INTRO TO MICROPYTHON

- The porting guide #1

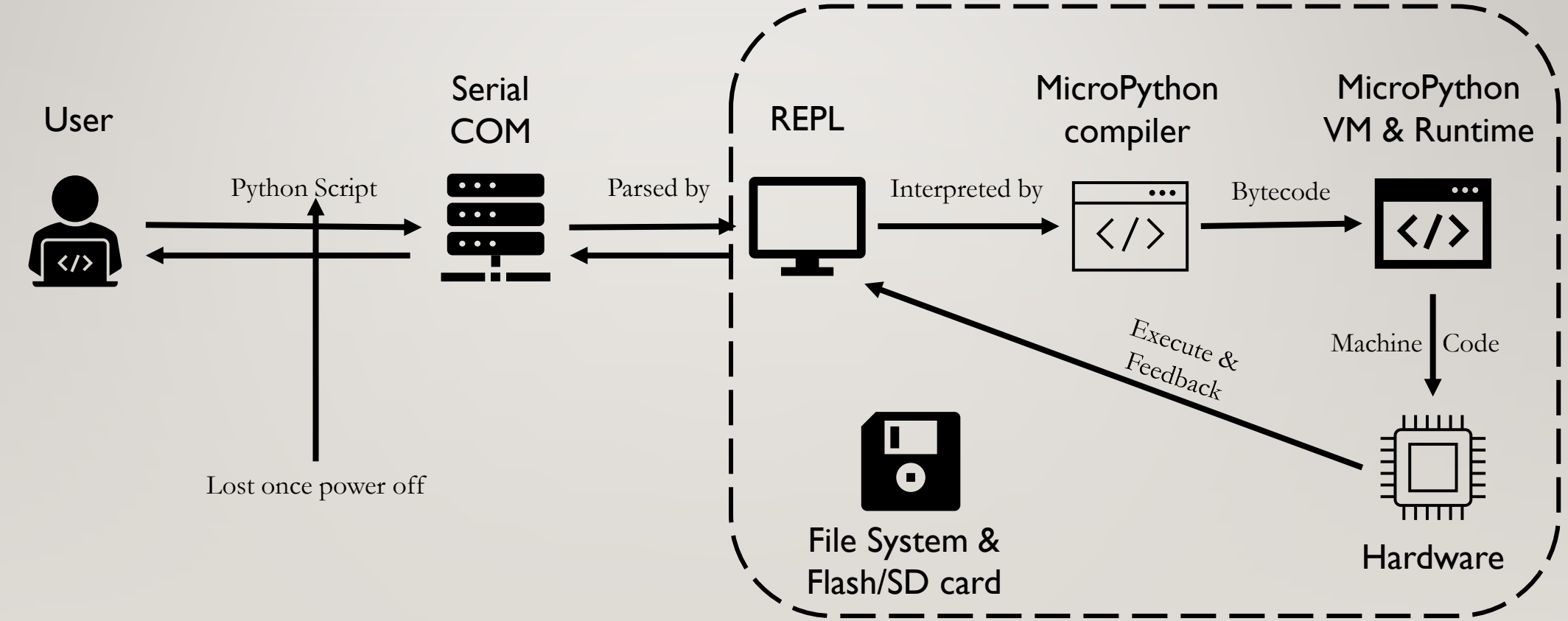
by Simon XI

MICROPYTHON是什麼?

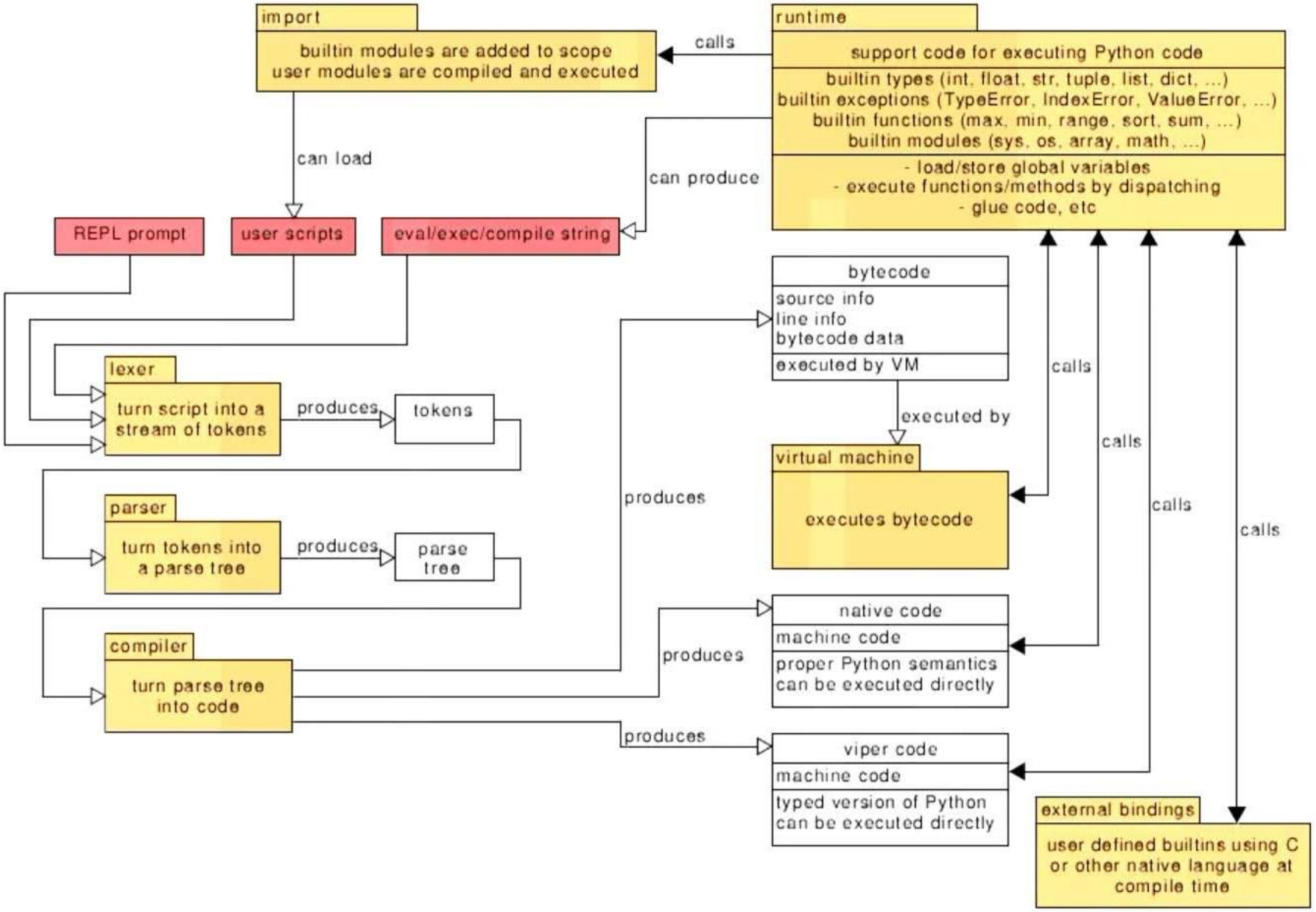
- MicroPython 就是一個為MCU等資源有限的設備設計的 **Python 3** 語言的解釋器

- Damien George, PhD, 物理學家

運行原理


User
Python Script
Serial
COM
Parsed by
REPL
Interpreted by
MicroPython
compiler
Bytecode
MicroPython
VM & Runtime
Execute &
Feedback
Machine Code
Lost once power off
File System &
Flash/SD card
Hardware

INTERNAL ARCHITECTURE



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.

EXAMPLE SCRIPT

```
def do_sleep(d):
    print("sleep", d)
    time.sleep(d)
```

```
00 LOAD_GLOBAL print
03 LOAD_CONST_STRING "sleep"
06 LOAD_FAST 0
07 CALL_FUNCTION n=2 nkw=0
09 POP_TOP
10 LOAD_GLOBAL time
13 LOAD_METHOD sleep
16 LOAD_FAST 0
17 CALL_METHOD n=1 nkw=0
19 POP_TOP
20 LOAD_CONST_NONE
21 RETURN_VALUE
```

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.

同一個專案，不同的開發

1. MicroPython層的開發
   - 比如MQTT library
   - Neopixel library
   - TFT LCD libraries，　等等

   } Python code

2. MicroPython Port層的開發
   - GPIO
   - SPI
   - WiFi，等等

   } C code

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

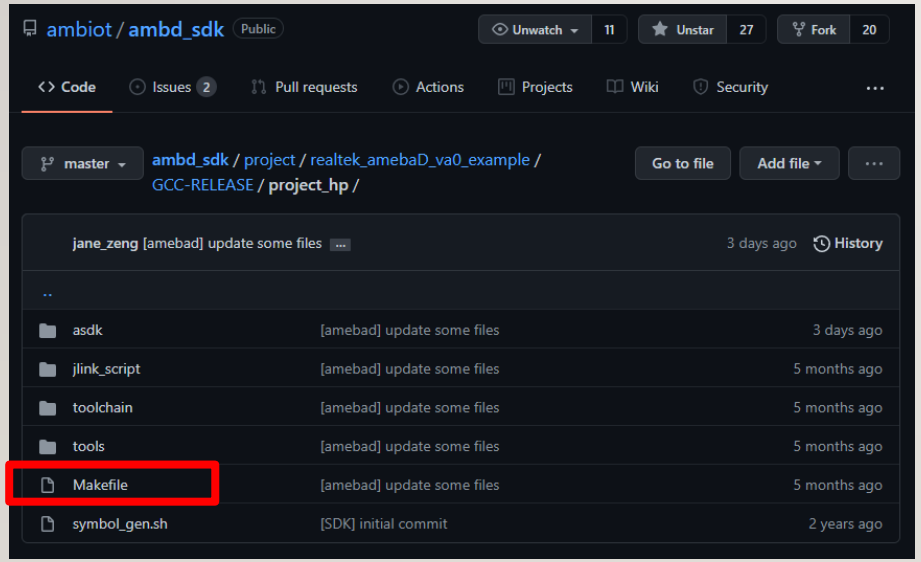
3. 移植芯片原廠SDK中的核心代碼，並於MicroPython的核心代碼適配
4. 實現MicroPython REPL作爲里程碑
5. 移植大量的外設控制API到MicroPython中

#2

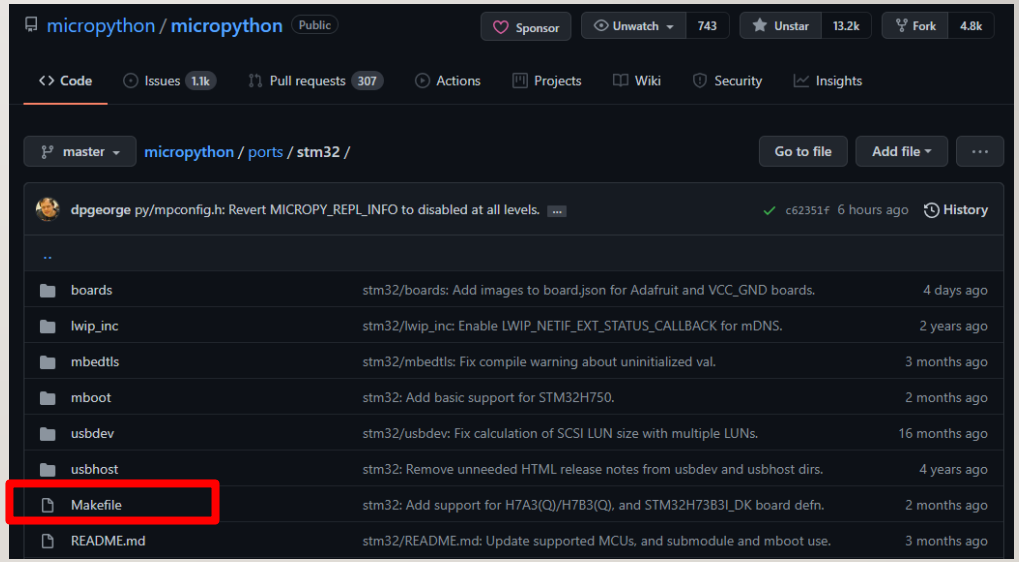
6. 大量的適配，測試還有文檔編寫

1. 研究原廠SDK以及MICROPYTHON的BUILD SYSTEM

- **Realtek RTL8722DM的原廠SDK**



- **MicroPython的Github Repo**



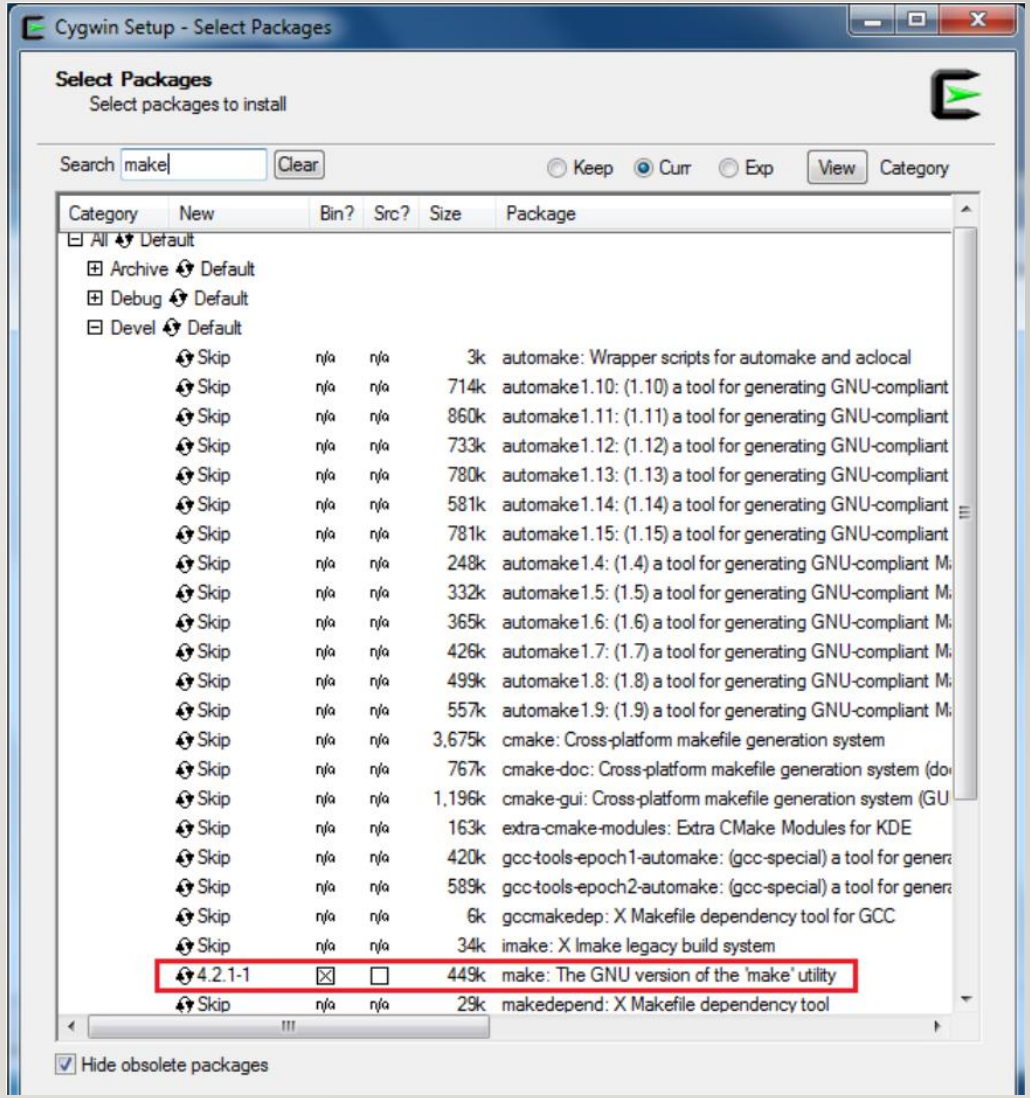
GCC + Make!

GCC + MAKE

- GCC -- GNU Compiler Collection
  - 可能是最受歡迎的開源開發**Toolchain工具鏈**
- Make -- GNU Make
  - 管理並生成**executable可執行文檔**的開源工具

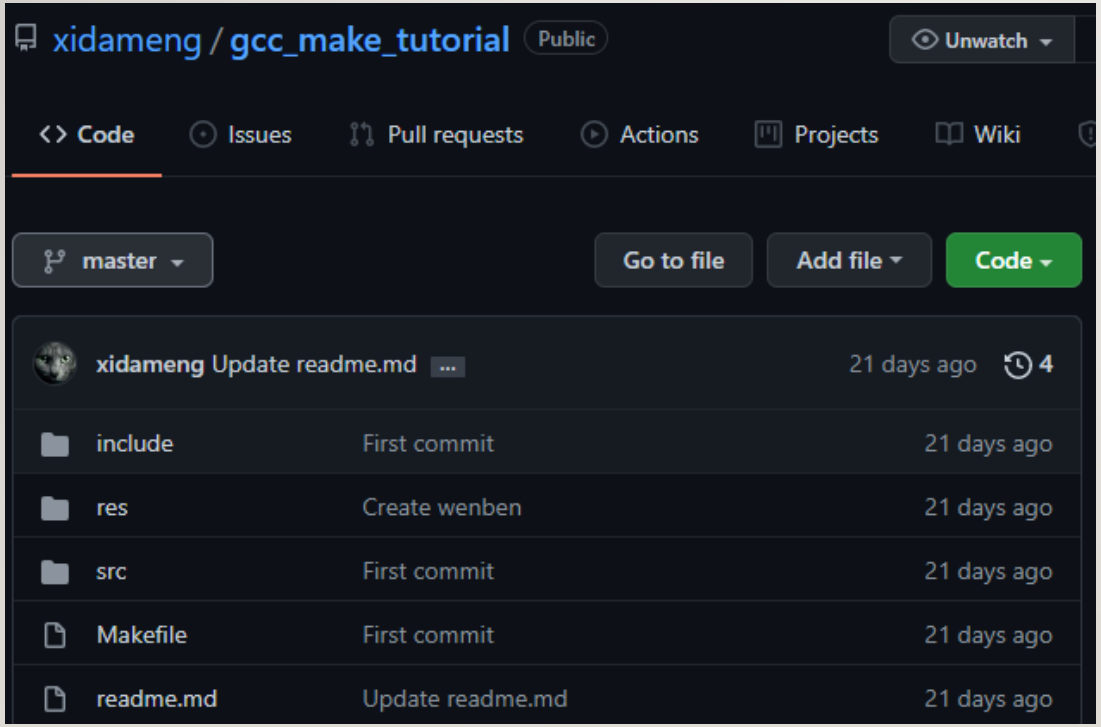
環境配置

- Cygwin
  - Cygwin is a POSIX-compatible programming and runtime environment that runs natively on Microsoft Windows.
  - make + bc
- Python 3



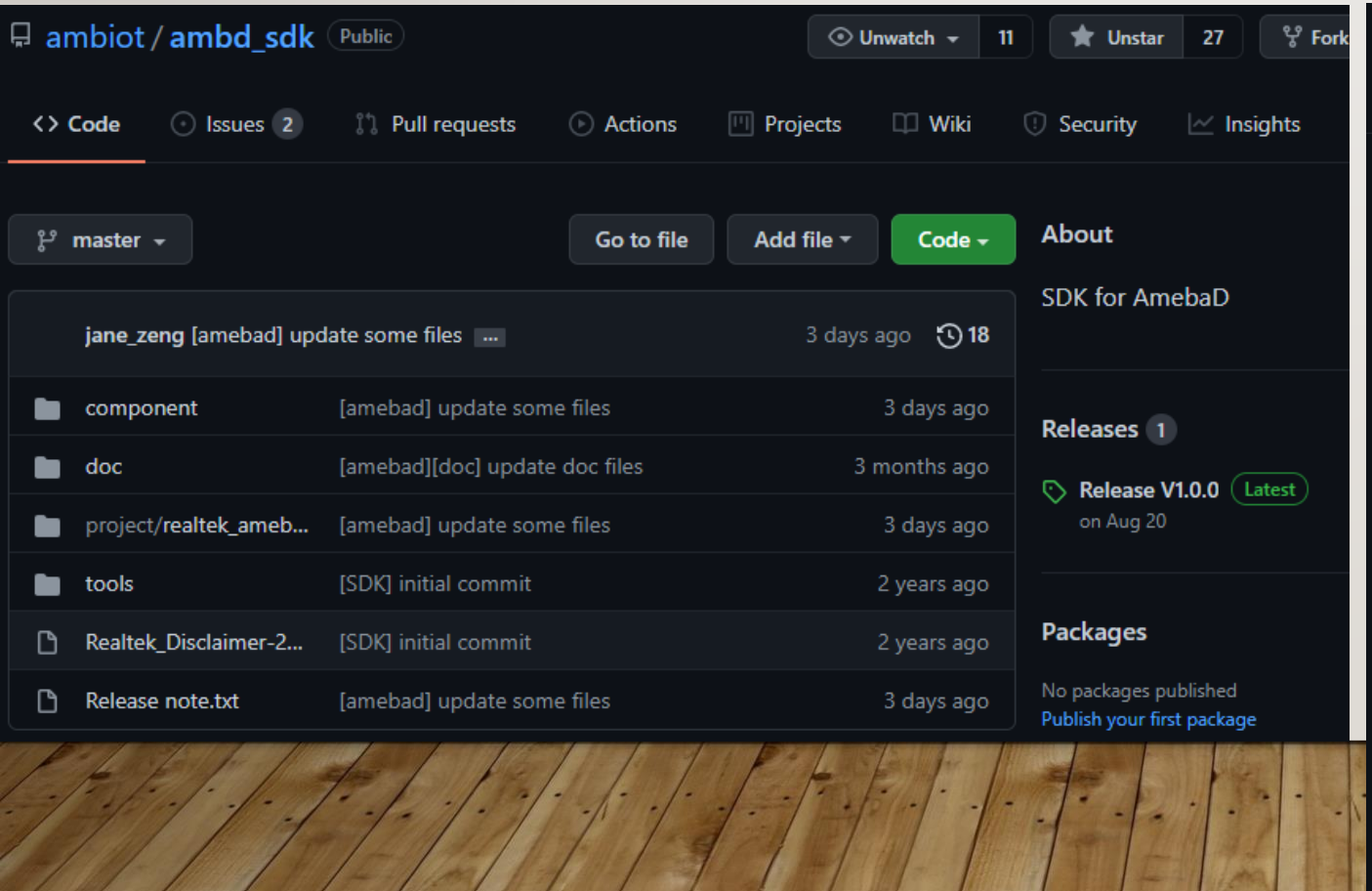
玩一下GCC+MAKE

- https://github.com/xidameng/gcc_make_tutorial

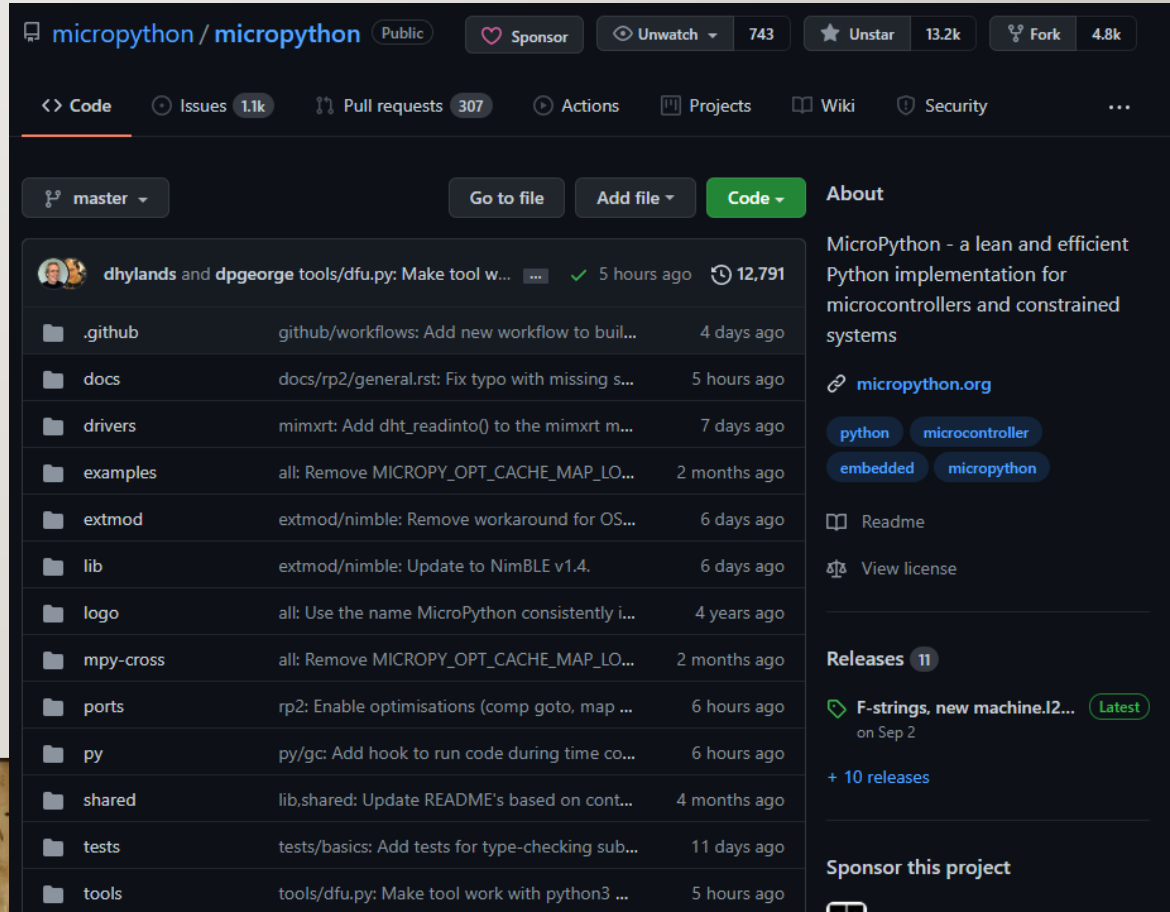


2. 熟知原廠SDK以及MICROPYTHON的使用方法和核心代碼

- **Realtek RTL8722DM的原廠SDK**



- **MicroPython的Github Repo**



原廠SDK的使用方法和核心代碼

- Components: 核心功能代碼
- Doc: 技術文檔
- Project: 已經配置好的build system
- Tools: 桌面級工具

MICROPYTHON的使用方法和核心代碼

- doc： read the docs綫上文檔的源碼
- drivers: 某些硬體的驅動程式
- examples： 示例代碼和C模塊的模板
- extmod： 非核心的C模塊
- lib：第三方開源的C語言庫
- **mpy-cross**: MicroPython的cross compiler
- **ports**： 這種不同硬體的核心移植代碼
- **py**：MicroPython解釋器最核心的代碼
- **shared**：與所有ports共用的核心代碼
- tests：測試用的MicroPython代碼
- tools：桌面級工具

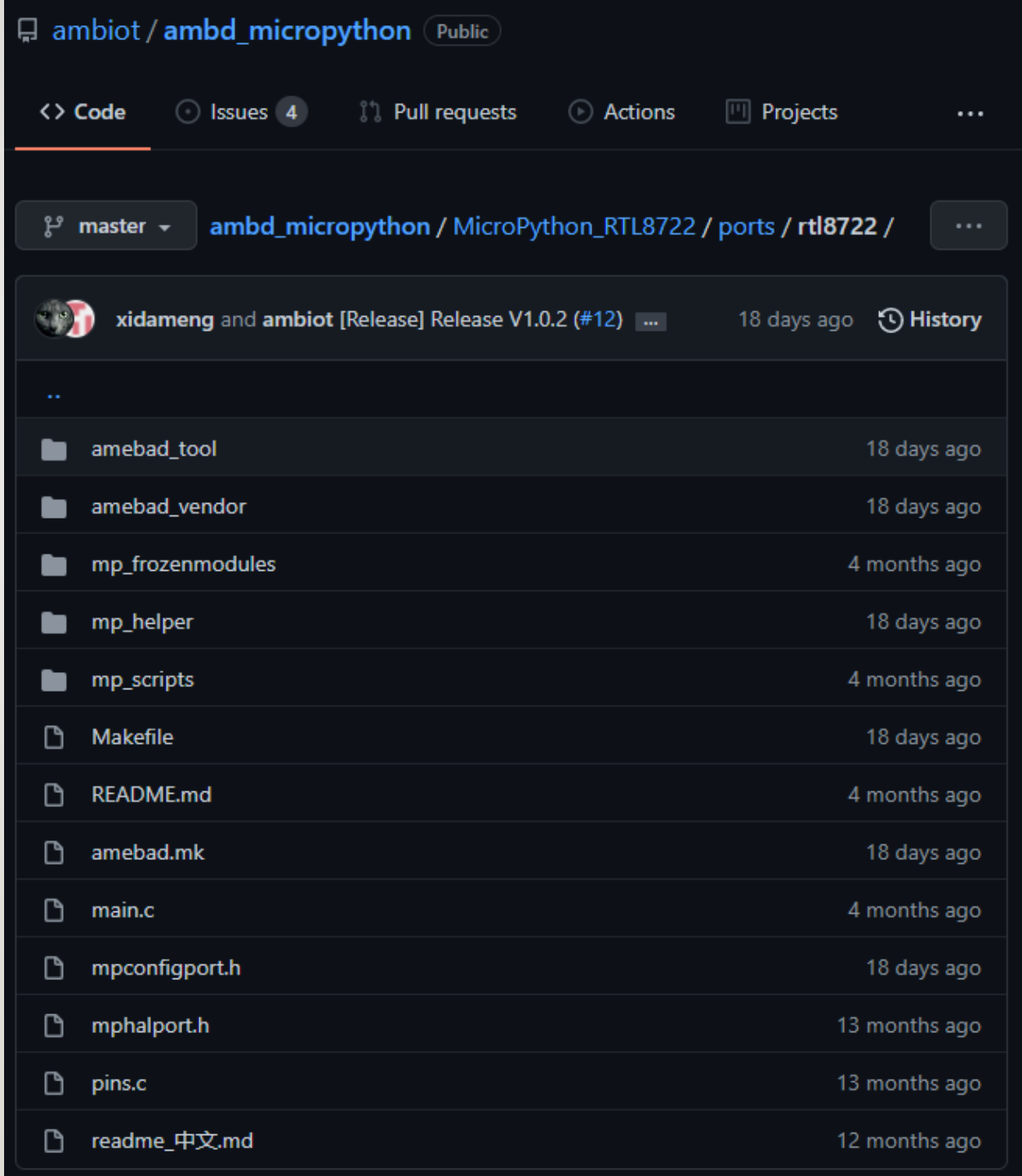
核心編譯流程

1. mpy-cross
2. py/mkenv.mk
3. py/py.mk
4. extmod/extmod.mk
5. ports/rtl8722/amebad.mk
6. py/mkrules.mk

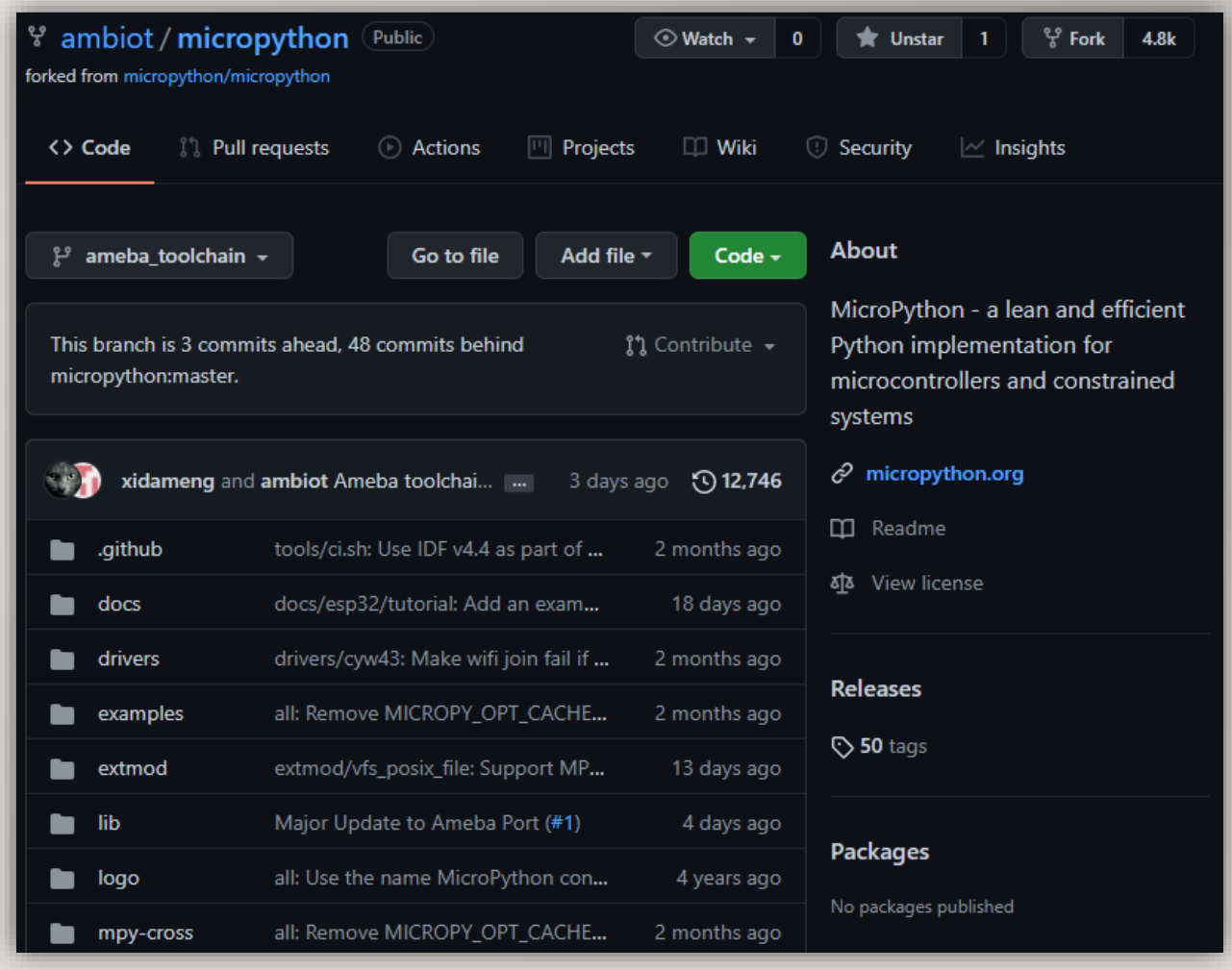
1. cross compiler
2. 配置環境
3. 標明核心解釋器代碼 + 創建qstr庫
4. 標明額外C模塊代碼
5. 編譯與硬體有關的代碼
6. 編譯所有MicroPython代碼

MICROPYTHON AMEBA PORT

https://github.com/ambiot/ambd_micropython



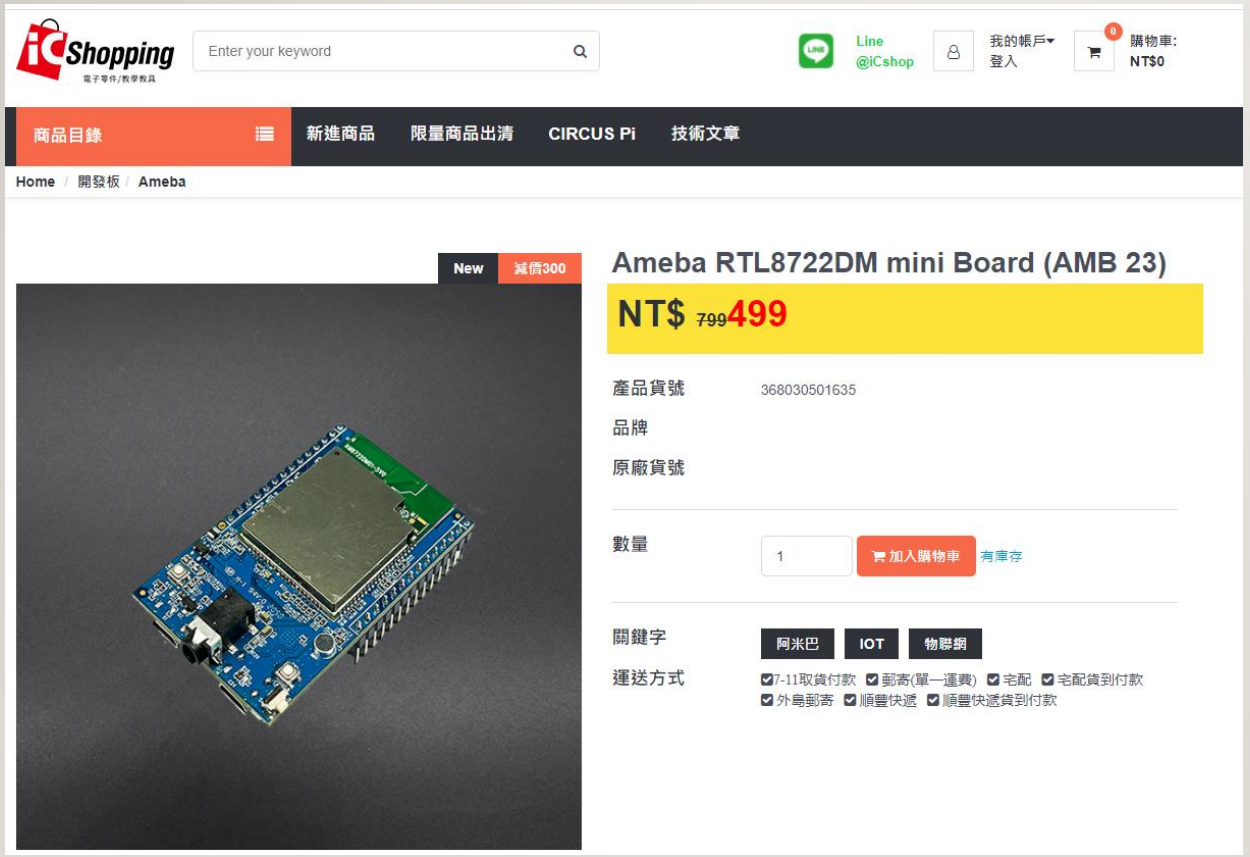
AMEBA FORKED MICROPYTHON



可以用ARDUINO開發MICROPYTHON嗎?

- 可以!
- 但是**非常+極其**不好用!!
- 三個原因:
  1. MicroPython編譯+上傳一次就可以了，之後的開發主要靠在文檔系統中保存和修改Python code，這點上Arduino IDE幾乎做不到
  2. Arduino IDE對Python語言的支援是負數的
  3. Arduino的Serial Monitor根本無法配合REPL使用

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