**Informe de proyecto primer parcial**

**Mesa Roldan José David**

**Mesa Roldan Carlos Andrés**

**Análisis del problema:**

Con la lectura del parcial y los problemas propuestos en el mismo, identificamos que se presenta un caso complejo el cual debe ser procesado e investigado de forma adecuada, considerando implementar la plataforma de Tinkercard y la implementación del lenguaje C++ para una solución tanto eficiente como correcta a la situación formulada.

Se formula un caso hipotético del uso de leds para apoyar fachadas llamativas con Arduino, buscando que estos presenten patrones ya predeterminados como también algunos que el mismo usuario quiera mostrar para comprobar un correcto funcionamiento además para cumplir tareas específicas; el usuario busca que el sistema muestre los patrones y funcione adecuadamente, por lo tanto se debe considerar cumplir con un hardware bien estructurado y adecuado, como también un código claro y eficiente con el cual se cumplan las expectativas de tiempo entre cada parpadeo de los leds como los diversos patrones que el usuario quiera ingresar por el puerto serial.

**Tareas definidas para el desarrollo:**

Buscando la eficiencia y un correcto desarrollo del problema planteado, se buscó un paso a paso con el fin de tratar el parcial de manera modular, asignando tareas claras y precisas en funciones, tanto que se pidan por parte de los docentes, como las que consideremos necesarias:

1. Comprender lo que se pide durante el trabajo, ya que un mejor entendimiento del problema ayuda a que haya una solución más clara y precisa.
2. Analizar detalladamente el funcionamiento de la plataforma Tinkercard ya que este nos ayudara a desarrollar toda la parte del hardware y los algoritmos que nos ayudaran a dar solución. Además de que es allí donde se harán las pruebas a los distintos cambios que se propongan durante la etapa de diseño.
3. Comprender el funcionamiento de la plataforma de desarrollo Arduino. Aspirando a tener claridad en el uso de los distintos puertos, tanto digitales como análogos del dispositivo. Además de identificar sus particularidades con respecto al lenguaje C++.
4. Implementar una función principal, a la cual se tenga acceso inmediato una vez se inicialice el programa y a partir de esta llamar a las funciones necesarias para el correcto desarrollo del programa.
5. Tener total claridad acerca del uso de matrices, punteros, memoria dinámica y todos los distintos requerimientos que se solicitan en el parcial de manera prioritaria.
6. Consultar acerca de la respectiva polarización de circuito integrado 74HC595, su adaptación a Arduino y adaptarlo a las necesidades que se tienen en la realización del parcial.
7. Realizar un código modular, que ayude a la legibilidad del mismo.

**Algoritmos implementados:**

Se procuro tener un código modular, el cual realiza una serie de llamados a distintas funciones, las cuales se numeran a continuación.

* setup(): Para la configuración inicial del programa.
* loop(): Función principal o de entrada del programa.
* publik(): Mostrará cada una de la opciones disponibles para proceder en el algoritmo
* verificacion(): Endenderá y apagará la matriz de leds, con un tiempo y número de prepeticiones especificados por el usuario.
* apagarLeds(): Hará parte de la función de verificación.
* encenderLeds(): Hará parte de la función de verificación.
* imagen(): Usada para captar la imagen deseado por el usuario, mediante la captación de cadenas de 8 bits.
* secuencias(): Lllamara a las 4 funciones que acontinuación aparecen.
* secuenciaRombo()
* secuenciaX()
* secuenciaCuadrados()
* secuenciaFlecha()
* binarioAEntero: Realizará una transformación de binario de 8 bits a decimal, para posteriormente enviar información a los 74HC595.

**Problemas de desarrollo que afrontó:**

Como en todo proyecto a desarrollar, los problemas se hicieron presente de forma frecuente conforma se iba progresando y avanzando en las tareas que se tenían que completar. Como lo fue en un comienzo de buscar la ayuda e investigar el montaje y la configuración necesaria para el repositorio de GitHub, ya que era una tarea bastante ardua y era el comienzo de todo el proyecto.

En un segundo momento se presentaron los problemas para el uso de Tinkercard ya que esta plataforma presentaba una serie de dispositivos en los cuales no se tenía un conocimiento previo, pero apoyándonos de la clase que se nos brindó y los recursos en las diversas plataformas como Google y YouTube encontramos diversas formas del montaje cada vez más eficientes en búsqueda de un mejor programa.

Finalmente, y es en la etapa que nos encontramos es empezar a implementar los algoritmos necesarios con el lenguaje de C++ para poder buscar eficiencia y un funcionamiento correcto, tal vez en esta hay más problemas ya que a veces el programa puede romperse o se busca que este sea lo más completo para brindarle al usuario una experiencia a la altura. Aunque con la presentación de nuevos y diversos problemas, podemos encontrar soluciones que mejoren adecuadamente el programa completo.

Realizar un correcto merge, similar a realizado por el docente. Sin embargo, los merges son bastante eficientes si se realizan sobre archivos de texto plano, incluso si llegan a presentar problemas. Cuando se realiza un merge con archivos binarios, como en el que se albergo el informe, se presentan muchos problemas.

**Evolución de la solución y consideraciones en la implementación:**

Con la compresión mucho más amplia del problema y con el poco desarrollo que le hemos venido dando, encontramos cada vez más soluciones que pueden ser más eficientes y ese es siempre el objetivo que se busca del trabajo. Poco a poco encontramos que montando un sistema de leds simples se puede solucionar y con dispositivos sencillos, pero con un entendimiento adecuado y con asesoramiento hemos encontrado que con un hardware mucho más avanzado e implementado un código cada vez más eficiente, por medio de punteros y matrices, el problema se puede adecuar y desarrollar para una solución mucho más satisfactoria en cuanto a la entregaba que en un caso real se le debería hacer al cliente. A la hora de la entrega se puede considerar que está sujeta a cambios en pro del mejoramiento del sistema y que cada vez se busque estar más satisfecho con el producto final, y probablemente será una solución avanzada, pero buscando que en ningún momento se incumplan las restricciones ya señaladas en el parcial.

Se desea realizar el montaje del circuito, sin embargo, en necesario comprender el funcionamiento del circuito integrado a utilizar 74HC595. A continuación, se muestra la polarización del 74HC595, en la ***ilustración 1.***

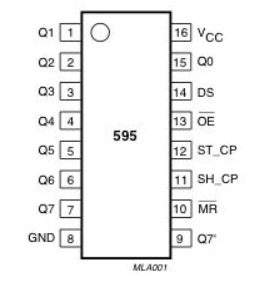


Ilustración 1: Polarización 74HC595.

Antes que nada, se usaran los pines 10 y 16, los cuales reciben voltaje. El PIN 16 como se puede observar en la ilustración es la encargada de alimentar el circuito integrado. El voltaje mínimo es de 2v y el máximo es de 6v. El PIN 10 recibe pequeños pulsos de energía, los cuales reiniciaran a los valores predeterminados del circuito, es decir cada bit será igual a 0, en el caso de que así se requiera.

El circuito integrado es capaz de manipular cadenas de 8 bits, por tanto, es necesario utilizar 8 dispositivos 74HC595, con el fin de representar una matriz de 8 x 8.



Ilustración 2: LED

Con el fin de representar cada una de las posiciones de la matriz de 8x8, se hará uso de una serie de LEDS, como se puede apreciar en la ***ilustración 2.*** El ánodo se identifica como la salida que tiene un dobles, es decir, la que se localiza a la derecha de la ilustración. El ánodo será conectado a la Fuente de voltaje, la cual será dado por el 74HC595. El cátodo será conectado a la salida de GND del circuito integrado o PIN de tierra que a su vez se conecta a la salida GND del Arduino.

El voltaje de salida por cada uno de los registros de desplazamiento oscila entre 0 y 5V. Por tanto, se hace uso de una serie de resistencias, de 470 Ohm, como se puede observar en la ***ilustración 3.***

Diagrama

Descripción generada automáticamente

Ilustración 3: Salidas y uso de resistencias.

Si bien la cantidad de corriente para cada LED varía según el flujo de la ejecución del programa, es importante resaltar que en la anterior ecuación se da un aproximado a la cantidad de corriente que puede llegar a tener un LED. Un LED por lo general puede funcionar entre 5mA y 20mA, sin embargo, se insiste, esto es una variable que depende de la configuración y el código a utilizar. El simulador no tuvo problemas a utilizar esta configuración.

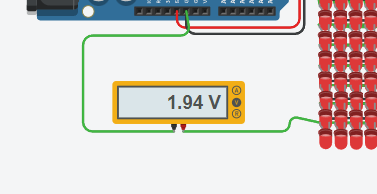


Ilustración 4: Voltaje a cada LED.

En la ***ilustración 4*** se puede verificar el voltaje que llega a cada LED durante la ejecución de la función verificación (). El voltaje esta dentro de los estándares para el buen funcionamiento de los LEDS.

En cuanto a un avance significativo del código, se tiene una forma de enviar los datos de manera óptima. Utilizando memoria dinámica se envía la información de manera ordenada a cada registro de desplazamiento, se le envía un decimal, el cual será transformado por el circuito integrado a un binario de 8 bits, Cada pin de salida del 74HC595 tomará un bit, que corresponderá a un estado, sea arriba o abajo (encendido o apagado), el reto yace en el proporcionar el decimal adecuado para cumplir con la expectativas solicitada en la guía.

Cada miembro del grupo estará encargado de realizar dos funciones, las cuales tendrán su lógica particular, pero tendrán en común la forma en la que recepción y envía la información al circuito integrado 74HC595. José David se hará cargo de la función que mostrará la X y la flecha. Carlos, tendrá la función del rombo y del cuadrado. Las demás funciones se harán en conjunto, durante las reuniones del equipo de trabajo.