

Comunicación entre RB3+, IMU y STM32"

1st Esteban Olmos
, 2st Vanesa Lara, 3st Mario Galeano
Ingeniería Mecatrónica
Universidad ECCI
Bogotá, Colombia

johne.olmosc@ecc.edu.co, vanesay.larac@ecc.edu.co, marioe.galeanor@ecc.edu.co

I. RESUMEN

Resumen: En este documento se encuentra concentrada la información acerca del proceso, requerimientos entre otros aspectos que se deben tener para establecer comunicación entre la tarjeta de adquisición de datos "Núcleo STM 32", el sensor "MPU5060" y la tarjeta de procesamiento "Raspberry 3B+"

El establecimiento de la comunicación tiene condicionales para poder obtener la conectividad de acuerdo al orden requerido; por esta razón se establece una distribución de tareas o subrutinas que cada una de las tarjetas debe cumplir, ya que, al momento de realizar la integración, se pueda obtener la lectura del sensor y a partir de esto las demás tarjetas realicen su funcionalidad.

Uno de los principales propósitos que surgieron para establecer la conectividad es distribuir los procesos que manejaría la tarjeta Raspberry, puesto que es mucho más eficiente realizar la adquisición de datos por medio de un sistema embebido, esto con el fin de liberar procesamiento en la tarjeta de implementación principal (RB 3B+)

Palabras claves—Conexión VNC, procesamiento, gráficas, giroscopio, acelerómetro.

II. INTRODUCCIÓN

La comunicación es uno de los aspectos más importantes en cualquier aspecto; sin embargo, acá se remite netamente a las comunicaciones entre dispositivos y es cuando se inicia a hablar de protocolos de comunicación, medio de comunicación, entre otros aspectos que se pueden considerar antes de establecer un canal de comunicaciones entre 2 o más dispositivos como lo es este caso del que se hará una profunda explicación.

Establecer la comunicación entre las tarjetas antes mencionadas es en otras palabras unificar un lenguaje para lograr una sincronía en la transmisión y recepción de datos, de acuerdo a la secuencia lógica y funcional que se determine en el sistema, teniendo como base principal en la implementación del sistema: adquisición, procesamiento, y visualización. Estos tres procesos se distribuyen en cada una de las tarjetas, esto con el fin de optimizar los procesos cargados en la tarjeta principal (Raspberry 3B+)

La distribución del funcionamiento tiene como elemento principal la tarjeta Raspberry, esta realiza las funciones de procesamiento y visualización, como adquisición de datos la

tarjeta STM32, el sensor realiza la toma de datos, de esta manera es posible establecer la comunicación completa del sistema.

III. OBJETIVOS

OBJETIVO GENERAL

- Establecer comunicación directa entre las tarjetas: sensor MPU5060, Núcleo STM32 y Raspberry 3B

OBJETIVOS ESPECÍFICOS

- Realizar la captura de datos en rangos de tiempos específicos con el fin de obtener una análisis de la calibración del sensor
- Graficar los la señal de los sensores: giroscopio y acelerómetro, obteniendo datos en estado: calibrado y sin calibración.

IV. COMUNICACIÓN

La comunicación de las tarjetas se establece una vez se haya instalado correctamente el sistema operativo de la Raspberry 3B+ y se haya instalado la aplicación "**VNC Viewer**", luego de esto existen dos formas para establecer la comunicación. Sin embargo antes de ejecutar cualquiera de las dos formas, es necesario cumplir con los siguientes requisitos:

- Estación de trabajo (PC)
- Software Coolterm V1.5.0
- Mbed Compiler (Página Web)
- Script para lectura de datos
- VNC Viewer

En términos generales 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".

1. **Vía Ethernet** El establecimiento de la comunicación por medio de vía ethernet se re

2. **Vía Wi-Fi**

En las Figura 1 y 2 se muestra la gráfica del sensor acelerómetro y giroscopio, respectivamente sin calibrar; por lo que es evidente las variaciones existentes en los tres planos (X, Y, Z), adicional a esto es posible visualizar el comportamiento de aceleración y la velocidad angular aplicando el filtro complementario.

En las Figuras 3 y 4, los sensores están calibrados correctamente y listos para realizar pruebas de funcionamiento de acuerdo a la aplicación implementada.

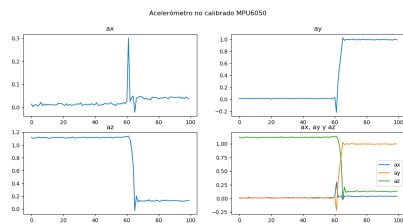


Figura 1. Acelerómetro sin calibración

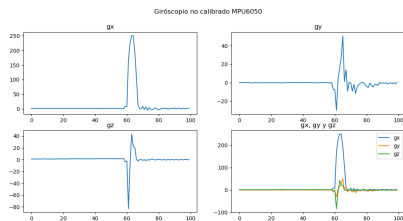


Figura 2. Giroscopio sin calibración

En las Figuras 5 y 6 se muestra el comportamiento de los ejes *Roll* y *Pitch*, con la respectiva calibración, la aplicación del filtro complementario y una gráfica adicional en la que se contrastan los resultados anteriormente nombrados.

Los valores de las variables son los siguientes:

Resultados acelerómetro:

- Offset-accelx = 404.00
- Offset-accey = -280.00
- Offset-accelz = 17886.00

Resultados giroscopio:

- Offset-gyrox = 199.00
- Offset-gyroy = 30.00

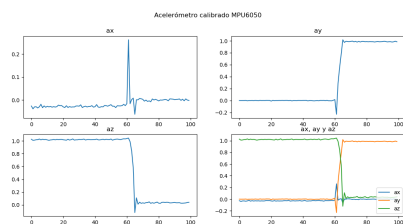


Figura 3. Acelerómetro con calibración

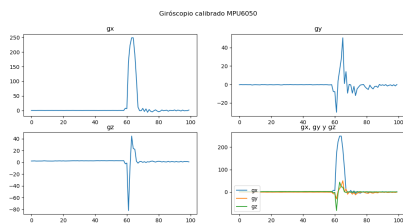


Figura 4. Giroscopio con calibración

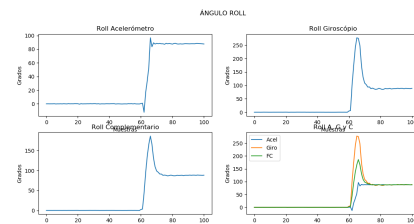


Figura 5. Movimientos en Roll

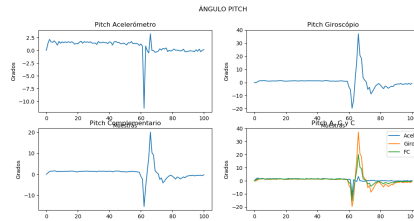


Figura 6. Movimientos en Pitch

- 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.

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 una 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.

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.

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 (convertor 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 "upload" 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.

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, n° 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.
- 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.
- En términos generales la programación por medio del lenguaje de desarrollo "**Python**" permite fácilmente la optimización funcional del código, ya que los procesos no se ejecutan directamente en una tarjeta, sino que están distribuidos estratégicamente.

VI. CONCLUSIONES

- Al determinar el protocolo de comunicación compatible para las tarjetas, es posible establecer canales de comu-