

PARCIAL 1

INFORME

Diego Fernando Urbano Palma
Michael Stiven Zapata Giraldo
Brayan Steven Avila Marin

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

Índice

1. Análisis del problema	2
2. problemas en el desarrollo	3
3. Esquema donde describa las tareas que usted definió en el desarrollo del algoritmo.	4
4. Algoritmo implementado.	5
5. Evolución del algoritmo y consideraciones a tener en cuenta en la implementación.	7

1. Analisis del problema

Para enfrentar el problema planteado en cual debemos crear una matriz de leds en los cuales debemos encenderlos y apagarlos para representar patrones ingresados por el usuario proponemos el uso de 8 circuitos integrados 74HC595 los cuales conectaremos haciendo un puente entre los pines del mismo tipo y así llevarlos al los pines digitales del arduino.

Tenemos un circuito montado en la plataforma Tinkercard, donde estamos implementado por medio de un Arduino Uno R3, una protoboard, ocho circuitos integrados, 64 cables cafés, y demás cables para realizar la comunicación de datos entre el arduino, la placa de pruebas, los chips y los leds. Mediante los pines del Arduino, SER=5, reloj-des=3, reloj-salida=4 y mediante la variable opciones=-1, este último es una variable para guardar el número que pide el Serial.available() y guardarlo con Serial.parseInt(), para llevar a cabo una presentación de lo que el proyecto puede mostrarle al usuario. Nuestro proyecto, se compone de las funciones void: setup, loop, inicialización, apagar, patrón, ingreso, grupo8, orden, guardador, verificación, imagen, publik.

La función loop() desprende la información del menú por consola con tres opciones; la primera corresponde presionando la tecla 1 para invocar la función verificación() y si al presionar la tecla uno sale el mensaje “todos los leds función correctamente”, por favor presione cualquier tecla alfanumérica para volver al menú principal”, por lo tanto se puede proseguir con el funcionamiento del proyecto, este menú por consola también ofrece la opción 2, la cual sirve para llamar la función imagen() y la tercera opción con la tecla 3 ofrecerá al ejecutar la función publik(). En caso contrario, cuando no se presionó la tecla indicada en el menú propuesto de la consola se imprimirá también por consola “el dato ingresado no se contempla en el menú” lo quiere decir que se ingresó un valor diferente a 1,2 o 3, esta captura de datos se hace por medio del dato creado opciones=Serial.parseInt();.

Inmediatamente después de la función loop() viene la función inicializacion() la cual al ejecutarse inicializara los “relojes” reloj-des y reloj-salida, después de esta función esta la función apagar() que se encargara de apagar los leds del led 1 al led 64 mediante un ciclo for y con la ayuda de la instrucción digitalWrite(SER,0), para escribir el valor lógico digital en pin de salida de nuestro Arduino Uno R3, dentro de este ciclo también se llamara la mencionada función inicialización().

La función patron() tiene como parámetros int guardado[] e int num-arre. Esta función ofrece al usuario la posibilidad de que escoja el número de leds que se van a prender por fila, y también ingresar el número que representan los leds prendidos. Esta función guardara en la variable num el número en las posiciones en que se va encender el led, siendo las posiciones desde 1 hasta ocho. Por ejemplo, si se ingresa 1468, estará pidiendo encender los leds 1 4 6 8. La función ingreso() mediante su ejecución guardará en una arreglo los números dígitos excluyendo el cero y el nueve. Esto para para mostrar los leds encendidos en esas posiciones excluyendo los mencionados, es decir, entre uno y ocho. La siguiente función a tener en cuenta es grupo8() se encarga de recorrer las filas

de la matriz de leds y pide una cadena de enteros, dentro de esta función que llama la función `inicializacion()` para encender o apagar los leds según el carácter requerido por el usuario. La función `orden()` permite guardar en el arreglo `fin[]` los “1” las posiciones que ingreso el usuario por la consola. Mientras que la función `guardador()` va a guardar los arreglos de arreglos, que posteriormente guardara los valores de encendido y apagado de los led. La función `verificación()` prende todos los leds esto es solamente para verificar que cada uno de los leds enciendan o funcionen correctamente. La función `imagen()` juega un papel muy importante en la ejecución del circuito y del código, ya que ésta permite que el usuario interactúe con la consola; en una primera opción podrá el usuario ingresar los números de las posiciones en que quiera que se encienda los leds, de segunda opción podrá ingresar las letras que formaran el encendido de los leds, mostrando detalladamente los números que debe presionar para que se encienda los leds y forme una letra según el número que le haya sugerido el programa por la consola. La función `publik()` permitirá al usuario ingresar el número de patrones que el usuario desea mostrar, también permitirá al usuario ingresar los milisegundos entre cada visualización de patrones, el usuario también deberá ingresar por consola el tiempo de visualización entre cada patrón. Dentro de esta función hay dos for anidados, el primero consta de tres for el cual el primera recorrerá el número de patrones y los guardara; el segundo for recorrerá las filas de la matriz para guardar las secuencias, mientras que el tercer for es para recorrer los leds que no se van a encender. El ultimo for anidado de esta función contiene cuatro for que ayudaran a llamar la función `grupo8()` cuyo parámetro es el arreglo 2x2 secuencia; este último for anido tiene un delay cuyo argumento es la variable tiempo, que el usuario habrá ingresado por consola en esta misma función.

2. problemas en el desarrollo

- Lograr la conexión de los 8 circuitos integrados 74HC595 al arduino para controlar los leds.
- Leds y circuitos quemados, debido al mal cálculo de la capacidad de ohmios de las resistencias.
- Conectar la matriz al circuito integrado 74HC595 de tal forma que podamos controlar el encendido de todos los leds.
- Entender cómo se manipulan los leds por medio del circuito integrado 74HC595 y el registro de desplazamiento.
- Buscar las funciones más óptimas en c++ para hacer amena la escritura y estructura del código.
- Abuso de la memoria disponible por la plataforma de Tindercard.

- Falta de conocimiento de las funciones básicas de Arduino como la captura de datos y manipulación de punteros, por falta de conocimiento procesos tardíos de ejecución del código.
- Plataforma Tindercard no respondía, dejaba de funcionar o la página se caía.

3. Esquema donde describa las tareas que usted definió en el desarrollo del algoritmo.

Esquema donde describa las tareas que usted definió en el desarrollo del algoritmo().

Proceso Configuración. Definir serial, pines. llenar Digitalmente los reloj. Fin proceso.

Proceso cíclico. Imprimir Menu. Sugerir presión de teclas al usuario. Seleccionar opción para el funcionamiento de los leds. Seleccionar la imagen del led. Llamar la funcion publik. Fin proceso.

Proceso inicialización . Iniciar los relojes en 0 y 1. Fin proceso.

Proceso apagado. De 0 a 64 apagar cada led. Inicializar . Fin Proceso.

Proceso patron. Definir num, aux, tamaño, cifra. Pedir letra. Capturar letra. Calcular los dígitos del número ingresado. Declarar arreglo. Ingresar número anterior al arreglo. Fin proceso.

Función grupo8. Recorrer de 0 a 8 . Escribir valores lógicos digitales en el pin ser. Llamar a la función inicialización. Fin proceso.

Funcion orden. Declarar arreglo int fin. Para i = hasta tamano-1 incrementado de a 1. a es espacios en la posicion i-1. fin[a]=1. fin para. fin proceso.

Funcion guardador. Para i=1 hasta 64, aumentar de a 1 . Escribir digital en el puerto serial 1 para encender. Inicializar. Fin proceso.

Funcion imagen. Imprimir información del menu. Seleccionar opción con cambiar o switch. Caso 1:. Para i=0 hasta hasta 7 . Llamar a la funcion patron y la funcion grupo8. Fin ciclo for. break.

función publik. declarar variable tiempo y num-patrones. Imprimir el mensaje para que el usuario ingrese la. cantidad de patrones . Capturar dato con parseint. Mientras num!=0. Ingresar el tiempo de visualizacion. Tiempo=seria.parseint. Fin mientras.

Declarar secuencia como arreglo de 2x2 y de 8 elementos Para i=0 hasta j=num-patrones, i++. Patronnn[8][8]. Llamar imagen(patronn). Para e=0 hasta 8,e++. Para a=0 hasta a=8,a++. Hacer que el arreglo seuencia en la posicon. [(e*1)-1][a]. sea igual a patronn[e-1][a]. Fin ultimo for. Fin penúltimo for. Fin primer for.

Para q=0 hasta q=2,q++. Para m=1 hasta m=num-patrones;m++. Para i=0 hasta i=8,ii++. Para a=0 hasta a=8,a++. Llamar a la funcion grupo8(secuencia[a*m]);. Fin cuarto for. Delay-¿según el el valor que le usuario haya ingresado. Fin tercer for. Fin segundo for. Fin primer for.

4. Algoritmo implementado.

```
//nombre de los pines que estamos usando const int SER =5; const int
reloj-des=3;. const int reloj-salida=4;.
//para el menu.
int opciones;.
void setup(). .
Serial.begin(9600);.
pinMode(SER,OUTPUT);. pinMode(relojdes, OUTPUT);.
pinMode(relojsalida, OUTPUT);.for(int i = 3; i == 5; i ++).
digitalWrite(i,0); //Serial.print("Bienvendio al menu programa del par-
cial");.
void loop() //esto para que no pase de este campo sin el dato a ingregar. Ser-
ial.print(F("bienvenido al menu del parcial 1 de info 2!!!")); Serial.print(F("presiona
1. para invocar la funcion verificacionpresiona 2. para invocar la. funcion ima-
genpresiona 3. para la funcion publik "));. while(Serial.available()==0) . opcio-
nes=Serial.parseInt();.
switch(opciones). case 0: break; case 1: verificacion();. Serial.println(F("todos
los leds funcion correctamente porfavor presione cualquier l para. volver al me-
nu principal ")); while(Serial.available()==0) apagado(); break; case 2: int pa-
tronn[8][8]; imagen(patronn);
break; case 3: publik(); break; default: Serial.println(F(".el dato ingresado no
se contempla en el menu "));
// funciones auxiliares
void inicializacion()
// funcion para inicializar los relojes digitalWrite(relojdes, 0); digitalWrite(relojdes, 1); digitalWrite(relojdes, 0);
digitalWrite(relojsalida, 0); digitalWrite(relojsalida, 1); digitalWrite(relojsalida, 0); void apagado() for (by
void grupo8(int cadena[]) //recorrer las filas dejando datos, pide una cadena
de enteros for(short int e=0; e<8; e++) digitalWrite(SER, *(cadena+e)); iniciali-
zacion();
void patron(int guardado[][8], short int num-arre) long int num, aux;.
short int tamano=0, cifra;.
Serial.print(F("ingrese el numero que representan los leds de la fila: "));.
Serial.print(numarre);.
Serial.print(F(""));
while(Serial.available()==0).
num=Serial.parseInt();.
Serial.print(num);.
Serial.print(F(""));
while(num !=0) Serial.print(F("ingrese el numero que representan los leds pren-
didos ")); while(Serial.available()==0) num=Serial.parseInt(); aux= num; whi-
le(num !=0) cifra= num if(cifra !=9 and cifra !=0) tamano++; num=num/10;
int espacios[tamano],ingresor[8]=0,0,0,0,0,0,0,0;
ingreso(espacios,aux,cifra);.
orden(espacios,tamano,ingresor);.
guardador(guardado,ingresor,numarre);.
```

```

    void ingreso(int espacios[],long int aux,short int cifra ) byte e=0; while(aux!=0) cifra = aux if (cifra !=9 and cifra !=0) espacios[e]=cifra; e++; aux=aux / 10;
    void orden(int espacios[],short int tamano,int fin[]) int a; for(short int i=0; i< tamano ;i++){
        a=espacios[i]-1;
        fin[a]=1;
        void guardador(int arreglodarreglos[][8],int ingreso[],short int num-arre).
        for(byte i= 0 ; i < 8;i++) arreglodarreglos[num-arre][i] = *(ingreso + i);
        void verificacion() //funcion que prende todos los leds para verificar que funcionan for(int i=1;i<64;i++) digitalWrite(SER,1); inicializacion();
        void imagen(int patronn[][8]) Serial.print(F("Menu de la funcion imagen"));
        Serial.print(F("presiona 1. para ingresar tu propio patron")); while(Serial.available()==0)
        opciones=Serial.parseInt();
        switch(opciones) case 1: Serial.print(F("ingrese el numero que representan los leds prendidos ")); Serial.print(F("ejemplo:si usted ingresa 1468, encenderan los led en posicion 1 4 6 8 "));
            for(int i=0;i<8;i++){
                patron(patronn,i);
                grupo8(patronn[i]);
                break;
            }
            void publik().
            int tiempo=0;
            short int num-atrones = 0;
            Serial.println(F("Ingrese la cantidad de patrones que desea: "));
            while(Serial.available()==0).
            num-atrones = Serial.parseInt();
            while (num-atrones < 0).
            Serial.println(F("debes ingresar el numero de patrones que deseas mostrar"));
            Serial.println(F("Ingrese la cantidad de patrones que desea: ")); while(Serial.available()==0).
            num-atrones = Serial.parseInt();
            Serial.println(F("Ingrese los milisegundos entre cada visualizacion entre patrones : ")); while(Serial.available()==0).
            tiempo = Serial.read();
            while (tiempo < 0).
            Serial.println(F("Ingrese el tiempo de visualizacion entre cada patron :"));
            while(Serial.available()==0) tiempo = Serial.parseInt(); int secuencia[8*num-atrones][8]; for(short int i = 0; i <= num-atrones; i++){
                for(short int e=1;e<8;e++){
                    for(short int a=0;a<8;a++){
                        secuencia[(e*i)-1][a] = patronn[e-1][a];
                        for(short int q=0;q<8;q++){
                            for(short int m=1; m <= num-atrones; m++){
                                for(short int i = 0; i < 8; i++){
                                    for(short int a = 0; a < 8; a++){
                                        grupo8(secuencia[(a * m)]);
                                        delay(tiempo);
                                    }
                                }
                            }
                        }
                    }
                }
            }

```

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

El algoritmo implementado fue cambiado múltiples veces dado que al hacer pruebas de escritorios cada vez éstas nos sugerían cambios como en los tipos de datos como punteros y la eliminación de funciones que, aunque reducían el tiempo de ejecución del código y eliminación de líneas de código también, se hacían tardías ya que se llegó a un punto donde habían más de 800 líneas de código y el código editor de Tindercard no permitía la edición. También se dispuso de algunos diagramas de flujo para identificar procesos de las funciones y el comportamiento del proyecto en general; por lo que estas herramientas ayudaron a reducir componentes del hardware y reducción de variables y líneas de código.