Inicialmente se crea la simulación en tinkercad

Se instalan 2 arduinos y Paralelamente se crea en qt el código fuente para esta prueba separando en diferentes funciones el código a implementar en los dos arduinos.

Luego se conecta el puerto 5 del Arduino emisor a el puerto 4 del Arduino recepetor y también el puerto r del Arduino emisor al puerto 5 del Arduino receptor, esto solo por ordenamiento del cableado.

En el Arduino 1 se definen los puertos 4 y 5 como salidas digitales, siendo 4 la señal de datos Y siendo 5 la señal de reloj.

En el Arduino 2 se definen los puertos 4 y 5 como entradas digitales, siendo 4 la señal de reloj Y siendo 5 la señal de datos.

Inicialmente se escribe una señal de alto en el pin 5 del Arduino 1 para que esta señal llegue al Arduino 2 y monitorearla asi:

Cuando la entrada 4 se encuentra en HIGH enciende el LED\_BUILTIN y en caso contrario se apaga; asi de esta manera verificamos el correcto envio de la señal de 5v desde Arduino 1 hasta Arduino 2.

El primer ensayo fue enviar 8 señales con retardo de 500 ms entre ellas las cuales representan el correspondiente binario a un numero entero entre 0 y 255, utilizamos la señal de datos para dterminar cuando el puerto serial del Arduino 1 recibia datos para asi activar la lectura de estos en el Arduino 2, durante este proceso se presentaron muchas interferencias al momento de activar 2 puertos simultáneamente, se encontró que este fallo se debia a que no estaban correctamente equipotencializados los dos arduinos, se corrijio conectando entre ellos sus pines GND.

Luego observamos que al enviar la señal de inicio de lectura de datos el Arduino receptor comienza recibir datos, valga la redundancia, antes de lo que nosotros deseamos que realmente reciba; luego de realizar diferentes pruebas de funcionamiento, observamos que este se debia a un retraso de envio de datos del Arduino transmisor debido a que en nuestro código la salida de activación de lectura, encontraba un retraso frente al envio real de datos ocasionado por un ciclo for que se encarga de descomponer el numero ingresado, al realizar la modificación y colocar la señal de activación justo antes del ciclo de envio de datos se logra corregir el problema.

Finalmente, con el circuito y los códigos implementados se logra:

-Implementar un Arduino emisor que recibe por serial un numero entero entre o y 255, luego lo descompone mediante un cilo for en un arreglo de 8 bits, luego activa una salida digital correspondiente a la señal de envio de datos llamada reloj, y luego en otro ciclo for activa o desactiva la señal de datos correspondiente a el estado de cada uno de los elementos del arreglo de bits.

-Implementar un Arduino receptor que mientras una entrada digital correspondiente a la señal de recepción de datos reloj se encuentre activa estará leyendo e imprimiendo en el puerto serial simultáneamente el estado de la entrada digital correspondiente a datos, ya sabemos con antelación que estos representan el arreglo de datos binario correspondiente al numero ingresado al Arduino 1.