



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN



MICROPROCESADORES

PRIMAVERA 2015

EQUIPO 6

ALUMNOS:

MIGUEL HERNÁNDEZ RAMOS
EDGAR ESPINOSA ORDOÑES

PROYECTO FINAL

“ALARMA LUMINOSA”

PROFESORA:

ELSA CHAVIRA MARTINEZ

INDICE

Introducción general.....	3
PRÁCTICA I: Encendido de un LED usando Arduino.....	4
PRÁCTICA 2: Encendido de una matriz de LED usando Arduino.....	9
PRÁCTICA 3: Uso de sensor ultrasónico con Arduino.....	12
PRÁCTICA 4: Uso de un Servomotor con Arduino.....	15
PROYECTO FINAL: Alarma Luminosa.....	19
ACRÓNIMOS.....	26

INTRODUCCIÓN GENERAL

El desarrollo de software se ha convertido en una tarea que se puede abordar desde diferentes temáticas, y cada vez hay más personas que se involucran en el mundo de las tecnologías de la información. Es una lucha constante por desarrollar sistemas que faciliten a los humanos las tareas cotidianas, reduzcan tiempo y por lo tanto recursos. Y aunque el desarrollo de sistemas cada vez abarca más ámbitos es necesario orientarlos hacia problemas que causen impacto en la sociedad.

Este documento es una colección de prácticas que se realizaron a lo largo de un curso universitario con el propósito de aterrizar todos los conocimientos en un proyecto que pudiera ser aplicado a la solución de problemas reales.

Al final de este documento encontrará “Alarma Luminosa” un proyecto orientado a aplicaciones específicas, por ejemplo a ofrecer una solución para atender sus tareas de personas con problemas auditivos.



PRÁCTICA I: Encendido de un LED usando Arduino

Edgar Espinosa Ordoñez, Miguel Hernández Ramos, Elsa Chavira Martínez

Facultad de Ciencias de la Computación.

14 Sur y San Claudio, Edif. 104C, Ciudad Universitaria, Col. San Manuel, C.P. 72570, Puebla, Pue., México.
(edgar.espinosa.fcc@gmail.com, miguellhdez@hotmail.com), elsachavira56@hotmail.com

Abstract— The following report of practice opens the theoretical framework of what will be the development of multiple jobs throughout the document. We believe as the Arduino development board more complete until these moments, therefore we are going to define it and give them to know every detail of this plate. However to test its effectiveness is the ignition of an LED as proof of its workings. This will test the simplicity of development as an advantage that leaves a side to other development boards, but as everything: it is a question of choosing the best hardware in accordance to the project that you want to do.

Keywords— Arduino, circuits, CPU, LED, microcontroller, microprocessor, project, USB.

Resumen—El siguiente reporte de práctica abre el marco teórico de lo que va ser el desarrollo de varios trabajos a lo largo del documento. Consideramos Arduino como la placa de desarrollo más completa hasta estos momentos, por lo tanto vamos a definirla y darles a conocer cada detalle de esta placa. Sin embargo para probar su eficacia se hace el encendido de un LED como prueba de su funcionamiento. Esto probará la sencillez de desarrollo como una ventaja que deja a un lado a otras placas de desarrollo, pero como todo: es cuestión de elegir el mejor hardware de acuerdo al proyecto que se quiera realizar.

Índices— Arduino, circuitos, CPU, LED, microcontrolador, microprocesadores, proyecto, USB.

NOMENCLATURA

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 multidisciplinarios [1].

LED. Light Emitter Diode.

USB. Universal Serial Bus.

Microcontrolador. Un microcontrolador es un circuito integrado o chip que incluye en su interior

las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de E/S, es decir, se trata de un computador completo en un solo circuito integrado.

CPU. Unidad Central de Procesamiento..

INTRODUCCIÓN

EN esta entrega hacemos uso de la placa de desarrollo “Arduino” para hacer el encendido de un LED. El propósito es hacer una introducción a Arduino para ir mostrando lo eficiente que es desarrollar en esta placa. Al final de esta primera entrega del documento usted podrá conocer por completo la placa de desarrollo Arduino y podrá hacer esta prueba. Con todo lo anterior será capaz de programar en el lenguaje de Arduino ya que se le mostrará la sintaxis básica del lenguaje.

PREPARACIÓN DEL TRABAJO (MARCO TEÓRICO)

En primer lugar vamos a mostrar las partes que componen Arduino, esto con la intención de hacer de su conocimiento el funcionamiento tanto interno como externo. La siguiente figura muestra como es Arduino físicamente [2].

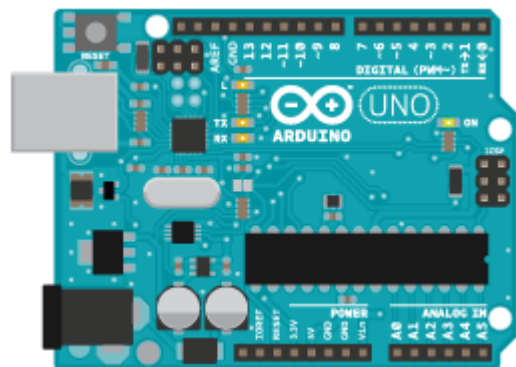


Fig. 1. Placa de desarrollo Arduino UNO

Arduino se distribuye y categoriza en 4 diferentes productos.

Placas:

Arduino Uno

Arduino Leonardo

Arduino Due

Arduino Yún

Arduino Tre (En Desarrollo)

Arduino Zero (En Desarrollo)

Arduino Micro

Arduino Esplora

Arduino Mega ADK

Arduino Ethernet

Arduino Mega 2560

Arduino Robot

Arduino Mini

Arduino Nano

LilyPad Arduino Simple

LilyPad Arduino SimpleSnap

LilyPad Arduino

LilyPad Arduino USB

Arduino Pro Mini

Arduino Fio

Arduino Pro

Escudos (shields):

Arduino Proto Shield

Arduino GSM Shield

Arduino Ethernet Shield

Arduino WiFi Shield

Arduino Wireless SD Shield

Arduino USB Host Shield

Arduino Motor Shield

Arduino Wireless Proto Shield

Arduino Proto Shield

Kits:

The Arduino Starter Kit

Arduino Materia 101

Accesorios:

TFT LCD Screen

USB/Serial Light Adapter

Arduino ISP

Mini USB/Serial Adapter [2].

A la fecha Arduino ha sido usado para el desarrollo de miles de aplicaciones que ya están siendo usadas. Estos son algunos ejemplos:

- Xoscillo: Osciloscopio de código abierto.
- Equipo científico para investigaciones.
- Arduinome: Un dispositivo controlador MIDI.

- OBDuino: un económetro que usa una interfaz de diagnóstico a bordo que se halla en los automóviles modernos.
- Humane Reader: dispositivo electrónico de bajo coste con salida de señal de TV que puede manejar una biblioteca de 5000 títulos en una tarjeta microSD.¹⁷
- The Humane PC: equipo que usa un módulo Arduino para emular un computador personal, con un monitor de televisión y un teclado para computadora.¹⁸
- Ardupilot: software y hardware de aeronaves no tripuladas.
- ArduinoPhone: un teléfono móvil construido sobre un módulo Arduino.¹⁹
- Impresoras 3D.

Funciones básicas y operadores.

Arduino está basado en C y soporta todas las funciones del estándar C y algunas de C++.

Sintaxis básica

Delimitadores: ;, { }

Comentarios: //, /* */

Cabeceras: #define, #include

Operadores aritméticos: +, -, *, /, %

Asignación: =

Operadores de comparación: ==, !=, <, >, <=, >=

Operadores Booleanos: &&, ||, !

Operadores de acceso a punteros: *, &

Operadores de bits: &, |, ^, ~, <<, >>

Operadores compuestos:

Incremento y decremento de variables: ++, --

Asignación y operación: +=, -=, *=, /=, &=, |=

Estructuras de control[editar]

Condicionales: if, if...else, switch case

Bucles: for, while, do... while

Bifurcaciones y saltos: break, continue, return, goto

Variables

En cuanto al tratamiento de las variables también comparte un gran parecido con el lenguaje C.

Constantes

HIGH/LOW: representan los niveles alto y bajo de las señales de entrada y salida. Los niveles altos son aquellos de 3 voltios o más.

INPUT/OUTPUT: entrada o salida.

false (falso): Señal que representa al cero lógico.

A diferencia de las señales HIGH/LOW, su nombre se escribe en letra minúscula. [2].

true (verdadero): Señal cuya definición es más amplia que la de false. Cualquier número entero diferente de cero es "verdadero", según el álgebra de Boole, como en el caso de -200, -1 o 1. Si es cero, es "falso".

Ejemplo sencillo de programación en Arduino

```
#define LED_PIN 13
void setup () {
  // Activado del contacto 13 para salida digital
  pinMode (LED_PIN, OUTPUT);
}
// Bucle infinito
void loop () {
  // Encendido del diodo LED enviando una señal
  alta
  digitalWrite (LED_PIN, HIGH);
  // Tiempo de espera de 1 segundo (1000 ms)
  delay (1000);
  // Apagado del diodo LED enviando una señal
  baja.
  digitalWrite (LED_PIN, LOW);
  // Tiempo de espera de 1 segundo
  delay (1000);
}[1].
```

Conexiones

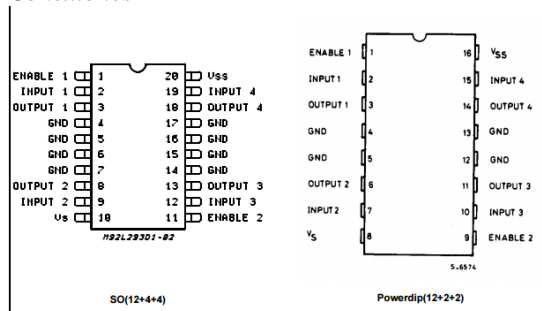


Fig. 2. Esquema de conexiones Arduino

DESARROLLO DEL TRABAJO EXPERIMENTAL

Ahora que ya conocemos Arduino comenzaremos con el objetivo de este trabajo. El material necesario es el siguiente:

Arduino Uno



Fig. 3. Arduino UNO disponible para usar

LED



Fig. 4. LED en sus diferentes colores

Para la conexión del Arduino UNO con el PC usaremos un cable USB de tipo A-B.



Fig. 5. Cable USB de tipo A-B

Instalación software IDE de desarrollo y controladores para conectar Arduino con un PC con Windows 7.

Para poder programar el Arduino UNO usaremos el propio IDE que proporciona Arduino, podremos descargar la última versión desde:
<http://arduino.cc/en/Main/Software>

En el sitio está la versión más actual a su fecha de lectura de este documento. La carpeta que usted descargará contiene tanto el IDE de desarrollo de Arduino como los drivers (controladores) para la conexión USB en Windows.

Tras descargar el fichero y descomprimirlo, en el caso de Linux no será necesario instalar drivers, pero en el caso de Microsoft Windows deberemos instalar los controladores para el emulador de puerto USB a puerto serie. Aunque Arduino se conecta al PC mediante el puerto USB en realidad, internamente, emula un puerto serie, por ello, en Microsoft Windows realizaremos los siguientes pasos para instalar los drivers:

1. Conectaremos Arduino UNO al PC mediante el cable USB de tipo A-B.
2. Microsoft Windows detectará el dispositivo e intentará instalar los drivers.
3. Una vez instalados los drivers ya podremos abrir el IDE de desarrollo.

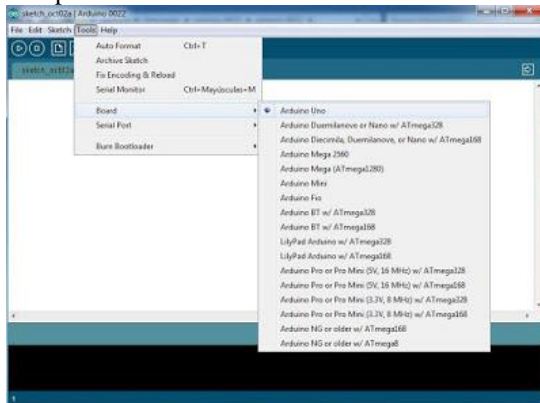


Fig. 6. Entorno de IDE de desarrollo

Una vez adquirido Arduino, conectado al PC y configurado el IDE, vamos a explicar cómo realizar una primera aplicación en Arduino IDE y cómo enviarla al dispositivo y testarla. Como ya hemos comentado encenderemos un LED, si no disponemos de un LED no es problema pues la placa de Arduino UNO incorpora un LED que nos servirá para las pruebas:



Fig. 7. LED de prueba integrado en Arduino UNO

El LED quedará conectado directamente a los conectores del Arduino, lo más profesional es usar una placa prototipo y conectar ahí el LED y, a su vez, conectar la placa prototipo al Arduino con sus cables correspondientes. El LED conectado directamente al Arduino:



Fig. 8. LED conectado

Con una Protoboard quedará de la siguiente forma:

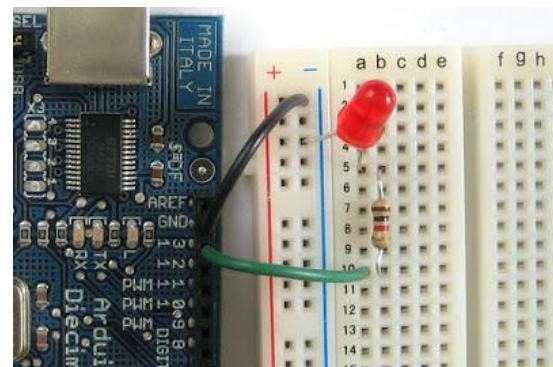


Fig. 9. LED y Protoboard junto a Arduino

En el IDE de Arduino colocamos el siguiente código que corresponde a la serie de instrucciones para esta prueba.

```
void setup(){  
  // Inicializamos el pin 13 como una salida, donde  
  va a estar conectado el LED  
  Serial.begin(9600);  
  pinMode(13, OUTPUT);  
}
```

```
void loop(){  
  digitalWrite(13, HIGH);  
  // Ponemos el LED en estado alto  
  delay(10); // Tiempo de estado alto en  
  milisegundos  
  digitalWrite(13, LOW);  
  // Ponemos el LED en estado bajo  
  delay(10); // Tiempo de estado bajo en msec  
}
```

Para verificar que el código es correcto pulsamos el botón Verify dentro del IDE.

RESULTADOS

Si la compilación del código anterior es correcta ya podemos mandar estas instrucciones a Arduino. La salida es la siguiente.



Fig. 9. LED encendido

Como podemos notar la práctica no presenta problemas mayores. En general la conexión del LED y Arduino queda de la siguiente forma.

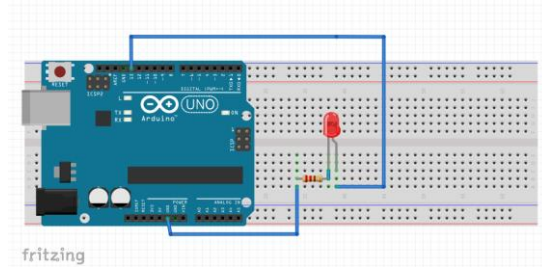


Fig. 10. Conexión general de LED, Arduino y Protoboard

CONCLUSIONES Y DISCUSIÓN DE RESULTADOS

Una de las ventajas de este trabajo como muestra de lo eficiente y fácil de usar que es Arduino es que nos abre la mente a posibles proyectos, donde esta placa puede ser sin duda la mejor opción.

APÉNDICE

Este trabajo puede ser expandido de forma incremental si en lugar de encender solamente un LED encendemos muchos simultáneamente, cada uno con un tiempo de encendido y apagado diferente, por ejemplo podemos simular unas luces navideñas.

AGRADECIMIENTOS

Agradecemos la atención prestada para la revisión de este trabajo a nuestra profesora Dra. Elsa Chavira Martínez y a todos aquellos que se interesaron en nuestros trabajos.

REFERENCIAS

- [1] ARDUINO (2012, Aug.). *Arduino Board - Serial Interface*. En: <http://www.arduino.cc/en/Main/ArduinoBoardSerial>
- [2] ARDUINO (2012, Aug.). *Arduino y todo lo que debes saber*. En: <http://es.wikipedia.org/wiki/Arduino>

PRÁCTICA 2: Encendido de una matriz de LED usando Arduino

Edgar Espinosa Ordoñez, Miguel Hernández Ramos, Elsa Chavira Martínez

Facultad de Ciencias de la Computación.

14 Sur y San Claudio, Edif. 104C, Ciudad Universitaria, Col. San Manuel, C.P. 72570, Puebla, Pue., México.

(edgar.espinosa.fcc@gmail.com, miguellhdez@hotmail.com), elsachavira56@hotmail.com

Abstract— Once you identify everything you need to work with Arduino and already know the programming language we are going to turn not only an LED, now will be an array of LEDs 8x8.

Keywords— Arduino, LED, microcontroller, microprocessor, project, USB.

Resumen— Una vez que ya identificamos todo lo necesario para trabajar con Arduino y ya conocemos el lenguaje de programación vamos a encender no solo un LED, ahora será una matriz de LEDs de 8x8.

Índices— Arduino, circuitos, LED, microcontrolador, microprocesadores, proyecto, USB.

NOMENCLATURA

Arduino. En 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 multidisciplinarios [1].

CPU. Unidad Central de Procesamiento.

LED. Light Emitter Diode.

USB. Universal Serial Bus.

Matriz se denomina matriz de n filas y m columnas a un conjunto de $n \times m$ elementos de X , dispuestos en un arreglo rectangular de n filas y m columnas [2].

Microcontrolador. Un microcontrolador es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de E/S, es decir, se trata de un computador completo en un solo circuito integrado.

RGB. Combinación de colores Red-Green-Blue para formar los pantones de colores.

INTRODUCCIÓN

EN esta entrega hacemos uso de la placa de desarrollo “Arduino” para hacer el encendido de una matriz LED. En este trabajo podremos seguir viendo las ventajas que nos ofrece Arduino.

PREPARACIÓN DEL TRABAJO (MARCO TEÓRICO)

Parece que los LEDs se fabrican en todos los tamaños y formatos imaginables, y este componente que hoy presentamos, hace gala de esa creatividad. Las matrices de LEDs (o LED arrays) son, como su nombre indica, una matriz de diodos LED normales y corrientes que se comercializa en multitud de formatos y colores. Desde las de un solo color, a las que tienen varios colores posibles, e incluso las hay de una matriz RGB [3].

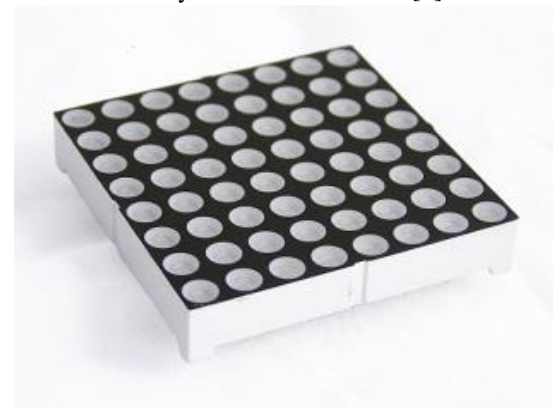


Fig. 1. Matriz de LED 8X8

La figura uno muestra como es externamente una matriz, cada circulo corresponde a un LED sin

embargo es necesario conocer su estructura interna para saber cómo funciona.

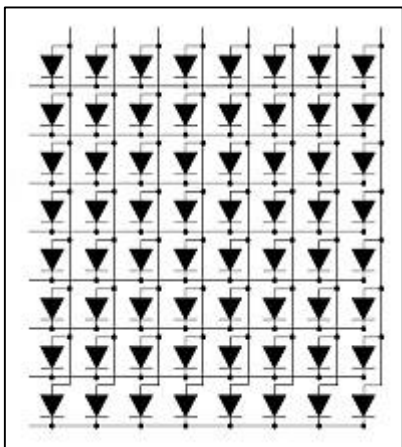


Fig. 2. Estructura interna de la Matriz

Conocida la matriz vamos a ver la forma en que funciona cada LED contenido en la matriz.

Funcionamiento del LED diodo.

La estructura del chip de los diodos LED, al contrario de lo que ocurre con los diodos comunes, no emplea cristales de silicio como elemento semiconductor, sino una combinación de otros tipos de materiales, igualmente semiconductores, pero que poseen la propiedad de emitir fotones de luz de diferentes colores cuando lo recorre una corriente eléctrica.

Un diodo LED emisor de luz roja, por ejemplo, emplea un chip compuesto por arseniuro de galio y aluminio (GaAlAs), mientras que para emitir luz azul utiliza un chip de nitruro de galio (GaN). Todas las combinaciones empleadas en la fabricación del chip de un diodo LED, poseen también dos polaridades o regiones diferentes: una negativa "N" correspondiente al cátodo y otra positiva "P" correspondiente al ánodo, al igual que ocurre con los diodos comunes de silicio (Si).

Para crear un diodo LED se unen también dos regiones "N" y "P", como si de un diodo común se tratara. En el punto de unión o juntura de esas dos regiones se forma, igualmente, una barrera de potencial, cuya función es impedir el paso de los electrones desde la región negativa "N" hacia la positiva "P" cuando no se encuentran debidamente polarizados y los electrones no poseen la suficiente energía para poder atravesarla.

Cuando aplicamos a los extremos del LED una tensión o voltaje que permita polarizarlo

directamente, los electrones provenientes de la fuente de suministro de corriente directa (C.D.) comienzan a fluir a través del diodo. Bajo esas condiciones, cada vez que un electrón en exceso con carga negativa (-) presente en la región "N" adquiere la suficiente energía como para poder vencer la resistencia que le ofrece la barrera de potencial, la atraviesa y se combina con un hueco positivo en exceso en la región "P". En el mismo instante que ocurre esa combinación, la energía en exceso que adquirió dicho electrón para poder atravesar la barrera de potencial, se transforma en energía electromagnética, que libera, en ese preciso momento, en forma de fotón de luz [4].

El color de la luz emitida por los fotones de un LED en particular se corresponde con una determinada frecuencia del espectro electromagnético visible al ojo humano. Sin embargo, existen LEDs cuya luz no es visible, como ocurre con los materiales que emiten fotones de rayos infrarrojos "IR" y ultravioletas "UV". En cualquier caso, la luz y color de un determinado LED depende de la composición química de los materiales semiconductores utilizados en la fabricación del chip [4].

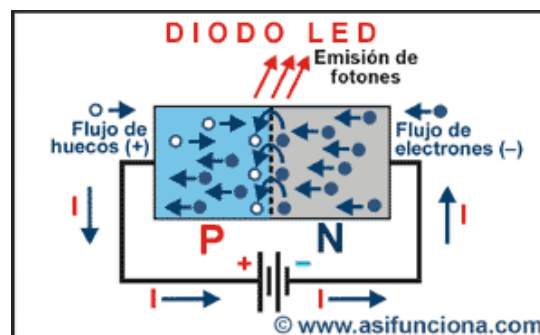


Fig. 3. Funcionamiento de un LED

DESARROLLO DEL TRABAJO EXPERIMENTAL

Lo primero que hacemos es conectar Arduino a nuestra computadora mediante el cable USB.

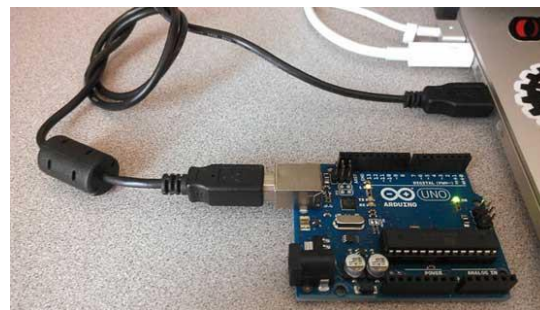


Fig. 4. Arduino conectado a la computadora

El esquema general de conexión para la matriz es el que se muestra en la figura 5.

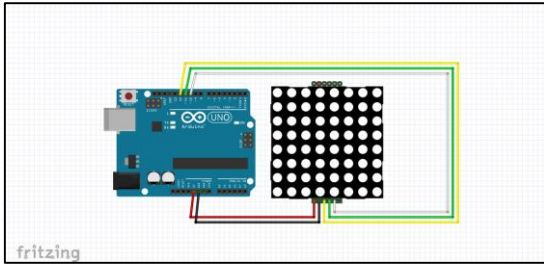


Fig. 5. Esquema general de conexión.

El código correspondiente a este trabajo es el siguiente:

```
#include <LedControl.h> //siempre incluimos la libreria de control

const byte DIN   = 12; //Lo conectamos en din
const byte CS    = 11; //Lo conectamos a Load (cs)
const byte CLK   = 10; //Lo conectamos a CLK
const byte QTD_DISP = 1; //El numero de matriz con
                        //controlador M72XX

LedControl ledMatrix = LedControl(DIN, CLK, CS,
                                   QTD_DISP);

void setup() {
  // El MAX72XX está en modo de ahorro de energía en el
  // arranque, tenemos que hacer que despierte
  ledMatrix.shutdown(0, false); //modo 'shutdown' no
  // display '0' y FALSE
  // Establecer el brillo a un valor medio
  ledMatrix.setIntensity(0, 5); //intensidad del display '0' y
  // 5 (0~16)
  // y borramos la pantalla
  ledMatrix.clearDisplay(0); //borrar pantalla '0'
}

void loop(){
  // imprimir un caracter en la pantalla led
  ledMatrix.clearDisplay(0); //borrar pantalla '0'
  ledMatrix.printChar(0, 0, '-');
  delay(500);
}
```

RESULTADOS

Ante la compilación del código podemos ver la siguiente salida:

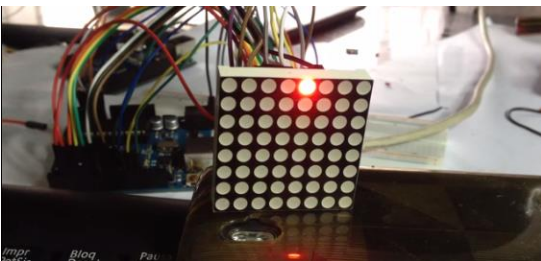


Fig. 6. Matriz de LEDs encendida

CONCLUSIONES Y DISCUSIÓN DE RESULTADOS

Conforme vayamos adentrándonos en el desarrollo en Arduino vamos a ir generando resultados más sorprendentes. El objetivo no deja de ser mostrar la importancia que tiene Arduino para el desarrollo de proyectos, así como ir mencionando a detalle los pasos para cada trabajo.

AGRADECIMIENTOS

Agradecemos la atención prestada para esta segunda parte del documento. En especial agradecemos a nuestros padres por los recursos que nos proporcionan para continuar estudiando y querer hacer de nosotros mejores personas.

REFERENCIAS

- [1] *ARDUINO (2012, Aug.)*. Arduino Board - Serial Interface. En: <http://www.arduino.cc/en/Main/ArduinoBoardSerial>
- [2] Carl D. Meyer's (2000): "Matrix analysis and applied linear algebra", Philadelphia SIAM, 461, 468-470
- [3] *Matriz LED. Uso de las matrices en LED (2014, Nov.)*. En: <http://www.prometec.net/matriz-led-8x8/>
- [4] E. García (2013, Aug). ASÍ FUNCIONAN LOS DIODOS LEDs. En http://www.asifunciona.com/fisica/af_leds/af_leds_3.htm

PRÁCTICA 3: Uso de sensor ultrasónico con Arduino

Edgar Espinosa Ordoñez, Miguel Hernández Ramos, Elsa Chavira Martínez

Facultad de Ciencias de la Computación.

14 Sur y San Claudio, Edif. 104C, Ciudad Universitaria, Col. San Manuel, C.P. 72570, Puebla, Pue., México.

(edgar.espinosa.fcc@gmail.com, miguellhdez@hotmail.com), elsachavira56@hotmail.com

Abstract— In this practice is up to us to do the operation of an ultrasonic sensor using the Arduino development board. At the end of this section we are going to be able to see the potential applications that may have the fact to have something that we determine the proximity of an object either solid, liquid or dusty. We have a sensor that detect different types of composition in the field makes it an ideal choice in order to apply it in a larger number of projects.

Keywords— Arduino, LED, microcontroller, microprocessor, project, Sensor, USB.

Resumen— En esta práctica nos corresponde hacer el funcionamiento de un sensor ultrasónico usando la placa de desarrollo Arduino. Al final de este apartado vamos a poder ver las posibles aplicaciones que puede tener el hecho de tener algo que nos determine la proximidad de un objeto ya sea sólido, líquido o polvoriento. Que un sensor nos detecte diferentes tipos de composición en la materia lo hace una elección ideal para poder aplicarlo en un número más amplio de proyectos.

Índices— Arduino, circuitos, LED, microcontrolador, microprocesadores, proyecto, sensor, USB.

NOMENCLATURA

Arduino. En 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 multidisciplinarios [1].

CPU. Unidad Central de Procesamiento.

LED. Light Emitting Diode.

USB. Universal Serial Bus.

Sensor Ultrasónico. Son detectores de proximidad que trabajan libres de roces mecánicos

y que detectan objetos a distancias que van desde pocos centímetros hasta varios metros. El sensor emite un sonido y mide el tiempo que la señal tarda en regresar. Estos reflejan en un objeto, el sensor recibe el eco producido y lo convierte en señales eléctricas, las cuales son elaboradas en el aparato de valoración.

Microcontrolador. Un microcontrolador es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de E/S, es decir, se trata de un computador completo en un solo circuito integrado [2].

INTRODUCCIÓN

EL sensor ultrasónico nos va a servir para medir la proximidad de un objeto. Con esto podemos aplicar este trabajo a diferentes actividades cotidianas, por ejemplo que la luz de las escaleras de nuestro edificio se enciendan al detectar que estamos entrando.

PREPARACIÓN DEL TRABAJO (MARCO TEÓRICO)

Los sensores de ultrasonidos son detectores de proximidad que trabajan libres de roces mecánicos y que detectan objetos a distancias que van desde pocos centímetros hasta varios metros. El sensor emite un sonido y mide el tiempo que la señal tarda en regresar. Estos reflejan en un objeto, el sensor recibe el eco producido y lo convierte en señales eléctricas, las cuales son elaboradas en el aparato de valoración. Estos sensores trabajan solamente en el aire, y pueden detectar objetos con diferentes formas, colores, superficies y de diferentes materiales. Los materiales pueden ser sólidos, líquidos o polvorientos, sin embargo han de ser

deflectores de sonido. Los sensores trabajan según el tiempo de transcurso del eco, es decir, se valora la distancia temporal entre el impulso de emisión y el impulso del eco [3].

Ventajas e inconvenientes:

Este sensor, al no necesitar el contacto físico con el objeto, ofrece la posibilidad de detectar objetos frágiles, como pintura fresca, además detecta cualquier material, independientemente del color, al mismo alcance, sin ajuste ni factor de corrección. Los sensores ultrasónicos tienen una función de aprendizaje para definir el campo de detección, con un alcance mínimo y máximo de precisión de 6 mm. El problema que presentan estos dispositivos son las zonas ciegas y el problema de las falsas alarmas. La zona ciega es la zona comprendida entre el lado sensible del detector y el alcance mínimo en el que ningún objeto puede detectarse de forma fiable [3].

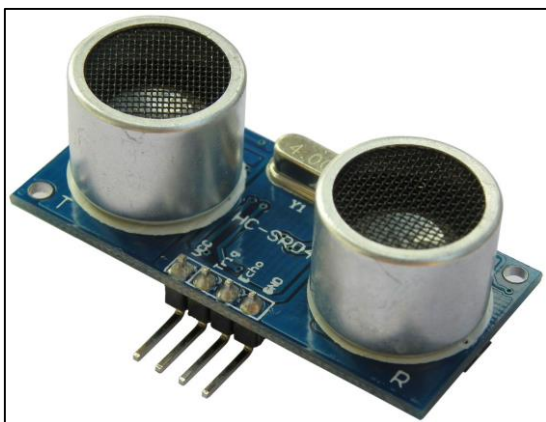


Fig. 1. Sensor Ultrasónico

En general son dispositivos autónomos de estado sólido diseñados para la detección sin contacto de objetos sólidos y líquidos. Para muchas aplicaciones, tales como el monitoreo del nivel de agua en un tanque, la tecnología ultrasónica permite que un dispositivo haga el trabajo que de otro modo requeriría varios sensores. Nuestros sensores están disponibles en varios rangos y estilos de detección y tienen una salida analógica o discreta según el modelo.

Algunos ejemplos de aplicación:

Trabaje con sensores versátiles y flexibles.

Los sensores ultrasónicos de Pepperl+Fuchs cubren un amplio abanico de situaciones de aplicación de automatización industrial. Pueden detectar una gran variedad de materiales, no se ven afectados por superficies problemáticas y presentan una gran inmunidad frente a las

influencias medioambientales. Sea cual sea su tarea, ya se trate de manipulación de materiales, equipos móviles, alimentación y bebidas, medición de nivel de llenado o detección en entradas y puertas, los sensores ultrasónicos aportan soluciones para los requisitos de aplicación más diversos. Eche un vistazo a 11 ejemplos de aplicación detallados que le muestran la versatilidad de los sensores ultrasónicos [4].

Sensores ultrasónicos para el control del nivel de llenado en silos de grava

Las minas de grava a cielo abierto contienen materiales de construcción como arena, roca fragmentada y grava. Estos materiales se excavan a profundidades de hasta 50 m y se deben almacenar de forma adecuada hasta que se puedan transportar a otro lugar. Las cintas transportadoras llevan los materiales de construcción a los silos. Los sensores ultrasónicos detectan si se alcanza el nivel de llenado máximo [4].

Para garantizar una cobertura de pulverización óptima en terrenos desiguales y cultivos con diferentes alturas, además de evitar que el brazo impacte con los cultivos o el suelo, es necesario controlar y ajustar su altura. Los sensores ultrasónicos, inmunes a los efectos del polvo, la suciedad y los químicos y con capacidad para detectar superficies de cualquier color con la misma precisión, resultan ideales para esta aplicación [4].

Sensores ultrasónicos para la protección antichoques en plataformas de trabajo aéreas

Las plataformas de trabajo aéreas con conductor son comunes en muchos emplazamientos de obras. Estas plataformas facilitan el trabajo en zonas de gran altura y aumentan la productividad. Sin embargo, dado que pueden producirse accidentes graves relacionados con las plataformas de trabajo aéreas debido a colisiones, no deben ignorarse las cuestiones de seguridad. Los sensores ultrasónicos protegen el funcionamiento de este equipo [4].

Transporte de placas de circuitos impresos

Desde los smartphones y los electrodomésticos hasta los coches que conducimos, los sistemas electrónicos (y, por tanto, las placas de circuito impreso o PCB) son una característica esencial de prácticamente todas las máquinas que usamos en la actualidad. Los PCB constituyen el núcleo de cualquier dispositivo y deben manipularse con el mayor cuidado. Los sensores ultrasónicos

contribuyen a controlar el proceso de producción de estos delicados PCB [4].

DESARROLLO DEL TRABAJO EXPERIMENTAL

Similar a las veces anteriores debemos asegurarnos que nuestro Arduino esté en perfecto estado y funcionando sin ningún problema. Después de conectarlo y ver que la computadora lo reconoce podemos cargar el código:

```
#define Pecho 8
#define Ptrig 9
long duracion, distancia;
void setup() {
  Serial.begin (9600);
  pinMode(Pecho, INPUT);
  pinMode(Ptrig, OUTPUT);
  pinMode(13, 1);
}
void loop() {
  digitalWrite(Ptrig, LOW);
  delayMicroseconds(2);
  digitalWrite(Ptrig, HIGH);
  delayMicroseconds(10);
  digitalWrite(Ptrig, LOW);
  duracion = pulseIn(Pecho, HIGH);
  distancia = (duracion/2) / 29;
  if (distancia >= 500 || distancia <= 0){
    Serial.println("---");      // no mide nada
  }
  else {
    Serial.print(distancia);
    Serial.println("cm");
    digitalWrite(13, 0);      // en bajo el pin 13
  }

  if (distancia <= 10 && distancia >= 1){
    digitalWrite(13, 1);
    Serial.println("Alarma.....");
  }
  delay(400);
}
```

La conexión del sensor con el Arduino queda como en la figura 2.

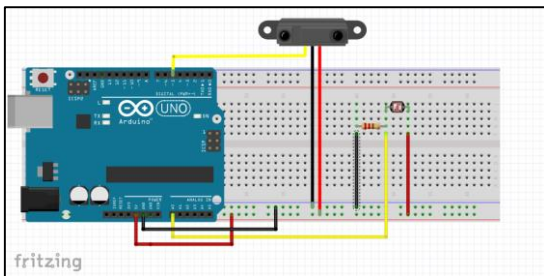


Fig. 2. Conexión del sensor con Arduino

RESULTADOS

Una vez realizada la implementación, el armado y la ejecución del programa correctamente podemos notar que el sensor determina la distancia a la que se encuentra el objeto.

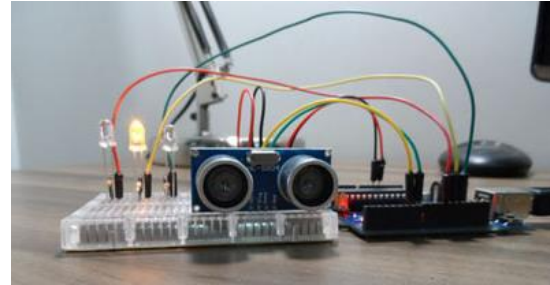


Fig. 3. Sensor detectando obstáculos

CONCLUSIONES Y DISCUSIÓN DE RESULTADOS

Este tipo de sensores tiene muchas aplicaciones, todas ellas con un gran campo de trabajo por experimentar. Nos queda como trabajo a futuro aterrizar el conocimiento adquirido hasta este momento en una situación donde podamos observar que la sociedad se ve beneficiada.

AGRADECIMIENTOS

A las personas externas al grupo de clases por haber aportado su conocimiento mediante las asesorías que nos dieron.

REFERENCIAS

- [1] *ARDUINO (2012, Aug.)*. Arduino Board - Serial Interface. En: <http://www.arduino.cc/en/Main/ArduinoBoardSerial>
- [2] Canto E. *Autómatas Programables Vol. I México: México 2012* pag. 15
- [3] *Sensores Ultrasónicos (2011, May.)*. Sensores – Sensor Ultrasónico. En: http://es.wikipedia.org/wiki/Sensor_ultras%C3%B3nico
- [4] 11 aplicaciones para los *Sensores Ultrasónicos (2011, Jan.)*. Aplicaciones – Sensor Ultrasónico. En: <http://www.pepperl-fuchs.es/spain/es/24854.html>

PRÁCTICA 4: Uso de un Servomotor con Arduino

Edgar Espinosa Ordoñez, Miguel Hernández Ramos, Elsa Chavira Martínez

Facultad de Ciencias de la Computación.

14 Sur y San Claudio, Edif. 104C, Ciudad Universitaria, Col. San Manuel, C.P. 72570, Puebla, Pue., México.

(edgar.espinosa.fcc@gmail.com, miguellhdez@hotmail.com), elsachavira56@hotmail.com

Abstract— We must now introduce us to the topic of the motors. In this paper does not include the comparisons between the different types of engines that exist because it is a very lengthy to treat. All the information that you will find referenced here is with respect to the servos therefore the implementation is done based on them. It was going to mention the reasons for which he was elected, as well as the advantages that present and what they might be used in everyday life.

Keywords— Arduino, circuits, microcontroller, microprocessors, motors, project, Servo, USB.

Resumen— Ahora nos toca introducirnos en el tema de los motores. En este documento no se incluyen las comparaciones entre los diferentes tipos de motores que existen debido a que es un tema muy extenso de tratar. Toda la información referenciada que encontrará acá es con respecto a los servomotores por lo tanto la implementación se hace en base a ellos. Se van a mencionar los motivos por los que fue elegido, así como las ventajas que presentar y el uso que se les puede dar en la vida cotidiana.

Índices— Arduino, circuitos, microcontrolador, microprocesadores, motores, proyecto, Servo, USB.

NOMENCLATURA

Arduino. En 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 multidisciplinarios [1].

CPU. Unidad Central de Procesamiento.

USB. Universal Serial Bus.

Microcontrolador. Un microcontrolador es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de E/S, es decir, se trata

de un computador completo en un solo circuito integrado.

Motor. En general, Un motor es la parte sistemática de una máquina capaz de hacer funcionar el sistema, transformando algún tipo de energía (eléctrica, de combustibles fósiles, etc.), en energía mecánica capaz de realizar un trabajo. En los automóviles este efecto es una fuerza que produce el movimiento [2].

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 [3].

INTRODUCCIÓN

EL conocimiento de los motores, el servo en este caso, nos va a servir para que dado un proyecto podamos usar su fuerza para moverlo. Es el caso de los proyectos que desarrollan autos seguidores de línea, drones, cuadricópteros, etc. Vamos a profundizar en este tema para que al concluir con la lectura no tenga duda de que el servo puede ser su mejor opción.

PREPARACIÓN DEL TRABAJO (MARCO TEÓRICO)

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 [2].

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 [2].

Los servomotores hacen uso de la modulación por ancho de pulsos (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 antihorario; 1,5 milisegundos representa un estado neutro para los servomotores estándares.



Fig. 1. Servomotor, también llamado servo

Los Servos son sumamente útiles en robótica. Los motores son pequeños, tiene internamente una circuitería de control interna y es sumamente poderoso para su tamaño. Un servo normal o Standard como el HS-300 de Hitec tiene 42 onzas por pulgada o mejor 3kg por cm. De torque que es bastante fuerte para su tamaño. También potencia proporcional para cargas mecánicas. Un servo, por consiguiente, no consume mucha energía. Se muestra la composición interna de un servo motor en el cuadro de abajo. Podrá observar la circuitería de control, el motor, un juego de piñones, y la caja. También puede ver los 3 alambres de conexión externa. Uno es para alimentación Vcc (+5volts), conexión a tierra GND y el alambre blanco es el alambre de control.

Partes.

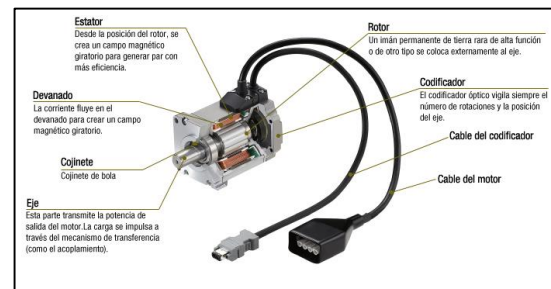


Fig. 2. Partes de un servo

Motor de corriente continúa

Es el elemento que le brinda movilidad al servo. Cuando se aplica un potencial a sus dos terminales, este motor gira en un sentido a su velocidad máxima. Si el voltaje aplicado sus dos terminales es inverso, el sentido de giro también se invierte.

Engranajes reductores

Se encargan de convertir gran parte de la velocidad de giro del motor de corriente continua en torque.

Circuito de control

Este circuito es el encargado del control de la posición del motor. Recibe los pulsos de entrada y ubica al motor en su nueva posición dependiendo de los pulsos recibidos.

Tiene además de los circuitos de control un potenciómetro conectado al eje central del motor. Este potenciómetro permite a la circuitería de control, supervisar el ángulo actual del servo motor. Si el eje está en el ángulo correcto, entonces el motor está apagado. Si el circuito chequea que el ángulo no es correcto, el motor volverá a la dirección

correcta, hasta llegar al ángulo que es correcto. El eje del servo es capaz de llegar alrededor de los 180 grados. Normalmente, en algunos llega a los 210 grados, pero varía según el fabricante.

Un servo normal se usa para controlar un movimiento angular de entre 0 y 180 grados. Un servo normal no es mecánicamente capaz de retornar a su lugar, si hay un mayor peso que el sugerido por las especificaciones del fabricante.

Los servomotores tienen 3 terminales:

Terminal positivo: Recibe la alimentación del motor (4 a 8 voltios)

Terminal negativo: Referencia tierra del motor (0 voltios)

Entrada de señal: Recibe la señal de control del motor

Los colores del cable de cada terminal varían con cada fabricante: el cable del terminal positivo siempre es rojo; el del terminal negativo puede ser marrón o negro; y el del terminal de entrada de señal suele ser de color blanco, naranja o amarillo [5].

DESARROLLO DEL TRABAJO EXPERIMENTAL

El código correspondiente a la implementación de un servomotor con Arduino es el siguiente.

```
#include<Servo.h>
Servo miServo;
const int pinPot=0;
const int pinServo=2;
const int pulsoMin=650;//pulso en us para girar el
servo a 0
const int pulsoMax=2550;//pulso en us para girar el
servo a 180
int valor;
int angulo;
void setup(){
miServo.attach(pinServo,pulsoMin,pulsoMax);
}
void loop(){
valor=analogRead(pinPot);
angulo=map(valor, 0, 1023, 0, 180);
miServo.write(angulo);
delay(1);
}
```

La conexión con Arduino queda de la forma en que se muestra en la figura tres.

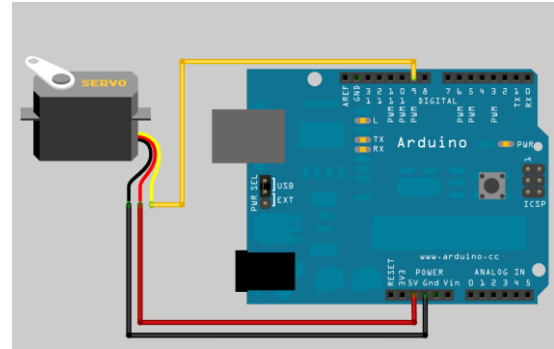


Fig. 3. Esquema de conexión Servomotor – Arduino

RESULTADOS

Dependiendo del modelo del servo, la tensión de alimentación puede estar comprendida entre los 4 y 8 voltios. El control de un servo se reduce a indicar su posición mediante una señal cuadrada de voltaje. El ángulo de ubicación del motor depende de la duración del nivel alto de la señal. Cada servo motor, dependiendo de la marca y modelo utilizado, tiene sus propios márgenes de operación. Para el servomotor Futaba S3003, los valores posibles de la señal en alto están entre 0,3 y 2,1 ms, que posicionan al motor en ambos extremos de giro (0° y 180°, respectivamente). El valor 1,2 ms indica la posición central, y otros valores de duración del pulso dejarían al motor en la posición proporcional a dicha duración.

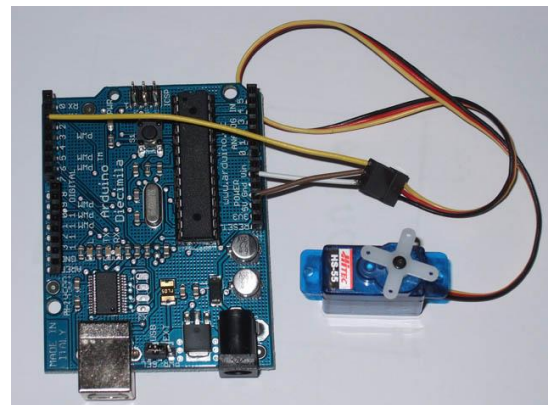


Fig. 4. Muestra el Servo conectado a Arduino

CONCLUSIONES Y DISCUSIÓN DE RESULTADOS

Como se mencionó antes: cada Servo tiene diferentes características de funcionamiento. Para futuros trabajos es necesario que se busque uno dependiendo del tipo de proyecto

REFERENCIAS

- [1] *ARDUINO (2012, Aug.)*. Arduino Board - Serial Interface. En: <http://www.arduino.cc/en/Main/ArduinoBoardSerial>
- [2] *Motor (2014, Aug.)*. Motores – Tipos de motores, ventajas y todo lo que debes saber de motores. En: <http://es.wikipedia.org/wiki/Motor>
- [3] *Irving L. Kosow (1 de mayo de 1993)*. Máquinas eléctricas y transformadores. Pearson Educación. pp. 429-. ISBN 9789688802939. Consultado el 26 de enero de 2011.
- [4] *Servomotores(2011, Aug.)*. Funcionamiento y definición. En: <http://www.info-ab.uclm.es/labelec/solar/electronica/elementos/servomotor.htm>
- [5] *Y. RamirezS (2008, Feb.)*. Servo y sus partes En <http://www.monografias.com/trabajos60/servo-motores/servo-motores.shtml>

PROYECTO FINAL: Alarma Luminosa

Edgar Espinosa Ordoñez, Miguel Hernández Ramos, Elsa Chavira Martínez

Facultad de Ciencias de la Computación.

14 Sur y San Claudio, Edif. 104C, Ciudad Universitaria, Col. San Manuel, C.P. 72570, Puebla, Pue., México.

(edgar.espinosa.fcc@gmail.com, miguellhdez@hotmail.com), elsachavira56@hotmail.com

Abstract— All the projects in the area of technology need to be focused on any area, whether social or economic. This ensures that you have an impact and that the scope are expected. This project develops an alarm light implemented with the Arduino development board, which as we know today has thousands of applications in everyday life, not to mention the incredible number of projects that young people like us are being developed by all parts of the world. In this document you will find some of the applications you can have our project and all the advantages to existing projects that have been applied to a specific group of people. We include also the specification and justification of the project so that anyone can understand that this is "Warning Light".

Keywords— Arduino, circuit, LED, light, light, microcontroller, microprocessor, engine, breadboard, project, ultrasonic sensor, Servo, USB.

Resumen— Todos los proyectos del área de la tecnología necesitan estar enfocados a algún ámbito, ya sea social o económico. Con esto se garantiza que tenga un impacto y que los alcances sean los esperados. Este proyecto desarrolla una alarma luminosa implementada con la placa de desarrollo Arduino, que como ya sabemos en la actualidad tiene miles de aplicaciones en la vida cotidiana, sin mencionar el número increíble de proyectos que jóvenes como nosotros están desarrollando por todas partes del mundo. En este documento podrá encontrar algunas de las aplicaciones que puede tener nuestro proyecto y todas las ventajas que presenta ante proyectos existentes que se aplicaron a un grupo en específico de personas. Incluimos también la especificación y justificación del proyecto para que cualquier persona pueda entender de qué trata "Alerta Luminosa".

Índices— Arduino, circuito, LED, luminosidad, luz, microcontrolador, microprocesador, motor, protoboard, proyecto, sensor ultrasónico, Servo, USB.

NOMENCLATURA

Arduino. En 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 multidisciplinarios [1].

Circuito. Un circuito eléctrico consiste en un conjunto de elementos u operadores que, unidos entre sí, permiten establecer una corriente entre dos puntos, llamados polos o bornes, para aprovechar la energía eléctrica [2].

LED. Light-Emitting Diode (Diodo Emisor de Luz)

CPU. Unidad Central de Procesamiento.

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 [4].

Microcontrolador. Un microcontrolador es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de E/S, es decir, se trata de un computador completo en un solo circuito integrado [3].

Motor. En general, Un motor es la parte sistemática de una máquina capaz de hacer funcionar el sistema, transformando algún tipo de energía (eléctrica, de combustibles fósiles, etc.), en energía mecánica capaz de realizar un trabajo. En los automóviles este efecto es una fuerza que produce el movimiento [4].

Protoboard. O placa de pruebas, es una placa de uso genérico reutilizable de proyectos de electrónica.

USB. Universal Serial Bus.

INTRODUCCIÓN

EL proyecto “Alarma Luminosa” es una caja que tiene LEDs para hacer luminosidad con el objetivo de informar a un usuario sobre algún suceso o actividad. Es un proyecto que puede adaptarse a diversos problemas para personas con capacidades diferentes o que requieran ser avisados de formas más llamativas. “Alarma Luminosa” consiste de una caja realizada con material ligero para facilitar su movilidad en caso de ser necesario transportarla. Fue desarrollada en Arduino, y por la necesidad de usar el menor espacio posible usamos Arduino NANO.

PREPARACIÓN DEL TRABAJO (MARCO TEÓRICO)

Para la mejor comprensión del proyecto vamos a definir algunos conceptos para poder comprender mejor el objetivo principal de “Alarma Luminosa”

Luminosidad.

Es la cantidad de luz que puede llegar a entrar a través de la lente frontal de un objetivo. Con mucha luminosidad en un objetivo, se pueden realizar buenas imágenes aunque haya poca luz. La exposición también depende de la cantidad de luz que pasa a través de las lentes de nuestro objetivo durante un tiempo determinado [5].

Alarma.

Señal o aviso que advierte sobre la proximidad de algo.

Al año 2010 las personas en México con algún tipo de discapacidad son 5 millones 739 mil 270, lo cual representa 5.1% de la población total [6]. Se calcula que del total de personas con capacidades diferentes el 20% corresponde a personas con problemas auditivos. Esto representa un problema al tener que convivir diariamente en un mundo donde los avances tecnológicos están cada vez más avanzados.

Por cuestiones de espacio fue conveniente usar el Arduino NANO para este trabajo. La imagen uno muestra como es la parte frontal de esta placa de desarrollo.

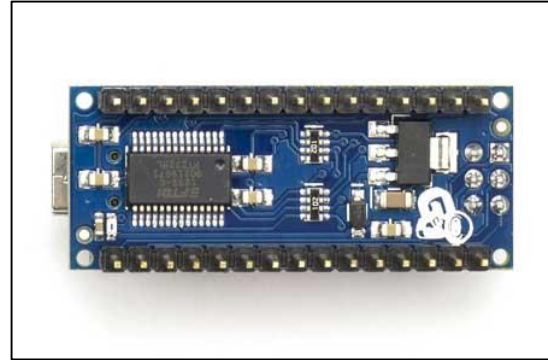


Fig. 1. Parte trasera de Arduino NANO

Como podemos notar el tamaño de este Arduino es más pequeño a comparación del UNO.

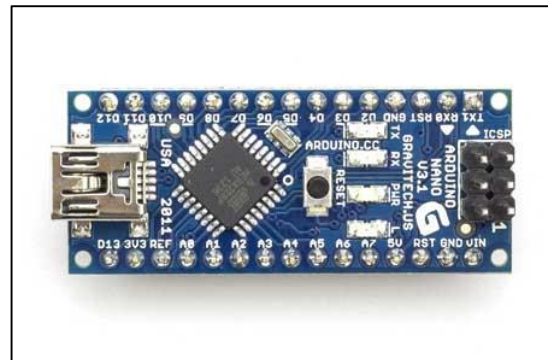


Fig. 2. Parte Frontal de Arduino NANO

La siguiente es una lista de las especificaciones generales [7].

Microcontroller: Atmel ATmega168 or ATmega328
Operating Voltage (logic level): 5 V
Input Voltage (recommended): 7-12 V
Input Voltage (limits) 6-20 V
Digital I/O Pins 14 (of which 6 provide PWM output)
Analog Input Pins: 8
DC Current per I/O Pin: 40 mA
Flash Memory: 16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM 1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM: 512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed: 16 MHz
Dimensions: 0.73" x 1.70"
Length: 45 mm
Width: 18 mm
Weight: 5

DESARROLLO DEL TRABAJO EXPERIMENTAL

Este proyecto fue desarrollado para ayudar a las personas sordas a visualizar fácilmente cuando reciben por el móvil una llamada de voz, o más bien un mensaje de texto, ya que aunque la mayoría de las móviles disponen de sistema de vibración, también es cierto que la gran mayoría de las ocasiones no se percatan de cuando este está vibrando. Para desarrollar el proyecto fue necesario contar con:

- 1 placa de desarrollo Arduino NANO
- 1 Sensor Ultrasónico
- Una placa de Estireno para armar la caja
- Placa grabada para montar los dispositivos
- Matriz de LEDs 8X8



Fig. 3. Estireno

Es necesario considerar las dimensiones finales para hacer los cortes del estireno para armar la caja. Las dimensiones dependen del tamaño de la placa grabada, la matriz y los demás componentes que van adentro.



Fig. 4. Placa grabada con los detalles del equipo y sus integrantes

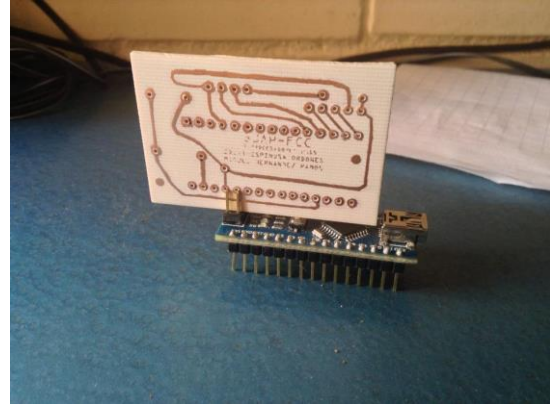


Fig. 5. Arduino NANO y placa



Fig. 6. Parte superior del Arduino NANO y la placa



Fig. 7. Caja de estireno. Armada manualmente

Teniendo todo lo necesario armado el siguiente paso es meter a la caja estos componentes. La figura 8 muestra la caja con los componentes adentro.

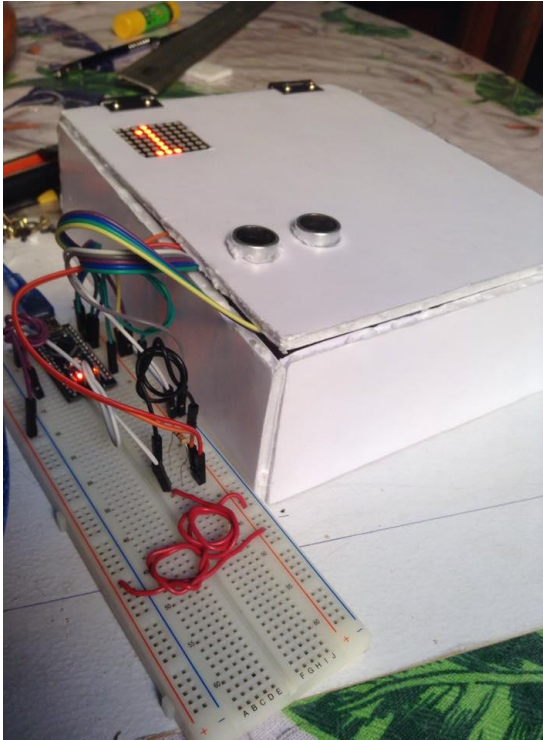


Fig. 8. Caja armada completamente

El código que corresponde al proyecto es el siguiente.

```
/*
Inicialmente la matriz se encuentra apagada
(todos los leds en 0)
por cada notificación o actividad que se tenga
apareciera un numero indicando el número de
veces que ha habido actividad del teléfono
para reiniciar la matriz solo bastara pasar la
mano sobre el sensor ultrasónico suponiendo
que se reinicia porque la persona ya hizo caso
de su actividad el teléfono
*/
#include <LedControl.h>
//puertos para matriz leds 8x8
const byte DIN    = 12; //Lo conectamos en
    din
const byte CS      = 11; //Lo conectamos a
    Load (cs)
const byte CLK     = 10; //Lo conectamos a
    CLK
const byte QTD_DISP = 1; //El nuenmro de
    matriz con controlador M72XX
```

```
//puerto para sensor LDR
int pinLDR = 0; // Pin analogico 0 de entrada
    para el LDR
int valorLDR = 0; // Variable donde se
    almacena el valor del LDR

//puertos y variables para sensor ultrasonico
#define Ptrig 9 //pin 7 para pin trigger del
    sensor ultrasonico
#define Pecho 8 //pin 6 para pin echo del
    sensor ultrasonico
long duracion, distancia; //variables para
    realizar la medicion del pulso y distancia

//inicializacion de matriz
LedControl ledMatrix = LedControl(DIN,
    CLK, CS, QTD_DISP);
int contador; //variable global para el conteo
    de actividad(la matriz y el sensor ldr
    dependen de ella)

void setup() {

    // El MAX72XX está en modo de ahorro de
        energía en el arranque, tenemos que hacer
        que despierte
    ledMatrix.shutdown(0, false); //modo
        'shutdown' no display '0' y FALSE
    ledMatrix.setIntensity(0, 5); //intensidad del
        brillo de la matriz '0' y 5 (0~16)
    ledMatrix.clearDisplay(0); //borrar pantalla
    Serial.begin(9600); //comunicacion
        puerto serial

    //declaramos los puertos del sensor
        ultrasonico
    pinMode(Pecho, INPUT); // define el pin 6
        como entrada (echo)
    pinMode(Ptrig, OUTPUT); // define el pin
        7 como salida (trigger)
}

//CONJUNTO DE FUNCIONES QUE VAN
    DE 0 A 9 ES DECIR DESPLIEGA
    HASTA 9 NOTIFICACIONES EN LA
    MATRIZ, UNA VEZ LLEGANDO A 10
    O MAS SOLO MOSTRARA EL TEXTO
    "MENSAJE/LLAMADA" SEGUIDO
    DEL SIMNOLO '!'
void uno()
{
    delay(200);
    ledMatrix.printChar(0, 0, '1');
}
```

```

void dos()
{
    ledMatrix.printChar(0, 0, '2');
}
void tres()
{
    ledMatrix.printChar(0, 0, '3');
}
void cuatro()
{
    ledMatrix.printChar(0, 0, '4');
}
void cinco()
{
    ledMatrix.printChar(0, 0, '5');
}
void seis()
{
    ledMatrix.printChar(0, 0, '6');
}
void siete()
{
    ledMatrix.printChar(0, 0, '7');
}
void ocho()
{
    ledMatrix.printChar(0, 0, '8');
}
void nueve()
{
    ledMatrix.printChar(0, 0, '9');
}
void buzon()
{
    ledMatrix.printChar(0, 0, 'M');
    delay(300);
    ledMatrix.clearDisplay(0);
    ledMatrix.printChar(0, 0, 'e');
    delay(300);
    ledMatrix.clearDisplay(0);
    ledMatrix.printChar(0, 0, 'n');
    delay(300);
    ledMatrix.clearDisplay(0);
    ledMatrix.printChar(0, 0, 's');
    delay(300);
    ledMatrix.clearDisplay(0);
    ledMatrix.printChar(0, 0, 'a');
    delay(300);
    ledMatrix.clearDisplay(0);
    ledMatrix.printChar(0, 0, 'j');
    delay(300);
    ledMatrix.clearDisplay(0);
    ledMatrix.printChar(0, 0, 'e');
    delay(300);

```

```

ledMatrix.clearDisplay(0);
ledMatrix.printChar(0, 0, '/');
delay(300);
ledMatrix.clearDisplay(0);
ledMatrix.printChar(0, 0, 'l');
delay(300);
ledMatrix.clearDisplay(0);
ledMatrix.printChar(0, 0, 'l');
delay(300);
ledMatrix.clearDisplay(0);
ledMatrix.printChar(0, 0, 'a');
delay(300);
ledMatrix.clearDisplay(0);
ledMatrix.printChar(0, 0, 'm');
delay(300);
ledMatrix.clearDisplay(0);
ledMatrix.printChar(0, 0, 'a');
delay(300);
ledMatrix.clearDisplay(0);
ledMatrix.printChar(0, 0, 'd');
delay(300);
ledMatrix.clearDisplay(0);
ledMatrix.printChar(0, 0, 'a');
delay(300);
ledMatrix.clearDisplay(0);
}

void loop(){
    //INICIALIZAMOS LECTURA DE
    SENSOR LDR
    valorLDR= analogRead(pinLDR); // lectura
    del sensor LDR mediante el puerto
    analogo 0

    //INICIALIZAMOS LECTURA DE
    SENSOR ULTRASONICO
    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

    //MONITOREO DE ACTIVIDAD DE
    SENSORES MEDIANTE PUERTO
    SERIAL
    Serial.print(valorLDR); // envia el
    valor de la resistencia por el puerto serial
    Serial.print(" ldr");
    Serial.println("");
    Serial.print(distancia); // envia el valor
    de la distancia por el puerto serial

```


REFERENCIAS

- [1] *ARDUINO* (2012, Aug.). Arduino Board - Serial Interface. En: <http://www.arduino.cc/en/Main/ArduinoBoardSerial>
- [2] Guijarro, G.. (2014). *Circuitos serie y paralelo*. Abril 15, 2015, de IES Bahía de Algeciras Sitio web: http://luis.tarifasoft.com/2_eso/electricidad/2ESO/circuitos_serie_y_paralelo.html
- [3] Galindo A.. (Diciembre 12, 2010). Microcontroladores. Abril 15, 2015., de MikroElektronika Sitio web: <http://www.mikroe.com/chapters/view/79/capitulo-1-el-mundo-de-los-microcontroladores/>
- [4] Irving L. Kosow (1 de mayo de 1993). *Máquinas eléctricas y transformadores*. Pearson Educación. pp. 429-. ISBN 9789688802939. Consultado el 26 de enero de 2011.
- [5] Hernández L.. (2007). La luminosidad. Abril 17, 2015, de FOTONOSTRA Sitio web: <http://www.fotonostra.com/fotografia/objetivo.htm>
- [6] INEGI. (2010). Discapacidad en México. Abril 20, 2015, de INEGI Sitio web: <http://cuentame.inegi.org.mx/poblacion/discapacidad.aspx?tema=P>
- [7] Arduino. (2015). Arduino NANO. Abril 20, 2015, de ARDUINO Sitio web: <http://www.arduino.cc/en/Main/arduinoBoardNano>

ACRÓNIMOS

AMD: Advanced Micro Devices

ISP: Integrate Serial Port

SD: Security Digital

MODEM: modulator demodulator

PIC: Programmable Interrupt Controller

CMOS: Complementary metal oxide
semiconductor

PCB; Printed Circuit Board

CNR: Communication and Network Riser

MIDI: Musical Instrument Digital Interface

ASICs: application Especific Integrate Circuits

MIPS: Microprocessor without Interlocked
Pipeline Stages

ASCII: Definición de ascii: American
Standard Code of Information
Interchange

DSP: Digital Signal Procesing