

**IMPLEMENTACIÓN DE MÓDULOS PEDAGÓGICOS PARA LA ENSEÑANZA DE  
ELECTRÓNICA Y AUTOMATIZACIÓN EN PROGRAMAS DE BASE  
TECNOLÓGICA**

**FRANCISCO JAVIER CÓRDOBA CIFUENTES  
ORLANDO ALBEIRO CORREA SANTAMARÍA  
GUSTAVO ALONSO QUINTANA ARANGO**

**INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO  
FACULTAD DE INGENIERÍA  
TECNOLOGÍA SISTEMAS MECATRONICOS  
MEDELLÍN  
2016**

**IMPLEMENTACIÓN DE MÓDULOS PEDAGÓGICOS PARA LA ENSEÑANZA DE  
ELECTRÓNICA Y AUTOMATIZACIÓN EN PROGRAMAS DE BASE  
TECNOLÓGICA**

**FRANCISCO JAVIER CÓRDOBA CIFUENTES  
ORLANDO ALBEIRO CORREA SANTAMARÍA  
GUSTAVO ALONSO QUINTANA ARANGO**

**TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE TECNÓLOGO MECATRÓNICO**

**ASESORES:**

**CARLOS VALENCIA**

**JUAN GONZÁLO ZULUAGA**

**INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO  
FACULTAD DE INGENIERÍA  
TECNOLOGÍA SISTEMAS MECATRONICOS  
MEDELLÍN  
2016**

## CONTENIDO

<b>INTRODUCCIÓN</b> .....	9
<b>2 JUSTIFICACIÓN</b> .....	11
<b>3 OBJETIVOS</b> .....	13
3.1 OBJETIVO GENERAL .....	13
3.2 OBJETIVOS ESPECÍFICOS .....	13
<b>4 MARCO TEÓRICO</b> .....	14
<b>4.1 HISTORIA</b> .....	14
<b>4.2 ARDUINO – MICROCONTROLADORES</b> .....	15
<b>4.3 LENGUAJE DE PROGRAMACIÓN EN PLACA ARDUINO</b> .....	20
<b>4.4 REFERENCIAS DEL LENGUAJE ARDUINO</b> .....	21
4.4.1 Estructura .....	21
4.4.2 Estructura De Control .....	22
4.4.3 Estructura De Sintaxis.....	22
4.4.4 Operador Comparativo.....	23
4.4.5 Operador De Composición.....	23
4.4.6 Variables .....	23
4.4.7 Funciones .....	24
<b>4.5 MODULOS DE COMUNICACIONES ARDUINO</b> .....	26
4.5.1 Módulo XBee .....	26
4.5.2 Módulo Bluetooth .....	29
4.5.3 Módulo Ethernet Shield .....	31
<b>4.6 MODULOS DE INSTRUMENTACION ARDUINOS</b> .....	33
4.6.1 Sensor Ultrasonido .....	33
4.6.2 Sensor De Distancia X Infrarrojo.....	35

4.6.3	Sensor De Presión .....	37
4.6.4	Sensor Termopar .....	41
<b>5</b>	<b>METODOLOGÍA.....</b>	<b>42</b>
5.1	TIPO DE PROYECTO.....	42
5.2	MÉTODO.....	42
5.3	POBLACIÓN .....	43
<b>6</b>	<b>RESULTADOS.....</b>	<b>44</b>
<b>6.1</b>	<b>DESCRIPCION DEL DESARROLLO DEL PROYECTO.....</b>	<b>44</b>
<b>6.2</b>	<b>POR QUE ARDUINO?.....</b>	<b>46</b>
<b>6.3</b>	<b>INTRUMENTACION A UTILIZAR EN EL ENSAMBLE MODULOS DIDACTICOS .....</b>	<b>48</b>
6.3.1	Sensor Ultrasónico HC-SR04 Arduino .....	48
6.3.2	Sensor De Presión FSR 402 .....	49
6.3.4	Sensor De Distancia X Infrarrojo GP2Y0A21YK.....	51
<b>6.4</b>	<b>MODULOS DE COMUNICACIONES UTILIZADOS EN ENSAMBLE DE LOS MODULOS DIDACTICOS.....</b>	<b>52</b>
6.4.1	Módulo XBee Pro .....	52
6.4.2	Modulo Ethernet Shield W5100 Arduino.....	52
6.4.3	Modulo Bluetooth HC06 Para Arduino.....	53
<b>6.5</b>	<b>TARJETA MICROCONTROLADOR MEGA AT2560 A UTILIZAR EN EL ENSAMBLE DE LOS MODULOS DIDACTICOS .....</b>	<b>54</b>
<b>6.6</b>	<b>REGISTRO FOTOGRAFICO DEL ENSAMBLE MODULOS DIDACTICOS .....</b>	<b>55</b>
6.6.1	Módulo De Potencia.....	55
6.6.2	Módulo De XBee .....	56
6.6.3	Módulo Ethernet Shield W5100.....	57

6.6.4	Módulo Bluetooth HC06.....	58
	<i>Figura 26:</i> Modulo Didáctico Bluetooth HC06. ....	58
6.6.5	Planos Eléctricos De los Modulos Didácticos .....	59
<b>6.7</b>	<b>REGISTRO DE PRUEBAS DE FUNCIONAMIENTO ALGORITMOS...</b>	<b>61</b>
<b>6.8</b>	<b>REGISTRO FOTOGRAFICO DE PRUEBAS DE FUNCIONAMIENTO.</b>	<b>67</b>
<b>6.9</b>	<b>MATERIALES Y COSTOS.....</b>	<b>70</b>
<b>7</b>	<b>CONCLUSIONES.....</b>	<b>73</b>
	<b>BIBLIOGRAFÍA.....</b>	<b>74</b>

## LISTA DE FIGURAS

Figura 1: Arduino Mega 2560.....	15
Figura 2: Software de Programación de Arduino.....	17
Figura 3: Esquema De Conexión Modulo XBee.....	24
Figura 4: Modulos XBee .....	25
Figura 5: Nodos De Red.....	27
Figura 6: Modulos Bluetooth.....	28
Figura 7: Módulo Ethernet Shield Arduino.....	31

Figura 8: Sensor Ultrasónico	
Arduino.....	32
Figura 9: Sensor Ultrasónico	
Arduino.....	33
Figura 10: Sensor De Distancia	
Infrarrojo.....	34
Figura 11: Curva Característica De Trabajo Sensor Infrarrojo	
Arduino.....	35
Figura 12: Sensor De	
Presión.....	37
Figura 13: Curva característica sensor de	
presión.....	38
Figura 14: Sensor Termopar,	
Arduino.....	39
Figura 15: Sensor Ultrasónico HC-SR04	
Arduino.....	46
Figura 16: Sensor De Presión FSR	
402.....	47
Figura 17: Sensor Termopar Tipo	
K.....	48
Figura 18: Sensor De Distancia X Infrarrojo	
GP2Y0A21YK.....	49
Figura 19: Módulo XBee	
PRO.....	50
Figura 20: Módulo Ethernet Shield W5100	
Arduino.....	51
Figura 21: Módulo Bluetooth Hc06 Para Arduino	
.....	51
Figura 22: Módulo AT MEGA 2560 Arduino.	
.....	52

Figura 23: Módulo Didáctico de Potencia.	53
.....	53
Figura 24: Modulo Didáctico	
XBee.....	54
Figura 25: Modulo Didáctico Ethernet Shield W	
5100.....	55
Figura 26: Modulo Didáctico	
Bluetooth.....	56
Figura 27: Planos	
Eléctricos.....	57

**LISTA DE TABLAS**

Tabla 1. Especificaciones Técnicas Arduino Mega 2560.....	16
Tabla 2. Costo Recursos materiales.....	65
Tabla 3. Costos varios.....	67



## INTRODUCCIÓN

Los retos que impone la época actual exige a las empresas, organizaciones y demás entes estar a la vanguardia en conocimientos y desarrollos tecnológicos, de forma que les permita potenciar sus procesos con los más altos estándares de calidad, eficiencia y eficacia, para cumplir no sólo las metas propuestas, sino también tener un desarrollo sostenible y amigable con su entorno social y ambiental. La Institución Universitaria Pascual Bravo, no es ajena a esta necesidad de constantes actualizaciones y desarrollos tecnológicos, que le permitan y faciliten específicamente, implementar una formación teórico-práctica a los alumnos que forman parte de su programa de regionalización.

Por lo anterior, y basados en que los circuitos electrónicos nos ofrecen múltiples funciones para procesar la información, en este proyecto se buscará desarrollar mediante un Arduino, módulos didácticos de fácil manipulación, con diversas aplicaciones y que puedan ser desplazados a los diferentes centros educativos en municipios en donde la Institución Universitaria Pascual Bravo, llega con su programa de regionalización.

Los módulos didácticos facilitan el proceso de enseñanza tanto a los docentes como a los estudiantes. En muchos casos se eliminarían los desplazamientos de los estudiantes hasta la sede central para la realización de prácticas en los laboratorios, optimizando tiempo y recursos, además de facilitar a los estudiantes acceder al conocimiento y la manipulación de estos sistemas propios del ámbito académico.

Adicionalmente el Arduino, que es el hardware a utilizar en este proyecto, además de ser un microcontrolador libre, se ha convertido en una herramienta de gran utilidad para la tecnología por su flexibilidad de configuración, su sencillez, robustez y bajo costo lo que le permite ser utilizado en gran cantidad de aplicaciones y procesos como teléfonos móviles, mandos a distancia, consolas portátiles, cámaras, sistemas de calefacción, dispositivos para discapacitados, en el campo del arte, en juegos de luces, controles para domótica y, en tiempos recientes, en automatización y control industrial.

## 2 JUSTIFICACIÓN

La finalidad de realizar estos módulos, es implementar sistemas integrados que sirvan como herramienta fundamental para desempeñar de una forma más óptima las labores del docente en el proceso de enseñanza en lugares apartados en donde se tienen programas de regionalización, partiendo de la experiencia real y la capacitación técnica recibida, y buscando lograr una optimización de los recursos.

La importancia radica en la optimización de los recursos, para transmitir un conocimiento que permita la prestación de un servicio de óptima calidad con la eficiencia y eficacia que el proceso amerita, con la utilización de los módulos didácticos.

Dichos conocimientos se materializan cuando los estudiantes obtienen mayor confianza y conocimiento de sistemas embebidos, permitiéndoles un mejor desempeño con altos estándares de calidad, gracias a las competencias adquiridas mediante la utilización de estos módulos. Los beneficios para la institución se verán reflejados en una mayor facilidad, agilidad y asertividad en las prácticas.

El impacto inicial será el económico, evitando los costos que implica para muchos estudiantes su estadía en Medellín, ahorro de tiempo para los docentes, permitiéndoles que programen actividades de laboratorio en cada región y por último, para la institución porque libera espacio para la programación de laboratorios en la sede central.

Con estos módulos didácticos, se espera subsanar la falencia de recursos pedagógicos en la enseñanza de electrónica y automatización, basado en el sistema Arduino, para los programas de regionalización con que cuenta la institución.

### **3 OBJETIVOS**

#### **3.1 OBJETIVO GENERAL**

Implementar módulos pedagógicos para la enseñanza de la automatización en programas de base tecnológica en los programas de regionalización de la Institución universitaria Pascual Bravo.

#### **3.2 OBJETIVOS ESPECÍFICOS**

- Implementar los módulos didácticos, prácticos y pedagógicos con aplicaciones diferentes, basados en Módulos Arduino.
- Realizar pruebas de funcionamiento.
- Realizar informe sobre el funcionamiento de los equipos Arduino.

## 4 MARCO TEÓRICO

### 4.1 HISTORIA

Con el transcurrir de los años, el carácter de la Institución Universitaria Pascual Bravo se fortalece, aún más, en los procesos de educación superior que se adelantan en la sede de Medellín y en más de 30 municipios de Antioquia y Colombia, en donde actualmente tiene presencia la Institución.

La Institución Universitaria, se ha consolidado como una entidad líder en programas técnicos, tecnológicos y ahora universitarios, ajustados para el sector productivo del país.

Hace un tiempo, la institución incursionó en la modalidad de enseñanza a nivel regional, llegando a lugares apartados del departamento y de Colombia, donde por causas diversas, se hace difícil el desplazamiento de los estudiantes hacia los establecimientos educativos en los que se dictan los diferentes programas descentralizados, además, varias de las asignaturas que están incluidas en los programas, requieren de la asistencia a prácticas de laboratorio en la sede central, causando dificultades en los alumnos y en ocasiones deserción estudiantil por la escasa posibilidad de los desplazamientos hasta la ciudad, situación que crea la necesidad de buscar alternativas que brinden comodidad y economía al grupo estudiantil, a los docentes y a la institución.

En la actualidad, se acepta que se ya no estamos en una era que la enseñanza basada en la transmisión de contenidos, sea la forma en que se desarrollan las capacidades. Para evitar lo

anterior, hay que establecer estrategias mediante las cuales, al estudiante se le haga protagonista principal de su proceso de enseñanza – aprendizaje.

Se procederá a implementar un conjunto de módulos didácticos, para la enseñanza de automatización, basado en el sistema Arduino, cual se caracteriza por su facilidad para instalar y usar, agilizando la comunicación y conectividad, y con los cuales se espera subsanar la falencia de recursos pedagógicos, para los programas de regionalización con que cuenta la institución.

#### **4.2 ARDUINO – MICROCONTROLADORES.**

Desde que se ha ido profundizando el conocimiento en el comportamiento de los electrones a nivel microscópico, el ámbito de la electrónica se ha ido desarrollando y estructurando, de manera que es necesario trabajar con diferentes y muy específicos componentes electrónicos, que corresponden a cualquier elemento que forme parte de un circuito, como por ejemplo un condensador, una resistencia, un diodo o un LED. Si tomáramos diferentes componentes electrónicos, y los conectásemos de una manera adecuada, podríamos construir diferentes dispositivos que pueden resultar más comunes en nuestra vida diaria, como por ejemplo una radio, una fuente de poder o un temporizador.

En la actualidad, el avance tecnológico es tan grande y tan masivo, que la digitalización de las cosas ha llegado a alcanzar a aparatos tan comunes para nosotros, que hasta finales del siglo XX, seguían siendo en su gran mayoría analógicos. Para ello fue necesario agregar a los circuitos un tipo diferente de componente electrónico, llamado microcontrolador, éste es un circuito integrado (también llamado chip o microchip) programable, el cual posee memoria, una unidad

central de procesamiento (CPU) y dispositivos de entrada y salida (I/O) para la comunicación con otros dispositivos.

La función de un microcontrolador es la de almacenar y ejecutar ciertas instrucciones lógicas digitales grabadas en su memoria a través de un grabador manejado por un software (por ejemplo ICProg., software para programar los microcontroladores). Dichas instrucciones dependerán de la tarea específica que deba cumplir el microcontrolador. Se debe tener en cuenta que al poseer una CPU en su interior, la frecuencia de la ejecución de las instrucciones grabadas en el microcontrolador, dependerá de la frecuencia del cristal externo al microcontrolador siempre que no se supere la frecuencia máxima de trabajo del mismo.

Una de las grandes ventajas de los microcontroladores es su bajo costo, su bajo consumo y su ahorro de espacio dentro de un circuito, ya que dentro del encapsulado existe un circuito microscópico creado a través de fotolitografía, el cual es de muy bajo consumo.

Los microcontroladores Arduino incorporan una entrada de alimentación directa, y un programador mediante USB que a su vez funciona como alimentación, además de un lenguaje y software de programación gratuita y de código abierto.

“Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos.” (Arduino.cc, 2011)

Para trabajar con Arduino, es necesario que la o el docente y el estudiantado, tengan la capacidad de manejar un computador a nivel de usuario, y puedan comprender las nociones



básicas de la programación (partiendo de la base de la lógica proposicional), ya que mediante manuales o guías de ejemplo, es posible aplicar sin dificultad las diferentes instrucciones lógicas que requiere un programa.

En la ilustración a continuación se puede apreciar el modelo “Arduino Mega 2560”



*Figura 1.* Arduino Mega 2560.

Fuente: <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>

Las características electrónicas del Módulo Arduino Mega 2560, con el cual trabajaremos, son las siguientes:

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7 – 12V
Input Voltage (limit)	6 – 20V
Digital I/O Pins	54 (of which 15 provide PWM output)

Analog Input Pins	16
DC Current per I/O	20mA
DC Current for 3.3V Pin	50mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Length	101.52 mm
Width	53.3 mm
Weight	37 g

---

Tabla 1. Especificaciones Técnicas Arduino Mega 2560.

Fuente: <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>

La plataforma Arduino ofrece una gran variedad de opciones a la hora de querer trabajar con ella, y es por eso que la educación de la Tecnología Sistemas Mecatrónicas, Automatización, se puede llegar a ver muy favorecida con un dispositivo y software de estas características.

Mediante guías de trabajo y una correcta aplicación por parte de los docentes, el dispositivo Arduino puede aterrizar a las salas de clases instrumentos de laboratorio, tales como una gran gama de sensores, con los cuales realizar diferentes experimentos, gracias a la gran versatilidad que posee. Además es muy importante recalcar, que para desarrollar los diferentes programas con los cuales se trabajará, es necesario aprender Electrónica y Automatización, ya que la gran capacidad que ofrece la plataforma Arduino para poder visualizar ciertos fenómenos físicos que antes eran imperceptibles dentro de una sala de clases, llamará enormemente la atención de los estudiantes, debido a lo innovador que resulta trabajar en el aula utilizando

tecnologías poco manejadas por el público general y que son propias del quehacer científico actual.

El lenguaje de programación utilizado es el lenguaje propio de Arduino, el cual está basado en los lenguajes de programación C y C++. El entorno de desarrollo integrado o también llamado IDE (de la sigla en inglés *integrated development environment*) es una herramienta para desarrollar el software que se escribirá en el microcontrolador. En este caso el IDE para desarrollar los software de manejo de sensores será el programa de mismo nombre de la interfaz y lenguaje de programación (Arduino), el cual está constituido por un editor de texto para escribir el código, un área de mensajes, una consola de texto, una barra de herramientas con botones de acceso rápido y una serie de menús tal como se muestra en la ilustración.

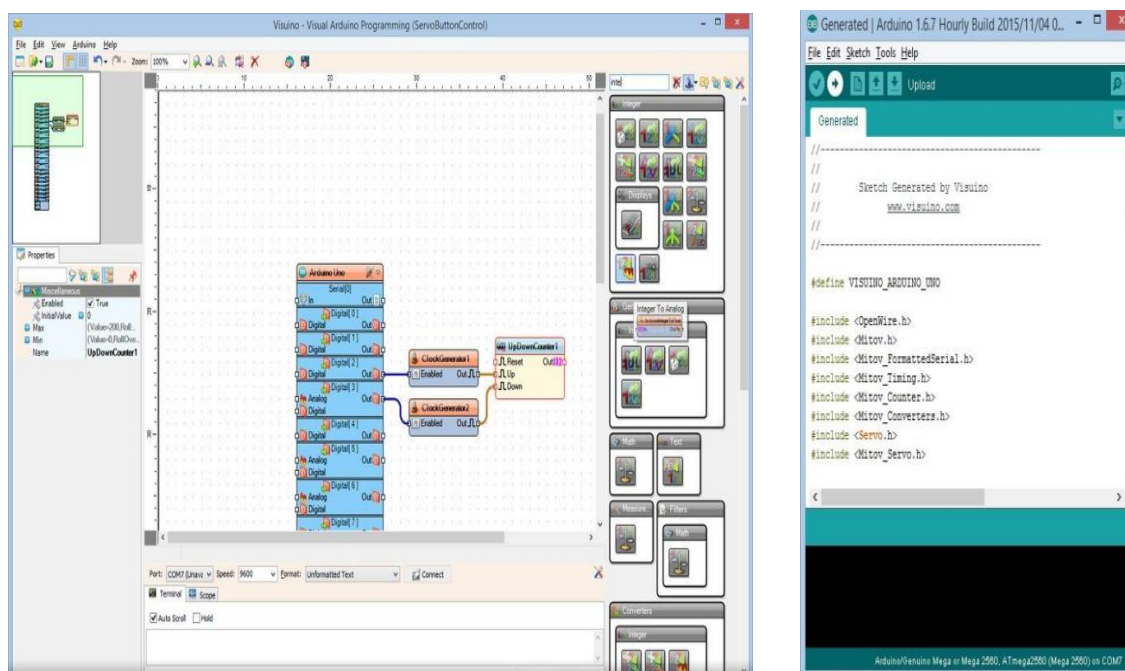


Figura 2. Software de Programación de Arduino.

Fuente: [www.visuino.com](http://www.visuino.com)

### 4.3 LENGUAJE DE PROGRAMACIÓN EN PLACA ARDUINO

La programación de Arduino se desarrolla en torno a 3 partes principales, la estructura, los valores y las funciones, además de esto es posible agregar librerías, esto es un subprograma dentro del software que realiza una función específica para el funcionamiento general, pero que sirve para tener un desarrollo modular dentro de la programación, la librerías pueden ser externas o bien creadas por el usuario.

En resumen, si las variables, las funciones y las librerías poseen una debida estructura (referida a la lógica), se crea un software para que el microcontrolador complementado con hardware apropiado, realice una función específica, en este caso, la lectura de sensores, crear comunicaciones por bluetooth, Xbee, Ethernet.

La estructura básica del lenguaje de programación de Arduino es bastante simple y se compone de al menos dos partes. Estas dos partes necesarias, o funciones, encierran bloques que contienen declaraciones, estamentos o instrucciones.

```
Void setup()
{estamentos;}

Void loop()
{estamentos;}
```

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.

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.

La función bucle (loop) siguiente 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.

#### **4.4 REFERENCIAS DEL LENGUAJE ARDUINO**

Para el uso de los sensores utilizados dentro del presente trabajo, es necesario definir ciertas funciones, variables y la estructura utilizada dentro del código creado para la lectura del sensor. La descripción se dividirá en torno a los tres componentes principales: la estructura, las variables y las funciones.

##### 4.4.1 Estructura

- **“setup()”**: se establece cuando se inicia un programa. Se utiliza para iniciar variables, establecer el estado de los pins, inicializar librerías, etc. Esta función se ejecuta una única vez después de que se conecte la placa o se reinicie la placa.

- **loop():** se utiliza después de la función `setup()`, como dice su nombre, la función `loop()` crea un bucle y se ejecuta consecutivamente, de manera que la placa arduino funciona de manera activa y sin detenerse.

#### 4.4.2 Estructura De Control

- **if:** es la función lógica que se ejecuta cuando se cumple una cierta condición de comparación. Posee la siguiente estructura:  

```
if (condición) {entonces}
```
- **for:** es un bucle contador, el cual ejecuta las funciones que se encuentran dentro de él un determinado número de veces, cada vez que realiza un bucle se utiliza un incremento del contador hasta terminar el bucle y continuar con la función que siga dentro del programa. Su estructura es la siguiente:

```
for (declaración de variable (opcional) valor de inicio de la  
variable; test; incremento o decrecimiento)
```

#### 4.4.3 Estructura De Sintaxis

- “;” (punto y coma): es utilizado dentro del programa para terminar una declaración.
- “{}” (llaves): se utilizan dentro de funciones de estructura, loops y sentencias condicionales con el fin de generar bloques de instrucciones a ejecutar.

- “//” (comentarios): se utilizan para aclarar o comentar alguna instrucción dentro del programa, todo lo que se escriba a la derecha de estas líneas son ignoradas por el compilador (IDE) por lo cual no ocupan espacio en el microcontrolador.

#### 4.4.4 Operador Comparativo

- “==” (igual a): se utiliza cuando se quiere comparar dos variables, de modo que se utiliza junto con las estructuras de control, de modo que el programa comprueba si una variable es idéntica a otra.

#### 4.4.5 Operador De Composición

- “++” (incremento): incrementa una variable en una unidad

#### 4.4.6 Variables

- Constantes

**HIGH/LOW:** definen el nivel o estado de un pin digital cuando este se lee o escribe, el pin estará en HIGH cuando en el pin hay más de 3 voltios y el pin estará en LOW cuando en el pin hay menos de 3 voltios.

**INPUT/OUTPUT:** definen si un pin se utilizará como entrada o salida de información. Si el pin se configura como INPUT, el pin se OUTPUT, éste se encontrará en un estado de baja impedancia.

- Tipos De Datos

**byte:** se declara una variable con un valor desde 0 a 255

**int:** se declara una variable con un valor entero dentro de un rango entre -32768 y 32767

**long:** se declara una variable dentro de un rango entre -2147483648 hasta 2147483647

**void:** se utiliza sólo cuando se quieren declarar funciones, ya sea loop(), setup(), o alguna función creada por el usuario.

**Array:** también llamado matriz, es una colección de variables que son accedidas mediante un número de índice. Los arrays son zero index, lo que referido a una matriz quiere decir que el primer elemento está en índice cero, por lo que si se quiere acceder a al primer elemento dentro de la colección se debe invocar la posición 0.

#### 4.4.7 Funciones

- E/S Digitales

**pinMode():** se utiliza para configurar un pin como INPUT o OUTPUT. Se escribe de la forma: pinMode(pin,valor)



**digitalWrite()**: escribe un valor HIGH o LOW en un pin digital. Se escribe de la forma: digitalWrite(pin, valor)

- E/S Analógicas

**AnalogRead()**: lee el voltaje en un pin analógico específico de la interfaz.

- E/S Avanzadas

**Tone()**: genera una onda cuadrada con una frecuencia especificada, en un pin digital determinado y con una determinada duración. Se escribe de la forma:

tone(pin, frecuencia, duración)

- Tiempo

**Millis ()**: devuelve el tiempo en milisegundos transcurridos desde que arrancó la placa Arduino con el programa actual.

**Delay()**: pausa el programa por un determinado tiempo en milisegundos.

**DelayMicroseconds()**: pausa el programa en un determinado tiempo en microsegundos.

- Matemática

**map()**: cambia un numero desde un rango hacia otro, es decir, un cierto valor que varía dentro de un rango, lo cambia a otro valor proporcional dentro de otro rango determinado. Se escribe de la forma:

map(valor, límite inferior actual, límite superior actual, límite inferior deseado, límite superior deseado)". (Serie: Comunicaciones Modulo, [www.ardumania.es/](http://www.ardumania.es/) )

## 4.5 MODULOS DE COMUNICACIONES ARDUINO

### 4.5.1 Módulo XBee

Los módulos XBee son dispositivos que integran un transmisor – receptor de ZigBee y un procesador en un mismo modulo, lo que le permite desarrollar aplicaciones rápidas y sencillas a los usuarios.

“ZigBee es un protocolo de comunicaciones inalámbrico basado en el estándar de comunicaciones para redes inalámbricas **IEE 802.15.4**. Creado por **ZigBee Alliance**, una organización, teóricamente sin ánimo de lucro”.

(<https://www.arduino.cc/en/Guide/ArduinoXbeeShield>)

ZigBee permite que dispositivos electrónicos de bajo consumo puedan realizar sus comunicaciones inalámbricas. Es especialmente útil para redes de sensores en entornos industriales y sobre todo domótica.

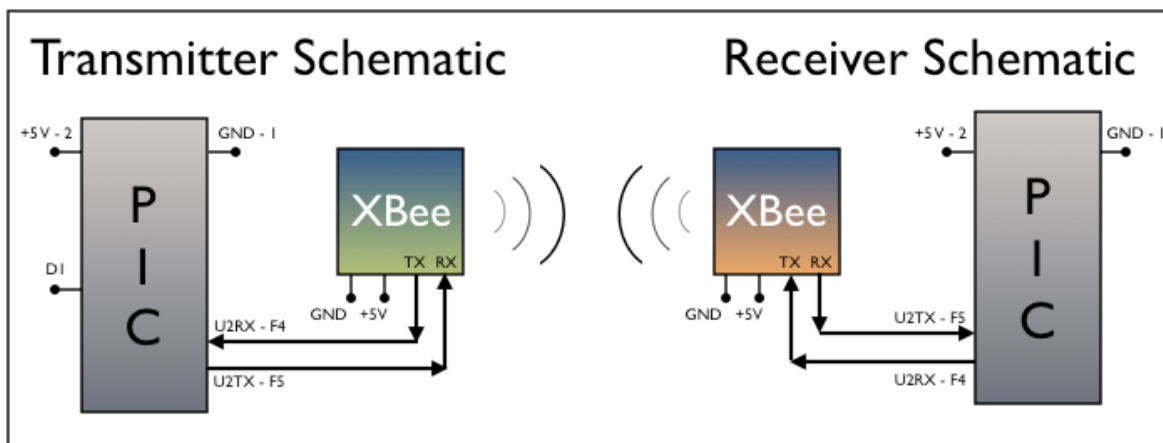


Figura 3: Esquema De Conexión Modulo Xbee

Fuente: [Www.Tuelectronica.Es](http://www.tuelectronica.es)

Las comunicaciones ZigBee se realizan en la banda libre de 2.4 GHz. esto quiere decir que realizan su comunicación en una única frecuencia, en un canal. Tiene 16 canales para escoger. El alcance depende de la potencia del emisor así como el tipo de antenas utilizadas. La velocidad de transmisión de datos es de 256 Kbps. Se puede conformar una red con 65535 equipos, es decir, el protocolo está preparado para poder controlar en la misma red esta cantidad enorme de dispositivos.

Los módulos XBee son económicos, potentes y fáciles de utilizar.

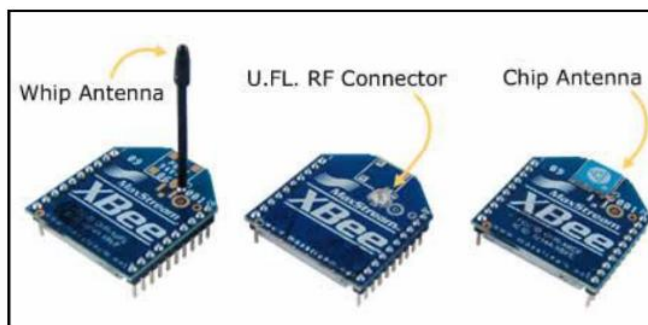


Figura 4: Modulos XBee

Fuente: <https://www.arduino.cc/en/Guide/ArduinoXbeeShield>

#### Arquitectura Básica De Una Red XBee

Una red ZigBee la forman básicamente 3 tipos de elementos. Un único dispositivo Coordinador, dispositivo Routers y dispositivo finales (end points).

- “Coordinador: es el nodo de la red que tiene la única función de formar una red. es el responsable de establecer el canal de comunicaciones y del PAN ID (identificador de red) para toda la red. una vez establecidos estos parámetros el Coordinador puede

formar una red, permitiendo unirse a él a dispositivos Routers y End points. Una vez formada la red, el Coordinador hace las funciones de Router, esto es, participar en el enrutado de paquetes y ser origen y/o destinatario de información”.

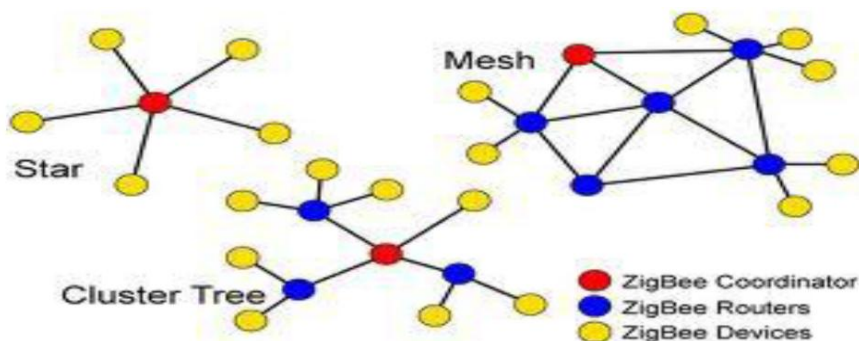
(<http://www.zigbeeoperator.com/>)

- “Los Routers: es un nodo que crea y mantiene información sobre la red para determinar la mejor ruta para enrutar un paquete de información. Lógicamente un router debe unirse a una red ZigBee antes de poder actuar como Router retransmitiendo paquetes de otros routers a de end points”.

(<http://www.zigbeeoperator.com/>)

- “End Device: los dispositivos finales no tienen capacidad de enrutar paquetes. Deben interactuar siempre a través de su nodo padre, ya sea este un Coordinador o un router, es decir, no pueden enviar información directamente a otro end device. Normalmente estos equipos van alimentados a baterías. el consumo es menor al no tener que realizar funciones de enrutamiento”. (<http://www.zigbeeoperator.com/>)

Con los módulos XBee se pueden configurar diferentes topologías de red, como se muestra en la siguiente figura:



*Figura 5: Nodos De Red*

Fuente: <http://www.zigbeeoperator.com/>

Una vez se decida el tipo de red con la que se hará el proyecto, se deben escoger los módulos XBee.

#### Modos De Operación De Módulo Xbee.

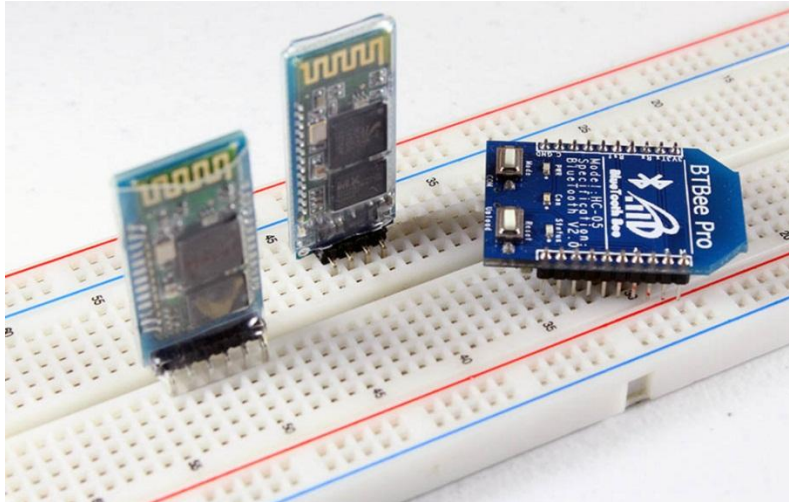
Los módulos pueden ser configurados para trabajar en 3 modos:

- Modo Recibir/Transmitir: al módulo RF le llega un paquete de datos a través de la antena (modo Receive) y luego será transmitida (modo Transmit)
- Modo Bajo Consumo: se hace posible cuando el modulo RF entra en modo de bajo consumo de energía cuando no se encuentra en uso.
- Modo De Comando: este modo permite ingresar comandos al XBee, para configurar, ajustar o modificar parámetros.

#### 4.5.2 Módulo Bluetooth

Los módulos de Bluetooth, se utilizan más comúnmente en el sector de comunicaciones e informática e inicialmente se desarrolló para permitir la comunicación entre ordenadores y teléfonos móviles.

Los Módulo Bluetooth HC-06 y el HC-05, gracias a sus puertos TxD y RxD nos permite realizar comunicaciones inalámbricamente de 10mts máximo, resultan muy fáciles de manejar gracias a su SPP (Serial Port Protocolo) de bajo consumo ya que trabajan a 3.3V.



*Figura 6:* Módulos Bluetooth

Fuente: <http://saber.patagoniatec.com/modulo-bluetooth-hc-06-bluetooth-arduino-slave-hc06-esclavo-iot/>

El módulo de bluetooth HC-06 que solo opera de modo Esclavo, a diferencia de su hermano HC-05, Que ofrece una mejora, con respecto a precio y características, ya que es un módulo Maestro-Esclavo, esto quiere decir que además de recibir conexiones desde una PC o Tablet. También es capaz de generar conexiones hacia otros dispositivos Bluetooth.

#### Conexiones De Los Módulos Bluetooth

- **KEY:** Este pin se puede conectar a cualquier otro pin da la placa Arduino, siempre y cuando este tenga salida digital. El pin KEY se debe activar a nivel alto cuando queremos entrar en el modo de configuración del módulo Bluetooth, una vez que hayamos entrado en este modo podremos cambiar sus parámetros con los comandos AT que luego explicaremos.

- 27 -RXD: Pin por donde se recibirán los datos que nos lleguen desde la placa Arduino. Este pin tiene que ir preciso cableado hasta el pin 1 TX de la placa Arduino que es por donde se transmiten todos los datos que van por el puerto Serie o, en el caso de haber creado un puerto Serie virtual con otros dos pines de la placa, tendría que ir al que hayamos escogido como pin de TX.
- TXD: Pin por donde se transmitirán datos desde el módulo Bluetooth hacia la placa Arduino. Este pin tiene que ir preciso conectado al pin 0 RX de Arduino o, en el caso de haber creado un puerto Serie virtual con otros dos pines de la placa Arduino, al pin que hayamos escogido como pin de RX.
- 5.0: Pin de alimentación a 5 voltios del módulo Bluetooth. Irá conectado al pin 5V de la placa Arduino. 3.3: Pin de alimentación a 3.3 voltios. Irá conectado al pin 3.3V de la placa Arduino. El HC-05 se puede alimentar con cualquiera de estas dos tensiones.
- GND: Pin de masa del módulo HC-05.

#### 4.5.3 Módulo Ethernet Shield

Es una tarjeta electrónica que permite la comunicación entre dos equipos y tiene múltiples usos en la industria de la electrónica y las comunicaciones. Tiene una capacidad de soporte de hasta cuatro conexiones de socket en forma simultánea. El shield Ethernet permite conexión a una placa Arduino con headers wire-wrap largos y que se extienden a través del shield.

Hay una ranura para insertar una tarjeta micro- SD, que puede ser usado para almacenar archivos para servir a través de la red.

El shield actual tiene una alimentación a través de Ethernet, modulo diseñado para extraer energía de un convencional par trenzado categoría 5 cable Ethernet.

El shield dispone de un conector RJ45 para conectarse a una red Ethernet. El botón de reinicio sirve para reiniciar el shield y la propia tarjeta arduino.

El shield contiene un número de LEDs informativos:

- PWR: indica que arduino y el shield están alimentados
- LINK: indica la conexión a una red y parpadea cuando el shield transmite o recibe datos
- FULLD: indica que la conexión de red es full dúplex.
- 100M: indica la presencia de una conexión de red 100 Mbps
- RX: parpadea cuando el shield recibe datos
- TX: parpadea cuando el shield envía datos
- COLL: parpadea cuando se detectan colisiones de red.



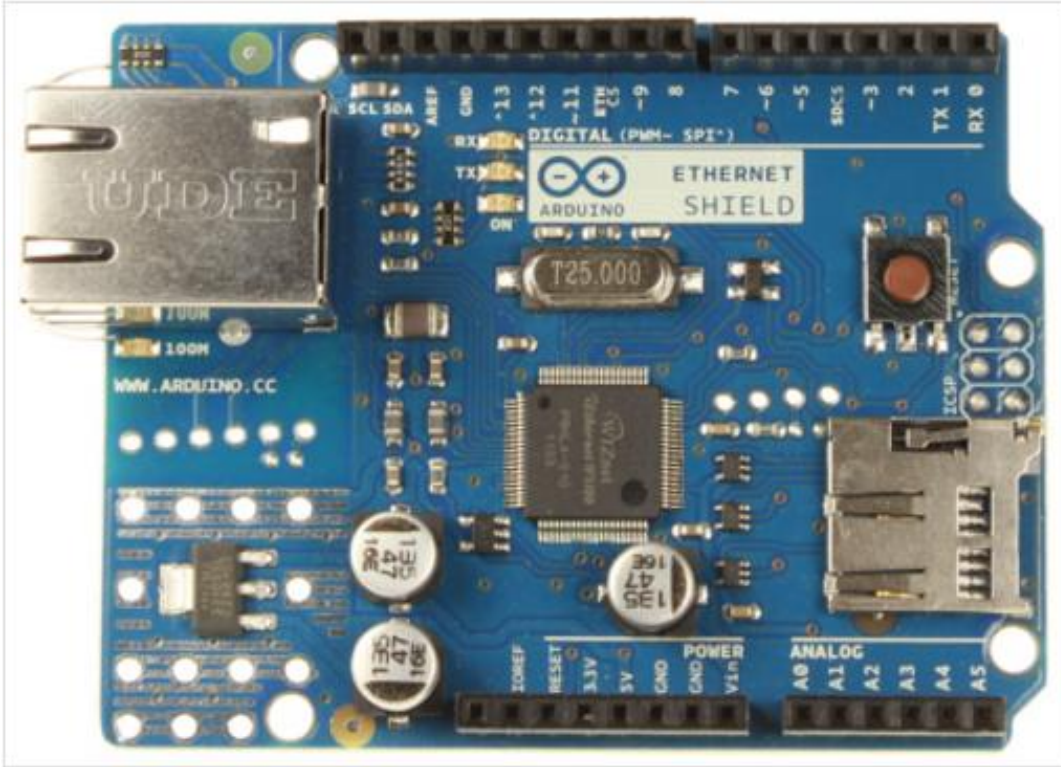


Figura 7: Modulo Ethernet Shield Arduino

Fuente: <https://www.arduino.cc/en/Guide/Arduino>

## 4.6 MODULOS DE INSTRUMENTACION ARDUINOS

### 4.6.1 Sensor Ultrasonido

El sensor de ultrasonidos forma parte del grupo de sensores para medir distancias o superar obstáculos, entre otras posibles funciones. Por eso cuando se produce el sonido éste viaja en el aire a una velocidad casi constante, que permite al sensor captar las ondas de éste cuando rebota en una superficie y es ahí que se puede estimar la distancia recorrida. Este sensor trae toda la circuitería que se requiere para generar el pulso ultrasónico y su correspondiente recepción.

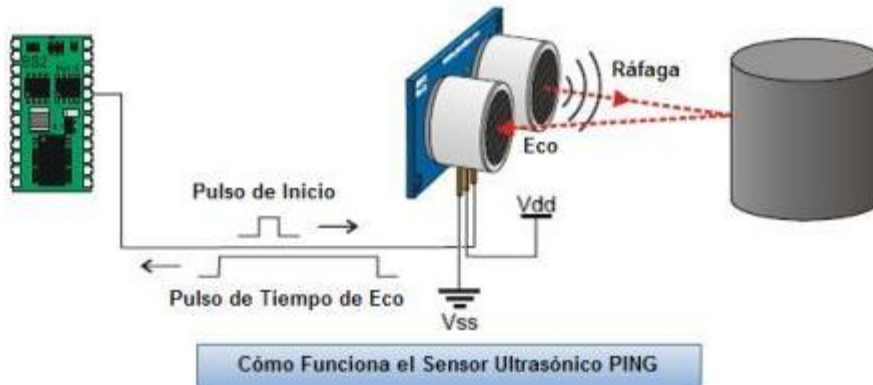


Figura 8: Sensor Ultrasónico Arduino

Fuente: [Www.Tuelectronica.Es](http://www.tuelectronica.es)

El funcionamiento de este módulo es muy sencillo. Está alimentado con 5V y se debe suministrar un pulso de  $10\mu\text{s}$  para activar el módulo a través del pin Trig. En ese momento, el módulo lanzará una ráfaga de 8 pulsos ultrasónicos a 40Khz y la salida Echo pasa a nivel alto hasta que el módulo recibe un eco, momento en el que volverá de nuevo a pasar a un nivel bajo.

“Por tanto, la salida Echo es un pulso cuyo ancho será proporcional a la distancia respecto a un objeto. Si no se detecta un objeto, la salida Echo pasará a nivel bajo después de 30ms.

Si el ancho del pulso se mide en  $\mu\text{s}$ , el resultado se debe dividir entre 58 para saber la distancia en centímetros, y entre 148 para saber la distancia en pulgadas. Estos valores son obtenidos de:

- Si la velocidad del sonido es 340 metros por segundo o  $29\mu\text{s}$  por centímetro, y como el sonido tiene que viajar dos veces la distancia hacia el objeto, una de ida y otra de vuelta, entonces cada  $2 \times 29 = 58\mu\text{s}$  recorrerá un centímetro.

- El módulo debe activarse cada 50ms como mínimo, de esta manera se asegura que la ráfaga ultrasónica haya desaparecido completamente y no provocará un falso eco en la siguiente medición de distancia”. ([Www.Tuelectronica.Es](http://Www.Tuelectronica.Es))

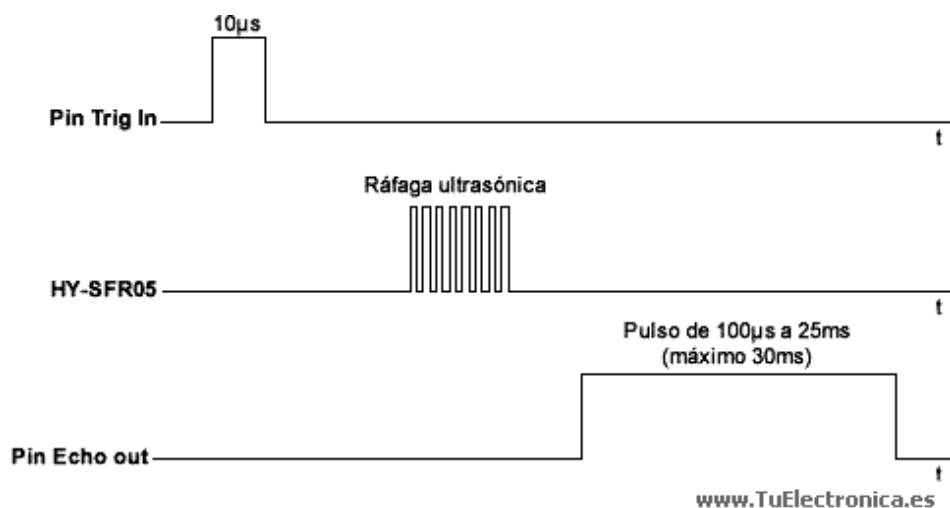
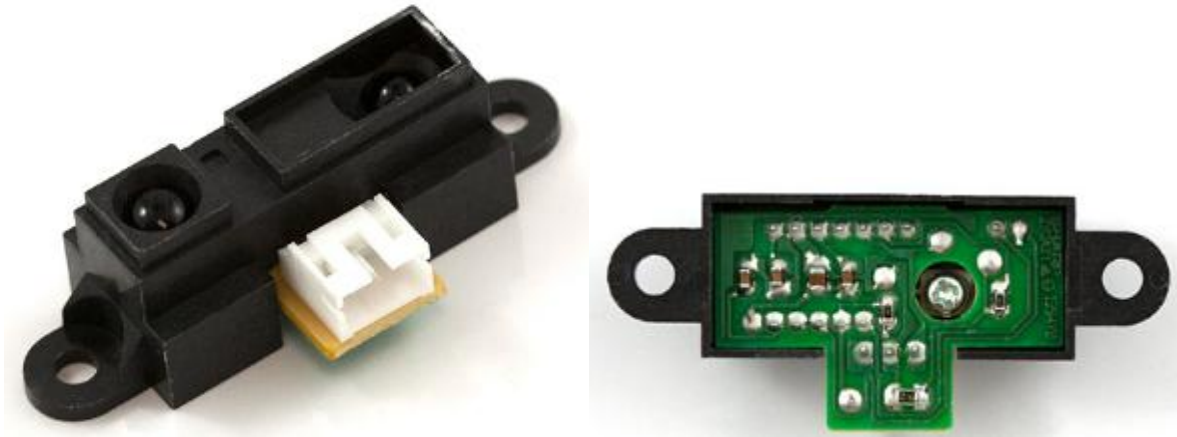


Figura 9: Sensor Ultrasónico Arduino

Fuente: [Www.Tuelectronica.Es](http://Www.Tuelectronica.Es)

#### 4.6.2 Sensor De Distancia X Infrarrojo

Este tipo de sensores permite detectar objetos en condiciones climáticas extremas y está diseñado para clasificar colores y diferencias de las superficies captadas. Su presentación normal es similar a la de un Led normal, con la diferencia que este sensor emite una luz invisible al ojo humano, que sólo es percibida y captada por otros dispositivos electrónicos. Su uso más común está en los controles remotos, en los mandos a distancia de televisión, video y música.



*Figura 10:* Sensor De Distancia Infrarrojo.

Fuente: <http://electronilab.co/>

La interface con la mayoría de los microcontroladores es muy sencilla: una sola línea de salida digital puede ser tomada y leída por el convertidor analógico-digital para tomar la medida de la distancia, o la salida puede ser conectada a un comparador para detección con histéresis. La detección de distancia de esta versión es aproximadamente 10cm a 80cm (4" to 32").

Tiene un conector de 3 pines JST incluido, y además es muy sencillo soldar tres cables para lograr la conexión, mirando de abajo los conectores de izquierda a derecha son: VCC, GND, OUTPUT.

#### Características

- Voltaje de operación: +4.5V a +5.5V
- Consumo de corriente promedio: 30mA (Típico)
- Rango de medición: 10 cm a 80 cm (4" to 32")
- Tipo de salida: Voltaje analógico
- Diferencial de voltaje sobre el rango de distancia: 1.9V Típicos

- Tiempo de respuesta:  $38 \pm 10$  ms
- Tamaño:  $29.5 \times 13.0 \times 13.5$  mm ( $1.16 \times 0.5 \times 0.53$ " )
- Peso: 3.5 g (0.12 oz)

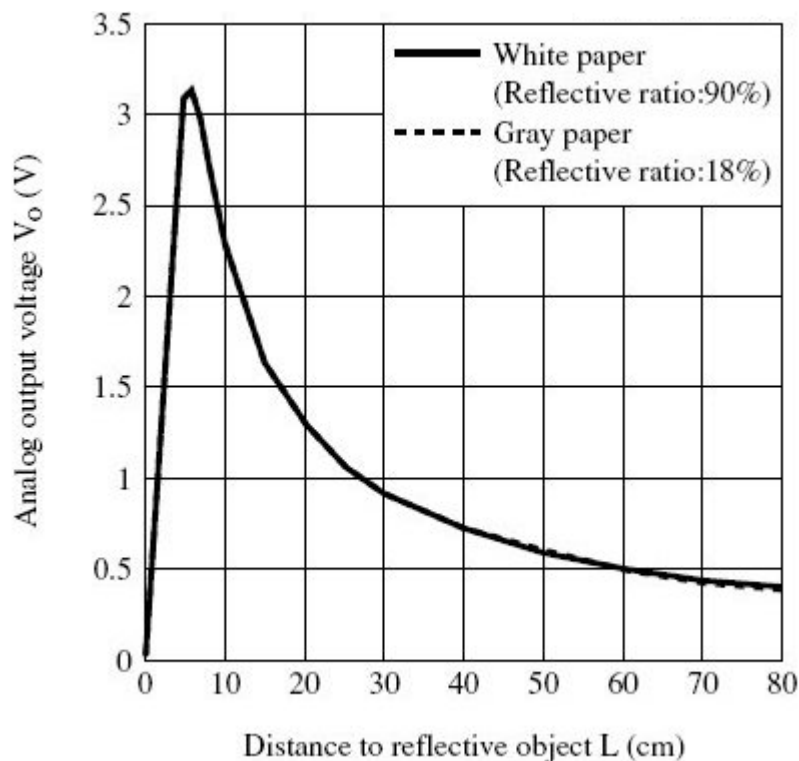


Figura 11: Curva Característica De Trabajo Sensor Infrarrojo Arduino.

Fuente: <http://electronilab.co>

#### 4.6.3 Sensor De Presión

Teniendo en cuenta que un sensor es un dispositivo que está diseñado para recibir información y transformarla de forma que luego se pueda manipular y cuantificar, es necesario precisar las bondades que tiene el sensor de presión, como es que su resistencia varía según el esfuerzo o la presión ejercida y a pesar de que su desventaja es que no es muy preciso.

En sus mediciones tiene amplias aplicaciones como son:

- Medición cualitativa de la fuerza
- Control por toque
- Construcción de interfaces más intuitivas al poder evaluar mayor o menor fuerza del toque y/o toque accidental o intencionado
- Aplicaciones donde se requiere diferenciar entre un toque o un agarre continuo (por ejemplo para mejorar la seguridad de una máquina o herramienta)
- Detección de presencia, movimiento o posición de una persona o paciente en una silla, cama, camilla, o dispositivo médico, etc.
- Detección de bloqueo de fluidos en tubos elásticos detectando el aumento de presión del tubo contra otra superficie
- Controles de juegos, instrumentos musicales, controles remotos, dispositivos médicos, etc.

El sensor de fuerza resistivo (FSR) es un dispositivo de película de polímero (PTF) que presenta una disminución de la resistencia cuando aumenta la fuerza aplicada a la superficie activa.

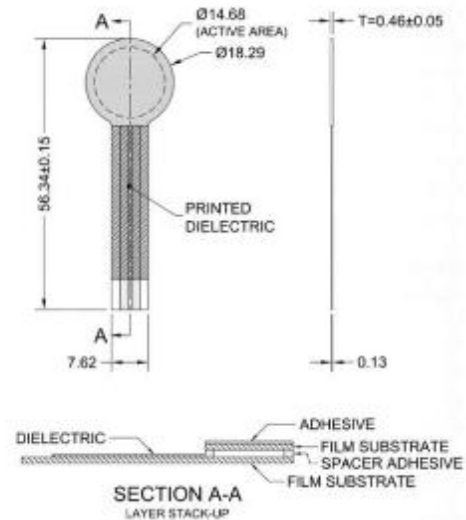


Figura 12: Sensor De Presión

Fuente: <http://www.electronicoscaldas.com/>

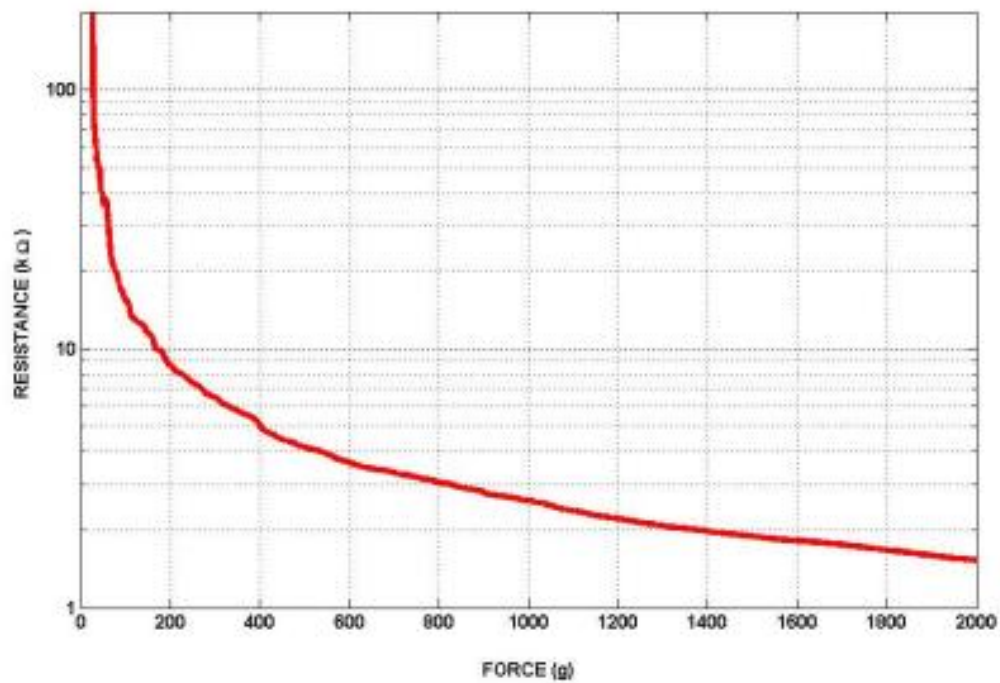
Su sensibilidad a la fuerza está optimizada para uso en el control por toque humano de dispositivos electrónicos. Las FSRs no son células de carga o galgas extensiométricas aunque tengan propiedades similares. Las FSRs no son adecuadas para medidas de precisión. (Carrión y Otros. Programa de Pasantías Académicas, UPS Cuenca, Análisis del funcionamiento del sensor de fuerza resistivo (fsr) con LabView. 2009.

[http://www.datalights.com.ec/site2/images/stories/robotica/nap/nap\\_fsr.pdf](http://www.datalights.com.ec/site2/images/stories/robotica/nap/nap_fsr.pdf).)

#### Características

- Rango de fuerza: 0.2 N a 20 N (20.4 gf a 2.039 kgf)
- Resistencia sin actuación: > 10 MΩ
- Repetibilidad (mismo elemento): ± 2%
- Repetibilidad (entre diferentes elementos): ± 6%
- Rise time: < 3 μs

- Hysteresis: +10% en promedio
- Diámetro total: 18.29 mm
- Diámetro área sensible: 14.68 mm
- Ultradelgado: grosor de 0.46 mm
- Recorrido para activarse: 0.15 mm
- Auto adhesivo
- Vida útil: 10 000 000 de activaciones
- No genera EMI, no es sensible a ESD



*Figura 13:* Curva característica sensor de presión

Fuente: <http://www.electronicoscaldas.com/>



#### 4.6.4 Sensor Termopar

Los termopares se basan en el efecto seebeck. “Al unir dos materiales diferentes aparece una pequeña diferencia de potencial que depende de la naturaleza de los materiales y que es variable con la temperatura”. Sabido esto podemos medir esa pequeña tensión y conocer así la temperatura. Esta tensión es muy débil por lo que hay que amplificarla.

(<http://searchnetworking.techtarget.com/> )

Además al conectar el sensor a la placa aparece otro termopar por el hecho de ser materiales diferentes. Dicho termopar es lo que denominamos unión fría y deberemos compensarla para conocer la temperatura real en el termopar que nos interesa.

Este módulo está listo y emplea un termistor para detectar la temperatura ambiente y con ello compensar la unión fría. Ver figura.



*Figura 14:* Sensor Termopar, Arduino.

Fuente: <http://www.electronicaembajadores.com/>

## 5 METODOLOGÍA

La metodología se levanta sobre la base de una problemática evidenciada por los integrantes del equipo, ya con la información recopilada, se lleva a una mesa de concertación y análisis de manera que dé a lugar la confirmación o el rechazo de las propuestas que se plantean en los objetivos del proyecto. En este análisis se describe y explica el nivel de profundidad que se le da a cada módulo y las respectivas pruebas que se realizarán con ellos, basados en las necesidades planteadas y en la información adquirida en diversas fuentes. Este proceso va acompañado de un material escrito, el cual sirve de soporte y evidencia, para quienes necesiten hacer uso de estos equipos, cabe aclarar que dicho manual se realizará por el mismo grupo estudiantil que realizó este trabajo.

### 5.1 TIPO DE PROYECTO

Este proyecto se toma como práctica en laboratorio, por el interés que surgió de realizar mejoras a los talleres y laboratorios de la institución, proyección del laboratorio y tender al mejoramiento de equipos de laboratorios y talleres en las distintas regiones donde llega la institución Pascual Bravo.

### 5.2 MÉTODO

Para este proyecto se utiliza el método aplicado mediante la técnica de campo, la cual permite por medio de la observación y el contacto directo con la comunidad educativa tanto docentes como alumnos que hacen parte del programa de regionalización del Pascual Bravo, identificar la

necesidad e implementar una solución, mediante la aplicación de los conocimientos teóricos y prácticos en la elaboración de los módulos pedagógicos, para proveer una solución práctica a la problemática identificada.

### 5.3 POBLACIÓN

El alcance de este proyecto, tiende a beneficiar a la institución Universitaria Pascual Bravo y a los docentes, pero, a quien más beneficia es a los estudiantes de asignaturas que requieren de la asistencia a prácticas de laboratorio en la sede central hacia los establecimientos educativos en los que se dictan los diferentes programas descentralizados

## 6 RESULTADOS

### 6.1 DESCRIPCION DEL DESARROLLO DEL PROYECTO

En la ejecución, se realizan las siguientes actividades:

Los módulos se ensamblan en 4 maletas de 37 x 26 cms y 13 cms de profundidad, están diseñadas con recubrimiento de material sintético y espuma, que brindará protección a sus componentes, además, de ser livianas permiten un fácil transporte.

Al interior de estos módulos se dispone de una lámina de acrílico, la cual la cual lleva empotrados una serie de dispositivos con una función específica según la finalidad de cada módulo, esta lámina, está rotulada para la adecuada identificación de los componentes alojados en ella.

Los componentes se instalaron con tornillos espaciadores según el caso, la protoboard queda adherida al acrílico, igualmente se instalaron terminales tipo banana y en la parte posterior del acrílico se ubicaron las terminales que salen de los pulsadores y las conexiones permanentes que hay entre algunos elementos.

Se perforaron las maletas para adicionarle el cable de entrada de potencia, el cableado se realizó en la parte inferior del acrílico, los relevos también se cablean, se instalan soportes en las

maletas para asegurar el acrílico, se adecúa un sistema eléctrico para la alimentación de potencia de cada módulo con salidas de alimentación de 110 VCA, 5 VCD, 12 VCD y tierra.

Son diversos módulos que contienen en su interior fuentes reguladas de voltaje, procesadores Arduino Mega 2560, módulos de comunicaciones y sensores, estos elementos son instalados según la aplicación específica que se le dio a cada módulo.

Luego de finalizado el montaje de los módulos, se procede a cargar los algoritmos propuestos para realizar las pruebas y verificar el buen funcionamiento de dichos equipos, para estas pruebas es necesario realizar primero el proceso de programación.

- Se prueban los módulos con el fin de verificar el correcto funcionamiento.

La mira central de este proyecto estará puesta en la posibilidad de brindar una alternativa para hacer frente a una serie de situaciones que son susceptibles de mejora, mediante unas herramientas que diversifican las oportunidades de aprendizaje de los estudiantes, la optimización del tiempo de los docentes para la preparación de clases, pues disponen de nuevos recursos tecnológicos que apalancan las metodologías usualmente trabajadas para el tratamiento de contenidos.

Son los Módulos Didácticos Digitales, un conjunto de materiales y recursos asociados a un contenido, creados con el objetivo de favorecer el proceso de enseñanza aprendizaje en las

asignaturas que requieren casi con obligatoriedad, el desplazamiento de los estudiantes a la ciudad de Medellín para la asistencia a prácticas de laboratorio en los talleres de la institución.

Será necesario entonces, plantear algunos parámetros que sirvan de ejes a este nuevo cambio, pues, cada módulo está encabezado por una planificación, en la que se especifican los objetivos de los distintos recursos, sirviendo de apoyo al docente para facilitar la integración de estos en las clases.

## **6.2 POR QUE ARDUINO?**

Hay muchos otros microcontroladores y plataformas microcontroladoras disponibles para computación y automatización. Todas estas herramientas toman los desordenados detalles de la programación de microcontrolador y la encierran en un paquete fácil de usar.

Arduino también simplifica el proceso de trabajo con microcontroladores, pero ofrece algunas ventajas para profesores y estudiantes como son las siguientes:

- **Economía:** Las placas Arduino son relativamente económicas comparadas con otras plataformas microcontroladoras. La versión menos costosa del módulo Arduino puede ser ensamblada a mano, e incluso los módulos de Arduino pre ensamblados cuestan menos de \$ 80.000 Col.

- **Multiplataforma:** El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux. La mayoría de los sistemas microcontroladores están limitados a Windows.
- **Entorno De Programación Simple Y Clara:** El entorno de programación de Arduino es fácil de usar para principiantes, pero suficientemente flexible para que usuarios avanzados puedan aprovecharlo también. Para profesores, está convenientemente basado en el entorno de programación Processing, de manera que estudiantes aprendiendo a programar en ese entorno estarán familiarizados con el aspecto y la imagen de Arduino.
- **Código Abierto Y Software Extensible:** El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++, y la gente que quiera entender los detalles técnicos pueden hacer el salto desde Arduino a la programación en lenguaje AVR-C en el cual está basado. De forma similar, puedes añadir código AVR-C directamente en tus programas Arduino si lo requiere.
- **Código Abierto Y Hardware Extensible:** El Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo. Incluso usuarios relativamente inexpertos pueden construir la versión de la placa del módulo para entender cómo funciona y ahorrar dinero.

### **6.3 INSTRUMENTACION A UTILIZAR EN EL ENSAMBLE MODULOS DIDACTICOS**

Para la construcción de los módulos didácticos se utilizaron los siguientes elementos que se describen más adelante, los cuales fueron seleccionados por su fácil forma de operar y su economía.

#### **6.3.1 Sensor Ultrasónico HC-SR04 Arduino**

Específicamente en este caso se utiliza para la medición de distancias, el sensor ultrasónico HC-SR04 Arduino.

El sensor de ultrasonidos se enmarca dentro de los sensores para medir distancias o superar obstáculos, entre otras posibles funciones.

En este caso se utiliza para la medición de distancias. Esto lo consigue enviando una señal de ultrasonido (inaudible para el oído humano por su alta frecuencia) a través de uno de la pareja de cilindros que compone el sensor (un transductor) y espera a que dicho sonido rebote sobre un objeto y regrese, retorno captado por el otro cilindro

Este sensor en concreto tiene un rango de distancias sensible entre 3cm y 3m con una precisión de 3mm.





Figura 15: Sensor Ultrasónico HC-SR04 Arduino

Fuente: <http://www.geekfactory.mx/>

### 6.3.2 Sensor De Presión FSR 402

En este caso se utiliza para la medición de presión, el sensor de presión FSR 402 Arduino.

Teniendo en cuenta que este sensor es un dispositivo que está diseñado para recibir información y transformarla de forma que luego se pueda manipular y cuantificar, es necesario precisar las bondades que tiene el sensor de presión FSR 402 como es que su resistencia varía según el esfuerzo o la presión ejercida.

La forma más fácil de entender cómo funciona un FSR es conectar un multímetro en el modo de medición de resistencia a los dos pines del sensor y ver los cambios en la resistencia.

Como hemos dicho, la resistencia del FSR cambia a medida que se aplica más presión. Cuando no hay presión, el sensor se parece a una resistencia infinita (circuito abierto), ya que la presión aumenta, la resistencia disminuye



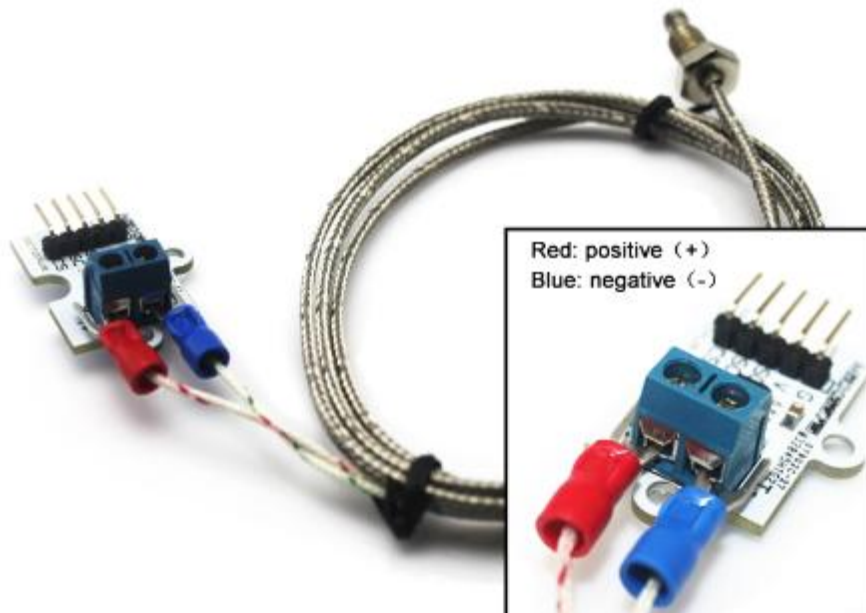
*Figura 16:* Sensor De Presión FSR 402

Fuente: <http://learn.teslabem.com/>

### 6.3.3 Sensor Termopar Tipo K

En este caso se va a utilizar para la medición de temperatura, el sensor de temperatura TERMOPAR TIPO K Arduino.

El detector empleado es un termopar de tipo K y nos permite medir temperaturas de -50 a 600°C con una precisión del  $\pm (2,0\% + 2^\circ\text{C})$ .



*Figura 17:* Sensor Termopar Tipo K

Fuente: <http://learn.teslabem.com/>

### 6.3.4 Sensor De Distancia X Infrarrojo GP2Y0A21YK.

En este caso en particular se utiliza para la medición de distancia, el sensor infrarrojo de proximidad y distancia de medio alcance GP2Y0A21YK.

Este detector funciona con una alimentación de 5VDC, y su rango de medición es de 10 – 80 cm, su señal de salida es de 0 -3.0VDC análoga.

Su descripción de funcionamiento se encuentra en el marco teórico de este trabajo.

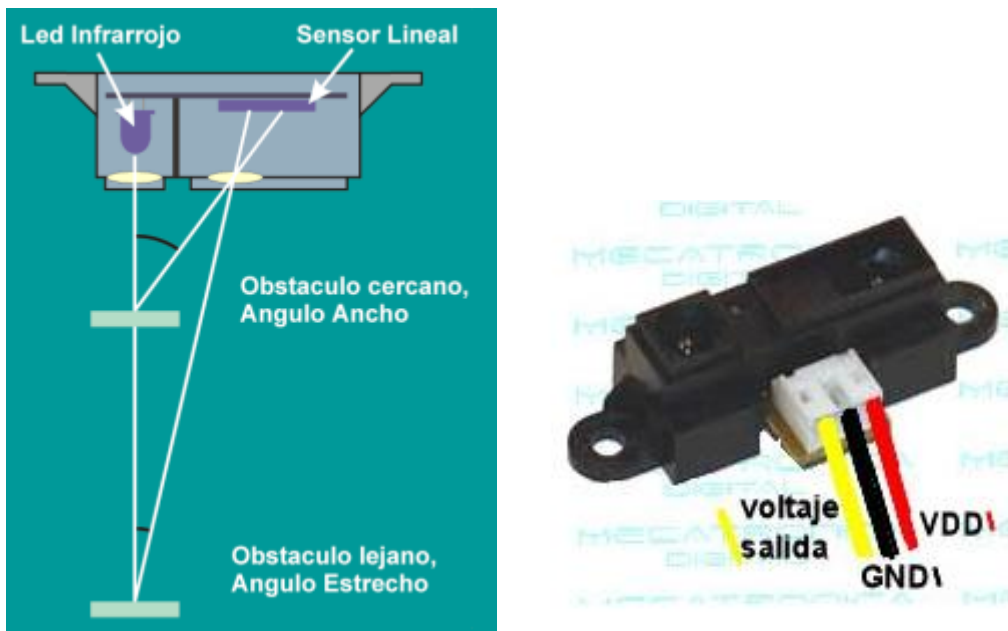


Figura 18: Sensor De Distancia X Infrarrojo GP2Y0A21YK.

Fuente: <http://www.x-robotics.com/>

## 6.4 MODULOS DE COMUNICACIONES UTILIZADOS EN ENSAMBLE DE LOS MODULOS DIDACTICOS

### 6.4.1 Módulo XBee Pro

Se utiliza el Modulo XBee Pro. Ya que se pueden hacer cosas simples, como reemplazar un par de cables en una comunicación serial.

Su descripción de funcionamiento se encuentra en el marco teórico de este trabajo.



*Figura 19: Módulo XBee PRO*

Fuente: <http://www.x-robotics.com/>

### 6.4.2 Modulo Ethernet Shield W5100 Arduino.

Específicamente en este caso se utiliza el Modulo Ethernet Shield W5100 Arduino. Ya que se pueden hacer cosas simples, como configurar una red de comunicación a un costo muy bajo. Su descripción de funcionamiento se encuentra en el marco teórico de este trabajo.



Figura 20: Módulo Ethernet Shield W5100 Arduino.

Fuente: <http://www.x-robotics.com/>

### 6.4.3 Modulo Bluetooth HC06 Para Arduino.

Para este caso se utiliza el Modulo Bluetooth Hc06 Para Arduino. Ya que se pueden hacer cosas simples, como configurar una red de comunicación a un costo muy bajo sin necesidad de cables. Su descripción de funcionamiento se encuentra en el marco teórico de este trabajo.

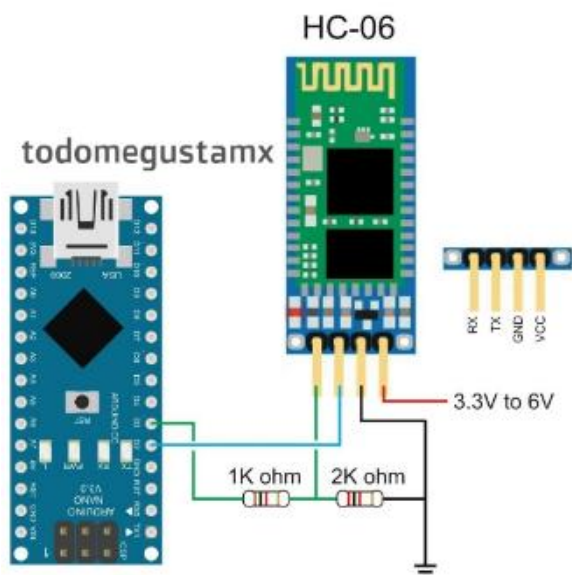


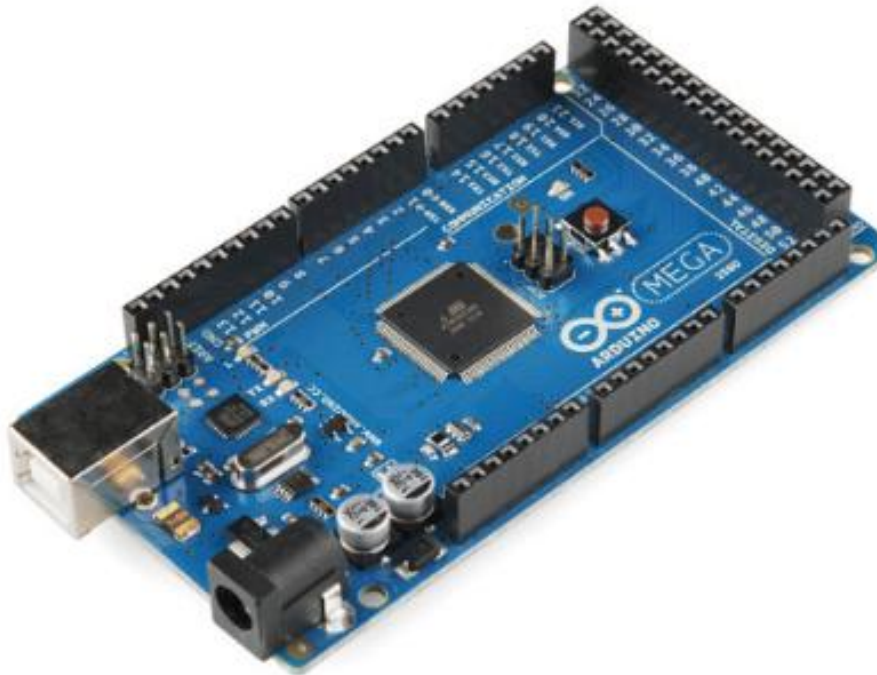
Figura 21: Modulo Bluetooth HC06 Para Arduino

Fuente: <http://www.x-robotics.com/>

## 6.5 TARJETA MICROCONTROLADOR MEGA AT2560 A UTILIZAR EN EL ENSAMBLE DE LOS MODULOS DIDACTICOS

Específicamente en este caso se va a utilizar el Modulo AT MEGA 2560 Arduino. Ya que se pueden hacer cosas simples, como configurar una red de comunicación, crear programas y obtener señales, análogas y/o digitales. Así como tener salidas para controlar procesos, de forma remota o local.

A diferencia de otros dispositivos tiene un costo muy bajo. Su descripción de funcionamiento se encuentra en el marco teórico de este trabajo.



*Figura 22:* Modulo AT MEGA 2560 Arduino.

Fuente: <https://www.arduino.c>

## 6.6 REGISTRO FOTOGRAFICO DEL ENSAMBLE MODULOS DIDACTICOS

### 6.6.1 Módulo De Potencia



*Figura 23:* Módulo Didáctico de Potencia.  
Fuente: Propia Alumnos.

## 6.6.2 Módulo De XBee



*Figura 24:* Modulo Didáctico XBee.  
Fuente: Propia Alumnos.

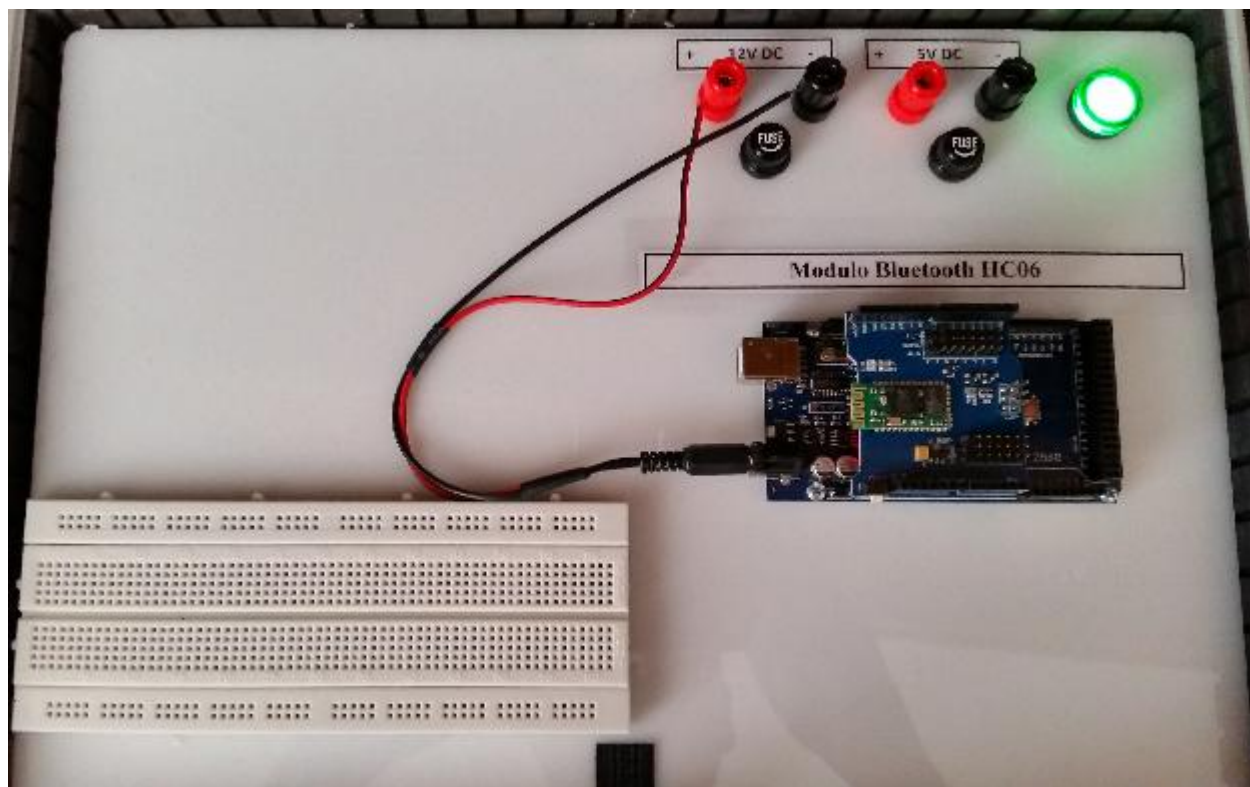


### 6.6.3 Módulo Ethernet Shield W5100



*Figura 25:* Modulo Didáctico Ethernet Shield W5100.  
Fuente: Propia Alumnos.

## 6.6.4 Módulo Bluetooth HC06

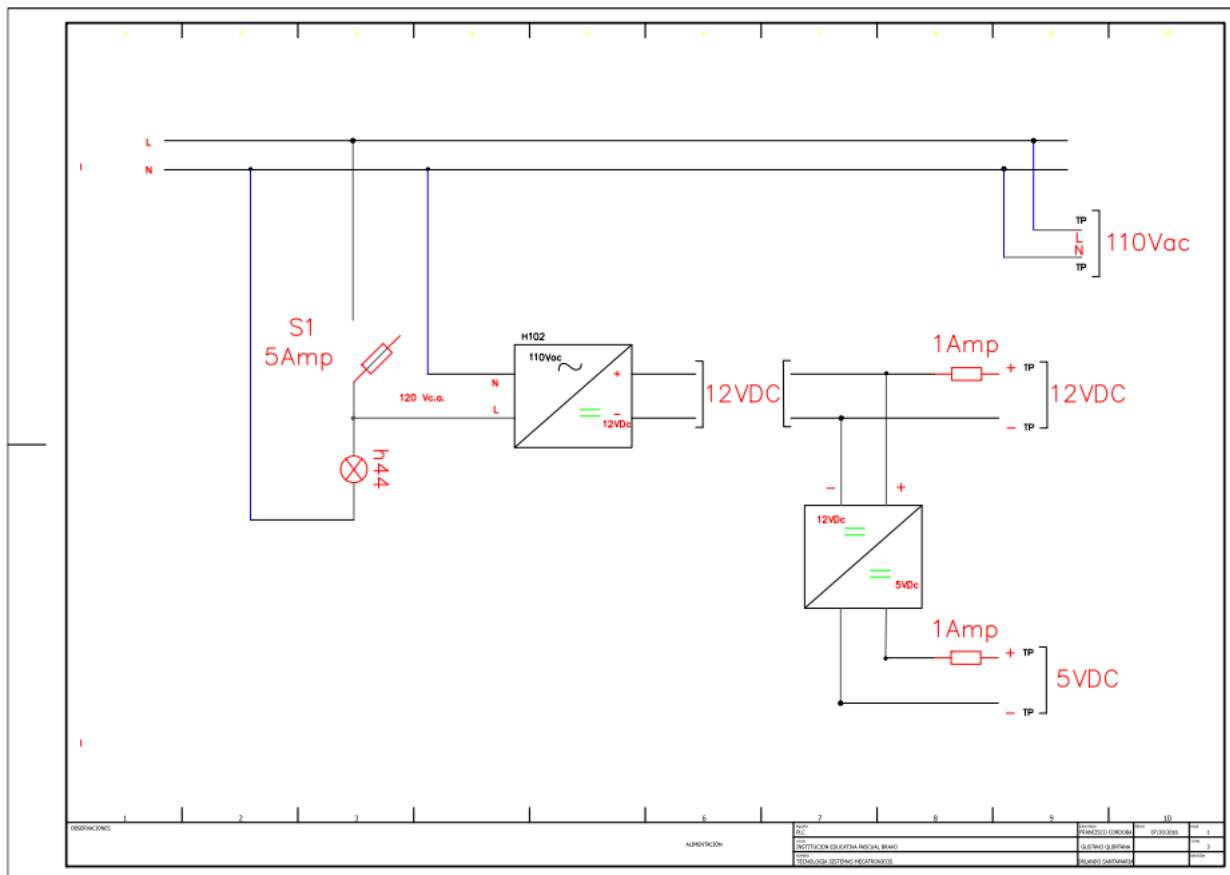


*Figura 26:* Modulo Didáctico Bluetooth HC06.

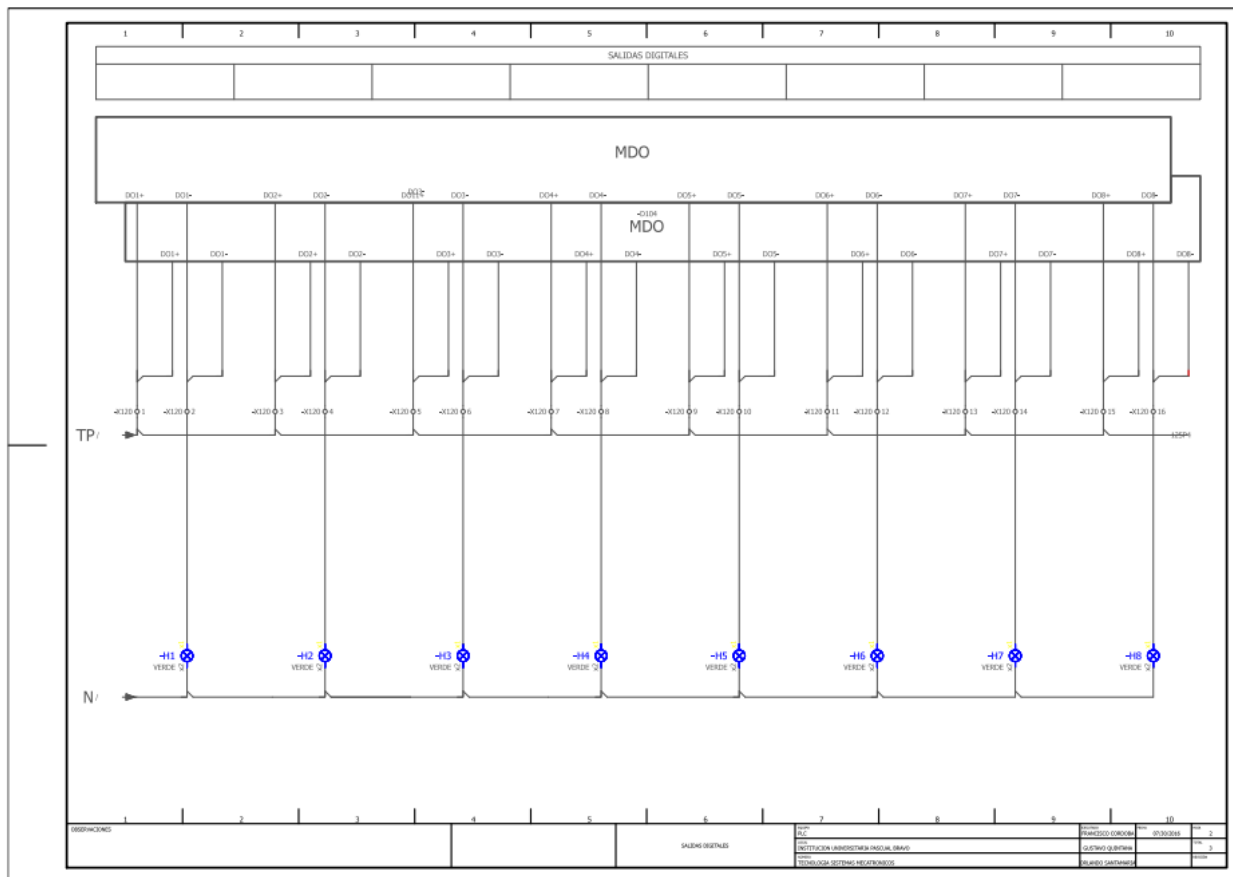
Fuente: Propia Alumnos.

## 6.6.5 Planos Eléctricos De los Modulos Didácticos

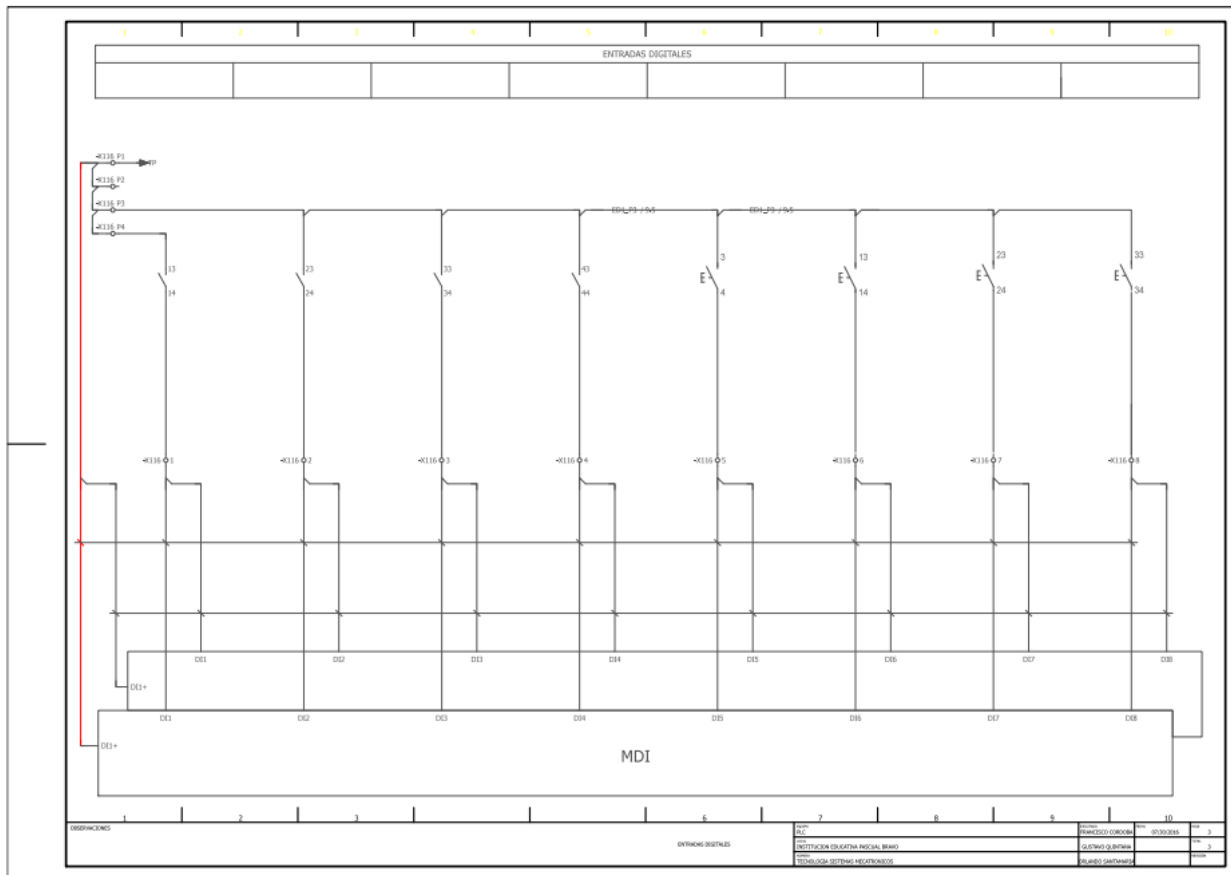
## Conexiones fuente de Alimentación



### Conexiones Salidas Digitales a 110 VAC Para El Modulo De Potencia



## Conexiones De Entradas Digitales Para El Módulo De Potencia



*Figura 27:* Planos Eléctricos Modulo Didáctico.

Fuente: Propia Alumnos.

## 6.7 REGISTRO DE PRUEBAS DE FUNCIONAMIENTO ALGORITMOS

### MODULO MOTORES

La prueba consiste gobernar un motor AC monofásico con la tarjeta de Arduino, empleando el módulo de relevos dispuesto en la maleta. Este módulo servirá como interfaz de

potencia para prevenir daños en el Arduino. El código implementado es el que se aprecia a continuación.

```

/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  Most Arduinos have an on-board LED you can control. On the Uno and
  Leonardo, it is attached to digital pin 13. If you're unsure what
  pin the on-board LED is connected to on your Arduino model, check
  the documentation at http://arduino.cc

  This example code is in the public domain.

  modified 8 May 2014
  by Scott Fitzgerald
  */

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(6, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  digitalWrite(12, HIGH);
  digitalWrite(11, HIGH);
  digitalWrite(10, HIGH);
  digitalWrite(9, HIGH);
  digitalWrite(8, HIGH);
  digitalWrite(7, HIGH);
  digitalWrite(6, HIGH);
  delay(1000); // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  digitalWrite(12, LOW);
  digitalWrite(11, LOW);
  digitalWrite(10, LOW);
  digitalWrite(9, LOW);
  digitalWrite(8, LOW);
  digitalWrite(7, LOW);
  digitalWrite(6, LOW);
  delay(1000); // wait for a second
}

```

## MODULO BLUETOOTH

Se utiliza un dispositivo móvil (celular con versión android 5.02) para comunicar con la tarjeta bluetooth implementado en la maleta. El objetivo es enviar un dato al arduino por medio de protocolo bluetooth para que sea reenviado al dispositivo móvil y así verificar la comunicación bidireccional entre el Smartphone y el arduino. El código implementado es el que se aprecia a continuación.

```

/* Upload this sketch into Crowduino and press reset*/

#include <SoftwareSerial.h> //Software Serial Port
#define RxD 6
#define TxD 7

SoftwareSerial blueToothSerial(RxD,TxD);

void setup()
{
  Serial.begin(9600);
  pinMode(RxD, INPUT);
  pinMode(TxD, OUTPUT);
  setupBlueToothConnection();
}

void loop()
{
  char recvChar;
  while(1){
    if(blueToothSerial.available()){//check if there's any data sent from the remote bluetooth shield
      recvChar = blueToothSerial.read();
      Serial.print(recvChar);
    }
    if(Serial.available()){//check if there's any data sent from the local serial terminal, you can add the other
applications here
      recvChar = Serial.read();
      blueToothSerial.print(recvChar);
    }
  }
}

void setupBlueToothConnection()
{
  blueToothSerial.begin(38400); //Set BluetoothBee BaudRate to default baud rate 38400
  blueToothSerial.print("\r\n+STWMOD=0\r\n"); //set the bluetooth work in slave mode
  blueToothSerial.print("\r\n+STNA=CrowBTSlave\r\n"); //set the bluetooth name as "CrowBTSlave"
  blueToothSerial.print("\r\n+STPIN=0000\r\n");//Set SLAVE pincode"0000"
  blueToothSerial.print("\r\n+STOAUT=1\r\n"); // Permit Paired device to connect me
  blueToothSerial.print("\r\n+STAUTO=0\r\n"); // Auto-connection should be forbidden here
  delay(2000); // This delay is required.
  blueToothSerial.print("\r\n+INQ=1\r\n"); //make the slave bluetooth inquirable
  Serial.println("The slave bluetooth is inquirable!");
  delay(2000); // This delay is required.
  blueToothSerial.flush();
}

```

## MODULO ETHERTNET

Se programó en el Arduino una página simple en HTML que muestra la conversión análoga digital de cinco canales, la dirección de la página es 190168.1.10, se conecta por cable de red standar. El código implementado es el que se aprecia a continuación.

```

/*
  Web Server

  A simple web server that shows the value of the analog input pins.
  using an Arduino Wiznet Ethernet shield.

  Circuit:
  * Ethernet shield attached to pins 10, 11, 12, 13
  * Analog inputs attached to pins A0 through A5 (optional)

  created 18 Dec 2009
  by David A. Mellis
  modified 9 Apr 2012
  by Tom Igoe

  */

#include <SPI.h>
#include <Ethernet.h>

// Enter a MAC address and IP address for your controller below.
// The IP address will be dependent on your local network:
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED
};
IPAddress ip(192, 168, 1, 177);

// Initialize the Ethernet server library
// with the IP address and port you want to use
// (port 80 is default for HTTP):
EthernetServer server(80);

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for Leonardo only
  }

  // start the Ethernet connection and the server:
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.print("server is at ");
  Serial.println(Ethernet.localIP());
}

```



```

}

void loop() {
  // listen for incoming clients
  EthernetClient client = server.available();
  if (client) {
    Serial.println("new client");
    // an http request ends with a blank line
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);
        // if you've gotten to the end of the line (received a newline
        // character) and the line is blank, the http request has ended,
        // so you can send a reply
        if (c == '\n' && currentLineIsBlank) {
          // send a standard http response header
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println("Connection: close"); // the connection will be closed after completion of the response
          client.println("Refresh: 5"); // refresh the page automatically every 5 sec
          client.println();
          client.println("<!DOCTYPE HTML>");
          client.println("<html>");
          // output the value of each analog input pin
          for (int analogChannel = 0; analogChannel < 1; analogChannel++) {
            int sensorReading = analogRead(analogChannel);
            client.print("el voltaje es: ");
            client.print(analogChannel);
            client.print(" is ");
            client.print(sensorReading);
            client.println("<br />");
          }
          client.println("</html>");
          break;
        }
        if (c == '\n') {
          // you're starting a new line
          currentLineIsBlank = true;
        }
        else if (c != '\r') {
          // you've gotten a character on the current line
          currentLineIsBlank = false;
        }
      }
    }
    // give the web browser time to receive the data
    delay(1);
    // close the connection:
    client.stop();
    Serial.println("client disconnected");
  }
}

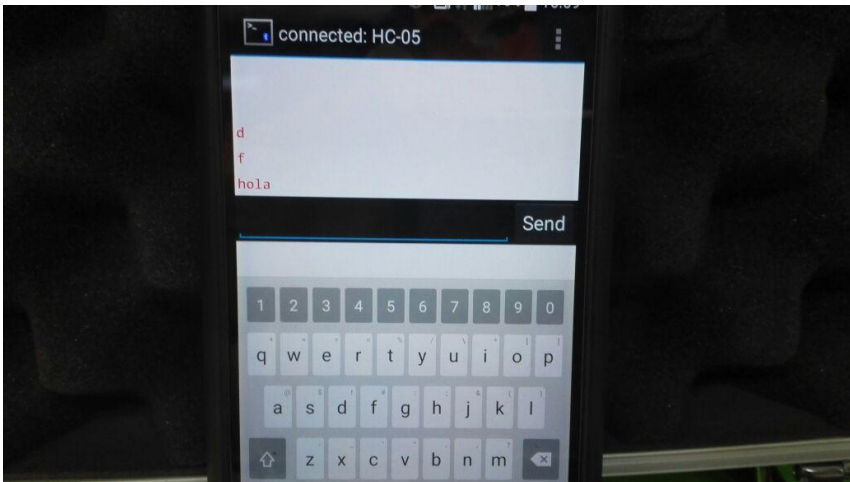
```

## MODULO Xbee

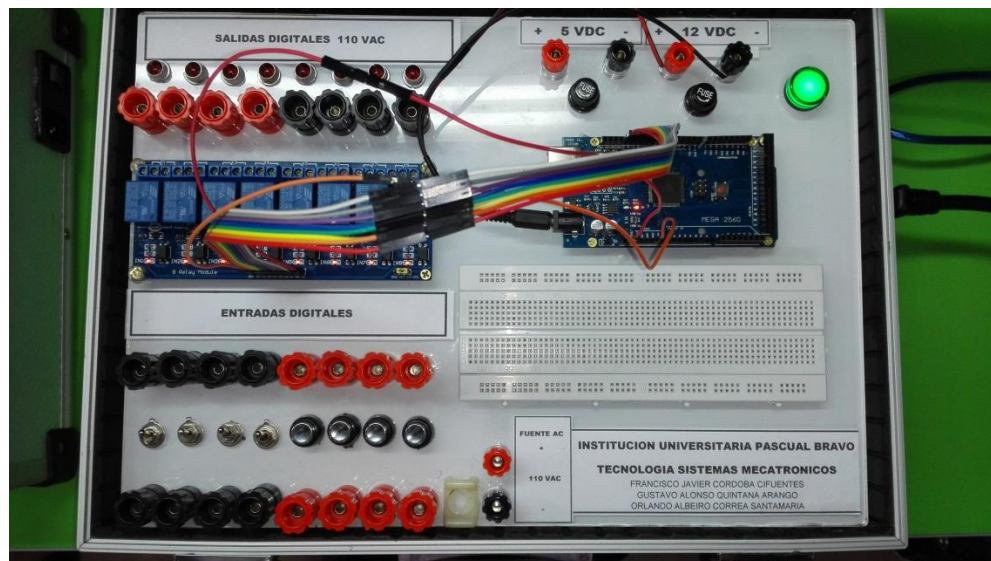
Se configuraron los módulos uno como coordinador y el otro como router. Igualmente se configuraron para realizar comunicación cruzada solo entre ellos con el fin de evitar interferencias de otros modulos. Se enviaron una serie de caracteres de un modulo a otro conectando el coordinador al Arduino y el router a un PC obteniendo una comunicación full dúplex exitosa. . El código implementado es el que se aprecia a continuación.

```
#include <SoftwareSerial.h>
      // RX, TX
char on = 'i';           // Carácter para prender el led
char off = 'o';         // Carácter para apagar el led
char lectura = 0;
int led = 13;
void setup() {
  Serial.begin(9600);    // Baud rate monitor serial
  Serial.println( "XBees conectados" );
  pinMode(led, OUTPUT); // Led como salida
  delay(1000);          // Baud rate Xbee
}
void loop() {
  lectura = Serial.read(); // Lectura del xbee
  if( lectura == on){
    digitalWrite(led, HIGH); // Prender el led
    Serial.println( lectura ); // Imprimir lectura en el monitor serial
    Serial.println( "LED ON" );
  }
  if(lectura == off)
  {
    digitalWrite(led, LOW); // Apagar el led
    Serial.println( lectura ); // Imprimir en el monitor serial
    Serial.println( "LED OFF" );
  }
  delay( 100);
}
```

## 6.8 REGISTRO FOTOGRAFICO DE PRUEBAS DE FUNCIONAMIENTO



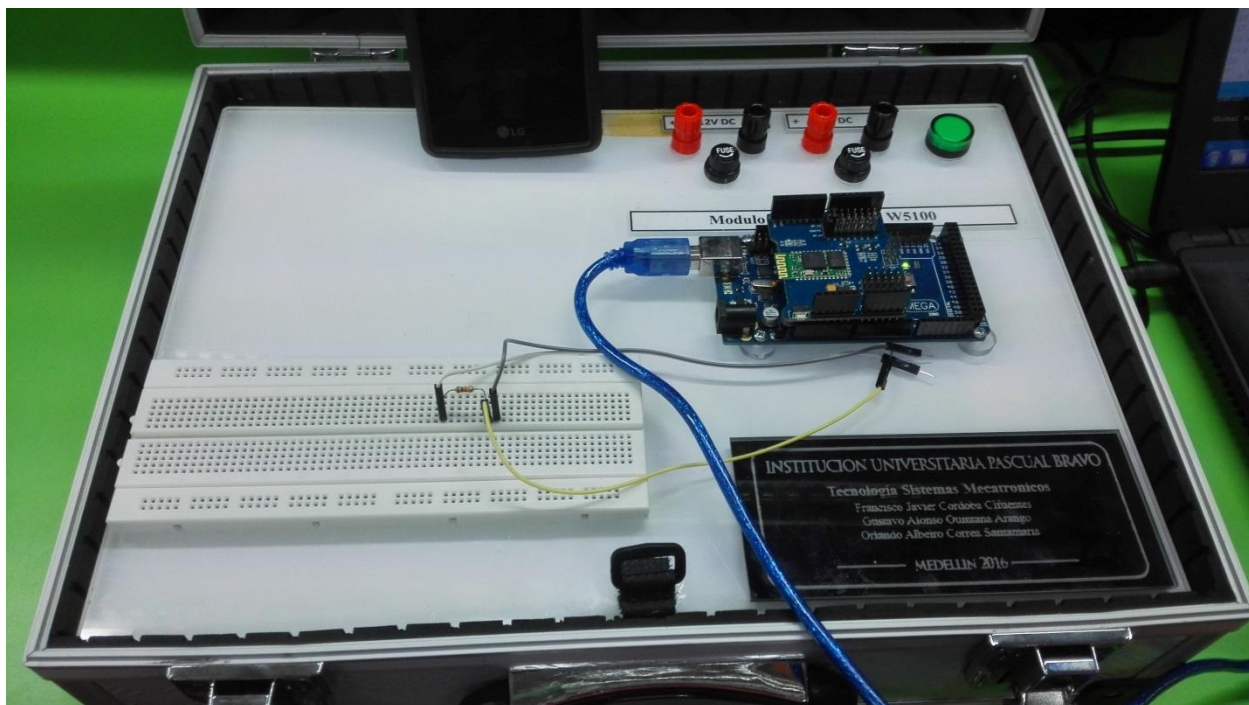
*Figura 28:* Prueba de funcionamiento de Modulo Didáctico XBee.  
Fuente: Propia Alumnos.



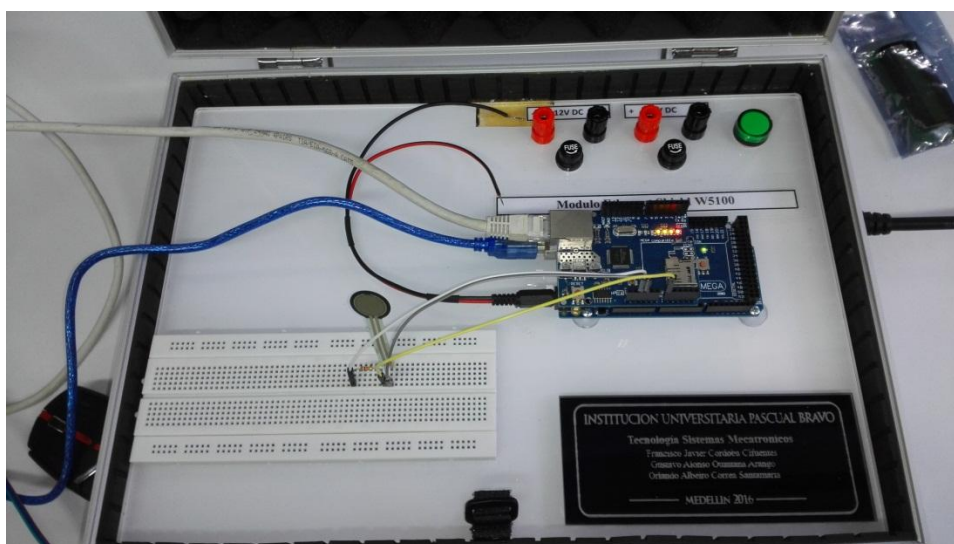
*Figura 29:* Prueba de funcionamiento de Módulo Didáctico Motores.  
Fuente: Propia Alumnos.



*Figura 30:* Prueba de funcionamiento de Módulos Didácticos.  
Fuente: Propia Alumnos.



*Figura 31:* Prueba de funcionamiento de Módulo Didáctico Bluetooth.  
Fuente: Propia Alumnos.



*Figura 32:* Prueba de funcionamiento de Módulo Didáctico Ethernet.  
Fuente: Propia Alumnos.

## 6.9 MATERIALES Y COSTOS

- Maletas multifuncionales plásticas,
- Láminas de acrílico,
- Bisagras,
- Abrazaderas plásticas,
- Fuentes reguladas CA 100-240V a DC 12 V 5A 60W,
- Reguladores 12V a 5V 3A 15W, protoboard 830 puntos MB 102,
- Cables conectores para protoboard,
- Pantalla LCD 16 x 2,
- Conectores hembra tipo banana,
- Accionamientos tipo start NA,
- Accionamientos tipo stop NC,
- Tarjetas Arduino Mega 2560 R3,
- Módulo Relé 8 canales con opto Arduino,
- Módulos X-Bee serie 2 con antena- cable B24-Z7WIT-004, cable
- Módulo Bluetooth + cable USB,
- Shields Bluetooth,
- Adaptador X-Bee USB,
- Shields Ethernet R3,
- Sensor ultrasonido HC-SR04,
- Sensor de distancia x infrarrojos 0A41SK,
- Sensor de presión FSR402,
- Sensor termopar tipo K,

**Tabla 2. Costo Recursos materiales**

MATERIALES	CANTID AD	VALOR UNIT. EN \$	VALOR TOTAL EN \$
Resma de papel	1	10.000	10.000
Maleta	5	45.000	225.000
Lámina de acrílico	5	50.000	250.000
Bisagra	10	1.000	10.000
Abrazadera plástica pqte. x 50	1	5.000	5.000
Fuente regulada CA 100-240V a DC 12V 5A 60W	5	60.000	300.000
Regulador 12V a 5V 3A 15W	5	20.000	100.000
Protoboard 830 puntos MB 102	5	10.000	50.000
Cables conectores para protoboard pqte. x 120	1	30.000	30.000
Pantalla LCD 16 x 2	1	8.500	8.500
Conector hembra tipo banana	12	1.000	12.000
Accionamientos tipo start NA	10	6.000	60.000
Accionamientos tipo stop NC	10	6.000	60.000
Tarjetas Arduino Mega 2560 R3	5	50.000	250.000
Módulos X-Bee serie 2 con antena-cable B24-Z7WIT-004	1	80.000	80.000
Módulo Relé 8 canales con opto Arduino	1	36.000	36.000
Módulo Bluetooth + cable USB	2	22.000	44.000
Shields Bluetooth	2	30.000	60.000
Adaptador X-Bee USB	1	24.000	24.000

Shields Ethernet R3	1	55.000	55.000
Sensor ultrasonido HC-SR04	2	7.000	14.000
Sensor de distancia x infrarrojos 0A41SK	2	24.000	48.000
Sensor de presión FSR402,	2	55.000	110.000
Sensor termopar tipo K,	2	20.000	40.000

---

Elaboró: Francisco Javier Córdoba Cifuentes

**Tabla 3. Costos varios**

VARIOS			
CONCEPTO	CANTIDAD	VALOR UNITARIO EN \$	VALOR TOTAL EN \$
TRANSPORTE	20	6.500	130.000

---

Elaboró: Francisco Javier Córdoba Cifuentes

Este proyecto tiene un costo aproximado de \$2.571.500



## 7 CONCLUSIONES

7.1 luego de finalizada la prueba se evidenció el correcto funcionamiento del programa para el arranque de un motor ejecutando desde el Arduino.

7.2 Se realizó la prueba del módulo Ethernet y se observó cómo se transfirió la información desde un PC a la tarjeta Ethernet mostrando una respuesta ágil y rápida.

7.3 en el caso del sistema bluetooth se utilizaron varios dispositivos que tenían una versión de Android inferiores a cinco, con los cuales no fue posible la comunicación. Solo fue posible con un celular que posee una versión Android 5.02. Por lo tanto es esencial una versión Android superior a cinco para una comunicación efectiva.

7.4 Con los Xbee pro se estableció una comunicación inalámbrica entre dos módulos, los cuales fueron configurados uno como emisor y otro como receptor, desde el teclado del PC se enviaron una serie de caracteres entre los módulos evidenciando con claridad cuál fue la ruta de origen y la respuesta rápida en su destino, se imprime en el monitor serial, comprobando la comunicación entre ambas tarjetas.

7.5 Fue posible implementar los módulos didácticos para las prácticas de los estudiantes de regionalización haciendo el montaje y pruebas de funcionamiento, cumpliendo a cabalidad el objetivo principal del proyecto.

## BIBLIOGRAFÍA

- ZigBee Tutorial. Tomado el 29 de febrero de 2016, de <http://www.tutorial-reports.com/wireless/zigbee/tutorial.php>
- Consideraciones de diseño aplicadas al uso de ZigBee. Tomado el 29 de febrero de 2016, de <http://www.rfdesignline.com/howto/lowpower/rf/197007510>
- Ernest, E.: Sistemas de medición e instrumentación. Recuperado el 7 de marzo de 2016, de <http://www.tirant.com/derecho/libro/sistemas-de-medicion-e-instrumentacion--diseno-y-aplicacion-ernest-e-doebelin-9789701049778>
- Creus, Antonio: Instrumentación industrial. Recuperado el 7 de marzo de 2016, de <http://www.freelibros.org/ingenieria-industrial/instrumentacion-industrial-8va-edicion-antonio-creus-sole.html>
- Dmitruk, Andrés. E., Departamento de Ingeniería e Investigaciones Tecnológicas, UNLaM.  
Recuperado el 15 de marzo de 2016, de [El software y los sistemas embebidos Universidad de la Plata.](#)
- Proyectos con arduino paso a paso. Recuperado el 16 de marzo de 2016, de <https://arduino paso a paso.wordpress.com/>
- Cómo un proyecto Arduino puede convertirse en un producto real. Tomado el 16 de marzo de 2016, de <https://es.wikipedia.org/wiki/Arduino>
- RFC0826: Un Protocolo Para la Resolución de Dirección Ethernet. Tomado el 22 de marzo de 2016, de <http://www.rfc-es.org/rfc/rfc0826-es.txt>
- Introduces\_bluetooth\_low\_energy\_wireless\_technology\_the\_next\_generation\_of\_bluetooth\_wireless\_te. Tomado el 22 de marzo de 2016 de, <https://www.bluetooth.com/>
- Bluezona. ¿Qué es Bluetooth?. Recuperado el 22 de marzo de 2016, de

[http://www.bluezona.com/index.php?option=com\\_content&task=view&id=25&Itemid=50/](http://www.bluezona.com/index.php?option=com_content&task=view&id=25&Itemid=50/)

Arduino mega. Tomado el 19 de abril de 2016, de <https://www.arduino.cc/en/Main/arduinoBoardMega>

Comunicación de arduino con puerto serie. Tomado el 19 de abril de 2016, de <http://www.luisllamas.es/2014/04/arduino-puerto-serie/>