



Dirección General de Asuntos  
del Personal Académico

# MÓDULO 7-SEGMENTOS

MANUAL DE USUARIO

V2.3

Miguel Ángel Bañuelos Saucedo, ICAT  
Milagros Pacheco Castañeda, ENP-5  
Rebeca Guillermina Villegas Salas, ENP-7

Proyecto PAPIME PE101620

Desarrollo de material didáctico para la asignatura de Informática Aplicada a  
la Ciencia y la Industria

Universidad Nacional Autónoma de México

Dirección General de Asuntos del Personal Académico

Proyecto PAPIME PE101620

Ciudad de México, 2021



Esta obra está licenciada bajo la Licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional.

Para ver una copia de esta licencia, visite:

<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.es>

O envíe una carta a:

Creative Commons

PO Box 1866

Mountain View, CA 94042, USA.

## Contents

INTRODUCCIÓN.....	4
1. DESCRIPCIÓN.....	5
2. FUNCIONAMIENTO.....	8
3. ACTIVIDADES EXPERIMENTALES .....	12
3.1. Manejo de arreglos .....	12
3.2. Manejo de ciclos FOR.....	12
3.3. Contador de eventos.....	12
3.4. Conversión binario a decimal (0, 1, ..., 9).....	12
3.5. Conversión de binario a hexadecimal (0, 1, ..., F) .....	13
3.6. Control de intensidad.....	13
3.7. Dado electrónico .....	13
3.8. Ruleta .....	13
3.9. Cerradura electrónica.....	14
ANEXO A. Programa para verificar la operación del módulo 7-segmentos.....	14
ANEXO B. Diagrama esquemático.....	16
ANEXO C. Lista de partes.....	16

## INTRODUCCIÓN

Un display de siete segmentos es un elemento útil para visualizar información numérica. Al realizar ejercicios de programación con una tarjeta Arduino y que involucran este display, se requiere emplear parte del tiempo en realizar la conexión de ocho resistencias. Con la finalidad de simplificar el uso de este display se ha diseñado un Módulo 7-Segmentos que contiene todas las conexiones necesarias y se inserta fácilmente como un *shield* en una tarjeta Arduino UNO o tarjetas compatibles (ver Figura 1). El módulo cuenta con cuatro botones de presión que pueden ser usados en conjunto con el display de 7 segmentos.

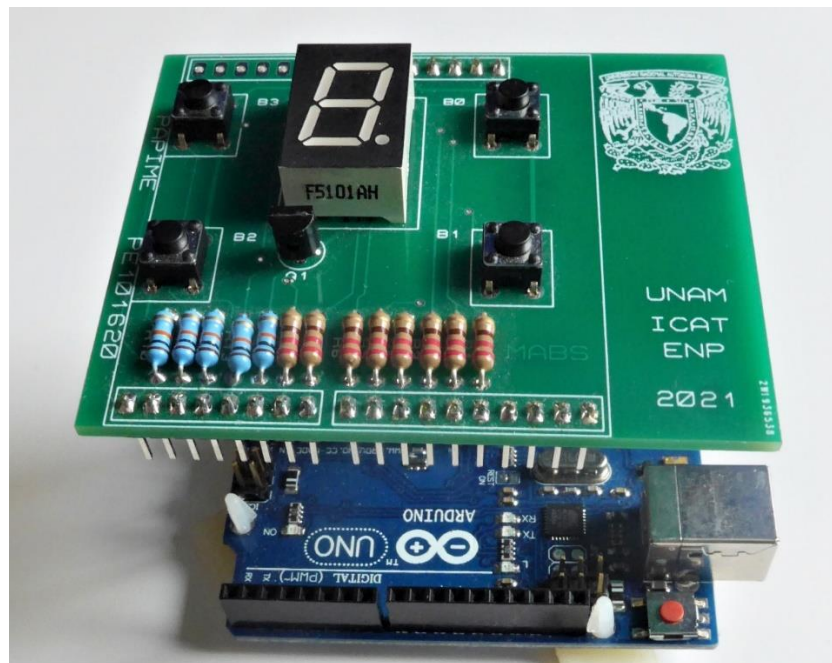


Figura 1. El módulo 7-segmentos desarrollado, antes de ser insertado en una placa Arduino.

Este manual describe los componentes que integran el módulo 7-segmentos, su diagrama de conexiones y el procedimiento para su operación. Está pensado en lectores con un conocimiento básico del manejo de una tarjeta Arduino, diagramas de conexión y componentes electrónicos.

## 1. DESCRIPCIÓN

El módulo 7-Segmentos está orientado a la práctica y aprendizaje de elementos básicos de la programación, tales como: sistema binario, sentencias condicionales, ciclos iterativos, manejo de arreglos, y control de líneas digitales de entrada y salida. Adicionalmente permite aplicar señales PWM para control de la intensidad luminosa del display.

Los elementos principales del módulo 7-Segmentos son: un display de 7 segmentos cátodo común, cuatro botones de presión (B0, B1, B2 y B3) y un transistor (ver Figura 2). El transistor se utiliza para activar el encendido del display y permite controlar la intensidad de encendido si se utiliza una señal PWM (pulsewidth modulation). La distribución de los botones de presión permite que un usuario pueda operarlos de manera simultánea y con relativa comodidad.

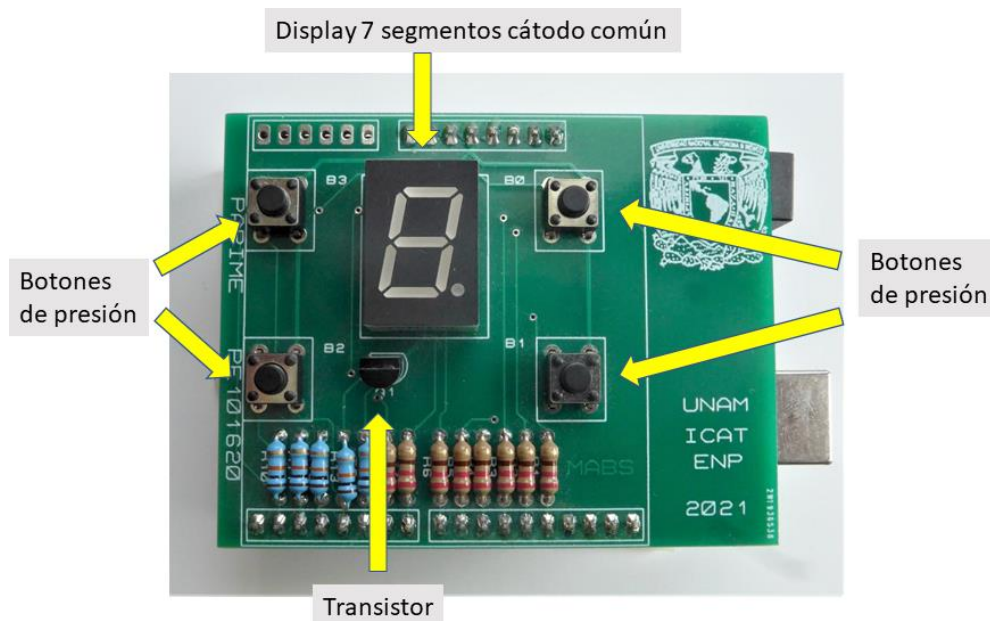


Figura 2. Componentes principales del módulo 7-Segmentos.

En la Figura 3, se muestra el diagrama de conexiones del módulo 7-segmentos donde destaca el transistor N (tipo NPN) que se utiliza para activar la conexión cátodo común del display LED de siete segmentos. Aparecen también los cuatro botones etiquetados B3, B2, B1, B0, en referencia al bit 3, bit 2, bit 1 y bit 0 de un número binario.

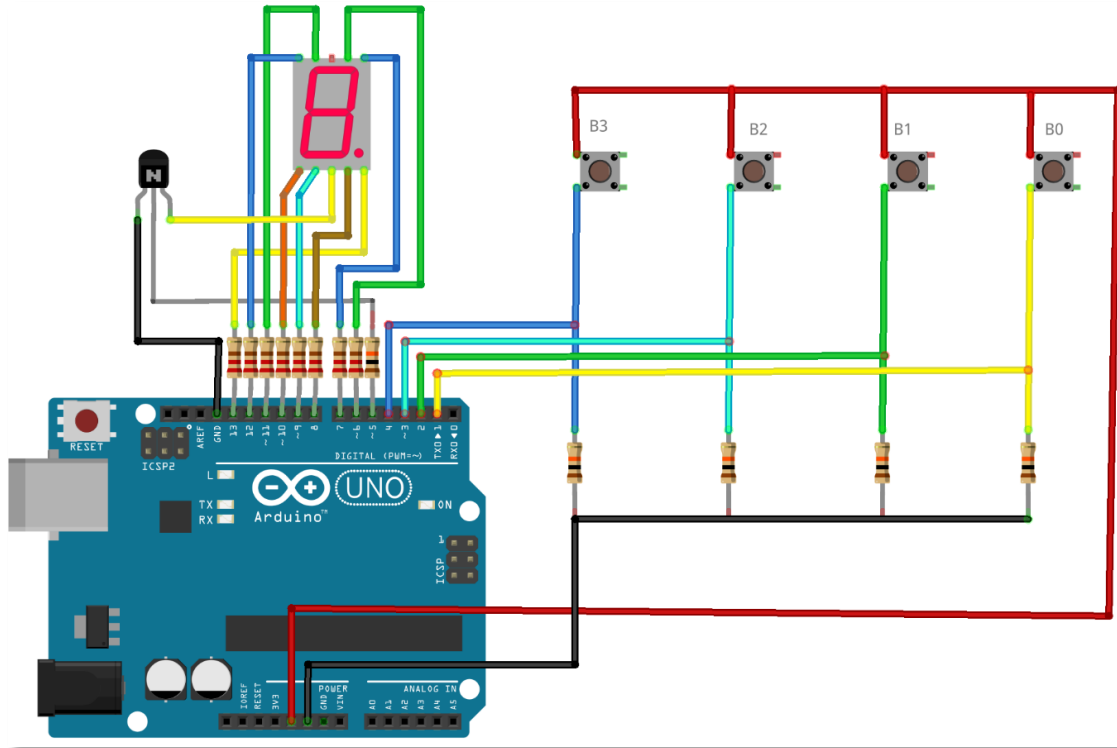


Figura 3. Diagrama de conexiones del módulo 7-segmentos.

En la Figura 4 se muestran en detalle los puntos de conexión del display LED de siete segmentos donde PD=Punto decimal y CC=cátodo común. Las dos terminales CC están conectadas entre sí, por lo que se puede utilizar cualquiera de ellas de forma indistinta.

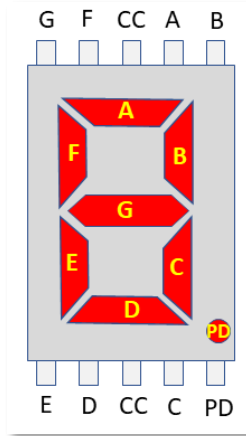


Figura 4. Conexiones del display LED de siete segmentos.

En la Figura 5 se indican las conexiones del transistor NPN utilizado (BC547C) para activar el display LED de siete segmentos donde E=emisor, B=base, C=colector. Observe que la terminal E está conectada a tierra, la terminal B al pin digital 5 del Arduino y la terminal C está conectada a la terminal CC del display LED de siete segmentos. Con esta disposición, al enviar un voltaje alto (HIGH) al PIN D5 del Arduino, se activa el encendido del display, por lo que será posible encender cualquier

segmento al enviar simultáneamente un voltaje alto (HIGH) al PIN correspondiente del mismo (A, B, C, D, E, F, G, y punto decimal).

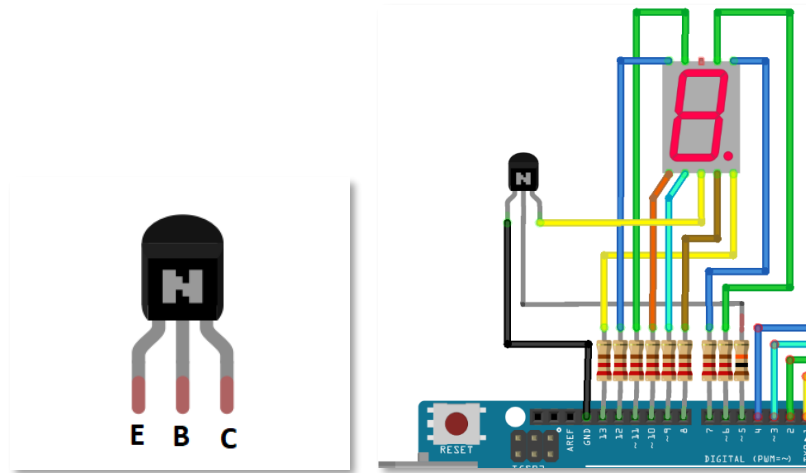


Figura 5. Conexión del transistor NPN a la tarjeta Arduino UNO.

En la Tabla 1 se especifican los pines de conexión de la tarjeta Arduino UNO para cada uno de los segmentos del display LED y los 4 botones de presión. Observe que a través del pin digital 5 (D5) el dispositivo tiene la posibilidad de controlar la intensidad luminosa del display LED de siete segmentos, dado que este pin tiene la capacidad de producir una señal de PWM.

Tabla 1. Asignación de pines digitales del Arduino UNO

Pin Arduino UNO	Conexión
D13	Punto decimal
D12	Segmento G
D11	Segmento F
D10	Segmento E
D9	Segmento D
D8	Segmento C
D7	Segmento B
D6	Segmento A
D5	Activación del display LED/ Control de luminosidad PWM
D4	Botón bit 3
D3	Botón bit 2
D2	Botón bit 1
D1	Botón bit 0

## 2. FUNCIONAMIENTO

En un display LED de siete segmentos, la terminal CC (cátodo común) se conecta a tierra (GND) para completar la conexión de los LED que se deben conectar a los pines de la tarjeta Arduino mediante su respectiva resistencia de limitación de corriente (p. ej.  $220 \Omega$ ), ver la Figura 6. En el módulo 7-segmentos, esta conexión se hace por medio del transistor, el cual funciona como interruptor. Es decir, cuando se aplica voltaje a la terminal B el transistor se comporta como un interruptor cerrado (entre las terminales C y E), ver la Figura 7A. Por otro lado, cuando no se aplica voltaje a la terminal B del transistor, este se comporta como un interruptor abierto entre sus terminales C y E, ver la Figura 7B. La terminal B del transistor está conectada al pin 5 de la placa Arduino, por lo tanto, para activar el display LED es necesaria la instrucción:

```
digitalWrite(5, HIGH);
```

además de las instrucciones necesarias para activar cada uno de los segmentos. Una representación del funcionamiento del transistor se muestra en la Figura 7.

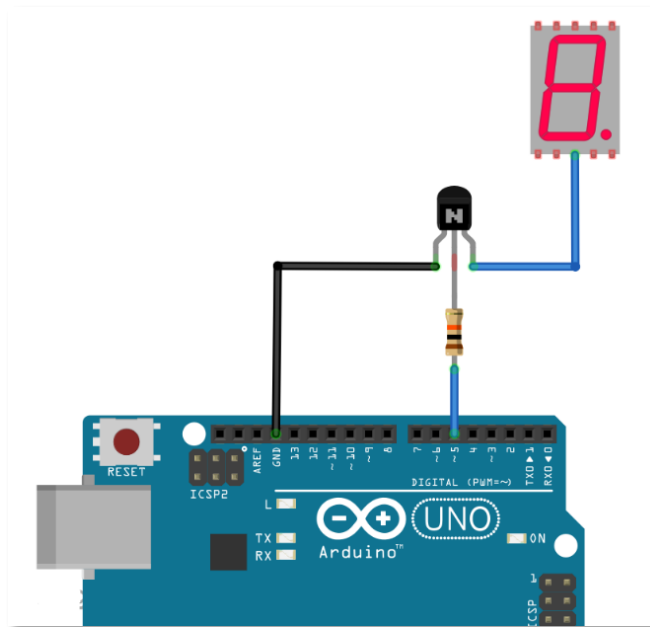


Figura 6. Detalle de la conexión del transistor al cátodo común del display LED.



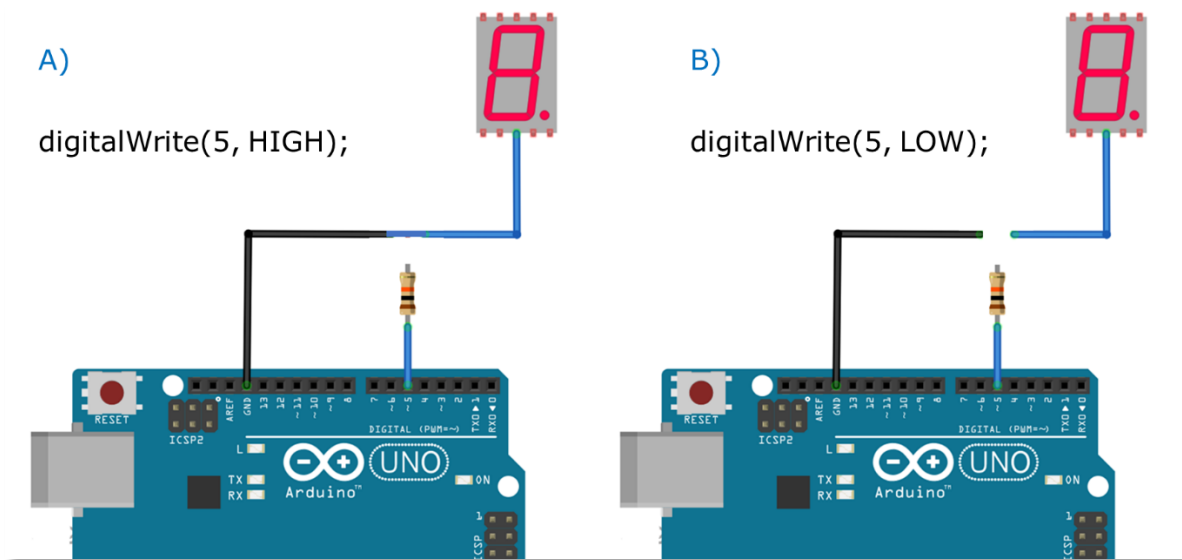


Figura 7. Representación del funcionamiento del transistor como interruptor. A) el transistor está encendido. B) el transistor está apagado.

El transistor conectado al cátodo común del display LED también permite controlar la intensidad del display mediante la función *analogwrite* a través del envío de una señal PWM al pin 5 del Arduino, lo cual le da versatilidad al módulo.

Los botones de presión, etiquetados como B3, B2, B1 y B0, están conectados de forma tal que al presionar un botón, la terminal correspondiente recibe un voltaje alto (“1” lógico) y al no presionarlo, se genera un voltaje bajo (“0” lógico). Esta operación se muestra en la Figura 8.

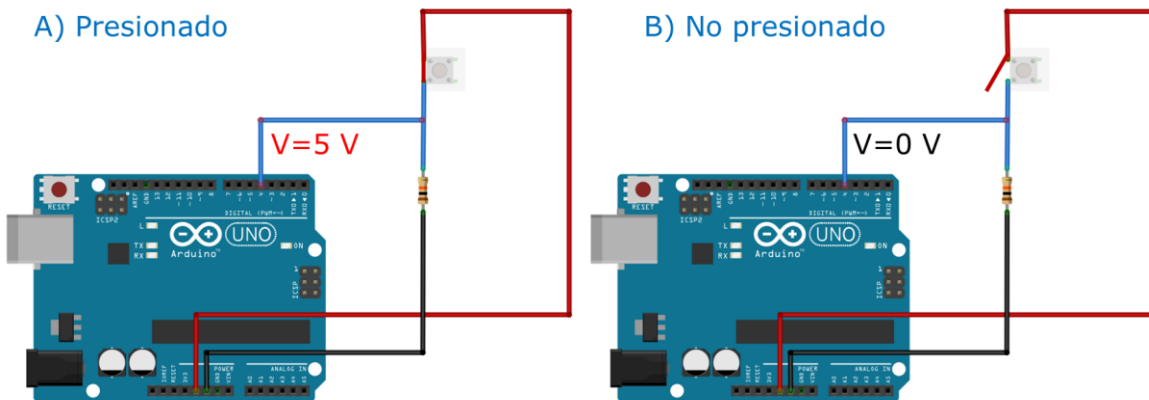


Figura 8. Representación del funcionamiento del botón de presión.

La disposición de los cuatro botones se muestra en la Figura 9. Se prefirió ubicarlos alrededor del display LED de siete segmentos para hacer posible que un solo usuario pueda accionarlos simultáneamente y sin que se estorben sus dedos.

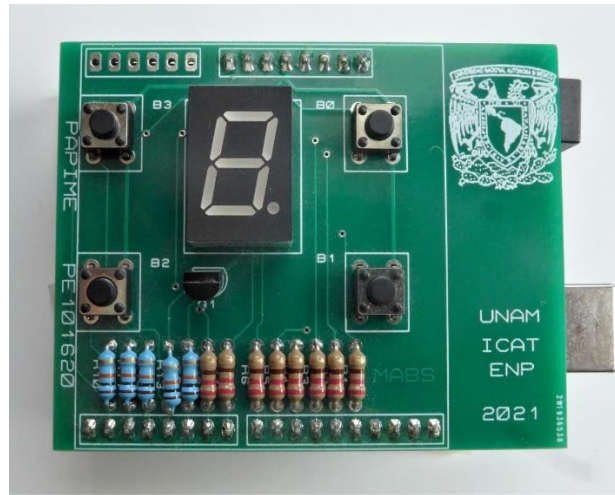


Figura 9. Disposición de los botones de presión en el módulo 7-segmentos.

En la Tabla 2 se muestra un código de ejemplo, el cual enciende de manera consecutiva cada uno de los siete segmentos (del A al G) y el punto decimal.

Tabla 2. Programa de prueba secuencial de los segmentos y el punto decimal.

```

/*
  _7Seg_single_segment_Test_dot
  Programa que prueba el uso de un Display de 7 segmentos
  El display se maneja con un transistor en D1
  Enciende cada segmento y el PD de manera consecutiva
  (2021) UNAM
*/

// Arreglo con la tabla de segmentos
//          P A B C D E F G pd
const byte numeros[12] = {B01111110, // 0
                          B00110000, // 1
                          B01101101, // 2
                          B01111001, // 3
                          B00110011, // 4
                          B01011011, // 5
                          B00011111, // 6
                          B01110000, // 7
                          B01111111, // 8
                          B01110011, // 9
                          B10000000, // punto decimal
                          };

// Un arreglo con los pines del display
//          A B C D E F G pd
const int pinSegmento[] = {6,7,8,9,10,11,12,13};
const int Q=5; //Pin base del transistor

void setup() {
  // Para configurar los pines de salida
  for(int i=0; i < 8; i++){
    pinMode(pinSegmento[i], OUTPUT);
  }
  pinMode(Q, OUTPUT);
  digitalWrite(Q, HIGH); //Permanece activo el display
}

void loop() {
  for(int i=0; i < 8; i++) {
    digitalWrite(pinSegmento[i],HIGH);
    delay(1000);
    digitalWrite(pinSegmento[i],LOW);
    delay(100);
  }
}

```

### 3. ACTIVIDADES EXPERIMENTALES

El módulo 7-segmentos puede ser utilizado para un sinnúmero de actividades de programación, algunas de las cuales se mencionan a continuación.

#### 3.1. Manejo de arreglos

La programación del display se simplifica si se utiliza un arreglo para definir los segmentos que deberán encender.

#### 3.2. Manejo de ciclos FOR

Se puede programar un contador decimal (0, 1, ..., 9) ascendente, descendente o ascendente-descendente.

#### 3.3. Contador de eventos

Utilizando un par de botones, se puede realizar un contador decimal (0, 1, ..., 9) que se incremente o decremente al presionar alguno de los dos botones.

#### 3.4. Conversión binario a decimal (0, 1, ..., 9)

Se puede realizar un programa que utilice los cuatro botones para generar un código binario de cuatro bits, y desplegar el resultado encendiendo el dígito decimal correspondiente. Se debe recordar que un botón no presionado se lee como un "0" lógico y un botón presionado se lee como un "1" lógico.

B3	B2	B1	B0	Decimal
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

### 3.5. Conversión de binario a hexadecimal (0, 1, ..., F)

De manera similar al caso anterior se puede realizar un programa que utilice los cuatro botones para generar un dígito hexadecimal.

### 3.6. Control de intensidad

Utilizando el modo PWM se puede controlar la intensidad de encendido del display LED. El control de intensidad puede ir variando de manera autónoma con una secuencia de instrucciones, mediante un ciclo FOR, o al presionar uno o dos botones. En el caso de dos botones se puede utilizar uno para incrementar el brillo y el otro para decrementarlo. Se recomiendan pasos de 20 unidades en el control del PWM.

### 3.7. Dado electrónico

Se puede utilizar un botón y una instrucción para la generación de números pseudoaleatorios *random()* para generar un número del 1 al 6, o alguna otra variante.

### 3.8. Ruleta

Se puede utilizar un botón, y la instrucción de generación de números pseudoaleatorios para generar una ruleta sencilla que encienda en secuencia los segmentos A, B, C, D, E y F, vayan disminuyendo su velocidad hasta detenerse en alguno de ellos (ver

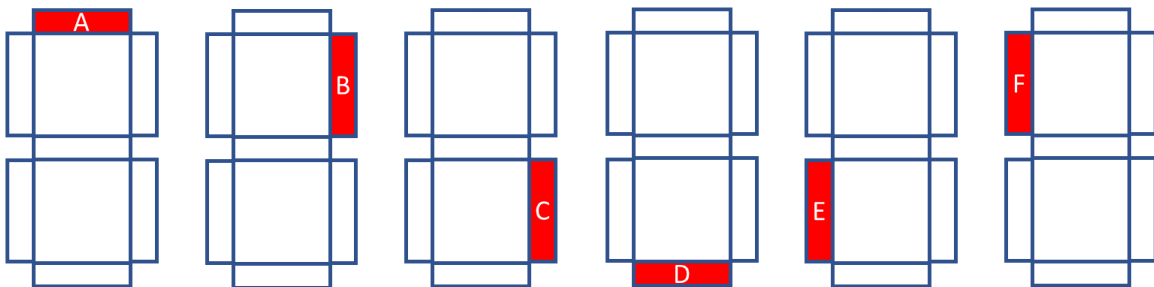


Figura 10). Utilizando la generación de una señal de PWM es posible incluso variar la luminosidad del LED de la ruleta conforme va “girando”.

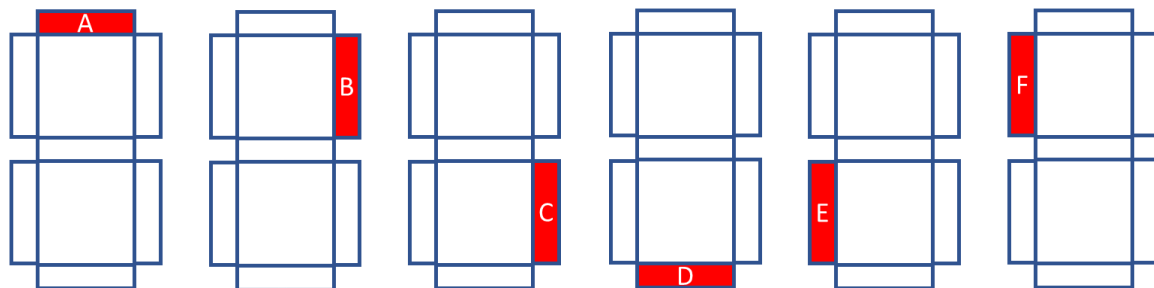


Figura 10. Secuencia de encendido en la ruleta.

### 3.9. Cerradura electrónica

Se pueden utilizar los botones para detectar una secuencia de presión que active o no una cerradura electrónica simulada. En este caso, al presionar la secuencia correcta pueden encenderse los segmentos que formen una letra A, si la secuencia es incorrecta puede aparecer una E y si la cerradura permanece cerrada una letra C.

## ANEXO A. Programa para verificar la operación del módulo 7-segmentos

El siguiente programa muestra de forma sucesiva los dígitos del 0 al 9 y luego enciende el punto decimal. Lo anterior lo ejecuta dos veces. A continuación, se queda esperando a que el usuario presione alguno de los cuatro botones. Al presionar algún botón, se despliega un dígito, de acuerdo con la siguiente tabla:

Botón	Dígito que se enciende
B0	0
B1	1
B2	2
B3	3

```

/*  _7Seg_test_all.ino
   Programa que prueba el uso de un Display de 7 segmentos
   El display se maneja con un transistor en D1
   No está conectado el punto decimal
   UNAM 2021
   */

#define boton0 1
#define boton1 2
#define boton2 3
#define boton3 4
// Arreglo con la tabla de segmentos
//
//          ABCDEFGP
const byte numeros[11] = {B11111100, // 0
                          B01100000, // 1
                          B11011010, // 2
                          B11110010, // 3
                          B01100110, // 4
                          B10110110, // 5
                          B00111110, // 6
                          B11100000, // 7
                          B11111110, // 8
                          B11100110, // 9
                          B00000001 // 10=PUNTO DECIMAL
                          };
// Un arreglo con los pines del display
//          A B C D E F G PD
const int pinSegmento[] = {6,7,8,9,10,11,12, 13};
const int Q=5; //Pin base del transistor

void setup() {
  // Para configurar los pines de salida
  for(int i=0; i <= 7; i++){
    pinMode(pinSegmento[i], OUTPUT);
  }
  pinMode(Q, OUTPUT);
  digitalWrite(Q, HIGH);
  pinMode(boton0, INPUT);
  pinMode(boton1, INPUT);
  pinMode(boton2, INPUT);
  pinMode(boton3, INPUT);
}

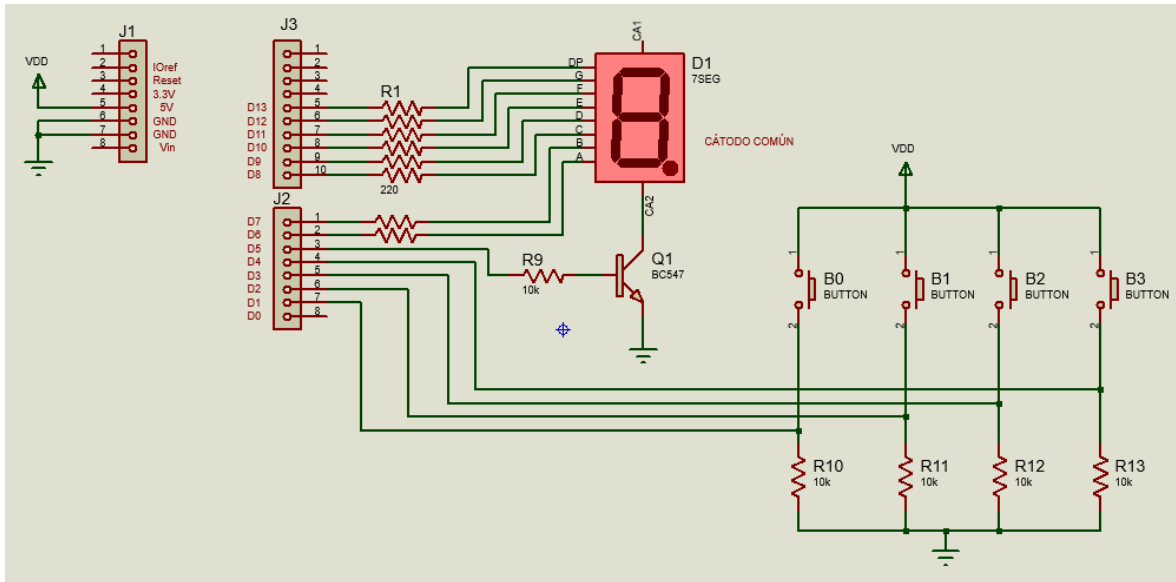
void loop() {
  for(int i=0; i <= 10; i++) {
    prendeDigito(i);
    delay(500);
  }
  for(int i=0; i <= 10; i++) {
    prendeDigito(i);
    delay(500);
  }
  while(1){
    if(digitalRead(boton0))
      prendeDigito(0);
    if(digitalRead(boton1))
      prendeDigito(1);
    if(digitalRead(boton2))
      prendeDigito(2);
    if(digitalRead(boton3))
      prendeDigito(3);
  }
}

// ***** Definición de funciones *****

void prendeDigito(int numero) {
  boolean prendido;
  for(int segmento = 0; segmento <= 7; segmento++) {
    prendido = bitRead(numeros[numero], 7-segmento);
    digitalWrite( pinSegmento[segmento], prendido);
  }
}

```

ANEXO B. Diagrama esquemático



ANEXO C. Lista de partes

Número de parte	Cantidad	Descripción
J1, J2	2	Header 8 pines, macho
J3	1	Header 10 pines, macho
D1	1	Display LED 7 segmentos cátodo común (F5101AH)
Q1	1	BC547 transistor NPN
B0-B3	4	Push button 6 mm x 6 mm
R1-R8	8	Resistencia carbón 220 $\Omega$ 5% 1/2 W
R9-R13	5	Resistencia carbón 10 k $\Omega$ 5% 1/2 W
Placa 7-segmentos	1	Circuito impreso doble cara