Overview of SDK Framework on Real-time OS (RTOS)

Wei-Lun Hsu

Agenda

- Issues of development
- Target features of SDK framework
- Design points of device level
- Design points of a component

Issues of development (1)

- Internal
  - Source code
    - Stable vs. Developing (co-working)
    - Multi classes of components or libraries
    - Many compiling options
    - Coding style
- External
  - Release SDK
  - Version mapping
    - App vs. SDK
    - Version of pre-build libraries
  - Reproduce issues
  - Special executable image for the customer

Issues of development (2)

- Debugging
  - Toolchain for cross-compiling
  - Binutils usage
    - objdump
    - objcopy
    - addr2line
    - Map file
  - GDB and GDB server
- Image selection of multi-cores or DSP
- Document
- License (release source or not)

3-th party libraries for RTOS

- Operating System
  - ThreadX (license)
  - FreeRTOS
  - Zephyr
- TCP/IP stack
  - NetX (license)
  - Lwip
  - uip
- File system
  - Fatfs
  - Ext2/3/4

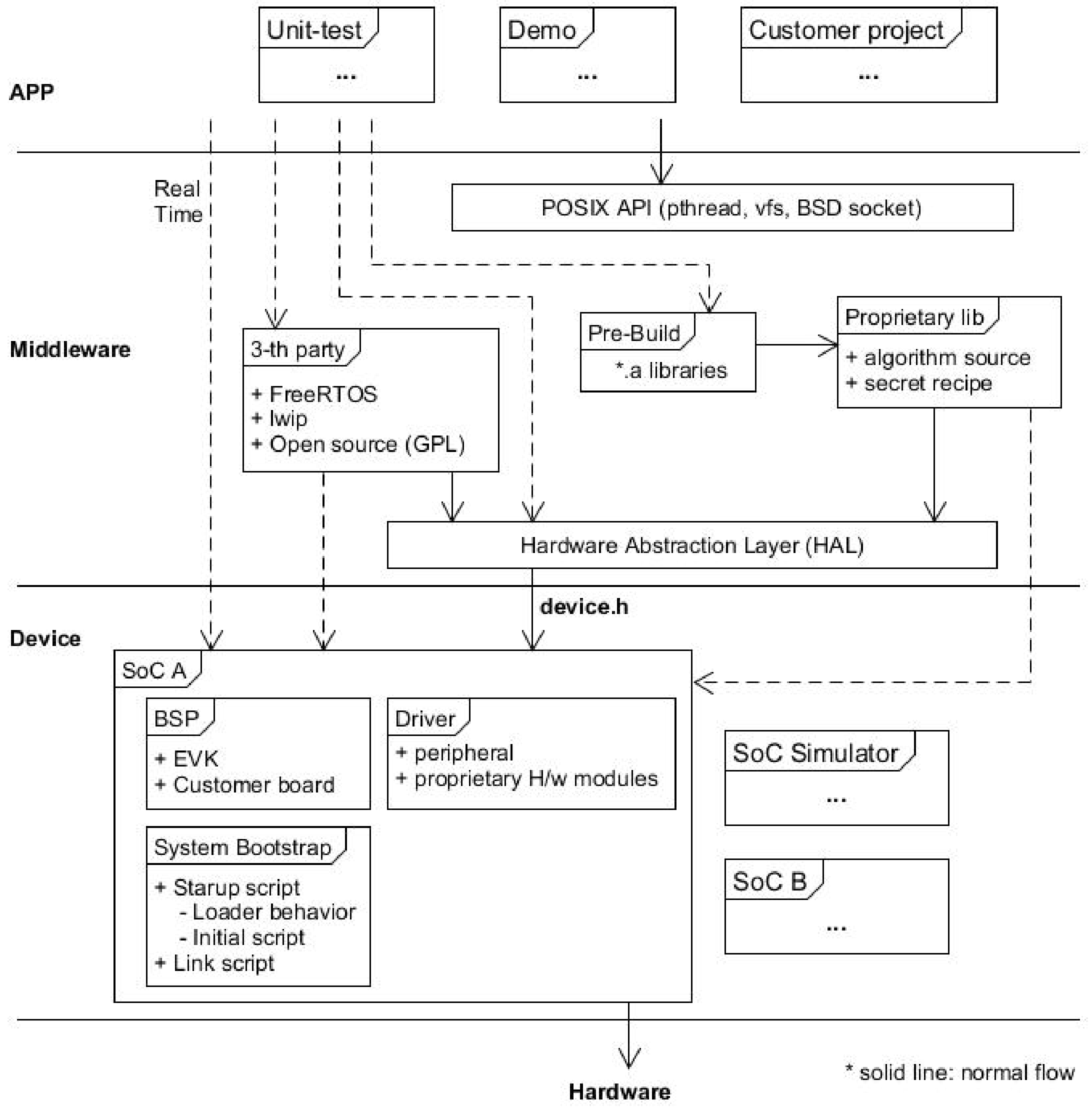
Cross-compiling tools

- Linux kernel
  - Kbuild framework
    - Kconfig (front-end)
    - Kbuild
  - Buildroot
    - Toolchain/kernel source compiling
    - Integrate libraries (> 1000)
  - Yocto
    - OpenEmbedded
- RTOS (MCU)
  - Depend on IDE
    - Keil MDK
    - IAR
    - Proprietary IDE
      - Base on Eclipse or other open source
  - Focus on GCC compiling
    - LinkIt (MTK)
    - ESP32
  - Lack a regular framework for development

Target features of SDK framework

- Portable
  - Reduce the time cost of build-up development environment
- Cross-compiling
  - Easily switch toolchain (by SoC)
  - Friendly manage compiling options
  - Flexibly add/remove components or libraries
    - Private compiling options
- Friendly development interface
  - Easy-to-use for Binutils/GDB
    - Insert a section to ELF file
    - GDB server connection flow
- Automate release flow
  - Avoid mistake from human
  - Generate pre-build libraries
- MISC
  - Document generator
  - Syntax format
  - Version tracing and authenticating

SDK Architecture



* solid line: normal flow

Code base management (1)

- Use small blocks to form a big goal
  - Modularize components and minimize the dependency
    - A lot of components need to management
  - Recipe mechanism with manifest file
  - Abstract features and enhance reuse
    - e.g. link-list utility in linux kernel
- Version control tool and code server
  - repo-git
    - Version control
    - Manifest file to connect multi-repositories
      - Reduce the effort of project management
  - Gerrit (from google)
    - All repo features support
    - Permission control
    - Code review mechanism
  - Gitlab
    - Support CI (Continuous Integration)
      - Automatic test, daily build
    - No code review server to response ‘repo upload’

Code base management (2)

- Compiling options
  - Kconfig
    - Generate the dependent or exclusive relation of options
    - Support defconfig to quickly setup setting
    - Integrate verables to definitions of source codes
    - Interactive front-end
      - help, search, …etc

Design points of device level (1)

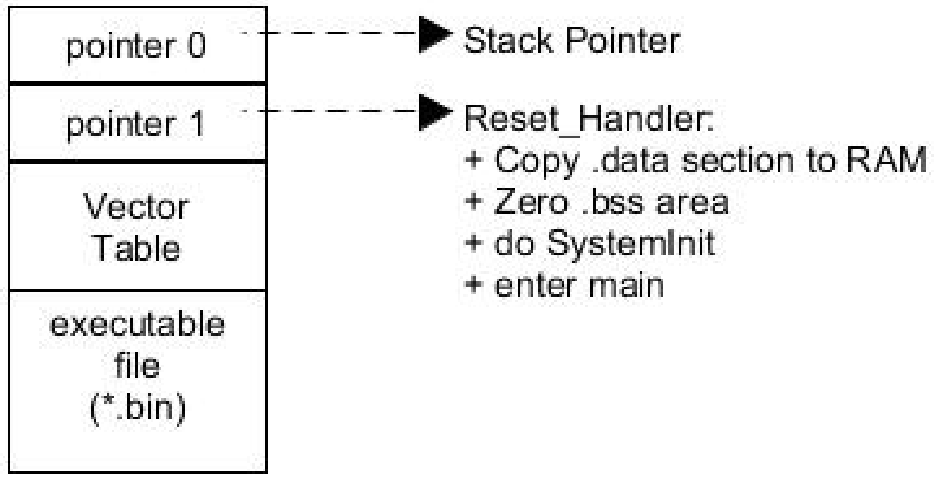
- Define the prototypes (interface) of HAL
  - Unified vs. specialized (cross discussion)
    - A category has unified interface
  - Use object/file description

```
struct peripheral_description {
    err_t (*cb_init)(void *pArgv);
    err_t (*cb_deinit)(void *pArgv);
    err_t (*cb_read)(void *pArgv);
    err_t (*cb_write)(void *pArgv);
};
```

- Bootstrap flow
  - Link script to set enter pointer
  - LMA vs. VMA
    - Execute at RAM or Flash

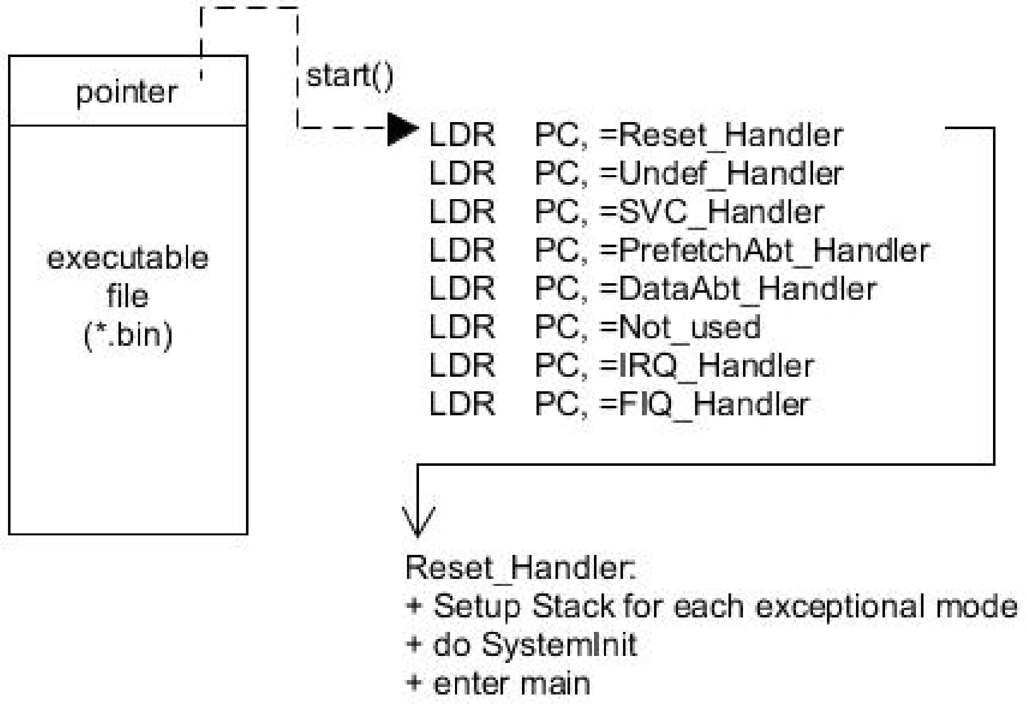
Design points of device level (2)

- Enter pointer of Cortex-M series



Design points of device level (3)

- Enter pointer of ARM9, Cortex-A5/7/9



Design points of device level (4)

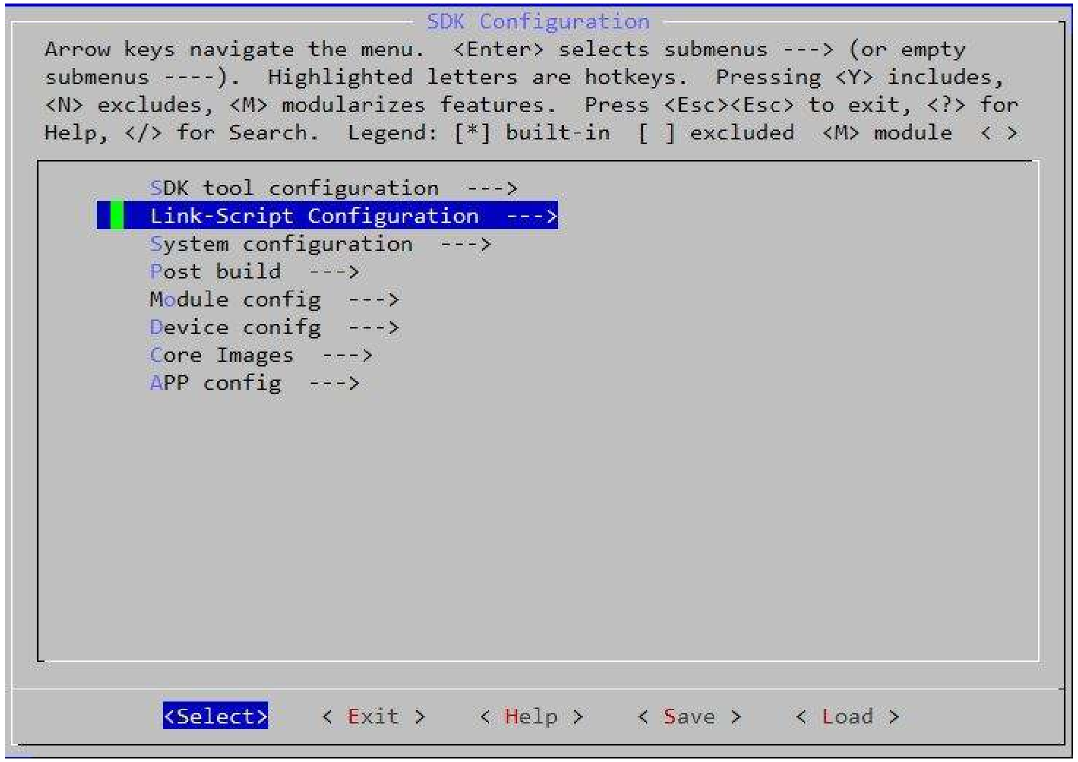
- Interrupt system
  - Vector table (Cortex-M series)
  - Hardware Interrupt Controller
  - Working mode (CPSR)
    - User/System
    - FIQ
    - Supervisor (SWI)
    - Data abort
    - IRQ
    - Undefined
- 32-bits vs. 64-bits (ARM CPU)
  - Data type (pointer vs. integer)
  - Usage of general-purpose registers
- Multi-cores
  - RPC/IPC
  - TrustZone (Arm TrustZone Firmware, ATF)
- Board selection
  - Flexibly configure by application (e.g. pin mux)
- Simulator
  - Qemu

Design points of a component

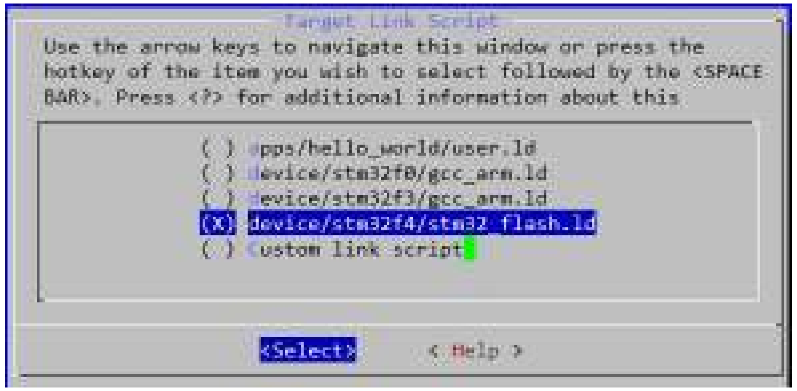
- Cross-platform
  - Simulation on PC
  - Target ASIC
- Reuse and porting effort
  - I/O abstraction
  - Memory usage and configuration
    - Hard code vs. user configure
    - Callback malloc/free
    - Memory pool
  - The dependency of libraries
    - System API (POSIX)
    - Other open sources
- Release model
  - Source code or library (*.a)
    - Library can **NOT** block program flow

Annex: kconfig

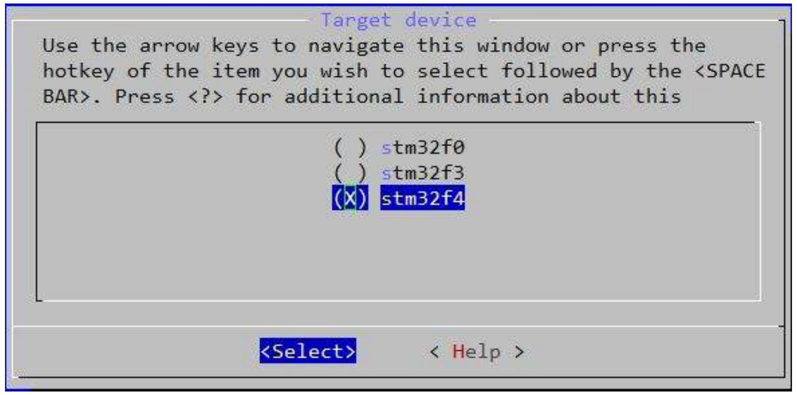
- Real-time detection for selection



```
                              SDK Configuration
 Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
 submenus ----).  Highlighted letters are hotkeys.  Pressing <Y> includes,
 <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for
 Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < >

         SDK tool configuration  --->
         Link-Script Configuration  --->
         System configuration  --->
         Post build  --->
         Module config  --->
         Device conifg  --->
         Core Images  --->
         APP config  --->

      <Select>    < Exit >    < Help >    < Save >    < Load >
```



```
                         Target Link Script
 Use the arrow keys to navigate this window or press the
 hotkey of the item you wish to select followed by the <SPACE
 BAR>. Press <?> for additional information about this

              ( ) apps/hello_world/user.ld
              ( ) device/stm32f0/gcc_arm.ld
              ( ) device/stm32f3/gcc_arm.ld
              (X) device/stm32f4/stm32_flash.ld
              ( ) Custom link script

               <Select>    < Help >
```



```
                           Target device
 Use the arrow keys to navigate this window or press the
 hotkey of the item you wish to select followed by the <SPACE
 BAR>. Press <?> for additional information about this

                    ( ) stm32f0
                    ( ) stm32f3
                    (X) stm32f4

               <Select>    < Help >
```