

Arduino

程式語法

入門篇 – BeeCar 基礎套件

Day3

Contents



Arduino

程式下載&設定

下載方式



The screenshot shows a Google search interface. The search bar contains 'arduino ide'. A red arrow points to the search bar with the text '搜尋Arduino'. Below the search bar, the search results are displayed. The first result is from 'www.arduino.cc' and is titled 'Software | Arduino'. A red arrow points to this title with the text '2.點他'. The result description states: 'Arduino IDE 1.8.13. The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino ...'. Below the result, there is a box titled '其他人也搜尋了以下項目' (Others also searched for the following items) containing three suggestions: 'arduino ide介紹', 'arduino打不開', and 'Arduino code'. On the right side of the search results, there is a card for 'Arduino IDE' software, showing the Arduino logo and a photo of an Arduino board.

Google

arduino ide ← 搜尋Arduino

全部 圖片 影片 新聞 書籍 更多 設定 工具

約有 14,900,000 項結果 (搜尋時間：0.47 秒) 在新分頁中開啟結果

www.arduino.cc › software ▾ 翻譯這個網頁

Software | Arduino ← 2.點他

Arduino IDE 1.8.13. The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino ...

Arduino IDE · Donate · Release Notes

其他人也搜尋了以下項目

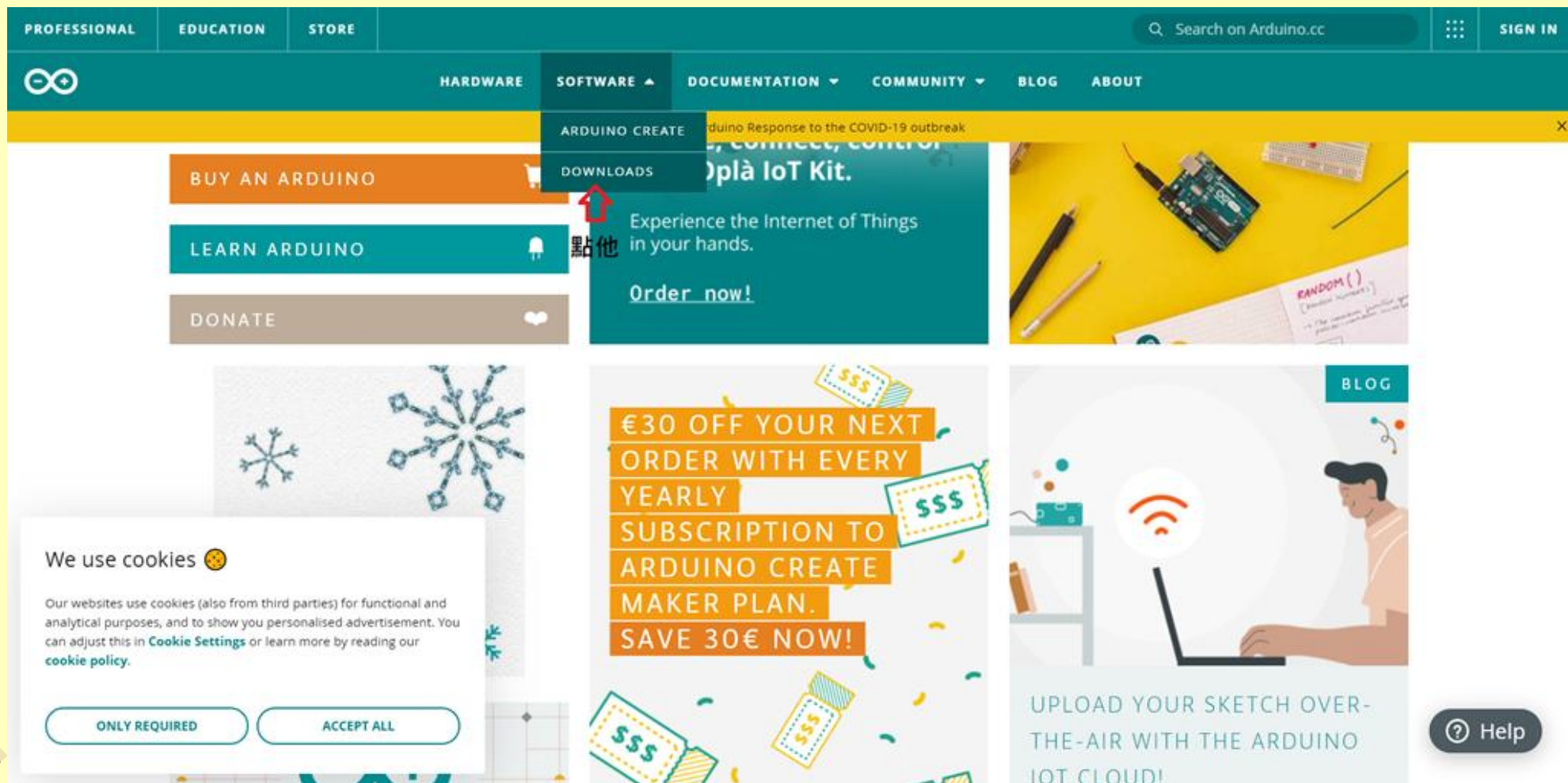
- arduino ide介紹 [Https arduino cc](https://arduino.cc)
- arduino打不開 [Arduino x64](#)
- Arduino code [Arduino online upload](#)

Arduino IDE

軟體

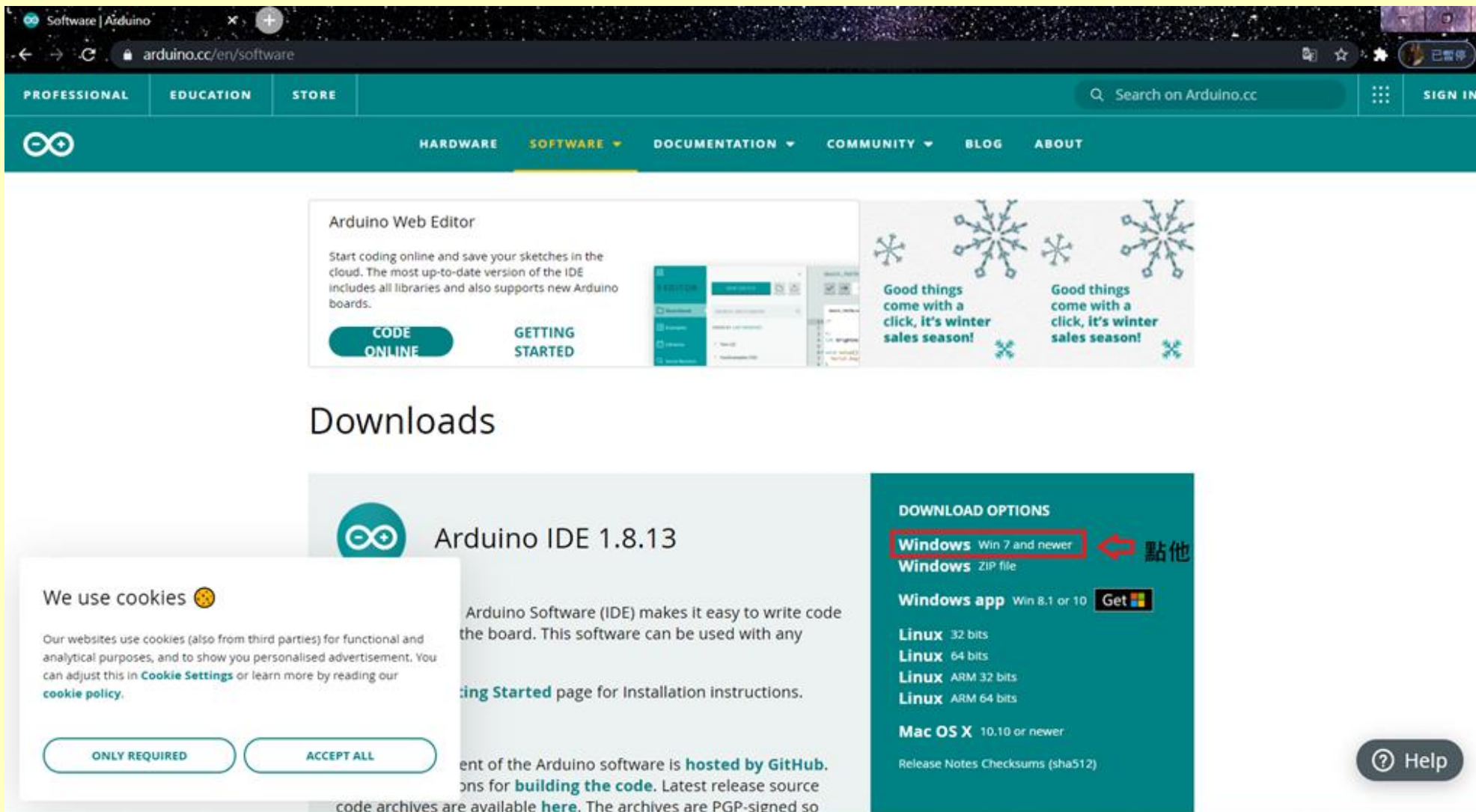
譯自英文 - Arduino集成開發
數編寫的跨平台應用程序。

下載方式



The screenshot shows the Arduino.cc website interface. The top navigation bar includes links for PROFESSIONAL, EDUCATION, and STORE. Below this, a secondary navigation bar features links for HARDWARE, SOFTWARE, DOCUMENTATION, COMMUNITY, BLOG, and ABOUT. The 'SOFTWARE' menu is expanded, showing options for ARDUINO CREATE and DOWNLOADS. A red arrow points to the 'DOWNLOADS' option, with the text '點他' (Click it) next to it. The main content area features a 'BUY AN ARDUINO' button, a 'LEARN ARDUINO' button, and a 'DONATE' button. A large banner promotes the 'Arduino Response to the COVID-19 outbreak' and the 'Dplà IoT Kit'. A cookie consent banner is visible at the bottom left, and a promotional banner for a €30 discount is in the center. A blog post titled 'UPLOAD YOUR SKETCH OVER-THE-AIR WITH THE ARDUINO IOT CLOUD!' is shown on the right.

下載方式



The screenshot shows the Arduino.cc website's 'Software' page. The navigation bar includes links for PROFESSIONAL, EDUCATION, STORE, and a search bar. The main content area features a section for the 'Arduino Web Editor' with a 'CODE ONLINE' button and a 'GETTING STARTED' link. Below this, the 'Downloads' section highlights 'Arduino IDE 1.8.13'. A 'DOWNLOAD OPTIONS' sidebar on the right lists various download methods: Windows (Win 7 and newer), Windows ZIP file, Windows app (Win 8.1 or 10), Linux (32 bits, 64 bits, ARM 32 bits, ARM 64 bits), and Mac OS X (10.10 or newer). A red box highlights the 'Windows Win 7 and newer' option, with a red arrow pointing to it and the text '點他' (Click it). A cookie consent banner is visible in the bottom left corner, and a 'Help' button is in the bottom right corner.

Software | Arduino

arduino.cc/en/software

PROFESSIONAL EDUCATION STORE

Search on Arduino.cc

SIGN IN

HARDWARE SOFTWARE DOCUMENTATION COMMUNITY BLOG ABOUT

Arduino Web Editor

Start coding online and save your sketches in the cloud. The most up-to-date version of the IDE includes all libraries and also supports new Arduino boards.

CODE ONLINE GETTING STARTED

Downloads

Arduino IDE 1.8.13

Arduino Software (IDE) makes it easy to write code the board. This software can be used with any

Getting Started page for Installation instructions.

ent of the Arduino software is hosted by GitHub. ons for building the code. Latest release source code archives are available here. The archives are PGP-signed so

DOWNLOAD OPTIONS

Windows Win 7 and newer

Windows ZIP file

Windows app Win 8.1 or 10 Get

Linux 32 bits

Linux 64 bits

Linux ARM 32 bits

Linux ARM 64 bits

Mac OS X 10.10 or newer

Release Notes Checksums (sha512)

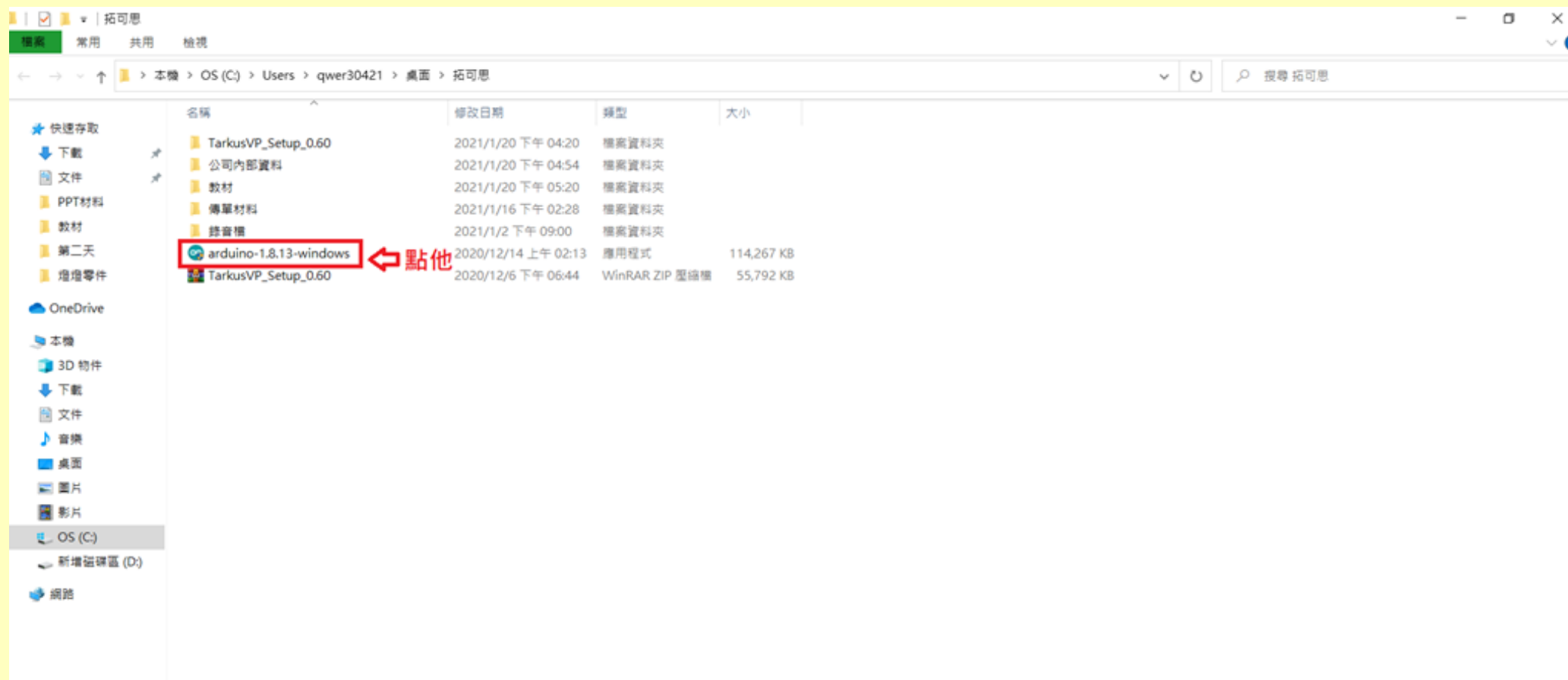
Help

We use cookies

Our websites use cookies (also from third parties) for functional and analytical purposes, and to show you personalised advertisement. You can adjust this in [Cookie Settings](#) or learn more by reading our [cookie policy](#).

ONLY REQUIRED ACCEPT ALL

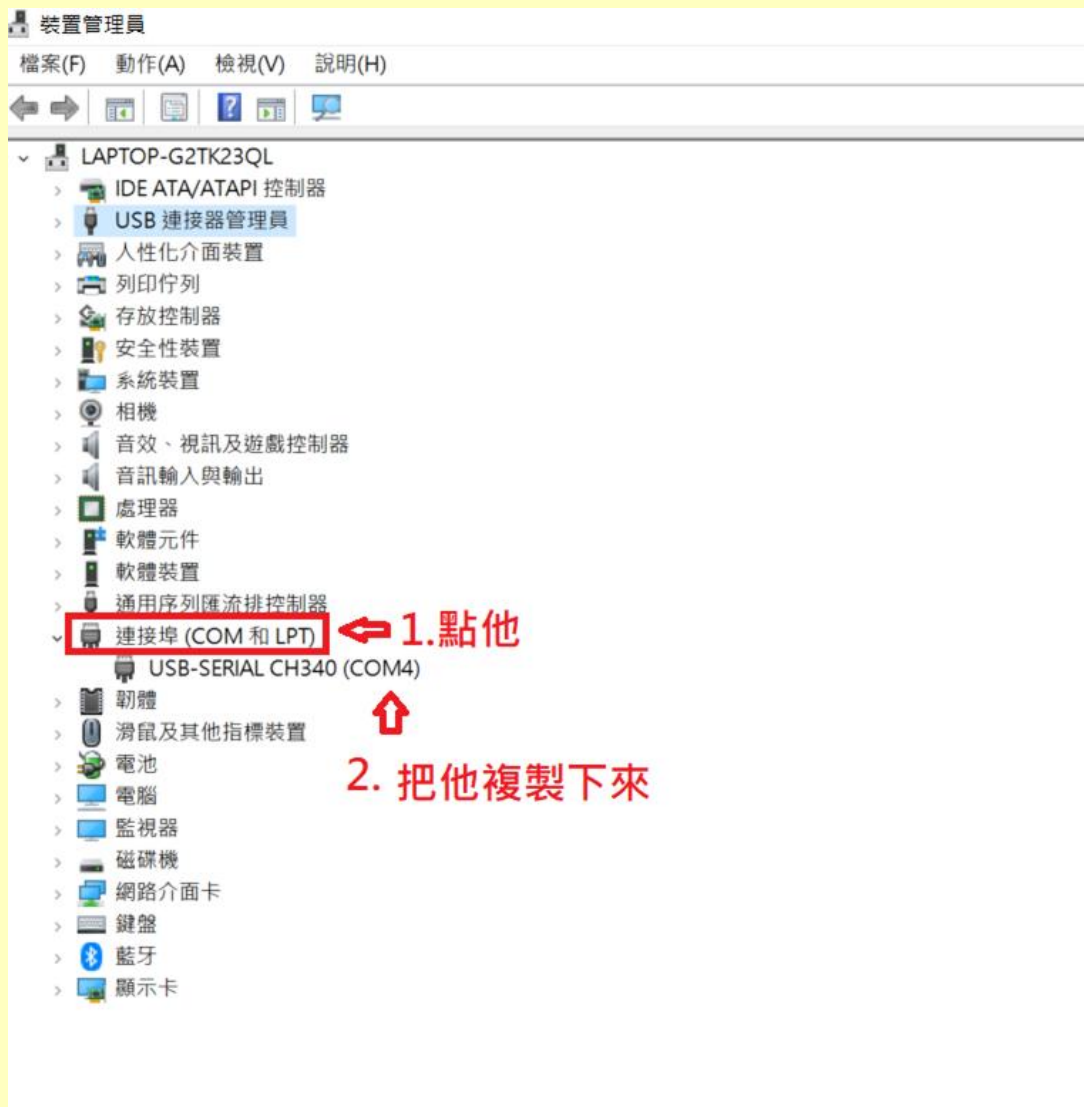
下載方式



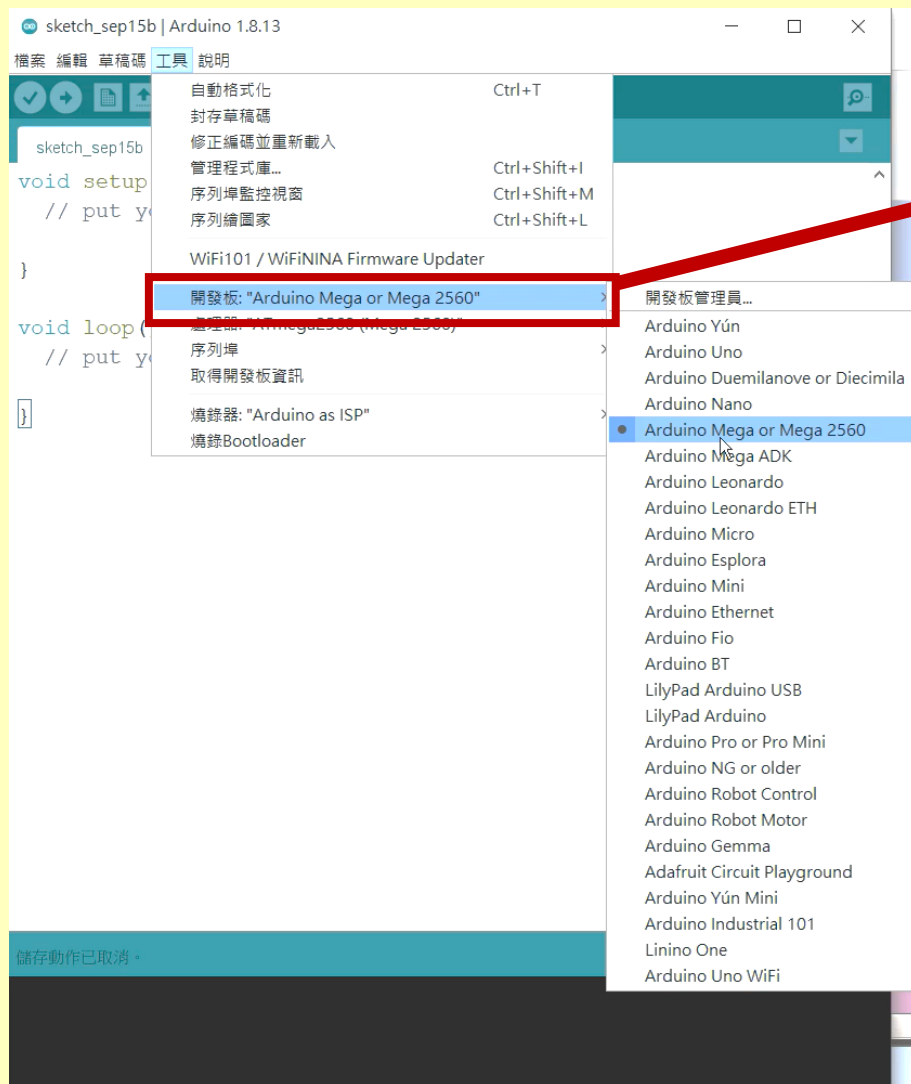
系統設定



系統設定



系統設定



選擇開發板型號，BeeCar 為
Arduino Mega 2560。

系統設定



Arduino

基本程式架構

認識語言的架構

常用的資料型態：

```

int i=0;
byte
const
const int k=3;
char ID=A;
float val=12.34;

void setup() {

pinMode(led1, INPUT);
}

void loop() {

digitalWrite(led1, HIGH/LOW);

delay(1000);

analogWrite(led1, 0-255);

delayMicroseconds();

true
false

}

```

//這區塊宣告全域變數的地方，也就是你在這邊設定的變數在這裡全部區域都能使用
//主要宣告：變數/對應名稱/引入函式庫/設定函式
//整數"變數"為i，設定初始值為0，在c語言裡以 %i , %d 來呈現
//主要是在二位元的時候才會使用到，主要是輸出的功能
//宣告的指標為固定指標
//整數"固定"為k，值為3
//命令字元變數ID，初始值為A，在c語言裡以 %c 來呈現
//實數變數為val，初始值為12.34，遇到小數點的時候使用，可以把它當作計算機的功能使用，在c語言裡以 %f , %e , %g 來呈現

// put your setup code here, to run once (將您的設置代碼放在這裡，一次運行)
//將你要設定的硬體資訊，其功能，放在這個區塊，讓他去做一個功能設定的啟動
//這裡設定的變數，只會在這邊區域執行：

//這是設定PIN角的輸出或輸入，通常會接 (我們設定的工作接腳"接腳編號數字或大寫字母"，模式"INPUT或OUTPUT")；

// put your main code here, to run repeatedly (將您的主程式放在這裡，它會重複運行你想執行的動作)
//這裡設定的變數，只會在這邊區域執行：
//要設定你的指令前一定要加"{"這個符號，代表這裡是你的程式執行的起點

//這是設定pin角的通電或斷電，通常會接 (HIGH通電，LOW斷電)

//這邊是設定你的動作要持續多久，通常這邊都以毫秒為單位 (1秒=1000毫 秒)

//調整其亮度或者驅動一個電機，控制其轉速，設定PIN角+你要輸出的數值： 从 0 (off) 到 255 (on) .

//這邊是設定你的動作要持續多久，通常這邊都以微秒為單位 (1秒=1000毫 秒)

//代表 (是) 或 (1)
//代表 (否) 或 (0)

//最後一定要加"}"這個符號，代表這一連串指令的終點



認識語言的架構

算術運算子：

運算子	動作	範例	說明
+	加法	$a+b$	a與b相加
-	減法	$a-b$	a與b相減
*	乘法	$a*b$	a與b相乘
/	除法	a/b	a除以b的商數
%	餘數	$a\%b$	a除以b的餘數
++	遞增	$a++$	a值加1，即 $a=a+1$
--	遞減	$a--$	a值減1，即 $a=a-1$
=	等於	$a=b$	a等於b的值

關係運算子：

運算子	動作	範例	說明
==	等於	$a==b$	a等於b? 是為true，否為false
!=	不等於	$a!=b$	a不等於b? 是為true，否為false
<	小於	$a<b$	a小於b? 是為true，否為false
>	大於	$a>b$	a大於b? 是為true，否為false
<=	小於等於	$a<=b$	a小於等於b? 是為true，否為false
>=	大於等於	$a>=b$	a大於等於b? 是為true，否為false

認識語言的架構

邏輯運算子：

運算子	動作	範例	說明
&&	AND	a&&b	a與b執行邏輯AND運算 兩數皆為真時 結果才為真
	OR	a b	a與b執行邏輯OR運算 有任一數為真時 結果為真
!	NOT	!a	a與b執行邏輯NOT運算 原數為真，結果為假 原數為假，結果為真

在邏輯運算中，非0的數即為真(true)，為0的數即為假(false)

位元運算子：

運算子	動作	範例	說明
&	AND	a&b	a與b兩數每一同位元 執行AND運算
	OR	a b	a與b兩數每一同位元 執行OR運算
^	XOR	a^b	a與b兩數每一同位元 執行XOR運算
~	補數	~a	將a中的每一位元相反
<<	左移	a<<2	將a內含的值 左移2個位元
>>	右移	a>>2	將a內含的值 右移2個位元

認識語言的架構

複合運算子：

運算子	動作	範例	說明
<code>+=</code>	加法	<code>a+=b</code>	同 <code>a=a+b</code> 運算
<code>-=</code>	減法	<code>a-=b</code>	同 <code>a=a-b</code> 運算
<code>*=</code>	乘法	<code>a*=b</code>	同 <code>a=a*b</code> 運算
<code>/=</code>	除法	<code>a/=b</code>	同 <code>a=a/b</code> 運算
<code>%=</code>	餘數	<code>a%=b</code>	同 <code>a=a%b</code> 運算
<code>&=</code>	遞增	<code>a&=b</code>	同 <code>a=a&b</code> 運算
<code> =</code>	遞減	<code>a =b</code>	同 <code>a=a b</code> 運算
<code>^=</code>	等於	<code>a^=b</code>	同 <code>a=a^b</code> 運算

C 運算子：

遞增：`++a`

先自增在參與運算

遞減：`--a`

先自減在參與運算

Arduino 語法

簡易除錯 –

確認 Arduino IDE 中開發板及序列埠是否選擇正確。
Tarkus VP 與 Arduino IDE 不可重複登入。

```
do{  
  s=s+i;  
  i++;  
  Serial.println(s);  
  delay(100);  
}|
```

確認括號是否有對應，{}、[]、()。

```
void loop() {  
  // put your main code here, to run repeatedly:  
  
  int i=0,s=0;  
  do{  
    s=s+i  
    i++;  
    Serial.println(s);  
    delay(100);  
  }  
  while(i<=10);  
}
```

確認在該輸入分號的結尾是否有加上，在上傳程式的時候，Arduino IDE 也會提醒錯誤的地方。

expected ';' before 'i'

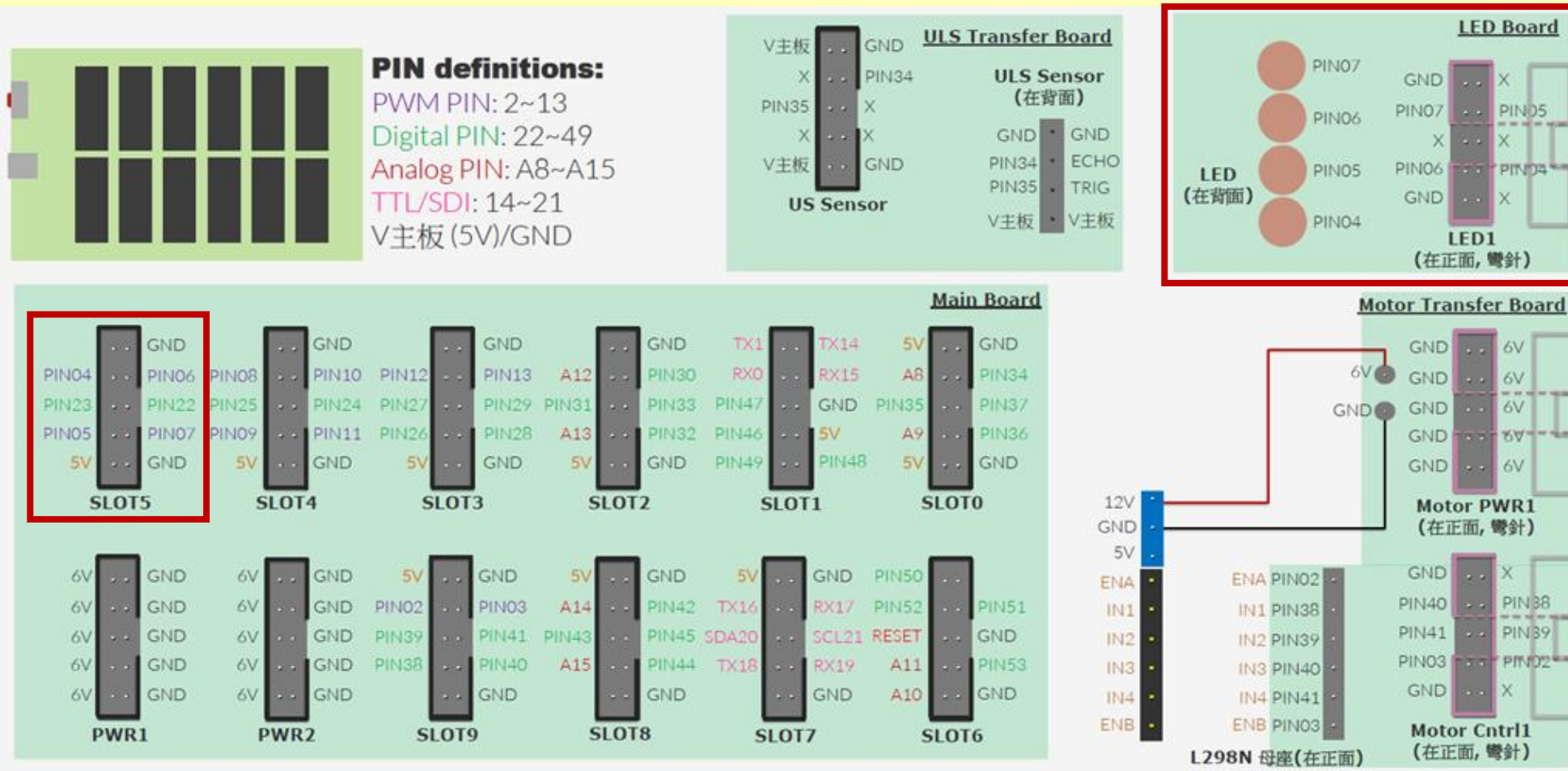
複製錯誤訊息

exit status 1
expected ';' before 'i'

LED控制

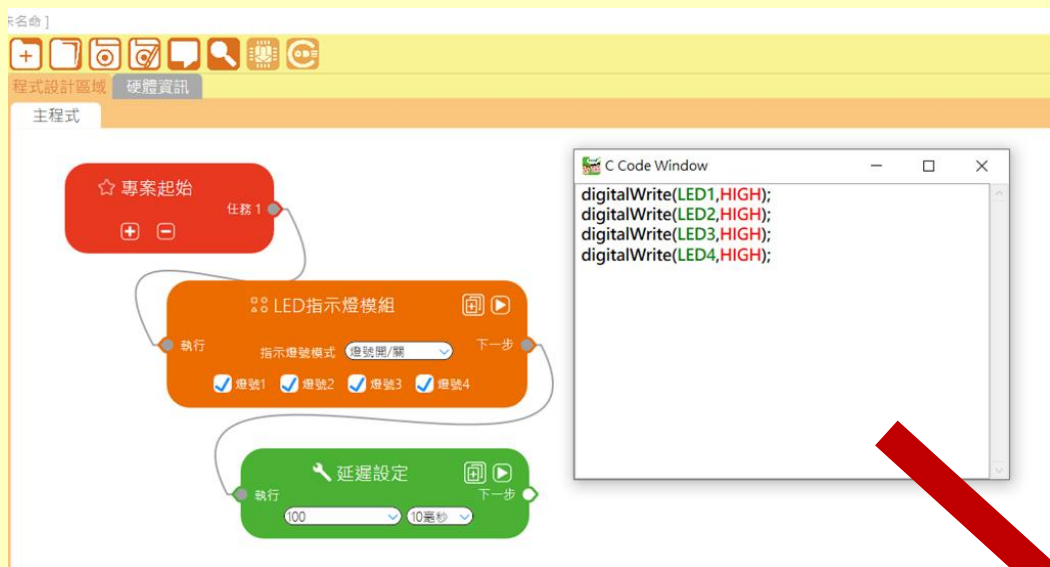
Arduino 語法

LED 腳位 -



Arduino 語法

LED 宣告(數位digital) –



Tarkus VP 畫

面

```

//LED開關的語法
//設定變數為led1~4為腳位4,5,6,7

int led1=4;
int led2=5;
int led3=6;
int led4=7;

void setup() {
    // put your setup code here, to run once:
    pinMode(led1,OUTPUT);
    pinMode(led2,OUTPUT);
    pinMode(led3,OUTPUT);
    pinMode(led4,OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    {digitalWrite(led1,HIGH);
    digitalWrite(led2,HIGH);
    digitalWrite(led3,HIGH);
    digitalWrite(led4,HIGH);
    delay(1000);
}
}
    
```

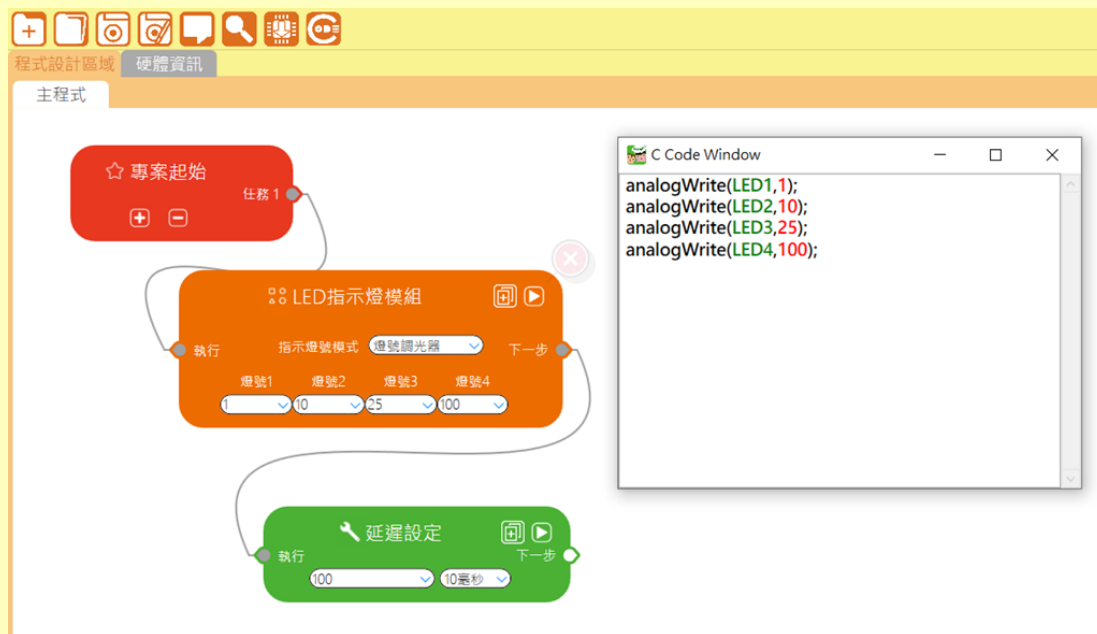
※這邊設定模式後會默認4、5、6、7為對應開發板的腳位。

//這邊將LED1~4腳位都執行輸出的設定

//這邊將Led1~4都設定為通電(高電位)

Arduino 語法

LED 宣告(類比analog) –



Tarkus VP 畫

面

```

//這是調光器的語法
//設定變數為led1~4為腳位4,5,6,7

int led1=4;
int led2=5;
int led3=6;
int led4=7;
void setup() {
    // put your setup code here, to run once:
    pinMode(led1,OUTPUT);
    pinMode(led2,OUTPUT);
    pinMode(led3,OUTPUT);
    pinMode(led4,OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    {analogWrite(led1,1);
    analogWrite(led2,10);
    analogWrite(led3,25);
    analogWrite(led4,100);
    delay(1000);
    }
}
    
```

//這邊將LED1~4腳位都執行輸出的設定

//設定LED燈號亮度，範圍：0~255。

※如果要將Tarkus VP 內的 C Code 複製出來貼至 Arduino IDE 主程式，要注意變數的宣告部分，必須輸入 Arduino 中我們宣告的變數名。

分組討論

閃爍燈

如何用Arduino讓LED燈一閃一閃呢？

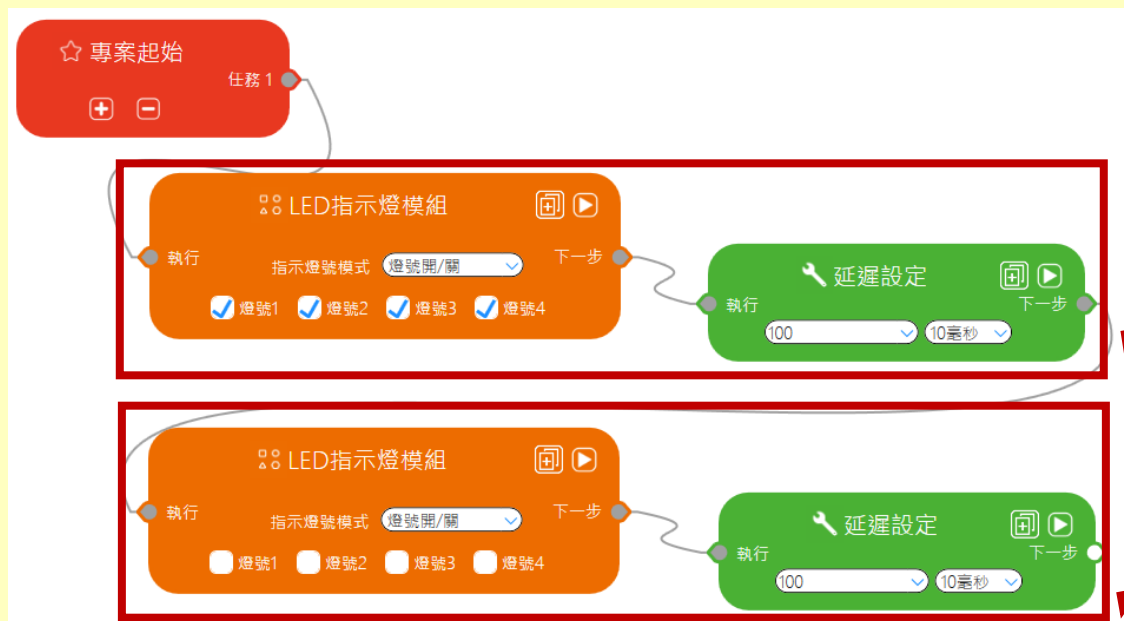


解答

Tarkus VP 畫面

Arduino 畫面

閃爍燈範例 –



```
int led1=4; //設定變數為led1~4為腳位4,5,6,7
int led2=5;
int led3=6;
int led4=7;
void setup() {
    // put your setup code here, to run once:
    pinMode(led1,OUTPUT); //這邊將LED1~4腳位都執行輸出的設定
    pinMode(led2,OUTPUT);
    pinMode(led3,OUTPUT);
    pinMode(led4,OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    {digitalWrite(led1,HIGH); //這區塊將Led1~4設定都設定為通電(高電位)
    digitalWrite(led2,HIGH);
    digitalWrite(led3,HIGH);
    digitalWrite(led4,HIGH);
    delay(1000); //延遲秒數我們設定1000毫秒
    }
    {digitalWrite(led1,LOW); //這區塊將Led1~4設定都設定為斷電(低電位)
    digitalWrite(led2,LOW);
    digitalWrite(led3,LOW);
    digitalWrite(led4,LOW);
    delay(1000); //延遲秒數我們設定1000毫秒
    }
}
```

Arduino 語法

陣列(array) –

陣列是指存放在記憶體中一群資料型態相同的集合體，在宣告時需要包含：1. 資料型態、2. 陣列名稱、3. 陣列大小及4. 陣列的初始值。

陣列：通常一個變數只能存一個值，陣列可以讓變數存放很多不同的數值，再用裝盒子的方式，把每個不同數值間隔分類好，把大括號當作盒子，數值用逗號區隔

一維陣列：

```
int i;  
const int led[]={4,5,6,7};      //製作出一個led的大盒子，並去把裡面的腳位分成:led[0]=4, led[1]=5, led[2]=6, led[3]=7
```

二維陣列：

```
int a,b;  
int c[2][4]=  
{  
    {0,1,2,3},  
    {1,2,3,4},  
}
```

```
c[0][0]=0 =>指調出[0,1,2,3]的0號資料  
c[0][1]=1 =>指調出[0,1,2,3]的1號資料  
c[0][2]=2 =>指調出[0,1,2,3]的2號資料  
c[0][3]=3 =>指調出[0,1,2,3]的3號資料  
c[1][0]=1 =>指調出[1,2,3,4]的1號資料  
c[1][1]=2 =>指調出[1,2,3,4]的2號資料  
c[1][2]=3 =>指調出[1,2,3,4]的3號資料  
c[1][3]=4 =>指調出[1,2,3,4]的4號資料
```



Arduino 語法

For 迴圈 –

```
for  
int i;  
const int led[]={4,5,6,7};
```

```
void setup() {  
    // put your setup code here, to run once:  
    for(int i=0;i<4;i++)
```

```
{  
    pinMode(led[i],OUTPUT);
```

```
}  
}
```

```
void loop() {  
    // put your main code here, to run repeatedly:  
    for(int i=0;i<4;i++){
```

```
        digitalWrite(led[i],HIGH);
```

```
}  
}
```

//迴圈的寫法

//將整數"變數"設為i

//整數"固定"為led，值設定成腳位4,5,6,7，並用一維陣列區隔開來

//for(初始值(起始)；條件式(終點)；遞增或是遞減)

//當條件式成立時，大括號裡的程式區塊就會去執行迴圈裡所設的條件次數

//將i值帶入；設定輸出輸入

執行迴圈 0、1、2、3，共4次。

//for(初始值(起始)；條件式(終點)；遞增或是遞減)

//當條件式成立時，大括號裡的程式區塊就會去執行迴圈裡所設的條件次數

//如i<4，所以他只會執行0、1、2、3，4次的迴圈循環動作

//將i值帶入；寫入你要執行的指令

執行迴圈 0、1、2、3，共4次。

分組討論

跑馬燈

如何用Arduino讓LED燈跑起來呢？



解答

Tarkus VP 畫面

Arduino 畫面

跑馬燈範例1 -



```
void loop() {  
    // put your main code here, to run repeatedly:
```

```
{digitalWrite(led1,HIGH);  
digitalWrite(led2,LOW);  
digitalWrite(led3,LOW);  
digitalWrite(led4,LOW);  
}  
delay(1000);
```

led 1 高電位HIGH，
持續時間1秒。

```
{digitalWrite(led1,LOW);  
digitalWrite(led2,HIGH);  
digitalWrite(led3,LOW);  
digitalWrite(led4,LOW);  
}  
delay(1000);
```

led 2 高電位HIGH，
持續時間1秒。

```
{digitalWrite(led1,LOW);  
digitalWrite(led2,LOW);  
digitalWrite(led3,HIGH);  
digitalWrite(led4,LOW);  
}  
delay(1000);
```

led 3 高電位HIGH，
持續時間1秒。

```
{digitalWrite(led1,LOW);  
digitalWrite(led2,LOW);  
digitalWrite(led3,LOW);  
digitalWrite(led4,HIGH);  
}  
delay(1000);  
}
```

led 4 高電位HIGH，
持續時間1秒。

解答

Arduino 畫面

迴圈的寫法-

```
int i; //將整數"變數"設為i
const int led[]={4,5,6,7}; //整數"固定"為led，值設定成腳位4,5,6,7
void setup() {
    // put your setup code here, to run once:
    for(int i=0;i<4;i++) //for( 初始值; 條件式; 增量或是減量)
    {
        pinMode(led[i],OUTPUT); //將i值帶入; 設定輸出
    }
}

void loop() {
    // put your main code here, to run repeatedly:
    for(int i=0;i<4;i++) //for( 初始值; 條件式; 增量或是減量)
    {
        digitalWrite(led[i],HIGH); //將i值帶入; 執行的開電的動作(高電位)
        delay(1000); //持續100毫秒
        digitalWrite(led[i],LOW); //在執行斷電的動作(低電位)
    }
}
```

※ Tinking :
LED由左至右依序亮燈後再由右至左依序亮燈。



分組討論

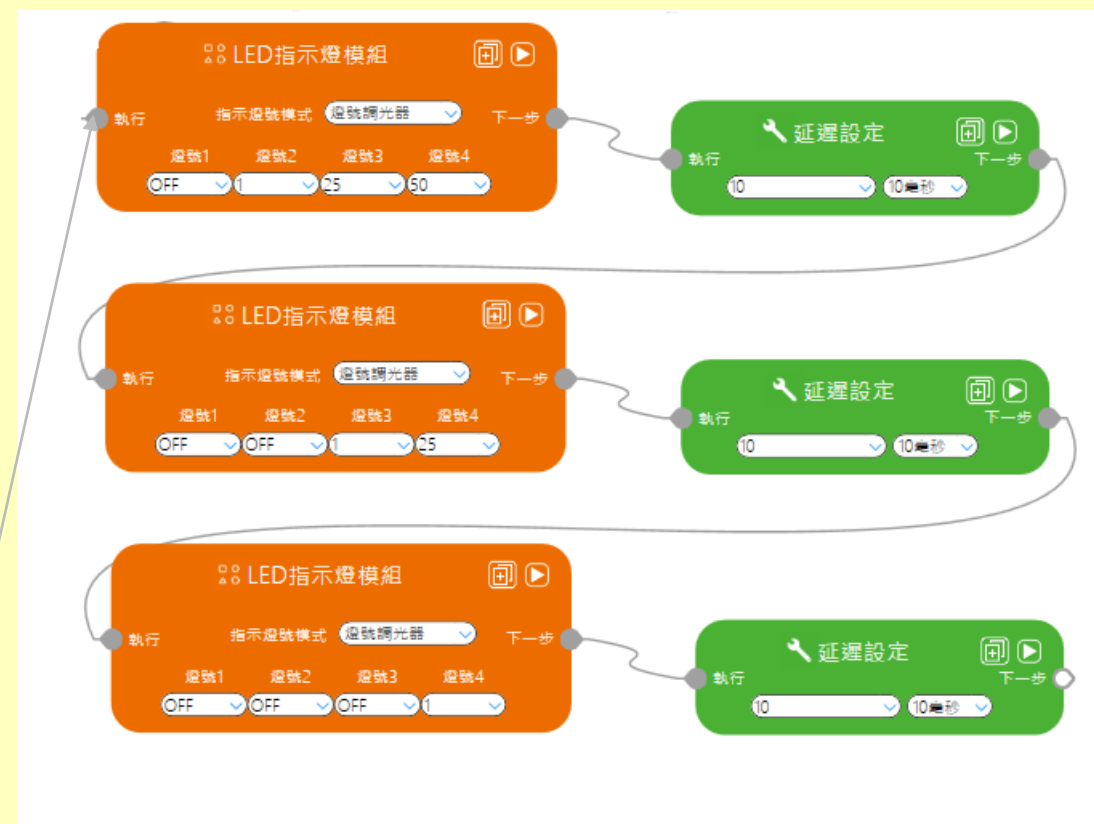
拖尾燈

如何用Arduino做出拖尾的效果?



解答

Tarkus VP 畫面



解答1

Arduino 畫面

```
void loop() {
    // put your main code here, to run repeatedly:
```

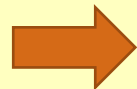
```
{ analogWrite(led1,0);
  analogWrite(led2,0);
  analogWrite(led3,0);
  analogWrite(led4,0);
}
delay(100);
```

設定led 1~4 的亮度
並持續時間100毫秒。

```
{ analogWrite(led1,100);
  analogWrite(led2,0);
  analogWrite(led3,0);
  analogWrite(led4,0);
}
delay(100);
```

```
{analogWrite(led1,50);
analogWrite(led2,100);
analogWrite(led3,0);
analogWrite(led4,0);
}
delay(100);
```

```
{analogWrite(led1,25);
analogWrite(led2,50);
analogWrite(led3,100);
analogWrite(led4,0);
}
delay(100);
```



```
{analogWrite(led1,1);
analogWrite(led2,25);
analogWrite(led3,50);
analogWrite(led4,100);
}
delay(100);
```

```
{analogWrite(led1,0);
analogWrite(led2,1);
analogWrite(led3,25);
analogWrite(led4,50);
}
delay(100);
```

```
{analogWrite(led1,0);
analogWrite(led2,0);
analogWrite(led3,1);
analogWrite(led4,25);
}
delay(100);
```

```
{analogWrite(led1,0);
analogWrite(led2,0);
analogWrite(led3,0);
analogWrite(led4,1);
}
delay(100);|
}
```

解答2

Arduino 畫面

陣列與迴圈寫法

迴圈執行結：

(led[0],brightness[0][0])
(led[1],brightness[0][1])
(led[2],brightness[0][2])
...以此類推至
(led[3],brightness[6][3])

```
const int led[]={4,5,6,7};
int varNums,ledNums;
const int brightness[7][4]= {
    {250,0,0,0},
    {100,250,0,0},
    {25,100,250,0},
    {1,25,100,250},
    {0,1,25,100},
    {0,0,1,25},
    {0,0,0,1},
};
void setup() {
    // put your setup code here, to run once:
    for(int a=0;a<4;a++){
        pinMode(led[a],OUTPUT);
    }
}
```

```
void loop() {
    // put your main code here, to run repeatedly:
    for(varNums=0;varNums<7;varNums++)
    {
        for(ledNums=0;ledNums<4;ledNums++)
        {
            analogWrite(led[ledNums],brightness[varNums][ledNums]);
        }
        delay(100);
    }
}
```

//整數"固定"為led，值設定成腳位4,5,6,7
//var (變數)的數值，led的數值
//整數"固定"為亮度的二維陣列，分別分個7個盒子，每個盒子分成4個間隔
//二維陣列的排列

//for(a=0 (起始) ; a<4 (終點) ; a做遞增)
//把a輸出到led的4,5,6,7腳位

//執行一個迴圈，varNums=0 (起點) ; varNums<7 (終點) ; varNums遞增

//執行一個迴圈，ledNums=0 (起點) ; ledNums<4 (終點) ; ledNums遞增

//這區塊將Led亮度數值設定為，led的數值，亮度的部分則是以二維陣列裡的數值去填入

//延遲100毫秒



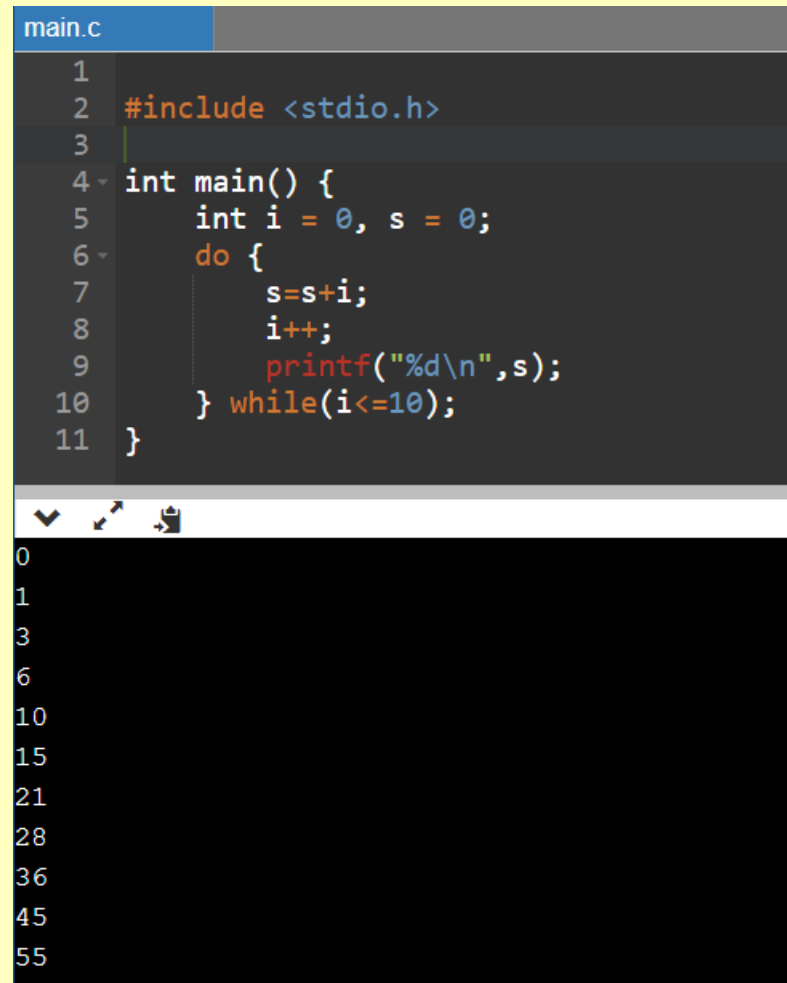
Arduino 語法

do while 迴圈 –

do while 為後判斷行迴圈，會先執行 {} 中的敘述一次，再判斷條件式，當條件為是時，繼續執行 {} 中的動作，直到條件為否則結束 do while 迴圈，此迴圈至少會執行一次。

```
void setup() {  
    // put your setup code here, to run once:  
    Serial.begin(9600);  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
    int i=0,s=0;  
    do{  
        s=s+i;  
        i++;  
        Serial.println(s);  
        delay(100);  
    }  
    while(i<=10);  
}
```

當 i 小於等於 10 持續執行迴圈，結束迴圈時 s 的值為 55。



```
main.c  
1  
2 #include <stdio.h>  
3  
4 int main() {  
5     int i = 0, s = 0;  
6     do {  
7         s=s+i;  
8         i++;  
9         printf("%d\n",s);  
10    } while(i<=10);  
11 }
```

0
1
3
6
10
15
21
28
36
45
55

Arduino 語法

if – else 敘述 –

if-else會先判斷條件式，如條件為真則執行後續內容，若條件式為假則執行else 的內容。



```
if - else //這就是比較條件的語法

void loop() {
  // put your main code here, to run repeatedly:
  if (A<50) {
    A=(A+1);
    analogWrite(led(1),A);
  }
  else
  {
    A=0;
  }
}
```

//if設定的條件

//成立了，會執行這個動作 (這是做幾何運算)

//再把運算出來的結果帶入這程式裡

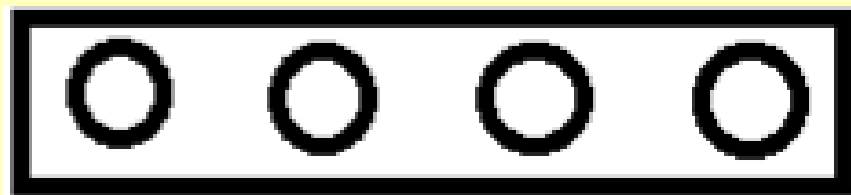
//當if設定的條件不成立時

//會執行這個動作

分組討論

呼吸燈

如何用Arduino做出呼吸的效果？

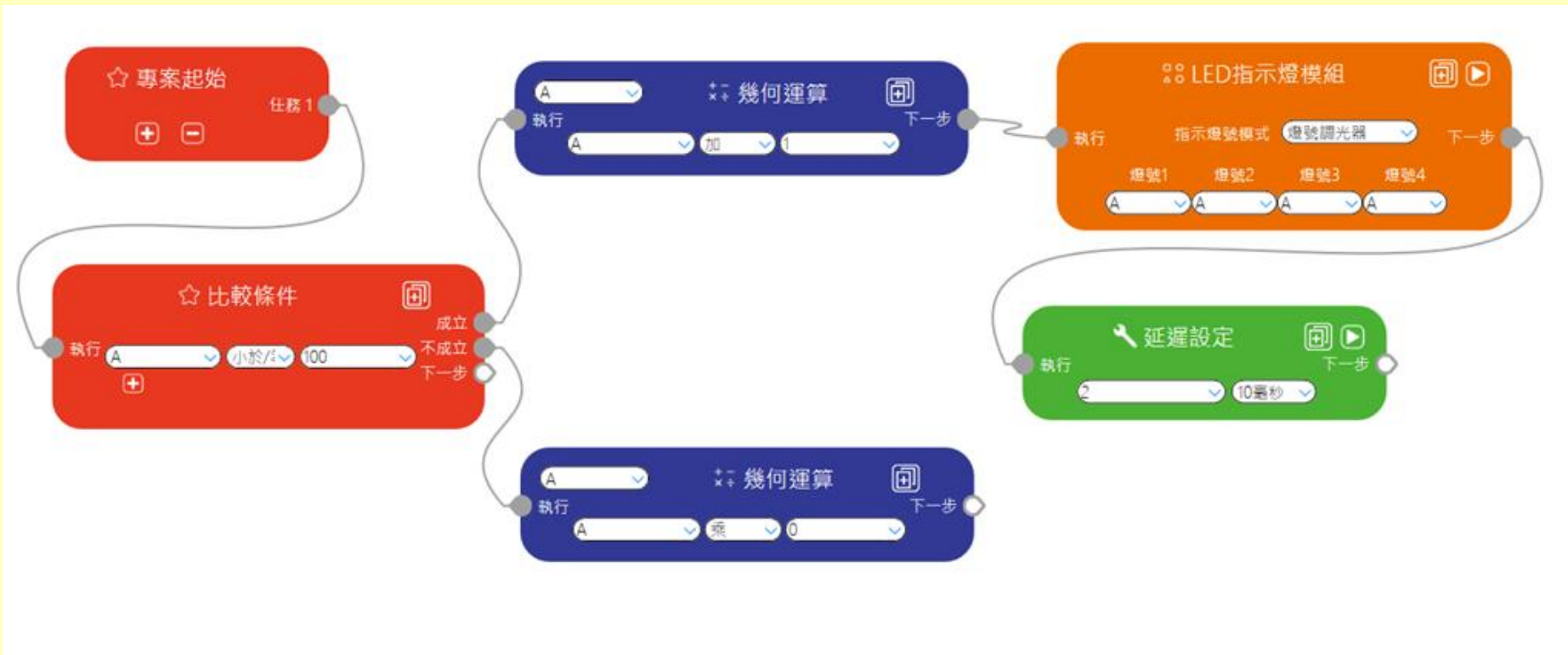


解答

Tarkus VP 畫

面

呼吸燈範例1 -



解答1

Arduino 畫面

if – else寫法-

```
int A=0;
const int led[]={4,5,6,7};
void setup() {
    // put your setup code here, to run once:
    for(int i=0;i<4;i++)
    {
        pinMode(led[i],OUTPUT);
    }
}

void loop() {
    // put your main code here, to run repeatedly:
    if (A<100) {
        A=(A+1);
        analogWrite(led[0],A);
        analogWrite(led[1],A);
        analogWrite(led[2],A);
        analogWrite(led[3],A);
        delay(50);
    }
    else
    {
        A=(A*0);
    }
}
```

//整數"變數"為A，值為0
//整數"固定"為led，值設定成腳位4,5,6,7

//for(i=0 (起始) ; i<4 (終點) ; i做遞增)

//把i的值，輸出到led的4,5,6,7腳位

//判斷變數A是否小於100
//如果是執行A=(A+1)

//將算出來的A值帶入LED的亮度
//持續50毫秒

//當A不小於100

//A的數值會直接歸零



解答2

Arduino 畫面

while 迴圈寫法

```
int A=0; //整數*變數*為A，值為0
const int led[]={4,5,6,7}; //整數*固定*為led，值設定成腳位4,5,6,7
void setup() {
    // put your setup code here, to run once:
    for(int i=0;i<4;i++) { //for( i=0(起始); i<4(終點); i欲遞增)

        pinMode(led[i],OUTPUT); //把i的值，輸出到led的4,5,6,7腳位
    }
}

void loop() {
    // put your main code here, to run repeatedly:
    if(A<100) { //判斷變數A是否小於100
        A= (A+1); //如果是執行A= (A+1)
        analogWrite(led[0],A); //將算出來的A值帶入led的亮度
        analogWrite(led[1],A);
        analogWrite(led[2],A);
        analogWrite(led[3],A);
        delay(50); //持續50毫秒
    }
    else //當A不小於100
    { //執行A大於0的條件式
        while(A>0) //A的數值會執行-1的動作
        {
            A= (A-1);

            analogWrite(led[0],A); //將算出來的A值帶入led的亮度
            analogWrite(led[1],A);
            analogWrite(led[2],A);
            analogWrite(led[3],A);
            delay(50); //持續50毫秒
        }
    }
}
```

```
int A=0;
const int led[]={4,5,6,7};

void setup() {
    // put your setup code here, to run once:
    for(int i=0;i<4;i++)
    {
        pinMode(led[i],OUTPUT);
    }
}

void loop() {
    // put your main code here, to run repeatedly:

    if(A<100)
    {
        A= (A+5);

        analogWrite(led[0],A); //HIGH高電位 LOW低電位
        analogWrite(led[1],A);
        analogWrite(led[2],A);
        analogWrite(led[3],A);
        delay(50);
    }
    else
    {
        while (A>0)
        {
            A= (A-5);
            analogWrite(led[0],A); //HIGH高電位 LOW低電位
            analogWrite(led[1],A);
            analogWrite(led[2],A);
            analogWrite(led[3],A);
            delay(50);
        }
    }
}
```

※需要在while迴圈內加入運算式，否則無法結束迴圈。

if 寫法-



Arduino 語法

函式(function) –

函式是將一些常用的敘述集合起來，在主程式需要使用到這些敘述集合時，再去呼叫執行此函式，可以減少程式碼的重複性及增加整體程式的可讀性。

範例：

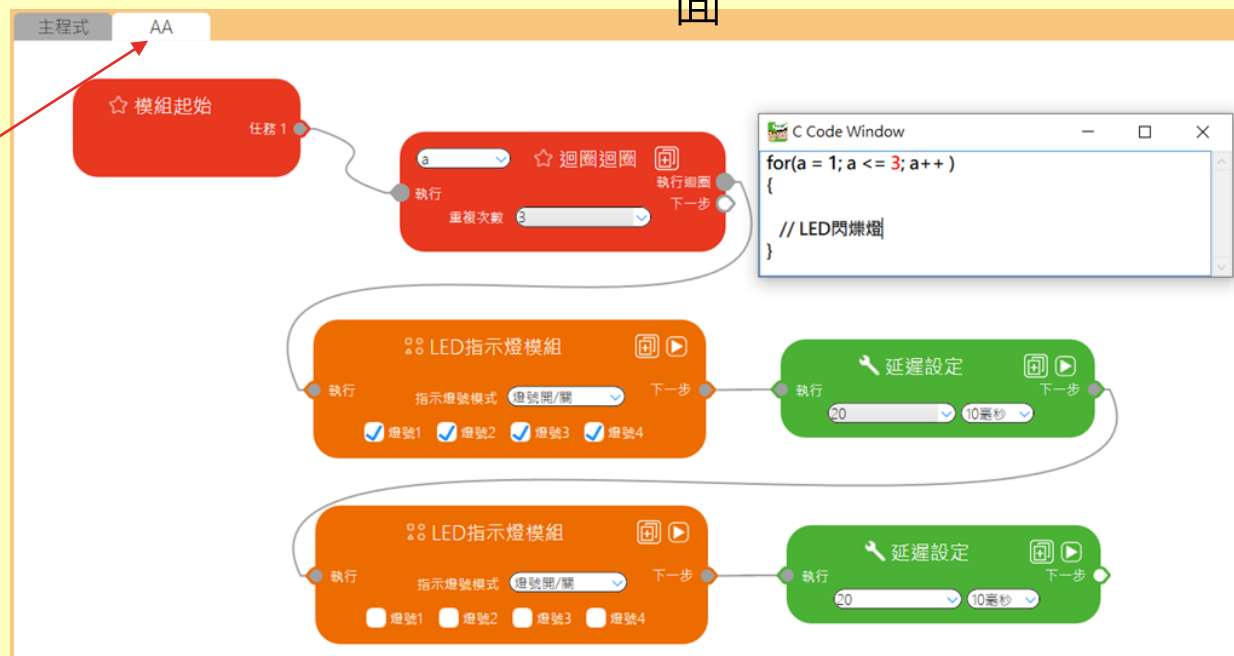
```
int i; //整數變數設為i
int a; //整數變數設為a
const int led[]={4,5,6,7};
void AA() { //設一個函式庫
  for(int a=1;a<=3;a++){ //裡面執行3次閃爍的動作
    digitalWrite(led[0],HIGH);
    digitalWrite(led[1],HIGH);
    digitalWrite(led[2],HIGH);
    digitalWrite(led[3],HIGH);
    delay(200);

    digitalWrite(led[0],LOW);
    digitalWrite(led[1],LOW);
    digitalWrite(led[2],LOW);
    digitalWrite(led[3],LOW);
    delay(200);
  }
}

void setup() {
  // put your setup code here, to run once:
  pinMode(led[0],OUTPUT);
  pinMode(led[1],OUTPUT);
  pinMode(led[2],OUTPUT);
  pinMode(led[3],OUTPUT);
}
```

Tarkus VP 畫

面



Arduino 語法

函式(function) –

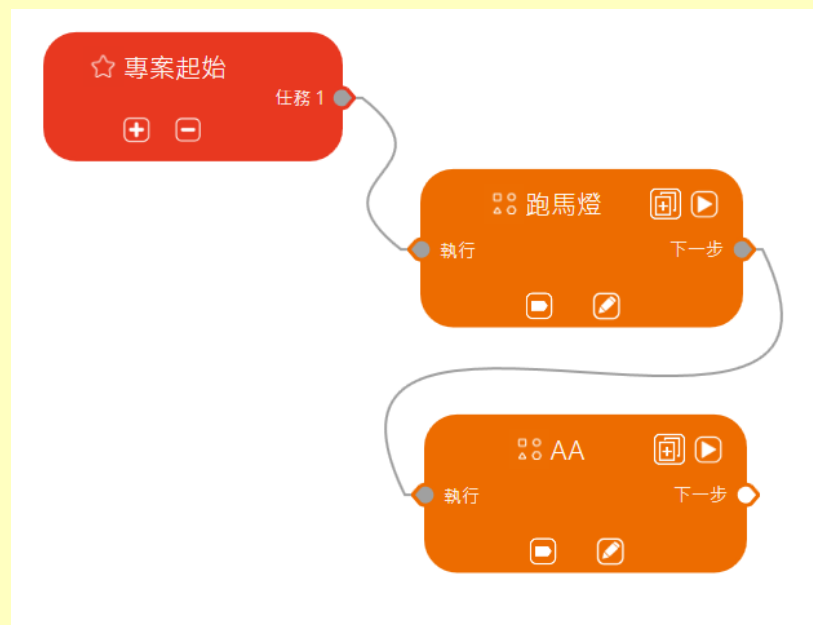
範例：執行跑馬燈一次後執行函式 AA()

```
int i;
int a;
const int led[]={4,5,6,7};
void AA()
{for(int a=1;a<=3;a++)
{
    digitalWrite(led[0],HIGH);
    digitalWrite(led[1],HIGH);
    digitalWrite(led[2],HIGH);
    digitalWrite(led[3],HIGH);
    delay(200);
}
{
    digitalWrite(led[0],LOW);
    digitalWrite(led[1],LOW);
    digitalWrite(led[2],LOW);
    digitalWrite(led[3],LOW);
    delay(200);
}
}
```

```
void setup() {
    // put your setup code here, to run once:
    for(int i=0;i<4;i++)
    {
        pinMode(led[i],OUTPUT);
    }
}

void loop() {
    // put your main code here, to run repeatedly:
    for(int i=0;i<4;i++){
        digitalWrite(led[i],HIGH);
        delay(200);
        digitalWrite(led[i],LOW);
    }
    AA();
}
```

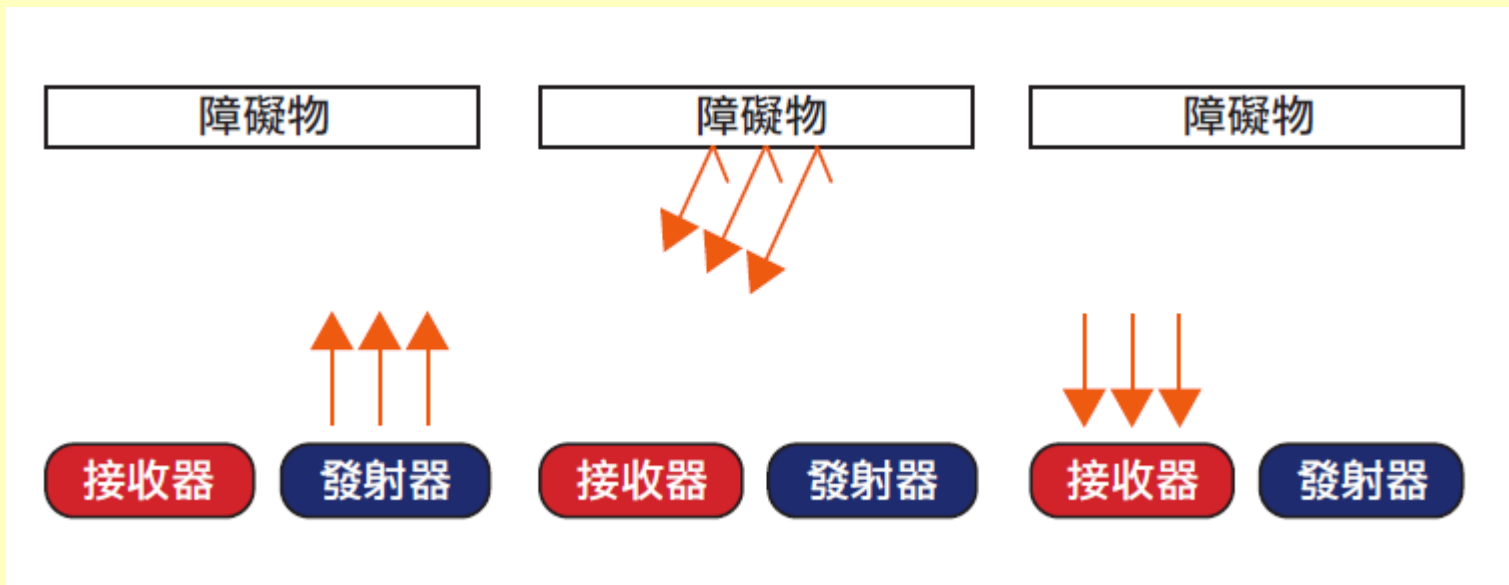
執行AA()函式內容



Tarkus VP 畫面

距離感測器

發射與接收的原理

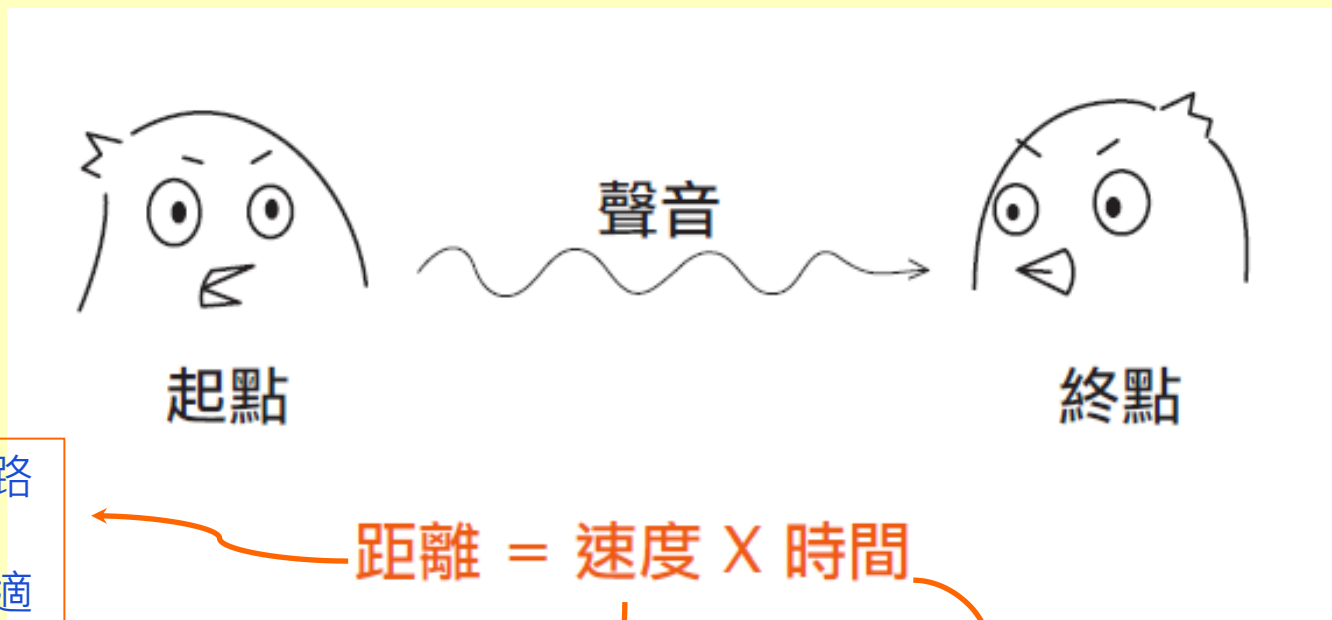


● 原理說明

超音波感測器與丟接球原理相同，感測器的右眼發出訊號，將超音波丟出，遇到障礙物時部分波反射，而感測器的左眼接收反射波。利用整個過程所花費的時間，可以推算出障礙物與反射器的距離。



聲音的傳達方式



聲波傳遞的總路徑長。
超音波感測器適用相同原理去計算車子與障礙物之間的距離。

$$\text{距離} = \text{速度} \times \text{時間}$$

由起點(講者)至終點(聽者耳朵)所花費的時間。

每單位時間所走的距離，
室溫下聲速是340m/s(每秒走340公尺)。

Arduino 語法

距離感測器腳位 -



Arduino 語法

距離感測器宣告 -



Tarkus VP 畫

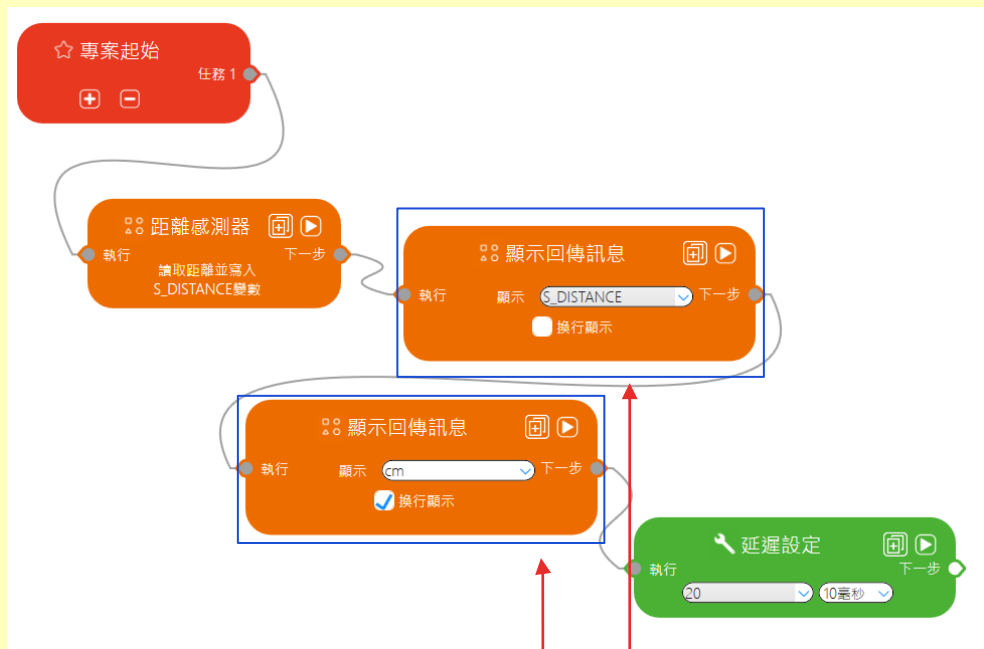
面



```
const int TrigPin = 35;           //設定整數"常數"為TrigPin角(發射)，值為35
const int EchoPin = 34;          //設定整數"常數"為EchoPin角(接收)，值為34
int duration = 0;                //設定整數變數為 持續時間，值為0
int S_DISTANCE;                  //整數變數為距離感測器
void distance()                  //傳回值型態為距離
{
    digitalWrite(TrigPin,LOW);    //將TrigPin角寫入低電位
    delayMicroseconds(2);         //延時2微秒
    digitalWrite(TrigPin,HIGH);  //將TrigPin角寫入高電位
    delayMicroseconds(10);        //延時10微秒(1秒=100萬微秒)
    digitalWrite(TrigPin,LOW);    //將TrigPin角寫入低電位
    duration = pulseIn(EchoPin,HIGH); //持續時間為脈衝輸入EchoPin的高電位
    S_DISTANCE = duration * 0.034 / 2; //計算距離：時間乘上聲音在空氣中傳送的速度(0.034)再除以 2。
}
```

Arduino 語法

讀取距離 -



Tarkus VP 畫面

```

const int TrigPin = 35;           //設定整數"常數"為TrigPin角(發射), 值為35
const int EchoPin = 34;          //設定整數"常數"為EchoPin角(接收), 值為34
int duration = 0;                //設定整數變數為 持續時間, 值為0
int S_DISTANCE;                  //整數變數為距離感測器
void distance()                  //傳回值型態為距離
{
    digitalWrite(TrigPin, LOW);   //將TrigPin角寫入低電位
    delayMicroseconds(2);         //延時2微秒
    digitalWrite(TrigPin, HIGH);  //將TrigPin角寫入高電位
    delayMicroseconds(10);        //延時10微秒
    digitalWrite(TrigPin, LOW);   //將TrigPin角寫入低電位
    duration = pulseIn(EchoPin, HIGH); //持續時間為脈衝輸入EchoPin的高電位
    S_DISTANCE = duration * 0.034 / 2; //計算距離: 時間乘上聲音在空氣中傳送的速度(0.034)再除以 2。
    delay(200);
}

void setup() {
    // put your setup code here, to run once:
    pinMode(TrigPin, OUTPUT);      //設定腳位模式 TrigPin 為輸出
    pinMode(EchoPin, INPUT);       //設定腳位模式 EchoPin 為輸入
    Serial.begin(9600);            //初始化序列埠監控視窗, 傳輸速率為 9600 鮑率
}

void loop() {
    // put your main code here, to run repeatedly:
    distance();
    Serial.print(S_DISTANCE);      //顯示計算完後的距離 S_DISTANCE
    Serial.print("cm");           //顯示字元 cm
    Serial.print("\n");           //換行顯示
    delay(200);                   //延遲 200 毫秒
}
  
```

Arduino 語法

感測器控制燈光(if-else if-else 敘述) –

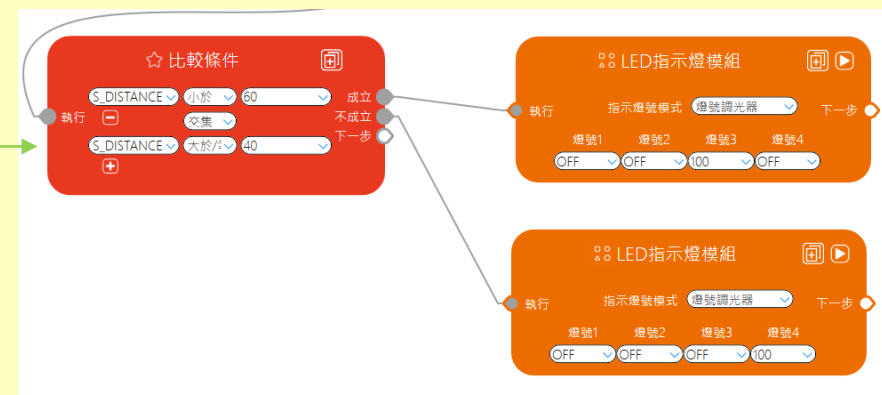
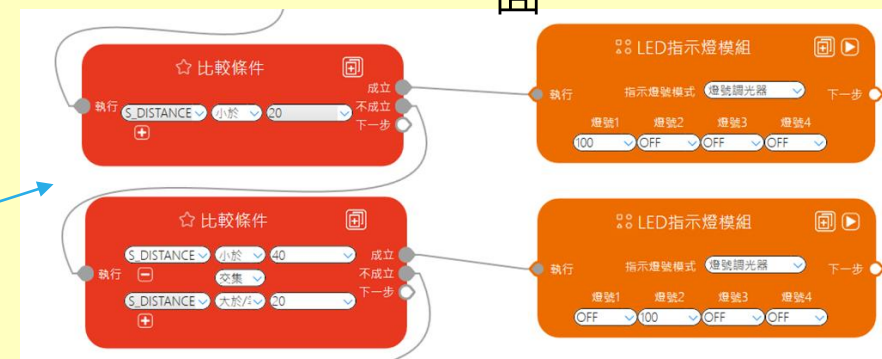
if-else if 會先判斷第一個條件式，如條件為真則執行 {} 內容，若條件式為假則執行第二個 else if 的條件式判斷，以此類推，如所有條件式皆不成立則執行 else 的內容。

Tarkus VP 畫

面

```
void loop() {
    // put your main code here, to run repeatedly:
    distance();

    if(S_DISTANCE < 20) {
        //系統會判斷第一個條件距離小於20，是否成立
        //成立時，會亮一號燈
        analogWrite(led[0],100);
        analogWrite(led[1],0);
        analogWrite(led[2],0);
        analogWrite(led[3],0);
        delay(1000);
    }
    else if((S_DISTANCE >=20 ) && (S_DISTANCE <40 ))
        //不成立，系統會判斷第二個條件距離大於等於20，小於40，是否成立 (這邊&&是指交集，所以一定要兩個條件都達成)
        //成立時，會亮二號燈
    {
        analogWrite(led[0],0);
        analogWrite(led[1],100);
        analogWrite(led[2],0);
        analogWrite(led[3],0);
        delay(1000);
    }
    else
        //都不成立，系統會判斷直接執行下面的指令
    {
        analogWrite(led[0],0);
        analogWrite(led[1],0);
        analogWrite(led[2],0);
        analogWrite(led[3],0);
        delay(1000);
    }
}
```



Arduino 語法

感測器控制燈光(巢狀 if-else 敘述) –

以右圖程式碼為例：

條件皆成立時執行敘述 1 ；

條件 1、2 成立，3 不成立時執行敘述 2 ；

只有條件 1 成立時執行敘述 3 ；

所有條件不成立則執行敘述 4 。

條件：

距離小於 20 亮 1 號燈，

距離介於 20 ~ 40 時亮 2 號燈，

距離介於 40 ~ 60 時亮 3 號燈，

距離大於 60 時亮 4 號燈。

```
distance();
Serial.print(S_DISTANCE);
Serial.print("cm");
Serial.print("\n");
delay(200);
if(S_DISTANCE>=20) {
    if(S_DISTANCE>=40) {
        if(S_DISTANCE>=60) {
            digitalWrite(led[0],LOW);
            digitalWrite(led[1],LOW);
            digitalWrite(led[2],LOW);
            digitalWrite(led[3],HIGH);
        }
        else {
            digitalWrite(led[0],LOW);
            digitalWrite(led[1],LOW);
            digitalWrite(led[2],HIGH);
            digitalWrite(led[3],LOW);
        }
    }
    else {
        digitalWrite(led[0],LOW);
        digitalWrite(led[1],HIGH);
        digitalWrite(led[2],LOW);
        digitalWrite(led[3],LOW);
    }
}
else{
    digitalWrite(led[0],HIGH);
    digitalWrite(led[1],LOW);
    digitalWrite(led[2],LOW);
    digitalWrite(led[3],LOW);
}
}
```

//顯示計算完後的距離 S_DISTANCE
//顯示字元 cm
//換行顯示
//延遲 200 毫秒

條件

//第一個條件距離大於等於20，是否成立
//第二個條件距離大於等於40，是否成立
//第三個條件距離大於等於60，是否成立
//當全部條件都成立時，會亮4號

敘述1

//只有1、2的條件成立時，會亮3號燈

敘述2

//只有第1個條件成立時，會亮2號燈

敘述3

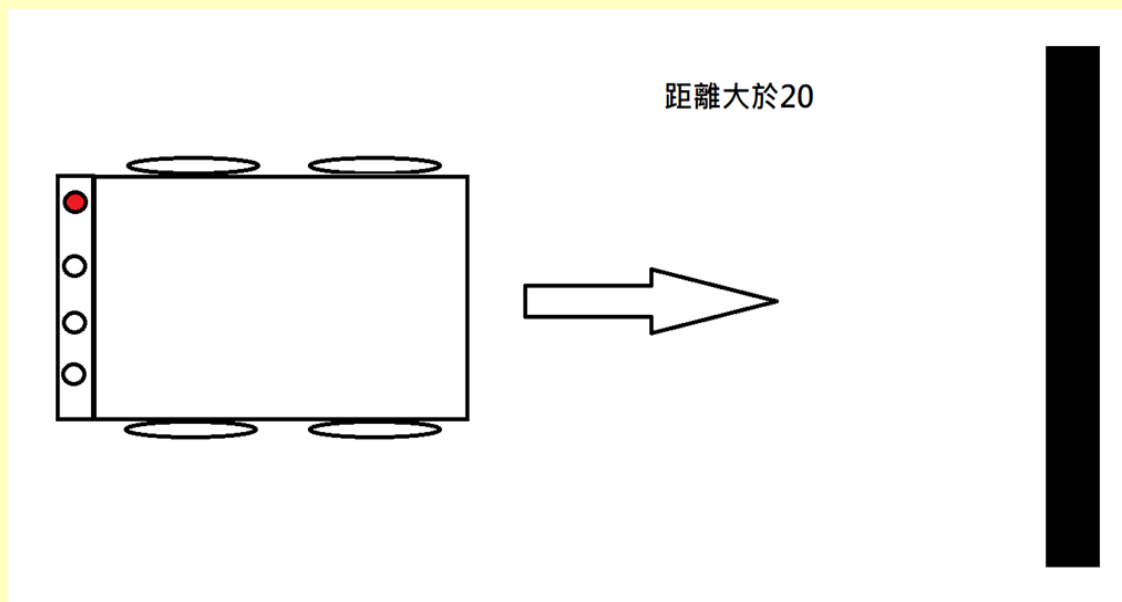
敘述4

//全部條件都不成立時，會亮1號燈

分組討論

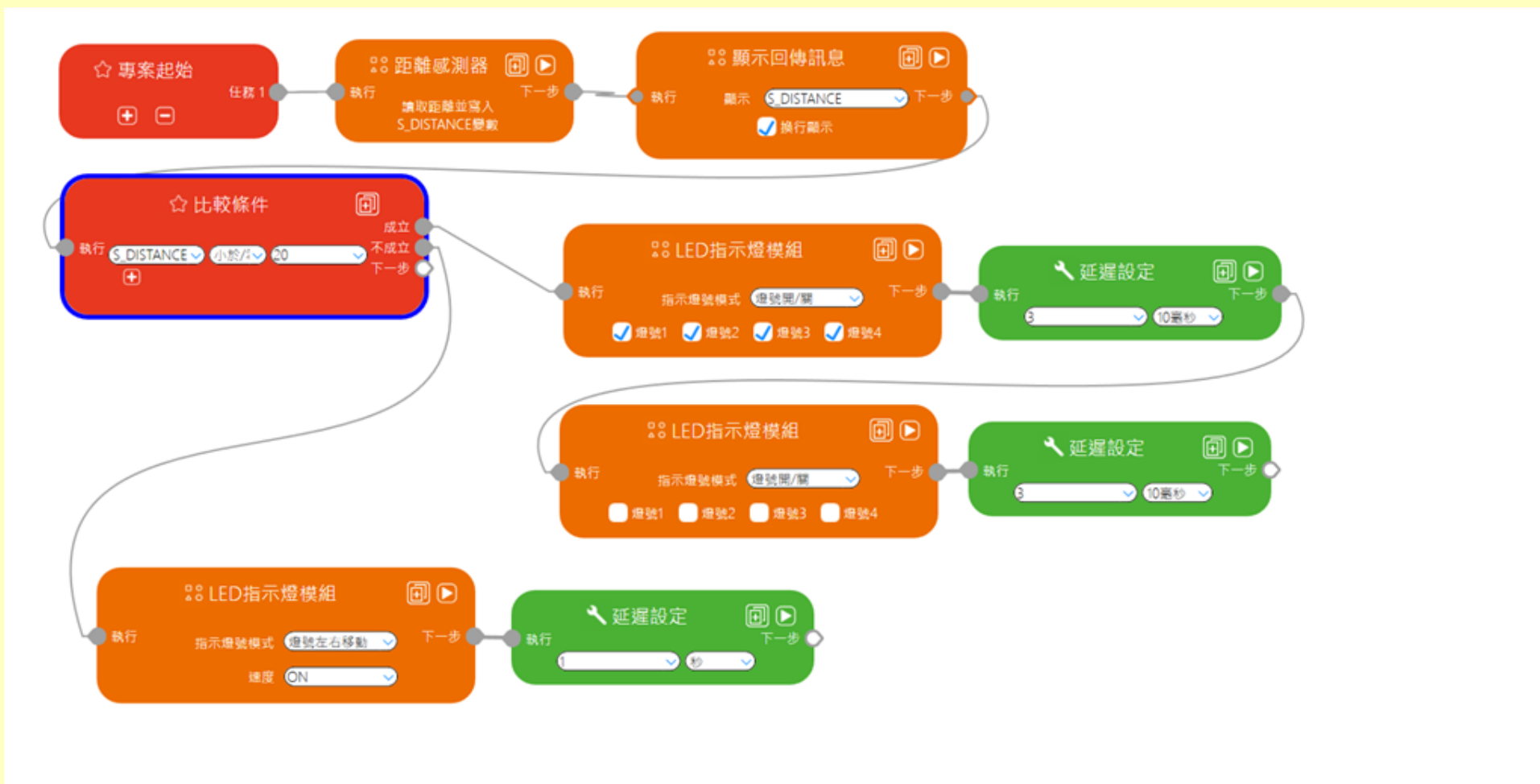
1. 距離大於20，LED燈會左右移動
2. 距離小於20，LED燈會閃爍

如何用Arduino做出這效果？



解答

Tarkus VP 畫面



解答

Arduino 畫面

```
const int led[]={4,5,6,7};
const int TrigPin=35;           //發射為35PIN腳
const int EchoPin=34;          //接收為34PIN腳
int duration=0;
int S_DISTANCE;
void distance()                 //傳回值型態為距離
{
    digitalWrite(TrigPin,LOW);
    delayMicroseconds(2);
    digitalWrite(TrigPin,HIGH);
    delayMicroseconds(10);
    digitalWrite(TrigPin,LOW);
    duration = pulseIn (EchoPin,HIGH);
    S_DISTANCE=duration*0.034/2; //計算距離：時間乘上聲音在空氣中傳送的速度(0.034)再除以 2。
    delay(100);
}

void setup()
{
    // put your setup code here, to run once:
    pinMode(led[1],OUTPUT);      //這邊將LED1~4腳位都執行輸出的設定
    pinMode(led[2],OUTPUT);
    pinMode(led[3],OUTPUT);
    pinMode(led[0],OUTPUT);
    pinMode(TrigPin,OUTPUT);     //把發射的腳位設定輸出
    pinMode(EchoPin,INPUT);      //把接收腳位設定輸入
    Serial.begin(9600);
}
```

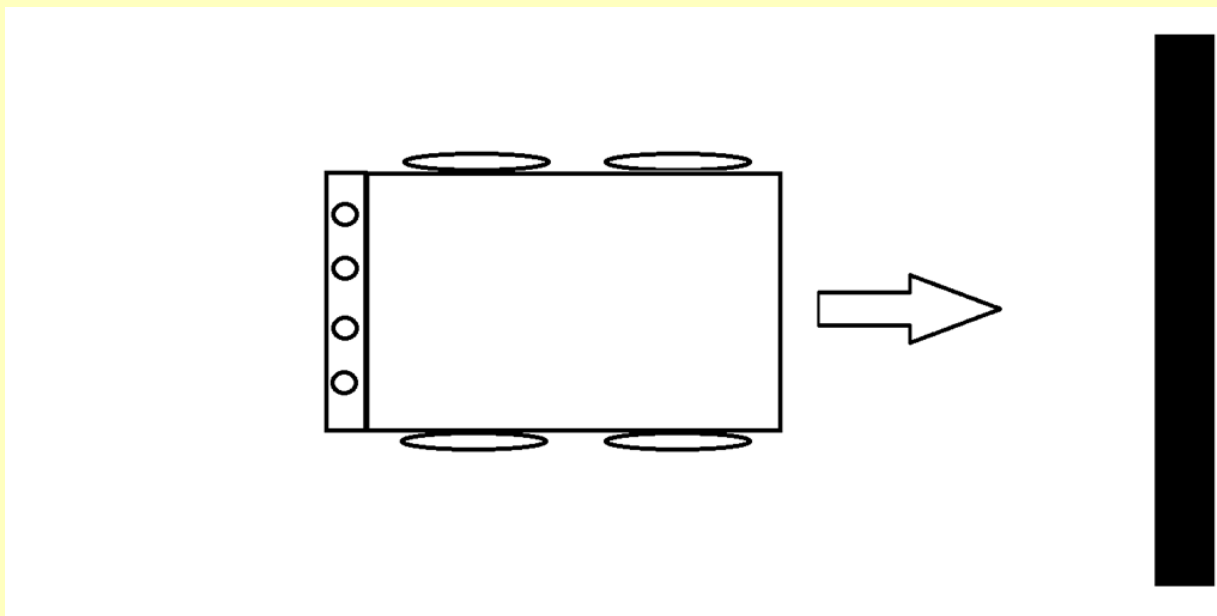
```
void loop() {
    // put your main code here, to run repeatedly:
    distance();                  //距離
    if(S_DISTANCE<20) {
        digitalWrite(led[0],HIGH);
        digitalWrite(led[1],HIGH);
        digitalWrite(led[2],HIGH);
        digitalWrite(led[3],HIGH);
        delay(100);

        digitalWrite(led[0],LOW);
        digitalWrite(led[1],LOW);
        digitalWrite(led[2],LOW);
        digitalWrite(led[3],LOW);
        delay(100);
    }
    else {
        for(int i=0;i<4;i++) {
            digitalWrite(led[i],HIGH);
            delay(100);
            digitalWrite(led[i],LOW);
        }
        for(int i=4;i>=0;i--) {
            digitalWrite(led[i],HIGH);
            delay(100);
            digitalWrite(led[i],LOW);
        }
    }
}
```

分組討論

距離越近，越亮
距離越遠，越暗

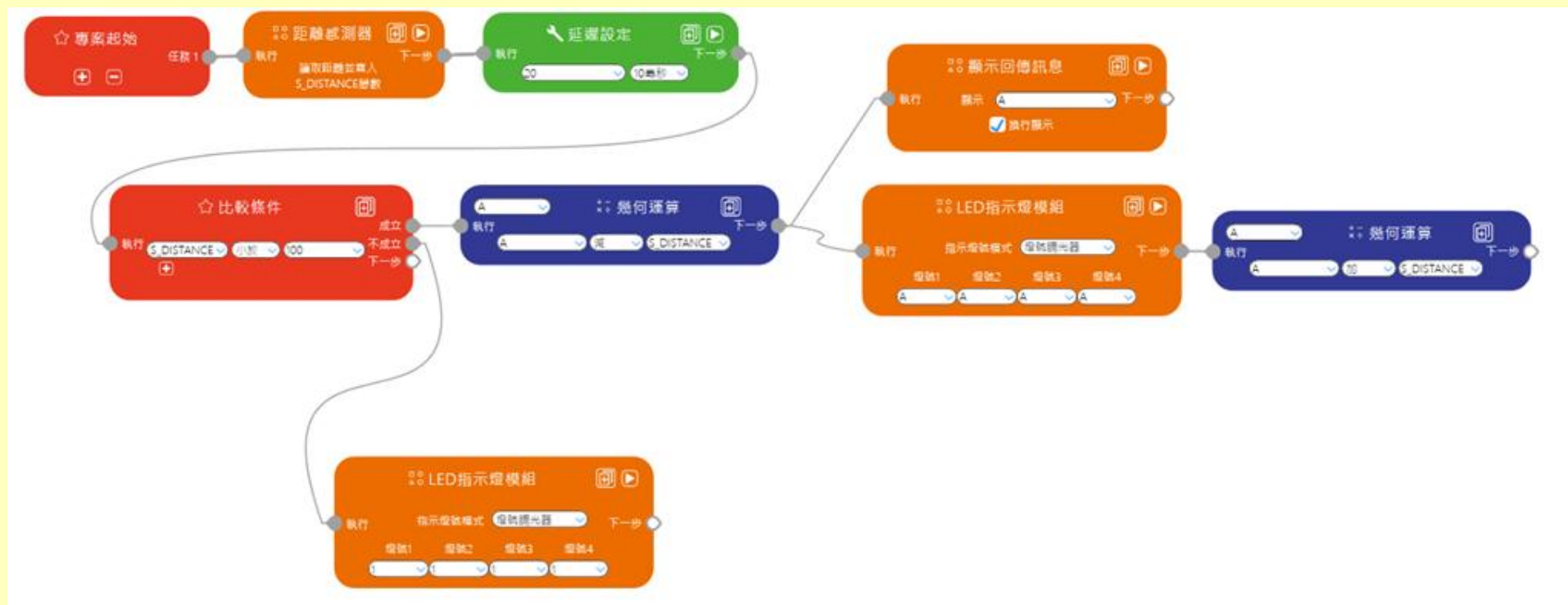
如何用Arduino做出這效果？



解答

Tarkus VP 畫面

面



解答

Arduino 畫面

```
const int led[]={4,5,6,7};
const int TrigPin=35;           //發射PIN腳35腳位
const int EchoPin=34;          //接收PIN腳34腳位
int duration=0;
int S_DISTANCE;
int A;
int B;
void distance()
{digitalWrite(TrigPin,LOW);
delayMicroseconds(2);
digitalWrite(TrigPin,HIGH);
delayMicroseconds(10);
digitalWrite(TrigPin,LOW);
duration = pulseIn (EchoPin,HIGH);
S_DISTANCE=duration*0.034/2;
}
void setup() {
    // put your setup code here, to run once:
pinMode(led[0],OUTPUT);
pinMode(led[1],OUTPUT);
pinMode(led[2],OUTPUT);
pinMode(led[3],OUTPUT);
pinMode(TrigPin,OUTPUT);
pinMode(EchoPin,INPUT);
Serial.begin(9600);
}
```

pulseIn:計算EchoPin(接收)
收到訊息所經過的時間，
這裡的經過是指訊息送出
到回來所花的時間

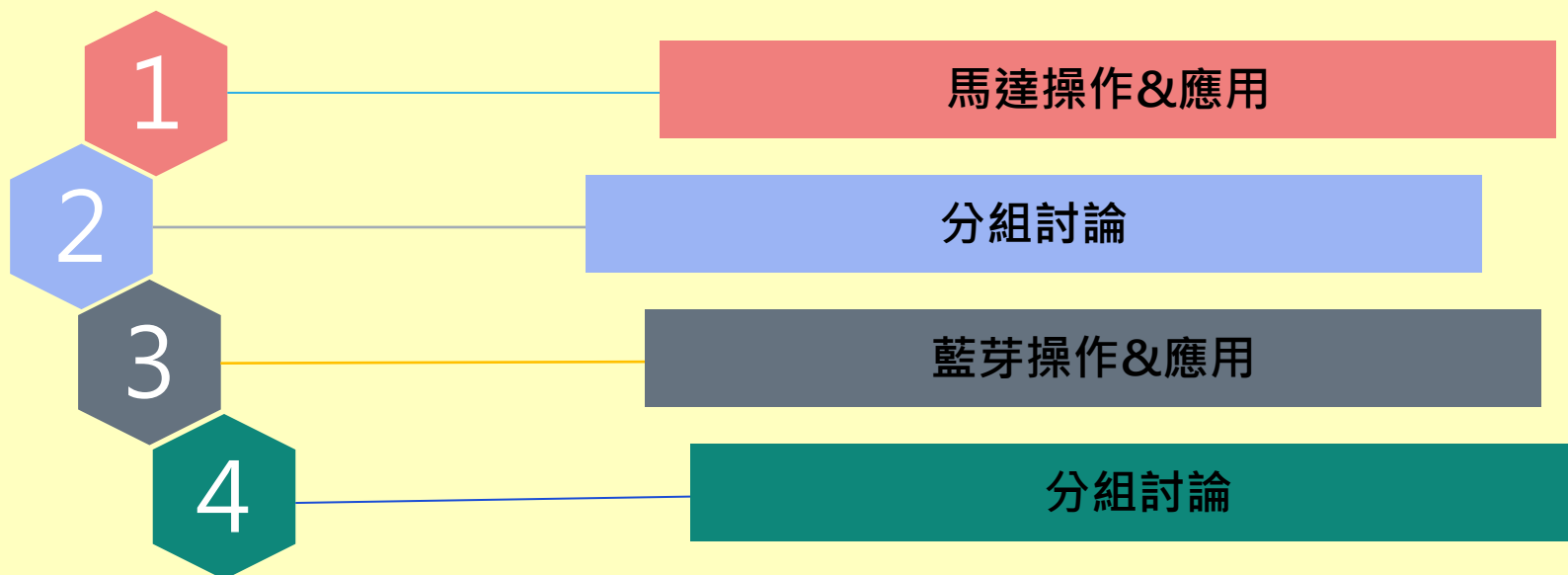
```
void loop() {
    // put your main code here, to run repeatedly:
    Serial.print(S_DISTANCE);           //顯示計算完後的距離 S_DISTANCE
    Serial.print("cm");                 //顯示字元 cm
    Serial.print("\n");                 //換行顯示
    delay(50);                          //延遲 50 毫秒，影響距離感測器的敏感度

    distance();
    B=255-S_DISTANCE;
    if(B<1){
        A=1;
        analogWrite(led[0],A);
        analogWrite(led[1],A);
        analogWrite(led[2],A);
        analogWrite(led[3],A);
        delay(50);
    }

    else{
        A=B;
        analogWrite(led[0],A);
        analogWrite(led[1],A);
        analogWrite(led[2],A);
        analogWrite(led[3],A);
        delay(50);
    }
}
```

Day4

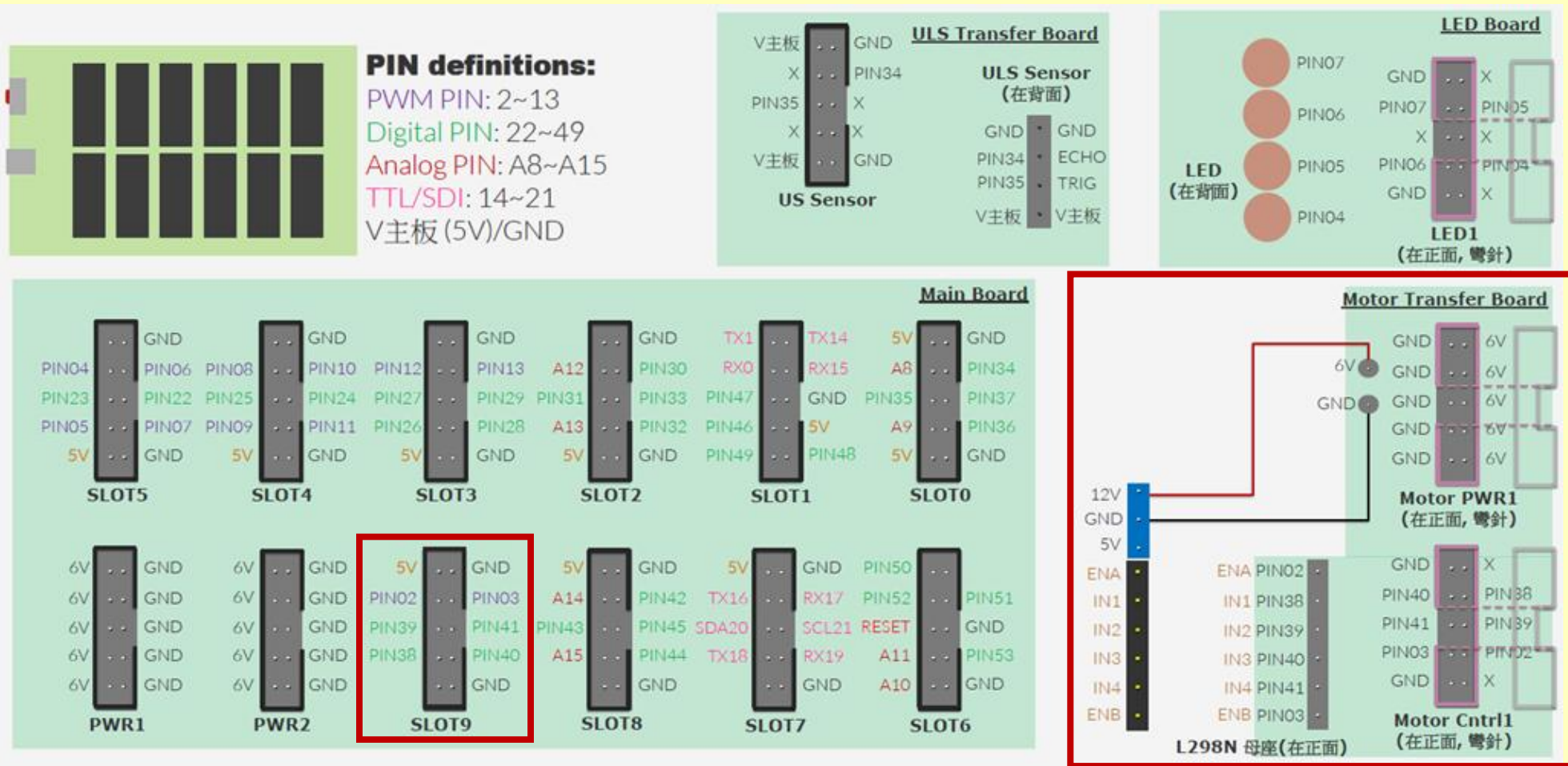
Contents



馬達控制

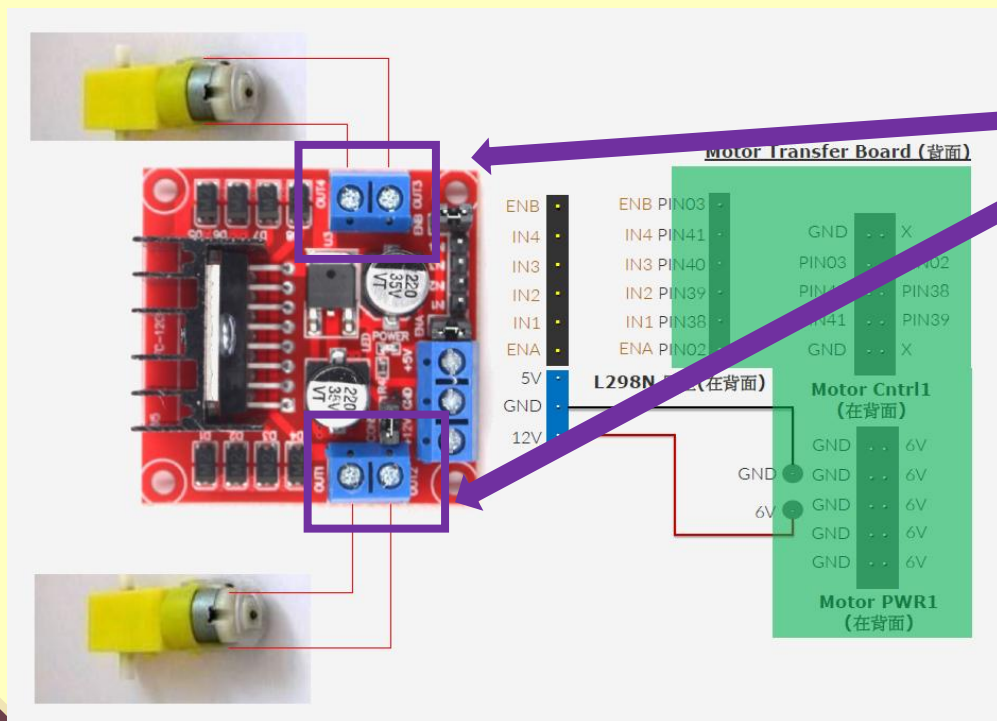
Arduino 語法

馬達腳位 -



Arduino 語法

馬達宣告、語法 –



```
const byte ENA=2;  
const byte ENB=3;  
const byte IN1=38;  
const byte IN2=39;  
const byte IN3=40;  
const byte IN4=41;
```

```
void setup() {  
    // put your setup code here, to run once:  
}
```

//這邊是啟動馬達1的部分，主要是控制馬達1的速率
//這邊是啟動馬達2的部分，主要是控制馬達2的速率
//控制馬達1的前進、後退、停止
//控制馬達1的前進、後退、停止
//控制馬達2的前進、後退、停止
//控制馬達2的前進、後退、停止

Arduino 語法

馬達-真值表(控制馬達轉向)

ENA	IN1	IN2	工作狀態
高	高	低	馬達正向轉動
高	低	高	馬達反向轉動
高	與IN2相同	與IN1相同	快速停止(緊急煞車)
低	任何狀態	任何狀態	停止(自由滑行)

ENB	IN3	IN4	工作狀態
高	高	低	馬達正向轉動
高	低	高	馬達反向轉動
高	與IN4相同	與IN3相同	快速停止(緊急煞車)
低	任何狀態	任何狀態	停止(自由滑行)



Arduino 語法

馬達宣告語法-前進



```
const byte ENA=2; //這邊是啟動馬達1的部分，主要是控制馬達1
const byte ENB=3; //這邊是啟動馬達2的部分，主要是控制馬達2
const byte IN1=38; //控制馬達1的前進、後退、停止
const byte IN2=39; //控制馬達1的前進、後退、停止
const byte IN3=40; //控制馬達2的前進、後退、停止
const byte IN4=41; //控制馬達2的前進、後退、停止

void setup() {
    // put your setup code here, to run once:
    pinMode(ENA, OUTPUT); //ENA設定輸出
    pinMode(ENB, OUTPUT); //ENB設定輸出
    pinMode(IN1, OUTPUT); //IN1設定輸出
    pinMode(IN2, OUTPUT); //IN2設定輸出
    pinMode(IN3, OUTPUT); //IN3設定輸出
    pinMode(IN4, OUTPUT); //IN4設定輸出
}

void loop() {
    // put your main code here, to run repeatedly:
    {analogWrite(ENA, 255); //ENA馬達速率設定為255最高扭力
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    delay(50);
    }
    {analogWrite(ENB, 255); //ENB馬達速率設定為255最高扭力
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    delay(50);
    }
}
```

Arduino 語法

馬達宣告語法 – 後退



```
const byte ENA=2;
const byte ENB=3;
const byte IN1=38;
const byte IN2=39;
const byte IN3=40;
const byte IN4=41;

void setup() {
    // put your setup code here, to run once:
    pinMode(ENA, OUTPUT);
    pinMode(ENB, OUTPUT);
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(IN3, OUTPUT);
    pinMode(IN4, OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    {analogWrite(ENA, 255);
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    delay(50);
    }
    {analogWrite(ENB, 255);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    delay(50);
    }
}
```

//這邊是啟動馬達1的部分，主要是控制馬達1的速率
 //這邊是啟動馬達2的部分，主要是控制馬達2的速率
 //控制馬達1的前進、後退、停止
 //控制馬達1的前進、後退、停止
 //控制馬達2的前進、後退、停止
 //控制馬達2的前進、後退、停止

//ENA設定輸出
 //ENB設定輸出
 //IN1設定輸出
 //IN2設定輸出
 //IN3設定輸出
 //IN4設定輸出

//ENA馬達速率設定為255最高扭力
 //ENB馬達速率設定為255最高扭力

Arduino 語法

馬達宣告語法 – 原地右轉



```
const byte ENA=2;
const byte ENB=3;
const byte IN1=38;
const byte IN2=39;
const byte IN3=40;
const byte IN4=41;
```

//這邊是啟動馬達1的部分，主要是控制馬達1的速率
//這邊是啟動馬達2的部分，主要是控制馬達2的速率
//控制馬達1的前進、後退、停止
//控制馬達1的前進、後退、停止
//控制馬達2的前進、後退、停止
//控制馬達2的前進、後退、停止

```
void setup() {
  // put your setup code here, to run once:
  pinMode(ENA, OUTPUT);
  pinMode(ENB, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
}
```

//ENA設定輸出
//ENB設定輸出
//IN1設定輸出
//IN2設定輸出
//IN3設定輸出
//IN4設定輸出

```
void loop() {
  // put your main code here, to run repeatedly:
  {analogWrite(ENA, 229);
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  delay(50);
  }
  {analogWrite(ENB, 255);
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  delay(50);
  }
}
```

//ENA馬達速率設定為229最高扭力

//ENB馬達速率設定為255最低扭力



Arduino 語法

馬達宣告語法 – 原地左轉



```
const byte ENA=2;
const byte ENB=3;
const byte IN1=38;
const byte IN2=39;
const byte IN3=40;
const byte IN4=41;

void setup() {
    // put your setup code here, to run once:
    pinMode(ENA, OUTPUT);
    pinMode(ENB, OUTPUT);
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(IN3, OUTPUT);
    pinMode(IN4, OUTPUT);
}

void loop() {
    // put your main code here, to run repeatedly:
    {analogWrite(ENA, 255);
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    delay(50);
    }
    {analogWrite(ENB, 229);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    delay(50);
    }
}
```

//這邊是啟動馬達1的部分，主要是控制馬達1的速率
 //這邊是啟動馬達2的部分，主要是控制馬達2的速率
 //控制馬達1的前進、後退、停止
 //控制馬達1的前進、後退、停止
 //控制馬達2的前進、後退、停止
 //控制馬達2的前進、後退、停止

//ENA設定輸出
 //ENB設定輸出
 //IN1設定輸出
 //IN2設定輸出
 //IN3設定輸出
 //IN4設定輸出

//ENA馬達速率設定為255最高扭力
 //ENB馬達速率設定為229最低扭力

Arduino 語法

馬達宣告語法 – 右轉



```
const byte ENA=2;
const byte ENB=3;
const byte IN1=38;
const byte IN2=39;
const byte IN3=40;
const byte IN4=41;

//這邊是啟動馬達1的部分，主要是控制馬達1的速率
//這邊是啟動馬達2的部分，主要是控制馬達2的速率
//控制馬達1的前進、後退、停止
//控制馬達1的前進、後退、停止
//控制馬達2的前進、後退、停止
//控制馬達2的前進、後退、停止

void setup() {
    // put your setup code here, to run once:
    pinMode(ENA, OUTPUT);
    pinMode(ENB, OUTPUT);
    pinMode(IN1, OUTPUT);
    pinMode(IN2, OUTPUT);
    pinMode(IN3, OUTPUT);
    pinMode(IN4, OUTPUT);

    //ENA設定輸出
    //ENB設定輸出
    //IN1設定輸出
    //IN2設定輸出
    //IN3設定輸出
    //IN4設定輸出
}

void loop() {
    // put your main code here, to run repeatedly:
    {analogWrite(ENA, 0);
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    delay(50);
    }
    {analogWrite(ENB, 255);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
    delay(50);
    }
    //ENA馬達速率設定為0最低扭力
    //ENB馬達速率設定為255最低扭力
}
```

Arduino 語法

馬達宣告語法 – 左轉



```
const byte ENA=2;
const byte ENB=3;
const byte IN1=38;
const byte IN2=39;
const byte IN3=40;
const byte IN4=41;
```

//這邊是啟動馬達1的部分，主要是控制馬達1的速率
 //這邊是啟動馬達2的部分，主要是控制馬達2的速率
 //控制馬達1的前進、後退、停止
 //控制馬達1的前進、後退、停止
 //控制馬達2的前進、後退、停止
 //控制馬達2的前進、後退、停止

```
void setup() {
    // put your setup code here, to run once:
    pinMode(ENA,OUTPUT);
    pinMode(ENB,OUTPUT);
    pinMode(IN1,OUTPUT);
    pinMode(IN2,OUTPUT);
    pinMode(IN3,OUTPUT);
    pinMode(IN4,OUTPUT);
}
```

//ENA設定輸出
 //ENB設定輸出
 //IN1設定輸出
 //IN2設定輸出
 //IN3設定輸出
 //IN4設定輸出

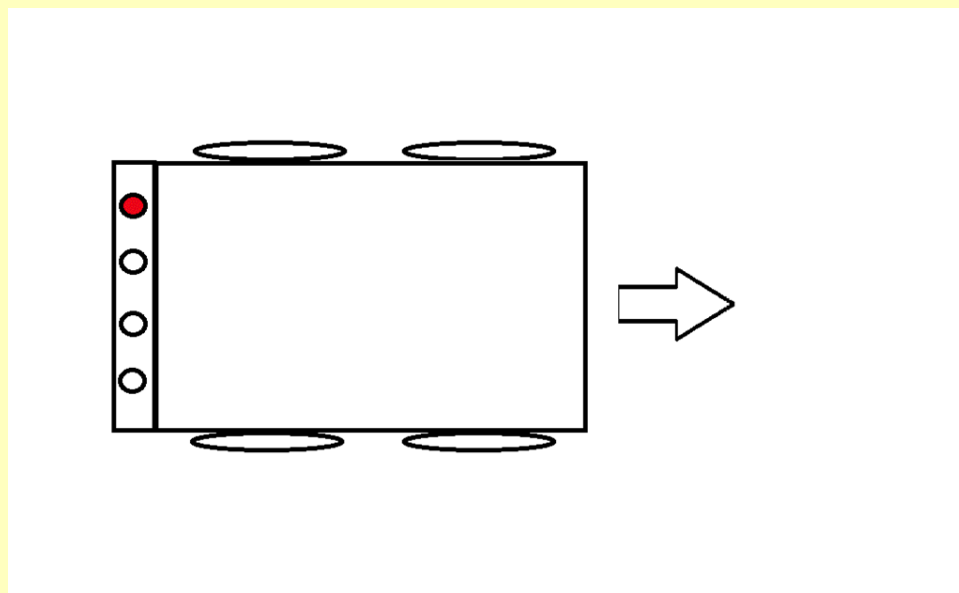
```
void loop() {
    // put your main code here, to run repeatedly:
    {analogWrite(ENA,255);
    digitalWrite(IN1,HIGH);
    digitalWrite(IN2,LOW);
    delay(50);
    }
    {analogWrite(ENB,0);
    digitalWrite(IN3,HIGH);
    digitalWrite(IN4,LOW);
    delay(50);
    }
}
```

//ENA馬達速率設定為255最高扭力
 //ENB馬達速率設定為0最低扭力

分組討論

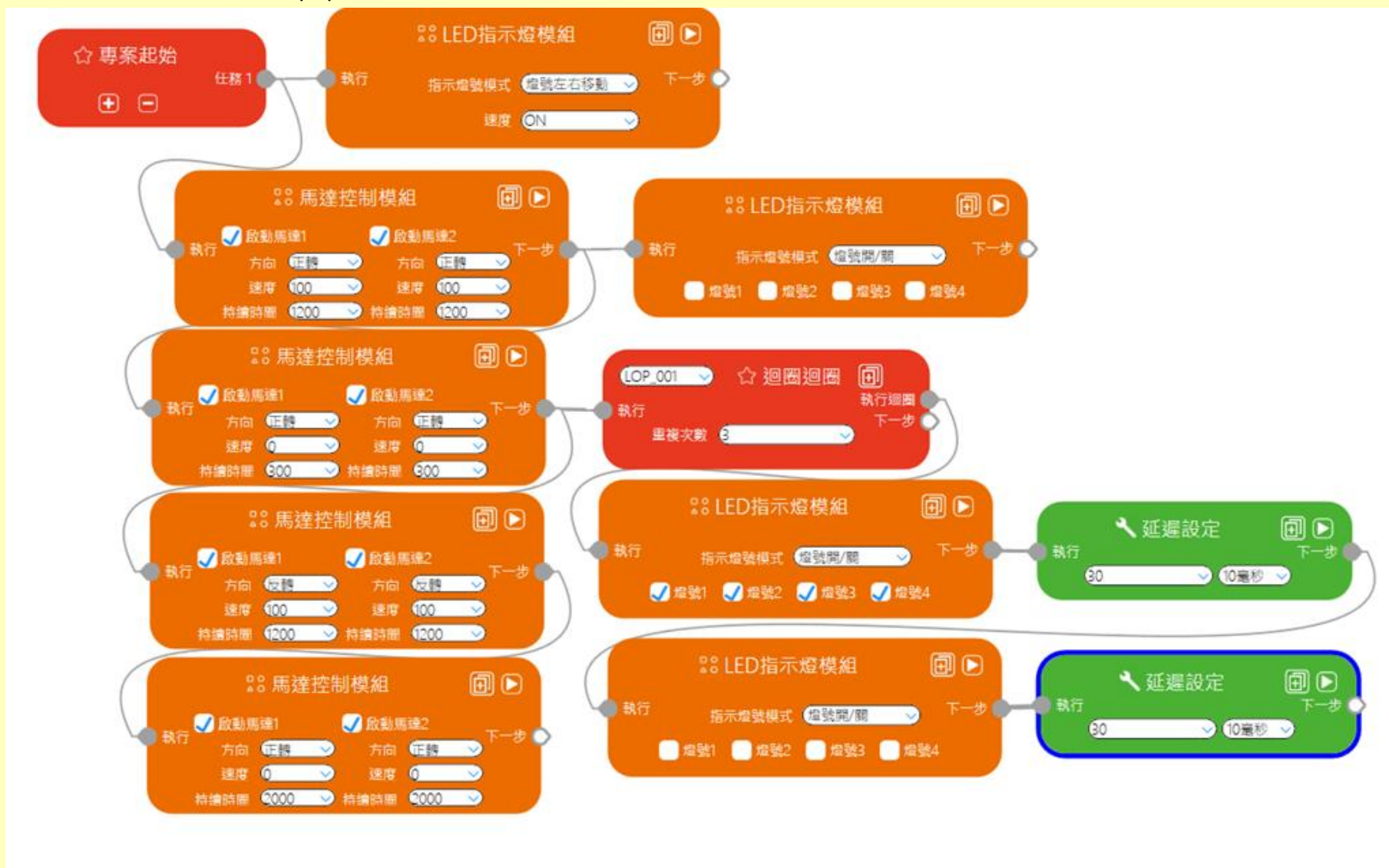
前進，LED會左右移動
後退，LED會閃爍

如何用Arduino做出這效果？



解答

Tarkus VP 畫面



解答

Arduino 畫面

宣告語法 – 全域

```
const int led[]={4,5,6,7};  
const byte ENA=2;  
const byte ENB=3;  
const byte IN1=38;  
const byte IN2=39;  
const byte IN3=40;  
const byte IN4=41;
```

//這邊是啟動馬達1的部分，主要是控制馬達1的速率

//這邊是啟動馬達2的部分，主要是控制馬達2的速率

//控制馬達1的前進、後退、停止

//控制馬達1的前進、後退、停止

//控制馬達2的前進、後退、停止

//控制馬達2的前進、後退、停止



解答

Arduino 畫面

宣告語法－函式庫

```
void led_move() {  
    for(int i=0;i<4;i++) {  
        digitalWrite(led[i],HIGH);  
        delay(100);  
        digitalWrite(led[i],LOW);  
    }  
    for(int i=4;i>=0;i--) {  
        digitalWrite(led[i],HIGH);  
        delay(100);  
        digitalWrite(led[i],LOW);  
    }  
} //LED燈左右移動  
  
void led_blink() {  
    for (int j=0;j<5;j++){  
        digitalWrite(led[0],HIGH);  
        digitalWrite(led[1],HIGH);  
        digitalWrite(led[2],HIGH);  
        digitalWrite(led[3],HIGH);  
        delay(100);  
  
        digitalWrite(led[0],LOW);  
        digitalWrite(led[1],LOW);  
        digitalWrite(led[2],LOW);  
        digitalWrite(led[3],LOW);  
        delay(100);  
    }  
} //LED燈閃爍
```

```
void gogo() {  
    analogWrite(ENA,255); //ENA馬達速率設定為255最高扭力  
    digitalWrite(IN1,HIGH);  
    digitalWrite(IN2,LOW);  
    analogWrite(ENB,255); //ENB馬達速率設定為255最高扭力  
    digitalWrite(IN3,HIGH);  
    digitalWrite(IN4,LOW);  
    delay(100);  
} //前進  
  
void back() {  
    analogWrite(ENA,255); //ENA馬達速率設定為255最高扭力  
    digitalWrite(IN1,LOW);  
    digitalWrite(IN2,HIGH);  
    analogWrite(ENB,255); //ENB馬達速率設定為255最高扭力  
    digitalWrite(IN3,LOW);  
    digitalWrite(IN4,HIGH);  
    delay(100);  
} //後退
```

解答

Arduino 畫面

宣告語法 – 設定 **setup()**

```
void setup() {  
    // put your setup code here, to run once:  
    pinMode(led[1], OUTPUT);           //這邊將LED1~4腳位都執行輸出的設定  
    pinMode(led[2], OUTPUT);  
    pinMode(led[3], OUTPUT);  
    pinMode(led[0], OUTPUT);  
    pinMode(ENA, OUTPUT);              //ENA設定輸出  
    pinMode(ENB, OUTPUT);              //ENB設定輸出  
    pinMode(IN1, OUTPUT);              //IN1設定輸出  
    pinMode(IN2, OUTPUT);              //IN2設定輸出  
    pinMode(IN3, OUTPUT);              //IN3設定輸出  
    pinMode(IN4, OUTPUT);              //IN4設定輸出  
}
```



解答

Arduino 畫面

宣告語法 - 主程式 **loop()**

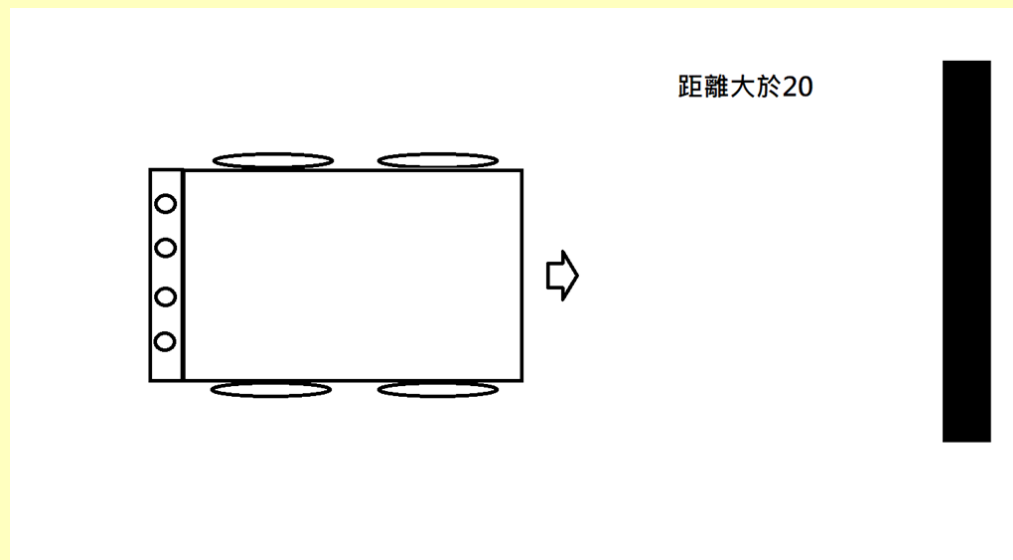
```
void loop() {  
    // put your main code here, to run repeatedly:  
    gogo();  
    led_move();  
    delay(500);  
    back();  
    led_blink();  
    delay(500);  
  
}
```



分組討論

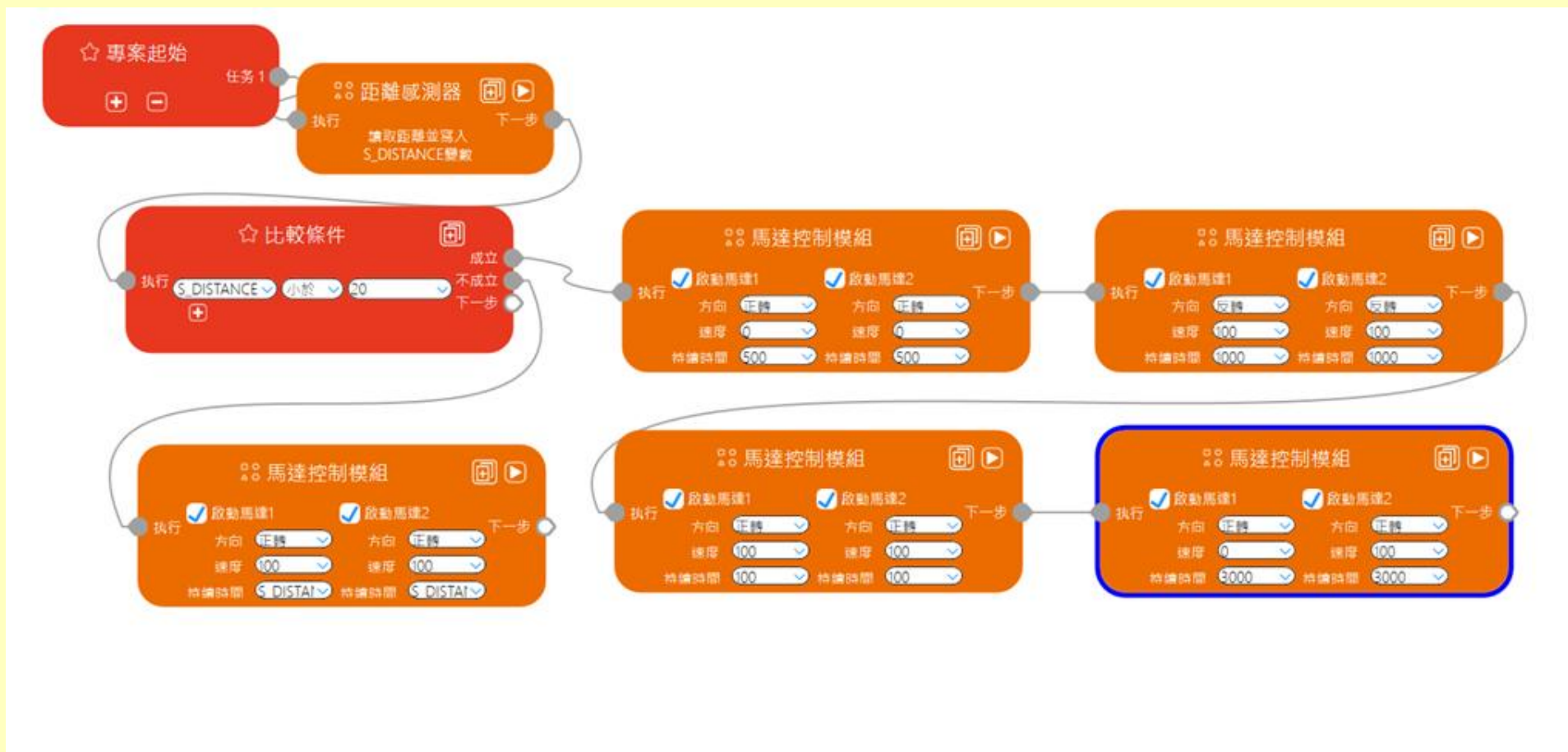
自動避障

1. 車子距離障礙物小於20時，會停車
2. 車子會倒退，再前進右轉彎



解答

Tarkus VP 畫面



解答

Arduino 畫面

宣告語法 – 全域

```
const int TrigPin=35;           //發射為35PIN腳
const int EchoPin=34;           //接收為34PIN腳
int duration=0;
int S_DISTANCE;

const byte ENA=2;                //這邊是啟動馬達1的部分，主要是控制馬達1的速率
const byte ENB=3;                //這邊是啟動馬達2的部分，主要是控制馬達2的速率
const byte IN1=38;               //控制馬達1的前進、後退、停止
const byte IN2=39;               //控制馬達1的前進、後退、停止
const byte IN3=40;               //控制馬達2的前進、後退、停止
const byte IN4=41;               //控制馬達2的前進、後退、停止
```



解答

Arduino 畫面

宣告語法－函式庫

```
void distance() {                                //傳回值型態為距離
  digitalWrite(TrigPin,LOW);
  delayMicroseconds(2);
  digitalWrite(TrigPin,HIGH);
  delayMicroseconds(10);
  digitalWrite(TrigPin,LOW);
  duration = pulseIn (EchoPin,HIGH);
  S_DISTANCE=duration*0.034/2;                    //計算距離：時間乘上聲音在空氣中傳送的速度(0.034)再除以 2。
  delay(100);
}

void cease() {
  analogWrite (ENA,255);                          //ENA馬達速率設定為255最高扭力
  digitalWrite (IN1,HIGH);
  digitalWrite (IN2,HIGH);
  analogWrite (ENB,255);                          //ENB馬達速率設定為255最高扭力
  digitalWrite (IN3,HIGH);
  digitalWrite (IN4,HIGH);
  delay(100);
}                                                  //緊急停止

void right() {
  analogWrite (ENA,0);                            //ENA馬達速率設定為0最低扭力
  digitalWrite (IN1,HIGH);
  digitalWrite (IN2,LOW);
  analogWrite (ENB,255);                          //ENB馬達速率設定為255最高扭力
  digitalWrite (IN3,HIGH);
  digitalWrite (IN4,LOW);
  delay(100);
}                                                  //右轉
```

```
void gogo() {
  analogWrite (ENA,255);                          //ENA馬達速率設定為255最高扭力
  digitalWrite (IN1,HIGH);
  digitalWrite (IN2,LOW);
  analogWrite (ENB,255);                          //ENB馬達速率設定為255最高扭力
  digitalWrite (IN3,HIGH);
  digitalWrite (IN4,LOW);
  delay(100);
}                                                  //前進
```

解答

Arduino 畫面

宣告語法 – 設定 **setup()**

```
void setup() {  
  // put your setup code here, to run once:  
  pinMode(TrigPin, OUTPUT);           //把發射的腳位設定輸出  
  pinMode(EchoPin, INPUT);            //把接收腳位設定輸入  
  Serial.begin(9600);  
  pinMode(ENA, OUTPUT);                //ENA設定輸出  
  pinMode(ENB, OUTPUT);                //ENB設定輸出  
  pinMode(IN1, OUTPUT);                //IN1設定輸出  
  pinMode(IN2, OUTPUT);                //IN2設定輸出  
  pinMode(IN3, OUTPUT);                //IN3設定輸出  
  pinMode(IN4, OUTPUT);                //IN4設定輸出  
}
```

解答

Arduino 畫面

宣告語法 - 主程式 **loop()**

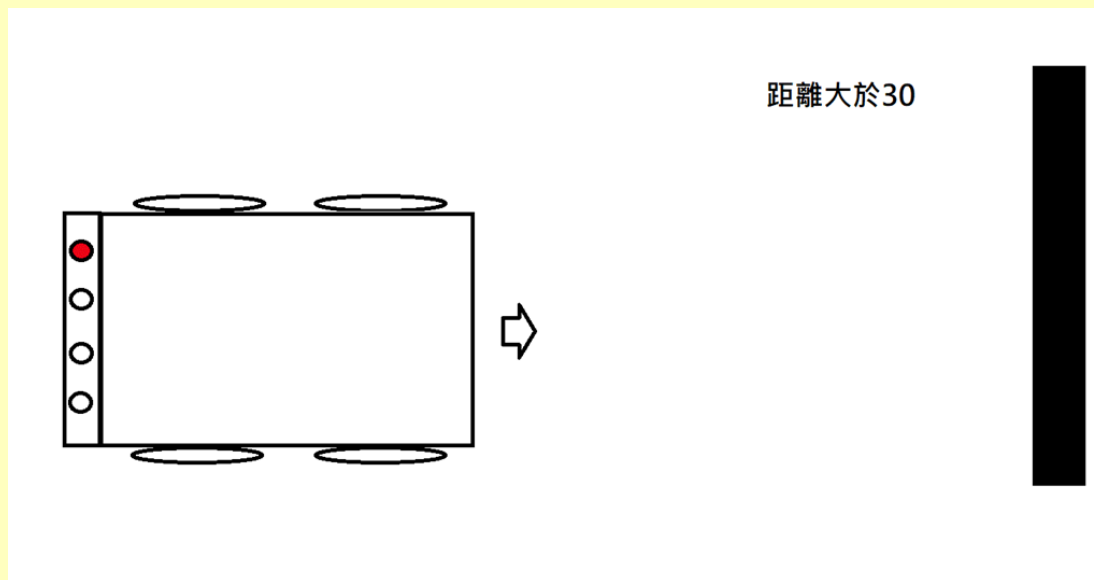
```
void loop() {  
  // put your main code here, to run repeatedly:  
  distance();  
  if(S_DISTANCE<20) {  
    cease();  
    delay(500);  
    right();  
    delay(500);  
  }  
  else {  
    gogo();  
    delay(500);  
  }  
}
```



分組討論

自動避障+燈

1. 車子距離障礙物大於30時，會前進+燈號左右移動
2. 車子距離障礙物小於30時，會停車閃爍
3. 再倒車，打左轉燈轉彎



解答

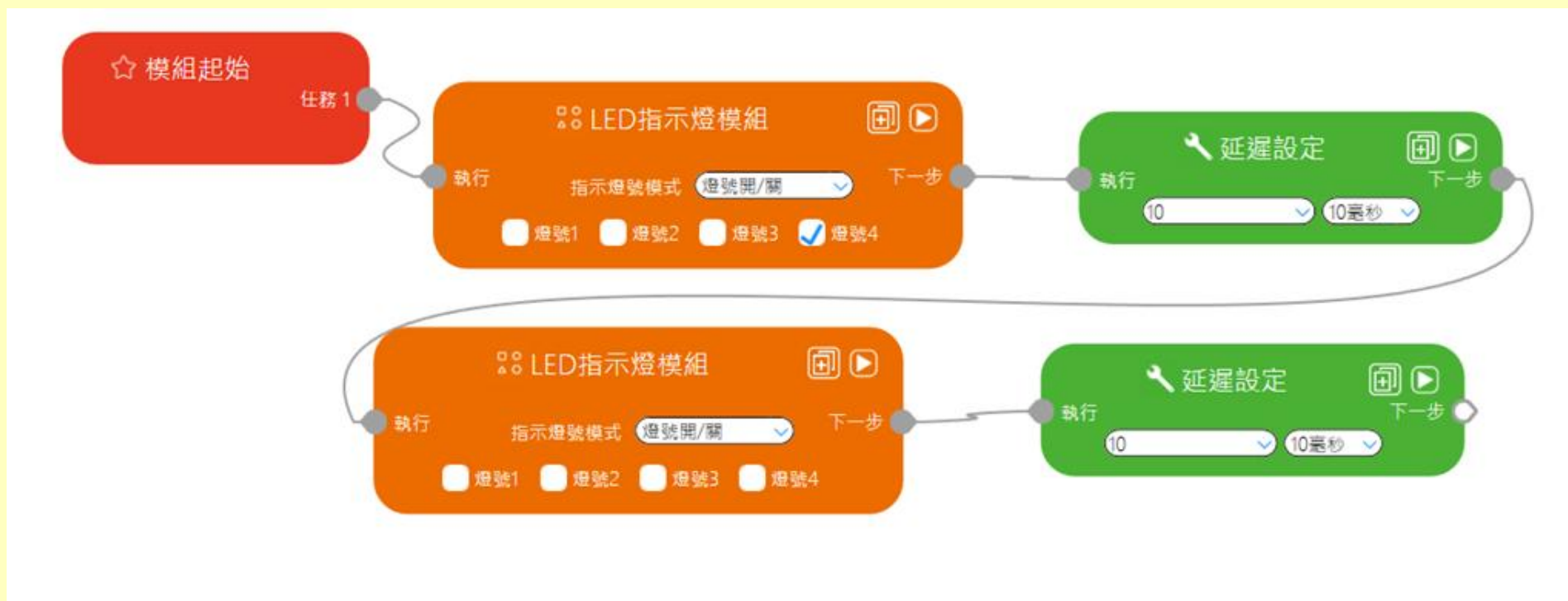
Tarkus VP 畫面



解答

Tarkus VP 畫

面



解答

Arduino 畫面

宣告語法 – 全域

```
const int led[]={4,5,6,7};  
const int TrigPin=35;  
const int EchoPin=34;  
int duration=0;  
int S_DISTANCE;  
const byte ENA=2;  
const byte ENB=3;  
const byte IN1=38;  
const byte IN2=39;  
const byte IN3=40;  
const byte IN4=41;
```

//宣告led的腳位
//發射為35PIN腳
//接收為34PIN腳

//這邊是啟動馬達1的部分，主要是控制馬達1的速率
//這邊是啟動馬達2的部分，主要是控制馬達2的速率
//控制馬達1的前進、後退、停止
//控制馬達1的前進、後退、停止
//控制馬達2的前進、後退、停止
//控制馬達2的前進、後退、停止



解答

Arduino 畫面

宣告語法－函式庫

```
void distance() {  
    digitalWrite(TrigPin, LOW);  
    delayMicroseconds(2);  
    digitalWrite(TrigPin, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(TrigPin, LOW);  
    duration = pulseIn(EchoPin, HIGH);  
    S_DISTANCE = duration * 0.034 / 2;  
    delay(100);  
}  
  
void cease() {  
    analogWrite(ENA, 255);  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, HIGH);  
    analogWrite(ENB, 255);  
    digitalWrite(IN3, HIGH);  
    digitalWrite(IN4, HIGH);  
    delay(100);  
}
```

//傳回值型態為距離

//計算距離：時間乘上聲音在空氣中傳送的速度(0.034)再除以 2。

//ENA馬達速率設定為255最高扭力

//ENB馬達速率設定為255最高扭力

//緊急停止

解答

Arduino 畫面

宣告語法－函式庫

```
void led_blink(){
  for (int j=0;j<5;j++){
    digitalWrite(led[0],HIGH);
    digitalWrite(led[1],HIGH);
    digitalWrite(led[2],HIGH);
    digitalWrite(led[3],HIGH);
    delay(100);
    digitalWrite(led[0],LOW);
    digitalWrite(led[1],LOW);
    digitalWrite(led[2],LOW);
    digitalWrite(led[3],LOW);
    delay(100);
  }
}                                     //LED燈閃爍

void left(){
  analogWrite(ENA,255);               //ENA馬達速率設定為255最高扭力
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
  analogWrite(ENB,0);                 //ENB馬達速率設定為0最低扭力
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);
  delay(100);
}                                     //左轉
```

```
void led_left(){
  for (int j=0;j<5;j++){
    digitalWrite(led[0],HIGH);
    digitalWrite(led[1],LOW);
    digitalWrite(led[2],LOW);
    digitalWrite(led[3],LOW);
    delay(100);
    digitalWrite(led[0],LOW);
    digitalWrite(led[1],LOW);
    digitalWrite(led[2],LOW);
    digitalWrite(led[3],LOW);
    delay(100);
  }
}                                     //LED左轉燈

void gogo(){
  analogWrite(ENA,255);               //ENA馬達速率設定為255最高扭力
  digitalWrite(IN1,HIGH);
  digitalWrite(IN2,LOW);
  analogWrite(ENB,255);               //ENB馬達速率設定為255最高扭力
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);
  delay(100);
}                                     //前進
```

解答

Arduino 畫面

宣告語法－函式庫

```
void back() {  
    analogWrite(ENA, 255); //ENA馬達速率設定為255最高扭力  
    digitalWrite(IN1, LOW);  
    digitalWrite(IN2, HIGH);  
    analogWrite(ENB, 255); //ENB馬達速率設定為255最高扭力  
    digitalWrite(IN3, LOW);  
    digitalWrite(IN4, HIGH);  
    delay(100);  
}  
//倒退  
void led_move() {  
    for(int i=0; i<4; i++) {  
        digitalWrite(led[i], HIGH);  
        delay(100);  
        digitalWrite(led[i], LOW);  
    }  
    for(int i=4; i>=0; i--) {  
        digitalWrite(led[i], HIGH);  
        delay(100);  
        digitalWrite(led[i], LOW);  
    }  
    //LED燈左右移動  
}
```



解答

Arduino 畫面

宣告語法－設定setup()

```
void setup() {  
  // put your setup code here, to run once:  
  pinMode(led[0], OUTPUT);  
  pinMode(led[1], OUTPUT);  
  pinMode(led[2], OUTPUT);  
  pinMode(led[3], OUTPUT);  
  pinMode(TrigPin, OUTPUT);           //把發射的腳位設定輸出  
  pinMode(EchoPin, INPUT);           //把接收腳位設定輸入  
  Serial.begin(9600);  
  pinMode(ENA, OUTPUT);              //ENA設定輸出  
  pinMode(ENB, OUTPUT);              //ENB設定輸出  
  pinMode(IN1, OUTPUT);              //IN1設定輸出  
  pinMode(IN2, OUTPUT);              //IN2設定輸出  
  pinMode(IN3, OUTPUT);              //IN3設定輸出  
  pinMode(IN4, OUTPUT);              //IN4設定輸出  
}
```



解答

Arduino 畫面

宣告語法 - 主程式 **loop()**

```
void loop() {  
  
    distance();  
    if (S_DISTANCE < 30) {  
        cease();  
        led_blink();  
        delay(500);  
        back();  
        delay(500);  
        left();  
        led_left();  
        delay(500);  
    }  
    else {  
        gogo();  
        led_move();  
        delay(500);  
    }  
}
```

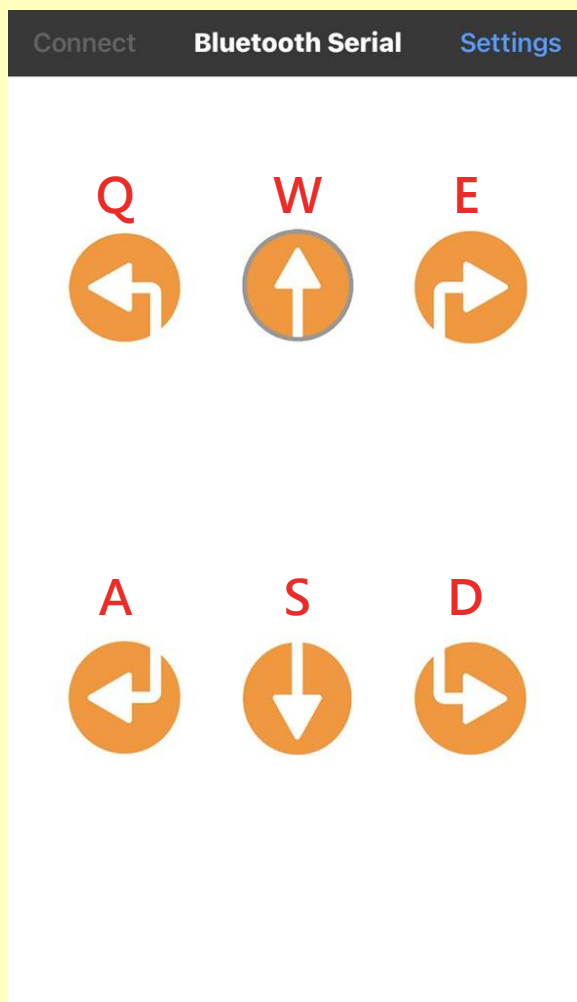


藍芽操控

藍芽手機設定

IOS設定-

你要讓手機能控制
你只能設定**QWEASD**
這幾個英文字
然後一定要大寫



安卓設定-

手機的控制設定可以自行設定
但你設定的內容一定
要跟你條件敘述的設定是一樣的



Arduino 語法

藍芽讀取 (Serial3.read()) – Tarkus VP 畫

面



Arduino 語法

藍芽讀取 (Serial3.read()) –

Serial 3 腳位在 Mega 2560 為 14、15，不需額外多做宣告，可以直接下讀取命令即可。

```
void setup() {  
  // put your setup code here, to run once:  
  Serial3.begin(57600);  
  Serial.begin(9600);  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  char cmd;  
  if(Serial3.available()){  
    cmd=Serial3.read();  
    Serial.println(cmd);  
    delay(50);  
  }  
}
```

//Serial3這邊是L3,不是13喔;它主要是指板子對外通訊藍芽的序列埠;這裡的begin是開啟藍芽連接的腳位,值為57600
//Serial這邊是指我們藍色這條傳輸線;這裡的begin是開啟他的序列埠,值為9600

//接收到的字元指令
//一旦接收到來自Serial3的字元訊號
//把讀取到Serial3的字元訊號放進cmd
//在視窗裡顯示cmd所代表的字元
//延遲個50毫秒

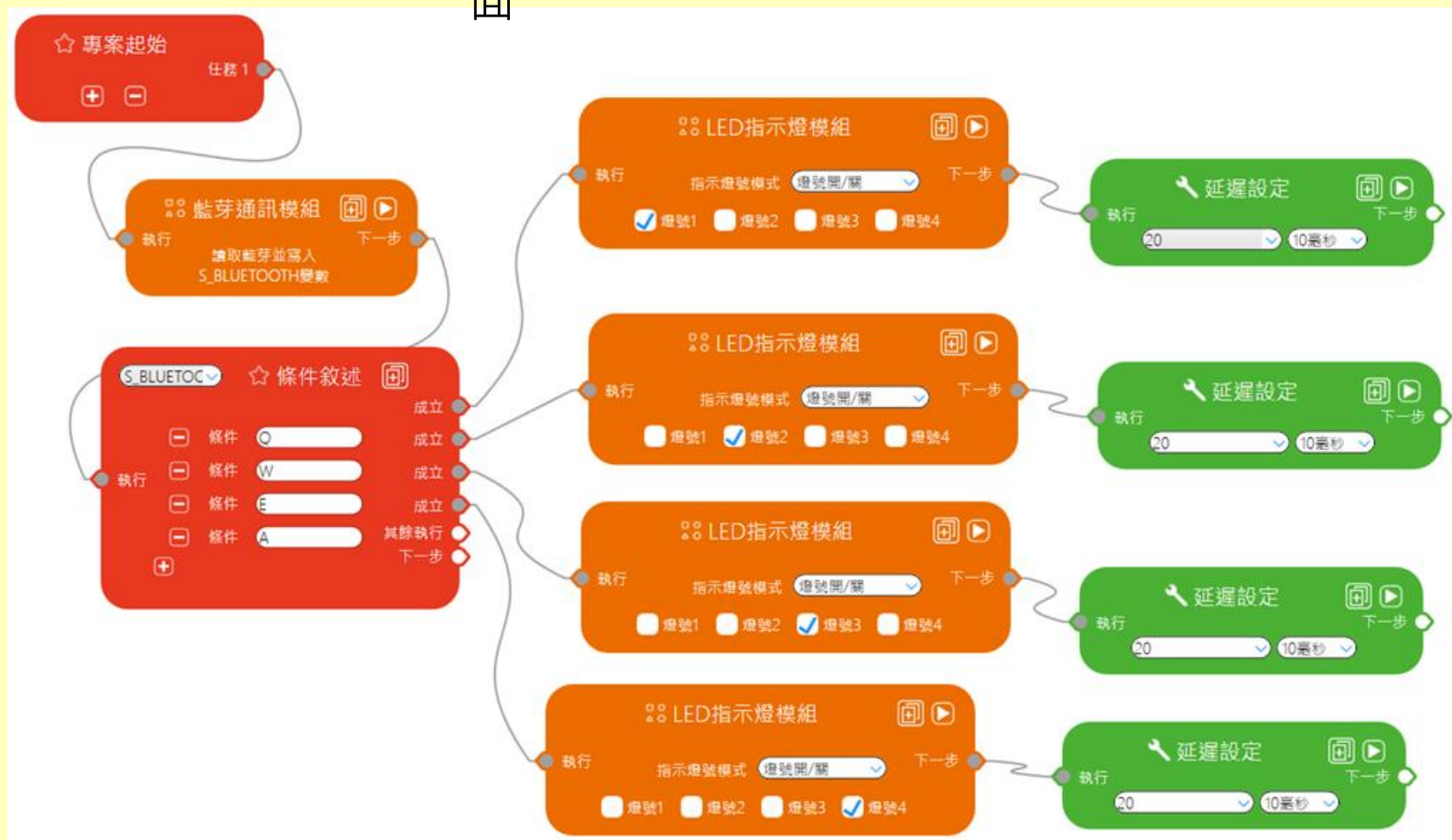


Arduino 語法

藍芽控制(switch - case) –

Tarkus VP 畫

面



Arduino 語法

藍芽控制(switch - case) –

```
void setup() {  
    // put your setup code here, to run once:  
    Serial3.begin(57600);  
    Serial.begin(9600);  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
    char cmd;  
    if(Serial3.available()){  
        cmd=Serial3.read();  
        Serial.println(cmd);  
        delay(50);  
    }  
}
```

//Serial3這邊是L3,不是13喔;它主要是指板子對外通訊藍芽的序列埠;這裡的begin是開啟藍芽連接的腳位,值為57600
//Serial這邊是指我們藍色這條傳輸線;這裡的begin是開啟他的序列埠,值為9600

//接收到的字元指令
//一旦接收到來自Serial3的字元訊號
//把讀取到Serial3的字元訊號放進cmd
//在視窗裡顯示cmd所代表的字元
//延遲個50毫秒



Arduino 語法

藍芽控制(switch - case) –

```
void loop() {
    // put your main code here, to run repeatedly:
    char cmd;
    if(Serial3.available()){
        cmd=Serial3.read();
        Serial.println(cmd);
        delay(50);
        switch(cmd){
            case 'Q':{
                digitalWrite(led[0],HIGH);
                delay(200);}
            break;
            case 'W':{
                digitalWrite(led[1],HIGH);
                delay(200);}
            break;
            case 'E':{
                digitalWrite(led[2],HIGH);
                delay(200);}
            break;
            case 'A':{
                digitalWrite(led[3],HIGH);
                delay(200);}
            break;
        }
        digitalWrite(led[0],LOW);
        digitalWrite(led[1],LOW);
        digitalWrite(led[2],LOW);
        digitalWrite(led[3],LOW);

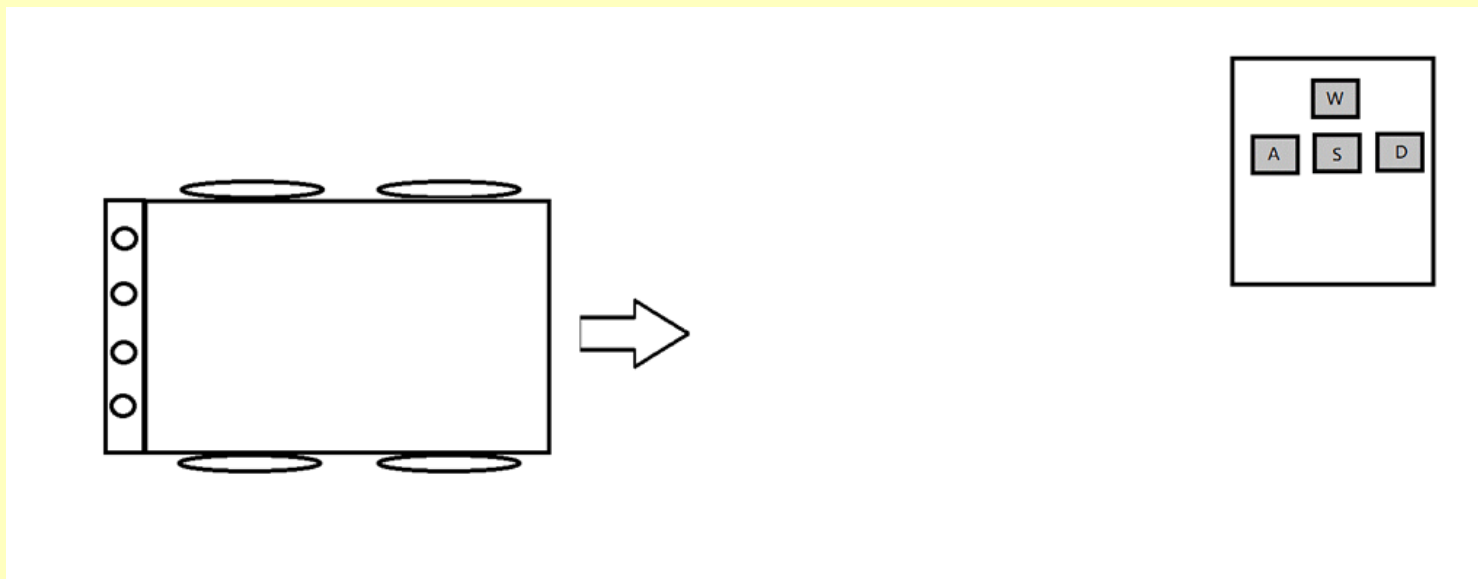
    }
}
```

//接收到的字元指令
 //一旦接收到來自Serial3的字元訊號
 //把讀取到Serial3的字元訊號放進cmd
 //在視窗裡顯示cmd所代表的字元
 //延遲個50毫秒
 //switch條件敘述
 //條件Q，這邊的''是指字元(單一個字母)的括弧；" "是指字串(一串英文字)的括弧
 //1號燈亮
 //延續200毫秒
 //跳出這個switch條件敘述
 //條件W，這邊的''是指字元(單一個字母)的括弧；" "是指字串(一串英文字)的括弧
 //2號燈亮
 //延續200毫秒
 //跳出這個switch條件敘述
 //條件E，這邊的''是指字元(單一個字母)的括弧；" "是指字串(一串英文字)的括弧
 //3號燈亮
 //延續200毫秒
 //跳出這個switch條件敘述
 //條件A，這邊的''是指字元(單一個字母)的括弧；" "是指字串(一串英文字)的括弧
 //4號燈亮
 //延續200毫秒
 //跳出這個switch條件敘述
 //不管你執行完哪個條件後，跳出switch條件敘述時，燈就會自動全關



分組討論

- 1.當藍芽送一個訊號為W時，車子會前進
- 2.當藍芽送一個訊號為S時，車子會後退
- 3.當藍芽沒送任何訊號時，車子會停止





解答

Arduino 畫面

```
const byte ENA=2;           //這邊是啟動馬達1的部分，主要是控制馬達1的速率
const byte ENB=3;           //這邊是啟動馬達2的部分，主要是控制馬達2的速率
const byte IN1=38;          //控制馬達1的前進、後退、停止
const byte IN2=39;          //控制馬達1的前進、後退、停止
const byte IN3=40;          //控制馬達2的前進、後退、停止
const byte IN4=41;          //控制馬達2的前進、後退、停止

void setup() {
  // put your setup code here, to run once:
  Serial3.begin(57600);      //Serial3這邊是L3,不是13喔;它主要是指板子對外通訊藍芽的序列埠;這裡的begin是開啟藍芽連接的腳位,值為57600
  Serial.begin(9600);        //Serial這邊是指我們藍色這條傳輸線;這裡的begin是開啟他的序列埠,值為9600
  pinMode(ENA,OUTPUT);       //ENA設定輸出
  pinMode(ENB,OUTPUT);       //ENB設定輸出
  pinMode(IN1,OUTPUT);       //IN1設定輸出
  pinMode(IN2,OUTPUT);       //IN2設定輸出
  pinMode(IN3,OUTPUT);       //IN3設定輸出
  pinMode(IN4,OUTPUT);       //IN4設定輸出
}
```

解答

Arduino 畫面

```
void loop() {  
    // put your main code here, to run repeatedly:  
    char cmd;  
    if(Serial3.available()) {  
        cmd=Serial3.read();  
        Serial.println(cmd);  
        delay(40);  
        switch(cmd) {  
            case 'W': {  
                analogWrite(ENA, 255);  
                digitalWrite(IN1, HIGH);  
                digitalWrite(IN2, LOW);  
                analogWrite(ENB, 255);  
                digitalWrite(IN3, HIGH);  
                digitalWrite(IN4, LOW);  
                delay(500);  
            }  
            break;  
            case 'S': {  
                analogWrite(ENA, 255);  
                digitalWrite(IN1, LOW);  
                digitalWrite(IN2, HIGH);  
                analogWrite(ENB, 255);  
                digitalWrite(IN3, LOW);  
                digitalWrite(IN4, HIGH);  
                delay(500);  
            }  
            break;  
        }  
    }  
}
```

//接收到的字元指令
//一旦接收到來自Serial3的字元訊號
//把讀取到Serial3的字元訊號放進cmd
//在視窗裡顯示cmd所代表的字元
//延遲個40毫秒
//switch條件敘述
//條件W，控制馬達前進
//ENA馬達速率設定為255最高扭力

//ENB馬達速率設定為255最高扭力

//前進
//跳出這個switch條件敘述
//條件S，控制馬達後退
//ENA馬達速率設定為255最高扭力

//ENB馬達速率設定為255最高扭力

//後退
//跳出這個switch條件敘述

解答

Arduino 畫面

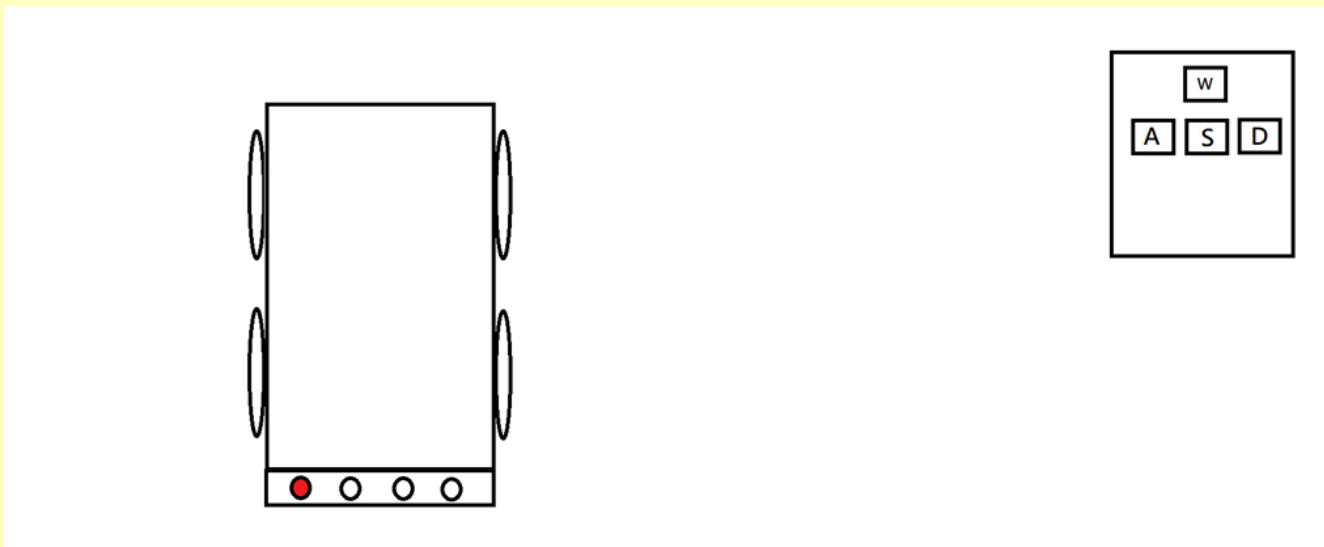
```
{  
  analogWrite (ENA, 255);           //ENA馬達速率設定為255最高扭力  
  digitalWrite (IN1, HIGH);  
  digitalWrite (IN2, HIGH);  
  analogWrite (ENB, 255);           //ENB馬達速率設定為255最高扭力  
  digitalWrite (IN3, HIGH);  
  digitalWrite (IN4, HIGH);  
  delay (500);  
}  
}  
}
```

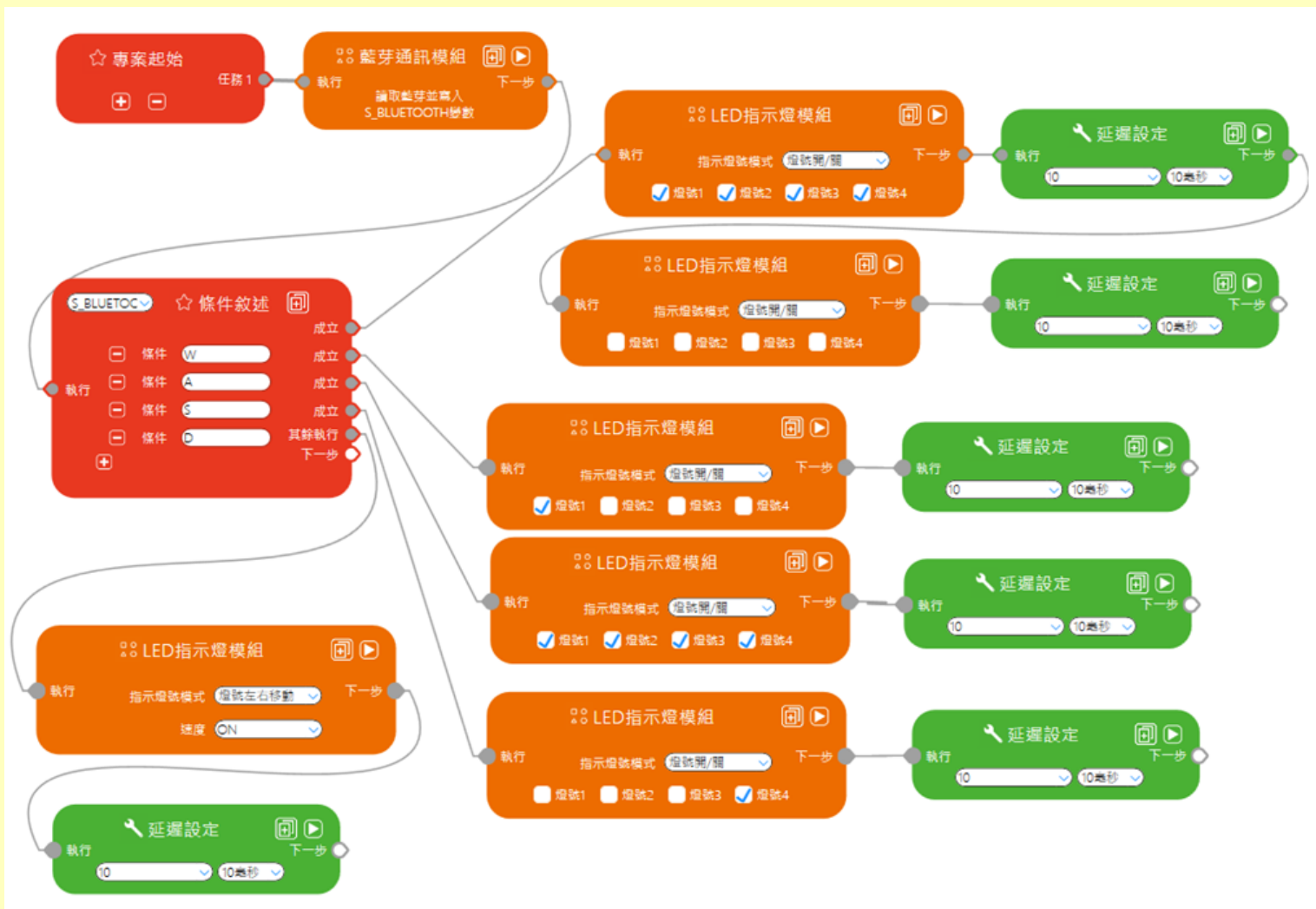


分組討論

1. 沒訊號時，車子燈號會左右移動
2. 當藍芽訊號為W時，車子會閃爍
3. 藍芽訊號為D時，訊號會打右轉燈

4. 當藍芽訊號為A時，燈號會打左轉燈
5. 當藍芽訊號為S時，車子會全亮





解答

Arduino 畫面

```
int led[]={4,5,6,7};

void led_bright(){
    { digitalWrite(led[0],HIGH);
      digitalWrite(led[1],HIGH);
      digitalWrite(led[2],HIGH);
      digitalWrite(led[3],HIGH);
      delay(1000);
    }
}

void led_blink(){
    for (int j=0;j<5;j++){
        digitalWrite(led[0],HIGH);
        digitalWrite(led[1],HIGH);
        digitalWrite(led[2],HIGH);
        digitalWrite(led[3],HIGH);
        delay(100);
        digitalWrite(led[0],LOW);
        digitalWrite(led[1],LOW);
        digitalWrite(led[2],LOW);
        digitalWrite(led[3],LOW);
        delay(100);
    }
}
```

//LED燈左右移動

//LED燈閃爍



解答

Arduino 畫面

```
void led_left() {  
  for (int j=0;j<5;j++){  
    digitalWrite(led[0],HIGH);  
    digitalWrite(led[1],LOW);  
    digitalWrite(led[2],LOW);  
    digitalWrite(led[3],LOW);  
    delay(100);  
  
    digitalWrite(led[0],LOW);  
    digitalWrite(led[1],LOW);  
    digitalWrite(led[2],LOW);  
    digitalWrite(led[3],LOW);  
    delay(100);  
  }  
} //LED左轉燈  
  
void led_right() {  
  for (int j=0;j<5;j++){  
    digitalWrite(led[0],LOW);  
    digitalWrite(led[1],LOW);  
    digitalWrite(led[2],LOW);  
    digitalWrite(led[3],HIGH);  
    delay(100);  
  
    digitalWrite(led[0],LOW);  
    digitalWrite(led[1],LOW);  
    digitalWrite(led[2],LOW);  
    digitalWrite(led[3],LOW);  
    delay(100);  
  }  
} //LED右轉燈
```



解答

Arduino 畫面

```
void setup() {  
  // put your setup code here, to run once:  
  Serial3.begin(57600);  
  Serial.begin(9600);  
  pinMode(led[0], OUTPUT);  
  pinMode(led[1], OUTPUT);  
  pinMode(led[2], OUTPUT);  
  pinMode(led[3], OUTPUT);  
}
```

//Serial3這邊是L3,不是13喔;它主要是指板子對外通訊藍芽的序列埠;這裡的begin是開啟藍芽連接的腳位,值為57600
//Serial這邊是指我們藍色這條傳輸線;這裡的begin是開啟他的序列埠,值為9600
//led0~4設定輸出



解答

Arduino 畫面

```

void loop() {
    // put your main code here, to run repeatedly:
    char cmd;
    if(Serial3.available()) {
        cmd=Serial3.read();
        Serial.println(cmd);
        delay(40);
        switch(cmd) {
            case 'W': {
                led_blink();
            }
            break;
            case 'D': {
                led_right();
            }
            break;
            case 'A': {
                led_left();
            }
            break;

            case 'S': {
                led_bright();
            }
            break;
        }
        {digitalWrite(led[0],LOW);
        digitalWrite(led[1],LOW);
        digitalWrite(led[2],LOW);
        digitalWrite(led[3],LOW);

        }
    }
}

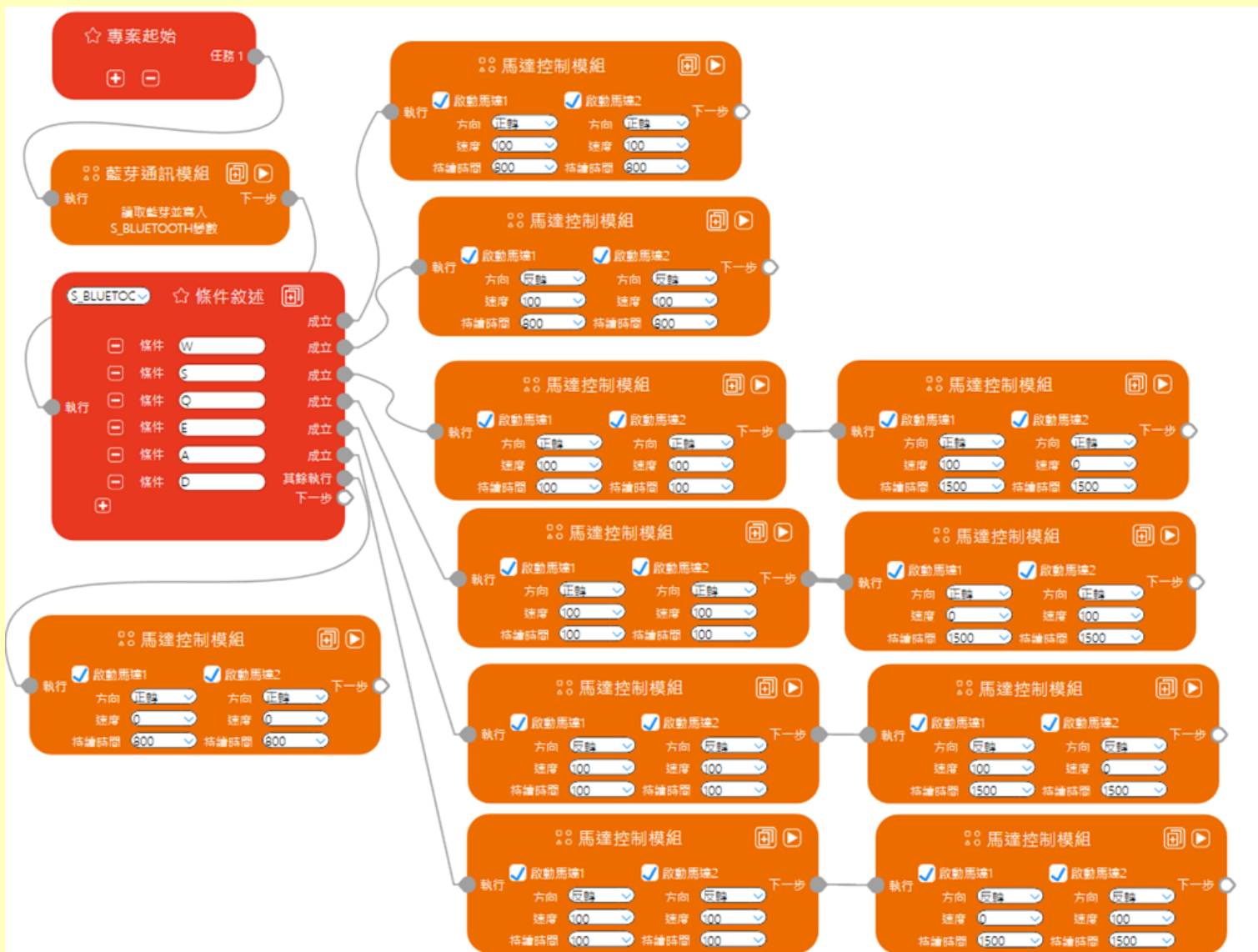
```

//接收到的字元指令
 //一旦接收到來自Serial3的字元訊號
 //把讀取到Serial3的字元訊號放進cmd
 //在視窗裡顯示cmd所代表的字元
 //延遲個40毫秒
 //switch條件敘述
 //條件W，控制LED燈
 //閃爍
 //跳出這個switch條件敘述
 //條件D，控制LED燈
 //右轉燈
 //跳出這個switch條件敘述
 //條件A，控制LED燈
 //左轉燈
 //跳出這個switch條件敘述
 //條件S，控制LED燈
 //亮
 //跳出這個switch條件敘述
 //不管你執行完哪個條件後，跳出switch條件敘述時，燈就會自動全關
 //不管執行哪個條件，跳出switch條件敘述時，車子都會緊急停止

解答

Tarkus VP 畫面

面



解答

Arduino 畫面

```
const byte ENA=2;           //這邊是啟動馬達1的部分，主要是控制馬達1的速率
const byte ENB=3;           //這邊是啟動馬達2的部分，主要是控制馬達2的速率
const byte IN1=38;          //控制馬達1的前進、後退、停止
const byte IN2=39;          //控制馬達1的前進、後退、停止
const byte IN3=40;          //控制馬達2的前進、後退、停止
const byte IN4=41;          //控制馬達2的前進、後退、停止

void setup() {
  // put your setup code here, to run once:
  Serial3.begin(57600);      //Serial3這邊是L3,不是13喔;它主要是指板子對外通訊藍芽的序列埠;這裡的begin是開啟藍芽連接的腳位,值為57600
  Serial.begin(9600);        //Serial這邊是指我們藍色這條傳輸線;這裡的begin是開啟他的序列埠,值為9600
  pinMode(ENA, OUTPUT);      //ENA設定輸出
  pinMode(ENB, OUTPUT);      //ENB設定輸出
  pinMode(IN1, OUTPUT);      //IN1設定輸出
  pinMode(IN2, OUTPUT);      //IN2設定輸出
  pinMode(IN3, OUTPUT);      //IN3設定輸出
  pinMode(IN4, OUTPUT);      //IN4設定輸出
}
```



專題思考

足球大賽



解答

Arduino 畫面

```
void loop() {  
    // put your main code here, to run repeatedly:  
    char cmd;  
    if (Serial3.available()) {  
        cmd = Serial3.read();  
        Serial.println(cmd);  
        delay(40);  
        switch (cmd) {  
            case 'W': {  
                analogWrite(ENA, 255);  
                digitalWrite(IN1, HIGH);  
                digitalWrite(IN2, LOW);  
                analogWrite(ENB, 255);  
                digitalWrite(IN3, HIGH);  
                digitalWrite(IN4, LOW);  
                delay(500);  
            }  
            break;  
            case 'S': {  
                analogWrite(ENA, 255);  
                digitalWrite(IN1, LOW);  
                digitalWrite(IN2, HIGH);  
                analogWrite(ENB, 255);  
                digitalWrite(IN3, LOW);  
                digitalWrite(IN4, HIGH);  
                delay(500);  
            }  
            break;  
        }  
    }  
}
```

//接收到的字元指令
//一旦接收到來自Serial3的字元訊號
//把讀取到Serial3的字元訊號放進cmd
//在視窗裡顯示cmd所代表的字元
//延遲個40毫秒
//switch條件敘述
//條件W，控制馬達前進
//ENA馬達速率設定為255最高扭力

//ENB馬達速率設定為255最高扭力

//前進
//跳出這個switch條件敘述
//條件S，控制馬達後退
//ENA馬達速率設定為255最高扭力

//ENB馬達速率設定為255最高扭力

//後退
//跳出這個switch條件敘述



解答

Arduino 畫面

```
case 'Q':{  
    analogWrite (ENA, 0);  
    digitalWrite (IN1, HIGH);  
    digitalWrite (IN2, LOW);  
    analogWrite (ENB, 255);  
    digitalWrite (IN3, HIGH);  
    digitalWrite (IN4, LOW);  
    delay (500);  
}  
break;  
case 'E':{  
    analogWrite (ENA, 255);  
    digitalWrite (IN1, HIGH);  
    digitalWrite (IN2, LOW);  
    analogWrite (ENB, 0);  
    digitalWrite (IN3, HIGH);  
    digitalWrite (IN4, LOW);  
    delay (500);  
}  
break;  
    case 'A':{  
        analogWrite (ENA, 255);  
        digitalWrite (IN1, LOW);  
        digitalWrite (IN2, HIGH);  
        analogWrite (ENB, 0);  
        digitalWrite (IN3, LOW);  
        digitalWrite (IN4, HIGH);  
        delay (500);  
    }  
break;
```

//條件Q，控制馬達右轉
//ENA馬達速率設定為0最低扭力

//ENB馬達速率設定為255最高扭力

//右轉
//跳出這個switch條件敘述
//條件E，控制馬達左轉
//ENA馬達速率設定為255最高扭力

//ENB馬達速率設定為0最低扭力

//左轉
//跳出這個switch條件敘述
//條件A，控制馬達後退右轉
//ENA馬達速率設定為255最高扭力

//ENB馬達速率設定為0最低扭力

//後退右轉
//跳出這個switch條件敘述



解答

Arduino 畫面

```
case 'D':{  
  analogWrite (ENA, 0);  
  digitalWrite (IN1, LOW);  
  digitalWrite (IN2, HIGH);  
  analogWrite (ENB, 255);  
  digitalWrite (IN3, LOW);  
  digitalWrite (IN4, HIGH);  
  delay (500);  
}  
break;  
}  
{  
  analogWrite (ENA, 255);  
  digitalWrite (IN1, HIGH);  
  digitalWrite (IN2, HIGH);  
  analogWrite (ENB, 255);  
  digitalWrite (IN3, HIGH);  
  digitalWrite (IN4, HIGH);  
  delay (500);  
}  
}  
}
```

//條件D，控制馬達後退左轉
//ENA馬達速率設定為0最低扭力

//ENB馬達速率設定為255最高扭力

//後退左轉
//跳出這個switch條件敘述

//ENA馬達速率設定為255最高扭力

//ENB馬達速率設定為255最高扭力

//不管執行哪個條件，跳出switch條件敘述時，車子都會緊急停止

專題思考

在藍牙遙控馬達中加入避障功能。



專題思考

手動模式與自動模式的切換。



營隊群組



-END-