



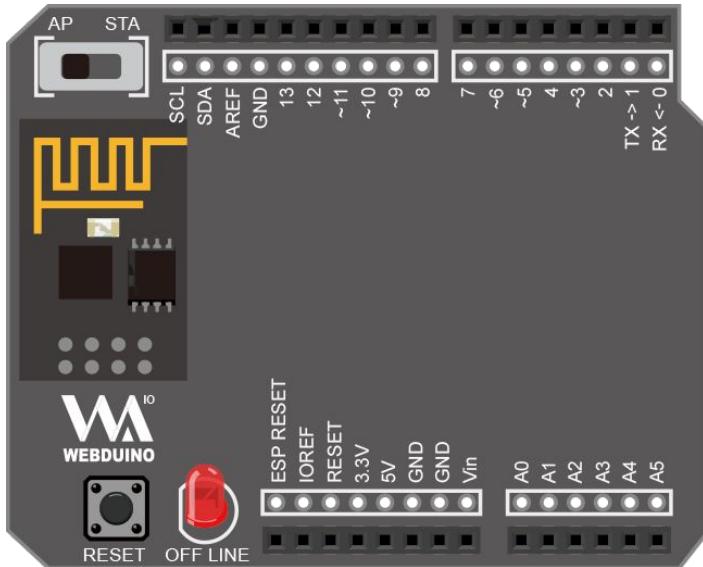
# Webduino Blockly

## 自訂積木模組開發

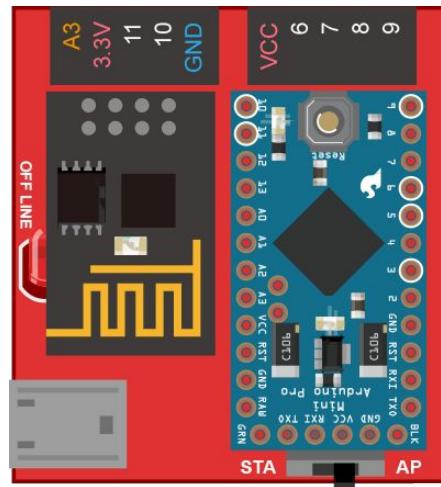
# 大綱

- Webduino 開發板介紹
- 介紹 Google Blockly Factory
- 積木擴充實作

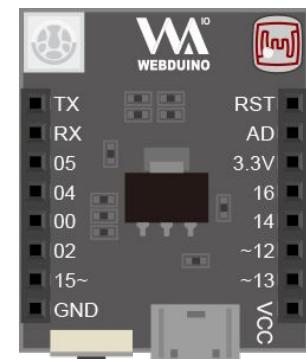
# Webduino 開發板系列



# Webduino Fly ( Arduino UNO 擴充板 )



Webduino 馬克1號



# Webduino Smart

# 開發板支援傳感器數量

51 種

17

種，不用驅動程式，使用 I/O 程式積木

16

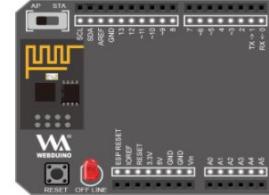
種，不用驅動程式，有提供程式積木

18

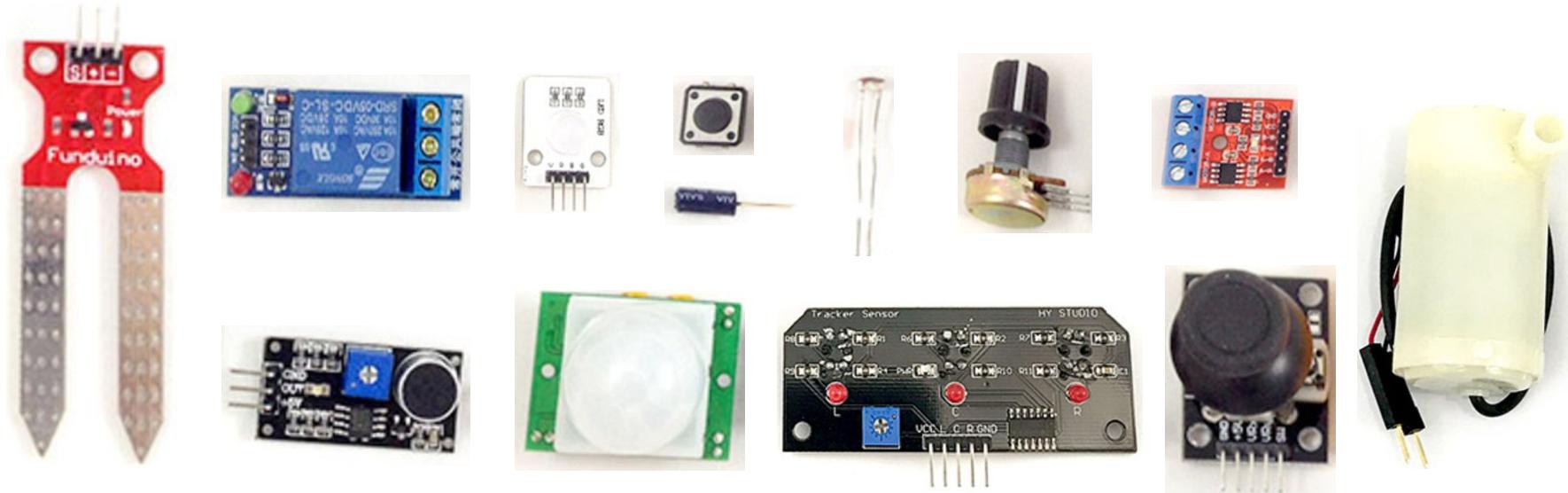
種，需要驅動程式，有提供程式積木

開源並且可自行擴充傳感器

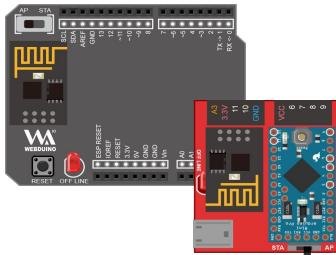
# 各種開發板的差異

	Smart	馬克一號	Fly
開發板長相			
I/O積木支援傳感器	17	17	17
客製化積木支援傳感器	16	16	16
需驅動程式支援傳感器	11	14	17
提供 Arduino 程式庫	✗	✓	✓
支援 Arduino IDE	✗	✓	✓
類比腳	1 (已使用)	1 (未使用)	6 (未使用)
教學範例數量	★★☆☆☆	★★★★★	★★★★★

# 不需要韌體的傳感器 x 13 (開發板都支援,除了搖桿)



# 需要韌體的傳感器 x 15



蜂鳴器



溫濕度



超音波



點距陣



MP3



液晶螢幕



PM2.5



秤重HX711

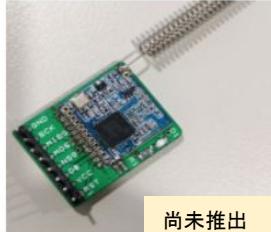


全彩矩陣



尚未推出

LORA



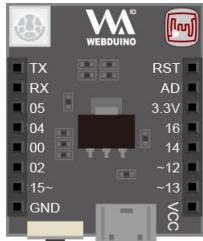
尚未推出

錄紅外線



# Smart 支援傳感器

按按鈕更新



蜂鳴器



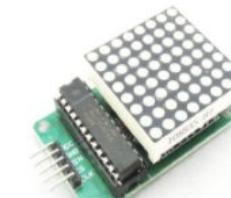
溫濕度



超音波



點距陣



註冊版增加  
5種

PM2.5



# 使用 Webduino開發板優點

- 最容易串接網頁、豐富教學、支援行動裝置、元件多樣化

開發板	發光元件	輸入輸出元件	環境偵測元件	網頁互動
- 開發板	- LED	- 紅外線發射	- 超音波	- 文字
- I/O 腳位	- 三色 LED	- 接收	- 溫濕度	- 圖片
- 資料傳輸	- LED 點矩陣	- 冷氣紅外線	- 光敏電阻	- 按鈕
<b>進階功能</b>	- OLED	- 按鈕開關	- 土壤濕度	- 顏色
- 控制台	- LCD	- 震動開關	- 人體紅外線	- 拉霸
- 等待	- 全彩燈條	- 繼電器	- 聲音偵測	- 遙控器
- 數值轉換	<b>無線感應元件</b>	- 搖桿	- 三軸感測	- 圖表
- 鍵盤行為	- RFID	- 可變電阻	- 細懸浮微粒	- Youtube
- 語音聲控	- LORA	- 蜂鳴器		- 影像追蹤
- 語音朗讀	- GPS	- MP3 播放器		- Google 地圖
- 行動裝置		- 電子磅秤		

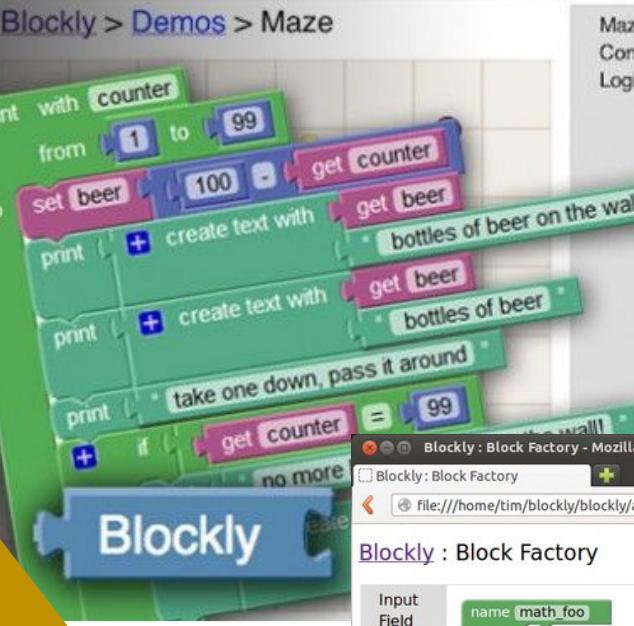
# WEBDUINO 開發平台特色

命令種類 (hex)	傳感器	型態
04 01	OLED	輸出
04 02	保留勿用	--
04 03	保留勿用	--
04 04	溫濕度 dht.h	輸入
04 05	保留勿用	--
04 06	保留勿用	--
04 07	蜂鳴器	輸出
04 08	點距陣	輸出
04 09	紅外線發射	輸出
04 0A	紅外線接收	輸入
04 0B	三軸	輸入
04 0C	GPS	輸入
04 0D	保留勿用	--
04 0F	RFID	輸入
04 10	G3 (PM2.5)	輸入
04 11	保留勿用	--
04 12	保留勿用	--
04 13	保留勿用	--
04 14	Joypad (搖桿)	輸入
04 15	HX711 (電子磅秤)	輸入
04 16	UART 通訊	輸入 / 輸出
04 17	循跡自走車	輸入 / 輸出
04 18	LCD1602	輸出
04 19	MP3 DFPlayer Mini	輸入 / 輸出
04 20	字串傳輸 (非傳感器)	輸入 / 輸出

# 積木工廠

## 用積木定義積木

Blockly > Demos > Maze



Maze Control Logic

- not wall ahead
- repeat while do move forward
- + if then and
- turn left
- turn right
- true
- repeat while do

Blockly : Block Factory - Mozilla Firefox

Blockly : Block Factory

file:///home/tim/blockly/blockly/apps/blockfactory/index.h

Blockly : Block Factory

Preview: LTR

Input Field Type Colour

name math\_foo  
inputs external inputs no connections colour

Language code:

```
Blockly.Blocks['math_foo'] = {
  init: function() {
    this.setHelpUrl('http://www.example.com/');
    this.setToolTip('');
  }
};
```

Generator stub: JavaScript

```
Blockly.JavaScript['math_foo'] = function(block) {
  // TODO: Assemble JavaScript into code variable.
  var code = '...';
  return code;
};
```

# 積木定義

LCD (1602) • SDA 10 ▾ SCL 11 ▾

```
1 Blockly.Blocks['lcd1602_new'] = {
2   init: function () {
3     this.appendDummyInput()
4       .appendField(
5         Blockly.Msg.WEBDUINO_LCD1602, "LCD 點陣液晶螢幕，SDA")
6       .appendField(new Blockly.FieldDropdown([
7         ["10", "10"],
8         ["11", "11"]
9       ]), "sda_")
10      .appendField(Blockly.Msg.WEBDUINO_LCD1602_SCL, " SCL")
11      .appendField(new Blockly.FieldDropdown([
12        ["10", "10"],
13        ["11", "11"]
14      ]), "scl_");
15      this.setOutput(true);
16      this.setColour(230);
17      this.setToolTip('');
18      this.setHelpUrl('https://webduino.io');
19    }
20  };
```

# 積木產生程式碼

LCD (1602) • SDA 10 ▾ SCL 11 ▾

```
1 Blockly.JavaScript['lcd1602_new'] = function (block) {  
2     var dropdown_sda_ = block.getFieldValue('sda_');  
3     var dropdown_scl_ = block.getFieldValue('scl_');  
4     var code =  
5         'getLCD1602(board,' +  
6             dropdown_sda_ + ',' + dropdown_scl_ + ')';  
7     return [code, Blockly.JavaScript.ORDER_FUNCTION_CALL];  
8 };
```

# Google Blockly Factory 連結

The screenshot shows the Google Blockly Factory interface. On the left, there's a sidebar with categories: Input, Field, Type, and Colour. In the main area, a block named "block\_type" is selected. The block has the following properties:

- Inputs: automatic, no connections
- tooltip: ““”
- help url: ““”
- colour: hue: 230

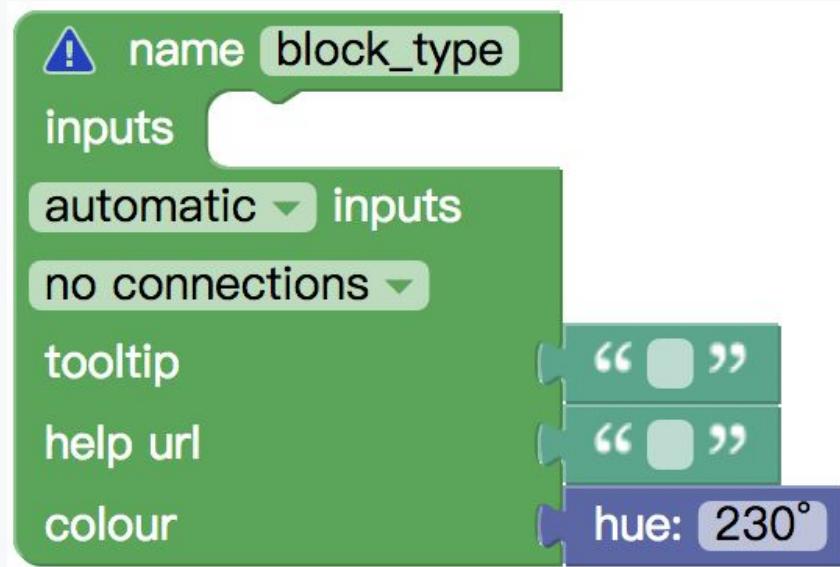
On the right, there are two sections:

- Block Definition:** (JavaScript) 

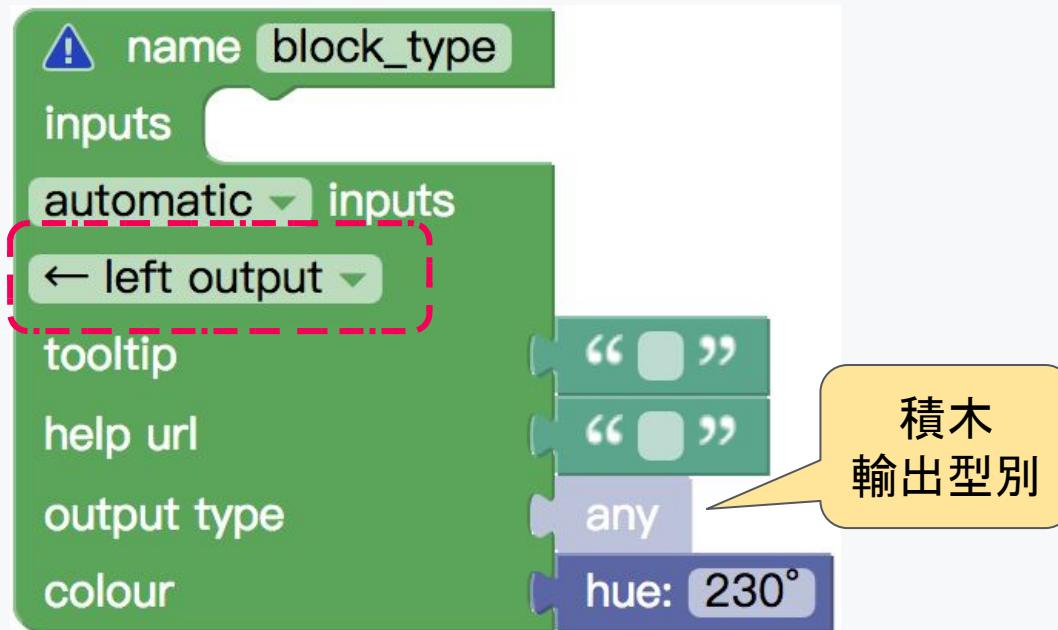
```
Blockly.Blocks['block_type'] = {
  init: function() {
    this.setColour(230);
    this.setTooltip('');
    this.setHelpUrl('');
  }
};
```
- Generator stub:** (JavaScript) 

```
Blockly.JavaScript['block_type'] = function(block) {
  // TODO: Assemble JavaScript into code variable.
  var code = ...;\n  return code;
};
```

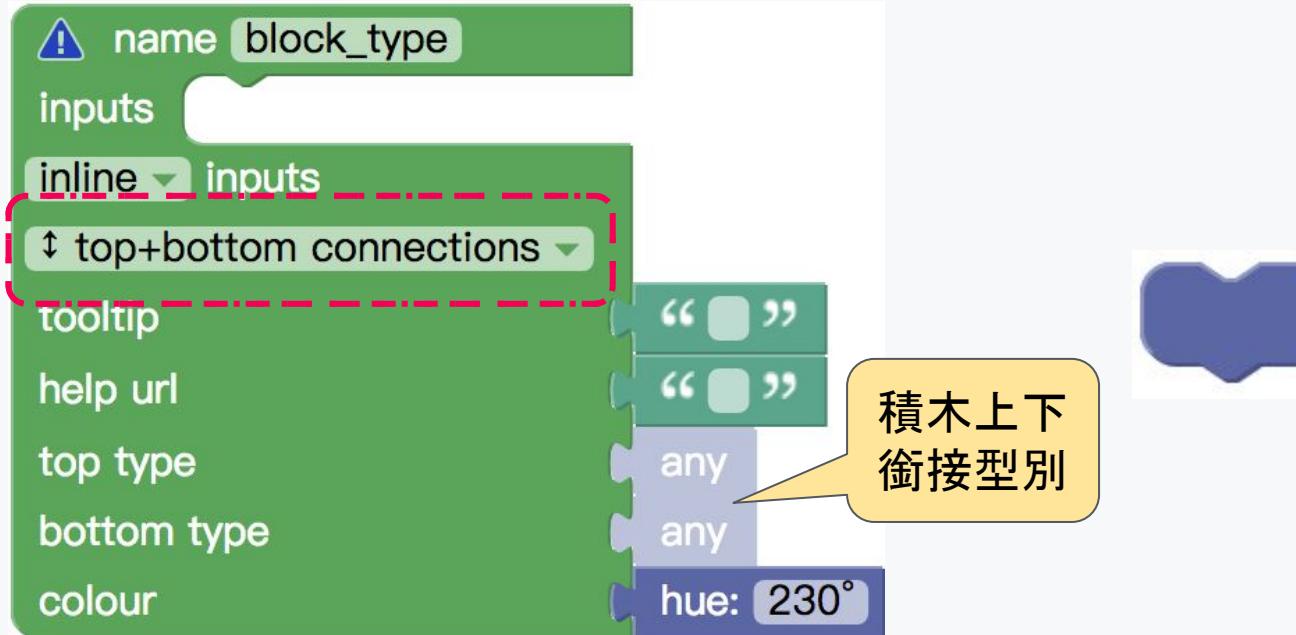
# Google Blockly Factory



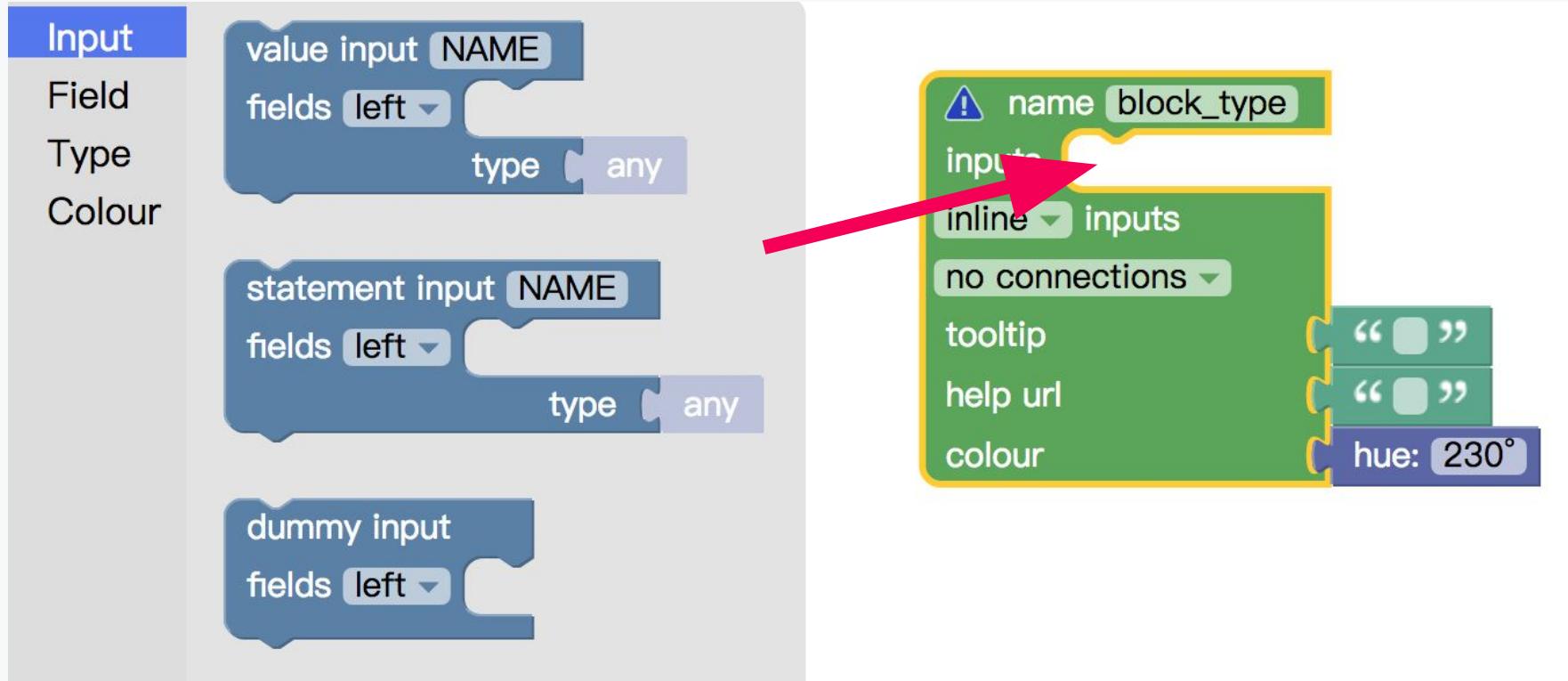
# Google Blockly Factory



# Google Blockly Factory



# Google Blockly Factory



# 單一實例積木

開發板 馬克1號 ▾ 使用 Wi-Fi ▾ 連線至 “ ” 類比取樣 250 ms ▾ 串聯 ■ 協同控制 ■



# 設定積木

LED 燈，腳位 10



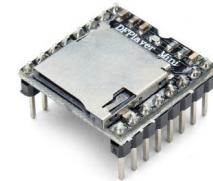
MP3 ( DFPlayer )，Rx 10 Tx 11



全彩點距陣，腳位 10 ，LED數量 16 顆



紅外線發射，腳位 9



LCD (1602) , SDA 10 SCL 11



電子秤，SCK 6 DT 7



# 設定積木變數

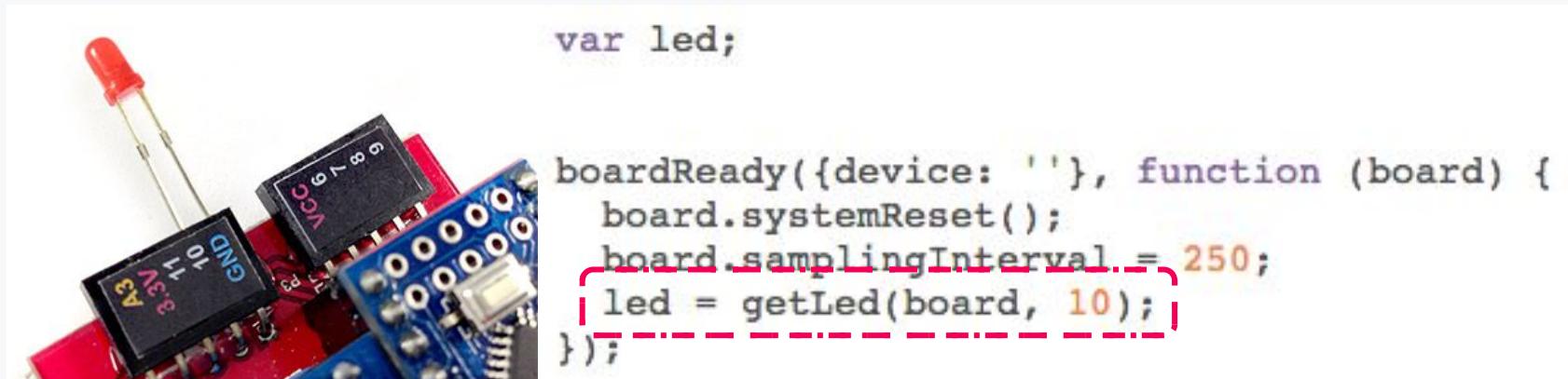
設定 led 為 LED 燈，腳位 10



```
var led;
```

```
led = getLed(board, 10);
```

# 點亮 LED 燈



# 動作積木



led 設定狀態 on

dfplayer MP3 DFPlayer 開始播放

點亮 LED 燈後  
開始播歌

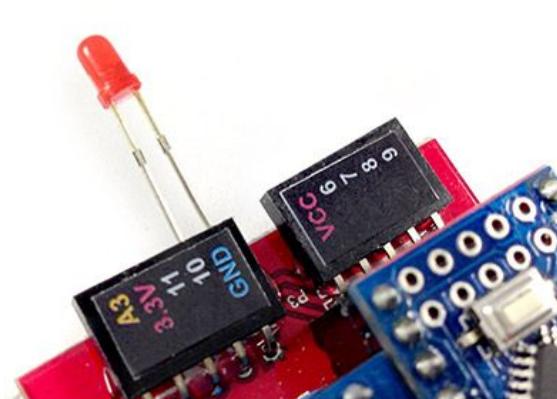


# 點亮 LED 燈

開發板 馬克1號 或 Fly ▾ 使用 Wi-Fi ▾ 連線至 “ ” 類比取樣 250 ms ▾ 串聯 協同控制

設定 led 為 LED 燈，腳位 10

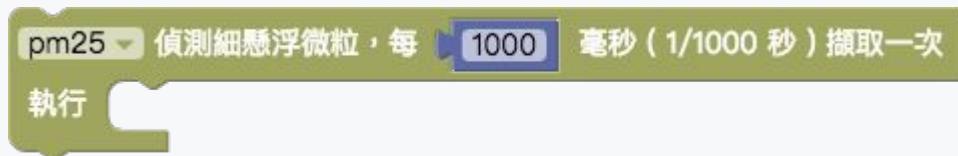
led 設定狀態 on



```
var led;

boardReady({device: ''}, function (board) {
    board.systemReset();
    board.samplingInterval = 250;
    led = getLed(board, 10);
    led.on();
});
```

# 動作積木 + Function Callback



```
var pm25;  
  
pm25.read(function(evt){  
  
}, 1000);
```



土壤濕度  
傳感器



空氣溫濕度  
傳感器

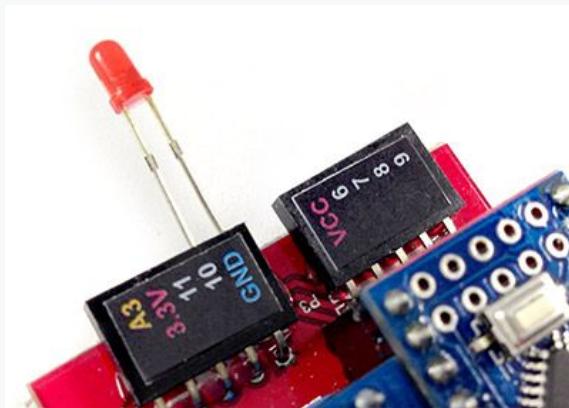
# 點亮 LED 燈後執行

開發板 馬克1號 或 Fly ▾ 使用 Wi-Fi ▾ 連線至 “ ” 類比取樣 250 ms ▾ 串聯 協同控制

設定 led 為 LED 燈，腳位 10

led 設定狀態 on

接著執行



```
boardReady({device: ''}, function (board) {  
    board.systemReset();  
    board.samplingInterval = 250;  
    led = getLed(board, 10);  
    led.on(function(){  
        // Your code here  
    });  
});
```

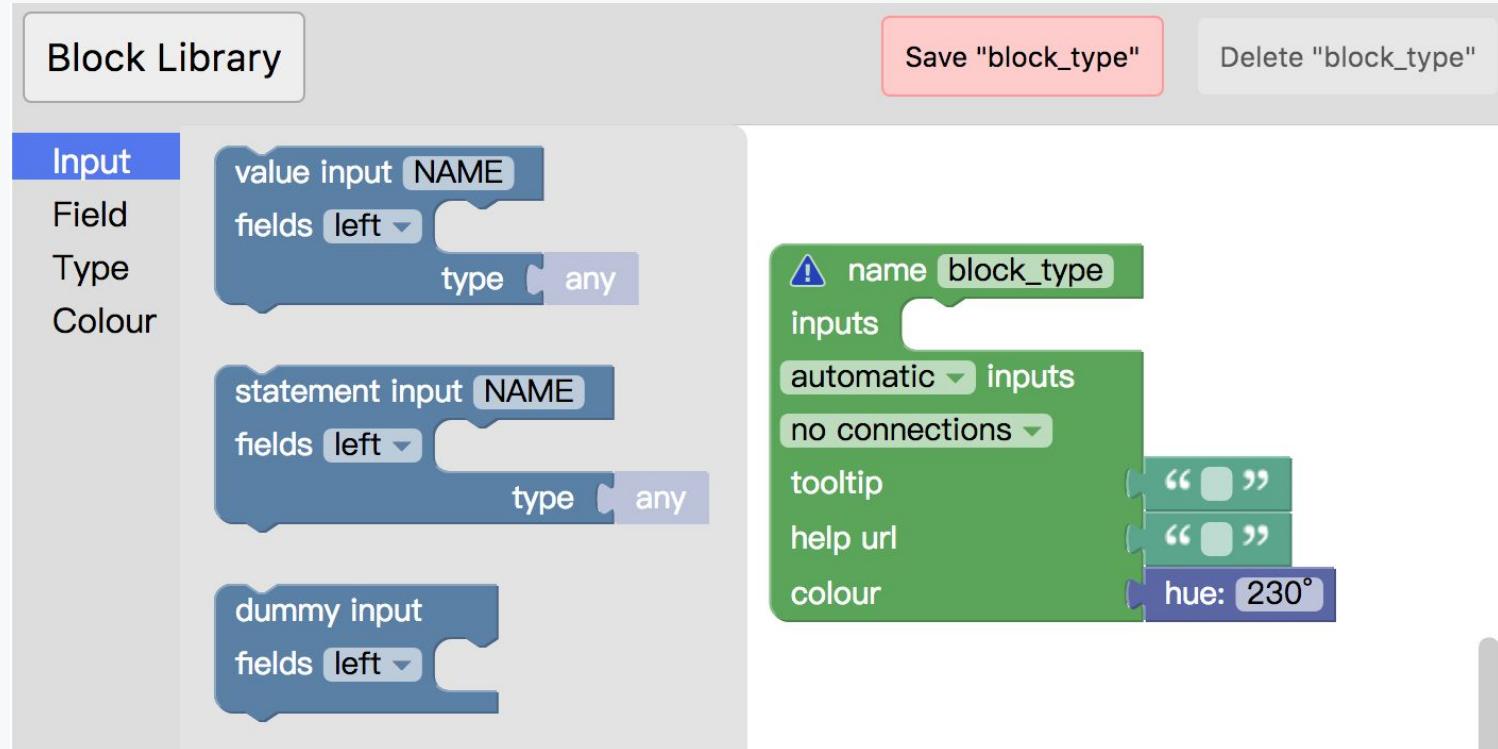
# 多重輸入積木



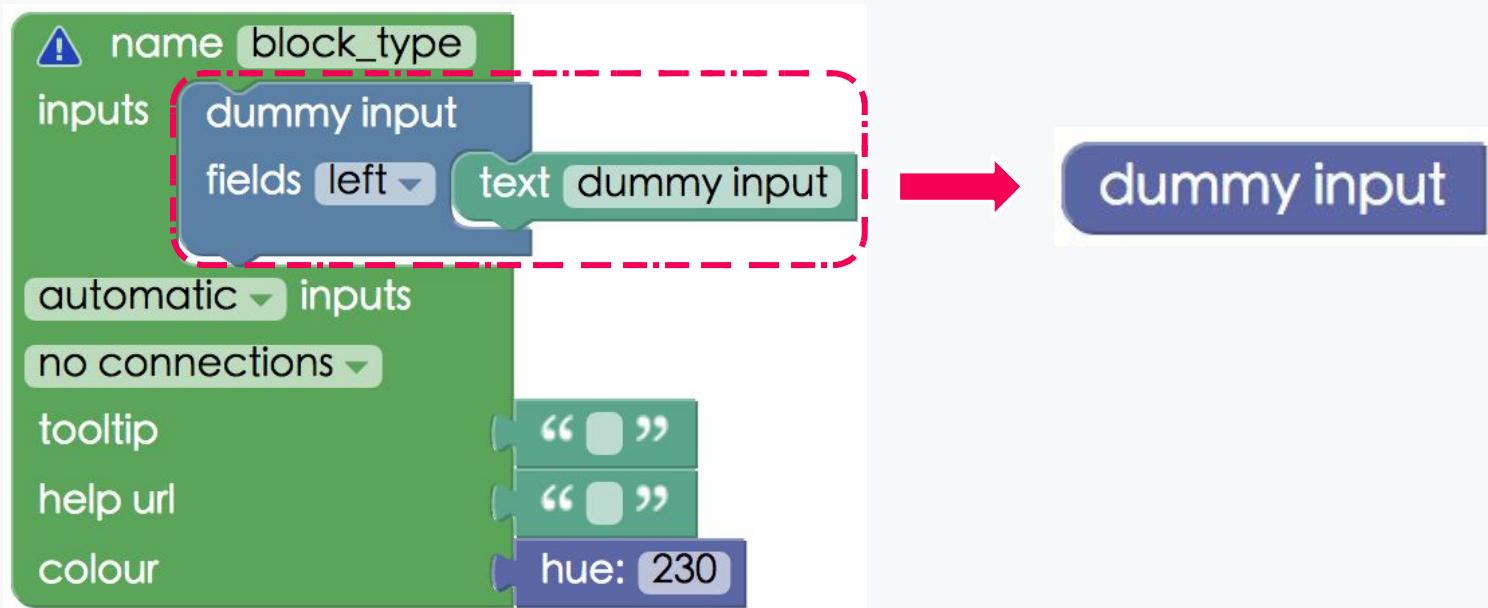
```
function colour_blend(c1, c2, ratio) {
    ratio = Math.max(Math.min(Number(ratio), 1), 0);
    var r1 = parseInt(c1.substring(1, 3), 16);
    var g1 = parseInt(c1.substring(3, 5), 16);
    var b1 = parseInt(c1.substring(5, 7), 16);
    var r2 = parseInt(c2.substring(1, 3), 16);
    var g2 = parseInt(c2.substring(3, 5), 16);
    var b2 = parseInt(c2.substring(5, 7), 16);
    var r = Math.round(r1 * (1 - ratio) + r2 * ratio);
    var g = Math.round(g1 * (1 - ratio) + g2 * ratio);
    var b = Math.round(b1 * (1 - ratio) + b2 * ratio);
    r = ('0' + (r || 0).toString(16)).slice(-2);
    g = ('0' + (g || 0).toString(16)).slice(-2);
    b = ('0' + (b || 0).toString(16)).slice(-2);
    return '#' + r + g + b;
}

colour_blend('#ff0000', '#3333ff', 0.5);
```

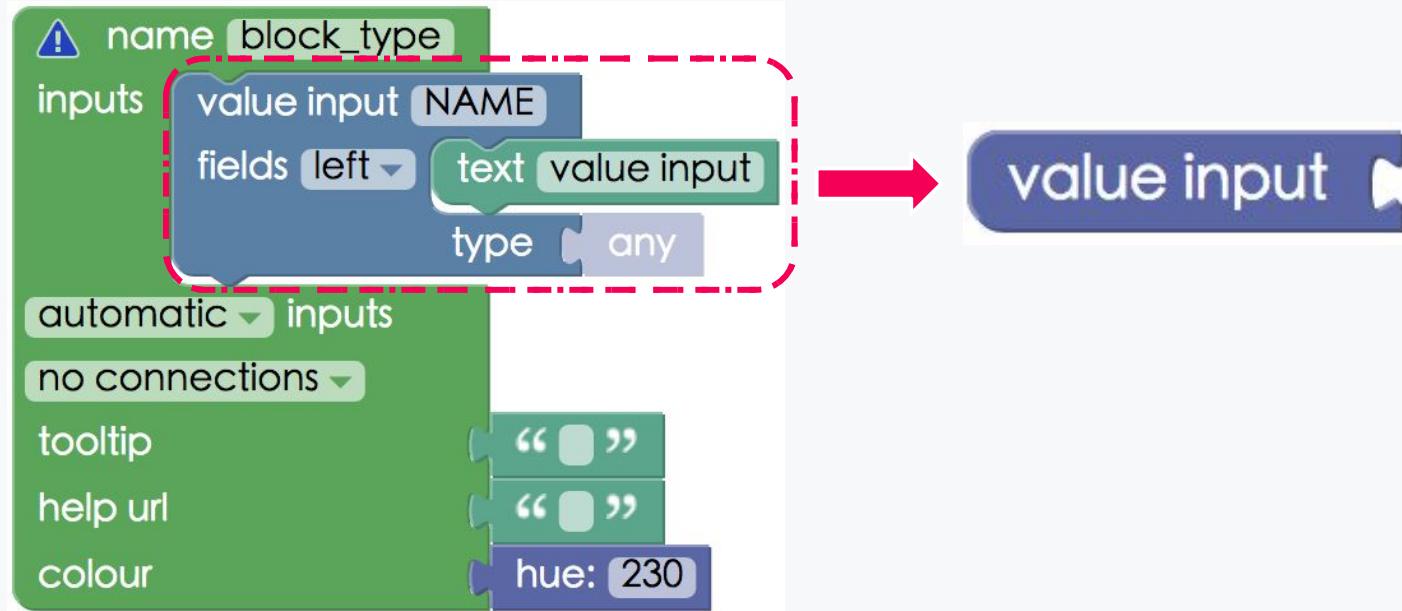
# 積木工廠 - 三種主要積木定義



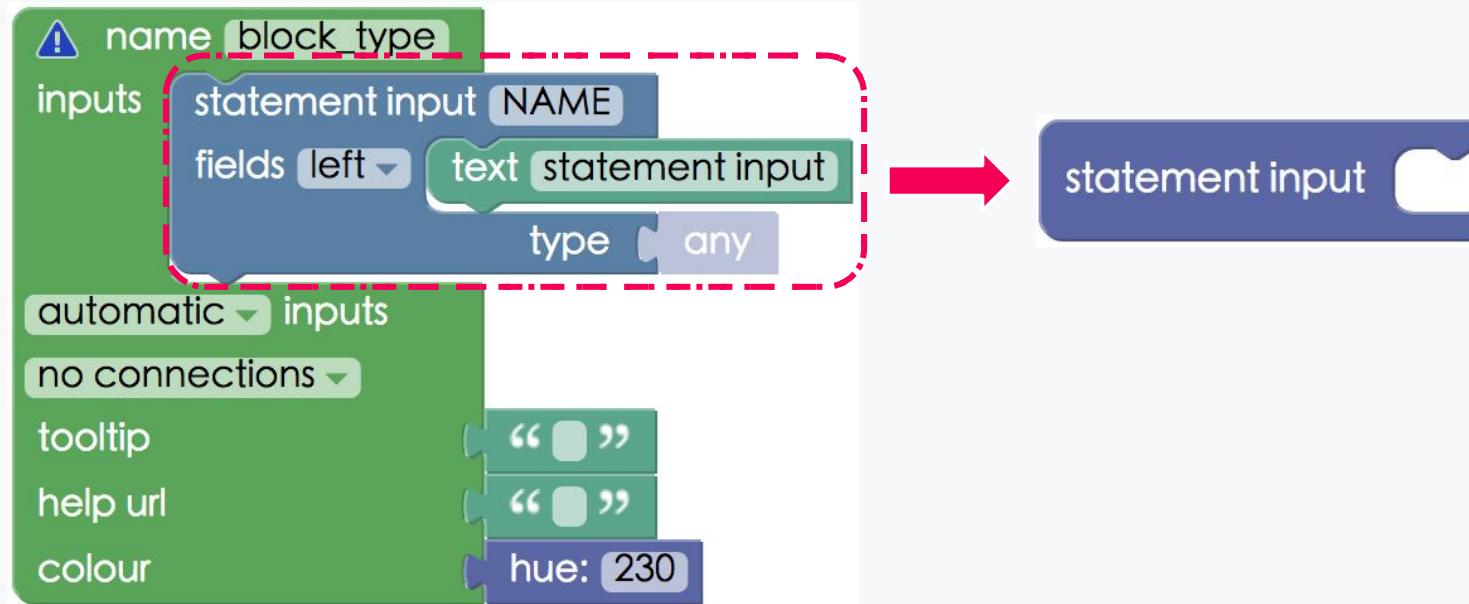
# 積木種類：dummy input



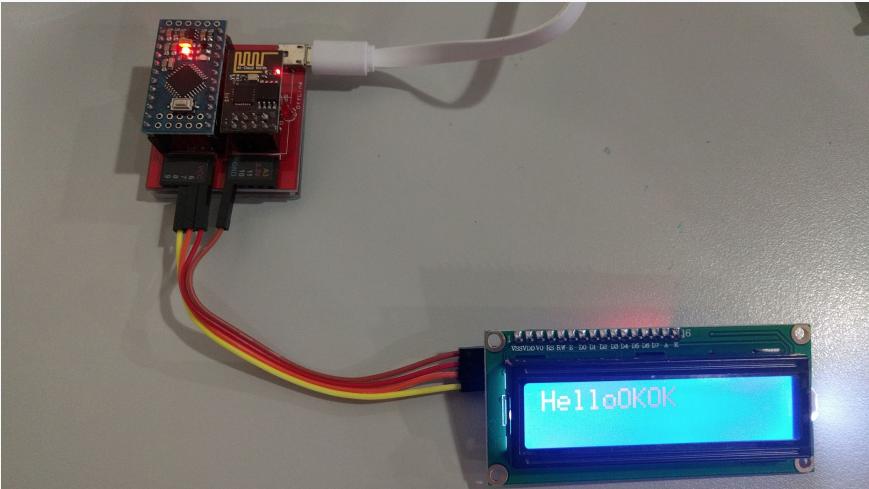
# 積木種類：value input



# 積木種類：statement input



# LCD 積木種類



# LCD 積木種類



```
toolbox.xml
<block type="variables_set">
  <field name="VAR">lcd1602</field>
  <value name="VALUE">
    <!-->
  </value>
</block>
```

```
<block type="lcd1602_new">
  <field name="sda_">4</field>
  <field name="scl_">5</field>
</block>
```

# LCD 設定積木



toolbox.xml

```
<block type="lcd1602_new">
  <field name="sda_>">10</field>
  <field name="scl_>">11</field>
</block>
```

blocks.js

```
1 Blockly.Blocks['lcd1602_new'] = {
2   init: function () {
3     this.appendDummyInput()
4       .appendField(
5         Blockly.Msg.WEBDUINO_LCD1602, "LCD 點陣液晶螢幕，SDA")
6       .appendField(new Blockly.FieldDropdown([
7         ["10", "10"],
8         ["11", "11"]
9       ]), "sda_")
10      .appendField(Blockly.Msg.WEBDUINO_LCD1602_SCL, " SCL")
11      .appendField(new Blockly.FieldDropdown([
12        ["10", "10"],
13        ["11", "11"]
14      ]), "scl_");
15     this.setOutput(true);
16     this.setColour(230);
17     this.setToolTip('');
18     this.setHelpUrl('https://webduino.io');
19   }
20};
```

# 積木組合與JavaScript程式對應關係

The screenshot shows the Weduino IDE interface. At the top, there is a pin configuration bar with the text "設定 lcd1602 為 LCD (1602) · SDA 10 SCL 11". Below this is a code editor window titled "JavaScript". The code contains two lines of JavaScript:

```
var lcd1602;  
lcd1602 = getLCD1602(board, 10, 11);
```

# LCD 積木種類

The image shows the Scratch interface with the "LCD 螢幕" category selected in the toolbox. A script is being built with the following blocks:

- Setting block: 設定 [lcd1602] 為 [LCD (1602) · SDA 10 · SCL 11]
- Control block: [LCD 顯示: “Hello”]

The "LCD 顯示" block has a red arrow pointing from it to the XML code in the toolbox. Another red arrow points from the "Hello" text in the block to the XML code.

toolbox.xml

```
<block type="lcd1602_print">
  <value name="value_">
    <block type="text">
      <field name="TEXT">Hello</field>
    </block>
  </value>
</block>
```

# LCD 顯示積木 + 字串積木



JavaScript

```
lcd1602.print('Hello');
```



# 可插拔積木定義

Blockly : Block Factory

Preview: LTR

Input Field Type Colour

name  inputs  inputs  colour

Language code:

```
Blockly.Blocks['math_foo'] = {
  init: function() {
    this.setHelpUrl('http://www.example.com/');
    this.setToolTip('');
  }
};
```

Generator stub:

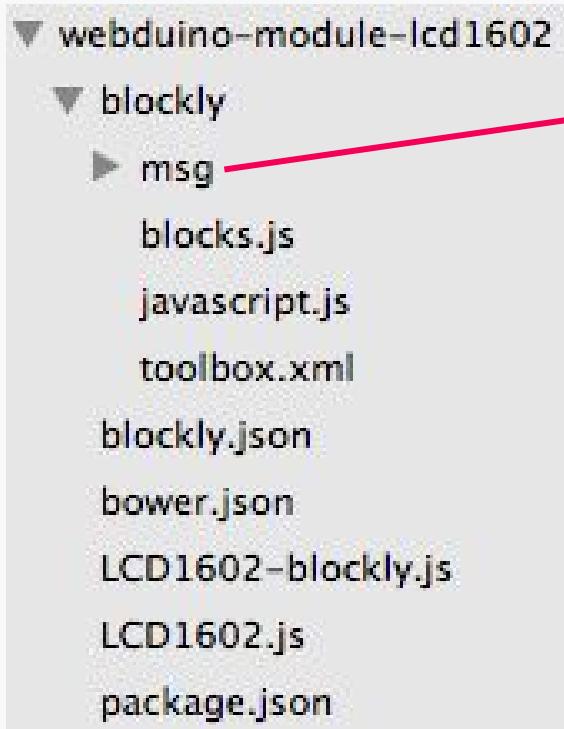
```
Blockly.JavaScript['math_foo'] = function(block) {
  // TODO: Assemble JavaScript into code variable.
  var code = '';
  return code;
};
```

# Webduino 自訂積木模組

```
▼ webduino-module-lcd1602
  ▼ blockly
    ► msg
    blocks.js
    javascript.js
    toolbox.xml
    blockly.json
    bower.json
    LCD1602-blockly.js
    LCD1602.js
    package.json
```

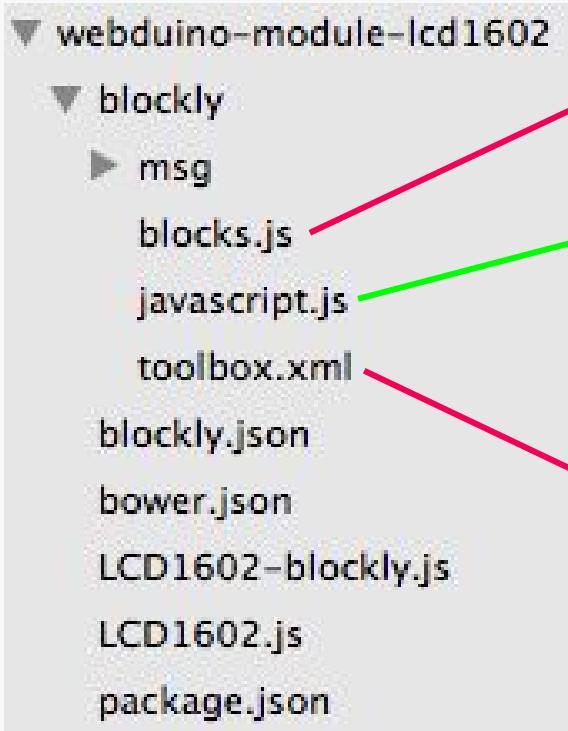
- 可定義 1 ~ n 個積木
- 撰寫積木對應的功能
- 積木模組可直接透過 URL 載入

# Webduino 自訂積木模組



積木語系定義

# Webduino 自訂積木模組



定義  
每個積木

```
Blockly.Blocks['lcd1602_print'] = {
  init: function () {
    this.appendValueInput("value_")
      .appendField(new Blockly.FieldVariable("lcd1602", "name_"))
      .appendField(Blockly.Msg.WEBDUINO_LCD1602_PRINT, "OLED 顯示:");
    this.setPreviousStatement(true);
    this.setNextStatement(true);
    this.setTooltip('');
    this.setColour(65);
    this.setHelpUrl('https://webduino.io');
  }
};
```

積木產生JavaScript程式

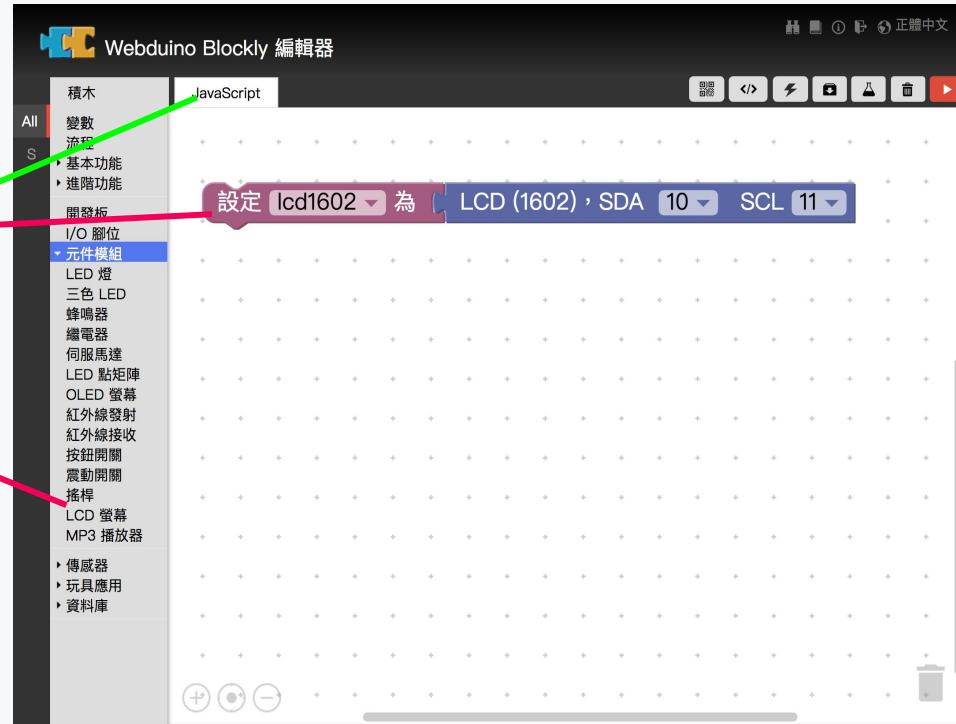
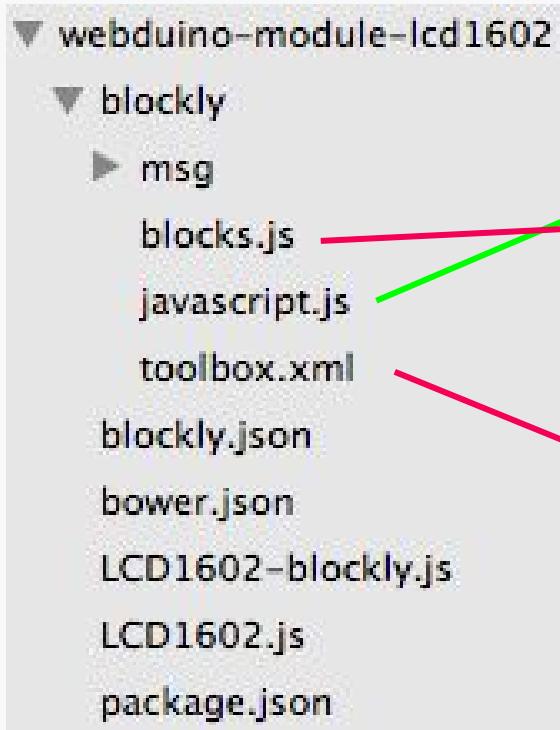
```
Blockly.JavaScript['lcd1602_new'] = function (block) {
  var dropdown_sda_ = block.getFieldValue('sda_');
  var dropdown_scl_ = block.getFieldValue('scl_');
  var code = 'getLCD1602(board,' + dropdown_sda_ + ',' + dropdown_scl_ + ')';
  return [code, Blockly.JavaScript.ORDER_FUNCTION_CALL];
};

Blockly.JavaScript['lcd1602_print'] = function (block) {
  var variable_name_ = Blockly.JavaScript.variableDB_.getName(block.getFieldValue('variable_name'));
  var value_value_ = Blockly.JavaScript.valueToCode(block, 'value_', Blockly.JavaScript.ORDER_ASSIGNMENT);
  var code = variable_name_ + '.print(' + value_value_ + ')';
  return code;
};
```

積木選單定義檔



# Webduino 自訂積木模組



# Webduino 自訂積木模組

```
▼ webduino-module-lcd1602
  ▼ blockly
    ► msg
      blocks.js
      javascript.js
      toolbox.xml
      blockly.json
      bower.json
      LCD1602-blockly.js
      LCD1602.js
      package.json
```

Blockly > Demos > Block Factory

Preview: LTR

Input  
Field  
Type  
Colour

name block\_type  
inputs value input NAME  
fields left text 1234  
text input default , NAME  
type any  
automatic Inputs  
bottom type any  
colour

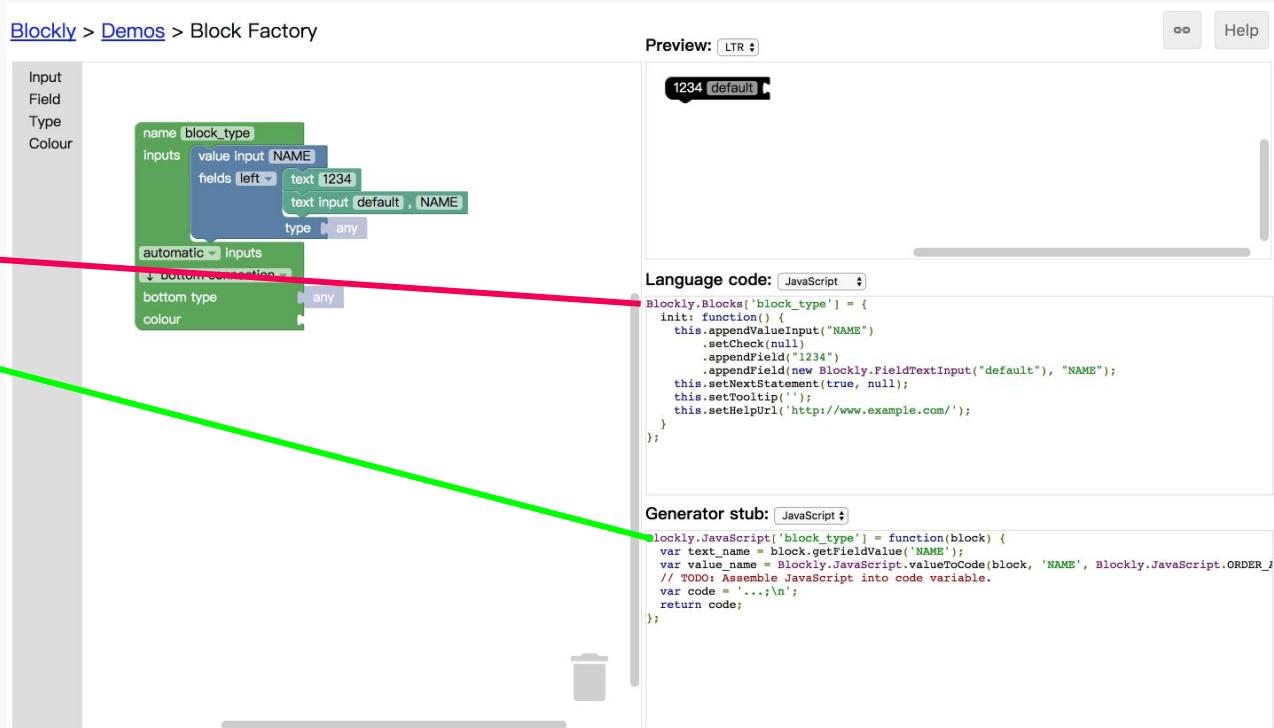
1234 default

Language code: JavaScript

```
Blockly.Blocks['block_type'] = {
  init: function() {
    this.appendValueInput("NAME")
      .setCheck(null)
      .appendField("1234")
      .appendField(new Blockly.FieldTextInput("default"), "NAME");
    this.setNextStatement(true, null);
    this.setTooltip('');
    this.setHelpUrl('http://www.example.com/');
  }
};
```

Generator stub: JavaScript

```
Blockly.JavaScript['block_type'] = function(block) {
  var text_name = block.getFieldValue('NAME');
  var value_name = Blockly.JavaScript.valueToCode(block, 'NAME', Blockly.JavaScript.ORDER_EQUAL);
  // TODO: Assemble JavaScript into code variable.
  var code = '...;\n';
  return code;
};
```



# LCD1602 積木模組

## [webduino-module-lcd1602](#)

LCD1602 Module for Webduino

JavaScript Updated 19 days ago

## [webduino-module-dfplayer](#)

DFPlayer Module for Webduino

JavaScript Updated 19 days ago

▼ webduino-module-lcd1602

▼ blockly

► msg

blocks.js

javascript.js

toolbox.xml

blockly.json

bower.json

LCD1602-blockly.js

LCD1602.js

package.json

輔助程式

主程式

# 自訂新增積木



# 實戰自訂積木

## 完成 LCD清除畫面的積木

# 模組 gh-pages URL

webduinoio.github.io/\${模組名稱}/blockly.json

The screenshot shows the GitHub search interface with the query "webduino-module". The results page displays four repositories:

- webduinoio/webduino-module-lcd1602**: Webduino Module for LCD1602. Language: JavaScript. Last updated 29 days ago.
- webduinoio/webduino-module-wabot**: Webduino Module for WABot. Language: JavaScript. Last updated on 9 Sep.
- webduinoio/webduino-module-dht**: Webduino Module for DHT. Language: JavaScript. Last updated on 8 Nov.
- webduinoio/webduino-module-gps**: Webduino Module for GPS. Language: JavaScript. Last updated on 4 Aug.

On the left sidebar, there are sections for Repositories (16), Code, Commits (10), Issues (5), Wikis (1), and Users. Below that is a Languages section showing 14 JavaScript, 1 CSS, and 1 HTML projects. At the bottom of the sidebar are links for Advanced search and Cheat sheet.

## 模組名稱

- webduino-module-lcd1602
- webduino-module-wabot
- webduino-module-dht
- webduino-module-gps
- ...

# 本機積木開發環境建置

## 在命令列模式下，抓積木模組

**git clone** <https://github.com/webduinoio/webduino-module-lcd1602>

## 在命令列模式下，安裝 WebServer

```
$ npm install -g http-server
```

## 在命令列模式下，啟動 WebServer

```
$ openssl req -newkey rsa2048 -new -nodes -x509 -days 3650 -keyout key.pem -out cert.pem  
$ http-server -S -C cert.pem -o --cors
```

```
✓ ~/projects/webduino-module-lcd1602 [master|✓]  
15:10 $ http-server --cors  
Starting up http-server, serving ./  
Available on:  
  http://127.0.0.1:8080  
  http://192.168.8.100:8080  
Hit CTRL-C to stop the server
```

# 本機積木開發環境建置

在瀏覽器的Webduino Blockly網頁，設定載入積木

