

# Parcial N° 1

**Ivonne Rosero**

1007687589

**Rigoberto Berrio**

1040327583

Departamento de Ingeniería Electrónica y

Telecomunicaciones

Universidad de Antioquia

Medellín

Abril de 2021

## Índice

1. Análisis del problema y consideraciones para la alternativa de solución propuesta.	2
2. Esquema donde describa las tareas que se definieron en el desarrollo del algoritmo.	2
3. Algoritmo implementado.	3
4. Problemas de desarrollo.	3
5. Evolución del algoritmo y consideraciones a tener en cuenta en la implementación.	5

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

Gracias a los apuntes y grabaciones de clase se están haciendo pruebas para afianzar el conocimiento para poder asimilarlo al nivel del examen.

Decidimos utilizar 8 integrados para que cada uno controlara 8 leds y así completar los 64 leds requeridos.

En la Figura (1) se muestra el circuito a utilizar en el examen.

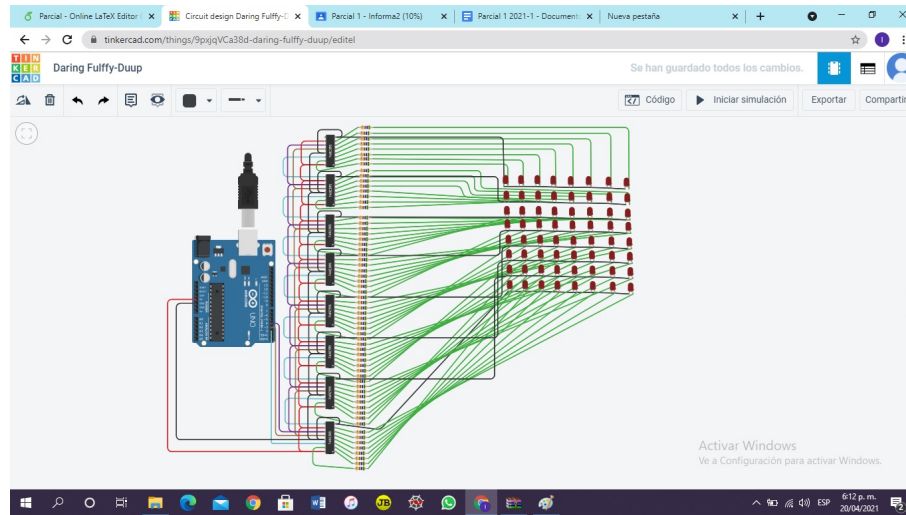


Figura 1: Circuito final

## 2. Esquema donde describa las tareas que se definieron en el desarrollo del algoritmo.

1. Buscando reducir al máximo el uso de pines digitales se buscó y a la final lograrlo, puentear 8 integrados 74HC595 en serie; al hacer esto la información dada al primer integrado será la misma que recibirán los demás, esto también nos lleva a conectar los LED'S en grupos de ocho para que cada integrado controle un grupo. Las resistencias tienen la labor de controlar el voltaje ya que los integrados y los leds solo soportan 5V.

2. Se usa el comando `shiftOut()` la cual se encarga de desplazar hacia la salida un bit cada vez, comienza a partir del bit mas significativo o menos significativo en este caso se inicio desde el bit menos significativo usando el comando `LSBFIRST`. Se programaron los 3 pines que van conectados del arduino al integrado en `OUTPUT` (salida). Se obtiene que los leds formen figura.

### 3. Algoritmo implementado.

Algoritmo base.

```
Proyecto
//Variables para definir los pines del arduino al integrado
int pindato=23;
int pindig=3;
int pinrelaj=4;
#define T 2000 //Tiempo

void ledWrite(int Alad, int Blad, int Clad, int Dlad, int Elad, int Flad, int Glad, int Hlad) { //Funcion con 8 variables para los 8 integrados del circuito
  shiftOut(pindato, pinrelaj, LSBFIRST, Alad); //Cada variable se debe programar con los pines anteriores
  shiftOut(pindato, pinrelaj, LSBFIRST, Blad);
  shiftOut(pindato, pinrelaj, LSBFIRST, Clad);
  shiftOut(pindato, pinrelaj, LSBFIRST, Dlad);
  shiftOut(pindato, pinrelaj, LSBFIRST, Elad);
  shiftOut(pindato, pinrelaj, LSBFIRST, Flad);
  shiftOut(pindato, pinrelaj, LSBFIRST, Glad);
  shiftOut(pindato, pinrelaj, LSBFIRST, Hlad);
  digitalWrite(pindig, HIGH);
  digitalWrite(pindig, LOW);
}

//Los tres pines se deben establecer en salida
void setup() {
  pinMode(pindato, OUTPUT);
  pinMode(pindig, OUTPUT);
  pinMode(pinrelaj, OUTPUT);
}

void loop() {
  ledWrite(255,255,254,254,7,199,255,126); delay(T); //línea de código para proyectar el 5
}
```

Figura 2: Algoritmo Implementado

### 4. Problemas de desarrollo.

Se usa el comando shiftOut() la cual se encarga de desplazar hacia la salida un bit cada vez, comienza a partir del bit mas significativo o menos significativo en este caso se inicio desde el bit menos significativo usando el comando LSBFIRST.

Se programaron los 3 pines que van conectados del arduino al integrado en OUTPUT (salida).

Se obtiene que los leds formen la figura.



Figura 3: Figura final

Siempre que el usuario ingrese alguna opcion siempre aparecera de la primero la figura.

Se le muestra al usuario por medio de la funcion "verificacion" que todos los leds funcionan.

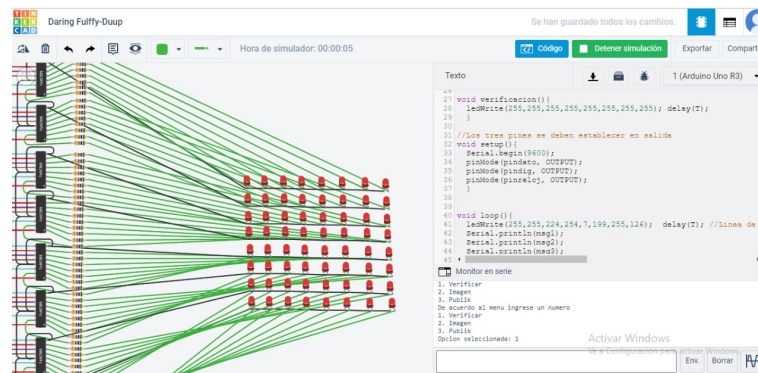


Figura 4: Verificacion

Creacion de la funcion imagen”, en este caso se ingreso que una fila alumbrara y la siguiente no.

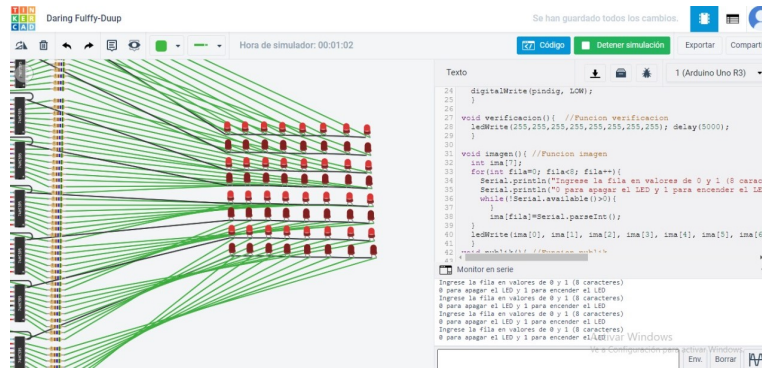


Figura 5: La imagen que ingreso el usuario.

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

Se crea un menú de manera que el usuario se le facilite la opción que quiera elegir.

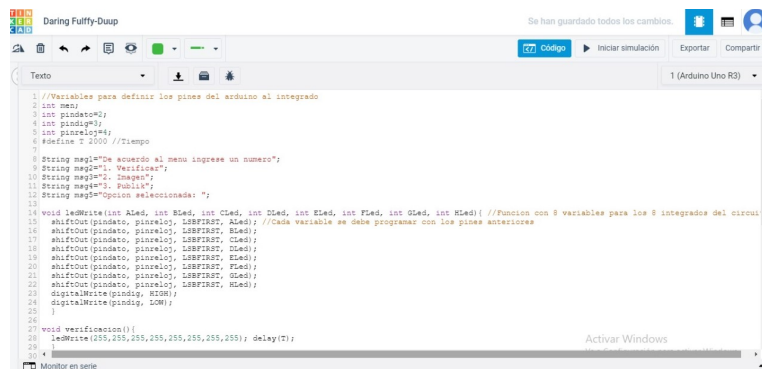


Figura 6: Avance del código

En la primera opcion esta la verificación donde el usuario observará que todos los leds funcionan.

```

27 void verificacion(){
28   digitalWrite(255,255,255,255,255,255,255,255); delay(10);
29 }
30
31 //Los tres pines se deben establecer en salida
32 void setup(){
33   Serial.begin(9600);
34   pinMode(pindato, OUTPUT);
35   pinMode(pindig, OUTPUT);
36   pinMode(pinele0, OUTPUT);
37 }
38
39 void loop(){
40   digitalWrite(255,255,255,255,255,255,255,255); delay(10); //Linea de codigo para proyectar el 5
41   Serial.println(msg1);
42   Serial.println(msg2);
43   Serial.println(msg3);
44   Serial.println(msg4);
45   Serial.println(msg5);
46   while(Serial.available()==0){
47     }
48     msg=Serial.parseInt();
49     Serial.println(msg5);
50     Serial.println(msg6);
51     if(msg=="1"){
52       verificacion();
53     }
54   }
55 }

```

Figura 7: Avance del codigo

Se crea la funcion "imagen" la cual da la orden a cada led de encender o apagar segun el usuario lo desee ingresando "1" si quiere encenderlo o "0" si desea apagarlo, formando asi la figura que desea.

```

24 digitalWrite(pindig, LOW);
25 }
26
27 void verificacion(){ //Funcion verificacion
28   digitalWrite(255,255,255,255,255,255,255,255); delay(5000);
29 }
30
31 void imagen(){ //Funcion imagen
32   int ima[7];
33   for(int fila=0; fila<8; fila++){
34     Serial.println("Ingrese la fila en valores de 0 y 1 (8 caracteres)");
35     Serial.println("C para apagar el LED y 1 para encender el LED");
36     while(Serial.available()>0){
37       }
38       ima[fila]=Serial.parseInt();
39     }
40     digitalWrite(ima[0], ima[1], ima[2], ima[3], ima[4], ima[5], ima[6], ima[7]); delay(5000);
41   }
42 }
43 void publik(){ //Funcion publik
44   }
45
46 //Los tres pines se deben establecer en salida
47 void setup(){
48   Serial.begin(9600); //Llamar el puerto serial
49   pinMode(pindato, OUTPUT);
50   pinMode(pindig, OUTPUT);
51   pinMode(pinele0, OUTPUT);
52 }
53 }

```

Figura 8: Avance del codigo