Motion Controller (1)

Making a wireless motion controller - motion

Demo

https://www.youtube.com/watch?v=gLKGu0S5caA

Implementation

3 weeks to implement

- **Sensing motion (10/19)**
- Unity (10/26)
- Build a case (11/02 after midterm) ※ TBD

Report due on 11/16 ※ TBD

Materials

(1) Breadboard x1

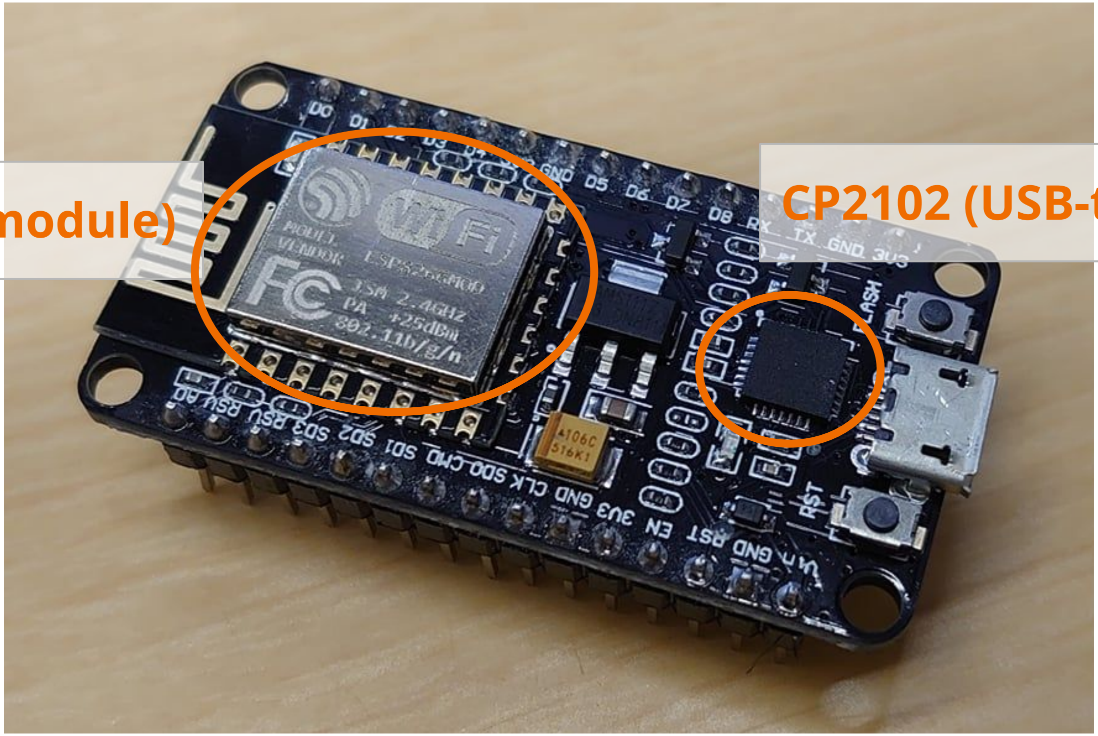
(2) NodeMCU ESP8266 x1

(3) 1kΩ Resistor x1

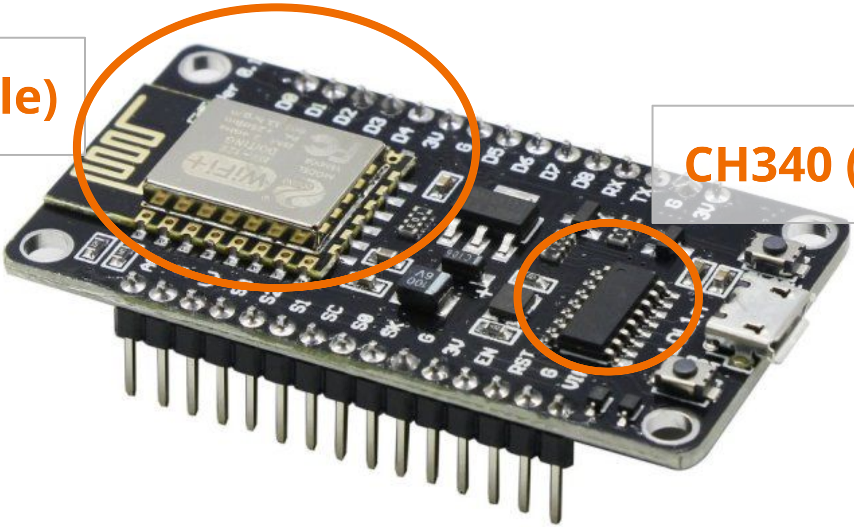
(4) Button x1

(5) IMU sensor (GY-521) x1

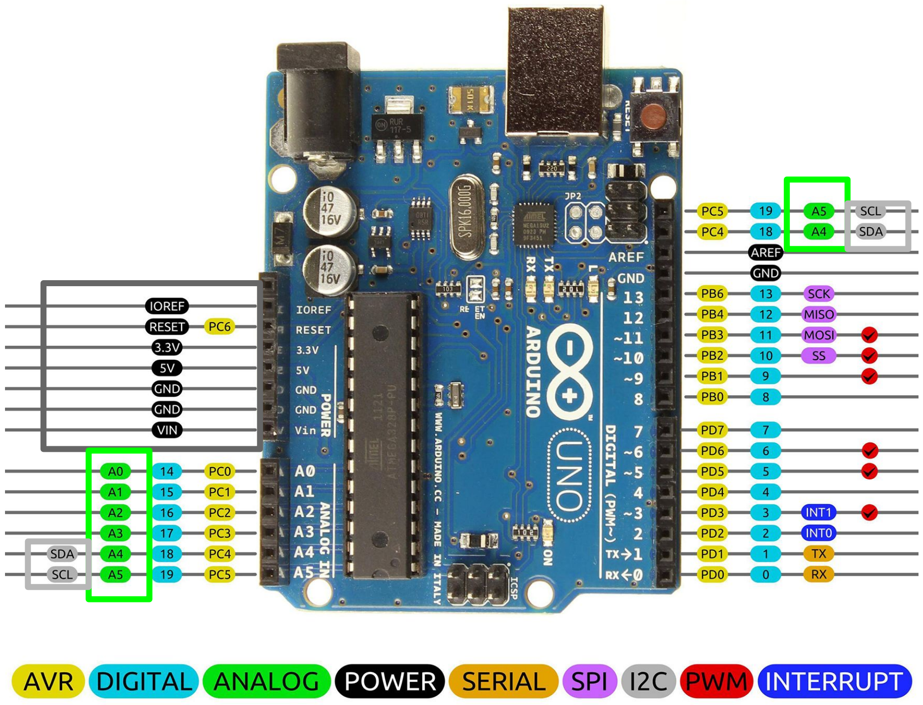
NodeMCU: Wifi-Capable Microcontroller

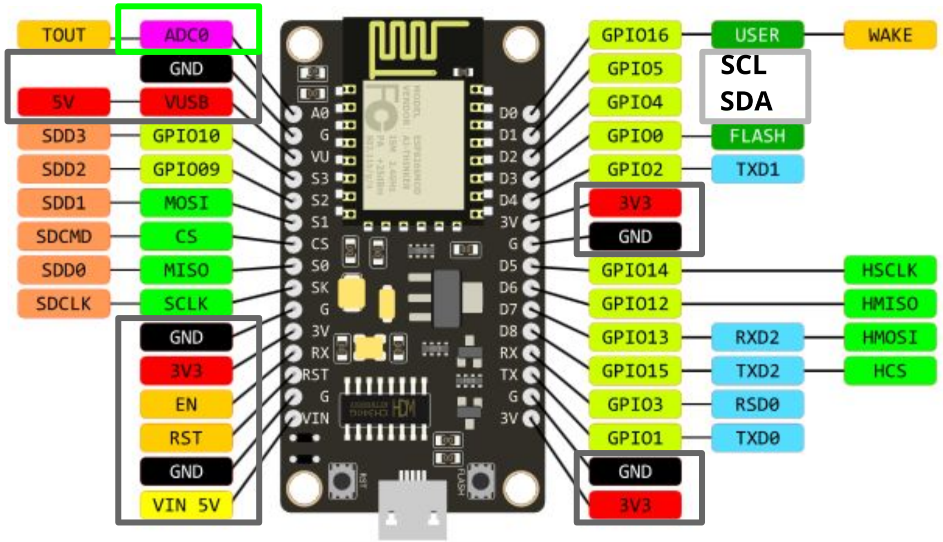


NodeMCU: Wifi-Capable Microcontroller



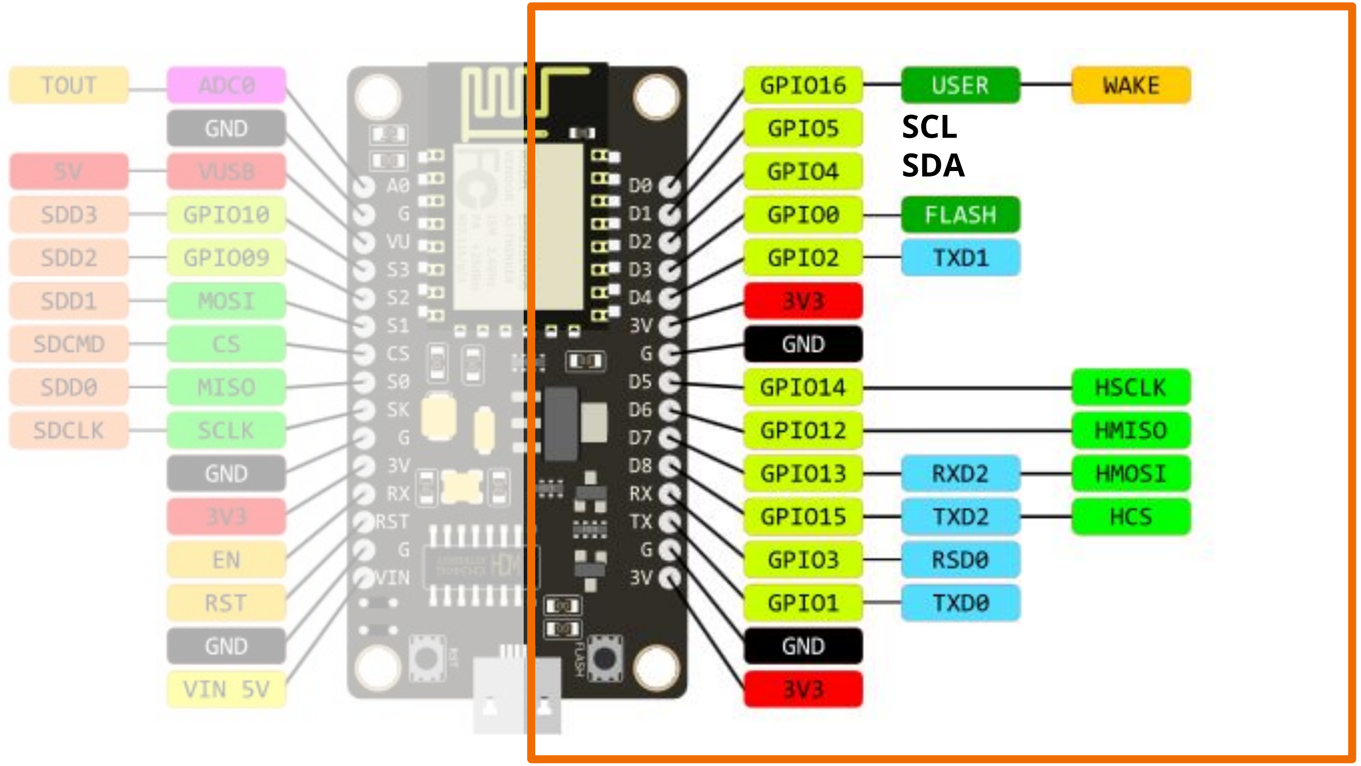
Arduino vs. NodeMCU Pinout



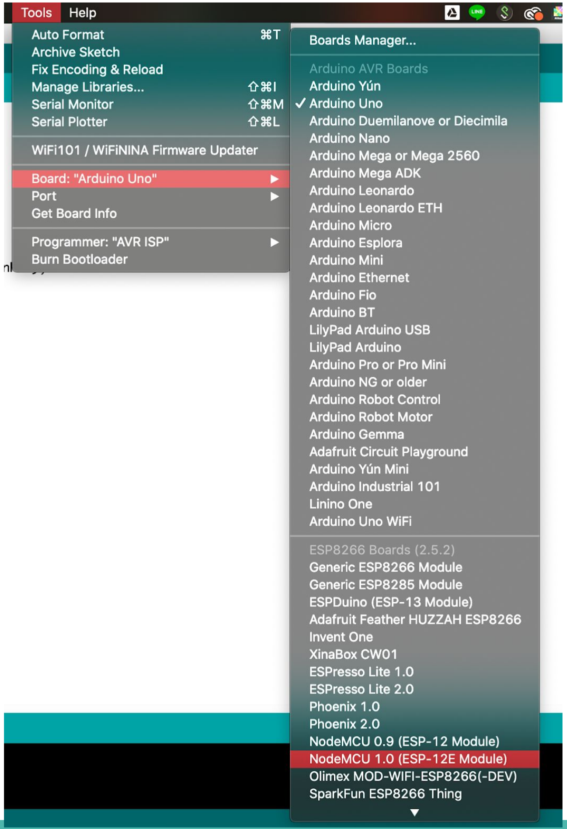


GPIO = **G**eneral **P**urpose **I**nput/**O**utput

NodeMCU Pinout Diagram



Time to implement!


Tools Help

Auto Format ⌘T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... ⇧⌘I
Serial Monitor ⇧⌘M
Serial Plotter ⇧⌘L

WiFi101 / WiFiNINA Firmware Updater

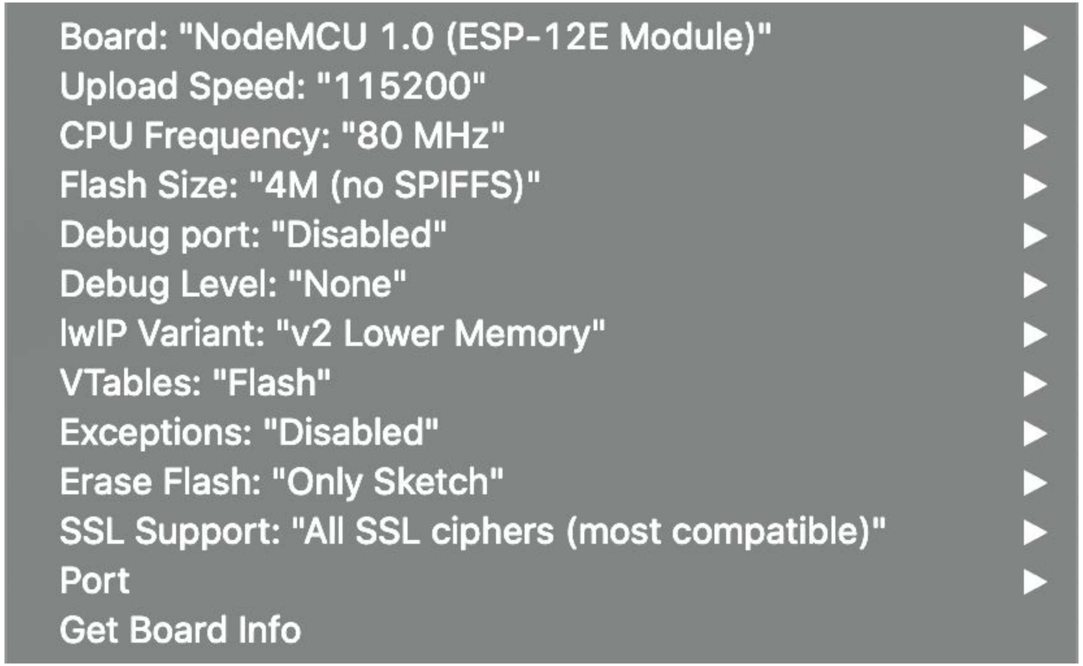
Board: "Arduino Uno" ▶
Port ▶
Get Board Info

Programmer: "AVR ISP" ▶
Burn Bootloader

Boards Manager...

Arduino AVR Boards
Arduino Yún
✓ Arduino Uno
Arduino Duemilanove or Diecimila
Arduino Nano
Arduino Mega or Mega 2560
Arduino Mega ADK
Arduino Leonardo
Arduino Leonardo ETH
Arduino Micro
Arduino Esplora
Arduino Mini
Arduino Ethernet
Arduino Fio
Arduino BT
LilyPad Arduino USB
LilyPad Arduino
Arduino Pro or Pro Mini
Arduino NG or older
Arduino Robot Control
Arduino Robot Motor
Arduino Gemma
Adafruit Circuit Playground
Arduino Yún Mini
Arduino Industrial 101
Linino One
Arduino Uno WiFi

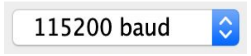
ESP8266 Boards (2.5.2)
Generic ESP8266 Module
Generic ESP8285 Module
ESPDuino (ESP-13 Module)
Adafruit Feather HUZZAH ESP8266
Invent One
XinaBox CW01
ESPresso Lite 1.0
ESPresso Lite 2.0
Phoenix 1.0
Phoenix 2.0
NodeMCU 0.9 (ESP-12 Module)
NodeMCU 1.0 (ESP-12E Module)
Olimex MOD-WIFI-ESP8266(-DEV)
SparkFun ESP8266 Thing


Board: "NodeMCU 1.0 (ESP-12E Module)" ▶
Upload Speed: "115200" ▶
CPU Frequency: "80 MHz" ▶
Flash Size: "4M (no SPIFFS)" ▶
Debug port: "Disabled" ▶
Debug Level: "None" ▶
lwIP Variant: "v2 Lower Memory" ▶
VTables: "Flash" ▶
Exceptions: "Disabled" ▶
Erase Flash: "Only Sketch" ▶
SSL Support: "All SSL ciphers (most compatible)" ▶
Port ▶
Get Board Info

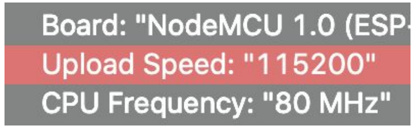
Print Hello World using this board!

The Baud in the following three places should be the same

- In your code `Serial.begin(115200);`
- In Serial Monitor
- In “Tools”

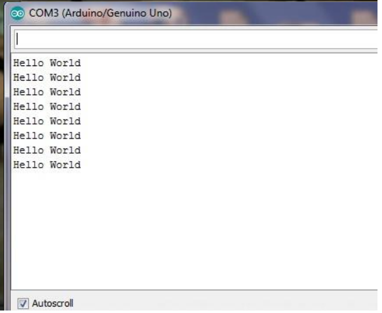


```
115200 baud
```



```
Board: "NodeMCU 1.0 (ESP
Upload Speed: "115200"
CPU Frequency: "80 MHz"
```

If you can not compile an empty sketch, restart the Arduino IDE.

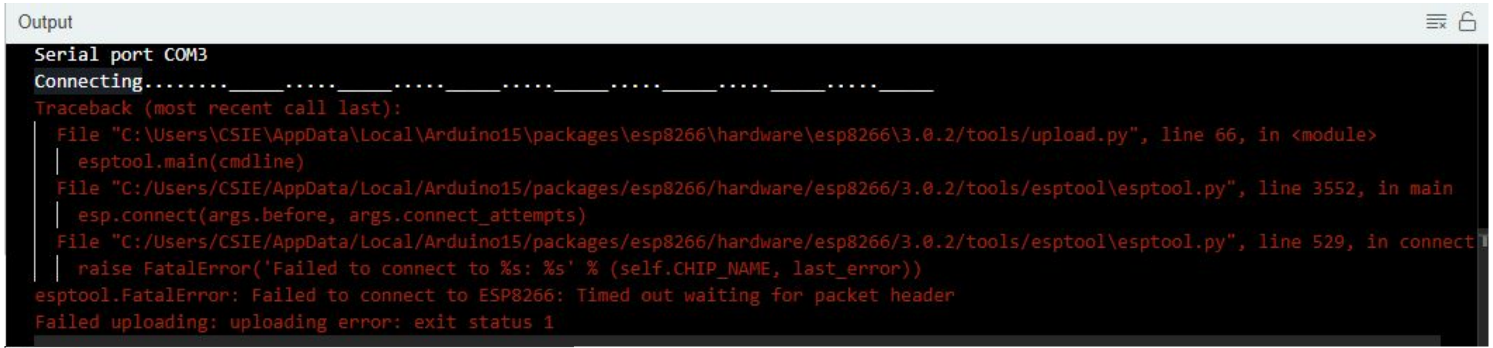


COM3 (Arduino/Genuino Uno)

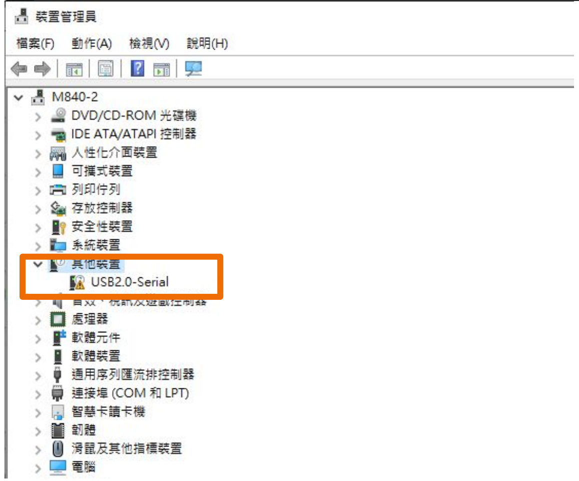
```
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
Hello World
```

Autoscroll

If the CP2102 driver doesn't work:

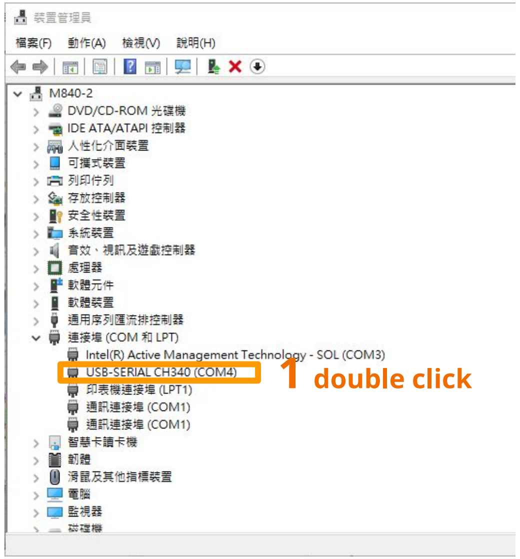

Output

```
Serial port COM3
Connecting........_____....._____....._____....._____....._____....._____....._____
Traceback (most recent call last):
  File "C:\Users\CSIE\AppData\Local\Arduino15\packages\esp8266\hardware\esp8266\3.0.2/tools/upload.py", line 66, in <module>
    esptool.main(cmdline)
  File "C:/Users/CSIE/AppData/Local/Arduino15/packages/esp8266/hardware/esp8266/3.0.2/tools/esptool\esptool.py", line 3552, in main
    esp.connect(args.before, args.connect_attempts)
  File "C:/Users/CSIE/AppData/Local/Arduino15/packages/esp8266/hardware/esp8266/3.0.2/tools/esptool\esptool.py", line 529, in connect
    raise FatalError('Failed to connect to %s: %s' % (self.CHIP_NAME, last_error))
esptool.FatalError: Failed to connect to ESP8266: Timed out waiting for packet header
Failed uploading: uploading error: exit status 1
```



If the CP2102 driver doesn’t work:

Download and install CH340 Drivers: https://sparks.gogo.co.nz/ch340.html

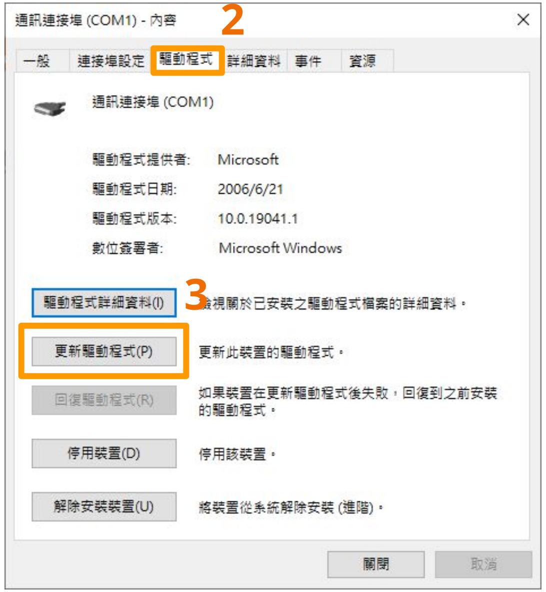


裝置管理員

檔案(F) 動作(A) 檢視(V) 說明(H)

- M840-2
  - DVD/CD-ROM 光碟機
  - IDE ATA/ATAPI 控制器
  - 人性化介面裝置
  - 可攜式裝置
  - 列印佇列
  - 存放控制器
  - 安全性裝置
  - 系統裝置
  - 音效、視訊及遊戲控制器
  - 處理器
  - 軟體元件
  - 軟體裝置
  - 通用序列匯流排控制器
  - 連接埠 (COM 和 LPT)
    - Intel(R) Active Management Technology - SOL (COM3)
    - USB-SERIAL CH340 (COM4)
    - 印表機連接埠 (LPT1)
    - 通訊連接埠 (COM1)
    - 通訊連接埠 (COM1)
  - 智慧卡讀卡機
  - 韌體
  - 滑鼠及其他指標裝置
  - 電腦
  - 監視器
  - 磁碟機

1 double click



通訊連接埠 (COM1) - 內容

2

一般 | 連接埠設定 | 驅動程式 | 詳細資料 | 事件 | 資源

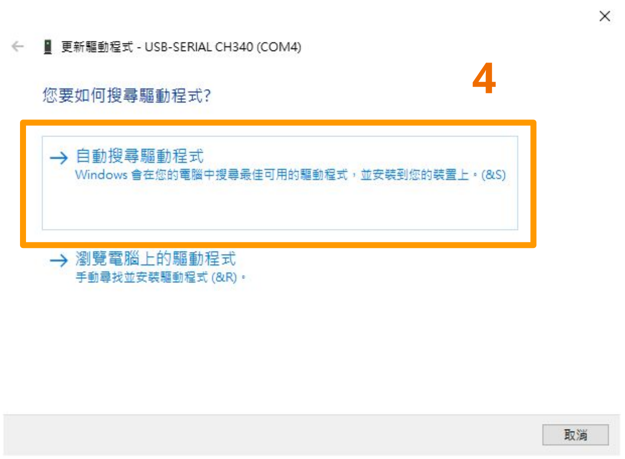
通訊連接埠 (COM1)

驅動程式提供者:	Microsoft
驅動程式日期:	2006/6/21
驅動程式版本:	10.0.19041.1
數位簽署者:	Microsoft Windows

3

驅動程式詳細資料(I)	檢視關於已安裝之驅動程式檔案的詳細資料。
更新驅動程式(P)	更新此裝置的驅動程式。
回復驅動程式(R)	如果裝置在更新驅動程式後失敗，回復到之前安裝的驅動程式。
停用裝置(D)	停用該裝置。
解除安裝裝置(U)	將裝置從系統解除安裝 (進階)。

關閉 取消



更新驅動程式 - USB-SERIAL CH340 (COM4)

4

您要如何搜尋驅動程式?

→ 自動搜尋驅動程式
Windows 會在您的電腦中搜尋最佳可用的驅動程式，並安裝到您的裝置上。(&S)

→ 瀏覽電腦上的驅動程式
手動尋找並安裝驅動程式 (&R)。

取消

Restart Arduino IDE if the port doesn’t appear

To build a wireless motion controller, we need to...

1. Detect whether the button is pressed.
2. Get the pose data from the IMU sensor.
3. System receives the data from the motion controller as user’s input.

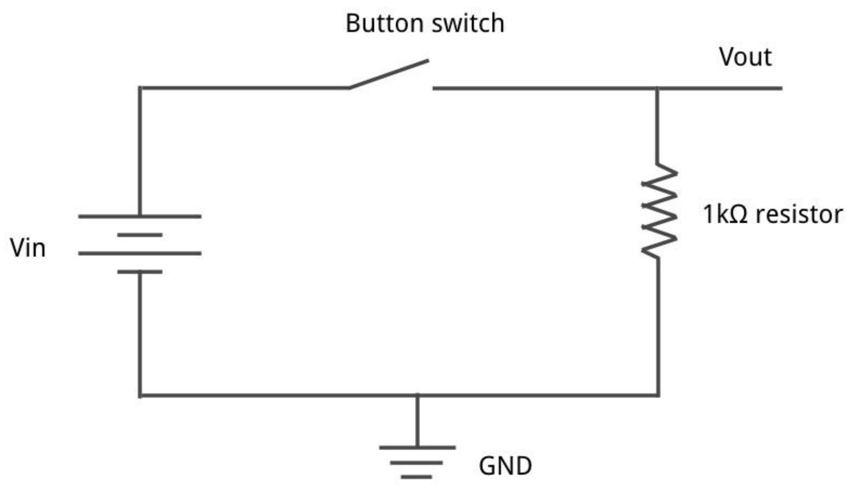
Step (1/3)

To build a wireless motion controller, we need to...

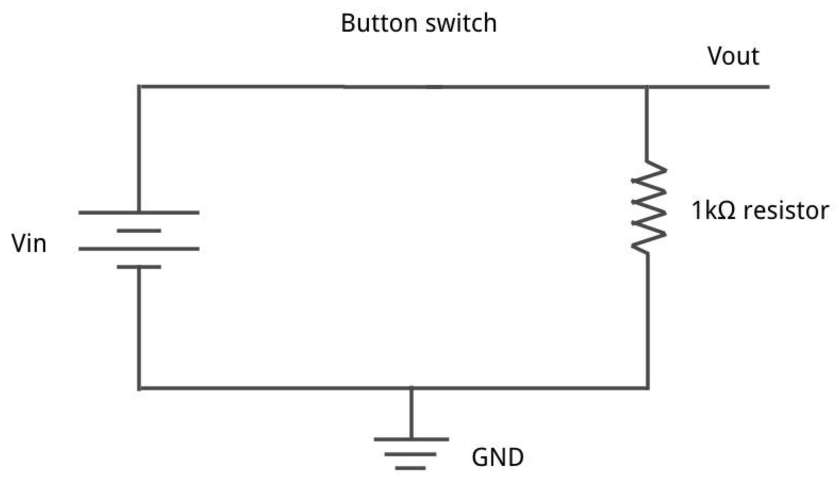
1. **Detect whether the button is pressed.**
2. Get the pose data from the IMU sensor.
3. System receives the data from the motion controller as user’s input.

How to detect the button status?

If we connect it to a resistor with a constant resistance...

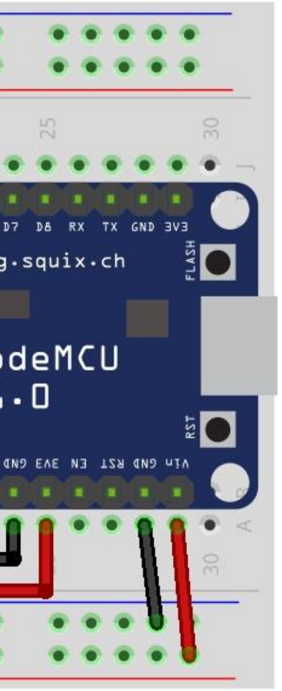


Vout = GND



Vout = Vin

Time to implement!

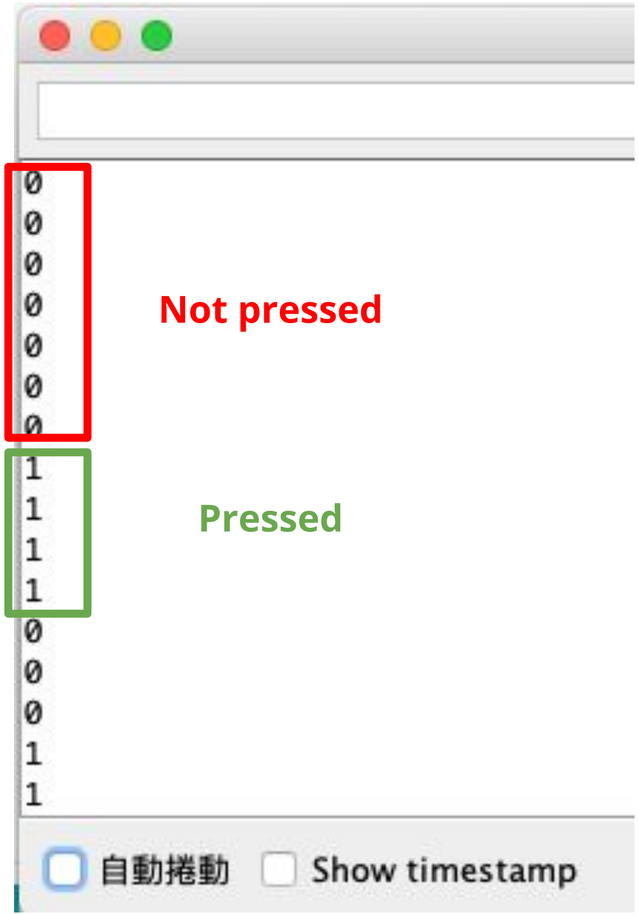


Based on the previous slide,

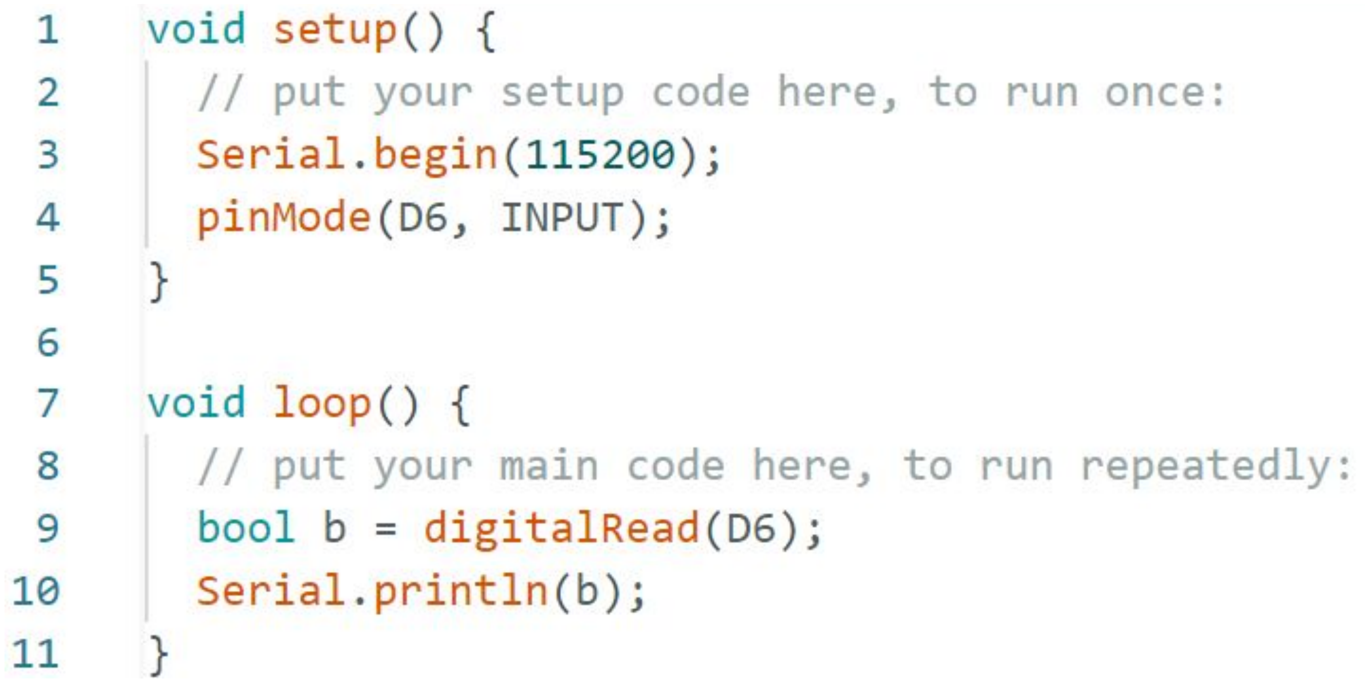
Implement a circuit and arduino code that prints 1 when the button is pressed, 0 when it isn't.

- **Use pin D6 as INPUT pin.**

HINT:
You can use the variable D6 directly.



```
0
0
0
0
0
0
0
1
1
1
1
0
0
0
1
1
```



```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  pinMode(D6, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  bool b = digitalRead(D6);
  Serial.println(b);
}
```

Step (2/3)

To build a wireless motion controller, we need to...

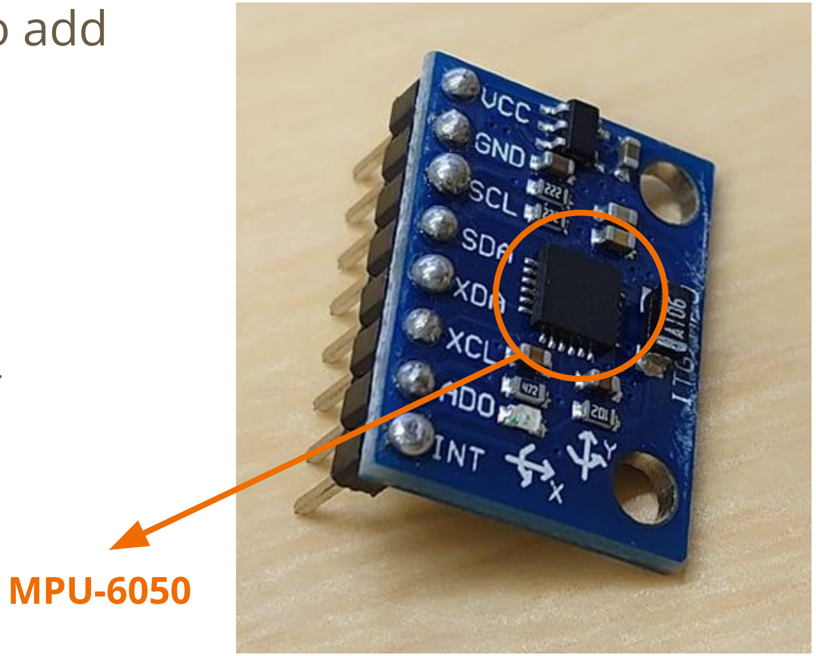
1. Detect whether the button is pressed.
2. **Get the pose data from the IMU sensor.**
3. System receives the data from the motion controller as user’s input.

IMU sensor - GY-521 (MPU-6050)

To get data from the IMU sensor, we need to add the following two libraries to Arduino.

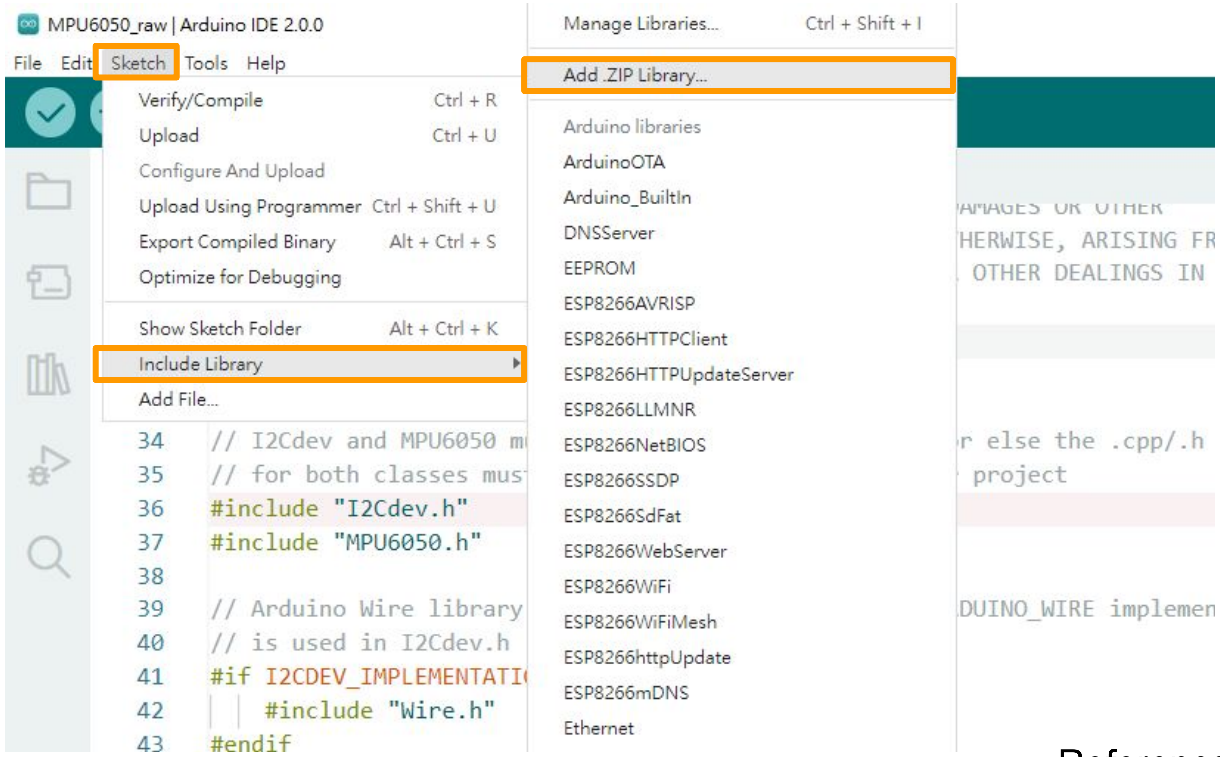
- I^{2}C protocol: **I2Cdev**
- IMU sensor: **MPU6050**

Download: NTUCOOL > 文件 > Lab > Lab03 > **I2Cdev.zip** and **MPU6050.zip**



Reference [1]: http://ming-shian.blogspot.com/2014/05/arduino21mpu6050row-data.html

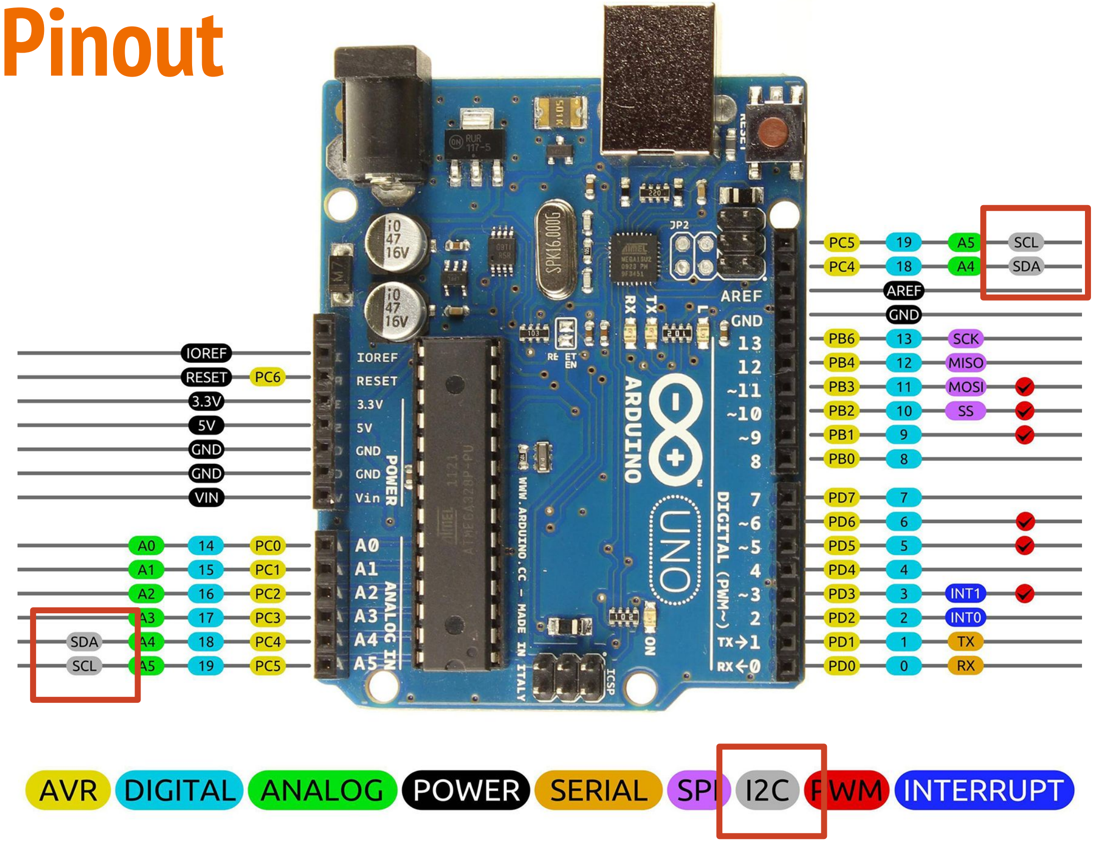
Install the Libraries



```
AMAGES OR OTHER
HERWISE, ARISING FR
 OTHER DEALINGS IN

// I2Cdev and MPU6050 m        r else the .cpp/.h
// for both classes mus         project
#include "I2Cdev.h"
#include "MPU6050.h"

// Arduino Wire library        DUINO_WIRE implemen
// is used in I2Cdev.h
#if I2CDEV_IMPLEMENTATI
    #include "Wire.h"
#endif
```

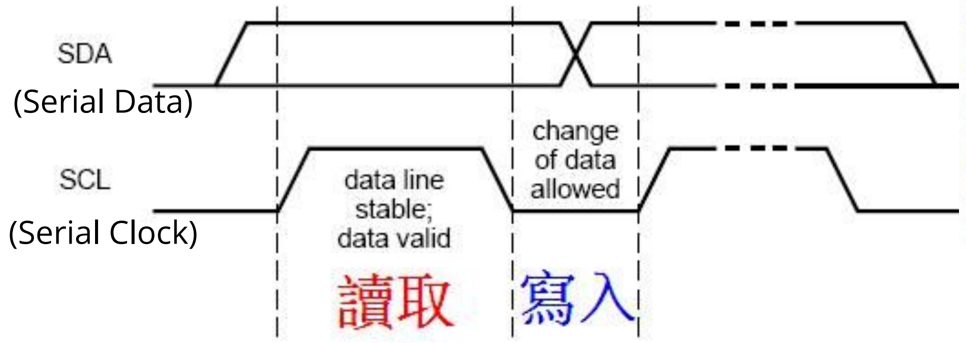
Reference: Installing Additional Libraries

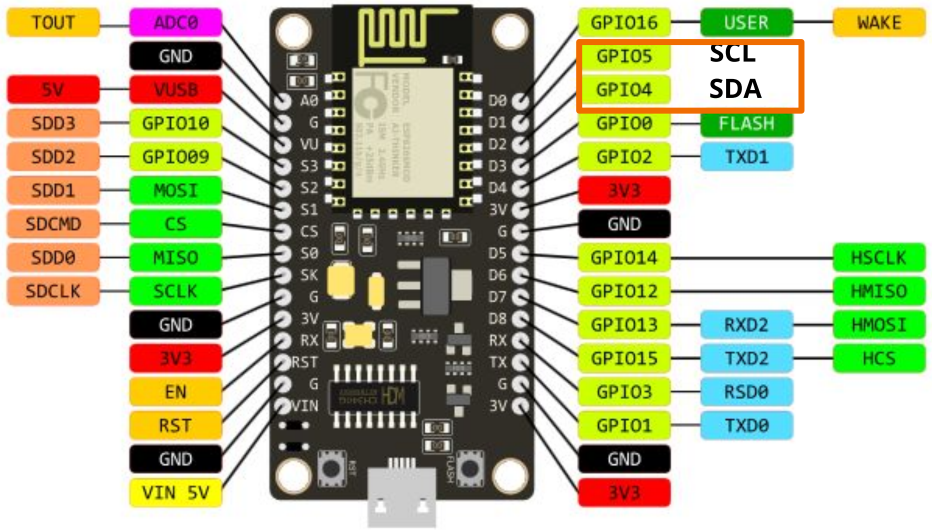
Arduino - Pinout



What is I²C?

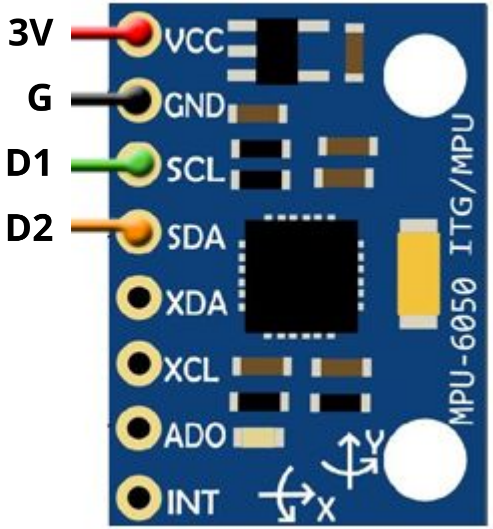
I²C (I2C) is a serial communication protocol.





For Arduino and NodeMCU, the default **SDA=GPIO4 and SCL=GPIO5.**

GY-521 Pinouts

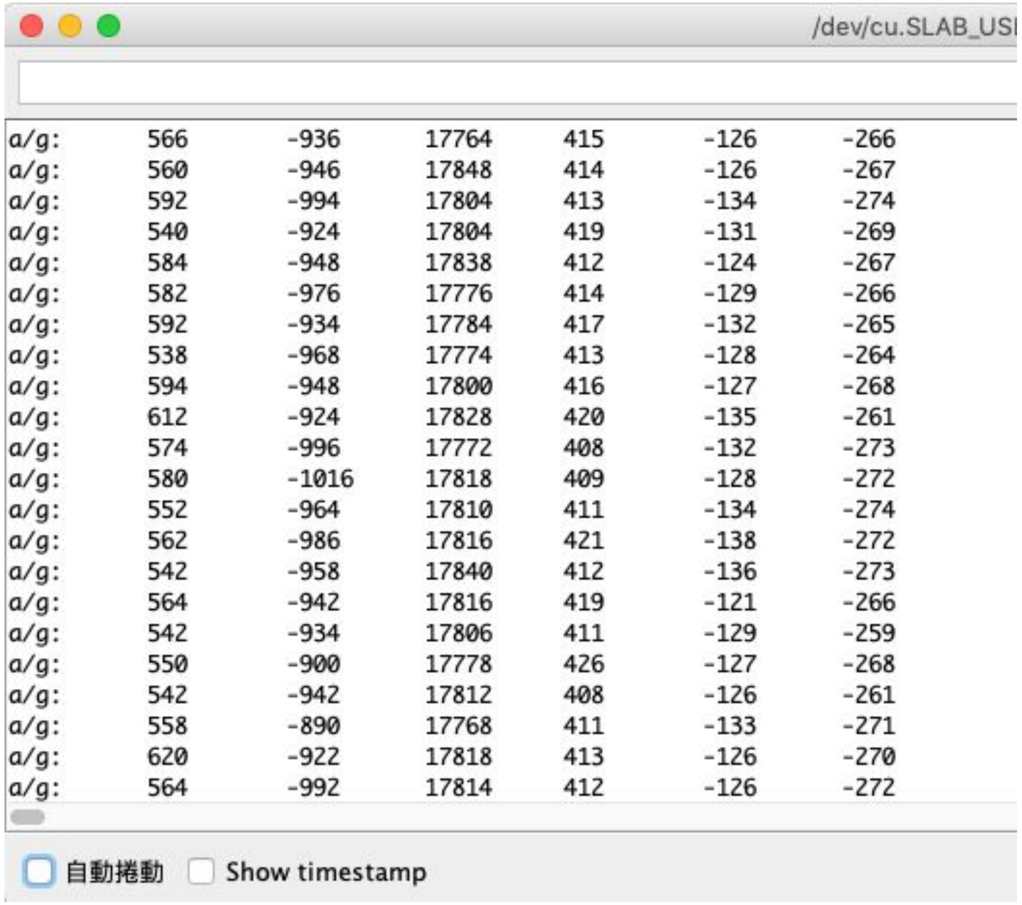


Time to implement!

Output (raw)

- a_x, a_y, a_z (acceleration) and ω_x, ω_y, ω_z (angular velocity)

Check **baud** setting if the output is weird or empty



```
a/g:    566    -936    17764   415    -126    -266
a/g:    560    -946    17848   414    -126    -267
a/g:    592    -994    17804   413    -134    -274
a/g:    540    -924    17804   419    -131    -269
a/g:    584    -948    17838   412    -124    -267
a/g:    582    -976    17776   414    -129    -266
a/g:    592    -934    17784   417    -132    -265
a/g:    538    -968    17774   413    -128    -264
a/g:    594    -948    17800   416    -127    -268
a/g:    612    -924    17828   420    -135    -261
a/g:    574    -996    17772   408    -132    -273
a/g:    580    -1016   17818   409    -128    -272
a/g:    552    -964    17810   411    -134    -274
a/g:    562    -986    17816   421    -138    -272
a/g:    542    -958    17840   412    -136    -273
a/g:    564    -942    17816   419    -121    -266
a/g:    542    -934    17806   411    -129    -259
a/g:    550    -900    17778   426    -127    -268
a/g:    542    -942    17812   408    -126    -261
a/g:    558    -890    17768   411    -133    -271
a/g:    620    -922    17818   413    -126    -270
a/g:    564    -992    17814   412    -126    -272
```

☐ 自動捲動 ☐ Show timestamp

NTU COOL > 文件 > Lab > Lab03 > ArduinoCode > **MPU6050_raw** > **MPU6050_raw.ino**

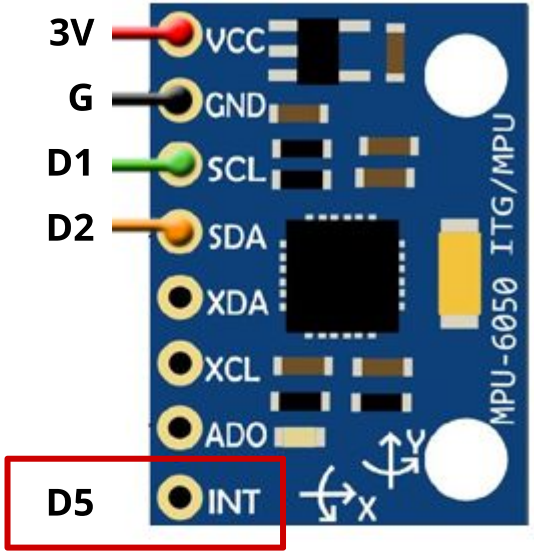
Question: How to process the raw data?

DMP: Digital Motion Processor.
It is embedded in MPU6050 and can help to **process the raw data to readable data**.
The calculated data will be push into a queue and **an interrupt signal will be sent**. **(Need to assign a pin for reading the interrupt signal from INT)**

Reference [1]: https://hackmd.io/@csielee/HkSQOMX1b?type=view

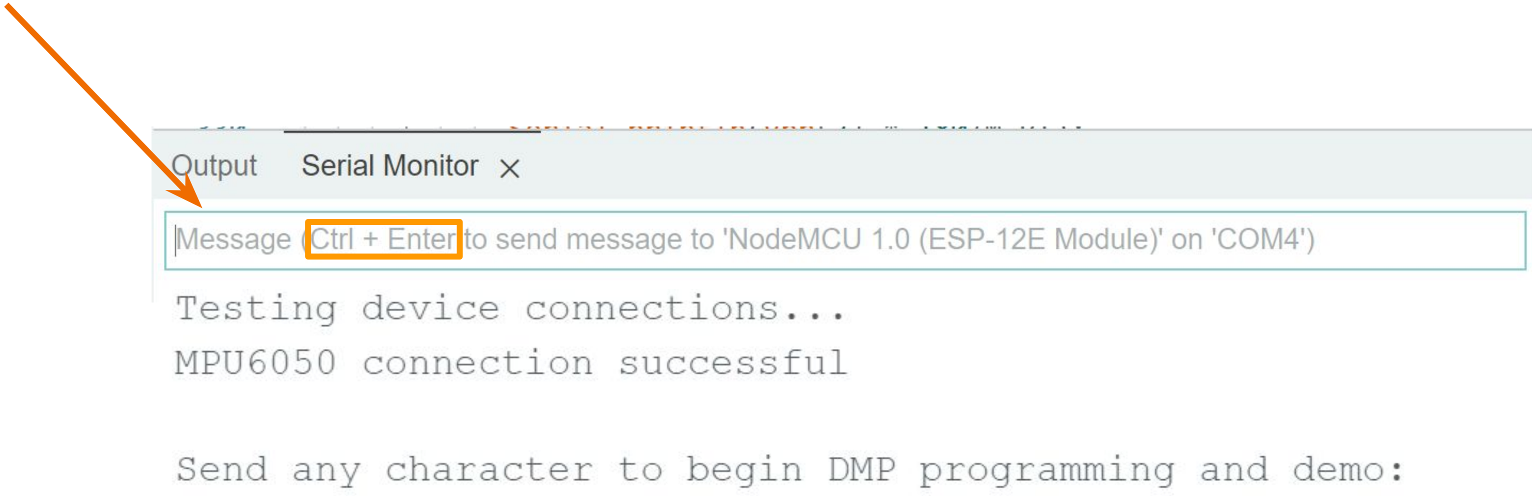
NTU COOL > 文件 > Lab > Lab03 > ArduinoCode > MPU6050_DMP6 > MPU6050_DMP6.ino

GY-521 Pinouts



NTU COOL > 文件 > Lab > Lab03 > ArduinoCode > MPU6050_DMP6 > MPU6050_DMP6.ino

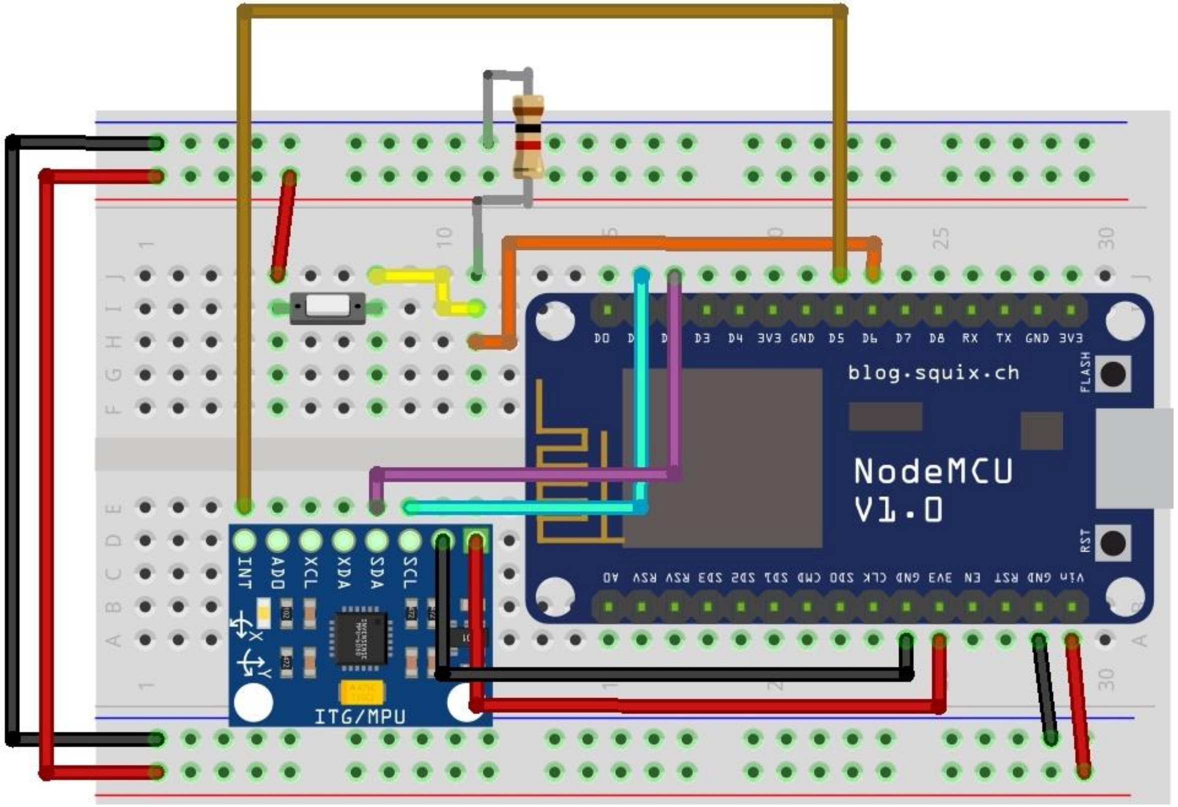
Note: Start DMP demo by sending any character in the serial monitor

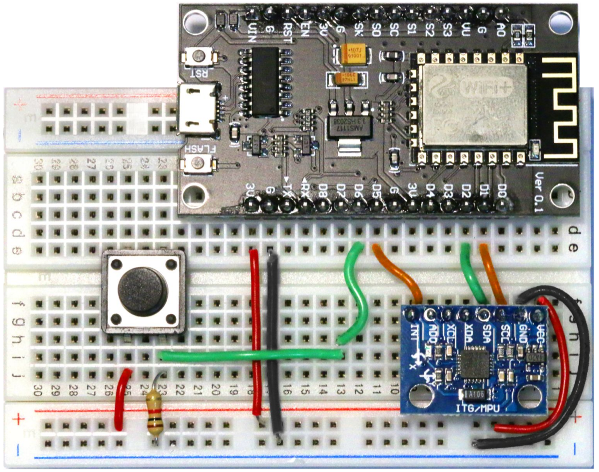


```
Message (Ctrl + Enter to send message to 'NodeMCU 1.0 (ESP-12E Module)' on 'COM4')

Testing device connections...
MPU6050 connection successful

Send any character to begin DMP programming and demo:
```

(Arduino 2.0.0)

Hardware wiring



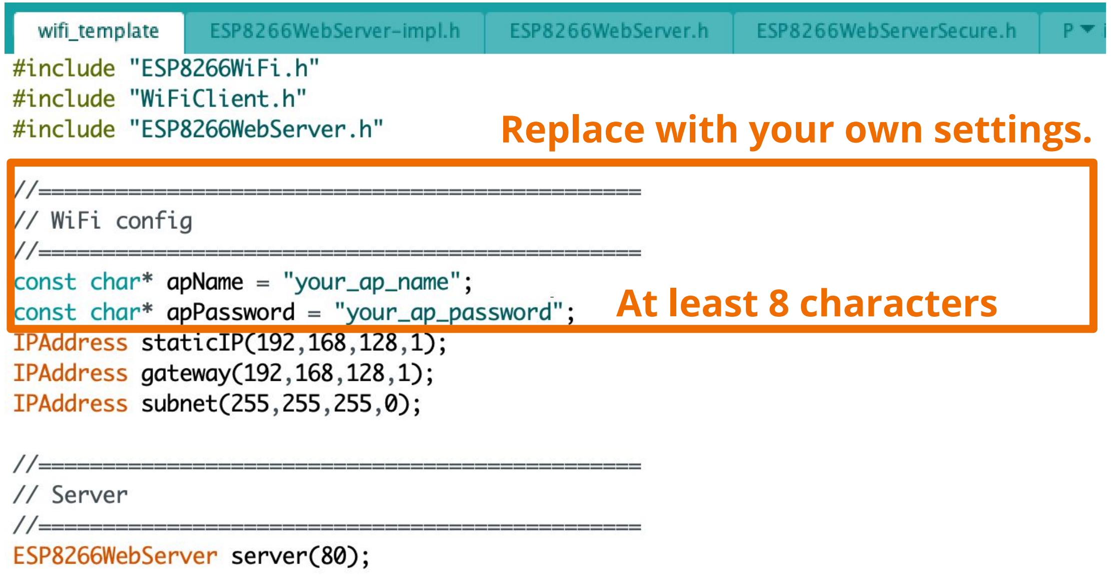


Step (3/3)

To build a wireless motion controller, we need to...

1. Detect whether the button is pressed.
2. Get the pose data from the IMU sensor.
3. **System receives the data from the motion controller as user’s input.**

Build a local network with your NodeMCU



```
#include "ESP8266WiFi.h"
#include "WiFiClient.h"
#include "ESP8266WebServer.h"

//================================================
// WiFi config
//================================================
const char* apName = "your_ap_name";
const char* apPassword = "your_ap_password";
IPAddress staticIP(192,168,128,1);
IPAddress gateway(192,168,128,1);
IPAddress subnet(255,255,255,0);

//================================================
// Server
//================================================
ESP8266WebServer server(80);
```

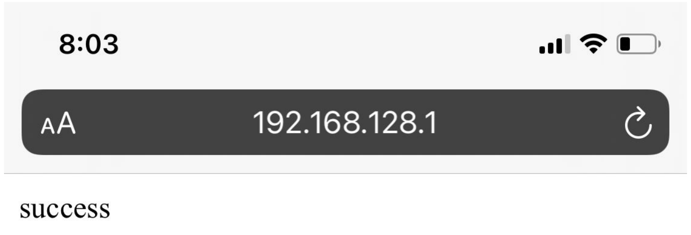
Replace with your own settings.

At least 8 characters

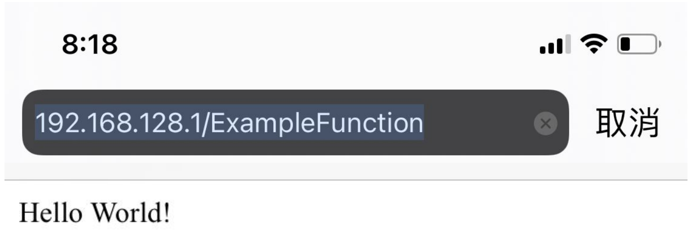
NTU COOL > 文件 > Lab > Lab03 > ArduinoCode > wifi_template > wifi_template.ino

Build a local network with your NodeMCU

Use another device connected to the same LAN, then you can access the web server on NodeMCU with its IP

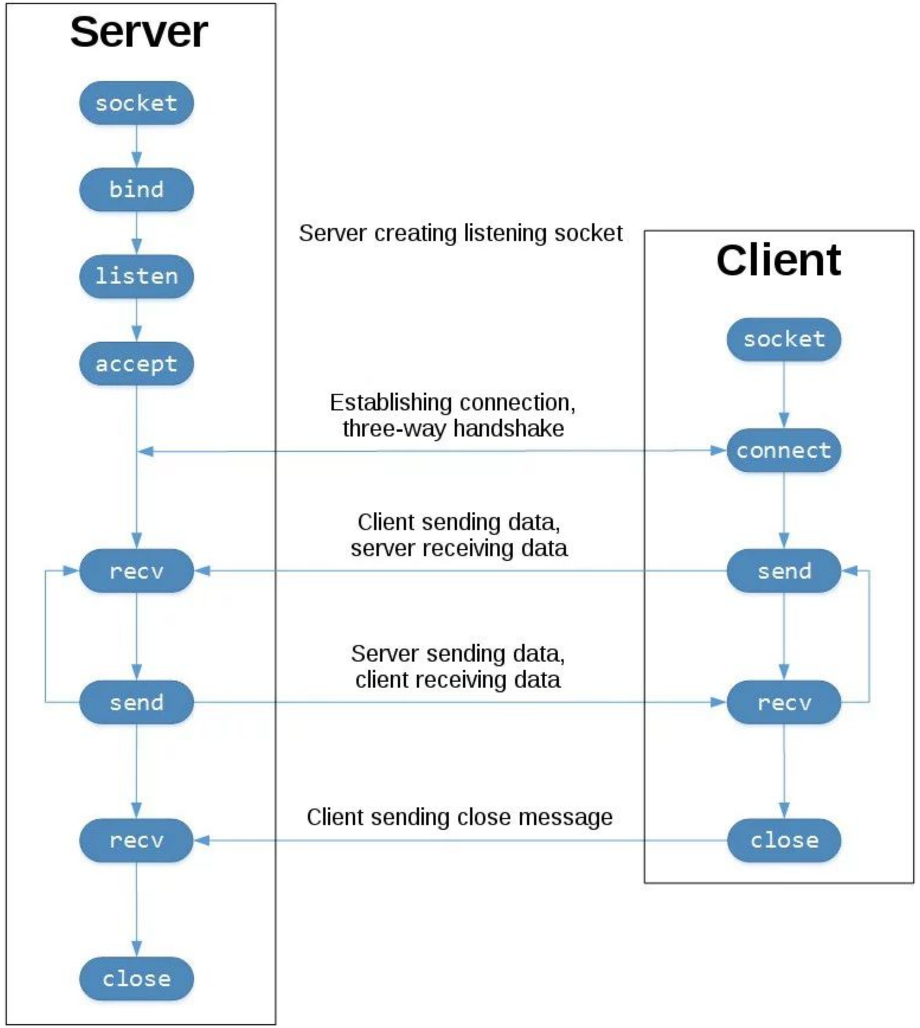


```
success
```

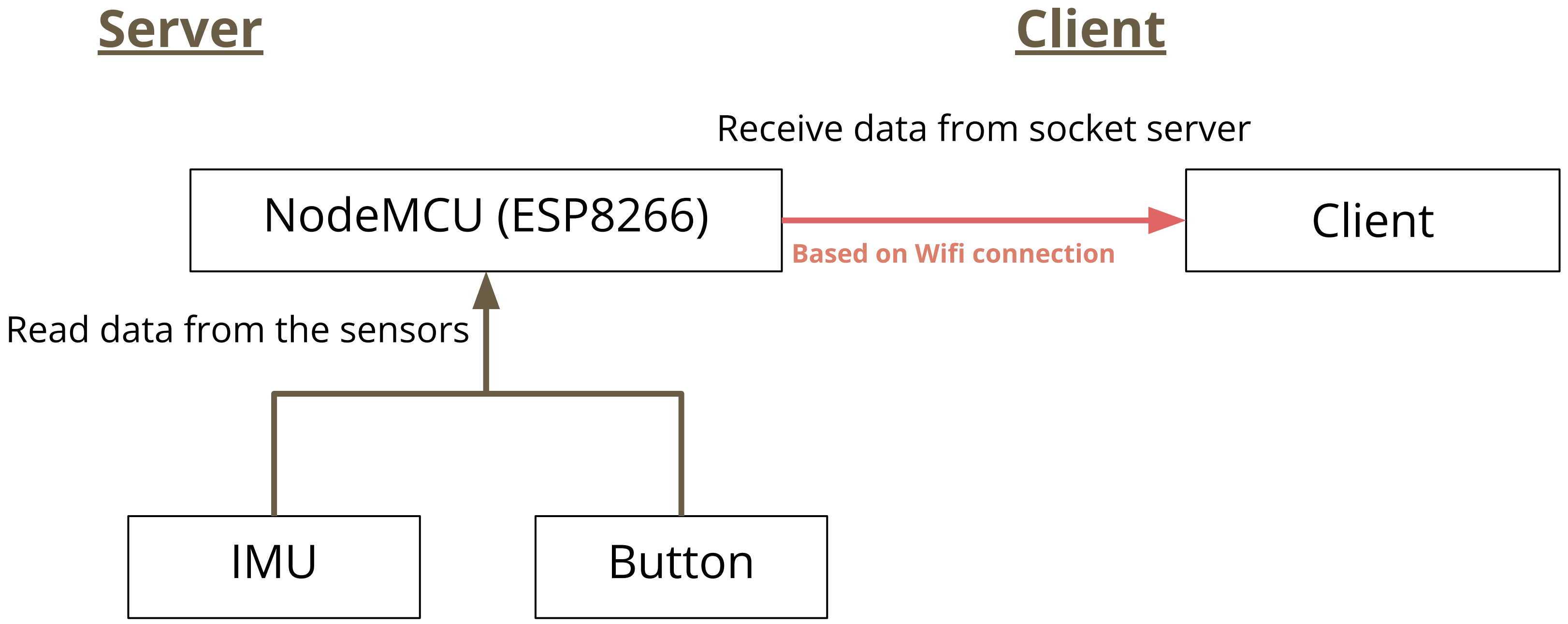


```
Hello World!
```

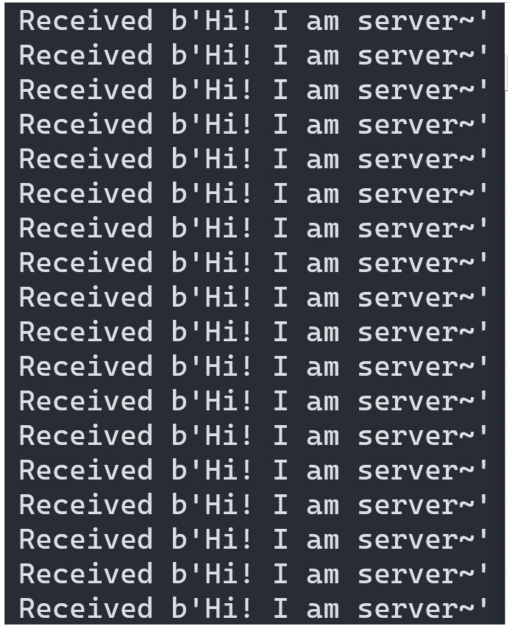
Socket


Server
socket
bind
listen
accept
recv
send
recv
close
Server creating listening socket
Client
socket
connect
send
recv
close
Establishing connection,
three-way handshake
Client sending data,
server receiving data
Server sending data,
client receiving data
Client sending close message

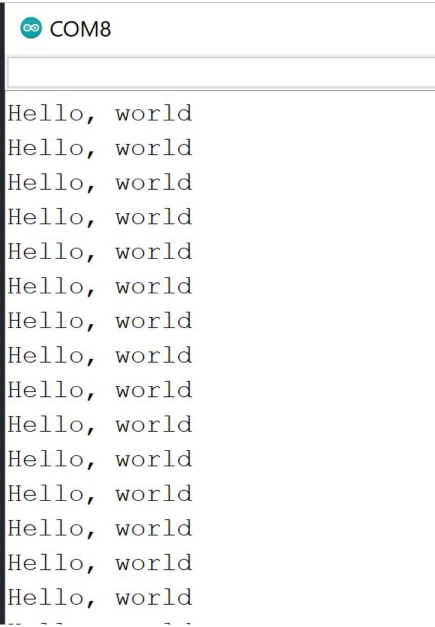
System Architecture



Time to implement!



```
Received b'Hi! I am server~'
Received b'Hi! I am server~'
Received b'Hi! I am server~'
Received b'Hi! I am server~'
Received b'Hi! I am server~'
Received b'Hi! I am server~'
Received b'Hi! I am server~'
Received b'Hi! I am server~'
Received b'Hi! I am server~'
Received b'Hi! I am server~'
Received b'Hi! I am server~'
Received b'Hi! I am server~'
Received b'Hi! I am server~'
Received b'Hi! I am server~'
Received b'Hi! I am server~'
Received b'Hi! I am server~'
Received b'Hi! I am server~'
Received b'Hi! I am server~'
```



```
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
```

Here are the sample codes for the system.

Test your code by running “socket_client_example.py”

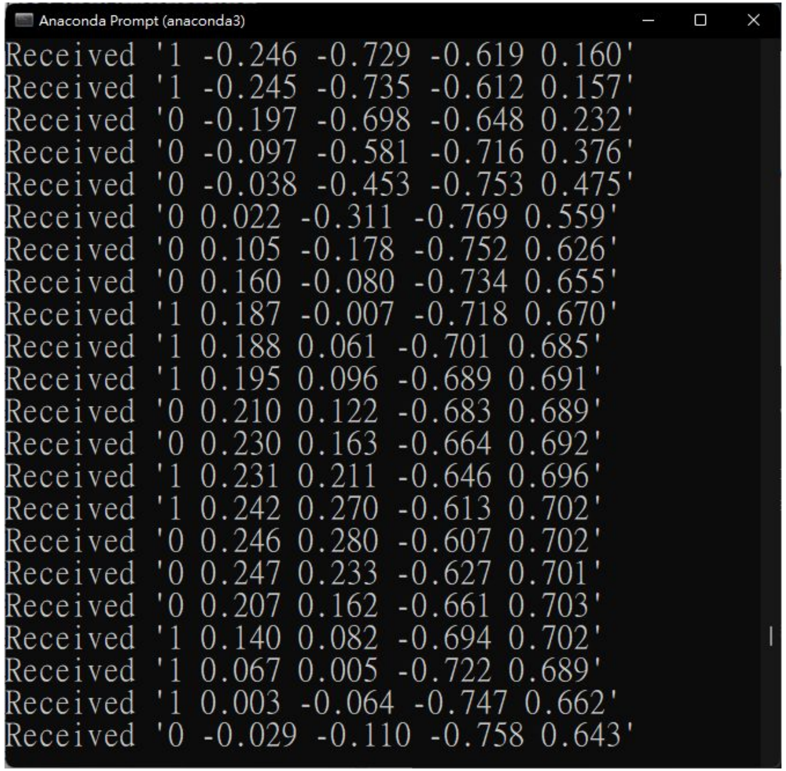
NTU COOL > 文件 > Lab > Lab03 > ArduinoCode > socket_server_example > socket_server_example.ino

NTUCOOL > 文件 > Lab > Lab03 > socket_client_example.py

Task to complete this week

Receive **the state of button (0/1)** and **IMU data (quaternion: *x, y, z, w*)** from NodeMCU then print them out in Python.

Hint: TCP sends data as a stream. You might need to put delimiter between messages / send the message length



```
Received '1 -0.246 -0.729 -0.619 0.160'
Received '1 -0.245 -0.735 -0.612 0.157'
Received '0 -0.197 -0.698 -0.648 0.232'
Received '0 -0.097 -0.581 -0.716 0.376'
Received '0 -0.038 -0.453 -0.753 0.475'
Received '0 0.022 -0.311 -0.769 0.559'
Received '0 0.105 -0.178 -0.752 0.626'
Received '0 0.160 -0.080 -0.734 0.655'
Received '1 0.187 -0.007 -0.718 0.670'
Received '1 0.188 0.061 -0.701 0.685'
Received '1 0.195 0.096 -0.689 0.691'
Received '0 0.210 0.122 -0.683 0.689'
Received '0 0.230 0.163 -0.664 0.692'
Received '1 0.231 0.211 -0.646 0.696'
Received '1 0.242 0.270 -0.613 0.702'
Received '0 0.246 0.280 -0.607 0.702'
Received '0 0.247 0.233 -0.627 0.701'
Received '0 0.207 0.162 -0.661 0.703'
Received '1 0.140 0.082 -0.694 0.702'
Received '1 0.067 0.005 -0.722 0.689'
Received '1 0.003 -0.064 -0.747 0.662'
Received '0 -0.029 -0.110 -0.758 0.643'
```

Next...

With the data from sensors, what can we do?

Let's make a kart game!

feedback