

# PARCIAL 1, INFORMÁTICA 2.

## Informados S.A.S.

**Fabian Camilo Falla Ramírez**

*Estudiante Ingeniería de Telecomunicaciones  
Universidad de Antioquia  
Medellín, Antioquia*

FABIAN.FALLA@UDEA.EDU.CO

**Sofia Marín Cacante**

*Estudiante Ingeniería de Telecomunicaciones  
Universidad de Antioquia  
Medellín, Antioquia*

SOFIA.MARINC@UDEA.EDU.CO

### 1. Análisis del problema.

En el parcial se presenta un reto grande que consiste en varios problemas menores, en el primero en el que nos enfocamos fue en la disposición de los componentes, el uso de los pines y las recomendaciones dadas, tales como investigar el uso del integrado 74HC595, El integrado tiene un registro de desplazamiento serie-paralelo entonces se accede a un pin de entrada y los datos de entrada se van desplazando a todos los pines de salida, lo cual resulta bastante útil para ahorrar en el uso de pines, incluso, como reto personal tratamos de utilizar menos cantidad de pines. Para el uso del integrado vimos algunos videos y trabajamos con el datasheet, cada integrado “controla” una fila de LED’s y están conectados entre sí, es una conexión cascada.

Las funciones las fuimos analizando en el orden en el que se nos iban pidiendo en la presentación del parcial, secuencia que tenía mucho sentido para ayudarnos a manejar cierto orden y haciendo más fácil el proceso, algo fundamental fue identificar en el código la matriz de LED’s 8x8, además de esto, verificar el funcionamiento de los 64 LED’s, para esto se le da la entrada de encendido a cada uno de los LED’s, para esta y la mayoría de funciones utilizamos mucho el uso de ciclos ya que a la hora de trabajar con patrones, muchos datos o en general matrices, es muy útil, además que ahorra muchas líneas de código.

Adicional a las funciones pedidas, pensamos que como el usuario debe ingresar varios patrones por medio del puerto serial, una buena opción sería informarle como hacerlo, entonces en el menú principal, se agregó una opción para acceder al menú.

### 2. Esquema de tareas a realizar.

- a) Organización y disposición de los componentes.
- b) Investigación del uso de integrados, conceptos de nodos y uso de la protoboard, configuración de pines de salida y uso de Arduino.
- c) Reducir el uso de pines utilizados.
- d) Implementar lo adquirido en la investigación.
- e) Verificar el funcionamiento de los 64 LED’s.
- f) Diseñar una función que nos permita recibir matrices.

- g) Crear una función que permita imprimir o enviar la señal de encendido de la matriz predeterminada.

### 3. Algoritmos implementados y diseño de funciones.

El algoritmo principal descrito en el loop del código primero invoca la función publik, que fue declarada antes, esta función contiene un menú en el cual el usuario puede decidir que desea hacer al ingresar la opción por el monitor serial. Se le presentan cuatro opciones, en la primera opción se invoca la función verificación, que comprueba que todos los LED's y las conexiones realizadas, están correctas y se prenden todos los LED's. La segunda opción invoca un par de funciones para que le usuario pueda ingresar fila por fila el patrón que desee, eso lo hace mediante unos y ceros, y se le informa al usuario como hacerlo. En la tercera opción se invocan otras funciones que se encargan de pedirle al usuario la cantidad de patrones que desea ver en pantalla o la matriz de LED's y un tiempo en milisegundos entre esos patrones y así se muestra una secuencia de los patrones que ingresó el usuario. La última opción es la del manual en la que simplemente se imprime por el monitor serial la forma en la que el usuario debe ingresar las filas de los patrones que desea.

### 4. Funciones requeridas implementadas

```
void verificacion(){
    for(int k=0; k<64; k++){
        digitalWrite(SER, 1);
        relojregistro(SRCLK); //Activar el flanco de subido
        relojregistro(RCLK); //Activar el flanco de subido del registro de salida
    }
    delay(5000);
    for(int k=0; k<64; k++){
        digitalWrite(SER, 0);
        relojregistro(SRCLK);
        relojregistro(RCLK);
    }
}

void imagen(){
    int m;
    for (int i=7; i>=0; i--){
        for (int j=7; j>=0; j--){
            m=*(*(puntero_matriz+i)+j);
            digitalWrite(SER, m);
            relojregistro(SRCLK);
            relojregistro(RCLK);
        }
    }
}
```

```

void publik() {
    Serial.println("Bienvenido al menu de inicio");
    Serial.println("Ingrese el numero de la opcion que desea realizar");
    Serial.println("1. Verificar el funcionamiento de los 64 LEDs");
    Serial.println("2. Mostrar un patron");
    Serial.println("3. Secuencia de patrones");
    Serial.println("4. Manual de uso");
}

```

### Funciones auxiliares que sirvieron en las principales

```

void definematrizbidi() {
    puntero_matriz = new int*[8];
    for(int i=0; i<8; i++){
        puntero_matriz[i] = new int[8];
    }
    for(int j=0; j<8; j++){
        for(int k=0; k<8; k++){
            *(*(puntero_matriz+j)+k)=0;
        }
    }
}

```

```

void definematriztridi(int cpatrones){
    puntero_matriz3d = new int**[cpatrones];
    for(int i=0; i<cpatrones; i++){
        puntero_matriz3d[i] = new int*[8];

        for(int j=0; j<8; j++){
            puntero_matriz3d[i][j] = new int[8];
        }
    }
    for(int i=0; i<cpatrones; i++){
        for(int j=0; j<8; j++){
            for(int k=0; k<8; k++){
                *(*(puntero_matriz3d+i)+j)+k)=0;
            }
        }
    }
}

```

```

void RecibirMatrizbidi(){
    char num_vol = 49;
    String valor;
    long int fila;
    Serial.println("Para ingresar su patron, debe tener en cuenta que:");
    Serial.println("- El numero 1 se reflejara como el led encendido ");
    Serial.println("- El numero 0 se reflejara como el led apagado ");
    Serial.println("- Debe ingresar fila por fila");
    for(int i=0; i<=7; i++){
        Serial.print("Ingrese la fila ");
        Serial.println(i+1);
        while (Serial.available()==0){}
        valor = Serial.readStringUntil('\n');
        valor = num_vol + valor;
        fila = valor.toInt();
        for (int j=7; j>=0; j--){
            *((puntero_matriz+i)+j)=fila%10;
            fila=fila/10;
        }
    }
}

```

```

void RecibirMatriztridi(int cpatrones){
    char num_vol = 49;
    String valor;
    long int fila;
    Serial.println("Para ingresar su patron, debe tener en cuenta que:");
    Serial.println("- El numero 1 se reflejara como el led encendido ");
    Serial.println("- El numero 0 se reflejara como el led apagado ");
    Serial.println("- Debe ingresar fila por fila");

    for(int i=0; i<cpatrones; i++){
        Serial.print("A continuacion ingrese la matriz ");
        Serial.println(i+1);
        for(int j=0; j<=7; j++){
            Serial.print("Ingrese la fila ");
            Serial.println(j+1);
            while (Serial.available()==0){}
            valor = Serial.readStringUntil('\n');
            valor = num_vol + valor;
            fila = valor.toInt();
            for (int k=7; k>=0; k--){
                *((*(puntero_matriz3d+i)+j)+k)=fila%10;
                fila=fila/10;
            }
        }
    }
}

```

```

void relojregistro(int parametro){
    digitalWrite(parametro, 0);
    digitalWrite(parametro, 1);
    digitalWrite(parametro, 0);
}

```

```

void auxiliar(int tpatrones, int cpatrones){
    int m;
    for (int i=0; i<cpatrones; i++){
        for (int j=7; j>=0; j--){
            for(int k=7; k>=0; k--){
                m=*( (* (puntero_matriz3d+i)+j)+k);
                digitalWrite(SER, m);
                relojregistro(SRCLK);
                relojregistro(RCLK);
            }
        }
        delay(tpatrones);
    }
}

```

## 5. Problemas de desarrollo que enfrentamos

Conectando los componentes de la forma planteada en la primer solución se detectó que, al prender todos los LED's (de forma "manual") se quemaban los circuitos integrados, no permitían el desarrollo del programa y se presentaban inconvenientes a la hora de realizar algunos patrones básicos, además la programación de las funciones, con el tiempo que teníamos fue un poco complejo, no alcanzamos a presentar una solución no trivial que era lo ideal ya que nos quedamos cortos de tiempo al “traducir” el código de qt a Arduino, el tema de la secuencia también fue un poco duro además todo el tema circuital, de conexiones, asegurar que nada se conectara con lo que no debía, tratar de mantener un montaje organizado fue realmente un reto.

## 6. Evolución de la solución y consideraciones a tener en cuenta

Los planes nos cambiaron bastante en la implementación que teníamos pensado, siempre miramos el problema con optimismo y aunque eso nos ayudó a no desistir ante los problemas también nos dejó cortos en otros. Al principio de la solución del parcial, estuvimos experimentando diferentes maneras de verificar el funcionamiento del montaje, esto nos permitió conocer mucho más sobre circuitos integrados y la sintaxis del lenguaje de Arduino. En el transcurso de los días veíamos que no le podíamos dedicar el tiempo suficiente con los demás parciales y entregas, por esto fuimos adelantando ideas de funciones sin tener completamente verificado el montaje. Finalmente, logramos mostrar un patrón de matrices reutilizando código de la función imagen (), reservando espacio de memoria para un array tridimensional y pudimos con ayuda de esto finalizar el programa.

Entre las consideraciones a tener en cuenta está que no logramos generar los patrones por medio de secuencias entonces el usuario tiene bastante protagonismo en el ingreso de dichos patrones, por esto decidimos dejarle un pequeño manual de ingreso.