



**UNIVERSIDAD NACIONAL DE CHIMBORAZO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA ELECTRONICA Y  
TELECOMUNICACIONES**

**“Trabajo de grado previo a la obtención del Título de Ingeniera en  
Electrónica y Telecomunicaciones”**

**TRABAJO DE GRADUACIÓN**

**DISEÑO E IMPLEMENTACIÓN DE UNA RADIOSONDA DE  
TRANSMISIÓN DE PARÁMETROS METEOROLÓGICOS PARA EL  
“PROGRAMA INTEGRADO DE MONITOREO DE CONTROL DE  
CALIDAD DEL AIRE EN LA CIUDAD DE RIOBAMBA USANDO  
MODELOS MATEMÁTICOS”.**

**Autor:** Geomara Viviana Pilco Ninabanda

**Director:** Ing. Giovanni Cuzco

**Riobamba – Ecuador**

**2016**

## PÁGINA DE REVISIÓN

Los miembros del Tribunal de Graduación del proyecto de investigación de título: Diseño e implementación de una radiosonda de transmisión de parámetros meteorológicos para el “Programa integrado de monitoreo de control de calidad del aire en la ciudad de Riobamba usando modelos matemáticos”, presentado por: Geomara Viviana Pilco Ninabanda y dirigida por: Ing. Giovanni Cuzo.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

----- <b>Ing. Paulina Vélez</b>	 ----- <b>Firma</b>
----- <b>Ing. Giovanni Cuzco</b>	 ----- <b>Firma</b>
----- <b>Ing. Cristian Rocha</b>	 ----- <b>Firma</b>

## **CERTIFICACIÓN DEL TUTOR**

Certifico que el presente trabajo de investigación previo a la obtención del título de Ingeniera en Electrónica y Telecomunicaciones. Con el tema: Diseño e implementación de una radiosonda de transmisión de parámetros meteorológicos para el “Programa integrado de monitoreo de control de calidad del aire en la ciudad de Riobamba usando modelos matemáticos” ha sido elaborado por Geomara Viviana Pilco Ninabanda, el mismo que ha sido revisado y analizado en un cien por ciento por el asesoramiento de mi persona en calidad de Director, por lo que se encuentra apta para su presentación y defensa respectiva. Es todo cuanto puedo informar en honor a la verdad.



Ing. Giovanni Cuzco  
Tutor del Proyecto de Investigación

## **AUTORÍA DE LA INVESTIGACIÓN**

“La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: Geomara Viviana Pilco Ninabanda e Ing. Giovanni Cuzco; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo”.



---

**Geomara Viviana Pilco Ninabanda**  
**C.I. 060413070-8**

## **AGRADECIMIENTO**

A las autoridades y al personal docente de la Escuela de Ingeniería en Electrónica y Telecomunicaciones quiero dejar un testimonio de gratitud y reconocimiento por su ardua labor desplegada en beneficio de la educación, al Ing. Giovanni Cuzco por la profunda calidad humana que supo compartir conmigo y de manera muy especial a mi familia por el apoyo que he recibido en el transcurso de mi vida.

## **DEDICATORIA**

Este trabajo les dedico a mis padres por el apoyo incondicional que me brindaron para la realización de un bello sueño, que con trabajo, con perseverancia sembraron en mi corazón la semilla de la responsabilidad, la voluntad, el amor y el respeto para alcanzar los objetivos propuestos

## INDICE GENERAL

PÁGINA DE REVISIÓN.....	I
AUTORÍA DE LA INVESTIGACIÓN.....	II
AGRADECIMIENTO .....	III
DEDICATORIA .....	IV
INDICE DE TABLAS .....	VIII
INDICE DE ILUSTRACIONES.....	IX
RESUMEN.....	XI
SUMMARY .....	XII
INTRODUCCIÓN .....	1

### CAPITULO I

1. MARCO TEÓRICO.....	2
1.1 METEOROLOGIA .....	2
1.1.1 RADIOSONDA .....	3
1.2 SENSORES.....	3
1.2.1 CARACTERÍSTICAS DE LOS SENSORES .....	3
1.2.1.1 Características Estáticas .....	4
1.2.1.2 Características Dinámicas .....	5
1.2.2 CLASIFICACIÓN DE LOS SENSORES .....	5
1.2.3 SENSORES DE TEMPERATURA.....	6
1.2.4 SENSORES DE HUMEDAD.....	6
1.2.4.1 Sensor de humedad relativa y temperatura SHT75.....	7
1.2.5 SENSORES DE PRESION ATMOSFERICA .....	8
1.2.5.1 Sensor de presión atmosférica BMP085 .....	9
1.2.6 SENSORES DE VELOCIDAD DEL VIENTO .....	10
1.2.6.1 Sensor de velocidad del viento MD0550 .....	10
1.3 ARDUINO .....	12
1.3.1 Arduino UNO.....	17
1.3.1 ENTORNO DE DESARROLLO DE ARDUINO .....	13
1.3.1.1 Inicio del IDE de Arduino.....	13
1.3.1.2 Opciones de la interfaz.....	16
1.4 TECNOLOGÍA ZIGBEE.....	18

1.4.1 Tipos de Dispositivos.....	18
1.4.2 Módulos RF Xbee-868MHZ.....	19
1.5 LABVIEW .....	20
1.5.1 Requerimientos del sistema.....	20
1.5.2 Ventajas.....	21
1.5.3 Entorno Labview.....	22
1.5.3.1 Panel Frontal .....	22
1.5.3.2 Diagrama de bloques.....	23
1.5.3.3 Paletas .....	24

## **CAPITULO II**

2 METODOLOGÍA .....	27
2.3 HIPÓTESIS.....	28
2.4 Operacionalización de las variables.....	28
2.5 PROCEDIMIENTOS .....	30
2.5.1 Identificación de las variables meteorológicas a medir .....	32
2.5.2 Selección de equipos y sensores .....	32
2.5.3 ENSAMBLAJE DE LAS ETAPAS DEL DISPOSITIVO .....	33
2.5.3.1 Etapa de Transmisión.....	33
2.5.3.1.1 Diseño de la placa para el sensor SHT75.....	33
2.5.3.1.2 Diseño de la placa para el módulo Xbee.....	34
2.5.3.2 Etapa de Recepción.....	35
2.5.4 PROGRAMACIÓN EN ARDUINO .....	37
2.5.4.1 Programación para el sensor de temperatura y humedad relativa SHT75	37
2.5.4.2 Programación para el sensor de presión atmosférica BMP085.....	39
2.5.4.3 Programación para el sensor de velocidad del viento MD0550.....	40
2.5.5 TRANSMISIÓN DE DATOS.....	41
2.5.6 INTERFAZ DE USUARIO EN LABVIEW .....	42
2.6 COMPROBACIÓN DE LA HIPOTESIS .....	45
2.6.1 Planteamiento de la hipótesis.....	45
2.6.2 Establecimiento de nivel de significancia.....	45
2.6.3 Muestra encontrada .....	45
2.6.4 Cálculo de la media y la varianza .....	46



2.6.5 Cálculo del valor crítico o $z_c$ .....	49
2.6.6 Representación gráfica de la regla de decisión .....	49
<b>CAPITULO III</b>	
3. RESULTADOS.....	51
3.1 Medición usando la Radiosonda .....	51
<b>CAPITULO IV</b>	
4. CONCLUSIONES Y RECOMENDACIONES.....	53
4.1 Conclusiones .....	53
4.2 Recomendaciones.....	54
<b>CAPITULO V</b>	
5. PROPUESTA.....	55
5.1 Título de la propuesta.....	55
5.2 Introducción .....	55
5.3 Objetivos .....	56
5.3.2 Específicos .....	56
5.4 Fundamentación Científico –Técnica .....	56
5.5 Descripción de la propuesta .....	57
5.6 Diseño Organizacional.....	57
5.7 Monitoreo y Evaluación de la propuesta.....	57
6. BIBLIOGRAFÍA .....	59
7. APÉNDICES Y ANEXOS.....	61
ANEXO 1. DISEÑO DE LAS PLACAS EN PDF .....	61
ANEXO 2. CAJAS DE TRANSMISIÓN Y RECEPCIÓN ENSAMBLADAS...	63
ANEXO 3. CONFIGURACIÓN MÓDULOS XBEE .....	65
ANEXO 4. INTERFAZ DE USUARIO EN LABVIEW.....	70
ANEXO 5. HOJAS DE ESPECIFICACIONES TÉCNICAS .....	79
ANEXO 6. PROGRAMACIÓN EN ARDUINO .....	82

## INDICE DE TABLAS

<b>Tabla 1.</b> Especificaciones técnicas SHT75 .....	11
<b>Tabla 2.</b> Especificaciones técnicas Arduino.....	22
<b>Tabla 3.</b> Especificaciones técnicas Modulo RF Xbee 868 MHz.....	23
<b>Tabla 4.</b> Requerimientos para Windows .....	24
<b>Tabla 5.</b> Requerimientos para Mac OS X .....	25
<b>Tabla 6.</b> Requerimientos para Linux .....	25
<b>Tabla 7.</b> Operacionalizacion de las variables .....	33
<b>Tabla 8.</b> Equipos y materiales utilizados.....	36
<b>Tabla 9.</b> Mediciones realizadas con el dispositivo implementado .....	49
<b>Tabla 10.</b> Mediciones realizadas con otros dispositivos .....	50
<b>Tabla 11.</b> Cálculo de la media y desviación de las mediciones realizadas de humedad .....	51
<b>Tabla 12.</b> Cálculo de la media y desviación de las mediciones realizadas de temperatura.....	51
<b>Tabla 13.</b> Cálculo de la media y desviación de las mediciones realizadas de presión.....	52
<b>Tabla 14.</b> Cálculo de la media y desviación de las mediciones realizadas de velocidad del viento .....	52

## INDICE DE FIGURAS

<b>Figura 1.</b> Conexión del sensor SHT75 al micro controlador .....	12
<b>Figura 2.</b> Sensor BMP085 .....	13
<b>Figura 3.</b> Sensor MD0550 .....	15
<b>Figura 4.</b> Versiones disponibles de IDE Arduino .....	17
<b>Figura 5.</b> Ejecución IDE Arduino .....	18
<b>Figura 6.</b> Selección de tipo de placa a usarse.....	19
<b>Figura 7.</b> Selección del puerto serie de Arduino .....	19
<b>Figura 8.</b> Opciones de la interfaz .....	20
<b>Figura 9.</b> Arduino UNO .....	21
<b>Figura 10.</b> Panel Frontal LabVIEW .....	27
<b>Figura 11.</b> Diagrama de bloques LabVIEW .....	28
<b>Figura 12.</b> Paleta de herramientas .....	29
<b>Figura 13.</b> Paleta de controles .....	29
<b>Figura 14.</b> Paleta de funciones .....	30
<b>Figura 15.</b> Esquema de las etapas del equipo diseñado .....	34
<b>Figura 16.</b> Diagrama del procedimiento .....	36
<b>Figura 17.</b> Elementos para la placa del sensor SHT75 .....	37
<b>Figura 18.</b> Vista 3D de la placa del sensor SHT75 .....	38
<b>Figura 19.</b> Elementos para la placa de los módulos Xbee .....	38
<b>Figura 20.</b> Vista 3D de la placa de los módulos Xbee .....	39
<b>Figura 21.</b> Elementos para la placa de Recepción .....	40
<b>Figura 22.</b> Vista 3D de la placa de recepción .....	40
<b>Figura 23.</b> Secuencia de inicio de transmisión del SHT75. ....	41
<b>Figura 24.</b> Envío de información al SHT75 .....	42
<b>Figura 25.</b> Recepción de datos del SHT75 .....	43
<b>Figura 26.</b> Secuencia para la obtención de datos del sensor BMP085.....	44
<b>Figura 27.</b> Secuencia para la obtención de la velocidad del viento .....	45
<b>Figura 28.</b> Enlace punto a punto con los módulos Xbee.....	46
<b>Figura 29.</b> Diagrama de flujo de la programación en labview .....	47
<b>Figura 30.</b> Panel frontal del programa en Labview.....	48

<b>Figura 31.</b> Diagrama de Bloques del programa en Labview. ....	48
<b>Figura 32.</b> Grafica de distribución normal .....	54
<b>Figura 33.</b> Verificar carga de la batería.....	55
<b>Figura 34.</b> Conexión de la caja de Recepción .....	56
<b>Figura 35.</b> Página de Digi para descargar XCTU.exe.....	69
<b>Figura 36.</b> Ventana principal de XCTU .....	70
<b>Figura 37.</b> Ventana donde escanea los puertos COM conectados .....	70
<b>Figura 38.</b> Configuración de los parámetros de la conexión del puerto COM ....	71
<b>Figura 39.</b> Características básicas del módulo conectado.....	71
<b>Figura 40.</b> Modulo conectado y su configuración actual .....	72
<b>Figura 41.</b> Firmwares disponibles para el modulo .....	72
<b>Figura 42.</b> Configuración del módulo Xbee .....	73
<b>Figura 43.</b> Paleta de funciones de VISA .....	74
<b>Figura 44.</b> Paleta de funciones VISA Read.....	75
<b>Figura 45.</b> Paleta de funciones Property Node.....	76
<b>Figura 46.</b> Paleta de funciones VISA Close.....	77
<b>Figura 47.</b> Lectura puerto serial .....	77
<b>Figura 48.</b> Concatenate Strings .....	78
<b>Figura 49.</b> Paleta de funciones Match Pattern.....	79
<b>Figura 50.</b> Paleta de funciones Fract/Exp String To Number .....	80
<b>Figura 51.</b> Paleta de funciones Max & Min .....	81
<b>Figura 52.</b> Funcion Write to Measurement File .....	82

## RESUMEN

El presente proyecto se centra en el diseño e implementación de un dispositivo de transmisión-recepción inalámbrico, basado en la tecnología ZigBee y de fácil movilización; para la medición de parámetros meteorológicos tales como: humedad, velocidad del viento, temperatura y presión atmosférica

Para el diseño y desarrollo del dispositivo se utilizan componentes de hardware tales como: sensores que permiten medir las variables meteorológicas, tarjeta Arduino usado para realizar la programación de los distintos sensores, módulos Xbee para la transmisión y recepción de datos; y componentes de software como: XCT-U interfaz que permite la configuración de los módulos Xbee, labview para el desarrollo de un programa de adquisición de datos que permite realizar el monitoreo y registro de la información recolectada.

El dispositivo implementado permite realizar mediciones hasta de 1 kilómetro con resultados óptimos; las mediciones se pueden realizar en distintos lugares por ser un equipo portable.



**UNIVERSIDAD NACIONAL DE CHIMBORAZO**  
**CENTRO DE IDIOMAS INSTITUCIONAL**

**Ms. Hugo Romero**

**25 de Julio de 2016**

**ABSTRACT**

The present project focuses on the design and implementation of a device of wireless transmission - receipt, it is based on the ZigBee technology and it is of easy mobilization; for the measurement of meteorological parameters such as: dampness, speed of the wind, temperature and atmospheric pressure.

For the design and development of the device were used components of hardware such as: sensors that allow to measure the meteorological variables, Arduino card used to do the programming of the different sensors, Xbee modules for the transmission and data reception; and components of software as : XCT - U interface that allows the configuration of the Xbee, labview modules for the development of a program of acquisition of information that allows to do the monitoring and record of the gathered data.

The implemented device allows doing measurements up to 1 kilometer with optimal results; the measurements can be made in different places for being a portable equipment.

A handwritten signature in blue ink, appearing to read "Hugo Romero".



## INTRODUCCIÓN

En el país no existe el fácil acceso al uso de dispositivos como la radiosonda que permiten medir varios parámetros meteorológicos en conjunto, excepto en instituciones donde es indispensable el uso de esta tecnología como el Instituto Nacional de Meteorología e Hidrología, Fuerza Aérea, entre otras.

Tomando en cuenta las restricciones que las personas tienen para el uso de dispositivos de esta clase, en la Universidad Nacional de Chimborazo en la Facultad de Ingeniería en Electrónica y Telecomunicaciones nace la idea de implementar un prototipo encaminado a estudiar variaciones, tendencias e impactos del clima.

Los dispositivos que se utilizan en las diferentes instituciones con en su mayor parte son adquiridas en el exterior a un costo elevado; el equipo implementado utiliza elementos de fácil adquisición a un costo muy accesible, el mismo que será usado en el proyecto “PROGRAMA INTEGRADO DE MONITOREO DE CONTROL DE CALIDAD DEL AIRE EN LA CIUDAD DE RIOBAMBA”

## **CAPITULO I**

### **1. MARCO TEÓRICO**

#### **1.1 METEOROLOGÍA**

Meteorología viene del griego, meteoros (alto), logos (tratado). Se asocia a las condiciones de la atmósfera en un tiempo y lugar específico, con relación a la temperatura, precipitaciones, presión atmosférica y otros factores climáticos.

Los factores climáticos fundamentales en el estudio y predicción del tiempo son: la presión atmosférica, la temperatura y la humedad.

- La presión atmosférica, variable también en el transcurso del día, es registrada en los mapas meteorológicos mediante el trazado de las isobaras o puntos de igual presión, que permiten identificar los centros de baja presión o borrascas, cuya evolución determina en gran parte el tiempo reinante.
- La temperatura, sometida a numerosas oscilaciones, se halla condicionada por la latitud y por la altura sobre el nivel del mar.
- La humedad es indicativo de que esa masa de aire ya no puede almacenar más vapor de agua, y a partir de ese momento, cualquier cantidad extra de vapor se convertirá en agua líquida o en pequeños cristales de hielo.

La meteorología utiliza instrumentos esenciales, como el barómetro, el termómetro y el higrómetro, para determinar los valores absolutos, medios y extremos de los factores climáticos. Para el trazado de mapas y la elaboración de predicciones es



fundamental la recogida coordinada de datos en amplias zonas. Existen equipos que permiten la medición de todas las variables meteorológicas en conjunto, estos pueden ser estaciones meteorológicas y radiosondas.

### **1.1.1 RADIOSONDA**

Una radiosonda es un dispositivo pequeño para la medición de parámetros meteorológicos compuesto por un transmisor y sensores que permiten analizar y medir las condiciones físicas de la atmósfera, se puede trasladar de un lugar a otro y transmite los datos a una estación terrestre en la que personal capacitado y calificado se ocupa de analizar dicha información,

## **1.2 SENSORES**

“Un sensor es un dispositivo que a partir de la energía del medio donde se mide, da una señal de salida transducible que es función de la variable medida” (Pallás, 2003, p.3).

Los sensores permiten captar variaciones de magnitudes físicas como temperatura, intensidad luminosa, entre otras; así como también otras alteraciones producidas en su entorno.

### **1.2.1 CARACTERISTICAS DE LOS SENSORES**

Un sensor ideal sería aquel donde la relación entre la magnitud de entrada y la magnitud de salida fuese proporcional y proporcione una respuesta instantánea e idéntica para todos los elementos de un mismo tipo.

Sin embargo, la respuesta real de los sensores nunca es del todo lineal, tiene un rango limitado de validez, suele estar condicionada por perturbaciones del entorno y tiene un cierto retardo en la respuesta. (Osuna, 2015, p.7)

Las características de los sensores se pueden agrupar en dos grupos: características estáticas y características dinámicas

#### 1.2.1.1 Características Estáticas

- ✓ **Rango de medida:** es el conjunto de valores que puede tomar la señal de entrada con una tolerancia de error aceptable.
- ✓ **Resolución:** Indica que variación de la señal de entrada produce una variación detectable en la señal de salida.
- ✓ **Precisión:** define la variación máxima entre la salida real obtenida y la salida teórica dada como patrón para el sensor.
- ✓ **Repetitibilidad:** Indica la máxima variación entre valores de salida obtenidos al medir varias veces la misma entrada con el mismo sensor y en condiciones ambientales similares.
- ✓ **Linealidad:** se considera un sensor lineal si existe una constante de proporcionalidad única que relaciona los incrementos de la señal de salida con los respectivos incrementos de la señal de entrada en todo el rango de medida.
- ✓ **Sensibilidad:** indica la mayor o menor variación de la señal de salida por unidad de la magnitud de entrada. Cuanto mayor sea la variación de la señal de salida producida por una variación en la señal de entrada, el sensor es más sensible.
- ✓ **Ruido:** es cualquier perturbación aleatoria del propio sistema de medida que afecta la señal que se quiere medir.

#### 1.2.1.2 Características Dinámicas

- ✓ **Velocidad de respuesta:** mide la capacidad del sensor para que la señal de salida siga sin retraso las variaciones de la señal de entrada.
- ✓ **Respuesta en frecuencia:** mide la capacidad del sensor para seguir las variaciones de la señal de entrada a medida que aumenta la frecuencia.
- ✓ **Estabilidad:** indica la desviación en la salida del sensor con respecto al valor teórico dado, al variar parámetros exteriores distintos al que se quiere medir.

### 1.2.2 CLASIFICACION DE LOS SENSORES

Los sensores clasificados según el tipo de señal que generan son: Digitales, Análogos y Temporales.

- ✓ **Sensores Digitales:** frente a un estímulo cambian de estado ya sea de cero a uno o de uno a cero (refiriéndose a lógica digital), en este caso no existen estados intermedios y los valores de tensión que se obtienen son únicamente dos, 5V y 0V.
- ✓ **Sensores Análogos:** como salida emiten una señal comprendida por un campo de valores instantáneos que varían en el tiempo, y son proporcionales a los efectos que se están midiendo
- ✓ **Sensores temporales:** entregan una señal variable en el tiempo la cual puede ser una onda sinusoidal, triangular o cuadrada.

### 1.2.3 SENSORES DE TEMPERATURA

Existen varios sensores que permiten la medición de la temperatura, los más utilizados son los sensores de circuito integrado.

Los circuitos integrados se basan en los transistores, uniendo las terminales base y colector y haciendo pasar una corriente constante por el circuito, se obtiene una tensión entre las terminales base y emisor, que es dependiente de la temperatura en aproximadamente  $2\text{mV}/^\circ\text{C}$ . Haciendo uso de esta característica se obtienen sensores de temperatura bastante lineales comúnmente en un rango de  $-40^\circ\text{C}$  a  $150^\circ\text{C}$  con un costo relativamente bajo.

#### **1.2.4 SENSORES DE HUMEDAD**

Principalmente se utilizan dos tipos de sensores electrónicos de humedad: los sensores de tipo resistivo y los sensores de tipo capacitivo.

- ✓ Los sensores de tipo resistivos miden el cambio en el valor de la resistencia eléctrica con respecto a la humedad, en algún polímero conductor, sal, o algún sustrato tratado. La relación que existe entre humedad relativa y resistencia es de tipo exponencial inversa.
- ✓ Los sensores de tipo capacitivo detectan un cambio en la humedad relativa del ambiente y la reflejan como una variación en el valor de su capacitancia. Se encuentran formados por dos placas conductoras separadas por un aislante que absorbe y expelle vapor de agua del ambiente conforme cambia la humedad, esto se refleja como un cambio en la constante dieléctrica.

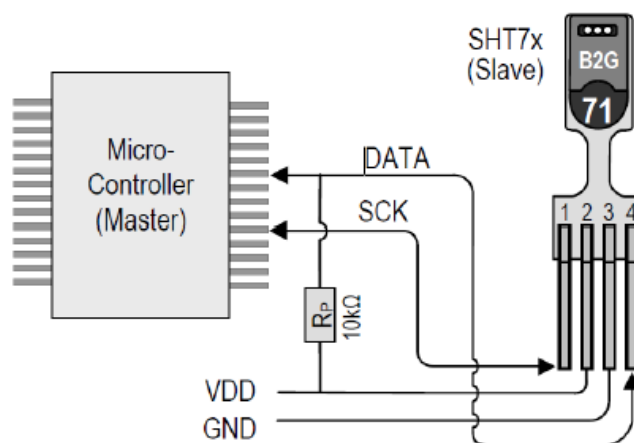
##### **1.2.4.1 Sensor de humedad relativa y temperatura SHT75**

SHT75 de Sensirion está fabricado con un capacitor de polímero para la humedad relativa y con silicio tecnología de banda prohibida (band-gap sensor) para la temperatura, estos son acoplados a un convertidor de analógico a digital de 14 bits y a un circuito de interface serial. En la tabla 1. Se indican las especificaciones técnicas del sensor.

**Tabla 1.** Especificaciones técnicas SHT75

Tensión de alimentación	2.4...5.5V CC
Campo de medición	0...100% RH
Exactitud de medición de temperatura	0,3°C
Exactitud de medición de humedad	±1.8%
Campo de medición de temperatura	-40...125°C

En la Figura 1. Se muestra la conexión recomendada por el fabricante. Se coloca una resistencia de 10K $\Omega$  entre el pin VDD (pin 2) y DATA (pin 4) del sensor, que tiene la función de poner en estado lógico alto el pin del micro controlador, para que así el micro controlador trabaje sólo estados lógicos bajos.



**Figura 1.** Conexión del sensor SHT75 al micro controlador

**Fuente:** Datasheet sensor SHT75

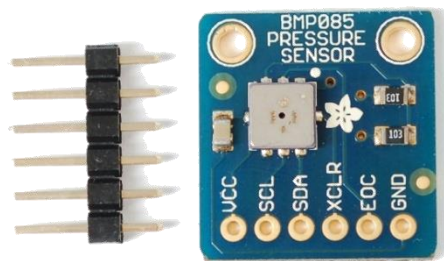
### 1.2.5 SENSORES DE PRESIÓN ATMOSFÉRICA

La Presión se define como fuerza por unidad de área que un fluido ejerce en sus inmediaciones.

Los sensores de presión atmosférica utilizan un dispositivo de suma de fuerzas para convertir la presión en un esfuerzo proporcional a la presión. Dicho esfuerzo se aplica luego a un elemento de transducción eléctrica para generar la señal requerida.

Los sensores de presión disponibles al público poseen una escala que proporciona un método para convertir a unidades de ingeniería. La unidad utilizada según el sistema internacional (SI) para medir la presión es el Pascal (N/m<sup>2</sup>).

### 1.2.5.1 Sensor de presión atmosférica BMP085



*Figura 2. Sensor BMP085*

*Fuente: Datasheet del fabricante*

Es un sensor de alta precisión y de bajo consumo de energía. El BMP085 (Figura 2) ofrece un rango de medición de 300 a 1100 hPa (Hecto Pascal), con una precisión absoluta de hasta 0,03 hPa. Se basa en la tecnología piezo-resistiva, alta precisión y linealidad, así como con estabilidad a largo plazo.

Este sensor es compatible con un voltaje entre 1,8 y 3.6VDC. Se ha diseñado para ser conectado directamente a un microcontrolador a través de I2C.

Éste sensor permite calcular la altitud con bastante precisión.

Para utilizarlo con Arduino, simplemente se debe conectar GND, VCC a 3.3V, SCL a I2C Clock, y SDA a I2C Data. Luego instalar la librería para BMP085.

### Características

- ✓ Digital interfaz de dos cables (I2C)
- ✓ Amplio rango de medición de presión barométrica
- ✓ Ultra-bajo consumo de energía
- ✓ Bajo ruido
- ✓ Completamente calibrado
- ✓ Medición de temperatura incluida
- ✓ Ultraplano y pequeño tamaño

### 1.2.6 SENSORES DE VELOCIDAD DEL VIENTO

El viento se define como el movimiento horizontal del aire. El dispositivo para medir la velocidad del viento es un anemómetro.

Existen algunos tipos de anemómetros, los principales son:

- ✓ **De empuje:** posee una esfera hueca o una pala, cuya posición respecto a un punto de suspensión varía con la velocidad del viento.
- ✓ **De rotación o de copelas:** está dotado de cazoletas o hélices unidas a un eje central. La velocidad de giro es proporcional a la velocidad del viento.
- ✓ **De compresión:** se basa en el tubo de Pitot (un tubo con forma de L, con un extremo abierto hacia la corriente de aire y el otro conectado a un dispositivo de medición de presión), y está formado por dos pequeños tubos, uno de ellos con dos orificios, uno frontal que mide la presión dinámica y otro lateral que mide la presión estática, el otro tubo sólo con un orificio lateral. La diferencia entre las presiones medidas permite determinar la velocidad del viento.
- ✓ **De hilo caliente:** detecta la velocidad del viento mediante pequeñas diferencias de temperatura entre un hilo enfrentado al viento y otro a sotavento.

- ✓ **Sónico o anemómetro de efecto Doppler:** detecta el desfase del sonido (anemómetro de ultrasonido) o de la luz coherente (anemómetro laser) reflejados por las moléculas de aire.

### 1.2.6.1 Sensor de velocidad del viento MD0550



*Figura 3. Sensor MD0550*

*Fuente: Modern device*

MD0550 (Figura 3.) de modern device es un anemómetro térmico o de hilo caliente cuyo funcionamiento consiste en que el elemento sensor es calentado a una temperatura constante por la corriente provista por el circuito electrónico del sensor. El flujo de aire que va a ser medido para y enfría el elemento sensor, lo que obliga al circuito a entregar mayor corriente para conservar la temperatura constante. La pérdida de calor del sensor se puede relacionar con la velocidad según la ley de King:

$$Q = A + B * U^n$$



Q= pérdida de calor

A, B= constantes que dependen de las propiedades físicas del fluido, sensor y temperaturas de operación.

U= velocidad del fluido

n= Constante que depende de las dimensiones del sensor

El algoritmo proporcionado por el fabricante que determina la velocidad del aire utilizando este sensor es el siguiente:

1.  $zeroWindADunits = -0.0006 \times (TMP\_ThermADunits \times TMP\_ThermADunits) + 1.0727 \times TMP\_ThermADunits + 47.172$
2.  $zeroWindVolts = (zeroWindADunits \times B) - zeroWindAdjustment$
3.  $WindSpeed\_MPH = \left( \frac{RV\_WindVolts - zeroWindVolts}{0.23} \right)^A$
4.  $WindSpeed\_m/s = \frac{WindSpeed\_MPH}{2.23694}$

Donde:

TMP\_ThermADunits = es la lectura analógica del pin TMP;

RV\_WindADunits = es la lectura analógica del pin RV;

RV\_WindVolts = se obtiene de: RV\_WindADunits x B;

zeroWindAdjustment = 0.2;

A = 2.7265

B = 0.0048828125

### 1.3 ARDUINO

Es una plataforma de desarrollo de código abierto basado en un hardware y software de fácil uso. Las placas arduino permiten leer datos de una gran variedad de interruptores y sensores y controlar multitud de tipos de luces, motores y otros actuadores físicos.

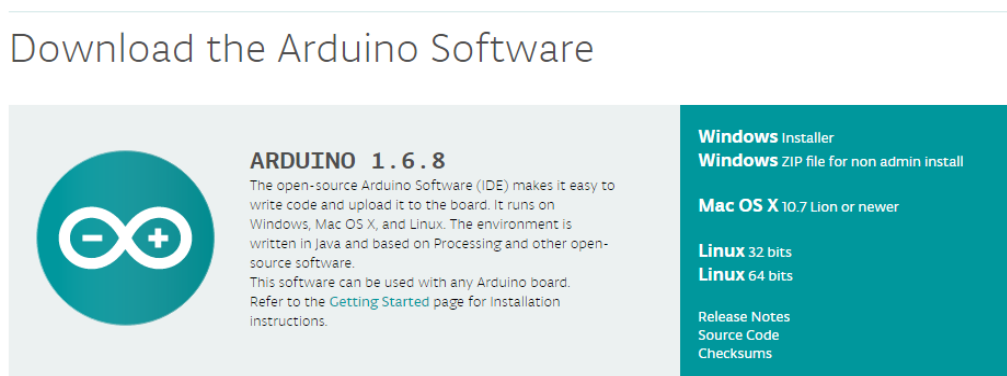
Arduino posee una sencilla y accesible experiencia de usuario por lo que es usado en miles de diferentes proyectos y aplicaciones. El software de Arduino (IDE) es fácil de usar para los principiantes, pero lo suficientemente flexible para los usuarios avanzados. Se ejecuta en Mac, Windows y Linux.

### 1.3.1 ENTORNO DE DESARROLLO DE ARDUINO

Arduino posee un entorno de programación fácil y funcional que permite avanzar con mayor rapidez en el desarrollo de un proyecto.

#### 1.3.1.1 Inicio del IDE de Arduino

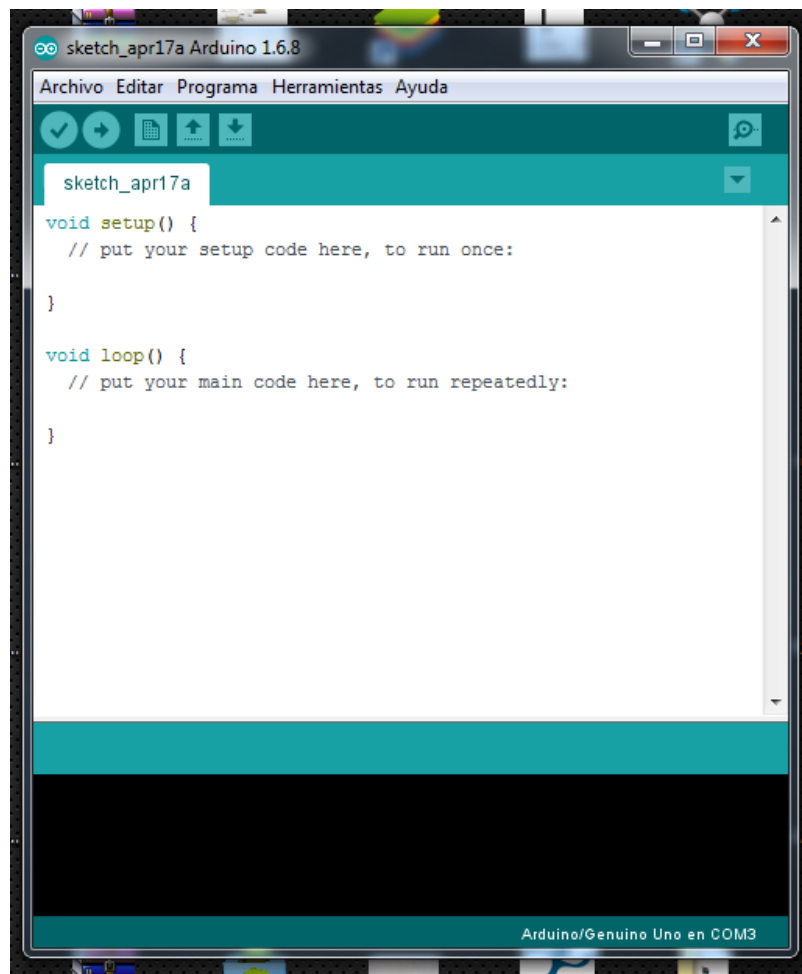
Para iniciar con la programación es necesario descargarse de la página web de Arduino el entorno de desarrollo (IDE). Están disponibles versiones para Windows, Linux y para MAC. (Figura 4.)



**Figura 4.** Versiones disponibles de IDE Arduino

**Fuente:** [www.arduino.cc](http://www.arduino.cc)

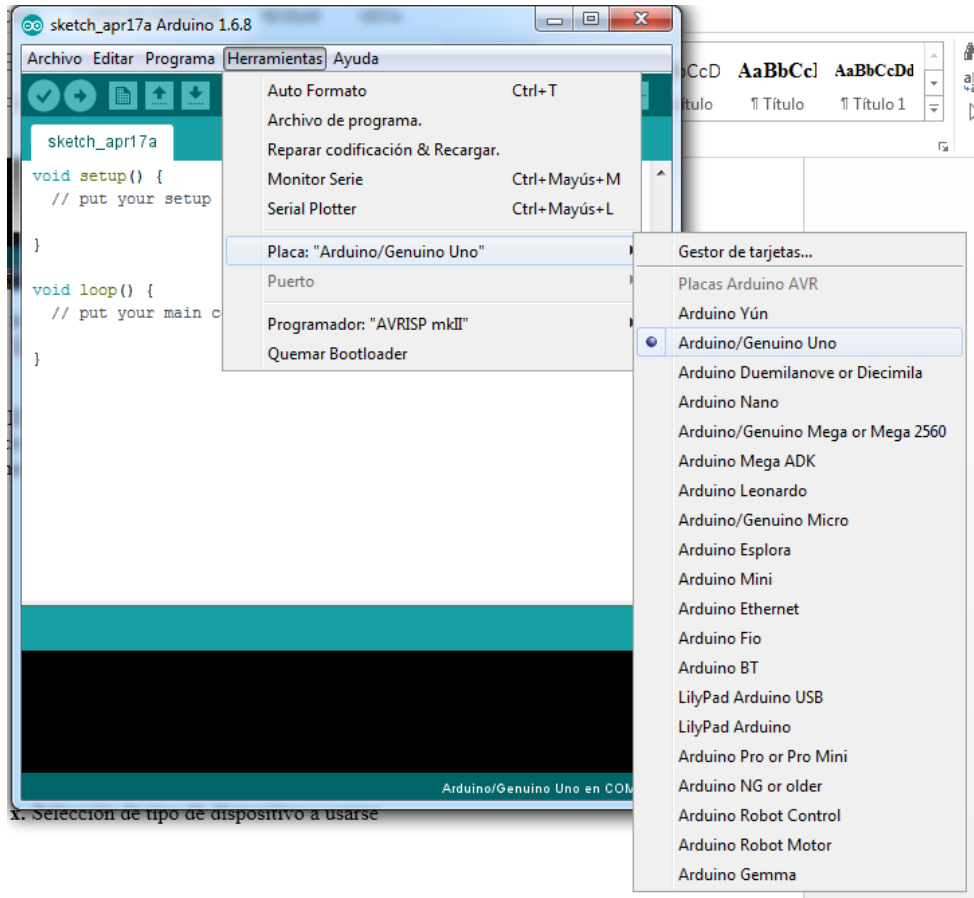
Tras realizar la descarga, se realiza la instalación ejecutando el archivo descargado y siguiendo los pasos que se van presentando. Ahora se procede con la ejecución para configurar el IDE de desarrollo, pulsaremos con el botón derecho del ratón sobre el ejecutable "arduino.exe". Ejecutar como administrador o doble clic solo el ejecutable. (Figura 5.)



**Figura 5.** Ejecución IDE Arduino

**Fuente:** Geomara Pilco

En el IDE de desarrollo de Arduino, en primer lugar se selecciona el tipo de dispositivo "Placa"; para ello se pulsa en el menú "Herramientas" - "Placa" y seleccionar el tipo de tarjeta Arduino que se esté usando "Arduino/Genuino Uno". (Figura 6.).

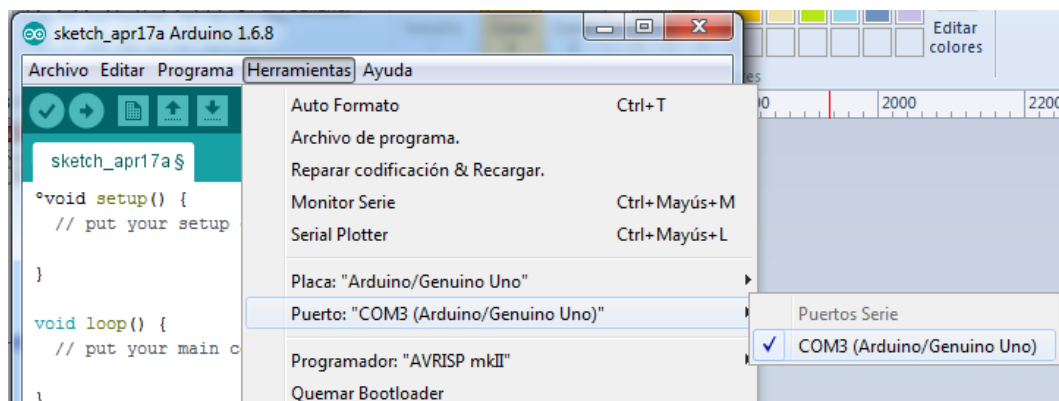


x. Selección de tipo de dispositivo a usarse

**Figura 6.** Selección de tipo de placa a usarse

*Fuente: Geomara Pilco*

Seleccionar también el puerto serie asignado al controlador de, para ello se accede al menú "Herramientas" - "Serial Port" - "COM3". (Figura 7.)



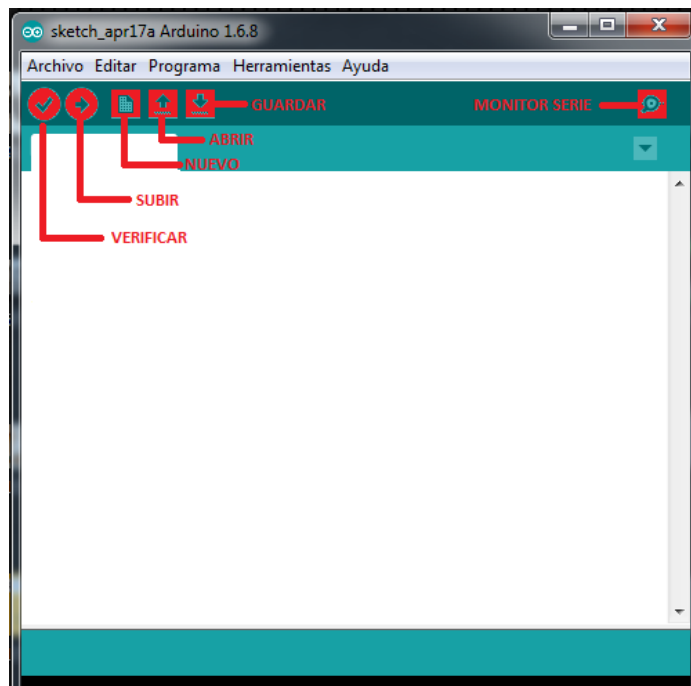
**Figura 7.** Selección del puerto serie de Arduino

*Fuente: Geomara Pilco*

### 1.3.1.2 Opciones de la interfaz

Las opciones (Figura 8.) indispensables se encuentran identificados con su respectivo icono y son los siguientes:

- ✓ **Verificar/Compilar:** Chequea el código, identificando los errores.
- ✓ **Subir:** Carga la programación realizada al Arduino.
- ✓ **Nuevo:** Crea un nuevo proyecto
- ✓ **Abrir:** presenta un menú para abrir proyectos previamente guardados y también muestra los proyectos de ejemplos
- ✓ **Guardar:** Guarda el proyecto actual
- ✓ **Monitor serie:** Realiza el monitoreo del puerto serial, visualiza la data enviada desde la tarjeta arduino.



*Figura 8. Opciones de la interfaz*

*Fuente: Geomara Pilco*

Los archivos son guardados con la extensión .ino, pero los archivos de las versiones anteriores con extensión .pde los reconoce sin problema.

### 1.3.2 Arduino UNO



*Figura 9. Arduino UNO*

*Fuente: [www.arduino.cc](http://www.arduino.cc)*

La Tarjeta Arduino Uno (Figura 9.) es una placa electrónica basada en el ATmega328P. Cuenta con 14 pines digitales de entrada / salida (de los cuales 6 se podrán utilizar como salidas PWM), 6 entradas analógicas, un cristal de cuarzo de 16 MHz, una conexión USB, un conector de alimentación y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador; basta con conectarlo a un ordenador con un cable USB o la corriente con un adaptador de corriente alterna a corriente continua o una batería.

Arduino Uno es el primero de una serie de placas Arduino USB, y el modelo de referencia para la plataforma Arduino; para una extensa lista de las tarjetas.

En la Tabla 2. Se detallan las especificaciones técnicas de Arduino Uno.

**Tabla 2. Especificaciones técnicas Arduino Uno**

Microcontrolador	ATmega328P
Tensión de funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límite)	6-20V
E / S digitales prendedores	14 (6 proporcionan salida PWM)
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
Memoria flash	32 KB
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Velocidad de reloj	16 MHz
Longitud	68,6 mm
Anchura	53,4 mm
Peso	25 g

## **1.4 TECNOLOGIA ZIGBEE**

ZigBee es un estándar de comunicaciones inalámbricas diseñado por ZigBee Alliance. Es un conjunto estandarizado de soluciones que pueden ser implementadas por cualquier fabricante. ZigBee está basado en el estándar IEEE 802.15.4 de redes inalámbricas de área personal y tiene como objetivo las aplicaciones que requieren comunicaciones seguras con maximización de la vida útil de sus baterías. (Glen,2013)

### **1.4.1 Tipos de Dispositivos**

Según su papel en la red se definen tres tipos distintos de dispositivo ZigBee:

- ✓ **Coordinador ZigBee:** es el dispositivo más completo, debería existir uno por red. Sus funciones son las de encargarse de controlar la red y los caminos que deben seguir los dispositivos para conectarse entre ellos, requiere memoria y capacidad de computación.
- ✓ **Router ZigBee:** Interconecta dispositivos separados en la topología de la red.
- ✓ **Dispositivo final:** Posee la funcionalidad necesaria para comunicarse con su nodo (coordinador o router), pero no puede transmitir información destinada a otros dispositivos. De esta forma, este tipo de nodo puede estar suspendido la mayor parte del tiempo, aumentando el tiempo de duración de sus baterías.

#### 1.4.2 Módulos RF Xbee-868MHZ

Este módulo es pin a pin compatible con el resto de módulos Xbee. Es un módulo pensado para comunicaciones punto-punto o punto-multipunto pero con una característica adicional, el largo alcance.

En la Tabla 3. Se detallan las especificaciones técnicas del módulo Xbee 868MHz.

*Tabla 3. Especificaciones técnicas Módulo RF Xbee 868 MHz*

<b>Tensión de alimentación</b>	3.0 - 3.6 V CC
<b>Alcance Interiores / Zona Urbana</b>	Hasta 1800 pies (550 m)
<b>Alcance al aire Libre</b>	Hasta 10 millas (16 km) con antena dipolo (13,7 dBm)
<b>Potencia de transmisión</b>	1 mW (0 dBm) a 315 mW (25 dBm)
<b>Sensibilidad del receptor (1% PER)</b>	-112 dBm
<b>Interfaz de datos Serie</b>	3.3V CMOS de serie UART
<b>Canales</b>	Un solo canal



## 1.5 LABVIEW

Labview es un lenguaje de programación de alto nivel de tipo gráfico, inicialmente enfocado al uso en instrumentación. Pero como lenguaje de programación, debido a que cuenta con todas las estructuras, puede ser usado para elaborar cualquier algoritmo que se desee, en cualquier aplicación, como en análisis, telemática, juegos, manejo de textos, etc. (Ruiz, Molina, 2010, p.275)

Cada programa realizado en Labview es llamado Instrumento Virtual (VI).

### 1.5.1 Requerimientos del sistema

Los requerimientos del sistema para instalar Labview varían de acuerdo al sistema operativo que se utilice; a continuación se detalla cada uno, en la Tabla 4. Para Windows, en la Tabla 5. Para Mac y en la Tabla 6. Para Linux.

*Tabla 4. Requerimientos para Windows*

<b>Procesador</b>	Pentium 4M (o equivalente) o posterior (32 bits) Pentium 4 G1 (o equivalente) o posterior (64 bits)
<b>RAM</b>	1 GB
<b>Resolución de Pantalla</b>	1024 x 768 píxeles
<b>SO</b>	Windows 8.1/8/7/Vista (32 bits y 64 bits) Windows XP SP3 (32 bits) Windows Server 2012 R2 (64 bits)/ 2008 R2 (64 bits)/ 2003 R2 (32 bits)
<b>Espacio en Disco</b>	5 GB (Incluye controladores predeterminados del DVD de Controladores de Dispositivos de NI)

**Tabla 5. Requerimientos para Mac OS X**

<b>Procesador</b>	Procesador basado en Intel
<b>RAM</b>	2 GB
<b>Resolución de Pantalla</b>	1024 x 768 píxeles
<b>SO</b>	OS X 10.9 o 10.10
<b>Espacio en Disco</b>	1.4 GB para la instalación completa (excluyendo controladores)

**Tabla 6. Requerimientos para Linux**

<b>Procesador</b>	Pentium 4M (o equivalente) o posterior (32 bits) Pentium 4 G1 (o equivalente) o posterior (64 bits)
<b>RAM</b>	1 GB
<b>Resolución de Pantalla</b>	1024 x 768 píxeles
<b>SO</b>	Red Hat Enterprise Linux Desktop + Workstation 6.5 o posterior, open SUSE 12.3 o 13.1 o Scientific Linux 6.5 o posterior
<b>Espacio en Disco</b>	1.1 GB para la instalación completa de cada bitness 2.2 GB para la instalación completa de LabVIEW de 32 y 64 bits

### 1.5.2 Ventajas

Las ventajas que proporciona Labview entre las principales tenemos:

- ✓ Reduce el tiempo de desarrollo de las aplicaciones

- ✓ Dota de gran flexibilidad al sistema, permitiendo cambios y actualizaciones tanto del hardware como del software.
- ✓ Tiene la posibilidad de incorporar aplicaciones escritas en otros lenguajes.
- ✓ Es muy simple de manejar, debido a que está basado en programación gráfica, llamada lenguaje G.
- ✓ Es un programa de mucho poder donde se cuentan con librerías especializadas para manejos de DAQ, Redes, Comunicaciones, Análisis Estadístico, Comunicación con Bases de Datos, etc.

### **1.5.3 Entorno Labview**

Un proyecto en Labview tiene un panel frontal y un diagrama de bloques. Están disponibles también las paletas que contienen las opciones que se emplean para crear y modificar los proyectos.

#### **1.5.3.1 Panel Frontal**

Es la interfaz gráfica del VI con el usuario. Esta interfaz recoge las entradas procedentes del usuario y representa las salidas proporcionadas por el programa, se puede observar en la Figura 10.

Un panel frontal está formado por una serie de botones, pulsadores, potenciómetros, gráficos, etc. Cada uno de ellos puede estar definido como un control o un indicador. Los controladores sirven para introducir parámetros al VI, mientras que los indicadores se emplean para mostrar los resultados producidos.



**Figura 10.** Panel Frontal LabVIEW

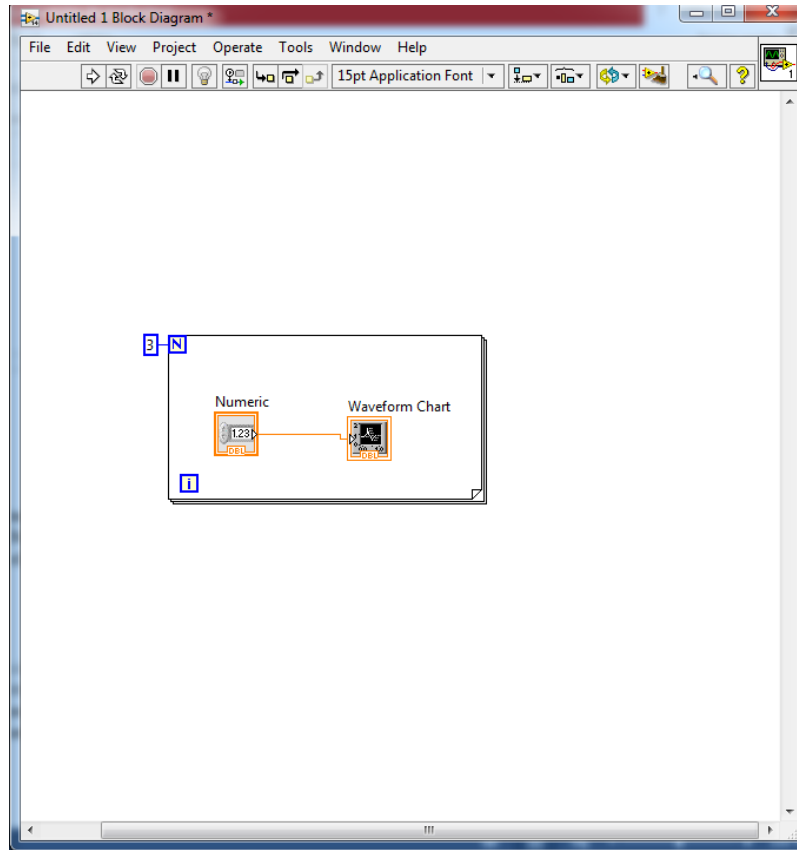
**Fuente:** Geomara Pilco

### 1.5.3.2 Diagrama de bloques

El diagrama de bloques constituye el código fuente del VI. En el diagrama de bloques es donde se realiza la implementación del programa del VI para controlar o realizar cualquier procesado u operación de las entradas y salidas.

El diagrama de bloques incluye funciones y estructuras integradas en las librerías de LabVIEW, entre ellas, aritméticas, comparaciones, conversiones, funciones de entrada/salida, de análisis, etc. En el lenguaje Grafico las funciones y las estructuras son nodos elementales.

El diagrama de bloques se construye conectando los distintos objetos entre sí, como si de un circuito se tratara. Los cables unen terminales de entrada y salida con los objetos correspondientes, y por ellos fluyen los datos.



**Figura 11.** Diagrama de bloques LabVIEW

**Fuente:** Geomara Pilco

Las estructuras, similares las usadas en lenguajes de programación convencionales, ejecutan el código que contienen de forma condicional o repetitiva (bucle, for, while, case, etc.).

Las líneas son las trayectorias que siguen los datos desde su origen hasta su destino, ya sea una función, una estructura, un terminal, etc. Cada una de estas tiene un color o un estilo diferente, lo que diferencia unos tipos de datos de otros.

### 1.5.3.3 Paletas

Las paletas proporcionan las herramientas que se requieren para crear y modificar tanto el panel frontal como el diagrama de bloques.

#### ✓ Paleta de herramientas

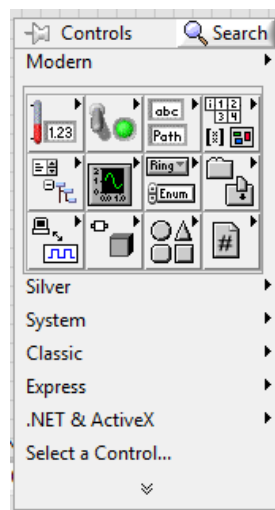


**Figura 12.** Paleta de herramientas

**Fuente:** Geomara Pilco

Utilizada en el panel frontal como en el diagrama de bloques. Contiene las herramientas necesarias para editar y depurar los objetos.

#### ✓ Paleta de controles

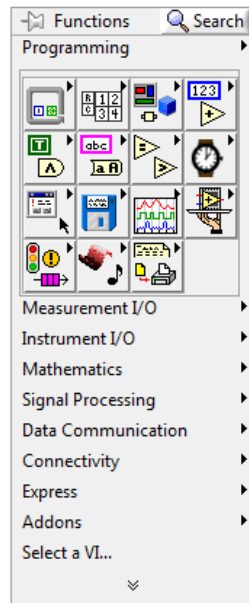


**Figura 13.** Paleta de controles

**Fuente:** Geomara Pilco

Está disponible únicamente en el panel frontal. Contiene todos los controles e indicadores que se emplearán para crear la interfaz del VI con el usuario.

✓ **Paleta de funciones**



**Figura 14.** Paleta de funciones

**Fuente:** Geomara Pilco

La paleta de funciones contiene todos los objetos que se emplean en la implementación del programa del VI, ya sean funciones aritméticas, de entrada/salida de señales, entrada/salida de datos a fichero, adquisición de señales, temporización de la ejecución del programa, etc.

Para seleccionar una función o estructura concretas, se debe señalar el grupo al que pertenece la función deseada para que se despliegue el menú y elegir la función requerida.

## CAPITULO II

### 2. METODOLOGÍA

#### 2.1 TIPO DE ESTUDIO

- **Exploratorio:** Es un estudio de tipo exploratorio ya que tiene como objeto esencial familiarizarse con un tema escasamente estudiado. Y servirá como punto de partida para estudios posteriores de mayor profundidad.
- **Metodología de medición:** Para determinar la eficiencia del dispositivo implementado.

#### 2.2 POBLACIÓN Y MUESTRA

##### 2.2.1 POBLACIÓN

Para la utilización de la radiosonda, la población está conformada por las mediciones realizadas usando otros dispositivos que permiten medir las variables meteorológicas.

##### 2.2.2 MUESTRA

Para el cálculo de la muestra se utiliza la siguiente formula:

$$n = \frac{N\sigma^2 Z^2}{(N-1)e^2 + \sigma^2 Z^2}$$



Donde:

- $n$  = el tamaño de la muestra.
- $N$  = tamaño de la población.
- $\sigma$  = Desviación estándar de la población que, generalmente cuando no se tiene su valor, suele utilizarse un valor constante de 0,5.
- $Z$  = Valor obtenido mediante niveles de confianza. Es un valor constante que, si no se tiene su valor, se lo toma en relación al 95% de confianza equivale a 1,96.
- $e$  = Límite aceptable de error muestral que, generalmente cuando no se tiene su valor, suele utilizarse un valor que varía entre el 1% (0,01) y 9% (0,09), valor que queda a criterio del encuestador. (Galarza 2015)

$$n = \frac{(10)(0.5)^2(1.96)^2}{(20 - 1)(0.09)^2 + (0.5)^2(1.96)^2}$$

$$n = 8,6133$$

$$n = 9 \text{ muestras}$$

## 2.3 HIPOTESIS

“El diseño e implementación de una radiosonda permitirá realizar la medición y monitoreo de parámetros meteorológicos”.

## 2.4 OPERACIONALIZACIÓN DE VARIABLES

En la tabla 7. Se indican las variables dependiente e independiente, el concepto de las mismas, los indicadores y las técnicas e instrumentos utilizados.

**Tabla 7. Operacionalización de las variables**

<b>VARIABLE</b>	<b>CONCEPTO</b>	<b>INDICADORES</b>	<b>TECNICAS E INSTRUMENTOS</b>
<b>Independiente:</b> Radiosonda	La radiosonda es un equipo capaz de monitorear condiciones climáticas utilizando sensores de diversos tipos para determinar constantes físicas.	Conexión Sistemas Dispositivos	<b>Técnica:</b> ➤ Implementación
<b>Dependiente:</b> Medición de Parámetros meteorológicos	Parámetros meteorológicos son las variables meteorológicas que determinan los cambios atmosféricos que se producen.	Medición de humedad, temperatura, presión atmosférica y velocidad del viento.	<b>Técnica:</b> ➤ Observación <b>Instrumento:</b> ➤ Software de adquisición de datos

## 2.5 PROCEDIMIENTOS



*Figura 15. Esquema de las etapas del equipo diseñado*

*Fuente: Geomara Pilco*

Para la implementación de este dispositivo, se requiere seguir una secuencia de pasos y etapas que permiten cumplir con el objetivo planteado. Cada etapa desempeña una función específica, y también sirve de base para la siguiente etapa, logrando así que el dispositivo implementado sea funcional. En la Figura 15 se puede observar la integración de las etapas para el funcionamiento del sistema.

La primera etapa del diseño consiste en analizar e identificar las variables meteorológicas a medir; según requerimientos se determina medir temperatura, humedad, presión atmosférica y velocidad del viento.

Ya identificadas las variables a medir se procede con la selección de sensores que permitan la medición de las mismas, esta selección se basa tanto en las características técnicas de los sensores así como también en el presupuesto y la factibilidad para su adquisición.

Luego de conocer las características de los sensores, se opta por utilizar la plataforma Arduino para la integración de todos los elementos del prototipo. Se

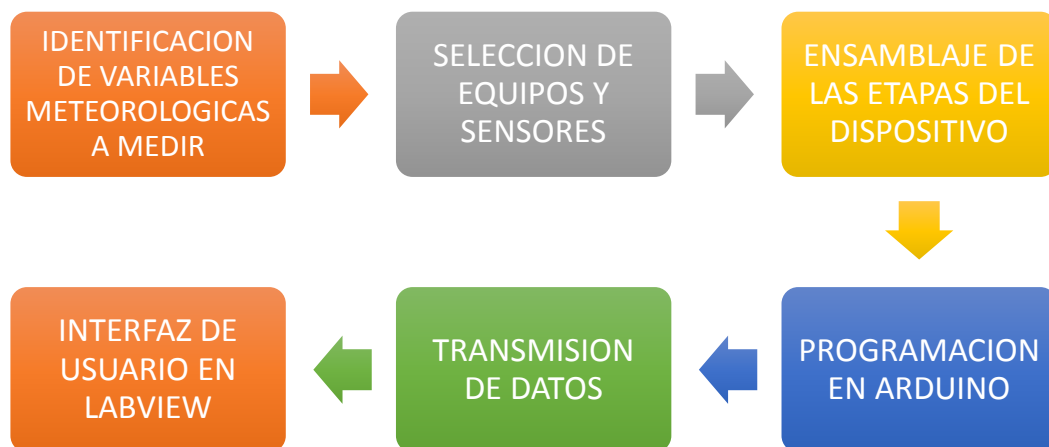
realiza la programación necesaria para obtener la información que proporciona cada sensor.

En la transmisión de datos se utiliza módulos Xbee Pro (Xbee-868MHZ) que trabajan con tecnología Zigbee en la frecuencia libre de 2.4 Ghz. Se usa el software XCTU propio del fabricante para la configuración de los mismos, logrando así un enlace punto a punto, útil para el proyecto.

Para el ensamblaje del dispositivo se usa la tarjeta Arduino uno en donde se integraron todos los sensores a utilizar, posteriormente se realiza una placa en donde están los elementos adicionales que necesitan ciertos sensores para su funcionamiento, así como también la conexión al módulo Xbee Pro que funciona como emisor. Para el módulo Xbee Pro que funciona como receptor se diseña una placa tipo shield la cual adapta los niveles de voltaje de la fuente a los requeridos por este dispositivo y permite la comunicación serial hacia un computador.

Finalmente para la recepción y procesamiento de datos se desarrolla una interfaz de usuario en Labview la cual permite visualizar en tiempo real los valores medidos, además muestra los valores máximos y mínimos durante el último minuto de medición. Los datos medidos se guardan en un archivo donde está registrada la hora y fecha de medición.

En la Figura 16. Se muestra el diagrama del procedimiento, cada etapa desarrollada para conseguir el diseño e implementación del dispositivo.



*Figura 16. Diagrama del procedimiento*

*Fuente: Geomara Pilco*

### **2.5.1 IDENTIFICACIÓN DE LAS VARIABLES METEOROLÓGICAS A MEDIR**

Existen algunas variables meteorológicas que pueden ser medidas; sin embargo se consideran para este proyecto la temperatura, humedad, presión atmosférica y velocidad del viento, dado que estas son la base para realizar un análisis, predicción o cualquier tipo de estudio climático.

### **2.5.2 SELECCIÓN DE EQUIPOS Y SENSORES**

Los equipos y sensores a utilizarse en la elaboración de la radiosonda se señalan en la tabla 8.

**Tabla 8.** Equipos y materiales utilizados

<b>CANTIDAD</b>	<b>DISPOSITIVO/SENSOR</b>
1	Tarjeta Arduino UNO
1	Sensor de humedad y temperatura SHT75

1	Sensor de velocidad del viento MD0550
1	Sensor de presión atmosférica BMP085
2	Módulos RF Xbee-868MHZ
1	Cable Serial- USB
1	Bateria de litio (recargable)

### 2.5.3 ENSAMBLAJE DE LAS ETAPAS DEL DISPOSITIVO

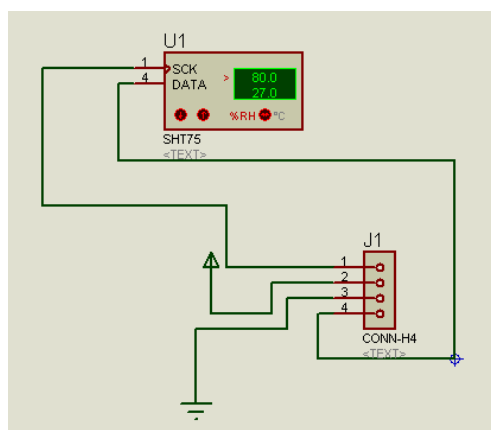
#### 2.5.3.1 Etapa de Transmisión

La etapa de transmisión consta de: sensores, tarjeta Arduino y transmisor.

Para comenzar con la etapa de transmisión se diseñan placas para el sensor SHT75 y para los módulos Xbee, ya que es necesario acoplar el tamaño de sus pines al tamaño estándar usado.

##### 2.5.3.1.1 Diseño de la placa para el sensor SHT75

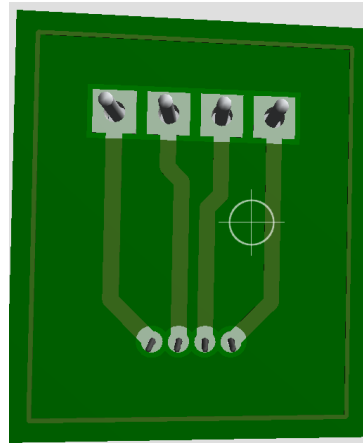
Para el diseño de la placa del sensor SHT75, en proteus ISIS se usaron dos elementos una regleta de 4 pines y el package propio del sensor que está disponible.



**Figura 17.** Elementos para la placa del sensor SHT75

**Fuente:** Geomara Pilco

En Proteus ARES se trazan las rutas conectando pin a pin de cada elemento, esto se realizó para acoplar el tamaño smd de lo pines del sensor al tamaño estándar de los mismos, en la figura 18. Se muestra la vista 3D del diseño. Anexo 1. Diseño de la placa del sensor SHT75 en PDF.

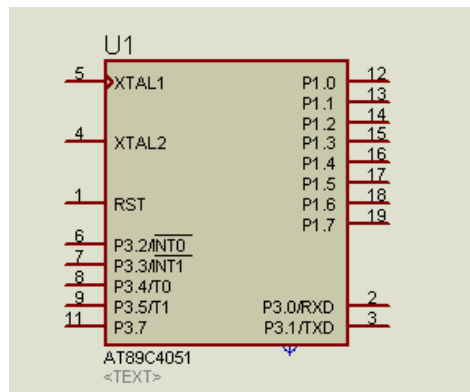


**Figura 18.** Vista 3D de la placa del sensor SHT75

*Fuente: Geomara Pilco*

### 2.5.3.1.2 Diseño de la placa para el módulo Xbee

Para el diseño de la placa para los módulos Xbee se coloca un solo elemento, cualquier package de un microprocesador que posea 20 pines.

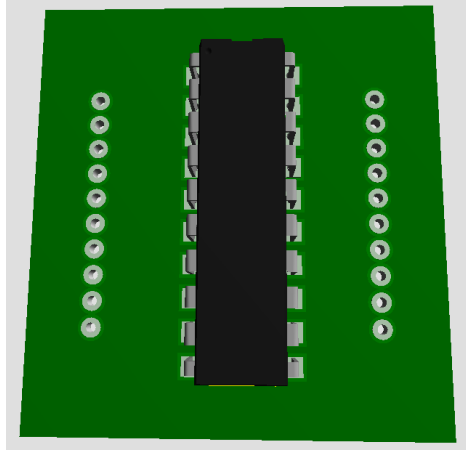


**Figura 19.** Elementos para la placa de los módulos Xbee

*Fuente: Geomara Pilco*

En Proteus ARES se coloca uno por uno los pines del módulo Xbee de acuerdo a sus medidas y se conecta cada uno de estos a los pines del microprocesador de 20

pinos y en la figura 20. Se muestra la vista 3D del diseño. Adicionalmente en el Anexo 1. Se muestra el diseño de la placa del sensor SHT75 en PDF.



*Figura 20. Vista 3D de la placa de los módulos Xbee*

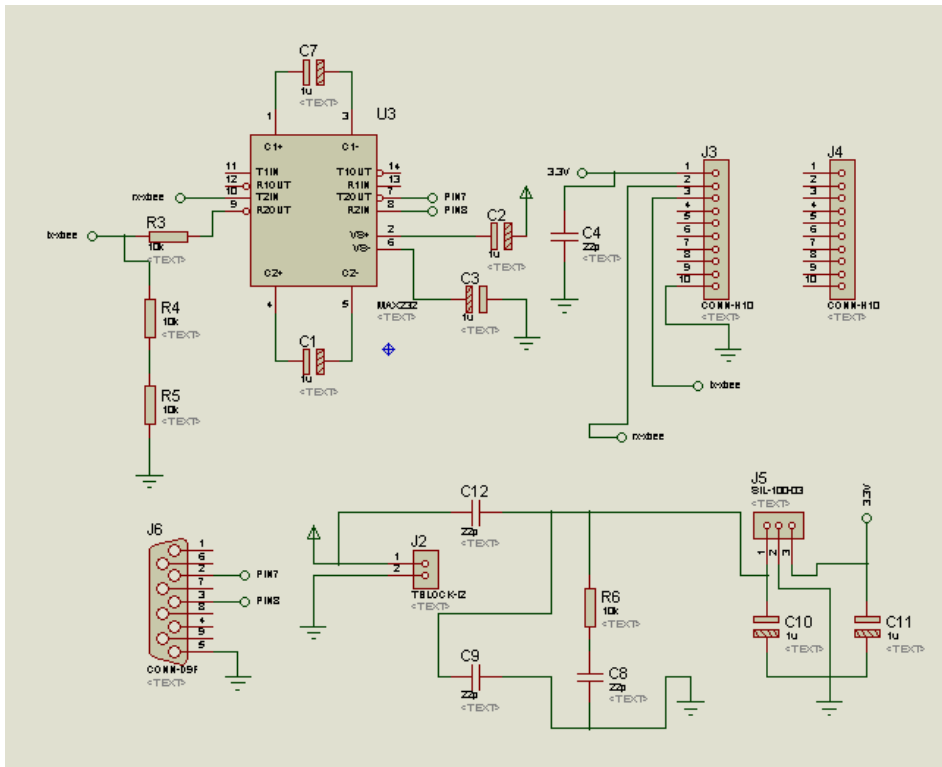
*Fuente: Geomara Pilco*

### **2.5.3.2 Etapa de Recepción**

La placa de recepción consta de los siguientes componentes:

- ✓ Circuito integrado MAX232: para convertir las señales de un puerto serie RS-232 a señales compatibles con los niveles TTL (lógica transistor a transistor) de circuitos lógicos.
- ✓ Adaptador de voltaje de 5V a 3.3V: para alimentar los componentes que trabajan con 5 voltios y con 3.3 voltios.
- ✓ Socket para el módulo Xbee: para la colocación del módulo Xbee.
- ✓ Package DB9: para la conexión del cable serial.
- ✓ Elementos complementarios: resistencias y capacitores.

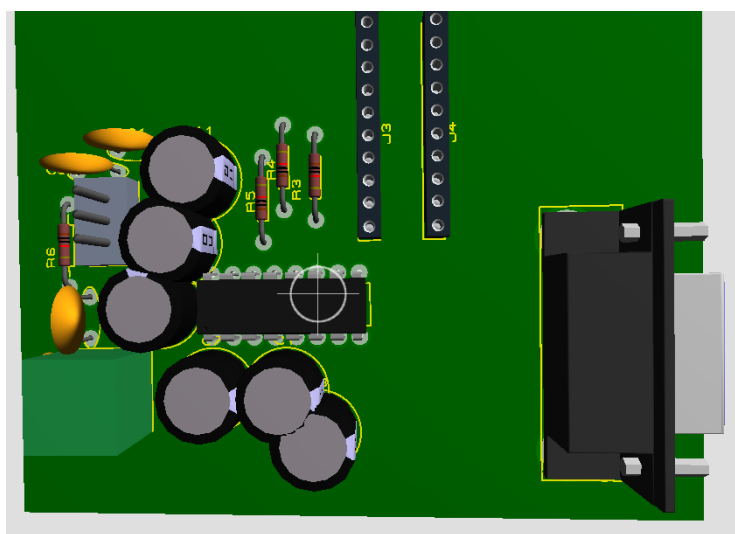




**Figura 21.** Elementos para la placa de Recepción

**Fuente:** Geomara Pilco

En proteus Ares se coloca cada elemento estratégicamente, considerando la utilización y posición requerida de cada elemento.



**Figura 22.** Vista 3D de la placa de recepción

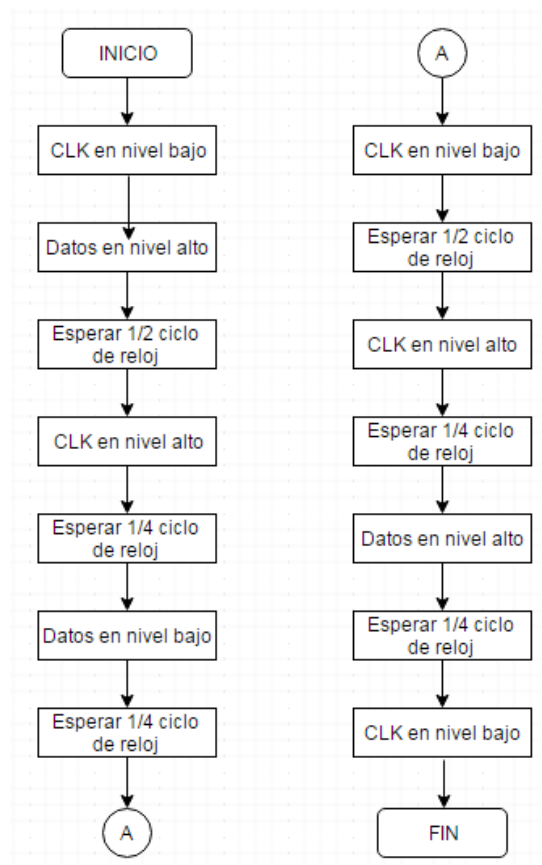
**Fuente:** Geomara Pilco

En la Figura 21. Se puede observar el circuito para la placa de recepción y en la Figura 22. Se muestra la vista 3D de la misma. Adicionalmente en el Anexo 1. Se indica el diseño de la placa de recepción en PDF, para la impresión.

## 2.5.4 PROGRAMACIÓN EN ARDUINO

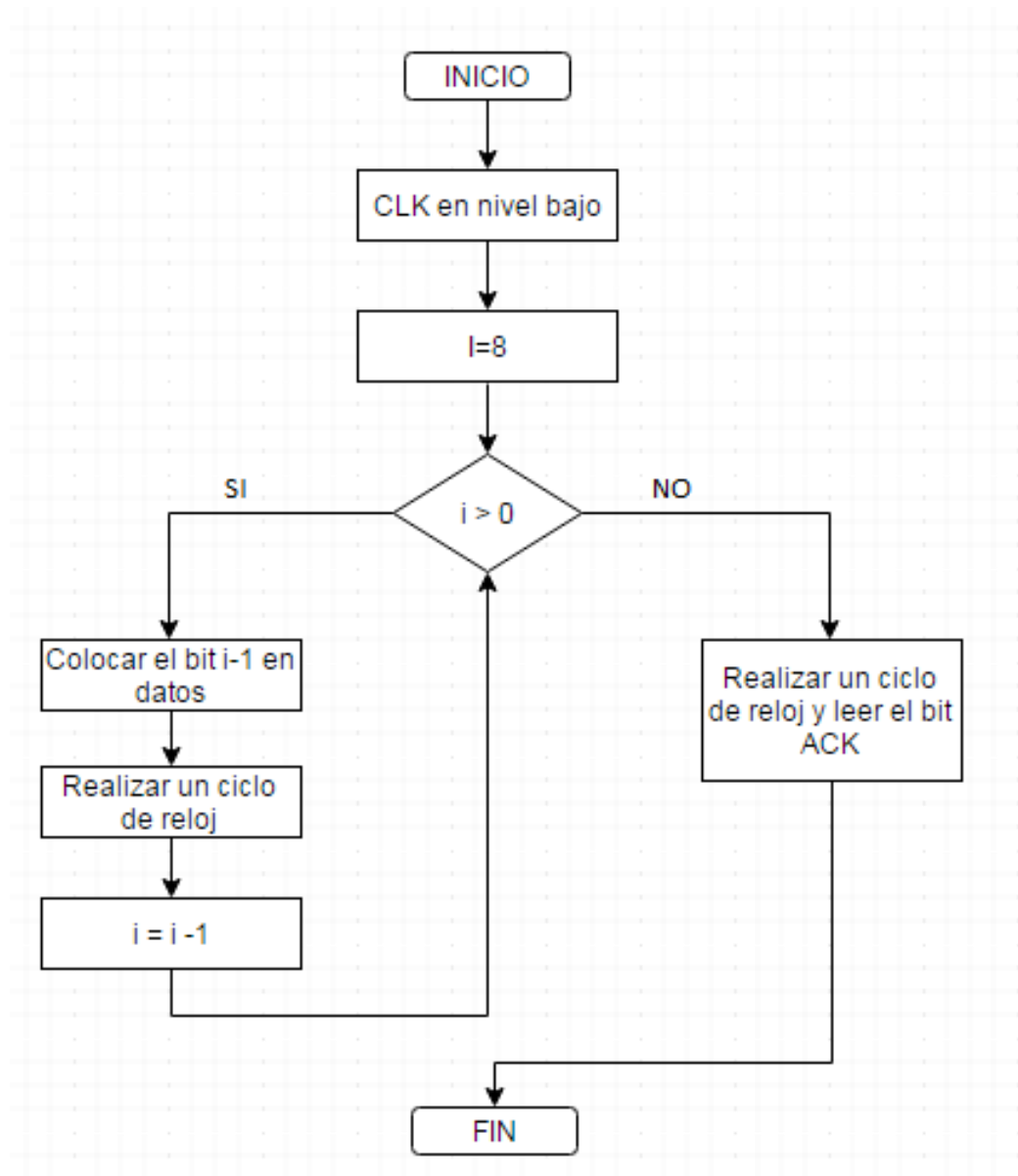
### 2.5.4.1 Programación del sensor de temperatura y humedad relativa SHT75

Para obtener la medición de la temperatura y humedad con el sensor se debe seguir el siguiente proceso: enviar secuencia de inicialización del sensor, enviar la petición de la variable a medir y recibir la información proporcionada por el sensor. En las Figuras 23, 24 y 25 se muestran los diagramas de flujo de la programación realizada que se puede observar en el Anexo 6.



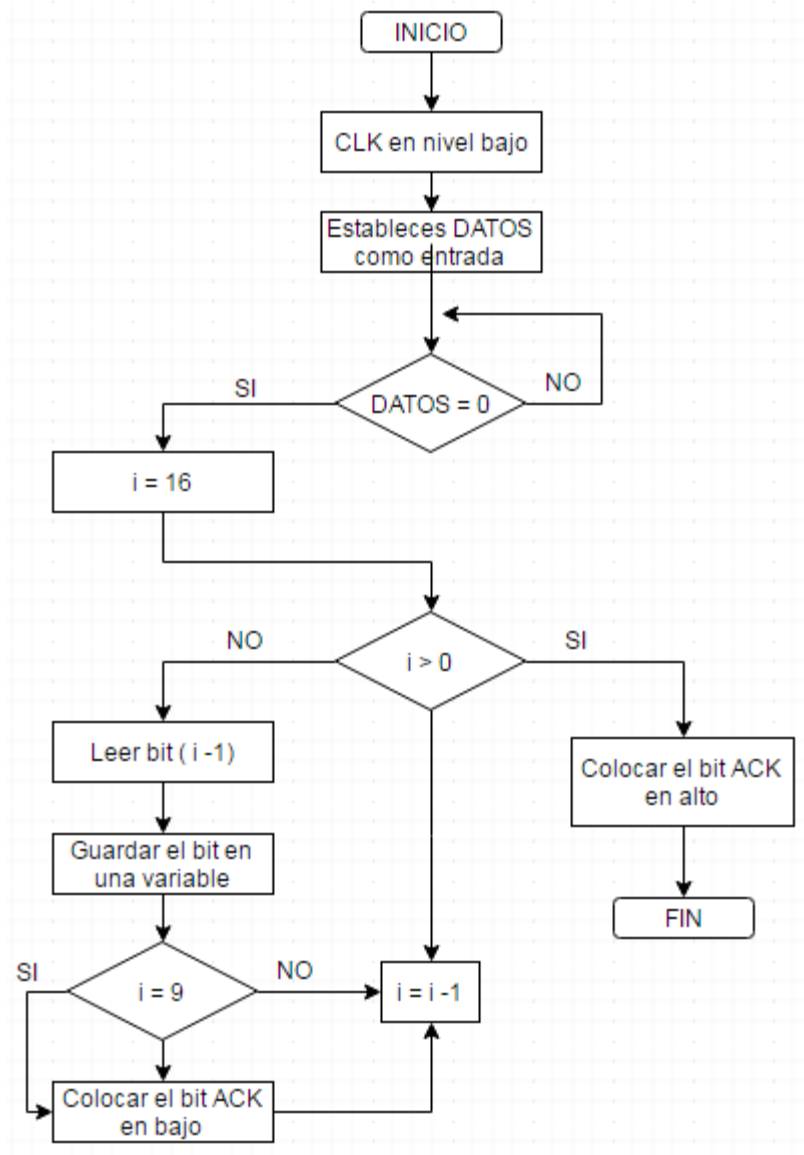
**Figura 23.** Secuencia de inicio de transmisión del SHT75.

**Fuente:** Geomara Pilco



**Figura 24.** Envío de información al SHT75

**Fuente:** Geomara Pilco

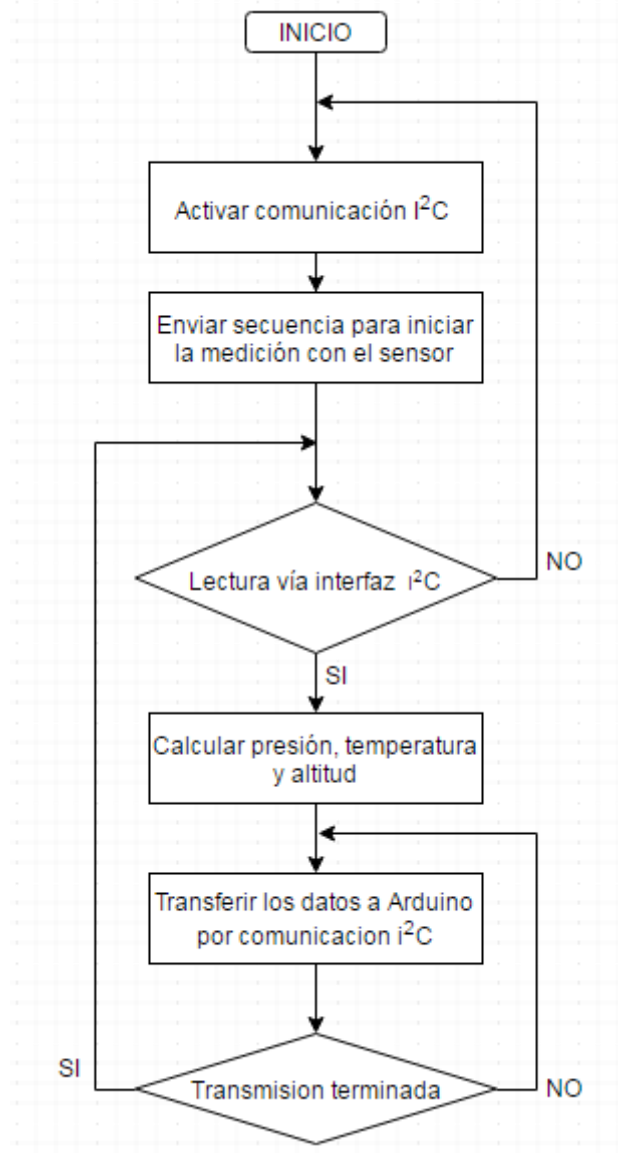


**Figura 25.** Recepción de datos del SHT75

*Fuente: Geomara Pilco*

#### 2.5.4.2 Programación del sensor de presión atmosférica BMP085

El diagrama de flujo de la Figura 26. Indica la lógica con la que trabaja el ATmega de la tarjeta Arduino para obtener la medición de la presión atmosférica y la altura.



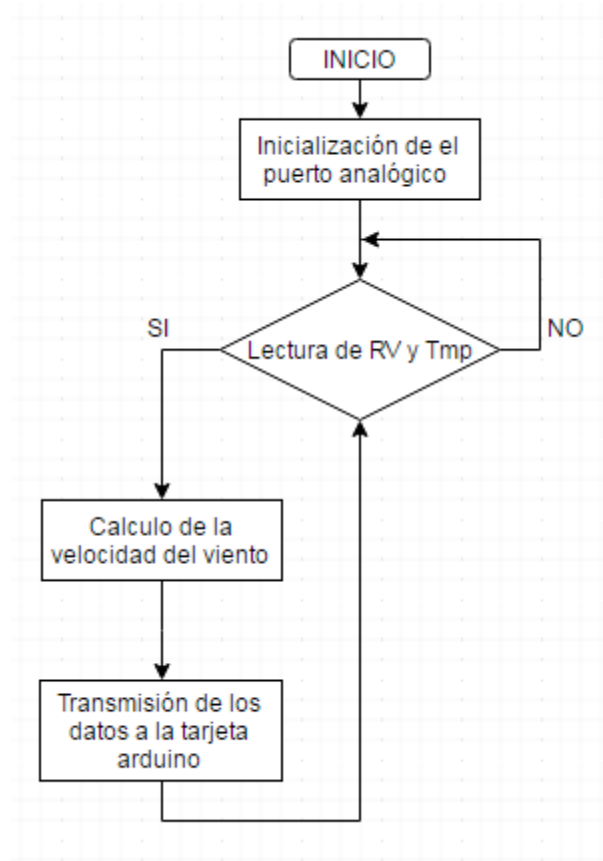
**Figura 26.** Secuencia para la obtención de datos del sensor BMP085

**Fuente:** Geomara Pilco

Para realizar la programación del Barómetro BMP085, se debe tener en cuenta las librerías con las cuales vamos a programar, y la comunicación tipo I<sup>2</sup>C con la cual se obtiene información de presión barométrica, altitud y temperatura.

### 2.5.4.3 Programación para el sensor de velocidad del viento MD0550

El diagrama de flujo de la Figura 27. Indica la lógica con la que trabaja el ATmega de la tarjeta Arduino para obtener la medición de la velocidad del viento.



**Figura 27.** Secuencia para la obtención de la velocidad del viento

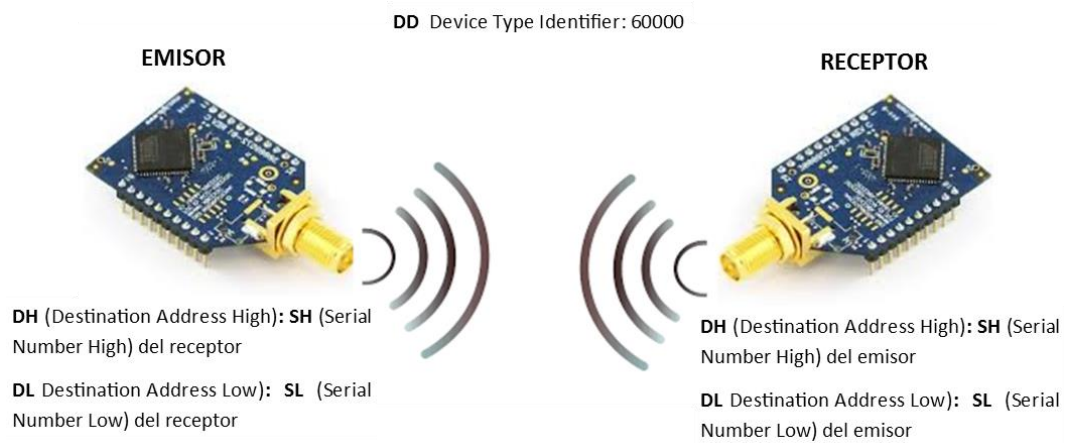
**Fuente:** Geomara Pilco

### 2.5.5 TRANSMISIÓN DE DATOS

La comunicación que el dispositivo requiere es un enlace punto a punto con los Xbee Pro.

Un enlace punto a punto se aplica para un tipo de arquitectura de red específica, se aplica para largas distancias. Los dispositivos que forman parte de esta red actuarán uno como transmisor y otro como receptor, teniendo entre los mismos una comunicación simplex ya que la información se transmitirá en un solo sentido.

Para lograr este enlace es necesario configurar cada uno de los Xbee, esto se realiza usando el Software XCTU que lo proporciona el fabricante. Básicamente se debe poner a los dos dispositivos la misma identificación de red para que estén dentro de la misma, y colocar el SL de un dispositivo en el DH del otro dispositivo, como se muestra en la figura 28. La configuración de los módulos paso a paso se detalla en el Anexo 3.

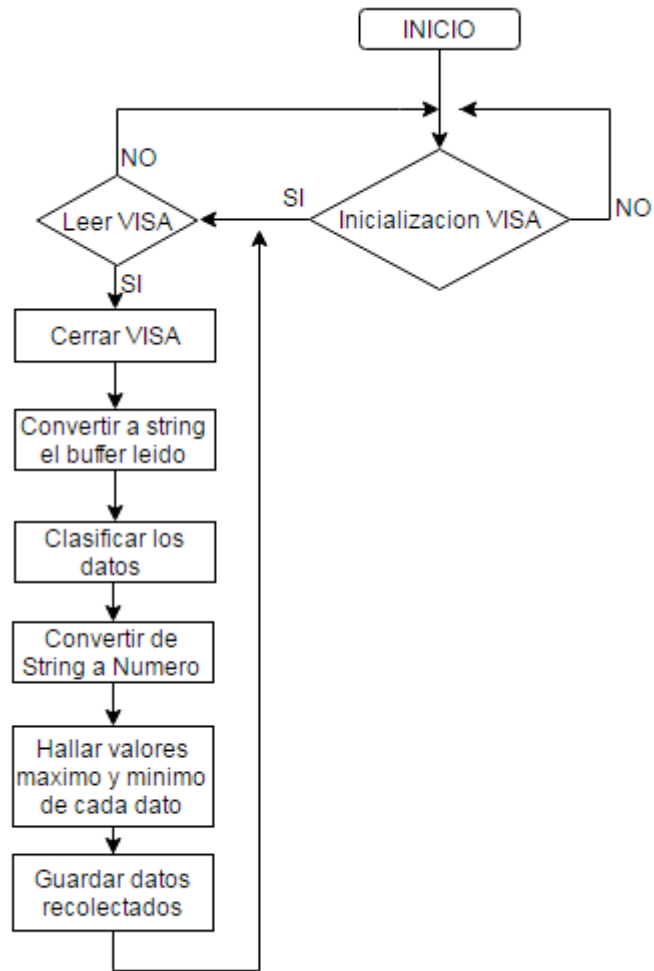


**Figura 28.** Enlace punto a punto con los módulos Xbee

**Fuente:** Geomara Pilco

### 2.5.6 INTERFAZ DE USUARIO EN LABVIEW

El programa en labview para la adquisición de datos consiste básicamente en iniciar la comunicación serial, leer los datos proporcionados por los sensores después clasificarlos usando los identificadores establecidos en la programación en Arduino, convertir estos datos a números para su posterior representación gráfica y procesamiento (valores máximos y mínimos); finalmente se almacenan los datos en un archivo de Excel que registra la hora y fecha de la medición. Las herramientas utilizadas y el procedimiento realizado para la realización del programa en Labview se detallan paso a paso en el Anexo 5.



**Figura 29.** Diagrama de flujo de la programación en labview

**Fuente:** Geomara Pilco

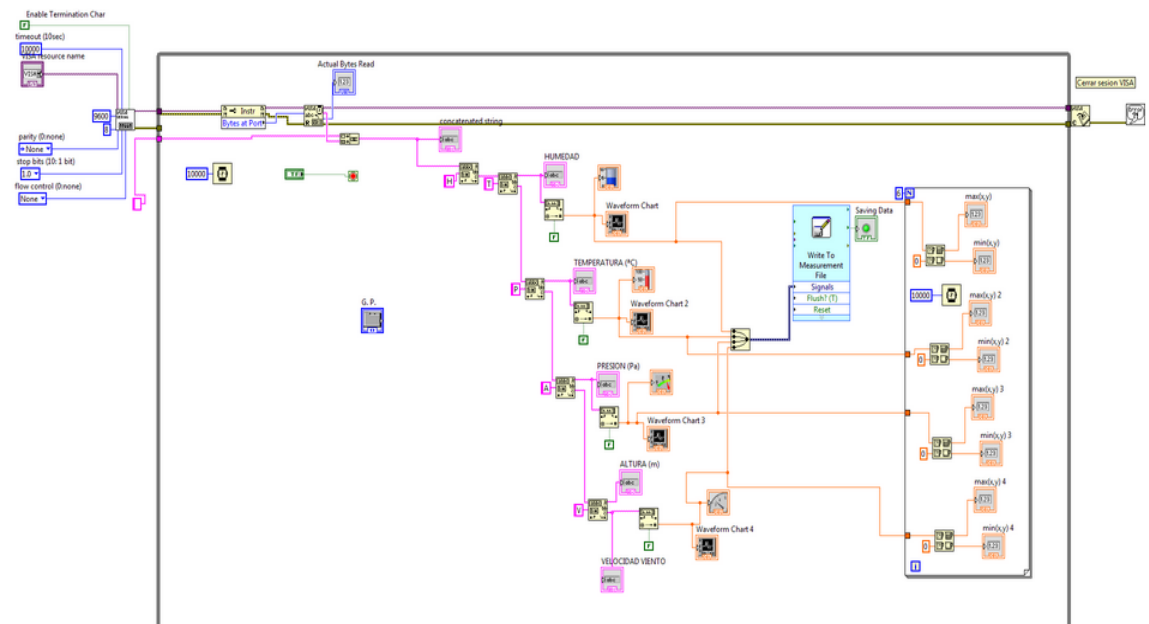


En las figuras 28 y 29 se representan el panel frontal y diagrama de bloques de labview respectivamente.



*Figura 30. Panel frontal del programa en Labview*

*Fuente: Geomara Pilco*



*Figura 31. Diagrama de Bloques del programa en Labview.*

*Fuente: Geomara Pilco*

## 2.6 COMPROBACION DE LA HIPOTESIS

### 2.6.1 PLANTEAMIENTO DE LA HIPOTESIS

#### **Hipótesis Nula Ho:**

El diseño e implementación de una radiosonda permitirá realizar la medición y monitoreo de parámetros meteorológicos.

#### **Hipótesis Alternativa H1:**

El diseño e implementación de una radiosonda no permitirá realizar la medición y monitoreo de parámetros meteorológicos.

### 2.6.2 ESTABLECIMIENTO DE NIVEL DE SIGNIFICANCIA

Para obtener un resultado confiable, se toma un valor de significancia en relación al 95% de confianza equivalente a 1.96; siendo este valor establecido por el criterio del investigador.

### 2.6.3 MUESTRA ENCONTRADA

De la población se toma una muestra de 9 pruebas de medición utilizando el dispositivo elaborado y utilizando otros dispositivos, para así probar su eficiencia. En la tabla 9. Y tabla 10. Se muestran los datos medidos por el dispositivo implementado y por otros dispositivos respectivamente.

**Tabla 9.** Mediciones realizadas con el dispositivo implementado.

HUMEDAD %	TEMPERATURA °C	PRESION Pa	VELOCIDAD V. m/s
73,1	22.5	73511	3,27
73	23	73511	1,79

73,7	15,2	73508	0,81
76	20	73509	0,81
75,1	21,1	73509	6,81
74	15,8	73509	0,82
72,6	19,9	73510	2,45
73,9	20	73510	0,46
73,2	22	73510	1,46

**Tabla 10.** Mediciones realizadas con otros dispositivos.

HUMEDAD %	TEMPERATURA °C	PRESION Pa	VELOCIDAD V. m/s
73,3	22,1	73513	4,1
72,8	22,4	73512	2,3
73,3	16,1	73506	0,7
75,8	20,8	73515	0,8
74,2	21,8	73509	6,86
73,6	15,1	73504	0,82
71,7	20,2	73512	2,52
74,6	20,4	73510	0,53
73,8	21,3	73510	1,54

#### 2.6.4 CALCULO DE LA MEDIA Y LA VARIANZA

Las fórmulas utilizadas para el cálculo de la media y la varianza son las siguientes:

##### MEDIA

$$\bar{x} = \frac{\sum_1^n x}{n}$$

$$\bar{Y} = \frac{\sum_1^n Y}{n}$$

##### VARIANZA

$$\sigma^2 = (X - \bar{X})^2$$

$$\sigma^2 = (Y - \bar{Y})^2$$

En donde:

y = valor medido utilizando la radio sonda

$\bar{y}$  = media utilizando la radio sonda

x = valor medido con otro dispositivo

$\bar{x}$  = media utilizando otro dispositivo

$\sigma^2$  = Varianza

$\sigma$  = Desviación estándar

En las tablas 11, 12, 13 y 14 se muestran los valores del cálculo de la media y desviación de cada uno de los parámetros medidos.

**Tabla 11.** Cálculo de la media y desviación de las mediciones realizadas de humedad.

Con la radiosonda			Con otros dispositivos		
Y	$\bar{y}$	$\vartheta$	x	$\bar{x}$	$\Theta$
73,10	73,84	0,55	73,30	73,68	0,14
73,00	73,84	0,71	72,80	73,68	0,77
73,70	73,84	0,02	73,30	73,68	0,14
76,00	73,84	4,67	75,80	73,68	4,49
75,10	73,84	1,59	74,20	73,68	0,27
74,00	73,84	0,03	73,60	73,68	0,01
72,60	73,84	1,54	71,70	73,68	3,92
73,90	73,84	0,00	74,60	73,68	0,85
73,20	73,84	0,41	73,80	73,68	0,01

**Tabla 12.** Cálculo de la media y desviación de las mediciones realizadas de temperatura.

Con la radiosonda			Con otros dispositivos		
y	$\bar{y}$	$\vartheta$	x	$\bar{x}$	$\vartheta$

22,50	19,94	6,55	22,10	20,02	4,33
23,00	19,94	9,36	22,40	20,02	5,66
15,20	19,94	22,47	16,10	20,02	15,37
20,00	19,94	0,01	20,80	20,02	0,61
21,10	19,94	1,35	21,80	20,02	3,17
15,80	19,94	17,14	15,10	20,02	24,21
19,90	19,94	0,00	20,20	20,02	0,03
20,00	19,94	0,00	20,40	20,02	0,14
22,00	19,94	4,24	21,30	20,02	1,64

**Tabla 13.** Cálculo de la media y desviación de las mediciones realizadas de presión.

Con la radiosonda			Con otros dispositivos		
y	$\bar{y}$	$\vartheta$	x	$\bar{x}$	$\Theta$
73511	73509,67	1,77	73513	73510,33	7,13
73511	73509,67	1,77	73512	73510,33	2,79
73508	73509,67	2,79	73506	73510,33	18,75
73509	73509,67	0,45	73515	73510,33	21,81
73509	73509,67	0,45	73509	73510,33	1,77
73509	73509,67	0,45	73504	73510,33	40,07
73510	73509,67	0,11	73512	73510,33	2,79
73510	73509,67	0,11	73510	73510,33	0,11
73510	73509,67	0,11	73510	73510,33	0,11

**Tabla 14.** Cálculo de la media y desviación de las mediciones realizadas de velocidad del viento.

Con la radiosonda			Con otros dispositivos		
y	$\bar{y}$	$\vartheta$	x	$\bar{x}$	$\Theta$
3,27	2,08	1,42	4,10	2,24	3,46
1,79	2,08	0,08	2,30	2,24	0,01
0,81	2,08	1,61	0,70	2,24	2,37
0,81	2,08	1,61	0,80	2,24	2,07
6,81	2,08	22,37	6,86	2,24	21,34
0,82	2,08	1,59	0,82	2,24	2,02
2,45	2,08	0,14	2,52	2,24	0,08
0,46	2,08	2,62	0,53	2,24	2,92
1,46	2,08	0,38	1,54	2,24	0,49

### 2.6.5 CALCULO DEL VALOR CRITICO O $Z_c$

La fórmula para el cálculo del valor crítico es la siguiente:

$$Z_c = \frac{\bar{X} - \bar{Y}}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

En donde

$\bar{X}$  y  $\bar{Y}$  : son medias obtenidas en las muestras

$\sigma_1^2$  y  $\sigma_2^2$  : varianzas

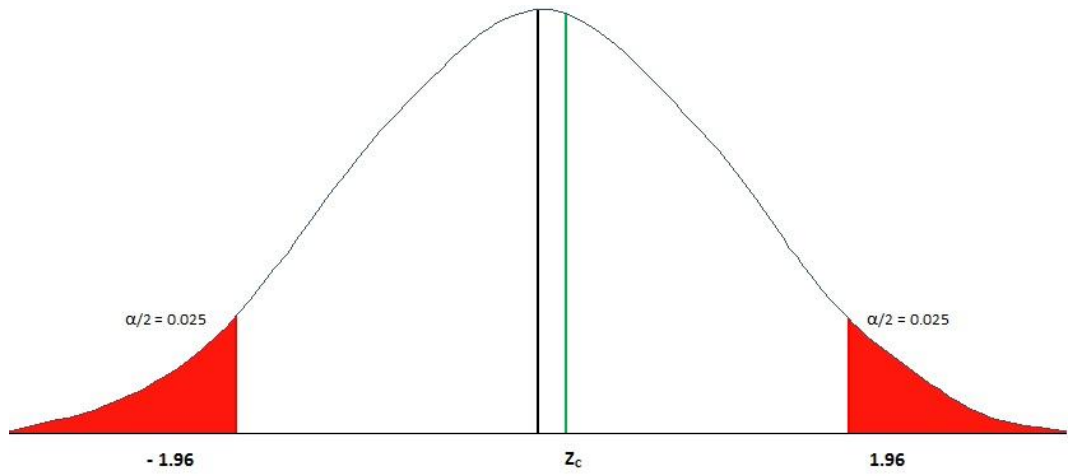
$n_1$  y  $n_2$  : número de muestras

$$Z_c = \frac{2,24 - 2,08}{\sqrt{\frac{3,86}{9} + \frac{3,54}{9}}}$$

$$Z_c = 0,18 \text{ valor critico}$$

### 2.6.6 REPRESENTACION GRAFICA DE LA REGLA DE DECISION

Tomando en cuenta las medidas recolectadas utilizando la radio sonda y utilizando otros dispositivos tomando en cuenta sus promedios el valor crítico se encuentra en la zona de aceptación de la hipótesis tomando en cuenta de que la confiabilidad en este caso es del 95% por lo tanto la implementación de la radio sonda es eficiente.



*Figura 32. Grafica de distribución normal*

*Fuente: Geomara Pilco*

## CAPITULO III

### 3. RESULTADOS

Para la presentación de los resultados se utilizó la comparación entre las medidas obtenidas utilizando otros dispositivos y el dispositivo diseñado, los parámetros meteorológicos medidos son la velocidad del viento en m/s, la humedad en %, la temperatura en grados centígrados, la Presión Atmosférica en Pascales

Comparando las medidas arrojadas tanto por otros dispositivos y por la radiosonda diseñada se puede ver claramente que las medidas promedio tienen una diferencia mínima en cada uno de los parámetros meteorológicos medidos, tomando en cuenta que al utilizar otros dispositivos se realiza la medición en diferentes tiempos y en forma particular, mientras que la medición de los diferentes parámetros meteorológicos utilizando la radiosonda diseñada se efectúa de manera simultánea, convirtiéndose la radiosonda en un dispositivo que permite el ahorro de tiempo y esfuerzo, evidenciando de esta manera que el dispositivo diseñado permite la medición de los parámetros meteorológicos de manera correcta en distintos lugares.

#### 3.1 Medición usando la Radiosonda

Para realizar una medición lo primero que se debe hacer es verificar que la carga de la batería de la caja de Tx sea correcta. Véase la Figura 33.

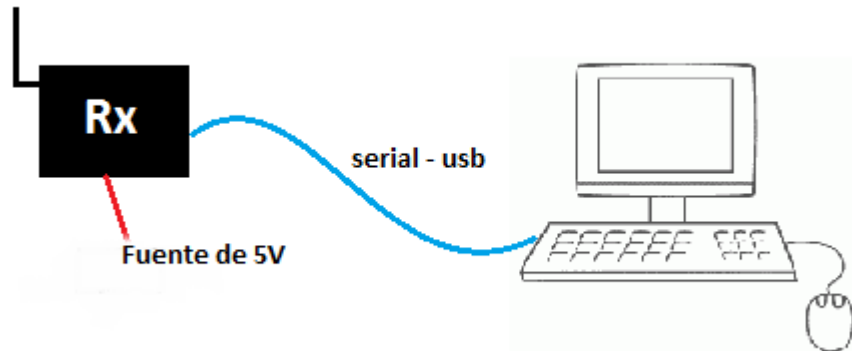


*Figura 33. Verificar carga de la batería*

*Fuente: Geomara Pilco*



Por otra parte la caja de recepción debe estar conectada a una fuente de 5V y conectada mediante el cable serial - usb al computador donde estará instalado Labview, como se muestra en la Figura 34.



**Figura 34.** *Conexión de la caja de Recepción*

**Fuente:** *Geomara Pilco*

Se coloca la caja de Tx en el punto deseado de medición, se verifica que exista línea de vista entre las dos cajas por tratarse de un enlace inalámbrico punto a punto.

Finalmente se ejecuta el la interfaz de usuario en labview para la adquisición de datos, se selecciona el puesto COM donde está conectada la caja de Rx y se especifica el archivo de Excel donde se guardara la información.

## CAPITULO IV

### 4. CONCLUSIONES Y RECOMENDACIONES

#### 4.1 Conclusiones

Culminado el trabajo de investigación se concluye lo siguiente:

- ✓ Previo a la utilización de un enlace inalámbrico punto a punto es imprescindible la comprobación de la existencia de visión directa entre las antenas.
- ✓ Los módulos Xbee a pesar de su relativa simplicidad comparada con otros estándares, proveen confiabilidad, flexibilidad y escalabilidad.
- ✓ La utilización del software de adquisición de datos, permite un estudio continuo de las variables meteorológicas medidas, lo cual posibilita un análisis inmediato y estadístico del comportamiento de las mismas.
- ✓ En las pruebas realizadas en la ciudad de Riobamba haciendo uso del dispositivo desarrollado y de sensores profesionales, se verifico que existe un 95% de efectividad de las mediciones realizadas con el dispositivo implementado.

## 4.2 Recomendaciones

- ✓ Si se requiere transmitir la información a una velocidad mayor a 250Kbps, hay que cambiar los dispositivos de transmisión – recepción porque esta es su velocidad máxima.
- ✓ Una radiosonda debe tener un peso no mayor a 250 gramos, para trabajar de manera óptima.
- ✓ Actualizar el firmware de los módulos ZigBee según la disponibilidad de nuevas versiones publicadas por el fabricante, ya que ofrecen una reducción de tiempos de respuesta, por ende optimizan la comunicación.
- ✓ Durante una medición se debe considerar que el enlace puede presentar atenuaciones de la señal debidas a la absorción y dispersión causadas por hidrometeoros como la lluvia, la nieve, el granizo o la niebla.

## **CAPITULO V**

### **5. PROPUESTA**

#### **5.1 Título de la propuesta**

Diseño de una radiosonda para la medición de parámetros meteorológicos

#### **5.2 Introducción**

En la actualidad la polución atmosférica ha aumentado considerablemente lo que obliga a los entes investigadores a obtener la instrumentación necesaria para realizar estudios meteorológicos principalmente sobre condiciones climáticas y la calidad del aire en nuestro entorno. La Universidad Nacional de Chimborazo siendo una entidad educativa cuya misión es fomentar la investigación, mediante la ejecución del proyecto “Programa integrado de monitoreo de control de calidad del aire en la ciudad de Riobamba usando modelos matemáticos” aspira contribuir en el estudio de la calidad del aire de nuestra ciudad.

Por la dificultad de la adquisición de instrumentación meteorológica en la actualidad no existen estudios de este tipo para la ciudad de Riobamba; con la implementación de la radiosonda en el “Programa integrado de monitoreo de control de calidad del aire en la ciudad de Riobamba usando modelos matemáticos” de la Universidad Nacional de Chimborazo en el futuro se pretende fomentar el desarrollo de este tipo de proyectos.

## **5.3 Objetivos**

### **5.3.1 Generales:**

Diseñar e implementar una radiosonda de transmisión de parámetros meteorológicos.

### **5.3.2 Específicos:**

- a) Diseñar e implementar una radiosonda que permita medir temperatura, humedad, presión y velocidad del viento.
- b) Determinar la forma de transmitir las señales desde la radiosonda a una base, así como también almacenar los datos en la misma.
- c) Desarrollar un Software en Labview para la adquisición y procesamiento de las señales.

## **5.4 Fundamentación Científico –Técnica**

El dispositivo está conformado por tres sensores, los cuales permiten medir la velocidad del viento, la humedad, la temperatura, la presión atmosférica y determinar la altitud.

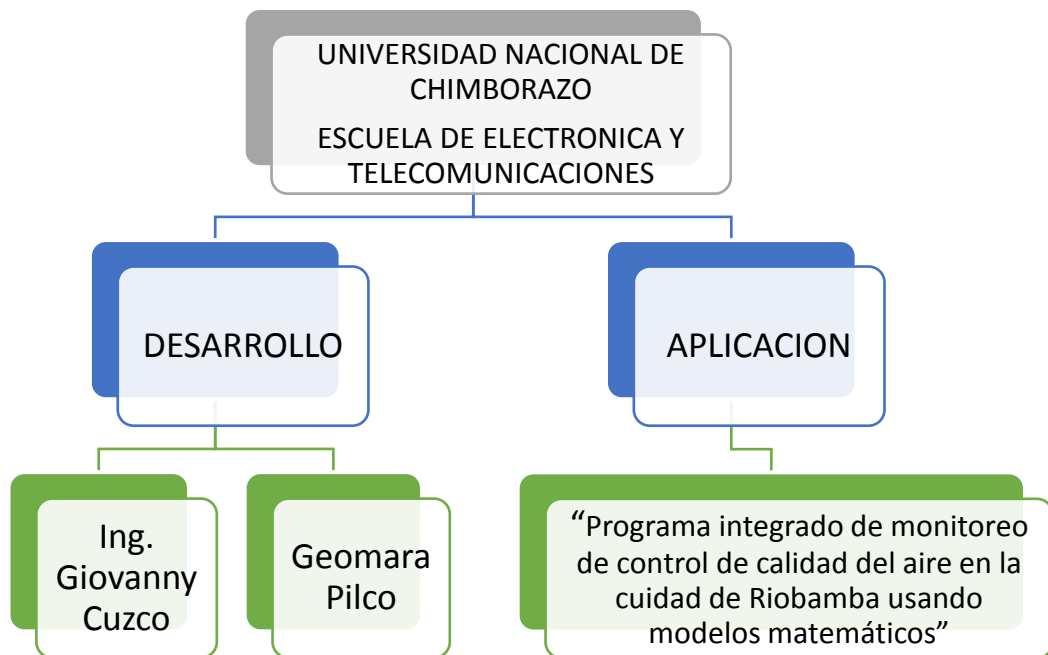
Como parte principal del prototipo se tiene la tarjeta Arduino UNO, basada en el ATmega328P; esta tarjeta será la encargada de analizar los datos recibidos de los sensores para luego poder enviar esta información vía serial al transmisor XBee, el cual transmitirá esta información de forma inalámbrica hacia el otro Xbee que actúa como receptor.

Para visualizar y analizar los datos recibidos, se diseñó un programa en labview el cual muestra las mediciones en tiempo real, se obtiene valores máximos y mínimos del último minuto y permite guardar estos datos en un archivo de Excel.

## 5.5 Descripción de la propuesta

Indicar en qué consiste la propuesta, el plan de acción que seguirá que para cumplir los objetivos, el tiempo requerido para desarrollarlas y los recursos que se utilizarán.

## 5.6 Diseño Organizacional.



## 5.7 Monitoreo y Evaluación de la propuesta

El monitoreo y la evaluación de la propuesta se la realizará a través de más mediciones que realice el “Programa integrado de monitoreo de control de calidad del aire en la ciudad de Riobamba usando modelos matemáticos” de la Universidad Nacional de Chimborazo.

Estas mediciones se harán con la utilización del prototipo en tiempos más prolongados y según las necesidades del proyecto de la Universidad para poder observar si su aplicación es útil o no durante el desarrollo de esta clase de proyectos.

Además, dentro de las mediciones que se realice, se propondrá la intervención de estudiantes de la carrera de electrónica y telecomunicaciones para que esta propuesta sea la base para que se siga investigando e implementando este tipo de dispositivos.

## 6. BIBLIOGRAFÍA

1. Ramón Pallás Areny. Sensores y Acondicionadores de Señal, Editorial Marcombo, 2003.
2. J. Fraden, Handbook of Modern Sensors, AIP Press third edition 2003.
3. David Ashlock, Serie de Fundamentos de Mediciones con Sensores, National Instruments, 2011
4. Ing. Edwin J., CLASIFICACIÓN DE SENSORES, Universidad Santiago de Cali Ortega 2010
5. CORDOBA H., Anemómetro, Argentina, 2008.
6. Reyes, *Microcontroladores PIC y programación en Basic*. Quito, 2008.
7. Rodríguez, W. *Software libre para educación y educación en Ingeniería*. Bogotá – Colombia, 2014
8. Arlanzón David, Introducción a la plataforma Open Source Arduino para aplicaciones domóticas y automatización, Catalunya – España, 2012.
9. Banzi, M. *Arduino*. Obtenido de la página de Arduino, 2012.

### Webgrafia

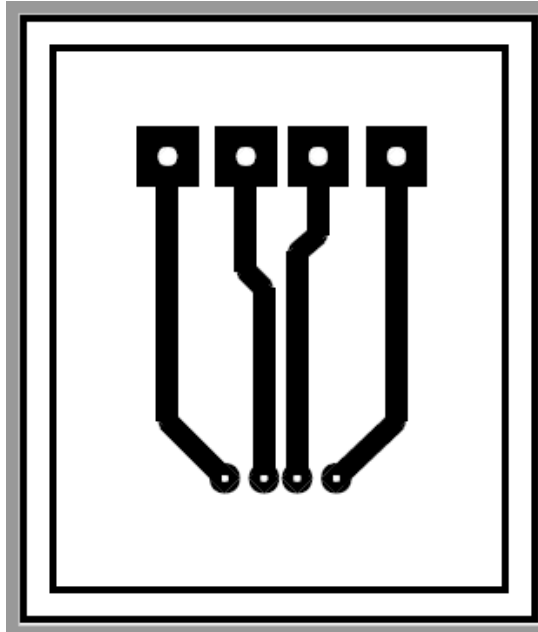
- ✓ <http://www.senamhi.gob.pe/?p=1003>
- ✓ <https://www.arduino.cc>
- ✓ [https://www.wmo.int/pages/prog/dra/documents/VCP\\_factsheets\\_5\\_UpperAir\\_es.pdf](https://www.wmo.int/pages/prog/dra/documents/VCP_factsheets_5_UpperAir_es.pdf)
- ✓ <http://artemisa.unicauca.edu.co/~gavasquez/res/Sensores.pdf>



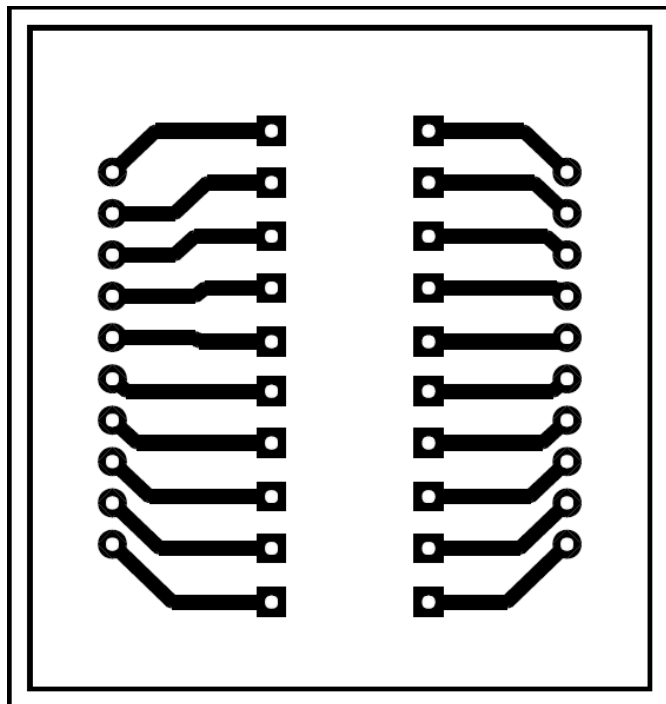
- ✓ [http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena11/4quincena11\\_contenidos\\_3d.htm](http://recursostic.educacion.es/secundaria/edad/4esotecnologia/quincena11/4quincena11_contenidos_3d.htm)
- ✓ <http://electronicaanalog.blogspot.com/2011/12/ldr-y-termistor.html>
- ✓ [http://www.tecnologiatecnica.com.ar/sistemadecontrol/index%20sistemasdecontrol\\_archivos/Page381.html](http://www.tecnologiatecnica.com.ar/sistemadecontrol/index%20sistemasdecontrol_archivos/Page381.html)
- ✓ <http://www.ni.com/white-paper/13034/es/>
- ✓ <http://www.ciese.org/curriculum/weatherproj2/es/docs/anemometro.shtml>
- ✓ <http://www.profesaulosuna.com/data/files/ENERGIA%20RENOVABLES/INSTR%20ER/1.3%20velocidad%20del%20viento.pdf>
- ✓ <http://www.ni.com/>
- ✓ <http://electronilab.co/>

## 7. APÉNDICES Y ANEXOS

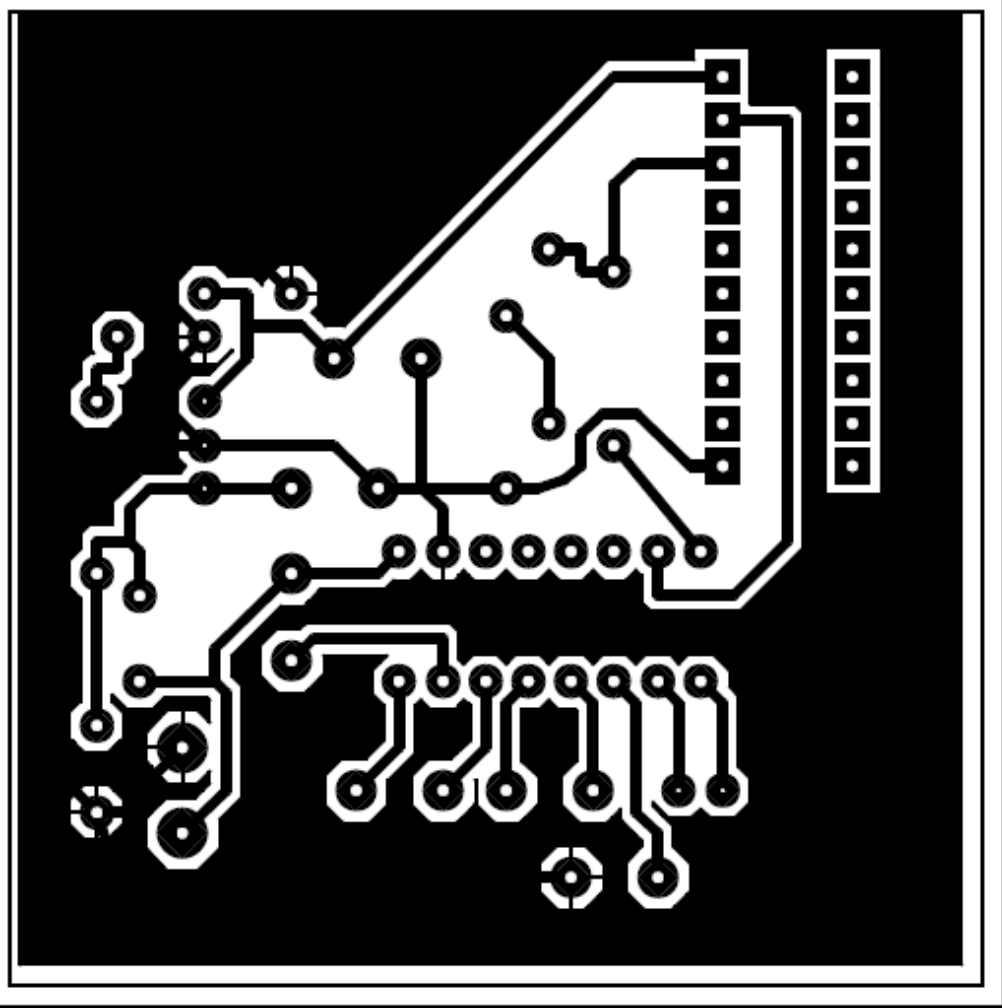
### ANEXO 1. DISEÑO DE LAS PLACAS EN PDF



Diseño de la placa del sensor SHT75 PDF.



Diseño de la placa para de los módulos XBee en PDF



Diseño de la placa de recepción PDF.

**ANEXO 2. CAJAS DE TRANSMISIÓN Y RECEPCIÓN ENSAMBLADAS.**



Caja de transmisión de datos ensamblada.

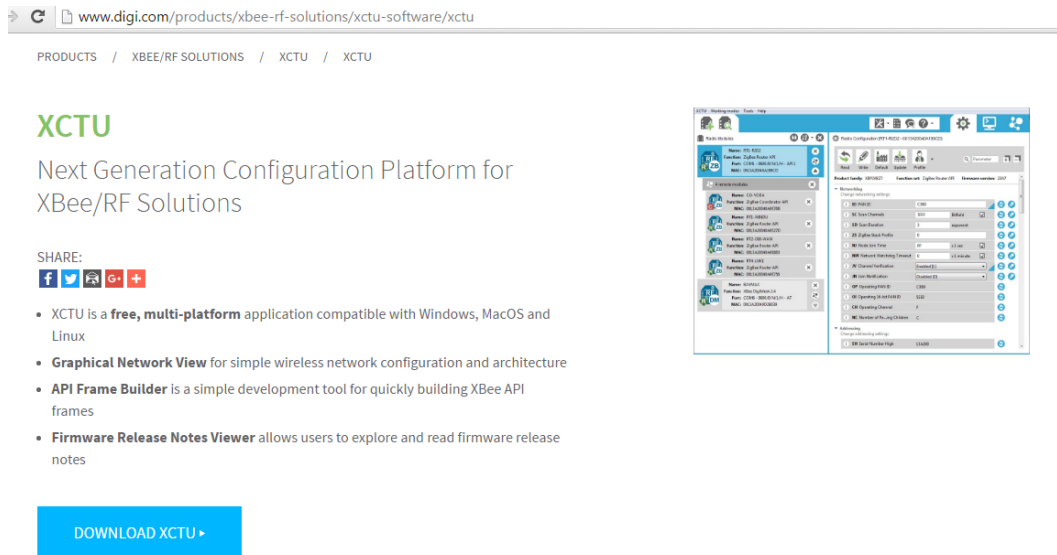


Caja de recepción de datos ensamblada.

### ANEXO 3. CONFIGURACION MODULOS XBEE

A continuación se detallan los pasos a seguir para la configuración:

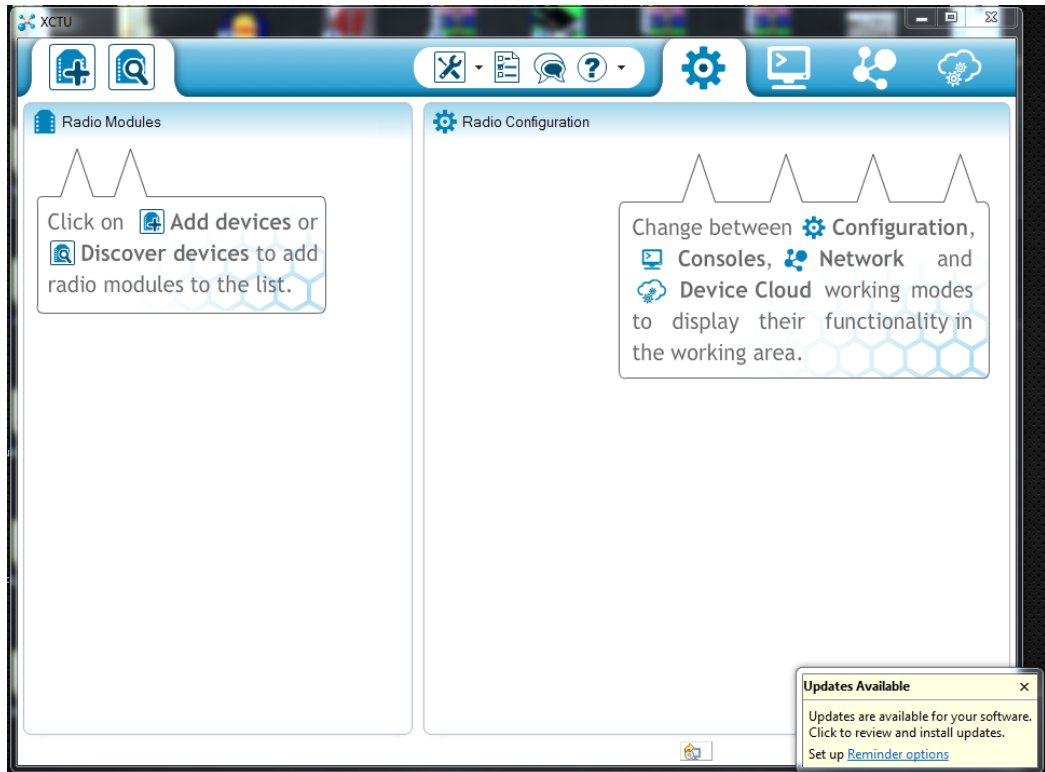
- ✓ Lo primero es descargar XCTU. Exe de la página oficial del fabricante.



*Figura 35. Página de Digi para descargar XCTU.exe*

*Fuente: Geomara Pilco*

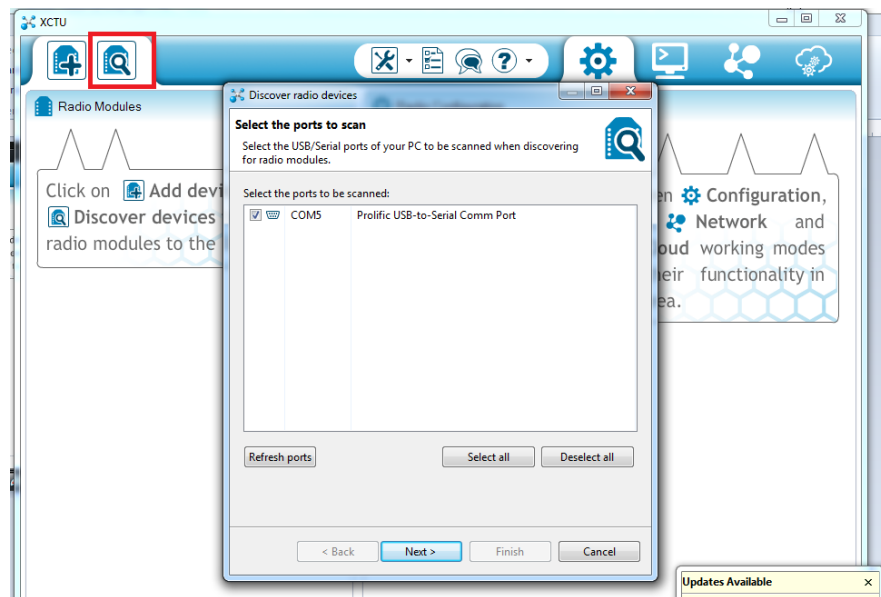
- ✓ Una vez instalado el archivo descargado, se procede a ejecutar el programa y nos aparece la siguiente ventana.



**Figura 36.** Ventana principal de XCTU

*Fuente: Geomara Pilco*

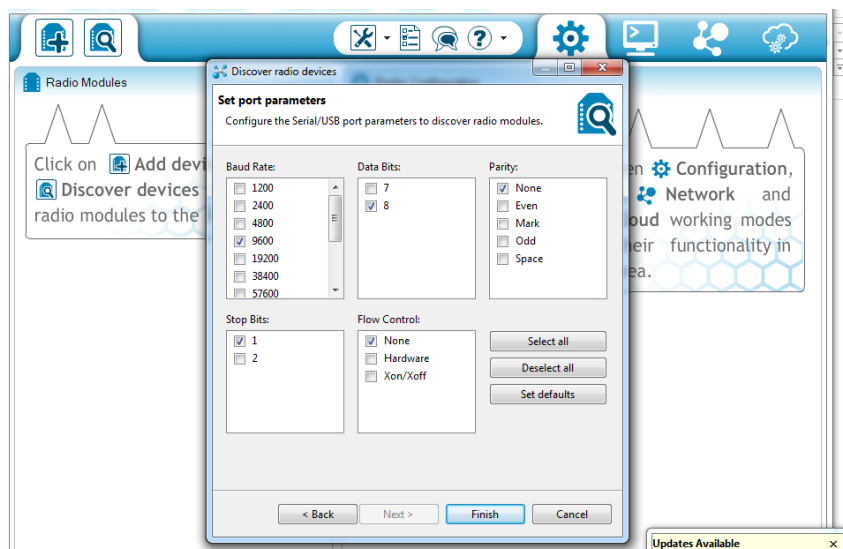
- ✓ Se da click en el icono de “Discover radio devices”, y escanea los puertos COM conectados en el computador.



**Figura 37.** Ventana donde escanea los puertos COM conectados

*Fuente: Geomara Pilco*

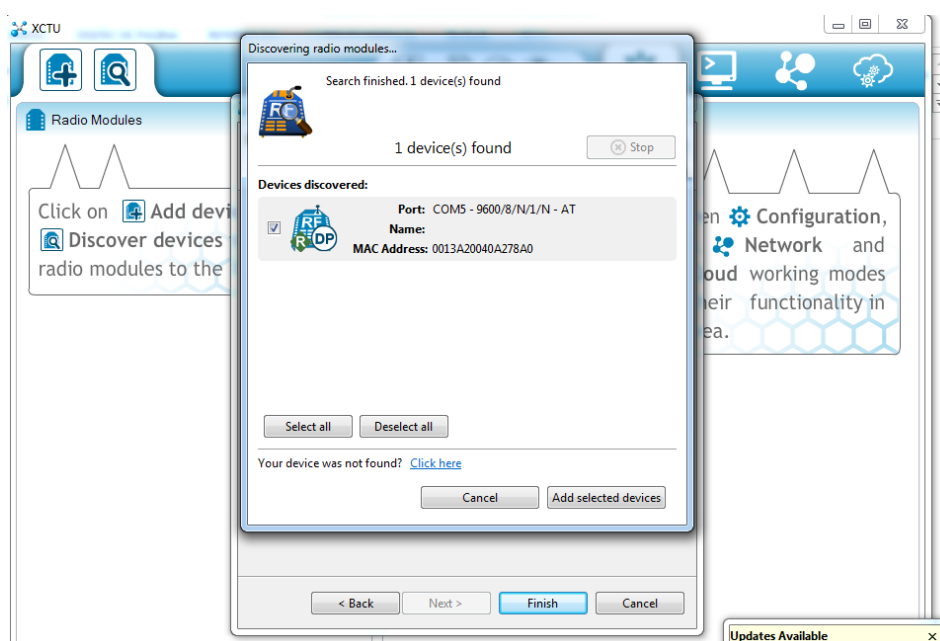
- ✓ Seleccionada la casilla del puerto COM donde está conectado el módulo Xbee, se da clic en "next" y aparece la siguiente ventana, en donde se pueden configurar los parámetros de la conexión del puerto COM.



**Figura 38.** Configuración de los parámetros de la conexión del puerto COM

**Fuente:** Geomara Pilco

- ✓ Se da clic en "Finish" y aparecen las características básicas del módulo conectado en el puerto seleccionado.

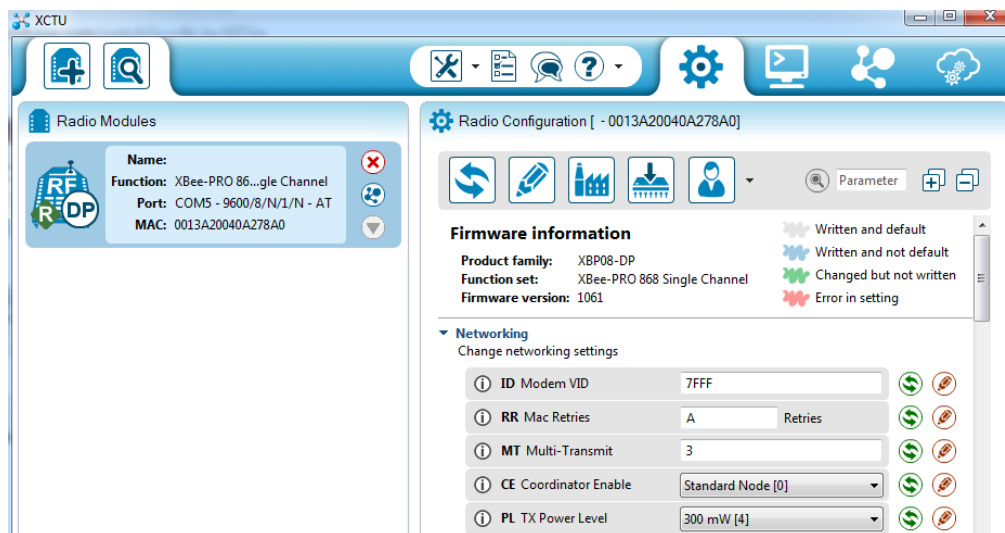


**Figura 39.** Características básicas del módulo conectado

**Fuente:** Geomara Pilco



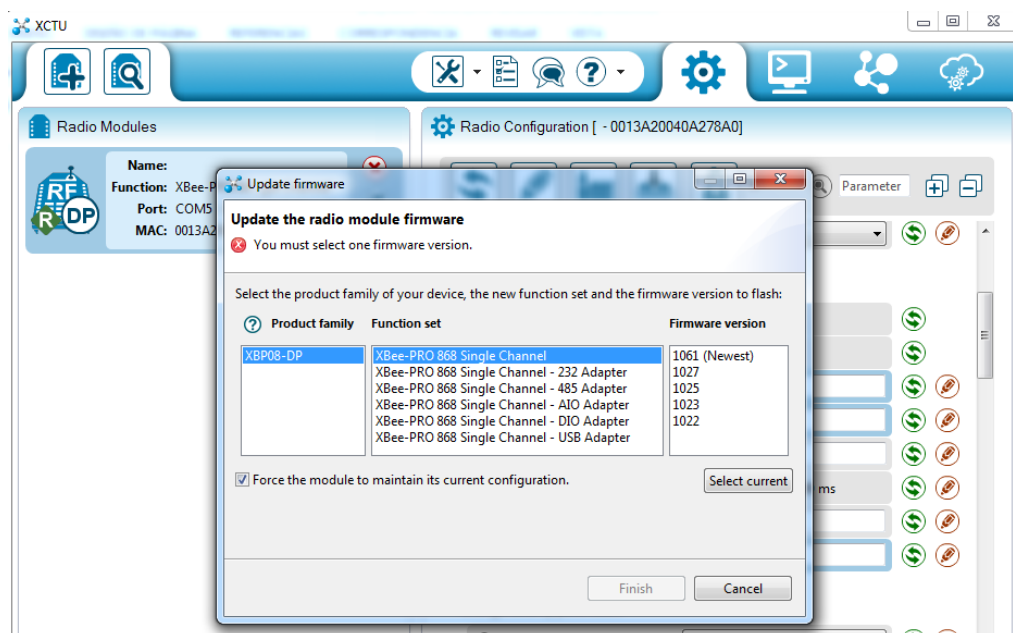
- ✓ Clic en “Add selected devices” y aparece el módulo a la izquierda de la ventana, se da click sobre el mismo y en la parte derecha aparece la configuración actual del módulo.



**Figura 40.** Módulo conectado y su configuración actual

*Fuente: Geomara Pilco*

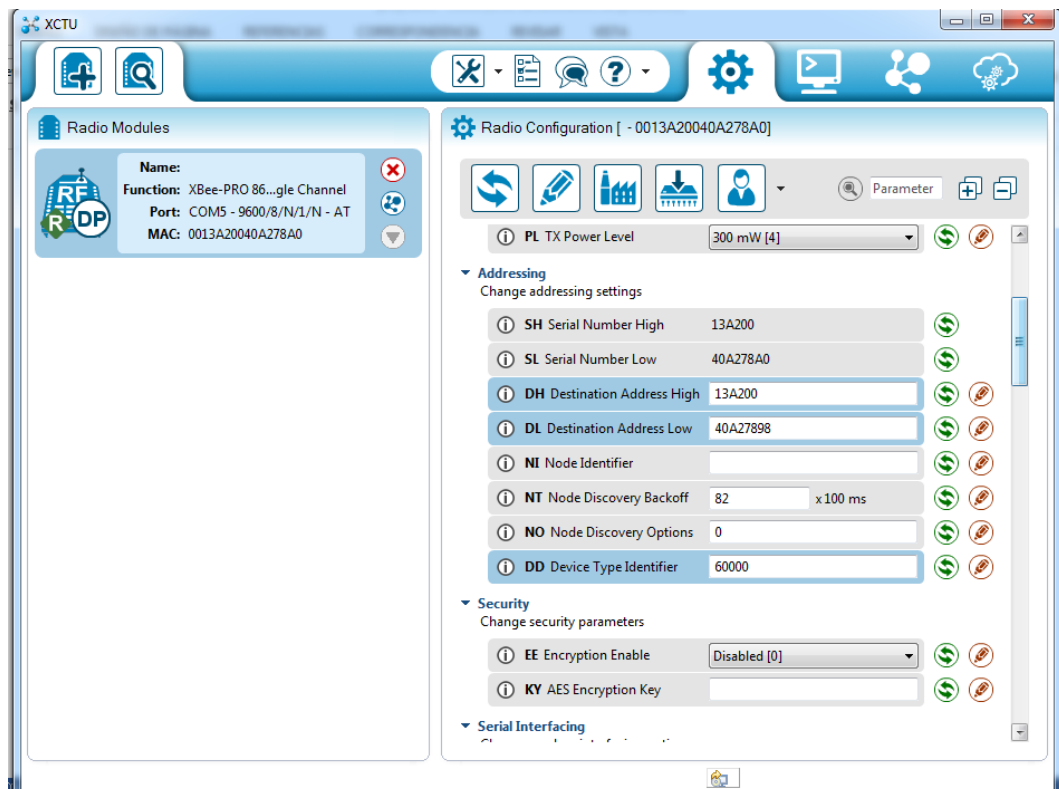
- ✓ En la opción “Update Firmware” hay que seleccionar el tipo de función que tendrá el módulo, en este caso single channel.



**Figura 41.** Firmwares disponibles para el módulo

*Fuente: Geomara Pilco*

✓ En cada uno de los módulos hay que identificar el “Serial Number High” y el “Serial Number Low”, que son distintos en cada dispositivo. En la “Destination Address High” de uno de los módulos colocar el “Serial Number High” del otro modulo y en la “Destination Address Low” colocar el “Serial Number Low” del otro modulo. En el “Decive Type Identifier” colocar el mismo valor en los dos módulos. Se da clic en escribir para grabar la configuración en cada uno de los módulos y con esto se consigue un enlace punto a punto entre los módulos Xbee pro.



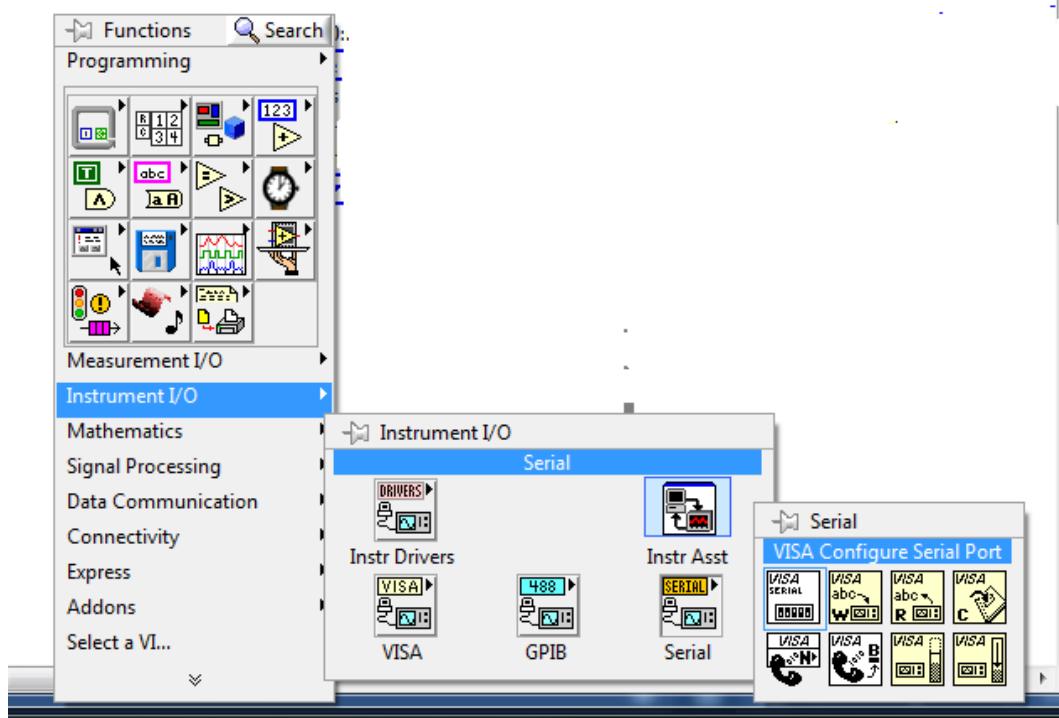
**Figura 42.** Configuración del módulo Xbee

**Fuente:** Geomara Pilco

## ANEXO 4. SOFTWARE EN LABVIEW

### LECTURA DEL PUERTO SERIAL

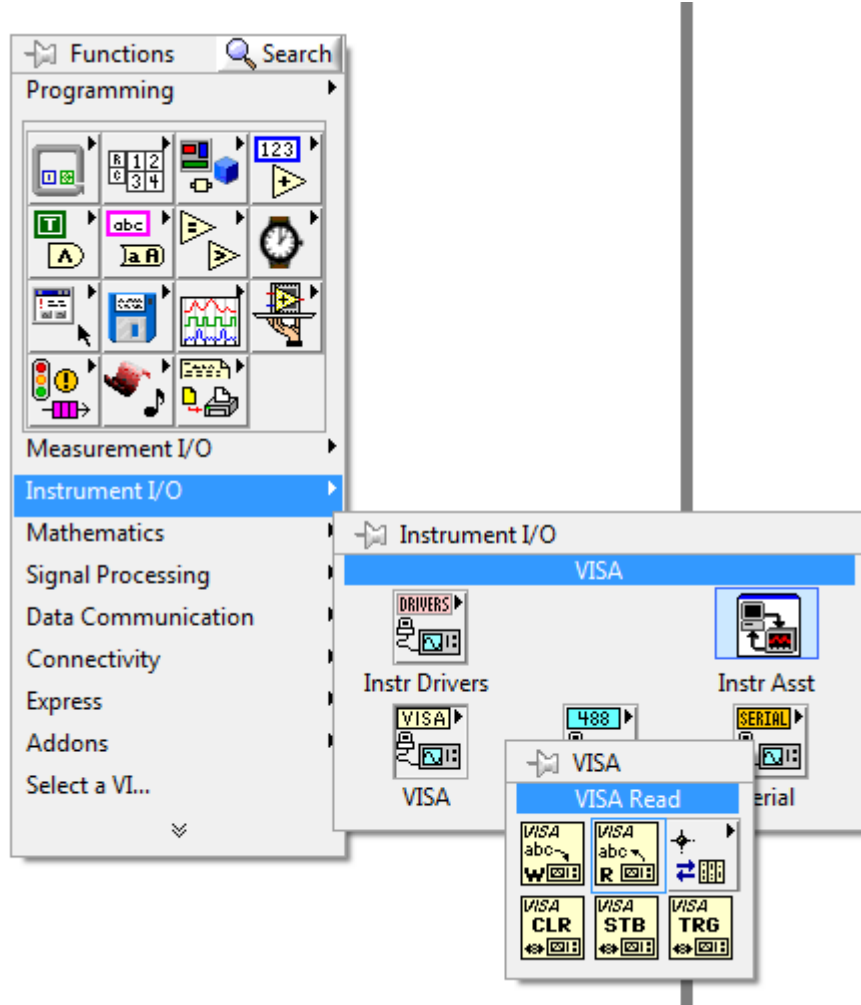
- ✓ Para tener acceso al puerto serial se debe iniciar una sesión VISA. La configuración del tipo de comunicación serial se hace con “VISA configure serial port”, que se puede encontrar en Functions - Instrument I/O – Serial - VISA configure serial port.



*Figura 43. Paleta de funciones de VISA*

*Fuente: Geomara Pilco*

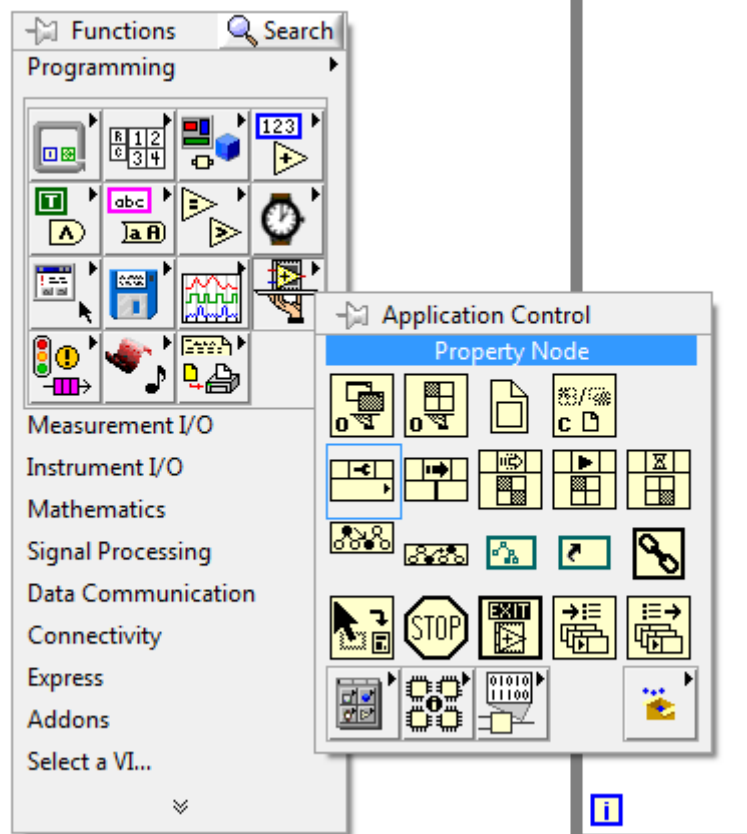
- ✓ Una vez inicializada la sesión VISA, se procede a configurar la lectura. Para lo cual se utiliza “VISA Read”.



*Figura 44. Paleta de funciones VISA Read*

*Fuente: Geomara Pilco*

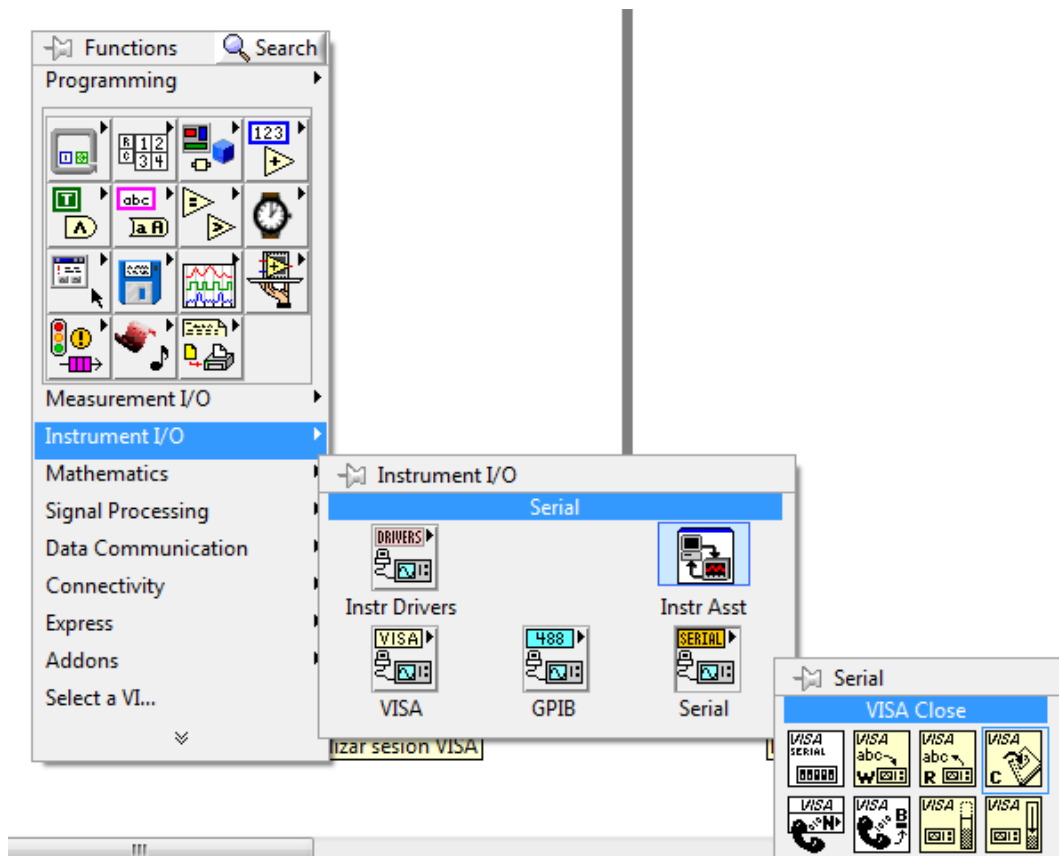
- ✓ El nodo de “byte count” recibe el tamaño de buffer que se escribió en el puerto. Para identificarlo, se coloca un Property Node ubicado en Functions – Programming - Property Node. Su nodo de referencia se conecta a la sesión VISA creada y luego, en property node se da un click para seleccionar Serial Settings - Number of bytes at serial port.



**Figura 45.** Paleta de funciones Property Node

**Fuente:** Geomara Pilco

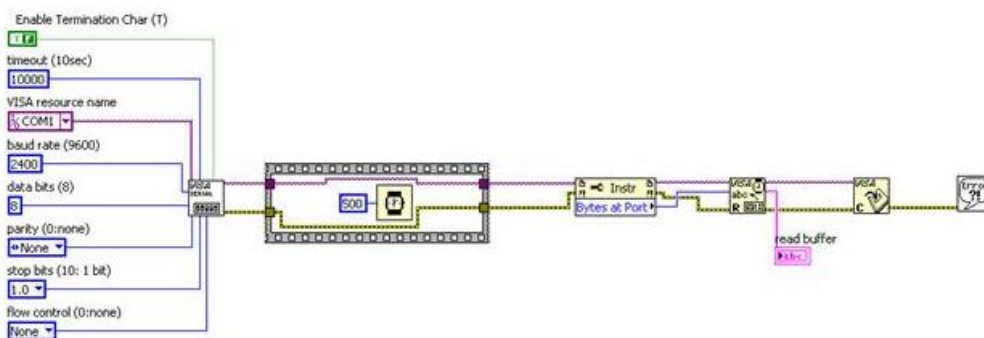
- ✓ Por último se debe cerrar la sesión VISA para liberar el puerto y poder dar otra función. Esto se logra con “VISA close” en Functions - Instrument I/O -Serial - Visa Close,



*Figura 46. Paleta de funciones VISA Close*

*Fuente: Geomara Pilco*

- ✓ La programación para leer el puerto serial seria el siguiente.

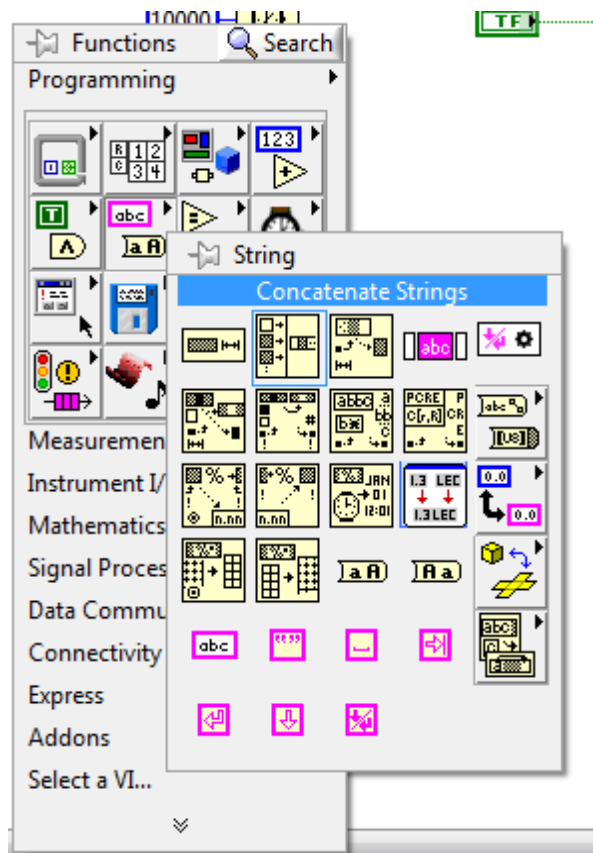


*Figura 47. Lectura puerto serial*

*Fuente: National Instruments*

## PROCESAMIENTO DE LOS DATOS RECIBIDOS

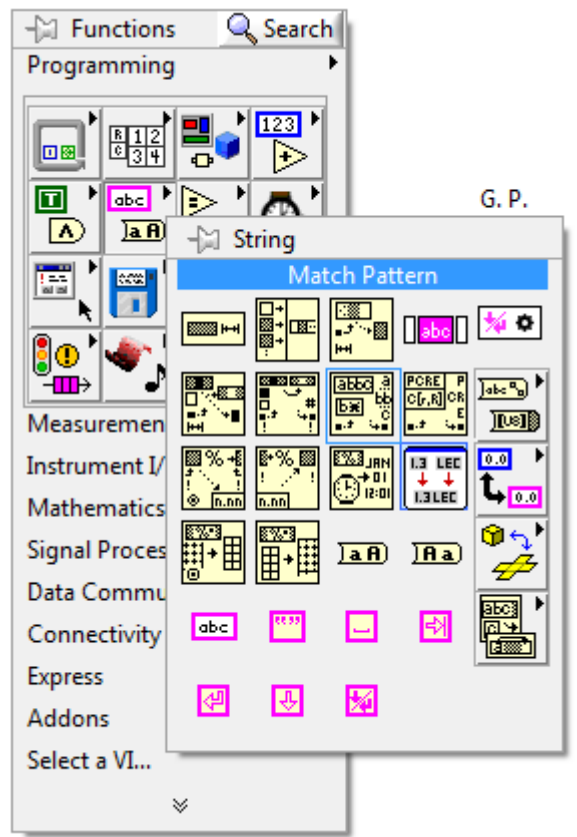
- ✓ A la cadena de datos leídos se convierte en string para poder hacer uso de los identificadores que se establecieron en la programación en Arduino. Se utiliza “Concatenate Strings” que se puede encontrar en Programming – String – Concatenate Strings.



*Figura 48. Concatenate Strings*

*Fuente: Geomara Pilco*

- ✓ Para separar cada uno de los datos recibidos se usó “Match Pattern”; que se encuentra en Programming – String – Match Pattern.

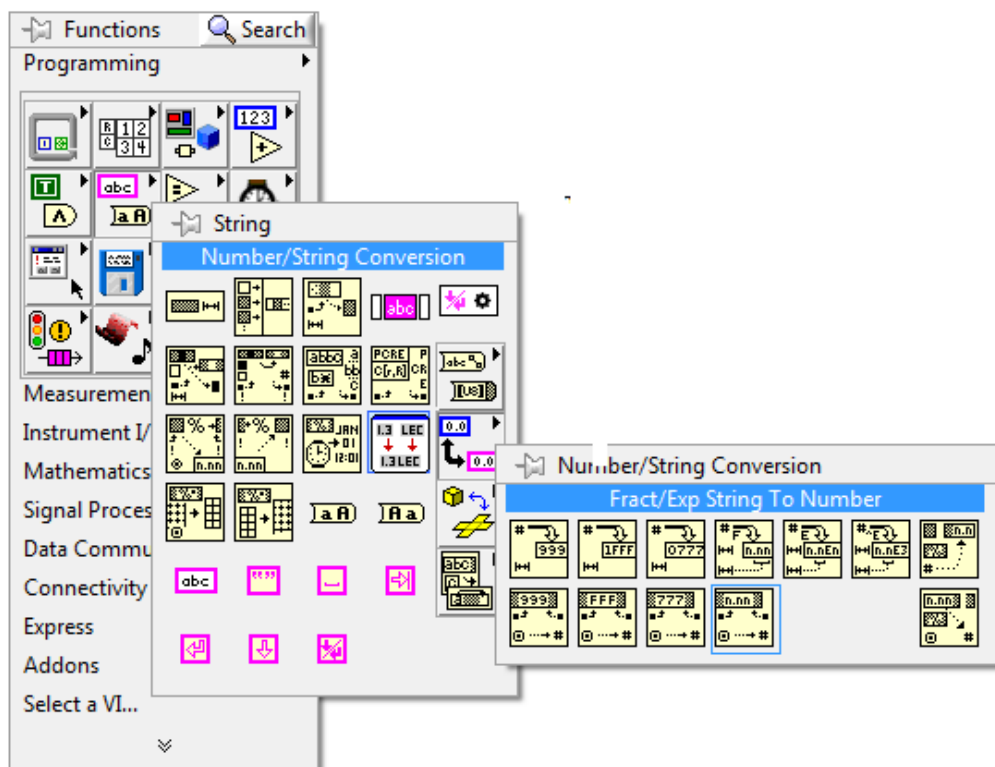


**Figura 49.** Paleta de funciones Match Pattern

**Fuente:** Geomara Pilco

- ✓ Una vez separados los datos el siguiente paso es convertir el dato a número para lo cual usamos Fract/Exp String To Number; que se encuentra en Programming – String – Number//String Conversion - Fract/Exp String To Number.

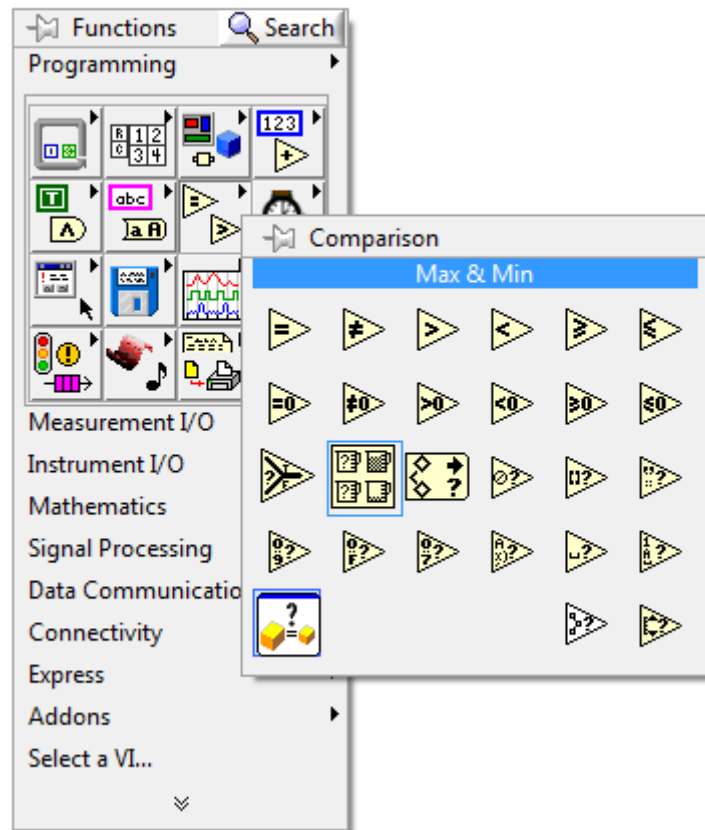




**Figura 50.** Paleta de funciones *Fract/Exp String To Number*

**Fuente:** *Geomara Pilco*

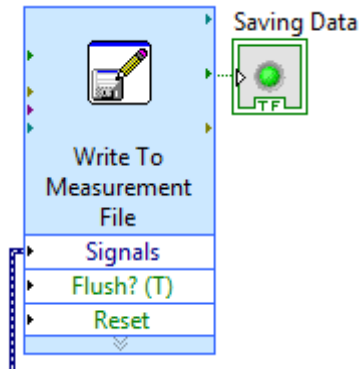
- ✓ Después de convertir el dato número mostramos este valor en gráficos, indicadores y hallamos el valor máximo y mínimo usando Max & Min; esta función se encuentra en Programming – Comparison - Max & Min.



*Figura 51. Paleta de funciones Max & Min*

*Fuente. Geomara Pilco*

- ✓ Finalmente se procede a guardar los datos medidos en un archivo de Excel usando la función Write to Measurement File.



*Figura 52. Función Write to Measurement File*

*Fuente: Geomara Pilco*

## ANEXO 5. HOJAS DE ESPECIFICACIONES TÉCNICAS

### SHT1x / SHT7x Relative Humidity & Temperature Sensor System

#### 1 Sensor Performance Specifications

Parameter	Conditions	Min.	Typ.	Max.	Units
<b>Humidity</b>					
Resolution <sup>(1)</sup>		0.5	0.03	0.03	%RH
		8	12	12	bit
Repeatability			±0.1		%RH
Accuracy <sup>(1)</sup>	linearized	see figure 1			
Uncertainty					
Interchangeability		Fully interchangeable			
Nonlinearity	raw data		±3		%RH
	linearized		<<1		%RH
Range		0		100	%RH
Response time	1/e (63%) slightly moving air		4		s
Hysteresis			±1		%RH
Long term stability	typical		< 0.5		%RH/yr
<b>Temperature</b>					
Resolution <sup>(2)</sup>		0.04	0.01	0.01	°C
		0.07	0.02	0.02	°F
		12	14	14	bit
Repeatability			±0.1		°C
			±0.2		°F
Accuracy		see figure 1			
Range		-40		123.8	°C
		-40		254.9	°F
Response Time	1/e (63%)		5	30	s

Table 1 Sensor Performance Specifications

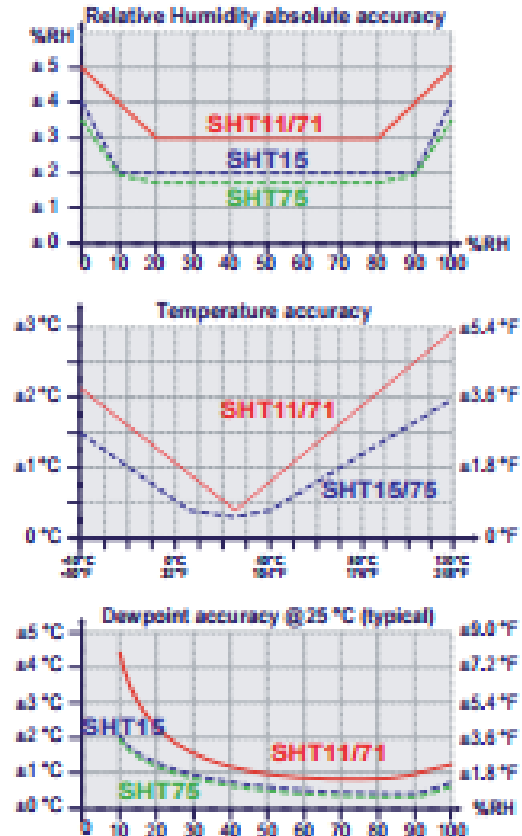


Figure 1 Rel. Humidity, Temperature and Dewpoint accuracies

#### 2 Interface Specifications

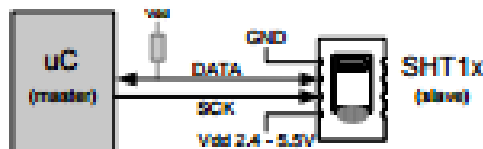


Figure 2 Typical application circuit

##### 2.1 Power Pins

The SHTx requires a voltage supply between 2.4 and 5.5 V. After powerup the device needs 11ms to reach its "sleep" state. No commands should be sent before that time. Power supply pins (VDD, GND) may be decoupled with a 100 nF capacitor.

##### 2.2 Serial Interface (Bidirectional 2-wire)

The serial interface of the SHTx is optimized for sensor readout and power consumption and is not compatible with PC interfaces, see FAQ for details.

<sup>(1)</sup> Each SHTx is tested to be fully within RH accuracy specifications at 25 °C (77 °F) and 48 °C (118.4 °F)

<sup>(2)</sup> The default measurement resolution of 14bit (temperature) and 12bit (humidity) can be reduced to 12 and 8 bit through the status register.

##### 2.2.1 Serial clock input (SCK)

The SCK is used to synchronize the communication between a microcontroller and the SHTx. Since the interface consists of fully static logic there is no minimum SCK frequency.

##### 2.2.2 Serial data (DATA)

The DATA tristate pin is used to transfer data in and out of the device. DATA changes after the falling edge and is valid on the rising edge of the serial clock SCK. During transmission the DATA line must remain stable while SCK is high. To avoid signal contention the microcontroller should only drive DATA low. An external pull-up resistor (e.g. 10 kΩ) is required to pull the signal high. (See Figure 2) Pull-up resistors are often included in I/O circuits of microcontrollers. See Table 5 for detailed I/O characteristics.

# BMP085

## Digital, barometric pressure sensor

Bosch Sensortec



**BOSCH**  
Invented for life

### General description

The BMP085 is a high-precision, ultra-low power barometric pressure sensor for use in advanced mobile applications.

With an absolute accuracy of 2.5hPa and a noise level of down to 0.03hPa (which is equivalent to an altitude change of merely 0.25m) the BMP085 offers superior performance. At the same time the BMP085 features ultra low power consumption of down to 3µA. This and the very small, ultra-thin package make the BMP085 the sensor of choice for any mobile application requiring precise barometric pressure measurement, like for example mobile phones, PDAs, personal GPS-based navigation devices and advanced outdoor equipment.

The BMP085 sensor is based on piezo-resistive MEMS technology for EMC robustness, high accuracy and linearity as well as long term stability. It comes in an ultra-thin, but robust 8-pin ceramic lead-less chip carrier (LCC) package. The BMP085 is designed to be connected directly to a micro-controller of a mobile device via the I<sup>2</sup>C bus.

The BMP085 is the direct successor of the successful SMD500 pressure sensor, that marked a new generation of high-precision digital pressure sensors for consumer applications.

### Key features BMP085

- ▶ Wide barometric pressure range
- ▶ Flexible supply voltage range
- ▶ Ultra-low power consumption
- ▶ Low noise measurement
- ▶ Fully calibrated
- ▶ Temperature measurement included
- ▶ Digital two-wire I<sup>2</sup>C interface
- ▶ Ultra-flat, small footprint ceramic package
- ▶ Pb-free and RoHS compliant

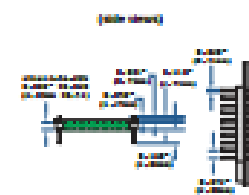
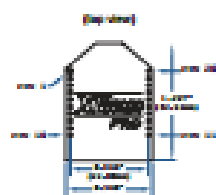
### BMP085 target applications

- ▶ GPS navigation enhancement
- ▶ Dead reckoning
- ▶ In- and outdoor navigation
- ▶ Leisure, sports and health monitoring
- ▶ Weather forecast
- ▶ Vertical velocity indication (rise/sink speed)
- ▶ Fan power control

Technical data	BMP085
Pressure range	300 ... 1100 hPa
RMS noise expressed in pressure	0.06 hPa, typ. (ultra-low power mode)
	0.03 hPa, typ. (ultra-high resolution mode)
RMS noise expressed in altitude	0.5 m, typ. (ultra-low power mode)
	0.25 m, typ. (ultra-high resolution mode)
Absolute accuracy p=700 ... 1100hPa (T=0 ... +65°C, V <sub>DDA</sub> =3.3V)	Pressure: ± 2.5 hPa, max. Temperature: ±2 °C, max.
Average current consumption (1Hz data refresh rate)	3 µA, typ. (ultra-low power mode)
	12 µA, typ. (ultra-high resolution mode)
Peak current	600 µA, typ.
Stand-by current	0.1 µA, typ.
Supply voltage V <sub>DD0</sub>	1.62 ... 3.6 V
Supply voltage V <sub>DDA</sub>	1.8 ... 3.6 V
Operation temp. range	-40 ... +85 °C
full accuracy	0 ... +65 °C
Pressure conv. time	7.5 msec, max.
I <sup>2</sup> C data transfer rate	3.4 MHz, max.
Package type / pin no.	LCC / 8
Package dimensions	5 mm x 5 mm x 1.2 mm

Hoja de especificaciones técnicas Datasheet del sensor BMO085.

Platform		XBee-PRO® 868
<b>Performance</b>		
RF Data Rate	24 kbps (limited to 10% duty cycle)	
Indoor/Urban Range	Up to 1000 ft (300 m)	
Outdoor/RF Line-of-Sight Range	Up to 24 miles (39 km) with dipole antenna (24% duty) up to 10 miles (16 km) with dipole antenna (12.7 dBm)	
Transmit Power	1 mW (23 dBm) to 100 mW (+20 dBm)	
Receiver Sensitivity (1% BER)	-102 dBm or 100 nW ERP	
<b>Features</b>		
Serial Data Interface	3.3V CMOS Serial (UART)	
Configuration Method	API and AT commands	
Frequency Band	868 MHz ISM	
Interference Immunity	Multiple transmissions, acknowledgments	
Serial Data Rate	1.2 kbps to 230.4 kbps (non-standard rates available)	
ADC Inputs	0 (10-bit)	
Digital I/O	10	
Antenna Options	Wired whip, UFL connector, SMA connector	
<b>Networking &amp; Security</b>		
Encryption	128-bit AES	
Reliable Packet Delivery	Retransmissions/acknowledgments	
Addressing Options	Network ID, 64-bit address	
Channels	Single channel	
<b>Power Requirements</b>		
Supply Voltage	3.0 - 3.6VDC	
Transmit Current	100 mA typical at 3.3V (300 mA max)	
Receiver Current	50 mA typical	
Power-Down Current	50 uA typical @3.3V	
<b>Regulatory Approvals</b>		
FCC (USA)	Yes	
IC (Canada)	Yes	
CE (Europe)	Yes (24% duty max)	
C-TICK (Australia)	Yes	
VCCI (Japan)	Yes	



You can purchase with confidence knowing that Digi is always available to serve you with expert technical support and our industry leading warranty. For detailed information visit [www.digi.com/support](http://www.digi.com/support)

910001090  
03/10/11

Digi International  
Worldwide HQ  
877-912-1888  
952-912-1888  
[www.digi.com](http://www.digi.com)

Digi International  
France  
+33-1-69-41-89-88  
[www.digi.fr](http://www.digi.fr)

Digi International  
Japan  
+81-3-6428-0061  
[www.digi-int.co.jp](http://www.digi-int.co.jp)

Digi International  
Singapore  
+65-6713-5388

Digi International  
China  
+86-21-50901199  
[www.digi.com.cn](http://www.digi.com.cn)



[www.digi.com](http://www.digi.com)

© 1996-2011 Digi International Inc. All rights reserved. All other trademarks are the property of their respective owners.

Hoja de especificaciones técnicas del módulo Xbee Pro.

## ANEXO 6. PROGRAMACIÓN EN ARDUINO

```
//HUMEDAD
int SHT_clockPin = 3; // pin usado para clock
int SHT_dataPin = 2; // pin usado para data

//PRESION
#include <util/delay.h> //Needed by BMP085 library
#include "Adafruit_BMP085.h" //Library to control the BMP085
Adafruit_BMP085 bmp
    //VIENTO
#define analogPinForRV 1
#define analogPinForTMP 0
const float zeroWindAdjustment = .2;
int TMP_Therm_ADunits;
float RV_Wind_ADunits;
float RV_Wind_Volts;
unsigned long lastMillis;
int TempCtimes100;
float zeroWind_ADunits;
float zeroWind_volts;
float WindSpeed_MPH;

void setup(){
  Serial.begin(9600);
  bmp.begin();
}

void loop(){

  float humidity = getHumidity();
```

```

//VIENTO

TMP_Therm_ADunits = analogRead(analogPinForTMP);
RV_Wind_ADunits = analogRead(analogPinForRV);
RV_Wind_Volts = (RV_Wind_ADunits * 0.0048828125);

TempCtimes100 = (0.005 * ((float)TMP_Therm_ADunits *
(float)TMP_Therm_ADunits)) - (16.862 * (float)TMP_Therm_ADunits) + 9075.4;

zeroWind_ADunits = -0.0006 * ((float)TMP_Therm_ADunits *
(float)TMP_Therm_ADunits) + 1.0727 * (float)TMP_Therm_ADunits + 47.172;

zeroWind_volts = (zeroWind_ADunits * 0.0048828125) - zeroWindAdjustment;

WindSpeed_MPH = pow(((RV_Wind_Volts - zeroWind_volts) /.2300) ,
2.7265);
WindSpeed_MPH = WindSpeed_MPH*0.44704;
}
Serial.print("H");
Serial.print(humidity);

Serial.print("T");
Serial.print(bmp.readTemperature());
//Serial.println(" *C");
Serial.print("P");
Serial.print(bmp.readPressure());
//Serial.println(" Pa");

Serial.print("A");
Serial.print(bmp.readAltitude());
//Serial.println(" Meters");

```



```

Serial.print("V");
if (WindSpeed_MPH>0)
{
Serial.println((float)WindSpeed_MPH);
}
else
{
Serial.println("0");
}
delay(1000);
}
float getTemperature(){
SHT_sendCommand(B00000011, SHT_dataPin, SHT_clockPin);
SHT_waitForResult(SHT_dataPin);
int val = SHT_getData(SHT_dataPin, SHT_clockPin);
SHT_skipCrc(SHT_dataPin, SHT_clockPin);
return (float)val * 0.01 - 40; //convert to celsius
}
float getHumidity(){
SHT_sendCommand(B00000101, SHT_dataPin, SHT_clockPin);
SHT_waitForResult(SHT_dataPin);
int val = SHT_getData(SHT_dataPin, SHT_clockPin);
SHT_skipCrc(SHT_dataPin, SHT_clockPin);
return -4.0 + 0.0405 * val + -0.0000028 * val * val;
}

void SHT_sendCommand(int command, int dataPin, int clockPin){
pinMode(dataPin, OUTPUT);
pinMode(clockPin, OUTPUT);
digitalWrite(dataPin, HIGH);
digitalWrite(clockPin, HIGH);
}

```

```

digitalWrite(dataPin, LOW);
digitalWrite(clockPin, LOW);
digitalWrite(clockPin, HIGH);
digitalWrite(dataPin, HIGH);
digitalWrite(clockPin, LOW);

shiftOut(dataPin, clockPin, MSBFIRST, command);

digitalWrite(clockPin, HIGH);
pinMode(dataPin, INPUT);

if (digitalRead(dataPin)) Serial.println("ACK error 0");
digitalWrite(clockPin, LOW);
if (!digitalRead(dataPin)) Serial.println("ACK error 1");
}

void SHT_waitForResult(int dataPin){
    pinMode(dataPin, INPUT);

    int ack;

    for (int i = 0; i < 1000; ++i){
        delay(2);
        ack = digitalRead(dataPin);
        if (ack == LOW) break;
    }

    if (ack == HIGH) Serial.println("ACK error 2");
}

int SHT_getData(int dataPin, int clockPin){

```

```

pinMode(dataPin, INPUT);
pinMode(clockPin, OUTPUT);
byte MSB = shiftIn(dataPin, clockPin, MSBFIRST);

pinMode(dataPin, OUTPUT);
digitalWrite(dataPin, HIGH);
digitalWrite(dataPin, LOW);
digitalWrite(clockPin, HIGH);
digitalWrite(clockPin, LOW);

pinMode(dataPin, INPUT);
byte LSB = shiftIn(dataPin, clockPin, MSBFIRST);
return ((MSB << 8) | LSB); //combine bits
}

void SHT_skipCrc(int dataPin, int clockPin){

pinMode(dataPin, OUTPUT);
pinMode(clockPin, OUTPUT);
digitalWrite(dataPin, HIGH);
digitalWrite(clockPin, HIGH);
digitalWrite(clockPin, LOW);
}

```