

**題目**

植物光照及澆水系統

**作者**

| 學號       | 姓名  | email                     |
|----------|-----|---------------------------|
| 00757136 | 王柏森 | 00757136@mail.ntou.edu.tw |

指導教授：辛華昀 博士

中華民國 110年 12 月 10 日

## 摘要

### 1. 發想：

在室內種植植物有時會忘記或無法澆水，又或是缺乏光照導致植物生長不利，此時若能用手機遠端管理光照及澆水便能解決此類問題。

### 2. 實作方法：

使用樹莓派以及麵包板連接陽光感測器、繼電器等電子元件，搭配光照燈和抽水馬達完成光照及澆水的嵌入式系統建置。再透過樹莓派端執行的 Python 程式接收手機端應用程式發送的訊息並執行程序，裝置間的溝通採用訊息佇列遙測傳輸協定完成。

### 3. 目標：

本專題的目標是實作出物聯網架構的植物光照及澆水系統，讓使用者能夠利用手機應用程式透過網路傳輸完成植物的光照及澆水管理，其中功能包括手動進行光照燈開關及澆水、透過設定時間完成定時光照及澆水、以及讓系統依陽光感測器讀數自動判定是否開啟光照。

## 一、簡介

### 1. 實驗動機

種植物常會有一些困擾，例如出門後才想起來還沒有給植物澆水、出遠門連續數天無法給植物澆水導致植物枯死、又或者是因為光照不足導致植物長得不好，此時若可以利用手機應用程式隨時進行光照及澆水的管理，就不用擔心這些問題了。

### 2. 相關概念及技術：

#### (1) 物聯網：

物聯網是一種透過將裝置連上網路，以實現機器對機器通訊，來達成各種機械作業、感測數據回報等程序的技術，其架構主要分為感測層、網路層及應用層，感測層的組成通常包括一微控制器或微處理器連接各式感測元件以及機械裝置；網路層有各種網路傳輸技術選擇，常見的例如 Wi-Fi、藍牙等；應用層則是一種服務提供，根據使用者需求可以是一種人機操作介面，又或是支援其他軟體應用的平台。

## (2) 嵌入式系統：

嵌入式系統是嵌入機械並具有專一功能和計算的電腦系統，通常是由一微控制器或微處理器進行機械部件的控制與操作。

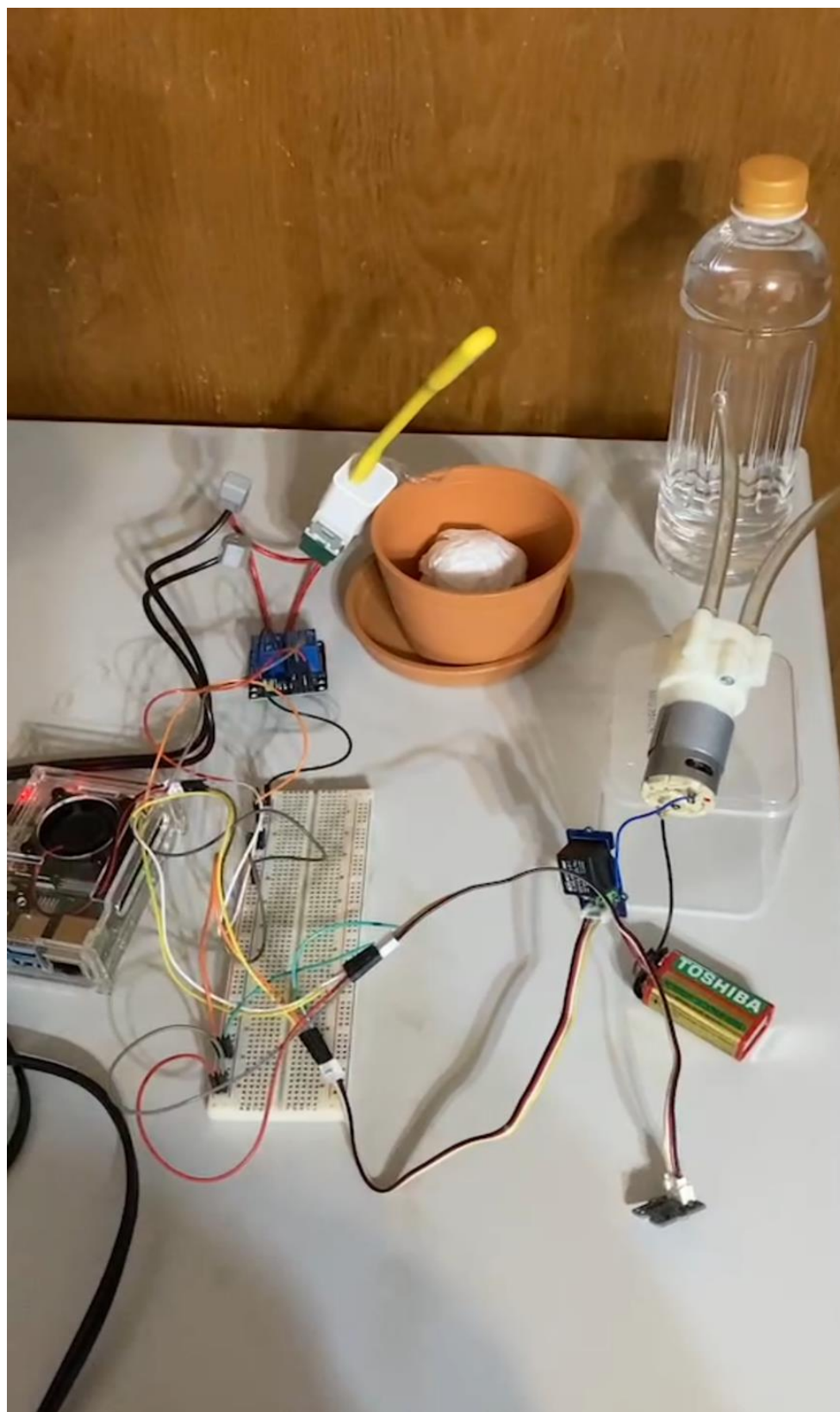
## (3) 訊息佇列遙測傳輸：

(MQTT, Message Queuing Telemetry Transport) 是一種基於發佈及訂閱模式的訊息傳輸協定，每個訊息的傳輸透過主題(topic)進行管理，主要實體分為訊息代理(message broker)與客戶端(client)，運作的模式為當訊息代理收到特定主題的訊息後，會向有訂閱該主題的客戶端發佈訊息。服務品質分為三個等級：0，最多傳送一次，即只負責將訊息傳送出去而不管傳送狀態；1，至少傳送一次，若過了一段時間沒收到回傳的訊息則再次傳送，可能導致接收方收到重複的封包；2，正好傳送一次，當確認收到訊息後便刪除重複的封包。

### 3. 實作

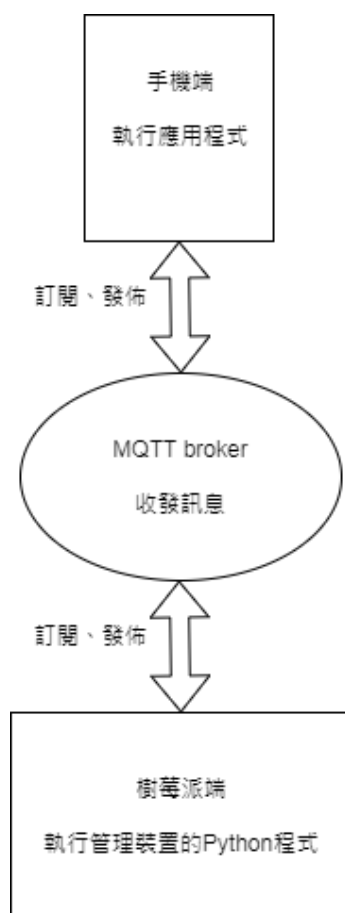
使用樹莓派以及麵包板連接陽光感測器、繼電器等電子元件，再透過繼電器分別連接光照燈和抽水馬達完成光照及澆水的嵌入式系統裝置建置。裝置建置完成後確認樹莓派是否有連接可用的網路例如 Wi-Fi，確認後在樹莓派端執行設計好的 Python 程式以進行訊息的收發來達成對繼電器的控制和接收陽光感測器的讀數。之後便利用手機端安裝的應用程式來對樹莓派端收發採用訊息佇列遙測傳輸協定的控制訊息，完成對裝置控制，其中手機端應用程式使用了 UrsAI2MQTT 擴充套件來完成 MQTT 協定的處理。

整體設備如下圖所示：

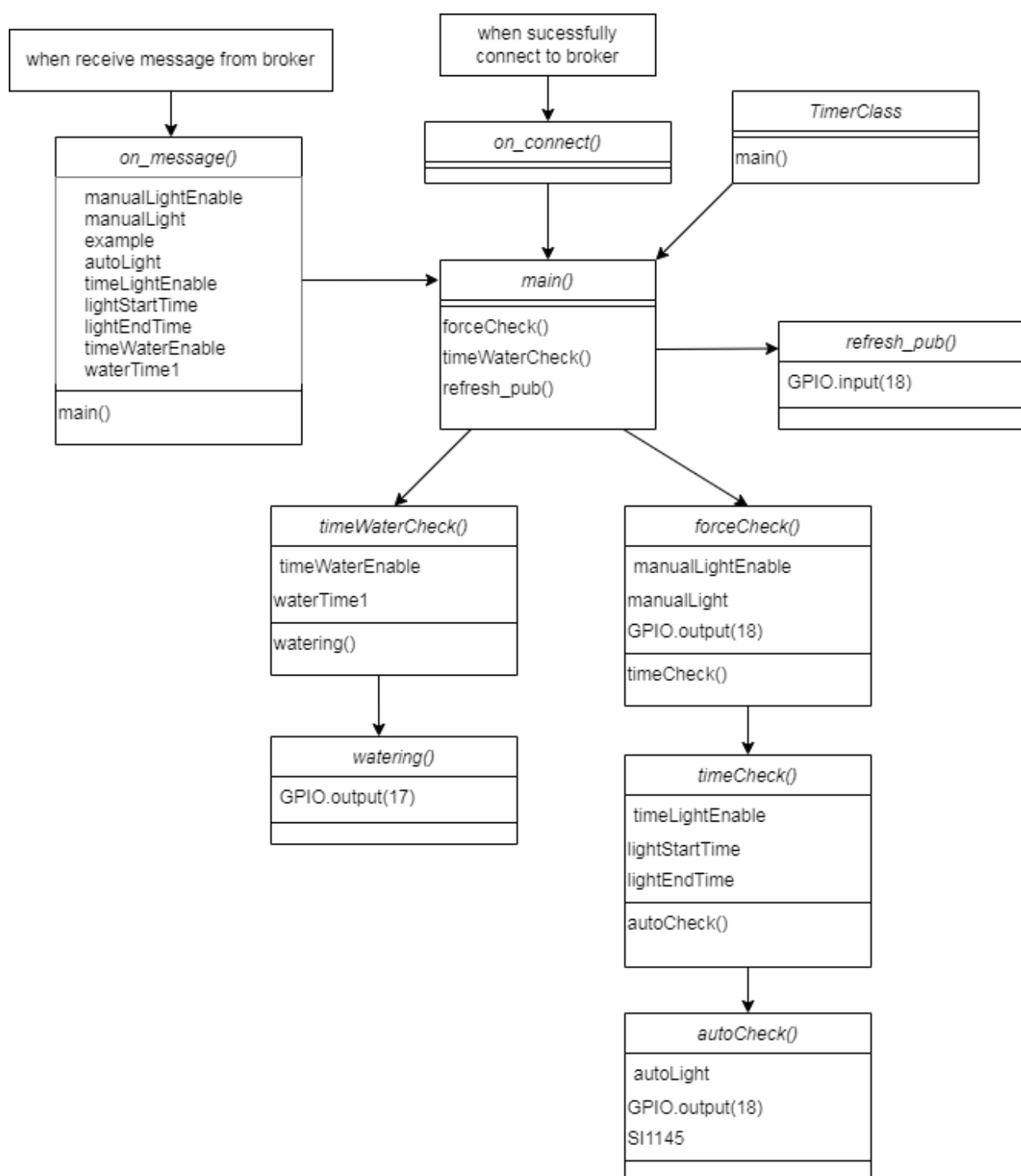


## 二、架構

### 1. 系統裝置架構



## 2. Python 程式架構





### 三、模組設計描述

#### 1. 檔案名稱：

##### (1)樹莓派端程式 plantLightingAndWatering.py

```
1  # -*- coding: utf-8 -*-
2  import paho.mqtt.client as mqtt
3  import random
4  import json
5  import datetime
6  import time
7  import seeed_si114x          #用來讀取陽光感測器讀數
8  import signal
9  import RPi.GPIO as GPIO
10 import threading
11
12
13 #設定樹莓派接線以及各變數初始值
14 GPIO.setup(17, GPIO.OUT, initial=GPIO.LOW)      #樹莓派接線 gpio 17 relay for pump
15 GPIO.setup(18, GPIO.OUT, initial=GPIO.HIGH)     #gpio 18 relay for light
16 SI1145 = seeed_si114x.grove_si114x()
17 manualLightEnable="off"
18 manualLight="off"
19 autoLight="off"
20 timeLightEnable="off"
21 lightStartTime=datetime
22 lightEndTime=datetime
23 timeWaterEnable="off"
24 waterTime1=datetime
25
26 # 設置日期時間的格式
27 ISOTIMEFORMAT = '%m/%d %H:%M:%S'
28
29 # 連線設定
30 # 初始化本地端程式
31 client = mqtt.Client()
32
33 # 設定連線資訊(IP, Port, 連線時間)
34 client.connect("test.mosquitto.org", 1883, 60)
35
36 #連線後訂閱client端會發佈的主題
37 def on_connect(client, userdata, flags, rc):
38     print("Connected with result code "+str(rc))
39     client.subscribe("manualLight")
40     client.subscribe("autoLight")
41     client.subscribe("timeLightEnable")
42     client.subscribe("lightStartTime")
43     client.subscribe("lightEndTime")
44     client.subscribe("clientRefresh")
45     client.subscribe("manualLightEnable")
46     client.subscribe("clientWatering")
47     client.subscribe("timeWaterEnable")
48     client.subscribe("waterTime1")
49     print("ok")
50
51 # 設定連線的動作
52 client.on_connect = on_connect
53
```

```

54 #發佈目前的狀態給用戶
55 def refresh_pub():
56     if GPIO.input(18) == GPIO.HIGH:
57         client.publish("currentLightState", "off")
58         print("pub currentLightState off")
59     if GPIO.input(18) == GPIO.LOW:
60         client.publish("currentLightState", "on")
61         print("pub currentLightState on")
62     client.publish("currentVisibleLight", str(SI1145.ReadVisible))
63     print("pub currentVisibleLight %s"%(str(SI1145.ReadVisible)))
64
65
66 #開啟澆水裝置
67 def watering():
68     GPIO.output(17,GPIO.HIGH)
69     time.sleep(12) #可調整澆水時間
70     GPIO.output(17,GPIO.LOW)
71
72
73 #確認是否開啟定時澆水
74 def timeWaterCheck():
75     global timeWaterEnable
76     global waterTime1
77     print("timeWaterCheck")
78     if timeWaterEnable=="on":
79         print("timeWaterEnable:on")
80         print("waterTime1=%s"%(waterTime1))
81         try:
82             now=datetime.datetime.now()
83             if now.minute==waterTime1.minute and now.second<15:
84                 watering()
85         except Exception as e:
86             print("%s"%(e))
87
88     elif timeWaterEnable == "off":
89         print("timeWaterEnable: off")
90
91
92 #確認是否開啟自動判斷開燈
93 def autoCheck():
94     global autoLight
95
96     print("auto check")
97     threshold=262 #auto的閾值
98     if autoLight == "on":
99         print("autoLight: on")
100         GPIO.output(18,GPIO.HIGH)
101         if SI1145.ReadVisible>=threshold:
102             GPIO.output(18,GPIO.HIGH)
103         elif SI1145.ReadVisible<threshold:
104             GPIO.output(18,GPIO.LOW)
105     elif autoLight=="off":
106         print("autoLight: off")
107         GPIO.output(18,GPIO.HIGH)
108

```

```

109
110 #確認是否開啟定時開關燈
111 def timeCheck():
112     global timeLightEnable
113     global lightStartTime
114     global lightEndTime
115
116     print("time check")
117     if timeLightEnable == "on":
118         print("timeLightEnable: on")
119         try:
120             now=datetime.datetime.now()
121             if lightEndTime.time() < lightStartTime.time():
122                 print("m1") #設定的時間段有經過半夜十二點
123                 if now.time()>=lightStartTime.time() and now.time()<=datetime.time.max:
124                     print("已到設定的時間-開燈")
125                     GPIO.output(18,GPIO.LOW)
126                 elif now.time()>=datetime.time.min and now.time()<=lightEndTime.time():
127                     print("已到設定的時間-開燈")
128                     GPIO.output(18,GPIO.LOW)
129                 else:
130                     print("不在設定的時間範圍內-關燈")
131                     GPIO.output(18,GPIO.HIGH)
132                     autoCheck()
133             elif lightEndTime.time() > lightStartTime.time():
134                 print("m2") #設定的時間段沒經過半夜12點
135                 if now.time()>=lightStartTime.time() and now.time()<=lightEndTime.time():
136                     print("已到設定的時間-開燈")
137                     GPIO.output(18,GPIO.LOW)
138                 else:
139                     print("不在設定的時間範圍內-關燈")
140                     GPIO.output(18,GPIO.HIGH)
141                     autoCheck()
142         except Exception as e:
143             print("%s"%(e))
144
145     elif timeLightEnable == "off":
146         print("timeLightEnable: off")
147         autoCheck()
148
149 #確認是否手動控制燈的開關
150 def forceCheck():
151     global manualLightEnable
152     global manualLight
153
154     print("force check")
155     if manualLightEnable == "on":
156         if manualLight=="on":
157             GPIO.output(18,GPIO.LOW)
158         elif manualLight=="off":
159             GPIO.output(18,GPIO.HIGH)
160     elif manualLightEnable == "off":
161         timeCheck()
162
163
164 #光照燈基本流程
165 #判斷是否手動控制開關forceCheck()->是否定時開關timeCheck()->
166 #->是否在變暗時自動開關autoCheck()-> 結束後更新資訊refresh_pub()
167 #澆水定時檢查時間到了沒timeWaterCheck()
168 #更新資訊refresh_pub()
169 def main():
170     print("main check")
171     forceCheck()
172     timeWaterCheck()
173     refresh_pub()
174     #GPIO.cleanup()
175

```

```

176
177 #開始線程每隔幾秒重複執行一次main()，以進行定時設定的確認和資訊的更新
178 class TimerClass(threading.Thread):
179     def __init__(self):
180         threading.Thread.__init__(self)
181         self.event = threading.Event()
182
183     def run(self):
184         while not self.event.is_set():
185             main()
186             self.event.wait( 5 ) #每隔幾秒執行一次
187
188     def stop(self):
189         self.event.set()
190
191 tmr = TimerClass()
192 tmr.start()
193
194 #收到訊息時的動作，儲存使用者發送的設定
195 def on_message(client, userdata, msg):
196     msg.payload = msg.payload.decode("utf-8") #一定要先decode才能正常操作，否則會出錯
197     #debug
198     print(msg.topic+" "+ msg.payload)
199     global manualLightEnable
200     global manualLight
201     global example
202     global autoLight
203     global timeLightEnable
204     global lightStartTime
205     global lightEndTime
206     global timeWaterEnable
207     global waterTime1
208
209     if msg.topic == "clientRefresh":
210         refresh_pub()
211
212     if msg.topic == "clientWatering":
213         watering()
214
215     if msg.topic == "timeWaterEnable":
216         if str(msg.payload)=="on":
217             timeWaterEnable="on"
218         if str(msg.payload)=="off":
219             timeWaterEnable="off"
220
221     if msg.topic == "waterTime1":
222         try:
223             waterTime1=datetime.datetime.strptime(str(msg.payload),"H-%M")
224             print("waterTime1= %s"%(waterTime1))
225         except Exception as e:
226             print("%s"%(e))
227
228     if msg.topic == "manualLightEnable":
229         if str(msg.payload) == "on":
230             manualLightEnable = "on"
231             autoLight="off"
232             print("manualLightEnable %s"%(manualLightEnable))
233         if str(msg.payload) == "off":
234             manualLightEnable = "off"
235             print("manualLightEnable %s"%(manualLightEnable))
236

```

```

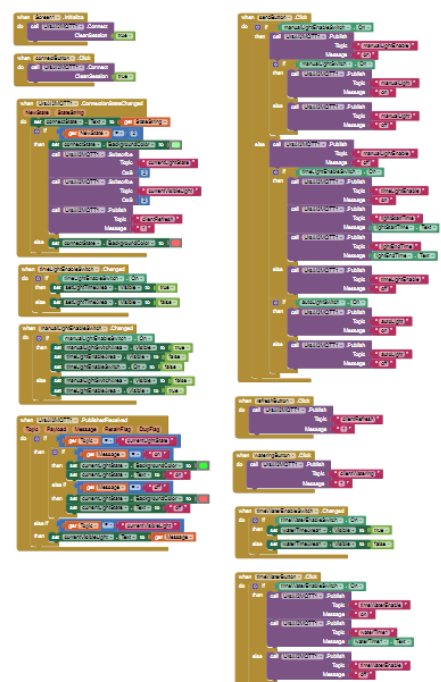
237
238 if msg.topic == "manualLight":
239     if str(msg.payload) == "on":
240         manualLight="on"
241     if str(msg.payload) == "off":
242         manualLight="off"
243
244 if manualLightEnable == "off":
245     if msg.topic == "autoLight":
246         if str(msg.payload) == "on":
247             autoLight="on"
248         if str(msg.payload) == "off":
249             autoLight="off"
250
251 if msg.topic == "timeLightEnable":
252     if str(msg.payload) == "on":
253         timeLightEnable="on"
254     if str(msg.payload) == "off":
255         timeLightEnable="off"
256 if msg.topic == "lightStartTime":
257     try:
258         lightStartTime=datetime.datetime.strptime(str(msg.payload), "%H-%M")
259         print("start= %s"%(lightStartTime))
260     except Exception as e:
261         print("%s"%(e))
262 if msg.topic == "lightEndTime":
263     try:
264         lightEndTime=datetime.datetime.strptime(str(msg.payload), "%H-%M")
265         print("end= %s"%(lightEndTime))
266     except Exception as e:
267         print("%s"%(e))
268
269 main()
270 refresh_pub()
271
272
273 # 設定接收訊息的動作
274 client.on_message = on_message
275
276 # 開始連線，執行設定的動作和處理重新連線問題
277 client.loop_forever()
278
279
280 if __name__ == '__main__':
281     main()
282

```

(2)手機端應用程式安裝檔 ProjectMQTT.apk

※圖為原 ai2 block 設計，於後面附件區附有

.aia 檔可詳細查看





## 2. 函式宣告及功能說明

(1) `def on_connect(client, userdata, flags, rc):`

當 broker 回應發送的請求時執行，功能為開始訂閱傳輸訊息所需要的主題。

(2) `def refresh_pub():`

功能為讀取後刷新及發佈當前光照燈的開關狀態以及陽光感測器讀取的數值。

(3) `def watering():`

功能為開啟控制抽水開關的繼電器，以達成澆水的目的。

(4) `def timeWaterCheck():`

功能為確認是否開啟定時澆水，若開啟則比較現在的時間及設定的時間來判斷是否應該澆水了，若時間到了則會執行 `watering()`。

(5) `def autoCheck():`

判斷自動開關是否開啟，若開啟則比較陽光感測器的讀數是否超過設定的閾值，若超過則自動關閉光照燈，若小於則自動開啟光照燈。

(6) `def timeCheck():`

判斷是否開啟定時開啟光照燈，若開啟則檢查當前的時間是否在設定的時間範圍內，如在範圍內則開啟光照燈，反之則關閉，擁有較 `autoCheck()` 高的執行權，結束後執行 `autoCheck`。

(7) `def forceCheck()`:

判斷使用者是否強制開啟或關閉光照燈，擁有較 `timeCheck()` 高的執行權，結束後執行 `timeCheck()`。

(8) `def main()`:

依序執行 `forceCheck()`、`timeWaterCheck()` 以及 `refresh_pub()`，來達成定時設置的檢查並刷新資料給使用者。

(9) `def on_message(client, userdata, msg)`:

當接收到訂閱主題的訊息時執行，主要功能為根據接收到的主題進行資料和設定的儲存，並在完成新設定的儲存後執行 `main()` 來套用設定。

### 3. 參數說明

manualLightEnable：字串型別，儲存手動開關的啟用與否。

manualLight：字串型別，儲存手動開關設定的值。

autoLight：字串型別，儲存自動判定照光的啟用與否。

timeLightEnable：字串型別，儲存定時照光的啟用與否。

lightStartTime：datetime 類別物件，儲存定時照光開始的時間。

lightEndTime：datetime 類別物件，儲存定時照光結束的時間。

timeWaterEnable：字串型別，儲存定時澆水功能的啟用與否。

waterTime1：datetime 類別物件，儲存澆水時間。

SI1145：由陽光感測器官方提供的 `seed_si114x.grove_si114x()` 物件，用以  
讀取陽光感測器數值。



## 四、實驗結果

實驗結果為各個目標功能正常，包括：

1. 利用手機應用程式手動進行光照燈開關及澆水。
2. 利用手機應用程式遠端發送時間設定來完成定時光照及澆水。
3. 利用手機應用程式遠端設定讓系統依陽光感測器讀數自動判定是否開啟光照。

※實際操作過程影片附於附件。

## 五、討論

專題目前功能如前面實驗結果所述正常，但由於網路傳輸速度及穩定度的關係會導致或多或少的延遲，然而此類物聯網裝置並不會因為幾秒的延遲導致嚴重的後果，以光照為例，因為植物生長燈通常一照便是幾小時，少照幾秒的誤差相對而言不算太嚴重。

本系統未來可以藉由搭配更多感測原件來達成更詳細的觀測以及更完善的操作，例如搭配土壤濕度感測器來完成濕度不夠時自動澆水的設定，具體達成方式如藉感測控制裝置讀取土壤濕度感測器讀數，再由程式判定比較讀數與合適的濕度自動開關澆水器。

## 六、結論

利用本系統便可在出門在外時完成植物的澆水，並且隨時開關光照燈解決植物光照不足的問題，也可透過設定時間來達成每日自動開關光照及澆水，又或是讓系統在陰天或有遮蔽物等環境變暗的情況自動開啟光照燈。

### 附件：

#### 1. 專題程式網址

<https://github.com/s89726559/plantLightingAndWatering>

#### 2. 實作影片網址

<https://youtu.be/fMw-t1IdG38>

<https://youtu.be/B6NcdouKGb4>

### 參照：

Grove 陽光感測器官方

[https://wiki.seeedstudio.com/Grove-Sunlight\\_Sensor/](https://wiki.seeedstudio.com/Grove-Sunlight_Sensor/)

AI2 MQTT Extension introduction page

<https://ullisroboterseite.de/android-AI2-MQTT-en.html>