MÓDULO DIDÁCTICO DE SENSORICA PARA LABORATORIO DE MECATRONICA
ETAPA 7

TOBÓN RESTREPO IDER ANDRÉS
AGUDELO GARCÍA VÍCTOR JAVIER

Asesor
CARLOS ALBERTO VALENCIA HERNÁNDEZ
Magister en Automatización y Control Industrial

INSTITUCIÓN UNIVERSITARIA PASCUAL BRAVO
FACULTAD DE INGENIERÍA
TEGNOLOGÍA MECATRÓNICA

MEDELLIN ANTIOQUIA
2016

Contenido

1. IDENTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA 7
2. JUSTIFICACIÓN 8
3. OBJETIVOS 9
3.1. GENERAL 9
3.2. ESPECÍFICOS 9
4. REFERENTES TEORICOS 10
4.1 Reseña Histórica de la Institución 10
4.2. MECATRONICA Y CONCEPTOS BÁSICOS 11
4.3. Clasificación de los instrumentos 13
4.4. Diagrama en bloques del sensor 13
4.5. Sensor infrarrojo 13
4.6. Sensor de humedad y temperatura 14
4.7. Sensor de luz 14
4.8. Sensor detector de lluvia 14
4.9. Sensor de movimiento 14
4.10. Sensor ultrasónico 15
4.11. Motores paso a paso 15
4.12. Servomotor 20
5. METODOLOGIA 22
5.1. Tipo de investigación 22
5.2. Método escogido 22
5.3. Aportes del proyecto 22
5.3.1. Aporte científico 22
5.3.2. Aporte social. 22

5.3.3. Aporte práctico. 23

5.4. Técnicas de recolección de datos 23

6. DISEÑO ESTRUCUTURAL DEL MODULO 24

Las medidas son: 25

7. IDENTIFICAR LOS DISPOSITIVOS DE MEDICIÓN Y CONTROL PERTINENTES A LA CARRERA DE MECATRÓNICA 26

7.1. Sensor óptico 26

7.2. Sensor de humedad y temperatura 27

7.3. Sensor detección de luz 28

7.4. Sensor detector de lluvia YL-83 30

7.5. Sensor de movimiento PIR 31

7.6. Sensor ultrasónico HC-SR04 32

7.7. Motor paso a paso 34

7.8. Servomotores 34

8. DISEÑO ELECTRONICO DE LOS MODULOS DESENSORICA 36

Conexión de sensor CNY-70 36

Conexión sensor temperatura y humedad 37

Conexión de sensor de luz GY-302 37

Conexión Sensor de lluvia 38

8.1. Conexión de sensor de movimiento PIR 39

8.2. Conexión sensor ultrasonido 40

8.3. Conexión motor paso a paso 41

8.4. Conexión de servomotor 41

9. DISEÑO DE ALGORITMOS DE CONTROL Y-O ADQUISICIÓN DE DATOS 43

9.1. Sensor CNY-70 43

9.2. Sensor de humedad y temperatura DHT-11 44
9.3. Sensor de luz GY-302 45
9.4. Sensor de lluvia 47
9.5. Sensor de movimiento PIR 49
9.6. Sensor ultrasónico HC-SR04 52
9.7. Motor paso a paso 54
9.8. Servomotores 56
10. RECURSOS ..... **¡Error! Marcador no definido.**
10.1. RECURSOS HUMANOS ..... **¡Error! Marcador no definido.**
10.2. RECURSOS FISICOS ..... **¡Error! Marcador no definido.**
11. BIBLIOGRAFIA 58
12. ANEXOS 61

TABLA DE FIGURAS

Figura 1. Distribución del bobinado de un motor unipolar 17
Figura 2. Secuencia Normal 18
Figura 3. Wave drive 19
Figura 4. Medio paso 20
Figura 5. Diseño modulo 24
Figura 6. Foto maletín 25
Figura 7. Módulo CNY70 26
Figura 8.Recepción y transmisión de gatos CNY_70 27
Figura 9. Modulo temperatura y humedad 28
Figura 10. Modulo detención de luz 30
Figura 11. Modulo lluvia 31
Figura 12. Modulo detención de movimiento 32
Figura 13. Modulo detencion de movimiento sin proteccion 32
Figura 14. Modulo proximidad 33
Figura 15. Paso a paso 34
Figura 16. Encoder 34
Figura 17. Servomotor 35
Figura 18. Conexión arduino y CNY7-0 37
Figura 19. Conexión arduino y DTH11 37
Figura 20. Conexión arduino y GY-302 38
Figura 21. Conexión al arduino con YL-83 39
Figura 22. Conexión arduino y PIR 40
Figura 23. Conexión arduino y HC-SR04 40
Figura 24. Conexión arduino y motor paso a paso 41
Figura 25. Conexión arduino y servomotor 42

INTRODUCCION

En la industria de procesos de manufactura es necesario garantizar que los productos que se fabrican, cumplan con características definidas por el usuario final. Algunas de estas características son una excelente calidad que nos permita competir en todo mercado a bajo costo.

A principios del siglo XX para garantizar estas especificaciones era necesario monitorear y controlar las variables de manera manual; esta función la cumplían los operarios que recibían capacitación y entrenamiento para cumplir este cargo de control.

La globalización a finales del siglo XX hasta nuestros días, exigió a las empresas competir con mayor calidad y a reducir los costos. Por lo anterior los procesos se hicieron más complejos y difíciles de controlar por los operarios y el desarrollo de la electrónica y la informática permitió desarrollar dispositivos electrónicos inteligentes que cumplieran esta labor de monitorear y controlar las variables del proceso de una manera precisa sin la intervención del operario y garantizar las especificaciones del producto sin importar lo complejo del proceso de fabricación.

En este proyecto se estudiaran los sensores, los actuadores, los conceptos involucrados en la medición y control de las variables y los demás instrumentos de control y la finalidad es que los estudiantes de Mecatrónica del Tecnológico Pascual Bravo I.U. tengamos un mejor conocimiento y entendimiento sobre la sensores y su funcionamiento practicando por medio del módulo ya dicho.

1. IDENTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA

La Institución Universitaria Pascual Bravo. Es una de las instituciones más reconocidas a nivel nacional por su calidad educativa y cuenta con una amplia oferta. También cuenta con una gran cantidad de laboratorios educativos para prácticas mecánicas, eléctricas, motores eléctricos, motores a combustión, soldadura, entre otros.

Para el laboratorio de Mecatrónica se cuenta con elementos para la práctica, pero como estudiantes apreciamos la falta de elementos didácticos para la comprensión del funcionamiento de sensores. Teniendo en cuenta lo anterior vemos la necesidad de implementar un módulo didáctico para estudiar el funcionamiento de diferentes sensores contribuyendo así a la mejora en la capacitación de estudiantes y profesores.

2. JUSTIFICACIÓN

La Mecatrónica tiene como finalidad ser un un sistema compuesto por mecanismos, actuadores, controles inteligentes y sensores. Por lo tanto, busca integrar diferentes esquemas de control y comunicación entre los componentes; como estudiantes de Mecatrónica centramos nuestro interés en el desarrollo y/o elaboración del módulo de sensórica, no solo permitirá que los alumnos del pascual bravo se les pueda brindar todos los elementos necesarios para el aprendizaje, sino también la adquisición de destrezas necesarias para un conocimiento más profundo de estos dispositivos, ya que observamos la falencia que existe en la institución actualmente en cuanto llevar estos conceptos como teórico practico.

Otro factor importante en la elaboración del módulo de sensórica es la utilización y uso académico que servirá a los demás estudiantes tanto de tecnología mecatrónica, como a los estudiantes de tecnología electrónica, eléctrica, electromecánica e ingeniería eléctrica, los cuales son un porcentaje alto de alumnos de la institución.

3. OBJETIVOS

3.1. GENERAL

Diseñar y construir un módulo didáctico de sensores para el laboratorio de Mecatrónica de la Institución Universitaria Pascual Bravo

3.2. ESPECÍFICOS

Diseñar de la estructura física del módulo; teniendo como idea un diseño elegante y a la vez hermético, facilidad para su transporte y práctica.

Identificar los dispositivos de medición y control pertinentes a la carrera de Mecatrónica; con los conocimientos adquiridos en nuestra disciplina y dar un enfoque creativo brindar conocimientos a los demás.

Implementar el diseño eléctrico de los módulos de sensórica; con asesoría, investigación y basándonos en trabajos ya realizados

Diseño de algoritmos de control y adquisición de datos; mediante el software arduino, investigación y análisis de laboratorio.

Realizar pruebas para verificar el funcionamiento del módulo; para estas pruebas será necesario el software de arduino, verificaciones en monitor serial y ajustes con instrumentos de medición

4. REFERENTES TEORICOS

4.1 Reseña Histórica de la Institución

En 1930 surge la necesidad de empezar a formar personal calificado, con un aprendizaje que pudiera ayudar a mejorar el nivel de vida de las familias. El desarrollo de la industria naciente era ya una preocupación para las autoridades locales de dirigir entonces un aprendizaje hacia un oficio que permitiera al obrero, vincularse al proceso productivo.

La creación de la Escuela de Artes y Oficios requirió inversión monetaria para su funcionamiento, ya que no se contaba con las instalaciones adecuadas: talleres, aulas de clase, maquinaria y herramientas.

Por disposición de la Honorable Asamblea Departamental, mediante Ordenanza No. 37 del 24 de julio de 1935, fue creada la escuela de Artes y Oficios como secciones de la Universidad de Antioquia.

La Asamblea Departamental de Antioquia por Ordenanza No. 56 del 4 de julio de 1938 cambió su nombre por " Escuela de Artes y Oficios Pascual Bravo", en honor al héroe antioqueño, uno de los más jóvenes y epónimos gobernantes que se registran en nuestra historia. Un año más tarde, mediante Decreto 2350, el Ministerio de Educación Nacional recibe esta Institución de la Universidad de Antioquia.

Ordenanza 56 de 1938 "Se construyó en los talleres de la misma Escuela, un busto en bronce y de algunos relieves sobre la vida de aquel patriota y estadista para ser colocados en el patio principal y frente a los edificios en el sector de Robledo donde está instalada la Institución."

Esta primera descripción se basa en los textos de la Ley y Ordenanzas departamentales referenciadas, el libro "Los Partidos Políticos en Colombia" de Jorge Ospina Londoño y en informes del Director de la Escuela de Artes y Oficios al Director de Educación Pública y al Rector de la Universidad de Antioquia en los años (1936-1939).

Por medio del Decreto 108 de 1950 el Congreso de la República convierte la "Escuela Industrial de Artes y Oficios Pascual Bravo" en "Instituto Técnico Superior Pascual Bravo" y continúa dependiendo del Ministerio de Educación Nacional.

En 1957 el Instituto Técnico Superior Pascual Bravo se convirtió en uno de los mejores de América del Sur; produjo maquinaria en sus propios talleres, sus laboratorios de electrotecnia se contaban entre los más modernos de todo el continente. La Institución estableció estrechos contactos con empresarios e industriales de la región, ofreciendo educación pertinente a sus necesidades.

4.2. MECATRONICA Y CONCEPTOS BÁSICOS

4.2.1. Arduino

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares.

El hardware consiste en una placa con un microcontrolador Atmega AVR y puertos de entrada y salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, Atmega8, por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque que es ejecutado en la placa. Se programa en el ordenador para que la placa controle los componentes electrónicos.

Arduino puede tomar información del entorno a través de sus entradas analógicas y digitales, puede controlar luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador.

También cuenta con su propio software que se puede descargar de su página oficial que ya incluye los drivers de todas las tarjetas disponibles lo que hace más fácil la carga de códigos desde el computador.

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software tal como Adobe Flash, Processing, Max/MSP, Pure Data. Una tendencia tecnológica es utilizar Arduino como tarjeta de adquisición de datos desarrollando interfaces en

software como JAVA, Visual Basic y LabVIEW. Las placas se pueden montar a mano o adquirirse. El entorno de desarrollo integrado libre se puede descargar gratuitamente.

4.3.1. Conceptos de Medición y Control

Para definir las características de los instrumentos de medición y control, se requiere conocer el lenguaje propio de la instrumentación. Los siguientes conceptos le permitirán entender las especificaciones de los instrumentos.

4.3.2. Campo de Medida

Es el conjunto de valores comprendidos entre el valor inferior y el valor superior de la escala de medida del instrumento. Por ejemplo para un medidor de temperatura se definen como campo de medida los valores entre 0°C y 200°C.

4.3.3. Alcance

Es la diferencia algebraica entre el valor superior y el valor inferior de la escala del instrumento. Por ejemplo, el alcance del instrumento de temperatura anterior, es de 200°C.

4.3.4. Error

Es la diferencia del valor medido de la variable y el valor verdadero. Matemáticamente se puede determinar el error en porcentaje, empleando la siguiente ecuación:

Error = (Valor medido – Valor verdadero) x 100%

4.3.5. Precisión

Es la tolerancia de medida del instrumento y define los errores que el instrumento arroja cuando se utiliza en condiciones normales de servicio. La precisión es definida en fábrica y no puede modificarse.

4.3. Clasificación de los instrumentos

Los instrumentos de medición y control se pueden clasificar de acuerdo con la función que cumplen en el proceso de la siguiente manera:

4.3.1. Sensor o transductor

Está en contacto con la variable (presión, temperatura, etc.) y genera una señal en función del cambio que experimenta dicha variable. La señal generada por el sensor por lo general es de tipo eléctrico y es de corriente o voltaje. Las señales normalizadas son de 4 a 20 mA y 0 a 10 Vdc respectivamente.

4.4. Diagrama en bloques del sensor

4.4.1. Indicador

Indica el valor de la variable. Se clasifican en análogos, cuando se componen de aguja indicadora y escala numérica; y en digitales compuestos por una pantalla donde se visualiza el valor.

4.4.2. Registrador

Registra el valor de la variable con trazos continuos sobre el papel. Los datos del comportamiento de la variable del proceso se almacenan en el papel. Su funcionamiento es similar al de un aparato que monitorea el funcionamiento del corazón (electrocardiograma).

También se puede registrar el valor de la variable utilizando tarjetas electrónicas de adquisición de datos que reciben las señales eléctricas de los sensores y envían los datos al computador por el puerto serial o USB. En este caso el registrador es el computador, que tiene la capacidad de almacenar los datos.

4.5. Sensor infrarrojo

Es un sensor óptico reflexivo que tiene una construcción compacta dónde los diodos tanto el emisor de luz como el receptor están ubicados en la misma dirección para detectar la presencia de un objeto utilizando la reflexión del infrarrojo sobre el objeto. La longitud de onda de trabajo es950nm. El detector consiste en un fototransistor.

4.6. Sensor de humedad y temperatura

Un sensor de temperatura y humedad es un dispositivo ideal para sistemas de medición climatológico o para controles de temperatura y humedad tanto para la industria, como la domótica y otros ambientes; de fácil manejo y programación.

4.7. Sensor de luz

Este es un sensor de luz que detecta la iluminancia en el lugar dispuesto. Miden la cantidad de luz que llega a una célula foto-eléctrica (básicamente una resistencia). La resistencia es baja con luz y alta con oscuridad (sensor de oscuridad). Un Sensor fotoeléctrico es un dispositivo electrónico que responde al cambio en la intensidad de la luz. Estos sensores requieren de un componente emisor que genera la luz, y un componente receptor que toma la luz dada por el emisor. Todos los diferentes modos de sensado se basan en este principio de funcionamiento. Están diseñados especialmente para la detección, clasificación y posicionamiento de objetos; además de todo lo que nuestras mentes pudan diseñar a partir de esta aplicación.

4.8. Sensor detector de lluvia

El sensor detector de lluvia es capaz de detectar gotas de agua lluvia, por lo que puede ser utilizado para sistemas de detección que requieran realizar funciones cuando comience a llover.

Este módulo consiste en una serie de pistas conductoras impresas sobre una placa de baquelita. La separación entre las pistas es muy pequeña. Lo que en este módulo se crea un corto circuito cada vez que a las pistas se les sometan al contacto con agua. El agua hace que se cree un camino de baja resistencia entre las pistas con polaridad positiva y las pistas conectadas al GND.

4.9. Sensor de movimiento

Un sensor de movimiento se utiliza para detectar si un ser humano, un objeto o un animal se ha movido dentro o fuera de la gama del sensor. Son pequeñas, de bajo costo, bajo consumo de energía, fácil de usar y no se desgastan. Por esa razón, se encuentran comúnmente en los

electrodomésticos y aparatos utilizados en los hogares o negocios. Ellos se refieren a menudo como PIR, "infrarrojo pasivo", "piroeléctrico", o "IR" sensores de movimiento.

Los sensores de movimiento son básicamente de un sensor piroeléctrico (que se puede ver arriba como la lata de metal redondo con un cristal rectangular en el centro), que puede detectar los niveles de radiación infrarroja. Todo emite algo de radiación de bajo nivel, y ese algo más caliente, más radiación es emitida. El sensor en un detector de movimiento es en realidad dividido en dos mitades. La razón de ello es que estamos buscando para detectar movimiento (modificar) no los niveles promedio de IR. Las dos mitades están conectadas de modo que se anulan entre sí. Si uno ve la radiación media más o menos IR que el otro, la salida oscilará alta o baja.

4.10. Sensor ultrasónico

El HC-SR04 es un sensor ultrasónico de bajo costo que no sólo puede detectar si un objeto se presenta, como un sensor PIR (Passive Infrared Sensor), sino que también puede sentir y transmitir la distancia al objeto.

Tienen dos transductores, básicamente, un altavoz y un micrófono.

Ofrece una excelente detección sin contacto (remoto) con elevada precisión y lecturas estables en un formato fácil de usar.

4.11. Motores paso a paso

Un motor “paso a paso” (o “PAP”) es un dispositivo electromecánico capaz de convertir una serie de impulsos eléctricos en desplazamientos angulares discretos. Esto significa que, a diferencia de un motor convencional (que gira de forma continua), es capaz de avanzar una serie de grados (o pasos) a la vez, dependiendo del estado de sus entradas de control. Un motor paso a paso se comporta de la misma manera que un convertidor digital-analógico y puede ser gobernado por impulsos procedentes de sistemas lógicos, tales como microcontroladores u ordenadores.

Los motores PAP son ideales para la construcción de mecanismos en donde se requieren movimientos muy precisos.

Los motores paso a paso se pueden ver como motores eléctricos sin escobillas. Es típico que todos los bobinados del motor sean parte del estator, y el rotor puede ser un imán permanente. Como se mencionó anteriormente está constituido esencialmente por dos partes:

Una fija llamada estator construida a base de cavidades en las que van depositadas las bobinas que excitadas convenientemente formarán los polos norte-sur de forma que se cree un campo magnético giratorio. Una móvil, llamada rotor construida mediante un imán permanente, con el mismo número de pares de polos, que el contenido en una sección de la bobina del estator; este conjunto va montado sobre un eje soportado por dos cojinetes que le permiten girar libremente.

La conmutación se debe manejar de manera externa con un controlador electrónico y, típicamente, los motores y sus controladores se diseñan de manera que el motor se pueda mantener en una posición fija y también para que se lo pueda hacer girar en un sentido y en el otro.

Comportamiento propio de los motores paso a paso:

Los motores paso a paso tienen un comportamiento del todo diferente al de los motores de corriente continua. En primer lugar, no giran libremente por sí mismos. Los motores paso a paso, como lo indica su nombre, avanzan girando por pequeños pasos. También difieren de los motores de CC en la relación entre velocidad y torque (un parámetro que también es llamado "par motor" y "par de giro"). Los motores de CC no son buenos para ofrecer un buen torque a baja velocidad sin la ayuda de un mecanismo de reducción. Los motores paso a paso, en cambio, trabajan de manera opuesta: su mayor capacidad de torque se produce a baja velocidad.

Si bien es cierto que los motores paso a paso funcionan controlados por un pulso de avance, el control de un motor paso a paso no se realiza aplicando en directo este pulso eléctrico que lo hace avanzar. Estos motores tienen varios bobinados que, para producir el avance de ese paso, deben ser alimentados en una adecuada secuencia. Si se invierte el orden de esta secuencia, se logra que el motor gire en sentido opuesto. Si los pulsos de alimentación no se proveen en el orden correcto, el motor no se moverá apropiadamente. Puede ser que zumbe y no se mueva, o puede ser que gire, pero de una manera tosca e irregular.

Tipos de motores paso a paso

Motor paso a paso unipolar

Los motores unipolares son relativamente fáciles de controlar, gracias a que poseen devanados duplicados. Aunque para facilitar el esquema se dibuja este devanado como una bobina con punto medio, en realidad tienen dos bobinas en cada eje del estator, que están unidas por extremos opuestos, de tal modo que al ser alimentada una u otra, generan cada una un campo magnético inverso al de la otra. Nunca se energizan juntas: por eso lo correcto es decir que tienen una doble bobina, en lugar de decir (como se hace habitualmente) que es una bobina con punto medio. Esta duplicación se hace para facilitar el diseño del circuito de manejo, ya que permite el uso, en la parte de potencia, de un transistor único por cada uno de los bobinados.

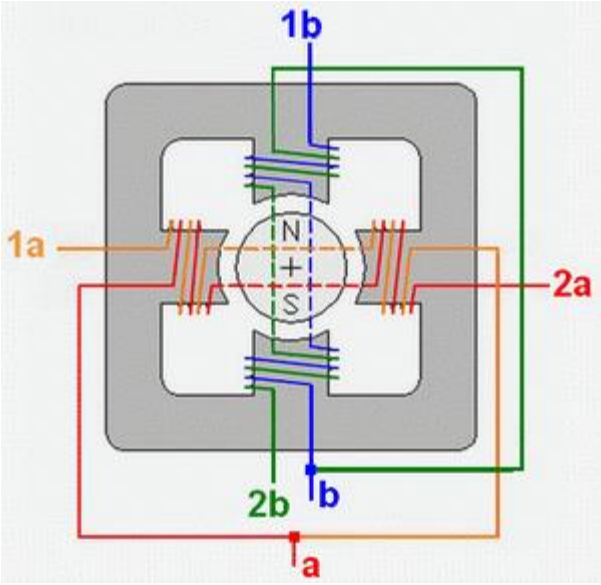


Figura 1. Distribución del bobinado de un motor unipolar

Fuente: (Travel, 2015)

En el esquema más común de conexión se unen los "puntos medios" de ambos ejes (a y b en el dibujo) y se les conecta al positivo de la alimentación del motor. El circuito de control de potencia, entonces, se limita a poner a masa los bobinados de manera secuencial.

La secuencia de pulsos de un motor unipolar se puede controlar con un contador binario de dos bits con un decodificador, como por ejemplo el integrado CD 4017. La parte de potencia puede ser implementada con un único transistor en cada bobinado.

Existen tres secuencias posibles para este tipo de motores:

Secuencia Normal: Esta es la secuencia más usada y la que generalmente recomienda el fabricante. Con esta secuencia el motor avanza un paso por vez y debido a que siempre hay al menos dos bobinas activadas, se obtiene un alto torque de paso y de retención.

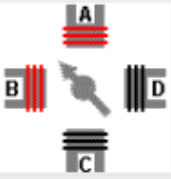
PASO	Bobina A	Bobina B	Bobina C	Bobina D	
1	ON	ON	OFF	OFF	
2	OFF	ON	ON	OFF	
3	OFF	OFF	ON	ON	
4	ON	OFF	OFF	ON	

Figura 2. Secuencia Normal

Fuente: (Valenzuela, 2006)

Secuencia del tipo wave drive: En esta secuencia se activa solo una bobina a la vez. En algunos motores esto brinda un funcionamiento más suave. Pero al estar solo una bobina activada, el torque de paso y retención es menor.

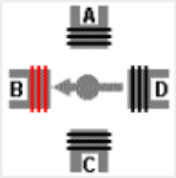
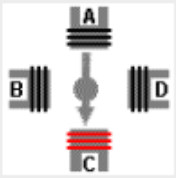
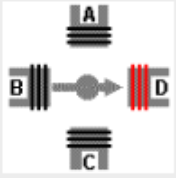
PASO	Bobina A	Bobina B	Bobina C	Bobina D	
1	ON	OFF	OFF	OFF	
2	OFF	ON	OFF	OFF	
3	OFF	OFF	ON	OFF	
4	OFF	OFF	OFF	ON	

Figura 3. Wave drive

Fuente: (Valenzuela, 2006)

Secuencia del tipo medio paso: En esta secuencia se activan las bobinas de tal forma de brindar un movimiento igual a la mitad del paso real. Para ello se activan primero 2 bobinas y luego solo 1 y así sucesivamente. Como vemos en la tabla la secuencia completa consta de 8 movimientos en lugar de 4.

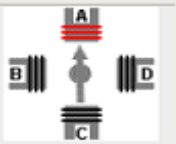
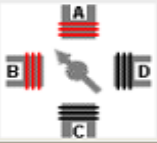
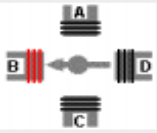
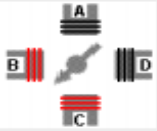
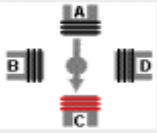
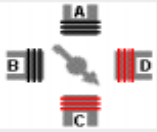
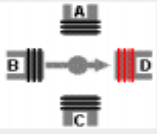
PASO	Bobina A	Bobina B	Bobina C	Bobina D	
1	ON	OFF	OFF	OFF	
2	ON	ON	OFF	OFF	
3	OFF	ON	OFF	OFF	
4	OFF	ON	ON	OFF	
5	OFF	OFF	ON	OFF	
6	OFF	OFF	ON	ON	
7	OFF	OFF	OFF	ON	
8	ON	OFF	OFF	ON	

Figura 4. Medio paso

Fuente: (Valenzuela, 2006)

4.12. Servomotor

Un servomotor (también llamado servo) es un dispositivo similar a un motor de corriente continua que tiene la capacidad de ubicarse en cualquier posición dentro de su rango de operación, y mantenerse estable en dicha posición.1

Un servomotor es un motor eléctrico que puede ser controlado tanto en velocidad como en posición.

Es posible modificar un servomotor para obtener un motor de corriente continua que, si bien ya no tiene la capacidad de control del servo, conserva la fuerza, velocidad y baja inercia que caracteriza a estos dispositivos.

Características

Está conformado por un motor, una caja reductora y un circuito de control. También potencia proporcional para cargas mecánicas. Un servo, por consiguiente, tiene un consumo de energía reducido.

La corriente que requiere depende del tamaño del servo. Normalmente el fabricante indica cuál es la corriente que consume. La corriente depende principalmente del par, y puede exceder un amperio si el servo está enclavado.

En otras palabras, un servomotor es un motor especial al que se ha añadido un sistema de control (tarjeta electrónica), un potenciómetro y un conjunto de engranajes. Con anterioridad los servomotores no permitían que el motor girara 360 grados, solo aproximadamente 180; sin embargo, hoy en día existen servomotores en los que puede ser controlada su posición y velocidad en los 360 grados. Los servomotores son comúnmente usados en modelismo como aviones, barcos, helicópteros y trenes para controlar de manera eficaz los sistemas motores y los de dirección.

Control

Los servomotores hacen uso de la modulación por ancho de pulso (PWM) para controlar la dirección o posición de los motores de corriente continua. La mayoría trabaja en la frecuencia de los cincuenta Hertz, así las señales PWM tendrán un periodo de veinte milisegundos. La electrónica dentro del servomotor responderá al ancho de la señal modulada. Si los circuitos dentro del servomotor reciben una señal de entre 0,5 a 1,4 milisegundos, éste se moverá en sentido horario; entre 1,6 a 2 milisegundos moverá el servomotor en sentido anti horario; 1,5 milisegundos representa un estado neutro para los servomotores estándares.

5. METODOLOGIA

5.1. Tipo de investigación

Para este proyecto se tomó en cuenta lo importante que es el desarrollo en todos los campos generados por los avances tecnológicos y electrónicos, a la vez los avances del conocimiento y estos superados continuamente.

La investigación para este trabajo fue importante tener presente la búsqueda de todo conocimiento acorde al desarrollo de este proyecto como el modelo de investigación experimental.

5.2. Método escogido

Para la realización de este proyecto se optó por un análisis profundo y amplio sobre los principales beneficiados generados por el fin de este proyecto y a su vez un análisis asertivo sobre su funcionalidad y cada uno de sus componentes.

Primero se buscó la principal necesidad en el aula de prácticas para los estudiantes de Mecatrónica, de ahí optamos por buscar una forma en la cual los estudiantes pudieran poner sus conocimientos a prueba, por eso escogió el modulo el cual contiene motores y sensores como una forma de práctica.

5.3. Aportes del proyecto

5.3.1. Aporte científico.

Científicamente el proyecto aporta modernización y ampliación del conocimiento dentro de la Institución Universitaria Pascual Bravo, no solo a las generaciones de ahora sino futuras, haciendo más fácil para los estudiantes en la incursión en el campo industrial, Dando calidad en el desarrollo de nuevas aplicaciones.

5.3.2. Aporte social.

En el área social generar un sentido de pertenencia y mejora en la calidad educativa. También para obtener conocimiento para el desarrollo de las aplicaciones en pro del bienestar humano.

5.3.3. Aporte práctico.

En la práctica, se puede decir que este es el fin de proyecto, ya que su objetivo a realizar es para que los estudiantes de la Institución Universitaria Pascual Bravo en el área mecatrónica tengan donde hacer su experimentos y mejoras practicando por medio de este módulo.

5.4. Técnicas de recolección de datos

En la realización del proyecto se investigó textos sobre la funcionalidad de los sensores, la funcionalidad de los motores.

Después de la investigación, se tomó la información básica y necesaria para la creación de cada uno de los puntos a seguir, generando positivamente una ruta segura y precisa para llegar al objetivo propuesto.

Ya con la información obtenida, se obtuvo de una manera clara las posibles mejoras y adecuaciones a realizarle al trabajo, también obteniéndola por medio de bibliografía, donde se toma una información más amplia y concisa.

6. DISEÑO ESTRUCUTURAL DEL MODULO

Se optó por una maleta de plástico con bordes de aluminio que soporta el paso del tiempo, las condiciones extremas de humedad, el cambio brusco del clima, por ende, es un material que nos permite el uso en todas las circunstancias, no se oxidan, no presentan herrumbres en su superficie, están diseñadas para soportar pesos importantes y permanecer en las mismas condiciones en su vida útil. Por estos motivos encontramos grandes beneficios para nuestro proyecto y se decidió incorporarla realizándole internamente una distribución con acrílico para allí incorporar los módulos de este proyecto.

Todo el montaje se hizo en un maletín de plástico con bordes de aluminio con doble broche de seguridad para un cierre seguro; de esta manera vimos una buena opción para asegurar los elementos allí contenidos y su fácil transporte. En su interior posee 6 sensores, 2 motores, fuente, protoboar, módulo arduino, cables de conexión.

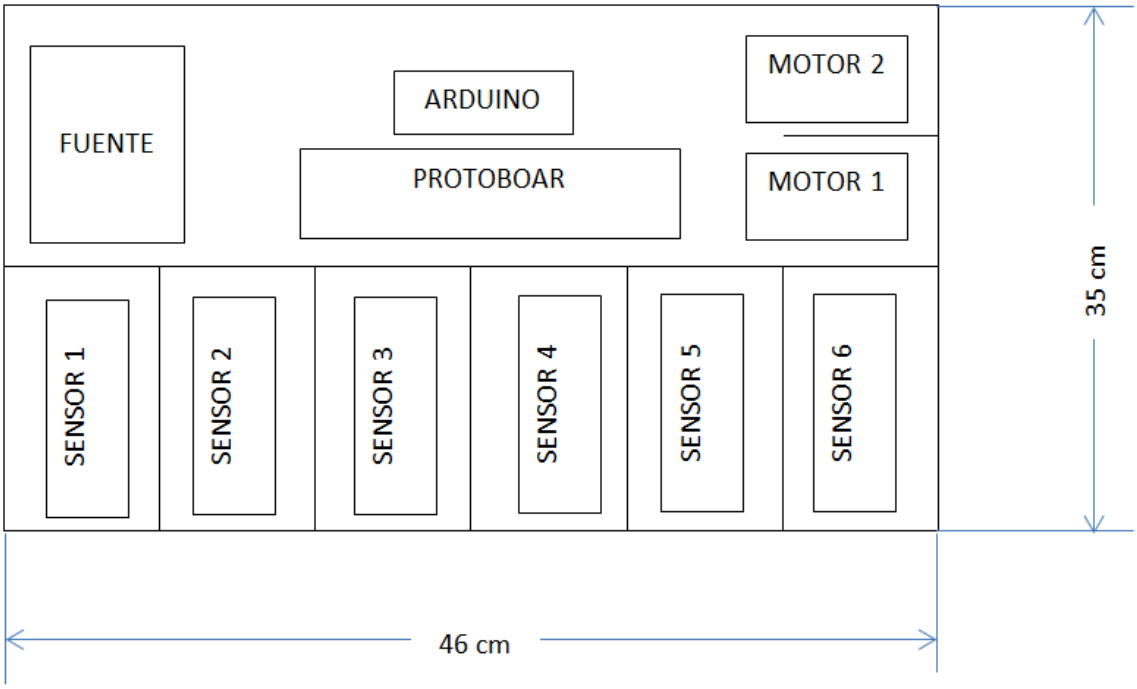


Figura 5. Diseño modulo

Fuente: Autores

Las medidas son:

Largo: 46 cm

Ancho: 15 cm

Alto: 35 cm

Figura 6. Foto maletín

Fuente: Autores

7. IDENTIFICAR LOS DISPOSITIVOS DE MEDICIÓN Y CONTROL PERTINENTES A LA CARRERA DE MECATRÓNICA

Se tomó como idea del proyecto la implementación y programación de algunos sensores con la tarjeta de desarrollo arduino como complemento de estos. De esta manera implementamos un módulo didáctico para el aprendizaje y practica de los alumnos de Mecatrónica y áreas afines para que con esta herramienta puedan facilitar su aprendizaje y uso de estos sensores.

Con el fin de llevar a cabo lo anterior se decidió utilizar los siguientes sensores:

7.1. Sensor óptico

Escogimos este sensor por su aplicación y su compatibilidad con arduino dentro de sus ventajas encontramos que puede funcionar como escáner optoelectrónico y detector de movimiento de objetos, es decir, sensor de índice, lectura de discos codificados etc., (codificador opto electrónico montado como sensor de cambio de marcha).

Dentro de sus características tenemos la construcción compacta con distancia de del centro a centro de 0.1 '(pulgadas), entre emisor y receptor, no necesita ningún ambiente especial Señal de salida alta, el coeficiente de temperatura bajo, detector provisto de filtro óptico, el ratio de corriente de transferencia (CTR) típico es del 5%.

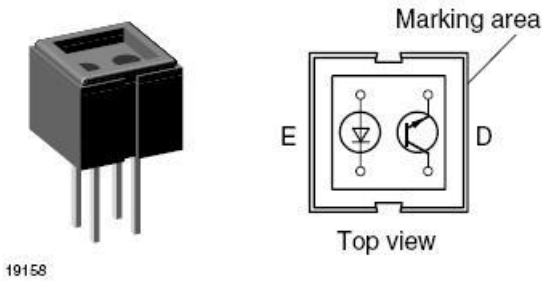


Figura 7. Módulo CNY70

Fuente:(Vishay, 2012)

Longitud de onda del haz infrarrojo de 950nm.

Intensidad del diodo emisor I=50mA.

Intensidad de colector Ic=50mA.

Tensión colector emisor Ice=32V

Tensión emisor colector Vec=7V.

Consumo aproximado de 200mW.

Distancia de detección de 0.3 a 5mm

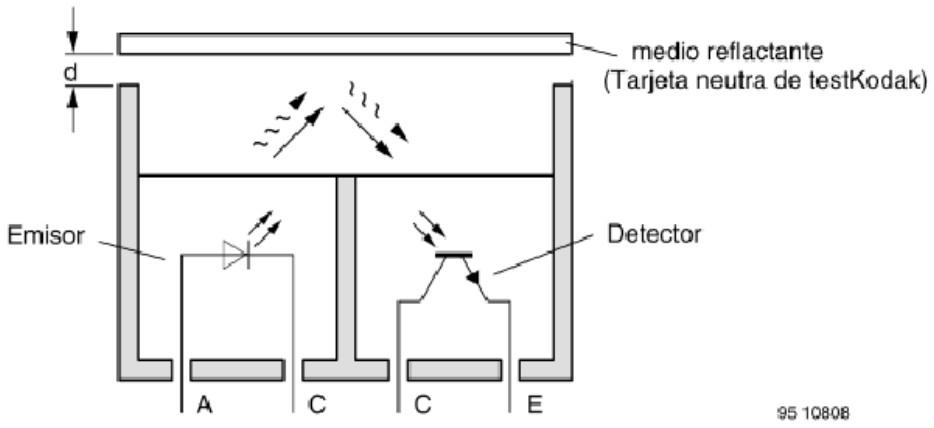


Figura 8.Recepción y transmisión de gatos CNY_70

Fuente: (Pinotolosa, 2015)

7.2. Sensor de humedad y temperatura

Este sensor de temperatura y humedad posee un sensor de capacidad para medir la humedad y un termistor para medir la temperatura del aire que lo rodea, esto nos pareció muy particular ya que estamos incluyendo dos sensores en uno, al investigar sobre este vimos fácil su manejo y programación y que este era compatible con arduino; además dispone de una salida de señal digital calibrada con la temperatura y humedad, asegurando la alta fiabilidad y una excelente estabilidad a largo plazo. Y cuenta con un microcontrolador de alto rendimiento de 8 bits. Cuenta con una excelente calidad, respuesta rápida, capacidad anti-interferencia y ventajas de costo.

Especificaciones:

Voltaje de entrada: $3Vdc \leq Vcc \leq 5Vdc$

Conector de alta calidad

Rango de temperatura: 0-50 °C error de ± 2 °C

Humedad: 20-90% RH ± 5% RH error

Interface: Digital

Tiempo de respuesta: 1s

Tamaño: 12 x 15.5 x 5.5mm

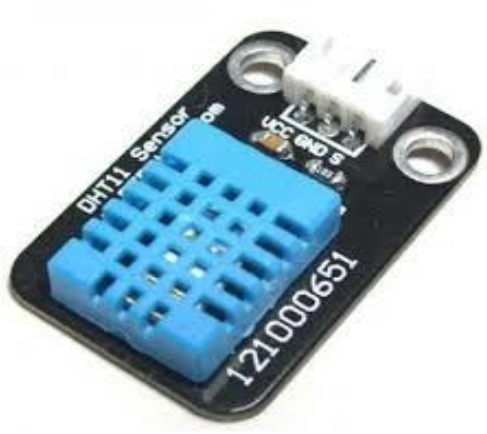


Figura 9. Modulo temperatura y humedad

Fuente: (D-robotics, 2010)

7.3. Sensor detección de luz

Este módulo nos pareció novedoso por su simpleza y tamaño reducido, su compatibilidad con arduino y su precio módico.

Es por tanto que el BH1750 como sensor digital de intensidad de luz ambiente, posee un conversor ADC de 16bits interno. Este sensor es una versión mejorada del típico sensor de luz a base de un LDR, el cual simplemente entrega un valor analógico. El BH1750, entrega automáticamente el valor en Lux (desde 1 lx hasta 65535 lx), y se comunica por I2C, con la posibilidad de seleccionar 2 direcciones. Este sensor nos permite medir la cantidad de luz por metro cuadrado que tenemos en algún lugar y poder modificar automáticamente las luces de los alrededores.

Ejemplos de luminosidad:

Noche: 0.001—0.02

Luz Lunar: 0.02—0.3

Nublado Interior: 5—50

Nublado Exterior: 50—500

Soleado Interior: 100-1000

Luz mínima para la lectura: 50—60

Intensidad estándar sistema de video hogareño: 1400

Características Técnicas

Interfaz digital a través de bus I2C con capacidad de seleccionar entre 2 direcciones

Respuesta espectral similar a la del ojo humano

Ejecuta mediciones de luz y convierte el resultado a una palabra digital

Amplio rango de medición 1-65535 lux

Modo de bajo consumo de energía

Rechazo de ruido a 50/60 Hz

Baja dependencia de la medición contra la fuente de luz: halógeno, led, luz incandescente, luz de día, etc.

Es posible seleccionar dos direcciones de esclavo (I2C).

La influencia del espectro infrarrojo es baja.

Voltaje 3.3v-5v

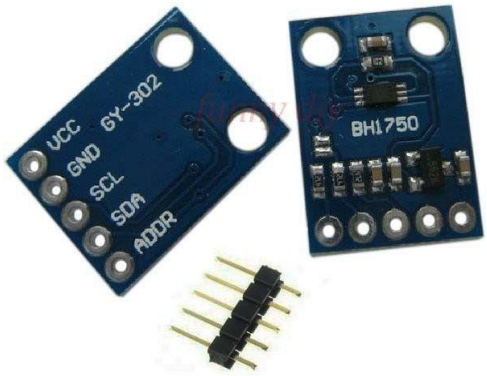


Figura 10. Modulo detención de luz

Fuente: (cmosgr.wordpress, 2015)

7.4. Sensor detector de lluvia YL-83

Nosotros tomamos este sensor por ser un sensor de bajo costo, de pequeño tamaño y es compatible con arduino lo cual nos facilita la aplicación del programa por medio de este dispositivo. El sensor posee un amplificador operacional, específicamente el circuito integrado LM392. Este es el encargado de amplificar el pequeño diferencial de voltaje que se genera cuando una gota de agua cae sobre las pistas del módulo.

Especificaciones:

Voltaje de trabajo: 3.3VDC a 5VDC.

Señal de salida: Análoga y digital.

Sensibilidad ajuste mediante potenciómetro.

Tarjeta amplificadora: 3.2 x 1.5 cm

Placa detectora de lluvia: 5.5 x 4 cm.

Figura 11. Modulo lluvia

Fuente:(k-electronica, 2015)

7.5. Sensor de movimiento PIR

El módulo Sensor DYP-ME003 movimiento PIR lo escogimos porque lo podemos utilizar en varios tipos de lámparas incandescentes, fluorescentes, zumbador, puertas automáticas, ventiladores eléctricos, lavadora automática y Máquinas secadora y otros dispositivos por su pequeño tamaño. Es un producto de alta tecnología, fácil de programar y es compatible con arduino, especialmente adecuado para las empresas, hoteles, centros comerciales, almacenes, casas familiares y corredores, tomando todas estas aplicaciones que le podemos dar a este sensor nos dimos a la tarea de incorporarlo a nuestro proyecto ya que como estudiantes tenemos más posibilidad de interactuar con él.

Característica

Voltaje de entrada: DC 4.5-20V

Corriente estática: 50uA

Disparador: H-Sí, L-n

Tiempo Bloque: 2.5 S (predeterminado)

Tiempo de retardo: 5 S (predeterminado)

Centro Ángulo: <110 grados

Centro Distancia: 3 m (por defecto) - max 7 m

Tamaño de la lente: Diámetro: 23 mm (predeterminado)

Dimensiones: 32mm * 24mm

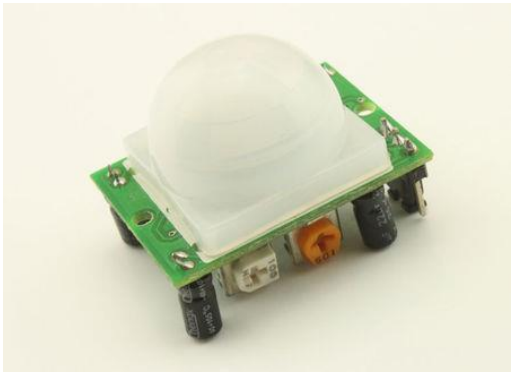


Figura 12. Modulo detención de movimiento

Fuente: (Elecfreaks, 2015)

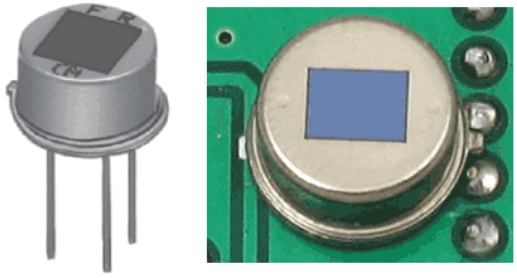


Figura 13. Modulo detencion de movimiento sin proteccion

Fuente: (Elecfreaks, 2015)

7.6. Sensor ultrasónico HC-SR04

Nosotros escogimos este sensor porque es muy utilizado por los estudiantes para sus proyectos de robótica por su precisión, facilidad para adquirirlos. Esta serie de sensores ya lo están adecuando a los vehículos por su gran rango para detectar obstáculos sin la necesidad de un contacto físico. Su funcionamiento no se ve afectado por la luz solar o el material negro como telémetros ópticos (aunque acústicamente materiales suaves como telas pueden ser difíciles de detectar).

La velocidad del sonido en el aire (a una temperatura de 20 ºC) es de 343 m/s. Por cada grado centígrado que sube la temperatura, la velocidad del sonido aumenta en 0,6 m/s

Características:

VCC: Alimentación +5V (4.5V min – 5.5V max)

TRIG: Trigger entrada (input) del sensor (TTL)

ECHO: Echo salida (output) del Sensor (TTL)

GND

Corriente de reposo: < 2mA

Corriente de trabajo: 15mA

Ángulo de medición: 30º

Ángulo de medición efectivo: < 15º

Detección de 2cm a 400cm o 1" a 13 pies (Sirve a más de 4m, pero el fabricante no garantiza una buena medición). “Resolución” La precisión puede variar entre los 3mm o 0.3cm.

Dimensiones: 45mm x 20mm x 15mm

Frecuencia de trabajo: 40KHz

Figura 14. Modulo proximidad

Fuente: (“Sensor Ultrasónico HC-SR04,” n.d.)

7.7.Motor paso a paso

Se tomó la alternativa de seleccionar el motor paso a paso referencia 28BYJ-48 con un encoder ULN2003 este es el encargado de funciones tales como la rotación del motor. Nos pareció importante incluirlo ya que es una pieza fundamental en la robótica, es utilizada en aplicaciones como relojes, pantallas, paneles u otros dispositivos que requieren algún desplazamiento. Para esta presentación tomamos la librería de arduino llamada CustomStepper esta librería nos permite control de ángulo y velocidad del motor

Figura 15. Paso a paso
Fuente: (García, 2014)

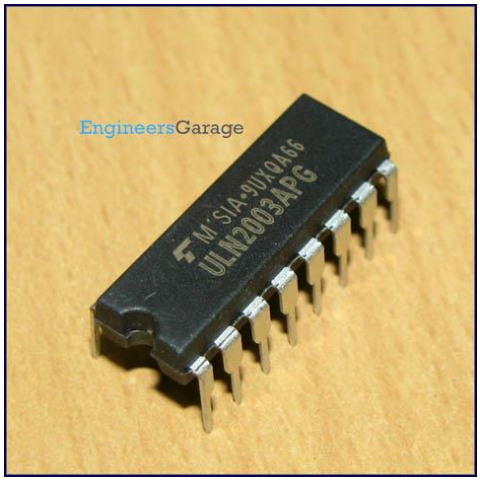


Figura 16. Encoder
Fuente: (Garage, 2012)

Servomotores

Nosotros decidimos incluir un servomotor SG90 Tower Pro a nuestro proyecto, ya que este tipo de servo es ideal para las primeras experiencias de aprendizaje y prácticas con servos, y sus requerimientos de energía son bastante bajos y se permite alimentarlo con la misma fuente de alimentación que el circuito de control. Por ejemplo, si se conecta a una tarjeta arduino, se puede alimentar durante las pruebas desde el puerto USB del PC sin mayor problema, también por su gran precisión, tamaño y costo. El servomotor es un tipo especial de motor de c.c. que se caracterizan por su capacidad para posicionarse de forma inmediata en cualquier posición dentro de su intervalo de operación.

Figura 17. Servomotor

Fuente: (Digest, 2016)

8. DISEÑO ELECTRONICO DE LOS MODULOS DESENSORICA

8.1. Modulo del sensor CNY-70

El sensor CNY-70 para este ejemplo de programa y el arduino uno del módulo va a estar conectado: El emisor (E) al pin análogo 1, y a su vez con una resistencia de 10k a tierra del arduino, el ánodo(A) va conectado al pin 2 de las salidas PWM el arduino, el cátodo (K) va conectado con una resistencia de 150 Ohm a tierra del arduino y el colector (C) va 5v del arduino, como lo podemos observar en la (figura 19).

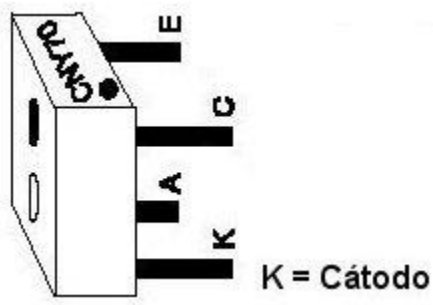


Figura 18. Identificación terminales del Sensor CNY70

Fuente: (GONZÁLEZ, n.d.)

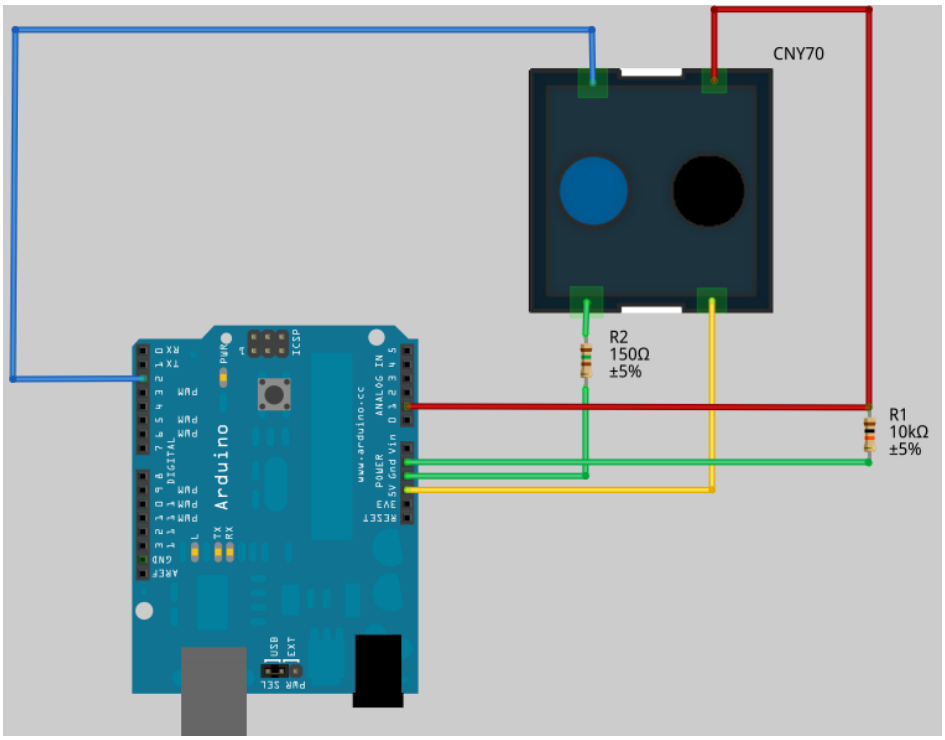


Figura 19. Conexión arduino y CNY7-0

Fuente:(PatagoniaTecnology, 2013)

8.2. Conexión sensor temperatura y humedad

El sensor DTH11 para este ejemplo de programa y el arduino uno del módulo va a estar conectado: El pin DATA del sensor va conectado al pin 2 Digital del arduino, el pin Vcc (+) del sensor va conectado a 5V del arduino y el pin GND (-) del sensor va conectado al GND del arduino, como lo podemos observar en la (figura 20).

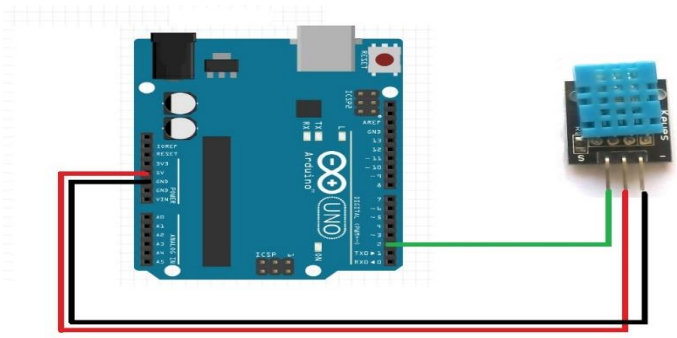


Figura 20. Conexión arduino y DTH11

Fuente: (Lara, 2015)

8.3. Conexión de sensor de luz GY-302

El sensor GY-302 para este ejemplo con el arduino uno del módulo, va a estar conectado de la siguiente manera: El pin VCC del sensor va conectado al pin 5v (+) del arduino, el pin GND del sensor va conectado a tierra del arduino, el pin SCL del sensor va conectado al puerto análogo 5 del arduino, el pin SDA del sensor va conectado al puerto análogo 4 del arduino y el

pin ADD del sensor va conectado al puerto análogo 3 del arduino, como lo podemos observar en la (figura 21).

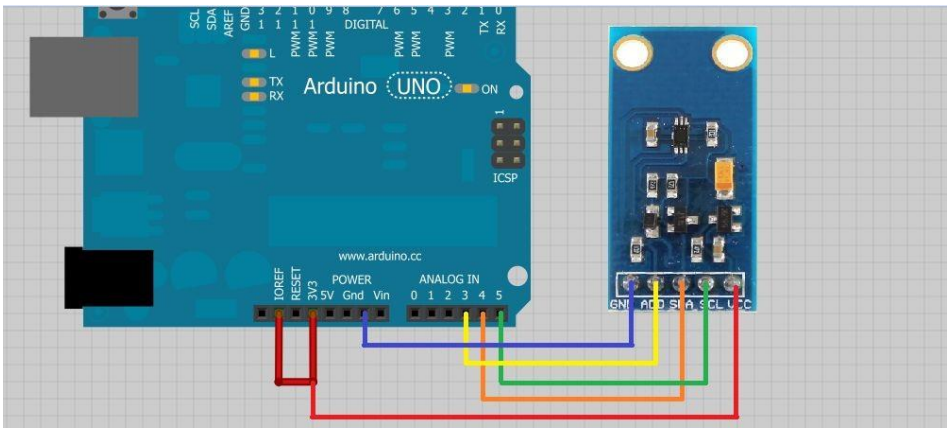


Figura 21. Conexión arduino y GY-302

Fuente: (Leandro, n.d.)

8.4. Conexión Sensor de lluvia

El sensor YL-83 para este montaje en el arduino uno del módulo va a estar conectado: Desde arduino al módulo transmisor por sus extremos se alimenta VCC y GND, en el centro cuenta con una salida análoga y va conectada al puerto análogo 0 del arduino y una salida digital que en este ejemplo no se conecta, al otro extremo va conectado a la placa metálica positivo con positivo y negativo con negativo como muestra la (figura 23).

Figura 22. Placa YL-83.

Fuente: (Antony García, 2014)

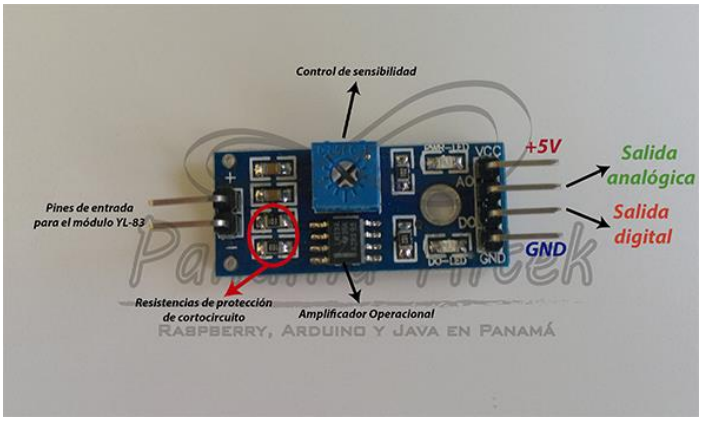


Figura 23. Conexión al arduino con YL-83

Fuente: (Antony García, 2014)

8.5. Conexión de sensor de movimiento PIR

Para el ejemplo del sensor PIR con conexión al arduino lo conectamos de la siguiente manera: El pin 1 que es 5v del sensor va conectado a VCC (+) del arduino, el pin 2 del sensor que es señal va conectado al puerto digital 8 y el pin 3 que es GND del sensor va conectado a tierra (GND) del arduino como muestra la (figura 24)

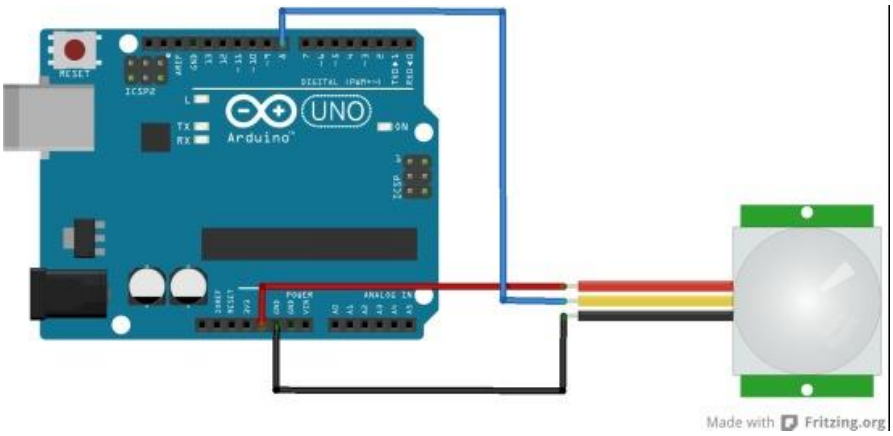


Figura 24. Conexión arduino y PIR

Fuente: (Bolivia, 2015)

8.6. Conexión sensor ultrasonido

Estas serían las respectivas conexiones para el ejemplo que tomamos en este módulo: El pin VCC del sensor va conectado al pin 5v del arduino, el pin ECHO del sensor va al pin 8 digital del Arduino, el pin TRIG del sensor al va al pin 9 digital del Arduino y el pin GND del sensor va conectado a tierra (GND) del arduino como lo observamos en la (figura 25).

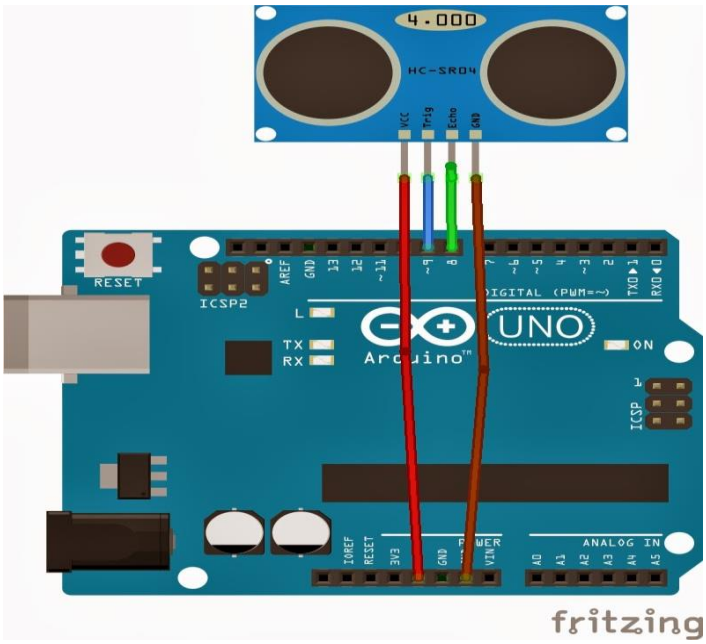


Figura 25. Conexión arduino y HC-SR04

Fuente: (Barbus, 2014)

8.7. Conexión motor pasó a paso

En este ejemplo de motor paso a paso dicha conexión se realiza desde los pines digitales del arduino 8, 9, 10 y11 a las respectivas entradas del encoder UNL2003, dicho encoder es alimentado a (5V, GND) y las salidas del encoder a las entradas del motor paso a paso, como se puede observar en la (figura.26).

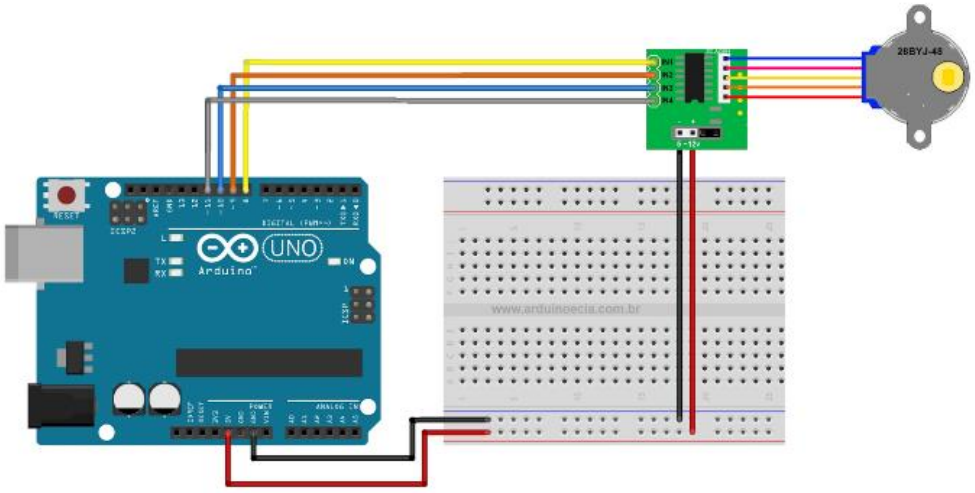


Figura 26. Conexión arduino y motor paso a paso

Fuente: (Cia, 2014)

8.8. Conexión de servomotor

En este montaje conectaremos un servo motor referencia SG90 a nuestro arduino uno de la siguiente manera: Los cables se pueden diferenciar por sus colores; rojo alimentación 5v desde el arduino, café con conexión a tierra del arduino(GND), naranja señal PWM del puerto 8 digital del arduino, como se observa en la (figura27).

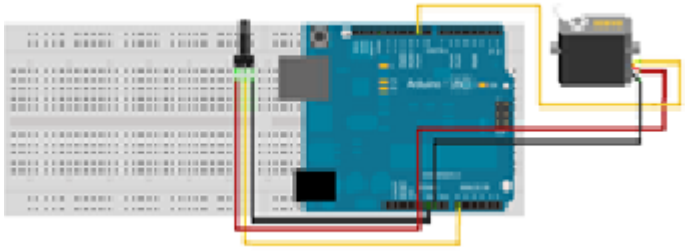


Figura 27. Conexión arduino y servomotor

Fuente: (ArduinoLab, 2012)

9. DISEÑO DE ALGORITMOS DE CONTROL Y-O ADQUISICIÓN DE DATOS

9.1. Sensor CNY-70

Este sería el código del sensor CNY-70. En este caso podemos observar que el sensor funciona detectándonos una presencia de un objeto negro y nos envía una diferencia de voltaje y nos muestra en el monitor serial un 1 a comparación con un color blanco que nos muestra un 0.

```
xxxxxxxxxxxxxxxxxxxxxxxxxxx Sensor CNY70xxxxxxxxxxxxxxxxxxxxxxxxx

/*Por Julio Megía, http://www.tecnosefarad.com liberado para la comunidad

This example code is in the public domain*/
nt ledPin1 = 8;     // pin de un LED
int ledPin2 = 9;     // pin de otro LED
int infraPin = 4;    // pin del infrarrojos utilizado como entrada digital
int valorInfra = 0;  // Valor inicial de la lectura digital del infrarrojos.

void setup() {
  pinMode(ledPin1, OUTPUT);     // Inicializa el pin del LED1 como salida digital
  pinMode(ledPin2, OUTPUT);     // Inicializa el pin del LED2 como salida digital
  pinMode(infraPin, INPUT);     // Inicializa el pin 4 como entrada digital
}

void loop() {
  valorInfra = digitalRead(infraPin);    // Lee el valor de la entrada 4, esto es, el valor que lee el infrarrojo
  digitalWrite(ledPin1, valorInfra);     /* Escribe en el pin 8 el valor que lee la entrada 4, esto es, el mismo valor que lee el infrarrojo
                                          Si el infrarrojo lee 0, entonces, el LED estará apagado
```

Si el infrarrojo lee 1, entonces, el LED estará encendido */

valorInfra = !valorInfra; // Se asigna a valorInfra el valorInfra negado. Si valorInfra es 1, el nuevo valorInfra será 0; y viceversa

digitalWrite(ledPin2, valorInfra); /* Escribe en el pin 9 el valor negado que lee la entrada 4, esto es, el negado del valor que lee el infrarrojo

Si el infrarrojo lee 0, entonces, el LED conectado al pin 9 estará encendido

Si el infrarrojo lee 1, entonces, el LED conectado al pin 9 estará apagado */

}

9.2. Sensor de humedad y temperatura DHT-11

Este sería el código del sensor DHT-11 donde podemos observar que el sensor nos entrega dos variables tanto de humedad como de temperatura y lo podemos observar en el monitor serial. Para este programa es necesario descargar la librería DHT.h en arduino.

Código

xxxxxxxxxxxxxxxxxxxxxxx Sensor DHT-11 xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

```
#include <DHT.h>
#define  DHTPIN 2 //  se selecciona el pin en el que conectamos el sensor
#define DHTTYPE DHT11 // seleccionamos el DHT11
DHT dht (DHTPIN, DHTTYPE); // se inicia una variable que será usada por Arduino para comunicarse con el sensor

void setup() {
  Serial.begin (9600); // se inicia con la comunicacion serial
dht.begin(); //  inicia sensor
}
void loop() {
  float h = dht.readHumidity(); // lectura de humedad
```

```
float t = dht.readTemperature(); // lectura temperatura
// se imprimen las variables
Serial.println("Humedad: ");
Serial.println(h);
Serial.println("Temperatura: ");
Serial.println(t);
delay (5000); // se espera 2 segundos para qur el programa pueda seguir leyendo los datos
}
```

9.3. Sensor de luz GY-302

Este sería el código del sensor GY-302 con arduino donde nos lee las variables de luz que tenemos en el momento y al entrar en contacto con otra luz mayor o menor nos muestra el porcentaje de luz que esta medido

.

Código

xxxxxxxxxxxxxxxxxxxxxxxx Sensor GY-302 xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

```
#include <Wire.h>
int BH1750_address = 0x23;
byte buff[2];

void setup() {

 Wire.begin();
 BH1750_Init(BH1750_address);
 delay(200);
 Serial.begin(9600);
 Serial.println(" iniciando sensor ");

}
void loop(){
```

```
float valf=0;

if(BH1750_Read(BH1750_address)==2){

 valf=((buff[0]<<8)|buff[1])/1.2;

 if(valf<0)Serial.print(">65535");
 else Serial.print ((int)valf,DEC);

 Serial.println(" lx");
}
delay(1000);
}
void BH1750_Init(int address){

Wire.beginTransmission(address);
Wire.write(0x10);//1 [lux] aufloesung
Wire.endTransmission();
}

byte BH1750_Read(int address){

  byte i=0;
  Wire.beginTransmission(address);
  Wire.requestFrom(address, 2);
  while(Wire.available()){
    buff[i] = Wire.read();
    i++;
  }
```

```
  Wire.endTransmission();
  return i;
}
```

9.4. Sensor de lluvia

Este sería el código para el sensor YL-83 con ARDUINO para este ejemplo se diseñó un programa que nos muestra tres estados: el primero cuando el sensor no está en contacto con agua nos muestra en el monitor serial “no hay lluvia”, en el segundo cuando entra en contacto con una gota de agua en una o dos pistas nos muestra “advertencia lluvia” y en el tercer estado cuando el agua entra en contacto con la mayoría de las pistas nos muestra “inundado”.

Código

xxxxxxxxxxxxxxxxxxxxxxx Sensor YL-83 xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

```
// Watch video here: https://www.youtube.com/watch?v=ZktoEX-wERA

/*
  - If the Sensor Board is completely soaked; "case 0" will be activated and " Flood " will be sent to the serial monitor.
  - If the Sensor Board has water droplets on it; "case 1" will be activated and " Rain Warning " will be sent to the serial monitor.
  - If the Sensor Board is dry; "case 2" will be activated and " Not Raining " will be sent to the serial monitor.
*/
// lowest and highest sensor readings:
const int sensorMin = 0;     // sensor minimum
const int sensorMax = 1024;  // sensor maximum

void setup() {
```

```
 Serial.begin(9600);
 pinMode(2, OUTPUT); // red led
 pinMode(3, OUTPUT); // yellow led
 pinMode(4, OUTPUT); // green led
}

void loop() {

 int sensorReading = analogRead(A0); // read the sensor on analog A0
 int range = map(sensorReading, sensorMin, sensorMax, 0, 3); // map the sensor range (four options)
 switch (range) { // range value
  case 0:    // Sensor getting wet
   Serial.println("inundado");
   digitalWrite(2, HIGH);
   digitalWrite(3, LOW);
   digitalWrite(4, LOW);
   break;

  case 1:    // Sensor getting wet
   Serial.println("advertencia lluvia");
   digitalWrite(2, LOW);
   digitalWrite(3, HIGH);
   digitalWrite(4, LOW);
   break;
  case 2:    // Sensor dry - To shut this up delete the " Serial.println("Not Raining"); " below.
   Serial.println("no hay lluvia");
     digitalWrite(2, LOW);
   digitalWrite(3, LOW);
   digitalWrite(4, HIGH);
```

```
  break;
 }
delay(1000);
}
```

9.5. Sensor de movimiento PIR

Este es el código para el módulo Sensor DYP-ME003 donde nos da presencia de un objeto o algo que pase por el lado del sensor, lo podemos visualizar por medio del monitor serial donde nos muestra en que tiempo tomo la lectura de movimiento y en qué tiempo dejo de moverse el objeto.

Código

```
xxxxxxxxxxxxxxxxxxxxxxx Sensor DYP-ME003 xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

 * @ Autor: Kristian Gohlke / krigoo (_) gmail (_) com / http://krx.at
 * @ Fecha: 3. De septiembre de 2006

* http://creativecommons.org/licenses/by-nc-sa/2.0/de/

 * El eje de salida del sensor va a ALTO si el movimiento está presente.
 * Sin embargo, incluso si hay movimiento va a LOW de vez en cuando,
 * Lo que podría dar la impresión de no hay movimiento.
 * Este programa se ocupa de este problema haciendo caso omiso de LOW-fases más corto que un momento dado,
 * Suponiendo que el movimiento continuo está presente durante estas fases.
 *
 */
//////////////////////////
//VARS
```

```
//El tiempo que nos da el sensor a calibrar (10-60 segundos de acuerdo con la hoja de datos)
int calibrationTime = 10;

//El tiempo cuando el sensor emite un impulso de baja
long unsigned int lowIn;

//La cantidad de milisegundos el sensor tiene que ser bajo
//Antes de que asuma todo el movimiento se ha detenido
long unsigned int pause = 5000;

boolean lockLow = true;
boolean takeLowTime;

int pirPin = 7;    //pin digital conectado a la salida del sensor PIR
int ledPin = 8;



////////////////////////////
//SETUP
void setup(){
  Serial.begin(9600);
  pinMode(pirPin, INPUT);
  pinMode(ledPin, OUTPUT);
  digitalWrite(pirPin, LOW);

  //Dar el sensor de un cierto tiempo para calibrar
  Serial.print(" calibrando sensor ");
    for(int i = 0; i < calibrationTime; i++){
      Serial.print(".");
      delay(1000);
```

```
    }
   Serial.println(" hecho");
   Serial.println("SENSOR ACTIVADO");
   delay(50);
  }

//////////////////////////
//LOOP
void loop(){

  if(digitalRead(pirPin) == HIGH){
   digitalWrite(ledPin, HIGH);   //LED visualiza el estado de la salida del sensor
   if(lockLow){
    //Se asegura de que esperar a que la transición a la baja antes de cualquier salida se
hace otra:
    lockLow = false;
    Serial.println("---");
    Serial.print("movimiento encontrado a los ");
    Serial.print(millis()/1000);
    Serial.println(" segundos ");
    delay(50);
    }
    takeLowTime = true;
   }

  if(digitalRead(pirPin) == LOW){
   digitalWrite(ledPin, LOW);  //LED visualiza el estado de la salida del sensor

   if(takeLowTime){
    lowIn = millis();          // guardar el momento de la transición de alta a baja
```

```
        takeLowTime = false;       //  asegurarse de que esto sólo se hace al comienzo de una
fase de baja
      }
      //Si el sensor es baja para más de la pausa dada,
      //Asumimos que el movimiento no más que va a pasar

      if(!lockLow && millis() - lowIn > pause){
        //Se asegura de este bloque de código sólo se ejecuta de nuevo después de
        //Una secuencia de movimiento se ha detectado nuevo
        lockLow = true;
        Serial.print("movimiento detenido a los ");
        Serial.print((millis() - pause)/1000);
        Serial.println(" segundos ");
        delay(50);
        }
      }
    }
```

9.6. Sensor ultrasónico HC-SR04

Este sería el código del sensor HC-SR04 sensor ultrasónico, este programa nos muestra en el monitor serial la distancia del objeto ubicado al frente del sensor hasta 3m, si el objeto que se está sensando entra en un espacio inferior a los 10cm nos muestra la distancia del objeto respecto al sensor acompañado de la palabra “Alarma”.

Código:

```
xxxxxxxxxxxxxxxxxxxxxxx Sensor HC-SR04 xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Sensor de proximidad y al ser inferior a 10cm
envia un pulso de alarma por  el pin 13
```

```
Descargar planos de conexiones en http://elprofegarcia.com/
 */

#define Pecho 6
#define Ptrig 7
long duracion, distancia;

void setup() {
  Serial.begin (9600);       // inicializa el puerto seria a 9600 baudios
  pinMode(Pecho, INPUT);     // define el pin 6 como entrada (echo)
  pinMode(Ptrig, OUTPUT);    // define el pin 7 como salida  (triger)
  pinMode(13, 1);            // Define el pin 13 como salida
  }

void loop() {

  digitalWrite(Ptrig, LOW);
  delayMicroseconds(2);
  digitalWrite(Ptrig, HIGH);   // genera el pulso de triger por 10ms
  delayMicroseconds(10);
  digitalWrite(Ptrig, LOW);

  duracion = pulseIn(Pecho, HIGH);
  distancia = (duracion/2) / 29;           // calcula la distancia en centimetros

  if (distancia >= 500 || distancia <= 0){  // si la distancia es mayor a 500cm o menor a
0cm
    Serial.println("---");                 // no mide nada
  }
  else {
```

```
  Serial.print(distancia);          // envia el valor de la distancia por el puerto serial
  Serial.println("cm");             // le coloca a la distancia los centimetros "cm"
  digitalWrite(13, 0);              // en bajo el pin 13
 }

 if (distancia <= 10 && distancia >= 1){
  digitalWrite(13, 1);                  // en alto el pin 13 si la distancia es menor a 10cm
  Serial.println("Alarma.....");        // envia la palabra Alarma por el puerto serial
 }
 delay(400);                            // espera 400ms para que se logre ver la distancia en la
consola
}
```

9.7. Motor paso a paso

Este programa es un ejemplo para comprender el movimiento de los pasos de el motor aplicando este código para ser controlado con una placa arduino uno.

Código:

```
xxxxxxxxxxxxxxxxxxxxxxx Motor 28BYJ-48 xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

// Programa: control motor 28BYJ-48 paso a paso - Revoluciones
// Autor: Arduino y Co.

#include <CustomStepper.h>

// Establecer los parámetros del motor de pasos de conexión
paso a paso CustomStepper (8, 9, 10, 11, (byte []) {8, B1000, B1100, B0100,
B0110, B0010, B0011, B0001, B1001}, 4075.7728395,   12, CW);
```

```
rotate1 Boolean = false;
rotatedeg Boolean = false;
boolean crotate = false;

void setup ()
{
  // Establecer la velocidad del motor
  stepper.setRPM (12);
   // Establecer el número de pasos por vuelta
  stepper.setSPR (4075.7728395);
}

void loop ()
{
  si (stepper.isDone () == false rotate1 &&)
  {
    retardo (2000);
     // Establecer la dirección de rotación (CW = Tiempo)
    stepper.setDirection (CW);
    // Establece el número de rotaciones
    stepper.rotate (3);
    rotate1 = true;
  }

  si (stepper.isDone () == true && rotate1)
  {
    retardo (2000);
     // Establecer el sentido de giro (CCW = izquierda)
    stepper.setDirection (CCW);
    // Establecer el número de rotaciones
    stepper.rotate (2);
```

```
    rotate1 = false;
   }
   // Comando obligatoria para el funcionamiento de la biblioteca
   stepper.run ();
}
```

9.8. Servomotores

Este es el programa respectivo para el servomotor SG90 Tower Pro donde se puede accionar el movimiento por medio de un potenciómetro.

Coigo:

```
xxxxxxxxxxxxxxxxxxxxxxx Servomotor SG90 xxxxxxxxxxxxxxxxxxxxxxxxxxxxx

#include <Servo.h>

Servo myservo;  // create servo object to control a servo

int potpin = 0;  // analog pin used to connect the potentiometer
int val;    // variable to read the value from the analog pin

void setup()
{
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop()
{
  val = analogRead(potpin);            // reads the value of the potentiometer (value between 0 and 1023)
```

```
    val = map(val, 0, 1023, 0, 179);     // scale it to use it with the servo (value between 0 and 180)
    myservo.write(val);                  // sets the servo position according to the scaled value
    delay(15);                           // waits for the servo to get there
  }
```

10. BIBLIOGRAFIA

http://platea.pntic.mec.es/~lmarti2/veloraton/sensoroptref.htmAntony García, K. N. (2014). Módulo YL-83. Retrieved from http://panamahitek.com/modulo-yl-83-un-detector-de-lluvia/

ArduinoLab. (2012). Controllando un servo con Arduino y un potenciómetro. Retrieved from https://arduinolab.wordpress.com/2012/05/07/controllando-un-servo-con-arduino-y-un-potenciometro-10/

Barbus, E. (2014). sensor ultrasonidos HC-SR04. Retrieved from http://elcajondeardu.blogspot.com.co/2014/03/tutorial-sensor-ultrasonidos-hc-sr04.html

Bolivia, R. y D. en. (2015). Modulo sensor de movimiento (PIR). Retrieved from http://www.systems-technology.net/blog/?p=23

Cia, arduino e. (2014). Controlador motor paso a paso. Retrieved from http://www.arduinoecia.com.br/search?q=28BYJ-48

cmosgr.wordpress. (2015). Sensor detección de luz (GY-302). Retrieved November 21, 2015, from https://cmosgr.wordpress.com/2013/09/02/bh1750fvi-gy-302-light-sensor-module/

Digest, C. (2016). Servo Motor Basics. Retrieved from http://circuitdigest.com/article/servo-motor-basics

D-robotics. (2010). DHT 11 Humidity & Temperature Sensor.

Elecfreaks. (2015). PIR Motion Sensor Module. Retrieved from http://www.elecfreaks.com/wiki/index.php?title=PIR_Motion_Sensor_Module:DYP-ME003

Garage, E. (2012). ULN2003. Retrieved from http://www.engineersgarage.com/electronic-components/uln2003-datasheet

García, V. (2014). Electrónica práctica aplicada.

GONZÁLEZ, S. S. (n.d.). Sensor CNY70. Retrieved from http://www.info-ab.uclm.es/labelec/solar/otros/infrarrojos/sensor_cny70.htm

k-electronica. (2015). Sensor de lluvia YL-83. Retrieved November 21, 2015, from http://k-electronica.es/complementos/227-sensor-de-lluvia-yl-83-compatible-arduino-en-tenerife-canarias-la-laguna-8436545519608.html

Lara, E. (2015). Sensor de humedad atmosférica DHT11. Retrieved from http://hetpro-

store.com/TUTORIALES/sensor-dht11/

Leandro. (n.d.). Sensor Digital de Luz GY30. Retrieved from http://saber.patagoniatec.com/sensor-digital-de-luz-bh1750/

PatagoniaTecnology. (2013). Sensor Optico Infrarrojo CNY70. Retrieved from http://saber.patagoniatecnology.com/sensor-optico-infrarrojo-cny70/

Pinotolosa. (2015). sensor de luz con cny70. Retrieved November 21, 2015, from http://pinotolosa.net/taller/sensor_luz/cny70/cny70.html

Sensor Ultrasónico HC-SR04. (n.d.). Retrieved November 21, 2015, from http://www.dirux.com/productos/electronica/ultrasonico.php

Travel, P. (2015). Muñoz,Mota,Aldana: motor paso a paso. Retrieved November 21, 2015, from http://mumoaldigitales1.blogspot.com.co/2010/12/motor-paso-paso.html

Valenzuela, R. M. (2006). Motores Paso a Paso. Retrieved November 22, 2015, from http://www.monografias.com/trabajos37/motores/motores2.shtml

Vishay. (2012). LOCKING FILTER TEST CONDITION SYMBOL VALUE SYMBOL. Vishay.

Antony García, K. N. (2014). Módulo YL-83. Retrieved from http://panamahitek.com/modulo-yl-83-un-detector-de-lluvia/

ArduinoLab. (2012). Controllando un servo con Arduino y un potenciómetro. Retrieved from https://arduinolab.wordpress.com/2012/05/07/controllando-un-servo-con-arduino-y-un-potenciometro-10/

Barbus, E. (2014). sensor ultrasonidos HC-SR04. Retrieved from http://elcajondeardu.blogspot.com.co/2014/03/tutorial-sensor-ultrasonidos-hc-sr04.html

Bolivia, R. y D. en. (2015). Modulo sensor de movimiento (PIR). Retrieved from http://www.systems-technology.net/blog/?p=23

Cia, arduino e. (2014). Controlador motor paso a paso. Retrieved from http://www.arduinoecia.com.br/search?q=28BYJ-48

cmosgr.wordpress. (2015). Sensor detección de luz (GY-302). Retrieved November 21, 2015, from https://cmosgr.wordpress.com/2013/09/02/bh1750fvi-gy-302-light-sensor-module/

Digest, C. (2016). Servo Motor Basics. Retrieved from http://circuitdigest.com/article/servo-motor-basics

D-robotics. (2010). DHT 11 Humidity & Temperature Sensor.

Elecfreaks. (2015). PIR Motion Sensor Module. Retrieved from http://www.elecfreaks.com/wiki/index.php?title=PIR_Motion_Sensor_Module:DYP-ME003

Garage, E. (2012). ULN2003. Retrieved from http://www.engineersgarage.com/electronic-components/uln2003-datasheet

García, V. (2014). Electrónica práctica aplicada.

GONZÁLEZ, S. S. (n.d.). Sensor CNY70. Retrieved from http://www.info-ab.uclm.es/labelec/solar/otros/infrarrojos/sensor_cny70.htm

k-electronica. (2015). Sensor de lluvia YL-83. Retrieved November 21, 2015, from http://k-electronica.es/complementos/227-sensor-de-lluvia-yl-83-compatible-arduino-en-tenerife-canarias-la-laguna-8436545519608.html

Lara, E. (2015). Sensor de humedad atmosférica DHT11. Retrieved from http://hetpro-store.com/TUTORIALES/sensor-dht11/

Leandro. (n.d.). Sensor Digital de Luz GY30. Retrieved from http://saber.patagoniatec.com/sensor-digital-de-luz-bh1750/

PatagoniaTecnology. (2013). Sensor Optico Infrarrojo CNY70. Retrieved from http://saber.patagoniatecnology.com/sensor-optico-infrarrojo-cny70/

Pinotolosa. (2015). sensor de luz con cny70. Retrieved November 21, 2015, from http://pinotolosa.net/taller/sensor_luz/cny70/cny70.html

Sensor Ultrasónico HC-SR04. (n.d.). Retrieved November 21, 2015, from http://www.dirux.com/productos/electronica/ultrasonico.php

Travel, P. (2015). Muñoz,Mota,Aldana: motor paso a paso. Retrieved November 21, 2015, from http://mumoaldigitales1.blogspot.com.co/2010/12/motor-paso-paso.html

Valenzuela, R. M. (2006). Motores Paso a Paso. Retrieved November 22, 2015, from http://www.monografias.com/trabajos37/motores/motores2.shtml

Vishay. (2012). LOCKING FILTER TEST CONDITION SYMBOL VALUE SYMBOL. Vishay.

11. ANEXOS

Conclusiones

Con las condiciones que nos brindó la maleta, en el diseño de su interior obtuvimos una distribución practica de los elementos para el momento de su uso, así nuestro modulo contara con una seguridad en su interior y fácil transporte, con esto los estudiantes tendrían la comodidad de interactuar de manera didáctica con este módulo.

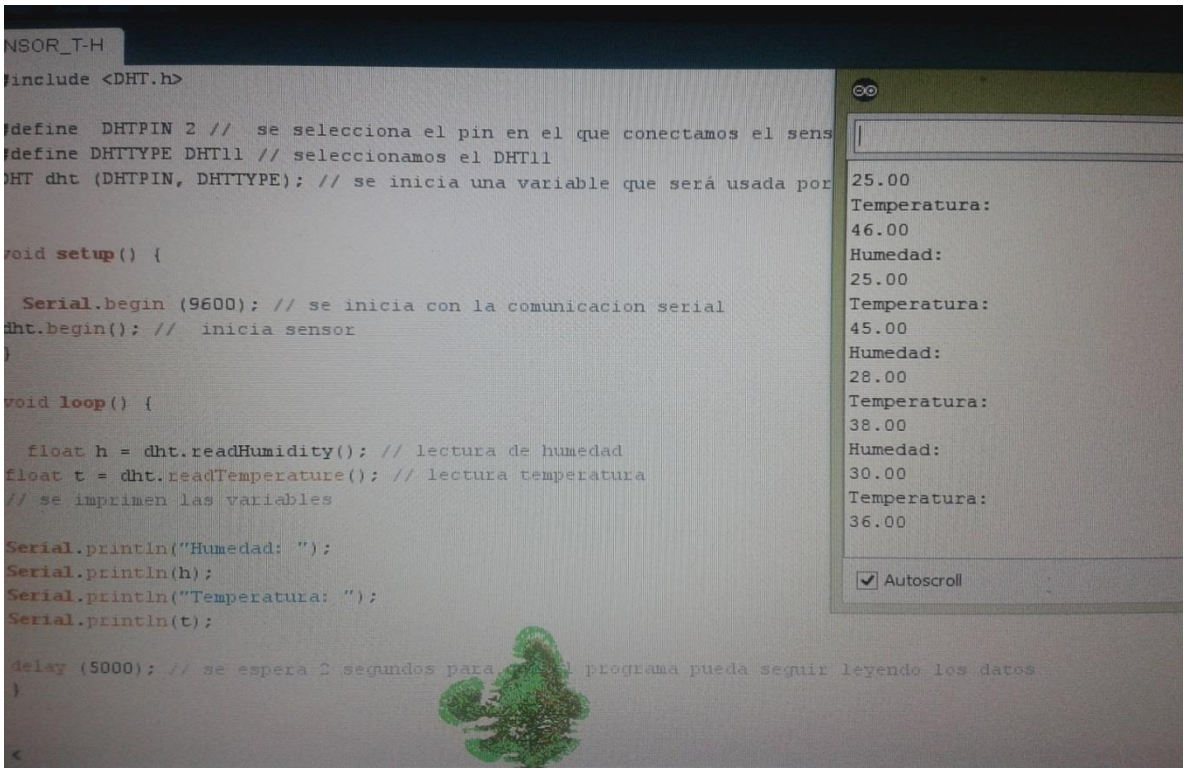
En la identificación de nuestros dispositivos se logró adquirir un conocimiento más profundo en ellos, nuestra inclinación por hacer de manera didáctica este proyecto donde buscamos incluir sensores cuyas variables son de nuestro común vivir y que nos acercan a un mejor aprendizaje.

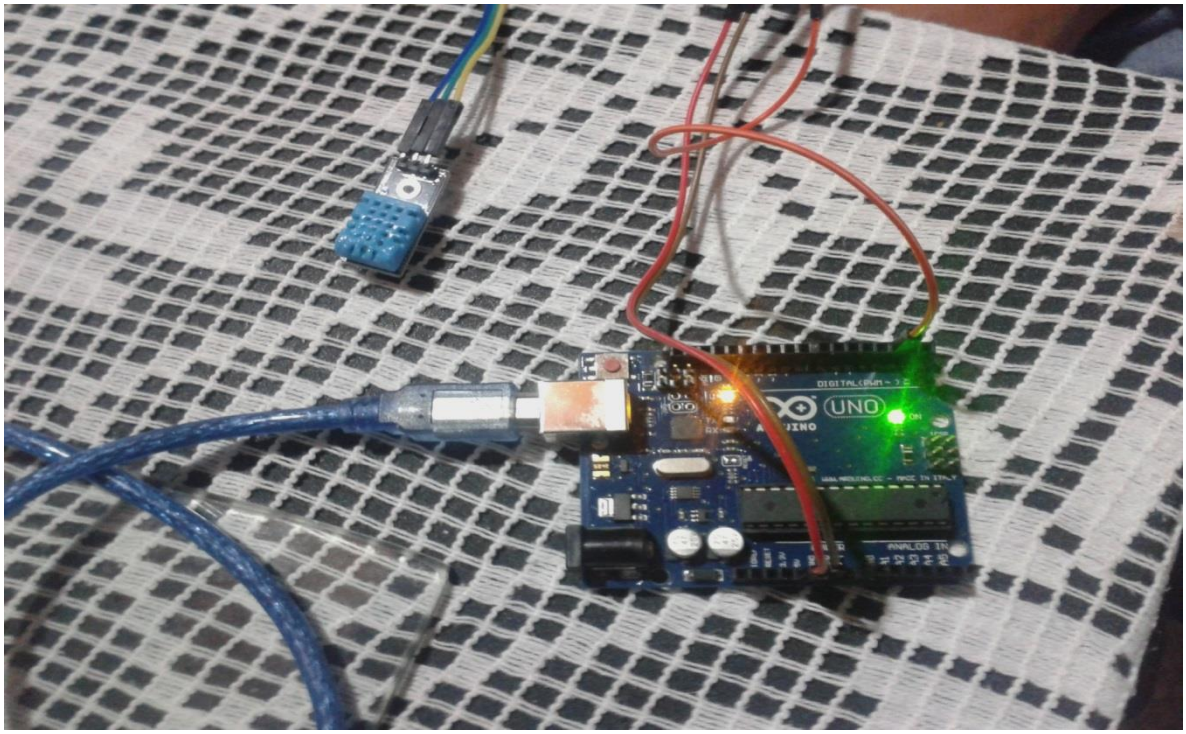
Se hizo los diseños eléctricos para su fácil comprensión de todos aquellos interesados en este módulo, con el fin de facilitar su funcionamiento, montaje e interpretación al momento de interactuar en este.

Se plasmaron los algoritmos de cada módulo, para que con ellos se tomen como base y los estudiantes puedan avanzar con sus propios proyectos.

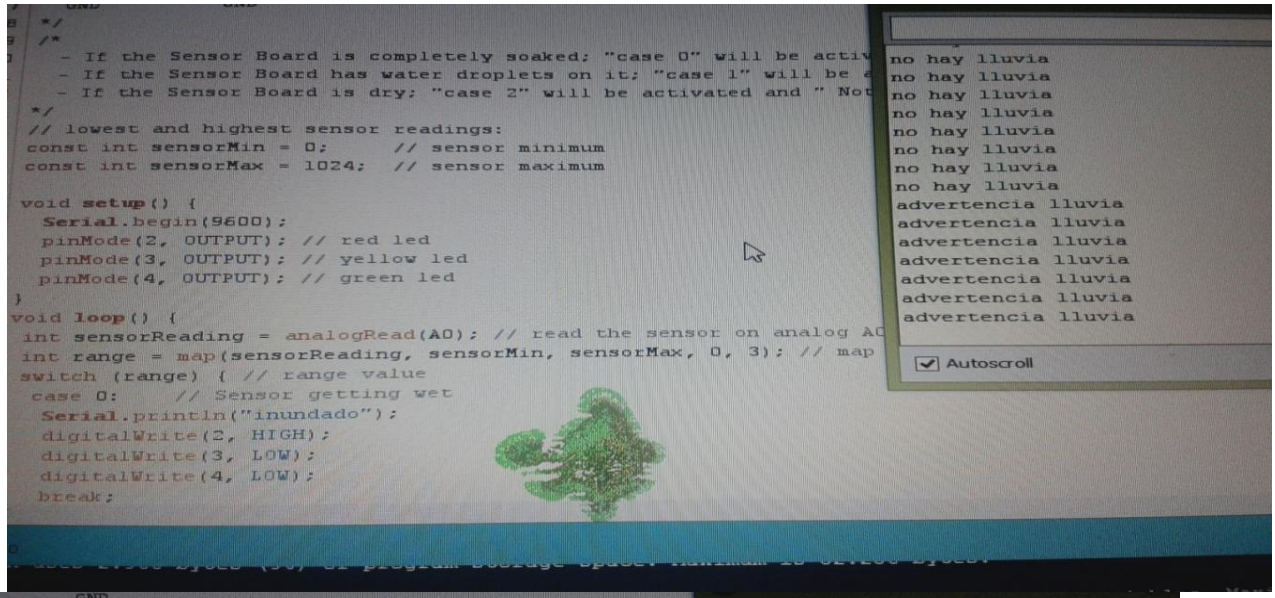
Durante la ejecución de este proyecto se realizaron pruebas de su programación, conexiones eléctricas y montajes físicos con el fin de garantizar su funcionamiento, logrando resultados satisfactorios con los cuales quedamos convencidos de que estamos de una gran herramienta como apoyo y contribución en nuestro campo

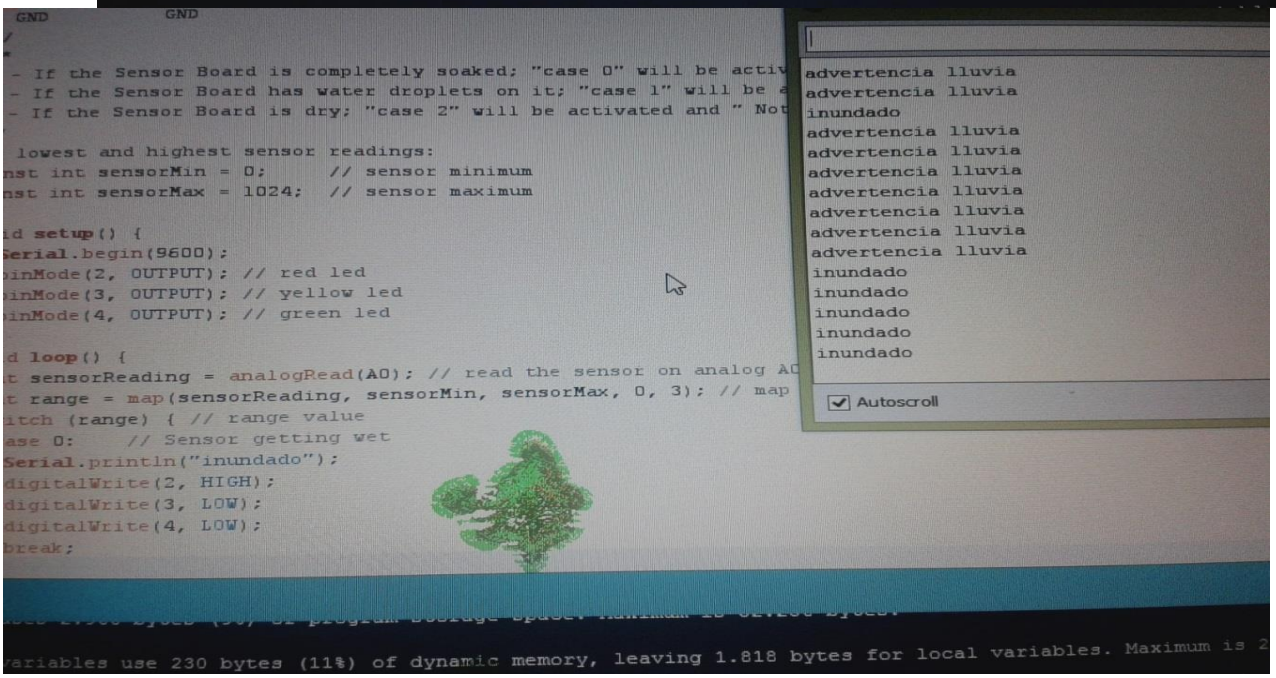
Sensor temperatura y humedad

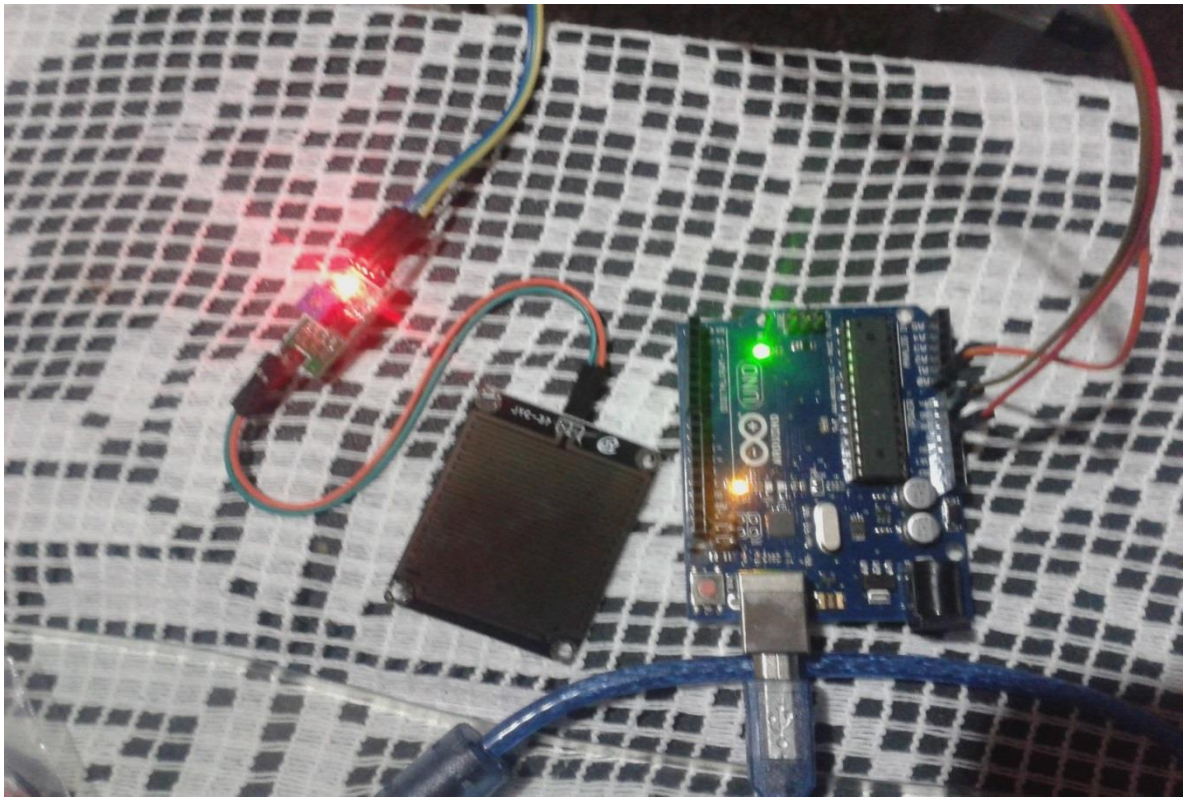




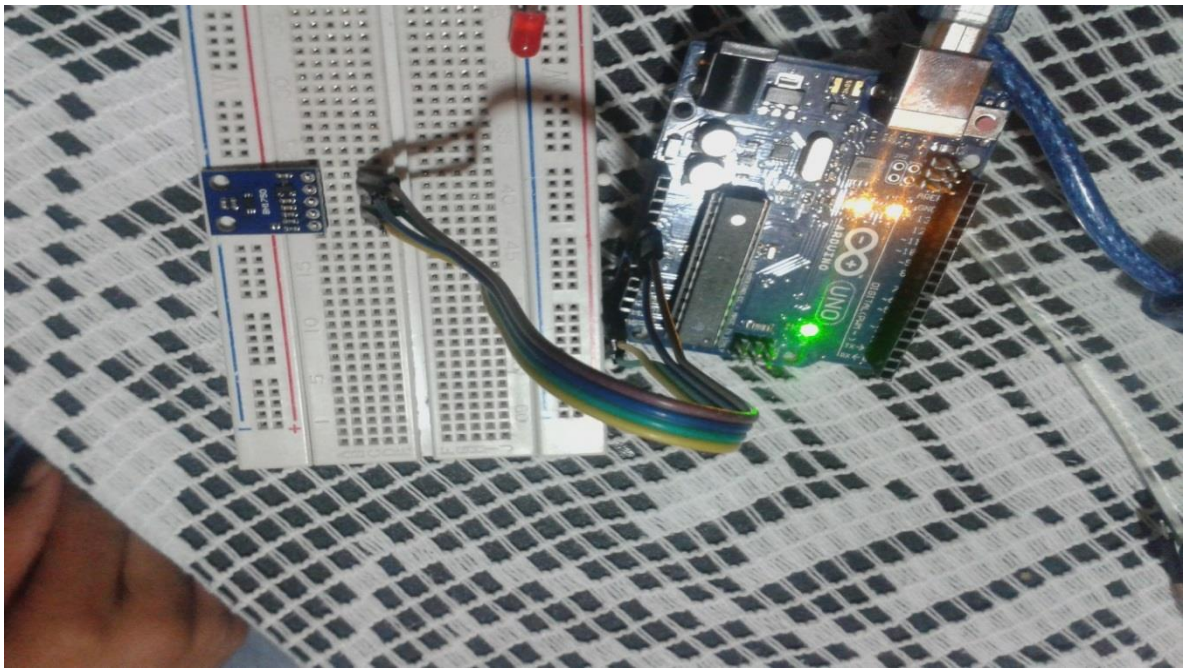
Sensor lluvia


*/
/*
- If the Sensor Board is completely soaked; "case 0" will be activ
- If the Sensor Board has water droplets on it; "case 1" will be
- If the Sensor Board is dry; "case 2" will be activated and " Not
*/
// lowest and highest sensor readings:
const int sensorMin = 0; // sensor minimum
const int sensorMax = 1024; // sensor maximum
void setup() {
Serial.begin(9600);
pinMode(2, OUTPUT); // red led
pinMode(3, OUTPUT); // yellow led
pinMode(4, OUTPUT); // green led
}
void loop() {
int sensorReading = analogRead(A0); // read the sensor on analog A
int range = map(sensorReading, sensorMin, sensorMax, 0, 3); // map
switch (range) { // range value
case 0: // Sensor getting wet
Serial.println("inundado");
digitalWrite(2, HIGH);
digitalWrite(3, LOW);
digitalWrite(4, LOW);
break;
no hay lluvia
no hay lluvia
no hay lluvia
no hay lluvia
no hay lluvia
no hay lluvia
no hay lluvia
no hay lluvia
advertencia lluvia
advertencia lluvia
advertencia lluvia
advertencia lluvia
advertencia lluvia
advertencia lluvia
advertencia lluvia
Autoscroll

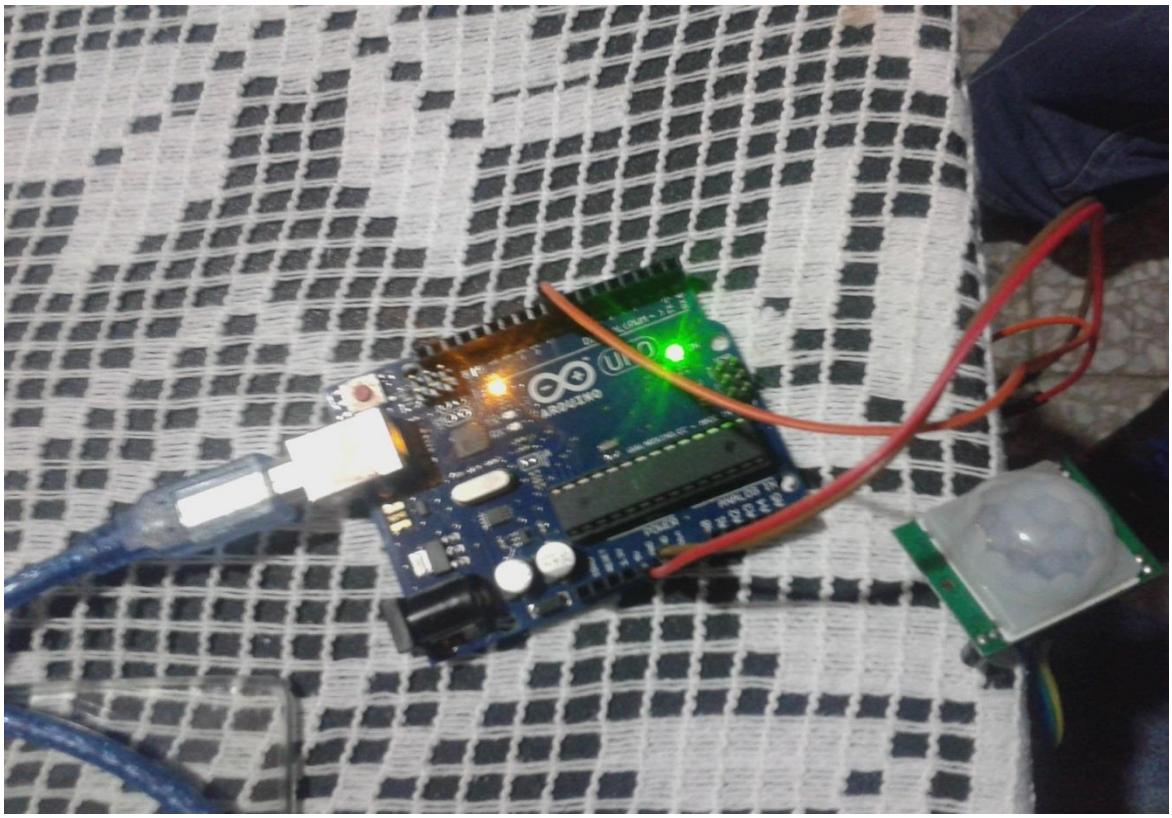

GND GND
- If the Sensor Board is completely soaked; "case 0" will be activ
- If the Sensor Board has water droplets on it; "case 1" will be
- If the Sensor Board is dry; "case 2" will be activated and " Not
lowest and highest sensor readings:
nst int sensorMin = 0; // sensor minimum
nst int sensorMax = 1024; // sensor maximum
id setup() {
erial.begin(9600);
inMode(2, OUTPUT); // red led
inMode(3, OUTPUT); // yellow led
inMode(4, OUTPUT); // green led
d loop() {
t sensorReading = analogRead(A0); // read the sensor on analog A
t range = map(sensorReading, sensorMin, sensorMax, 0, 3); // map
itch (range) { // range value
ase 0: // Sensor getting wet
Serial.println("inundado");
digitalWrite(2, HIGH);
digitalWrite(3, LOW);
digitalWrite(4, LOW);
break;
advertencia lluvia
advertencia lluvia
inundado
advertencia lluvia
advertencia lluvia
advertencia lluvia
advertencia lluvia
advertencia lluvia
advertencia lluvia
advertencia lluvia
inundado
inundado
inundado
inundado
inundado
Autoscroll
ariables use 230 bytes (11%) of dynamic memory, leaving 1.818 bytes for local variables. Maximum is 2

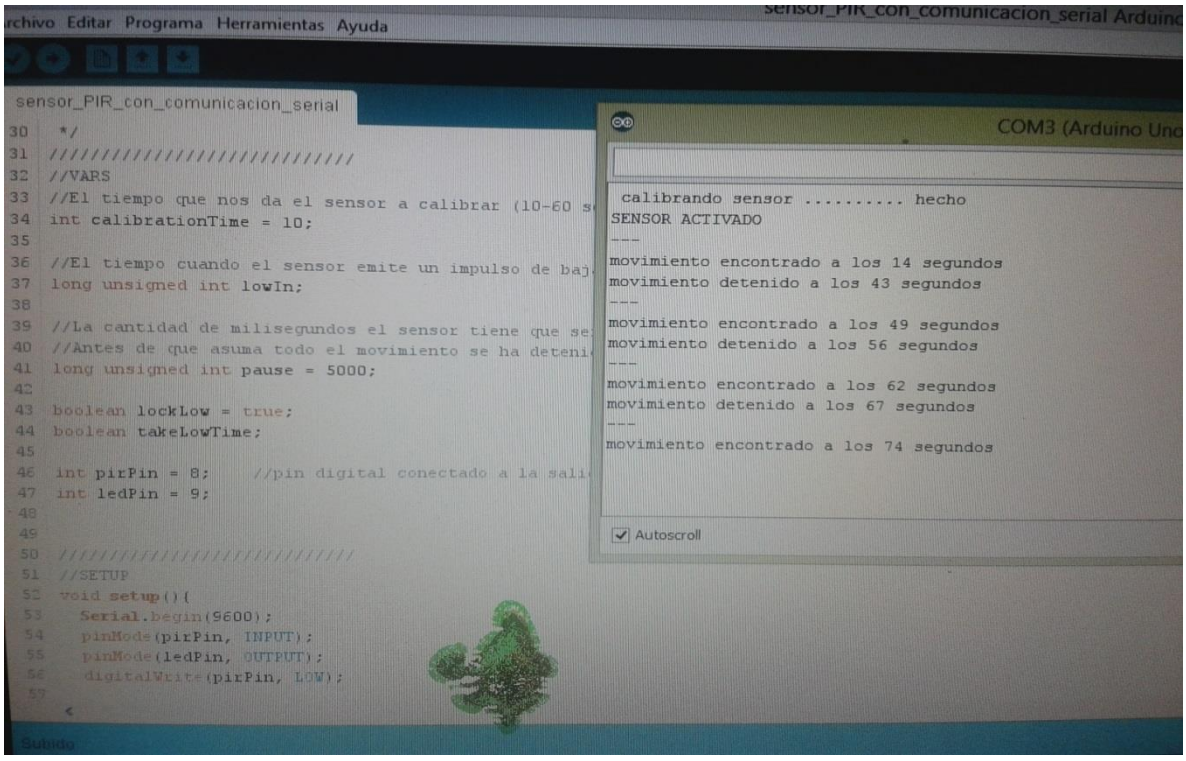


Sensor de iluminación

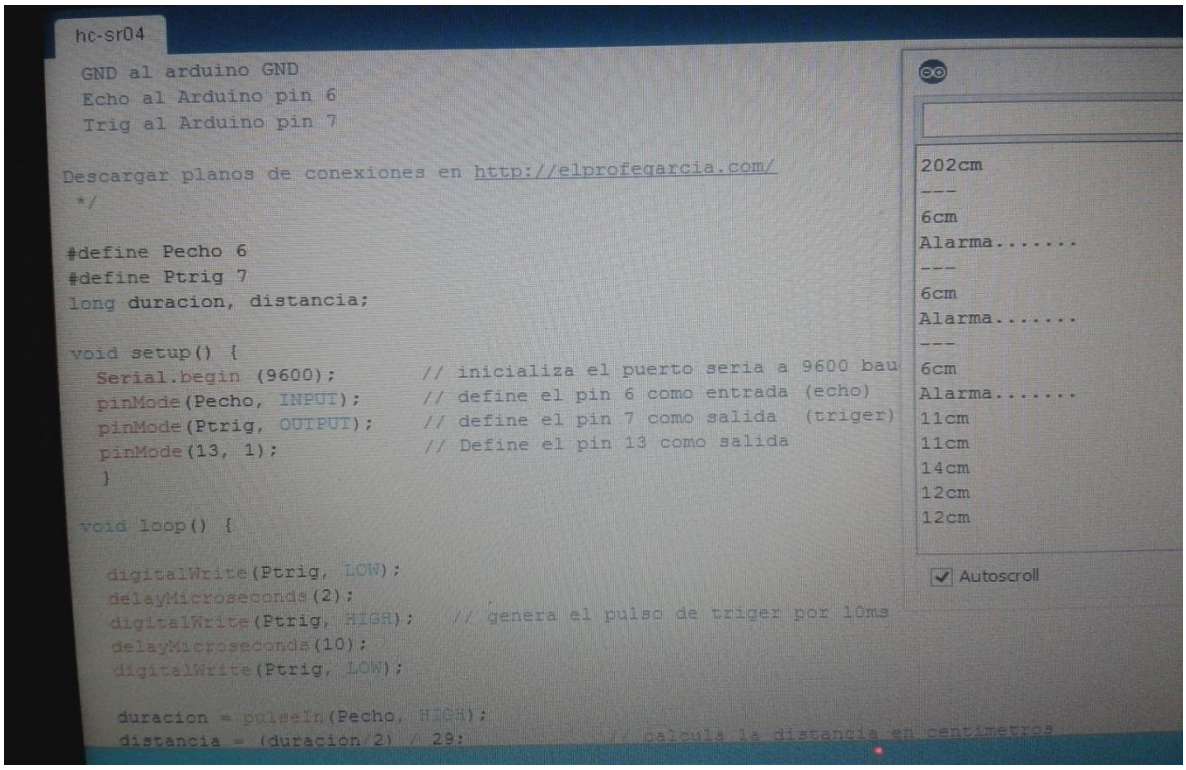


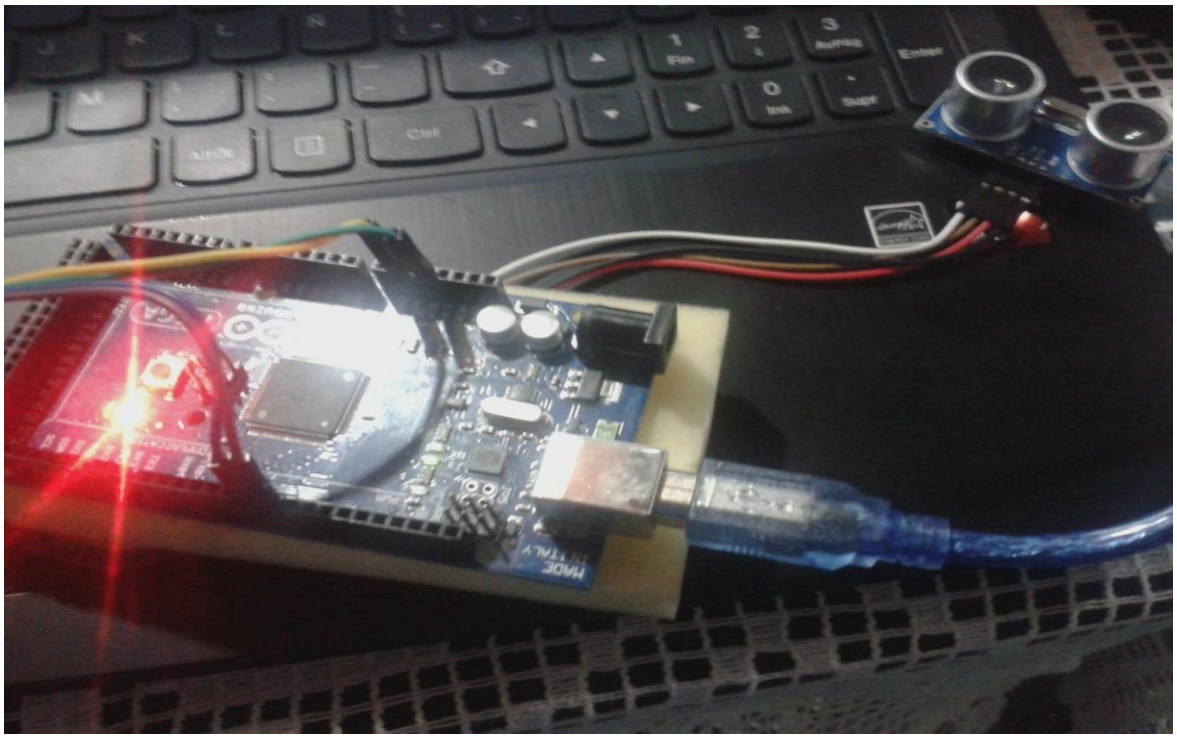
Sensor movimiento



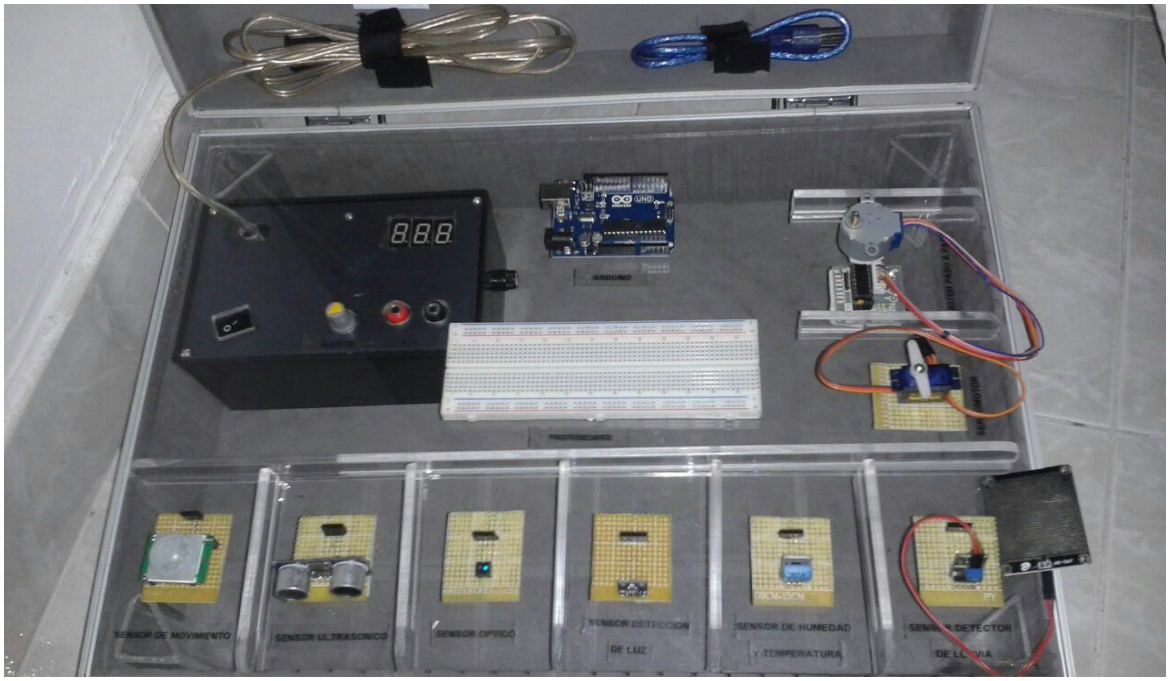


Sensor ultrasonido





Modulo terminado


SENSOR DE MOVIMIENTO
SENSOR ULTRASONICO
SENSOR OPTICO
SENSOR DETECCION
DE LUZ
SENSOR DE HUMEDAD
Y TEMPERATURA