
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

#1

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作爲里程碑
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

#2

RECAP

1. 研究芯片原廠SDK以及MicroPython的build system

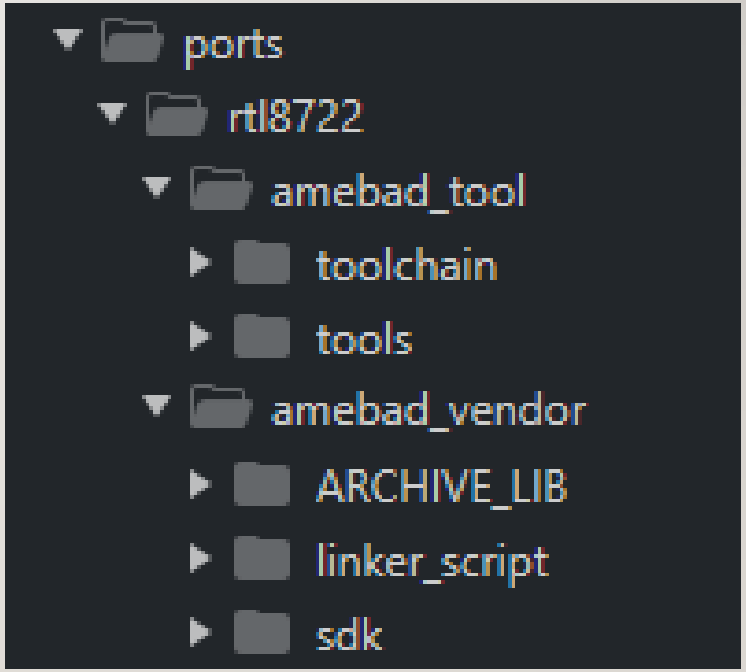
   GCC + Make

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脚本
    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

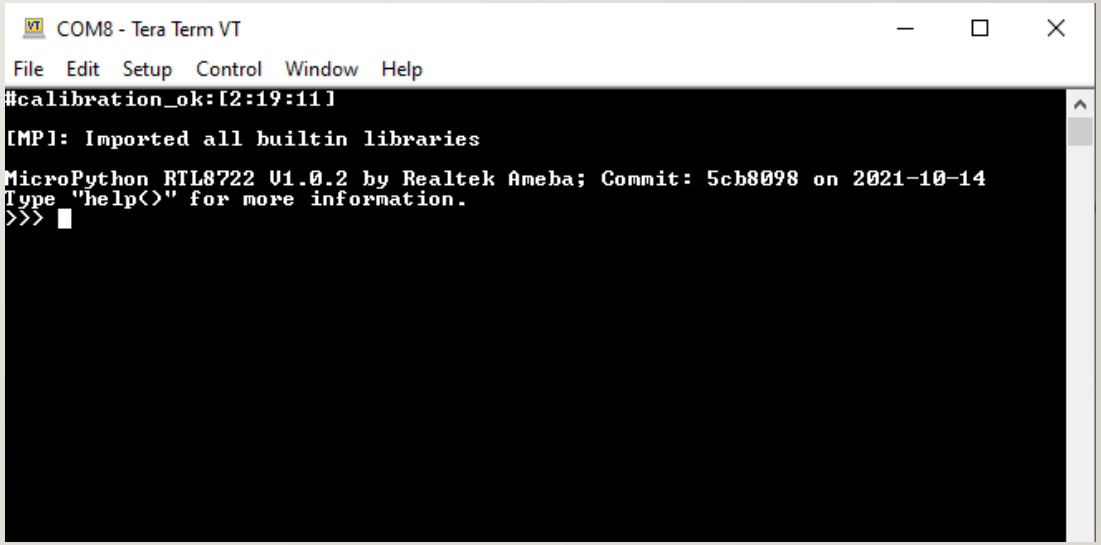


3. 移植芯片原廠SDK中的核心代碼，並於MICROPYTHON的核心程式碼適配

- 核心程式碼
  - main.c → 啓動+初始化MP
  - pins.c → 引脚的定義
  - 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）

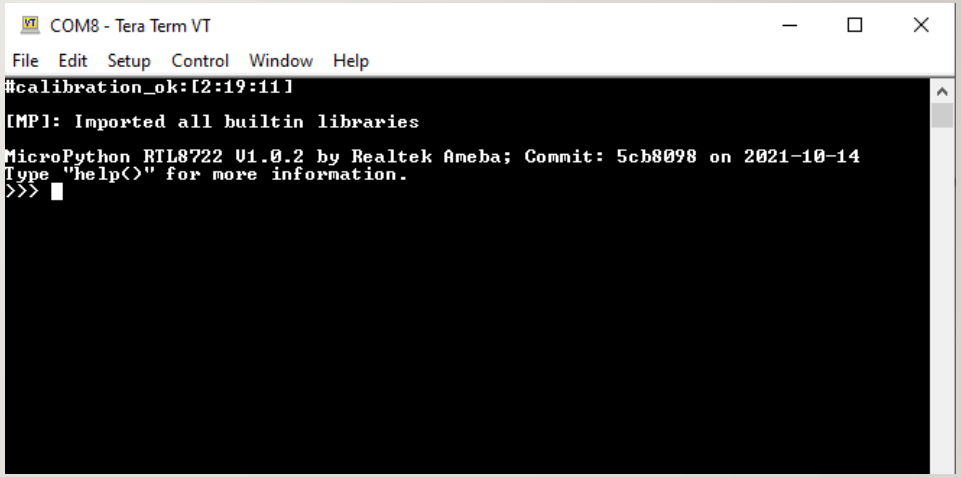
4.實現MICROPYTHON REPL作爲里程碑

- REPL是檢驗MicroPython Port的里程碑
  - 檢驗内建功能
  - 檢驗資源管理
  - 檢驗外設控制



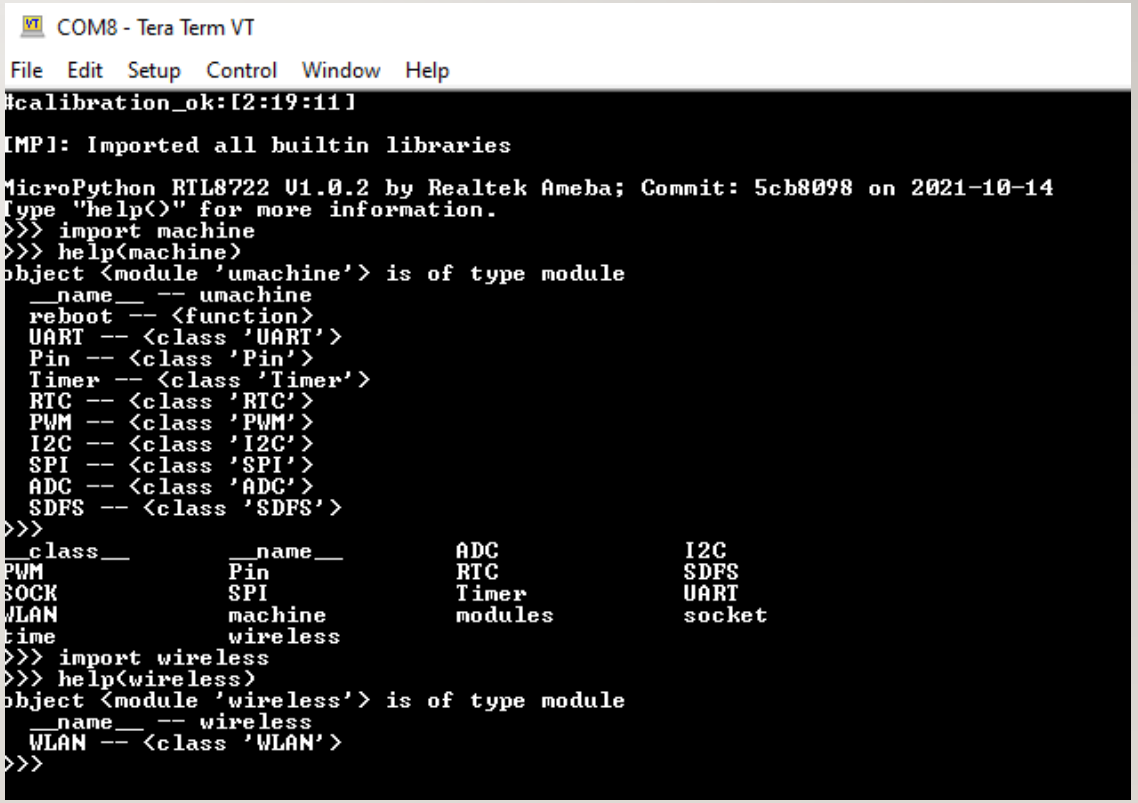
4.實現MICROPYTHON REPL作爲里程碑

- 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



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.



5.移植大量的外設控制API到MICROPYTHON中

- 以Pin模塊(GPIO)爲例

用C實現OOP

void* 類型的function →

必須declare所有用戶使用的API →

```c
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);
```

所有API都要被加入dictionary table →

API的名字要map到對應的function pointer上 →

```c
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) },
```

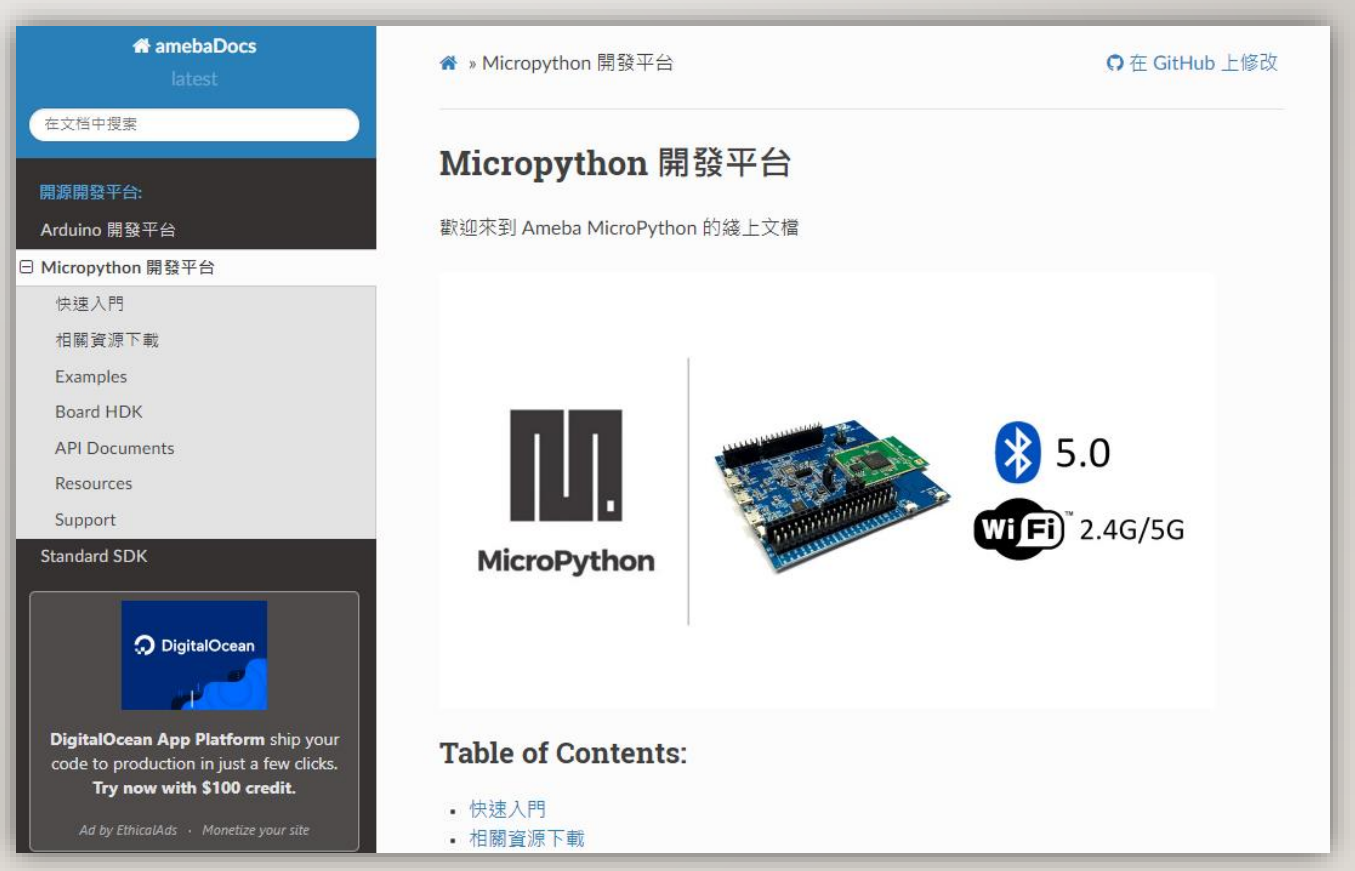
5.移植大量的外設控制API到MICROPYTHON中

- Demo
  - GPIO IRQ Test

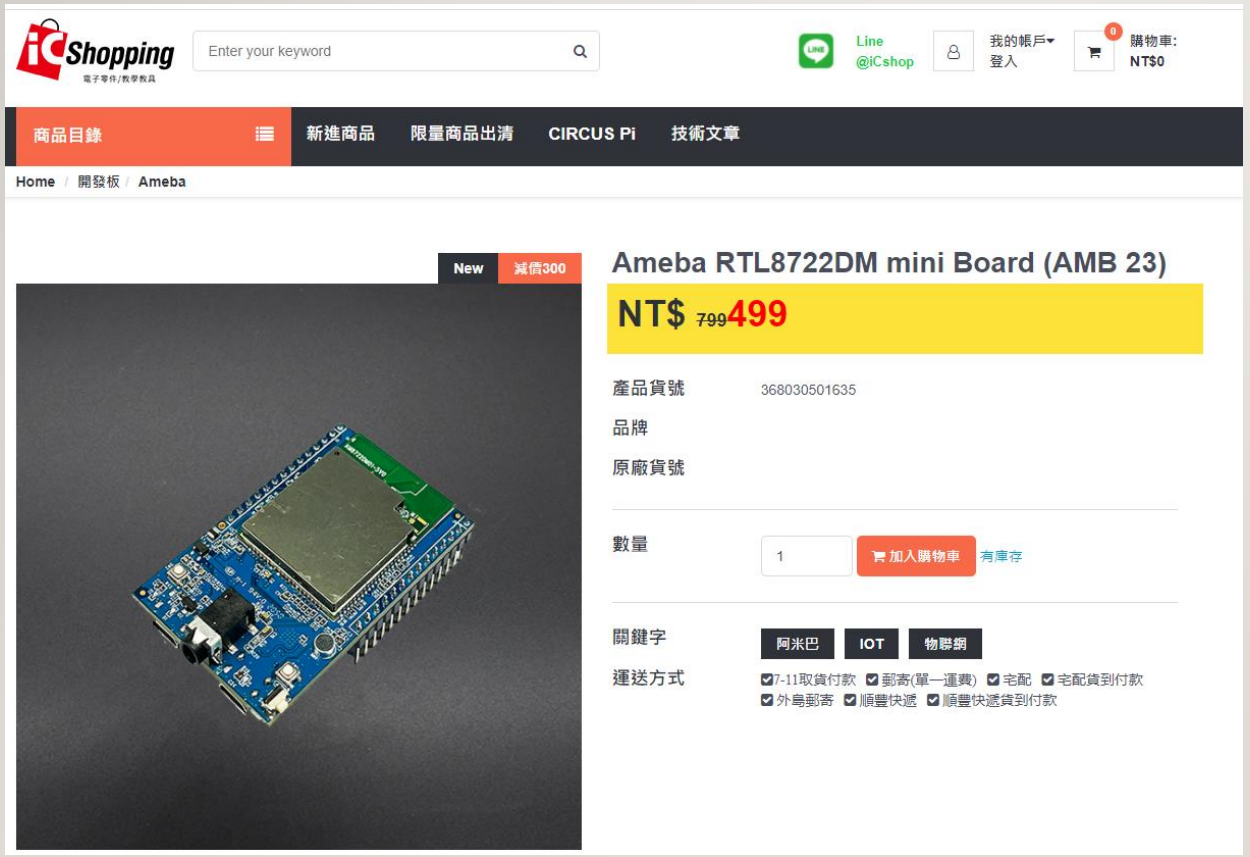
6.大量的適配，測試還有文檔編寫

- Readthedocs

https://amebaiotdocuments-zh-tw.readthedocs.io/zh_TW/latest/



ICSHOP X RTL8722DM_MINI



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