"Puesta a punto del sensor MPU 5060 por medio de la tarjeta Raspberry Pi 3+"

1st Esteban Olmos
, 2st Vanesa Lara, 3st Mario Galeano
Ingeniería Mecatrónica
Universidad ECCI
Bogotá, Colombia
johne.olmosc@ecci.edu.co, vanesay.larac@ecci.edu.co,marioe.galeanor@ecci.edu.co

I. Resumen

Resumen: En este documento se encuentra concentrada la información acerca de los procesos de calibración y adquisición de datos del sensor MPU5060 por medio de la tarjeta *"Núcleo-F411RE"*; adicional cuenta con el paso a paso para obtener la conectividad entre la tarjeta Núcleo STM32 (como adquisición de datos), la Raspberry y el sensor. Por último, se encuentra el análisis de la calibración de los sensores acelerómetro y giroscopio haciendo uso de Matlab para la visualización de las gráficas.

En conjunto con el análisis del funcionamiento y comportamiento del sistema, se encuentran las descripciones detalladas del orden en el que se debe ejecutar cada una de estas tareas para obtener los datos de la manera correcta, ya que de la calibración depende la precisión del sensor y con ello las mediciones generales que se pretendan tomar, de acuerdo a la aplicación para la que se disponga el sensor. Finalmente, se muestran las conclusiones y resultados de los procesos anteriormente nombrados, esto con el fin de socializar posibles errores que se pueden llegar a cometer en el desarrollo del código de programación o dentro de este mismo la parametrización de alguna variable asociada.

***Palabras claves*—Conexión VNC, lazo, calibración, giroscópio, acelerómetro.**

II. Introducción

En el desarrollo del análisis se presenta gráficamente el resultado de la medición de los sensores: giroscópio y acelerómetro con las unidades de medición correspondientes a cada uno de estos sensores (/s y m/s2 respectivamente), esto se logra a través del enlace entre dos tarjetas embebidas (Núcleo STM32 y Raspberry Pi 3+), para lo cual se acude a trabajos anteriores desarrollados en conjunto con el equipo de trabajo, en los que de implementa un código de programación desarrollado desde la plataforma Mbed de la tarjeta STM32,

En las anteriores oportunidades se han realizado trabajos relacionados con el sensor nombrado anteriormente, donde se ha realizado el análisis de dichos sensores en estado *"sin calibración"*, implementando como interfaz grafica Matlab y como interfaz de desarrollo Mbed.

El objetivo principal es realizar la calibración del sensor PMU 5060, adquirir y procesar los datos obtenidos de la calibración del sensor, puesto que a partir del análisis de la comparación de datos y gráficas de los sensores en estado de calibración óptimo y no tan óptimo (sin calibración alguna). Adicional a lo anterior se pretende realizar en enlace de comunicación directo entre la tarjeta Raspberry Pi3B+, un PC (Portatil Computer) y una pantalla auxiliar, para visualizar el entorno de trabajo que cooresponde a la tarjeta antes nombrada, no sin antes haber descargado e instalado el sistema operativo de la tarjeta usada.

III. objetivos

OBJETIVO GENERAL

- Calibrar el sensor MPU 6050, para obtener los resultados de offsets y gráficas para su correspondiente análisis.

OBJETIVOS ESPECÍFICOS

- Generar las gráficas de los dos estados de los sensores *çalibrados"* y *"sin calibrar"*
- Obtener los *.ºffset's"* de cada una de las señales asociadas a los dos sensores implementados, a partir del análisis de datos incluidos en las gráficas.
- Analizar las gráficas de los sensores en estado de calibración y en estado sin calibración.

IV. CALIBRACIÓN

El sensor MPU6050 contiene un giroscopio de tres ejes para la medición de velocidad angular y un acelerómetro para medir las componentes X, Y y Z de la aceleración a través del principio piezo-eléctrico, además de un sensor de temperatura. Dadas las anteriores condiciones se procede a hacer la calibración de los dos sensores nombrados como primera instancia, mediante las siguientes tres fases:

1. ***Programación de Núcleo F411RE*** La programación de la tarjeta núcleo, se realiza por medio de la página web Mbed compiler, al compilar el código esta página nos descarga un archivo de extensión ".bin", archivo que debemos copiar en el almacenamiento de la tarjeta Nucleo,
2. ***Lectura de .ºffset's calculados"*** Para la visualización de los datos se hace uso de un monitor serial, en este caso Coolterm (Figura 1), el cual permite visualizar los datos registrados y enviados por la tarjeta.

Para establecer la comunicación con la tarjeta se accede a la administración de dispositivos del equipo y en ítem “Ports (COM LPT), se observa la tarjeta y el puerto asignado para esta, en este caso el puerto “COM5”.

Figura 1. Asignación de puertos COM

Posterior a esto se debe configurar este mismo puerto en la aplicación serial desde la pestaña connection.

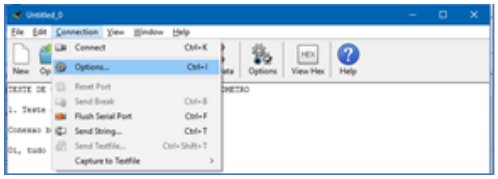
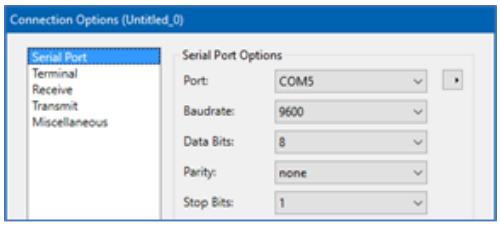


Figura 2. Configuración puerto serial

En la opción “Serial Port” se configura el puerto 5 y se acepta la configuración.



En la parte superior seleccionar la opción “Connect” y presionamos el botón reset de la tarjeta para verificar la comunicación entre los dispositivos.

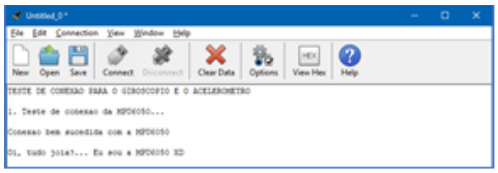


Figura 3. Conexión con tarjeta núcleo

Después de realizar la conexión, para realizar una adecuada lectura de los offsets el sensor se debe fijar de manera horizontal con la electrónica hacia arriba, y evitar moverlo durante la lectura de datos.
Dentro de esta ventana se puede observar los offsets calculados para cada uno de los sensores en cada eje.
Los valores de las variables son los siguientes:
Resultados acelerómetro:

- Offset-accelx = 404.00
- Offset-accely = -280.00

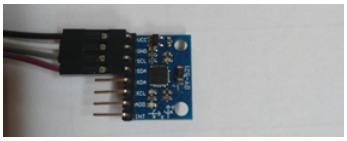


Figura 4. Posición del sensor para cálculo de *.ºffset's"*

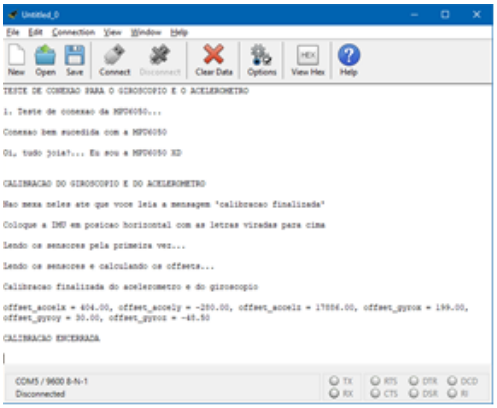


Figura 5. Visualización de datos a tráves de *Çoolterm"*

- Offset-accelz = 17886.00

Resultados giroscopio:

- Offset-gyrox = 199.00
- Offset-gyroy = 30.00
- Offset-gyroz = -48.50

3. **Programación de tarjeta núcleo –F411RE con los offsets calculados.**
   Con los datos obtenidos, se reasignan los valores en las variables del código para la calibración de los sensores.

Figura 6. Asignación de variables.

4. **Lectura de datos a través de Matlab.**

Al ejecutar el código para la lectura y adquisición de los datos, se obtienen las gráficas del sensor antes y después de realizar la calibración.

Los resultados obtenidos son los siguientes

- **Acelerómetro**

En la Figura 7 podemos observar los datos obtenidos del sensor sin calibrar, para el caso de la aceleración en los ejes X y Y, existe una desviación con respecto al valor cero y entre ellos, para el caso de la aceleración en el eje Z, de igual manera existe un desviación con respecto a 1, el cual corresponde a una gravedad.

En la Figura 8, los resultados para los ejes X y Y satisfactorios, el resultado es cero debido a que no está en movimiento, y para el caso del eje Z el valor es 1, de gual manera es un valor satisfactorio.

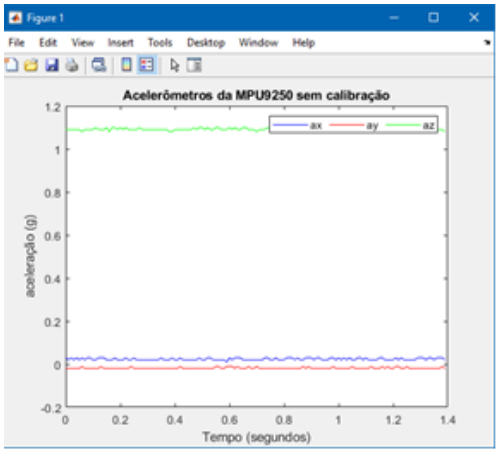


Figura 7. Acelerómetro sin calibración.

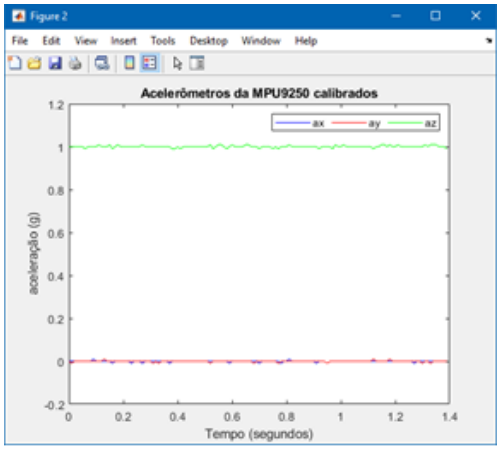


Figura 8. Acelerómetro calibrado.

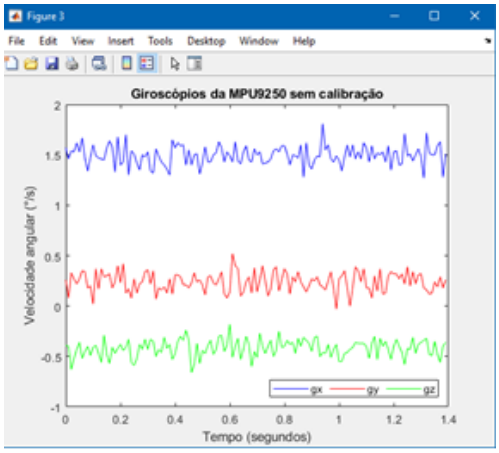


Figura 9. Giroscópio sin calibración.

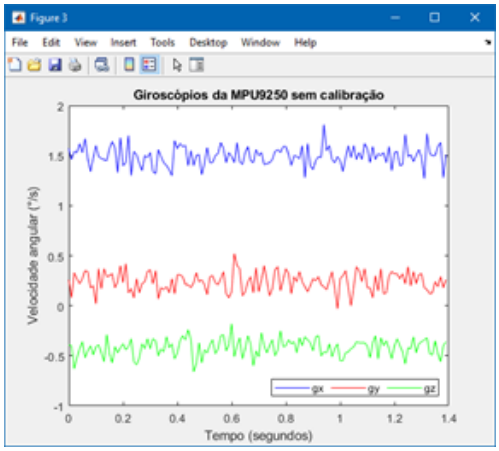


Figura 10. Giroscópio calibrado.

- **Giroscópio**

Teniendo en cuenta que el sensor no está en movimiento, valores obtenidos deberían estar cercanos a cero. Sin embargo los resultados en la Figura 9 demuestran desviaciones para cada uno de los ejes.

Despues de realizar la calibración encontramos que los resultados se acercan más a los ideales, debido a que los resultados son los esperados, aunque se observan variaciones en la señal la escala es muy pequeña, lo que no implica grandes afectaciones en las mediciones.

HERRAMIENTAS: Las herramientas implementadas en la ejecución de esta práctica fueron:

- Estación de trabajo (PC)
- Software Coolterm V1.5.0
- Mbed Compiler (Página Web)
- Script para lectura de datos
- Código de desarrollo para cálculo de offset's
- Script de Matlab

Los materiales necesarios para el desarrollo son:

- 1 sensor IMU 6050
- 1 tarjeta núcleo F411RE
- 4 Jumper de conexión (plug and play)
- 1 Cable T-mini USB

COMUNICACIÓN VNC La comunicación entre la tarjeta Raspberry y la estación de trabajo (computador portátil) se establece físicamente mediante un cable de poder entre la tarjeta y un cargador (conversor AC/DC), un cable para la transmisión de datos tipo mini USB, un cable tipo HDMI conectado a una pantalla auxiliar y una memoria *"micro SD"* de GB (como mínimo); todo lo anterior para permitir hacer el *üpload"* del sistema operativo de la tarjeta y así iniciar a visualizar las funcionalidades y aplicaciones que permite.

Luego de establecer comunicación, haber verificado la correcta instalación del sistema operativo y habiendo establecido previamente una IP para el dispositivo, se procede a visualizar el entorno.

El sistema operativo seleccionado para obtener la funcionalidad completa de la tarjeta es descargado directamente de la página web de Raspberry, esto se realiza por medio de una selección de software que contiene algunos programas preinstalados, lo cual permite facilidad y rapiddez en la instalación del sistema operativo en la tarjeta.

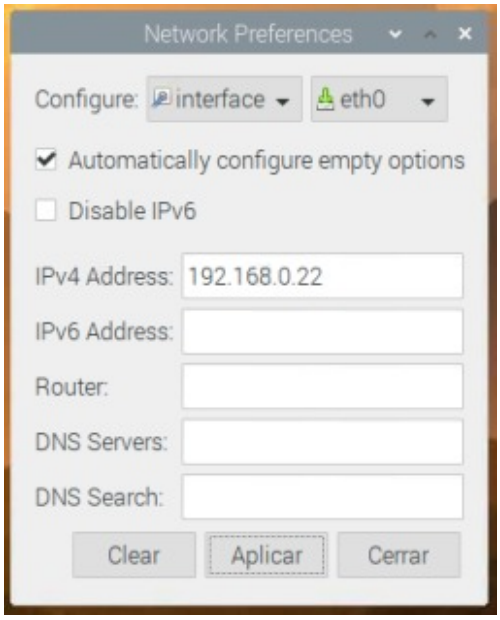


Figura 11. Establecimiento de comunicación entre el PC y la Raspberry Pi 3+.

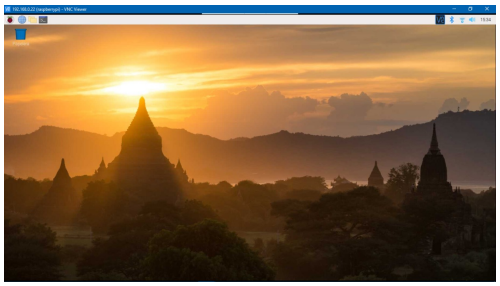


Figura 12. Entorno de trabajo inicial.

Entre las funcionalidades generales que posee la tarjeta se encuentran juegos, acceso de comando (cmd), entorno de programación para lenguaje Phyton, accesorios, gráficos, video, entro otros.

V. Referencias

- Olivares Garcés, D. (2018). Estudio e implementación de algoritmos para la estimación de la posición mediante sistemas inerciales con Arduino (Doctoral dissertation).
- Bourdelande González, A. (2016). Diseño y validación de un sistema para estimación del movimiento durante el entrenamiento de musculación mediante acelerómetros.
- Sheet, D. MPU-6000 and MPU-6050 Register Map and Description.
- Á. Hoyo, J. L. Guzmán, J. C. Moreno y M. Berenguel, «Teaching Control Engineering Concepts using Open Source tools on a Raspberry Pi board,» IFAC-PapersOnLine, vol. 48, nº 29, pp. 99-104, 2015.
- R. I. Pereira, I. M. Dupont, P. C. Carvalho y S. C. Jucá, «IoT embedded linux system based on Raspberry Pi applied to real-time cloud monitoring of a decentralized photovoltaic plant,» Measurement: Journal of the International Measurement Confederation, vol. 114, pp. 286-297, 1 1 2018.
- V. S. Arumuga Perumal, K. Baskaran y S. K. Rai, «Implementation of effective and low-cost Building Monitoring System(BMS) using raspberry PI,» de Energy Procedia, 2017.
- S. W. Foster, M. J. Alirangues, J. A. Naese, E. Constans y J. P. Grinias, «A low-cost, open-source digital stripchart recorder for chromatographic detectors using a Raspberry Pi,» Journal of Chromatography A, 2019.
- S. Ferdoush y X. Li, «Wireless sensor network system design using Raspberry Pi and Arduino for environmental monitoring applications,» de Procedia Computer Science, 2014.
- M. Jayapriya, S. Yadav, A. R. Ram, S. Sathvik, R. R. Lekshmi y S. Kumar, «Implementation of Fuzzy Based Frequency Stabilization Control Strategy in Raspberry Pi for a Wind Powered Microgrid,» de Procedia Computer Science, 2017.

VI. Conclusiones

- Realizar un correcto proceso de calibración del dispositivo, permite obtener datos de mayor confiabilidad, debido a que en este proceso de calibración se busca reducir al máximo los errores de punto cero.
- Durante el proceso adquisición de datos y el cálculo de los offset, es necesario conservar la misma posición del sensor, de manera que se realice un correcto proceso de calibración y se puedan observar los resultados en base a la misma posición.
- Según los resultados obtenidos, la medición realizada con el acelerómetro no presenta ruido elevado en ninguno de los ejes, por lo que no es del todo necesario la implementación de un filtro para estas mediciones.
- La señal obtenida del giroscopio, a diferencia del acelerómetro, presenta mayor ruido y aunque este muy cercana a cero, se debe implementar un filtro con el fin de tener mayor linealidad en la medición.