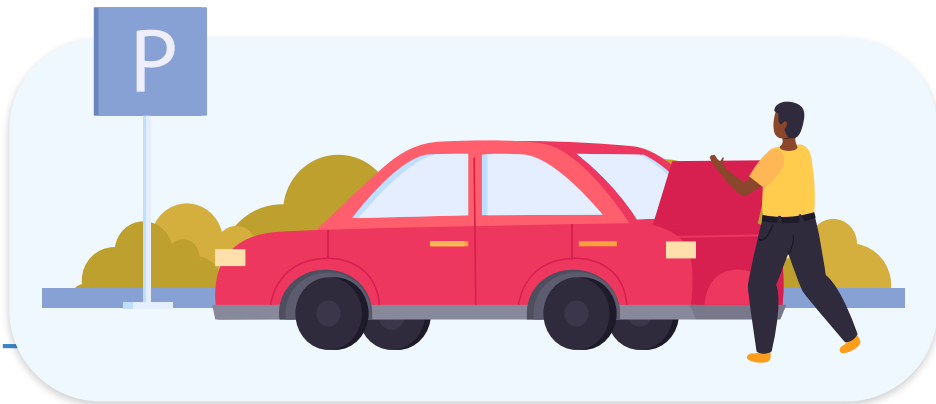


# 智慧校園 汽車eTag比對系統

組員：

A1083313 黃紹倫、A1093324 黃怡婷  
A1093357 張竣維、A1093359 謝益嘉  
A1093361 陳亭瑄、A1093367 陳俐卉

指導教授：蕭漢威 教授





# 目錄



1

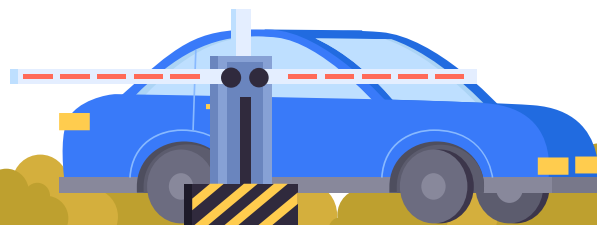
系統架構介紹

2

控制程式說明

3

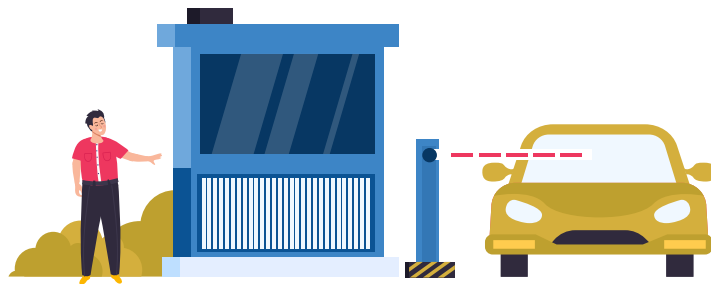
進度說明





1

# 系統架構介紹



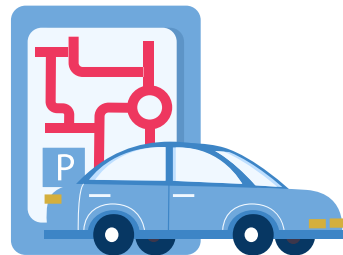


# 系統流程

在各車道口架設RFID感應器，偵測欲進到校內的車輛eTag。

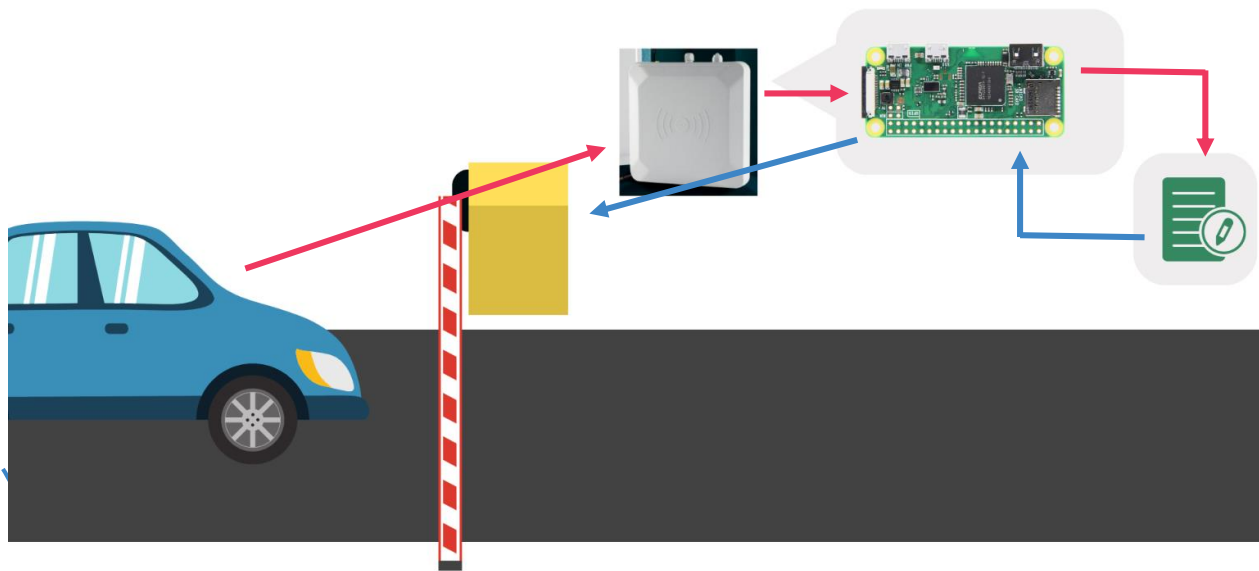
讀取到車輛eTag資料後，與校內資料庫做比對，**查看是否為已授權過的車輛。**

確認車輛已授權後，由透過程式**控制繼電器開關柵欄。**





# 使用示意圖





# 相關器材

eTag 貼紙

eTag 讀取器

繼電器

樹梅派

轉接頭/USB HUB

外接螢幕/鍵盤

# E-tag 格式

設備工作於定時模式下讀到標籤數據輸出格式如下：

1B 39 01 00 12 00 10 81 30 00 AA BB 20 16 03 01 10 80 A0 00 A0 01 00 00 XX

其中：

1B 39---包頭 (這是固定的)

01---設備群組號 (取值範圍0x00-0xFF即0-255，此值可在設備用戶參數中修改)

00---設備地址 (設備地址取值範圍0x00-0xFF即0-255，此值可在設備用戶參數中修改)

12 00---包數據長度(低字節在前)

10---標籤基本數據長度=RSSI(1Byte)+PC(2Byte)+標籤數據(N Byte,N="EPC或TID或EPC+ TID")+天線號(1 Byte)

81---標籤RSSI值

30 00--- 標籤PC值

AA BB 20 16 03 01 10 80 A0 00 A0 01---標籤EPC數據

00---天線號(00表示1號天線讀的數據,01表示2號天線讀到的數據)

00---附加的數據類別

XX---數據校驗和 ( 從標籤基本數據長度到數據校驗和前的所有數據校驗和 )

# E-tag 格式

設備工作於定時模式下讀到標籤數據輸出格式如下：

1B 39 01 00 12 00 10 81 30 00 AA BB 20 16 03 01 10 80 A0 00 A0 01 00 00 XX

其中：

1B 39---包頭 (這是固定的)

01---設備群組號 (取值範圍0x00-0xFF即0-255，此值可在設備用戶參數中修改)

00---設備地址 (設備地址取值範圍0x00-0xFF即0-255，此值可在設備用戶參數中修改)

12 00---包數據長度(低字節在前)

10---標籤基本數據長度=RSSI(1Byte)+PC(2Byte)+標籤數據(N Byte,N="EPC或TID或EPC+ TID")+天線號(1 Byte)

81---標籤RSSI值

30 00--- 標籤PC值

AA BB 20 16 03 01 10 80 A0 00 A0 01---標籤EPC數據

00---天線號(00表示1號天線讀的數據,01表示2號天線讀到的數據)

00---附加的數據類別

XX---數據校驗和 ( 從標籤基本數據長度到數據校驗和前的所有數據校驗和 )



# E-tag 格式

設備工作於定時模式下讀到標籤數據輸出格式如下：

1B 39 01 00 12 00 10 81 30 00 AA BB 20 16 03 01 10 80 A0 00 A0 01 00 00 XX

其中：

1B 39 ---包頭 (這是固定的)

01---設備群組號 (取值範圍0x00-0xFF即0-255，此值可在設備用戶參數中修改)

00---設備地址 (設備地址取值範圍0x00-0xFF即0-255，此值可在設備用戶參數中修改)

12 00---包數據長度(低字節在前)

10---標籤基本數據長度=RSSI(1Byte)+PC(2Byte)+標籤數據(N Byte,N="EPC或TID或EPC+ TID")+天線號(1 Byte)

81---標籤RSSI值

30 00--- 標籤PC值

AA BB 20 16 03 01 10 80 A0 00 A0 01---標籤EPC數據

00---大線號(00表示1號大線讀的數據,01表示2號天線讀到的數據)

00---附加的數據類別

XX---數據校驗和 ( 從標籤基本數據長度到數據校驗和前的所有數據校驗和 )



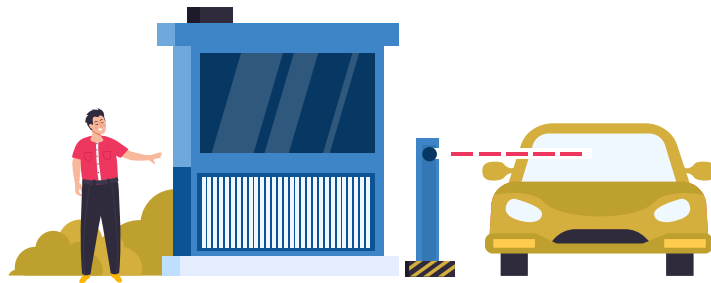
# 架設地





2

# 程式碼說明



# 程式碼

```
1. from ast import Global
2. import RPi.GPIO as GPIO
3. import paho.mqtt.client as mqtt
4. import serial
5. import time
6. import os
```



import 需要的函式庫

```
7. GPIO.setmode(GPIO.BOARD)
8. pin = 7
9. GPIO.setup(pin,GPIO.OUT)
10. sr = serial.Serial("/dev/ttyUSB0",115200)
11. token,answer,tempstr=[],[],[]
12. flag,correct=False,False
13. tag=""
```



設定繼電器控制腳位和RFID  
設備接口、宣告相關變數

```
14. def hex2dec(string_num):
15.     return str(int(string_num, 16))
```



定義函式(將16進制轉成10  
進制，以取得所需etag長度)

# 程式碼

```
18. def output(packs,t):
19.     global tag,tempstr,correct

20.     if len(packs) < 7 or int(hex2dec(packs[6])) > 20:
21.         return

22.     if len(packs) >= (7+int(hex2dec(packs[6]])):
23.         if int(packs[6]) == 0:
24.             ...
25.         else:
26.             correct = False
27.             answer=packs[10:10+int(hex2dec(packs[6]))-4]
28.             tag = ""
29.             for row in answer:
30.                 tag = tag + str(row)
31.
32.         with open("result2.csv","r") as f1:
33.             lines = f1.readlines()
34.             f1.close()
```



篩選掃描到的  
eTag資料內容



開啟CSV檔案讀取  
現有資料

# 程式碼

```
37.     for line in lines:
38.         if line != '\n':
39.             tempstr = line.split(",")
40.             tempstr[1] = tempstr[1].lower().rstrip()
41.             if(tempstr[1] == tag):
42.                 correct = True
43.                 break
```

比對eTag和檔案內  
資料是否相符

```
44.     if correct:
45.         GPIO.output(pin,True)
46.         time.sleep(1)
47.     else:
48.         GPIO.output(pin,False)
49.         str1 = ""
```

若相符則開啟繼電器  
開關

# 程式碼

```
52. try:
53.     while(True):
54.         data = sr.read().hex()
55.         if data != "":
56.             t1=time.time_ns()
57.             if data == '1b':
58.                 flag=True
59.                 continue
60.             if flag:
61.                 if data == '39':
62.                     output(token,t1)
63.                     t1=time.time_ns()
64.                     flag=False
65.                     token=['1b','39']
66.                     continue
67.             token.append(data)
68.             if len(token) > 50:
69.                 token = []
70. finally:
71.     GPIO.cleanup()
```



確認掃描到的eTag資料是1b 39開頭，紀錄當前時間，放入output函式篩選資料並比對

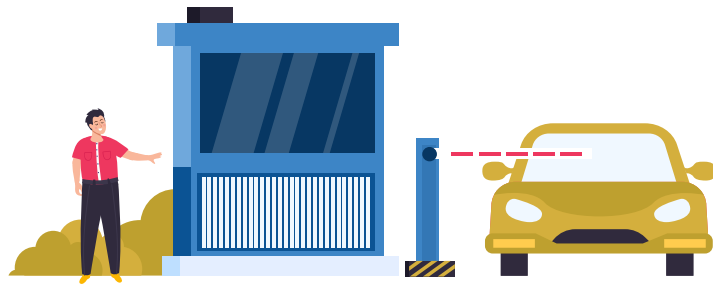


初始化繼電器設定



3

# 進度說明





## 控制程式

```

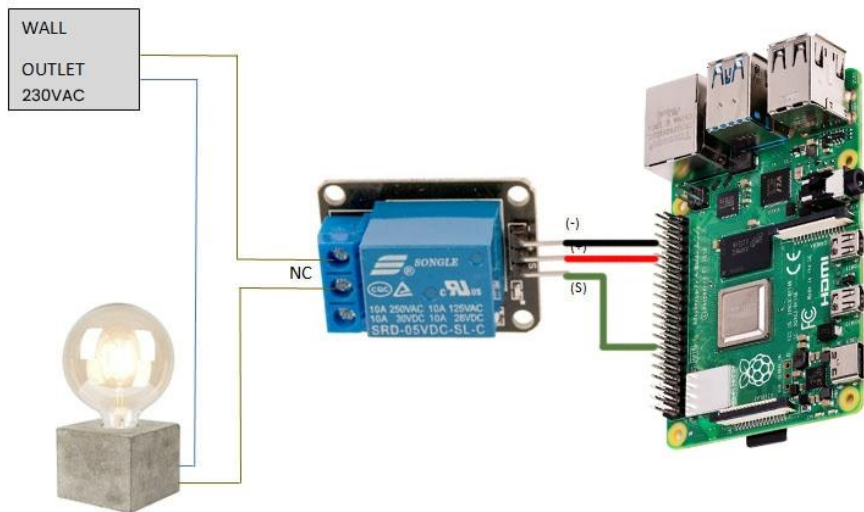
1 from sys import stdout
2 import RPi.GPIO as GPIO
3 import paho.mqtt.client as mqtt
4 import serial
5 import time
6 import os
7
8 GPIO.setmode(GPIO.BOARD)
9 pin = 7
10 GPIO.setup(pin, GPIO.OUT)
11 sr = serial.Serial("/dev/ttyUSB0", 115200)
12 token, answer, temp = [], [], []
13 flag, correct = False, False
14 tag = ""
15
16 def hex2dec(string_num):
17     return str(int(string_num, 16))
18
19 def output(packs, i):
20     global tag, tempstr, correct
21
22     if len(packs) < 7 or int(hex2dec(packs[i])) > 20:
23         return
24
25     if len(packs) >= 7:
26         if int(hex2dec(packs[i])):
27             correct = True
28         else:
29             correct = False
30         answer = packs[10:10+int(hex2dec(packs[i]))-4]
31         tag = ""
32         for row in answer:
33             tag = tag + str(row)
34
35     with open("result2.csv", "a") as f1:
36         lines = f1.readlines()
37         f1.close()
38     for line in lines:
39         if line != '\n':
40             tempstr = line.split(":").lower().rstrip()
41             if tempstr[i] == tag:
42                 correct = True
43                 break
44
45     if correct:
46         GPIO.output(pin, True)
47     else:
48         GPIO.output(pin, False)
49     str1 = ""
50
51 try:
52     while(True):
53         data = sr.read().hex()
54         if data != '':
55             ti = time.time_ns()
56             if data == '\b':
57                 flag = True
58                 continue
59             if flag:
60                 if data == '\n':
61                     output(token, ti)
62                     ti = time.time_ns()
63                     flag = False
64                     token = ["\b", "\n"]
65                     continue
66             token.append(data)
67
68             if len(token) > 50:
69                 token = []
70
71 finally:
72     GPIO.cleanup()

```

## 車牌紀錄

	A	B
1	2912R5	105023ab8105440846455443
2	6089UZ	105023ab8105440846455443
3	BDY2739	6810166868555e0000000001
4	BAH7150	105013ad1036924a46455443
5	7672MJ	102523cf101676a946455443
6	RCZ8103	105033af504711da46455443
7	RCZ8113	105033af504711da46455443
8	ASY5505	105013aa8476161d46455443
9	BJD3639	105013ae2959594446455443
10	BAN7976	105013ad761732ad46455443
11	BDP1997	105033ad031725ef46455443
12	6117XC	100613cd114944d446455443
13	BJT3309	105033af0788441746455443
14	AQU0698	102813db314373f146455443
15	AML8730	102813cf1170159046455443
16	APH5100	105033ae1258990e46455443
17	2377ZX	105033ad0564398e46455443
18	RCZ8133	105033af5182787746455443
19	BNF9610	105033ba0600834c46455443
20	KN9755	105023ae812477146455443
21	BJT6720	105033af306446ce46455443
22	AGT8227	102813bd106035d346455443
23	4378E9	100813bee0299ce46455443
24	652633	103323aa114710ec46455443
25	BED5625	105033ae3824945546455443
26	ZR3767	101413db1038779346455443
27	0565TG	105023ac2688503146455443
28	BDT1100	105033af0226887046455443
29	AFT5766	102113aa104520fd46455443
30	BDJ6336	105033ac730775de46455443
31	6T6208	100723bc6116728d46455443
32	BND2895	105033af6906388b46455443
33	1403ZL	100723bd111570fd46455443
34	ANY3986	102813be5071813546455443
35	7102UY	100713ce1071973c46455443
36	AKE0702	105013ba0932994b46455443
37	AQL6200	105013aa5197734c46455443
38	BAC9298	105033ae108993d746455443
39	ADA6060	101013acd15189b646455443
40	AQW8822	105023ab2128915046455443
41	6223F3	104223add020456946455443
42	BFL3102	105033ae166250946455443
43	7768XM	101423aa103271de46455443
44	AFZ8135	102713ca412368e446455443
45	BML8162	105033af2309738546455443
46	0019ZX	105013ae0553923946455443
47	BJT1610	100713ce1116694a46455443
48	AML8715	103813ac5015548646455443
49	5372G9	100713bd511233c946455443
50	AVR7975	105013ab5109379746455443
51	2326LS	101513aa5080513d46455443
52	AAJ5193	105013ac6819961846455443
53	BAF5572	105013aa533651cd46455443

## 繼電器介接



## GPIO

### Raspberry Pi GPIO Header A+, B+, Zero, Pi2

Pin#	NAME		NAME	Pin#
01	3.3v DC Power	Red	DC Power 5v	02
03	GPIO02 (SDA1 , I <sup>2</sup> C)	Blue	DC Power 5v	04
05	GPIO03 (SCL1 , I <sup>2</sup> C)	Blue	Ground	06
07	GPIO04 (GPIO_GCLK)	Green	(TXD0) GPIO14	08
09	Ground	Black	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	Green	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Green	Ground	14
15	GPIO22 (GPIO_GEN3)	Green	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	Red	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Purple	Ground	20
21	GPIO09 (SPI_MISO)	Purple	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	Purple	(SPI_CE0_N) GPIO08	24
25	Ground	Black	(SPI_CE1_N) GPIO07	26
27	ID_SD (I <sup>2</sup> C ID EEPROM)	Yellow	(I <sup>2</sup> C ID EEPROM) ID_SC	28
29	GPIO05	Green	Ground	30
31	GPIO06	Green	GPIO12	32
33	GPIO13	Green	Ground	34
35	GPIO19	Green	GPIO16	36
37	GPIO26	Green	GPIO20	38
39	Ground	Black	GPIO21	40

Rev. 1.5  
09/02/2016

www.element14.com/RaspberryPi



# Thanks!

