



# INTRO TO MICROPYTHON

- The porting guide #2

by Simon XI

# MICROPYTHON是什麼？

---

- MicroPython 就是一個為MCU等資源有限的設備設計的 **Python 3** 語言的解釋器
- Damien George, PhD, 物理學家



# 關鍵詞匯

---

- **Bytecode**

A compact representation of a Python program that generated by compiling the Python source code. This is what the VM actually executes. Bytecode is typically generated automatically at runtime and is invisible to the user. Note that while [CPython](#) and MicroPython both use bytecode, the format is different. You can also pre-compile source code offline using the [cross-compiler](#).

- **QSTR**

stands for unique STRing, it's the name they give for **interned strings**, which is meant for saving RAM and ROM, every built-in variables and functions' names are QSTR, they are created at compile time and their value is an index into a linked list of QSTR pool.

# 關鍵詞匯

---

- **REPL**

An acronym for “Read, Eval, Print, Loop”. This is the interactive Python prompt, useful for debugging or testing short snippets of code. Most MicroPython boards make a REPL available over a UART, and this is typically accessible on a host PC via USB.

- **Frozen Module**

A Python module that has been cross compiled and bundled into the firmware image. This reduces RAM requirements as the code is executed directly from flash.



# MICROPYTHON PORT的开发移植流程

---

1. 研究芯片原廠**SDK**以及**MicroPython**的build system
  2. 熟知芯片原廠**SDK**以及**MicroPython**的使用方法和核心代碼
  3. 移植芯片原廠**SDK**中的核心代碼，並於**MicroPython**的核心代碼適配
  4. 實現**MicroPython REPL**作為里程碑
  5. 移植大量的外設控制**API**到**MicroPython**中
  6. 大量的適配，測試還有文檔編寫
- #1
- #2

# RECAP

---

1. 研究芯片原廠**SDK**以及**MicroPython**的build system  
**GCC + Make**
2. 熟知芯片原廠**SDK**以及**MicroPython**的使用方法和核心代碼
  - py
  - ports
  - shared
  - mpy-cross

### 3. 移植芯片原廠**SDK**中的核心代碼，並於**MICROPYTHON**的核心程式碼適配

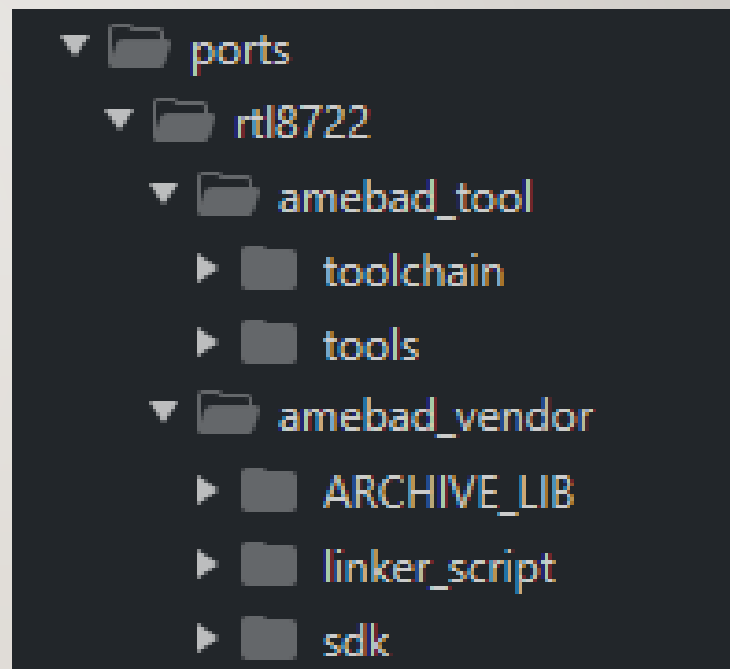
---

#### 1. 核心原廠程式碼 -> amebad\_vendor

1. ARCHIVE\_LIB ———> 靜態庫
2. linker\_script ———> linker腳本
3. sdk ———> headers檔

#### 1. 原廠工具 -> amebad\_tool

1. toolchain ———> 編譯工具鏈
2. tools ———> 桌面級工具



### 3. 移植芯片原廠**SDK**中的核心代碼，並於**MICROPYTHON**的核心程式碼適配

---

- 核心程式碼

- main.c                      —————> 啟動+初始化MP
- pins.c                        —————> 引腳的定義
- mpconfigport.h            —————> MP核心功能的開關、定義
- Makefile                    —————> Include MP的makefile從而編譯MP和Port相關的程式碼
- mphal.c                     —————> 實現最基本的硬體控制（UART + Delay）



## 4.實現MICROPYTHON REPL作為里程碑

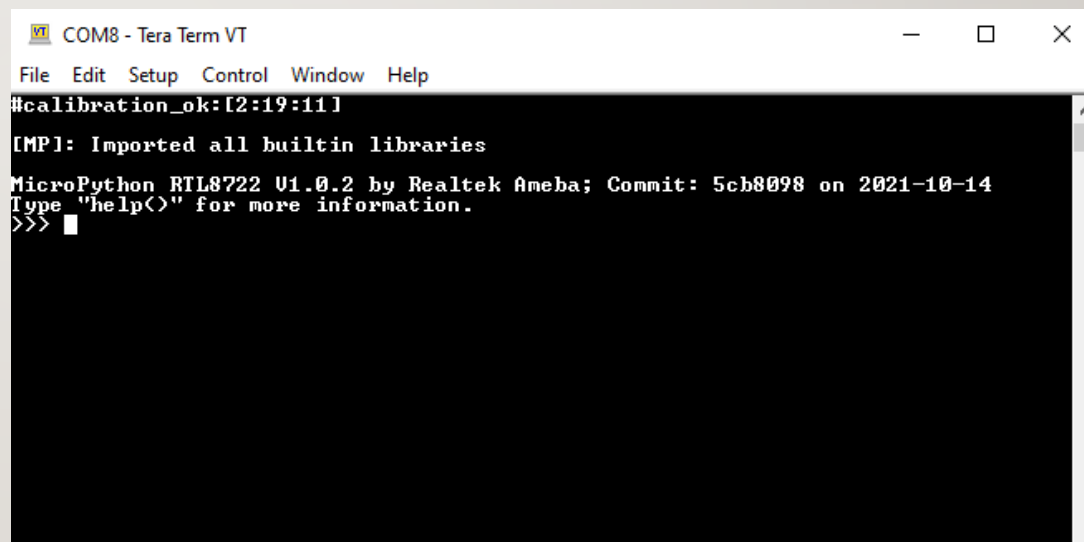
---

- REPL是檢驗MicroPython Port的里程碑

- 檢驗內建功能

- 檢驗資源管理

- 檢驗外設控制



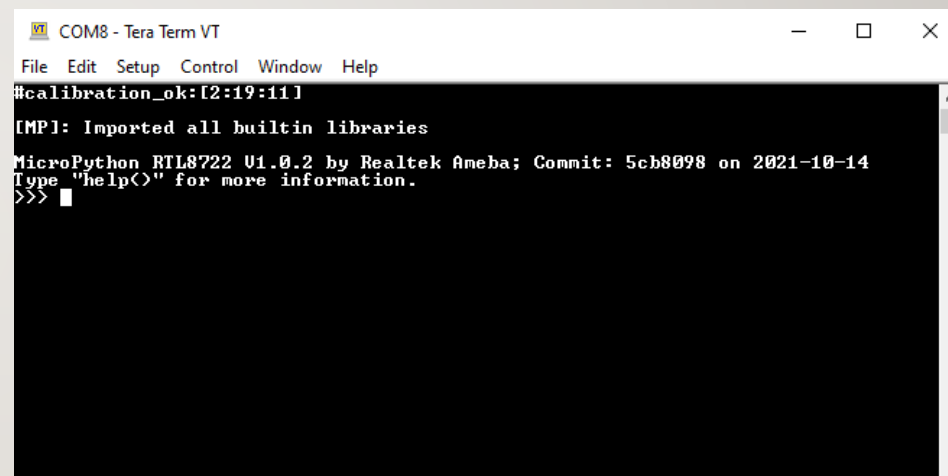
A screenshot of a Tera Term VT window titled 'COM8 - Tera Term VT'. The window has a menu bar with 'File', 'Edit', 'Setup', 'Control', 'Window', and 'Help'. The terminal output shows the following text: '#calibration\_ok:[2:19:11]', '[MP]: Imported all builtin libraries', 'MicroPython RTL8722 U1.0.2 by Realtek Ameba; Commit: 5cb8098 on 2021-10-14', 'Type "help(<)" for more information.', and a prompt '(>>>)' with a cursor. A vertical scrollbar is visible on the right side of the terminal area.

```
COM8 - Tera Term VT
File Edit Setup Control Window Help
#calibration_ok:[2:19:11]
[MP]: Imported all builtin libraries
MicroPython RTL8722 U1.0.2 by Realtek Ameba; Commit: 5cb8098 on 2021-10-14
Type "help(<)" for more information.
(>>> █
```

## 4.實現MICROPYTHON REPL作為里程碑

---

- 實現REPL的最基本要求
  - 正確初始化MP
  - 初始化UART並配置好“mp\_hal\_stdin/out
  - 在mpconfigport.h中enable
    - #if MICROPY\_ENABLE\_COMPILER
    - #if MICROPY\_REPL\_EVENT\_DRIVEN



```
COM8 - Tera Term VT
File Edit Setup Control Window Help
#calibration_ok:[2:19:11]
[MP]: Imported all builtin libraries
MicroPython RTL8722 U1.0.2 by Realtek Ameba; Commit: 5cb8098 on 2021-10-14
Type "help(<)" for more information.
>>> █
```

## 5.移植大量的外設控制API到MICROPYTHON中

- 真正的Porting才剛剛開始，還有大量的外設需要一一加入
  - UART
  - Pin
  - Timer
  - RTC
  - PWM
  - I2C
  - SPI
  - ADC
  - SDFS
  - WLAN
  - etc.

```
COM8 - Tera Term VT
File Edit Setup Control Window Help
#calibration_ok:[2:19:11]

[MP]: Imported all builtin libraries

MicroPython RTL8722 V1.0.2 by Realtek Ameba; Commit: 5cb8098 on 2021-10-14
Type "help()" for more information.
>>> import machine
>>> help(machine)
object <module 'umachine'> is of type module
  __name__ -- umachine
  reboot -- <function>
  UART -- <class 'UART'>
  Pin -- <class 'Pin'>
  Timer -- <class 'Timer'>
  RTC -- <class 'RTC'>
  PWM -- <class 'PWM'>
  I2C -- <class 'I2C'>
  SPI -- <class 'SPI'>
  ADC -- <class 'ADC'>
  SDFS -- <class 'SDFS'>
>>>
  class__      name__      ADC      I2C
  PWM          Pin          RTC      SDFS
  SOCK         SPI          Timer     UART
  WLAN         machine     modules  socket
  time         wireless
>>> import wireless
>>> help(wireless)
object <module 'wireless'> is of type module
  __name__ -- wireless
  WLAN -- <class 'WLAN'>
>>>
```

## 5.移植大量的外設控制API到MICROPYTHON中

- 以Pin模塊(GPIO)為例

### 用C實現OOP

void\* 類型的function →

```
STATIC mp_obj_t pin_on(mp_obj_t self_in) {  
    pin_obj_t *self = self_in;  
    gpio_write(&(self->obj), 1);  
    return mp_const_none;  
}  
  
STATIC MP_DEFINE_CONST_FUN_OBJ_1(pin_on_obj, pin_on);
```

必須declare所有用戶使用的API →

所有API都要被加入dictionary table →

```
STATIC const mp_map_elem_t pin_locals_dict_table[] = {  
    // instance methods  
    { MP_OBJ_NEW_QSTR(MP_QSTR_id),          MP_OBJ_FROM_PTR(&pin_id_obj) },  
    { MP_OBJ_NEW_QSTR(MP_QSTR_init),        MP_OBJ_FROM_PTR(&pin_init_obj) },  
    { MP_OBJ_NEW_QSTR(MP_QSTR_value),       MP_OBJ_FROM_PTR(&pin_value_obj) },  
    { MP_OBJ_NEW_QSTR(MP_QSTR_off),         MP_OBJ_FROM_PTR(&pin_off_obj) },  
    { MP_OBJ_NEW_QSTR(MP_QSTR_on),          MP_OBJ_FROM_PTR(&pin_on_obj) },  
    { MP_OBJ_NEW_QSTR(MP_QSTR_toggle),      MP_OBJ_FROM_PTR(&pin_toggle_obj) },  
};
```

API的名字要map到對應的function pointer上 →



## 5.移植大量的外設控制**API**到**MICROPYTHON**中

---

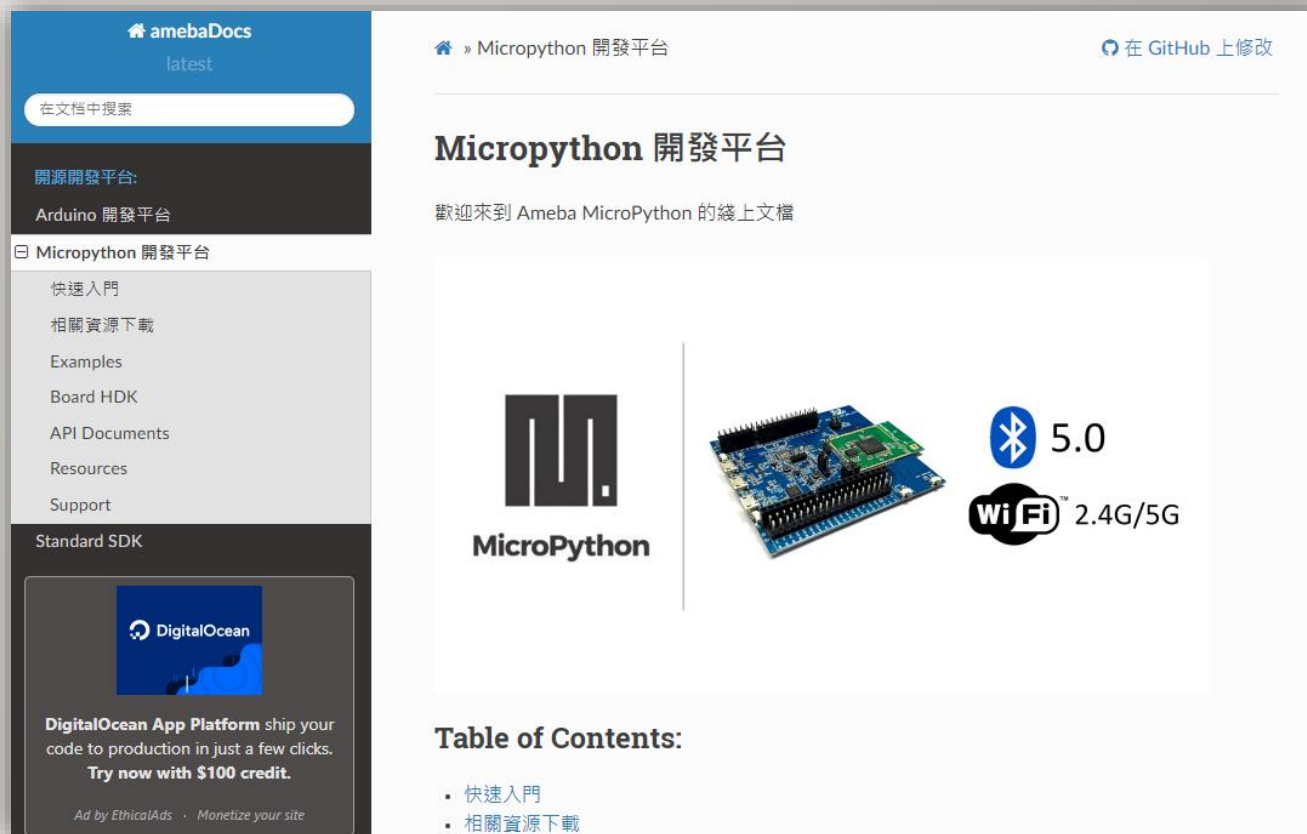
- Demo
  - GPIO IRQ Test



## 6.大量的適配，測試還有文檔編寫

- Readthedocs

[https://amebaiotdocuments-zh-tw.readthedocs.io/zh\\_TW/latest/](https://amebaiotdocuments-zh-tw.readthedocs.io/zh_TW/latest/)



The screenshot displays the AmebaDocs website interface. The left sidebar features a search bar, a navigation menu with links to '快速入門' (Quick Start), '相關資源下載' (Download Related Resources), 'Examples', 'Board HDK', 'API Documents', 'Resources', 'Support', and 'Standard SDK'. The main content area is titled 'Micropython 開發平台' (Micropython Development Platform) and includes a welcome message, a table of contents, and a DigitalOcean advertisement.

amebaDocs latest

在文档中搜索

開源開發平台:

Arduino 開發平台

☰ Micropython 開發平台

快速入門

相關資源下載

Examples

Board HDK

API Documents

Resources

Support

Standard SDK

DigitalOcean

DigitalOcean App Platform ship your code to production in just a few clicks.




Try now with \$100 credit.

Ad by EthicalAds · Monetize your site

» Micropython 開發平台 在 GitHub 上修改

### Micropython 開發平台


歡迎來到 Ameba MicroPython 的線上文檔





#### Table of Contents:


- 快速入門
- 相關資源下載

# ICSHOP X RTL8722DM\_MINI



 Line @iCshop

 我的帳戶  
登入

 購物車: NT\$0

商品目錄

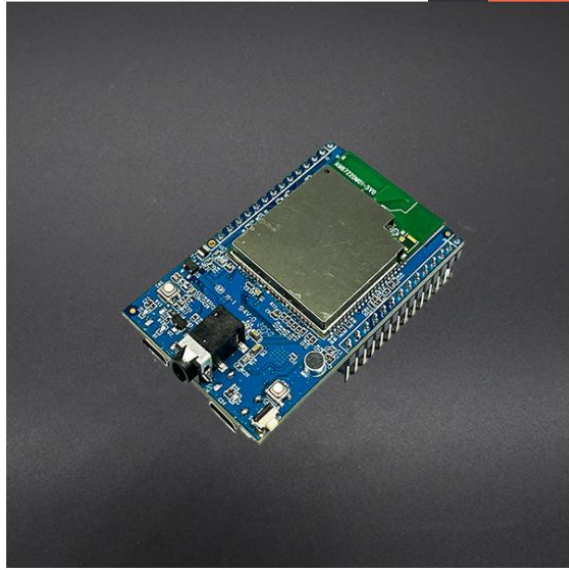
新進商品

限量商品出清

CIRCUS PI

技術文章

Home / 開發板 / Ameba



New 減價300

Ameba RTL8722DM mini Board (AMB 23)

NT\$ ~~799~~499

產品貨號 368030501635

品牌

原廠貨號

數量  [加入購物車](#) [有庫存](#)

關鍵字 [阿米巴](#) [IOT](#) [物聯網](#)

運送方式 ☒ 7-11取貨付款 ☒ 郵寄(單一運費) ☒ 宅配 ☒ 宅配貨到付款 ☒ 外島郵寄 ☒ 順豐快速 ☒ 順豐快速貨到付款





# PORT開發指南

---

## 1. Background and Project Structure

<https://forum.amebaiot.com/t/introduction-to-developing-micropython-1-background-and-structure/93>

## 2. Environment Setup

<https://forum.amebaiot.com/t/introduction-to-developing-micropython-2-environment-setup/99>

## 3. Learning the Build System

<https://forum.amebaiot.com/t/introduction-to-developing-micropython-3-getting-started/112>

## 4. Developing New Module for RTL8722 MicroPython Port

<https://forum.amebaiot.com/t/introduction-to-developing-micropython-4-developing-new-module-for-rtl8722-port/130>

## 5. MicroPython API Design and Methods

<https://forum.amebaiot.com/t/introduction-to-developing-micropython-5-micropython-api-design-and-methods/143>