

# Manual de prácticas

Materia:

## Sistemas programables

Instituto tecnológico superior del sur de  
Guanajuato.

Carrera:

Ingeniería en sistemas computacionales

Semestre: 7°

Autor: M.C. Jeziel Vázquez Nava

22 de marzo de 2022

## INDICE

[PRACTICA 4: DISPLAY 7 SEGMENTOS](#)

[PRACTICA 5: CHARLIEPLEXING](#)

## PRACTICA 4: DISPLAY 7 SEGMENTOS

---

### OBJETIVO:

Conocer cuales son los tipos de display (Visualizador) de 7 segmentos y utilizarlos correctamente después de esta práctica el alumno será capaz de:

- ✓ Utilizar correctamente los display de 7 segmentos.

---

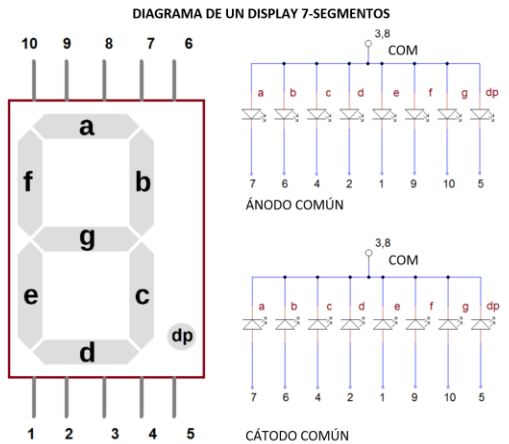
### RECURSOS MATERIALES Y EQUIPO:

- ❖ Hojas de datos del microcontrolador
- ❖ Arduino uno R3
- ❖ Resistencias
- ❖ Visualizador de 7 segmentos
- ❖ Pulsadores (Push buttons)
- ❖ IDE Arduino
- ❖ Software Proteus
- ❖ Protoboard

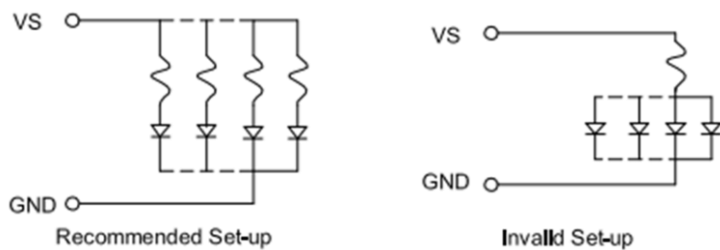
---

### MARCO TEÓRICO:

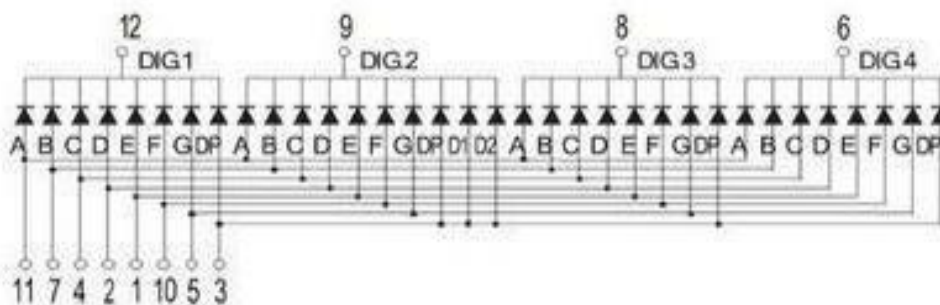
El visualizador (display) de siete segmentos es una forma de representar caracteres en equipos electrónicos. Esta formado de siete segmentos compuestos por LEDs los cuales se pueden encender o apagar individualmente, estos están conectados de forma interna uniando las terminales del mismo tipo (ánodo o cátodo), por lo que existen dos tipos básicos de visualizadores de 7 segmentos conocidos como ánodo común y cátodo común lo cual indica que terminal es la que está unida.



La forma de conectar estos dispositivos es conectar cada terminal por medio de resistencias, en la figura xxxx se muestra la forma recomendada y la no recomendada.



También existen visualizadores de siete segmentos con más dígitos (2, 3, 4, ...), estos normalmente por dígito se encuentra una terminal común ya sea de ánodo o de cátodo, por ejemplo un visualizador de 4 dígitos tendrá cuatro terminales por cada dígito, y las terminales no comunes para cada dígito se unen entre ellas, como se muestra en la figura xxx.



---

## DESCRIPCIÓN DEL DESARROLLO DE LA PRÁCTICA:

Esta práctica consiste en el uso del display de 7 segmentos de 4 dígitos, con el objeto de resolver la problemática descrita a continuación.

Una empresa que administra una cancha de basquetbol quiere implementar un marcador digital para lo cual compraron un display de 7 segmentos de 4 dígitos, los cuales deben representar los marcadores de ambos equipos por lo que los primeros dos dígitos son para representar el marcador de un equipo y los siguientes para el otro equipo, también compro cuatro pulsadores dos para cada marcador, uno debe incrementar en uno la cuenta y el otro debe decrementarla en la misma cantidad, esta empresa lo contrata a usted para que implemente este sistema en su cancha de basquetbol.

El alumno realizara un programa que resuelva la problemática descrita anteriormente y la implementara usando una simulación mediante el software PROTEUS e implementara físicamente.

---

## DESARROLLO:

Una vez que se revisó en clase el funcionamiento básico de un display de siete segmentos y con ayuda del esquema de pines, se procedió a crear la simulación del circuito pertinente para resolver la problemática descrita, obteniendo como resultado:

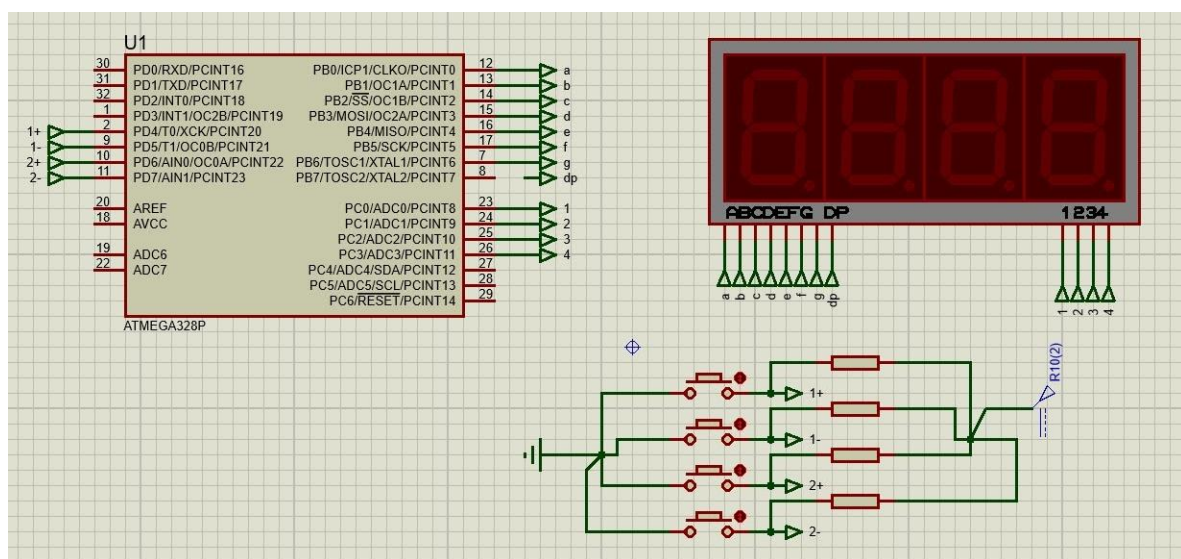


Ilustración 1. Simulación Práctica 4

Tomando como guía las conexiones previamente realizadas, se construyó el circuito de forma física. Como se pudo observar en la Ilustración 6, se omitieron las resistencias en las conexiones de los botones, ya que estas no eran necesarias si se incluía en el código la función INPUT\_PULLUP.

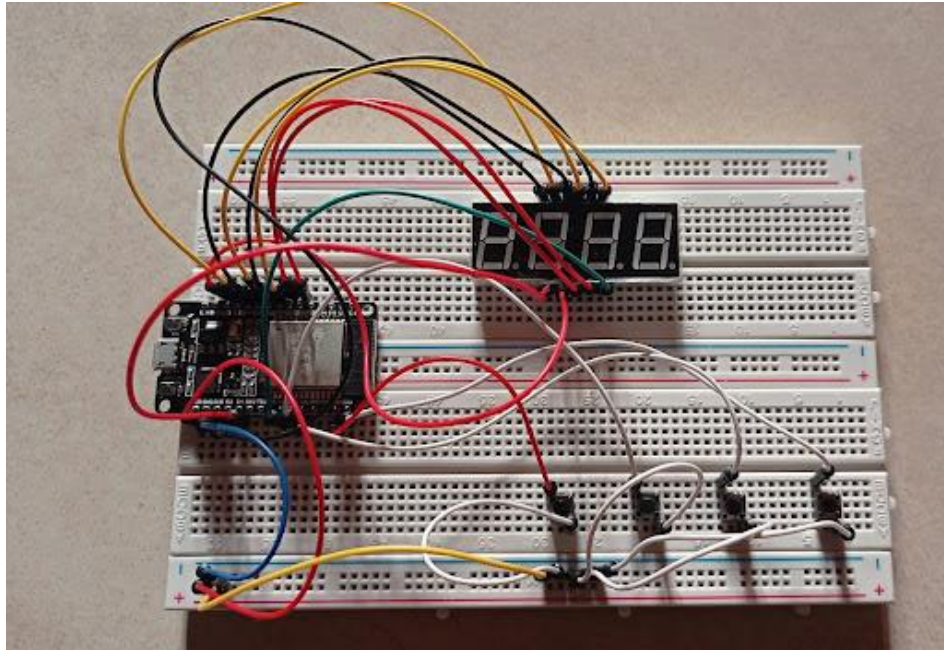


Ilustración 2. Conexiones Físicas Práctica 4

El código usado en este circuito, es el siguiente:

```
volatile int cont1 = 0, cont2 = 0;
const int a = 12, b = 25, c = 32, d = 13, e = 4, f = 14, g = 33;
const int d1 = 15, d2 = 27, d3 = 26, d4 = 18;
const int b11 = 22, b1 = 19, b22 = 21, b2 = 23;

void IRAM_ATTR sum1(){
    cont1++;
}

void IRAM_ATTR res1(){
    cont1--;
}

void IRAM_ATTR sum2(){
    cont2++;
}

void IRAM_ATTR res2(){
    cont2--;
}
```

```

}

void setup() {
  //Leds
  pinMode(a,OUTPUT);
  pinMode(b,OUTPUT);
  pinMode(c,OUTPUT);
  pinMode(d,OUTPUT);
  pinMode(e,OUTPUT);
  pinMode(f,OUTPUT);
  pinMode(g,OUTPUT);
  //Digitos
  pinMode(d1,OUTPUT);
  pinMode(d2,OUTPUT);
  pinMode(d3,OUTPUT);
  pinMode(d4,OUTPUT);
  //Botones
  pinMode(b11,INPUT_PULLUP);
  attachInterrupt(b11,sum1,CHANGE);
  pinMode(b1,INPUT_PULLUP);
  attachInterrupt(b1,res1,CHANGE);
  pinMode(b22,INPUT_PULLUP);
  attachInterrupt(b22,sum2,CHANGE);
  pinMode(b2,INPUT_PULLUP);
  attachInterrupt(b2,res2,CHANGE);
}

void loop() {
  if (cont1>99) cont1 = 0;
  if (cont2>99) cont2 = 0;

  //cont1
  limpiar();
  digito(1);
  num((cont1/10)%10);
  delay(5);
  limpiar();
  digito(2);
  num(cont1%10);
  delay(5);
  //cont2
  limpiar();
  digito(3);
  num((cont2/10)%10);
  delay(5);
  limpiar();

```

```

    digito(4);
    num(cont2%10);
    delay(5);
}

void digito(int x){
    digitalWrite(d1,HIGH);
    digitalWrite(d2,HIGH);
    digitalWrite(d3,HIGH);
    digitalWrite(d4,HIGH);

    switch(x){
        case 1: digitalWrite(d1,LOW); break;
        case 2: digitalWrite(d2,LOW); break;
        case 3: digitalWrite(d3,LOW); break;
        case 4: digitalWrite(d4,LOW); break;
    }
}

void num(int x){
    switch(x){
        case 1: uno(); break;
        case 2: dos(); break;
        case 3: tres(); break;
        case 4: cuatro(); break;
        case 5: cinco(); break;
        case 6: seis(); break;
        case 7: siete(); break;
        case 8: ocho(); break;
        case 9: nueve(); break;
        default: cero(); break;
    }
}

void cero(){
    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(d,HIGH);
    digitalWrite(e,HIGH);
    digitalWrite(f,HIGH);
    digitalWrite(g,LOW);
}

void uno(){
    digitalWrite(a,LOW);

```



```
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(d,LOW);
    digitalWrite(e,LOW);
    digitalWrite(f,LOW);
    digitalWrite(g,LOW);
}
```

```
void dos(){
    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,LOW);
    digitalWrite(d,HIGH);
    digitalWrite(e,HIGH);
    digitalWrite(f,LOW);
    digitalWrite(g,HIGH);
}
```

```
void tres(){
    digitalWrite(a,HIGH);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(d,HIGH);
    digitalWrite(e,LOW);
    digitalWrite(f,LOW);
    digitalWrite(g,HIGH);
}
```

```
void cuatro(){
    digitalWrite(a,LOW);
    digitalWrite(b,HIGH);
    digitalWrite(c,HIGH);
    digitalWrite(d,LOW);
    digitalWrite(e,LOW);
    digitalWrite(f,HIGH);
    digitalWrite(g,HIGH);
}
```

```
void cinco(){
    digitalWrite(a,HIGH);
    digitalWrite(b,LOW);
    digitalWrite(c,HIGH);
    digitalWrite(d,HIGH);
    digitalWrite(e,LOW);
    digitalWrite(f,HIGH);
    digitalWrite(g,HIGH);
}
```

```
}  
  
void seis(){  
    digitalWrite(a,HIGH);  
    digitalWrite(b,LOW);  
    digitalWrite(c,HIGH);  
    digitalWrite(d,HIGH);  
    digitalWrite(e,HIGH);  
    digitalWrite(f,HIGH);  
    digitalWrite(g,HIGH);  
}
```

```
void siete(){  
    digitalWrite(a,HIGH);  
    digitalWrite(b,HIGH);  
    digitalWrite(c,HIGH);  
    digitalWrite(d,LOW);  
    digitalWrite(e,LOW);  
    digitalWrite(f,LOW);  
    digitalWrite(g,LOW);  
}
```

```
void ocho(){  
    digitalWrite(a,HIGH);  
    digitalWrite(b,HIGH);  
    digitalWrite(c,HIGH);  
    digitalWrite(d,HIGH);  
    digitalWrite(e,HIGH);  
    digitalWrite(f,HIGH);  
    digitalWrite(g,HIGH);  
}
```

```
void nueve(){  
    digitalWrite(a,HIGH);  
    digitalWrite(b,HIGH);  
    digitalWrite(c,HIGH);  
    digitalWrite(d,LOW);  
    digitalWrite(e,LOW);  
    digitalWrite(f,HIGH);  
    digitalWrite(g,HIGH);  
}
```

```
void limpiar(){  
    digitalWrite(a,LOW);  
    digitalWrite(b,LOW);  
    digitalWrite(c,LOW);
```

```
digitalWrite(d,LOW);  
digitalWrite(e,LOW);  
digitalWrite(f,LOW);  
digitalWrite(g,LOW);  
}
```

---

## RESULTADOS Y CONCLUSIONES:

Después de realizar esta práctica queda comprendido el funcionamiento de un display, independiente del número de dígitos que se tenga, sabemos que se sigue la misma lógica.

Al probar el circuito de forma física, pudimos notar que en algunas ocasiones el contador aumentaba en más de una unidad, esto sucedía porque no teníamos nada que nos ayudara a controlar el voltaje (como son los capacitores); sin embargo, todo lo demás funcionaba de acuerdo a lo planeado.

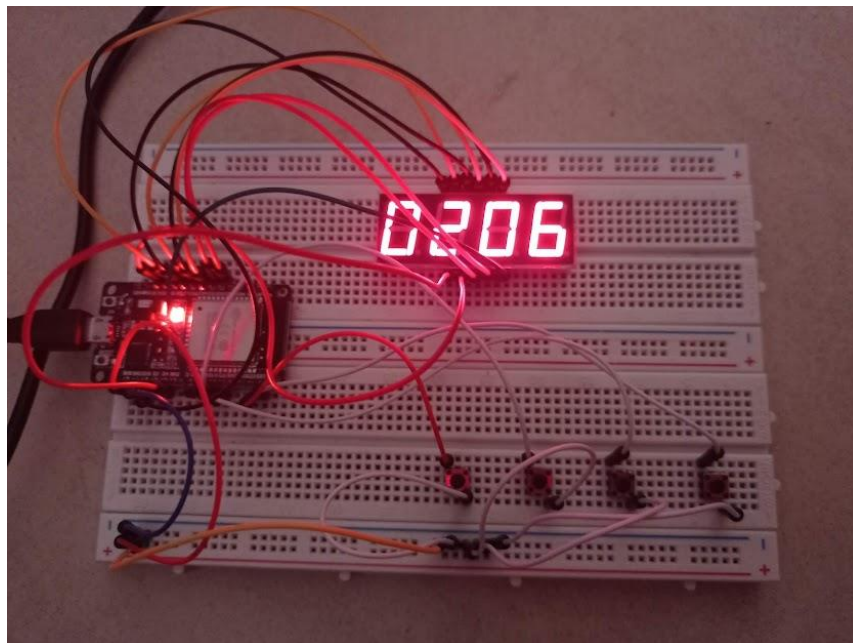


Ilustración 3. Resultado Práctica 4

---

## EVALUACIÓN:

- ¿Cuál es la diferencia entre un display de siete segmentos de ánodo común y uno de cátodo común en cuanto a su conexión eléctrica, alguno es de lógica negada, si es así cual es y por qué?

En un display de ánodo común todas sus líneas se encuentran unidas en su terminal positivo, y para encenderlas es necesario colocar el contacto GND en el terminal de la letra correspondiente.

En el caso de los display de cátodo común, los LEDs de las líneas se encuentran unidos en su terminal negativo, es decir, trabaja con lógica negada. Por lo que el encendido es posible al suministrar tensión en las terminales de las letras correspondientes.

- ¿Por qué no se recomienda poner una resistencia solamente en la terminal común?

Porque podría generar una sobrecarga y dañar el circuito.

- ¿Por qué razón se unen las terminales A, B ,.....G de los dispositivos de 7 segmentos de varios dígitos?

Porque todos los dígitos tienen los mismos leds y su funcionamiento es el mismo, por lo que solo necesitamos indicar el dígito que se utilizara y esto simplifica el proceso.

- ¿Cómo se debe programar un visualizador de 7 segmentos de varios dígitos, que es lo principal a tener en cuenta?

Es importante conocer el esquema de los pines correspondiente, es decir, saber cuáles pertenecen a los leds y con cuales podemos seleccionar el dígito a mostrar. De lo contrario, tendremos problemas en la conexión y su funcionamiento no será el correcto.

- ¿En qué aplicaciones todavía son utilizados estos visualizadores?

Se emplean en relojes digitales, medidores electrónicos, calculadoras básicas, pantallas de electrodomésticos, coches, y muchos otros dispositivos que muestran información numérica.

---

## BIBLIOGRAFÍA:

- ✓ Hoja de Datos del Microcontrolador ATMEGA328P
- ✓ <https://www.arduino.cc/reference/en/>
- ✓ Arduino cookbook Michael Margolis Editorial O'Reilly

## PRACTICA 5: CHARLIEPLEXING

---

### OBJETIVO:

Utilizar correctamente el estado conocido como alta impedancia, así como poder controlar una mayor cantidad de dispositivos de salida con menos recursos, después de esta práctica el alumno será capaz de:

- ✓ Utilizar el estado de alta impedancia
  - ✓ Administrar de mejor forma los recursos del microcontrolador..
- 

### RECURSOS MATERIALES Y EQUIPO:

- ❖ Hojas de datos del microcontrolador
  - ❖ Arduino uno R3
  - ❖ Resistencias (10k, 1k, 220k)
  - ❖ LEDs
  - ❖ Pulsadores (Push buttons)
  - ❖ IDE Arduino
  - ❖ Software Proteus
  - ❖ Protoboard
- 

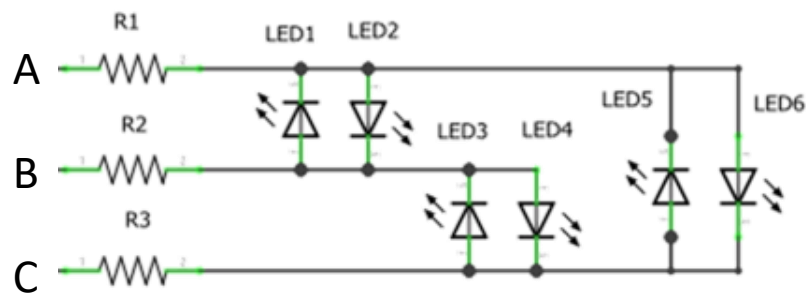
### MARCO TEÓRICO:

Entre los estados posibles se encuentran tres, Entrada, Salida y un tercer estado conocido como alta impedancia (Hi-Z) el cual es como si no hubiera una conexión o una impedancia alta.

Para poder seleccionar alguno de los estados (entrada, salida, Hi-Z) utilizando estos registros se muestra en la siguiente tabla.

DDxn	PORTxn	PUD (MCUCR)	I/O	Pull-up	Comentario
0	0	X	Entrada	NO	Hi-Z
0	1	0	Entrada	SI	Suministra corriente si se conecta a tierra externamente.
0	1	1	Entrada	NO	Hi-Z
1	0	X	Salida	NO	Salida en bajo
1	1	X	Salida	NO	Salida en alto

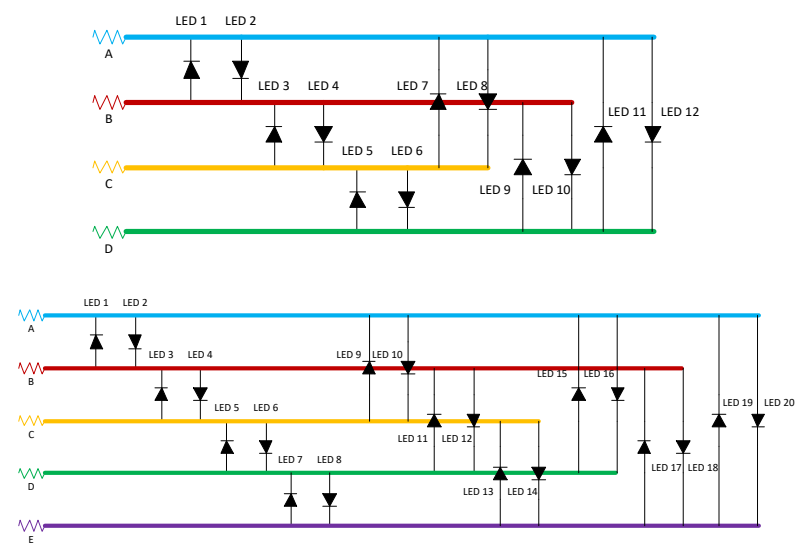
A veces es necesario desconectar el sistema y en esas ocasiones puede ser útil el estado Hi-Z, hay diferentes aplicaciones que utilizan este tipo de estado, una de las más conocida es llamada charliplexing, el nombre se debe a Charlie Allen quien publico el método.



Con esta técnica se pueden controlar seis salidas con tres pines, en la tabla xxx se muestra cómo se puede encender cada led.

	LED 1	LED 2	LED 3	LED 4	LED 5	LED 6
A	LOW	HIGH	Z	Z	LOW	HIGH
B	HIGH	LOW	LOW	HIGH	Z	Z
C	Z	Z	HIGH	LOW	HIGH	LOW

Finalmente, con esta técnica permite controla  $n(n-1)$  salidas digitales con  $n$  pines, por ejemplo, con cuatro pines y cinco pines el arreglo se muestra en la figura xxx



Además de la técnica de charlieplexing existen otras para manejar una mayor cantidad de señales digitales con menos recursos, algunas de ellas son Tucoplexing, Gugaplexing y chipiplexing.

---

### **DESCRIPCIÓN DEL DESARROLLO DE LA PRÁCTICA:**

Esta práctica consiste en el uso del estado conocido como alta impedancia (Hi-Z) mediante el uso de LEDs y pulsadores (push buttons), por medio de interrupciones y el manejo de las señales por medio de los registros de los puertos o por medio de las instrucciones propias del IDE de Arduino, con el objeto de resolver la problemática descrita a continuación.

En una ciudad una empresa que se dedica a la fabricación bandas que se mueven por atracción magnética la cual para moverse necesita activar en serie 6 electroimanes para que la banda se mueva, si quiere que gire en reversa debe cambiar el sentido y si quiere que se detenga debe desactivar los electroimanes, pero para hacer esto solamente dispone de un microcontrolador atmega328 pero solamente le quedan disponibles 6 pines pero 3 deben ser ocupados por pulsadores uno para el giro en sentido horario, otro para el giro en sentido antihorario y otro para detener la banda, cuando está girando un electroimán es encendido durante un tiempo de un segundo.

El alumno realizara una simulación en proteus que resuelva este problema para no comprar otro microcontrolador e implemente una maqueta con leds y tres pulsadores.



## DESARROLLO:

Primero realizamos las conexiones pertinentes en PROTEUS, obtenido como resultado:

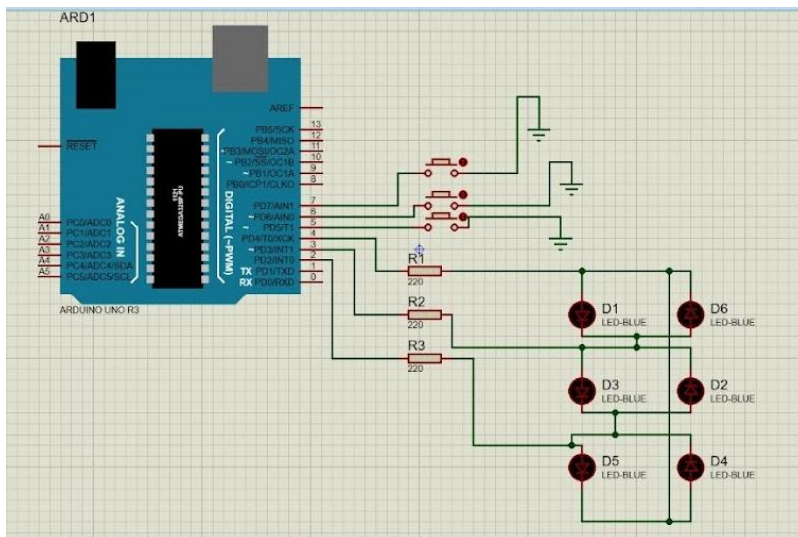


Ilustración 4. Simulación Práctica 5

Tomando como base el circuito anterior, se realizaron las conexiones físicas.

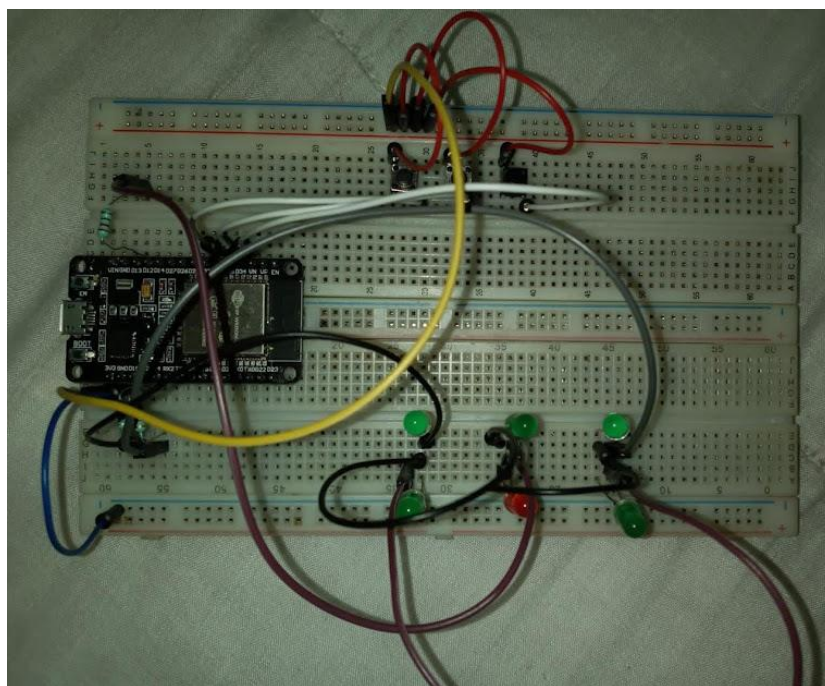


Ilustración 5. Conexiones Físicas Práctica 5

Para lograr el funcionamiento correcto de forma física, hicimos pequeños cambios en el código, por ejemplo, los pines usados; y quedo de la siguiente manera:

```
const int LED_1 = 4;      //fila 1
const int LED_2 = 13;     //fila 2
const int LED_3 = 15;     //fila 3
const int btn1 = 34;      //Botón 1
const int btn2 = 35;      //Botón 2
const int btn3 = 32;      //Botón 3
volatile bool reloj = true;
volatile bool antiRelej = true;
volatile bool detener = true;

void setup()
{
  attachInterrupt(digitalPinToInterrupt(34), bandaRelej, RISING);
  attachInterrupt(digitalPinToInterrupt(35), bandaAntiRelej, RISING);
  attachInterrupt(digitalPinToInterrupt(32), stopBanda, RISING);
}

void loop()
{
  pinMode(btn1, INPUT_PULLUP);
  int estadoBoton = digitalRead(btn1);
  pinMode(btn2, INPUT_PULLUP);
  int estadoBoton2 = digitalRead(btn2);
  pinMode(btn3, INPUT_PULLUP);
  int estadoBoton3 = digitalRead(btn3);

  if(estadoBoton==1){
    reloj=true;
    antiRelej=false;
    detener=false;
    bandaRelej();
  }
  if(estadoBoton2==1){
    antiRelej=true;
    reloj=false;
    detener=false;
    bandaAntiRelej();
  }
  if(estadoBoton3==1){
    detener=true;
    reloj=false;
    antiRelej=false;
    stopBanda();
  }
}
```

```

}
}

void bandaReloj(){
    if(reloj){
        //LED 1
        pinMode(LED_1, OUTPUT);    //fila 1
        digitalWrite(LED_1, LOW);
        pinMode(LED_2, OUTPUT);    //fila 2
        digitalWrite(LED_2, HIGH);
        pinMode(LED_3, INPUT);     //fila 3
        digitalWrite(LED_3, LOW);

        delay(30);

    }
    if(reloj){
        //LED 3
        pinMode(LED_1, INPUT);     //fila 1
        digitalWrite(LED_1, LOW);
        pinMode(LED_2, OUTPUT);    //fila 2
        digitalWrite(LED_2, LOW);
        pinMode(LED_3, OUTPUT);    //fila 3
        digitalWrite(LED_3, HIGH);

        delay(30);
    }
    if(reloj){
        //LED 6
        pinMode(LED_1, OUTPUT);
        digitalWrite(LED_1, HIGH);
        pinMode(LED_2, INPUT);
        digitalWrite(LED_2, LOW);
        pinMode(LED_3, OUTPUT);
        digitalWrite(LED_3, LOW);

        delay(30);
    }
    if(reloj){
        //LED 5
        pinMode(LED_1, OUTPUT);    //fila 1
        digitalWrite(LED_1, LOW);
        pinMode(LED_2, INPUT);     //fila 2
        digitalWrite(LED_2, LOW);
        pinMode(LED_3, OUTPUT);    //fila 3
        digitalWrite(LED_3, HIGH);
    }
}

```

```

delay(30);
}
if(reloj){

    //LED 4
    pinMode(LED_1, INPUT);    //fila 1
    digitalWrite(LED_1, LOW);
    pinMode(LED_2, OUTPUT);    //fila 2
    digitalWrite(LED_2, HIGH);
    pinMode(LED_3, OUTPUT);    //fila 3
    digitalWrite(LED_3, LOW);
    delay(30);
}
if(reloj){

    //LED 2
    pinMode(LED_1, OUTPUT);    //fila 1
    digitalWrite(LED_1, HIGH);
    pinMode(LED_2, OUTPUT);    //fila 2
    digitalWrite(LED_2, LOW);
    pinMode(LED_3, INPUT);    //fila 3
    digitalWrite(LED_3, LOW);

    delay(30);
}
}

void bandaAntiRelej(){
    if(antiRelej){
        //LED 2
        pinMode(LED_1, OUTPUT);    //fila 1
        digitalWrite(LED_1, HIGH);
        pinMode(LED_2, OUTPUT);    //fila 2
        digitalWrite(LED_2, LOW);
        pinMode(LED_3, INPUT);    //fila 3
        digitalWrite(LED_3, LOW);

        delay(30);

    }
    if(antiRelej){
        //LED 4
        pinMode(LED_1, INPUT);    //fila 1
        digitalWrite(LED_1, LOW);
        pinMode(LED_2, OUTPUT);    //fila 2

```

```

digitalWrite(LED_2, HIGH);
pinMode(LED_3, OUTPUT);    //fila 3
digitalWrite(LED_3, LOW);

delay(30);
}
    if(antiRelej){
        //LED 5
pinMode(LED_1, OUTPUT);    //fila 1
digitalWrite(LED_1, LOW);
pinMode(LED_2, INPUT);     //fila 2
digitalWrite(LED_2, LOW);
pinMode(LED_3, OUTPUT);    //fila 3
digitalWrite(LED_3, HIGH);

delay(30);
}
    if(antiRelej){
        //LED 6
pinMode(LED_1, OUTPUT);
digitalWrite(LED_1, HIGH);
pinMode(LED_2, INPUT);
digitalWrite(LED_2, LOW);
pinMode(LED_3, OUTPUT);
digitalWrite(LED_3, LOW);

        delay(30);
    }
    if(antiRelej){
//LED 3
pinMode(LED_1, INPUT);     //fila 1
digitalWrite(LED_1, LOW);
pinMode(LED_2, OUTPUT);    //fila 2
digitalWrite(LED_2, LOW);
pinMode(LED_3, OUTPUT);    //fila 3
digitalWrite(LED_3, HIGH);

delay(30);
}
    if(antiRelej){
        //LED 1
pinMode(LED_1, OUTPUT);    //fila 1
digitalWrite(LED_1, LOW);
pinMode(LED_2, OUTPUT);    //fila 2
digitalWrite(LED_2, HIGH);
pinMode(LED_3, INPUT);     //fila 3

```

```
digitalWrite(LED_3, LOW);

    delay(30);
}
}

void stopBanda(){
    if(detener){
        pinMode(4, INPUT);
        pinMode(13, INPUT);
        pinMode(15, INPUT);
    }
}
```

---

## RESULTADOS Y CONCLUSIONES:

En esta práctica fue necesario emplear conocimientos básicos de programación, así como varias funciones aplicadas en practicas anteriores. Aunque fue un circuito un poco más elaborado, en cuento a funcionamiento, cumplió con el objetivo de ilustrar la aplicación de la técnica Charliplexing.

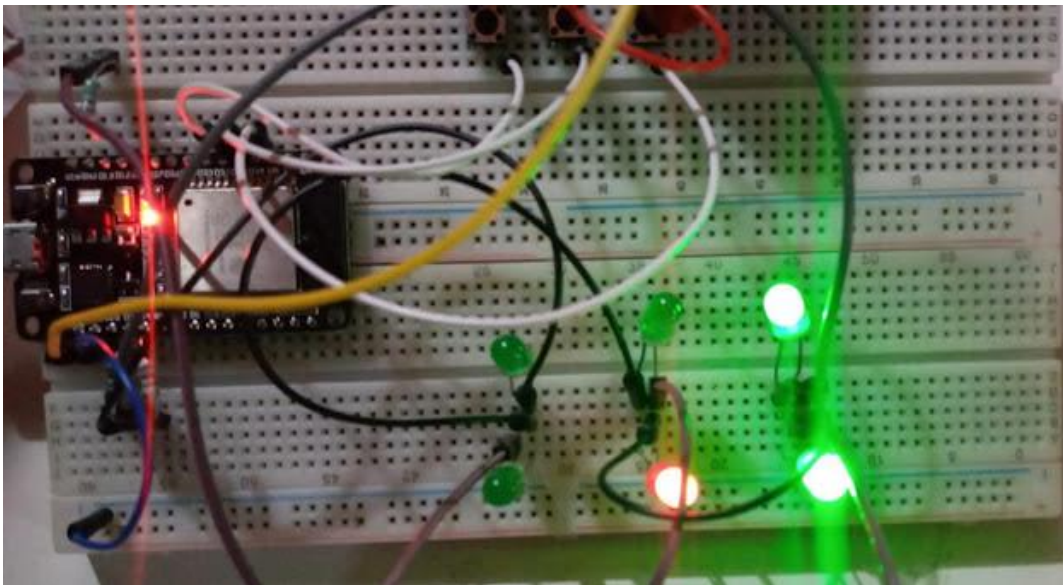


Ilustración 6. Resultado Práctica 5

---

## EVALUACIÓN:

- ¿Cuántas salidas se pueden controlar utilizando la técnica de charlieplexing con la tarjeta Arduino uno?

12 pines de E/S

- ¿En qué aplicaciones conviene utilizar esta técnica? Mencione ejemplos

Cuando no se cuenta con la cantidad suficiente de pines, por ejemplo, al conectar una pantalla LCD, LEDs o sensores.

- ¿Por qué es importante utilizar técnicas como la vista en esta práctica?

Para contar con diversas opciones o distintos métodos de solución para resolver el problema de no tener pines disponibles, y a su vez podemos ahorrar pines;

- ¿Cuáles son las desventajas de esta técnica?

Que el voltaje o energía se divide, por lo que en ocasiones podemos no obtener la cantidad de energía necesaria.

- ¿Describa en qué consisten las técnicas de tucoplexing, gugaplexing y chipioplexing?

Tucoplexing: Tenemos 4 LED de diseño Charlieplex estándar de 3 cables. En lugar de los 2 LED restantes del plex de 3 cables en la parte inferior, tenemos un par Charlieplex de dos botones más dos botones adicionales en un circuito RC.

Gugaplexing: Utiliza transistores como inversores de tres estados y nos permite controlar LED únicos adicionales en una matriz configurando dos pines hi/hi o lo/lo. Al igual que con Charlieplexing, todos los demás pines tendrán tres estados.

Chipioplexing: Es un método en el que se necesita agregar N transistores bipolares baratos. Este circuito usa tipos PNP, pero también puede usar dispositivos NPN.

---

## BIBLIOGRAFÍA:

- ✓ Hoja de Datos del Microcontrolador ATMEGA328P
- ✓ <https://www.arduino.cc/reference/en/>
- ✓ Arduino cookbook Michael Margolis Editorial O'Reilly