Práctica No.3. Protocolos de Comunicación.

1st Juan Carlos Garduño Gutiérrez 157302
Ingeniería en Telecomunicaciones
Instituto Tecnológico Autónomo de México
jgarduo6@itam.mx

2nd Humberto Martínez Barrón y Robles 166056
Ingeniería en Mecatrónica
Instituto Tecnológico Autónomo de México
hmartin3@itam.mx

3rd Sebastián Aranda Bassegoda 157465
Ingeniería en Computación e Ingeniería Industrial
Instituto Tecnológico Autónomo de México
sarandab3@itam.mx

***Resumen*—En esta tercera práctica del laboratorio de principios de mecatrónica, se introdujo a los alumnos a los protocolos de comunicación para la interacción de componentes, en especial: $I2C$, $UART$ y SPI. Cada protocolo de comunicación, funciona para conectar diferentes componentes, por ejemplo: el protocolo $I2C$ se utiliza para comunicar dos $Arduino$, el protocolo ~~$UART$~~ se utiliza para comunicar el acelerómetro con la $FPGA$. Se busca conocer las ventajas y desventajas de cada uno de ellos y cuando es conveniente usarlos. En teoría la experiencia del equipo en este ambiente de programación ha aumentado desde la práctica 2, sin embargo la dificultad de esta práctica es mayor a la pasada.**

***Index Terms*—Protocolos,comunicación,I^2C ,$UART$, SPI, $Xbee$.**

I. Introducción

Las comunicaciones entre dispositivos por medio de protocolos como $I2C$, SPI y $XBee$ son conceptos de suma importancia para ingenieros que trabajen con sistemas embebidos. Ya que éstos constituyen la base para el internet de las cosas o IoT, su aprendizaje es crucial para desarrollar tecnologías con dispositivos que necesiten interactuar entre sí para intercambiar información. Cabe mencionar que hoy, la mayoría de los dispositivos que utilizamos diariamente necesitan comunicarse con otros (nuestras computadoras se comunican entre ellas con $WiFi$, los drones se comunican con sus controles por radiofrecuencias y nuestros teléfonos se conectan a una infinidad de aparatos). Si bien los protocolos de comunicaciones han evolucionado con el tiempo y las tecnologías de IoT se han ido complicando conforme avanza su desarrollo, las bases de las comunicaciones siguen siendo iguales. Por eso, la relevancia de los protocolos estudiados en esta práctica no ha disminuido, sino que tienen aplicaciones innovadoras e interesantes. Por listar algunos ejemplos, mencionemos:

1. Sensores de temperatura
2. Sistemas de alarmas inalámbricas
3. Sensores de presión
4. Pantallas táctiles
5. Detección de incendios forestales
6. Lentes de cámaras
7. Memorias flash
8. Controles de consolas de videojuegos

Por si fuera poco, estos protocolos constituyen un primer acercamiento para otras tecnologías más usadas y más famosas como $WiFi$, $Bluetooth$ y tecnología de telefonía celular. Resulta entonces claro que estos protocolos son útiles no sólo para aplicaciones simples, sino que también puede tomarse el conocimiento de estos protocolos para facilitar la comprensión de otras disciplinas más avanzadas.

El objetivo de la práctica, para cada parte, fue:

1. Lograr que un microcontrolador $maestro$ mande órdenes a un $esclavo$ para controlar el giro de un motor por medio de un puente H.
2. Encontrar la forma de conectar un motor con un puente H a la $FPGA$ para leer los registros guardados en su acelerómetro y, con base en la información de los sensores, dar un giro específico al motor.
3. Mandar mensajes con $UART$ usando $XBee$s serie 1 para lograr comportamientos diferentes por medio del microcontrolador $Arduino$.

~~En cada sección de este documento se presenta~~: un breve marco teórico describiendo las herramientas utilizadas en la práctica (II), una descripción detallada del desarrollo para llegar a cada solución planteada en las especificaciones de la práctica (III), un análisis de los resultados obtenidos (IV), conclusiones y aprendizajes individuales (V), una explicación del papel que desempeñó cada miembro del equipo (VI) y, finalmente, referencias bibliográficas.

II. Marco teórico

~~Para la primera parte de la práctica se implementa el protocolo de comunicación $I2C$ para comunicar dos microcontroladores(A y B).~~Este protocolo es la comunicación síncrona $maestro$ - $esclavo$. $I2C$ significa Circuito Interintegrado (Por sus siglas en Inglés Inter-Integrated Circuit) es un protocolo de comunicación serial desarrollado por Phillips Semiconductors en la década de los 80s. Básicamente se creó para poder comunicar varios chips al mismo tiempo dentro de los televisores. El protocolo $I2C$ es un protocolo de comunicación serial. Toma e integra lo mejor de los protocolos SPI y UART. Con este protocolo podemos tener a varios maestros controlando uno o múltiples esclavos. Esto puede ser de gran ayuda cuando se van a utilizar varios microcontroladores para

almacenar un registro de datos hacia una sola memoria o cuando se va a mostrar información en una sola pantalla.

Para la segunda parte de la práctica se utilizó el acelerómetro de la tarjeta $DE0-Nano$ para establecer la dirección a la que rota un motor. El acelerómetro se comunica con la $FPGA$ mediante el protocolo SPI. Un acelerómetro es un dispositivo cuya finalidad es determinar la aceleración aplicada al mismo.Los cuáles pueden ser mecánicos o electrónicos. Y el protocolo SPI es un acrónimo para referirse al protocolo de comunicación serial:Serial $Peripherical\ Interface$. Este protocolo se ha convertido es uno de los más populares protocolos para trabajar con comunicación serial debido a su velocidad de transmisión, simplicidad, funcionamiento y también gracias a que muchos dispositivos en el mercado como pantallas LCD, sensores, microcontroladores pueden trabajar con el.

Por otro lado, para la tercera parte de la práctica se utiliza el protocolo $UART$ y $Xbee$. $UART$, son las siglas en inglés de Universal Asynchronous Receiver-Transmitter, en español: Transmisor-Receptor Asíncrono Universal, es el dispositivo que controla los puertos y dispositivos serie. Se encuentra integrado en la placa base o en la tarjeta adaptadora del dispositivo.

El $UART$ normalmente no genera directamente o recibe las señales externas entre los diferentes módulos del equipo. Usualmente se usan dispositivos de interfaz separados para convertir las señales de nivel lógico del $UART$ hacia y desde los niveles de señalización externos. En esta parte de la práctica, como interfaz se utiliza utilizan un $Xbees$. De acuerdo a $Digi$(fabricante), los módulos XBee son soluciones integradas que brindan un medio inalámbrico para la interconexión y comunicación entre dispositivos. Estos módulos utilizan el protocolo de red llamado IEEE 802.15.4 para crear redes FAST POINT-TO-MULTIPOINT (punto a multipunto); o para redes PEER-TO-PEER (punto a punto). Fueron diseñados para aplicaciones que requieren de un alto tráfico de datos, baja latencia y una sincronización de comunicación predecible. Por lo que básicamente XBee es propiedad de Digi basado en el protocolo Zigbee. En términos simples, los XBee son módulos inalámbricos fáciles de usar.

Las restricciones que hasta ahora tenemos corresponde con teoría acerca de protocolos de comunicación. Sin embargo, en el la sesión de laboratorio se dio una breve introducción y toda la información se encuentra en la web. Tanto el protocolo $I2C$ y el protocolo SPI, sólo el maestro le da ordenes al esclavo y este no le puede contestar. También una restricción importante en el protocolo de $I2C$ se necesitan 2 tarjetas, entonces puede ser un grave problema.

III. Desarrollo

III-A. Parte 1

Para cumplir con los objetivos de la práctica se empezó por combinar diferentes protocolos de comunicación para la interacción de componentes:

-I2C Comunicación serial síncrona maestro-esclavo. Se utilizará para comunicar dos Arduino.

-SPI Comunicación serial síncrona maestro-esclavo. Se utiliza para comunicar el acelerómetro con la FPGA.

-UART Comunicación asíncrona. Se utiliza para comunicar el Arduino con un Xbee utilizando un shield.

-Zigbee Comunicación inalámbrica utilizada para la comunicación entre Xbee. Se utiliza para comunicar un Arduino (con Xbee) con la computadora inalámbricamente.

En la parte 1 de la práctica se utilizó el protocolo I2C, anteriormente mencionado, para comunicar dos microcontroladores: A y B. El primero de estos, el A, tiene conectados 2 botones. El microcontrolador B tiene conectado un motor a través de un puente H. Los microcontroladores funcionaban de acuerdo a la tabla 1.

Botones	Acción del Motor
00	Freno pasivo
01	Rota en sentido de las manecillas
10	Rota en contrasentido de las manecillas
11	Freno activo

Tabla 1: Movimiento del motor

III-B. Parte 2

Protocolo SPI y acelerómetro

Para resolver este problema se utilizó el acelerómetro de la tarjeta DE0-Nano para definir la dirección del motor. Utilizando el protocolo es que se comunicó a la FPGA con el SPI.

Para leer los registros se utilizó los LEDs de la tarjeta.Estos fueron los pasos utilizados:

- Se establecieron valores adecuados para el componente *spi-3-wire-master* en *accel-rw.vhd*.

- Se leyó el registro que contiene el ID del acelerómetro y verificar que el valor es correcto.

- Se determinaron valores adecuados para: deshabilitar las funciones *TAP* y *DOUBLE TAP*, establecer *BANDWIDTH* de 200Hz, deshabilitar el modo *LINK* y *AUTOSLEEP*, habilitar el modo *MEASURE* y establecer el acelerómetro en modo *NORMAL*.

- Se escribieron los valores adecuados en los registros y se leyeron para verificar que su escritura fuera correcta.

- De los registros ser leyeron los valores para *X, Y y Z*, y se usó el *dip switch* para cambiar su valor. Utilizando los valores de las tres variables se prendían dos LEDs que indicaran la dirección del motor. Finalmente se sustituyeron los LEDs por el motor utilizando el puente H.

III-C. Parte 3

Protocolo UART y XBEE Se configuraron los *XBee* utilizando *XCTU* con parámetros adecuados y se probó la comunicación entre ambos desde *XCTU*.

Posteriormente se implementó el programa que enviara enviara comandos desde *XCTU* a *Arduino*, de manera que al recibir alguno de los tres comandos implementados, se prendiera o apagara un LED distinto. AL presionar un botón, *Arduino* enviaría un mensaje al otro *XBee*.

IV. Resultados

IV-A. *Primera parte de la práctica. Protcolo I2C*

En esta primera parte de la práctica no asociamos con otro equipo para realizar los dos microcontroladores. Nuestro equipo se encargó de realizar el microcontrolador B, que consistía en un motor mediante un puente H. Utilizando la hoja de especificaciones se conectó el motor utilizando una fuente de voltaje.

Encargándonos de la parte de esclavo, numerándolo dado que el maestro podía contar con diversos esclavos. Fue posible implementarlo de manera correcta y comunicarlo con el trabajo del equipo con el que se colaboró.

IV-B. *Segunda parte de la práctica. Protocolo SPI y acelerómetro*

En la segunda parte de la práctica al deshabilitar las funciones *TAP* y *DOUBLE TAP*, en la hoja de datos, en el valor del registro PQR, dirección 0X21, tuvo que ser igual a 0.

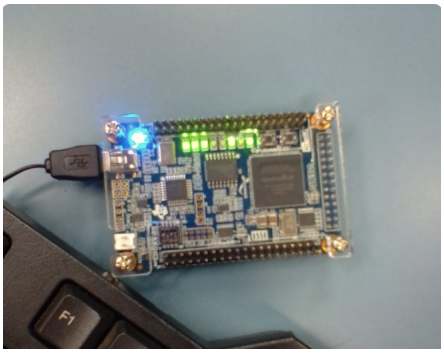

Figura 1. Se muestra el id del dispositivo

Al establecer el *BADNWIDTH* de 200Hz, tuvimos que establecer como 1100 los últimos 4 bits del registro PWRATE. Continuamos por deshabilitar el modo *LINK* y *AUTOSLEEP*,

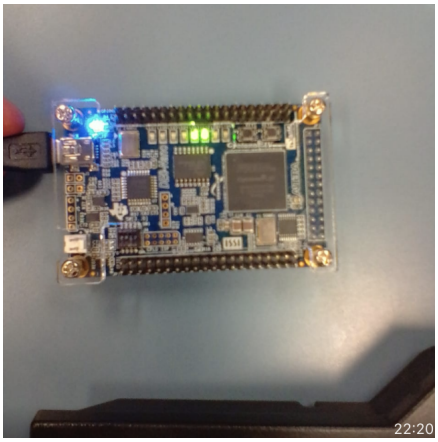

Figura 2. Se muestra el id del dispositivo

y habilitar el modo *MEASURE*, se hizo en el mismo registro, POWER-NCTL, desactivando los bits 5 y 4, y activando el 3. Posteriormente se probó con el circuito y el motor utilizando puente H, funcionó sin complicaciones.

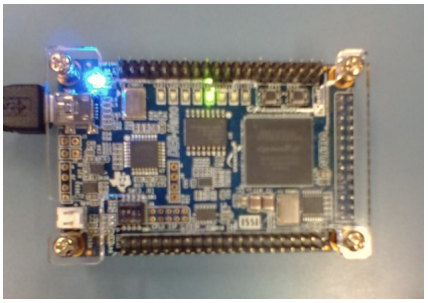

Figura 3. Ejemplo leyendo el valor en z

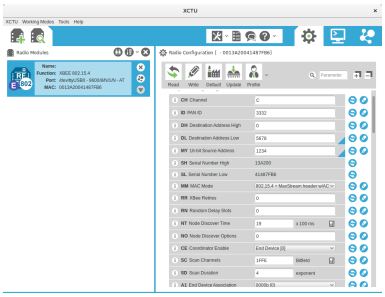

Figura 4. Configuración del primer $XBee$

IV-C. *Tercera parte de la práctica. Protocolo UART y XBee*

Al configurar los $XBee$s de la tercer parte, la mayor complicación con la que nos encontramos fue realizar la conexión de los $XBee$s con el microcontrolador. Fueron configurados correctamente usando el software $XCTU$ y, posteriormente, el código de Arduino fue verificado para asegurarnos de que corra sin problemas. Sin embargo, al momento de montar el $shield$ del $XBee$ en la tarjeta Arduino, experimentamos un $timeout$ de los datos y, en consecuencia, el $upload$ falló. Por falta de tiempo, no pudimos corregir problemas de la conexión del $shield$. Nos enteramos demasiado tarde de que la conexión de pines en el $shield$ tenían que conectarse entre ellos para que se lograran enviar comandos entre la tarjeta Arduino y el $XBee$.

De haber tenido más tiempo, quizá hubiera sido posible que el problema se hubiera arreglado. Parecería que aún quedaban varias cosas por intentar para configurar correctamente la comunicación. Entonces, los resultados obtenidos no fueron los esperados por una falta de conocimiento acerca de la implementación que se estaba utilizando a nivel de $Hardware$.

V. Conclusiones

V-A. *Humberto Martínez*

Las comunicaciones entre dispositivos resultó ser fascinante por la complejidad que presenta y por lo útil que resulta ser. Como se aprendió durante la práctica, no basta con conocer un dispositivo en particular para conocer la manera de hacer que éste se pueda comunicar con otros, sino que el protocolo que se utiliza es de suma importancia también. Las dificultades que se experimentaron con las comunicaciones SPI, por ejemplo, no consistieron en la comprensión de cada componente en

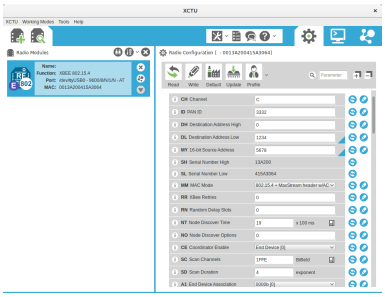

Figura 5. Configuración del segundo $XBee$

particular (acelerómetro y $FPGA$), sino en entender la forma en la que el $maestro$ se comunica con el $esclavo$ para escribir o leer valores. Quizá la explicación suena tan abstracta la primera vez que se escucha que no deja gran idea de cómo empezar a resolver el problema.

Otro ejemplo de la comprensión del protocolo como elemento de mayor importancia en este caso que la comprensión del componente se da en la comunicación por medio de $UART$, pues es necesario saber cuáles parámetros ajustar en la configuración para que los aparatos puedan comunicarse. Asimismo, una vez que se comunicaron, el siguiente reto es tratar de darle significado a los mensajes que se envían, y cómo manipularlos para obtener las reacciones deseadas en el sistema.

Finalmente, es posible concluir que, antes de escribir líneas de código sin saber por qué (simplemente siguiendo las instrucciones de la práctica), es necesario hacer la investigación necesaria. Resolver todas las dudas es una parte importante del trabajo, y las habilidades de investigación resultan cruciales. Como en cualquier aplicación que involucra el desarrollo de soluciones con tecnología, es necesario entender correctamente las especificaciones y el esbozo de la solución para lograr resolver el problema correctamente.

V-B. Juan Carlos Garduño

De esta práctica me llevo buenas experiencias para mi vida como ingeniero. En primer lugar, por primera vez estuve en contacto con protocolos de comunicación para la transferencia de datos. Es decir, comprendí lo útil que puede ser esta herramienta, no sólo para los ingenieros en mecatrónica, sino para ingeniería en telecomunicaciones.

En segundo lugar, logré comprender lo útiles que pueden ser cada un de estos protocolos y cuando es conveniente usar cada uno de ellos. En esta práctica se utilizan los protocolos para problemas relativamente sencillas, sin embargo, tienen múltiples aplicaciones en la vida real.

Siento que las sesiones de laboratorio no estuvieron bien coordinadas por varias razones. Personalmente, siento que eso afecto en la distribución en las horas y en la carga de trabajo.Finalmente, casi todo salió como lo esperábamos.

También me llevo la lección de que sabiendo buscar, todo se encuentra. Es decir, en múltiples ocasiones el equipo se enfrento con diferentes problemas, entonces nos pusimos investigar en $internet$ y todo esta ahí, no estoy diciendo que todo lo que se encuentre ahí este correcto, se tiene que hacer una depuración para lograr encontrar la información correcta.

V-C. Sebastián Aranda

Esta práctica fue realmente útil para trabajar con protocolos de comunicación, entender cómo funcionan, cuándo son útiles y cuándo es conveniente utilizar cada uno de ellos.

La mayor complicación durante la práctica fue cumplir con los plazos. Debido a inconvenientes durante los horarios asignados de laboratorio, se nos complicó como equipo cumplir con los plazos de manera ordenada y coherente.

Finalmente, en esta práctica particularmente fue de vital importancia realizar investigación previa a la realización del código. Teníamos que resolver todas la dudadas de manera ordenada para comprender las especificaciones y la solución y poder resolver el problema planteado por la gráfica.

VI. Rol o Papel

VI-A. Humberto Martínez

En las tres secciones de la práctica, Humberto tomó posesión por más tiempo del teclado. Humberto iba a escribiendo el código, incluyendo su propia lógica y las de sus compañeros. En cuanto al reporte Humberto elaboró las siguientes secciones:

- Introducción.
- Sus respectivas conclusiones.

VI-B. Juan Carlos Garduño

En la práctica, Juan Carlos se encargó junto con Sebastián a ir alambrando los componentes necesarios. En las tres partes de la práctica, iba dictando algunas secciones del código a Humberto que había investigado y podían ser de utilidad para la compilación del programa. De igual manera, iba revisando que la sintaxis de Humberto en el código, fuera la correcta. En cuanto al reporte Juan Carlos elaboró las siguientes secciones:

- Resultados.
- Marco teórico.
- Rol o papel.
- Resumen.
- Sus respectivas conclusiones.

VI-C. Sebastián Aranda

En la práctica, Sebastián se encargo del alambrado junto con Juan Carlos, así como de ir colaborando con los demás integrantes en la elaboración del código. En cuanto al reporte el elaboro las secciones siguientes:

- Desarrollo.
- Resultados.
- Sus respectivas conclusiones.

REFERENCIAS

[1] Tutorial Arduino con resistencia LDR, Geek Factory, 3 de noviembre de 2014. Disponible en: https://www.geekfactory.mx/tutoriales/tutoriales-arduino/tutorial-arduino-con-fotoresistencia-ldr/

[2] Anolog Devices. (2009). ADXL345. 7-Marzo-2019, de Analog Devices, Inc. Sitio web: https://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf

[3] DIGI. (2015). XBee/XBee-PRO S1 802.15.4 (Legacy). 15-Marzo-2019, de DIGI Sitio web: https://www.digi.com/resources/documentation/digidocs/PDFs/90000982.pdf

[4] Texas Instruments. (2016). L293x Quadruple Half-H Drivers. 15-Marzo-2019, de Texas Instruments Sitio web: http://www.ti.com/lit/ds/symlink/l293.pdf

[5] Arduino. (2017). SoftwareSerial Library. 10-Marzo-2019, de Arduino Sitio web: https://www.arduino.cc/en/Reference/SoftwareSerial

[6] Villafuerte, P.. (2014). Manejo del acelerómetro y brújula digital. 14-Marzo.2019, de Escuela Superior Politécnica del Litoral Sitio web: https://www.dspace.espol.edu.ec/bitstream/123456789/17023/1/Dise.pdf

[7] Navarro,K.. (2014). ¿Cómo funciona el protocolo SPI?. 10-Marzo-2019, de PANAMA HITEK Sitio web: http://panamahitek.com/como-funciona-el-protocolo-spi/