

Informe Parcial 1

Informatica 2

Daniela Andrea Gallego Diaz

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

Índice

1. Analisis del problema	2
2. Desarrollo del algoritmo	2
3. Algoritmo implementado	3
4. Problemas de desarrollo	8
5. Evolucion del algortimo	8

1. Analisis del problema

Para la solución de este problema planteado primero se tuvieron en cuenta los requisitos, por lo que se pensó desde el principio ahorrar los pines digitales que fueran posibles, para ello se investigó para saber el mínimo de pines que era posible utilizar; se llegó a la conclusión y decisión de utilizar 3 pines digitales. Luego se debía hacer un análisis para saber cuántos chips utilizar y cómo hacer todas las conexiones necesarias para poder llevar a cabo la solución. Al final se llegó a la conclusión de utilizar 8 chips ya que cada uno controla 8 leds, así que así podría controlar los 64 leds con éxito. Luego se procedió a hacer las conexiones teniendo en cuenta todo lo visto en clase anteriormente. Primeramente analice y me planteé para esta solución hacer un programa que fuera guiando al usuario para que digitara los datos correspondientes y que al hacer un patrón de alguna letra o una animación de esta estos ya estuvieran designados en el programa para que el usuario no tuviera que crear una letra desde cero, aun así también me planteé la opción de que el usuario pudiera ingresar un patrón creado por el mismo. Y así fui desarrollando mi programa logrando así el objetivo.

2. Desarrollo del algoritmo

Tareas designadas en el desarrollo del programa: 1. Hacer las conexiones para que el circuito funcione de manera correcta. Con ya mi análisis y el plan que tenía fui haciendo las conexiones con los 8 chips y los 3 pines digitales de manera que se viera ordenado y se pudiera entender, además de que funcionara correctamente, para probar su funcionamiento utilice un código que prendía todos los leds. 2. Hacer que los leds se encendieran en forma de S. Después de varios intentos pude hacer que mi programa mostrara una S. Para este proceso tuve que analizar con profundidad cómo funcionaba la función shiftout e implementarla. 3. Hacer que el programa mostrara la animación del abecedario. Mi próxima tarea a cumplir era que mi programa pudiera mostrar una animación de todas las letras del abecedario. Luego de muchos intentos y de dibujar bocetos de cómo quería que se vieran las letras y pasar todo esto a números hexadecimales pude por fin cumplir el objetivo. 4. Hacer que el usuario pudiera interactuar con el programa. Para ello empecé a implementar las funciones que previamente en el documento decía que debían crearse. Para poder crear uno por uno tuve que hacer un análisis de cómo iban a funcionar cada uno de ellos y lo que tenía que hacer para poder que lo que tenía en mente funcionara. Empecé a probar varias alternativas e ir cambiando poco a poco lo que no me parecía correcto. Además en este análisis supe exactamente dónde poner los arreglos, apuntadores y dónde utilizar memoria dinámica para la solución de problema. Al final, después de varios intentos con fallas, pude hacer que mi programa funcionara correctamente y cumpliera los objetivos que había planteado desde el principio.

3. Algoritmo implementado

A continuacion esta el codigo implementado en esta solucion del problema:

```
/*
Este programa pide al usuario que ingrese diferentes opciones
para visualizar un patron o secuencia de patrones

Manual de uso:
1.El usuario se encontrara con un menu de entrada, debera digitar el numero de l
2.Al entrar en la primera opcion(Verificacion del funcionamiento de leds) no ten
ningun tipo de dato, solo debe asegurarse que todos los leds se enciendan
3.Al entrar a la segunda opcion tiene dos alternativas: Crear su propio patron o
que ya esta establecido en el programa
3.1 Al elegir la opcion 1(crear patron) debera digitar unos y ceros por cada led
de cada fila. Va a digitar 1 solo si desea que ese led este prendido y va a digi
0 si desea que este apagado. Ejemplo: Yo quiero que el led 7 de la fila 1 este p
entonces digitare 1 cuando me corresponda darle valor a este led
3.2 Al entrar a la opcion 2(ver patron) tendra una lista de letras, lo unico que
hacer es digitar el numero de lista de la letra que quiere ver como patron de le
4.Al entrar a secuencia de patrones le pediran el numero de letras que quiere ve
el tiempo para pasar de una letra a otra y por ultimo los numeros de la lista co
a las letras que desea ver
LISTO!
*/
unsigned int pines[3]= {11, 12, 13};//Arreglo para pines digitales
//Variables necesarias
int patron[8];//Arreglo de 8 elementos no designados (luego este arreglo se llen
int *n;//Apuntador
int Chs;
int opcion;
int num;
int times=0;
float time=1000;//Tiempo entre un patron al siguiente

//Funcion para configurar los pines como salidas
void setup(){
    for (int i=0; i<=2; i++)
        pinMode (pines[i],OUTPUT);
    Serial.begin(9600);
}
//Funcion que recibe 8 valores que convierte a bianrios y los envia a cada dila
void leds(int F1, int F2, int F3, int F4, int F5, int F6, int F7, int F8, float
    shiftOut(pines[0], pines[2], LSBFIRST, F1); //El pin 0 corresponde al pin que
    shiftOut(pines[0], pines[2], LSBFIRST, F2); //El pin 2 corresponde al reloj d
    shiftOut(pines[0], pines[2], LSBFIRST, F3);
```

```

    shiftOut(pines[0], pines[2], LSBFIRST, F4);
    shiftOut(pines[0], pines[2], LSBFIRST, F5);
    shiftOut(pines[0], pines[2], LSBFIRST, F6);
    shiftOut(pines[0], pines[2], LSBFIRST, F7);
    shiftOut(pines[0], pines[2], LSBFIRST, F8);
    digitalWrite(pines[1], HIGH); //Este pin corresponde al Reloj de registro de s
    digitalWrite(pines[1], LOW);
    delay(time);
}
//Funcion que verifica si todo esta en orden encendiendo los leds fila por fila
void verificacion(){
    Serial.println("Debera_ver_como_se_enciende_cada_fila_de_leds");
    leds(0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, time);
    leds(0x00, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, time);
    leds(0x00, 0x00, 0xFF, 0x00, 0x00, 0x00, 0x00, 0x00, time);
    leds(0x00, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00, 0x00, time);
    leds(0x00, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00, 0x00, time);
    leds(0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0x00, 0x00, time);
    leds(0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, 0x00, time);
    leds(0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF, time);
    leds(0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, time);
}
//Funcion que envia a la funcion leds los numeros hexagecimales de la letra o le
void letters(int p, int time){
    if (p==1)
        leds(0x3C, 0x7E, 0x66, 0x66, 0x7E, 0x7E, 0x66, 0x66, time);
    else if (p==2)
        leds(0x7C, 0x7E, 0x66, 0x7C, 0x7C, 0x66, 0x7E, 0x7C, time);
    else if (p==3)
        leds(0x3C, 0x7E, 0x62, 0x60, 0x60, 0x62, 0x7E, 0x3C, time);
    else if (p==4)
        leds(0x78, 0x7C, 0x66, 0x66, 0x66, 0x66, 0x7C, 0x78, time);
    else if (p==5)
        leds(0x3C, 0x7E, 0x60, 0x7E, 0x7E, 0x60, 0x7E, 0x3C, time);
    else if (p==6)
        leds(0x7E, 0x7E, 0x60, 0x7E, 0x7E, 0x60, 0x60, 0x60, time);
    else if (p==7)
        leds(0x3C, 0x7E, 0x60, 0x6E, 0x6E, 0x66, 0x7E, 0x3C, time);
    else if (p==8)
        leds(0x66, 0x66, 0x66, 0x7E, 0x7E, 0x66, 0x66, 0x66, time);
    else if (p==9)
        leds(0x7E, 0x7E, 0x18, 0x18, 0x18, 0x18, 0x7E, 0x7E, time);
    else if (p==10)
        leds(0x7E, 0x7E, 0x18, 0x18, 0x18, 0x18, 0x78, 0x78, time);
    else if (p==11)
        leds(0x63, 0x66, 0x6C, 0x78, 0x78, 0x6C, 0x66, 0x63, time);
}

```

```

else if (p==12)
    leds(0x60, 0x60, 0x60, 0x60, 0x60, 0x60, 0x7E, 0x7E, time);
else if (p==13)
    leds(0x42, 0x66, 0x7E, 0x66, 0x66, 0x66, 0x66, 0x66, time);
else if (p==14)
    leds(0x00, 0x43, 0x63, 0x73, 0x6B, 0x67, 0x63, 0x00, time);
else if (p==15)
    leds(0x3C, 0x7E, 0x66, 0x66, 0x66, 0x66, 0x7E, 0x3C, time);
else if (p==16)
    leds(0x3C, 0x7E, 0x66, 0x7E, 0x78, 0x60, 0x60, 0x60, time);
else if (p==17)
    leds(0x7E, 0xC3, 0xD3, 0xCB, 0XC7, 0X7E, 0X01, 0X00, time);
else if (p==18)
    leds(0x3C, 0x7E, 0x62, 0x62, 0x7C, 0x70, 0x6C, 0x66, time);
else if (p==19)
    leds(0x3C, 0x7E, 0x64, 0x7E, 0x3E, 0x46, 0x7E, 0x3C, time);
else if (p==20)
    leds(0x7E, 0x7E, 0X18, 0X18, 0X18, 0X18, 0X18, 0X18, time);
else if (p==21)
    leds(0x24, 0x66, 0x66, 0x66, 0x66, 0x66, 0x7E, 0x3C, time);
else if (p==22)
    leds(0x66, 0x66, 0x66, 0x66, 0x66, 0x66, 0x3C, 0x18, time);
else if (p==23)
    leds(0x5A, 0xDB, 0xDB, 0xDB, 0xDB, 0xDB, 0xFF, 0x66, time);
else if (p==24)
    leds(0x00, 0x66, 0x66, 0x3C, 0x18, 0x3C, 0x66, 0x66, time);
else if (p==25)
    leds(0x6C, 0x6C, 0x6C, 0x3C, 0x1C, 0x4C, 0x7C, 0x38, time);
else if (p==26)
    leds(0x7E, 0x7E, 0x06, 0x0C, 0x18, 0x30, 0x7E, 0x7E, time);
}
//Funcion que le permite al usuario elegir o crear un patron
void imagen(int op){
    Serial.println(" Digite el numero 1 si quiere crear su propio patron");
    Serial.println(" Digite el numero 2 si desea utilizar la opcion predeterminada");
    Serial.println("(Esta opcion permite visualizar letras)");
    delay(10000);
    while (Serial.available()>0){
        op= Serial.parseInt();
        if (op==1){
            n= patron; //La variable n apunta al arreglo patron de 8 elementos
            Serial.println("Por cada fila (8 filas) tenemos 8 leds");
            Serial.println("Si desea que un led se prenda digite 1, por el contra");
            Serial.println(" Digite los numeros correspondientes al numero de fila");
            delay(10000);
            for (int i=0; i<8; i++){

```

```

Serial.print(" Fila");
Serial.println(i+1);
delay(1000);
for (int j=0; j<8; j++){
    Serial.print("Led");
    Serial.println(j+1);
    delay(1950);
    while (Serial.available()>0){
        op=Serial.parseInt();
        if (op==1){//Los unos y ceros que ingresa el usuario se convier
            if (i==0)
                num+=1;
            else if (j==1)
                num+=2;
            else if (j==2)
                num+=4;
            else if (j==3)
                num+=8;
            else if (j==4)
                num+=16;
            else if (j==5)
                num+=32;
            else if (j==6)
                num+=64;
            else if (j==7)
                num+=128;
        }
    }
    n[i]=num;//Se va sumando a cada posicion del apuntador el valor dec
    num=0;
}
Serial.println("Ahora debera ver el patron que ha creado");
leds(n[0], n[1], n[2], n[3], n[4], n[5], n[6], n[7], time);

}
else if (op==2){
    Serial.println("En esta seccion podra ver el patron de una letra");
    Serial.println("A continuacion se le mostrara la lista enumerada de la
        Serial.println("1. A     2. B     3. C     4. D     5. E     6. F
    Serial.println("9. I     10. J     11. K     12. L     13. M     14. N     15. O
    Serial.println("17. Q     18. R     19. S     20. T     21. U     22. V     23. W
    Serial.println("25. Y     26. Z");
    Serial.println("Digita el numero de la letra correspondiente");
    delay(10000);
    while (Serial.available()>0){

```

```

        op= Serial.parseInt();
        Serial.println("Ahora_debera_ver_el_patron");
        letters(op, time);//Se envia a la funcion Letters el numero que
    }
}

}
}
//Funcion que le permite al usuario escoger varios patrones y ver una animacion
void publik(int op, int num, float t){
    Serial.println("");
    Serial.println("En_esta_opcion_podra_ver_animaciones_de_letras");
    Serial.println("Usted_debera_digitar_las_letras_para_cada_patron");
    delay(10000);
    Serial.println("Por_favor_digite_el_numero_de_elementos_que_quiere_en_su_secuencia");
    Serial.println("Aclaracion: _cada_letra_corresponde_a_un_elemento");
    delay(10000);
    while (Serial.available()>0){
        num= Serial.parseInt();
        Serial.println("Ahora_digite_el_tiempo_en_el_que_quiere_que_se_muestre_cada_letra");
        Serial.println("Aclaracion: _El_tiempo_debe_ser_en_segundos");
        delay(10000);
        while (Serial.available()>0){
            t= Serial.parseFloat();
            t*=1000;//Conversion de segundos a milisegundos
            Serial.println("Ahora_se_le_mostrara_la_lista_de_letras_disponibles:");
            Serial.println("1._A_____2._B_____3._C_____4._D_____5._E_____6._F_____7._G_____");
            Serial.println("9._I_____10._J_____11._K_____12._L_____13._M_____14._N_____15._O_____");
            Serial.println("17._Q_____18._R_____19._S_____20._T_____21._U_____22._V_____23._W_____");
            Serial.println("25._Y_____26._Z");
            Serial.println("Por_favor_digite_los_numeros_correspondientes");
            Serial.println("a_las_letras_que_quiere_mostrar_en_cada_uno_de_los_patrones");
            n=new int [num];//Apuntador con memoria dinamica
            delay(10000);
            for (int i=0; i<num; i++){
                Serial.print("Patron_numero_");
                Serial.println(i+1);
                delay(10000);
                while (Serial.available()>0)
                    n[i]=Serial.parseInt();//Se suma cada numero a una posicion de la memoria
            }
            Serial.println("Ahora_debera_ver_la_animacion");
            while(times<5){
                for(int i=0; i<num; i++){
                    letters(n[i], t);//Se envia a la funcion Letters cada posicion de la memoria
                }
                times++;//Variable que suma 1, sirve para que la animacion se repita
            }
        }
    }
}

```



```

    }
    delete [] n; //Se libera el espacio de esta variable
  }
}

}
//Funcion que muestra el menu de opciones
void loop(){
  Chs=0; //Se utiliza para guardar el numero que digito el usuario para el
  Serial.println("Bienvenido");
  Serial.println(" Digite 1 para verificar que todos los leds funcionan cor");
  Serial.println(" Digite 2 para mostrar un solo patron");
  Serial.println(" Digite 3 para mostrar una serie de patrones");
  delay(10000);
  while (Serial.available()>0){
    Chs= Serial.parseInt();
    if (Chs==1)
      verificacion();
    else if (Chs==2)
      imagen(opcion);
    else if (Chs==3)
      publik(opcion , num, time);
  }
}

```

4. Problemas de desarrollo

En la realizacion de este problema hubieron varios inconvenientes, pero todos se trataban de falta de analisis del problema en si, así que luego de que algo fallara en mi programa me tomaba el tiempo de analizar el porque de ese error y como se haria para que funcione de la manera correcta.

5. Evolucion del algoritmo

Como ya he explicado anteriormente, analizar cada error ayudo a la evolucion del programa, de pasar de simplemente encender todos los leds, a poder formar letras, luego animaciones y luego poder hacer que el programa interactuara con el usuario y que todo se desarrollara de manera correcta. (En la seccion de desarrollo se explica todo mucho mas profundamente).