# 雷達教案前篇:BGT60LTR11AIP 雷達介紹

一、 實驗目的

1.1 使用 radar GUI(或 radar Fusion GUI)了解雷達感測狀態。

1.2 解焊 R3 電阻進入自主模式

改變 R10 電阻值以控制感測距離

改變 R20 電阻值以控制感測持續時間

1.3 使用雷達控制 Arduino MKR WIFI 1010

二、 材料

bgt60ltr11aip shield board、MCU7 底板、傳輸線、顯微鏡、焊接工具(焊槍、0.2mm 焊錫、松香、濕海綿)、針腳、貼片電阻、Arduino MKR WIFI 1010

三、 預先工作

1. 安裝 infineon radar GUI 及 Arduino IDE
2. Arduino IDE :工具/開發版/開發版管理員/Arduino MKR 安裝
3. Arduino IDE :草稿碼/匯入程式庫/管理程式庫/bgt60 安裝

四、 流程

## 4.1 radar GUI

bgt60ltr11aip radar 是一顆都普勒雷達，其中 BGT60LTR11AIP MMIC 是一款完全集成的微波運動傳感器，運行於 60GHz ISM 頻段，並具有 4 個四態(QS1-4) 輸入引腳，可工作於 SPI 模式和自主模式。shield board 在 SPI 模式下可接上 MCU7 底板且可由 radar GUI 觀測運動狀態。

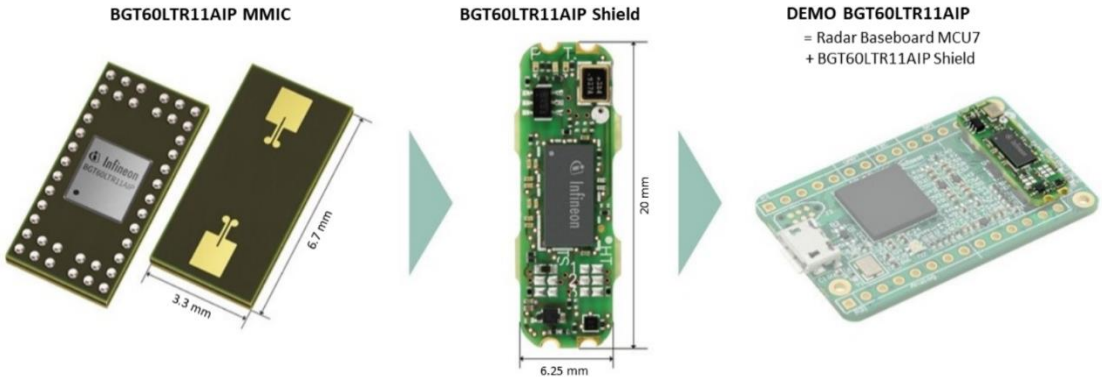



圖 1 MMIC、shield board、shield board 組合 MCU7 底板

1. 從 infineon 官網下載安裝 radar GUI (https://softwaretools.infineon.com/tools/)

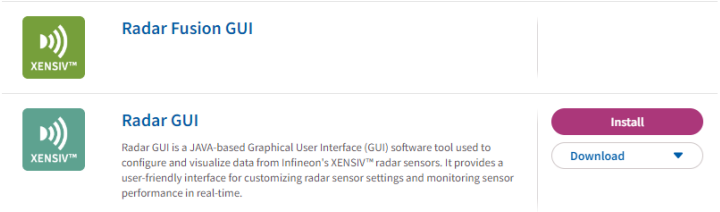



圖 2　radar GUI 下載及安裝

2. 將 bgt60ltr11aip shield board 組合 MCU7 底板並以傳輸線接上電腦後開啟 radar GUI(或 radar Fusion GUI)。

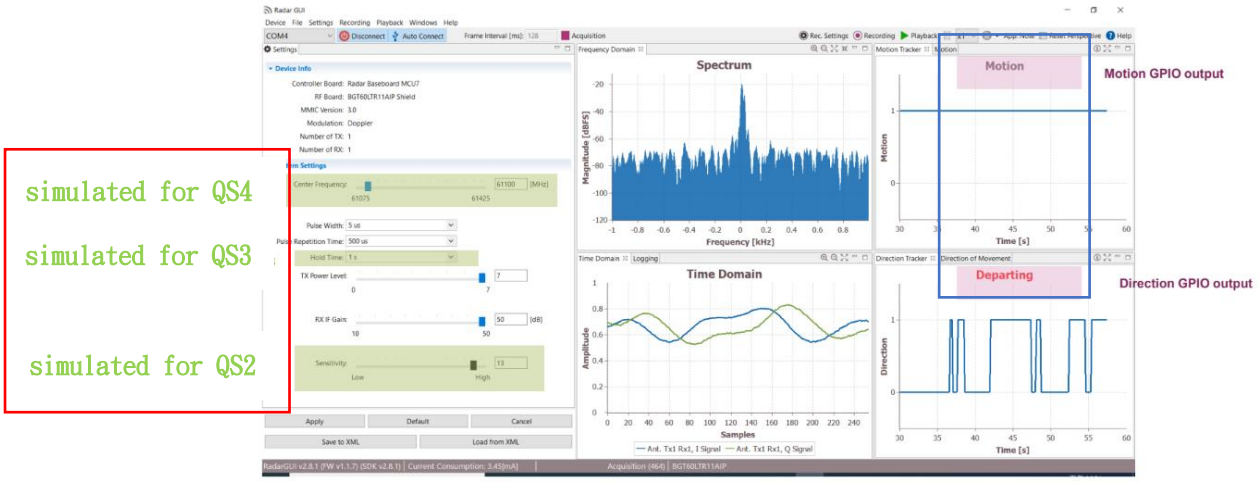



圖 3　radar GUI 開啟視窗

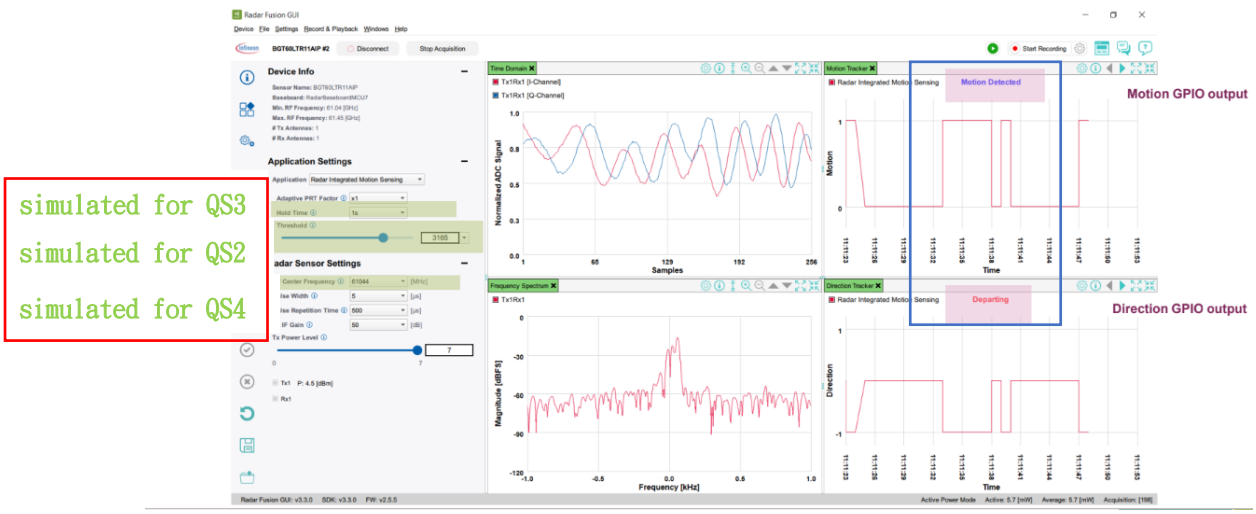



圖 4　radar Fusion GUI 開啟視窗

左側可控制雷達感測程度 ：

simulated for QS2 控制感測範圍

simulated for QS3 控制感測時間

simulated for QS4 控制感測頻率(約 60GHz)

可自由調整後點擊 Apply 保存格式，請注意在關閉 GUI 後雷達的所有設定將被還原。

右側可觀察到雷達感測情況:Motion 表示有無偵測物體、Departing 或 Approaching 表示偵測到遠離或靠近。

### 4.2 QS1 ： 解焊 R3 電阻進入自主模式

BGT60LTR11AIP 配置為在 SPI 模式下，可透過移除 R3 電阻進入自主模式。Shield board 具有 4 個(PD、TD、GND、VIN)腳位可對外連接。

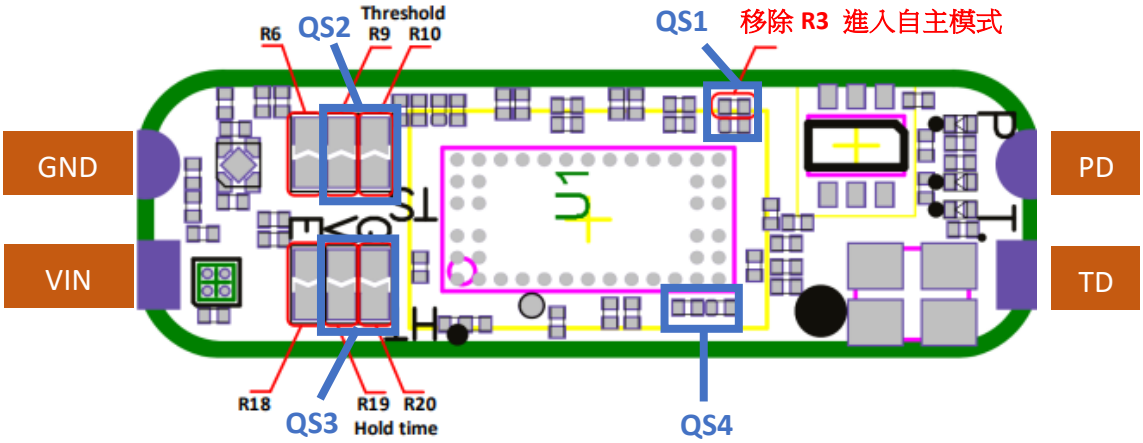



圖 5 radar shield board QS1 to QS4 schematic

將 shield board 置於顯微鏡下並以膠帶固定，調整目鏡至適當距離後移除 R3 電阻。

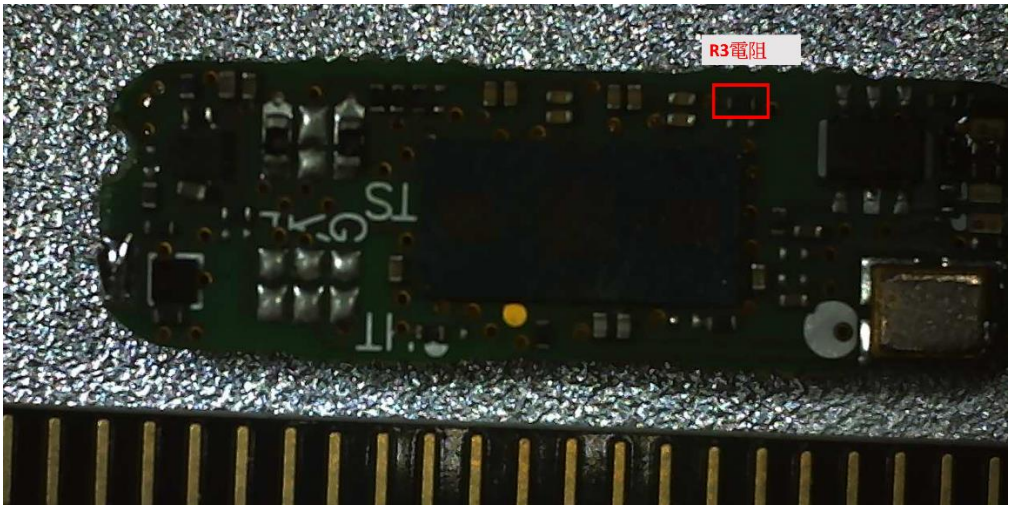



圖 6 解焊前 R3 電阻(1mm/刻度)

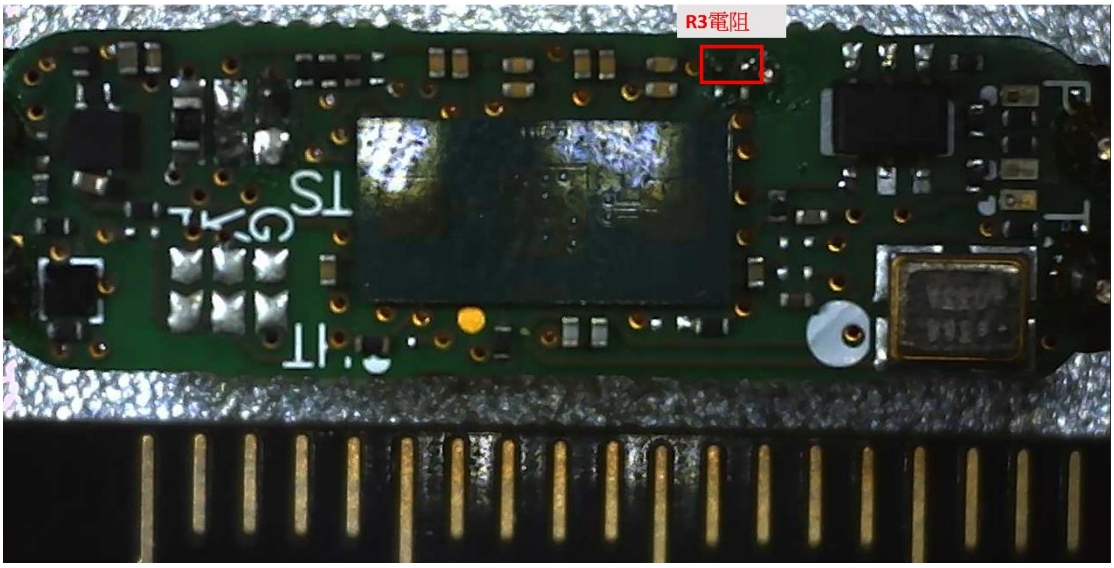



圖 7 解焊後 R3 電阻(1mm/刻度)

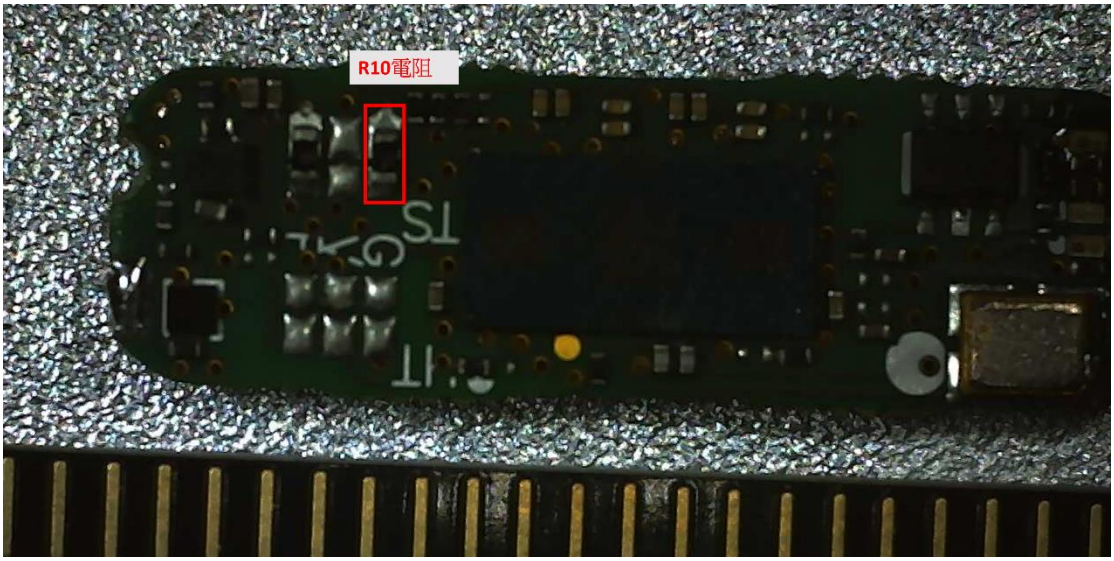
QS2 ： 改變 R9、R10 電阻值調整感測範圍(Sensitivity)

| Resistors settings | | Detector Threshold | Sensitivity |
|---|---|---|---|
| R9 | R10 | (Radar Fusion GUI setting) | (Radar GUI setting) |
| 10 kΩ | 330 Ω | 61 | 15* |
| 10 kΩ | 1 kΩ | 66 | 14* |
| 10 kΩ | 1.8 kΩ | 80 | 13 |
| 10 kΩ | 2.7 kΩ | 90 | 12 |
| 10 kΩ | 3.9 kΩ | 112 | 11 |
| 10 kΩ | 5.6 kΩ | 136 | 10 |
| 10 kΩ | 6.8 kΩ | 192 | 9 |
| 10 kΩ | 8.2 kΩ | 248 | 8 |
| 10 kΩ | 12 kΩ | 320 | 7 |
| 10 kΩ | 15 kΩ | 284 | 6 |
| 10 kΩ | 18 kΩ | 480 | 5 |
| 10 kΩ | 27 kΩ | 640 | 4 |
| 10 kΩ | 39 kΩ | 896 | 3 |
| 10 kΩ | 56 kΩ | 1344 | 2 |
| 10 kΩ | 100 kΩ | 1920 | 1 |
| 10 kΩ | 220 kΩ | 2560 | 0 |

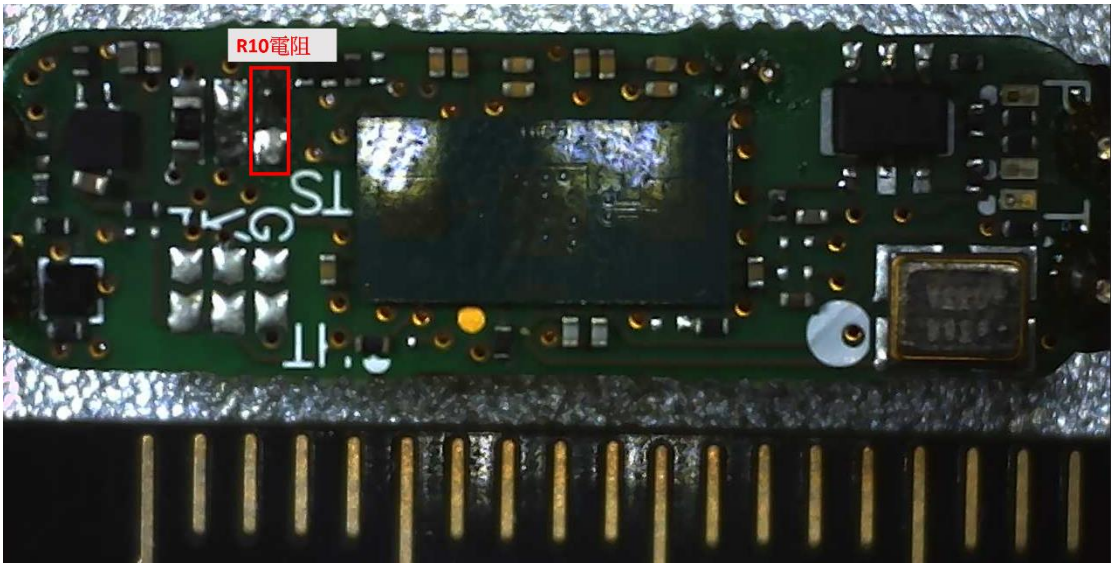
*High sensitivity levels could lead to false detections, are not shown on the Radar GUI settings.

圖 8　R9、R10 電阻值對應雷達感測範圍

將 shield board 置於顯微鏡下並以膠帶固定，調整目鏡至適當距離後移除 R10 電阻，提高感測精度。




解焊前 R10 電阻(1mm/刻度)




解焊後 R10 電阻(1mm/刻度)

**QS3 ： 改變 R19、R20 電阻值調整感測持續時間(Hold time)**

Hold time 是指感測後保持訊號的時間，在所偵測目標快速移動相對需要即時回復的設計需降低 Hold time，反之若是作為感測物體的有無，如陌生人靠近發出警報聲音的時間也由 Hold time 控制。

**Table 6 QS3 settings**

| Resistors setting | | Detector Hold time |
|---|---|---|
| R19 | R20 | |
| 10 kΩ | 330 Ω | Minimum (16 ms, 32 ms, 64 ms or 128 ms) dep. on PRT |
| 10 kΩ | 1 kΩ | 500 ms |
| 10 kΩ | 1.8 kΩ | 1 s |
| 10 kΩ | 2.7 kΩ | 2 s |
| 10 kΩ | 3.9 kΩ | 3 s |
| 10 kΩ | 5.6 kΩ | 5 s |
| 10 kΩ | 6.8 kΩ | 10 s |
| 10 kΩ | 8.2 kΩ | 30 s |
| 10 kΩ | 12 kΩ | 45 s |
| 10 kΩ | 15 kΩ | 1 min |
| 10 kΩ | 18 kΩ | 90 s |
| 10 kΩ | 27 kΩ | 2 min |
| 10 kΩ | 39 kΩ | 5 min |
| 10 kΩ | 56 kΩ | 10 min |
| 10 kΩ | 100 kΩ | 15 min |
| 10 kΩ | 220 kΩ | 30 min |

**圖 9 R19、R20 電阻值對應雷達感測時間**

1. 將 shield board 置於顯微鏡下並以膠帶固定，調整目鏡至適當距離後移除 R20 電阻。
2. 將四個針腳分別焊上 shield board 的四個腳位。
3. 對於特殊需求，雷達偵測時間可以由外部控制，預設 QS3 設定 detector hold time 較低，作為一般雷達應用，偵測已足夠靈敏，因此跳過這部分的焊接。

## 4.3 使用 shield board 控制 Arduino MKR WIFI 1010

在移除 R3 電阻後進入自主模式可和其他元件搭配使用，先前設定完感測程度後將 shield board 組合上 Arduino MKR WIFI 1010 可於 Arduino IDE 上觀察雷達的偵測情況。Arduino MKR WIFI 1010 可從 radar shield board 獲得是否感測到物體(motion、no motion)及物體運動情況(approaching、departing、no direction)共 5 個狀態。

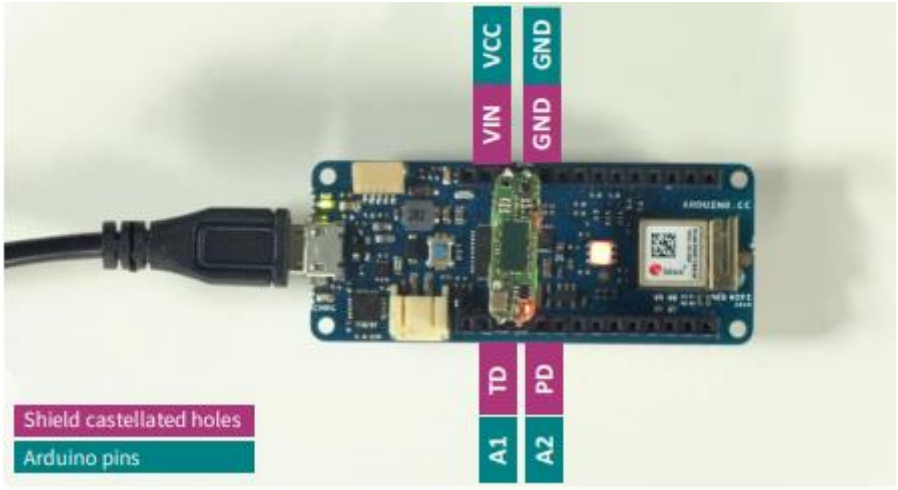



圖 10　shield board 連接 Arduino MKR WIFI 1010

1. 將 shield board 接上 Arduino MKR WIFI 1010，如圖 8，使用傳輸線連接電腦。
2. 開啟 Arduino IDE，於工具設定開發版”Arduino MKR WIFI 1010”及序列埠 COM，寫入 code 並上傳至 MKR 以取得 shield board 偵測靠近或遠離。你可以透過開啟序列埠監控視窗得知雷達的偵測情況。

code ：(https://github.com/Infineon/arduino-radar-bgt60/blob/master/examples/directionDetection/directionDetection.ino)

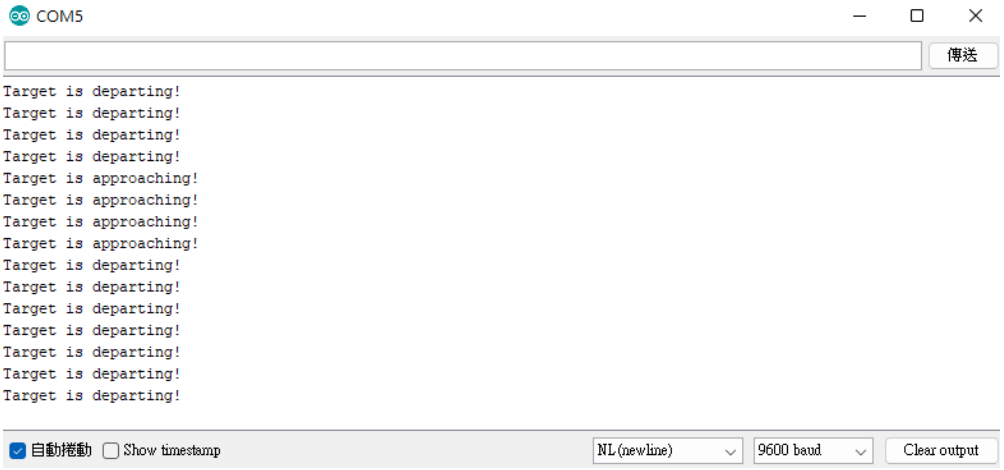



圖 11　Arduino IDE 開啟序列埠監控視窗

5.BGT60LTR11AIP 指導文件

https://www.infineon.com/dgdl/Infineon-AN608_BGT60LTR11AIP_Shield-ApplicationNotes-v01_80-EN.pdf?fileId=5546d4627550f4540175558b817a4d22

# 雷達教案後篇:結合 Arduino 應用

一、 實驗目的

1.1 使用雷達控制 LED

1.2 使用雷達控制伺服馬達

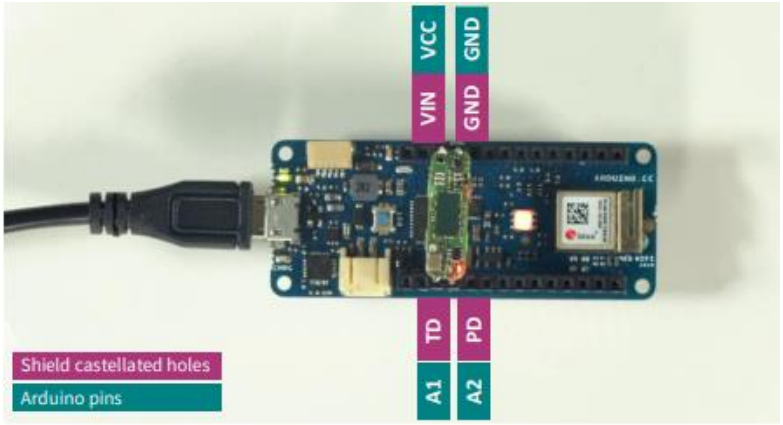
二、 材料

麵包版 1 個、radar shield broad(焊接後)1 個、公對母杜邦線 3 條、LED 燈 1 顆、伺服馬達 1 個、伺服馬達小扇葉 1 片、傳輸線、電阻 1 顆、Arduino MKR WIFI 1010 一個、小型紙盒 1 個、手電筒 2 個、紙板碎片 1 個、泡棉膠 1 捲

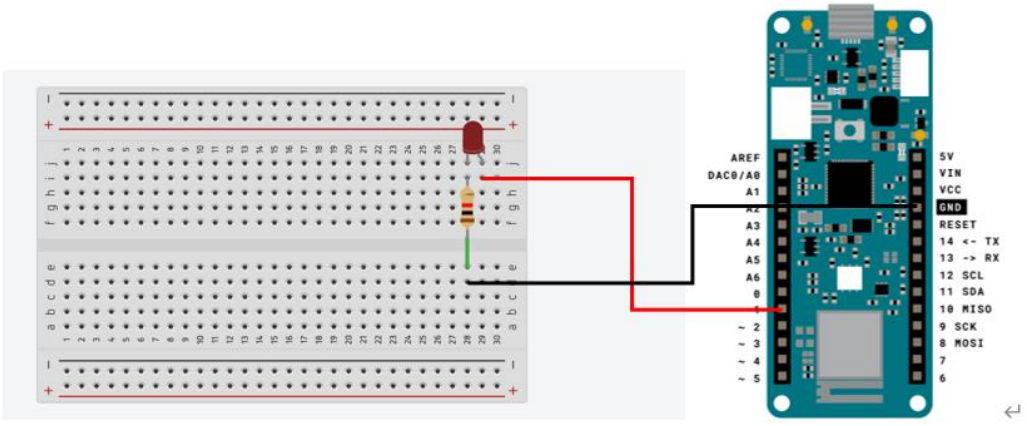
三、 流程

**3.1 使用 radar shield broad 控制 LED**

編寫 Arduino IDE 以讀取 radar shield board 輸出的資料並由 Arduino MKR 控制 LED 燈在靠近時發光，離開時關閉，模擬日常生活的人體偵測省電燈。




**圖一 radar shield broad 組合 Arduino MKR**




**圖二 Arduino MKR 控制 LED 線路圖**

1. 將 radar shield broad 組合上 Arduino MKR，雷達的四個腳位(TD/PD/VIN/GND)分別對應 Arduino MKR 四個腳位(A1/A2/VCC/GND)，如圖一。
2. LED 燈長接腳為正極，短接腳為負極。取一公對母杜邦線，公端接上麵包版，母端接上 Arduino MKR 底部接腳，連接線路如圖二。
3. 使用傳輸線連接 Arduino MKR 至電腦，編寫 Arduino IDE 並上傳至 Arduino MKR 使偵測到靠近時輸出 5V 電壓給 LED 燈使發亮，離開時關閉。

Code ：

```
#include <Arduino.h>

#include <bgt60-ino.hpp>

#include <bgt60-platf-ino.hpp>


#ifndef TD
#define TD  15
#endif

#ifndef PD
#define PD  16
#endif

Bgt60Ino radarShield(TD, PD);

void setup()
{

    Serial.begin(9600);

    Error_t init_status = radarShield.init();

    if (OK != init_status) {
        Serial.println("Init failed.");
    }
    else {
        Serial.println("Init successful.");
    }

    pinMode(1, OUTPUT);  // 腳位 1 設定為輸出
}


void loop()
{

    Bgt60::Direction_t direction = Bgt60::NO_DIR;

    Error_t err = radarShield.getDirection(direction);
```

```
    if (err == OK)
    {
        switch (direction)
        {
            case Bgt60::APPROACHING:
                Serial.println("Target is approaching!");
                digitalWrite(1, true); // 靠近時腳位 1 輸出高電位(5V)
                break;

            case Bgt60::DEPARTING:
                Serial.println("Target is departing!");
                digitalWrite(1, false); // 離開時腳位 1 輸出低電位(0V)
                break;

            case Bgt60::NO_DIR:
                Serial.println("Direction cannot be determined since no
motion was detected!");
                break;
        }
    }
    /* API execution returned error */
    else{
        Serial.println("Error occurred!");
    }

    delay(500);
}
```

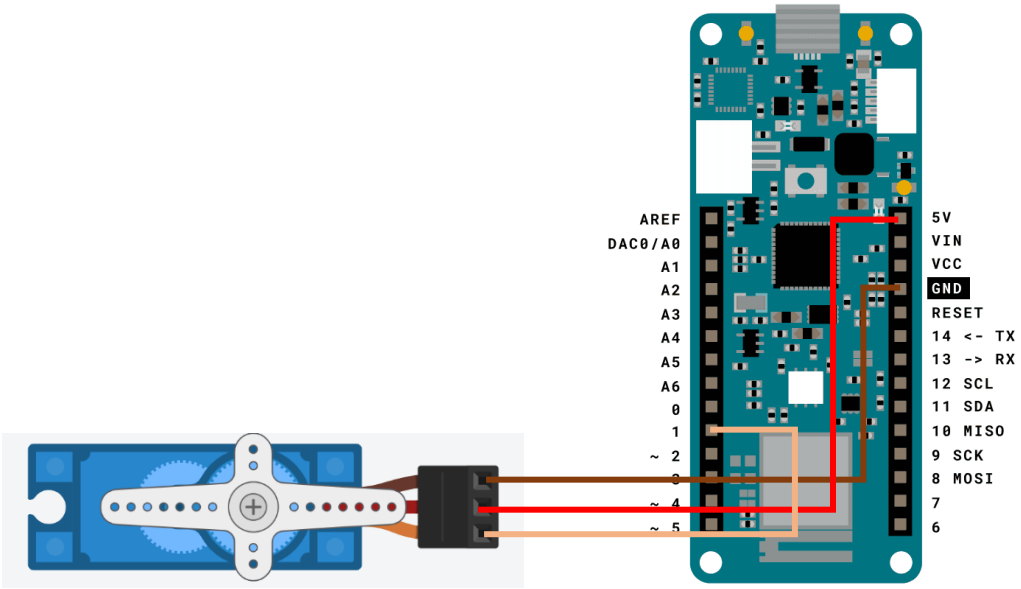
Code 1　雷達控制 LED 開關

## 3.2 使用 radar shield broad 控制伺服馬達

由雷達控制伺服馬達是基於助行器自動開關腳步指示燈所使用的另類方法。考慮到一般在醫院工作的復健師提供病人使用的助行器無腳步指示，容易造成使用者復健時步態不標準導致復健成效有限，因此提供腳步指示燈讓使用者可多加留意自身的步態準確度。

**圖三　開啟腳步指示燈的助行器**

伺服馬達是 Arduino 可用來控制轉動的一個元件，一般形式的伺服馬達轉軸可轉動至 180 度。使用伺服馬達並於扇葉黏合紙板，模擬在未使用時，雷達無偵測物體移動使伺服馬達轉動紙板遮擋出光口;於使用中，雷達偵測物體靠近或遠離控制伺服馬達開啟出光口。




**圖四 Arduino MKR 控制伺服馬達線路圖**

1. 將 radar shield broad 組合上 Arduino MKR，雷達的四個腳位(TD/PD/VIN/GND)分別對應 Arduino MKR 四個腳位(A1/A2/VCC/GND)，如圖一。

2. 連接伺服馬達和 Arduino MKR 如圖四。
3. 使用傳輸線連接 Arduino MKR 及電腦，編寫 Arduino IDE 並上傳至 Arduino MKR 使偵測到靠近或遠離時伺服馬達轉軸轉動至 90 度;雷達無偵測移動時，伺服馬達轉軸轉動至 180 度。
4. 測試通電後伺服馬達是否正常運作，使 radar shield broad 感測物體移動讓伺服馬達轉軸固定在 90 度後移除電腦端傳輸線(切斷電源)使伺服馬達轉軸維持不變。

5. 將小扇葉接上伺服馬達後與手電筒、小紙盒以泡棉膠固定如圖五並打開手電筒，將紙板碎片黏上扇葉模擬打開出光口。
6. 將傳輸線接回電腦，雷達 shield board 面對牆壁靜置模擬未使用狀態，觀察伺服馬達是否成功遮擋出光口。

**圖五　固定手電筒、伺服馬達**

Code ：

```
#include <Arduino.h>

#include <bgt60-ino.hpp>

#include <bgt60-platf-ino.hpp>

#include <Servo.h>
Servo myservo;

#ifndef TD
#define TD  15
#endif

#ifndef PD
#define PD  16
#endif

Bgt60Ino radarShield(TD, PD);

void setup()
{

    Serial.begin(9600);

    Error_t init_status = radarShield.init();

    if (OK != init_status) {
        Serial.println("Init failed.");
    }
    else {
        Serial.println("Init successful.");
    }

    myservo.attach(1);  // 腳位 1 設定為伺服馬達訊號腳位
}
```

```
void loop()
{

    Bgt60::Direction_t direction = Bgt60::NO_DIR;

    Error_t err = radarShield.getDirection(direction);


    if (err == OK)
    {
        switch (direction)
        {
            case Bgt60::APPROACHING:
                Serial.println("Target is approaching!");
                myservo.write(90); // 伺服馬達轉軸轉動至90度
                break;

            case Bgt60::DEPARTING:
                Serial.println("Target is departing!");
                myservo.write(90); // 伺服馬達轉軸轉動至90度
                break;

            case Bgt60::NO_DIR:
                Serial.println("Direction cannot be determined since no
motion was detected!");
                myservo.write(180);  // 伺服馬達轉軸轉動至180度
                break;
        }
    }
    /* API execution returned error */
    else{
        Serial.println("Error occurred!");
    }

    delay(500);
}
```

Code 2　雷達控制伺服馬達轉軸轉動