



Parcial I

Informatica II Sistema de comunicación

Autores

Wilmer Banquez Flórez, David Arango Pineda y Juliana
Martínez Hurtado

Profesores

Augusto Enrique Salazar Jiménez
Jonathan Ferney Gómez Hurtado

FACULTAD DE INGENIERÍA
2022

Índice

Lista de Figuras	III
1. Objetivos	1
1.1. Objetivo general	1
1.2. Objetivos específicos	1
2. Introducción	1
3. Marco Teórico	3
4. Análisis del problema	5
4.1. Transmisión	5
4.2. Descriptación	5
4.3. Recepción	5
4.4. Visualización	5
5. Marco experimental	7
6. Implementación del chip 74HC595	9
7. Desarrollo del diseño final	11
8. Conclusiones	13
9. Bibliografía	15

Lista de Figuras

5.1.	Diseño Final	7
5.2.	Diseño Final	8
5.3.	Diseño Final	8
6.1.	Circuito Ejemplo 1.	9
6.2.	Circuito ejemplo 2.	10
6.3.	Código del segundo ejemplo.	10
7.1.	Diseño Final	11

Capítulo 1

Objetivos

1.1. Objetivo general

Desarrollar un sistema que permita la comunicación y la descriptación de un mensaje en Tinkercard para posteriormente visualizarlo en una pantalla LCD.

1.2. Objetivos específicos

- Implementar los conocimientos de c++ y tinkercad de forma eficiente.
- Lograr una comunicación óptima entre arduinos utilizando el chip 74HC595.
- Estructurar un sistema basado principalmente en hardware.

NADA ES LO QUE PARECE

RESUMEN

A la hora de implementar un sistema en Arduino que dé con la solución al problema se deben tener en cuenta factores como diseño, componentes, eficiencia, entre otros. Precisamente el objetivo de este informe es detallar a fondo la cantidad de pasos y percances que se tuvieron, en este caso en un escenario hipotético en el cual se necesita enviar un mensaje encriptado desde una PC 1 y que sea desencriptado mediante una bandera para que llegue el mensaje real a la PC 2 y posteriormente visualizarlo

Capítulo 2

Introducción

El chip 74HC595 es un circuito impresor, concretamente es un registro de desplazamiento de 16 pines que sirve para convertir datos de serie a paralelo de una forma eficaz y sencilla.[1] En el siguiente informe se encuentran dos ejemplos de cómo utilizar dicho circuito, después detallamos la utilidad que podemos darle para solucionar la implementación del sistema de encriptación en los dos arduinos. Por último se plantea un posible diseño.

Capítulo 3

Marco Teórico

Los circuitos integrados son pequeños circuitos electrónico usados en tareas específicas acompañados de otros componentes para así consolidar un sistema más complejo.

”Han hecho posible la fabricación del microordenador o microcomputadora. Sin ellos, los circuitos individuales y sus componentes ocuparían demasiado espacio como para poder conseguir un diseño compacto.”[2] La tarea de éstos no es diferente al aplicarlo a un Arduino, pueden ser de gran utilidad al sumarlo a un sistema ya que aumenta la cantidad de salidas digitales con tan solo 3 pines del microcontrolador o Arduino.

En este caso, un 74HC595 bien empleado puede ser el protagonista de todo el armado, puesto que quita un peso enorme a todo el sistema sin comprometer la efectividad y eficiencia del mismo, hecho que también se pudo evidenciar con investigación y hallazgo de soluciones propuestas.[3]

Capítulo 4

Análisis del problema

Una vez que se tiene claro como funciona el circuito 74HC595, los pasos que realizaremos son:

4.1. Transmisión

Generaremos un arreglo de datos enteros el cual va a estar codificado, para eso lo realizaremos primero en QT, para verificar su correcto funcionamiento.

4.2. Descriptación

Para realizar este proceso, usaremos una clave (bandera) la cual nos indicará la posición de los verdaderos datos y aplicaremos la condicion que nos indican.

4.3. Recepción

Ya validada la bandera, con ayuda del reloj, clasificaremos estos datos para obtener el mensaje real y los guardaremos.

4.4. Visualización

Con los datos guardados, mediante un LCD imprimiremos el mensaje real.

Capítulo 5

Marco experimental

Pruebas de funcionamiento

La condición para el sistema de comunicación: Los datos válidos son el quinceavo y vigésimo segundo después de que aparezca la clave dos veces 5.1

Cadena	Resultado
22,25,22,123,54,212,124,65,43,23,54,67,60,70,34,23,67,85,23,125,75,12,13,14,15	85,15
22,22,10,5,123,234,54,80,212,124,65,43,23,54,67,60,70,34,23,67,85,23,125,75,12	70,75
10,123,22,22,80,212,124,65,43,23,54,67,60,70,34,23,67,85,23,125,75,12,11,20,21,23	23,23

Tabla 5.1: Pruebas

En éstas tres pruebas se pudo verificar que tanto la transmisión como recepcion del mensaje funcionan correctamente, pues ubicamos la clave (22) en diferentes posiciones.

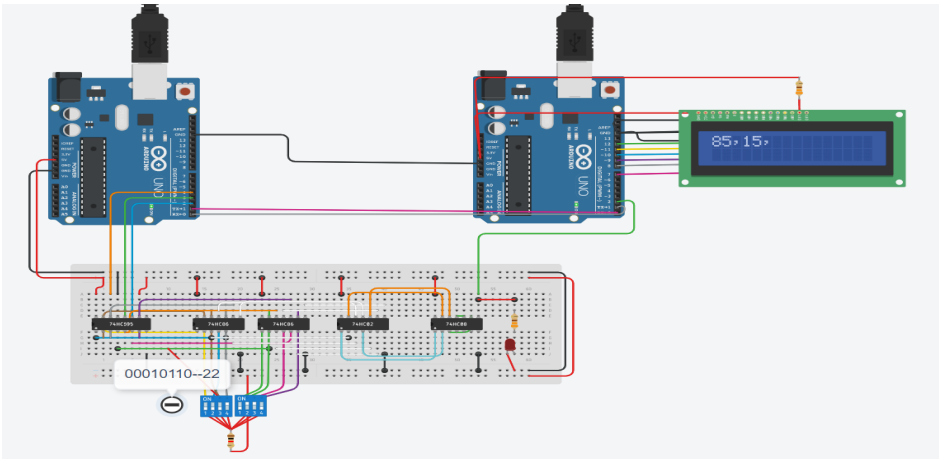


Figura 5.1: Diseño Final

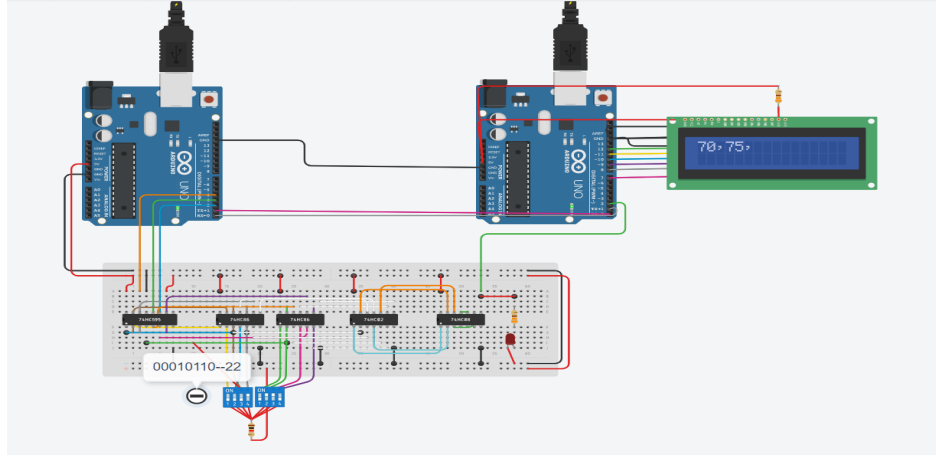


Figura 5.2: Diseño Final

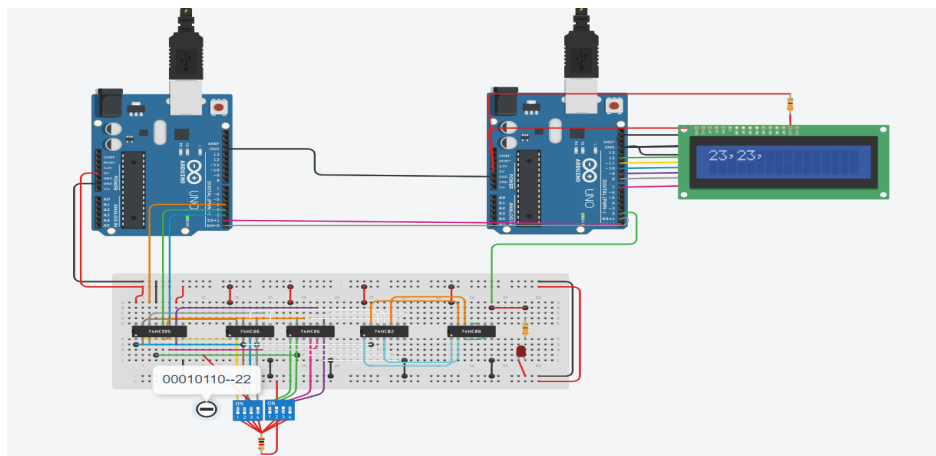


Figura 5.3: Diseño Final

Capítulo 6

Implementación del chip 74HC595

Como primera medida, se propuso el ejercicio de representar números binarios de forma manual usando una protoboard, un suministro de energía, un switch deslizante, dos pulsadores, luces led, el circuito integrado 74HC595, resistencias y cables. La primera etapa del ejercicio se basa en la estructuración general de los cables para efectuar la conexión entre el circuito integrado, los pulsadores, el switch y demás componentes. El funcionamiento constaba de un switch deslizante (data) que dependiendo del estado en el que se encontrara (encendido o apagado) comunicaba cierta información (1 o 0), a continuación el primero de los pulsadores recibe esta información las veces que fuera accionado. Por último, el segundo pulsador toma el registro y lo expone a la salida en este caso reflejándolo en las led.

En el segundo ejemplo aplicado se utilizaron luces leds, resistencias, un arduino, una

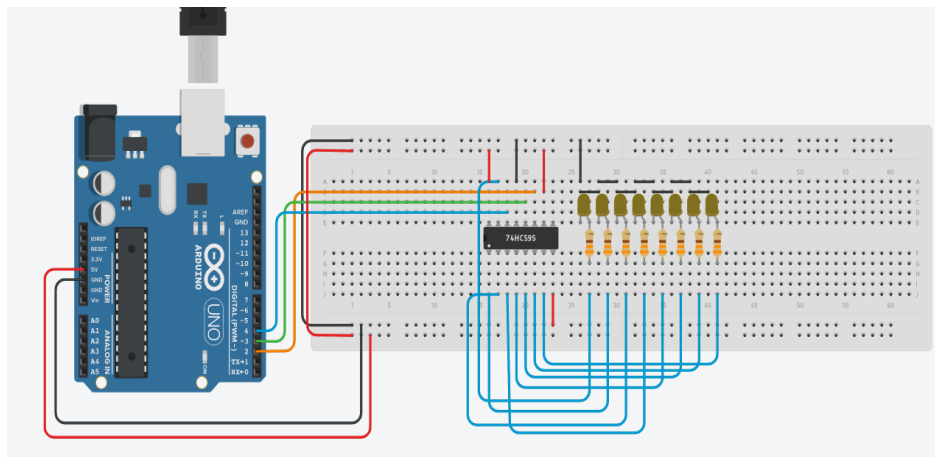


Figura 6.1: Circuito Ejemplo 1.

protoboard, cables y obviamente el circuito integrado 74HC595. El objetivo a cumplir era convertir un caracter ingresado por el usuario a codigo binario, principalmente mediante hardware. Una vez más se llevaron a cabo todos los cableados necesarios para que la conexión entre las componentes fuera óptima.

A continuación se usaron las resistencias y las LED para representar la conversión del

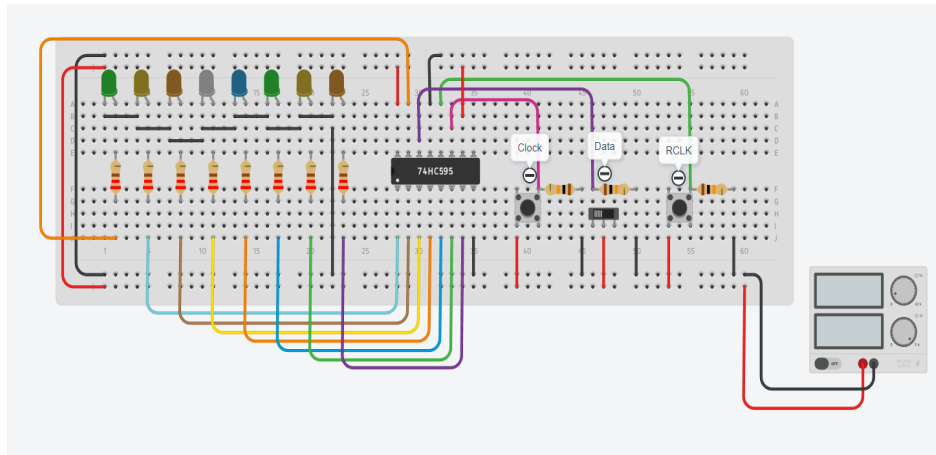


Figura 6.2: Circuito ejemplo 2.

caracter a código binario realizada por el propio circuito. En lo que respecta al código, se tomaron medidas específicas como restringir el rango de los caracteres ya que si éste se encontraba fuera de 0 y 255 no correspondería a código ASCII lo cual arrojaría un error.

```
1
2 #define PIN_CLOCK 2
3 #define PIN_RCLK 3
4 #define PIN_DATA 4
5 void setup() {
6   pinMode(PIN_CLOCK, OUTPUT);
7   pinMode(PIN_RCLK, OUTPUT);
8   pinMode(PIN_DATA, OUTPUT);
9
10  Serial.begin(9600);
11
12 }
13 void loop() {
14
15   if (Serial.available()) {
16     char character = Serial.read(); //Si es que si, lo leemos. En este caso la variable será de tipo char
17     if (character >= 0 && character < 256) { //No podemos representar caracteres cuyo código no esté entre 0 y 255
18       digitalWrite(PIN_RCLK, LOW); //Le decimos que vamos a escribir algo...
19       shiftOut(PIN_DATA, PIN_CLOCK, MSBFIRST, character); //Escribimos el carácter que el usuario proporcionó
20       digitalWrite(PIN_RCLK, HIGH); //Y le indicamos que lo exponga a la salida dando un pulso del reloj
21     }
22   }
23 }
```

Figura 6.3: Código del segundo ejemplo.

Capítulo 7

Desarrollo del diseño final

Fue una ardua elaboración de prototipos y bosquejos con inconvenientes en el apartado de la representación del mensaje real, fallos en el armado impedían que se diera la correcta validación del mensaje debido a la falta de conocimiento sobre cómo comunicar el ecosistema. Asimismo nos topamos con dificultades en el lado del código, por fallas de lógica el sistema se veía estancado y el mensaje que se veía reflejado en la pantalla no era el esperado.

Con el pasar de los intentos fue necesario consultar información que acelerara el proceso y en esa búsqueda exhaustiva dimos con herramientas imprescindibles como los circuitos 74HC86 (con lógica XOR necesario para comparar bits), circuito 74HC02 (lógica NOR) y el circuito 74HC08 (entrada AND).

Teniendo ya los cimientos del proyecto solo hacía falta pulir el apartado que iba a dar paso al mensaje: el código. La lógica era muy sencilla, se creó una variable llamada `cont1` que contara y validara si la condición de la bandera se había cumplido (que hayan dos 22), `cont2` a su vez contaba en qué posición iba con respecto a la cadena, siendo 15 y 22 nuestras posiciones de interés.

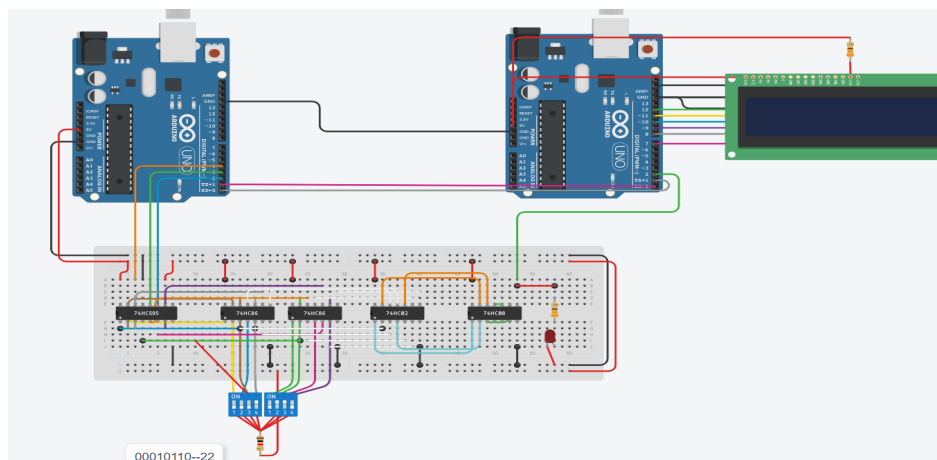


Figura 7.1: Diseño Final

Capítulo 8

Conclusiones

Definitivamente el trabajo que desempeña un chip en la elaboración de un sistema Arduino es muy grande, optimiza procesos y vuelve más compacto el armado. Sabiendo que el 74HC595 paraleliza los datos que recibe, éste facilitó enormemente descryptar la información mediante el método que se requería. Toda la construcción del proyecto tuvo que ser hecha alrededor de esta pieza debido a la obvia necesidad de un ecosistema en armonía que le sacara el mayor jugo a la funcionalidad del componente en cuestión.

Capítulo 9

Bibliografía

- [1] L. Valles. 74hc595 registro de desplazamiento con arduino. <https://programarfacil.com/blog/74hc595-registro-de-desplazamiento-arduino/>, 2020.
- [2] K. Tapias. Circuito integrado. <https://www.espaciahonduras.net/circuito-integrado-que-son-y-par>, 2022.
- [3] Bitwise Ar. Arduino desde cero en español - capítulo 70 - 74hc595 registro de desplazamiento (shift register). https://www.youtube.com/watch?v=LFqIA3ZvZE8t=1242sab_channel = BitwiseAr, 2021.