Parcial 1

Informática II

Alejandra Calle Vasquez
Jesús David Mercado Machado
Juan Sebastian Garavito Gallo

Despartamento de Ingeniería Electrónica y
Telecomunicaciones
Universidad de Antioquia
Medellín
Abril de 2021

Índice

1. Analisis del problema

Organización de los leds: Para la optimización de la conexión y organización de los leds usamos 8 integrados 74HC595 los cuales nos permiten formar un sistema de ubicación de cada bombillo, a cada uno de estos integrados le asignamos 8 leds; que sería lo máximo que podríamos controlar por cada uno de ellos, de este modo formariamos la matriz 8x8 que se es requerida para realizar el ejercicio. Esta manera de conexión de los leds nos pareció optima por que solo conlleva al uso de tres pines.

2. Circuito inicial en Tinkercad

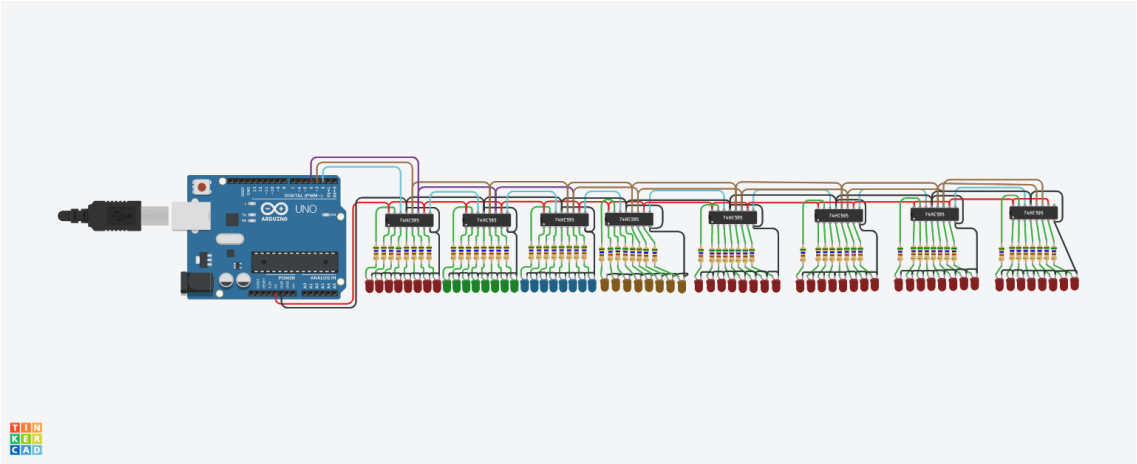

Figura 1: Primeras conexiones

2.1. Patron determinado

Organización matricial de los leds (Para verificar funcionamiento): En este caso organizamos nuestros leds de modo que se pueda ver el tablero en forma de matriz y usamos unos patrones establecidos para corroborar su funcionamiento.

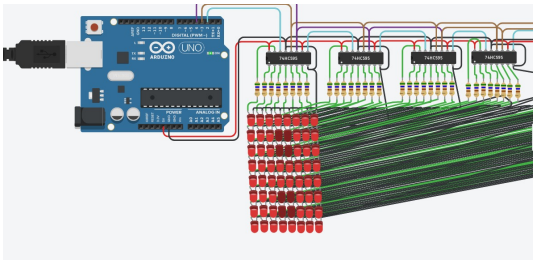

Figura 2: Simulación del tablero, signo C

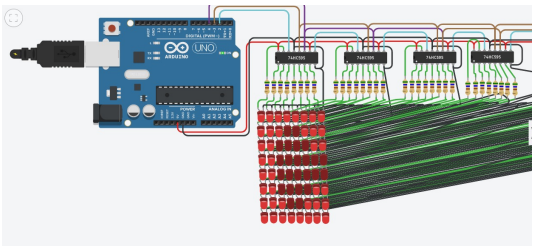

Figura 3: Simulación del tablero, signo B

2.2. Interpretación del patron predeterminado

Para representar estos codigos predeterminados usaremos entradas de 1 y 0 para representar los patrones con las siguientes clasificaciones **1=Endendido 0=Apagado**

[Tablero signo C]

```
1 1 1 1 1 1 1 1
1 1 1 0 0 1 1 1
1 1 0 0 0 0 1 1
1 1 0 0 0 0 0 0
1 1 0 0 0 0 0 0
1 1 0 0 0 0 1 1
1 1 1 0 0 1 1 1
1 1 1 1 1 1 1 1
```

[Tablero signo B]

```
1 1 1 1 1 1 1 1
1 1 1 0 0 1 1 1
1 1 1 0 0 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 0 0 1 1 1
1 1 0 0 0 0 1 1
1 1 1 1 1 1 0 1
```

3. Algoritmo implementado

Main principal
3

```
int incomingByte = 0; // for incoming serial data
int aux=0;
int cont=0;
void setup() {
  Serial.begin(9600); // opens serial port, sets data rate to 9600 bps
}

void loop() {
  // send data only when you receive data:
  if (Serial.available() > 0) {
    // read the incoming byte:
    if(cont<9)
    {
      aux=Serial.read();
      incomingByte+=(aux-48)*pow(2,cont-1);
    }

    else
    {
        Serial.print("I received: ");
        Serial.println(incomingByte);
        cont=0;
    }
    cont++;
  }
}
```

3.1. implementación

Lo que tratamos de lograr con el código anteriormente presentado es pedir fila por fila al usuario lo que quiere representar en el tablero de leds, de esta manera estaremos recibiendo los datos en asciipara luego pasarlos a binario y así llegar a los tableros con las entradas en unos y ceros (como se ve representado en el punto 2.2), pero al momento de hacer la conversión nos quedaría un solo valor, esto significa que si el usuario ingresa una fila entera de "1"solo tendríamos que almacenar el valor 255; que es el equivalente, para luego imprimirlo.

3.2. Código que llevamos elaborado hasta el momento

date: 21/04/2021

```
char digito='0';
int cont=7,columna=0,*matriz;
int pinData  = 2;
int pinLatch = 3;
int pinClock = 4;
void setup() {
  Serial.begin(9600);
  pinMode(pinData, OUTPUT);
  pinMode(pinLatch, OUTPUT);
  pinMode(pinClock, OUTPUT);
}
int* Lectura() {
   // send data only when you receive data:
        int incomingByte[8];
        for(int i=0;i<8;i++)
        {
        incomingByte[i]=0;
        }

    while(Serial.available() > 0)
    {
      digito=Serial.read();
      Serial.println(digito);
      Serial.println(columna);
      if(digito=='1')
          {
            incomingByte[columna]+=(pow(2,cont)+0.5);
                cont--;
          }
      else if(digito=='0')
      {
        cont--;
      }
      if(cont==-1){
        Serial.println(incomingByte[columna]);
        Serial.print("Ahora ingrese la siguiente fila de matrix:");
        Serial.println(columna+1);
        cont=7;
        columna++;
      }
      if(columna==7)
      {
```

```
        return incomingByte;
      }
    }
}
void ledWrite(int matriz[8]){
   for(int i=0;i<8;i++)
   {
        shiftOut(pinData, pinClock, LSBFIRST, *(matriz+i));
   }
                digitalWrite(pinLatch, HIGH);
                digitalWrite(pinLatch, LOW);
}

void loop()
{
  matriz=Lectura();
  ledWrite(matriz);

}
```

Hasta este punto del código tenemos completa la recepción de datos.

4. Problemas de desarrollo

```
char digito='0';
int incomingByte=0;
void setup() {
  Serial.begin(9600); // opens serial port, sets data rate to 9600 bps
}

void loop() {
   // send data only when you receive data:
    while(Serial.available() > 0)
    {
        for(int cont=0;cont<8;cont++)
        {
          digito=Serial.read();
          if(digito=='1')
          {
            //Serial.print("contador:");
            //Serial.println(cont);
            incomingByte+=pow(2,cont)+0.5;
            //Serial.print("potencia:");
            //Serial.println(pow(2,cont)+0.5);
            Serial.print("Sumatoria:");
            Serial.println(incomingByte);
          }
        }
        }
}
```

1er error: En las lineas 151 y 154 de esta parte del código tenemos dos comentarios que hacen que el código se dañe, lo que no nos parece lógico ya que los comentarios no deberían afectar en nada.

2do error: El 0.5 que pusimos en en el ¡pow¿es por que ¡pow¿retorna un flotante y que si lo conviertes al entero no dal el resultado en tipo entero si no un numero menor.