

# INFORME PRACTICA 2 CORTE

1<sup>st</sup> Johan Felipe Parraga Urian  
*Ingeniería Mecatrónica*  
*Universidad ECCI*  
Bogotá, Colombia  
johanf.parragau@ecci.edu.co

2<sup>nd</sup> Diego Sammir Moreno Sanchez  
*Ingeniería Mecatrónica*  
*Universidad ECCI*  
Bogotá, Colombia  
diegos.morenos@ecci.edu.co

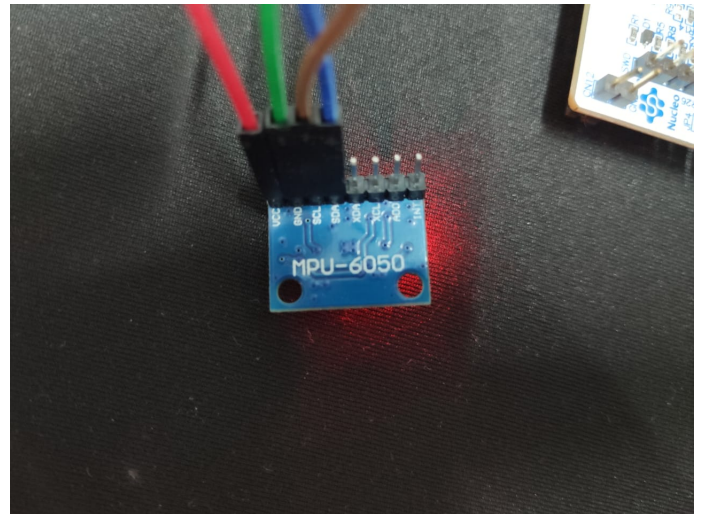
**Resumen—** El presente informe ilustra la practica de usar una tarjeta STM para adquirir los datos de un sensor MPU6050 y enviarlos para su procesamiento a MatLab, tambien se encontrara de manera practica la forma de conectar una Raspberry Pi a una computadora de manera remota por media de una conectividad alámbrica o inalámbrica.

**Palabras claves—**Sistema enbebido, MEMS, Comunicacion I2C, Registro de datos, Raspberry pi, conectividad.

1) **MPU6050:** EL MPU6050 es una unidad de medición inercial o IMU (Inertial Measurment Units) de 6 grados de libertad (DoF) pues combina un acelerómetro de 3 ejes y un giroscopio de 3 ejes. Este sensor es muy utilizado en navegación, goniometría, estabilización, etc.

## I. INTRODUCCIÓN

Los Sistemas Micro-Electro-Mecánicos (MEMS) usualmente son utilizados en el procesamiento de datos, comunicación y medición de variables físicas, entre otros. Implementados en aplicaciones como videojuegos, sistemas de seguridad vehicular, dispositivos móviles y demás. Dentro de los sistemas MEMS se encuentran las IMUs, Dispositivos electrónicos utilizados como unidades de medición inercial detectando la fuerza gravitacional, velocidad angular y en algunos casos campo magnético. El objetivo de esta practica es realizar una adquisición de datos y una calibración de los datos adquiridos por la tarjeta STM a través de Matlab, vamos a poder visualizar los datos recibidos en Matlab y realizar la respectiva calibración visualizando los datos recibidos y los datos calibrados a través de diferentes gráficas obtenidas por medio de Matlab



[?]

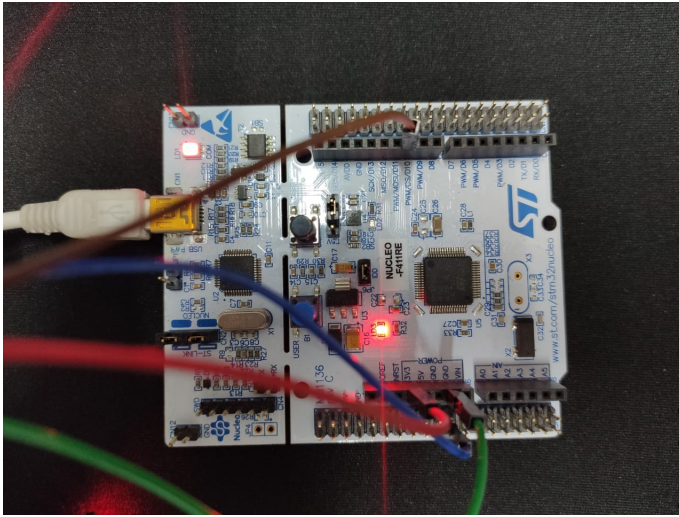
## II. MARCO TEÓRICO

Como medio de adquisición de datos se uso la tarjeta STM NUCLEO411R la cual se encargo realizar la tarea de adquisicion de datos enviados por la MPU a través de comunicacion i2c y enviar estos datos por media de una comunicacion Asíncrona por puerto serial a la computadora, estos datos fueron recibidos en la computadora por la aplicación Matlab la cual fue la encargada de realizar el procesamiento de los datos con el cual se grafican los datos y se puede observar un comportamiento frente a los datos después de ser calibrados correctamente.

### A. Elementos Utilizados

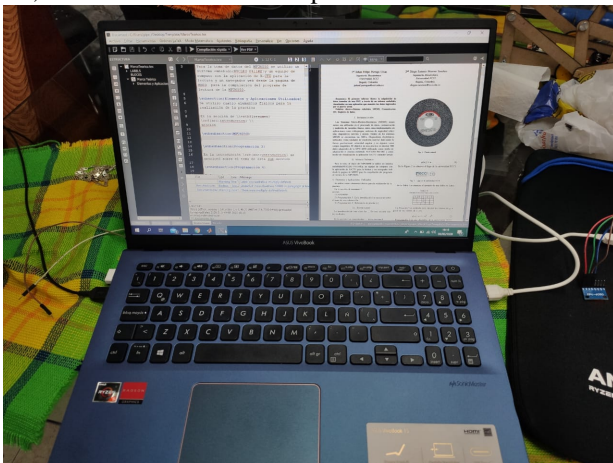
Se utilizo cuatro elementos físicos para la realización de la practica

2) **NUCLEO411RE:** La tarjeta de desarrollo STM32 provee una solución accesible y flexible para el desarrollo de prototipos y nuevas ideas bajo una plataforma ARM Cortex-M4 de ST y la plataforma mbed ARM. La tarjeta se diseño bajo características tanto en el microcontrolador como en sus componentes de un sistema balanceado en potencia de procesamiento con un consumo de energía limitado. La tarjeta es compatible en pines con un Arduino y con ST Morpho.



[?]

3) *Computadora Portátil:* computadora portátil de peso y tamaño ligero, su tamaño es aproximado al de un portafolio (hay más pequeñas como Palmtop y Handheld). Ésta pertenece al grupo de las computadoras personales, las cuales son sistemas de computación relativamente pequeños y de bajo costo, también llamados microprocesadores.



[?]

4) *Programación utilizada:* La programación utilizada fue brindada por el docente de la clase Prof. Msc. Fabián Barrera Prieto. Primero utilizamos un programa de calibración mediante la cual adquirimos dos muestras de 100 datos de las cuales la primera muestra es descartada como dato estadístico, de la segunda muestra de datos son tomados para análisis y obtener los datos de interés para la práctica los cuales son los offset en cada uno de los ejes de los dos sensores, estos datos son obtenidos a través de condicionales aplicadas a los puntos máximos y mínimos de los valores de cada uno de los ejes, estos valores son promediados para obtener un offset por cada eje. A continuación podemos encontrar el link al código utilizado para la calibración :

<https://drive.google.com/file/d/1Zq4SNt6mHB1RLBcFIvcTe6h7G8vUV/view>

Para la parte de lectura de datos utilizamos un código en el cual se realizaba la lectura del puerto i2c de la tarjeta

NUCLEO, esta programación realiza la lectura cuando a través del puerto serial recibe un carácter 'H', posteriormente escribimos al registro 3B de la IMU, después realizamos la lectura de la IMU a través de i2c de 14 Bytes y almacenando los datos recibidos en un vector de 14 posiciones, luego de recolectados estos datos son enviados a través del puerto serial a la computadora.

A continuación podemos encontrar el link al código utilizado para la calibración :

<https://drive.google.com/file/d/1LdIBcjZGbrdfhYsP65o36baeUxm4zoM/view>

Para el procesamiento de datos utilizamos Matlab mediante este procesamiento realizamos una lectura de los datos enviados por la STM411R, estos datos son calibrados y mostrados a través de gráficas de los valores de los ejes sin calibrar y calibrados

A continuación podemos encontrar la programación utilizada en Matlab:

```
close all;
clear all;
clc;
oldobj = instrfind;
if not(isempty(oldobj))
fclose(oldobj);
delete (oldobj);
end
if exist('s','var')
s = serial('COM5','BaudRate',9600,'DataBits',8,'Parity','None','StopBits')
end
if isvalid(s)
s = serial('COM5','BaudRate',9600,'DataBits',8,'Parity','None','StopBits')
end
if strcmp(get(s,'status'),'closed')
fopen(s);
end
```

```
SENSITIVITYACCEL = 2.0/32768.0;
SENSITIVITYGYRO = 250.0/32768.0;
offsetacclx = 120.00;
offsetacclz = 16932.00;
offsetgyrox = 4053.00;
offsetgyroy = 111.00;
offsetgyroz = -124.00;
```

```
disp('En sus marcas. Posicione el sensor en la posición inicial')
pause();
```

```
disp('comienza')
fprintf(s, 'H');
i = 1;
while(1)
stri = fscanf(s);
if(stri(1) == 'A')
```

```

disp('termina')
break;
end
i = i + 1;
end

fclose(s);
n = length(str)-1;
c = 1;

for i=1:n
temp = cellfun(@str2num, strsplit(str, ','));
if numel(temp) == 8
values(i,:) = temp;
end
end

save DatosTest1 values

Nsamples = length(values);
dt = 0.01;
t = 0:dt:Nsamples*dt-dt;

```

```

figure;
plot(t, values(:,3)*SENSITIVITY_ACCCEL,'b')
hold on
plot(t, values(:,4) * SENSITIVITY_ACCCEL,'r')
plot(t, values(:,5) * SENSITIVITY_ACCCEL,'g')
title('Acelerometrosincalibrar')
xlabel('Tiempo(segundos)')
ylabel('aceleracion(g)')
legend('ax','ay','az','Location','northeast','orientation','horizontal')

```

```

figure;
plot(t, (values(:,3)-offset_accelx) * SENSITIVITY_ACCCEL,'b')
hold on
plot(t, (values(:,4) - offset_accely) * SENSITIVITY_ACCCEL,'r')
plot(t, (values(:,5) - (offset_accelz - (32768/2))) * SENSITIVITY_ACCCEL,'g')
title('Acelerometrocalibrado')
xlabel('Tiempo(segundos)')
ylabel('aceleracion(g)')
legend('ax','ay','az','Location','northeast','orientation','horizontal')

```

```

figure;
plot(t, values(:,6)*SENSITIVITY_GYRO,'b')
hold on
plot(t, values(:,7) * SENSITIVITY_GYRO,'r')
plot(t, values(:,8) * SENSITIVITY_GYRO,'g')
title('Giroscopiosincalibrar')
xlabel('Tiempo(segundos)')
ylabel('velocidadangular')
legend('gx','gy','gz','Location','northeast','orientation','horizontal')

```

```

figure;
plot(t, (values(:,6)-offset_gyro_x) * SENSITIVITY_GYRO,'b')
hold on
plot(t, (values(:,7) - offset_gyro_y) * SENSITIVITY_GYRO,'r')
plot(t, (values(:,8) - offset_gyro_z) * SENSITIVITY_GYRO,'g')
title('Giroscopio calibrado')
xlabel('Tiempo(segundos)')
ylabel('velocidad angular')
legend('gx','gy','gz','Location','northeast','orientation','horizontal')

```

### III. RESULTADOS

#### IV. ANÁLISIS

A partir del muestreo de datos en los resultados se puede notar una variación (aun cuando este permanecía estático). En una superficie plana los datos adquiridos deberán dar lectura de una gravedad (1G), de lo contrario se tomaba lectura de valores por debajo o por encima, lo que puede indicar que el sensor se encuentra des calibrado.

#### V. CONCLUSIONES

1) : La elaboración de esta practica ha permitido conocer la forma en que se leen los datos generados por la MPU6050 respecto a los ángulos de euler.

2) : La adquisición de datos en estado estable indica si la MPU6050 esta calibrada correctamente, por ello el tiempo de muestreo es importante porque al promediar el valor de cada angulo se puede tener un acercamiento al valor real medido.

3) : Este documento tiene como fin conocer y desarrollar habilidades en el manejo del formato de texto latex.