

UNIVERSIDAD DE ANTIOQUIA

Facultad de Ingeniería

Informe del Parcial 1

Jeronimo Herrera Sanchez

Juan Camilo Sierra Gomez

Juan Carlos Murillo Florez

Informatica 2

Grupo Triple J

**Profesores: Augusto Enrique Salazar Jimenez , Jonathan
Ferney Gomez Hurtado**

2 de marzo de 2022

Índice

1. Introducción	2
2. Investigación del 74HC595	2
3. Ejemplo del 74HC595	3
4. Ejemplos de envío de datos de un Arduino a otro Arduino	3
5. Resultados	4
6. Conclusión	4
7. Referencias	4

Resumen

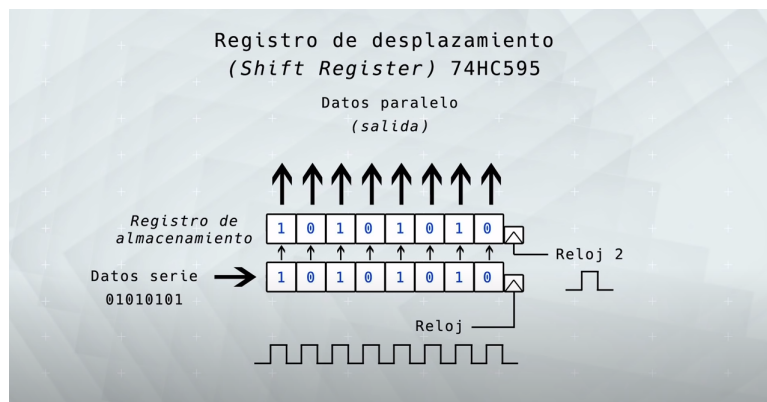
El propósito de este informe es conocer a profundidad la comunicación de dos Arduinos a su vez su implementación por medio de los puertos digitales los cuales serán: señal de reloj y datos. dando así la transferencia de información de datos a un sistema de recepción. con la ayuda del 74HC595 se podría llegar a este fin y lograr resolver este problema.

1. Introducción

En el presente informe de Informática 2, referente al parcial 1 sobre la comunicación de dos Arduinos donde se puede enviar la información de manera codificada y por medio del 74HC595 se trate la información. además de dar a conocer las capacidades de los integrantes a la hora de desarrollar la solución de un problema con las herramientas y conocimientos dados en el curso de informática 2

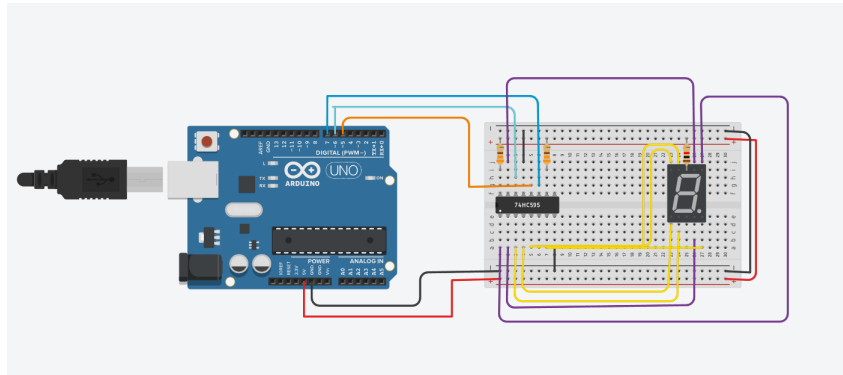
2. Investigación del 74HC595

El 74HC595 permite controlar 8 salidas con tan solo 3 pines digitales, con este se puede encadenar a un segundo circuito y controlar 16 salidas con estos mismos 3 pines digitales. Este se tiene una entrada en serie y una salida paralelo de 8 bits. Con cada pulso del reloj se le dice al circuito integrado que lea la entrada y comience el proceso.[Bitwise Ar.(6 nov 2021)]⁷ Al entrar los datos se desplaza a la derecha dándole espacio para el próximo bit, Pero sucede que al cargar los datos las salidas van cambiando de serie. Por esto lo ideal es que la salida no cambia hasta que se halla finalizado el problema de carga(esto lo resuelve el circuito integrado con un circuito de almacenamiento. El cual posee también un reloj) [CIRCUITS, I. (2003). 74HC595; 74HCT595.]⁷



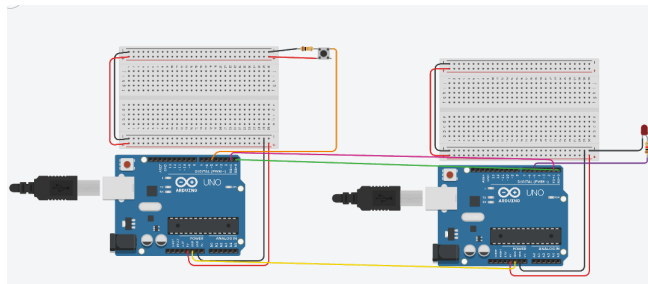
FUENTE: Bitwise Ar.(6 nov 2021)Arduino desde cero en Español

3. Ejemplo del 74HC595



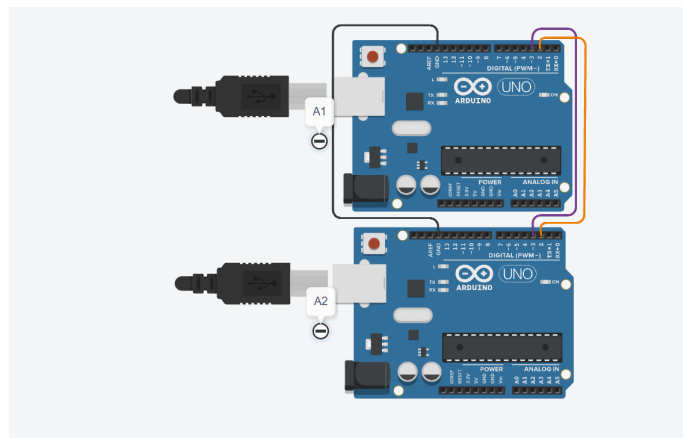
FUENTE: Autores

4. Ejemplos de envio de datos de un Arduino a otro Arduino



FUENTE: Autores

Aqui se muestra dos ejemplos del envio de datos de un arduino a otro.



FUENTE: Autores

5. Resultados

En el ejemplo anterior se muestra el uso del chip de expansión 74HC595 para mostrar la palabra HOLA en un visualizador de 7 segmentos. Para hacer esto primero escogimos los puertos (en este caso 5,6,7), luego creamos una variable constante tipo byte, para este caso le asignamos los bits y el orden manualmente (la idea es usar una función que lo haga) luego hacemos un ciclo que se repetirá tantas veces como la cantidad de bytes que queramos mostrar en el visor(para la palabra HOLA se repite 4 veces). Como ya sabemos o podemos ver en la documentación del chip, si se está usando un dispositivo con un reloj, debemos poner el pin del reloj en bajo con la función `digitalWrite(clockPin, LOW)`. Luego invocamos la función `shiftOut` la cual desplaza un byte uno a la vez, comenzando desde el más significativo (el primero por la izquierda) o el menos significativo (el primero por la derecha). Esta función como parámetros de entrada pide; el pin de datos (este indica que barrita se va a encender), el `clockPin`: es un pin que estará apagado y se encenderá cuando ya se hayan pasado todos los bits, esto para que muestre completamente y al mismo tiempo el byte, lo que hace que se muestre la letra completa y no se vaya armando barrita por barrita. el `bitOrder`: aquí se debe escoger el orden en el que se va a desplazar los bits, si desde el más significativo (`MSBFIRST`) o el menos significativo (`LSBFIRST`). El Valor: esto es lo que queremos que se desplace (en este caso lo que queremos mostrar en el visor que sería nuestra variable constante tipo byte). Por último cabe resaltar que cuando el pin que se le pasa a la entrada del visor, es decir, el pin que representa el estado de una de las barritas está en cero (`LOW`), la barrita se enciende y si esta en uno(`HIGH`) se apaga. Es por esto que luego de invocar la función con sus respectivos parámetros de entrada ponemos el pin del reloj en alto con la función `digitalWrite(clockPin, HIGH)` para que se apague el visor y esté listo para la siguiente iteración y mostrar la siguiente letra.

6. Conclusión

En conclusión, 74HC595 es una herramienta muy útil. con tan solo 3 salidas digitales podemos tener 8 salidas y si se juntara más de estos en serie se podría obtener el número de salidas que sean necesarias. también con esta podremos tratar la información con su registro de almacenamiento el cual entra al terminar la entrada de datos en serie y gracias a la ayuda de su reloj integrado podremos enviar la información sin ninguna pérdida de bits. con los resultados obtenidos con el ejemplo del 74HC595 se pudo hacer una mejor idea de su implementación e importancia a la hora de realizar la comunicación de los dos Arduino para el envío de información. ya con los ejemplos de la comunicación de los dos arduinos se pudo implementar la transmisión de datos para el circuito final.

7. Referencias

1. Logosímbolo UdeA
2. CIRCUITS, I. (2003). 74HC595; 74HCT595.
3. Bitwise Ar.(6 nov 2021)Arduino desde cero en Español - Capítulo 70 - 74HC595 Registro de desplazamiento (Shift register)

Youtube.https://www.youtube.com/watch?v=LFqIA3ZvZE8&t=426s&ab_channel=BitwiseAr

4. Imagen de funcionamiento del 74HC595 (Autores)