

EMBEDDED SYSTEM COURSE

LECTURE 2: EMBEDDED SOFTWARE DEVELOPMENT

- Hồi lại kiến thức bài cũ.
 - Phân biệt vi điều khiển/vi xử lý
 - Kiến trúc Harvard/Von Neumann, kiến trúc nào có performance tốt hơn
 - ISA là gì ?
- Giới thiệu mục đích buổi học:
 - Giúp học viên hiểu được quy trình phát triển một ứng dụng cho hệ thống nhúng.

Learning Goals



- Understanding how the code has been compiled, and generated to an image.
- Understand how does loading/debugging process happen.
- Understand most basic concepts regarding software engineering: pooling & interrupt.
- Having knowledge on how to access peripheral via memory mapped.

2

-Embedded Software, or firmware, is program that specialized for particular processor

-Embedded software developments including: Create project, compile & link to generate image; load & debug in hardware

-There are two kinds of software flow: pooling & interrupt.

-Peripheral (IO) registers are memory-mapped and therefore can be accessed as the memory.

Table of contents



1. Embedded Software Overview
2. Embedded Software Development Flow
3. Software Flow
4. Input/output Basic
5. Summary

Nội dung buổi học gồm:

- 1- Giới thiệu chung về phần mềm nhúng. Các tiêu chuẩn của phần mềm nhúng.
- 2- Giới thiệu quy trình/ Các bước phát triển một ứng dụng/phần mềm nhúng
- 3- Software Flow (base on interrupt or pooling)
- 4- Cách thức Giao tiếp với ngoại vi
- 5- Tổng kết

Table of contents



1. Embedded Software Overview
2. Embedded Software Development Flow
3. Software Flow
4. Input/output Basic
5. Summary

Nội dung buổi học gồm:

- 1- Giới thiệu chung về phần mềm nhúng. Các tiêu chuẩn của phần mềm nhúng.
- 2- Giới thiệu quy trình/ Các bước phát triển một ứng dụng/phần mềm nhúng
- 3- Thiết kế cơ bản của hệ thống nhúng (base on interrupt or pooling)
- 4- Cách thức Giao tiếp với ngoại vi
- 5- Tổng kết

Embedded Software Overview

Definition

Embedded software is **computer software**, written to control machines or devices that are not typically thought of as computers. It is typically specialized for the particular hardware that it runs on and has **time and memory constraints**. This term is sometimes used interchangeably with **firmware**

(wiki)

Topic thảo luận: Các vấn đề cần chú ý khi thiết kế hệ thống nhúng ?

- Ứng dụng có cần đáp ứng thời gian thực không ? Nếu có thì tới mức độ nào?

Phân biệt khái niệm: Hard real time (Ko cho phép có độ trễ, hệ thống safety tuyệt đối / Ví dụ nhà máy điện nguyên tử) / Soft real time (chấp nhận thời gian thực ở mức độ nào đấy. Ví dụ ứng dụng điều khiển tivi/ nhà thông minh,...).

- Độ phức tạp của ứng dụng. Cần thiết phải sử dụng RTOS hay chỉ BM/ Có cần thiết sử dụng bộ nhớ ngoài ko ?

- How to Load & Debug an application ?

Embedded Software Overview

Features:

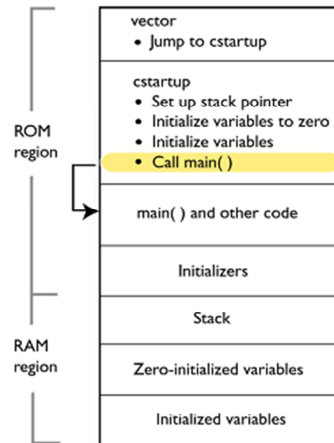
- Acts directly with and on the hardware
- Quite limited resources.
- Using a “Non-hosted environment”

- Topic thảo luận: Firmware chạy như thế nào?

Embedded Software Overview

Common Components:

- Reset vector
- Startup code
- Application code
- Libraries
- Interrupt/Exception Handler



- Topic thảo luận: Các thành phần cơ bản của 1 firmware

Reset vector – Là điểm bắt đầu của processor khi power-on or reset. Một số dòng có thể support hardware-code để khởi tạo trạng thái của vi xử lý/ Còn đa phần thì reset vector sẽ trỏ đến start-up code.

Startup code – Là phần chương trình chạy đầu tiên, có nhiệm vụ setup môi trường, khởi tạo các giá trị của processor, setup/define head & stack. Khởi tạo các biến được gán giá trị đầu, clear các vùng nhớ còn lại. Sau đó sẽ call `_main()` function. (Thường start-up code được cung cấp bởi hãng)

Application code – Ứng dụng của lập trình viên. Thường ứng dụng này được phát triển = asm, C or C++.

→ topic: Ưu nhược điểm của từng ngôn ngữ.

Libraries - Là tập thư viện được cung cấp bởi compiler (cho các hàm primitive C/lib cho xử lý số tín hiệu) hoặc thư viện của nhà sản xuất phát triển (device driver)

Interrupt/Exception handler – Là các đoạn code được viết phục vụ cho/ (và được chạy ngay khi) sự kiện được trigger bởi hardware (sometimes: có thể do software). Các sự kiện này thường xảy ra ngẫu nhiên.

→ Giới thiệu khái niệm: interrupt latency/ interrupt time .

→ Topic: Tiêu chí của Hàm phục vụ ngắt.

Embedded Software Overview

What is needed to start:

- Development suites
- Development board
- Debug Adapter
- Software device driver
- Documents and other resources.

- Trình bày cho học viên: Những thứ cần để phát triển một firmware bao gồm:
- Tool (Môi trường phát triển): IAR, Keil, gcc, kds, etc
- Board để phát triển: TWRK64, TWRK53, KL46 freedom.
- Debug Adapter: Jlink, Ulink, built-in USB debug adaptor
- Driver của ngoại vi
- Tài liệu (User guide, RM); các ví dụ có sẵn của nhà sản xuất.

Table of contents



1. Embedded Software Overview
2. Embedded Software Development Flow
3. Software Flow
4. Input/output Basic
5. Summary

Embedded Software Development Flow

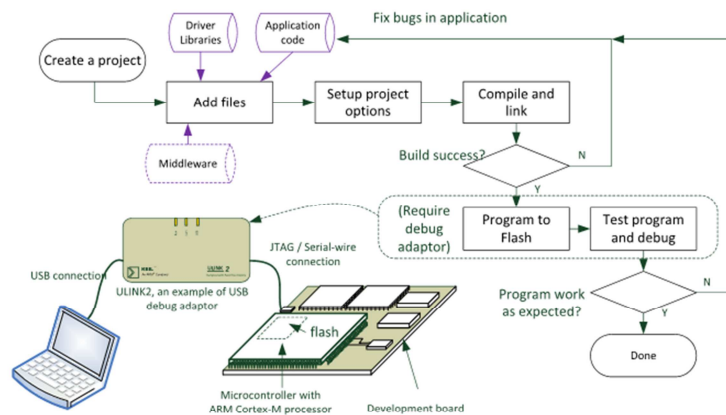
Software Development Steps in IDE

- Create project
- Setup project option
- Compile & Link
- Flash Program
- Execute & Debug

- Giới thiệu quy trình chung để phát triển firmware.

Embedded Software Development Flow

Development Flow



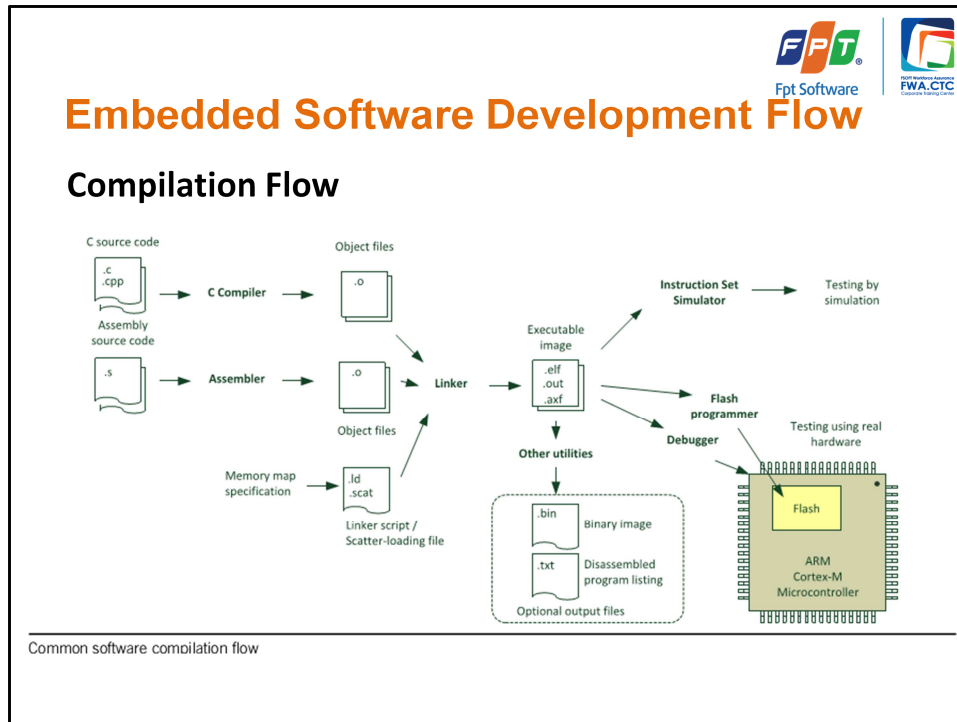
A simplified software development flow

Giới thiệu các bước Implement một embedded application./ Giảng viên có thể minh họa trực tiếp trên Keil (Tham khảo tại: https://www.youtube.com/watch?v=Oy_cA1d6TqE).

1- Tạo project:

- Add các thư viện (driver lib, middleware) cần dùng
- Implement application code
- Modify default linker → which is suitable for the application.

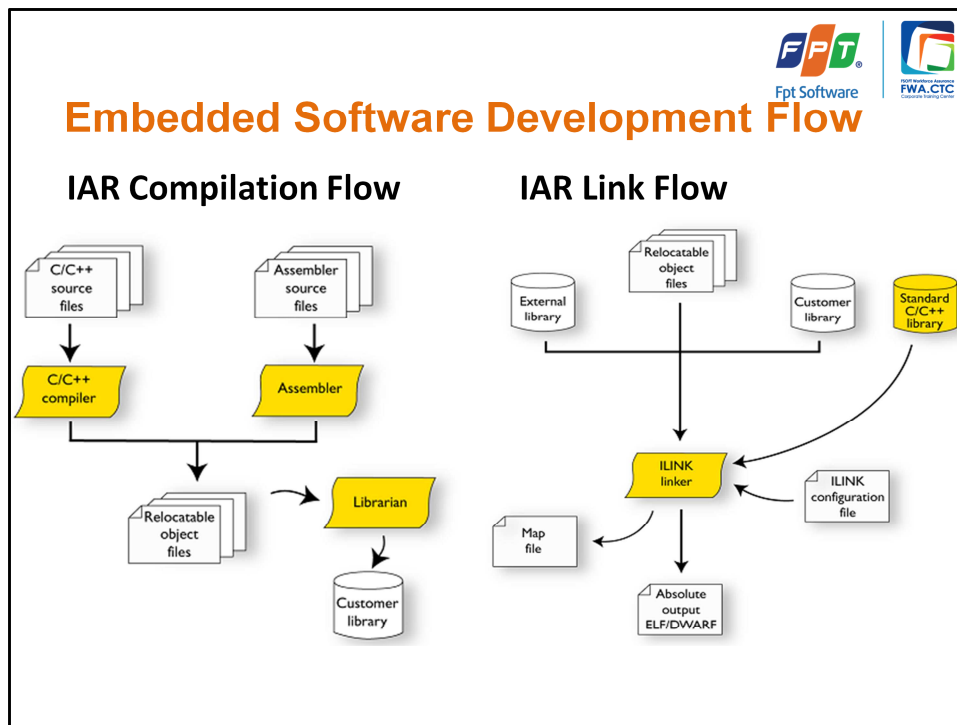
2- Chỉnh project options để phù hợp với ứng dụng (nếu cần thiết).



Giới thiệu quá trình compile từ một project gồm các file .c, .s thành file .bin để load được xuống board:

- + compiler: compile C source file to object
- + assembler: compile asm file → object file

Linker: using linker script to link/combine all object files to one executable image.



Minh họa quá trình build.compile.link trên IAR

Table of contents



1. Embedded Software Overview
2. Embedded Software Development Flow
3. Software Flow
4. Input, output and Peripherals access
5. Summary

Topic thảo luận: Phân biệt Interrupt & Pooling.

Software Flow

Pooling

- Continuously checking the status of a peripheral; e.g. read data from an input keyboard.
- Polling is relatively straightforward in design and programming with the sacrifice of system performance.

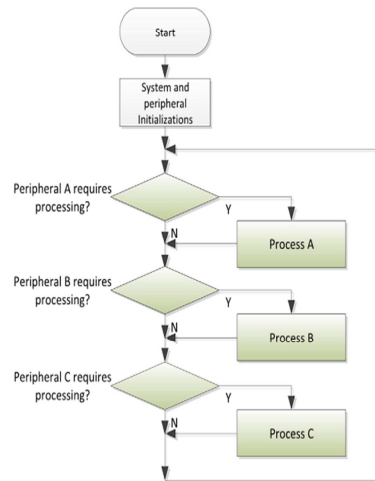
Interrupt

- Device “interrupts” CPU to indicate that it needs service.
- These events only occur if the interrupt is enabled.
- A handler (software to service the interrupt) is executed.
- CPU returns to where it left off in the main program.

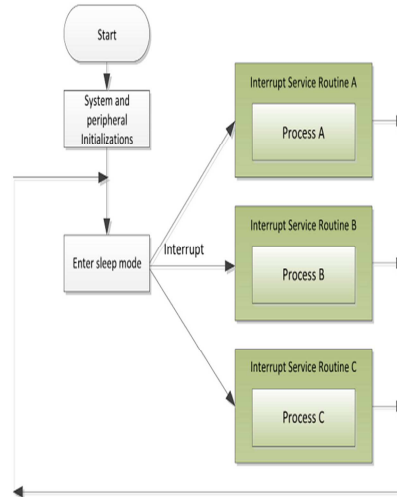
Topic thảo luận: Phân biệt Interrupt & Pooling.

Software Flow

Pooling



Interrupt



Thảo luận: ưu nhược điểm của pooling & interrupt handler

Software Flow



Interrupt Process:

- CPU waits until the current instruction has finished being executed.
- Save the contents of internal registers of the CPU & the state information within Control Unit
- The PC is loaded with address of the Interrupt Service Routine (ISR)
- ISR is executed.
- Return program from interrupt.

Topic thảo luận: Trình bày flow của một chương trình ngắt.

Software Flow



Interrupt Handler Features:

- Differs from subroutine because it is executed at any time due to interrupt, not due to Call
- Should be implemented as small as possible
- Should be executed in short-time.

- Thảo luận: Tiêu chí một chương trình ngắt tốt ?

Table of contents

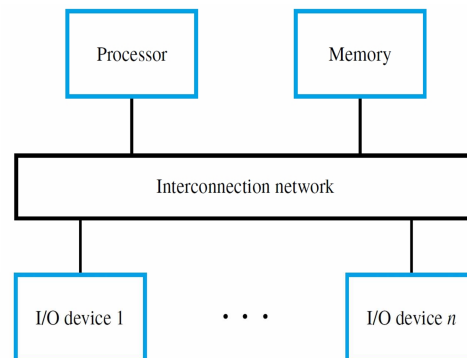


1. Embedded Software Overview
2. Embedded Software Development Flow
3. Software Flow
- 4. Input/output Basic**
5. Summary

Input Output Basic

Accessing I/O Devices

- Computer system components communicate through an interconnection network
- Memory-mapped I/O allows I/O registers to be accessed as memory locations. As a result, these registers can be accessed using only Load and Store instructions



Input Output Basic



I/O Device Interface

- Provides the means for data transfer and exchange of status and control information
- Includes data, status, and control registers accessible with Load and Store instructions
- Memory-mapped I/O enables software to view these registers as locations in memory

Table of contents



1. Embedded Software Overview
2. Embedded Software Development Flow
3. Software Flow
4. Input/output Basic
5. Summary

Summary



- Embedded Software, or firmware, is program that specialized for particular processor
- Embedded software developments including: Create project, compile & link to generate image; load & debug in hardware
- There are two kinds of software flow: pooling & interrupt.
- Peripheral (IO) registers are memory-mapped and therefore can be accessed as the memory.

Question and Answer



Thanks for your attention !

Trả lời câu hỏi nếu có.

Có thể hỏi lại học viên để đánh giá mức độ tiếp thu bài giảng.

Copyright



- This course including **Lecture Presentations, Quiz, Mock Project, Syllabus, Assignments, Answers** are copyright by FPT Software Corporation.
- This course also uses some information from external sources and non-confidential training document from Freescale, those materials comply with the original source licenses.