



Universidad Abierta Interamericana

Facultad de Tecnología Informática

Ingeniería en Sistemas Informáticos

Electromagnetismo en Estado Sólido II

Integrantes:

Bacelli Julian,
Cantoia Noel,
Dominguez Jacobo,
Mariani Julia

Año:

2015

Docente:

Ing. Lopez, Pedro



Trabajo Final

Matriz de LED comandada por joystick

Introducción

El proyecto consiste en la utilización de una placa Arduino y su IDE de desarrollo, que es utilizada para manejar una serie de LEDs dentro de una matriz comandada por un joystick analógico. En este trabajo se explican los materiales requeridos para su construcción, la forma en que se conectan, la funcionalidad del sistema y el software utilizado para la realización del proyecto.

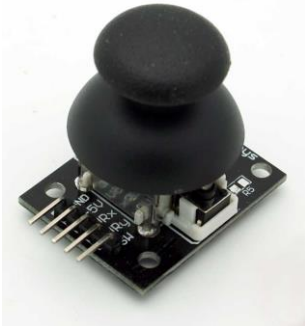
Materiales

- ❖ Joystick:
- ❖ Placa Arduino
- ❖ Matriz LED
- ❖ Protoboard
- ❖ Zócalo
- ❖ Cables

Todos estos materiales forman el Hardware que se utiliza en el proyecto. Dichos elementos son comandados por un software específico desarrollado por el equipo de trabajo. A continuación se detallan:



JOYSTICK



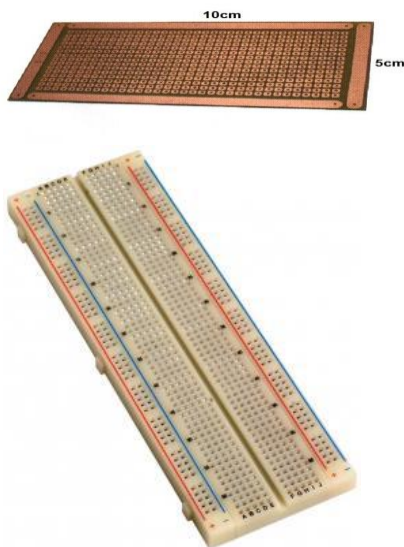
Los joysticks son dispositivos de entrada que indican la posición o el ángulo al dispositivo que controla. Los mismos pueden ser digitales (solo indican estados de encendido y apagado) o análogos (indican posición mediante potenciómetros).

Arduino ofrece la posibilidad de obtener lecturas de dispositivos análogos dando la posibilidad de conocer el estado o posición en la que se encuentra. Esta característica es aprovechable mediante la implementación de dispositivos de control como son los joysticks análogos.

Los joysticks análogos son dispositivos que, por lo general, constan de dos ejes los cuales controlan dos potenciómetros que indicaran al dispositivo que maneja, la dirección que el usuario desea.

En este proyecto se utiliza el joystick Arduino KY-023, que es un joystick analogo que controla los ejes X e Y.

PROTOBOARD O ZOCALO



Un protoboard o zócalo es un tablero con orificios que se encuentran conectados eléctricamente entre sí de manera interna, habitualmente siguiendo patrones de líneas, en el cual se pueden insertar componentes electrónicos y cables para el armado y prototipado de circuitos electrónicos y sistemas similares. Está hecho de dos materiales, un aislante, generalmente un plástico, y un conductor que conecta los diversos orificios entre sí. Uno de sus usos principales es la creación y comprobación de prototipos de circuitos electrónicos antes de llegar a la impresión mecánica del circuito en sistemas de producción comercial.

Estos 2 dispositivos realizan la misma tarea, la de transportar corriente a través de los cables y pasando por todos los circuitos electrónicos conectados a las mismas. En el proyecto se utiliza el zócalo para la conexión con la matriz de LED y se utiliza la protoboard para conectar el joystick.



PLACA ARDUINO



Se trata de una placa open hardware por lo que su diseño es de libre distribución y utilización, que incluso podemos construirnos nosotros mismos.

La placa Arduino Duemilanove apareció en 2009, con microcontrolador ATmega168 o 328, 14 pines digitales (6 con posibilidad PWM) y 6 analógicos. Contiene conexión USB, entrada de alimentación, una cabecera ISCP, botón de reset. Todo bastante parecido a UNO ya que se trata de una versión previa que fue sustituida por UNO en la actualidad.

Microcontroller	ATmega168
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 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



MATRIZ LED



Parece que los LEDs se fabrican en todos los tamaños y formatos imaginables, y este componente que utilizamos en este trabajo, hace referencia a 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.

En este proyecto, vamos a usar una matriz de 8x8 LEDs de color rojo.

CABLES



Los cables que se usan para conducir electricidad se fabrican generalmente de cobre, debido a la excelente conductividad de este material, o de aluminio que aunque posee menor conductividad es más económico.

Un cable eléctrico se compone de:

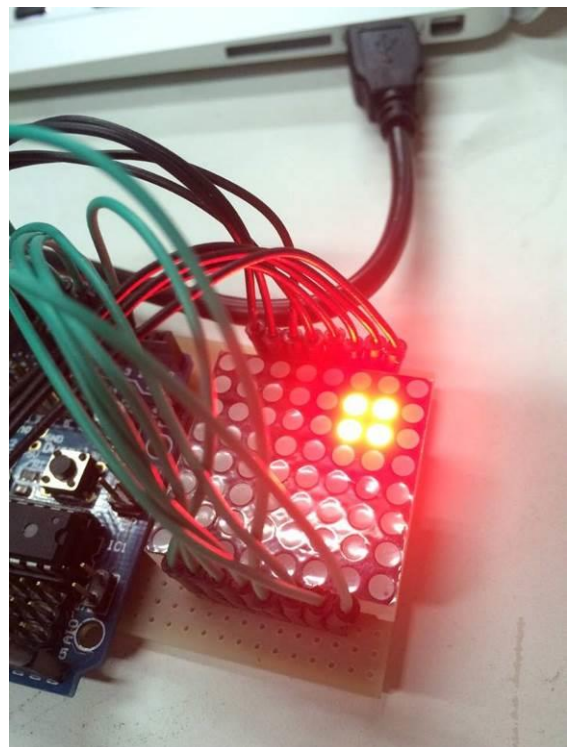
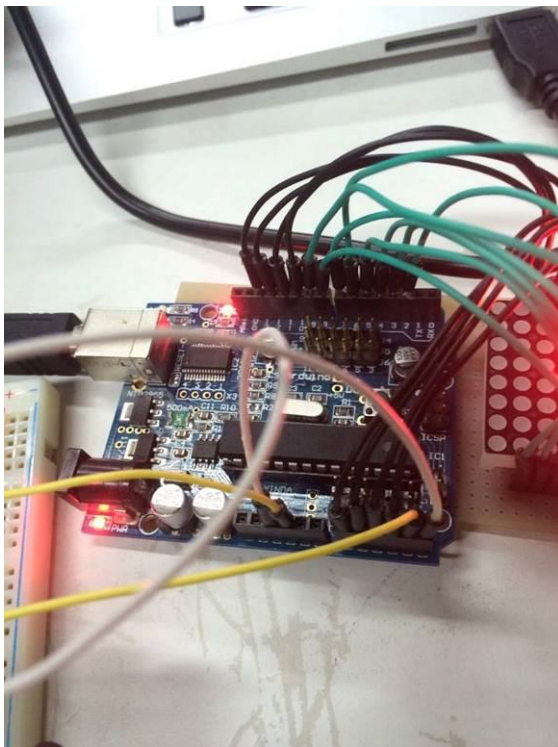
- Conductor: Elemento que conduce la corriente eléctrica y puede ser de diversos materiales metálicos. Puede estar formado por uno o varios hilos.
- Aislamiento: Recubrimiento que envuelve al conductor, para evitar la circulación de corriente eléctrica fuera del mismo.
- Capa de relleno: Material aislante que envuelve a los conductores para mantener la sección circular del conjunto.
- Cubierta: Está hecha de materiales que protejan mecánicamente al cable. Tiene como función proteger el aislamiento de los conductores de la acción de la temperatura, sol, lluvia, etc.



Conexión

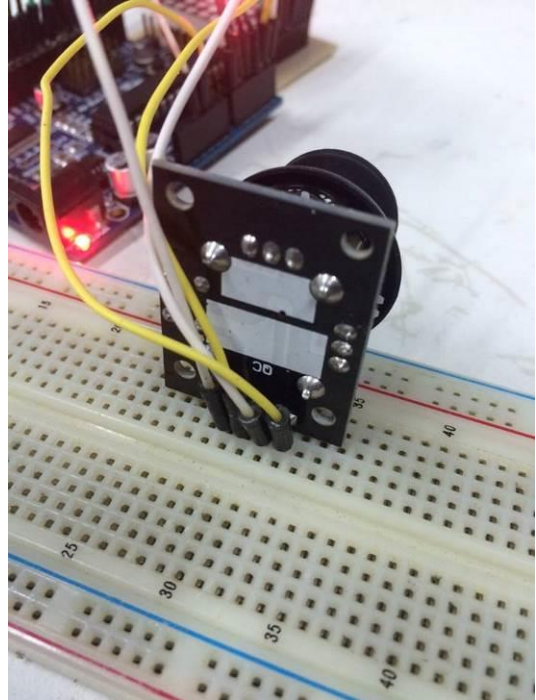
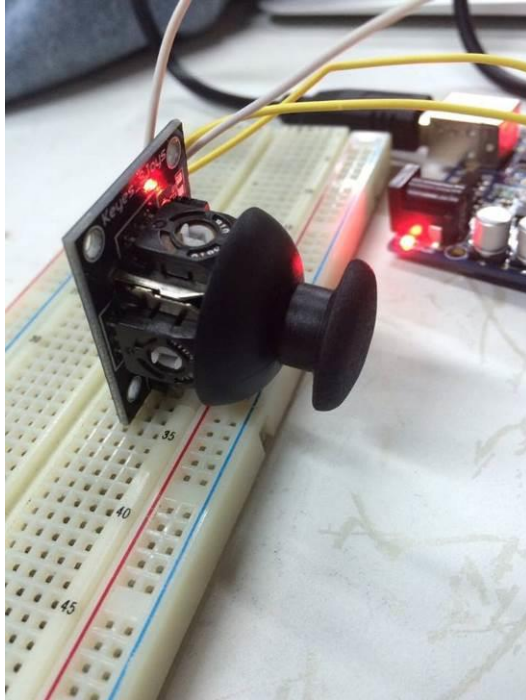
La conexión tanto de la matriz LED como del joystick a la placa Arduino se realizó de la siguiente manera:

- Matriz led: los 16 cables necesarios para realizar la conexión la matriz fueron conectados a los pines 2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17 de la placa arduino.





- Joystick: Se conectan los extremos VRx y VRy a los pines analógicos A4 y A5 respectivamente. También se conectan los extremos GND al pin de Ground de la placa y lo mismo sucede con el extremo 5V que coincide con el de 5V de la placa.



Además se conectó la placa al software de Arduino por medio del cable USB.



Funcionalidad

Como se explicó anteriormente, la matriz led cuenta con 8 filas y 8 columnas. La funcionalidad principal de este proyecto es comandar mediante el joystick conectado a la placa Arduino los LEDs a través de las filas y las columnas de manera deseada. Para un futuro es posible agregar matrices al proyecto y poder realizar un comando mediante joystick a través de una superficie más grande. Básicamente, se comienza dibujando en la matriz un cuadrado con 4 de los LEDs encendidos ubicados en la parte media de la matriz (0,0). De ahí se utiliza el joystick para controlar el encendido y apagado de los LEDs acorde la dirección en la que se movilice el mando. Siempre se encenderán de a 4 emulando un cuadrado como el del principio.

En el caso que el movimiento implique que el cuadrado de 4 LEDs “se salga” de la zona de la matriz, por alguno de los bordes, el cuadrado de LEDs aparecerá en el borde contrario.

Para determinar el encendido y apagado de los LEDs dentro de la matriz, primero es vital identificar cómo se comportan las columnas y filas de la misma. Para ello se utilizó un código a modo de prueba en el que se determinaron que pines correspondían a las columnas y cuales a las filas para luego crear el software pertinente.

Dicha prueba arrojó los siguientes resultados que se resumen en el siguiente esquema:

	1	2	3	4	5	6	7	8
1	5 12	2 12	7 12	1 12	13 12	8 12	11 12	16 12
2	5 3	2 3	7 3	1 3	13 3	8 3	11 3	16 3
3	5 4	2 4	7 4	1 4	13 4	8 4	11 4	16 4
4	5 15	2 15	7 15	1 15	13 15	8 15	11 15	16 15
5	5 6	2 6	7 6	1 6	13 6	8 6	11 6	16 6
6	5 14	2 14	7 14	1 14	13 14	8 14	11 14	16 14
7	5 10	2 10	7 10	1 10	13 10	8 10	11 10	16 10
8	5 9	2 9	7 9	1 9	13 9	8 9	11 9	16 9



Software

El código utilizado a continuación está configurado y regulado para el caso particular de los materiales utilizados.

```
int Filas[8] = {13,4,5,16,7,15,11,10}; //asignación de pines
correspondientes a cada fila, en orden ascendente
int Columnas[8] = {6,3,8,2,14,9,12,17}; //asignación de pines
correspondientes a cada columna, en orden ascendente
int ColumnaActual=3; //Declaración de columna inicial
int FilaActual=3; //Declaración de fila inicial
int Xreposo; //Variable que se utilizará para guardar el valor del eje X
en reposo
int Yreposo; //Variable que se utilizará para guardar el valor del eje Y
en reposo
//el primer valor de los vectores se encuentra en la posicion 0 de cada
uno, por lo cual para referirse a una fila o columna dentro del vector,
se toma un numero menor a la
// fila o columna; ejemplo: para la fila 4, el valor es Filas[3].
void setup()
{
    for (int j=2; j<19; j++) //Declaración de los pines de salida
correspondientes a cada led
    {
        pinMode(j, OUTPUT);
    }
    ApagarTodo(); //Apagar todos los leds
    Yreposo = analogRead(A5); //lectura del valor de reposo del
joystick en el eje Y
    Xreposo = analogRead(A4); //lectura del valor de reposo del
joystick en el eje X
    EncenderFila(3); //Encender la fila inicial
    EncenderColumna(3); //Encender la columna inicial
    EncenderFila(4); //Encender la fila siguiente
    EncenderColumna(4); //Encender la columna siguiente para
conseguir un cuadrado
}

//Siempre se encenderá un cuadrado de 2x2, por lo cual la fila y
columna actual se tomará como referencia para encender todo el cuadrado
//La primer fila/columna será la que se guarda en las variables
FilaActual y ColumnaActual, y la siguiente será la posición siguiente
//En el eje correspondiente; ejemplo: FilaActual = 4, se encienden
las filas 4 y 5(lo mismo para las columnas)
//El valor de la fila actual es el número real en la matriz, por lo
que será necesario leer una posición menos en el vector.

void ApagarTodo() //Funcion que apaga todos los leds
{
    for(int p=0;p<8;p++)
    {
```



```
        digitalWrite(Filas[p],HIGH); //Las filas se apagan enviado una
señal HIGH
        digitalWrite(Columnas[p],LOW); //Las columnas se apagan
enviando una señal LOW
    }
}

void loop()
{
    int y= analogRead(A5); //Lectura del valor actual de Y
    int x= analogRead(A4); //Lectura del valor actual de X
    ApagarTodo(); //Apagar todos los leds
    if(y > Yreposo+1) //Si el nuevo valor de Y es mayor al de
    reposo
    {
        if(FilaActual <=5) //Y la fila actual no es la
    anteultima(si se avanza mas se sale de la matriz led)
        {
            EncenderFila(FilaActual+2); //Encender la fila siguiente
            delay(100);
            ApagarColumna(FilaActual); //Apagar la fila actual
            FilaActual = FilaActual + 1; //Incrementar la fila
    actual a la nueva posición
        }
        else //Si la fila es la anteultima
        {
            EncenderFila(0); //Encender la primer fila
            EncenderFila(7); //Encender la ultima fila
            delay(100);
            ApagarColumna(6); //Apagar la fila actual
            FilaActual = 7; //Asignar la fila actual a la nueva
    posicion
        }
    }
    else //si el nuevo valor de Y no es mayor al de reposo
    {
        if(y < Yreposo-1) //si el nuevo valor de Y es menor al
    de reposo
        {
            if(FilaActual>=1) //y la fila actual es mayor o igual
    a la segunda
            {
                EncenderFila(FilaActual-1); //Encender la fila
    anterior
                delay(100);
                ApagarColumna(FilaActual+1); //Apagar la columna
    Actual
                FilaActual = FilaActual - 1; //Asignar la fila
    actual a la nueva posición
            }
        }
    }
}
```



```
else //si la fila actual es la primera
{
    EncenderFila(7); //Encender la ultima fila
    EncenderFila(0); //Encender la primer fila
    delay(100);
    ApagarColumna(1); //Apagar la segunda fila
    FilaActual = 7; //Asignar la fila actual a la
nueva posición
}
}
else //Se repite todo para el eje X
{
    if(x > Xreposo+1)
    {
        if(ColumnaActual<=5)
        {
            EncenderColumna(ColumnaActual+2);
            delay(100);
            ApagarFila(ColumnaActual);
            ColumnaActual = ColumnaActual + 1;
        }
        else
        {
            EncenderColumna(0);
            EncenderColumna(7);
            delay(100);
            ApagarFila(6);
            ColumnaActual = 0;
        }
    }
    else
    {
        if( x < Xreposo-1)
        {
            if(ColumnaActual>=1)
            {
                EncenderColumna(ColumnaActual-1);
                delay(100);
                ApagarFila(ColumnaActual+1);
                ColumnaActual = ColumnaActual - 1;
            }
            else
            {
                EncenderColumna(7);
                EncenderColumna(0);
                delay(100);
                ApagarFila(1);
                ColumnaActual = 7;
            }
        }
    }
}
```



```
    }

    while((x == Xreposo && x == Xreposo) && (y == Yreposo &&
y == Yreposo)) //Mientras el joystick este en reposo, encender la fila y
columna actual y seguir leyendo los valores de entrada hasta que cambien
    {
        ApagarColumna(ColumnaActual);
        ApagarColumna(ColumnaActual+1);
        ApagarFila(FilaActual+1);
        ApagarFila(FilaActual);
        delay(200);
        EncenderColumna(ColumnaActual);
        EncenderColumna(ColumnaActual +1);
        EncenderFila(FilaActual);
        EncenderFila(FilaActual + 1);
        y= analogRead(A5);
        x= analogRead(A4);
    }
}

void EncenderFila(int fila)
{
    digitalWrite(Filas[fila],HIGH);
}

void EncenderColumna(int col)
{
    digitalWrite(Columnas[col],LOW);
}

void ApagarFila(int fila)
{
    digitalWrite(Filas[fila],LOW);
}

void ApagarColumna(int col)
{
    digitalWrite(Columnas[col],HIGH);
}
```