

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DIVISIÓN DE INGENIERÍA ELÉCTRICA

Lab. Dispositivos de Almacenamiento y de E/S.

PRÁCTICA °N: 8

**“Utilizando Arduino como dispositivo
de entrada y salida.”**

28/Mayo/2020

Grupo: 3



Moreno Madrid María Guadalupe

PRÁCTICA °N: 8

“Utilizando Arduino como dispositivo de entrada y salida.”

Introducción:

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo (IDE, por sus siglas en Ingles), diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

La plataforma Arduino se ha vuelto muy popular entre personas que acaban de empezar con la electrónica, por una buena razón, a diferencia de la mayoría de las tarjetas de circuitos programables, el Arduino no necesita una pieza separada de hardware (llamado un programador) para cargar nuevo código al microcontrolador, simplemente se usa un cable USB. El Arduino IDE utiliza una versión simplificada de C ++, por lo que es más fácil aprender a programar.

El hardware y el software de Arduino fue diseñado para artistas, diseñadores, aficionados, hackers novatos, y cualquier persona interesada en la creación de proyectos electronicos. Arduino puede interactuar con botones, LEDs, motores, altavoces, unidades de GPS, cámaras, internet, e incluso smartphones. Esta flexibilidad combinada con el hecho de que el software de Arduino es gratuito, las tarjetas de hardware son bastante barato, y tanto el software como el hardware son fáciles de aprender ha llevado a una gran comunidad de usuarios que han contribuido con código libre para una base de proyectos didácticos. Por esto, el Arduino se puede utilizar como el cerebro detrás de casi cualquier proyecto de electrónica.

Objetivos:

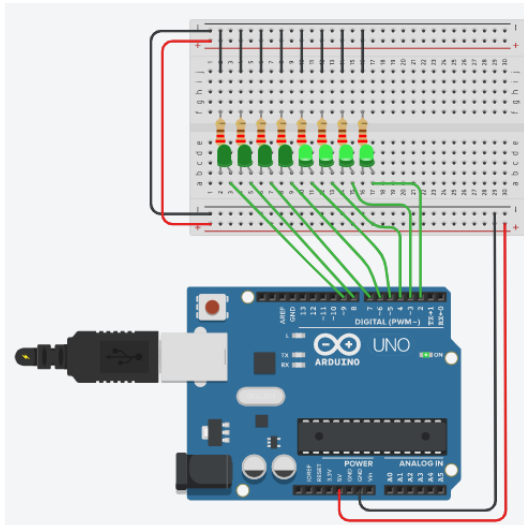
1. Conocer las partes y configuración de la tarjeta de desarrollo Arduino UNO, así como aprender su funcionamiento básico.
2. Conocer y emplear la tarjeta de desarrollo Arduino UNO, como medio sencillo y poderoso, para el diseño de interfaces (entrada /salida) y el control de dispositivos externos

Material:

- Cable USB A-B, para conectar la tarjeta Arduino
- Tarjeta Arduino UNO.
- 1 Display 7 segmentos.
- 10 Resistencias 330 ohms.
- 10 Leds.
- Dip switch 4 posiciones.
- 4 Resistencias 1 Kohm.
- Cables de conexión para protoboard.

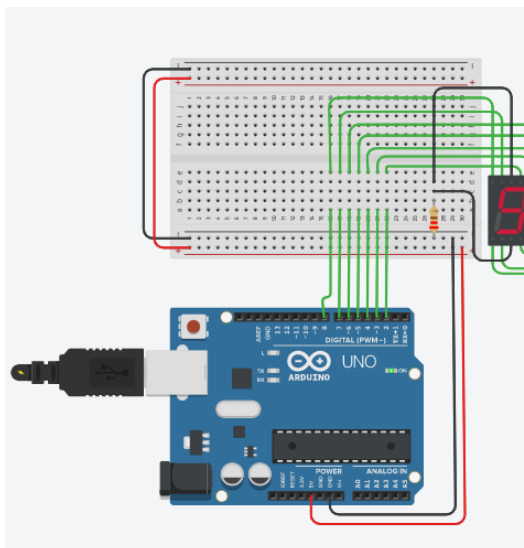
Desarrollo:

1. Demostrar el funcionamiento de las terminales digitales, en la salida del puerto deberá presentar el corrimiento de 8 bits de datos de 0 a 255.



```
1 //Definicion de constantes para la asignacion de pines y tiempo.
2 const int pinesSalida[8] = { 2, 3, 4, 5, 6, 7, 8, 9 };
3 const int tiempoRetardo = 600;
4
5 void setup() {
6     //Asignacion de pines de salida.
7     for(int i = 0; i < 8; i++) {
8         pinMode(pinesSalida[i], OUTPUT);
9     }
10 }
11
12 void loop() {
13     //Codigo del contador binario de 8 bits.
14
15     for (int i = 0; i < 256; i++)
16     {
17         for (int j = 0; j < 8; j++)
18         {
19             if ( ( (i >> j) & 1 ) == 1 )
20                 digitalWrite(pinesSalida[j], HIGH);
21             else digitalWrite(pinesSalida[j], LOW);
22         }
23         delay(tiempoRetardo);
24     }
25 }
```

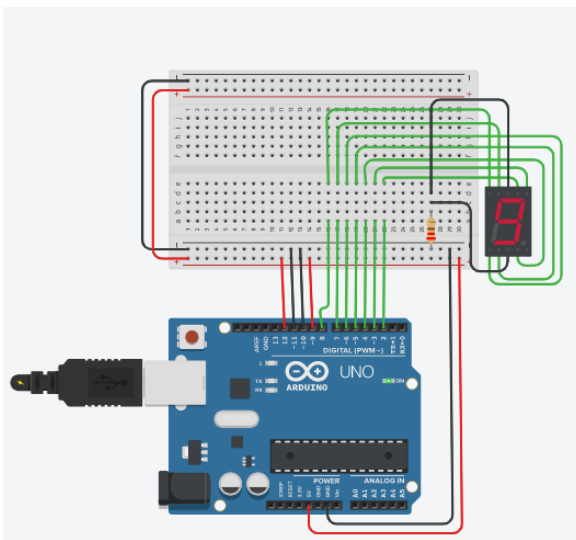
2. Contador de 0 a 9 conectando el display de 7 segmentos.



```
1 //Definicion de constantes para la asignacion de pines y tiempo.
2 const int pinesSalida[8] = { 2, 3, 4, 5, 6, 7, 8, 9 };
3 const int tiempoRetardo = 1000;
4 const byte numerosDisplay[10] = { 0b0111111, //0
5                                   0b0000110, //1
6                                   0b1011011, //2
7                                   0b1001111, //3
8                                   0b1100110, //4
9                                   0b1101101, //5
10                                  0b1111101, //6
11                                  0b0000111, //7
12                                  0b1111111, //8
13                                  0b1101111}; //9
14
15 void setup() {
16     //Asignacion de pines de salida.
17     for(int i = 0; i < 8; i++) {
18         pinMode(pinesSalida[i], OUTPUT);
19     }
20 }
21
22 void loop() { //Codigo del contador del 0 al 9
23     for(int i = 0; i < 10; i++) {
24         prenderSegmentos(i);
25         delay(tiempoRetardo);
26     }
27 }
28 //Funcion para desplegar el numero en el display
29 void prenderSegmentos(int numero) {
30     byte numeroBinario = numerosDisplay[numero];
31     for (int i = 0; i < 7; i++) {
32         int bit = bitRead(numeroBinario, i);
33         digitalWrite(pinesSalida[i], bit);
34     }
35 }
```

- Utilizando Arduino demostrar el funcionamiento del programa diseñado en el inciso (3) del previo, para observar la salida deberá conectar un Display de 7 segmentos en las terminales configuradas como salidas, y para la entrada introducir valores digitales utilizando los pines configurados como entradas, producir la salida según la tabla que se muestra en el inciso (3) del previo, para ambos alumnos, por lo cual deberá demostrar 3 entradas por alumno.

ENTRADA				SALIDA	
S3	S2	S1	S0	Display 7 segmentos	
1	0	0	1	9
0	1	0	1	5
1	0	0	0	8



```

1 //Definicion de constantes para la asignacion de pines y tiempo.
2 const int pinesSalida[8] = { 2, 3, 4, 5, 6, 7, 8, 9 };
3 const int pinesEntrada[4] = { 9, 10, 11, 12 };
4
5 //Definicion de la tabla para el despliegue en un display de 7
6 //segmentos con catodo comun
7 const byte numerosDisplay[10] = { 0b0111111, //0
8                                     0b0000110, //1
9                                     0b1011011, //2
10                                    0b1001111, //3
11                                    0b1100110, //4
12                                    0b1101101, //5
13                                    0b1111101, //6
14                                    0b0000111, //7
15                                    0b1111111, //8
16                                    0b1101111}; //9
17
18 //Variable que va a guardar el numero que se va a leer
19 int numero = 0;
20
21 void setup() {
22     //Asignacion de pines de salida.
23     for(int i = 0; i < 8; i++) {
24         pinMode(pinesSalida[i], OUTPUT);
25     }
26     //Asignacion de pines de entrada
27     for(int i = 0; i < 4; i++) {
28         pinMode(pinesEntrada[i], INPUT);
29     }
30 }
31 void loop() { //Codigo del contador del 0 al 9
32     /*Entradas binarias*/
33     int x = digitalRead(pinesEntrada[0]);
34     int y = digitalRead(pinesEntrada[1]);
35     int z = digitalRead(pinesEntrada[2]);
36     int w = digitalRead(pinesEntrada[3]);
37
38     if(x == HIGH && w == HIGH){
39         prenderSegmentos(9);
40     }
41     else if(x == HIGH && z == HIGH){
42         prenderSegmentos(5);
43     }
44     else if(x == LOW && z == HIGH){
45         prenderSegmentos(8);
46     }
47     else{
48         prenderSegmentos(0);
49     }
50 }
51 //Funcion para desplegar el numero en el display
52 void prenderSegmentos(int numero) {
53     byte numeroBinario = numerosDisplay[numero];
54     for (int i = 0; i < 7; i++) {
55         int bit = bitRead(numeroBinario, i);
56         digitalWrite(pinesSalida[i], bit);
57     }
58 }

```

Conclusiones:

El microcontrolador de Arduino esta diseñado para trabajar como un controlador de periféricos de entrada y salida. Es sumamente didáctico, entendible y fácil de programar.

Referencias:

- Arduino en Tinkercad #1: Creando una cuenta de usuario en tinkercad.com
["https://www.youtube.com/watch?v=UrLkigcU1G4&feature=youtu.be"](https://www.youtube.com/watch?v=UrLkigcU1G4&feature=youtu.be).