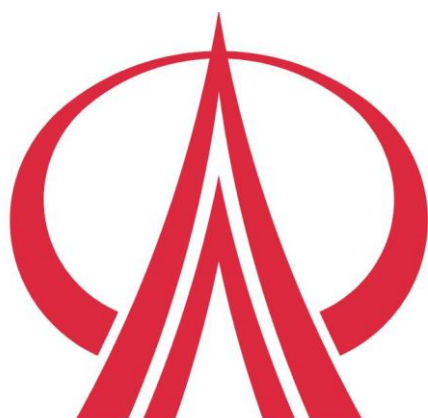


修平科技大學 電機工程系

DEPARTMENT OF ELECTRICAL ENGINEERING
HSIUPING UNIVERSITY OF SCIENCE AND TECHNOLOGY

實務專題報告書

物聯網應用於智慧空間環境



指導老師：劉文韻

專題製作學生：

四技電三乙 林家丞 BD104088

四技電三乙 林柏寬 BD104093

中華民國 107 年 6 月 25 日

修平科技大學

電機工程系

**HSIU-PING UNIVERSITY OF SCIENCE
AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING**

指導老師：劉文韻

專題製作學生：林家丞、林柏寬

製作日期：107 年 6 月 25 日

摘要

我們的專題是物聯網應用於智慧空間環境，要模擬一個停車場的監控跟遠端操控開門的實驗，此次我們運用 Arduino nano328 晶片進行專題模擬，使用光敏電阻傳感器接收光照度的訊號，再藉由 cc2541 藍芽模組收發訊號，傳送至 Arduino nano328 進程式編譯，以及 App Inventor2 寫入手機程式，再經由 cc2541 藍芽模組發射訊號製手機，用手機對整個停車場進行監控與控制，在進行的過程中：(1)先架設好光敏電阻傳感器(2)架設 cc2541 藍芽模組(3)架設 Arduino nano328 晶片(4)將 cc2541 和光敏電阻傳感器接至 Arduinonano328 晶片上進行實驗(5)最後再調整光敏電阻傳感器的接收強度已完成整個實驗。對於現今的科技來說，物聯網已經是一項不可或缺的技术，本次專題應用物聯網在智慧停車場做實驗，相信未來的停車場，一定都能走向科技化，為生活帶來更方便的選擇。

目錄

摘要	2
圖目錄	5
第一章 緒論	7
1.1 研究動機	7
1.2 作品特色	8
第二章 工作原理探討	9
2.1 晶片原理:cc2541	9
2.2 晶片原理:Arduino nano328	15
2.3 光敏電阻傳感器	18
第三章 系統架構	21
3.1 硬體架設	21
3.1.1 Arduino nano328 架設	22
3.1.2 cc2541 架設	23
3.2 App Inventor 2	24
3.3 程式說明	25
3.3.1 Arduino nano328 程式	25
3.3.2 App Inventor 2 程式	28

第四章 測試	30
4.1 系統整合測試	30
4.1.1 第一階段-光敏電阻強度測試	30
4.1.2 第二階段-停車狀況測試	32
4.1.3 第三階段-閘門開關測試	33
第五章 結論	34
參考文獻	35

圖目錄

圖 2. 1: cc2541 藍芽	9
圖 2. 2: Arduino nano	15
圖 2. 3: 光敏電阻傳感器	18
圖 2. 4: 光敏電阻動作圖	20
圖 3. 1: 系統架構圖	21
圖 3. 2: 板子後方	21
圖 3. 3: Arduino nano 放大圖	22
圖 3. 4: cc2541 板子正面	23
圖 3. 5: cc2541 板子背面	23
圖 3. 6: APP Inventor 2 介面	24
圖 3. 7: APP Inventor 手機介面	24
圖 3. 8: Arduino 程式	25
圖 3. 9: Arduino 程式	26
圖 3. 10: Arduino 程式	27
圖 3. 11: Arduino 程式	27
圖 3. 12: APP Inventor 程式	28
圖 3. 13: APP Inventor 程式	28

圖 3. 14: APP Inventor 程式	29
圖 3. 15: APP Inventor 程式	29
圖 3. 16: APP Inventor 程式	29
圖 4. 1: 測試過程	30
圖 4. 2: 有光照	31
圖 4. 3: 無光照	31
圖 4. 4: 停車測試	32
圖 4. 5: 停車測試	32
圖 4. 6 閘門開啟時	33
圖 4. 7 閘門關閉時	33

第一章

緒論

1.1 研究動機

由於觀光景點常常塞車，我去了解原因發現停車場外過多排隊車潮進而影響交通，如果有個 APP 能夠先知道當地的停車場位置以及剩餘的車位，那我想就能有效地去紓解因為排隊而占用到車道擁塞的情形。除了觀光景點的停車場之外，像是市區停車場月租型用戶，也常常要跟著排隊車流排隊，明明是月租型卻好像一般停車用戶一樣，所以如果能運用手機藍芽技術，就可以使月租型用戶更方便的進出。

1.2 作品特色

(1)Arduino nano328 是一款相當實用的晶片，是一個市面上廣用的晶片，具有多隻腳位可供使用，能讓整體程式更加靈活。

(2) CC2541 非常適合應用於需要超低能耗的系統。這由多種不同的運行模式指定。運行模式間較短的轉換時間進一步使低能耗變為可能。

(3)光敏電阻傳感器採用靈敏型光敏電阻感測器，比較器輸出，信號乾淨，波形好，驅動能力強，超過 15mA。配可調電位器可調節檢測光線亮度，設有固定螺栓孔，方便安裝。

第二章

工作原理探討

2.1 晶片原理:cc2541



圖 2.1:cc2541 藍芽

如圖 2.1 CC2541 是一款針對低能耗以及擁有 2.4-GHz 應用的功率優化的真正片載系統(SoC)解決方案。它使得使用低整體物料清單成本建立強健網路節點成為可能。CC2541 將領先 RF 收發器的出色性能和一個業界標準的增強型 8051 MCU、系統內可編程快存儲存器、8-KB RAM 和很多其它功能強大的特性和外設組合在一起。

iBeacon 技術核心 BLE，目前已有整合型嵌入式晶片。德州儀器(TI) 推出最新 CC2541 藍牙 (Bluetooth®) 低耗能系統單晶片 (SoC)，旨在滿足消費性醫療、運動健身、安全、娛樂以及家庭自動化 Bluetooth Smart 感測器應用需求。與 TI 前代 CC2540 SoC 相比，該 Soc 可在輸出功率為 1mW 時將功耗降低 33%，充分發揮省電優勢。2.4 GHz CC2541 SoC、免專利費 TI 協定堆疊、基本軟體 (profiles software) 樣品應用、開發套件、技術檔案與支援。使用高彈性、高整合度的內建解決方案簡化設計；支援藍牙 4.0 版以及 250Kbps、500Kbps、1Mbps 及 2Mbps 的專有模式；經驗證、穩健的藍牙技術建立在 CC2540 的成功基礎之上。實現可在單一硬幣型 (coin cell) 電池上運行超過一年的低功耗感測器產品。

射頻

- (1) 2.4-GHz 符合低能耗規範和私有的 RF 片載系統。
- (2) 支持 250-kbps，500-kbps，1-Mbps，2-Mbps 的數據速率。
- (3) 出色的鏈路預算，不使用外部前段而支持長距離運用。
- (4) 高達 0 dBm 的可編程輸出功率。
- (5) 出色的接收器靈敏度（1 Mbps 時為 -94 dBm），可選擇性，和阻擋性能。
- (6) 適合於針對符合世界範圍內的無線電頻率調節系統：ETSI EN 300 328 和 EN 300 440 2 類（歐洲），FCC CFR47 15 部分（美國），和 ARIB STD-T66（日本）。

布局

- (1)極少的外部組件
- (2)提供參考設計
- (3)6-mm × 6-mm 方形扁平無引腳 (QFN)-40 封裝
- (4)與 CC2540 接腳相容 (當不使用 USB 或者 I²C 時)

低功率

- (1)工作模式 RX 低至： 17.9 mA
- (2)工作模式 TX (0 dBm)： 18.2 mA
- (3)功率模式 1 (4- μ s 喚醒)： 270 μ A
- (4)功率模式 2 (睡眠定時器打開)： 1 μ A
- (5)功率模式 3 (外部中斷)： 0.5 μ A
- (6)廣泛的電源電壓範圍 (2 V - 3.6 V)

工作模式下 TPS62730 相容低功率

- (1)RX 低至： 14.7 mA (3-V 電源)
- (2)TX (0 dBm)： 14.3 mA (3V 電源)

微控制器

- (1) 具有代碼預取功能的高性能和低功率 8051 微控制器內核。
- (2) 系統內可編程快存，128 或者 256 KB。
- (3) 在所有功率模式下具有保持功能的 8-KB RAM。
- (4) 支持硬件調適。
- (5) 擴展基礎自動化，包括自動確認和地址解碼。
- (6) 所有功率模式中對所有相關寄存器的保持。

外設

- (1) 功能強大的 5 通道直接內存訪問 (DMA)。
- (2) 通用定時器 (1 個 16 位，2 個 8 位)。
- (3) 紅外 (IR) 生成電路。
- (4) 具有捕捉功能的 32-kHz 睡眠定時器。
- (5) 精確數字接收到的數字信號強度指示器 (RSSI) 支持。
- (6) 電池監控器和溫度傳感器。
- (7) 含 8 通道和可配置分辨率的 12 位模數轉換器 (ADC)。
- (8) 高級安全標準 (AES) 安全處理器。

(9) 2 個功能強大的支持幾個串行協議的通用異步接收發器。

(10) 23 個通用 I/O 接腳。

(11) I²C 接口。

(12) 2 個具有 LED 驅動功能的 I/O 接腳。

(13) 安全裝置定時器。

(14) 集成的高性能比較器。

2.2 晶片原理:Arduino nano

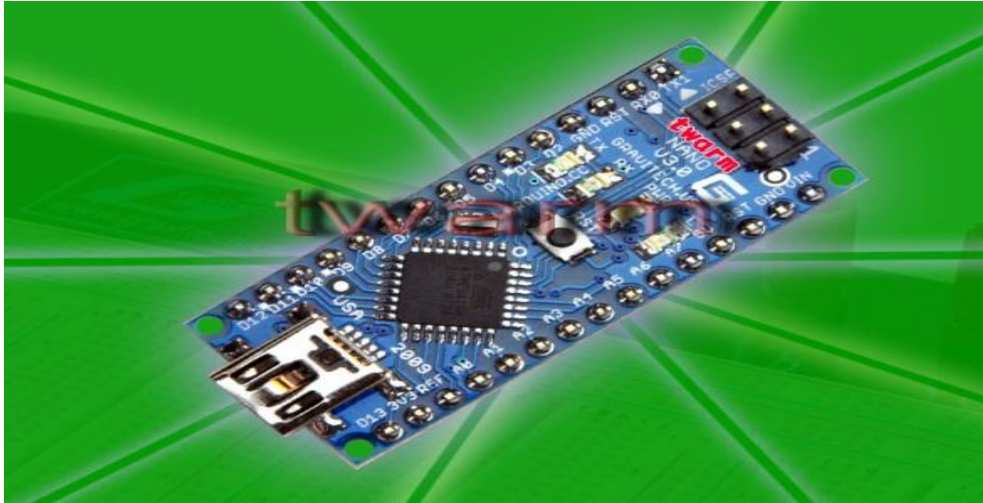


圖 2.2 Arduino nano

Arduino nano 如圖 2.2 是一塊基於開放原始代碼的 Simple i/o 平台，並且具有使用類似 java 跟 C 語言的開發環境。讓您可以快速使用 Arduino 語言與 Flash 或 Processing... 等軟件，作出互動作品。Arduino 可以使用開發完成的電子元件例如 Switch 或 Sensors 或其他控制器、LED、步進電機或其他輸出裝置。Arduino 也可以獨立運作成為一個可以跟軟件溝通的平台，例如說：flash processing Max/MSP VVVV 或其他互動軟件...，Arduino 開發 IDE 界面基於開放原始碼原則，可以讓您免費下載使用開發出更多令人驚奇的互動作品。

Arduino nano 技術規格

- 1 · Digital I/O 數字輸入/輸出端共 0~13。
- 2 · Analog I/O 模擬輸入端共 0~7。
- 3 · 支持 USB 接口協議及供電(不需外接電源)。
- 4 · 支持 ISP 下載功能。
- 5 · 支持單片機 TX/RX 端子。
- 6 · 支持 AREF 端子。
- 7 · 支持六組 PWM 端子(Pin11, Pin10, Pin9, Pin6, Pin5, Pin3)。
- 8 · 輸入電壓：接上 USB 時無須外部供電或外部 7V~12V DC 輸入。
- 9 · 輸出電壓：5V DC 輸出和 3.3V DC 輸出 和外部電源輸入。
- 10 · 採用 Atmel Atmega328PAU 單片機。
- 11 · nano 大小尺寸：寬 18mm X 高 45mm。

Arduino nano 特點

1. 開放原始碼的電路圖設計，程式開發界面免費下載，也可依需求自己修改!!
2. Arduino 可使用 ISP 下載線，自我將新的 IC 程序燒入「bootloader」;
3. 可依據官方電路圖，簡化 Arduino 模組，完成獨立運作的微處理控制器；
4. 可簡單地與傳感器、各式各樣的電子元件連接(如：紅外線，超聲波，熱敏電阻，光敏電阻，伺服電機等)；
5. 支持多樣的互動程式 如: Flash，Max/Msp，PD，C，Processing 等；
6. 使用低價格的微處理控制器(ATMEGA168-10AU)；
7. USB 接口供電，不需外接電源，電流保護二極管避免你的電腦 USB 電源過載，USB 與外接電源自動切換；
8. 應用方面，利用 Arduino，突破以往只能使用鼠標，鍵盤，CCD 等輸入的裝置的互動內容，可以更簡單地達成單人或多人遊戲互動。

2.3 光敏電阻傳感器

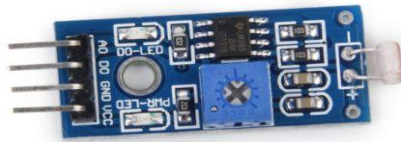


圖 2.3: 光敏電阻傳感器

光敏電阻如圖 2.3 所示，其介紹如下

- (1) 光敏電阻模組對環境光線最敏感，一般用來檢測周圍境的光亮度觸發單片機或繼電器模組等。
- (2) 模組在環境光線亮度達不到設定閾值時，D0 端輸出高電平，當外界環境光線亮度超過設定閾值時，D0 端輸出低電平。
- (3) D0 輸出端可以與單片機直接相連，通過單片機來檢測高低電平，由此來檢測環境的光線亮度改變。
- (4) D0 輸出端可以直接驅動本店繼電器模組，由此可以組成一個光控開關。

光敏電阻其特色介紹如下

1. 採用靈敏型光敏電阻感測器。
2. 比較器輸出，信號乾淨，波形好，驅動能力強，超過 15mA。
3. 配可調電位器可調節檢測光線亮度。
4. 工作電壓 3.3V-5V。
5. 輸出形式：D0 數位開關量輸出（0 和 1）和 A0 類比電壓輸出。
6. 設有固定螺栓孔，方便安裝。
7. 小板 PCB 尺寸：3.2cm x 1.4cm。
8. 使用寬電壓 LM393 比較器。

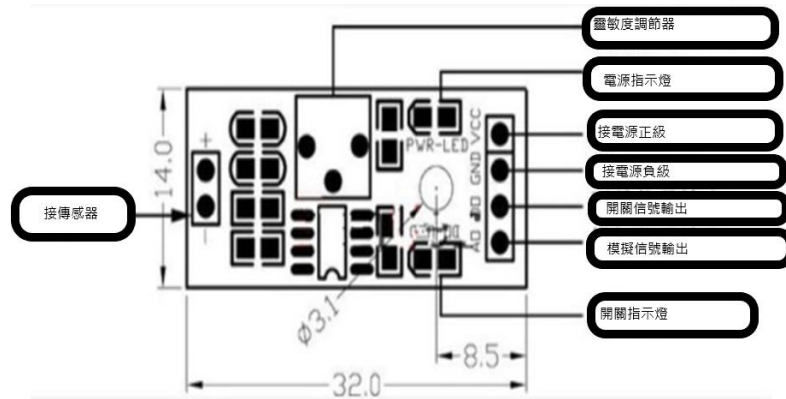


圖 2.4: 光敏電阻動作圖

如圖 2.4 所示為光敏電阻動作圖，其動作介紹如下

- 1、光敏電阻模組對環境光線最敏感，一般用來檢測周圍環境的光線的亮度，觸發單片機或繼電器模組等。
- 2、模組在環境光線亮度達不到設定閾值時，D0 端輸出高電平，當外界環境光線亮度超過設定閾值時，D0 端輸出低電平。
- 3、D0 輸出端可以與單片機直接相連，通過單片機來檢測高低電平，由此來檢測環境的光線亮度改變。
- 4、D0 輸出端可以直接驅動本店繼電器模組，由此可以組成一個光控開關。
- 5、小板模擬量輸出 A0 可以和 AD 模組相連，通過 AD 轉換，可以獲得環境光強更精準的數值。

第三章 系統架構

3.1 硬體架設

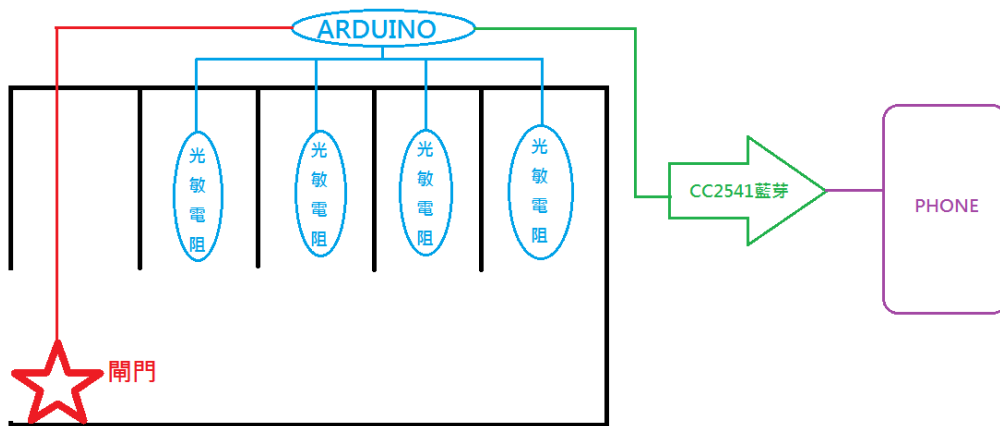


圖 3.1:系統架構圖

我們規劃在壓克力板上，呈現一座小型停車場，首先在四個停車格上鑽上小洞，讓線可以從小洞中拉至板子後方接上 Arduino nano，之後再將 Arduino nano 架設於板子後方使所有元件接至 Arduino nano 上如圖 3.2

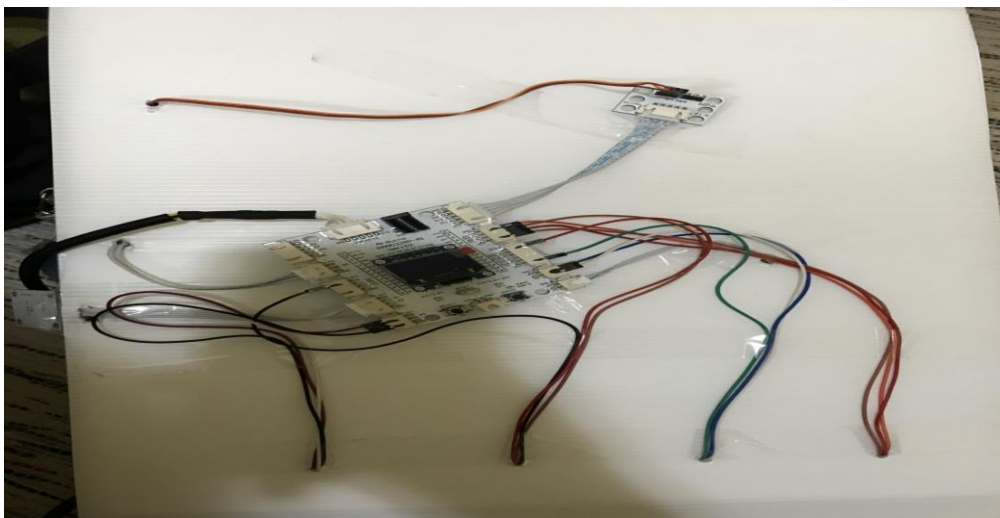


圖 3.2:板子後方

3.1.2 CC2541 架設

由於 CC2541 藍芽模組，是我們收發訊號的主要元件，因此必須將它設置在正面才能使其能夠準確的去收發我們手機端所發出來的訊號，所以先開洞，使線能穿至後方連接到 Arduino nano328。如下圖 3.4 跟圖 3.5 所示。



圖 3.4:cc2541 板子正面

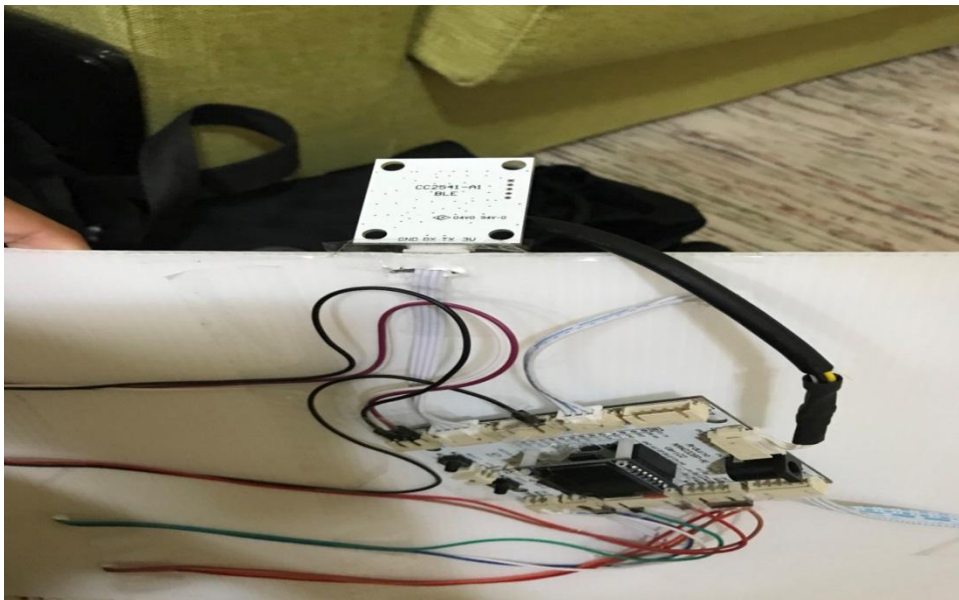


圖 3.5:cc2541 板子背面

3.2 App Inventor 2

APP Inventor 2 是一款用來寫手機 app 的程式軟體，此次我們專題也有運用到這款程式，透過程式將手機與藍芽跟停車場連結，使手機可以控制停車場。如下圖 3.6 所示，將我們停車場的模擬圖檔輸入至 APP Inventor 2 中，再透過藍芽連線，傳送至手機，使手機能跟我們設計的程式同步如圖 3.7。

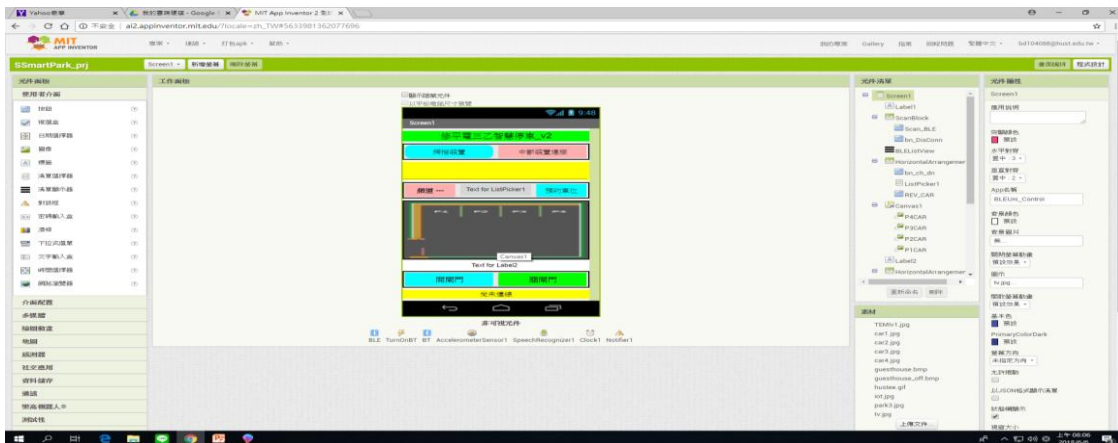


圖 3.6: APP Inventor 2 介面

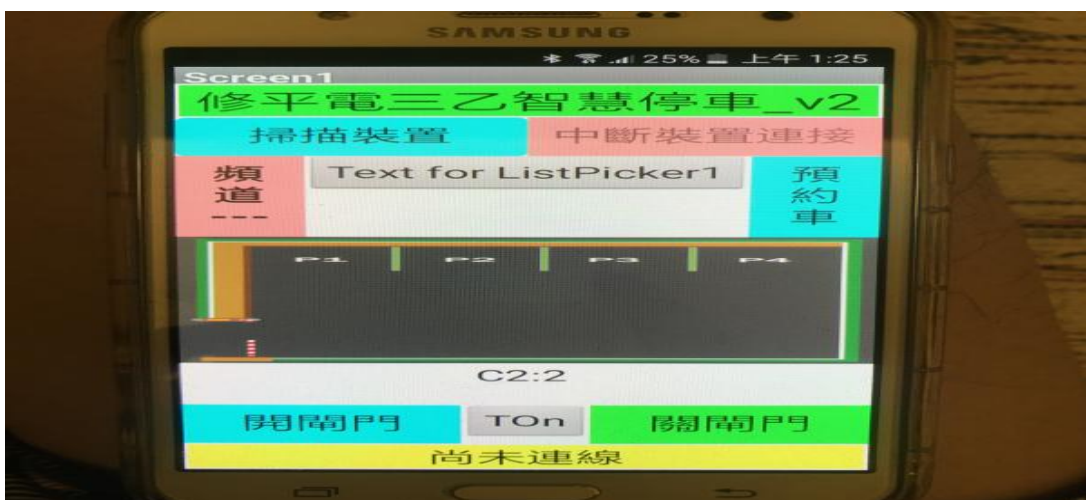


圖 3.7: APP Inventor 手機介面

3.3 程式說明

3.3.1 Arduino nano 328 程式

```
BLE_smartpark | Arduino 1.8.1 Hourly Build 2016/12/29 04:34
檔案 編輯 草稿碼 工具 說明
BLE_smartpark
//-----
// Title:BLE_smartpark
// Firmware:Ex0_KEY_LED
// Devices: NanoKIT + KEY_LED
// Function:KEY_LED
// UpDate:2018/05/22
#include <Servo.h>// 建立一個 Servo 物件lude <Servo.h>// 建立一個 Servo 物件
#define CAR_DOOR_PIN 10
Servo myservo;
void CarDoorInit();
void CarDoorSet(uint8_t istatus);
//-----
//步驟1
#define LUX_DO_PIN1 A3
#define LUX_DO_PIN2 A4
#define LUX_DO_PIN3 A5
#define LUX_DO_PIN4 5

//-----Declare-LED1W-MD1-----
#define LED 13
#define LED4 3
//-----Declare-RGB LED WS2812Driver -----
#include "FastLED.h"
#define LED_PIN AO
#define NUM_LEDS 3
CRGB leds[NUM_LEDS];
#define BRIGHTNESS 64
//-----Declare-KEY-----
#define KEY1 2
#define KEY2 6
//-----Declare-OLED-----
#include <OLED.h>
OLED myOLED;
//-----BLE-----
void BLE_setName(char blename[]);
#include <SoftwareSerial.h>
SoftwareSerial myBLE(A1, A2); // RX, TX
int DF=0;
int PartLotState[]={0,0,0,0,0};
int PartTempState[]={0,0,0,0,0};

Using proxy DIRECT
Using proxy DIRECT
Using proxy DIRECT
在D:\arduino\libraries\TCS3200: D:\arduino\libraries\TCS3200裡找到無效的程式庫
33
```

圖 3.8:Arduino 程式

如上圖 3.8 所示，此段程式是在定義光敏電阻傳感器的腳位，我們將

腳位定在 A3、A4、A5、D5。



```
BLE_smartpark | Arduino 1.8.1 Hourly Build 2016/12/29 04:34
檔案 編輯 草稿碼 工具 說明
BLE_smartpark

void setup() {
  //=====Setup-RGB=====
  delay( 100 ); // power-up safety delay
  FastLED.addLeds<WS2812B, LED_PIN, GRB>(leds, NUM_LEDS).setCorrection( TypicalLEDStrip );
  FastLED.setBrightness(BRIGHTNESS );
  SET_RGB(1 , 'D');SET_RGB(2 , 'D');SET_RGB(3 , 'D');
  CarDoorInit();
  // put your setup code here, to run once:
  //=====Setup-ED1W-MD1=====
  pinMode(LED, OUTPUT);
  pinMode(LED4, OUTPUT);
  //=====Setup-KEY=====
  // declare KEY1_PIN|KEY2_PIN to be an output:
  pinMode(KEY1, INPUT_PULLUP);
  pinMode(KEY2, INPUT_PULLUP);
  //=====LuxDo=====
  //步驟2
  pinMode(LUX_DO_PIN1, INPUT_PULLUP);
  pinMode(LUX_DO_PIN2, INPUT_PULLUP);
  pinMode(LUX_DO_PIN3, INPUT_PULLUP);
  pinMode(LUX_DO_PIN4, INPUT_PULLUP);
  //=====Setup-OLED FONT_8x16=====
  myOLED.begin(FONT_8x16); //FONT_6x8 , FONT_8x16
  myOLED.clearScreen();
  //=====SETUP BLE=====
  myBLE.begin(9600);
  BLE_setName("SmartPark");
  Serial.begin(9600);
}
//-----
void loop() {

  //步驟3
  uint16_t Lux_Val1 = 0;
  uint16_t Lux_Val2 = 0;
  uint16_t Lux_Val3 = 0;
  uint16_t Lux_Val4 = 0;

  //步驟4
  Lux_Val1 = digitalRead(LUX_DO_PIN1);
  // Lux_Val2 = digitalRead(LUX_DO_PIN2);
  // Lux_Val3 = digitalRead(LUX_DO_PIN3);
  // Lux_Val4 = digitalRead(LUX_DO_PIN4);

  Using proxy DIRECT
  Using proxy DIRECT
  Using proxy DIRECT
  在D:\arduino\libraries\TCS3200: D:\arduino\libraries\TCS3200裡找到無效的程式庫

  23
```

圖 3.9:Arduino 程式

如上圖 3.9 所示，此段程式是在定義光敏電阻訊號的讀取值，先定義接腳，再將初始值設為 0，並且讀取數位訊號。

```

BLE_smartpark | Arduino 1.8.1 Hourly Build 2016/12/29 04:34
檔案 編輯 監視器 工具 說明

BLE_smartpark
Lux_Va11 = digitalRead(LUX_DO_FIN1);
Lux_Va12 = digitalRead(LUX_DO_FIN2);
Lux_Va13 = digitalRead(LUX_DO_FIN3);
Lux_Va14 = digitalRead(LUX_DO_FIN4);
//=====
if(Lux_Va11==0)
    PartLotState[0]=2;
else
    PartLotState[0]=0;
//=====
if(Lux_Va12==0)
    PartLotState[1]=2;
else
    PartLotState[1]=0;
//=====
if(Lux_Va13==0)
    PartLotState[2]=2;
else
    PartLotState[2]=0;
//=====
if(Lux_Va14==0)
    PartLotState[3]=2;
else
    PartLotState[3]=0;

noOLED.setPos(0,0);
myOLED.println("Smart Park BLE");
//=====
int BnState = digitalRead( KEY1);
if(BnState==0)
{
    delay(500); // delay in between reads for stability
    myOLED.setPos(2,0);
    if(DF==5)
        DF=1;
    else
        DF=DF+1;
    myOLED.setPos(6,0);
}

Using proxy DIRECT
Using proxy DIRECT
Using proxy DIRECT
在D:\arduino\libraries\TCS3200: D:\arduino\libraries\TCS3200裡找到無效的程式碼庫
57

```

圖 3.10: Arduino 程式

```

BLE_smartpark | Arduino 1.8.1 Hourly Build 2016/12/29 04:34
檔案 編輯 監視器 工具 說明

BLE_smartpark
myBLE.write(0x9A);
myBLE.write(0xBB);
delay(200);
while(myBLE.available())
{
    myBLE.read();
}
}
void SEND_PARK()
{
    if(PartLotState[0]!=PartTempState[0])
    {
        PartTempState[0]=PartLotState[0];
        Serial.print("C1:");
        Serial.println(PartLotState[0]);
        myBLE.print("C1:");
        myBLE.println(PartLotState[0]);
        myOLED.print("Send P1:");
        myOLED.println(PartLotState[0]);
    }
    if(PartLotState[1]!=PartTempState[1])
    {
        PartTempState[1]=PartLotState[1];
        Serial.print("C2:");
        Serial.println(PartLotState[1]);
        myBLE.print("C2:");
        myBLE.println(PartLotState[1]);
        myOLED.print("Send P2:");
        myOLED.println(PartLotState[1]);
    }
    if(PartLotState[2]!=PartTempState[2])
    {
        PartTempState[2]=PartLotState[2];
        Serial.print("C3:");
        Serial.println(PartLotState[2]);
        myBLE.print("C3:");
        myBLE.println(PartLotState[2]);
        myOLED.print("Send P3:");
        myOLED.println(PartLotState[2]);
    }
    if(PartLotState[3]!=PartTempState[3])
}

Using proxy DIRECT
Using proxy DIRECT
Using proxy DIRECT
在D:\arduino\libraries\TCS3200: D:\arduino\libraries\TCS3200裡找到無效的程式碼庫
182

```

圖 3.11: Arduino 程式

如上圖 3.10、圖 3.11 所示，此兩段程式為本次專題主程式，圖(十四)是在控制光敏電阻收發訊號，如接收到有光照則為 2，無光照為 1，再透過圖(十五)程式跟手機上停車資訊做反應。

3.3.2 APP Inventor 程式



圖 3.12:APP Inventor 程式

如上圖 3.12 所示，此段 APP 程式，是在定義手機上開門的開關程式，當按下開時開門開，反之按下關則關。



圖 3.13:APP Inventor 程式

如上圖 3.13 所示，此段程式是開關程式，開門控制必須先點選開關才能夠控制開門。



圖 3.14: APP Inventor 程式



圖 3.15: APP Inventor 程式



圖 3.16: APP Inventor 程式

如上圖 3.14 、3.15 、3.16 所示，此段程式是在說明停車場停車狀況的程式，參照圖 3.10 當讀取的值為 2 時為有光照，代表沒車，反之值為 0 時為無光照，代表有車。

第四章 測試

4.1 系統整合測試

4.1.1 第一階段-光敏電阻強度測試

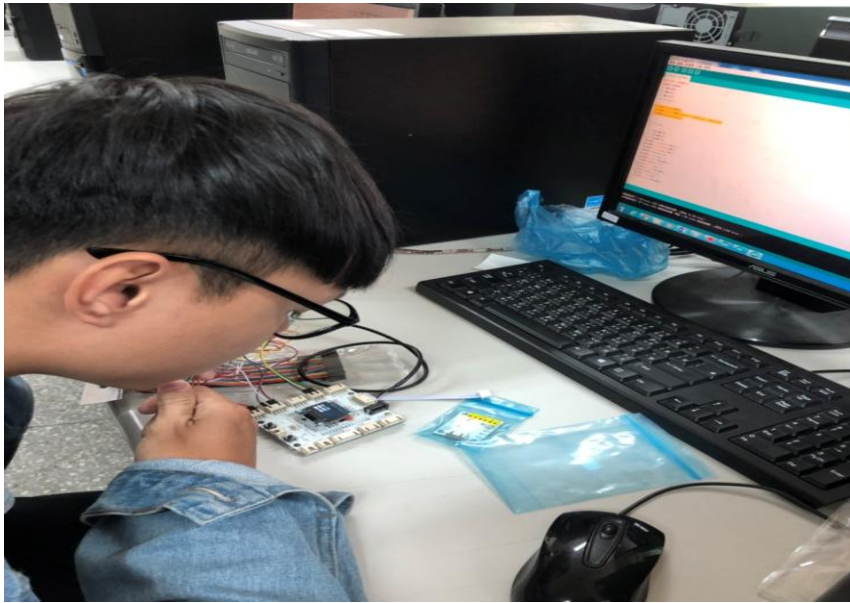


圖 4.1:測試過程

如圖 4.1，將 Arduino 程式編譯完後，上傳至 Arduino nano 328，透過 Arduino nano 328 上的小型螢幕，去做光敏電阻傳感器調整光線接收強度的訊號，已方便我們去改寫程式。



圖 4.2:有光照



圖 4.3:無光照

如圖 4.2 為圖 4.1 敘述的放大圖，由圖 4.1 我們可以看見三個數值都偏小，此顯示狀態代表有光照度反應，意思就是當前停車場為沒有車的狀態，相反的如圖 4.3 所示，三個數值都偏大，此顯示狀態代表無光照度反應，意思也就是當前停車場為有車狀態，透過小螢幕顯示，我們就能透過數值的顯示大小，適當地去調整我們光敏電阻傳感器的光感強度。

4.1.2 第二階段-停車狀況測試

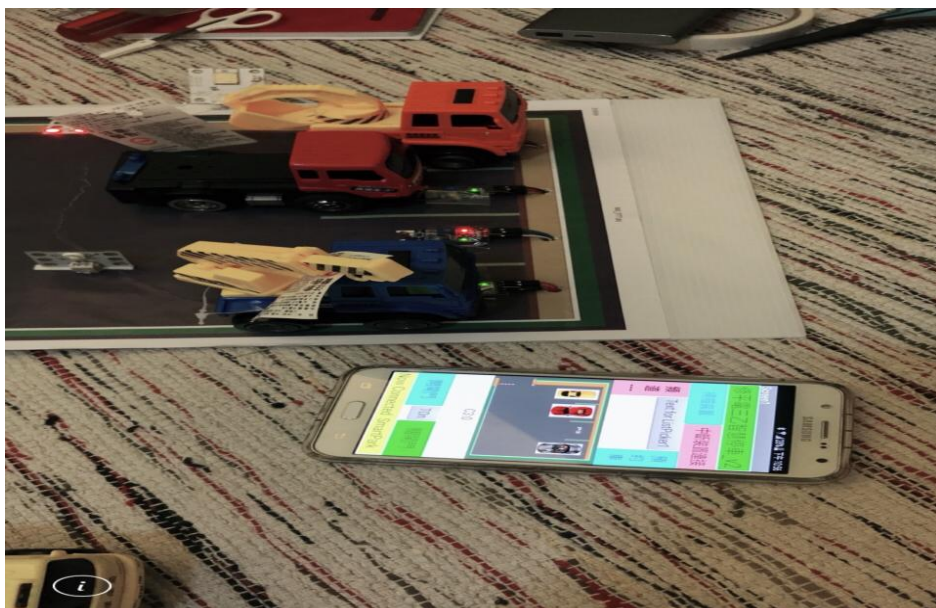


圖 4.4:停車測試



圖 4.5:停車測試

如上圖 4.4 、圖 4.5，將車子蓋過光敏電阻時，代表有車子在停車格內，手機上面就會顯示。

第五章

結論

隨著科技的進步，物聯網會越來越普遍的應用在我們的生活中，此次專題雖然只用 cc2541 藍芽來做短程收發訊號的控制，但模擬智慧停車場跟監控卻是非常成功。Arduino 程式編譯相當方便，運用這套軟體可以創造出很多物聯網相關的運用技術，有了小型停車場的模擬，往後如果要應用在大型停車場，我想一定能為生活帶來更多方便的選擇。

參考文獻

[1] TEXAS INSTRUMENTS

<http://www.ti.com.cn/product/cn/CC2541>

[2] 台灣物聯網科技 TAIWANIOT

<https://www.taiwaniot.com.tw/product>

[3] B217 無線感測實驗室

<http://120.109.165.16/b217/>

[4] 德源科技

http://twarm.com/commerce/product_info.php?products_id=1707

[5] APP Inventor 2

ai2.appinventor.mit.edu/login/