

Práctica 3. Protocolos de Comunicación

Emiliano Sotomayor González
155902
uboot46@gmail.com

Jaime Limón Samperio
162395
jalisa2808@gmail.com

Abstract—Se examinan las ventajas y desventajas de los protocolos I2C, UART y SPI utilizando módulos de comunicación XBee para conocer la configuración básica de sensores a través de una red inalámbrica.

I. INTRODUCCIÓN

Los protocolos de comunicación serial son una característica común en la mayoría de los microcontroladores. Su manejo permite comunicar dos o más dispositivos a través de una red en la cual cada nodo contiene un identificador y un rol específico. En general, 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.

Protocolos tales como I2C se valen de una comunicación sincrónica, es decir, la tasa de velocidad de envío y cantidad de datos que fluyen a través de la red, están reguladas por una señal de reloj común para todos los dispositivos. En este protocolo de comunicación todos los dispositivos 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 contiene el identificador del destinatario o los destinatarios del comando, siendo esto lo que da la pauta para que algunos dispositivos reaccionen o no respecto a cada mensaje.

II. MARCO TEÓRICO

Para invertir el sentido de giro de un motor hay que cambiar la polaridad, es decir, cambiar el sentido con el que la corriente pasa a través del motor. En electrónica esto normalmente se logra con una configuración llamada puente H. Es un circuito de dos estados: avance y retroceso. Está elaborado con 4 interruptores. Se realizará la conexión de un Arduino esclavo y uno maestro para la demostración.

III. DESARROLLO

Se deberán seguir las siguientes instrucciones:

- Realizar las conexiones correspondientes para habilitar dos push-buttons en el Arduino A y un puente H en el Arduino B, conectar un motorreductor de corriente directa a la salida del puente H.
- Habilitar las librerías necesarias para iniciar una comunicación serial entre ambos Arduinos a través del protocolo I2C.
- Habilitar del Arduino A como maestro (transmisor) y al Arduino B como esclavo (receptor).

- Realizar la programación necesaria para que el motor conectado al Arduino B se comporte en función de las pulsaciones en el Arduino A tal como se muestra en la siguiente tabla. El giro ha de realizarse a una velocidad constante definida por el usuario.
- Modificar el código de manera tal que pulsar cada uno de los dos push-buttons incremente la velocidad en un sentido distinto, partiendo de una velocidad cero.
- Realizar el ejercicio anterior sustituyendo la comunicación I2C por el protocolo SPI, reporte una comparativa entre ambos protocolos.
- Crear un programa en el Arduino A que tome lectura de un acelerómetro analógico y que mapee los datos a un valor de posición angular para desplegar apropiadamente los mismos a través del monitor serial.
- Hacer que la velocidad en el motor del Arduino B sea proporcional a la posición angular del acelerómetro en un recorrido de -90 a 90 grados, mediante una conexión I2C o SPI.
- Habilitar y configurar la comunicación inalámbrica entre dos módulos XBee mediante el software XCTU. (Establecer un maestro y un esclavo y seleccionar el mismo canal de comunicación.)

IV. RESULTADOS

Pudimos observar que al utilizar librerías que nos permiten utilizar los protocolos I2C y SPI, el manejo de las comunicaciones entre microcontroladores Arduino es relativamente sencillo. La mayoría de la configuración para comunicarse entre dos distintos Arduinos se puede lograr con unas pocas líneas de código, así como se ejemplifica en nuestro repositorio de git, y el resto del programa quedará libre para manejar las distintas situaciones donde el microcontrolador puede ser utilizado. (Lectura de señales analógicas, lectura de señales digitales, etc...)

V. CONCLUSIONES

Como se observa en la sección de resultados, existen ventajas y desventajas al momento de utilizar lenguaje ensamblador. El poder de procesamiento actual permite que el tiempo de compilación y ejecución de un lenguaje de alto nivel sea muy rápido.

La manipulación de microcontroladores a bajo nivel es en ocasiones más eficiente aunque puede ser más complicado su manejo y requiere tener conocimiento del hardware en cuestión.

VI. REPOSITORIO

<https://github.com/uboa46/principios-de-mecatronica.git>

REFERENCES

- [1] "Puente H [https://es.wikipedia.org/wiki/Puente H \(electr](https://es.wikipedia.org/wiki/Puente_H_(electr)
- [2] "Arduino reference" Internet: "<https://www.arduino.cc/reference/en/>"
- [3] "Acelero metro <https://es.wikipedia.org/wiki/Aceler>