**INFORME PRIMER PARCIAL**

Andres Felipe Restrepo López

Imagen que contiene Patrón de fondo

Descripción generada automáticamente

Universidad de Antioquia

2023

**Informe sobre proceso del trabajo de informática 2.**

­**-**Primero se procede a repasar los conceptos vistos en clase y se repite el ejerció que el profesor realizo el día 30/08/2023, para afianzar los conceptos y recordar el funcionamiento de Arduino.

Texto

Descripción generada automáticamente

[**https://www.tinkercad.com/things/ke9un5DtuRU?sharecode=Wwn4YjNOFoVF5qapq0QC6NIvtoquhcW9xX\_1tu3fyfA**](https://www.tinkercad.com/things/ke9un5DtuRU?sharecode=Wwn4YjNOFoVF5qapq0QC6NIvtoquhcW9xX_1tu3fyfA)

**-**Se busca información de este microcontrolador, para poder entender su funcionamiento.

Luego de investigar, y ver varios videos, se concluye que es muy útil ya que gracias a este controlador es posible encender una fila de 8 leds gracias a sus 8 salidas.

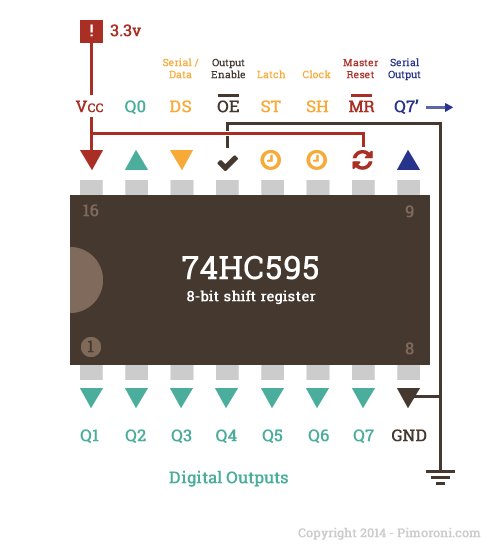
entre sus propiedades están:

**El DS**: El cual nos permite ingresar 0 y 1, a través de pulsaciones las cuales permiten habilitar o no las 8 salidas.

**El SH:** Que permite ir avanzando bit por bit, hasta completar los 8 bits.

**ST:** El cual permite la ejecución final de los bits registrados.

Gracias a la investigación se procede a crear un modelo en tinkercad para comprobar su funcionamiento.



Diagrama, Esquemático

Descripción generada automáticamente

Diagrama, Esquemático

Descripción generada automáticamente

<https://www.tinkercad.com/things/hCR6gSjDbb4?sharecode=bA8AiU0IJyl43MAQP-Wu4kFHDWgzVG0V8-wbinA1voY>

**-**Se procede a hacer la primera implementación en tinkercad, utilizando un Arduino y el integrado 74HC595, en primera instancia se planea utilizar la propiedad de este integrado, para hacer una cascada de 8 de estos, para poder formar las 8 filas de los 64 leds.

Se planea utilizar los puertos del Arduino de la siguiente manera:

-El pin 4 es el DS(Serial Data).

-El pin 3 es el SH(Clock).

-El pin 2 es el ST(Latch).

De esta manera podremos manipular y controlar los leds según la necesidad requerida.

En él .cpp de este repositorio de la fecha 16/09/2023, se podrá ver una de las primeras implementaciones a las que se logro llegar, esta ves se manipulo una fila de 8 leds para poder ir familiarizándome y hacer pruebas.

Se logro crear un primer prototipo de código que permitía comprobar el funcionamiento de los leds, y dibujar una figura que se deseara.

Imagen de la pantalla de un celular

Descripción generada automáticamente con confianza media

<https://www.tinkercad.com/things/lNQ0e91NL7v?sharecode=tblMOxz9G68nl2ZBz1fmp_AjZGKZgCkF2GDT1Pc2YDA>

**-**Se procede a conectar 8 74HC595 en cascada para poder encender los 64 leds, realizando las conexiones apropiadas.

En esta conexión se debe conectar el puerto Q7’ de primer controlador al puerto DS del segundo controlador y así sucesivamente.  
los puertos ST y SH se conectan en paralelo con los otros 74HC595.

Diagrama, Esquemático

Descripción generada automáticamente con confianza media

Luego se organizan los leds de manera que queden organizados por filas y columnas formando una pequeña pantalla de 8x8 y se organiza la parte del código que tenía implementada para poder utilizar los 64 leds.

Diagrama, Esquemático

Descripción generada automáticamenteUn conjunto de letras negras en un fondo blanco

Descripción generada automáticamente con confianza media

<https://www.tinkercad.com/things/6jTQK9Wu7fH?sharecode=mQ24OmKHufzweNN_DCyA6kYamk6JqHvLO4MkDUBHhJ4>

**-**Se procedió a seguir y avanzar en el código y corrigiendo partes de este.   
Se empieza a plantear la implementación de la función de patrones, y se logra progresar en la formulación y creación de los 2 primeros.

Hasta el momento ya tengo unas funciones establecidas las cuales son:

**void verificacion();** Esta función me permite encender y apagar la pantalla según lo indique el usuario, con un numero de parpadeos y segundos establecidos.

**void imagen(int arreglo\_leds[]);** Esta función me permite crear una imagen, en la cual es usuario tiene que ingresar el estado de los 64 leds uno por uno, indicando si quiere que el led este encendido o pagado, está función recibe como parámetro un arreglo entero de 64 elementos, cullo contenido está formado por ceros y unos, los cuales conforman la pantalla de leds.

**void activacionSH();** Esta función, activa el pin SH del integrado 74HC595, lo que me permite ir avanzando en el registro de leds encendidos o apagados.

**void activacionST();** Esta función, activa el pin ST del integrado 74HC595, lo que me permite ejecutar y encender los leds, según el caso dado.

**void patrones(int arreglo\_leds[]);** Esta función permite ingresar a un menú el cual tiene registrado los 4 tipos de patrones diferentes propuestos en el parcial. Hoy se logra progresar en el avance de los 2 primeros patrones, está función recibe como parámetro un arreglo entero de 64 elementos, cullo contenido está formado por ceros y unos, los cuales conforman la pantalla de leds.

**void patron1(int arreglo\_leds[]);** Esta función, me permite crear el primer patrón solicitado, esta función recibe como parámetro un arreglo entero de 64 elementos, cullo contenido está formado por ceros y unos, los cuales conforman la pantalla de leds.

**void patron2(int arreglo\_leds[]);** Esta función, me permite crear el segundo patrón solicitado, está función recibe como parámetro un arreglo entero de 64 elementos, cullo contenido está formado por ceros y unos, los cuales conforman la pantalla de leds.

**void ejecucion\_leds(int arreglo\_leds[]);** Esta recibe un arreglo de 64 elementos y procede a hacer la ejecución para poder mostrar los leds, si el elemento es en cero apaga el led y si está en uno lo enciende.

En él .cpp está el progreso de los avances.

<https://www.tinkercad.com/things/6jTQK9Wu7fH?sharecode=mQ24OmKHufzweNN_DCyA6kYamk6JqHvLO4MkDUBHhJ4>

**Principales dificultades y su solución:**

**\*Problemas con la corriente en los leds e integrados:**

Al principio tuve problemas con la corriente en el circuito, por que al encender todos los leds el integrado se quemaba.

La solución fue investigar y poner resistencias las cuales me permitieron solventar este problema.

**\*El cambio de algunas palabras en el Arduino:**

Al principio, algunas funciones no eran validas tales como el cin o el cout, y estas son muy importantes ya que permiten la interacción de Arduino con el usuario tanto para mostrar como para ingresar datos.

Para la corrección se procedió a buscar el nuevo tipo de comandos permitidos y se encontró la solución con los siguientes comandos.  
**Serial.println:** Que permite mostrar en pantalla.

**Serial.parseInt():** Que permite ingresar datos.

**\*Que Arduino esta constante mente en loop:**

Esta característica significó un reto, ya que, al estar en un ciclo permanente, tenia problema con las variables ya que estas en ocasiones se modificaban, sin yo querer esta acción o simplemente no dejaba ingresar datos por que el código no se pausaba.

Investigando logre encontrar **while(!Serial.available());** el cual no permitía que el código no siguiese ejecutándose, hasta que el usuario no ingresara un dato en consola.

**-**Se procede a terminar de implementar los 2 últimos patrones del punto 4 y se realizan diferentes pruebas en tinkercad.

Estas funciones son llamadas:  
**void patron3();**

**void patron4();**

A continuación, se continua en el punto 5 del parcial el cual consiste en crear una función llamada **publik** la cual permite integrar todos los puntos anteriores. Se realizan pruebas y se verifica el funcionamiento.

en el archivo .cpp se pueden encontrar los cambios realizados.

**-**Se procede a agregar restricciones al código, las cuales permiten que el usuario no ingrese otros números diferentes en el menú de opciones y una restricción mas en la función de imagen para que solo admita un 0 para apagar un led o 1 para encender el led.

Luego se crea la función: **transferencia\_de\_matriz\_a\_arreglo(int matriz[][8],int arreglo\_leds[]);** la cual se implementa en el punto 4 esta nos ayuda a pasar una matriz la cual tiene dibujado los diferentes patrones, y se trasfiere a un arreglo de 64 elementos la cual posterior mente es ejecutada para mostrar los 4 tipos de patrones.

**-**Ya teniendo el código terminado, procedo a corregir y a trabajar con la memoria dinámica y punteros, para los cuales son requeridos en el parcial y luego los cambios son aplicados en el .cpp