

1. “Protocolo de Comunicación – Contador de Pulsos”

1. Objetivo General

Conocer, identificar y comprobar el funcionamiento al desarrollar un protocolo de comunicación, en esta primera parte de un contador de pulsos en específico

2. Objetivos específicos

- Utilizando 1 Arduino, diseñar una función que permita enviar un número determinado de pulsos (Variación de 0 a 1) a una frecuencia específica, la función debe tener la siguiente estructura **void pulsos(int cantidad, int duración).** (Comprobar el correcto funcionamiento utilizando el osciloscopio).
- Utilizando un segundo Arduino, diseñar una función que permita contar el número de pulsos que ha recibido (Variación de 0 a 1) y mostrarlos en el monitor serial.
- Utilizando ambos arduinos, conectar un pin de los Arduinos para comprobar el funcionamiento del sistema enviando un numero de pulsos específicos a la velocidad de 1 pulso por segundo, 10, 100 y 1000 pulsos por segundo.

NOTA: Recuerda que para que pueda existir comunicación ambas tierras de los Arduino deben estar conectadas.

3. Material

- 2 Arduino
- Protoboard
- Botones
- LED's



2. “Protocolo de Comunicación – Contador de tiempo”

1. Objetivo General

Conocer, identificar y comprobar el funcionamiento al desarrollar un protocolo de comunicación, en esta segunda parte de un contador de tiempo de los pulsos.

2. Objetivos específicos

- Utilizando 1 Arduino, diseñar una función que permita enviar un pulso con un tiempo en High determinado (Comprobar el correcto funcionamiento utilizando el osciloscopio).
- Utilizando un segundo Arduino, diseñar una función que permita contar el tiempo de la duración de un pulso que ha recibido del primer Arduino (Variación de 0 a 1) y mostrarlos en el monitor serial.
- Utilizando ambos arduinos, conectar un pin de los Arduinos para enviar una serie de 4 bits desde el primer Arduino al segundo y mostrar los bits enviados por monitor serial. Tomar en consideración que la serie de 4 bits debe comenzar con un bit en High, cada uno de los bits debe de ser de un tamaño específico, de tal manera que el segundo Arduino sabe que tiene que esperar 4 pulsos de un tamaño determinado y que esta serie de pulsos comienza con el cambio de 0 a 1.

NOTA: Recuerda que para que pueda existir comunicación ambas tierras de los Arduino deben estar conectadas.

3. Material

- 2 Arduino
- Protoboard
- Botones
- LED's



3. “Protocolo de Comunicación – Convertidor a binario”

1. Objetivo General

Conocer, identificar y comprobar el funcionamiento al desarrollar un protocolo de comunicación, en esta tercera parte de un convertidor a binario.

2. Objetivos específicos

- Utilizando 1 Arduino, diseñar un sistema que permita convertir el valor analógico de un potenciómetro en un arreglo de variables que representen el numero en binario de 8 bits, de tal forma que podamos ver cada uno de los valores como salidas digitales en los pines del Arduino.
- Utilizando un segundo Arduino, diseñar una función que permita convertir el valor en binario obtenido de 8 puertos digitales del Arduino en un valor entero en sistema decimal y mostrarlos en el monitor serial. Utilizar el primer Arduino para generar los valores digitales en binario y el segundo Arduino utilizarlo para decodificarlos y convertirlos en un valor decimal.

NOTA: Recuerda que para que pueda existir comunicación ambas tierras de los Arduino deben estar conectadas.

3. Material

- 2 Arduino
- Protoboard
- Botones
- LED's
- Potenciometro

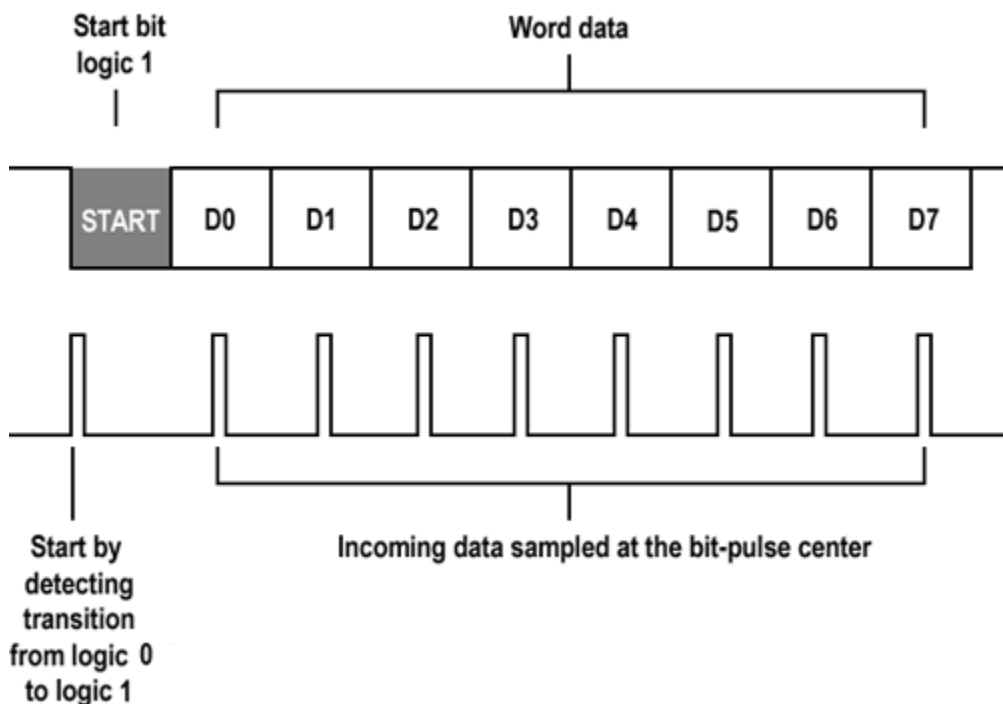
4. “Protocolo de Comunicación Asíncrono”

1. Objetivo General

Conocer, identificar y comprobar el funcionamiento al desarrollar un protocolo de comunicación asíncrono.

2. Objetivos específicos

- Utilizando 1 Arduino, diseñar un sistema que permita convertir el valor analógico de un potenciómetro en un arreglo de variables que representen el número en binario de 8 bits, y los deposite en un arreglo [], Finalmente elaborar un código de programación que permita enviar los bits del arreglo en orden y en serie siguiendo las siguientes características:
 - Iniciar el envío de los datos siempre con una señal HIGH (En total se enviarán 9 datos, el primer bit en HIGH y posteriormente los 8 bits de información).
 - Enviar los datos con una duración del pulso determinada, ejemplo 1000 ms
 - Enviar siempre la misma cantidad de bits.



- Utilizando un segundo Arduino, diseñar una función que permita adquirir el valor en bits de la señal de comunicación del Arduino anterior, la procese y la convierte en un dato decimal que finalmente enviara al monitor serial.

NOTA: Recuerda que para que pueda existir comunicación ambas tierras de los Arduino deben estar conectadas.

3. Material

- 2 Arduino
- Protoboard
- Botones
- LED's
- Potenciometro