



Tecnológico Nacional de México  
Instituto Tecnológico de Tijuana  
Su dirección Académica  
Departamento de Sistemas y Computación  
Enero – junio 2017

Serie SC8A

Materia:

Sistemas Programables

Unidad 2

Título:

Práctica 4

Maestro:

Mitre Padilla Luis Alberto

Alumno:

Álvarez Corral Miguel Ángel 13211384

Espinoza Covarrubias Alejandro 13211465

Fecha:

Mayo 8 del 2017

## **Introducción**

En esta práctica se requiere con la ayuda de Arduino, un dip switch y una aplicación en C#, controlar el encendido y apagado de cuatro diodos leds, de manera que muestren los numeros del 0 al 15 en forma binaria, ya se colocándolos de manera manual con el dip switch o enviando el número desde la aplicación en C#. En Arduino UNO se deberá cargar el código desarrollado para que este cumpla el funcionamiento requerido.

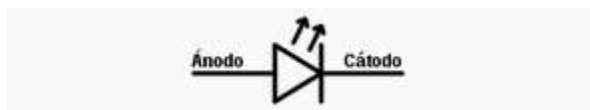
## Marco teórico

Esta práctica será realizada con los siguientes componentes:

- Protoboard
- Diodo LED
- Dip switch
- Resistencias 220 Ohms
- Arduino UNO

**Protoboard:** es una especie de tablero con orificios, en la cual se pueden insertar componentes electrónicos y cables para armar circuitos. Como su nombre lo indica, esta tableta sirve para experimentar con circuitos electrónicos, con lo que se asegura un buen funcionamiento del mismo.

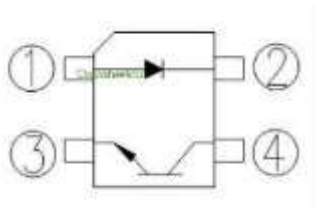
**Diodo LED:** El LED (Light-Emitting Diode: Diodo Emisor de Luz), es un dispositivo semiconductor que emite luz incoherente de espectro reducido cuando se polariza de forma directa la unión PN en la cual circula por él una corriente eléctrica. Su símbolo electrónico es:



**Dip switch:** Un DIP se trata de un conjunto de interruptores eléctricos que se presenta en un formato encapsulado (en lo que se denomina dual in-line-Package), la totalidad del paquete de interruptores se puede también referir como interruptor dip en singular.



**ITR 8102:** consiste en un diodo emisor infrarrojo y un Fototransistor NPN de silicio encerrado uno frente a otro convergiendo por medio de un eje óptico. Su representación en diagrama es:



**Resistencia:** se le denomina resistencia eléctrica a la oposición al flujo de electrones al moverse a través de un conductor. La unidad de resistencia en el Sistema Internacional es el

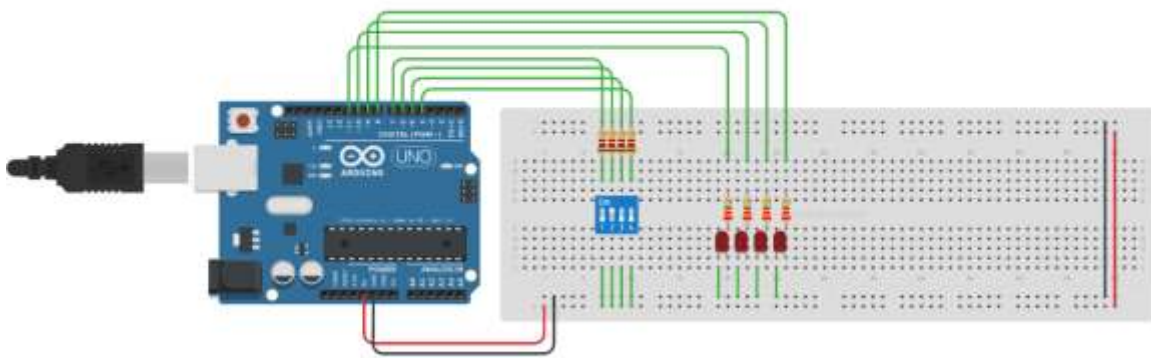
ohmio, que se representa con la letra griega omega ( $\Omega$ ), en honor al físico alemán Georg Simon Ohm, quien descubrió el principio que ahora lleva su nombre. Su símbolo eléctrico es:



**Arduino UNO:** El Arduino Uno es una placa electrónica basada en el ATmega328 (ficha técnica). Cuenta con 14 pines digitales de entrada / salida (de los cuales 6 se pueden utilizar como salidas PWM), 6 entradas analógicas, un 16 MHz resonador de cerámica, de una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio.



### Diagrama de Circuit.io



### Código de la práctica

- \* En los pines 4 - 7 se conecta el dip switch donde se tomaran las entradas.
- \* En los pines 8 - 11 se conectan los leds.

- \* El dip switch se conectada con un pin a la alimentación positiva y el otro
- \* al pin del arduino junto con una resistencia a tierra.
- \*/

```
byte state = 0;
```

```
void setup(){
```

```
    DDRD = B00001111; //Configuramos los pines 0-3 como salidas y los pines  
    4 - 7 como entradas.
```

```
    DDRB = B11111111; //Configuramos los pines 8-13 como salidas.
```

```
    Serial.begin(9600); //Iniciamos la comunicación serial por los pines 0 y 1.
```

```
}
```

```
void loop(){
```

```
    if(state != (PIND & B11110000)) //Revisamos si la lectura del dip switch es  
    diferente de la última vez
```

```
{
```

```
    state = PIND & B11110000; //Guardamos el estado de la lectura actual
```

```
    PORTB = state >> 4; //Imprimimos el resultado en el puerto B (pines 8 - 11)
```

```
    Serial.println(PORTB);
```

```
}
```

```
    delay(100);
```

```
    if(Serial.available()) //Si tenemos una lectura en el puerto serial.
```

```
{
```

```
    char i = Serial.read(); //Tomamos el carácter enviado por el usuario
```

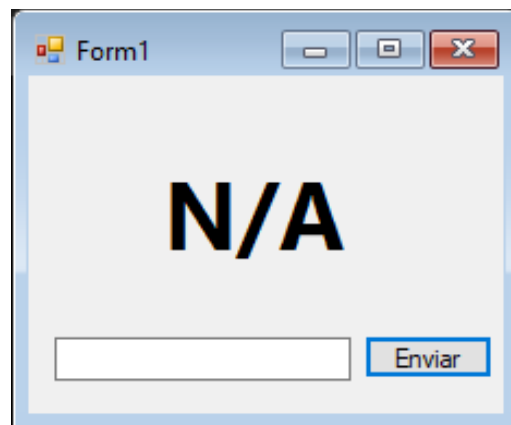
```
    byte n = (i-48); //Asumimos que es un número y restamos 48 para convertir  
    de ascii a numerico
```

```

if(Serial.available())
{
    i = Serial.read(); //Si tenemos otro número (en caso de números de dos
    dígitos)
    n = n * 10 + (i-48); //Agregamos el último dígito
}
PORTB = n; //Imprimimos el valor del número en los LEDs.
}
}

```

### Interfaz de la aplicación en C#



### Código de la aplicación en C#

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;

```

```
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO.Ports;

namespace Practica_4_Unidad_2
{
    public partial class Form1 : Form
    {
        SerialPort puerto;

        public Form1()
        {
            InitializeComponent();
            puerto = new SerialPort();
            puerto.BaudRate = 9600;
            puerto.PortName = "COM3";
            puerto.Close();
            puerto.Open();
            timer1.Enabled = true;
        }

        private void btn_Send_Click(object sender, EventArgs e)
        {
            try
```

```

{
    int i = int.Parse(txt_Input.Text);
    if (i < 0 || i > 15)
    {
        MessageBox.Show("Ingrese numeros dentro del rango 0 - 15.");
        return;
    }
    puerto.Write(txt_Input.Text);
    lbl_Output.Text = txt_Input.Text;
}
catch (System.Exception)
{
    MessageBox.Show("Entrada invalida, intente nuevamente con
números enteros en el rango 0 - 15");
}
}

```

```

private void timer1_Tick(object sender, EventArgs e)
{
    if (puerto.BytesToRead > 0)
        lbl_Output.Text = "";

    while (puerto.BytesToRead > 0)
    {
        lbl_Output.Text += puerto.ReadLine();
    }
}

```

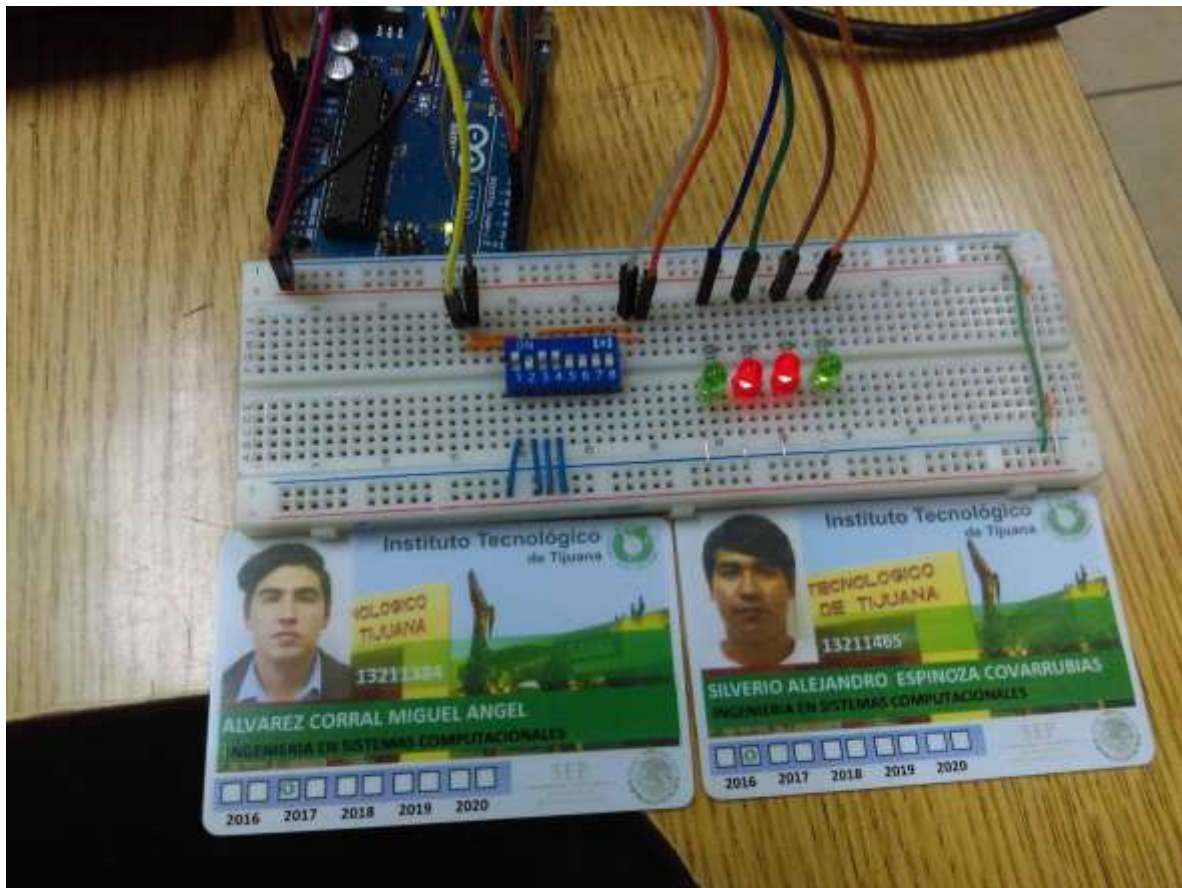


}

}

}

## Diagrama real



## **Conclusión**

Utilizando ArduinoUNO, un protoboard y una aplicación en C# se pudo lograr que la práctica funcionara de la manera requerida. Se aprendió a realizar una comunicación entre un Arduino y una aplicación en C#, al igual que esto no solo se puede hacer con un solo lenguaje de programación ya que se pueden implementar otros lenguajes como Python. Es importante conocer los componentes que tienen polaridad y los que no, ya que esto es un factor importante al momento de conectar los componentes. También se desarrolló un código el cual sería necesario para poder realizar el funcionamiento adecuado de los componentes antes mencionados. Es una forma útil de utilizar tanto Arduino como C#, ya que no solo se pueden implementar aplicaciones como las de esta práctica sino que también muchas otras.