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

	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	Sensor介紹; 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.**

Last 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

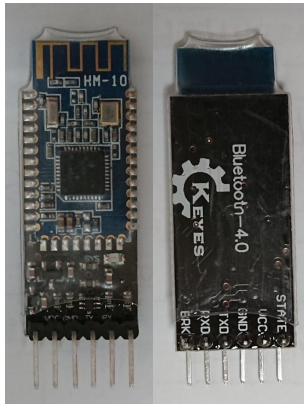
- MPU-6050 = Accelerometer + Gyroscope
  http://playground.arduino.cc/Main/MPU-6050

Accelerometer + gyroscope

This week

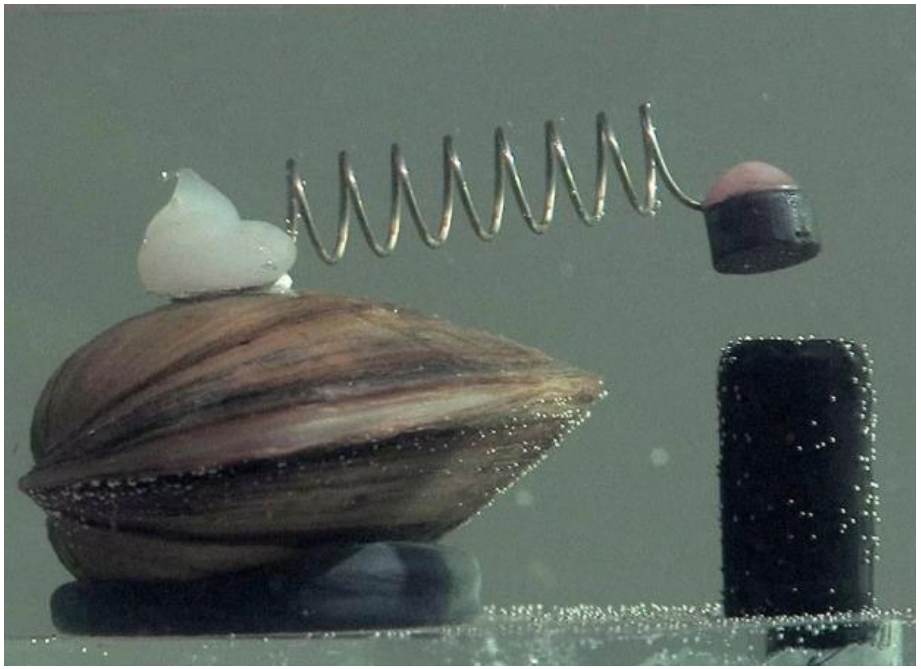
- Magnetometer (3-axis compass): 磁力計
  - I2C sensor
  - measure the earth's magnetic field in three axes
- Communication module: Bluetooth
  - Use Bluetooth to transmit data between Arduino and smart phone

HMC5883L



HM-10

Internet of Clams



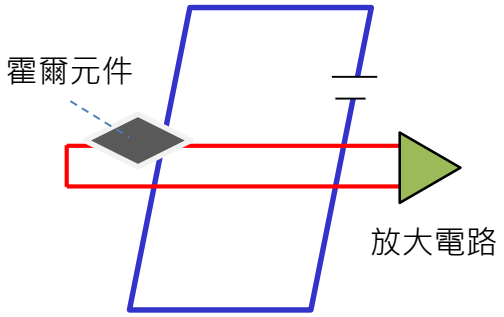
Clam Sensor Wiring Mississippi River Project

They are more commonly known as mucket clams or mucket mussels, and are particularly **sensitive to water pollution** — they will clam-up, so to speak, in the presence of contaminated water.

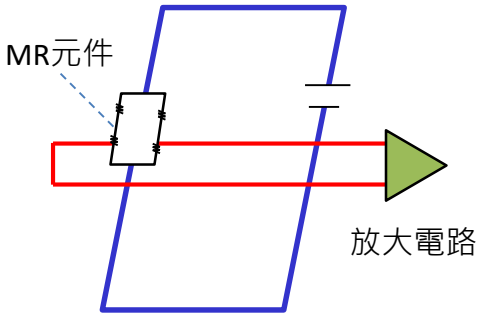
Lab1. HMC5003L

Magnetometer (compass), measure the earth’s magnetic field in three axes

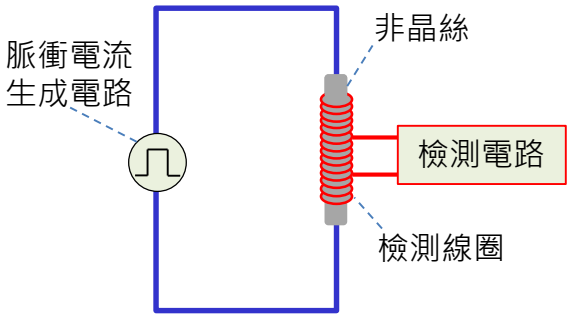
How does it work?



基於磁場變化
電壓隨著霍爾效應發生變化



基於磁場變化
MR元件的電阻發生變化



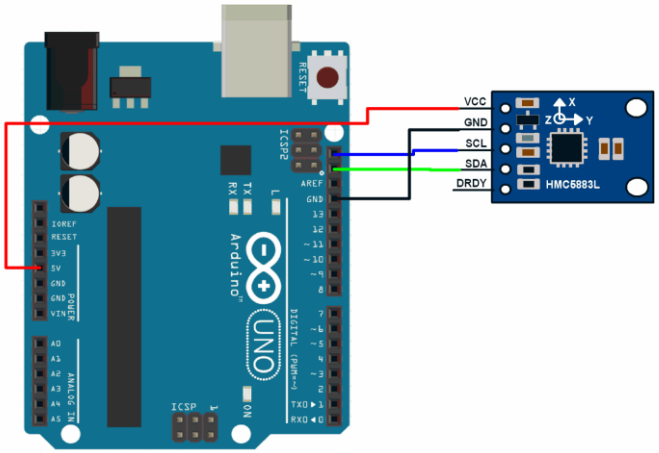
基於磁場變化
利用脈衝電流檢測線圈電壓

Magneto Resistance 磁阻效應感測器

cost compassing and magnetometry. The HMC5883L includes our state-of-the-art, high-resolution HMC118X series magneto-resistive sensors plus an ASIC containing amplification, automatic degaussing strap drivers, offset cancellation, and a 12-bit ADC that enables 1° to 2° compass heading accuracy. The I^2C

Magnetometer (HMC5883L)

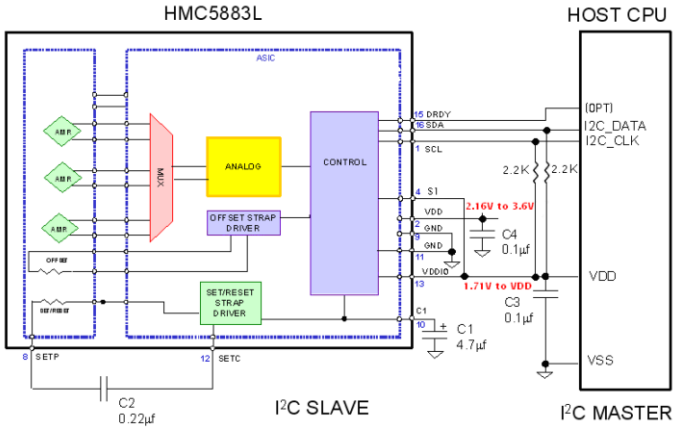
- Goal: Use CJ-M49/GY-271 sensor board with HMC-5883L in a single chip to read the values for compass.
- Measures the **earth's magnetic field** in three axes, and can be used to find earth's **magnetic north**.
- Hardware Required:
  - Arduino Board
  - HMC-5883L



Arduino UNO: SDA(pin A4), SCL(pin A5)

Magnetometer (HMC5883L)

- Measures the **earth's magnetic field** in three axes, and can be used to find earth's magnetic north.



Characteristics	Conditions*	Min	Typ	Max	Units
Field Range	Full scale (FS)	-8		+8	gauss
Mag Dynamic Range	3-bit gain control	±1		±8	gauss
Sensitivity (Gain)	VDD=3.0V, GN=0 to 7, 12-bit ADC	230		1370	LSb/gauss

https://cdn-shop.adafruit.com/datasheets/HMC5883L_3-Axis_Digital_Compass_IC.pdf

Magnetometer (HMC5883L)

- HMC5883L procedure

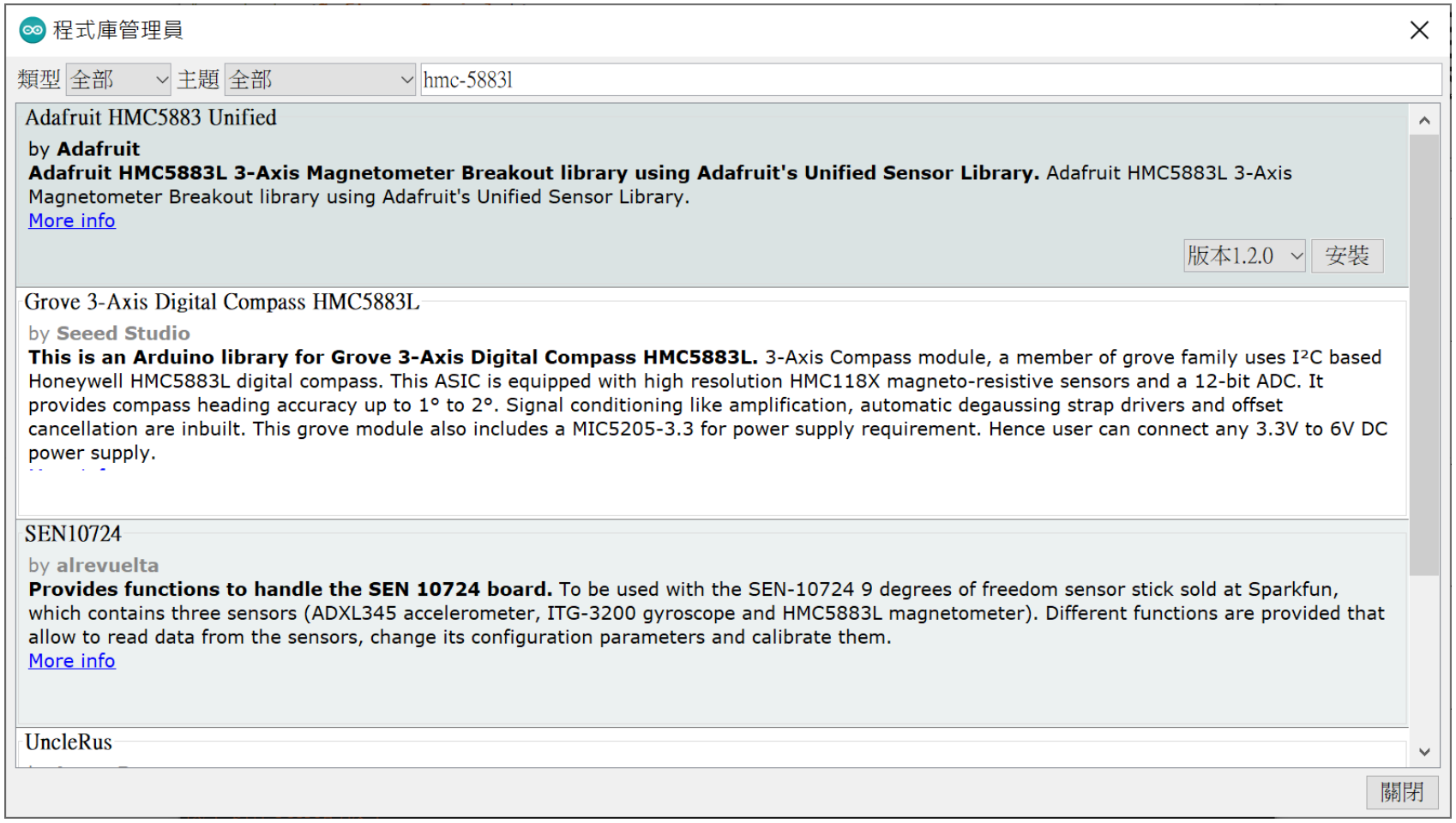
Below is an example of a (power-on) initialization process for “continuous-measurement mode”:

1. Write CRA (00) – send **0x3C 0x00 0x70** (8-average, 15 Hz default, normal measurement)
2. Write CRB (01) – send **0x3C 0x01 0xA0** (Gain=5, or any other desired gain)
3. Write Mode (02) – send **0x3C 0x02 0x00** (Continuous-measurement mode)
4. Wait 6 ms or monitor status register or DRDY hardware interrupt pin
5. Loop
   Send **0x3D 0x06** (Read all 6 bytes. If gain is changed then this data set is using previous gain)
   Convert three 16-bit 2’s compliment hex values to decimal values and assign to X, Z, Y, respectively.
   Send **0x3C 0x03** (point to first data register 03)
   Wait about 67 ms (if 15 Hz rate) or monitor status register or DRDY hardware interrupt pin

End_loop

The default (factory) HMC5883L 8-bit slave address is 0x3C for write operations, or 0x3D for read operations.

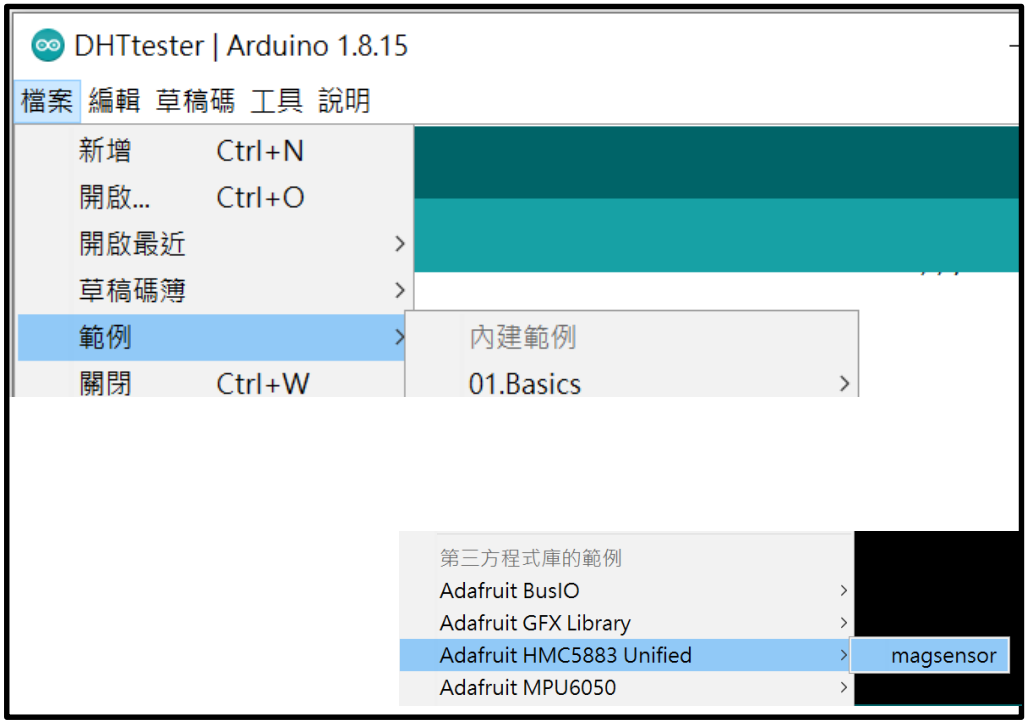
Library manager



Lab1. HMC5003L

Arduino IDE

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



magsensor code

```
void loop(void)
{
  /* Get a new sensor event */
  sensors_event_t event;
  mag.getEvent(&event);

  /* Display the results (magnetic vector values are in micro-Tesla (uT)) */
  Serial.print("X: "); Serial.print(event.magnetic.x); Serial.print("  ");
  Serial.print("Y: "); Serial.print(event.magnetic.y); Serial.print("  ");
  Serial.print("Z: "); Serial.print(event.magnetic.z); Serial.print("  ");Serial.println("uT");

  // Hold the module so that Z is pointing 'up' and you can measure the heading with x&y
  // Calculate heading when the magnetometer is level, then correct for signs of axis.
  float heading = atan2(event.magnetic.y, event.magnetic.x);
```

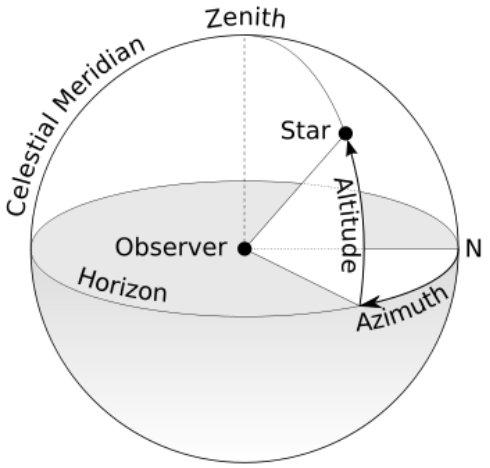
```
 // Once you have your heading, you must then add your 'Declination Angle', which is the 'Error' of the magnetic field in your location.
 // Find yours here: http://www.magnetic-declination.com/
 // Mine is: -13* 2' W, which is ~13 Degrees, or (which we need) 0.22 radians
 // If you cannot find your Declination, comment out these two lines, your compass will be slightly off.
 float declinationAngle = 0.22;
 heading += declinationAngle;

 // Correct for when signs are reversed.
 if(heading < 0)
   heading += 2*PI;

 // Check for wrap due to addition of declination.
 if(heading > 2*PI)
   heading -= 2*PI;

 // Convert radians to degrees for readability.
 float headingDegrees = heading * 180/M_PI;

 Serial.print("Heading (degrees): "); Serial.println(headingDegrees);

 delay(500);
}
```

Heading

- How to estimate the heading?
  - 1. measure two orthogonal components of the magnetic vector (mx, my)
  - 2. calculate the heading as the arctangent of their ratio

$$\operatorname{atan2}(y,x) = \begin{cases} \arctan\left(\frac{y}{x}\right) & x > 0 \\ \arctan\left(\frac{y}{x}\right) + \pi & y \geq 0, x < 0 \\ \arctan\left(\frac{y}{x}\right) - \pi & y < 0, x < 0 \\ +\frac{\pi}{2} & y > 0, x = 0 \\ -\frac{\pi}{2} & y < 0, x = 0 \\ \text{undefined} & y = 0, x = 0 \end{cases}$$

https://en.wikipedia.org/wiki/Azimuth



An azimuth is an angular measurement in a spherical coordinate system

https://www.pveducation.org/pvcdrom/properties-of-sunlight/azimuth-angle

Declination angle

- The declination angle (磁傾角) can be calculated by the equation:
  - $\delta = -23.45^\circ \times \cos\frac{360}{365} \times d + 10$
  - where d is the day of the year with Jan 1 as d = 1

http://www.magnetic-declination.com/

Latitude: 24.4700° North
Longitude: 120.5900° East
Date: 2020-02-17

Magnetic declination: 4° 23.11' West

Annual Change (minutes/year): 4.8 '/y West

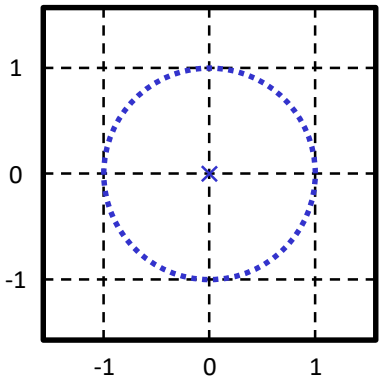
https://www.geomag.nrcan.gc.ca/calc/mdcal-en.php

https://www.pveducation.org/pvcdrom/properties-of-sunlight/declination-angle

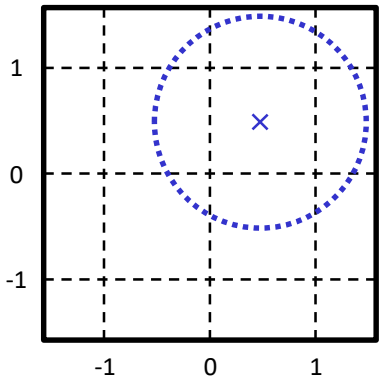
Compass Calibration

- Compass distortion
  - Hard Iron (硬磁干擾: 固定強度的磁干擾物)
  - Soft Iron (軟磁干擾: 會改變強度及方向, 可扭曲磁力線的干擾物)

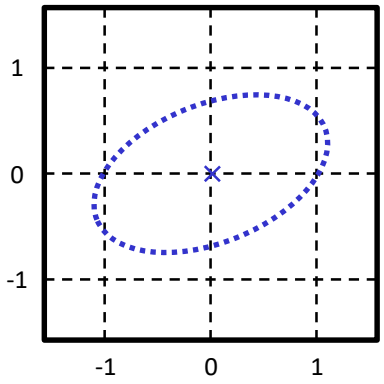
(Perfect)



No distortion

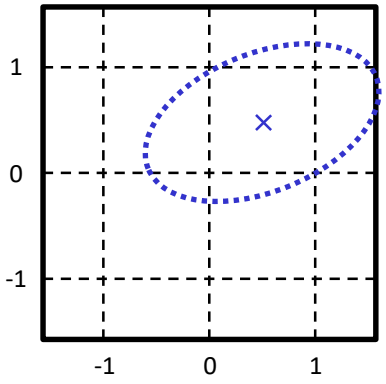


Hard Iron distortion



Soft Iron distortion

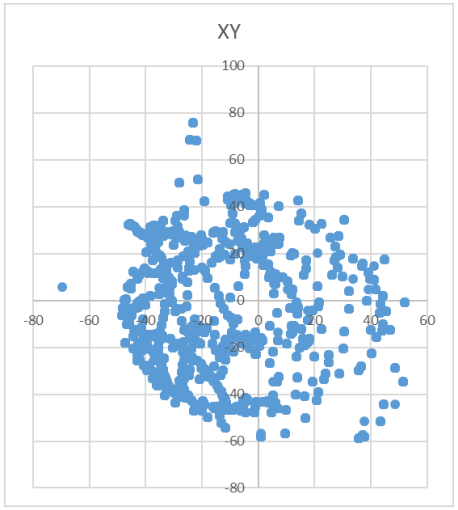
(Reality)

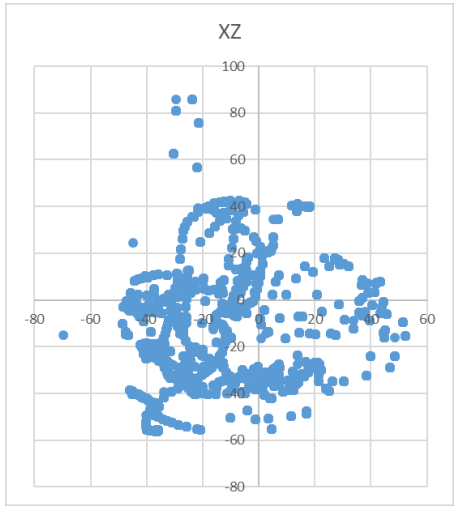


Soft & Hard Iron distortion

Discussion-1

- What is the current declination angle?
- Does your compass have distortion? Record the sensing value and draw it.
- What can we do to eliminate the distortion effect?





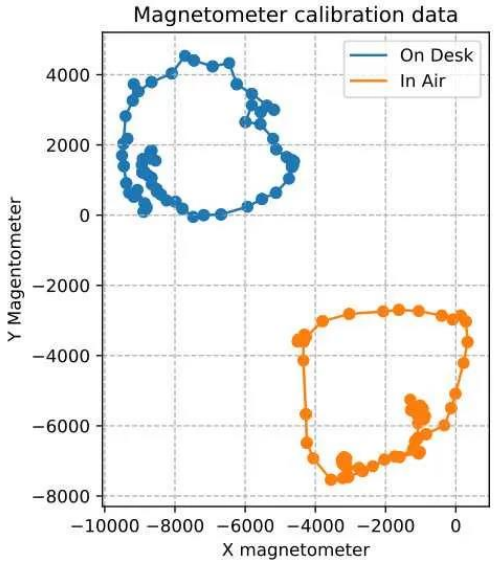
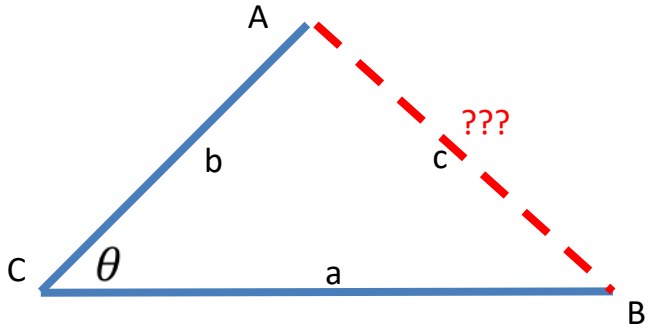


Fig source: https://www.best-microcontroller-projects.com/hmc5883l.html

Quiz 1

- Do you remember Trigonometric function or forget it?
- Compass can be used to design a distance estimator (the distance between A and B, as follows)
- Please use **ultrasonic** sensor to measure the distances of AC and BC, and use **magnetometer** to estimate the angle of ∠ACB.

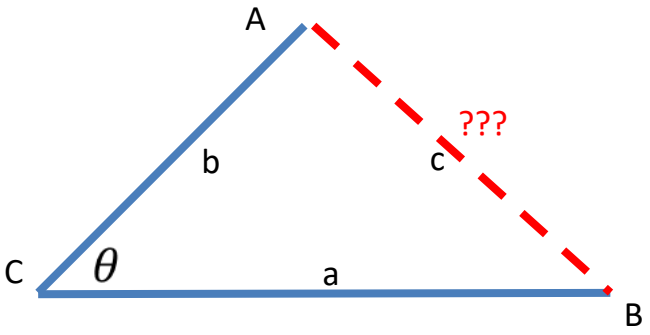
$$c^2 = a^2 + b^2 - 2ab\cos(\theta)$$

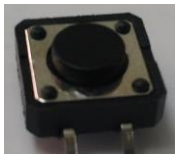


Hint: Law of cosines.
http://en.wikipedia.org/wiki/Law_of_cosines

Quiz 1 (cont.)

- Steps (you are in point C):
  1. Press button to measure the distance of AC
  2. Rotate from pointing to A to pointing to B
  3. Press button to measure the distance of BC
  4. Print out the distance between A and B





$$c^2 = a^2 + b^2 - 2ab\cos(\theta)$$

Lab2. Bluetooth

Use Bluetooth to transmit data between Arduino and smart phone

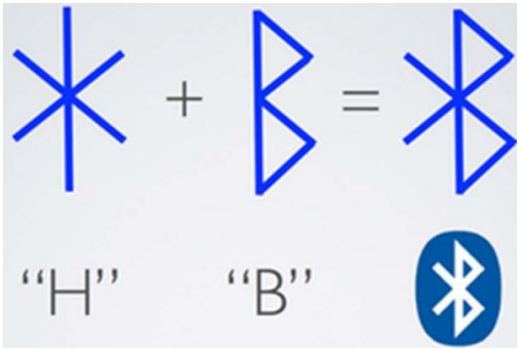
Bluetooth

- **Bluetooth** is a wireless technology standard used for exchanging data between fixed and mobile devices over short distances using UHF radio waves in ISM band, from 2.402 GHz to 2.480 GHz and building personal area networks (PANs).
- UHF (Ultra High Frequency) band: 300 MHz ~ 3 GHz
- ISM (Industrial Scientific Medical) band
  - ISM band is not unified in various countries.
  - License free
  - 2.4 GHz is unified ISM frequency among countries.

藍牙起源

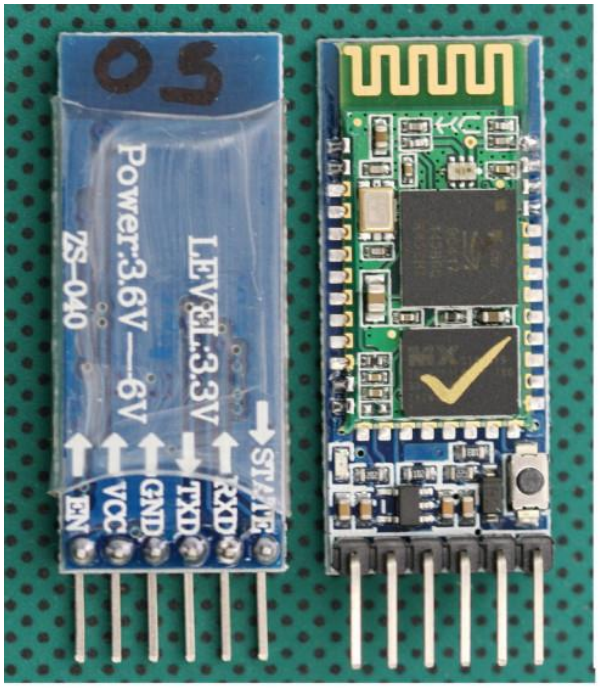
- 歷史
  - 十世紀國王的名字 (Harald Blåtand)
    - 統一了因宗教戰爭和領土爭議而分裂的挪威與丹麥而聞名於世
    - 喜歡吃藍莓，因此牙齒都變成藍色 (Blue tooth)
    - 另一說，他的牙齒很差，看起來像藍色(blue, dark, black)
    - 他喜歡穿藍色的服飾，當時的藍色有昂貴、尊爵、不凡的意思
  - 由 Ericsson 在 1994 年創製 ,希望為裝置間的通訊創造一組統一規則（標準化協定），以解決用戶間互不相容的移動電子裝置

不要寫成藍**芽**喔!

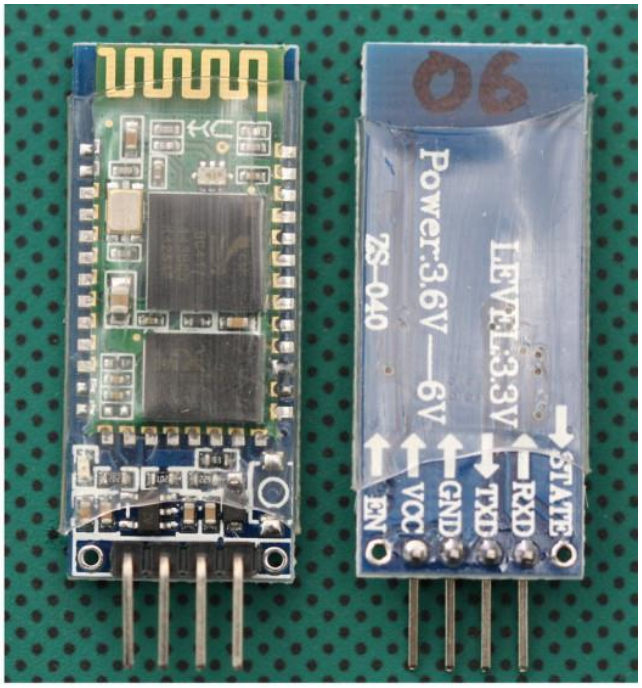


https://zh.wikipedia.org/zh-tw/%E8%97%8D%E7%89%99
https://en.wikipedia.org/wiki/Harald_Bluetooth

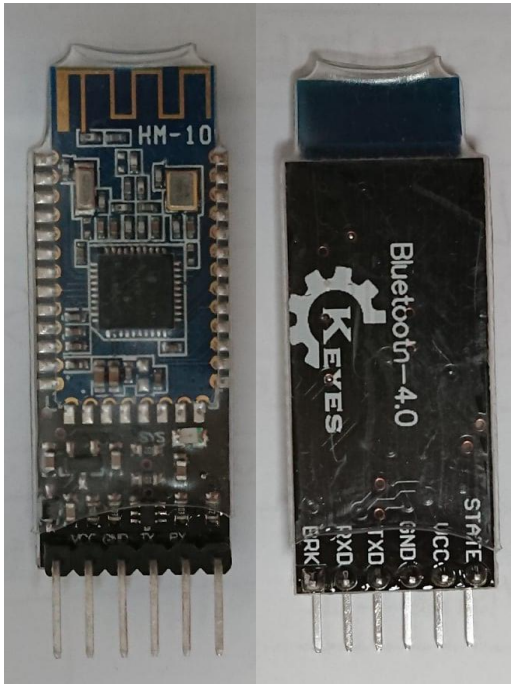
Bluetooth modules



HC-05



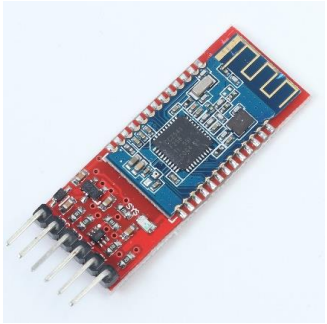
HC-06

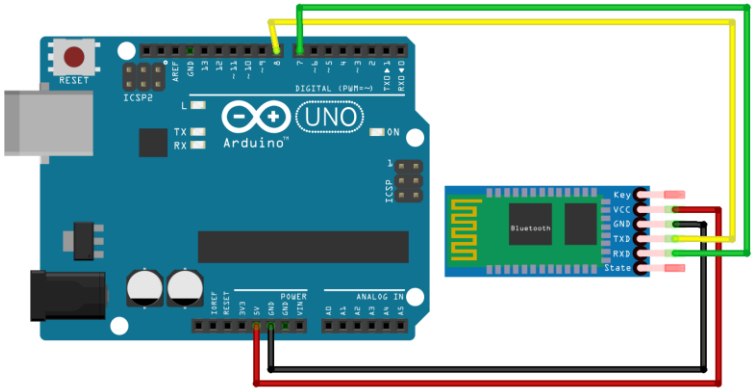


HM-10

Lab2. Bluetooth

- Goal: Use Bluetooth to transmit data between master and slave
- Hardware Required
  - Arduino
  - HM-10 Bluetooth





Sample code

```
#include <SoftwareSerial.h>

unsigned long BAUD_RATE = 9600;

const int BLUETOOTH_TX = 8;
const int BLUETOOTH_RX = 7;

const int READ_TIME = 500;
unsigned long prevMillis;

SoftwareSerial bluetooth(BLUETOOTH_TX, BLUETOOTH_RX);

void setup(){
  Serial.begin(BAUD_RATE);
  bluetooth.begin(BAUD_RATE);
  Serial.println("Setup Complete");
}
```

The SoftwareSerial library has been developed to allow serial communication on other digital pins of the Arduino, using software to replicate the functionality (hence the name "SoftwareSerial"). It is possible to have multiple software serial ports with speeds up to 115200 bps. A parameter enables inverted signaling for devices which require that protocol.

https://www.arduino.cc/en/Reference/softwareSerial

```
void loop(){
 if (Serial.available()) {              // 是否有來自Serial monitor的資料
  String str ="";
  Serial.print("Input: ");
  prevMillis = millis();
  while (millis() - prevMillis < READ_TIME) {
   if (Serial.available()) {
    char c = Serial.read();
    if (c != 10 && c != 13) {          // 檢查換行符號, 包含 \r (ASCII 13) 與 \n (ASCII 10)
     str += c;
 } } }
 bluetooth.print(str);                 // 將資料傳到藍牙的TX pin
 Serial.println(str);                  // 在serial monitor顯示資料
 }

 if (bluetooth.available()) {          // 是否有來自藍牙模組的資料
  String str = "";
  Serial.print("HM10: ");
  prevMillis = millis();
  while (millis() - prevMillis < READ_TIME) {
   if (bluetooth.available()) {
    str += (char) bluetooth.read()              // 讀取來自藍牙RX pin的資料
  } }
 Serial.println(str);                  // 在serial monitor顯示資料
} }
```

Syntax

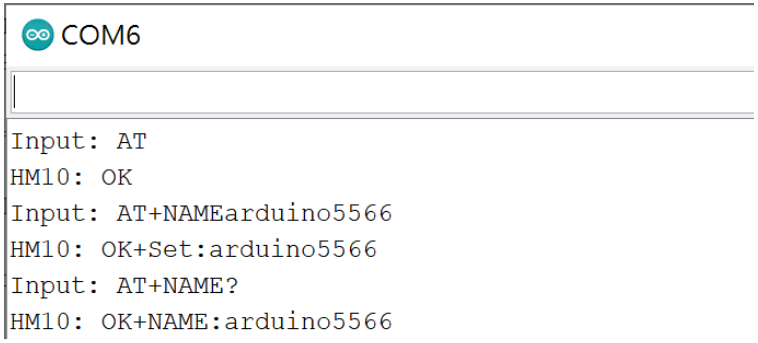
- Syntax
  - SoftwareSerial(rxPin, txPin, inverse_logic)
- Description
  - SoftwareSerial is used to create an instance of a SoftwareSerial object.
- Parameters
  - rxPin: the pin on which to receive serial data
  - txPin: the pin on which to transmit serial data
  - inverse_logic: is used to invert the sense of incoming bits (the default is normal logic)

Syntax

- SoftwareSerial: print(data)
  - Prints data to the transmit pin of the software serial port. Works the same as the Serial.print() function.
  - https://www.arduino.cc/en/Reference/SoftwareSerialPrint
- SoftwareSerial: read
  - Return a character that was received on the RX pin of the software serial port.
  - https://www.arduino.cc/en/Reference/SoftwareSerialRead

What can we do?

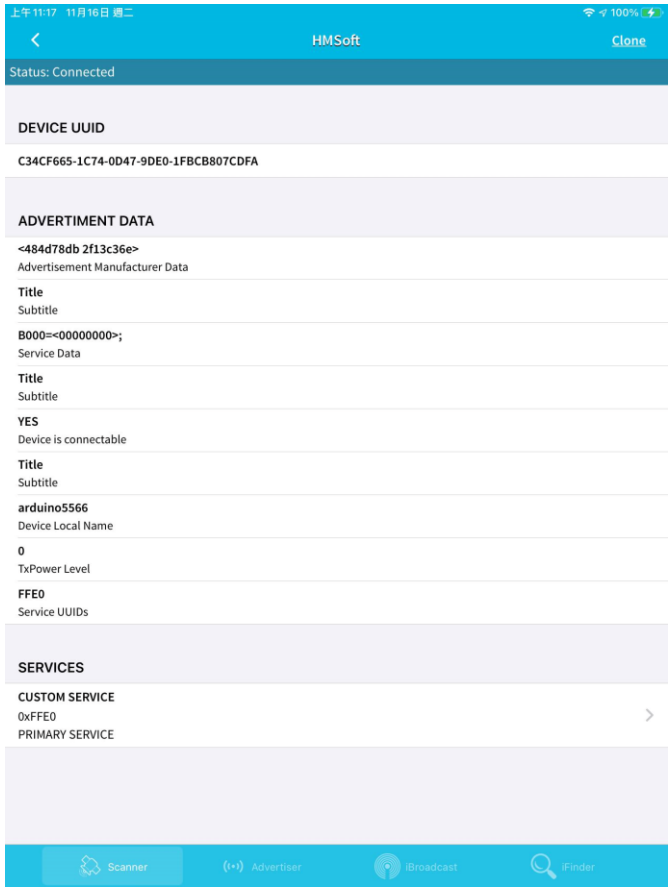
- AT commands
  - AT // Check if the command terminal work normally
  - AT+RENEW // Restore all setup value to factory setup
  - AT+RESET // Restart module
  - AT+NAMEarduino5566 // Set Module name to arduino5566
  - AT+NAME? // Query Module name
  - AT+ADDR? // Query module MAC address

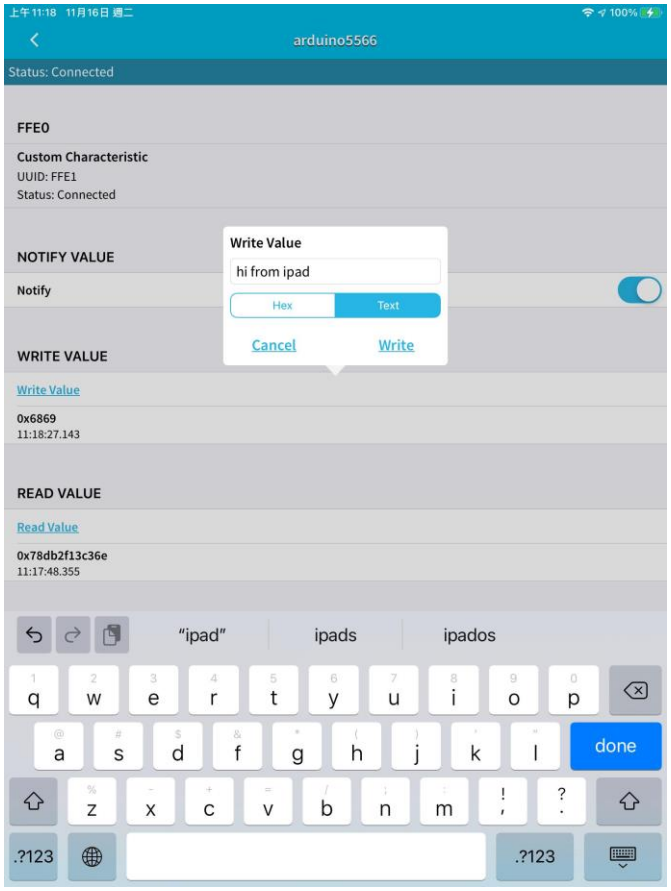


https://people.ece.cornell.edu/land/courses/ece4760/PIC32/uart/HM10/DSD%20TECH%20HM-10%20datasheet.pdf

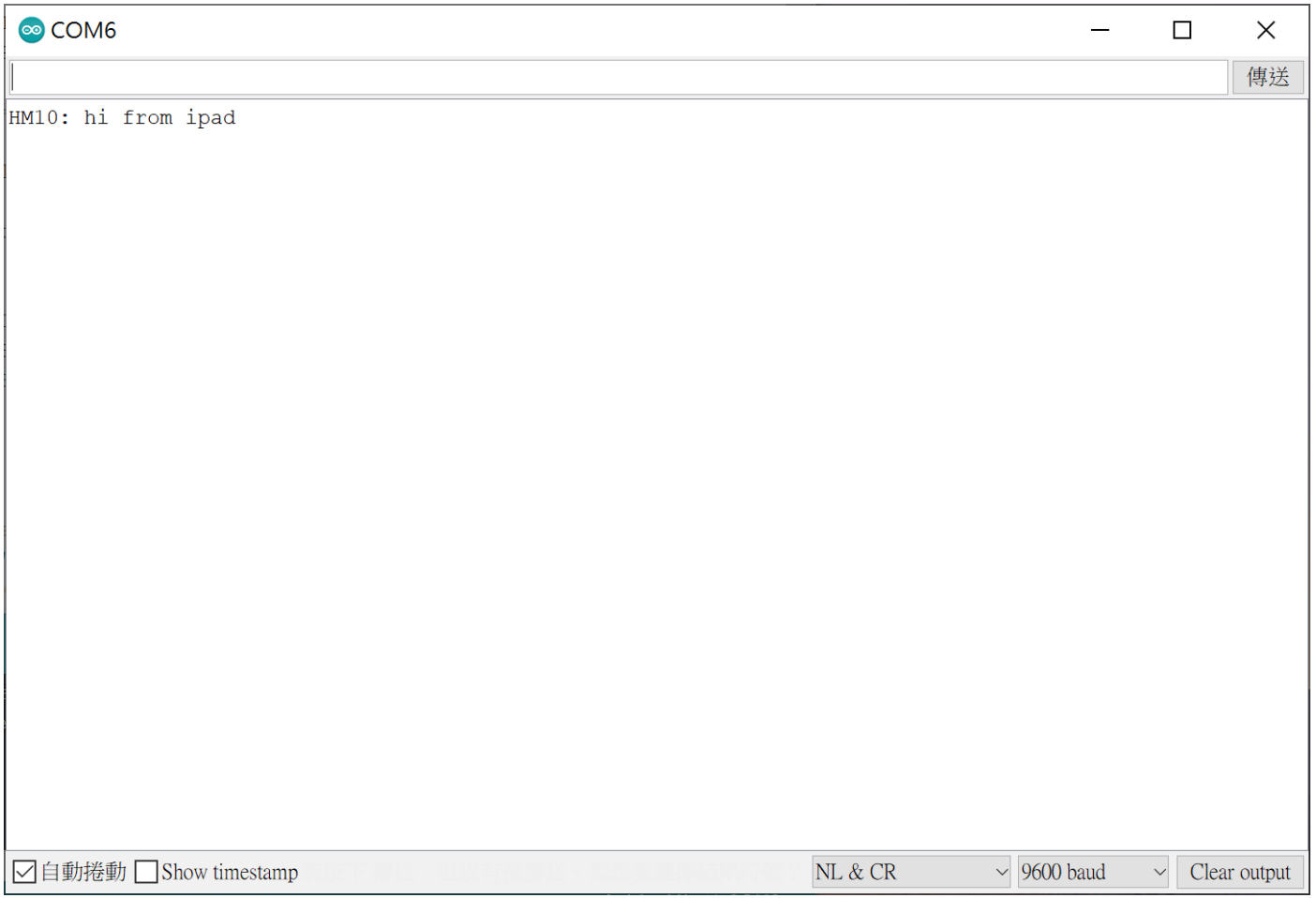
Smartphone & HM-10

- Install BLE terminal (ex: lightblue)





Smartphone & HM-10 (cont.)



Quiz 2

- Use AT command to set your HM-10
  - Change module name to your student ID
  - Query your module MAC address
- Use your smartphone, send control message to HM-10, then Arduino will execute specific actions
- Ex:
  - Send “on” => turn on LED
  - Send “off” => turn off LED
  - Send “whereau” => report heading

Summary

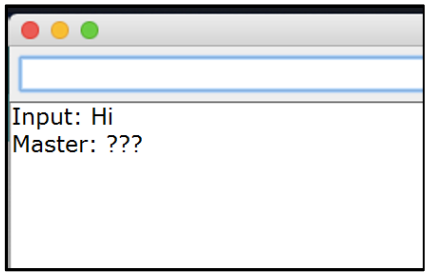
Summary

- **“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. Design a distance estimator
  - Quiz 2. send control message to HM-10

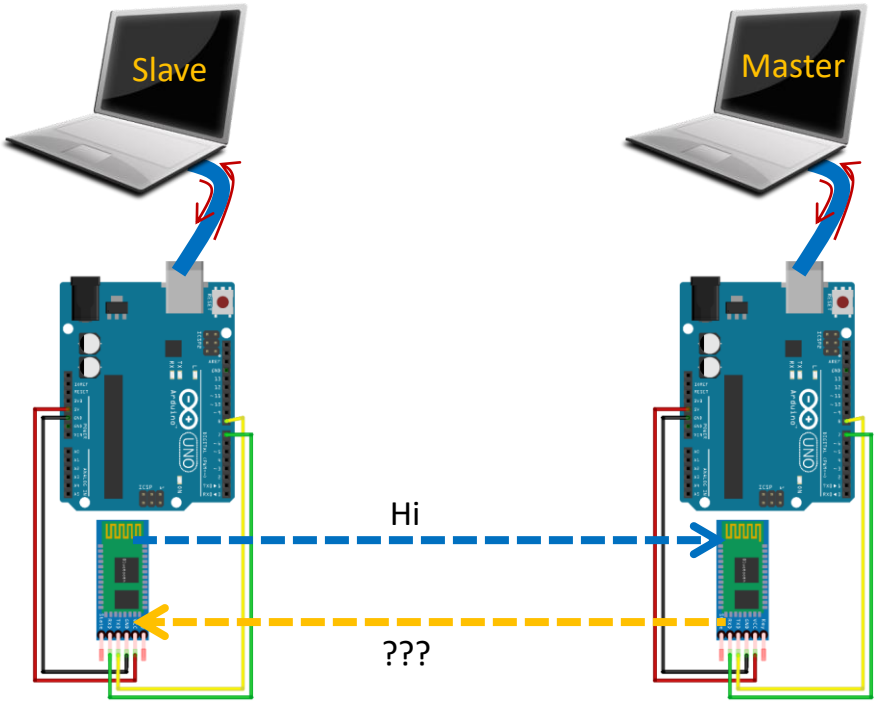
Appendix. Bluetooth

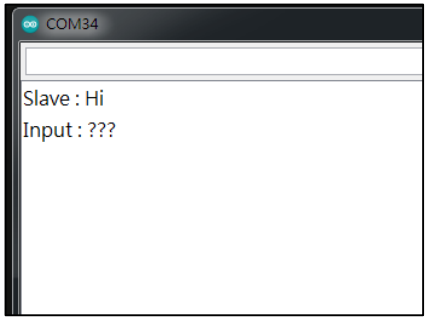
Two Arduino + HM-10

Bluetooth



Slave





Master

BT master and slave

Communication between master and slave

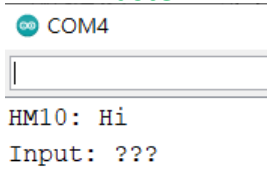
Master:

- Restore all setup value to factory
  - AT+RENEW (HM10: OK+RENEW)
- RESET (AT+RESET)
  - AT+RESET (HM10: OK+RESET)
- Query/Set Module work type
  - AT+IMME1(HM10: OK+Set:1)
- Set Master and Slaver Role
  - AT+ROLE1 (設定成Central)
- Start a device discovery scan
  - AT+DISC? (HM10:OK+DIS0:0CB2B7787E3AOK+RSSI:-042)
  - Copy the salve address (0CB2B7787E3A)
- Try connect an address
  - AT+CON0CB2B7787E3A (HM10:OK+CONNAOK+CONN)

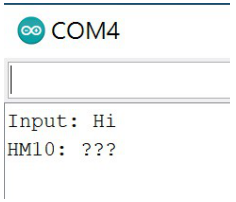
Salve

- Restore all setup value to factory
  - AT+RENEW
- RESET (AT+RESET)
  - AT+RESET
- Query/Set Module work type
  - AT+IMME0
- Set Master and Slaver Role
  - AT+ROLE0 (HM10:設定成Peripheral)
- Query module address
  - AT+ADDR? (HM10:OK+ADDR:0CB2B7787E3A)

Master



Slave

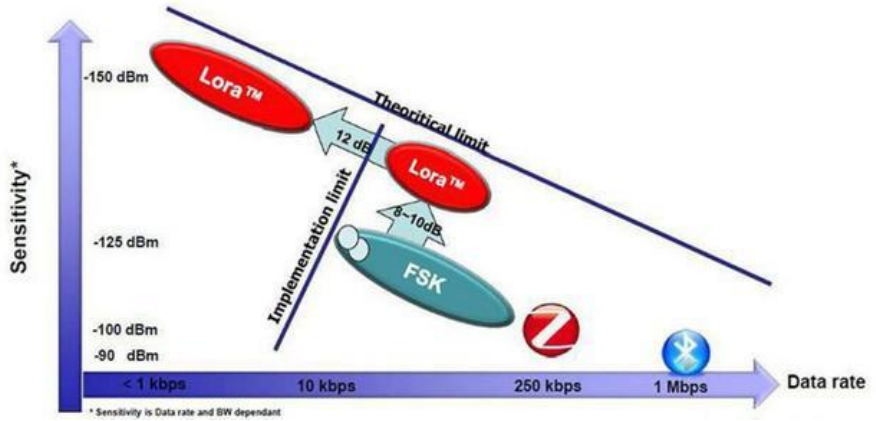


Appendix. LoRa

LoRa

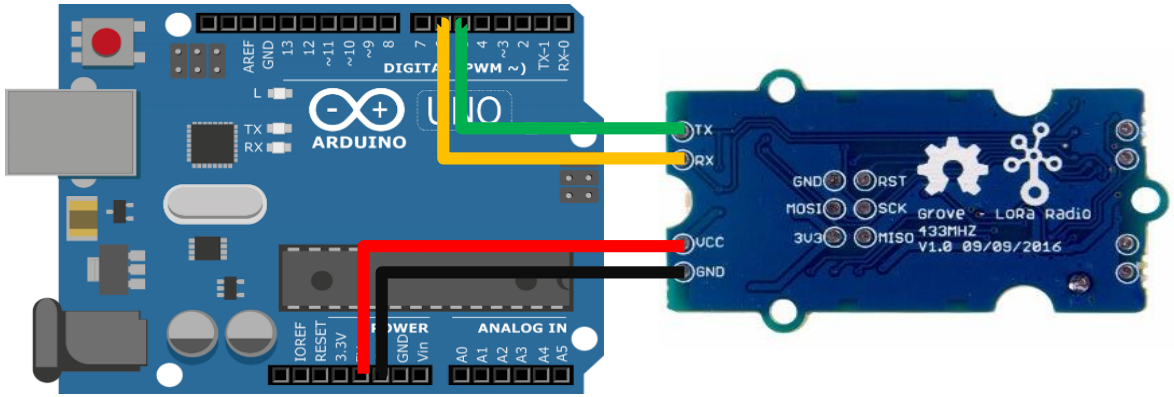
- **LoRa** is an acronym for **Lo**ng **Ra**nge using uses license-free sub-gigahertz radio frequency bands like 433 MHz, 868 MHz (Europe), 915 MHz (Australia and North America), 865 MHz to 867 MHz (India) and 923 MHz (Asia)
- Long distance: LoRa enables long-range transmissions (more than 10 km in rural areas)
- Low power consumption

https://wiki.seeedstudio.com/cn/Grove_LoRa_Radio/



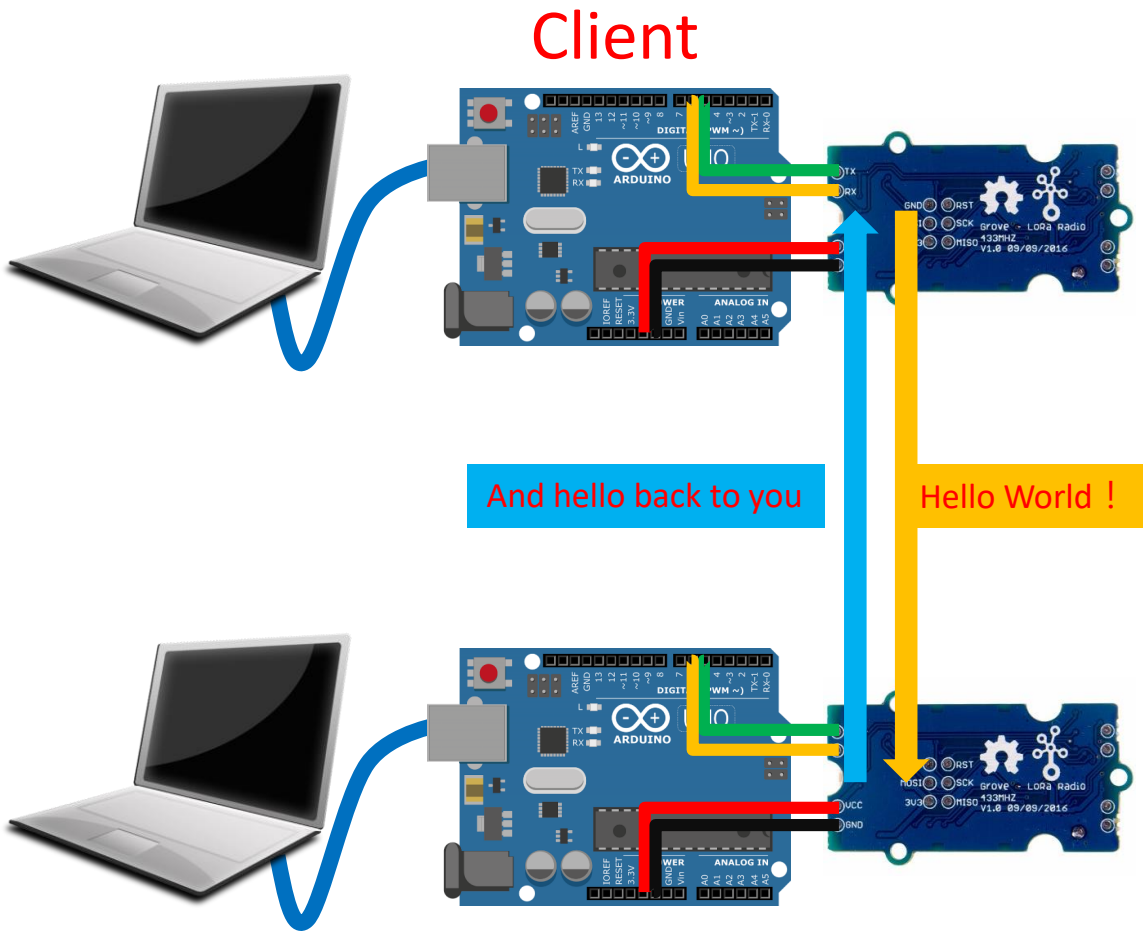
LoRa

- Goal: Use LoRa to transmit data between client and server
- Hardware Required
  - Arduino
  - Grove-LoRa Radio



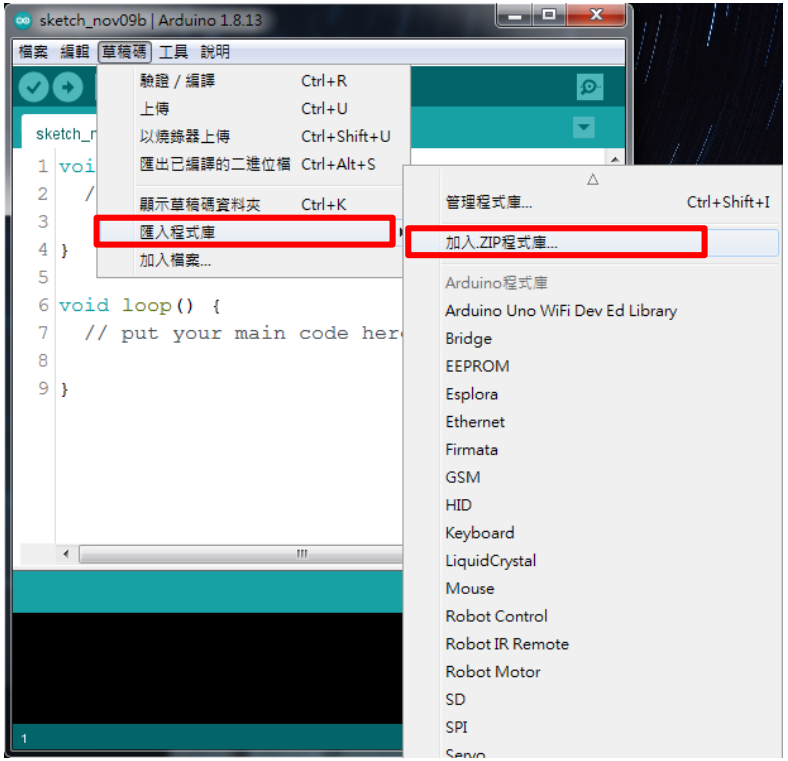
LoRa

引脚	名字	功能
1	TX	TX of UART
2	RX	RX of UART
3	VCC	电源输入, 3.3V or 5V
4	GND	连接到地



LoRa library

- Download and import library



https://github.com/Seeed-Studio/Grove_LoRa_433MHz_and_915MHz_RF/archive/master.zip

LoRa code

Client & Server Declare and Setup

rf95_client

```
10 #include <RH_RF95.h>
11 判斷是哪種晶片架構的控制
12 #ifdef __AVR__
13     #include <SoftwareSerial.h>
14     SoftwareSerial SSerial(5, 6); // RX, TX
15     #define COMSerial SSerial
16     #define ShowSerial Serial
17     使用RH_RF95函式庫宣告LoRa裝置為rf95
18     RH_RF95<SoftwareSerial> rf95(COMSerial);
38 void setup() {
39     ShowSerial.begin(115200);
40     ShowSerial.println("RF95 client test.");
41         將LoRa裝置初始化
42     if (!rf95.init()) {
43         ShowSerial.println("init failed");
44         while (1);
45     設定LoRa裝置的通訊頻段
54     rf95.setFrequency(434.0);
55 }
```

If you are the server, you need to modify here by manual

Set a proper frequency with your partner

Table 48 RFM95/96/97/98 Device Variants and Key Parameters

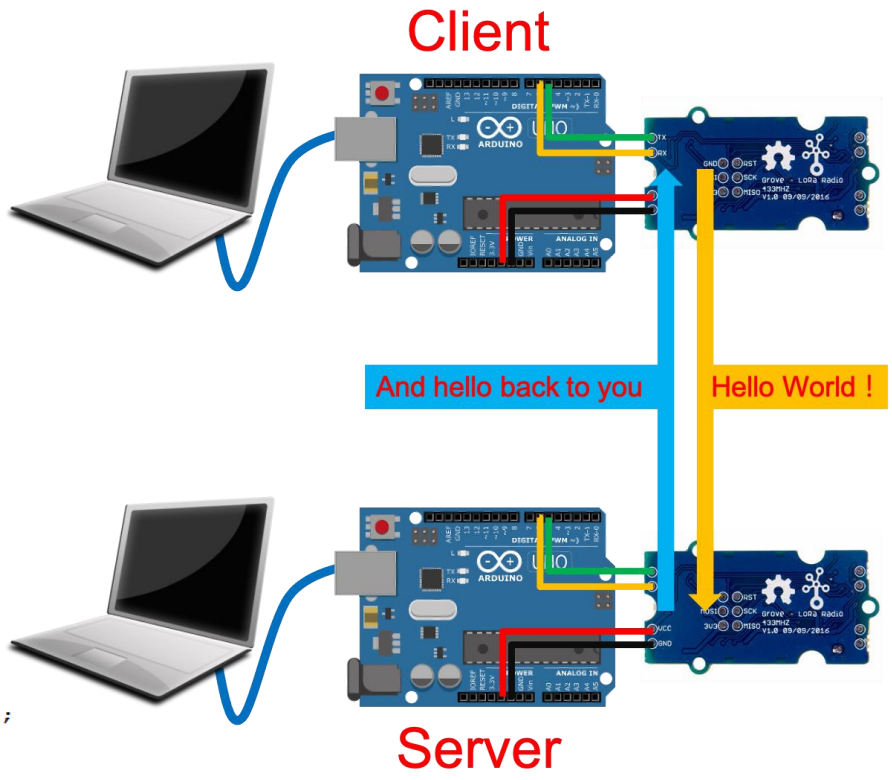
Part Number	Frequency Range	Spreading Factor	Bandwidth	Effective Bitrate	Est. Sensitivity
RFM95	868/915 MHz	6 - 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm
RFM97	868/915 MHz	6 - 9	7.8 - 500 kHz	0.11 - 37.5 kbps	-111 to -139 dBm
RFM96/RFM98	433/470MHz	6- 12	7.8 - 500 kHz	.018 - 37.5 kbps	-111 to -148 dBm

https://files.seeedstudio.com/wiki/Grove_LoRa_Radio/res/RFM95_96_97_98_DataSheet.pdf

LoRa client

Client

```
57 void loop() {
58     ShowSerial.println("Sending to rf95_server");
59     // Send a message to rf95_server
60     uint8_t data[] = "Hello World!";
61     rf95.send(data, sizeof(data));
62
63     rf95.waitPacketSent();
64
65     // Now wait for a reply
66     uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
67     uint8_t len = sizeof(buf);
68
69     if (rf95.waitAvailableTimeout(3000)) {
70         // Should be a reply message for us now
71         if (rf95.recv(buf, &len)) {
72             ShowSerial.print("got reply: ");
73             ShowSerial.println((char*)buf);
74         } else {
75             ShowSerial.println("recv failed");
76         }
77     } else {
78         ShowSerial.println("No reply, is rf95_server running?");
79     }
80
81     delay(1000);
82 }
```

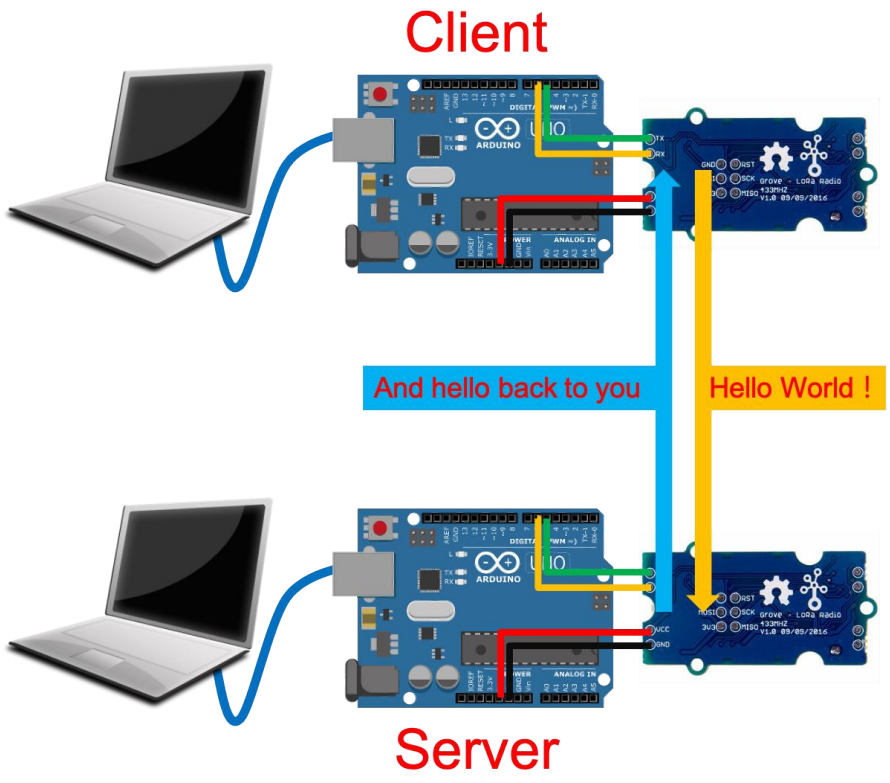


LoRa server

Server

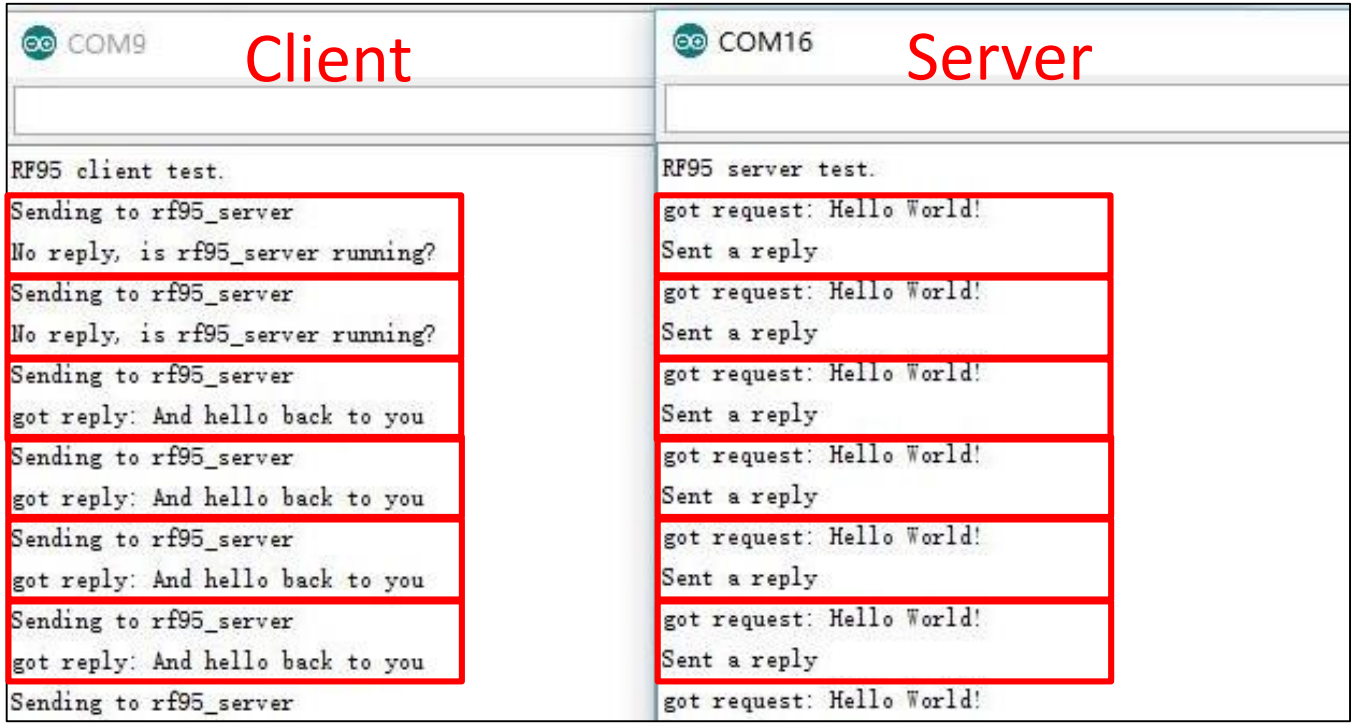
rf95_server §

```
59
60 void loop() {
61     if (rf95.available()) {
62         // Should be a message for us now
63         uint8_t buf[RH_RF95_MAX_MESSAGE_LEN];
64         uint8_t len = sizeof(buf);
65         if (rf95.recv(buf, &len)) {
66             digitalWrite(led, HIGH);
67
68             ShowSerial.print("got request: ");
69             ShowSerial.println((char*)buf);
70
71             // Send a reply
72             uint8_t data[] = "And hello back to you";
73             rf95.send(data, sizeof(data));
74             rf95.waitPacketSent();
75             ShowSerial.println("Sent a reply");
76
77             digitalWrite(led, LOW);
78         } else {
79             ShowSerial.println("recv failed");
80         }
```



https://www.airspayce.com/mikem/arduino/RadioHead/classRH__RF95.html

Results


COM9
Client
RF95 client test.
Sending to rf95_server
No reply, is rf95_server running?
Sending to rf95_server
No reply, is rf95_server running?
Sending to rf95_server
got reply: And hello back to you
Sending to rf95_server
got reply: And hello back to you
Sending to rf95_server
got reply: And hello back to you
Sending to rf95_server
got reply: And hello back to you
Sending to rf95_server
COM16
Server
RF95 server test.
got request: Hello World!
Sent a reply
got request: Hello World!
Sent a reply
got request: Hello World!
Sent a reply
got request: Hello World!
Sent a reply
got request: Hello World!
Sent a reply
got request: Hello World!
Sent a reply
got request: Hello World!