**物聯網核心技術  
實驗二 NB-IoT 雙向控制系統**

**國立中正大學資訊工程系 黃仁竑教授**

**實驗目的：**

透過小型智慧溫室來使用NB-IoT通訊模組進行雙向控制實驗。包含空氣溫溼度感應器、土壤溫溼度感應器、風扇、照光、滴灌澆水系統。將智慧溫室的感應器數據透過NB-IoT模組傳送到雲端平台，經過分析數據後再透過NB-IoT適時進行開啟風扇、燈光、以及滴灌澆水馬達，讓溫室環境適合植物成長。

**實驗所需軟硬體**

軟體：

* Windows 10
* Arduino IDE
* VSCode
* Mobaxterm

硬體：

* LinkIt Smart 7688 Duo
* SIM7000E ( NB-IoT模組 )
* SIM Card
* 土壤溫濕度感應器
* 光線感應器
* 溫濕度感應器
* LED 燈條、燈條轉接頭
* 風扇、螺絲組
* 繼電器
* 馬達、水管、滴灌分接頭
* 電源供應模組、電源供應器
* 溫室外箱
* 杜邦線
* Micro-USB

目錄

* 材料
* 目標一：MQTT PUB及SUB指令

1. MQTT
2. PUBLISH及SUBSCRIBE封包格式
3. 建立broker
4. 傳送PUBLISH封包
5. 傳送SUBSCRIBE封包

* 目標二：組裝及控制硬體設備

1. 原理與構造
2. 土壤濕度感測器
3. 光線感測器
4. 溫濕度感測器
5. 馬達
6. 繼電器
7. 電源供應模組
8. 控制設備及顯示資訊
9. 設備安裝
10. 控制及顯示

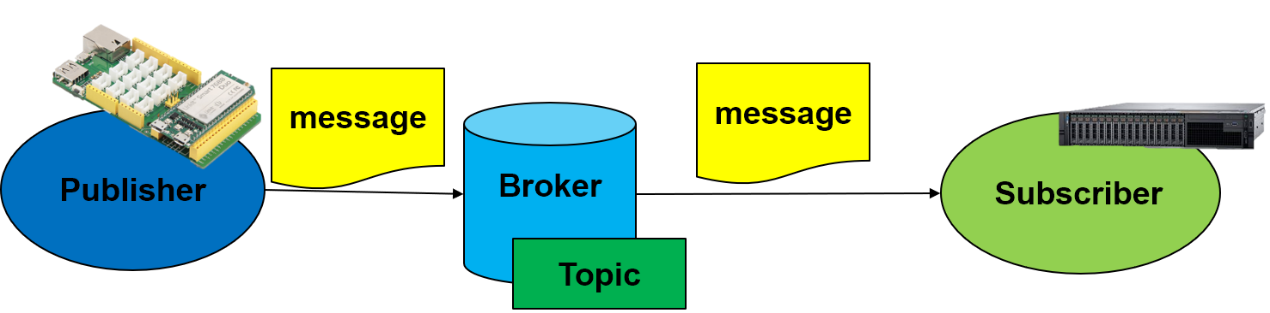
* 目標三：MQTT雙向控制

1. 以PUB上傳光感值
2. 以SUB控制開關燈
3. 雙向控制

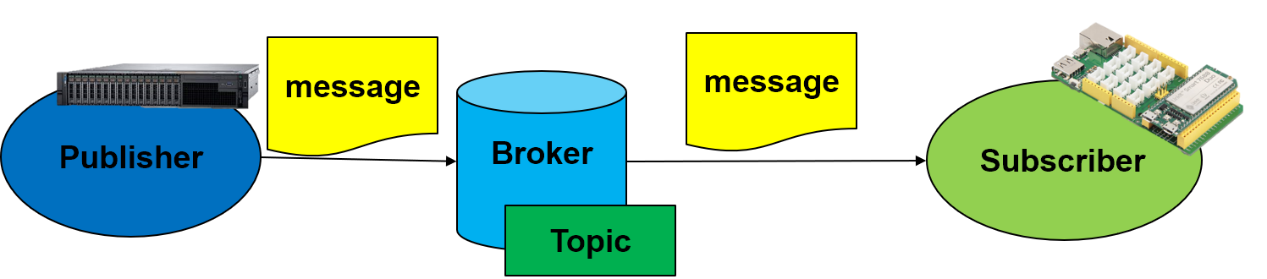
* 材料

|  |  |  |
| --- | --- | --- |
| LinkIt Smart 7688 Duo  (含擴充板) | SIM7000E | SIM Card |
|  |  |  |
| 土壤濕度感測器 | 光線感測器 | 溫濕度感測器 |
|  |  |  |
| LED 燈條 & 燈條轉接頭 | 風扇 & 螺絲組 | 繼電器 |
|  |  |  |
| 馬達、水管、滴灌分接頭 | 電源供應模組 | 電源供應器 |
|  |  |  |
| 溫室外箱 |  |  |
|  |  |  |

* 目標一：MQTT PUB及SUB指令

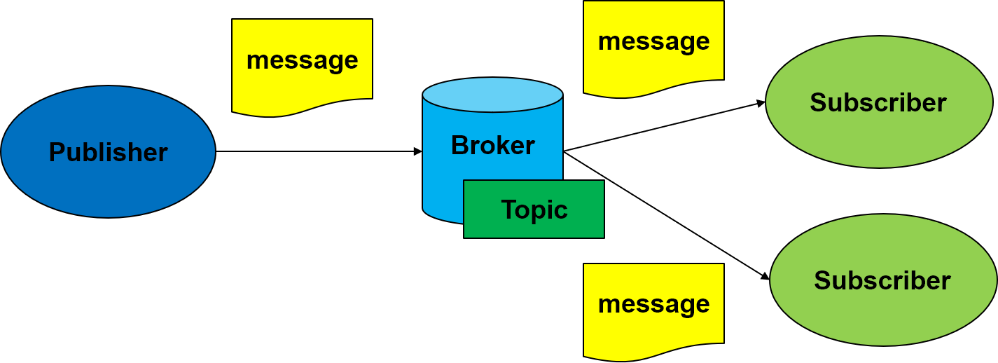


我們以7688作為publisher，透過MQTT將訊息上傳到server的broker中，並在server中建立subscriber，監控是否又上傳成功。

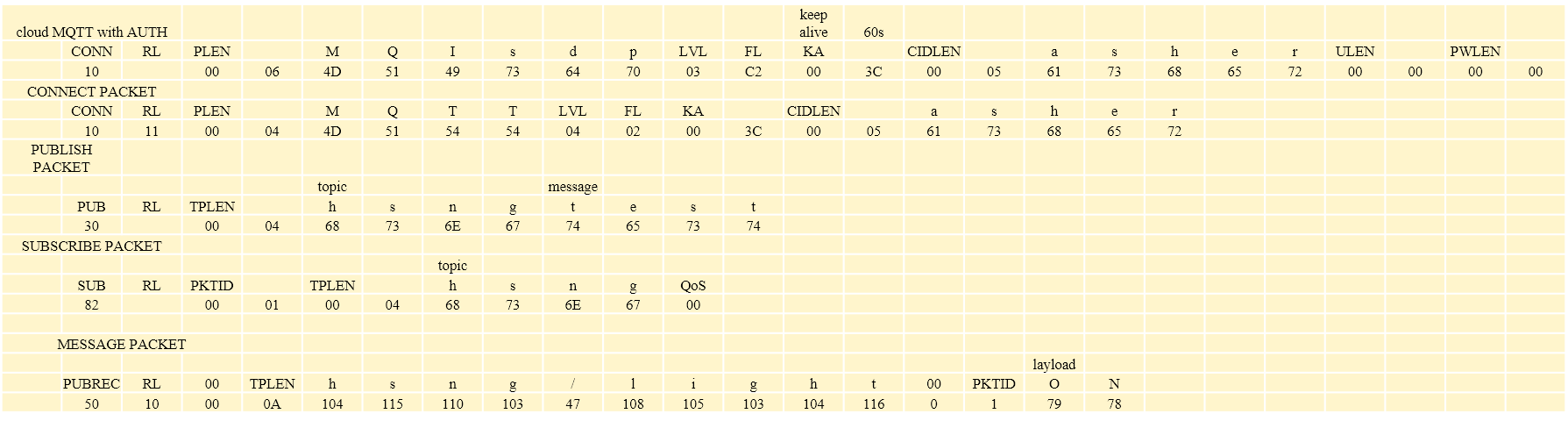


同時我們在server中publish指令傳輸訊息，透過7688來接收訊息。

* MQTT簡介



* + MQTT 全名是Message Queueing Telemetry Transport，上圖是示意圖，其重要的組成元素如下所示：
    1. Publisher：負責上傳訊息至broker，會指定topic及其他參數，例如：QoS
    2. Broker：負責將收到的訊息轉傳至有訂閱該訊息的Subscriber，其中能設定各項參數，例如：重新傳輸時間間隔
    3. Subscriber：向Broker訂閱Topic，並接收該Topic的訊息
    4. Topic：用來分類訊息，也是用來分類訊息要傳給哪個用戶的標籤，例如：CCU/CSIE/name，其大小寫不同，‘/’是主題階層分隔字元，特殊字元(例如：’-’，‘\*’)有特殊意義，不建議使用
  + 封包由Header+Topic+Payload組成
  + 使用的Port為1883
  + MQTT 的QoS及其代表意義：
    1. QoS = 0，at most once
    2. QoS = 1，at least once
    3. QoS = 2，exactly once
* PUBLISH及SUBSCRIBE封包格式
  + 其範例格式如下圖所示：



* + 封包應使用十六進位來表示，以下列舉重要的欄位及其意義：
    1. Command type：2bytes，代表要傳輸的封包是什麼，例如connect是10、PUB是30等
    2. Remaining Length (RL)：2bytes，代表此packet在本欄位之後還有幾個byte，例如’0X00’、’0X11’代表還有17個bytes
    3. Keep alive (KA)：連線時間，單位是秒，例如0X3C是60秒
    4. Length：1~2個byte，描述後面對應的欄位長度，以下列舉描述及縮寫對應：
       - Protocol Length (PLEN)
       - Client ID Length (CINLEN)
       - User Length (ULEN)
       - Password Length (PWLEN)
       - Topic Length (TPLEN)
       - Packet Length (PKLEN)
  + 一個packet傳遞後，需在最後攜帶’0X1A’
* 建立broker
  + 遠端連線至server
    1. 指令：ssh <user name>@IP, e.g., ssh asher@192.168.1.100
    2. user name: ccu<student ID>, e.g., ccu609410055
    3. Password: <student ID>, e.g., 609410055
    4. 若是使用個人網路，需先VPN至學校
       - 學校VPN：https://it.ccu.edu.tw/network\_source.php?section=7
       - 資工VPN：<https://vpn.cs.ccu.edu.tw/>
  + 建立MQTT及Broker
    1. 更新
       - sudo apt update
    2. 安裝 Mosquitto MQTT Broker
       - sudo apt install -y mosquitto
    3. 安裝 Mosquitto Client
       - sudo apt install -y mosquitto-clients
  + 測試broker
    1. 建立 Subscriber及Subscribe
       - mosquitto\_sub -h IP -t Topic
    2. 建立 Publisher及Publish
       - mosquitto\_pub -h IP -t Topic -m Message
* NB-IoT模組

|  |  |
| --- | --- |
|  |  |
| 實際相片 | 接線圖 |

* 傳送PUBLISH封包
  + 新增SIM7000 library至Arduino 函式庫
    1. 草稿碼->匯入程式庫->管理程式庫
    2. 搜尋SIM7000，找到DFRobot\_SIM7000，按下安裝1.0.1版本
    3. 若有詢問是否要連同其他lib同時安裝，選擇全部安裝，或是於安裝後點選更新

|  |  |
| --- | --- |
|  |  |
| 安裝library | 管理程式庫路徑 |

* + 修正lib中的bug及設定
    1. 至github下載lib：https://github.com/ashercy/iot.git
       - 位置： iot/lib/DFRobot\_SIM7000.cpp
       - 位置： iot/lib/DFRobot\_SIM7000.h
    2. 至Arduino library覆蓋上面的兩個檔案
       - 位置： C:\Users\user\Documents\Arduino\libraries\DFRobot\_SIM7000-master
  + 至github下載範例程式：<https://github.com/ashercy/iot.git>
    1. 位置：iot/example/MQTT\_PUB.ino
  + 修改參數
    1. serverIP
    2. IOT\_CLIENT: <student ID>
    3. IOT\_TOPIC: <student ID>/pub
  + 上傳內容為學號
  + 換行符號記得改為CRLF(NL&CR)

|  |  |
| --- | --- |
|  | Server端結果 |
|  | Arduino monitor端結果 |

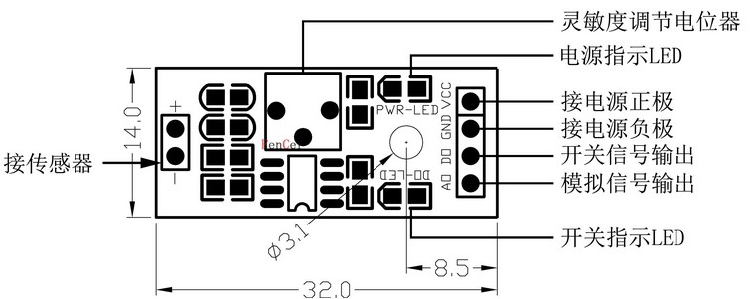
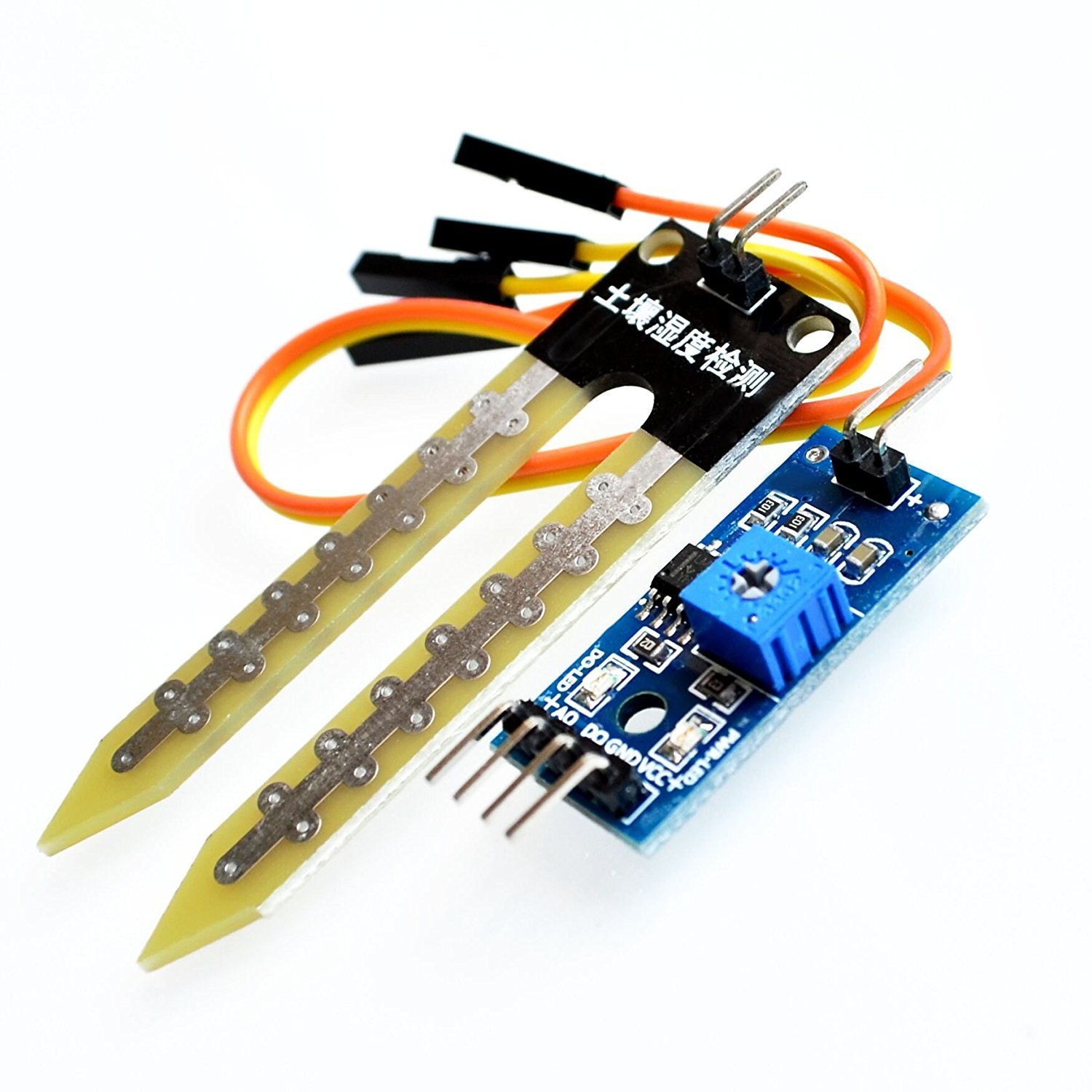
* 傳送SUBSCRIBE封包
  + 至github下載範例程式：<https://github.com/ashercy/iot.git>
    1. 位置：iot/example/MQTT\_SUB.ino
  + 修改參數
    1. serverIP
    2. IOT\_CLIENT: <student ID>
    3. IOT\_TOPIC: <student ID>/sub
  + 下載MQTT app，使用該應用發出PUB packet，並以模組接收，記得用QoS=1
    1. 訊息為學號

|  |  |
| --- | --- |
|  | Server端結果 |
|  | Arduino monitor端結果 |

* 目標二：組裝及控制硬體設備

1. 原理與構造
2. 土壤濕度感測器

土壤濕度感測器必須與比較器LM393一起使用，工作電壓為3.3V-5V。

1. 光線感測器

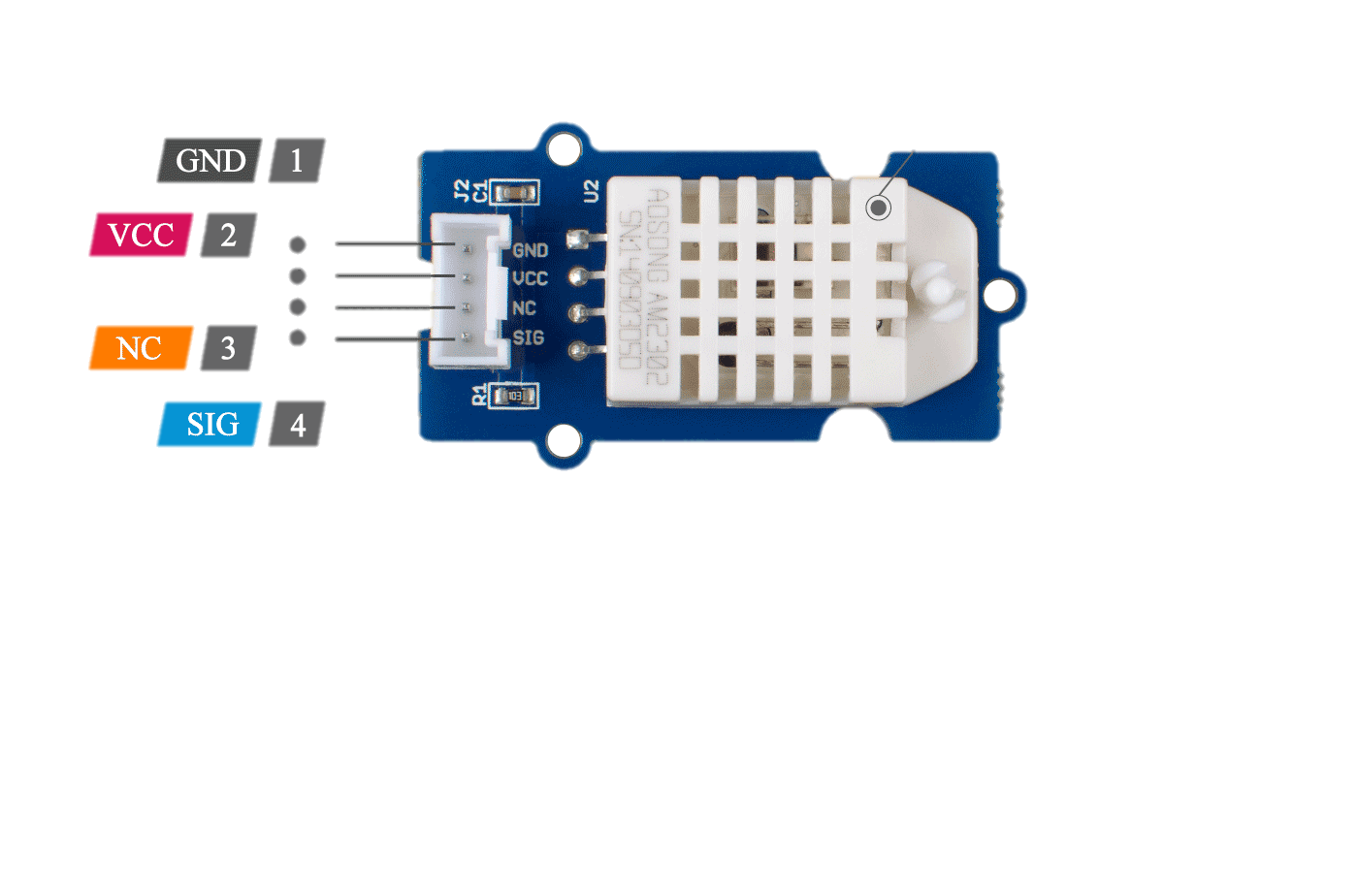
The resistance of photo-resistor decreases when the intensity of light increases.

A dual Op-Amp chip LM358 on board produces voltage corresponding to intensity of light(i.e. based on resistance value).

The output signal is analog value, the brighter the light is, the larger the value.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  | | --- | --- | | Operating voltage | 3~5V | | Operating current | 0.5~3 mA | | Response time | 20-30 milliseconds | | Peak Wavelength | 540 nm | |

1. 溫濕度感測器



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VCC | 3.3 | – | 6 | V |
| 支持測量電流 | 1 | – | 1.5 | mA |
| 待機電流 | 40 | – | 50 | uA |
| 測量範圍 (濕度) | 5% | – | 99% | RH |
| 測量範圍 (溫度) | -40 | – | 80 | °C |
| 精度 (濕度) | – | – | ±2% | RH |
| 精度 (溫度) | – | – | ±0.5 | °C |
| 解析度 (濕度) | – | – | 0.1% | RH |
| 解析度 (溫度) | – | – | 0.1 | °C |
| 再現性 (濕度) | – | – | ±0.3% | RH |
| 再現性 (溫度) | – | – | ±0.2 | °C |
| 長期穩定性 | – | – | ±0.5% | RH/year |
| 訊號採集週期 | – | 2 | – | S |
| 反應時間1 / e（63％） | 6 | – | 20 | S |

1. 馬達

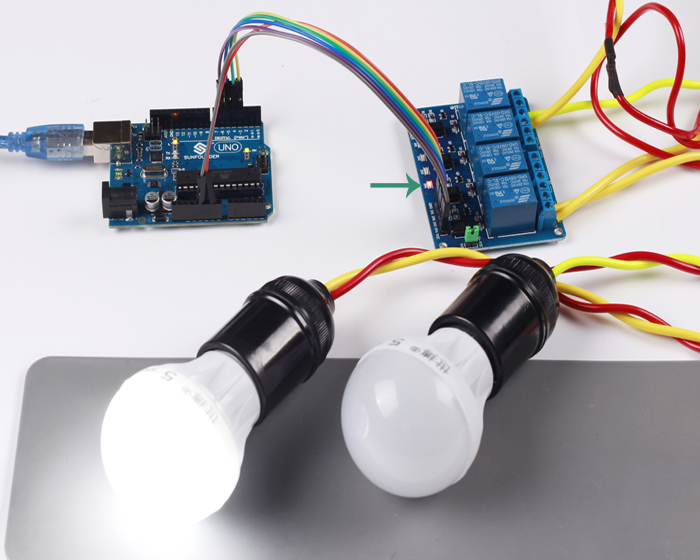
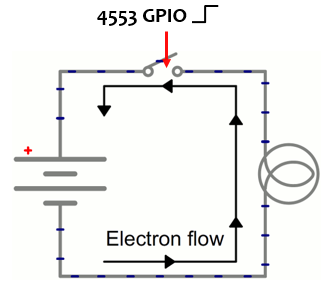
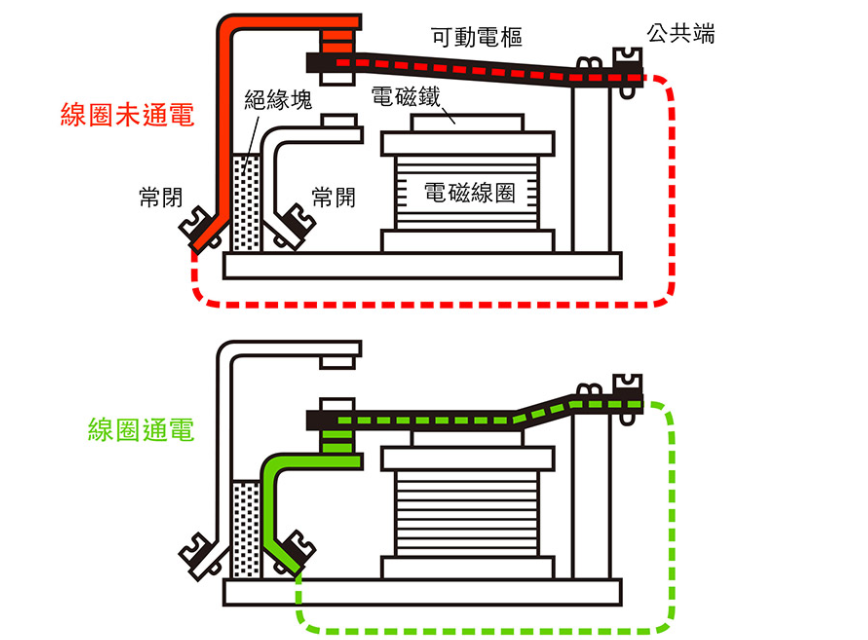
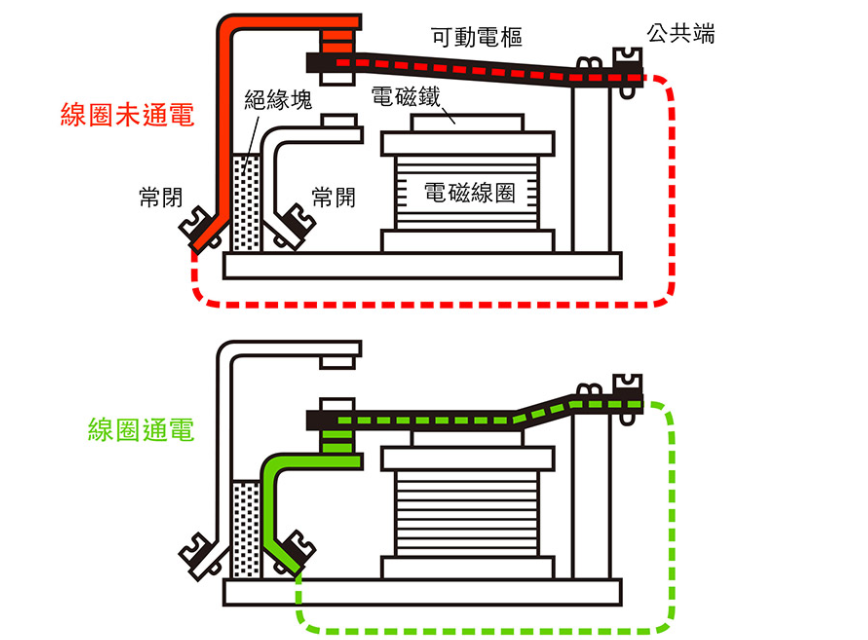
直流3~5V。

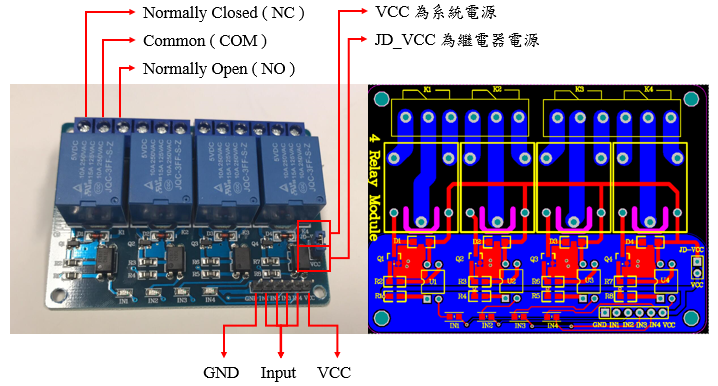


1. 繼電器( Relay )

用途是使小電力可以驅動大電力，並作為開關使用，它可分為高電位或低電位觸發，本次提供的繼電器為低電位觸發，主要構造有以下：

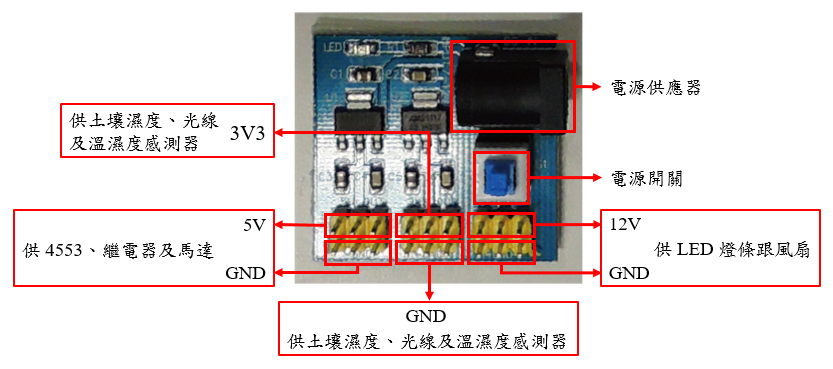
* COM：公共端。
* NO：常開，平時為斷路，繼電器開啟時與COM相接。
* NC：常閉，平時與COM為通路，繼電器開啟時斷路。
* Input：控制開關。



1. 電源供應模組(PDM)

用來將電源供應器提的的電壓分別轉換為3V3、5V、12V。



* + 控制設備及顯示資訊
    1. 設備及其電源強度
       - 注意：電源供應模組有兩個版本，3.3V及5V位置相反

|  |  |
| --- | --- |
| **名稱** | **電源** |
| 土壤濕度感測器 | 3.3V |
| 光線感測器 |
| 溫濕度感測器 |
| LED Bar |
| 繼電器 | 5V |
| 馬達 |
| LED燈條 | 12V |
| 風扇 |

* + 設備安裝
    1. 土壤濕度感測器

|  |  |
| --- | --- |
|  |  |
| 實際照片 | 接線圖 |

* + 1. 光線感測器

|  |  |
| --- | --- |
|  |  |
| 實際照片 | 接線圖 |

* + 1. 溫濕度感測器

|  |  |
| --- | --- |
|  |  |
| 實際照片 | 接線圖 |

* + 1. LED Bar

|  |  |
| --- | --- |
|  |  |
| 實際照片 | 接線圖 |

* + 1. 馬達

|  |  |
| --- | --- |
|  |  |
| 實際照片 | 接線圖 |

* + 1. 繼電器

|  |  |
| --- | --- |
|  |  |
| 實際照片 | 接線圖 |

* + 1. LED燈條

|  |  |
| --- | --- |
|  |  |
| 實際照片 | 接線圖 |

* + 1. 風扇

|  |  |
| --- | --- |
|  |  |
| 實際照片 | 接線圖 |

* + 1. 組合圖

|  |  |
| --- | --- |
|  |  |
| 實際照片 | 接線圖 |

* + 1. 放入箱子



* + 控制及顯示
    1. 至github下載範例程式：<https://github.com/ashercy/iot.git>
       - 位置：iot/example/Smart\_Greenhouse.ino

|  |  |
| --- | --- |
|  |  |
| 設備照片 | Arduino IDE 截圖 |

* 目標三：MQTT雙向控制

|  |
| --- |
|  |
| 示意圖 |

* 1. 以PUB上傳光感值

|  |  |
| --- | --- |
|  |  |
| 實際照片 | 接線圖 |

* + - 至github下載範例程式：<https://github.com/ashercy/iot.git>
      * 位置：iot/example/Pub\_Sensor.ino
    - 修改參數
      * serverIP
      * IOT\_CLIENT: <student ID>
      * IOT\_TOPIC: <represent student ID>/lab2

|  |  |
| --- | --- |
|  | Server端結果 |
|  | Arduino monitor端結果 |

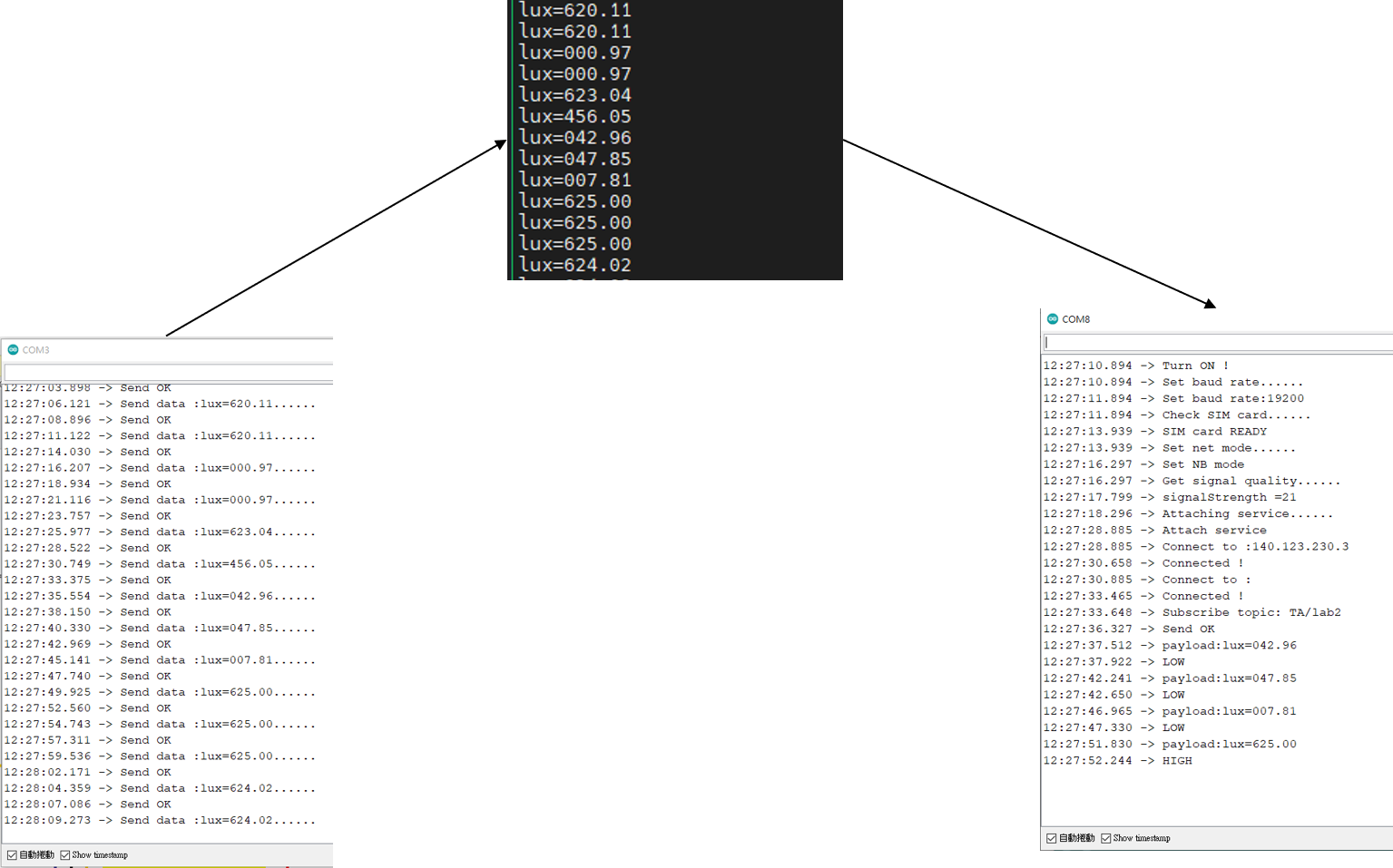
* 1. 以SUB控制開關燈

|  |  |
| --- | --- |
|  |  |
| 實際照片 | 接線圖 |

* 至github下載範例程式：<https://github.com/ashercy/iot.git>
  + 位置：iot/example/Sub\_LED.ino
* 修改參數
  + serverIP
  + IOT\_CLIENT: <student ID>
  + IOT\_TOPIC: <represent student ID>/lab2

|  |  |
| --- | --- |
|  | Server端結果 |
|  | Arduino monitor端結果 |

* 1. 雙向控制



* 常見問題
  1. 無法上傳檔案
     + 至裝置管理員看是否COM port寫錯，或是裝置未選擇LintIt 7688
  2. SIM 7000E: Fail to set baud rate
     + SIM7000E電源重新開關
  3. 繼電器不會動
     + 電源供應器的開關要記得按下去，按下後會亮紅燈
  4. 一台SIM7000E連上後，讓另一台已經連上的模組斷線
     + Client ID要設定不同，避免自己干擾自己