

Práctica 3

Protocolos de Comunicación

Stephanie Lizeth Malvaes Diaz
Ingeniería en Computación
Instituto Tecnológico Autónomo de México
Email:

Daniela Arely Morales Hernández
Ingeniería en Telecomunicaciones
Instituto Tecnológico Autónomo de México
Email: daniela.morales@itam.mx

Resumen—Se usarán diferentes protocolos de comunicación entre dispositivos digitales de los cuales deberemos identificar sus diferencias y ventajas de cada uno. Al comunicar dos dispositivos Xbee se probarán sus ventajas.

I. INTRODUCCIÓN

Los protocolos de comunicación serial permiten la conexión de dos o más dispositivos a través de una red en la cual cada nodo contiene un identificador y rol definido, se le llama maestro al dispositivo que envía comandos a través de la red de manera que uno o más elementos llamados esclavos los reciban y actúen en consecuencia. Son una característica común en la mayoría de los microcontroladores.

Durante la práctica se usarán protocolos tales como I2C que es usado para comunicación síncrona lo cual indica que la tasa de transmisión y la cantidad de datos enviadas están reguladas por una señal de reloj común para todos los dispositivos, los cuales están físicamente conectados entre sí, por lo que las señales en un momento dado son recibidas por cada dispositivo, sin embargo, parte de la información que se contiene son los identificadores del o los destinatarios del comando, siendo esto lo que da la pauta para que algunos dispositivos reaccionen o no respecto a cada mensaje.

Una alternativa a la conexión serial es la comunicación serial inalámbrica la cual está basada en el protocolo ZigBee. Dicho protocolo retransmite de manera inalámbrica a través de una portadora de radiofrecuencia aquello que es recibido en un puerto serial convencional, de manera que uno o más módulos de radio adicionales reciban el mensaje y actúen en función del comando y su respectivo rol en la red.

II. MARCO TEÓRICO

II-A. I2C

Define la trama de datos y las conexiones físicas para transferir bits entre dos dispositivos digitales. Su arquitectura permite tener una confirmación de los datos recibidos dentro de la misma trama. Hay diferentes tipos de elementos durante la comunicación.

- MAESTRO: se encarga de controlar el cable de reloj así como de iniciar y parar la comunicación y tiene dos modos, M-transmisor o M-receptor. Sus principales características son:
 - Iniciar la comunicación – S
 - Enviar 7 bits de dirección – ADDR

- Generar 1 bit de Lectura ó Escritura – R/W
- Enviar 8 bits de dirección de memoria
- Transmitir 8 bits de datos –
- Confirmar la recepción de datos – ACK – ACKnowledged
- Generar confirmación de No-recepción, NACK – No-ACKnowledged
- Finalizar la comunicación

- ESCLAVO: suele ser un sensor. Este es el elemento que suministra la información de interés al MAESTRO. No puede generar su señal de reloj. Sus funciones principales son:

- Enviar información en paquetes de 8 bits.
- Enviar confirmaciones de recepción, llamadas ACK

II-B. SPI

Es un protocolo de comunicación serial con alta velocidad de transmisión. Es un protocolo síncrono de 4 hilos que transmite y recibe información por canales diferentes en el mismo cable. Usa una línea adicional encargada de mantener la sincronización. Hay dos formas de conectar al maestro con los esclavos, encadenado o paralelo.

El bus SPI se define mediante 4 pines

- SCLK o SCK : Señal de reloj del bus. Esta señal rige la velocidad a la que se transmite cada bit.
- MISO(Master Input Slave Output): Es la señal de entrada a nuestro dispositivo, por aquí se reciben los datos desde el otro integrado.
- MOSI(Master Output Slave Input): Transmisión de datos hacia el otro integrado.
- SS o CS: Chip Select o Slave Select, habilita el integrado hacia el que se envían los datos. Esta señal es opcional y en algunos casos no se usa.

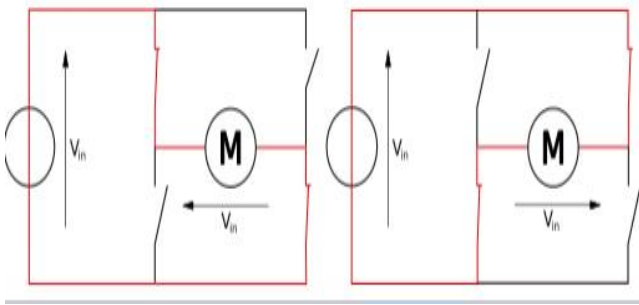
Desventajas de SPI:

- No hay control del flujo por hardware.
- Las comunicaciones tiene que estar perfectamente establecidas de antemano. No puedes enviar mensajes de diferentes longitudes cuando convenga.
- No hay confirmación de la recepción como ocurre en I2C con el ACK. Es decir no sabemos si el mensaje a llegado al destino.

- Usa más pines que otros buses, ya que necesita uno por cada esclavo. Eso implica que no hay direccionamiento en la propia trama. A menos que se diseñe por software.
- Funcionamiento a distancias cortas
- Master único y casi sin posibilidad de master múltiple

II-C. Puente H

Es un circuito electrónico que permite a un motor eléctrico DC girar en ambos sentidos. Esta constituido por 4 interruptores. Si S1 y S4 están cerrados genera el movimiento en un sentido.



II-D. Acelerómetro

Es un dispositivo electromecánico que mide la aceleración, que es la tasa de cambio de velocidad de un objeto. Se mide en metros sobre segundo al cuadrado o en la fuerzas G. Detecta fuerzas de aceleración sea estática (gravedad) o dinámica (arranques repentinos / frenado). Pueden medir la aceleración en uno, dos o tres ejes, contienen placas capacitivas internamente que se mueven cambiando la capacitancia, a partir de los cambios de capacitancia se mide la aceleración. El acelerómetro es útil para detectar vibraciones en sistemas, aplicaciones de orientación y detectar movimiento. Hay dos tipos de acelerómetros, el analógico y el digital. Durante la práctica usaremos el analógico.

II-E. Xbee

Brindan un medio inalámbrico para la interconexión y comunicación entre dispositivos. Crean redes FAST POINT-TO-MULTIPOINT o redes PEER-TO-PEER. Proveen una alta tasa de transmisión con baja latencia y sincronización de comunicación predecible.

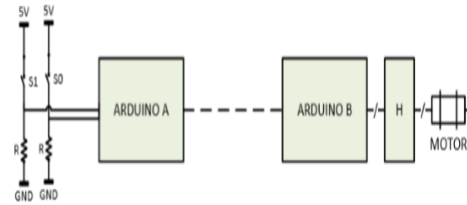
II-F. Zigbee

Comunicación inalámbrica utilizada para la comunicación entre Xbee. Se utilizará para comunicar un Arduino

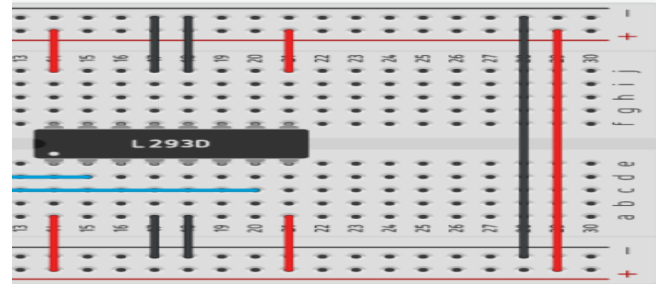
III. DESARROLLO

III-A. Protocolo I2C

Se tuvo que trabajar en conjunto con otro equipo para poder implementar el circuito del maestro y el esclavo. Nos basamos en el siguiente diagrama para conectar el Arduino A(maestro) con el Arduino B (esclavo)

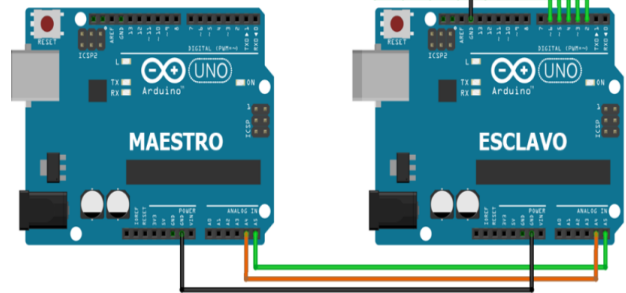


El puente H se implementa con un circuito integrado para el funcionamiento del motor. Se cablea de la siguiente manera:



Para que se pudiera habilitar la comunicación entre los arduinos usamos el protocolo I2C el cual ya está configurado en el arduino con los puertos establecidos.

En una protoboard se conecta el boton al arduino para configurar el maestro, en la otra protoboard se conecta el punte H con el motor.



En la imagen anterior se muestra la conexión I2C del arduino UNO solo que tuvimos que conectarlo a los pines correspondientes del arduino MEGA.

Adaptador LCD a I2C	Arduino Uno, Nano, Mini.	Arduino Mega , DUE
GND	GND	GND
VCC	5V	5V
SDA	A4	20
SCL	A5	21

En la práctica usaremos un acelerómetro que recibirá la posición angular de donde se encuentre y la transmitirá al arduino.

La conexión es simple, tan solo se alimenta el módulo desde

Arduino mediante GND y 5V y se conecta el pin SDA y SCL con los pines correspondientes del sensor.

Se logran obtener los datos recibidos de la posición, los códigos están adjuntos en el repositorio.

Se logró hacer que la velocidad se cambiara dependiendo de la posición angular del acelerómetro mediante la conexión I2C. Esto último no tuvimos oportunidad de mostrárselo en clase debido a que tardó en resultar esta parte.

III-B. XBEE

Para la parte de los XBEE primero tuvimos que entender este nuevo programa para usar. Se usó la plataforma XCTU en donde se configuran el receptor y el transmisor.

CH Channel	C
ID PAN ID	3332
DH Destination Address High	13A200
DL Destination Address Low	40B4C241
MY 16-bit Source Address	AAAA
SH Serial Number High	13A200
SL Serial Number Low	40F9D5D1
MM MAC Mode	802.15.4 + MaxStream header w/AC
RR XBee Retries	0
RN Random Delay Slots	0
NT Node Discover Time	19 x 100 ms
NO Node Discover Options	0
CE Coordinator Enable	Coordinator [1]

De esta parte tan solo logramos configurarlos como receptor y transmisor.

IV. CONCLUSIONES

En esta práctica se logró avanzar de manera lenta debido a que los protocolos fueron complicados de entender. Cada protocolo tiene sus ventajas las cuales pueden ser aprovechadas dependiendo el objetivo de la comunicación. Estas son algunas de las ventajas que encontramos de SPI sobre I2C:

- I2C No es Full-Duplex por lo que no permite envíos y recepciones al mismo tiempo.
- I2C un poco más complejo que SPI.
- I2C no tiene control de errores, por ejemplo mediante paridad etc. Aunque se puede realizar por Software.
- Velocidades de comunicación relativamente elevadas. En el caso de Arduino de hasta 8 Mhz.
- Completo control sobre la trama de bits al no exigir direccionamiento ni ACK.
- Se requiere un hardware sencillo (Barato)
- Requiere un menor consumo y menor electrónica de conexión que el I2C.
- Como el Clock lo proporciona el master, los esclavos no necesitan osciladores por lo que resulta más barato.

REFERENCIAS

- [1] <https://pygmalion.tech/tutoriales/electronica/tutorial-electronica-puente-h/>
- [2] <https://hetpro-store.com/TUTORIALES/i2c/>
- [3] [https://es.wikipedia.org/wiki/Puente_H_\(electr%C3%B3nica\)](https://es.wikipedia.org/wiki/Puente_H_(electr%C3%B3nica))