HUB 8735 ultra
災害救援影像傳輸車

國立高雄科技大學

電子工程系

目錄

使用情境

災害救援影像傳輸車，能夠使用在消防救難上面。車子能夠穿梭在人無法靠近的救災區域，提前感應該區域是否有生還者。

當車子辨識到人的時候，會回傳即時資訊給救災人員的移動裝置，方便救災人員可以在即時確認受難者位置的同時確認該位置有無救災阻礙，減少人為直接突圍浪費的時間差，進而增加罹難者的獲救機率。

材料清單

1.HUB 8735 ultra(做yolo)

2.遙控車底盤

3.8051開發板(做遙控)

4.超聲波模組(HC-SR04)

5.L9110馬達驅動模組

6.HT12D/HT12E IC

7.433MHz遙控接發模組

8.電池盒

9.電阻(220/51K/1M)

10.LED燈(紅/綠/黃)

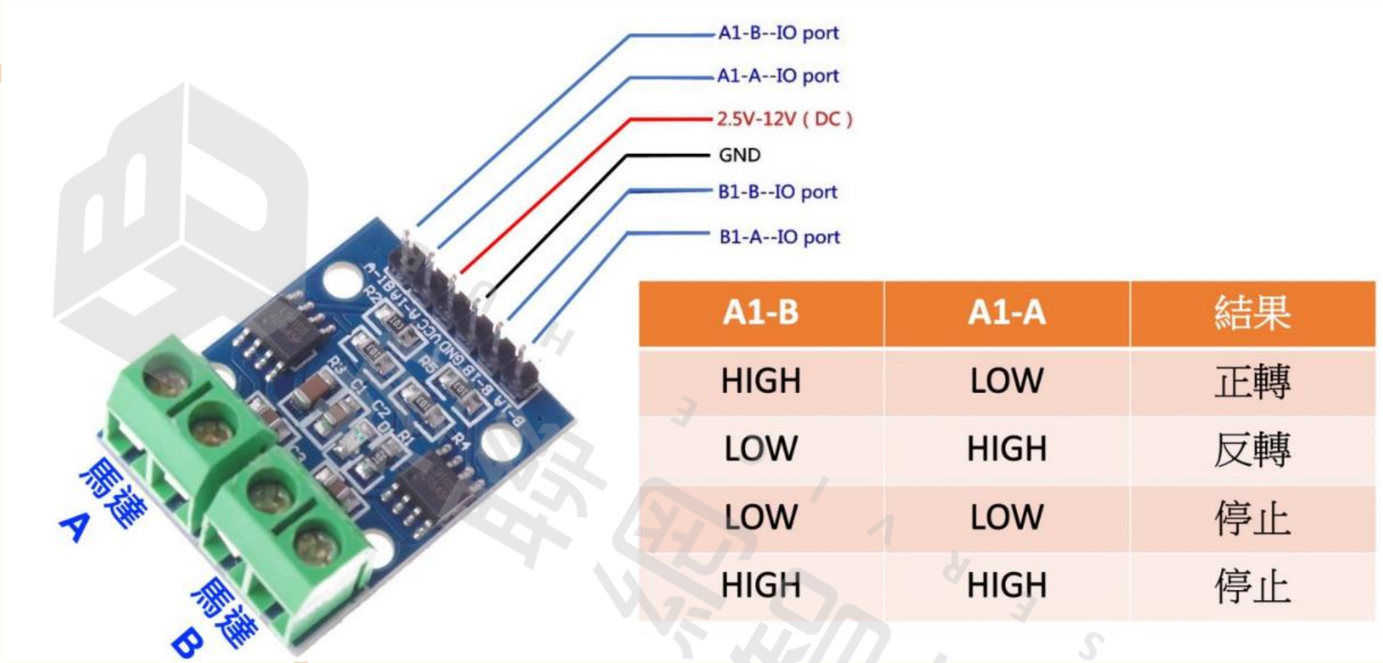
HUB 8735 Ultra

- **HUB 8735 ultra**

主要功能與特性

1. 支持Arduino原生開發環境
2. 內置NPU AI 運算引擎
3. 具備多功能影像處理的高度集成模組

L9110驅動馬達模組



A1-B	A1-A	結果
HIGH	LOW	正轉
LOW	HIGH	反轉
LOW	LOW	停止
HIGH	HIGH	停止

- **L9110驅動馬達**

主要功能與特性

1.雙路馬達驅動

2.高效H橋驅動

3.高電壓和大電流輸出

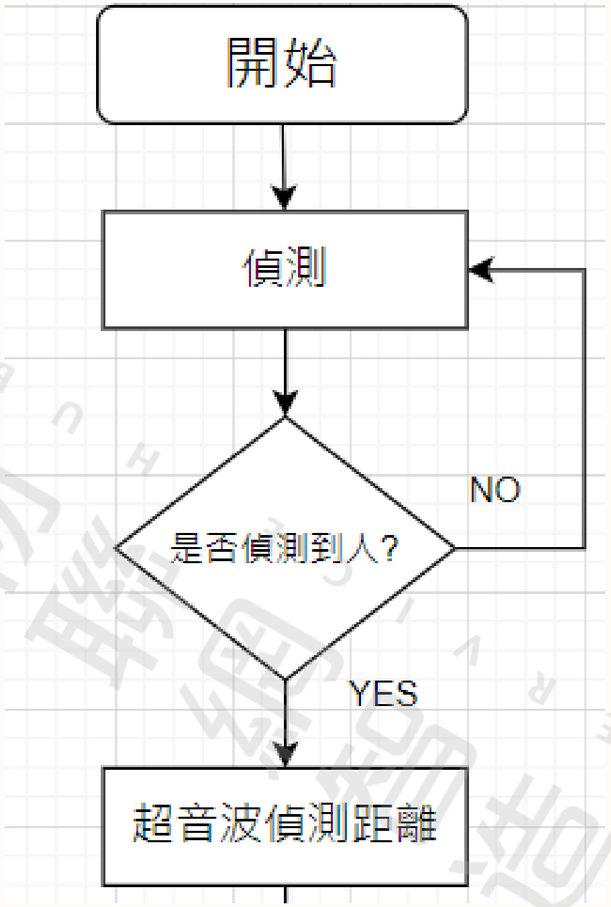
超音波測距模組

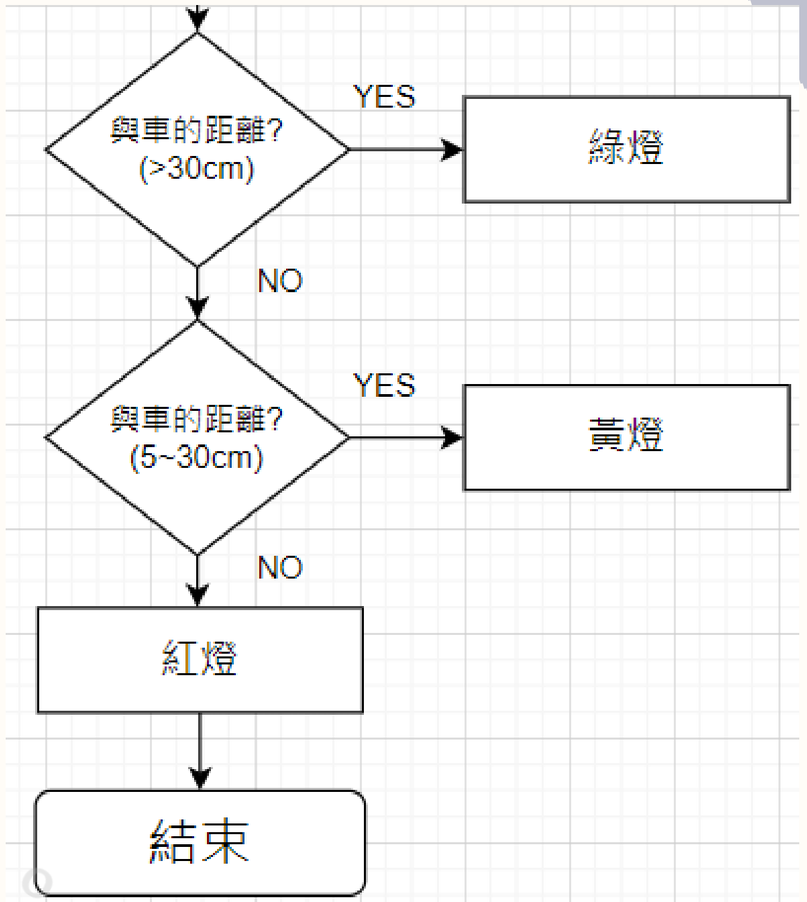
- **HC-SR04**

主要功能與特性

1. Arduino常用模組
2. 監測頻率: 8 kHz ~ 58 KHz
3. 監測距離: 0.3 m 到100 m

流程圖


開始
偵測
是否偵測到人?
NO
YES
超音波偵測距離


與車的距離?
(>30cm)
YES
綠燈
NO
與車的距離?
(5~30cm)
YES
黃燈
NO
紅燈
結束

- 上部分
- 下部分

8051程式圖

```
        ORG    0000H
START:  MOV    P2,    #11111111B
                MOV    P3,    #11111111B
MAIN:   JNB P3.0,    D1     ;
        JNB P3.1,    D2     ;
                JNB P3.2,    D3     ;
        JNB P3.3,    D4     ;
        JMP MAIN      ;

D1:     MOV    P2,    #10101111B
                ACALL DELAY
                MOV    P2,    #11111111B
                MOV    P3,    #11111111B
                SJMP MAIN
D2:     MOV    P2,    #01011111B
                ACALL DELAY
                MOV    P2,    #11111111B
                MOV    P3,    #11111111B
                SJMP MAIN
```

- 上部分

```
D3:     MOV    P2,    #10011111B
                ACALL DELAY
                MOV    P2,    #11111111B
                MOV    P3,    #11111111B
                SJMP MAIN
D4:     MOV    P2,    #01101111B
                ACALL DELAY
                MOV    P2,    #11111111B
                MOV    P3,    #11111111B
                SJMP MAIN

DELAY:  MOV     R3,    #100
NEXT:   MOV     R4,    #250
LOOP:   NOP
        NOP
                DJNZ    R4,    LOOP
                DJNZ    R3,    NEXT
                RET;

                END
```

- 下部分

YOLO程式圖

```
/*

 Example guide:
 https://www.amebaiot.com/en/amebapro2-arduino-neuralnework-object-detection/

 NN Model Selection
 Select Neural Network(NN) task and models using modelSelect(nntask, objdetmodel, facedetmodel, facerecogmodel).
 Replace with NA MODEL if they are not necessary for your selected NN Task.

 NN task
 =======
 OBJECT DETECTION/ FACE DETECTION/ FACE RECOGNITION

 Models
 =======
 YOLOv3 model          DEFAULT YOLOV3TINY    / CUSTOMIZED YOLOV3TINY
 YOLOv4 model          DEFAULT YOLOV4TINY    / CUSTOMIZED YOLOV4TINY
 YOLOv7 model          DEFAULT YOLOV7TINY    / CUSTOMIZED YOLOV7TINY
 SCRFD model           DEFAULT SCRFD         / CUSTOMIZED SCRFD
 MobileFaceNet model   DEFAULT MOBILEFACENET/ CUSTOMIZED MOBILEFACENET
 No model              NA MODEL
 */
```

- 第一部分

YOLO程式圖

```
#include "WiFi.h"
#include "StreamIO.h"
#include "VideoStream.h"
#include "RTSP.h"
#include "NNObjectDetection.h"
#include "VideoStreamOverlay.h"
#include "ObjectClassList.h"

#define CHANNEL   0
#define CHANNELNN 3
// Lower resolution for NN processing
#define NNWIDTH  576
#define NNHEIGHT 320
const int ledPin1 = 7;    // the number of the LED pin
const int ledPin2 = 8;    // the number of the LED pin
const int ledPin3 = 9;    // the number of the LED pin
const byte trigPin=5; //超音波測距的 觸發腳
const byte echoPin=6; //超音波測距的 回應腳
```

- 第二部分

```
VideoSetting config(VIDEO_FHD, 30, VIDEO_H264, 0);
VideoSetting configNN(NNWIDTH, NNHEIGHT, 10, VIDEO_RGB, 0);
NNObjectDetection ObjDet;
RTSP rtsp;
StreamIO videoStreamer(1, 1);
StreamIO videoStreamerNN(1, 1);

char ssid[] = "114514";    // your network SSID (name)
char pass[] = "0956795812";        // your network password
int status = WL_IDLE_STATUS;
float d;
IPAddress ip;
int rtsp_portnum;
```

- 第三部分

YOLO程式圖

```
void setup()
{
    Serial.begin(115200);
      // initialize the LED pin as an output:
    pinMode(ledPin1, OUTPUT);
    pinMode(ledPin2, OUTPUT);
    pinMode(ledPin3, OUTPUT);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);


       // initialize the pushbutton pin as an input:
    while (status != WL_CONNECTED) {
        Serial.print("Attempting to connect to WPA SSID: ");
        Serial.println(ssid);
        status = WiFi.begin(ssid, pass);

        // wait 2 seconds for connection:
        delay(2000);
    }
    ip = WiFi.localIP();

    // Configure camera video channels with video format information
    // Adjust the bitrate based on your WiFi network quality
    config.setBitrate(2 * 1024 * 1024);    // Recommend to use 2Mbps for RTSP streaming to prevent network congestion
    Camera.configVideoChannel(CHANNEL, config);
    Camera.configVideoChannel(CHANNELNN, configNN);
    Camera.videoInit();
```

- 第四部分

YOLO程式圖

```
// Configure RTSP with corresponding video format information
rtsp.configVideo(config);
rtsp.begin();
rtsp_portnum = rtsp.getPort();

// Configure object detection with corresponding video format information
// Select Neural Network(NN) task and models
ObjDet.configVideo(configNN);
ObjDet.modelSelect(OBJECT_DETECTION, DEFAULT_YOLOV4TINY, NA_MODEL, NA_MODEL);
ObjDet.begin();

// Configure StreamIO object to stream data from video channel to RTSP
videoStreamer.registerInput(Camera.getStream(CHANNEL));
videoStreamer.registerOutput(rtsp);
if (videoStreamer.begin() != 0) {
    Serial.println("StreamIO link start failed");
}

// Start data stream from video channel
Camera.channelBegin(CHANNEL);
```

- 第五部分

YOLO程式圖

```
// Start data stream from video channel
Camera.channelBegin(CHANNEL);

// Configure StreamIO object to stream data from RGB video channel to object detection
videoStreamerNN.registerInput(Camera.getStream(CHANNELNN));
videoStreamerNN.setStackSize();
videoStreamerNN.setTaskPriority();
videoStreamerNN.registerOutput(ObjDet);
if (videoStreamerNN.begin() != 0) {
    Serial.println("StreamIO link start failed");
}

// Start video channel for NN
Camera.channelBegin(CHANNELNN);

// Start OSD drawing on RTSP video channel
OSD.configVideo(CHANNEL, config);
OSD.begin();
```

- 第六部分

YOLO程式圖

```
void loop()
{
    std::vector<ObjectDetectionResult> results = ObjDet.getResult();

    uint16_t im_h = config.height();
    uint16_t im_w = config.width();
/*
    Serial.print("Network URL for RTSP Streaming: ");
    Serial.print("rtsp://");
    Serial.print(ip);
    Serial.print(":");
    Serial.println(rtsp_portnum);
    Serial.println(" ");
*/
    //printf("Total number of objects detected = %d\r\n", ObjDet.getResultCount());//object=ObjDet.getResultCount()
    OSD.createBitmap(CHANNEL);
```

- 第七部分

YOLO程式圖

```
if (ObjDet.getResultCount() > 0) {
    for (int i = 0; i < ObjDet.getResultCount(); i++) {
        int obj_type = results[i].type();
        if (itemList[obj_type].filter) {    // check if item should be ignored

            ObjectDetectionResult item = results[i];
            // Result coordinates are floats ranging from 0.00 to 1.00
            // Multiply with RTSP resolution to get coordinates in pixels
            int xmin = (int)(item.xMin() * im_w);
            int xmax = (int)(item.xMax() * im_w);
            int ymin = (int)(item.yMin() * im_h);
            int ymax = (int)(item.yMax() * im_h);

            // Draw boundary box
           // printf("Item %d %s:\t%d %d %d %d\n\r", i, itemList[obj_type].objectName, xmin, xmax, ymin, ymax);
            printf("Item %d %s:\t\n\r", i, itemList[obj_type].objectName);
            OSD.drawRect(CHANNEL, xmin, ymin, xmax, ymax, 3, OSD_COLOR_WHITE);

            // Print identification text
            char text_str[20];
            snprintf(text_str, sizeof(text_str), "%s %d", itemList[obj_type].objectName, item.score());
            OSD.drawText(CHANNEL, xmin, ymin - OSD.getTextHeight(CHANNEL), text_str, OSD_COLOR_CYAN);
            // turn LED off:
```

- 第八部分

YOLO程式圖

```
if  (itemList[obj_type].objectName == "person")
   //超聲波模組測距離
    d = sr04() * 0.017;
    Serial.print(d,1);
    Serial.println("cm");
    delay(1000);
    if (d > 30) {//綠
      digitalWrite(ledPin1, HIGH);
      digitalWrite(ledPin2, LOW);
      digitalWrite(ledPin3, LOW);
      delay(100);
      digitalWrite(ledPin1, LOW);
    }

    else if (d<5){//紅
      digitalWrite(ledPin3, HIGH);
      digitalWrite(ledPin1, LOW);
      digitalWrite(ledPin2, LOW);
      delay(100);
      digitalWrite(ledPin3, LOW);
    }
    else {//黃
      digitalWrite(ledPin2, HIGH);
      digitalWrite(ledPin1, LOW);
      digitalWrite(ledPin3, LOW);
      delay(100);
      digitalWrite(ledPin2, LOW);
    }
}
```

- 第九部分

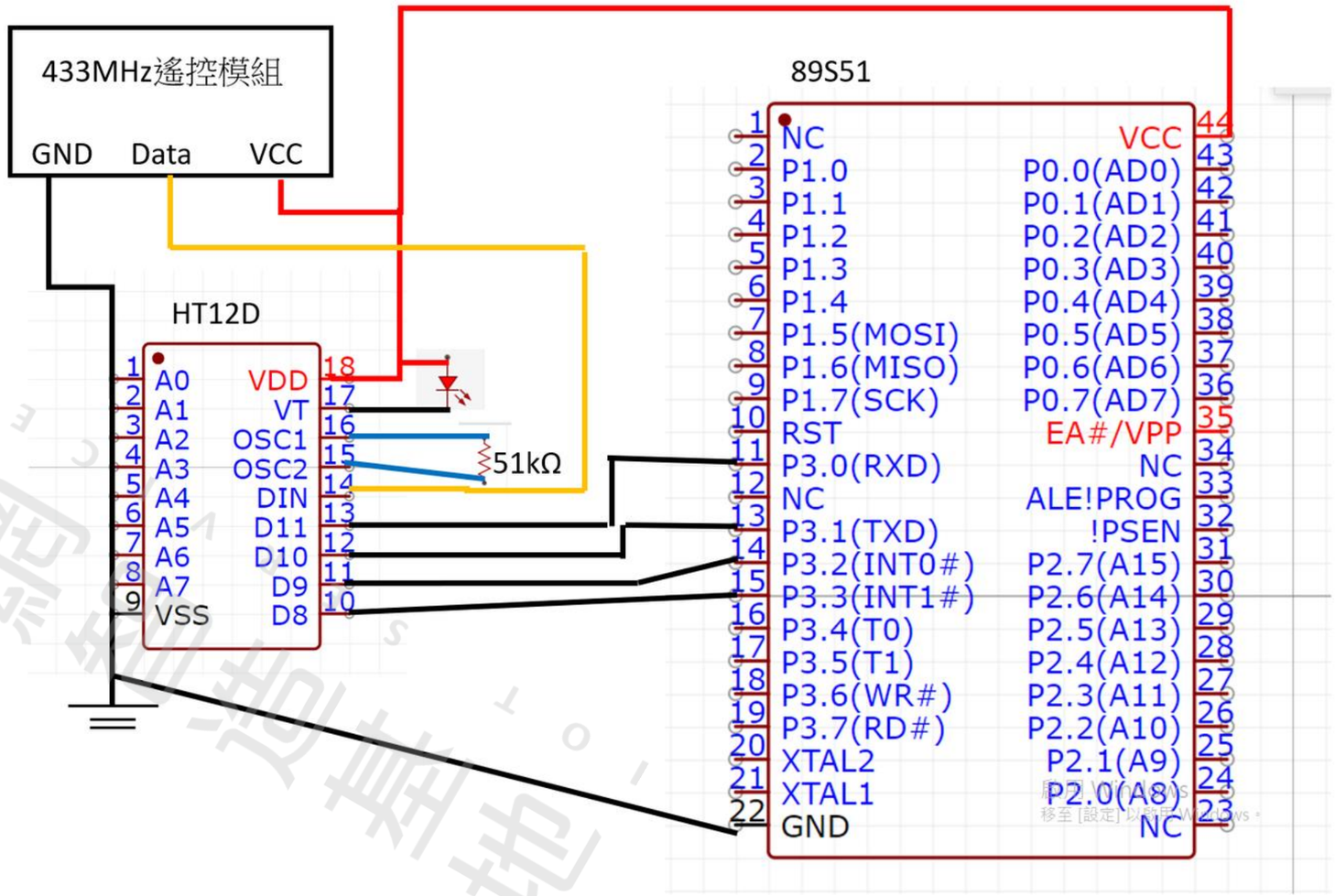
```
    OSD.update(CHANNEL);

    // delay to wait for new results
    delay(100);
}
unsigned long sr04() {
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  return pulseIn(echoPin, HIGH);
}
```

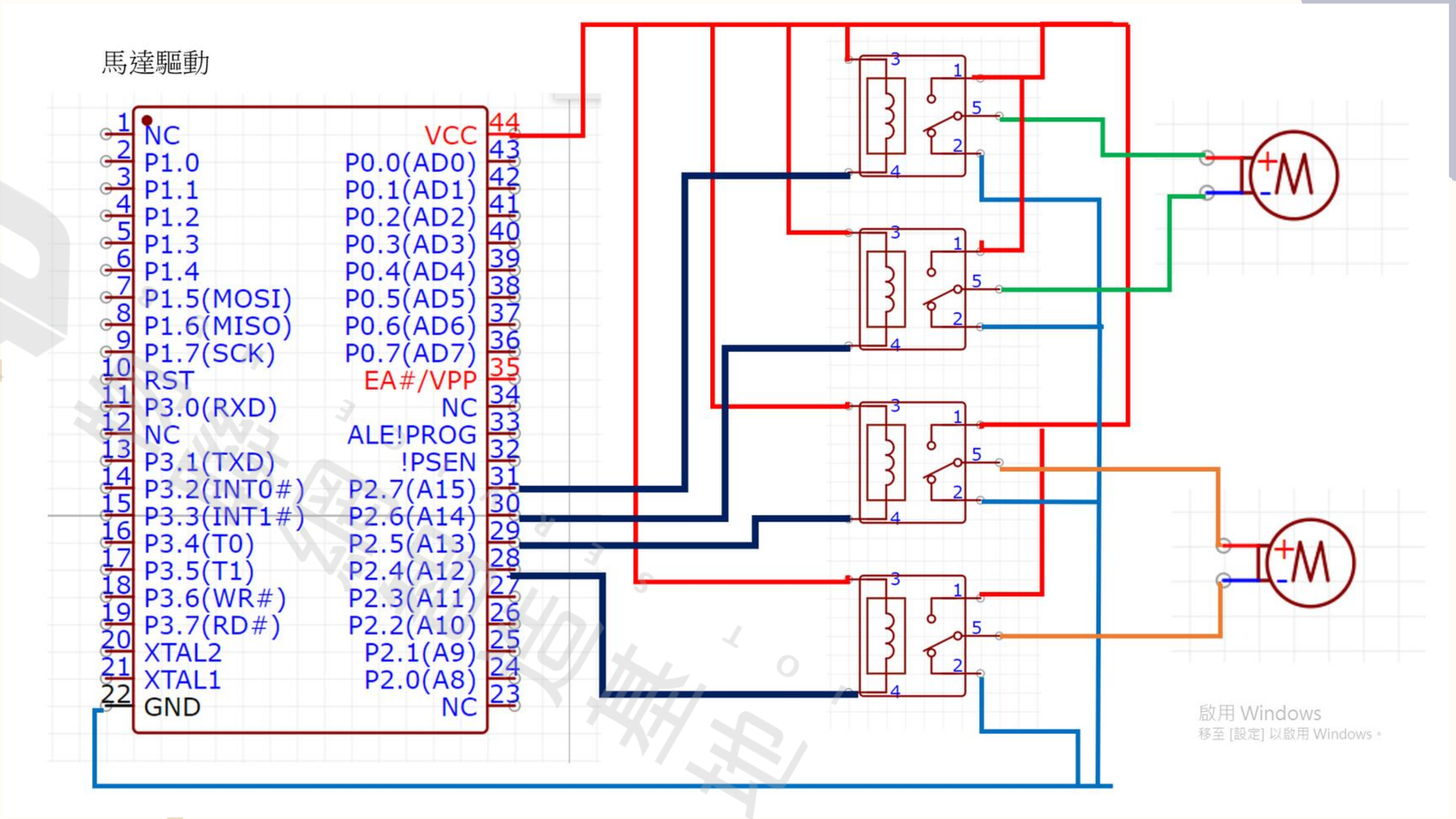
- 第十部分

接線圖

接收電路

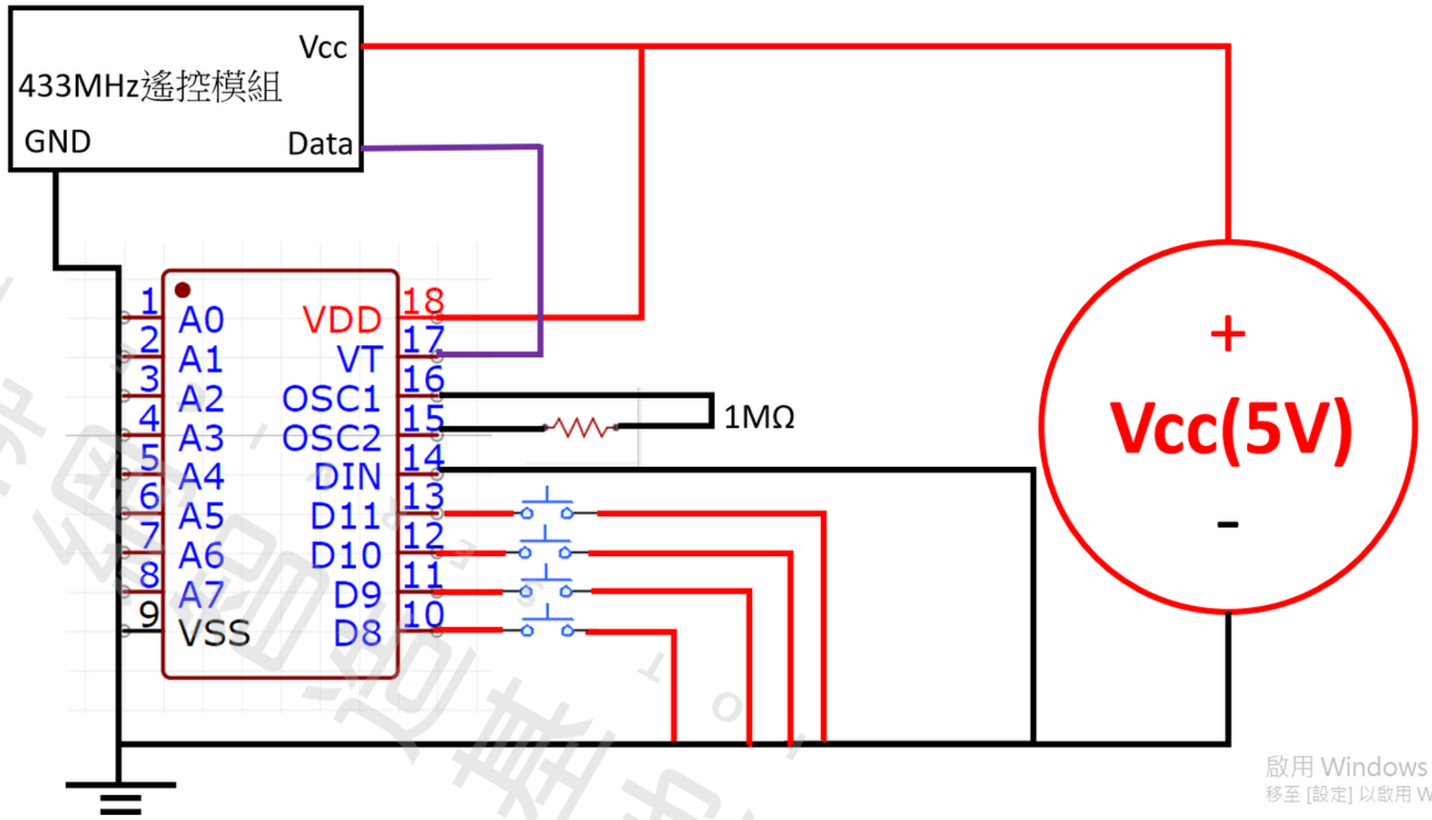

433MHz遙控模組
GND
Data
VCC
HT12D
A0
A1
A2
A3
A4
A5
A6
A7
VSS
VDD
VT
OSC1
OSC2
DIN
D11
D10
D9
D8
51kΩ
89S51
NC
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5(MOSI)
P1.6(MISO)
P1.7(SCK)
RST
P3.0(RXD)
NC
P3.1(TXD)
P3.2(INT0#)
P3.3(INT1#)
P3.4(T0)
P3.5(T1)
P3.6(WR#)
P3.7(RD#)
XTAL2
XTAL1
GND
VCC
P0.0(AD0)
P0.1(AD1)
P0.2(AD2)
P0.3(AD3)
P0.4(AD4)
P0.5(AD5)
P0.6(AD6)
P0.7(AD7)
EA#/VPP
NC
ALE!PROG
!PSEN
P2.7(A15)
P2.6(A14)
P2.5(A13)
P2.4(A12)
P2.3(A11)
P2.2(A10)
P2.1(A9)
P2.0(A8)
NC

接線圖


馬達驅動
1 NC
2 P1.0
3 P1.1
4 P1.2
5 P1.3
6 P1.4
7 P1.5(MOSI)
8 P1.6(MISO)
9 P1.7(SCK)
10 RST
11 P3.0(RXD)
12 NC
13 P3.1(TXD)
14 P3.2(INT0#)
15 P3.3(INT1#)
16 P3.4(T0)
17 P3.5(T1)
18 P3.6(WR#)
19 P3.7(RD#)
20 XTAL2
21 XTAL1
22 GND
44 VCC
43 P0.0(AD0)
42 P0.1(AD1)
41 P0.2(AD2)
40 P0.3(AD3)
39 P0.4(AD4)
38 P0.5(AD5)
37 P0.6(AD6)
36 P0.7(AD7)
35 EA#/VPP
34 NC
33 ALE!PROG
32 !PSEN
31 P2.7(A15)
30 P2.6(A14)
29 P2.5(A13)
28 P2.4(A12)
27 P2.3(A11)
26 P2.2(A10)
25 P2.1(A9)
24 P2.0(A8)
23 NC
+M
-M
+M
-M

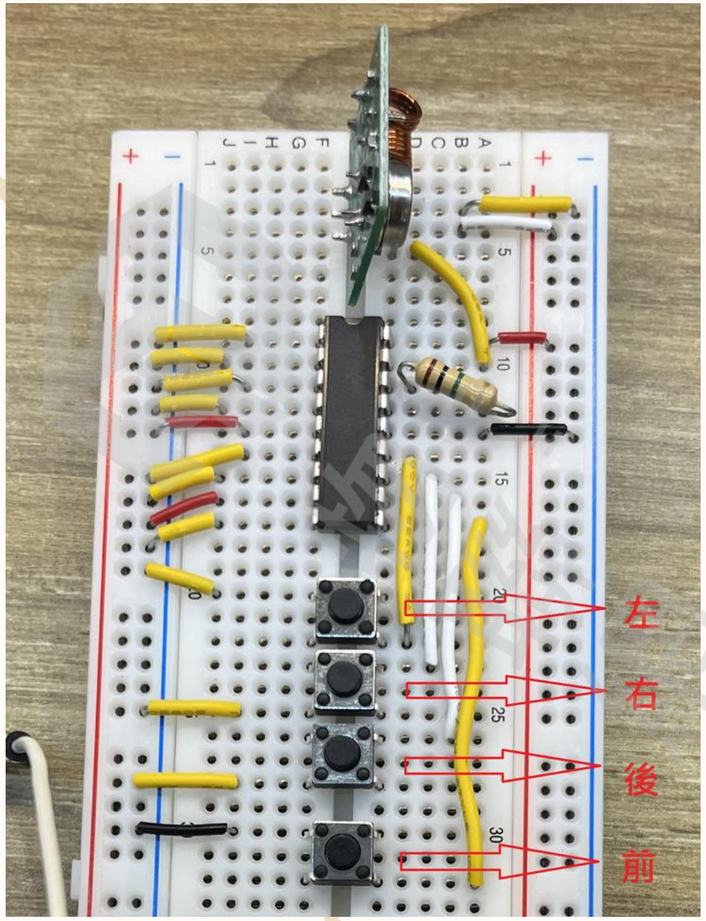
遙控器接線圖

發射電路

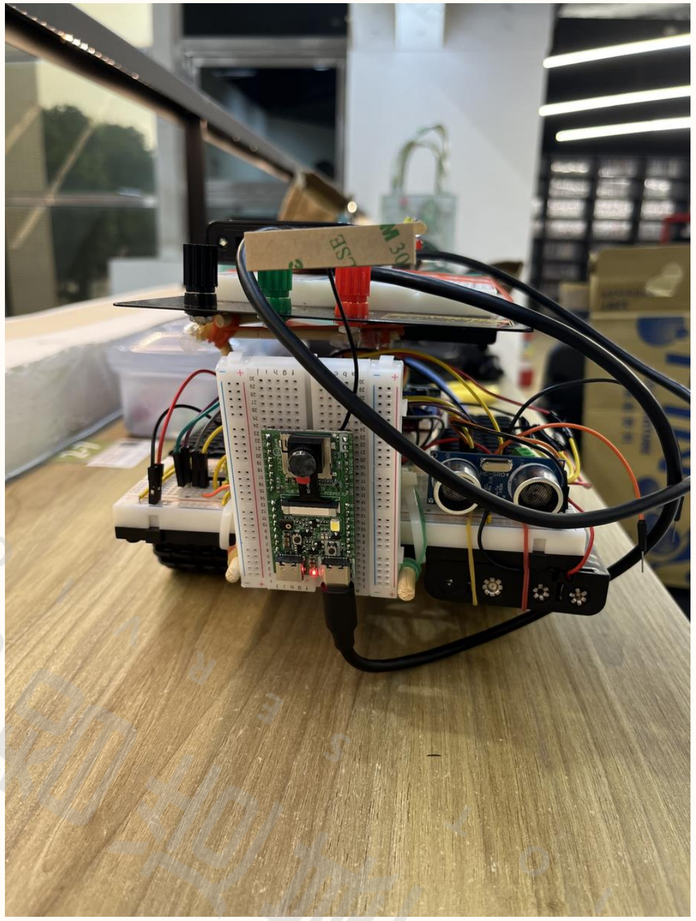


啟用 Windows
移至 [設定] 以啟用 Windows。

功能介紹

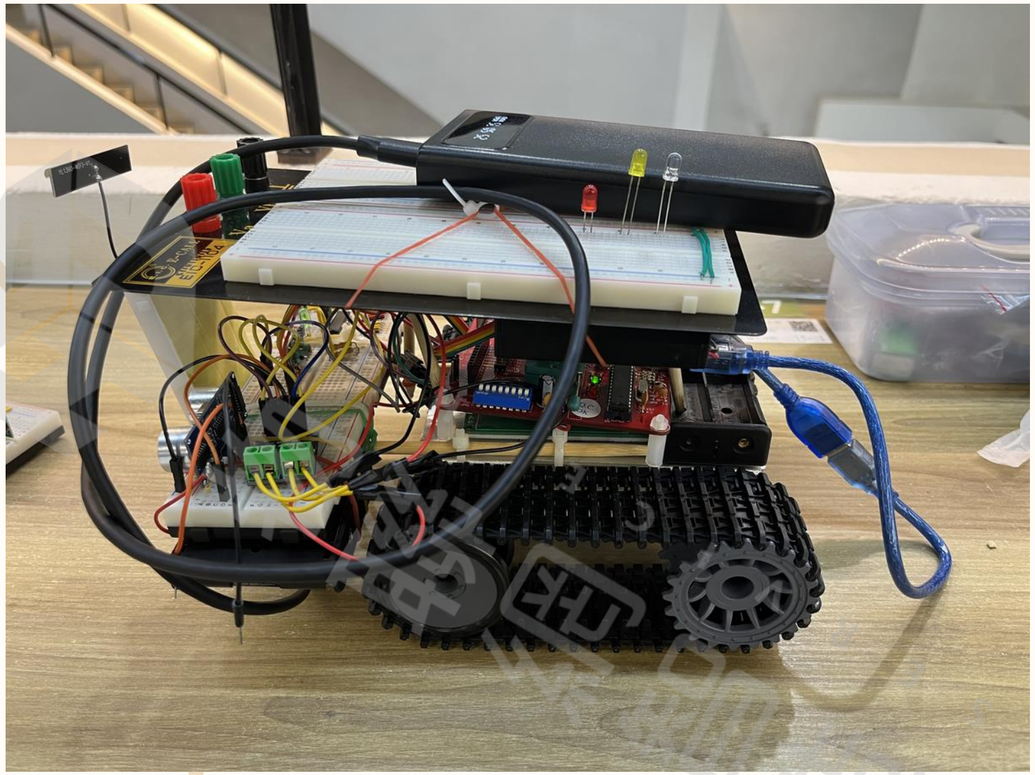


- 自走車遙控器



- 自走車(前方)

- 在自走車前方加裝HUB 8735 Ultra和超音波感測器。
- 使用YOLO人臉辨識可透過手機螢幕即時觀測。
- 自製遙控器可輕鬆操控自走車前行。



- 自走車(側面)

與物體的距離

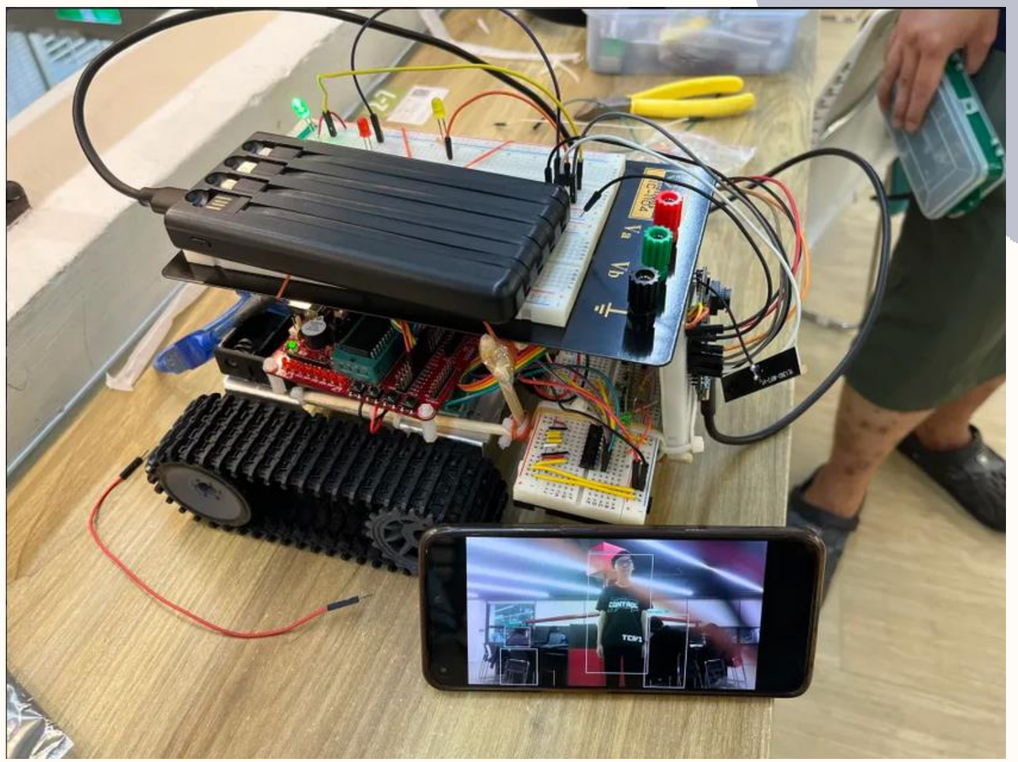
- LED顯示：
- 正常距離(30cm)：綠燈
- 稍微過近(5 ~ 30cm)：黃燈
- 過近(< 5cm)：紅燈

YOLO辨識

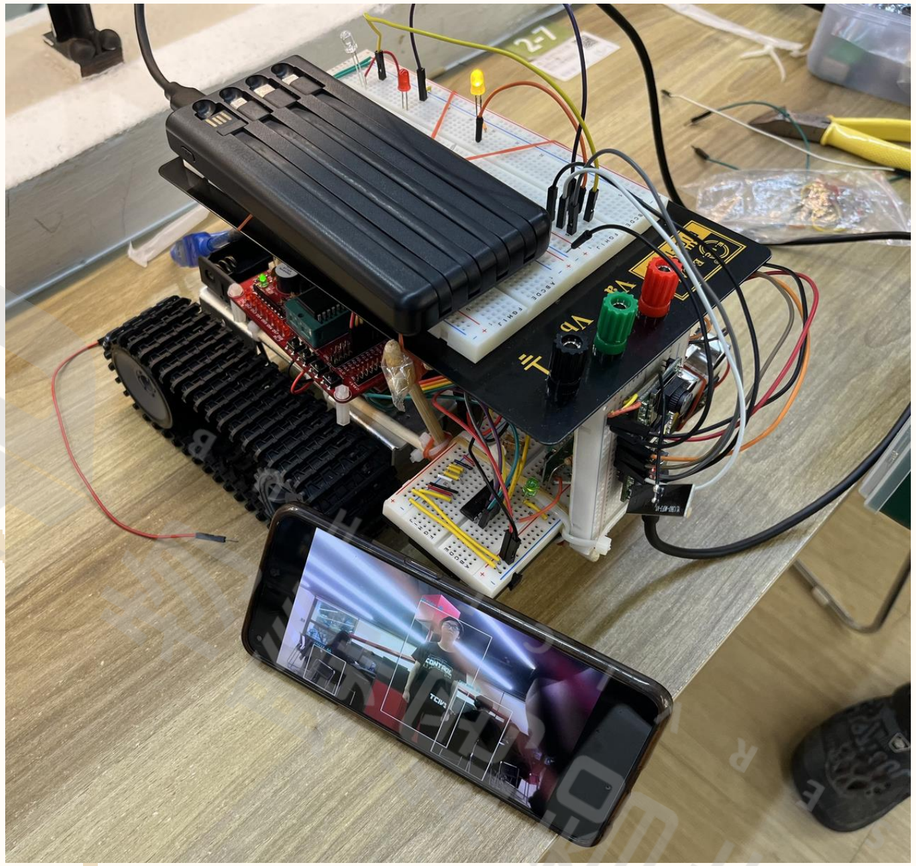
- YOLO辨識(人)

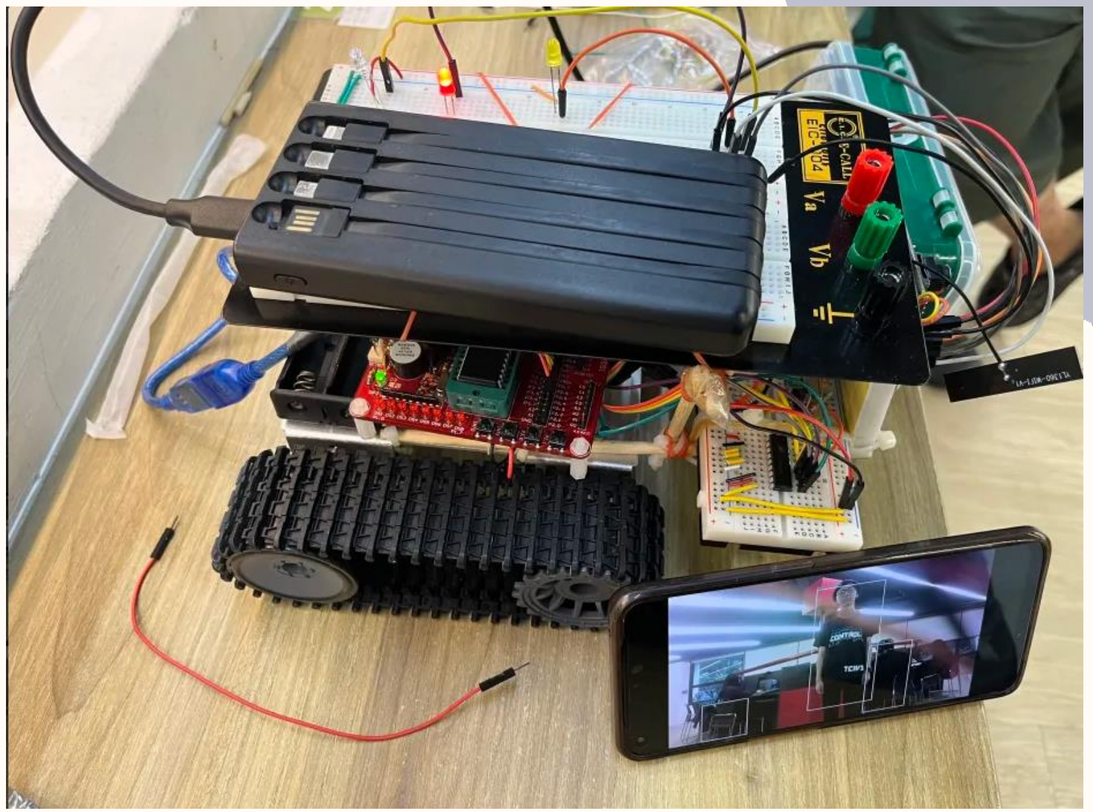
- 手機螢幕畫面

實作照片



- 距離自走車 > 30cm
- LED顯示綠燈





- 距離自走車 5 ~ 30cm
- LED顯示黃燈

- 距離自走車 < 5cm
- LED顯示綠燈

THE END

物联网知道基地 IOT SERVICE HUB