

Práctica No. 3 Protocolos de Comunicación

Departamento Académico de Sistemas Digitales
Instituto Tecnológico Autónomo de México
Primavera 2019

JEAN PAUL VIRUEÑA ITURRIAGA
FABIÁN ORDUÑA FERREIRA

155265
159001

Abstract - En esta práctica se aprendió a configurar protocolos de comunicación (I2C, SPI, UART) seriales o inalámbricos, síncronos o asíncronos, de tipo maestro-esclavo o transmisor receptor. Todos esto con el fin de poder implementarlos para ver sus ventajas y desventajas y al mismo tiempo realizar distintas funcionalidades como: buscaba establecer la dirección de un motor con ayuda de un puente H, ver cómo funciona un acelerómetro, prender y apagar Leds y enviar un mensaje cuando cierto evento ocurre, tal como al presionar un botón.

I. INTRODUCCIÓN

En la actualidad, la tecnología es un recurso que tenemos para optimizar nuestras actividades diarias. Las computadoras son un gran ejemplo de esto, mismas que pueden realizar diversas y muy complejas tareas.

Sin embargo, hay ocasiones en las que se desean realizar tareas sencillas, concretas y sin mucha complejidad. Es por esto que existen emplean los sistemas embebidos, sistemas pueden ser considerados como computadoras sencillas ya que cuentan los componentes básicos de las mismas. Muy seguido, se emplean es necesaria la comunicación entre un dispositivos y un sistema embebido para que este último sea capaz de realizar la tarea para la cual fue concebido.

Por tal motivo, es que en esta práctica tenemos como objetivos implementar y entender las ventajas y desventajas de tres protocolos de comunicación. Para esto se realizaron varios ejercicios donde se buscaba establecer la dirección de un motor con la ayuda de un puente en H, ver cómo funciona un acelerómetro, prender y apagar Leds y enviar un mensaje al presionar un botón. Todo esto mientras se comprendía cómo se

determinaban, se establecían y se leían los valores y parámetros adecuados de los registros de los distintos protocolos y cómo se implementan los dispositivos o herramientas externas a la placa del Arduino.

Este reporte está organizado en distintas secciones, comenzando con un marco teórico, en el que se presenta la información relevante y de importancia para una mejor comprensión de lo que aquí se aborda; seguido del desarrollo, donde se presenta la manera en que se trabajó para la obtención de los resultados; la sección de resultados, en la que se presenta un análisis de lo obtenido y la sección de conclusiones.

II. MARCO TEÓRICO

Los sistemas embebidos son sistemas basados en microcontroladores o microprocesadores los cuales son diseñados para realizar una tarea en específico; es decir, para realizar operaciones específicas, comúnmente simples y en tiempo real. Los microcontroladores son circuitos integrados, programables, compuestos por diversos grupos funcionales. Son “computadoras pequeñas” ya que cuentan con las unidades funcionales básicas de una computadora: unidad central de procesamiento, memoria y periféricos de entrada y salida [6] y [16].

Para que un sistema embebido pueda comunicarse con cualquier otro dispositivo es necesario emplear un protocolo de comunicación, es decir las reglas y procedimientos necesarios para poder enviar y transmitir datos en una red [3]. Estos pueden tener una comunicación de tipo serial, que a diferencia de la comunicación en paralelo, donde se transmiten los bits en buffers de información múltiples de 8 bits, la comunicación serial transmite los bits en una sola ráfaga [7]. También pueden ser de tipo asíncrona o síncrona

Otras maneras de clasificar a dichos protocolos es por su tipo de comunicación: cableada, donde el medio con cables o inalámbrica donde la comunicación es sin cables, usualmente ondas electromagnéticas, y sincrónica, es decir que necesita una señal de reloj, o asincrónica, que no necesita ninguna señal de reloj [13 y 15].

maestro-esclavo, donde un dispositivo que cumple con el rol de maestro comunica un requerimiento y otro dispositivo que cumple la función de esclavo se encarga de realizar dicho requerimiento [12]. Otro tipo de protocolos son los protocolos de telemetría donde un transmisor (Tx), dispositivo que transmite una señal por un medio, le manda información por medio de la señal a un receptor (Rx), dispositivo que recibe una señal por un medio [4, 10 y 11].

En nuestro caso se implementaron los siguientes tres protocolos:

- El protocolo I2C, ~~que quiere decir inter integrated circuits~~ para referirse a la interconexión de circuitos integrados es un protocolo serial síncrono maestro-esclavo cableado que conecta a dispositivos de baja velocidad mediante dos cables y donde cada esclavo tiene una dirección para recibir los requerimientos del maestro [5].
- El protocolo SPI, ~~que quiere decir serial peripheral interface~~ es un protocolo que permite la comunicación serial síncrona maestro-esclavo cableada en modo *Full duplex*, es decir que puede enviar información de los dos lados al mismo tiempo, suele usarse entre con microcontrolador como maestro y un componente periférico como esclavo [7].
- El protocolo UART, ~~que quiere decir Universal Asynchronous Receiver/Transmitter~~ es un microchip que permite la comunicación serial o paralela síncrona cableada de tipo transmisor-receptor entre un microcontrolador y sus periféricos [14].

Zigbee es un estándar para comunicación inalámbrica para redes de área personal, de bajo consumo que emplea señales de radio como medio [], para configurarlo se utilizó XCTU, un software multiplataforma con el que puedes visualizar, actualizar o configurar los módulos Xbee[1]

Un ~~servomotor~~ es un sistema compuesto de partes electrónicas y mecánicas para poder mantener el eje de un motor en una posición fija [8]. Con la ayuda de un puente en H (Figura 1), un circuito electrónico que transmite corriente directa, se le da la dirección al motor o se le indica si debe de frenar de manera pasiva o activa [18]. Para dirigirlo se utiliza la ayuda de un acelerómetro, un dispositivo que mide la vibración o el cambio de movimiento en una estructura [9].

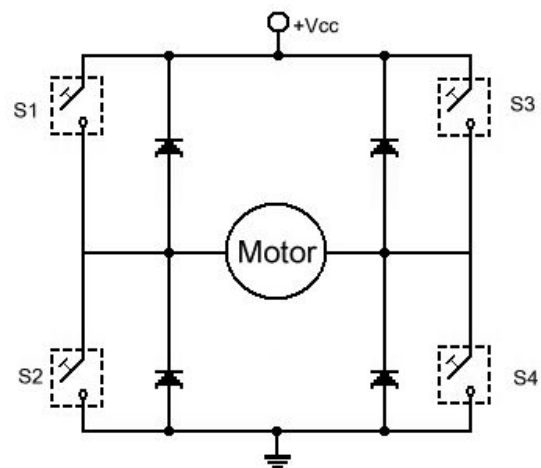


Figura 1 - Diagrama de un puente en H

III. DESARROLLO

Para cumplir con los objetivos de la práctica se realizaron tres ejercicios para aplicar distintos tipos de protocolos de comunicación para la interacción entre ciertos componentes. Consistió en realizar comunicación entre microcontroladores con Arduinos Mega 2560, con una FPGA y con Xbee. Los ejercicios se describen a continuación:

En el primer ejercicio fue necesario colaborar con otro equipo de laboratorio para poder enlazar su microcontrolador con el nuestro por medio del protocolo I²C, y poder generar pausas y movimiento de un motor mediante un puente H. Los códigos que se programaron fueron: un código que sirviera con un microcontrolador para que dados dos botones fuera capaz de codificar sus valores de entrada y para que pudiera transmitirlos al otro microcontrolador mediante el protocolo I²C y, un código que fuera capaz de leer valores transmitidos por otro microcontrolador mediante el protocolo I²C y que dependiendo del valor recibido fuera capaz de mandar valores de salida hacia un puente H para controlar movimientos de un motor.

El segundo ejercicio consistió en emplear el protocolo SPI junto con un acelerómetro. Se tomaron archivos iniciales que fueron proporcionados en el repositorio https://github.com/garygra/PM_pract_3, de allí se modificaron archivos con la finalidad de configurarlos para emplear la tarjeta FPGA junto con el acelerómetro que contiene. Asimismo, fue necesario configurar las frecuencias de lectura de registros ya que el acelerómetro reporta los valores a una frecuencia en específico y estos pueden leerse con errores si las frecuencias no empatan.

El tercer ejercicio consistía en configurar dispositivos XBee con la ayuda de XCTU, un programa empleado para configurarlos. También fue necesario escribir un código, empleando la librería *SoftwareSerial* para implementar el protocolo UART, con la finalidad de recibir los valores que se transmitían los Xbee para emplearlos y prender unos LEDs. En este mismo código, se implementó la opción de leer un valor obtenido de un botón y dependiendo de su valor se transmitía o no un mensaje del Xbee conectado al microcontrolador.

IV. RESULTADOS

En esta práctica, en lo que concierne al primer ejercicio fuimos capaces de transmitir información entre Arduinos Mega 2560 con el protocolo I²C.

Al trabajar con otro equipo, es importante acoplarse y entender la manera en que realizan las cosas; ya que, la forma en que debíamos leer los datos que transmitieron dependían de eso. Una vez que nos acoplamos, y después de haber entendido la manera en que funciona el puente H, transformamos la información recibida para hacer que el motor se moviera de tal forma que resultó exitoso nuestro trabajo en esta sección.

En cuanto al segundo ejercicio, tuvimos dificultades al tratar de configurar dentro del código los parámetros necesarios para llevar a cabo las tareas deseadas. Una de las cosas que afectó la forma en que hicimos las cosas fue el factor de que no conocemos a gran profundidad VHDL, el lenguaje empleado en la tarjeta FGA. Es por esto que, en este ejercicio, no llegamos al resultado esperado que era, mover por medio del acelerómetro de la FPGA un motor.

En el tercer ejercicio, conocimos la manera en que se realiza intercambio de información y lo implementamos exitosamente entre dispositivos conectados inalámbricamente con los XBee y el protocolo de comunicación UART. Fue fundamental la forma en que configuramos los dispositivos, ya que de esto dependía que la información llegara al destino esperado. Por otro lado, para poder configurar los dispositivos fue medular recordar que siempre debemos tener presente que los dispositivos pueden fallar físicamente y que, cambiar los dispositivos suele ser una buena práctica cuando las cosas no resultan como se espera. Lo anterior porque este tipo de dificultades se presentaron con los Xbee y con los cables del microcontrolador.

V. CONCLUSIONES

Jean Paul Virueña Iturriaga

Fue muy interesante ver cómo se configuraban los protocolos de comunicación para realizar distintas funcionalidades aunque fue confuso configurarlos en un principio. A pesar de que se sufrió mucho por culpa de errores de hardware que no podíamos

controlar los resultados fueron satisfactorios y recobré motivación para seguir con la carrera de Ingeniería en Telecomunicaciones.

Fabián Orduña Ferreira

Gracias a esta práctica aprendí que existen diversos protocolos de comunicación en dispositivos de distintos tipos; fui capaz de implementar comunicaciones en dispositivos conectados inalámbrica y alámbricamente; conocí la forma en que un puente H funciona y como afecta esto al funcionamiento de un motor; aprendí la forma de emplear los Xbees para transmitir datos inalámbricamente; aprendí la forma en que los acelerómetros funcionan; y disfruté las cosas interesantes que implementamos, precisamente porque aprendí nuevas cosas.

VI. ROLES

Para el desarrollo de las actividades, de las que mostramos previamente los resultados, ambos miembros del equipo colaboramos de forma conjunta. En lo que a este documento concierne, Paul se enfocó en mayor proporción al abstract, a la introducción y al marco teórico, mientras que Fabián se enfocó más al desarrollo y a los resultados.

VII. FUENTES DE CONSULTA

[1] Aprendiendo Arduino. (n.d.). ZigBee/ XBee. Retrieved March 14, 2019, from <https://aprendiendoarduino.wordpress.com/tag/xctu/>

[2] Byte Paradigm. (n.d.). Introduction to I²C and SPI protocols. Retrieved March 14, 2019, from <https://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/?/article/AA-00255>

[3] CCM.net. (n.d.). Protocolo de comunicación. Retrieved March 14, 2019 from

<https://es.ccm.net/contents/275-protocolo-de-comunicacion>

[4] Ecured. (n.d.). Transmisor. Retrieved March 14, 2019, from <https://www.ecured.cu/Transmisor>

[5] I2C. (n.d.). I2C Info. Retrieved March 14, 2019, from <https://i2c.info/>

[6] IoT Agenda. Microcontroller. Retrieved February 20, 2019, from <https://internetofthingsagenda.techtarget.com/definition/microcontroller>

[7] LigatureSoft. (n.d.). Serial and Parallel data transmission. Retrieved March 14, 2019, from https://web.archive.org/web/20111103174230/http://www.ligaturesoft.com:80/data_communications/data-transmission.html

[8] Menna. (n.d.). Como funciona un servomotor. Retrieved March 14, 2019, from <https://como-funciona.co/un-servomotor/>

[9] Omega. (n.d.). Acelerómetro. Retrieved March 14, 2019, from <https://es.omega.com/prodinfo/acelerometro.html>

[10] Real Academia de Ingeniería. (n.d.). Telemetría. Retrieved March 14, 2019, from <http://diccionario.raing.es/es/lema/receptor-de-telemetría>

[11] Real Academia de Ingeniería. (n.d.). Receptor de telemetría. Retrieved March 14, 2019, from <http://diccionario.raing.es/es/lema/telemetría-0>

[12] ResearchGate. (n.d.). Modelo Maestro-Eslavo. Retrieved March 14, 2019, from https://www.researchgate.net/figure/Figura-4-Modelo-Maestro-Eslavo-El-maestro-es-el-que-inicia-un-requerimiento-de_fig4_279531120

[13] Spark Fun. (n.d.). Serial Communication. Retrieved March 14, 2019, from <https://learn.sparkfun.com/tutorials/serial-communication/all>

[14] Techopedia. (n.d.). What is a Universal Asynchronous Receiver/Transmitter (UART)? - Definition from Techopedia. Retrieved March 14, 2019, from <https://www.techopedia.com/definition/3669/universal-asynchronous-receivertransmitter-uart>

[15] Techopedia. (n.d.). What is Synchronous Transmission? - Definition from Techopedia. Retrieved March 14, 2019, from <https://www.techopedia.com/definition/13389/synchronous-transmission>

[16] Tutorialspoint. Embedded Systems - Overview. Retrieved February 20, 2019, from https://www.tutorialspoint.com/embedded_systems/es_overview.htm

[17] WhatIs.com. (n.d.). What is serial peripheral interface (SPI)? - Definition from WhatIs.com. Retrieved March 14, 2019, from <https://whatis.techtarget.com/definition/serial-peripheral-interface-SPI>

[18] Wikibooks. (n.d.). Robótica/Puente H. Retrieved March 14, 2019, from https://es.wikibooks.org/wiki/Rob%C3%B3tica/Puente_H