

Comunicación IMU,STM32 Y RASPBERRY

Universidad ECCI

Luis Felipe Bravo 50211

Luis Vicente 68218

I. INTRODUCTION

En el siguiente trabajo se evidencia la implementación de dos controladores que permiten observar los datos generados por un Acelerómetro- Giroscopio; así como la formación de graficas degeneradas por el mismo sensor e implementación de ángulos de navegación (Roll y Pitch) de tal forma que permita analizar las variaciones que se pueden dar en un sistema que contenga información de tipo angular.

2. RESUMEN

En este trabajo se observara gráficamente los datos generados por un sensor Acelerómetro-giroscopio (MPU 6050) ; el cual estará anexo a un controlador STM32 el cual se encargara de la lectura de datos dados por el sensor. Así mismo se establece la implementación de un controlador Raspberry pi el cual de la calibración y segmentación de datos esto con el fin de generar graficas correspondientes a datos calibrados, no calibrados y ángulos de navegación (Ángulos de Euler) como lo son Roll y Pitch.

3. DESARROLLO CONCEPTUAL:

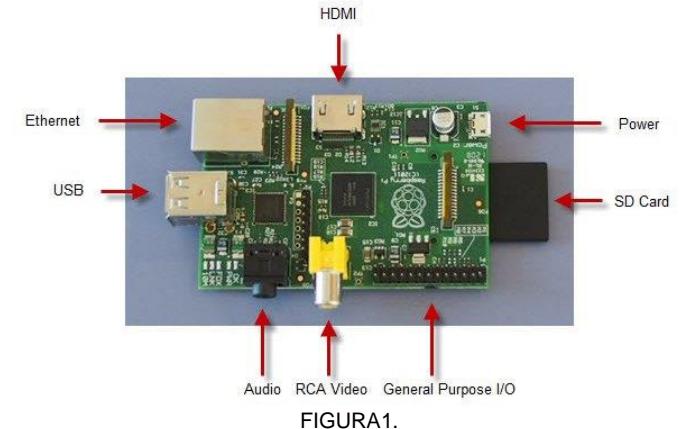
Raspberry pi:

Raspberry Pi, es un «es un ordenador de tamaño de tarjeta de crédito que se conecta a su televisor y un teclado». Es una placa que soporta varios componentes necesarios en un ordenador común.«Es un pequeño ordenador capaz, que puede ser utilizado por muchas de

las cosas que su PC de escritorio hace, como hojas de cálculo, procesadores de texto y juegos. También reproduce vídeo de alta definición», apuntan en la página web del producto.

Este proyecto fue ideado en 2006 pero no fue lanzado al mercado febrero de 2012. Ha sido desarrollado por un grupo de la Universidad de Cambridge y su misión es fomentar la enseñanza de las ciencias de la computación los niños. De hecho, en enero de este año Google donó más de 15.000 Raspberry Pi para colegios en Reino Unido.

La placa, que antes era más pequeña que una tarjeta de crédito tiene varios puertos y entradas, dos USD, uno de Ethernet y salida HDMI. Estos puertos permiten conectar el miniordenador a otros dispositivos, teclados, ratones y pantallas.



4. DESARROLLO TEORICO

Inicialmente debemos tomar en cuenta que la STM32 necesita capturar o leer datos, por ende debemos programarla para que reciba esos datos, esta programación se generara por medio de Mbed (plataforma añadida a este controlador). Esta permitirá recoger los datos generados por la IMU, esto debido a que se empleara la Stm32 como medio interlocutor entre la Información generada por la IMU y la Raspberry en esta última se hará la calibración y las gráficas correspondientes.

Después de ello se conecta la STM32 a Raspberry, para que se genere la calibración pertinente, así mismo se anexara la programación que referencia a la generación de los osffet estos tendrán la función de tomar un punto neutro en base a la orientación de la IMU, este osfet se tomara como el origen y de allí se presenta la calibración.

Después de haber capturado (Recolección de datos) el origen, se dispondrá a realizar la calibración esta se dará con respecto al giroscopio y al acelerómetro, estos deberán ser procesados como calibrados y no calibrados, esto con el fin de lograr comparar los datos referenciados por la IMU y la calibración pertinente.

Asi mismo se implementara un filtro complementario con el fin de hallar los datos de Roll y Pitch.

Tomando en cuenta que se esta trabajando con un giroscopio y acelerómetro MPU 6050 se emplean ángulos de navegación como lo son Roll y Pitch ; en este caso no se tomara en implementar yaw.

Para poder asimilar aplicaciones referentes a el uso de este tipo de ángulos como lo son en embarcaciones y muy especialmente en Aeronaves.

4.1 Procedimiento:

Primeramente debemos tener el cuenta el proceso que se efectuó para poder obtener las gráficas mencionadas:

1. Se debe configurar la stm32 para que lea los datos generados por la IMU, para ello se tomara en cuenta la ayuda proporcionada por MBED(Software de programación) esto con el fin de captar o generar una lectura de los datos.
2. Se configura la raspberry para que pueda recibir los datos suministrados por la STM32 que va a realizar las veces de intermediario entre la IMU y la Raspberry.
3. La configuración se realiza desde TERMINAL de el S.O de Raspberry, en donde se agregara el comando ls /dev/ACM* este me permitirá saber en que puerto se encuentra conectada la STM32.
4. Registrar el puerto en el código con el fin de generar una trasmisión continua de lo contrario no se efectuara de forma correcta.
5. Ejecutar el código de calibración.
6. Ejecutar el código de graficas el cual nos permitirá observar las distintas variaciones que tendrá la IMU y su posición de calibración.

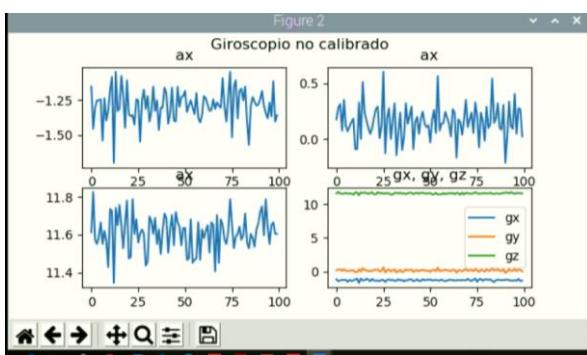
4.2 CONSIDERACIONES:

- El sistema operativo que maneja Raspberry trabaja en torno a una derivación de Linux.

5. ANALISIS DE RESULTADOS

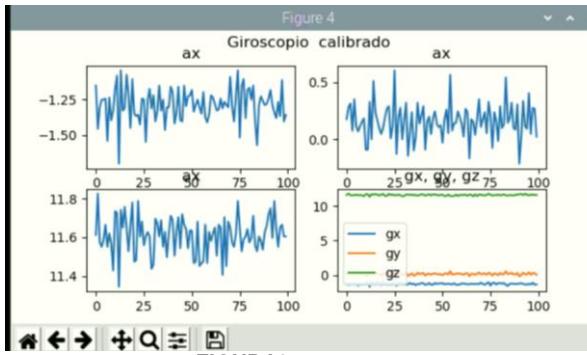
Lo anterior implica el análisis de las gráficas obtenidas en las cuales se tendrán en cuenta, datos calibrados, datos sin calibrar t a raíz de los datos calibrados las gráficas de Roll y Pitch.

1. Giroscopio no calibrado



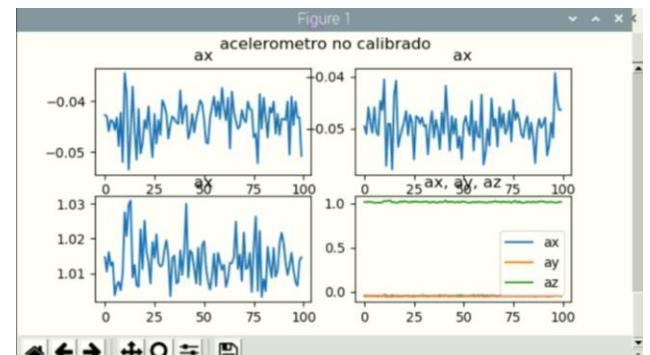
En la imagen_ tenemos la información neta generada por el sensor, esto quiere decir que el único filtro que ha tenido esta información fue el programa de lectura. En estas 4 graficas tenemos los componentes en x, y y z con respecto al giroscopio y una conjugación de las graficas descritas anteriormente.

2. Giroscopio calibrado



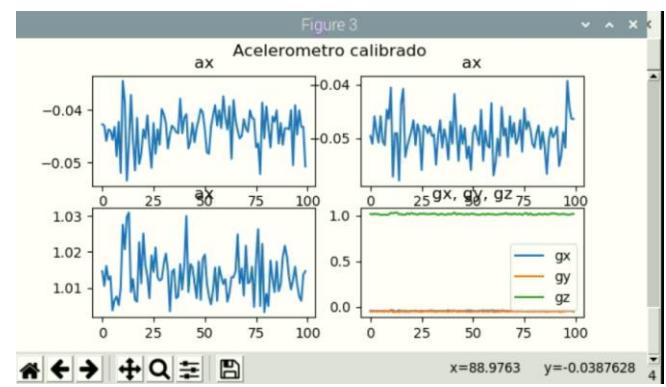
En la anterior grafica podemos observar las distintas variaciones que se han generado a partir la calibración. Que como se puede observar no es muy distante a lo que se tenia inicialmente, tomando en cuenta a la posición del sensor al inicio de la medición se puede decir que esta ubicación hace que no hallan variaciones evidentes.

3. Acelerometro no calibrado

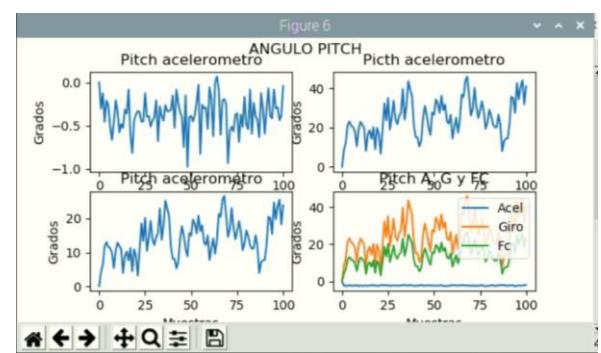


En la anterior grafica podemos observar las variaciones que tiene el sensor sin ningún filtro que limipe la señal.

4. Acelerómetro calibrado



5. Angulo pitch



En la anterior grafica tenemos la estimación de angulos con respecto a la

información recibida y mencionada anteriormente, la cual esta dada en términos de x, y, z, a estas graficas dadas por el acelerómetro , y el giroscopio pasan por un filtro complementario que tiene un función específica la cual es ajustar la estimación de los angulos.

6. Angulo Roll

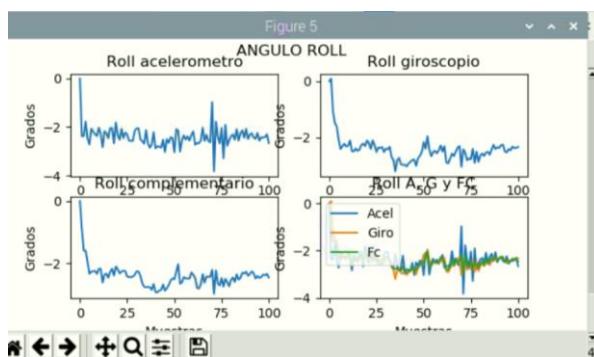


FIGURA.7

En la anterior grafica se pueden observar los datos generados a partir de una configuración matemática que se le aplica a la aceleración y el sentido de giro, además de esos datos generados se les aplica una conversión de radianes a grados ya que esta función hace referencia a los ángulos de Euler los cuales se trabajan en grados.

6. CONCLUSIONES:

1. Los ángulos Roll y Pitch son ángulos de navegación que permiten referenciar; estos permiten referencias ciertos puntos en el plano cartesiano en donde se toma como origen un punto distinto al original.
2. La calibración se debe a los offset generados por la primera captura de datos, que no es la misma captura dada por el código de calibración, esta captura se tomara con la IMU de forma horizontal en un punto fijo , esto con el fin de generar un origen(offset)

3. Los datos generados por el acelerómetro al ser calibrados no cambian demasiado con respecto a los datos no calibrados esto debido a la propiedad física del mismo.
4. La placa Raspberry tiene muchas características que lo hacen un controlador ideal, se puede tomar como un computador, tiene en sí mismo un propio sistema operativo, y así mismo se pueden agregar periféricos para su control.
5. Phyton tiene muchas similitudes con distintos lenguajes de programación, como por ejemplo **C**, así mismo es muy intuitivo, es de suma importancia considerar la indexación ya que esta reemplaza el uso de signos de agrupación .
6. Raspberry toma a la placa STM32 como un periférico similar a un teclado, y haciendo esta similitud, la información generada por las teclas se tomarían como la información generada la MPU6050

7. BIBLIOGRAFÍA

- ABC.ES. (s.f.). Recuperado el 20 de 05 de 2020, de ABC.ES: <https://www.abc.es/tecnologia/informatica-hardware/20130716/abci-raspberry-como-201307151936.html?ref=https%3A%2F%2Fwww.google.com%2F>
- Alvarado, M. S. (11 de Diciembre de 2011). Modelo matemático de un motor de corriente. *Instituto de Ciencias Físicas, Escuela Superior Politécnica del Litoral*, pág. 7.
- Alvarez, M. A. (12 de 11 de 2003). <https://desarrolloweb.com/>. Recuperado el 27 de 05 de 2020, de <https://desarrolloweb.com/articulos/1325.php>; <https://desarrolloweb.com/articulos/1325.php>
- César Augusto Peña Cortés, a. f. (2013). Inspection and monitoring system using an aerial robot guided by artificial vision. <https://www.researchgate.net/>, 10.
- SISTEMS, M. (3 de 3 de 2016). <https://la.mathworks.com/products/matlab.html>. Obtenido de <https://la.mathworks.com/products/matlab.html>; <https://la.mathworks.com/products/matlab.html>

