

3.3.4 Envío de datos Arduino Uno- Xbee S2

Toda esta información está siendo enviada por el pin1 TX del Arduino UNO hacia el pin RX del Xbee s2 como se muestran las conexiones de la figura 27.

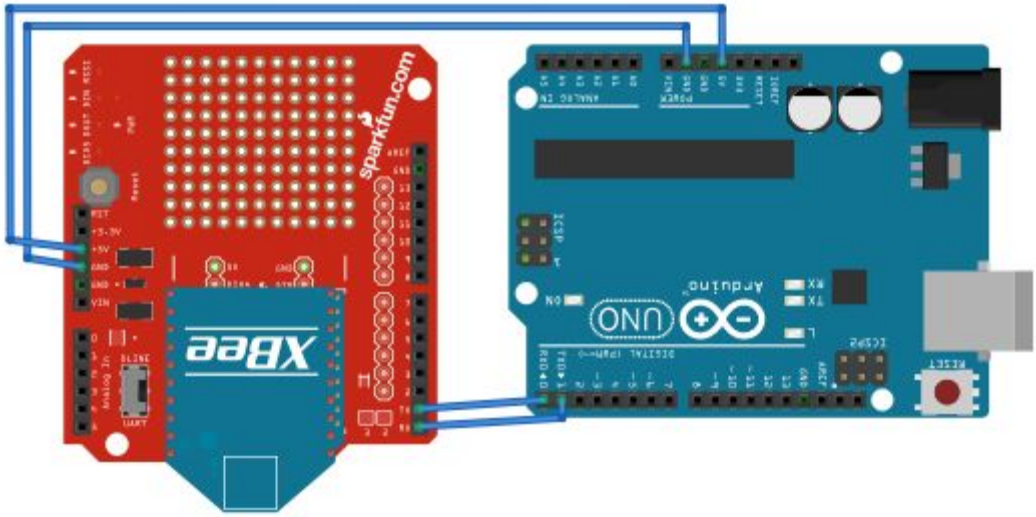


Figura 27 Conexión Arduino UNO- Xbee S2

De inmediato se conecta el otro Xbee S2 al puerto USB para poder interconectar

Figura 28 Conexión Xbee-puerto USB

Ahora se debe inicializar el software LABWIEV, para realizar la configuración y programación de adquisición de los datos que se obtienen de

la comunicación inalámbrica que se está generando entre los modulos Xbee S2

3.3.5 Adquisición de datos en mediante el software Labwiev

Para realizar la adquisición de datos se debe realizar la configuración del puerto que se va a utilizar de medio de transmisión de información para el efecto, inicializar LabView, y crear un nuevo VI como se muestra en la figura29.

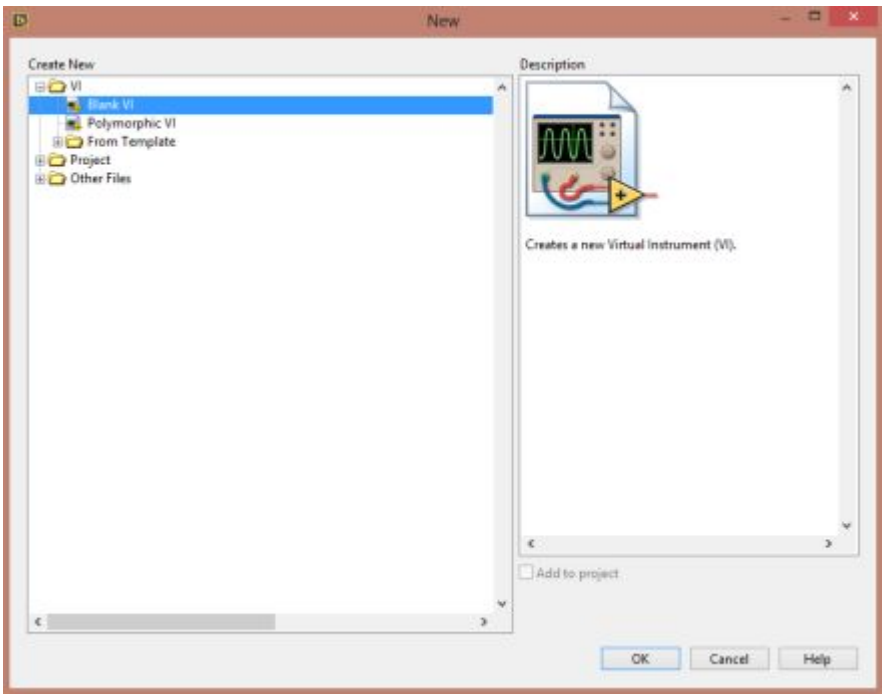


Figura 29 Creación de VI

Se despliega dos pantallas el diagrama de bloque y panel frontal; para tener acceso al puerto serial usando labview se debe iniciar una sesión VISA. La configuración del tipo de comunicación serial se hace con “VISA configure serial port”, que se puede encontrar en Functions >> Instrument I/O >> Serial >> VISA configure serial port. Como se muestra en la figura 30.

En el diagrama de bloque colocar el instrumento virtual Visa que se requiere para acceder al puerto serial, por el que se adquiere los datos análogos.

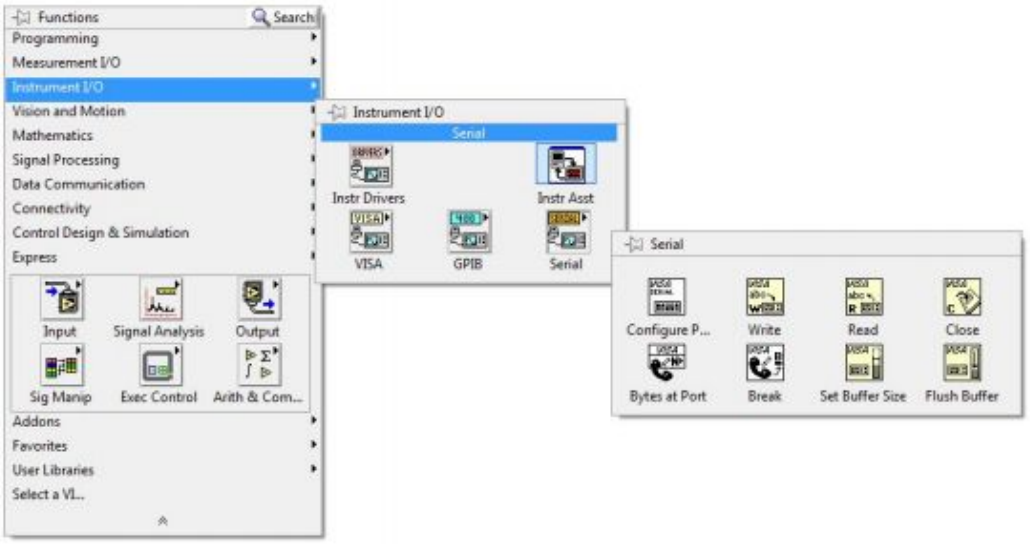


Figura 30 Configuración del puerto serial

Se selecciona la opción de configuración, la misma que se despliega en la pantalla de diagrama de bloques el siguiente control

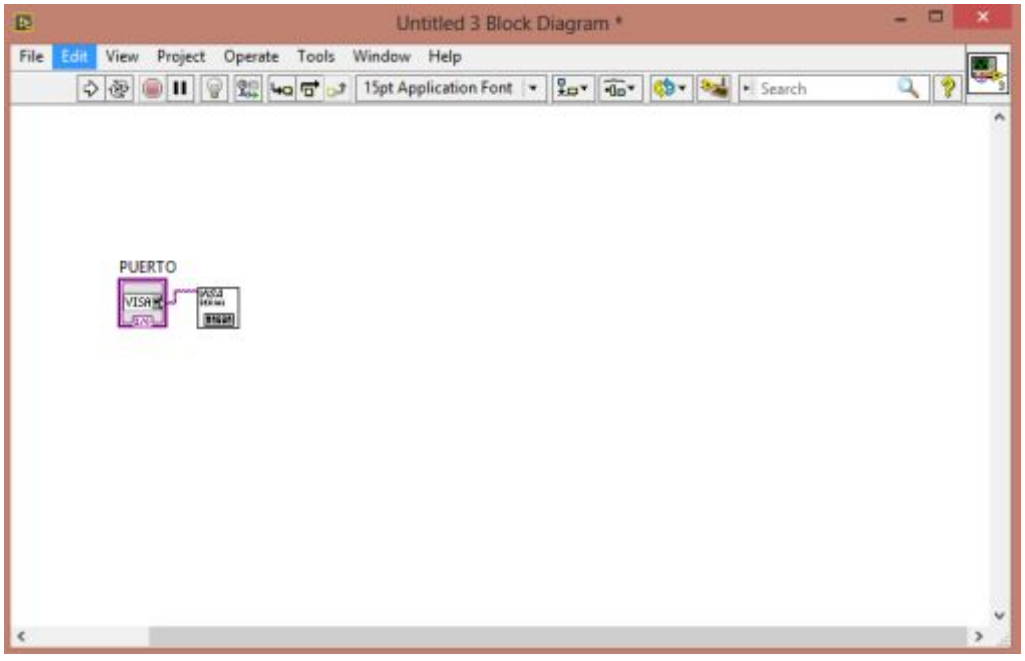


Figura 31 Control puerto serial

Sobre el control con el mouse se da click derecho y le aparecen varias opciones de las cuales se escoge la opción Create, de inmediato se despliega una ventana pequeña con las opciones como se muestra a continuación.

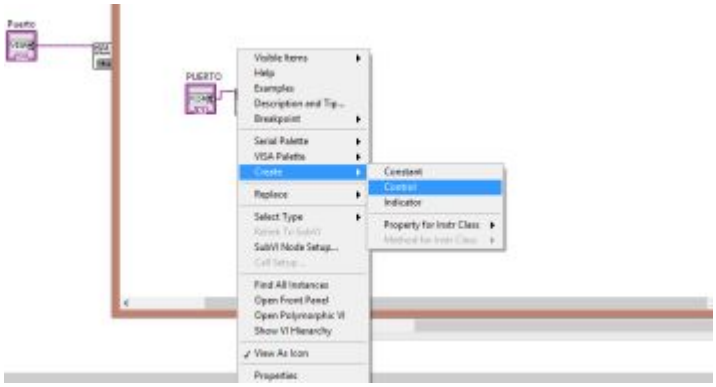


Figura 32 Creación de control

Para que la configuración se encuentre completa se agrega los bloques de lectura denominado Read y de cierre denominado CLOSE para que se pueda completar el ciclo de recepción de Datos, queda entonces la configuración como se observa en la figura 33.

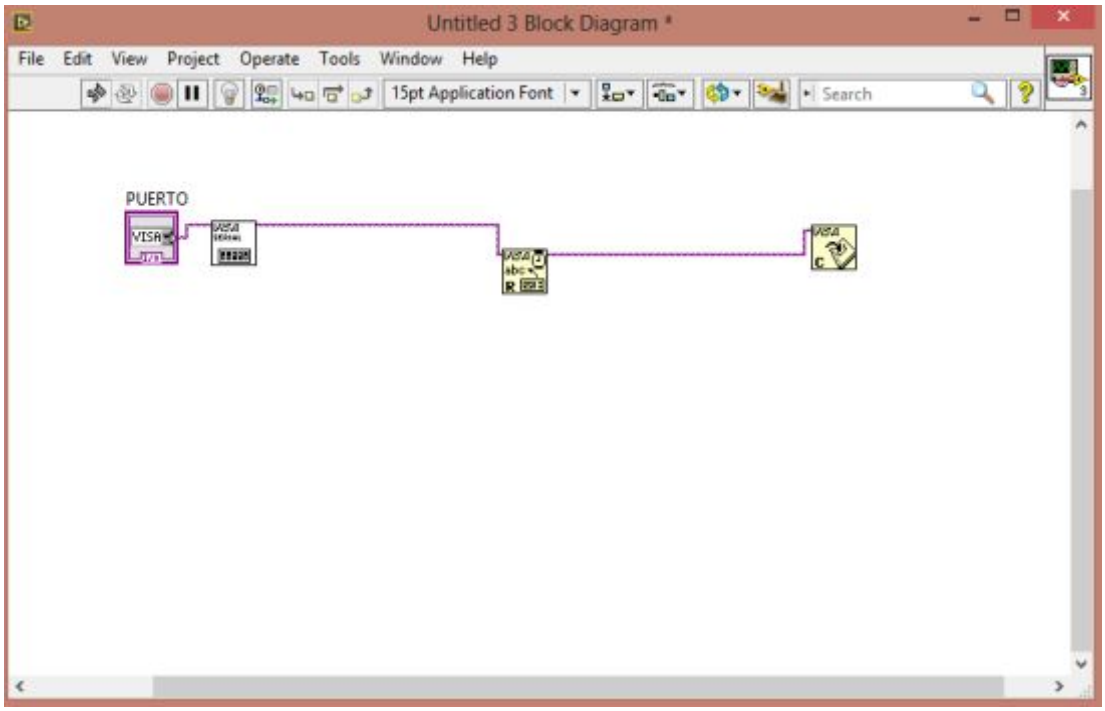


Figura 33 Configuración control Visa

3.3.6 Programación y procesamiento de datos en Labwiev

Se crea el siguiente diagrama de bloques para la lectura y proceso de los datos

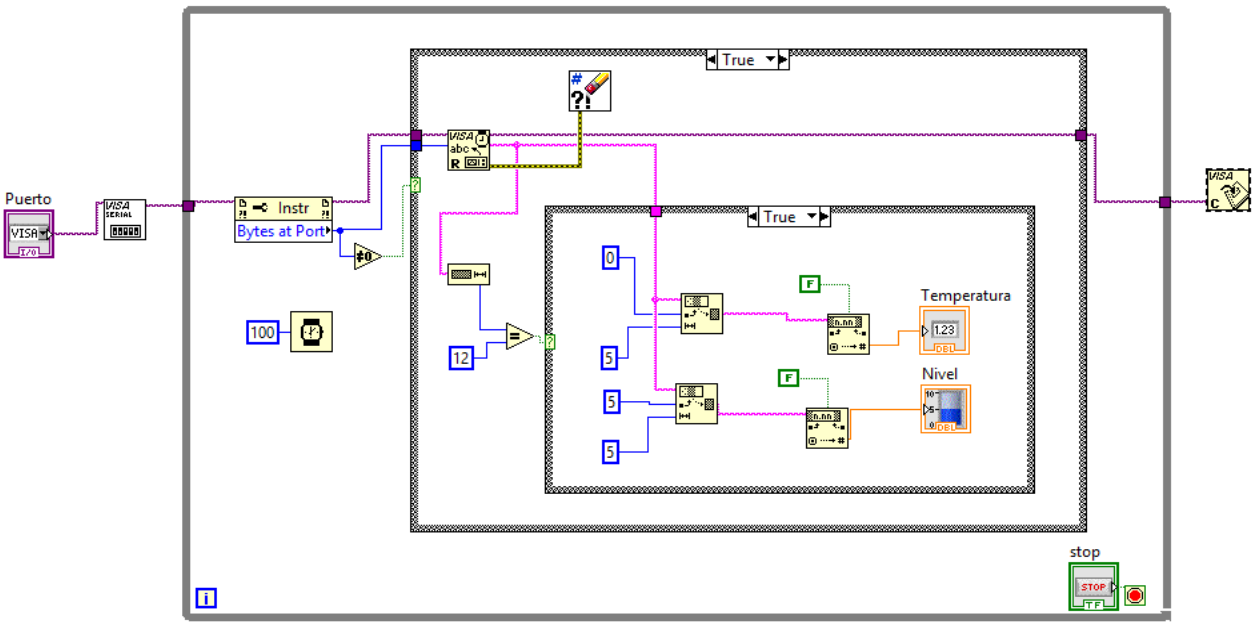


Figura 34 Diagrama de bloque de adquisición y proceso de datos

Para la explicación se utiliza zonas sombreadas que servirá como ayuda para identificar cada proceso

3.3.6.1 Ingreso de datos por el puerto serial

Dentro de la zona sombreada en la figura 35, se ha insertado la estructura While loop es similar a un Ciclo Do o a un Ciclo Repeat-Until en lenguajes de programación basados en texto, un ciclo while, ejecuta el código que contiene hasta que ocurre una condición.

La estructura While loop contiene a todo el bloque de programa en su primer nivel en la zona sombreada se puede observar que encierra al control de bytes at Port ( ), este es parte de la adquisición de los datos, recibe los bytes que ingresan por el puerto en su totalidad sin embargo para tener control de la información que recibe se insertó un comparador para que

solo ingresen al proceso todos los números que sean diferente de cero, dentro de la misma estructura se encuentra un timer para que el proceso tome los datos cada 100 milisegundos.

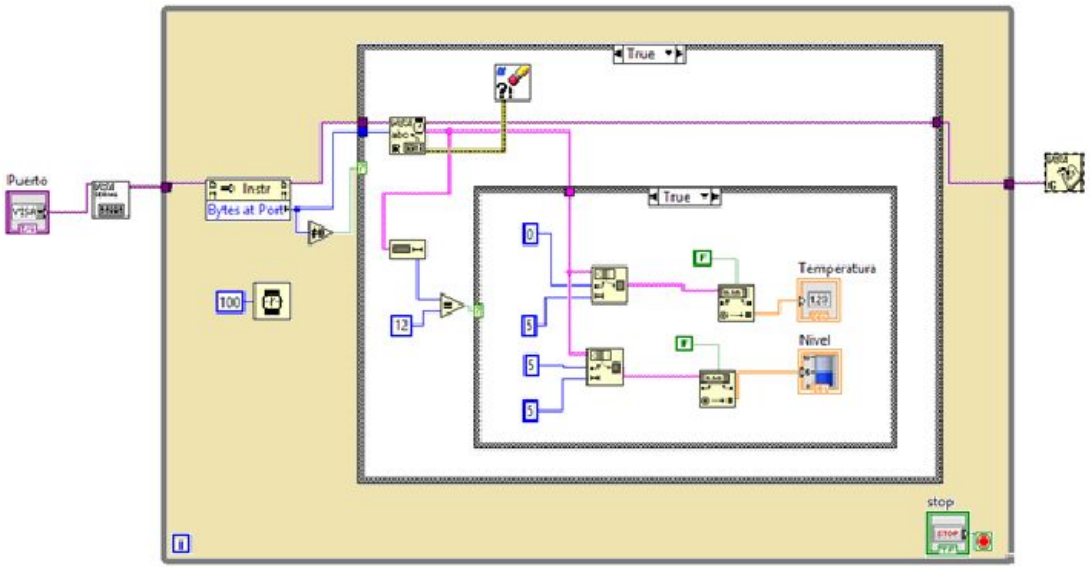


Figura 35 Bloque de programación While loop

La estructura While loop se encuntra dentro de la paleta de funciones>>structure como se puede observar en la figura 36.

Figura 36 Ubicación estructura While Loop

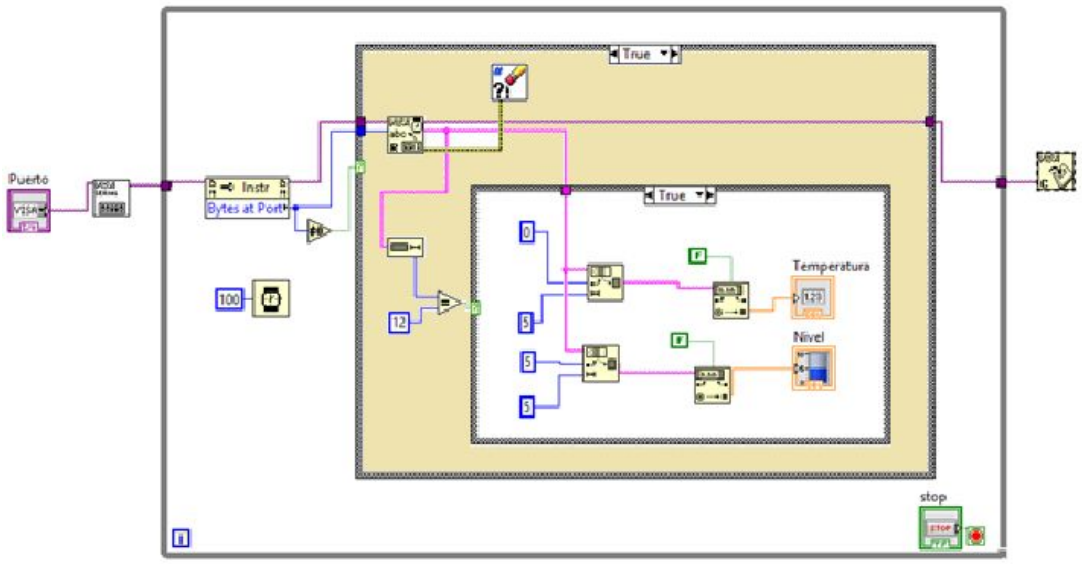


Figura 37 Programación Estructura Case

Dentro de la estructura Case se encuentra validado los datos que ingresan después de haber sido comparados y validados solo los números diferentes de cero con la condición verdadero sigue el proceso, caso contrario regresa a la parte de validación, los datos son leídos por el bloque Visa read la cual está conectado un control de borrador de errores que elimina los espacios en blanco, la salida del bloque es enviada a un control String y comparada con una constante 12, esto pasa a otro Case.

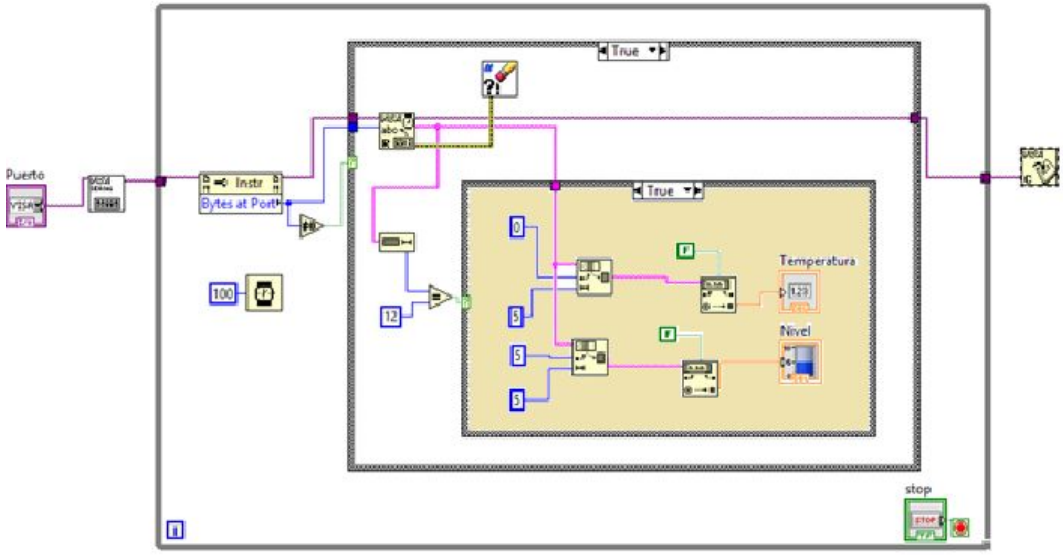


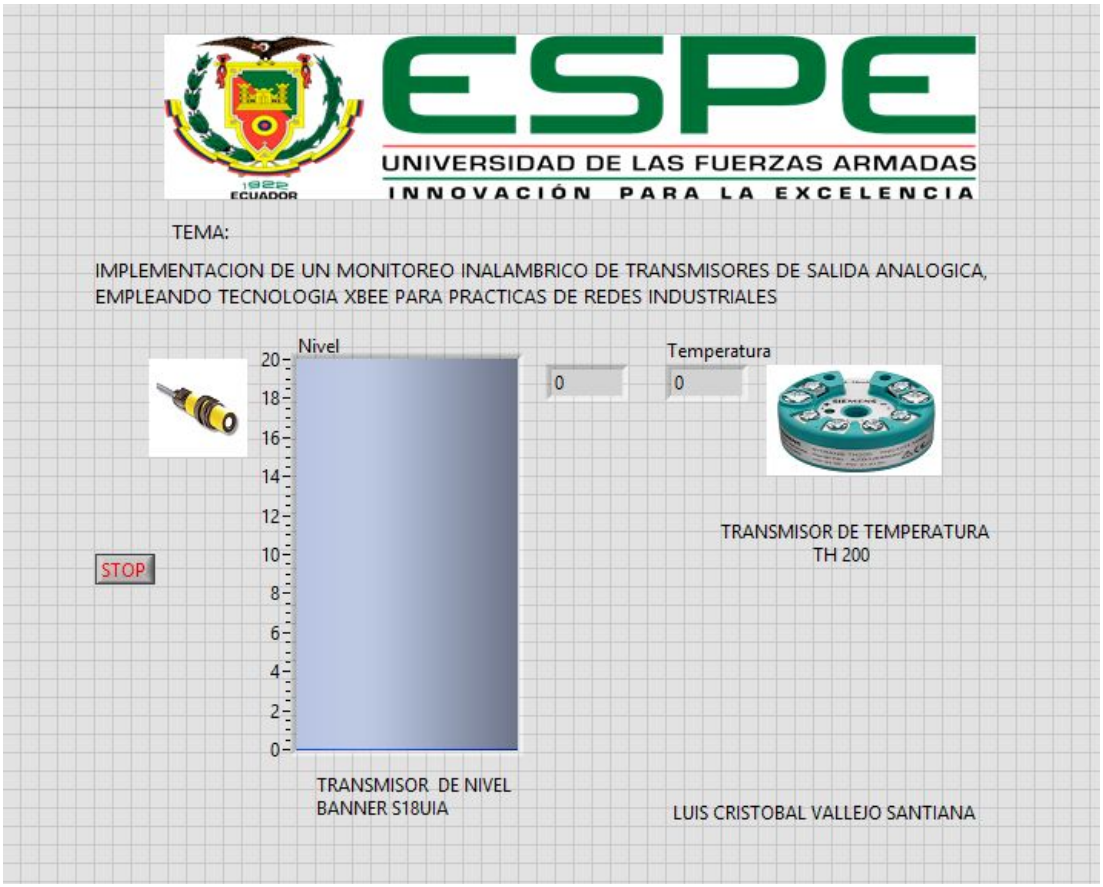
Figura 38 Programación case anidado

En este último segmento del bloque de programación se encuentra una estructura Case dentro de otra estructura case, se puede decir que en lenguaje de programación estructurado sería similar a un case anidado, dentro de este ingresan solo valores que tengan la longitud de texto de 12 espacios.

Adicional aquí se realiza la conversión del dato string a decimal, en este segmento se encuentra los controles que sirven como interfaz del usuario.

3.4 Monitoreo de señales analógicas con Labwiev

La aplicación del monitoreo inalámbrico mediante Xbee consiste en crear una interface mediante diagrama de bloques para monitorear y controlar señales analógicas. Se emplea la comunicación inalámbrica mediante tecnología Xbee.



3.4.1 Pruebas de funcionamiento

Para comprobar el funcionamiento del monitoreo inalámbrico, siga los siguientes pasos.

Paso 1.

Energice el Arduino Uno y conecte Xbee al puerto serial del computador

Paso 2.

Energice los transmisores analógicos de nivel y de temperatura

Paso 3.

Abra el software Labwiev y Ejecute el programa con el nombre Interface que se encuentra el disco

Paso 4.

De clic en la opción RUN del interface Labwiev para iniciar con el Monitoreo

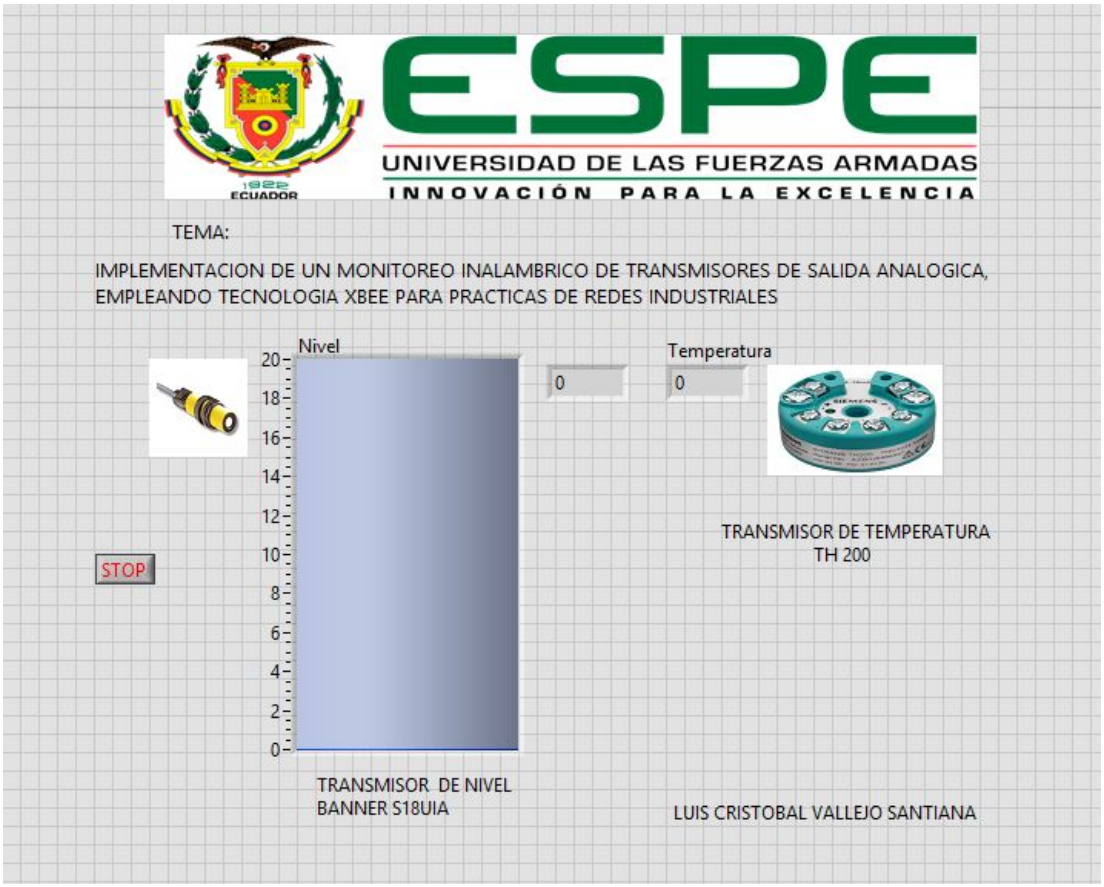


Figura 39 Interface para el monitoreo de señales

CAPÍTULO IV

CONCLUSIONES

4.1 Conclusiones

- El Arduino UNO es una tarjeta electrónica que posee un microcontrolador Atmel, que proporcionó la interacción con los módulos xbee con gran facilidad
- Para realizar la configuración de los módulos Xbee fue necesario contar con el Software XCTU.
- Los módulos Xbee no tienen regulador de voltaje y los pines son muy pequeños para la conexión al protoboard, para lo cual fue necesario recurrir a los shield de fabricación Digi.
- Al adquirir señales analógicas mediante el Arduino UNO se deben procesar y considerar el rango de voltaje que es de 0 a 5 VDC.
- Se debe instalar las librerías CDM21242 que sirve para crear el puerto virtual que es la interconexión con el computador
- Se realizó la comunicación y las conexiones de las placas de arduino se variaron las condiciones de las entradas análogas y se pudo visualizar el monitoreo en software labview

4.2 Recomendaciones

- Los Módulos Xbee deben ser alimentados con 3.3 V, ya que no poseen regulador interno
- Se debe interconectar los módulos Xbee, con los shields de acuerdo a la configuración que se necesite
- La investigación de la tecnología inalámbrica es muy importante, de tal manera que es sustancial explorar con este tipo de tecnología

GLOSARIO DE TÉRMINOS

LED	Diodo emisor de luz
TX	Transmisión de Datos
RX	Recepción de Datos
IEEE	Instituto de ingenieros eléctricos y electrónicos
IP	Protocolo de internet
REST	Transferencia de estado representacional
SSH	Intérprete de órdenes seguro
WiFi	Tecnología de comunicación inalámbrica

REFERENCIAS BIBLIOGRÁFICAS

Arduino. (2016). *Arduino Uno*. Obtenido de http://www.arduino.cc/en/Main/ArduinoBoardYun

Arduino. (2016). *Productos de Arduino*. Obtenido de http://www.arduino.cc/en/Main/Products

Artero, Ó. T. (2013). *Arduino: curso práctico de formación. RC Libros.*

Banner. (2016). *bannerengineering*. Obtenido de http://www.bannerengineering.com/

Colección Manuales Users, M. p. (s.f.). *Microcontroladores.*

DIGI. (2016). Recuperado el FEBRERO de 2017, de https://www.digi.com/products/xbee-rf-solutions/embedded-rf-modules-modems/digi-xbee-zigbee

Evans, B., & Ruiz Gutierrez, J. (2011). *Arduino Programming Notebook.* San Francisco California.

Gislason. (2008).

Instruments, N. (2017). *NI*.

Siemens. (1995-01). *SITRANS TR200/TR300 .*

Solé, A. C. (2011). *INSTRUMENTACION INDUSTRIAL.* España: AlphaOmega.

ANEXOS