

LABORATORIO NO.2



1st Juan David Leon
Ingeniería Mecatrónica
Universidad ECCI
Bogotá, Colombia
juand.leong@ecc.edu.co

2st Oscar Enrique La Rota
Ingeniería Mecatrónica
Universidad ECCI
Bogotá, Colombia
oscared.larotah@ecc.edu.co

3st Juan Sebastian Vera
Ingeniería Mecatrónica
Universidad ECCI
Bogotá, Colombia
Juans.verah@ecc.edu.co

I. RESUMEN

En este informe se pretende mostrar el desarrollo de la práctica número dos de electiva profesional, donde se adquirió datos y se calibró la IMU (MPU6050) para ello, utilizando una comunicación serial con la STM32F767ZI (programación en M BED) con el software MATLAB y la Raspberry PI observando la comunicación remota de esta con el computador. Adquiriendo una serie de gráficas a analizar su comportamiento tanto calibrado como no calibrado el sensor llegando a una conclusión de cuál es mejor. (ya que en clase se observó que a veces es mejor no tener el sensor MPU es mejor.)

Palabras claves—Adquisición, Datos, MPU6050, MBED, Programación, Raspberry PI, STM32F767ZI.

II. INTRODUCCIÓN

En el presente informe se mostrarán los resultados obtenidos en el desarrollo de la práctica con el fin de comprender las implicaciones como también el funcionamiento de la comunicación y adquisición de datos entre la tarjeta de desarrollo y el sensor; haciendo uso de la comunicación serial para adquirir los datos a través del sensor IMU (MPU6050) con la STM32F767ZI y la Raspberry PI con el software MATLAB, adquiriendo datos de forma real. Sin embargo, para una mejor comprensión de dicho funcionamiento se hace necesario dar énfasis tanto en el funcionamiento del bloque de programación como en el funcionamiento del montaje físico. Dicho análisis permitirá corroborar habilidades básicas para la configuración del sensor y la configuración general de un sistema embebido.

III. MARCO TEÓRICO

III-A. COMUNICACION SERIAL:

I2C es un puerto y protocolo de comunicación serial, define la trama de datos y las conexiones físicas para transferir bits entre 2 dispositivos digitales. ... Además el protocolo permite conectar hasta 127 dispositivos esclavos con esas dos líneas, con hasta velocidades de 100, 400 y 1000 kbits/s.

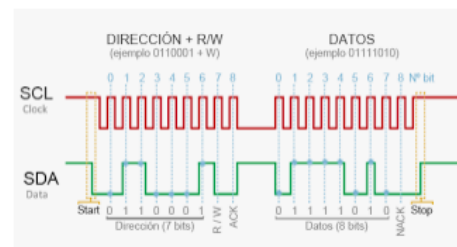


Figura 1. Protocolo i2c

III-B. NUCLEO STM 32F767ZI:

La tarjeta de desarrollo STM32F767ZI tiene unos módulos digitales y estos mismos están conectados a unos pines que están ya ensamblados por la tarjeta para verificar esto se tiene que tener el manual de la tarjeta que se va a usar, ya que cada tarjeta es diferente, porque para la configuración y programación de los pines de entrada o salida cambian y se tiene que estar seguros de lo que queremos hacer, para que cuando se programe sea la forma correcta y saber que buses de datos se manejan, para tener un mejor dominio de

la estructura y del dominio de la programación que está en Lenguaje C.



Figura 2. Tarjeta de desarrollo STM32F767ZI

III-C. SENSOR MPU 6050:

El sensor MPU6050 permite obtener datos en tiempo real de una posición determinada, el módulo acelerómetro MPU tiene un giroscopio de tres ejes con el que se puede medir velocidad angular y un acelerómetro también de 3 ejes para medir los componentes X, Y y Z de la aceleración.

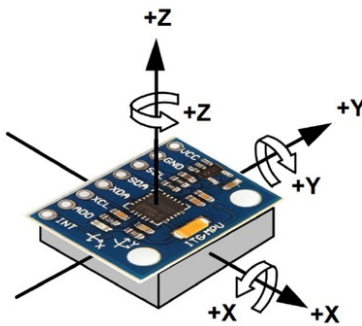


Figura 3. Sensor MPU6050 AXIS

III-D. MATLAB IDE COMUNICACION :

La comunicación en serie es el protocolo de bajo nivel más común para la comunicación entre dos o más dispositivos. Por lo general, un dispositivo puede ser un ordenador, mientras que el otro puede ser un módem, una impresora, hardware que en este caso es la tarjeta 32F767ZI, otro ordenador o un instrumento científico, como un osciloscopio o un generador de funciones. Para muchas aplicaciones de puerto serie, es posible comunicarse con el instrumento sin que sea necesario conocer en detalle cómo funciona el puerto serie. La comunicación a través de un puerto serie se establece con un objeto serialport, que se crea en el área de trabajo de MATLAB

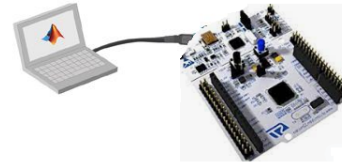


Figura 4. Sensor MPU6050 AXIS

III-E. RASPBERRY PI:

La Raspberry Pi es la placa de un ordenador simple compuesto por un SoC, CPU, memoria RAM, puertos de entrada y salida de audio y vídeo, conectividad de red, ranura SD para almacenamiento, reloj, una toma para la alimentación, conexiones para periféricos de bajo nivel, reloj, la Raspberry Pi básicamente es un ordenador reducido en el cual se pueden ejecutar distintas funciones y para distintas aplicaciones en específico, la Raspberry Pi usa un sistema operativo para su función de ordenador en este caso hay muchos tipos de sistemas operativos como lo son Raspbian, Kali Linux y Pidora, con este sistema operativo se pueden hacer funciones similares a windows 10.

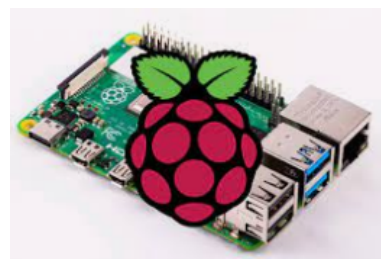


Figura 5. Sensor MPU6050 AXIS

III-F. COMUNICACION REMOTA O VNC:

VNC son las siglas en inglés de Virtual Network Computing (Computación Virtual en Red). VNC es un programa de software libre basado en una estructura cliente-servidor que permite observar las acciones del ordenador servidor remotamente a través de un ordenador cliente, permitiendo ejecutar acciones las cuales no son posibles en un control de campo, usualmente esta comunicacion se hace en aplicaciones de alta complejidad, acceso difícil, o para una conexión rápida y segura una herramienta útil para el ingeniero de campo el cual tiene a su disposición varios equipos que no se encuentran en su región de origen pero que necesitan un mantenimiento oportuno.



Figura 6. Sensor MPU6050 AXIS

IV. METODOLOGÍA

El desarrollo de esta práctica empieza con la debida programación del software MBED como también la configuración de la Tarjeta [2], de forma que el sensor [6] empiece la captación de datos necesaria para proceder a la comunicación I2C y serial propuesta. Con esta comunicación se quiere obtener datos y visualizarlos es por esto que se necesita la configuración del puerto con MATLAB y realizar la programación.

IV-A. Programación y protocolo I2C

En primer lugar se programan los registros de la tarjeta, para esto también se debe indicar que pines vamos a utilizar, en este caso se definieron los pines PB8 y PB9 que corresponden al reloj SCL y los datos SDA para el protocolo I2C, para que el sensor registre los datos se debe programar también su registro el cual está determinado por un número hexadecimal que en este caso es el **0x68** el cual es para la identificación del sensor y su conexión determinando si se estableció la conexión entre la tarjeta [2] y el sensor.

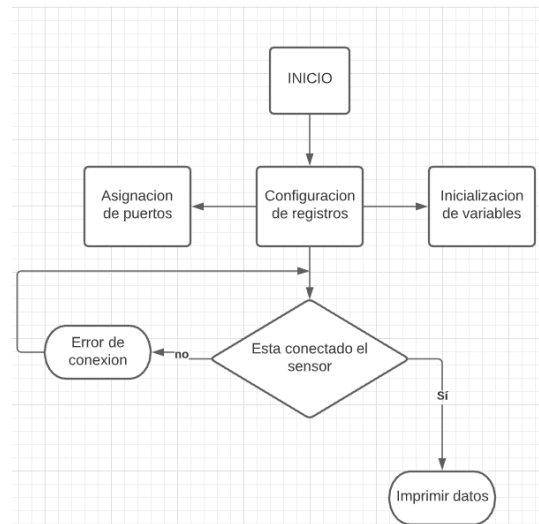


Figura 7. Diagrama de flujo del proceso

IV-B. Conexión STM32F767ZI con MPU:

La conexión de la tarjeta y el sensor es la siguiente:

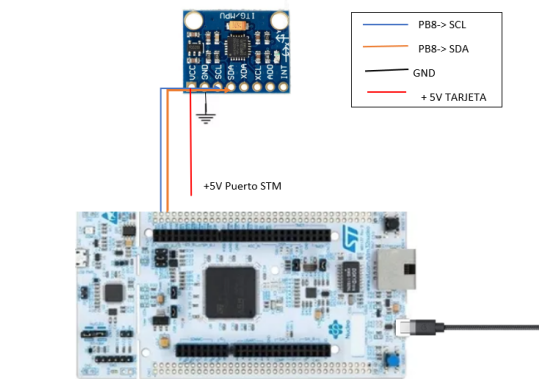


Figura 8. Circuito Esquemático

IV-C. Comunicación serial

La comunicación serial será la más importante esto debido a que allí se mostrarán los datos arrojados por el sensor [6], para esto se hace en la STM32F767ZI la cual establece la comunicación serial con la Raspberry PI, que a su vez establece el protocolo I2C y por consiguiente se empieza la captación de los datos actuando como "maestro", esto porque la STM32F767ZI en esta configuración actuará como ese puente entre la conversión de datos y pasar la información a la Raspberry y esta a su vez se visualizará en la pantalla del computador con los datos calibrados.

Se entiende que la conexión entre estas dos placas es a través de un cable USB / comunicación serial.

IV-D. Adquisicion de datos

Segun la figura 7, y sabiendo que la adquisicion de datos es a traves de la STM, es necesario visualizar esta adquisicion, por esto se procede a la utilizacion del software MATLAB, para la visualizacion de estos datos, configurando el puerto COMz plateando estos datos en graficas de los planos X,Y e Z.

IV-E. Conexion STM32F767ZI Con Raspberry PI:

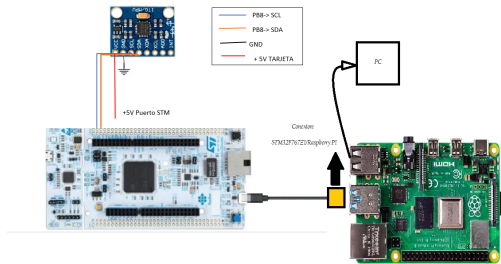


Figura 9. Conexion STM / Raspberry PI.

IV-F. Calibracion Sensor IMU:

Viendo es elquematico de la figura 7, y sabiendo que la visualizacion de datos es por MATLAB, pero a diferencia del anterior laboratorio es usado la raspberry PI, primero se procede a calibrar el sensor con phyton en el entonrno de la tarjeta ya mencionada.

IV-G. Programacion en PYTHON

A continuacion la pantalla de inicio y finalizacion para la captacion de datos en python.

```
===== RESTART: /home/pi/Actividad9AM.py =====
Serial<id=0x75468d70, open=True>(port='/dev/ttyACM1', baudrate=9600, bytesize=8,
parity='N', stopbits=1, timeout=10, xonxoff=False, rtscts=False, dsrdtr=False)

Quiere adquirir los datos S/N
n
Adios
```

Figura 10. Comunicacion Sereal Puertos.

En este caso al Mover el sensor IMU, se abriera se captaran los datos, para ello se inicializa otra vez la comunicacion sereal esto con el objetivo de que siempre se este leyendo la entrada del puerto.Siendo desicion del usuario con s, para si y n, para no iniciar la obtencion de datos.

IV-H. Programacion en MATLAB

Se pretende realizar La conexion con un puerto COM con el software MATLAB, para la adquisicion de datos y estos se guarden en una matriz del commad windows.

Donde la variable temp, cumple la funcion principal ya que es la encargade convertir los datos de string a numericos y colocarlos en el arreglo.

```
%crea un puerto serial
% s= serial('COM3','BaudRate',9600,'DataBits',8,'Parity','None','StopBits',1);
if ~exist('s','var')
s= serial('COM3','BaudRate',9600,'DataBits',8,'Parity','None','StopBits',1);
end
```

Figura 11. Comunicacion Sereal Puertos.

```
%lectura de un string
tic

t_ini=toc; t=0; i=1;
while 1
    a = fscanf(s);
    temp = cellfun(@str2num,strsplit(a','));
    values(i,:)=temp;
    figure(1);
    plot(values(:,1));
    figure(2);
    plot(values(:,2));
    pause(0.000000001);
    i=i+1;
    t = toc - t_ini;
end
fclose(s);
```

Figura 12. Lectura Datos.

```
temp = cellfun(@str2num,strsplit(a','));
```

Figura 13. Escaneo de datos.

Se observa en la figura 13 el comando @str2num que es el comando encargado de pasar los datos a numeros, todo guardado en parentesis por la funcion cellfun() para llenar las filas y strsplit(a',') a dentro con el fin de separar cada datos con coma ya que la letra a guarda el comando a = fscanf(s) que es es queesta constantemente escaneando el puerto para la recepcion de datos, hasta que ya se desida a traves de la Raspberry PI no seguir captando.

Ahora en la figura 12, en el comando value, se visualiza, values(i,:) = tempesto con el fin de visualizar las filas y llenar las filar , complementando el comando cellfun y se procede a plotear la salida de los datos.

V. RESULTADOS

V-A. GRAFICAS SENSOR MATLAB

V-B. ACELEROMETRO

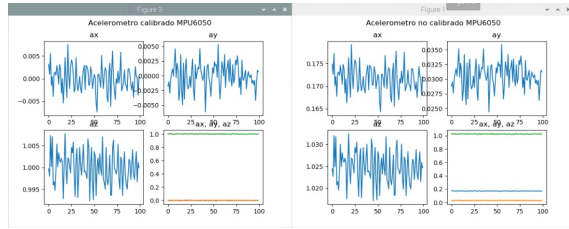


Figura 14. Graficas Acelerometro.

V-C. GIROSCOPIO

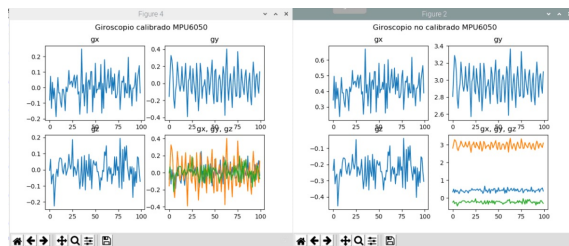


Figura 15. Graficas Giroscopio.

analizando las figuras 14 y 15 , se observo que la sensibilidad de datos en cero estan unidos y cambian a partir de este punto , mientras que al no estar calibradas, oscilan en diferentes puntos.

VI. CONCLUSIONES

- El protocolo I2C nos permite conectar varios esclavos a un solo maestro permitiendo tener mejor control sobre los perifericos que se eesten implementando.
- La tarjeta STM32F767ZI cumple un papel importante debido a que esta se encarga de suministrar los datos a un tiempo determinado, sin esto causaria que los datos de los ejes no estuvieran en sintonia con su tiempo exacto si no estarian aleatorios.
- La calibracion de un sensor es importante porque nos determina la fidelidad del instrumento y sus datos, en este caso se puede evidenciar una diferencia notoria entre datos calibrados y no calibrados
- La comunicacion vcn permite un manejo remoto de cualquier instrumento conectado a un cerebro o maestro facilitando su acceso en cualquier parte del mundo
- El sistema operativo de la raspberry pi nos permite graficar y tambien programar en lenguaje python esto nos determina que es un sistema operativo muy competitivo y con grandes ventajas industriales
- Es importante buscar el puerto por ACM debido a que windows maneja el puerto COM que basicamente es similar al ACM

REFERENCIAS

[1]I2C, Aprendiendo Arduino1 I2C, 2021. [Online]. Available: <https://aprendiendoarduino.wordpress.com/2017/07/09/i2c/>. [Accessed: 02- Mar- 2021].

[2]"Placa De Desarrollo Stm32 Nucleo-144 - Stm32f767zi2,A rticulo.mercadolibre.com.co, 2021. [Online]. Available: <https://articulo.mercadolibre.com.co/MCO-575029706-placa-de-desarrollo-stm32-nucleo-144-stm32f767zi-j.M>. [Accessed : 02 – Mar – 2021].

[3]N. mechatronics, "Tutorial MPU6050, Acelerómetro y Giroscopio", Naylamp Mechatronics 3 - Perú, 2021. [Online]. Available: https://naylampmechatronics.com/blog/45_tutorial-mpu6050-acelerometro-y-giroscopio.html. [Accessed : 02 – Mar – 2021].