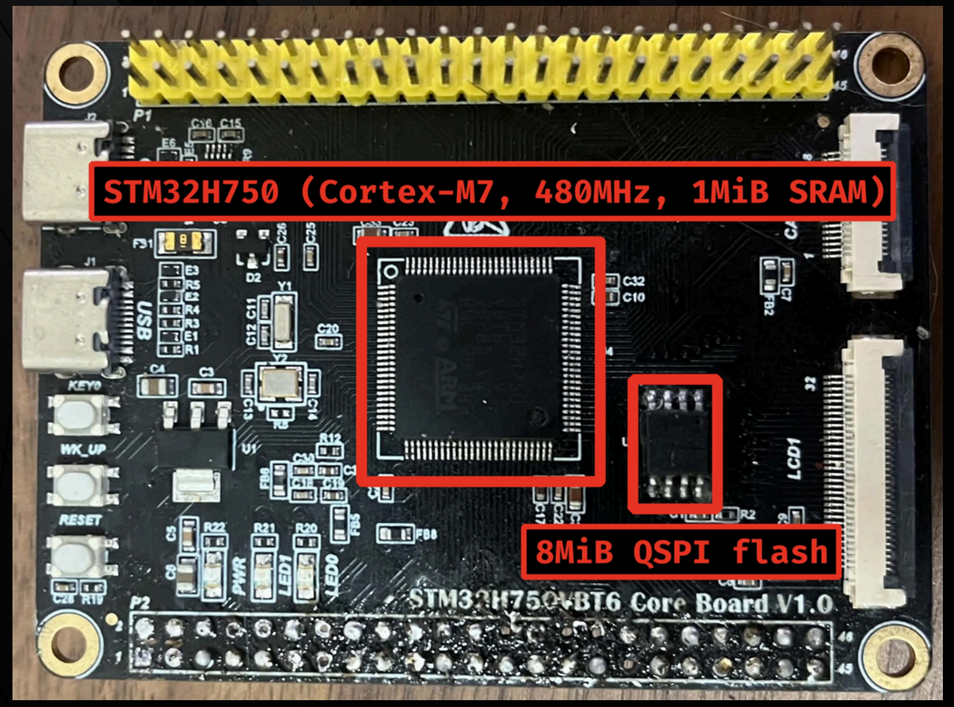
OPTIMIZE LINUX KERNEL TO FIT MICROCONTROLLER WITH 1MB RAM

作者：陳麒升
指導教授：陳奇業

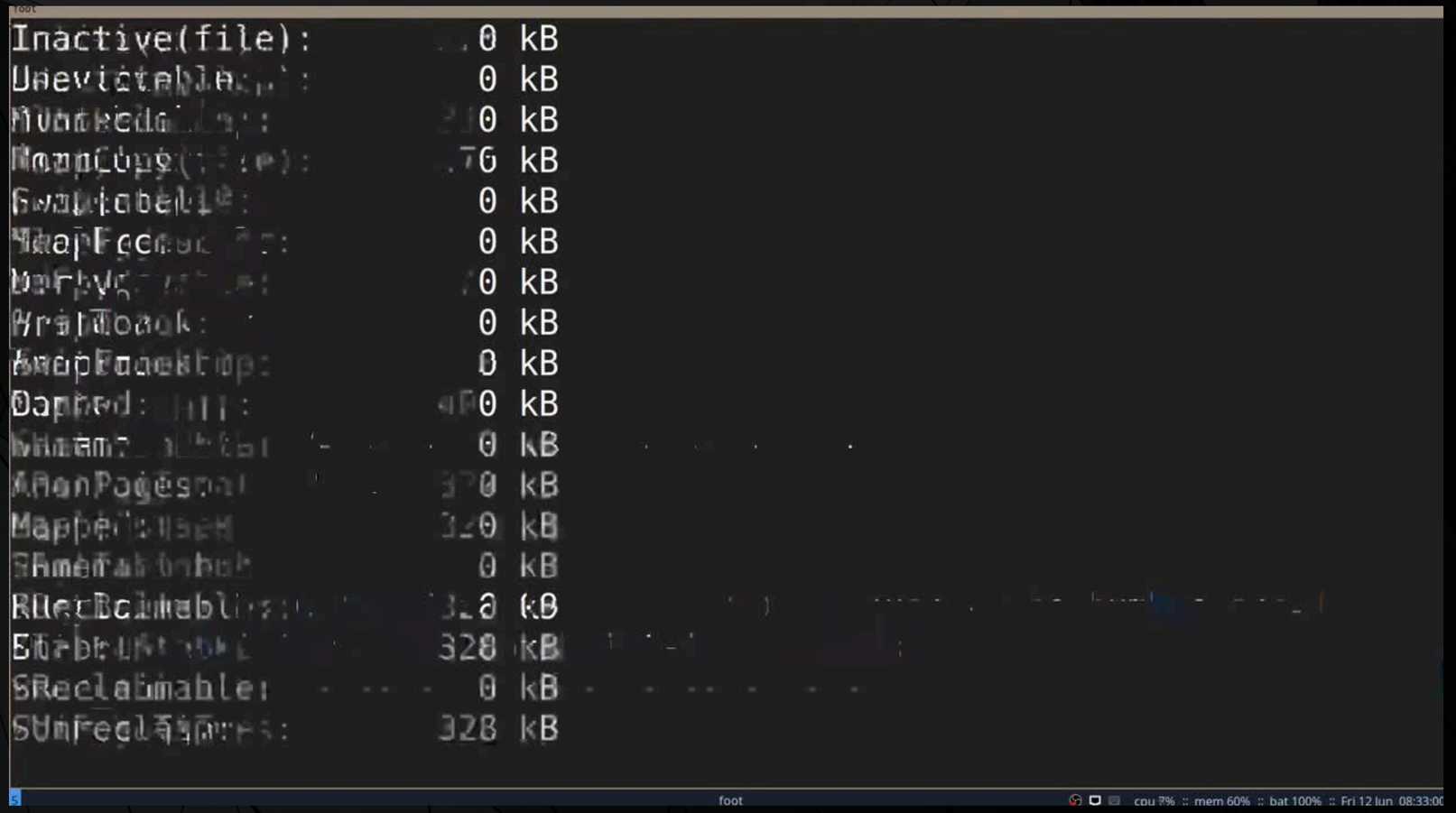
今天的目標

- Linux 7.0 in 1 MiB RAM
- Linux BSP
- 整合工具鏈與 libc
- 確保加上額外記憶體後即可因應更複雜的使用場景



→ 現有 Linux 軟體切換工具鏈就能移植上去

DEMO


Inactive(file): 0 kB
Unevictable: 0 kB
Mlocked: 0 kB
Mapped: 76 kB
Shmem: 0 kB
KReclaimable: 328 kB
Slab: 328 kB
SReclaimable: 0 kB
SUnreclaim: 328 kB
foot
cpu 7% :: mem 60% :: bat 100% :: Fri 12 Jun 08:33:00

動機

MCU：determinism、低延遲 → real-time 必要性質

高性能 MCU：更大 RAM、更高時脈 → 可運行 Linux noMMU

應用場景：

同時需要 real-time 與運行複雜程式

動機

MCU：determinism、低延遲 → real-time 必要性質

高性能 MCU：更大 RAM、更高時脈 → 可運行 Linux noMMU

應用場景：

同時需要 real-time 與運行複雜程式

做的是這裡的最小化驗證

Linux 好處

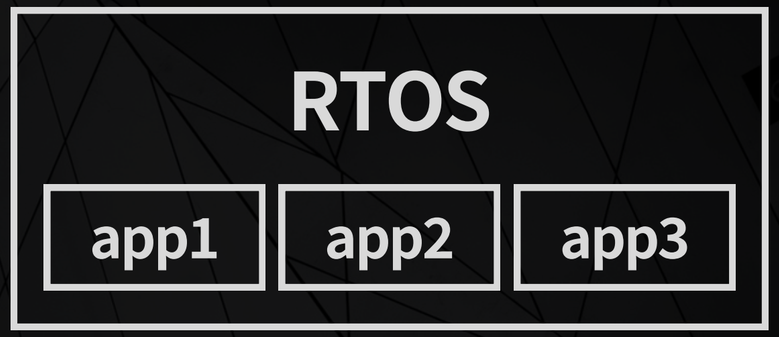
- 豐富的軟體生態、子系統、驅動程式
- 支援 uClibc-ng、FDPIC 執行檔 → 現有程式可重用
- FDPIC 支援動態連結、XIP → 減少記憶體用量、維護成本
- 內建系統分析工具 (debugfs、procfs、kprobe 等)

Reference: Linux 應用於微控制器的考量

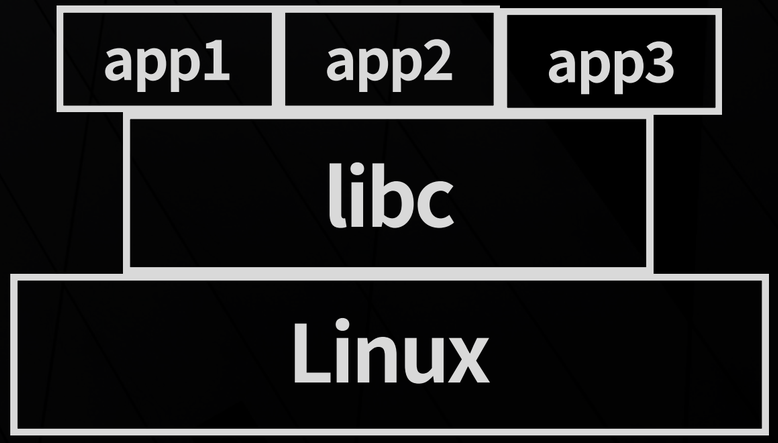
EU Cyber Resilience Act (CRA)

規定軟體要長期更新

RTOS：需重新整合、驗證



Linux：更新 app 與 libc



Reference: Linux 應用於微控制器的考量

開始實作吧

QEMU SoC Model（軟體模擬）

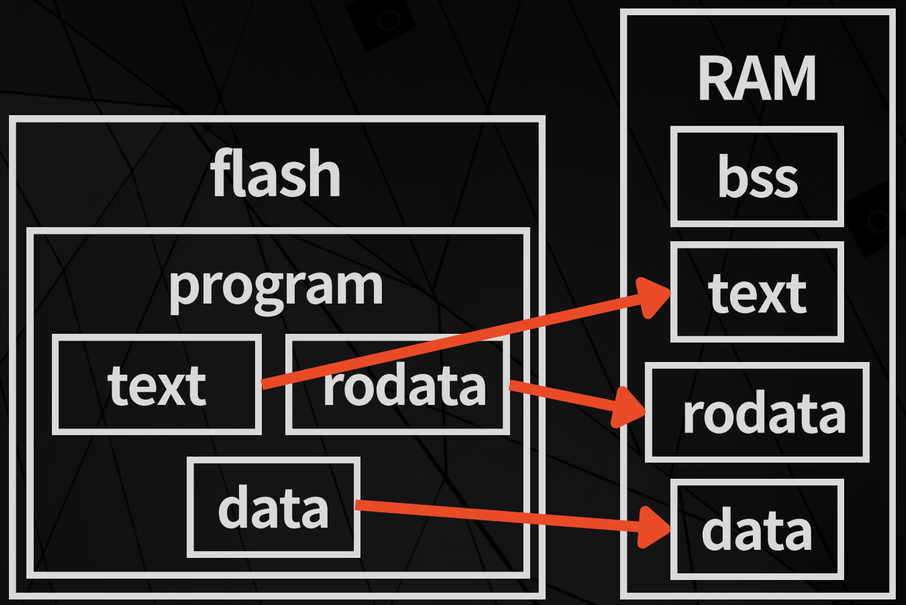
- 有效率的 GDB debug (比起 JTAG)
- runtime memory profile

需要多的 RAM，可以自己加

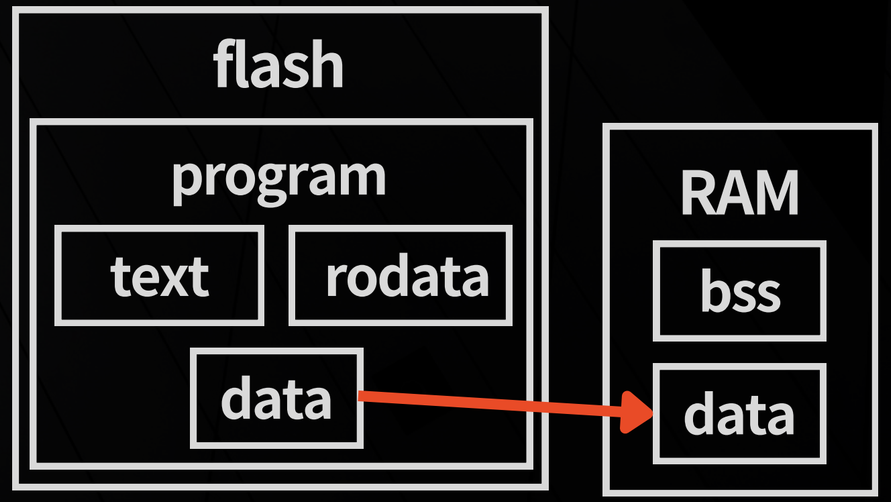
XIP (eXecuted In Place)

readonly 部份不載入 RAM

XIP



non-XIP



RAM↓

從 tinyconfig 開始 (minimal config)

以下統計 data + bss (靜態 RAM 佔用)

- tinyconfig + STM32 必要的 config：152 KiB
- 去除為 page table 預留的空間 (32KiB) 使用 128KiB
- 關掉沒用到的 Linux config (44KiB)
- 利用 System.map 發現多餘的結構體 (3KiB)

降到 105KiB，但還沒有 rootfs

```
Kernel panic - not syncing: No working init found
```

檔案系統（romfs）

以下統計 data + bss (靜態 RAM 佔用)

- 不壓縮的放在 flash 中
- 讀取時不會載入到 RAM
- 支援 Application XIP

```
VFS: Mounted root (romfs filesystem) readonly on device 31:0.
```

使用者程式

Linux noMMU 可用 FLAT binary 與 FDPIC 執行檔

都成功的運行了：

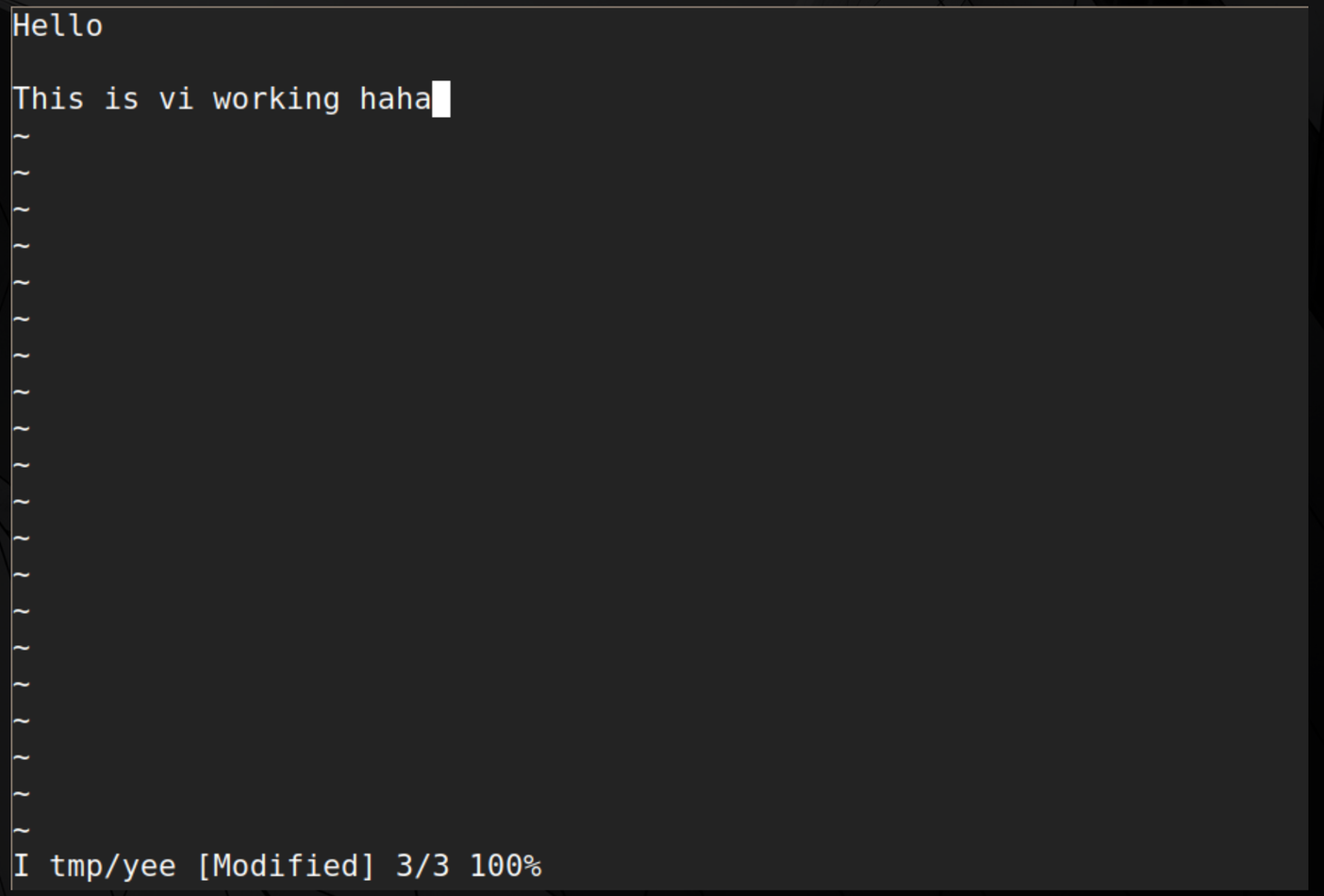
- uClibc-ng、FLAT binary (XIP)
- uClibc-ng、FDPIC (XIP)

運行 busybox

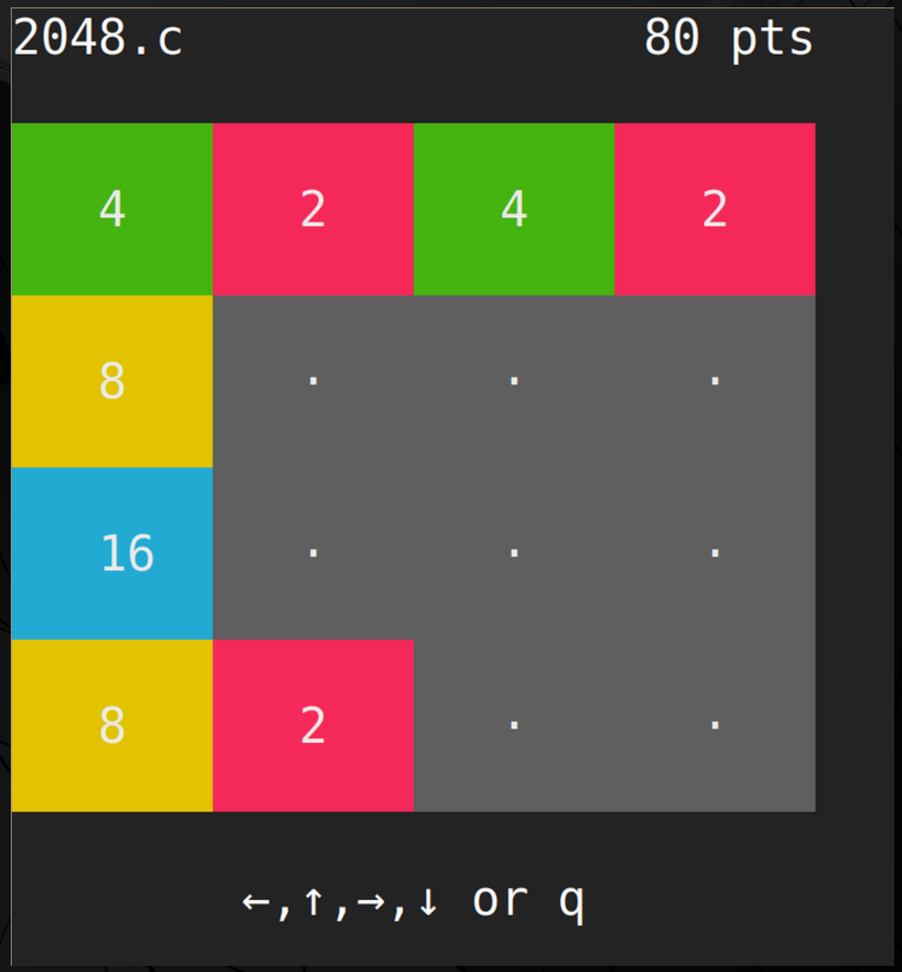
```
BusyBox v1.37.0 (2026-04-17 04:21:07 CST) hush - the humble shell
Enter 'help' for a list of built-in commands.

~ # uname -a
Linux (none) 7.0.0 #19 Fri Apr 17 04:51:22 CST 2026 armv7ml GNU/Linux
~ #
```

運行 vi (文字編輯器)



跑遊戲



用 debugfs 進行 memory profile

啟動後剩 116 KiB free RAM → 開始分析動態記憶體用量

- 在 QEMU 中加 RAM
- 用 debugfs 追蹤 slab 的用量來源

發現 stm32_pinctrl_gpio 相關 driver 分配了異常多空間

→ 修剪 device tree

啟動後還剩 228 KiB

延伸討論與未來展望

之後，我與黃敬群老師在 STM32F429 上進一步討論

- 相似軟體模型應用在 8 MiB RAM 的場景
- 啟用更多 Linux 核心軟體堆疊與週邊（LCD display、touchscreen）
- 在數 MiB RAM 情況下運行觸控式人機介面

並且近期在 Embedded Linux Conference North America 中發表我們的成果。

延伸討論與未來展望

- 評估 Linux 核心的即時處理能力
- 評估精簡核心對降低功耗與啟動時間的成效

Open Source Everything

開發紀錄

專案