



TARKUSTECH INNOVATIONS

Intro deck 2019

ABOUT

TARKUS

i n n o v a t i o n

課程大綱基本架構

1. 基礎農場套件課程架構

期	預計課堂	課程名稱	內容
1	2	課程整體介紹	1.課程方向/基本原理介紹; 2.植物特性介紹; 3.材料收集介紹(容器,管線)
2	4	硬體環境設置	1.基本農場環境配置; 2.收集材料整理加工; 3.環境控制組件配置
3	2	警示燈與蜂鳴器模組	1.警示燈與蜂鳴器模組介紹; 2. TARKUSVP實作; 3.程式啟動測試
4	1	土壤溼度偵測	1.土壤濕度感應器介紹; 2. TARKUSVP實作; 3.程式控制偵測測試
5	1	水位偵測	1.水位感應器介紹; 2. TARKUSVP實作; 3.程式控制偵測測試
6	1	雨滴偵測	1.雨滴感應器介紹; 2. TARKUSVP實作; 3.程式控制偵測測試
7	1	環境溫度與濕度偵測	1.溫度濕度感應器介紹; 2. TARKUSVP實作; 3.程式控制偵測測試
8	1	環境燈光偵測	1.光線感應器介紹; 2. TARKUSVP實作; 3.程式控制偵測測試
9	2	溫度與濕度控制 / 風扇控制	1.繼電器模組及風扇應用介紹; 2. TARKUSVP及配置方法實作; 3.對應感應器程式控制偵測測試
10	2	環境燈光控制 →燈光照明控制	1. 燈光模組應用介紹→植物燈背景知識; 莖葉類與根莖類的光線差別 2. TARKUSVP及配置方法實作; 3. 對應感應器程式控制偵測測試
11	2	土壤溼度控制 / 抽水幫浦 / 水位偵測 / 盆底雨滴偵測	1.繼電器模組及抽水幫浦應用介紹; 2. TARKUSVP及配置方法實作; 3.對應感應器程式控制偵測測試
12	2	實際動手種植物	1.整體農場整理; 2.實際動手栽種; 3.後續觀察規劃計畫

- 2 IOT課程架構
- 3 農場延伸套件
- 4 魚菜共生套件
- 5 水質監控套件

智能農場首套件功能模組探討：硬體呈現



IOT 模組

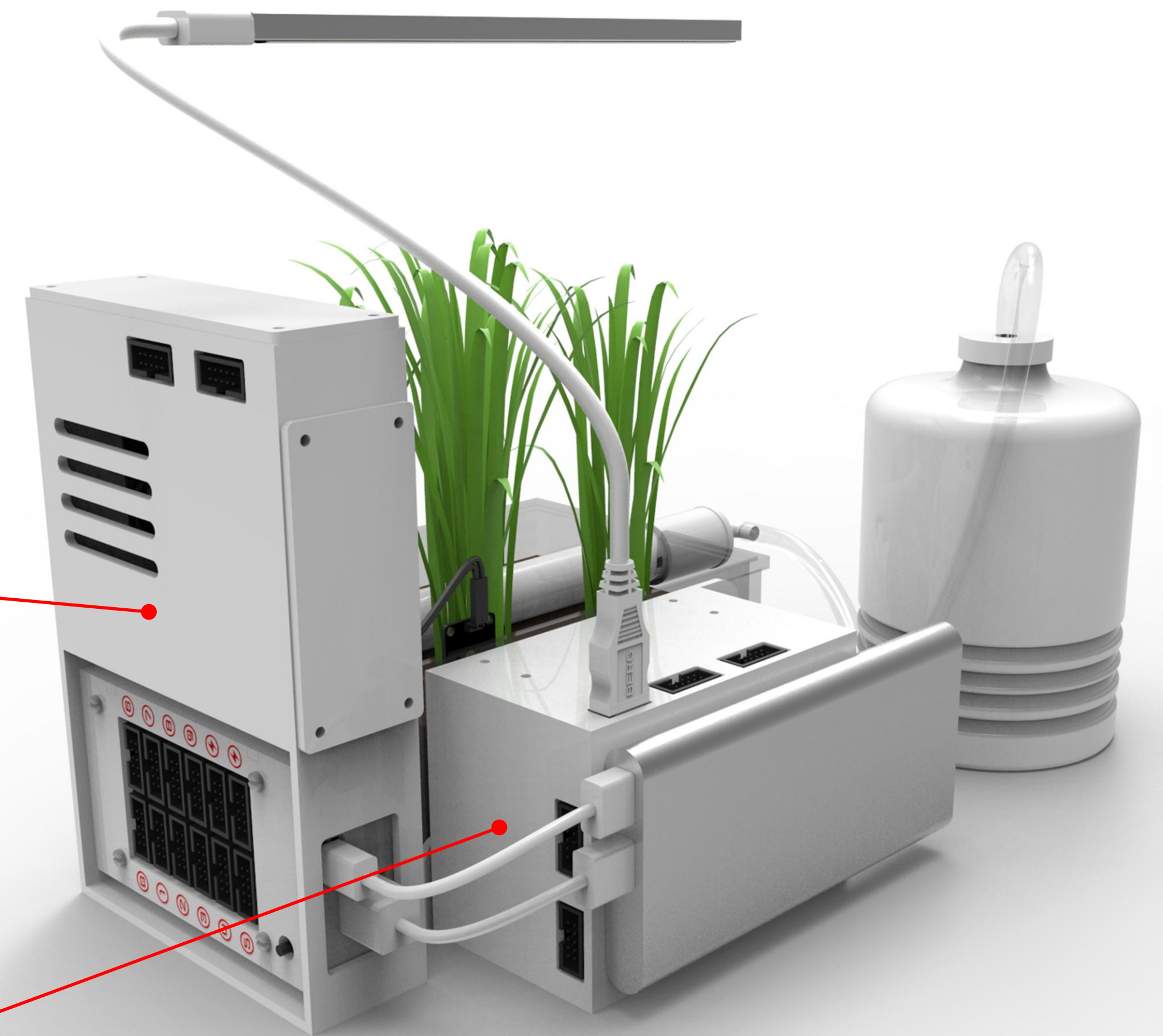
ESP32+CAM
物聯網/雲端網頁監控App
時鐘模組
SD卡存儲模組

百葉箱模組

空氣溫濕度 dht11
土壤濕度感測器soil
光敏電阻模組
雨滴感測器
RGB指示燈/蜂鳴器模組
電扇

電源模組

抽水幫浦
USB植物生長燈
電量偵測
電量指示燈



智能農場首套件功能模組探討：軟體呈現



百葉箱模組

空氣溫濕度 dht11
土壤濕度感測器soil
光敏電阻模組
(雨滴感測器)
RGB指示燈/蜂鳴器模組
電扇

電源模組

抽水幫浦
USB植物生長燈
電量偵測
電量指示燈

IOT 模組

ESP32+CAM
物聯網/雲端網頁監控App
時鐘模組
SD卡存儲模組

彩色警示燈模組

蜂鳴器模組

土壤濕度模組

雨滴感測模組

溫濕度感測模組

光線感測模組

風扇模組

水幫浦模組

植物生長燈模組

電池電量模組

雲端網頁監控App

WiFi讀取模組

影像監控模組

儲存至SD

讀取時鐘模組

WiFi匯出資料

A group of students in a classroom are seated at desks, working on laptops. The students are focused on their screens, and the classroom environment is visible in the background. The image has a slightly dim, warm-toned aesthetic.

百葉箱模組

Tarkustech Innovations Co. Ltd. Copyright
from 2017

百葉箱模組

空氣溫濕度 dht11
土壤濕度感測器soil
光敏電阻模組
雨滴感測器
RGB指示燈/蜂鳴器模組
電扇

彩色警示燈模組

蜂鳴器模組

土壤濕度模組

雨滴感測模組

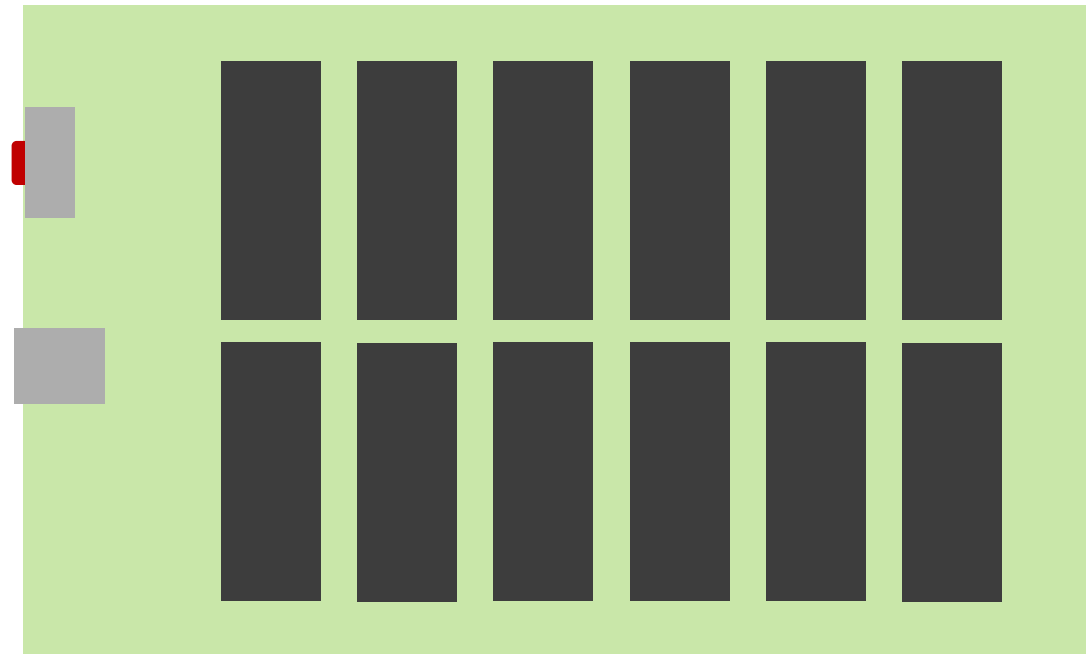
溫濕度感測模組

光線感測模組

風扇模組

DIGITAL		ANALOG		SERIAL			POWER	
IN	OUT	IN	PWM	TTL	I2C	SPI	VCC	GND
	PIN 43						V主板	GND
			PIN 45				V主板	GND
PIN42		A14					V主板	GND
PIN44		A15			SLOT6		V主板	GND
PIN33					SLOT5			
PIN30		A12					V主板	GND
	PIN31						V擴	GND

HOST_PCB (TOP面)



PIN definitions:

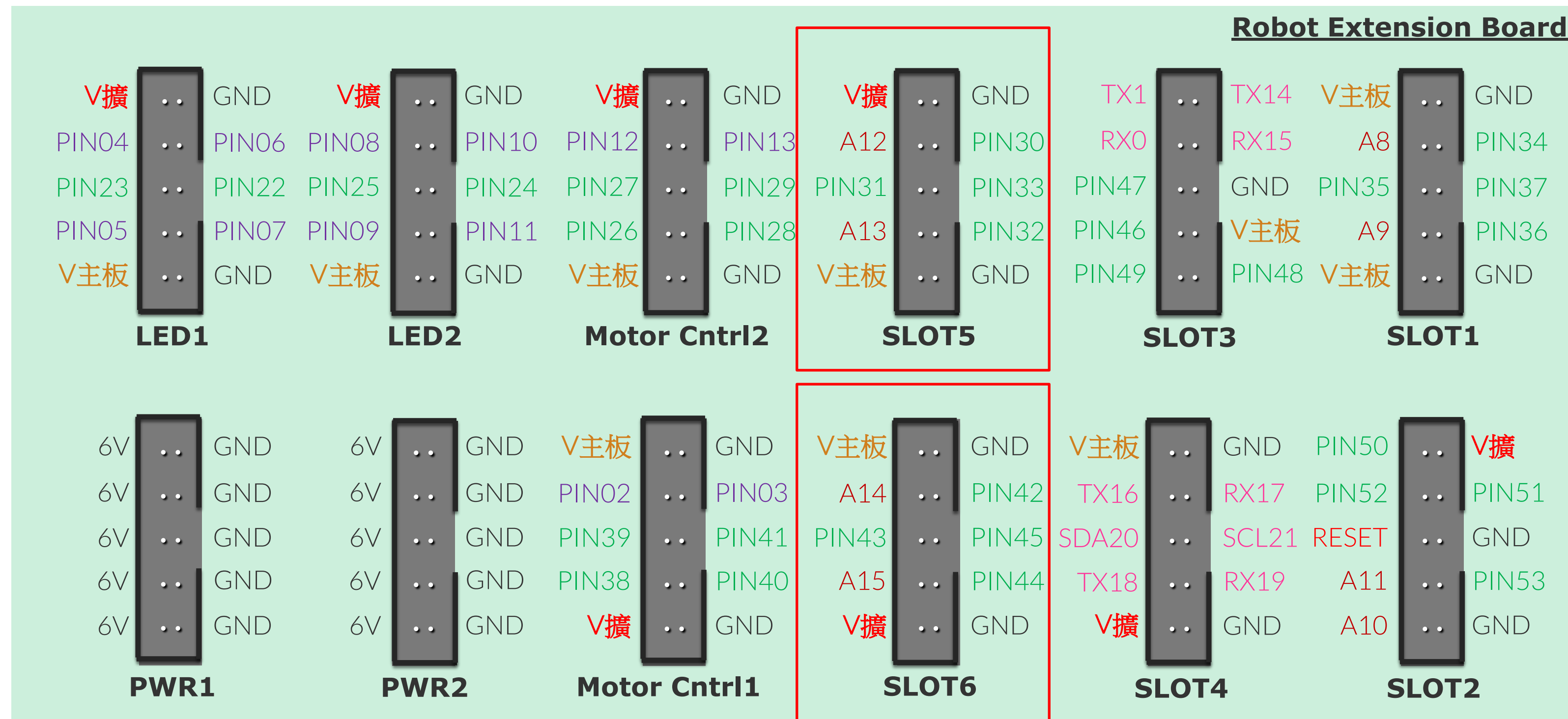
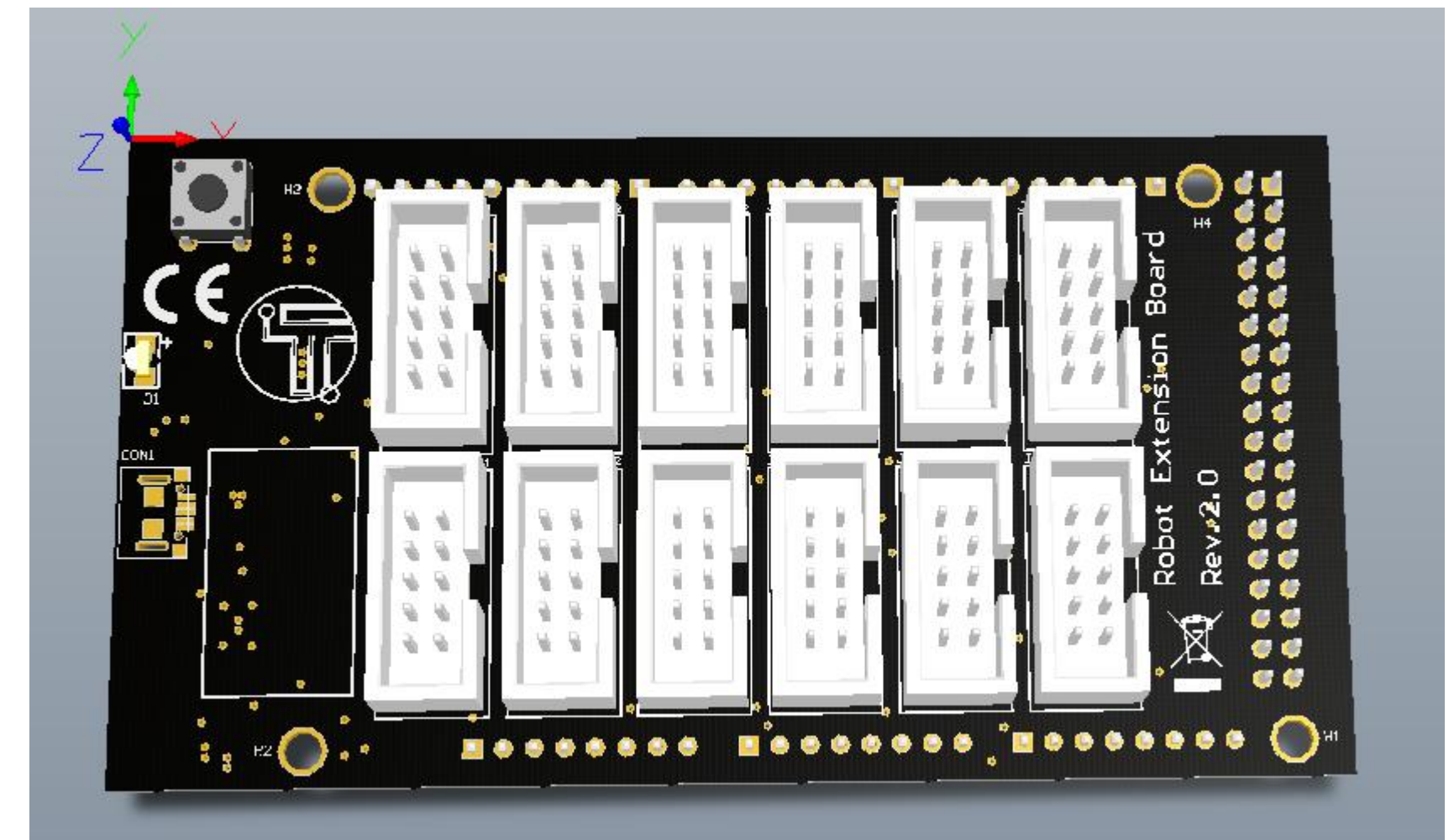
PWM PIN: 2~13

Digital PIN: 22~49

Analog PIN: A8~A15

TTL/SDI: 14~21

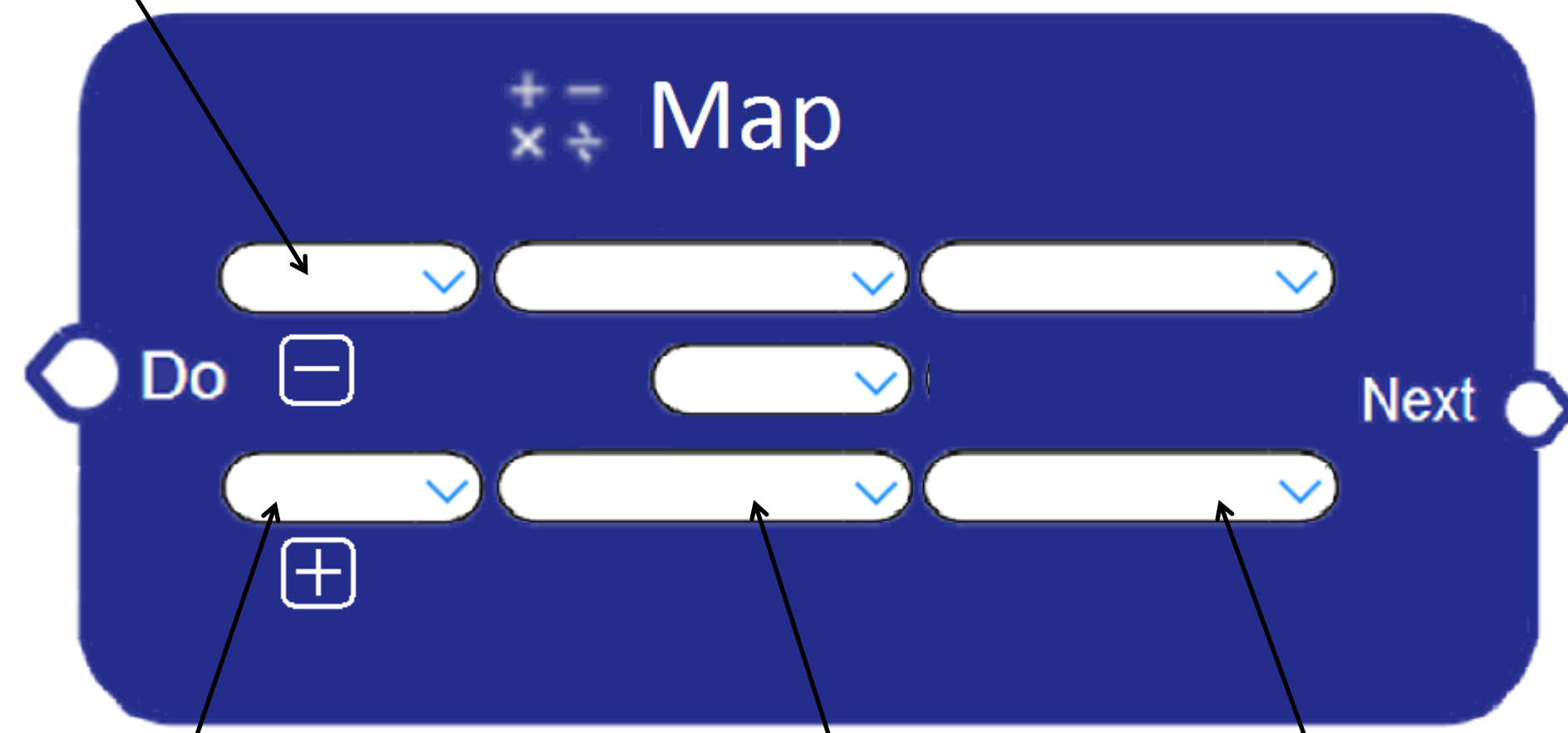
V主板/GND



RGB LED Indicator

- 4顆LED只是增加亮度用, 用同一個設定
- RGB分別有數位/Dimmer模式
- 使用MAP range變數做指定顏色的漸暗/漸亮
- 例如Brightness {(min. 0), (Max. 100)}
map至R(0,MaxR),G(0,MaxG),B(0,MaxB)的觀念

From變數(需為數字)

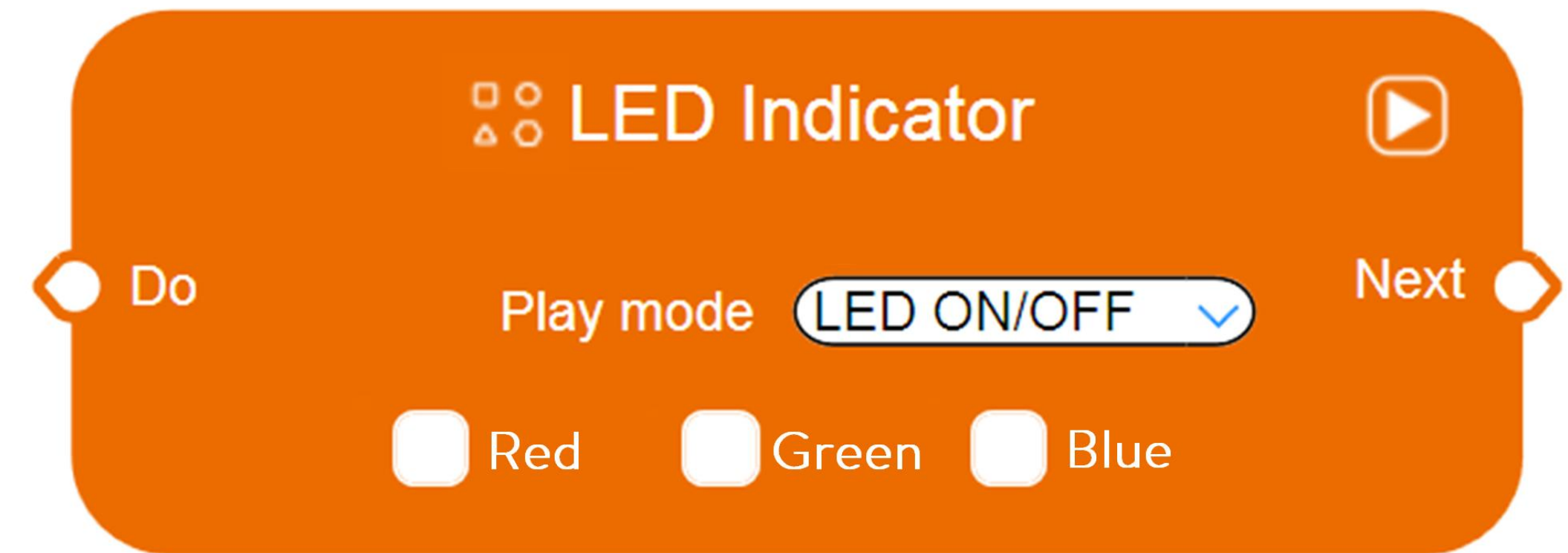
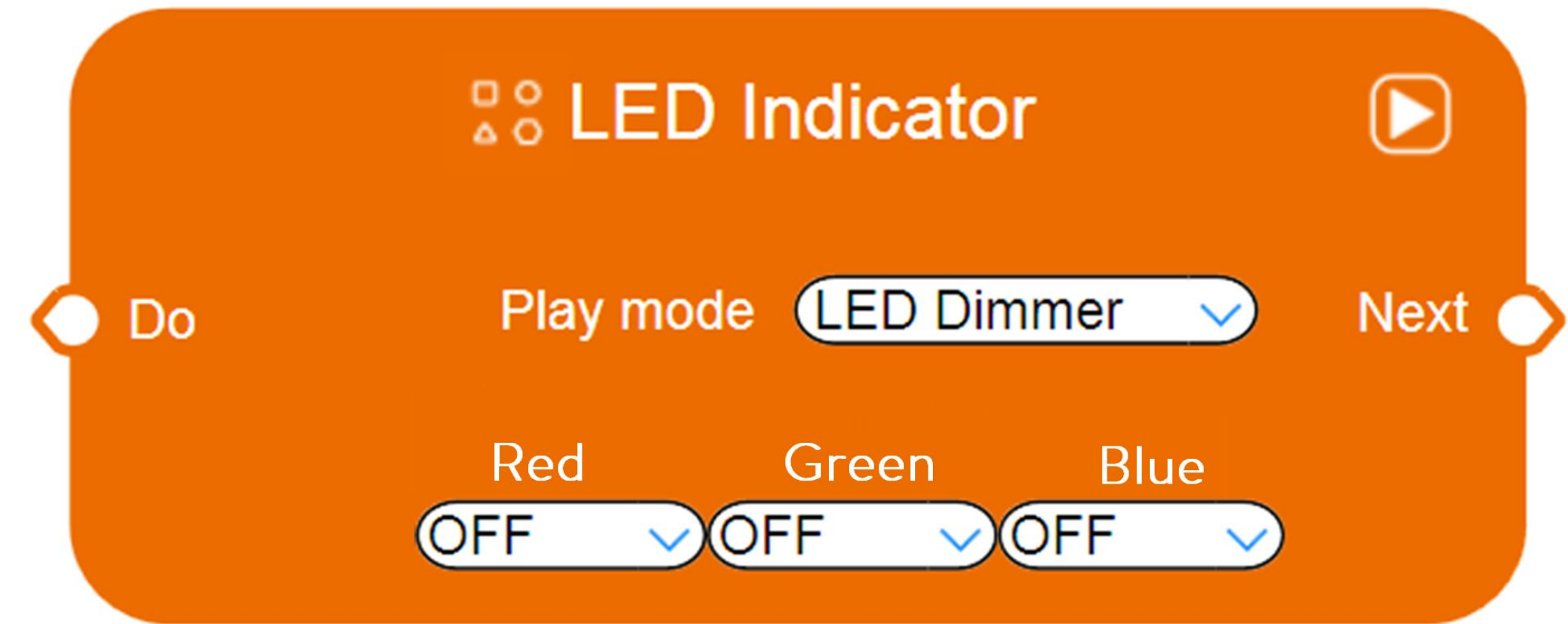


To變數(需為數字)

區間最大值

區間最小值

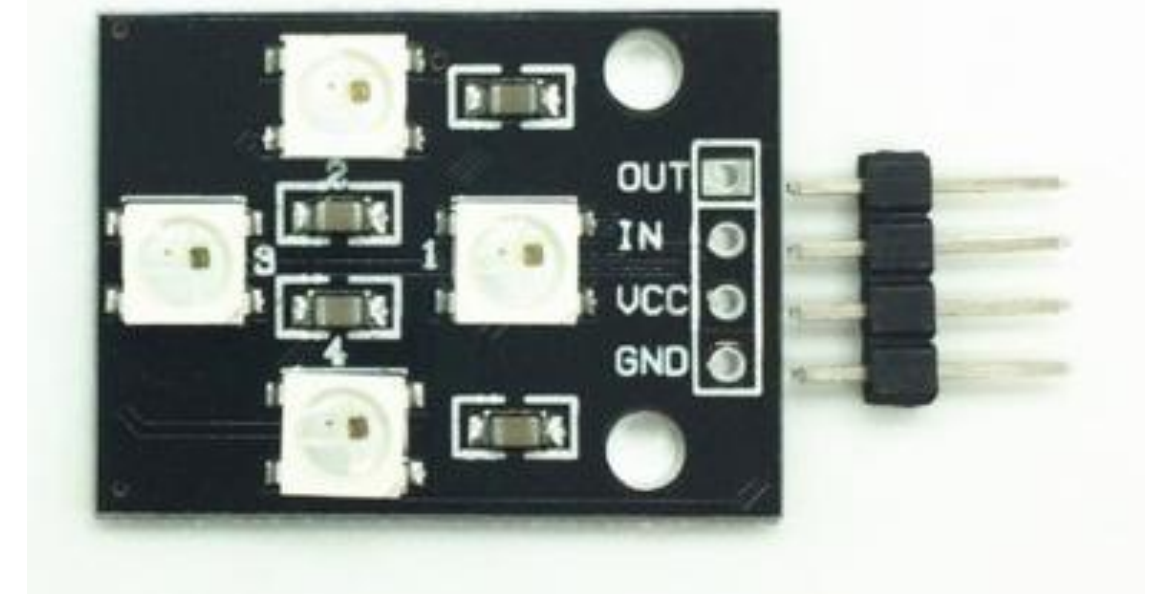
$$B = \text{map}(A, A_{\text{max}}, A_{\text{min}}, B_{\text{max}}, B_{\text{min}})$$



WS2812B-4 color LED chip #option1

```
#include <FastLED.h>
#define LED_PIN 7
#define NUM_LEDS 20
CRGB leds[NUM_LEDS];
void setup() {
  FastLED.addLeds<WS2812, LED_PIN, GRB>(leds, NUM_LEDS);
}
void loop() {
  leds[0] = CRGB(255, 0, 0);
  FastLED.show();
  delay(500);
  leds[1] = CRGB(0, 255, 0);
  FastLED.show();
  delay(500);
  leds[2] = CRGB(0, 0, 255);
  FastLED.show();
  delay(500);
  leds[5] = CRGB(150, 0, 255);
  FastLED.show();
  delay(500);
  leds[9] = CRGB(255, 200, 20);
  FastLED.show();
  delay(500);
  leds[14] = CRGB(85, 60, 180);
  FastLED.show();
  delay(500);
  leds[19] = CRGB(50, 255, 20);
  FastLED.show();
  delay(500);
}
```

```
#include <FastLED.h>
#define LED_PIN 7
#define NUM_LEDS 20
CRGB leds[NUM_LEDS];
void setup() {
  FastLED.addLeds<WS2812, LED_PIN, GRB>(leds, NUM_LEDS);
}
void loop() {
  for (int i = 0; i <= 19; i++) {
    leds[i] = CRGB ( 0, 0, 255);
    FastLED.show();
    delay(40);
  }
  for (int i = 19; i >= 0; i--) {
    leds[i] = CRGB ( 255, 0, 0);
    FastLED.show();
    delay(40);
  }
}
```



<https://howtomechatronics.com/tutorials/arduino/how-to-control-ws2812b-individually-addressable-leds-using-arduino/>

WS2812B-4 color LED chip #option2

- **下載函式庫**

1. 點選右側網址下載函式庫 https://github.com/adafruit/Adafruit_NeoPixel
2. 將下載的檔案解壓縮，複製到 Arduino IDE 的 `libraries` 資料夾裏。
3. 更改名稱為「`Adafruit_NeoPixel`」。

- **執行範例程式**

1. 啟動 Arduino IDE。
2. 點擊「檔案」>「範例」>「`Adafruit_NeoPixel`」>「`strandtest`」。
3. 找到這一行

```
Adafruit_NeoPixel strip = Adafruit_NeoPixel(60, PIN, NEO_GRB + NEO_KHZ800);
```

修改為

```
Adafruit_NeoPixel strip = Adafruit_NeoPixel(8, PIN, NEO_GRB + NEO_KHZ800);  看有幾顆LED燈？
```

- **Demo Code**(取自<https://learn.adafruit.com/getting-started-with-flora/blink-onboard-neopixel>)

```
#include <Adafruit_NeoPixel.h>
```

```
#define PIN 6 //接腳
```

```
#define MAX_LED 1 //LED的數量
```

```
//Adafruit_NeoPixel strip = Adafruit_NeoPixel(1, PIN, NEO_GRB + NEO_KHZ800);
```

```
Adafruit_NeoPixel strip = Adafruit_NeoPixel( MAX_LED, PIN, NEO_GRB + NEO_KHZ800 ); //NEO_GRB(顯示顏色是RGB)；NEO_RGB(顯示顏色是GRB)
```

```
//Adafruit_NeoPixel就是類別，其建構子有3個參數，分別為LED的數量、硬體連接的腳位、以及LED的型別。
```

```
//以上例來說，只有1個LED，連接在第6支腳，採用GRB，800KHZ通訊訊號的速率。
```

```
//為何LED跟通訊訊號的速率有關，因為它採用串接方式，當送出10顆LED命令時，接收到的第一顆LED會取下其命令，然後把9個命令再往下送，以此類推。
```

```
// Parameter 1 = number of pixels in strip
```

```
// Parameter 2 = Arduino pin number (most are valid)
```

```
// Parameter 3 = pixel type flags, add together as needed:
```

```
// NEO_KHZ800 800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)
```

```
// NEO_KHZ400 400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)
```

```
// NEO_GRB Pixels are wired for GRB bitstream (most NeoPixel products)
```

```
// NEO_RGB Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)
```

```
// NEO_RGBW Pixels are wired for RGBW bitstream (NeoPixel RGBW products)
```

```
void setup() {
```

```
strip.begin(); // 初始化庫
```

```
strip.show(); // Initialize all pixels to 'off' // 發送數據，默認每個點的顏色為0，所以初始的每個點都是不亮的
```

```
}
```

```
void loop() {
```

```
// Some example procedures showing how to display to the pixels:
```

```
colorWipe(strip.Color(255, 0, 0), 500); // Red (紅) 顯示顏色，參數為 R G B，範圍0-255(亮度)，後方參數為顯示時間。colorWipe第一個參數為顏色值，第二個為延遲的時間。
```

```
colorWipe(strip.Color(0, 255, 0), 500); // Green(綠)
```

```
colorWipe(strip.Color(0, 0, 255), 500); // Blue(藍)
```

```
rainbowCycle(20);
```

```
}
```

```
// Fill the dots one after the other with a color
```

```
// uint32_t color = strip.Color(val, val, val); // 設置顏色，參數為 R G B，範圍0-255
```

```
// strip.setPixelColor(0, color); // 設置某個點的顏色，若只有一個，所以只控制第0個點
```

```
void colorWipe(uint32_t c, uint8_t wait) {
```

```
for(uint16_t i=0; i<strip.numPixels(); i++) { //strip.numPixels() LED個數-1
```

```
strip.setPixelColor(i, c);
```

```
strip.show();
```

```
delay(wait);
```

```
}
```

```
}
```

<https://sites.google.com/site/wenyumaker/30-qi-ta/02--8-bit-2812-rgb-led>

Tone

```
#define E5 659  
#define C5 523  
#define G5 784
```

```
int buzzerPin=9;
```

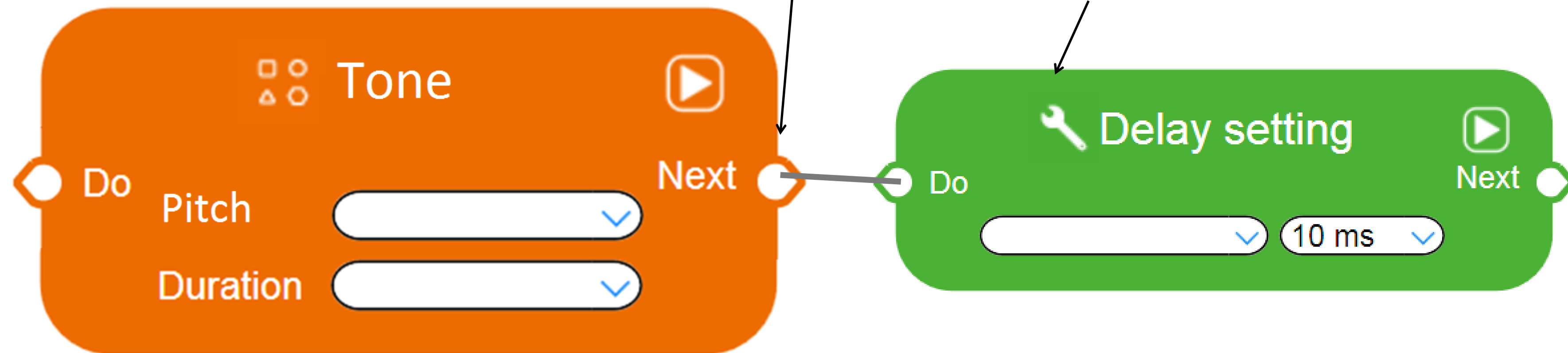
```
void setup() {  
  pinMode(buzzerPin, OUTPUT);  
}
```

```
void loop() {  
  mario(buzzerPin);  
}
```

```
void mario(int pin) {  
  tone(pin, E5, 150); //E5  
  delay(150); //pause for key transition  
  tone(pin, E5, 150); //E5  
  delay(150); //pause for key transition  
  tone(pin, E5, 150); //E5  
  delay(150); //pause for key transition  
  delay(150); //pause for a Quarter rest (1/4)  
  tone(pin, C5, 150); //C5  
  delay(150); //pause for key transition  
  tone(pin, E5, 150);  
  delay(150); //pause for key transition  
  delay(150); //pause for a Quarter rest (1/4)  
  tone(pin, G5, 150); //G5  
  delay(3000);  
}
```

與先前Buzzer一樣
最後狀態為關

做為休止符



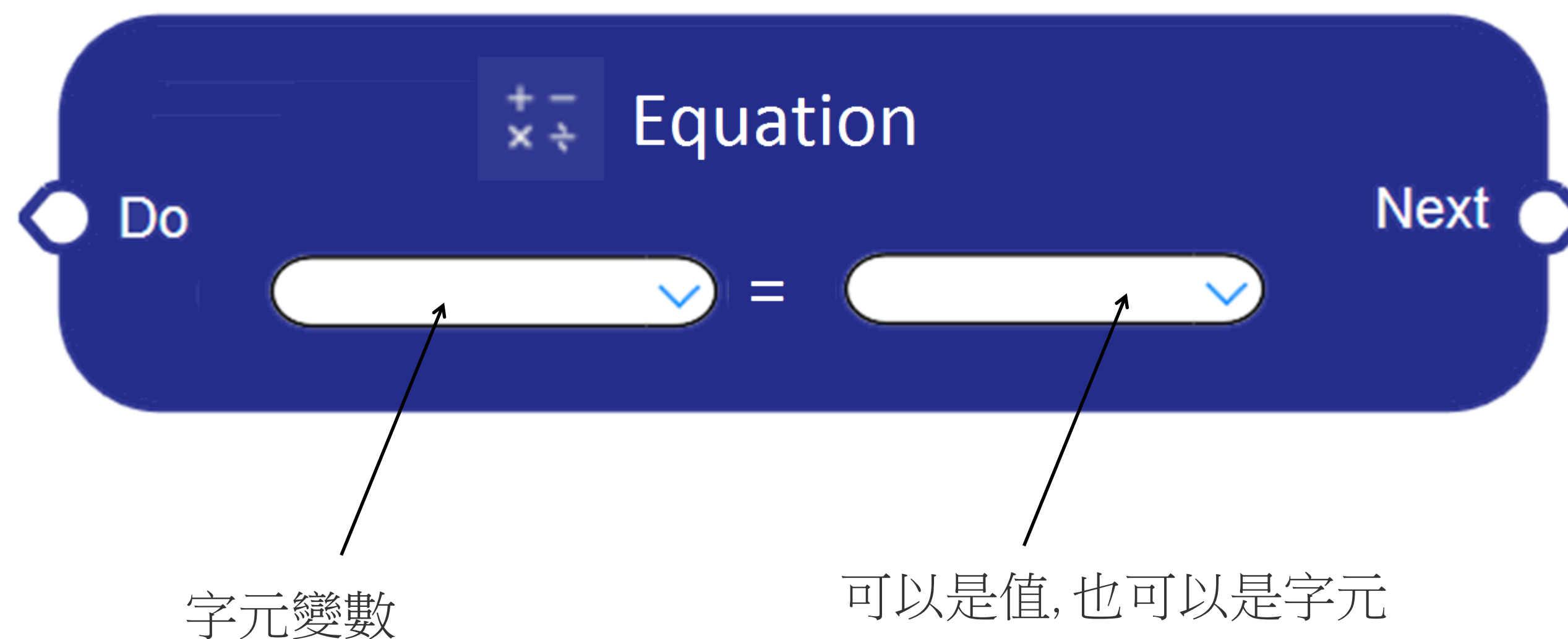
Equation

#define C0 18	#define DS1 39	#define CS2 69	#define E3 165	#define G4 392	#define AS5 932	#define CS7 2217	#define E8 5274
#define CS0 17	#define B0 31	#define D2 73	#define F3 175	#define GS4 415	#define B5 988	#define D7 2349	#define F8 5588
#define D0 18	#define C1 33	#define DS2 78	#define FS3 185	#define A4 440	#define C6 1047	#define DS7 2489	#define FS8 5920
#define DS0 19	#define CS1 35	#define E2 82	#define G3 196	#define AS4 466	#define CS6 1109	#define E7 2637	#define G8 6272
#define E0 21	#define D1 37	#define F2 87	#define GS3 208	#define B4 494	#define D6 1175	#define F7 2794	#define GS8 6645
#define F0 22	#define DS1 39	#define FS2 93	#define A3 220	#define C5 523	#define DS6 1245	#define FS7 2960	#define A8 7040
#define FS0 23	#define E1 41	#define G2 98	#define AS3 233	#define CS5 554	#define E6 1319	#define G7 3136	#define AS8 7459
#define G0 25	#define F1 44	#define GS2 104	#define B3 247	#define D5 587	#define F6 1397	#define GS7 3322	#define B8 7902
#define GS0 26	#define FS1 46	#define A2 110	#define C4 262	#define DS5 622	#define FS6 1480	#define A7 3520	
#define A0 28	#define G1 49	#define AS2 117	#define CS4 277	#define E5 659	#define G6 1568	#define AS7 3729	
#define AS0 29	#define GS1 52	#define B2 123	#define D4 294	#define F5 698	#define GS6 1661	#define B7 3951	
#define B0 31	#define A1 55	#define C3 131	#define DS4 311	#define FS5 740	#define A6 1760	#define C8 4186	
#define C1 33	#define AS1 58	#define CS3 139	#define E4 330	#define G5 784	#define AS6 1865	#define CS8 4435	
#define CS1 35	#define B1 62	#define D3 147	#define F4 349	#define GS5 831	#define B6 1976	#define D8 4699	
#define D1 37	#define C2 65	#define DS3 156	#define FS4 370	#define A5 880	#define C7 2093	#define DS8 4978	

這裡不用#define 但先前少了等式, 以至於 C0=18在先前的軟體裡面Geomatic需要表示成"C0=18+0"或"C0=18*1"

```
void loop() {  
  mario(buzzerPin);  
}
```

```
void mario(int pin) {  
  tone(pin, E5, 150);  
  delay(150);  
  tone(pin, E5, 150);  
  delay(150);  
  tone(pin, E5, 150);  
  delay(300);  
  tone(pin, C5, 150);  
  delay(150);  
  tone(pin, E5, 150);  
  delay(300);  
  tone(pin, G5, 150);  
  delay(3000);  
}
```



Tone

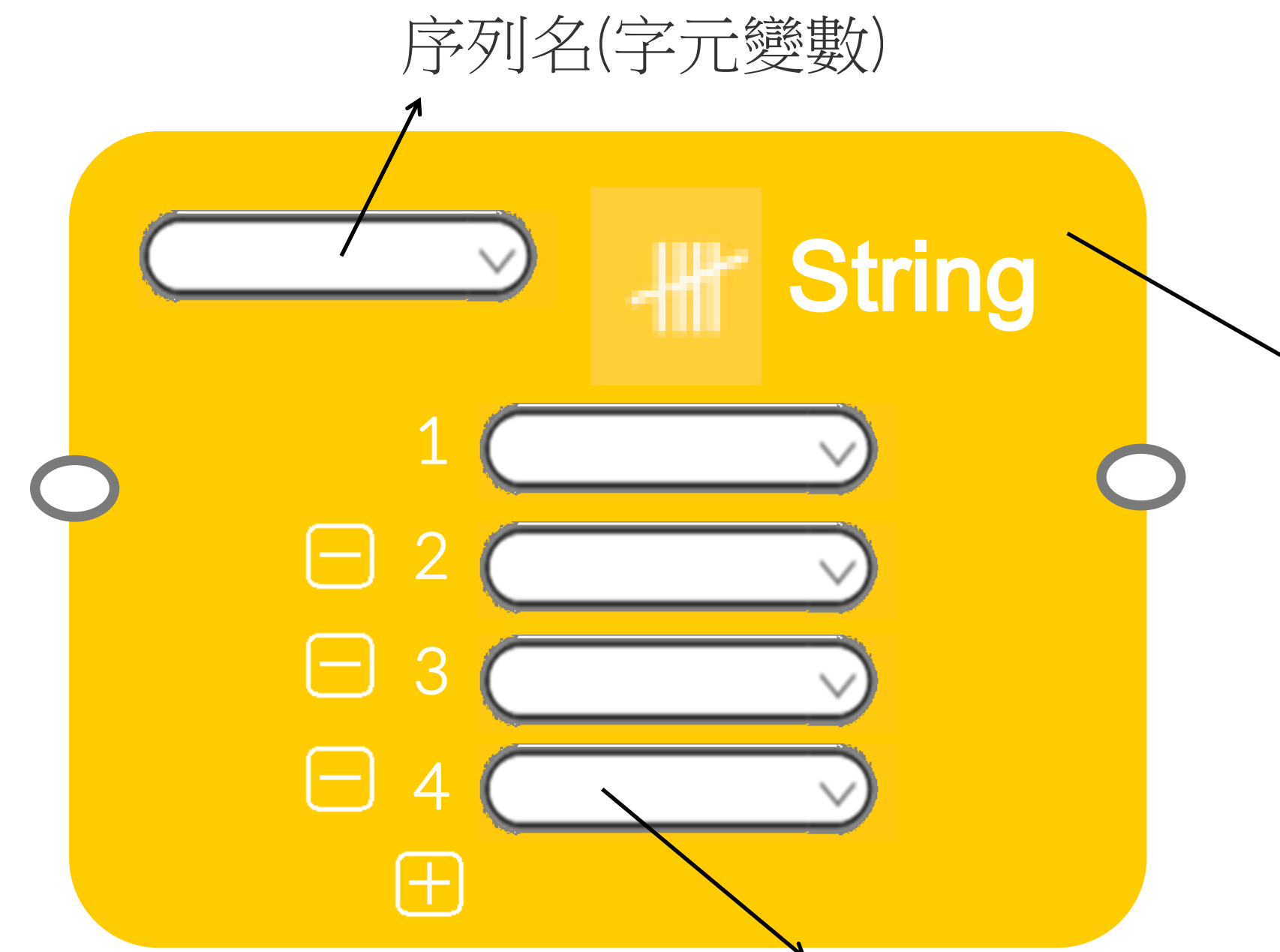
```
#include "notes.h"
```

```
int buzzerPin=9;  
int note[]={C4, G3, G3, A3, G3, 0, B3, C4}; //note name  
int duration[]={4, 8, 8, 4, 4, 4, 4, 4}; //note type (tempo)
```

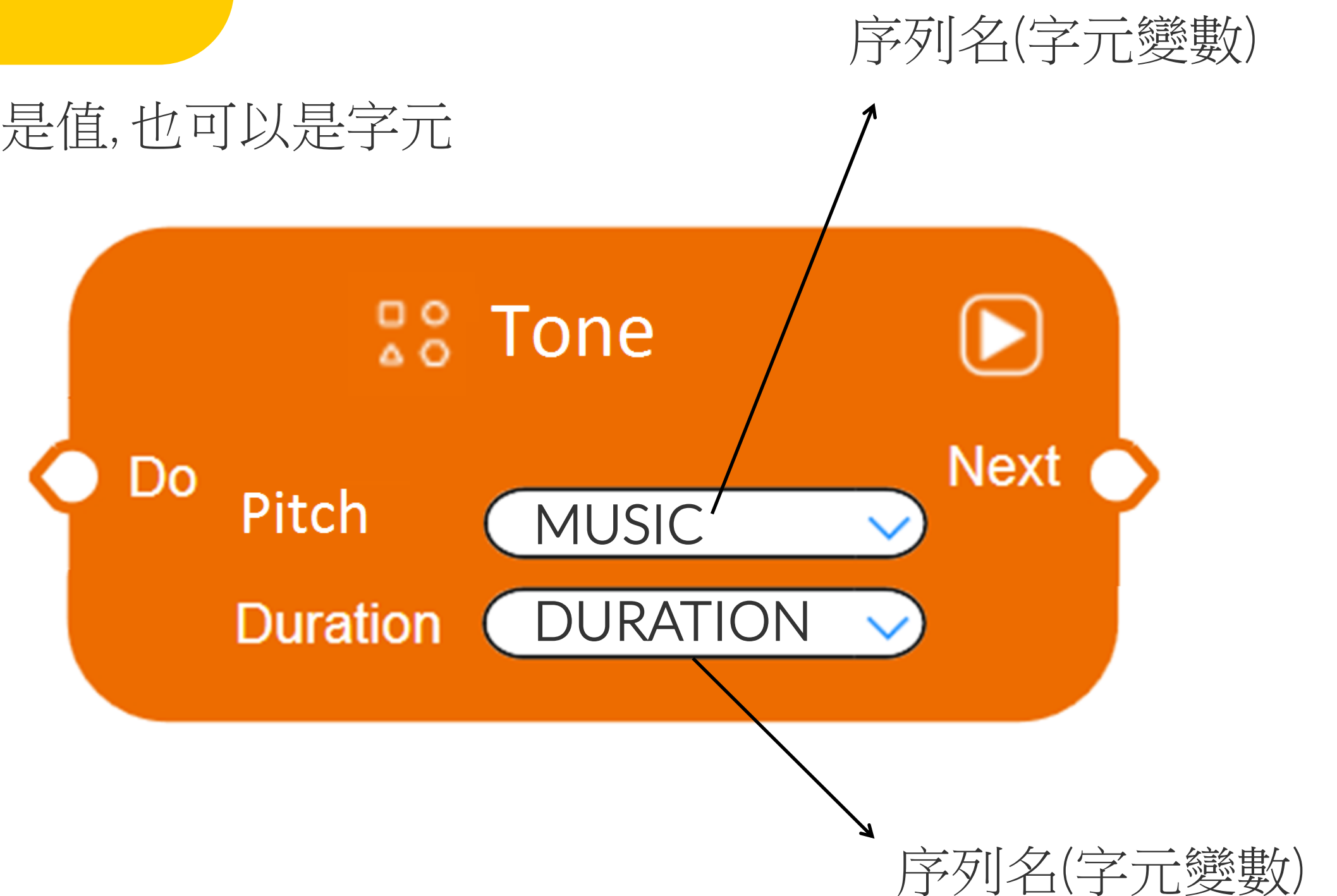
```
void setup() {  
  pinMode(buzzerPin, OUTPUT);  
}
```

```
void loop() {  
  melody(buzzerPin);  
}
```

```
void melody(int pin) {  
  for (int i=0; i<8; i++) { //visit 8 notes  
    int d=1000/duration[i]; //transform tempo to ms  
    tone(pin, note[i], d);  
    int p=d * 1.3; //suitable pause=plus 30% duration  
    delay(p); //pause between notes  
    noTone(8);  
  }  
  delay(2000);  
}
```



字串陣列, 有別於string字元陣列
<http://www.86duino.com/?p=1498&lang=TW>



DHT11

```
#include <dht11.h> //引用dht11庫檔案
#define DHT11PIN 9 //定義溫溼度針腳號為9號引腳
dht11 DHT11;
float tem;
float hum;
void setup() {
  Serial.begin(9600); //設定波特率引數
  pinMode(DHT11PIN,INPUT);
  Serial.println(F("reading Dh11 begin. \n"));
}

void loop() {
  static unsigned long sensortStamp = 0;
  if(millis()-sensortStamp>1000){
    sensortStamp=millis();
    int chk = DHT11.read(DHT11PIN); //將讀取到的值賦給chk
    tem=(float)DHT11.temperature; //將溫度值賦值給tem
    hum=(float)DHT11.humidity; //將溼度值賦給hum
    Serial.print(F("Real Time tem: "));
    Serial.println(tem);
    Serial.print(F("Real Time hum: "));
    Serial.println(hum);
  }
}
```

WHAT IS RELATIVE HUMIDITY?

The DHT11 measures *relative humidity*. Relative humidity is the amount of water vapor in air vs. the saturation point of water vapor in air. At the saturation point, water vapor starts to condense and accumulate on surfaces forming dew.

The saturation point changes with air temperature. Cold air can hold less water vapor before it becomes saturated, and hot air can hold more water vapor before it becomes saturated.

The formula to calculate relative humidity is:

$$RH = \left(\frac{\rho_w}{\rho_s} \right) \times 100\%$$

RH : Relative Humidity

ρ_w : Density of water vapor

ρ_s : Density of water vapor at saturation

Relative humidity is expressed as a percentage. At 100% RH, condensation occurs, and at 0% RH, the air is completely dry.

溫濕度感測模組

執行

下一步

讀取溫濕度感測器
並寫入S_TEM / S_HUM 變數

DHT11

```
#include <DHT.h>
#define DHTPIN 3
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  dht.begin();
  Serial.println("Hi Vcancy. Arduino is running...");
}
/**
  Arduino loop
  @param none
  @return none
 */
char target[] = "GET TH";

void loop() {
  //每次等待2秒後再輸出（這裡必須等大於1秒，不然不準確）
  delay(2000);
  //從串口讀數據並判斷
  if (Serial.available() > 0) {
    if (Serial.find(target)) {
      //獲取溫度或者濕度需要250毫秒!
      //感測器獲取到的溫度和濕度可能是2秒內的
      float h = dht.readHumidity();//讀取當前的濕度
      float t = dht.readTemperature();//讀取當前的溫度，單位C
      float f = dht.readTemperature(true);//讀取當前的溫度，單位F
```

```
//如果讀取失敗則退出，再讀取一次
if (isnan(h) || isnan(t) || isnan(f)) {
  Serial.println("Failed to read from DHT sensor!");
  return; } //讀取體感溫度，單位F
float hif = dht.computeHeatIndex(f, h);
//讀取體感溫度，單位C
float hic = dht.computeHeatIndex(t, h, false);
Serial.print("濕度: ");
Serial.print(h);
Serial.print(" %\t");
Serial.print("溫度: ");
Serial.print(t);
Serial.print(" *C ");
Serial.print(f);
Serial.print(" *F\t");
Serial.print("體感溫度: ");
Serial.print(hic);
Serial.print(" *C ");
Serial.print(hif);
Serial.println(" *F");
}
}
}
```

WHAT IS RELATIVE HUMIDITY?

The DHT11 measures *relative humidity*. Relative humidity is the amount of water vapor in air vs. the saturation point of water vapor in air. At the saturation point, water vapor starts to condense and accumulate on surfaces forming dew.

The saturation point changes with air temperature. Cold air can hold less water vapor before it becomes saturated, and hot air can hold more water vapor before it becomes saturated.

The formula to calculate relative humidity is:

$$RH = \left(\frac{\rho_w}{\rho_s} \right) \times 100\%$$

RH : Relative Humidity
 ρ_w : Density of water vapor
 ρ_s : Density of water vapor at saturation

Relative humidity is expressed as a percentage. At 100% RH, condensation occurs, and at 0% RH, the air is completely dry.

土壤濕度模組

雨滴感測模組

亮度感測模組



pulseIn()

Posted by roboard On 11 / 08 / 2013 0 Comment

描述

讀取 **pin** 腳上的脈衝 (高或低電位) 持續時間；舉個例子，如果**輸入值**是 HIGH，`pulseIn()` 會等 **pin** 腳上的電位變成 **HIGH** 之後開始計時，等到又變回 **LOW** 才停止；之後會回傳脈衝的持續時間，單位是微秒；假如在指定的時間內沒有出現任何脈衝或是脈衝長度太長，則會放棄讀取並回傳 **0**。

指定時間的長短是取決於經驗的，通常在長時間的脈衝下較容易發生錯誤，脈衝的測量長度介於 **10 毫秒**到 **3 分鐘**之間。

語法

```
pulseIn(pin, value)  
pulseIn(pin, value, timeout)
```

參數

pin: 用來讀取脈衝的 **pin** 腳 (int)

value: 讀取的脈衝型態，**HIGH** 或 **LOW** (int)

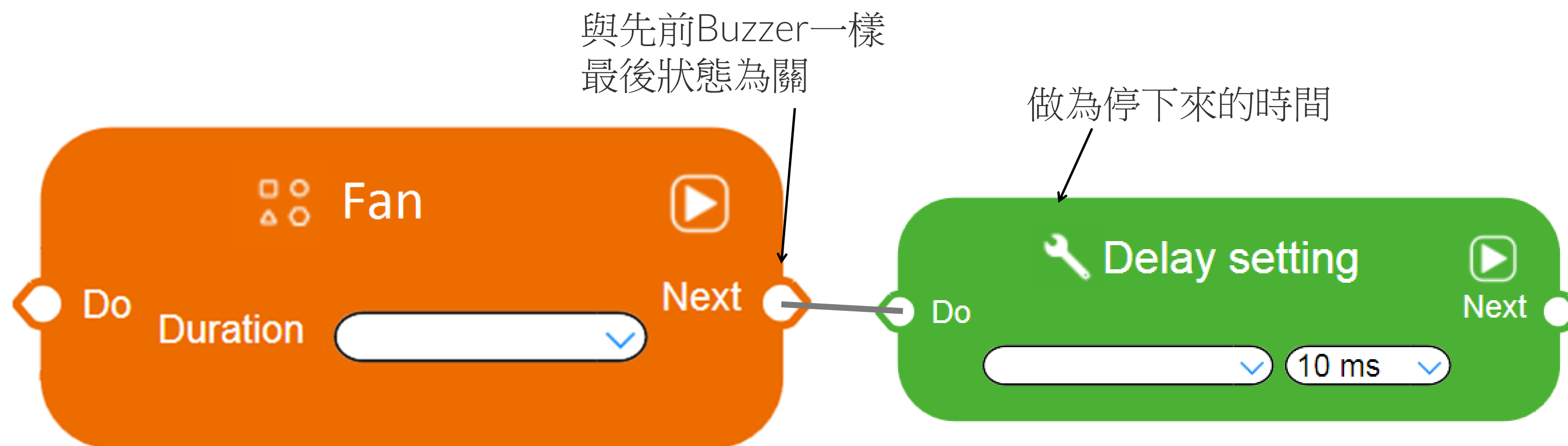
timeout (非必要): 設定一段時間來等待脈衝開始，預設是一秒 (unsigned long)

回傳

脈衝的時間長度 (單位是微秒)。如果在超時前沒有讀到脈衝的話則是回傳 **0** (unsigned long)。

Fan

用最簡單digital out語法就好



A photograph of a classroom where several students are seated at desks, working on laptops. The students are of various ethnicities and are focused on their screens. The lighting is warm and the atmosphere is educational. The text '電源模組' is overlaid in the center of the image.

電源模組

Tarkustech Innovations Co. Ltd. Copyright
from 2017

電源模組

抽水幫浦
USB植物生長燈
電量偵測
電量指示燈(App)

水幫浦模組

植物生長燈模組

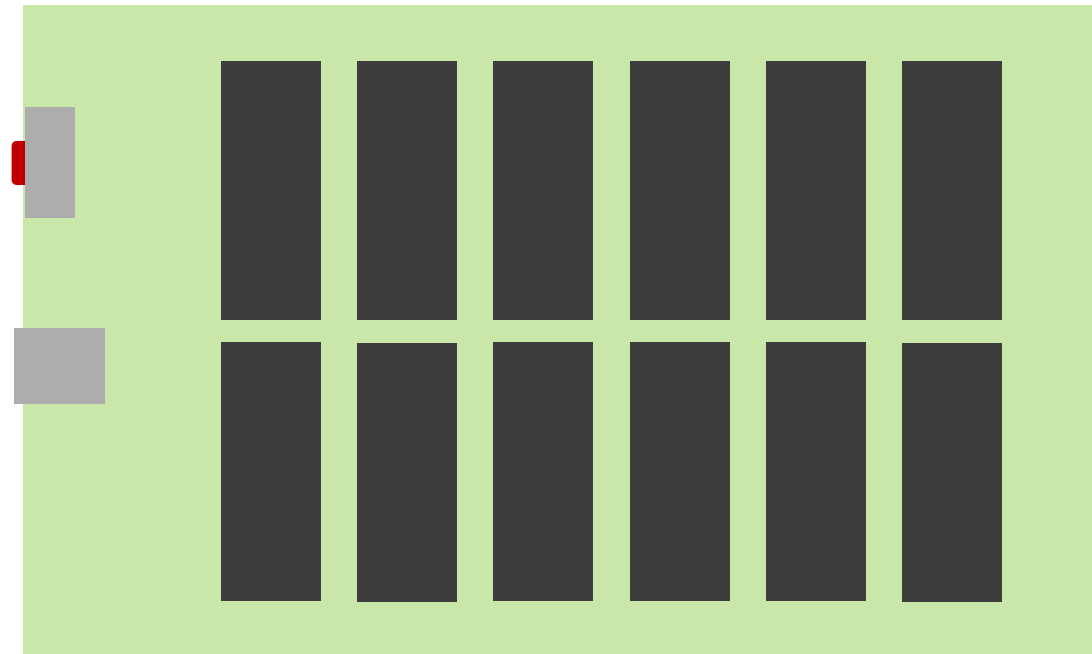
電池電量模組

DIGITAL		ANALOG		SERIAL			POWER	
IN	OUT	IN	PWM	TTL	I2C	SPI	VCC	GND
			PIN 02				V擴	GND
	PIN03						V擴	GND
PIN 38,39							V擴	GND
	PIN40,41						V主板	GND

電池電量: 讀取擴充板(38,39)為主, 若來自擴充板無訊號, 則讀取(40,41)

來自擴充板電量 V擴	來自主板電量 V主板	★★★★★	★★★	★★	★
PIN38	PIN40	1	1	0	0
PIN39	PIN41	1	0	1	0
		>80%	50~80%	20~80%	<20%

HOST_PCB (TOP面)



PIN definitions:

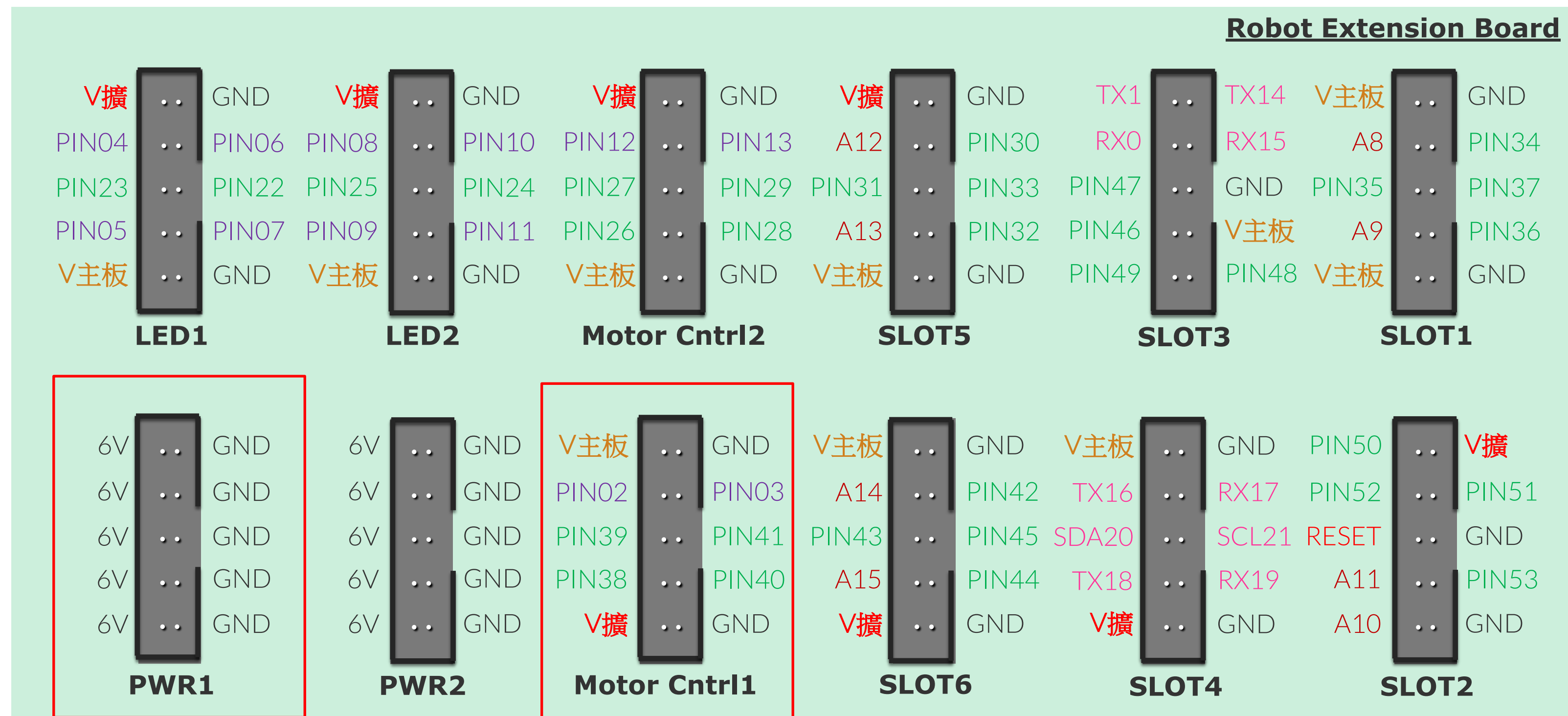
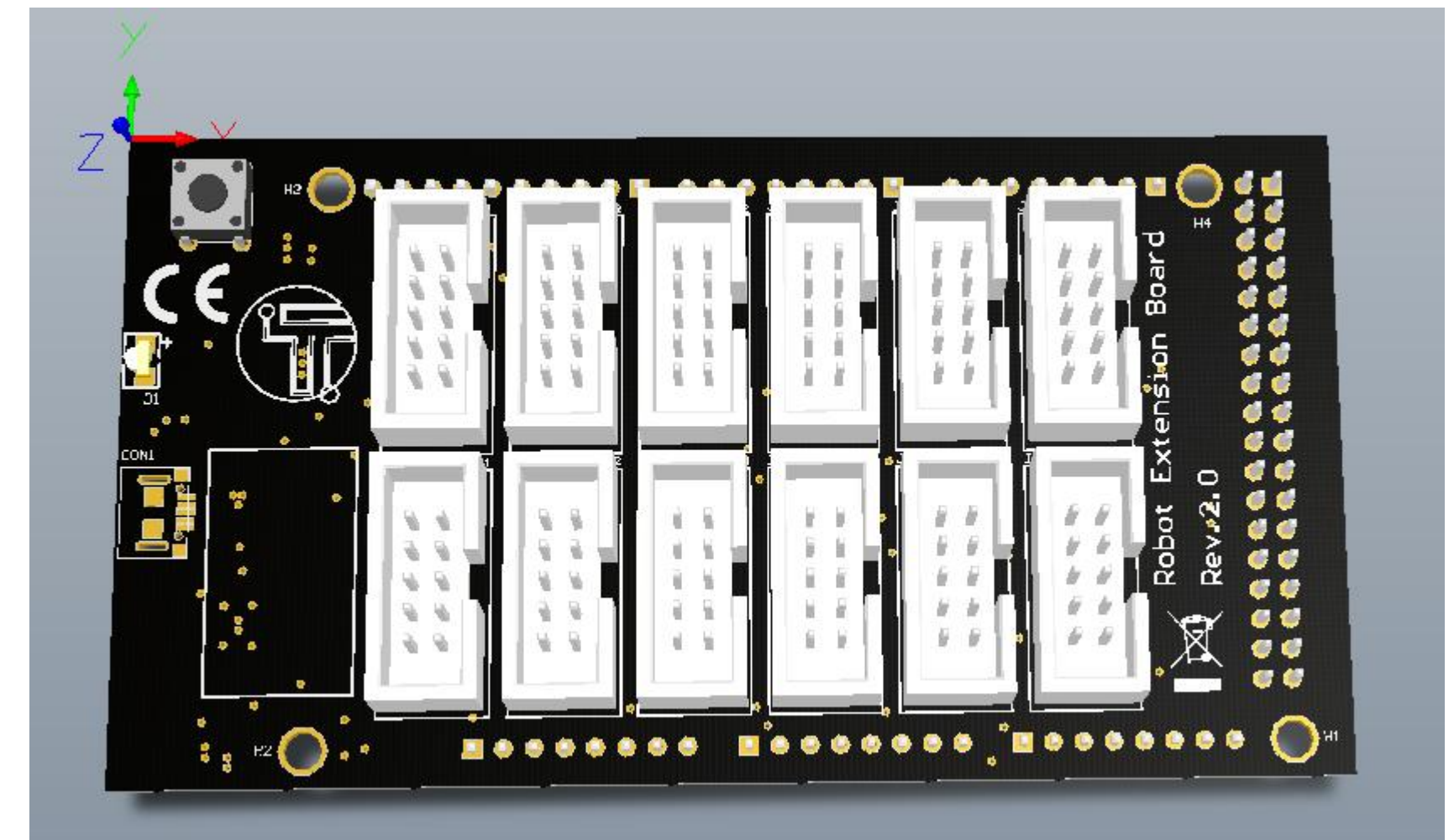
PWM PIN: 2~13

Digital PIN: 22~49

Analog PIN: A8~A15

TTL/SDI: 14~21

V主板/GND



Battery Power

Do

Read Battery power to S_BATTERY variable.

Next

Water pump

Do

Speed

Duration

Next

Plant Light

Do

Next

Lightening

植物生長燈模組

執行

下一步

照明

A group of students in a classroom are seated at desks, working on laptops. The students are focused on their screens, and the classroom environment is visible in the background. The text 'IOT 模組' is overlaid in the center of the image.

IOT 模組

Tarkustech Innovations Co. Ltd. Copyright
from 2017

IOT 模組

ESP32+CAM
物聯網/雲端網頁監控App
時鐘模組
SD卡存儲模組

雲端網頁監控App

WiFi讀取模組

影像監控模組

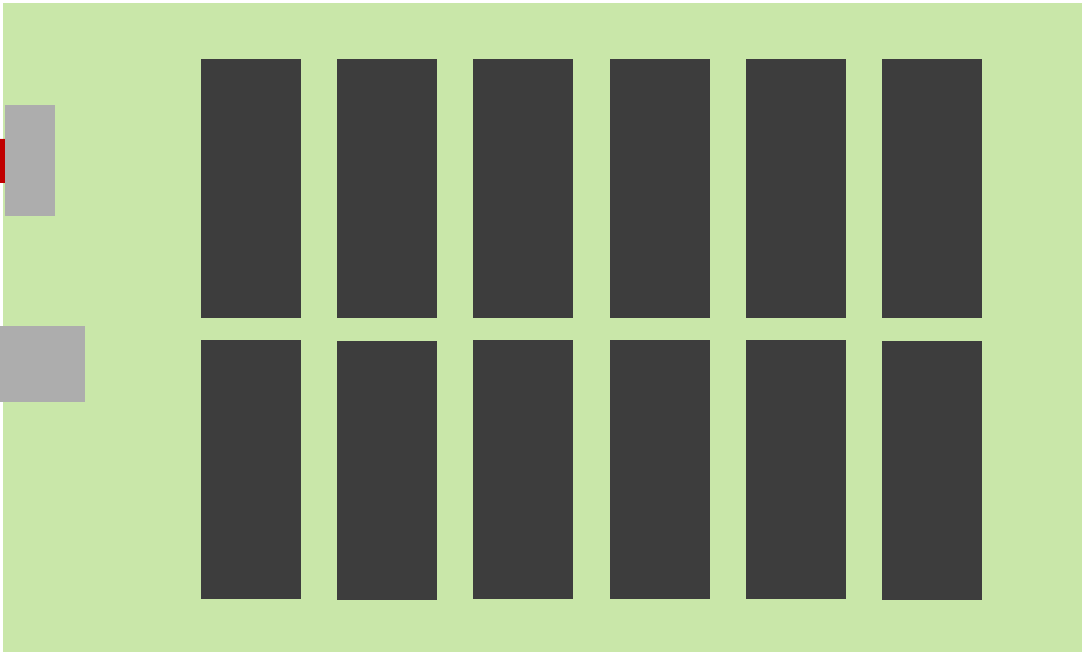
儲存至SD

讀取時鐘模組

WiFi匯出資料

DIGITAL		ANALOG		SERIAL			POWER	
IN	OUT	IN	PWM	TTL	I2C	SPI	VCC	GND
	PIN 43						V主板	GND
			PIN 45				V主板	GND
PIN33							V主板	GND
PIN42		A14					V主板	GND
PIN44		A15					V主板	GND
PIN30		A12					V主板	GND
	PIN31						V擴	GND

HOST_PCB (TOP面)



PIN definitions:

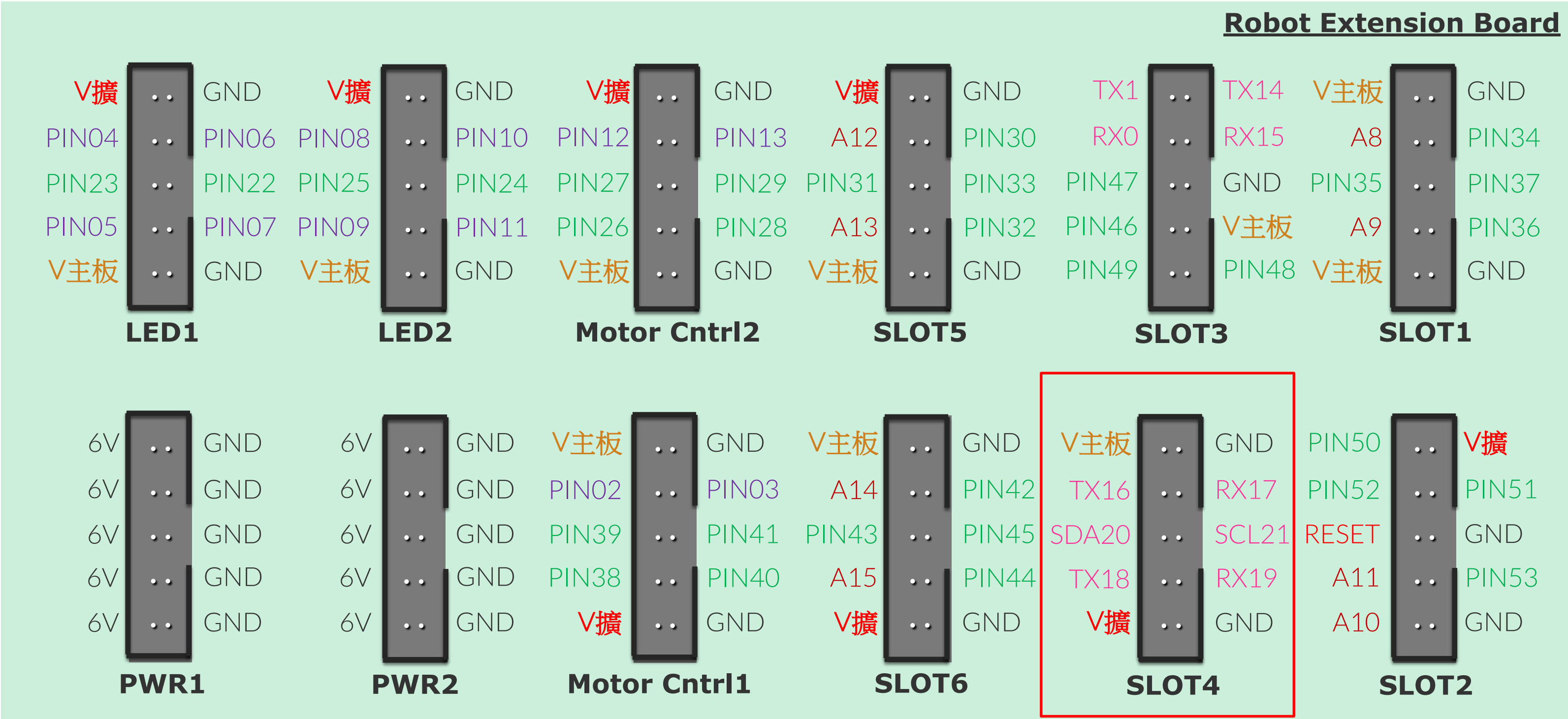
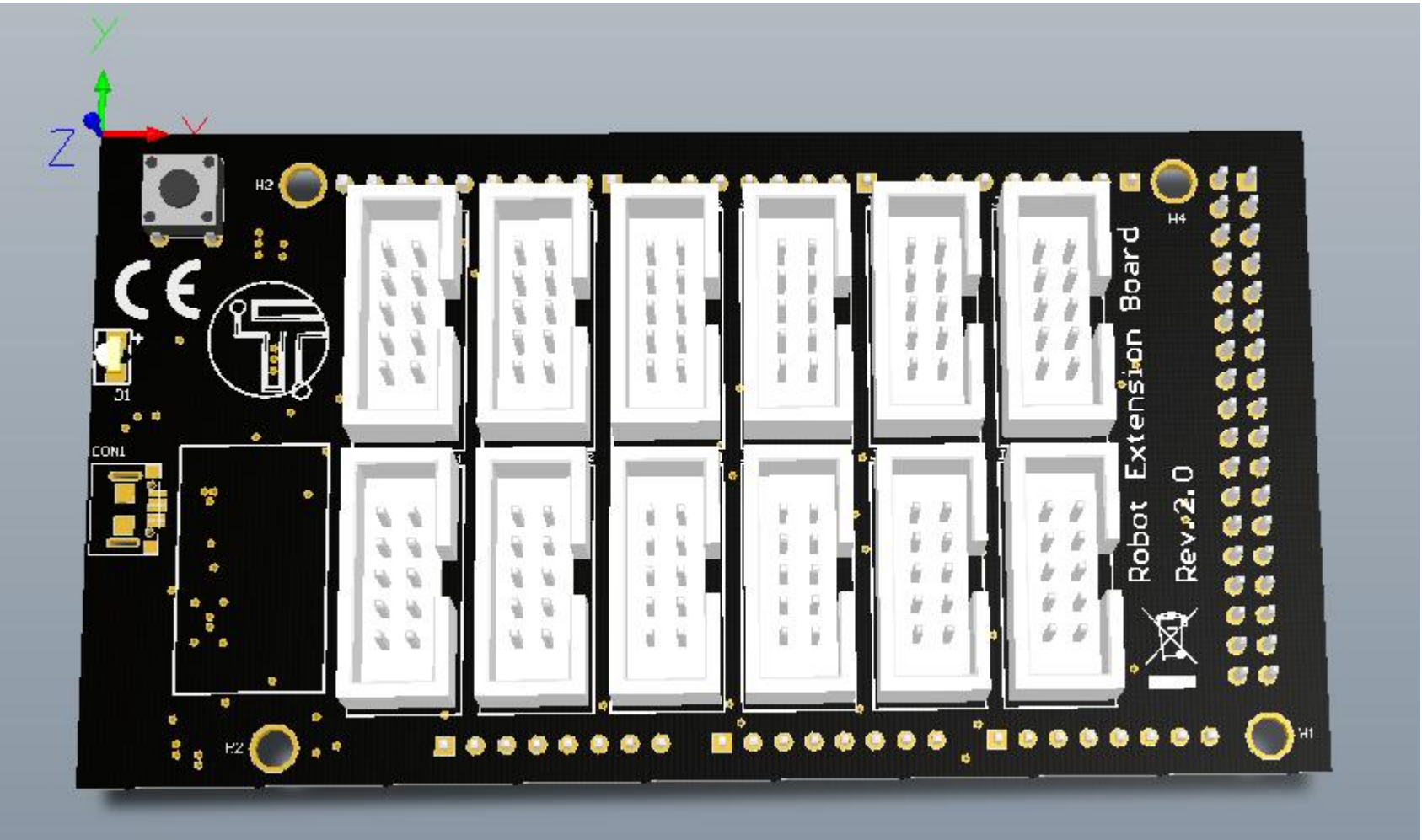
PWM PIN: 2~13

Digital PIN: 22~49

Analog PIN: A8~A15

TTL/SDI: 14~21

V主板/GND



1. WiFi傳送Camera /Video stream給device

Mega2560	ESP32
TX1 (PIN18)	PIN3
RX1 (PIN19)	PIN1
5V	5V
GND	GND

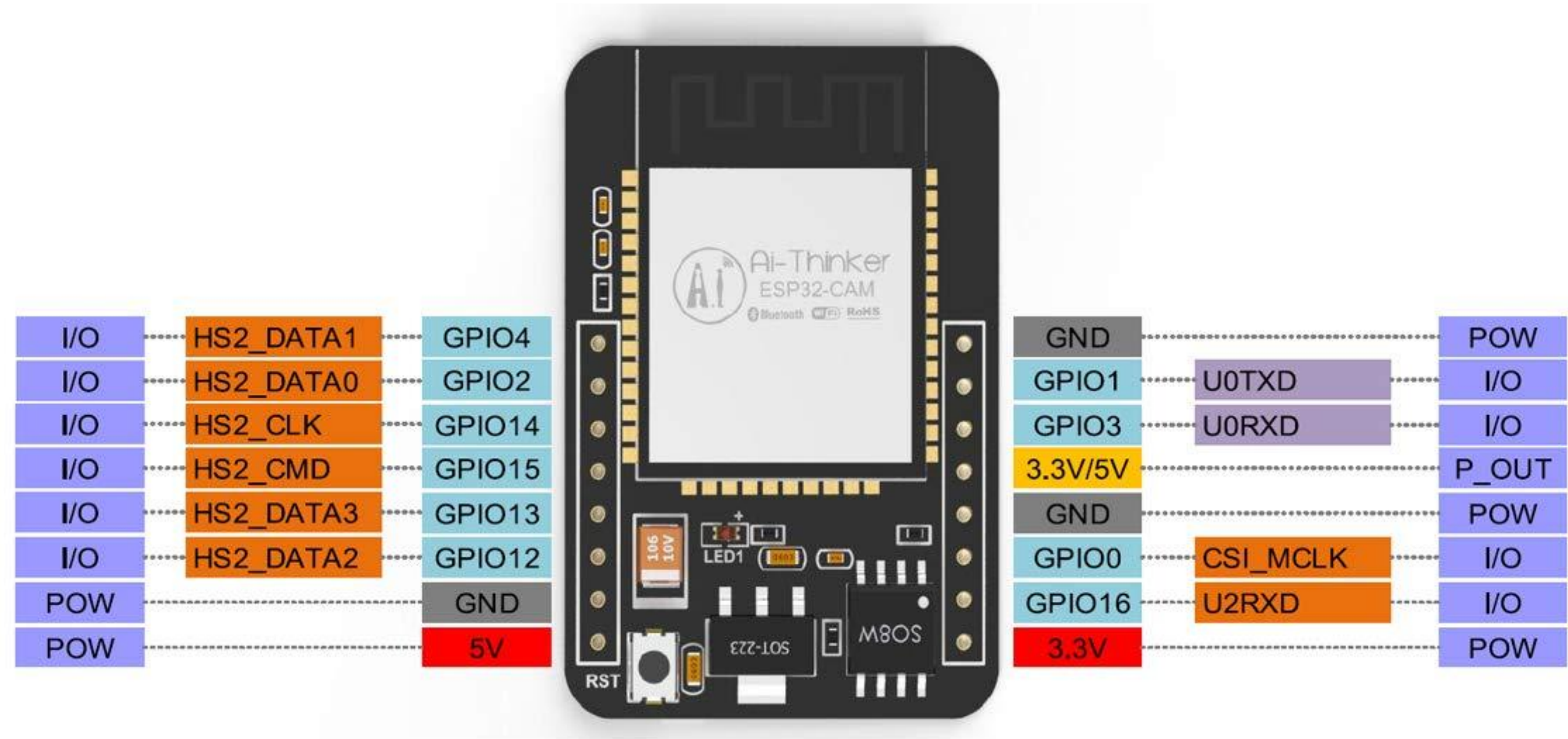
UART
channel

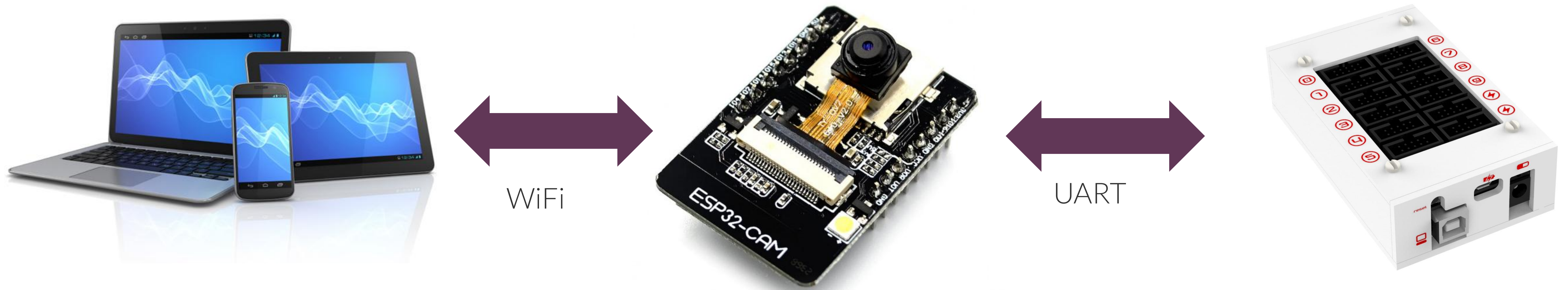
CAM	ESP32	SD	ESP32
D0	PIN5	CLK	PIN14
D1	PIN18	CMD	PIN15
D2	PIN19	DATA0	PIN2
D3	PIN21	DATA1/闪光灯	PIN4
D4	PIN36	DATA2	PIN12
D5	PIN39	DATA3	PIN13
D6	PIN34		
D7	PIN35		
XCLK	PIN0		
PCLK	PIN22		
VSYNC	PIN25		
HREF	PIN23		
SDA	PIN26		
SCL	PIN27		
POWER PIN	PIN32		

SPI
channel

I2C
channel

Pin Definition





Dashboard App on device

Phase1 資料圖形化Dashboard

1. 即時數據資料包圖形匯整
2. 即時影像監控
3. 功能鈕可以開關主板元件

Phase2 上Internet與網頁連結

1. 將感測表單傳送至網頁後台
2. 將影像傳送至網頁後台
3. 可以從網頁實現dashboard

Phase3 資料流VP

1. 網頁上dataflow介面可以編輯感測數據資料包資料流邏輯

ESP32燒錄固定的Code

1. WiFi傳送
 - a. Camera 照片/Video stream給Device
 - b. SD Card儲存的感測數據資料包
2. WiFi連線至Router設定: 用TarkusVP透過TXRX設定教室網路的WiFi帳號密碼
3. 感測資料封包固定格式
4. 可由Device傳指令至主板:
 - a. 功能元件動作: 開始澆水/開燈
 - b. 讀取感測元件單一數值
5. 可由主板傳感測資料至ESP32
6. 可由ESP32傳感測資料至Device

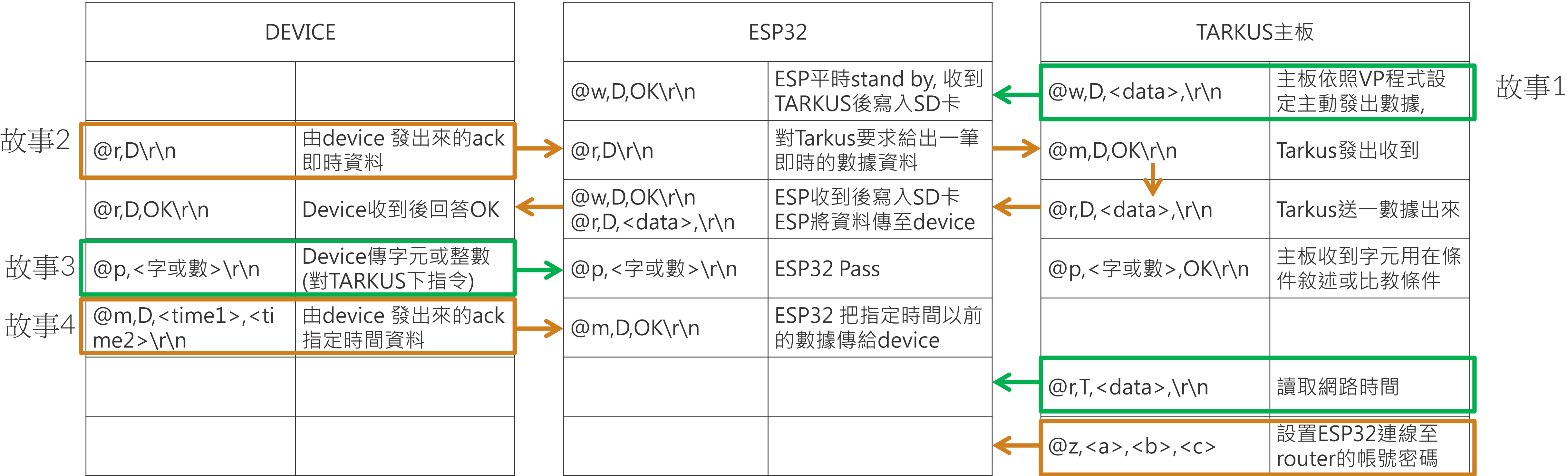
TARKUS主板

1. UART(TX1)送感測數據資料包給ESP
2. 收ESP指令更改功能元件動作

感測數據包格式

@w,D,1,1,1,1,1,1,1\r\n

指令



感測數據包格式

@w, D, 1, 1, 1, 1, 1, 1, 1\r\n

封包

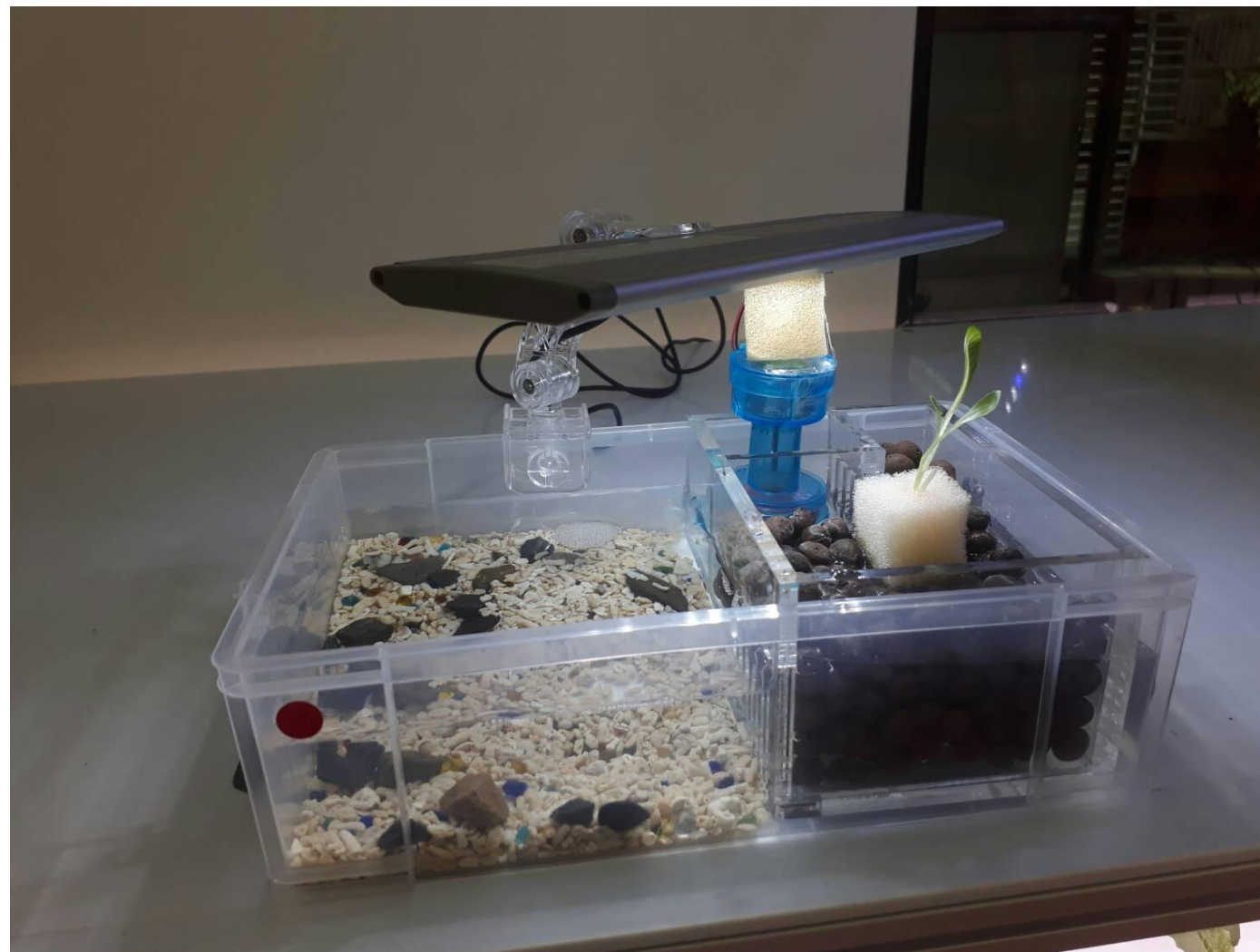
1st	2nd	3rd	4th	5th	6th	7th	8th	9th	10th
時間	燈stat 0/1	風扇stat 0/1	抽水馬達stat 0/1	土壤感測器stat 0/1	雨滴感測器stat 0/1	光感測器stat 0/1	電量PIN1stat 0/1	電量PIN2stat 0/1	未知device stat 0/1
11th	12th	13th	14th	15th	16th	17th	18th	19th	20th
未知device stat 0/1	未知device stat 0/1	未知device stat 0/1	未知device stat 0/1	空氣溫度 0~1023	空氣溼度 0~1023	土壤濕度 0~1023	雨滴感測 0~1023	光感測 0~1023	未知感測器 0~1023
21th	22th	23th	24th	25th	26th	27th	28th	29th	30th
未知感測器 0~1023	未知感測器 0~1023	未知感測器 0~1023	未知感測器 0~1023	未知感測器 0~1023	未知感測器 0~1023	未知感測器 0~1023	@P,<字元或整數> 的內容	Wifi連線 router的名稱	Wifi連線 router的帳號
31th	32th								
Wifi連線 router的密碼	ESP32的 MACaddress								

A group of students in a classroom are seated at desks, working on laptops. The students are focused on their screens, and the classroom environment is visible in the background. The image has a slightly dim, warm-toned aesthetic.

未來延伸模組

Tarkustech Innovations Co. Ltd. Copyright
from 2017

功能模組探討：結合綸堡國際魚菜共生



Power Board

電扇
抽水幫浦
USB植物生長燈
電池電量偵測

Sensor Board

空氣溫濕度 dht11
土壤濕度感測器soil
光敏電阻模組
水位感測器
(雨滴感測器)
RGB指示燈/蜂鳴器模組

IOT Board

ESP32+CAM
物聯網/雲端網頁監控App
時鐘模組
SD卡存儲模組
電池電量模組

農場基礎套件

彩色警示燈模組

蜂鳴器模組

風扇模組

水幫浦模組

植物生長燈模組

溫濕度感測模組

土壤濕感測模組

光線感測模組

水位感測模組

雨滴感測模組

雲端網頁監控App

WiFi讀取模組

影像監控模組

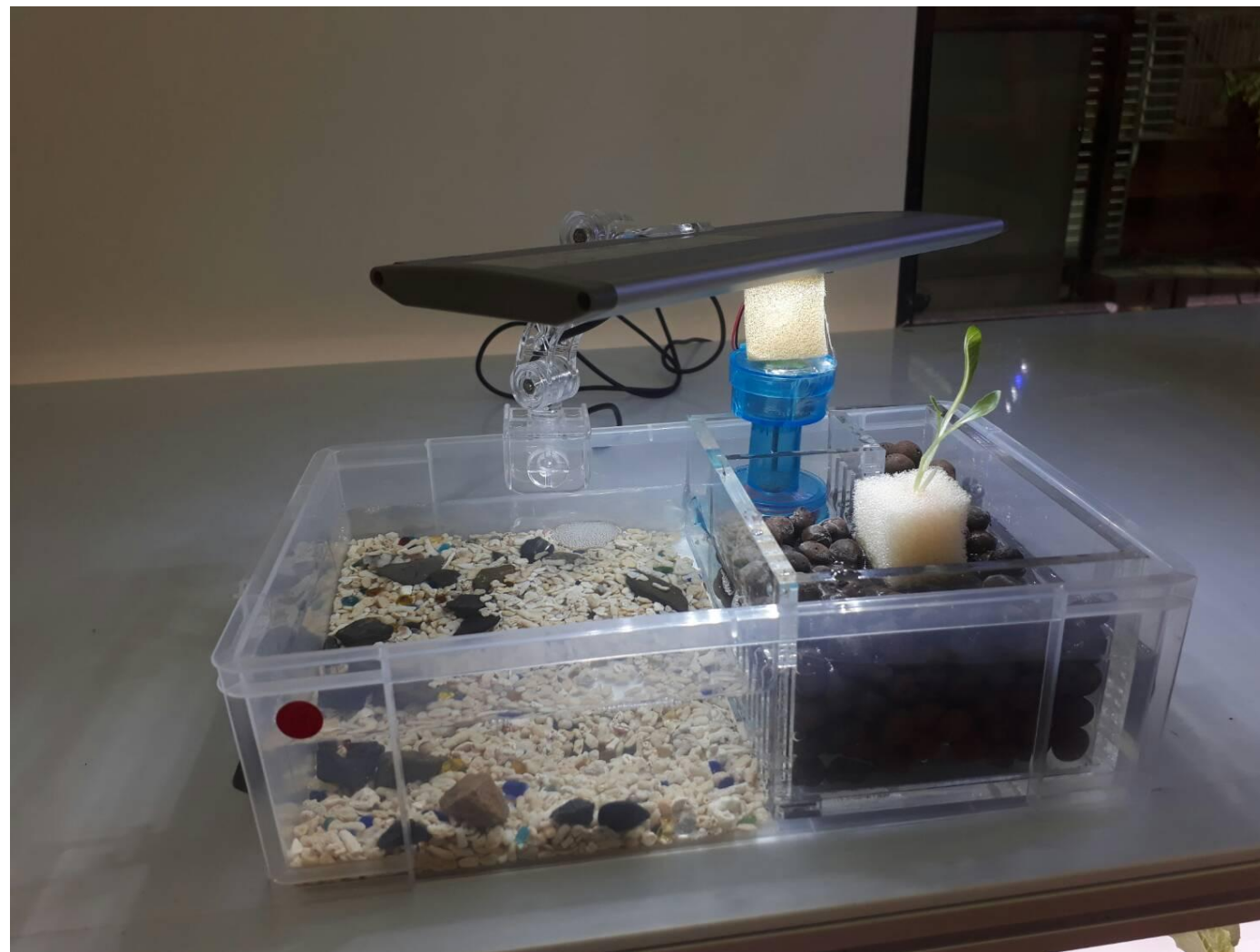
數據儲存模組

時鐘模組

WiFi匯出資料

電池電量模組

功能模組探討：結合綸堡國際魚菜共生



Sensor Board2

土壤氮磷鉀模組
土壤鹽鹼度模組
土壤PH值模組
七段數位顯示器模組
按鈕模組

Power Board2

定時飼料

Sensor Board3

水中溫度感測ds18b20
水溶氧量
PH值
TDS感測器
EC感測器

農場延伸套件

雲端網頁監控App

土壤氮磷鉀模組

土壤鹽鹼度模組

土壤PH值模組

數位顯示模組

按鈕模組

魚菜共生套件

室內電源模組

打氣幫浦模組

定時飼料模組

水溫感測模組

水質延伸套件

雲端網頁監控App

溶氧量感測模組

PH值感測模組

TDS感測模組

EC感測模組

