

# Práctica No.3 Protocolos de Comunicación.

## Principios de Mecatrónica

Arcadio Alexis Calvillo Madrid 159702  
Arlet Díaz Méndez 154840



**Resumen**—Durante esta práctica del laboratorio de Principios de Mecatrónica se exploraron varios medios de comunicación entre dispositivos. Para ello se implementaron distintos protocolos. La interacción de distintos componentes es importante en el desarrollo de sistemas embebidos dado que algunas tareas específicas pueden requerir señales de otros componentes para llevar a cabo sus propias tareas.

### 1. INTRODUCCIÓN

Los objetivos planteados para la realización de esta práctica son: aprender el funcionamiento de los protocolos 2C, UART y SPI para establecer comunicación entre dispositivos de manera alámbrica y utilizar XBee para comunicación inalámbrica. Además, establecer la configuración del acelerómetro de un FPGA para integrarlo a otro sistema. Esto contribuye al objetivo general del laboratorio de Principios de Mecatrónica de realizar un sistema embebido que interactúa con dispositivos distintos a la placa de Arduino. Para el desarrollo de esta práctica se tomaron tres sesiones, en la primera se desarrolló un ejemplo del protocolo I2C entre dos microcontroladores, para esto se trabajó en conjunto con otro de los equipos del laboratorio. En la segunda sesión se hizo una simulación del protocolo SPI utilizando el acelerómetro de un FPGA. Por último, en la última sesión se pusieron en práctica los protocolos UART y Xbee. Para el desarrollo de esta práctica se utilizó un Arduino Mega 2560, el IDE de Arduino, un FPGA de Altera, el IDE Quartus, un par de Xbee Series 1 y el IDE XCTU. En esta ocasión se usaron tres distintos IDEs dado que para cada dispositivo se utiliza un ambiente de programación diferente. Los IDEs Quartus y Arduino ya se han utilizado con anterioridad, el primero en el Laboratorio de Circuitos Lógicos y el segundo a lo largo de las prácticas anteriores. Por otra parte, XCTU se utiliza para la configuración de los XBee.

Este documento tiene la siguiente organización: un Marco Teórico en el que se explican las tecnologías utilizadas, así como los conocimientos teóricos que las respaldan, un Desarrollo en el que se explica cómo contribuyen los componentes a la solución, una sección de Resultados en la que se habla de los valores obtenidos durante la práctica, luego se presentan las Conclusiones de la práctica, los Roles de cada integrante del equipo y las Referencias utilizadas durante el proceso.

### 2. MARCO TEÓRICO

**Arduino Mega** Es una placa de Arduino (compañía de desarrollo abierto de hardware) que dispone de un microcontrolador AVR Atmel de 8 bits. Se considera la placa más potente de Arduino debido a que es la que más pines i/o tiene, es apta para trabajos más complejos que el que soportan otras placas. El Arduino Mega usa el microcontrolador Atmega2560.

**IDE Arduino** El entorno de desarrollo integrado de Arduino es una aplicación disponible para varios sistemas operativos desarrollado en Java que se utiliza para escribir y cargar programas en las placas Arduino. Este IDE permite el uso de los lenguajes C y C++ pero se utilizan reglas especiales de estructura de códigos.

**Lenguaje C** Es un lenguaje de programación imperativo, de propósito general, que admite programación estructurada, alcance de variable léxica y recursión, lo que se define como lenguaje de alto nivel. Sin embargo, cuenta con un sistema de tipo estático que evita muchas operaciones no intencionadas en bajo nivel. Lo anterior se debe a que es un lenguaje orientado a la implementación del sistema operativo Unix.

**FPGA** Una FPGA o matriz de puertas programables (Field-Programmable Gate Array). Dispositivo programable que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada en el momento mediante un lenguaje de descripción especializado. La lógica programable puede reproducir desde funciones tan sencillas como las llevadas a cabo por una puerta lógica o un sistema combinacional hasta complejos sistemas en un chip.

**XBee** Los 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 punto a multipunto o para redes punto a punto. Fueron diseñados para aplicaciones que requieren de un alto tráfico de datos, baja latencia y una sincronización de comunicación predecible.

**Protocolo I2C** I2C significa Circuito Interintegrado (Inter-Integrated Circuit). Con este protocolo se pueden integrar múltiples esclavos o múltiples maestros. De esta manera se pueden manejar varios microcontroladores. Es un protocolo de comunicación serial y síncrono. Usa dos vías de comunicación SDA Serial Data y SCL Serial Clock. Mediante

este protocolo el maestro manda señales al microcontrolador esclavo para que este ejecute alguna acción.

**Protocolo SPI** SPI significa Interfaz Periférica Serial (Serial Peripheral Interface). Es un protocolo serial que funciona en full-duplex para enviar y recibir información. El maestro envía información al esclavo. En este protocolo solo puede haber un maestro y varios esclavos, los cuales pueden estar en modo cascada o paralelo.

**Protocolo UART** UART significa Transmisor-Receptor Asíncrono Universal (Universal Asynchronous Receiver-Transmitter). A diferencia de I2C y SPI, se trata de un protocolo de comunicación asíncrona. Tiene designados pines de Transmisión de datos y de Recepción de datos. La mayor ventaja que ofrece este protocolo es que la comunicación entre los dispositivos se da de forma inalámbrica en una distancia de hasta 10m. Sin embargo, solo se pueden conectar 2 dispositivos.

### 3. DESARROLLO

La práctica constó de tres sesiones. Durante la primera sesión, se desarrolló en conjunto con otro equipo del laboratorio la implementación del protocolo I2C. El microcontrolador de nuestro equipo fue el maestro, mientras que el microcontrolador del otro equipo fue el esclavo. Mediante las señales enviadas de nuestro dispositivo al otro, se hizo girar un motor en ambos sentidos. Además, se implementó el freno pasivo y el freno activo.

En la segunda sesión, se realizó la implementación del protocolo SPI para realizar la comunicación del acelerómetro de la tarjeta DE0-Nano para cambiar el sentido de un motor. Para utilizar este protocolo fue necesaria la utilización de un FPGA, mediante la cual se llevó a cabo la comunicación entre el acelerómetro y el motor. Los documentos iniciales para la configuración el acelerómetro fueron provistos por el profesor. El quipo se encargó de establecer los valores adecuados para lograr el funcionamiento requerido.

En la tercera sesión, se realizó la implementación del protocolo UART para comunicar la tarjeta Arduino con otro dispositivo mediante los XBee. Para lograr esta comunicación es necesario establecer los parámetros iniciales que abren un canal inalámbrico mediante el cual son transmitidas las señales. Para este protocolo fue necesario utilizar leds que encendían dependiendo de la señal recibida.

### 4. RESULTADOS

Los resultados obtenidos durante la práctica fueron satisfactorios. Sin embargo, fue necesario invertir más tiempo que el de la duración de las tres sesiones para lograr el funcionamiento de todos los ejercicios. En especial para la implementación del protocolo SPI, pues la configuración de la FPGA para el manejo del acelerómetro no se había realizado con anterioridad por el equipo.

### 5. CONCLUSIONES

**Arlet:** Durante esta práctica pude conocer de manera práctica como funcionan tres distintos protocolos de comunicación. Es interesante conocer las diferencias que existen entre estos protocolos, tales como la capacidad de transmisión y recepción, la sincronía o asincronía. Otro aspecto

importante es la capacidad de comunicar multiples dispositivos, a pesar de que en esta práctica solo se conectaron 2 dispositivos por protocolo. Los medios de comunicación entre dispositivos son necesarios en sistemas embebidos para poder llevar a cabo algunas tareas del sistema global.

**Alexis:** La comunicación entre dispositivos ayuda a tener control sobre estos. Es decir, ya no sólo se puede tener el control de uno de los mecanismos de un sistema, sino hacer que entre ellos interactúen sin la necesidad de alguien que lo esté "vigilando". Un concepto importante que debemos recordar es la de maestro-esclavo; tal como ocurre en programación en la nube un "maestro." ordena qué es lo que hay que hacer teniendo así aún más control sobre los dispositivos.

### 6. ROL O PAPEL

**Arlet:** Programar algunas partes del código para los distintos protocolos. Actualizar las versiones de la práctica en GitHub. Redacción y corrección del reporte escrito.

**Alexis:** Programar algunas partes del código tanto en C como en ensamblador. armar el circuito electrónico. Redacción y corrección del reporte escrito.

### 7. FUENTES CONSULTADAS

[1] Drouiz. UART vs SPI vs I2C Diferencias entre protocolos. Aprendiendo Arduino. [En línea]. Disponible en: <https://www.drouiz.com/blog/2018/06/25/uart-vs-spi-vs-i2c-diferencias-entre-protocolos/>

[2] Tesla Bem. Fundamentos del Protocolo I2C. [En línea]. Disponible en: <https://teslabem.com/nivel-intermedio/fundamentos-del-protocolo-i2c-aprende/>

[3] Wikipedia. FPGA. [En línea]. Disponible en: [https://es.wikipedia.org/wiki/Field-programmable\\_gate\\_array](https://es.wikipedia.org/wiki/Field-programmable_gate_array)

[4] XBee Cl. ¿Qué es Xbee? [En línea]. Disponible en: <https://xbee.cl/que-es-xbee/>

[5] Wikipedia. Universal Asynchronous Receiver-Transmitter. [En línea]. Disponible en: [https://es.wikipedia.org/wiki/Universal\\_Asynchronous\\_Receiver-Transmitter](https://es.wikipedia.org/wiki/Universal_Asynchronous_Receiver-Transmitter)