Parcial I 2021-1 Informática 2

Informe de Desarrollo de Proyecto

Juan Diego Cabrera Moncada
Julian David Quintero Marin
Julian Montenegro Pinzón

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

Índice

1. Introducción y desarrollo del proyecto

Este es el documento correspondiente al grupo de 3 personas conformado por los estudiantes del curso de Informática 2: Juan Diego Cabrera Moncada, Julian David Quintero Marin y Julian Montenegro Pinzón. El proyecto a desarrollar busca la visualización de patrones en una matriz de LEDs de manera efectiva, donde dichos patrones y el intervalo de tiempo entre cada patrón a mostrar son dados por el usuario.

2. Análisis del problema y consideraciones para la alternativa de solución propuesta

El desarrollo de la solución al problema parte de la premisa de la posibilidad de controlar el encendido y apagado de LEDs de la matriz de manera organizada con base en la posición en la que cada LED se encuentre. De este modo, retomando como base el ejemplo dado por el profesor de Informática 2 Jonathan Ferney en el cual se hace uso de un circuito integrado 74HC595, se puede evidenciar que, si conectaramos la matriz de LEDs de forma que cada columna de LEDs se encuentre en serie usando el mismo código de ejemplo, se evidencia que se encenderían todos los LEDs de cada columna. Por lo cual se considera que la solución se debe fundamentar en el uso de las filas y columnas de la matriz como coordenadas de referencia para llevar a cabo la localización de cada LED de manera efectiva y organizada. Una consideración para el desarrollo de la solución es el uso de un componente capaz de inhibir la falta de voltaje y conservar la idea de plantear un sistema de coordenadas con filas y columnas. Con esto en mente, se busca la integración de transistores NPN y PNP como componentes de control de la electricidad que recorre el circuito, específicamente para el control de cada LED una vez localizados en su respectiva coordenada(Fila como X, Columna como Y). El día martes se opta por una alternativa de solución que deseche el uso de transistores y se crea un circuito cuya organización debe tenerse en cuenta para la creación del código correspondiente y así enlazar el circuito con el código como la solución propuesta del problema. De este modo, la solución propuesta radica en el uso del pin SER de un integrado que puede ser llamado el Integrado Master, pues las salidas de éste se encuentran conectadas a los pins SER de otros integrados, que llamaremos los Integrados Branch. Se busca que se ingrese electricidad por el pin SER del Integrado Master y, a través del correcto manejo de los pines SRCLK y RCLK de este integrado, la electricidad sea enviada por las salidas correctas de acuerdo con el integrado que se requiera usar para poder asignarle un estado a un LED en específico. De este modo, cada integrado está conectado a una columna de la matriz de LEDs, por lo cual se debe tomar estos integrados como la referencia para las columnas de la matriz mientras que las salidas correspondientes a cada integrado son establecidas como las coordenadas de la fila en la que se encuentra cada LED.

3. Esquema de descripción de las tareas a definir en el desarrollo del algoritmo

Las tareas a definir en el desarrollo del algoritmo corresponden a las siguientes:

4. Algoritmo implementado

5. Problemas de desarrollo presentados

Al probar el circuito descrito en la primera sección de análisis del problema, en el cual cada columna está conectada en serie, se evidencia, por medio de la medición por multímetro, que el voltaje que se transmite por cada LED (1.6 voltios aproximadamente) es menor al que registraba con el ejemplo dado por el profesor (4.8 voltios aproximadamente). El transistor NPN que aparece en la imagen no tiene ninguna relevancia en el circuito. Posterior a ello, se procede

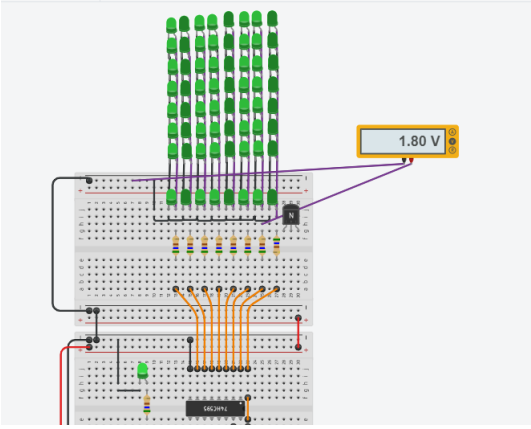

Figura 1: Prueba con columnas LED en serie basada en el circuito de ejemplo

a realizar una serie de conexiones introduciendo nuevos componentes con base en el análisis del problema inicialmente descrito (2). Para nuestro circuito base inicialmente usamos un Arduino uno, dos placas de prueba (protoboard´s), transistores NPN, PNP, resistencias, diodos emisores de luz y dos circuitos integrados SN74HC595 que serán los protagonistas en el proyecto pues serán los encargados de entregar el proceso de control coordinado con nuestro código, usaremos 5 pines del Arduino, cuatro para los puertos SRCLK y RCLK de cada uno de los integrados y uno para los dos puertos SER de los integrados, desde este pin entregaremos la información que será transmitida a la matriz de leds.

Se finaliza el montaje del circuito base, depues de esto se inicia el proceso de analisis de control para llegar al fin de proyectar el patron deseado a nuestra

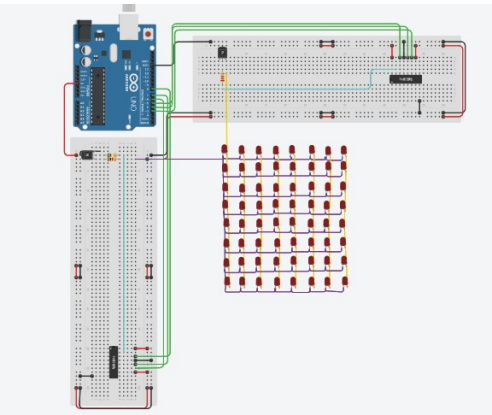

Figura 2: Circuito base de conexiones Versión 1

matriz de leds 8X8. Se replantea el montaje del circuito base, en el cual se elimina

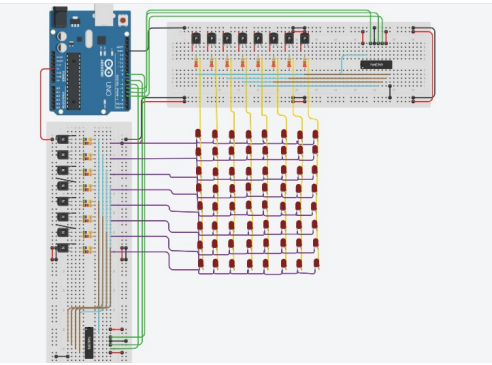

Figura 3: Circuito base de conexiones Versión 2

el uso de transistores y en su lugar, se utiliza como estructura de control una cantidad de 9 integrados 74HC595, uno actúa como el Master cuyas salidas se conectan a los integrados Branch (Rama), dicho nombre puesto arbitrariamente para mayor claridad, y con cada integrado Branch se controla el estado de una columna de la matriz de 8x8 de LEDs. Cabe destacar que se conectan resistencias de 560 ohmios con el objetivo de restringir el flujo de electricidad a una magnitud con la cual se evite el daño o deterioro de LEDs durante el proceso de ejecución de la simulación y el código. Se usan 5 puertos digitales donde el puerto digital 2 está conectado al puerto serial SER del Integrado Master 74HC595, el 4 al puerto RCLK del mismo integrado, el 5 al puerto RCLK, y el 11 y 12 a los puertos RCLK Y SRCLK respectivamente de los integrados Branch. Posteriormente, se hizo progreso con respecto al montaje de este circuito base modificado hasta el punto en el que dicho circuito sea

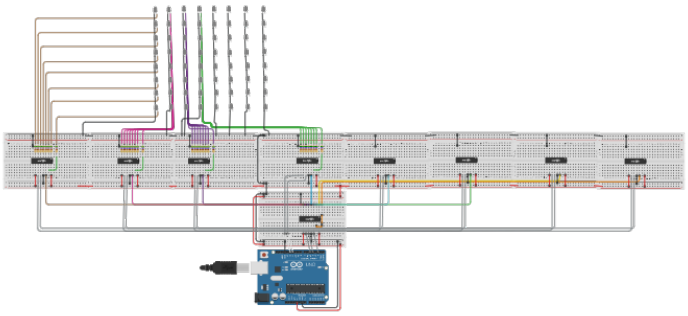

Figura 4: Modificación 1 del Circuito Base

funcional. Por lo cual, se procedió a desarrollar pruebas con código una vez el circuito fue finalizado en su totalidad. En cuanto al código correspondiente a

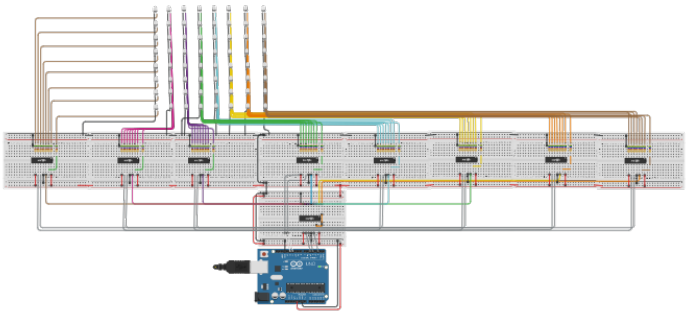

Figura 5: Modificación 2 del Circuito Base

las pruebas, en las cuales satisfactoriamente se encendieron todos los LEDs de la matriz fue la siguiente:

```
// Programa desarrollado, compilado y ejecutado en https://https://www.tinkercad
const int SER = 2; //Puerto digital 2 Puerto serial
const int RCLK = 4; //Puerto digital 4 Puerto registro de salida
const int SRCLK = 5; //Puerto digital 5 Puerto registro de desplazamiento
```

```
const int ARCLK = 11; //Puerto digital 11 Puerto de registro de salida controlad
const int ASRCLK = 12; //Puerto digital 12 Puerto de registro de desplazamiento

void setup()
{
  // PASO 1 Configuracion de puertos digitales como salida
  pinMode( SER , OUTPUT);
  pinMode( RCLK, OUTPUT);
  pinMode( SRCLK, OUTPUT);
  pinMode(ARCLK,OUTPUT);
  pinMode(ASRCLK,OUTPUT);

  //Paso 2 Inicializaci n de puertos digitales
  digitalWrite(SER,LOW);
  digitalWrite(RCLK,LOW);
  digitalWrite(SRCLK,LOW);
  digitalWrite(ASRCLK,LOW);
  digitalWrite(ARCLK,LOW);

  /*for(unsigned short i=0; i<=7; i++){
        digitalWrite(SER,1);
        digitalWrite(SRCLK,0);
        digitalWrite(SRCLK,1);
        digitalWrite(SRCLK,0);
  }
  digitalWrite(RCLK,0);
  digitalWrite(RCLK,1);
  digitalWrite(RCLK,0);
  */

  //digitalWrite(12,0);
  //digitalWrite(12,1);
  //digitalWrite(12,0);

  //digitalWrite(11,0);
  //digitalWrite(11,1);
  //digitalWrite(11,0);
  delay(1);
  for(unsigned short i=1; i<=8; i++){
        digitalWrite(SER,0);
        digitalWrite(SRCLK,0);
        digitalWrite(SRCLK,1);
        digitalWrite(SRCLK,0);
  }
  digitalWrite(RCLK,0);
  digitalWrite(RCLK,1);
```

```
  digitalWrite(RCLK,0);

  digitalWrite(12,0);
  digitalWrite(12,1);
  digitalWrite(12,0);

  digitalWrite(11,0);
  digitalWrite(11,1);
  digitalWrite(11,0);
  delay(1);
}

void loop()
{
  for(unsigned short i=0; i<=7; i++){
        digitalWrite(SER,1);
        digitalWrite(SRCLK,0);
        digitalWrite(SRCLK,1);
        digitalWrite(SRCLK,0);
  }
  digitalWrite(RCLK,0);
  digitalWrite(RCLK,1);
  digitalWrite(RCLK,0);

  digitalWrite(ASRCLK, 0);
  digitalWrite(ASRCLK, 1);
  digitalWrite(ASRCLK, 0);

  digitalWrite(ARCLK, 0);
  digitalWrite(ARCLK, 1);
  digitalWrite(ARCLK, 0);
  //digitalWrite(12,0);
  //digitalWrite(12,1);
  //digitalWrite(12,0);

  //digitalWrite(11,0);
  //digitalWrite(11,1);
  //digitalWrite(11,0);
  delay(1);
}
```

Posterior a ello, se define una serie de funciones, entre las cuales se encuentra la función verificación, con la cual se inician a realizar pruebas con el objetivo de entender y comprobar las ideas que se tienen sobre el funcionamiento del circuito dependiendo de los datos que se ingresen.

```
const int SER = 3; //Puerto digital 3 Puerto serial
```

```
const int SRCLK = 4; //Puerto digital 4 Puerto registro de desplazamiento
const int RCLK = 5; //Puerto digital 5 Puerto registro de salida
const int ASRCLK = 6; //Puerto digital 6 Puerto de registro de desplazamiento co
const int ARCLK = 7; //Puerto digital 7 Puerto de registro de salida controlador
char *ptr_user=NULL;
char user[8];
int i=0;

void subida(int);
void salida_datos(short, short, int, short, bool);
void verificacion();

void setup()
{
  Serial.begin(9600);
  for(unsigned short ini=3; ini<=7; ini++){
    pinMode(ini,OUTPUT);
    digitalWrite(ini,LOW);
  }
  Serial.write("Bienvenido/a. Ingrese 1 para verificar LEDs\n");
  Serial.write("Ingrese 2 para mostrar un unico patron de LEDs\n");
  Serial.write("Ingrese 3 para mostrar una serie de patrones de LEDs\n");
  verificacion();
}

void loop()
{
}
void salida_datos(short dato, int clock, short itera_limit, bool imprimir){
  for(unsigned short i=0;i<=itera_limit; i++){
        if(dato!=2) digitalWrite(SER,dato);
    subida(clock);
    if(imprimir==1) subida(clock+1);
  }
}
void subida(int clock){
  digitalWrite(clock,0);
  digitalWrite(clock,1);
  digitalWrite(clock,0);
}
void verificacion(){
  salida_datos(1,SRCLK,3,1);
  salida_datos(2,ASRCLK,7,1);
  salida_datos(1,SRCLK,3,1);
  salida_datos(1,ASRCLK,3,1);
}
```

6. Evolución del algoritmo y consideraciones a tener en cuenta en la implementación

En un inicio se plantea que el algoritmo requiere considerar el uso de componentes de modo que se establezca un sistema de coordenadas con filas y columnas manejable por medio del uso eficiente del circuito integrado (CI) 74HC595. Para ello, se establece como consideración para la implementación de este sistema el uso de 2 circuitos integrados 74HC595 donde uno de ellos se usa para controlar las filas de la matriz de LEDs y el otro para las columnas, así como el uso de 5 puertos digitales del Arduino: 1 de ellos se encuentra conectado al pin SER tanto del circuito integrado de las filas (CIF) como el de las columnas (CIC); y los 4 restantes para controlar por aparte los pines asignados a los relojes de registro de desplazamiento y a los relojes de registro de salida de cada CI, es decir, 1 puerto digital se asigna a cada pin. El día martes se replantea esta consideración con base en el replantamiento del circuito hecho, teniendo como consideración para la implementación del algoritmo que el circuito usa 9 integrados de la forma explicada en la sección de análisis del problema (2), se debe tener en cuenta que el algoritmo debe centrarse en el control del puerto digital conectado al pin SER del Integrado Master y el uso en el momento adecuado del código de los pines SRCLK y RCLK, principalmente con mayor cuidado en los Integrados Branch.

Referencias