陽明交大
NYCU

物聯網裝置與平台
IoT Devices and Platforms

曾煜棋、吳昆儒

National Yang Ming Chiao Tung University

陽明交大
NYCU

	日期	主題
1	9/17	(加退選9/13-27) 課程介紹, arduino簡介
2	9/24	物聯網裝置: Arduino basic introduction
3	10/1	物聯網裝置: Arduino Digital Interface
4	10/8	物聯網裝置: Arduino Analog Interface
5	10/15	sensor介紹 part 1
6	10/22	sensor介紹 part 2
7	10/29	sensor介紹 part 3
8	11/5	(期中考周11/1-5) sensor介紹 part 4
9	11/12	期中考
10	11/19	通訊模組 Bluetooth, Lora
11	11/26	通訊模組-wifi
12	12/3	Proposal
13	12/10	物聯網平台 - IoT Cloud Platform
14	12/17	AI應用 (SVM)
15	12/24	(期末考周 12/24-30) Project 準備周
16	12/31	(國定假日)
17	1/7	(彈性補充教學) Final demo
18	1/14	(彈性補充教學) Final demo part 2 (如果需要兩周進行)

Important date

- **Find your team member (each group: 2~4 persons)**
  - Total: 15 groups
- (12/3) Project proposal
  - Prepare slide (2 pages are enough) with 5 min introduction
    - P0. Project title
    - P1. Your idea/motivation
    - P2. What do you need (ex: sensors)
  - We will discuss and provide suggestions to each team
- (1/7, 1/14) Final project demo (via Teams)
  - Prepare both slide and live demo
  - Upload slide and demo video to e3
  - **Each team has 10 minute, so we might only need one week.**

11/12 - Midterm

- 筆試, 不用上機現場寫code
  - 題目會包含debug code, 可自行選擇帶零件過來

- Open book, open Internet
  - 可上網查詢資料, 也可看課程的相關講義檔案
  - 可以使用桌機, 筆電, 平板
  - 請勿使用手機, 禁止通訊軟體, 協作文件的comment…等
  - 所有可以交談的方式皆為禁止項目

- 考試時間12:20 – 15:10, 地點EC315, EC316
  - 座位之間須留空位
- 考試範圍: 上課提過的東西

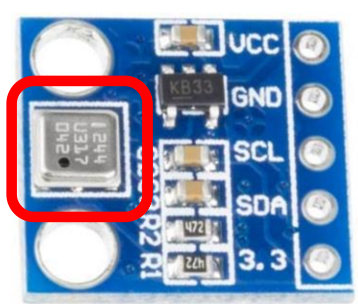
Ch 5, Sensors (4) - Summary

- Arduino IDE and how it interacts with the external world
  - Sensors
- Understanding this course:
  - Discussion: upload to e3
  - Quiz: show your code to TA
    - For remote-access, use discord to interact with TA

Last week

1. **Sweep:** Sweeps the shaft of a RC servo motor (Radio Control or Remote control) back and forth across 180 degrees.
2. **Knob:** Control the position of a RC servo motor with Arduino and a potentiometer.
3. **Altimeter:** Use BMP180 to read the pressure and temperature, and then calculate the altitude.
4. **Pulse sensor:** Optical Sensors for Heart Rate Monitor

SG-90
servo motor



Pressure (altimeter)

Pulse Sensor

This week

Use I2C-bus or a single data line to read the sensor values.

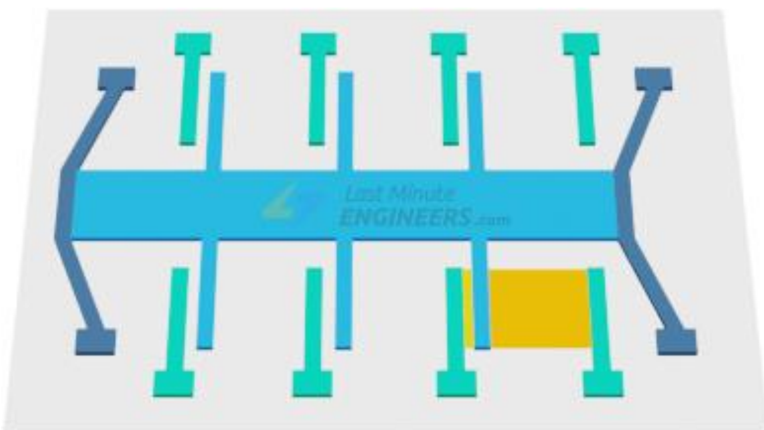
- IMU: Inertial measurement unit (6 DoF, degree of freedom)
- Accelerometer (3-axis): 加速度
- Gyroscope (3-axis): 角速度

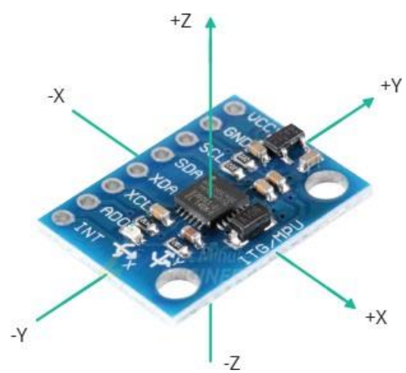
- MPU-6050 = Accelerometer + Gyroscope
  http://playground.arduino.cc/Main/MPU-6050

Accelerometer + gyroscope

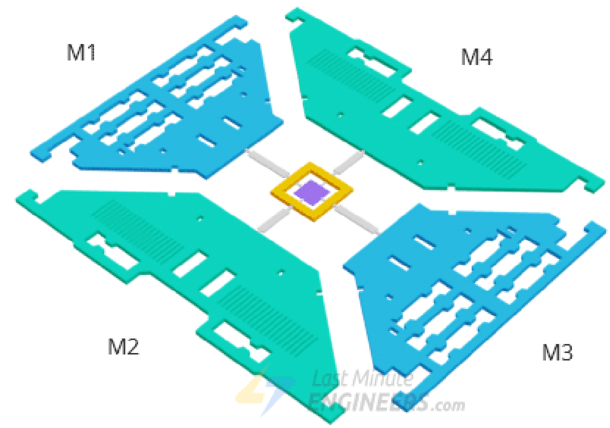
How does it work?

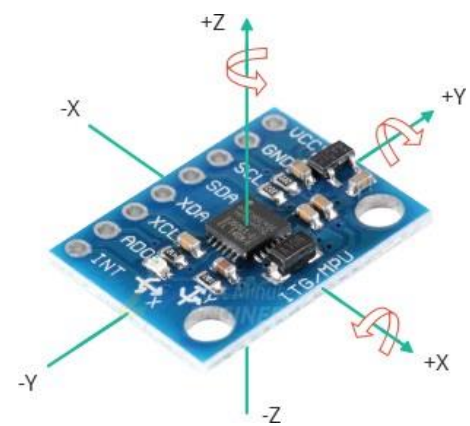
Accelerometer





Gyroscope



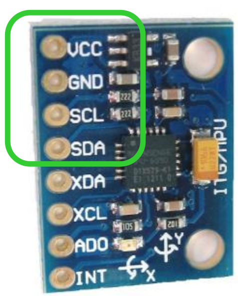


IMU applications

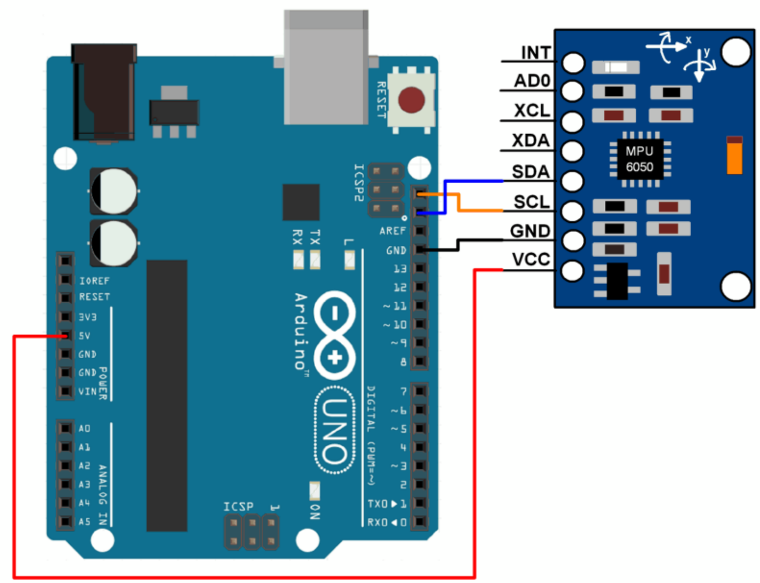
- Open Source IMU and AHRS Algorithm with x-IMU
  - https://www.youtube.com/watch?v=BXsGWoOMtmU
- 3D Box on screen rotated around its roll and pitch axes through an acceleration sensor
  - https://www.youtube.com/watch?v=85rl3KVAEGU
- Joycon 6-Axis sensor information
  - https://github.com/dekuNukem/Nintendo_Switch_Reverse_Engineering
- Nintendo Switch《健身環大冒險》CM 冒險篇1 (台灣)
  - https://www.youtube.com/watch?v=0Dze0TMI0dc
- The Legend of Zelda: Skyward Sword HD
  - https://youtu.be/X27t1VEU4d0?t=81
- Fitness Boxing 2
  - https://www.youtube.com/watch?v=GbmoH7sfoE0

Lab0. MPU-6050

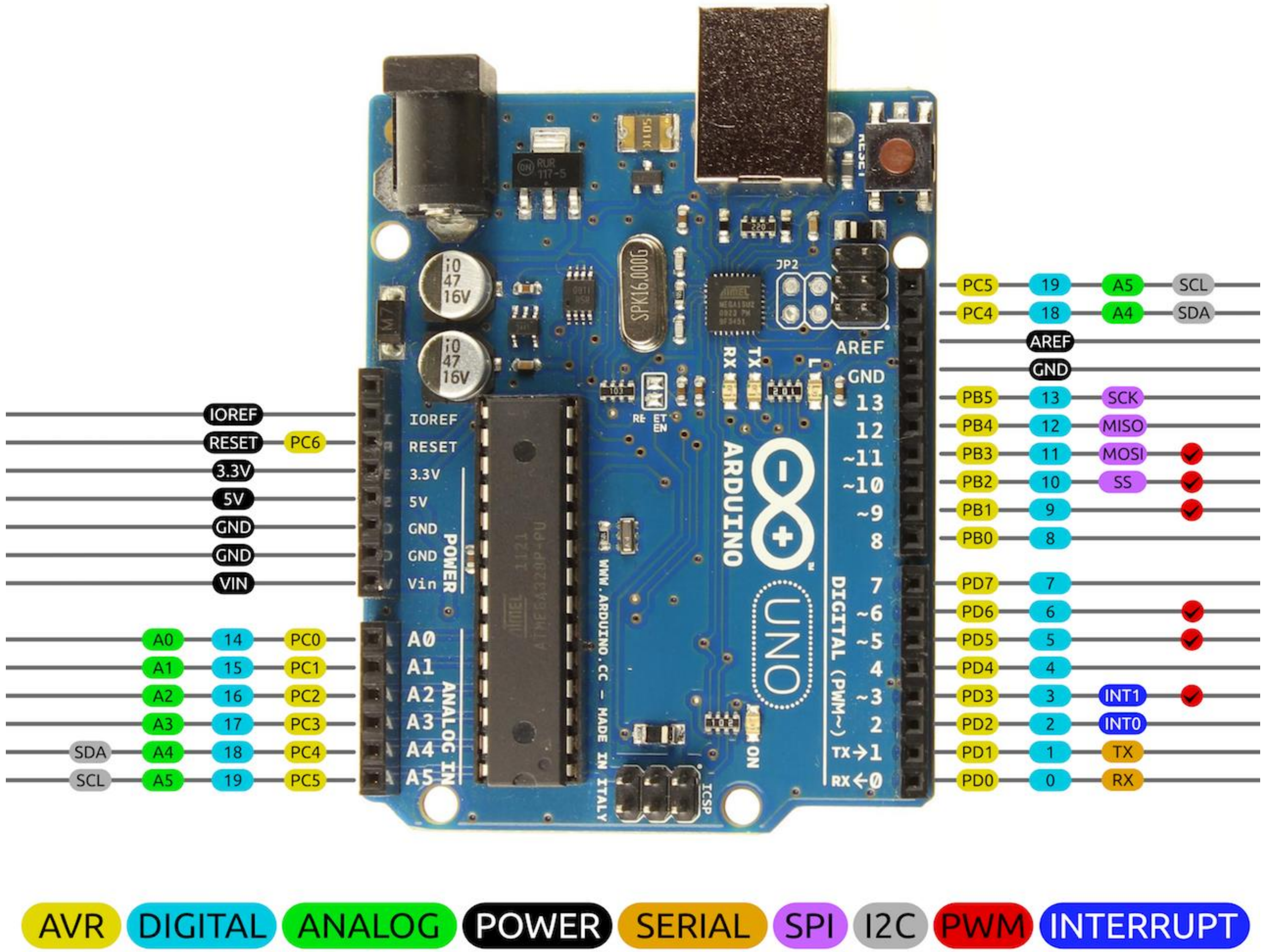
- Goal: Use GY-521 sensor board with MPU-6050 (chipset) in a single chip to read the sensor values for the accelerometer and gyro.
- Hardware Required
  - Arduino Board
  - MPU-6050



SCL - Serial clock line
SDA - Serial data line

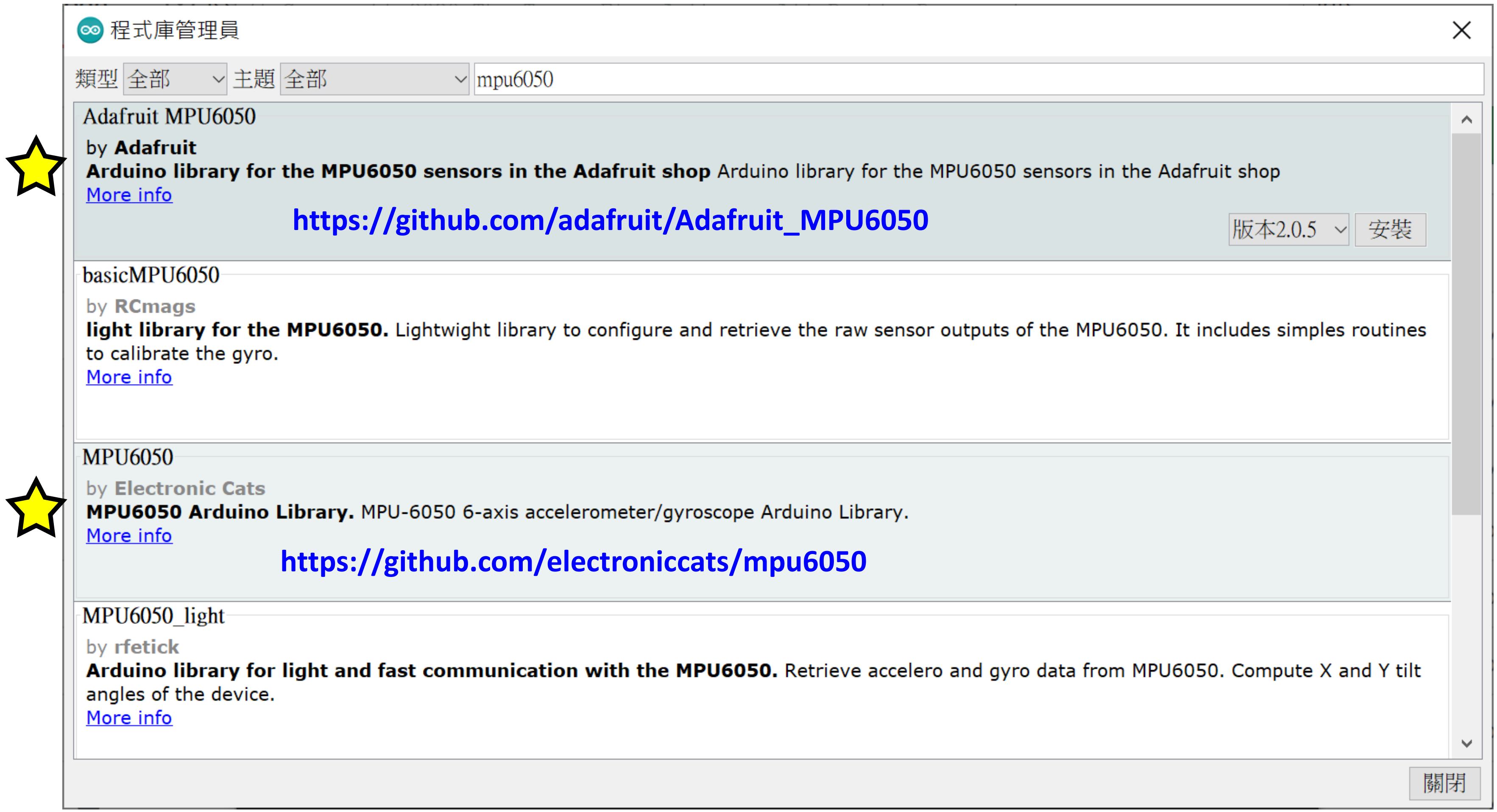


Arduino Uno R3 Pinout



When using SDA and SCL, **DO NOT** use analog **A4** and **A5** at the same time!!!

Library manager



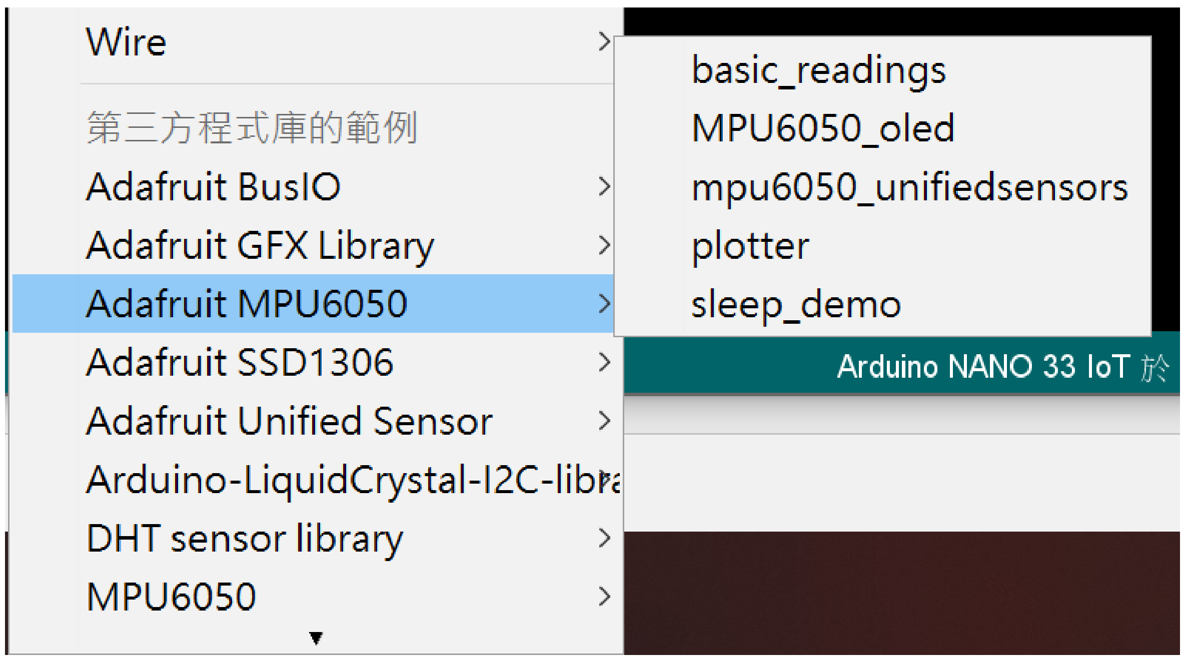
Lab1. MPU6050

Accelerometer + Gyroscope

Lab1. MPU6050 (acc+gyro)

Arduino IDE

Open--->File--->Examples---> 第三方程式庫 --> Adafruit MPU6050



Sample code: basic_reading

```
// Basic demo for accelerometer readings from Adafruit MPU6050

#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>

Adafruit_MPU6050 mpu;

void setup(void) {
 Serial.begin(115200);
 while (!Serial)
  delay(10); // will pause Zero, Leonardo, etc until serial console opens

 Serial.println("Adafruit MPU6050 test!");

 // Try to initialize!
 if (!mpu.begin()) {
  Serial.println("Failed to find MPU6050 chip");
  while (1) {
   delay(10);
  }
 }
 Serial.println("MPU6050 Found!");
```

```
  mpu.setAccelerometerRange(MPU6050_RANGE_8_G);
  Serial.print("Accelerometer range set to: ");
  switch (mpu.getAccelerometerRange()) {
  case MPU6050_RANGE_2_G:
    Serial.println("+-2G");
    break;
    ……
  }
  mpu.setGyroRange(MPU6050_RANGE_500_DEG);
  Serial.print("Gyro range set to: ");
   ……
  }

  // Sets the bandwidth of the Digital Low-Pass Filter
  mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
  Serial.print("Filter bandwidth set to: ");
   ……
  }

  Serial.println("");
  delay(100);
}
```

Unit conversion (acc)

- Read “byte value”
  => convert to “G”
  => convert to “m/s^2”

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
ACCELEROMETER SENSITIVITY						
Full-Scale Range	AFS_SEL=0		±2		*g*	
	AFS_SEL=1		±4		*g*	
	AFS_SEL=2		±8		*g*	
	AFS_SEL=3		±16		*g*	
ADC Word Length	Output in two’s complement format		16		bits	
Sensitivity Scale Factor	AFS_SEL=0		16,384		LSB/*g*	
	AFS_SEL=1		8,192		LSB/*g*	
	AFS_SEL=2		4,096		LSB/*g*	
	AFS_SEL=3		2,048		LSB/*g*	
Initial Calibration Tolerance			±3		%	
Sensitivity Change vs. Temperature	AFS_SEL=0, -40°C to +85°C		±0.02		%/°C	
Nonlinearity	Best Fit Straight Line		0.5		%	
Cross-Axis Sensitivity			±2		%	

Unit conversion (gyro)

- Read "byte value"
  => convert to "angular velocity" (degree per second)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
GYROSCOPE SENSITIVITY						
Full-Scale Range	FS_SEL=0		±250		º/s	
	FS_SEL=1		±500		º/s	
	FS_SEL=2		±1000		º/s	
	FS_SEL=3		±2000		º/s	
Gyroscope ADC Word Length			16		bits	
Sensitivity Scale Factor	FS_SEL=0		131		LSB/(º/s)	
	FS_SEL=1		65.5		LSB/(º/s)	
	FS_SEL=2		32.8		LSB/(º/s)	
	FS_SEL=3		16.4		LSB/(º/s)	

LOW PASS FILTER RESPONSE	Programmable Range	5		256	Hz	

```
void loop() {

  /* Get new sensor events with the readings */
  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);

  /* Print out the values */
  Serial.print("Acceleration X: ");   // print acc X,Y,Z
  Serial.print(a.acceleration.x);
  …
  Serial.println(" m/s^2");

  Serial.print("Rotation X: ");       // print gyro X,Y,Z
  Serial.print(g.gyro.x);
  …
  Serial.println(" rad/s");

  Serial.print("Temperature: ");
  Serial.print(temp.temperature);
  Serial.println(" degC");

  Serial.println("");
  delay(500);
}
```

Results (basic_readings)

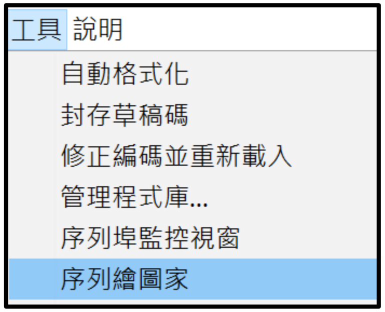
COM6

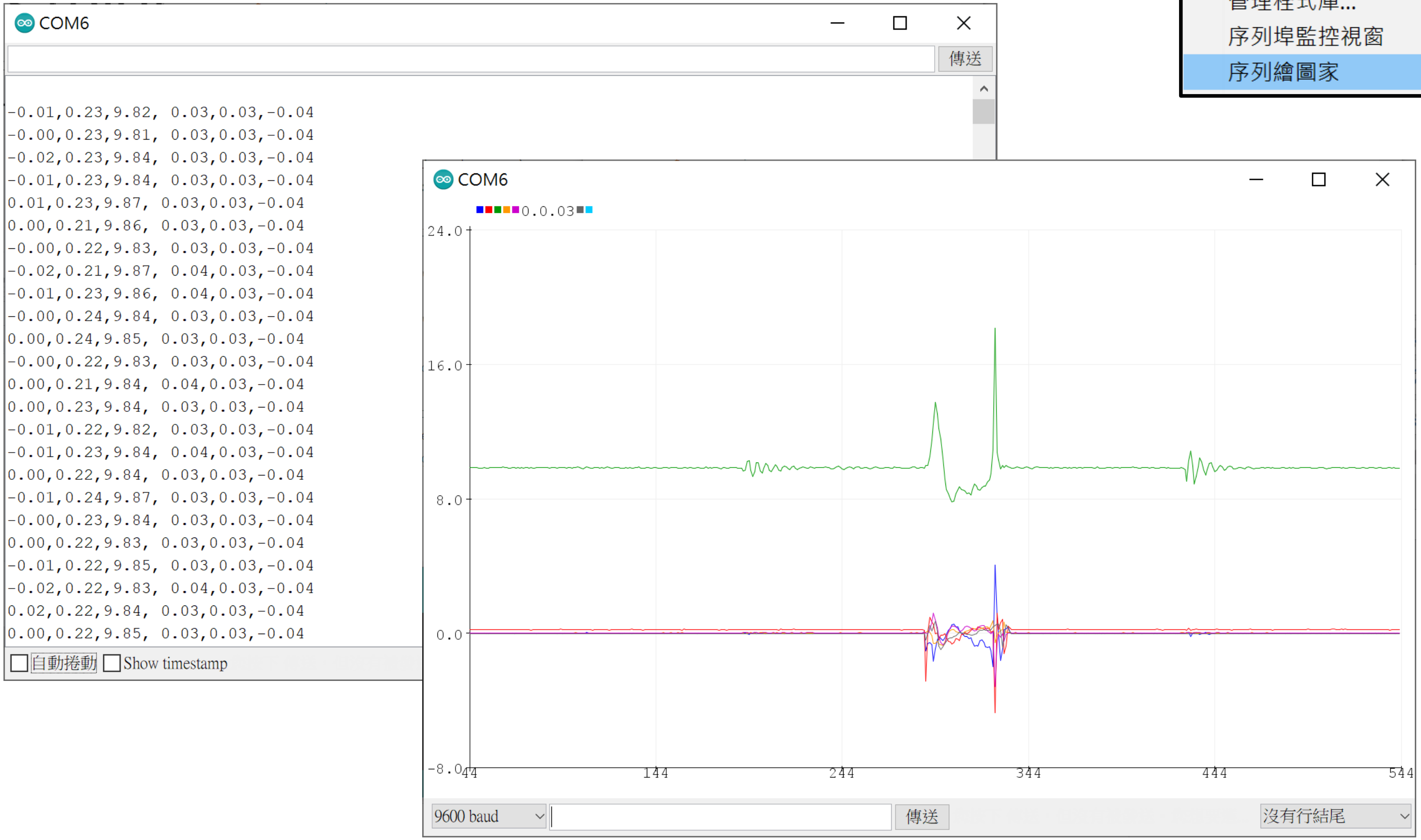
傳送

```
Adafruit MPU6050 test!
MPU6050 Found!
Accelerometer range set to: +-8G
Gyro range set to: +- 500 deg/s
Filter bandwidth set to: 21 Hz

Acceleration X: -0.09, Y: 2.21, Z: 9.61 m/s^2
Rotation X: 0.03, Y: 0.03, Z: -0.04 rad/s
Temperature: 29.17 degC

Acceleration X: -0.09, Y: 2.20, Z: 9.63 m/s^2
Rotation X: 0.03, Y: 0.03, Z: -0.04 rad/s
Temperature: 29.19 degC

Acceleration X: -0.09, Y: 2.20, Z: 9.59 m/s^2
Rotation X: 0.03, Y: 0.03, Z: -0.04 rad/s
```

自動捲動 Show timestamp NL(newline) 9600 baud Clear output

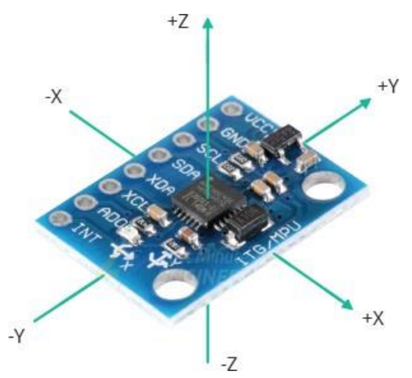
Results (plotter)

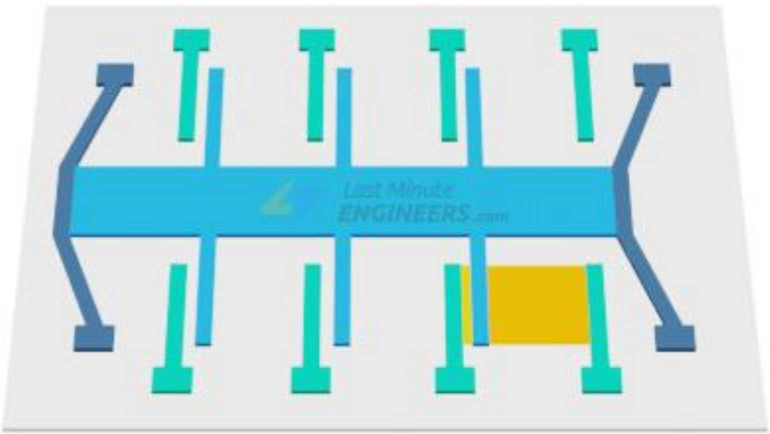




Brief summary - 1

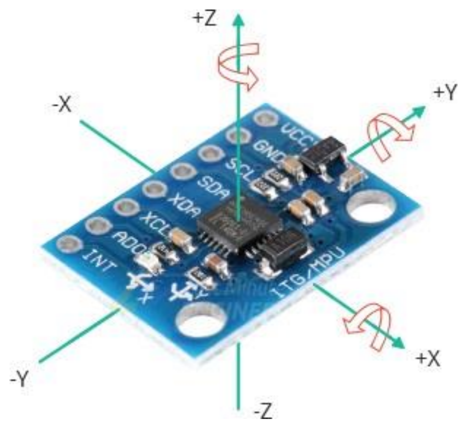
- What can we obtain from accelerometer?
  - 1. Movement along with xyz-axis (acceleration)
    - Unit: byte, G, or m/s^2
  - 2. Posture (Roll, Pitch)
    - Unit: Radians or Degree

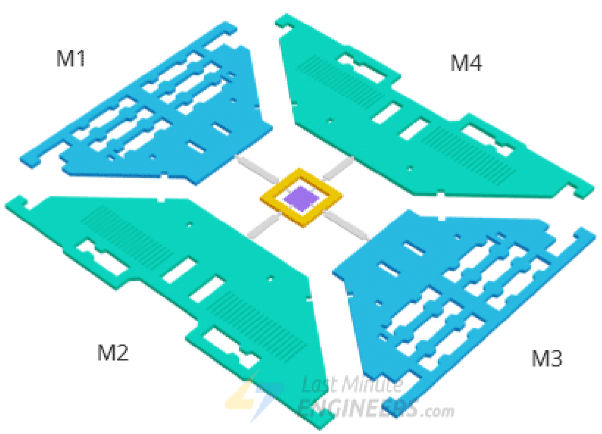




Brief summary - 2

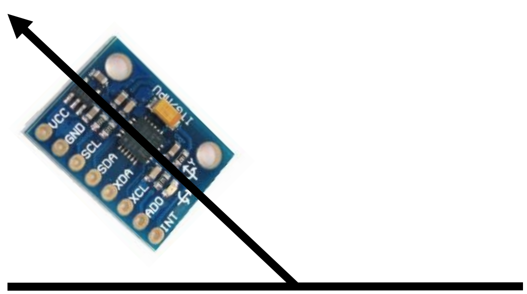
- What can we obtain from gyroscope?
  - 1. rotation (angular velocity)
    - Unit: byte, DPS (Degree per Second)
  - 2. rotated angle (integral “angular velocity” to “angle”)
    - Unit: degree. You can convert it to rad.

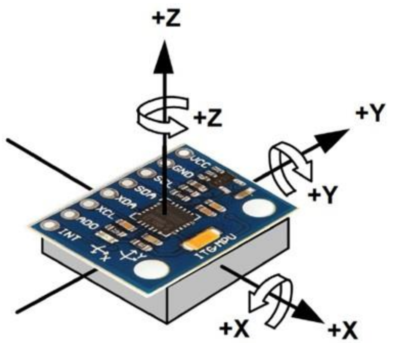




Discussion 1

- **How to find the xyz direction** of accelerometer and gyroscope?
  - Hint: Try to rotate MPU-6050 along the xy plane, yz plane, and xz plane. Observe the (gx, gy, gz) value.
- When you put the gyroscope at a certain angle, **what is the rotation value** for (gx, gy, gz)?
  - The gyroscope is static, not moving.





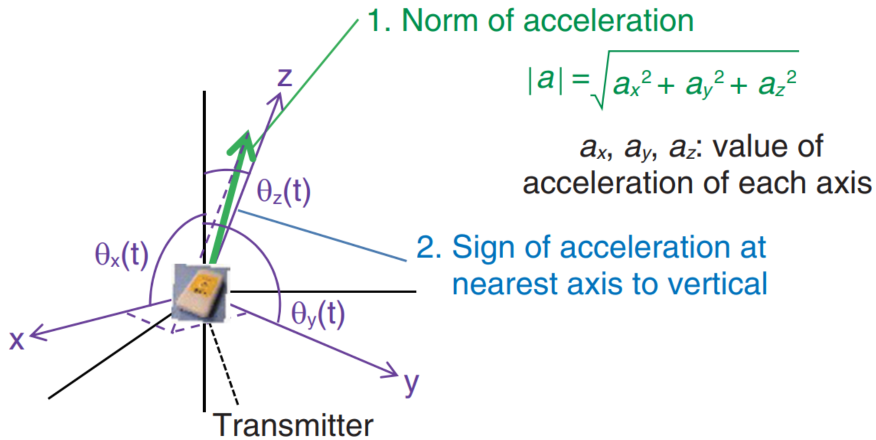
Quiz 1

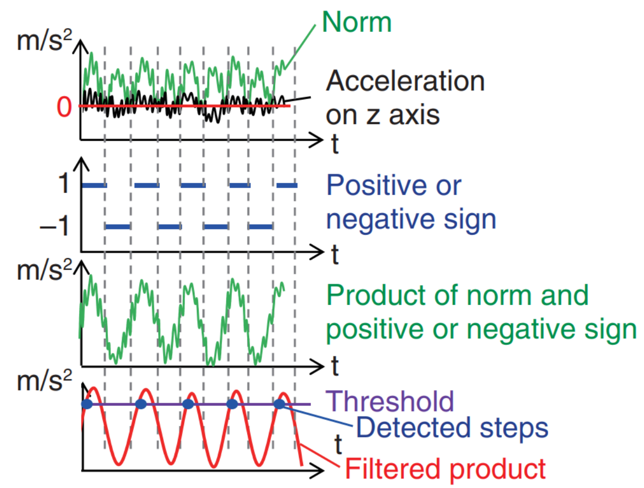
- **Calculate the norm of acceleration |a|**
- Formula:

  |a| = sqrt(x^2 + y^2 + z^2) ← Do not use ^. It is Bitwise Operators.

  = sqrt(pow(ax,2) + pow(ay,2) + pow(az,2))

  https://www.arduino.cc/reference/en/language/functions/math/sqrt/
  https://www.arduino.cc/reference/en/language/functions/math/pow/
  https://www.arduino.cc/reference/en/language/structure/bitwise-operators/bitwisexor/





Gait Analysis Using a Wearable T-shirt Type Sensor
https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201604ra1.html

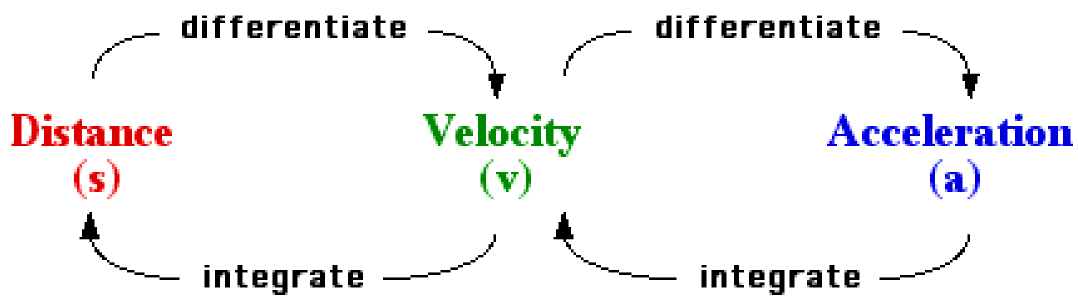
Quiz 2

- We can obtain acceleration (ax/ay/az) from accelerometer. How to **calculate the distance**?
- For gyroscope, we can obtain angular velocity (gx/gy/gz). How to **calculate the rotation angle**?

$$\mathbf{s}(t) = \mathbf{s}_0 + \mathbf{v}_0 t + \tfrac{1}{2}\mathbf{a}t^2 = \mathbf{s}_0 + \frac{\mathbf{v}_0 + \mathbf{v}(t)}{2}t$$

$$\mathbf{v}(t) = \mathbf{v}_0 + \mathbf{a}t$$

$$v^2(t) = {v_0}^2 + 2\mathbf{a}\cdot[\mathbf{s}(t) - \mathbf{s}_0]$$



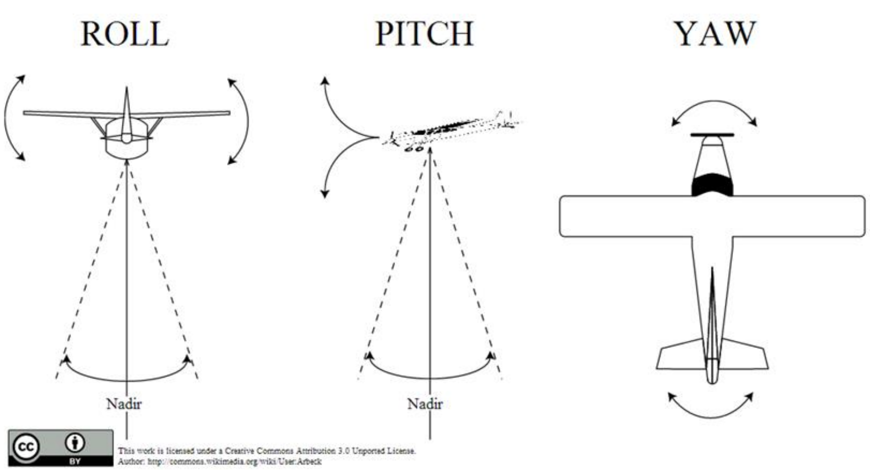
Quiz 3

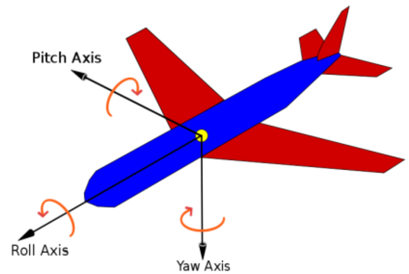
- **Calculate roll and pitch of the sensor**, then print in serial monitor
  - [Ref] atan2(y,x): arc tangent of y/x

公式有更新!

$$roll = arctan\left(\frac{-G_y}{G_z}\right)$$

$$pitch = arctan\left(\frac{G_x}{\sqrt{G_y^2 + G_z^2}}\right)$$



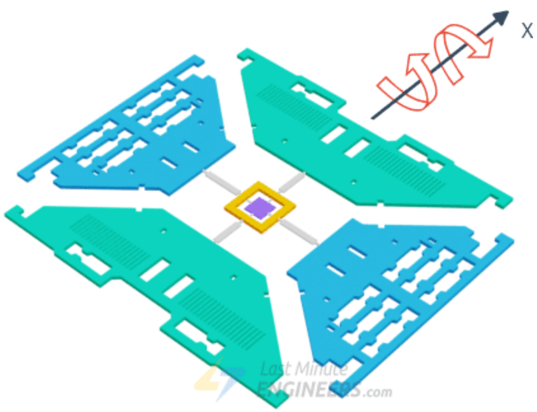


http://bluelemonlabs.blogspot.tw/2013/08/arduino-imu-pitch-roll-from-adxl345.html

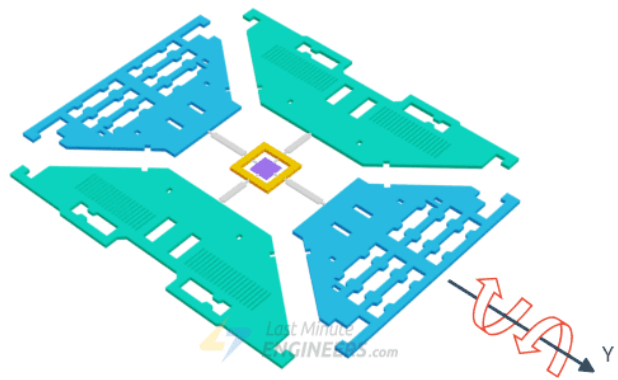
For better performance, sensor fusion (acc+gyro) is usually adopted to provide more accurate results, but it needs to calibrate and compensate the reading.

Quiz 3 (cont.)

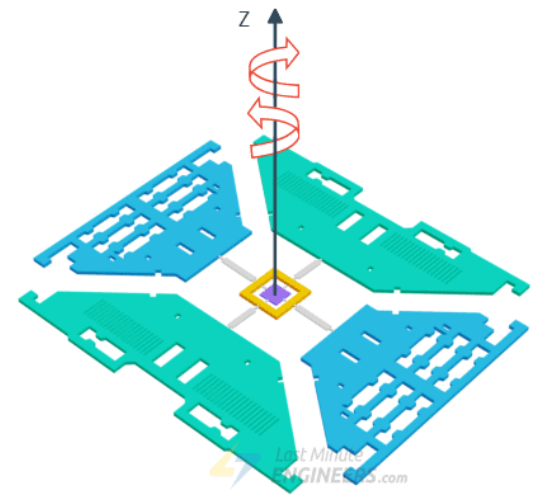
Roll



Pitch

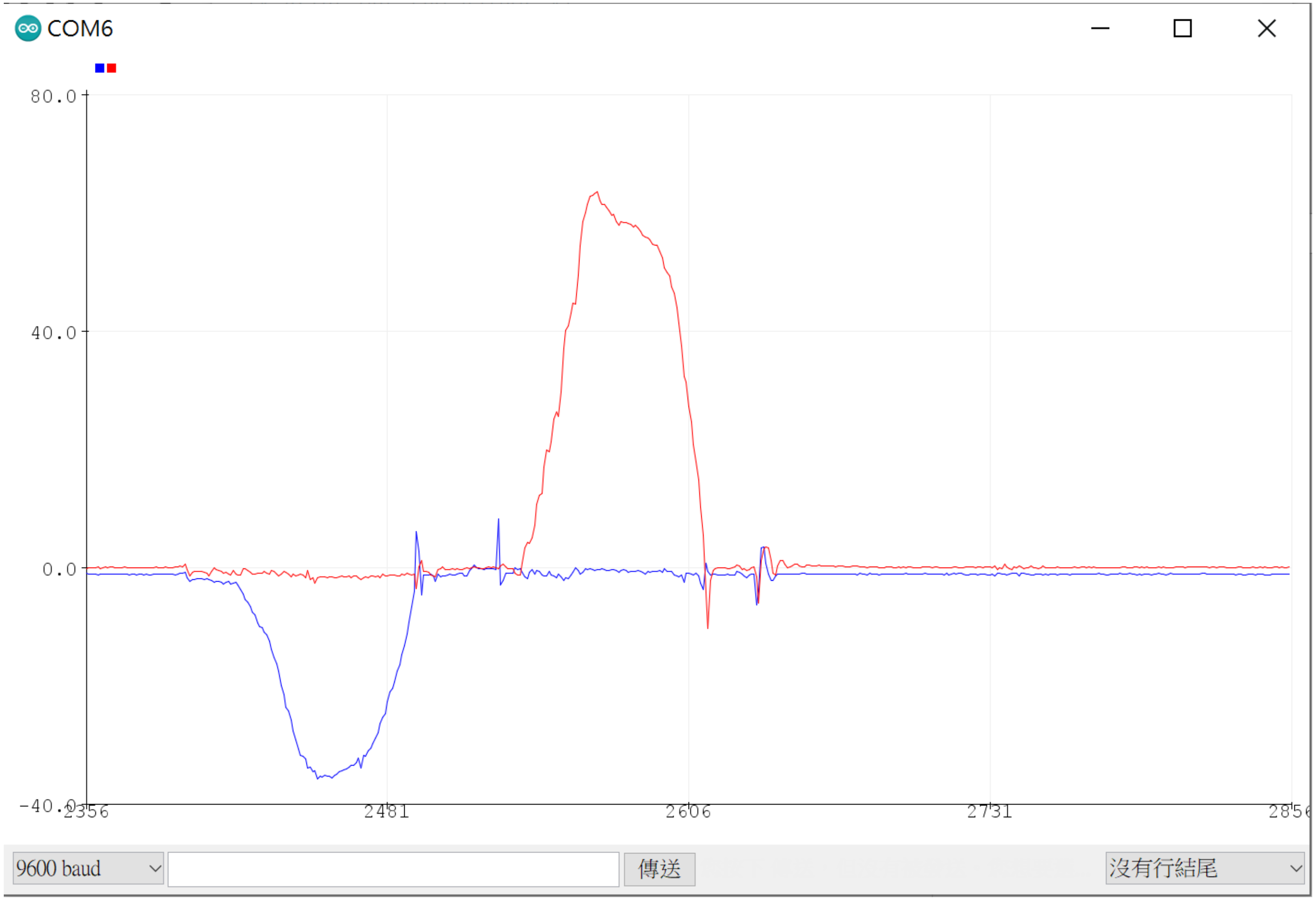


Yaw



https://lastminuteengineers.com/mpu6050-accel-gyro-arduino-tutorial/

Results


COM6
80.0
40.0
0.0
-40.0
2356
2481
2606
2731
2856
9600 baud
傳送
沒有行結尾

Lab2. Digital Motion Processor (DMP)

The MPU-60X0 is the world's first integrated 6-axis MotionTracking device that combines a 3-axis gyroscope, 3-axis accelerometer, and a Digital Motion Processor™ (DMP) all in a small 4x4x0.9mm package.

MPU6050 datasheet

7.9 Digital Motion Processor

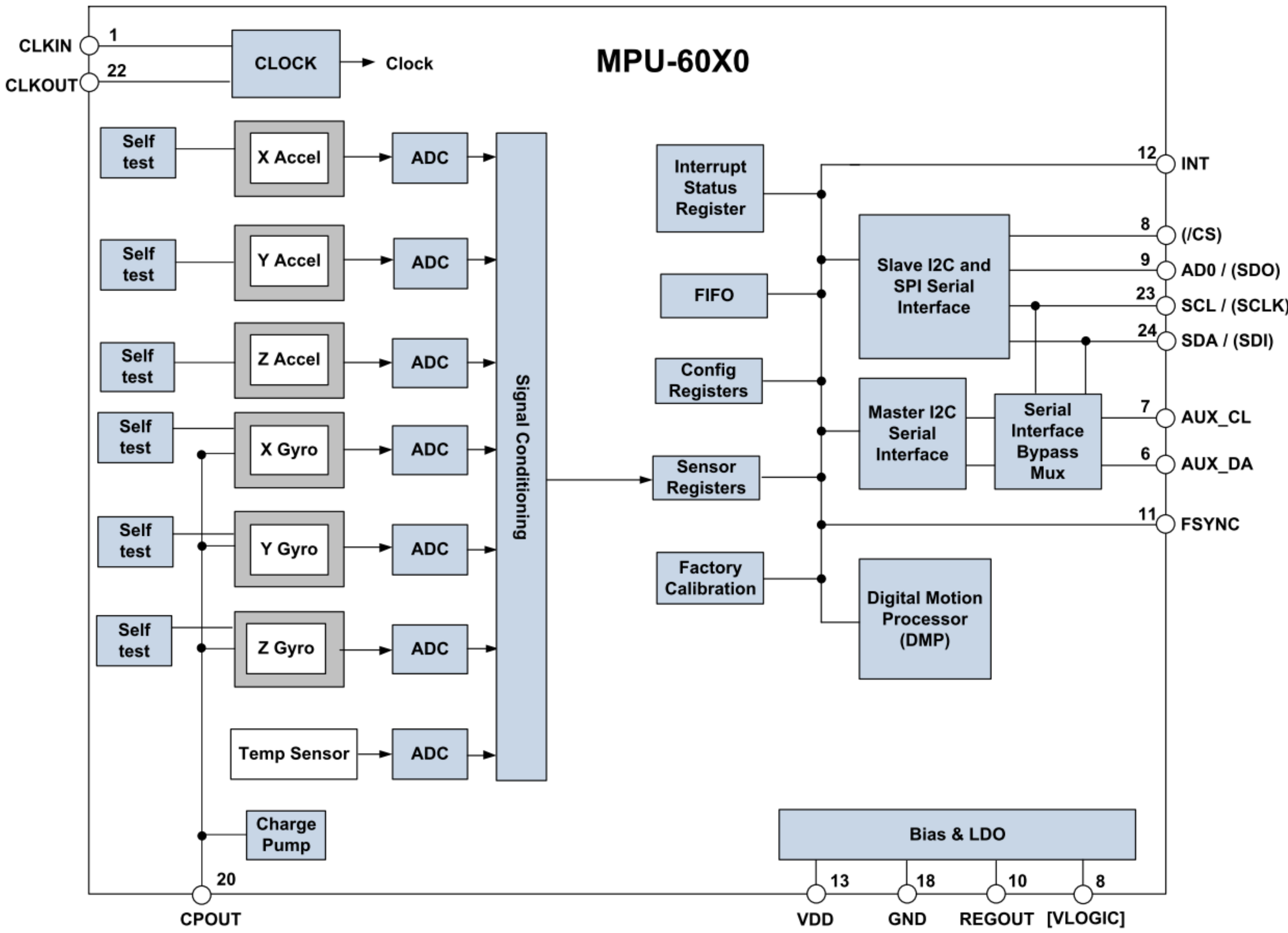
The embedded Digital Motion Processor (DMP) is located within the MPU-60X0 and offloads computation of motion processing algorithms from the host processor. The DMP acquires data from accelerometers, gyroscopes, and additional 3rd party sensors such as magnetometers, and processes the data. The resulting data can be read from the DMP's registers, or can be buffered in a FIFO. The DMP has access to one of the MPU's external pins, which can be used for generating interrupts.

The purpose of the DMP is to offload both timing requirements and processing power from the host processor. Typically, motion processing algorithms should be run at a high rate, often around 200Hz, in order to provide accurate results with low latency. This is required even if the application updates at a much lower rate; for example, a low power user interface may update as slowly as 5Hz, but the motion processing should still run at 200Hz. The DMP can be used as a tool in order to minimize power, simplify timing, simplify the software architecture, and save valuable MIPS on the host processor for use in the application.

https://invensense.tdk.com/products/motion-tracking/6-axis/
https://3cfeqx1hf82y3xcoull08ihx-wpengine.netdna-ssl.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf

```
/* ================================================================================================ *
 | Default MotionApps v2.0 42-byte FIFO packet structure:                                           |
 |                                                                                                  |
 | [QUAT W][      ][QUAT X][      ][QUAT Y][      ][QUAT Z][      ][GYRO X][      ][GYRO Y][      ] |
 |   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  |
 |                                                                                                  |
 | [GYRO Z][      ][ACC X ][      ][ACC Y ][      ][ACC Z ][      ][      ]                         |
 |  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41                          |
 * ================================================================================================ */
```

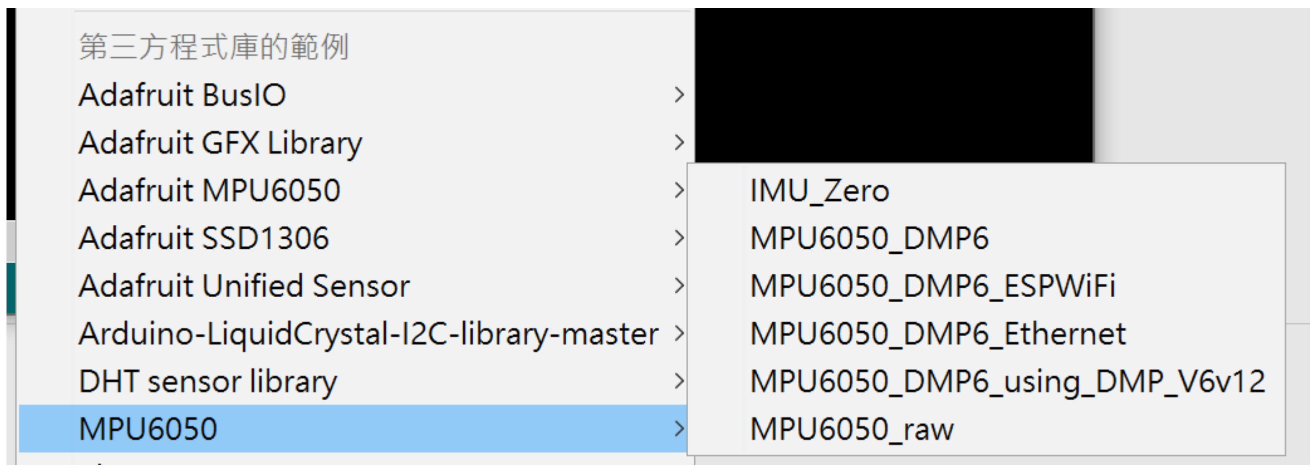
Block Diagram



Lab2. DMP

Arduino IDE

Open--->File--->Examples---> 第三方程式庫 --> MPU6050



陽明交大 NYCU

Sample code: MPU6050_DMP6

```
// see the actual quaternion components in a [w, x, y, z] format
#define OUTPUT_READABLE_QUATERNION

// see Euler angles (in degrees)
#define OUTPUT_READABLE_EULER

// see the yaw/pitch/roll angles (in degrees)
#define OUTPUT_READABLE_YAWPITCHROLL

// see acceleration components with gravity removed
#define OUTPUT_READABLE_REALACCEL

……

    // supply your own gyro offsets here, scaled for min sensitivity
    mpu.setXGyroOffset(220);
    mpu.setYGyroOffset(76);
    mpu.setZGyroOffset(-85);
    mpu.setZAccelOffset(1788); // 1688 factory default for my test chip
```

About offset: https://www.i2cdevlib.com/forums/topic/91-how-to-decide-gyro-and-accelerometer-offsett/

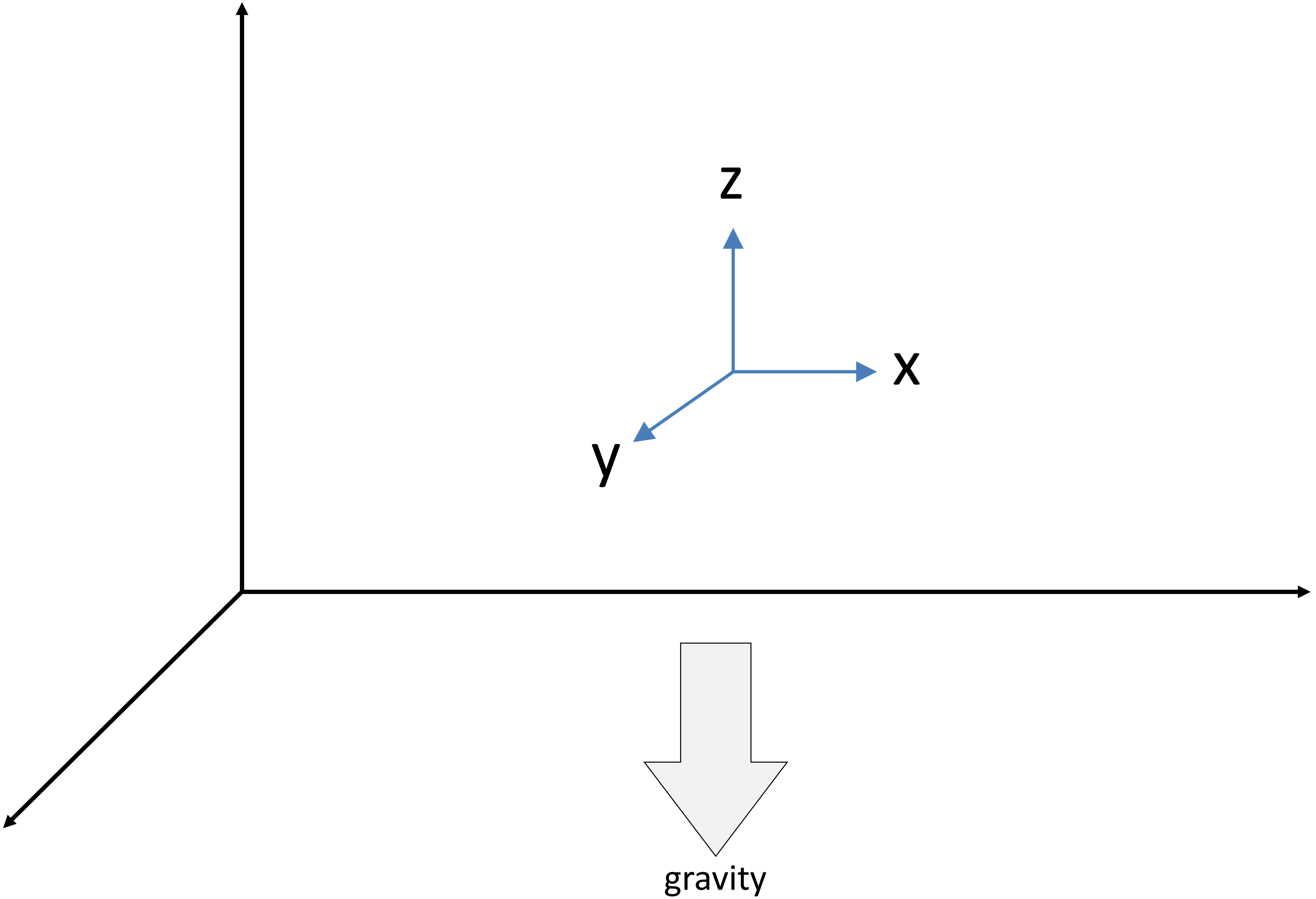
Sample code: MPU6050_DMP6

```
DEBUG_PRINTLN(F("Setting DLPF bandwidth to 42Hz..."));
setDLPFMode(MPU6050_DLPF_BW_42);

DEBUG_PRINTLN(F("Setting gyro sensitivity to +/- 2000 deg/sec..."));
setFullScaleGyroRange(MPU6050_GYRO_FS_2000);
```

> https://github.com/ElectronicCats/mpu6050/blob/master/src/MPU6050_6Axis_MotionApps20.h

```
void MPU6050::initialize() {
    setClockSource(MPU6050_CLOCK_PLL_XGYRO);
    setFullScaleGyroRange(MPU6050_GYRO_FS_250);
    setFullScaleAccelRange(MPU6050_ACCEL_FS_2);
    setSleepEnabled(false); // thanks to Jack Els
}
```

> https://github.com/ElectronicCats/mpu6050/blob/master/src/MPU6050.cpp


z
x
y
gravity

Result

COM6

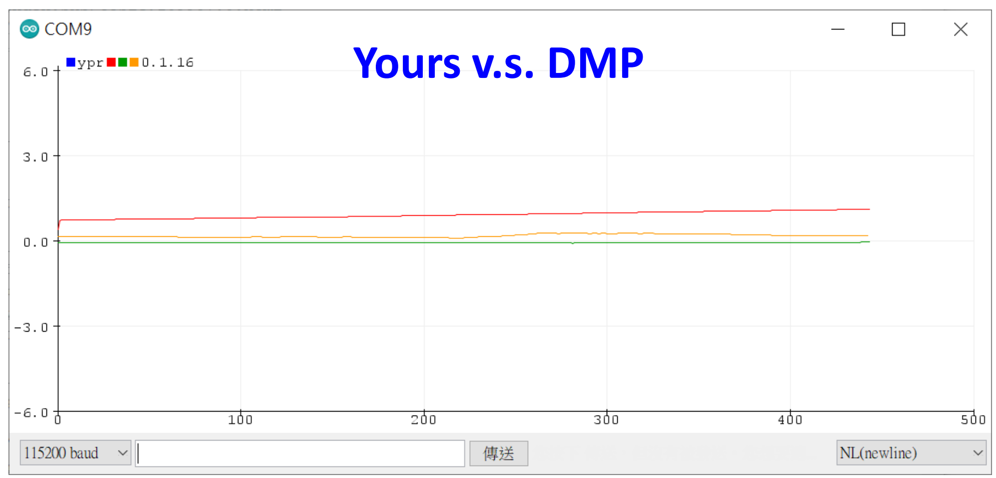
傳送

```
Initializing I2C devices...
Testing device connections...
MPU6050 connection successful

Send any character to begin DMP programming and demo:
Initializing DMP...
>......>......
//           X Accel  Y Accel  Z Accel   X Gyro   Y Gyro   Z Gyro
//OFFSETS     -468,   -1831,    1646,     -58,     -59,      73
Enabling DMP...
Enabling interrupt detection (Arduino external interrupt 1)...
DMP ready! Waiting for first interrupt...
quat    1.00    0.00    0.00    -0.00
euler   0.01    -0.01   -0.44
ypr     0.01    -0.01   0.44
areal   8       -67     -5143
aworld  6       -27     -5142
quat    1.00    0.00    0.00    -0.00
euler   0.01    -0.01   -0.44
ypr     0.01    -0.01   0.44
areal   9       -64     -3774
aworld  8       -34     -3773
quat    1.00    0.00    0.00    -0.00
euler   0.01    -0.01   -0.44
ypr     0.01    -0.01   0.44
```

自動捲動 Show timestamp NL(newline) 9600 baud Clear output

Discussion 2

- The embedded Digital Motion Processor can report the roll/pitch/yaw.
- In quiz 2, we also calculate roll and pitch.
- **Please compare the results between yours and DMP.**



Summary

Summary

- **"請記得填寫"教室座位實聯制**
  - https://docs.google.com/spreadsheets/d/1k4q-JP9Pk9cLGY70V04Nbc6XbUbBdYu_TXqJtHF6rGk
- **Practice Labs by yourself**
- **Write Answers for Discussion**
  - Upload to e3 before next class
- **Quiz: Write code for quiz, then demonstrate to TAs**
  - Quiz 1. Calculate the norm of acceleration |a|
  - Quiz 2. Calculate the distance and rotation angle
  - Quiz 3. Calculate roll and pitch of the sensor

陽明交大
NYCU

Parkinson's Disease Prediction based on Hand Tremor Analysis

Fig. 3. System Setup

Fig. 4. ADXL335

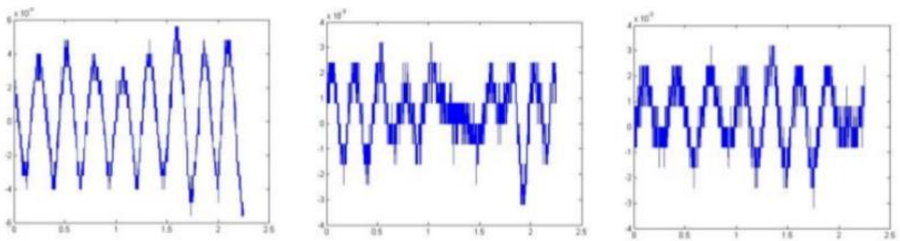


Fig. 5. Waveforms obtained from DSO for X, Y and Z axis respectively.

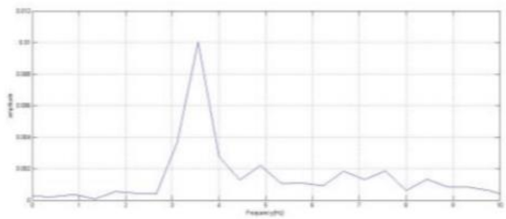


Fig. 6. FFT of X axis of the right hand of the Parkinson's disease patient

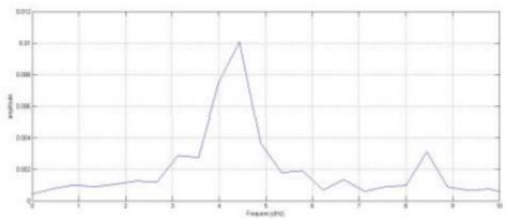


Fig. 7. FFT of Y axis of the right hand of the Parkinson's disease patient

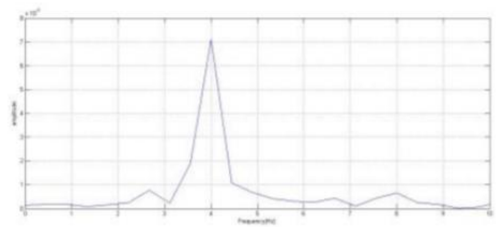


Fig. 8. FFT of Z axis of the right hand of the Parkinson's disease patient