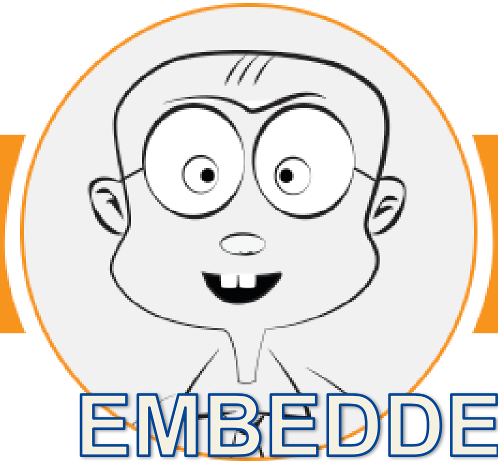


EMBEDDED SYSTEM COURSE

LECTURE 2: EMBEDDED SOFTWARE DEVELOPMENT

Learning Goals

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

Table of contents

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

Table of contents

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

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)

Embedded Software Overview

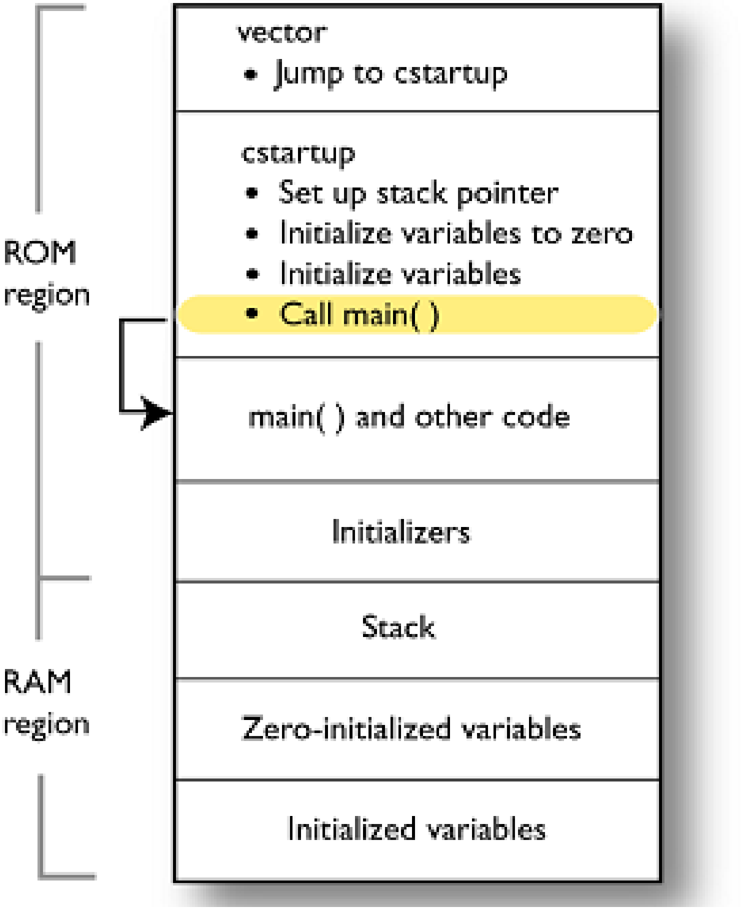
Features:

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

Embedded Software Overview

Common Components:

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



Embedded Software Overview

What is needed to start:

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

Table of contents

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

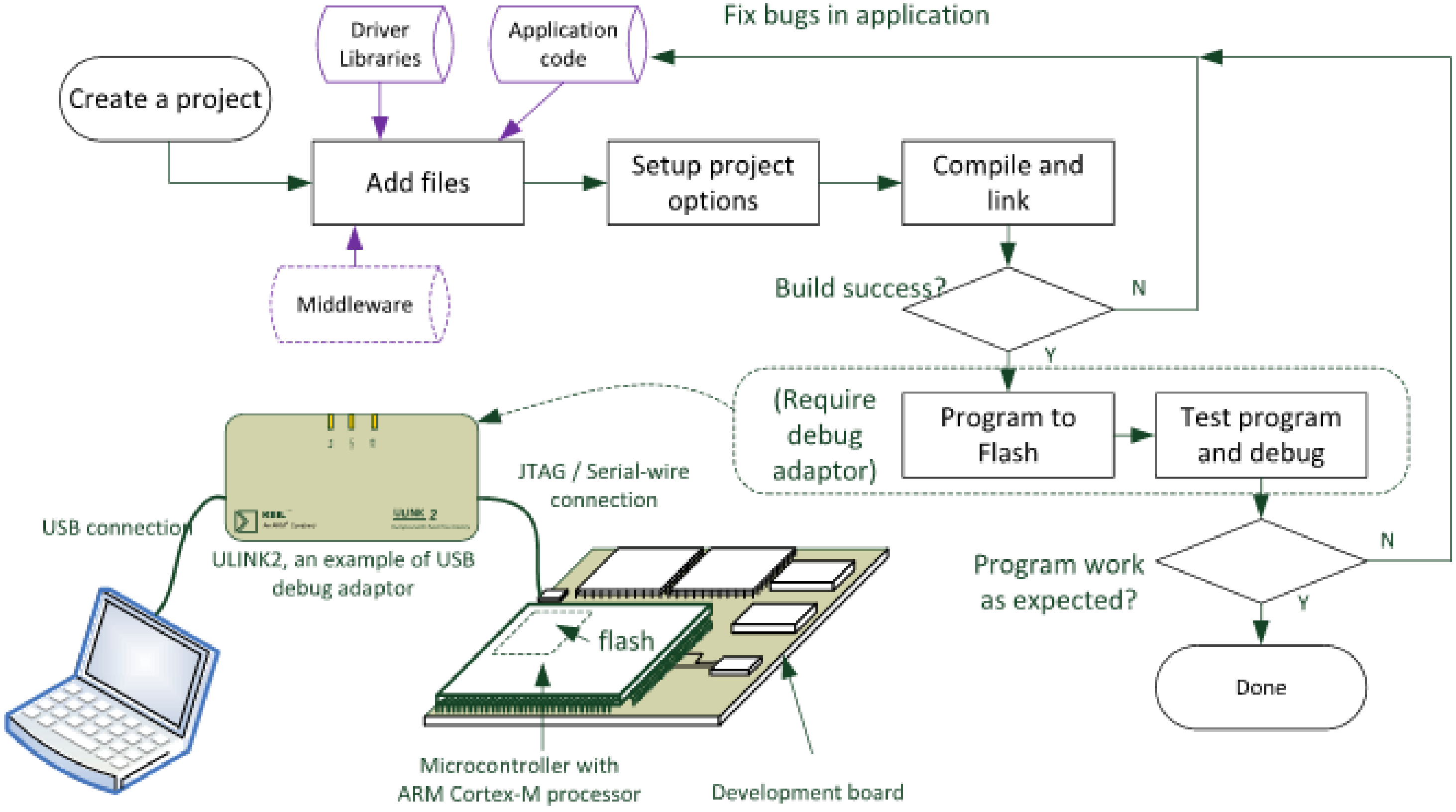
Embedded Software Development Flow

Software Development Steps in IDE

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

Embedded Software Development Flow

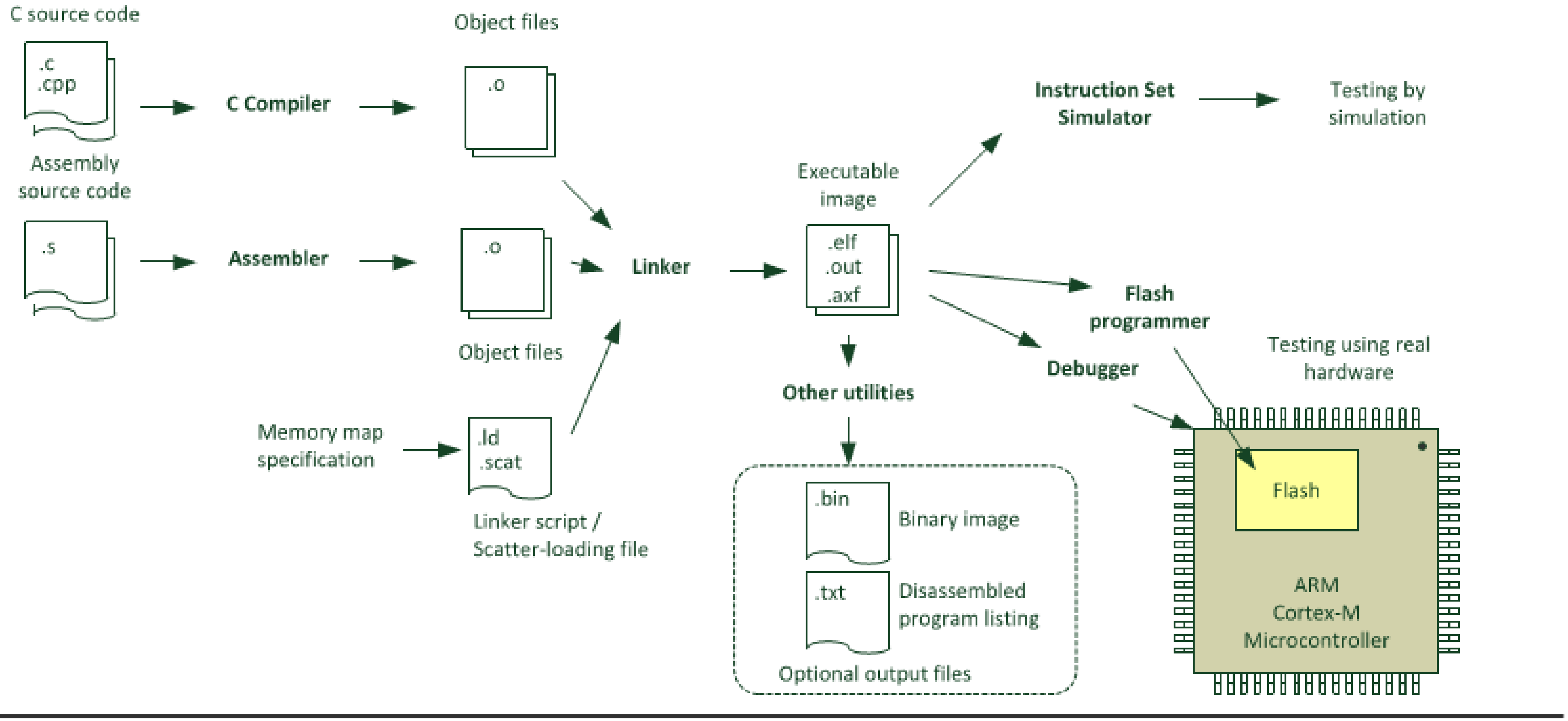
Development Flow



A simplified software development flow

Embedded Software Development Flow

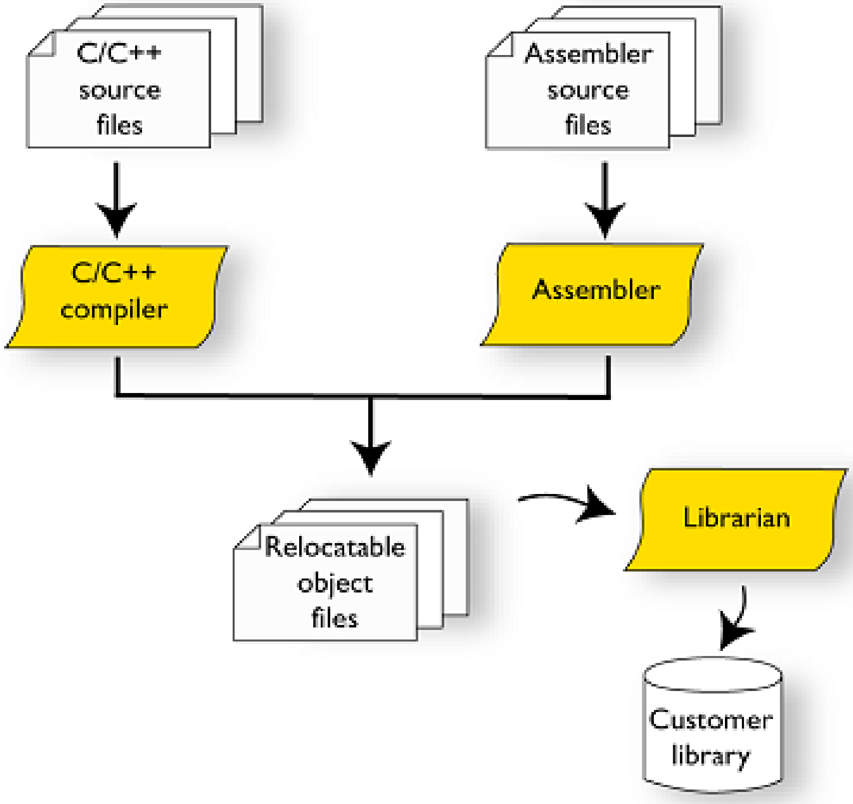
Compilation Flow



Common software compilation flow

Embedded Software Development Flow

IAR Compilation Flow



IAR Link Flow

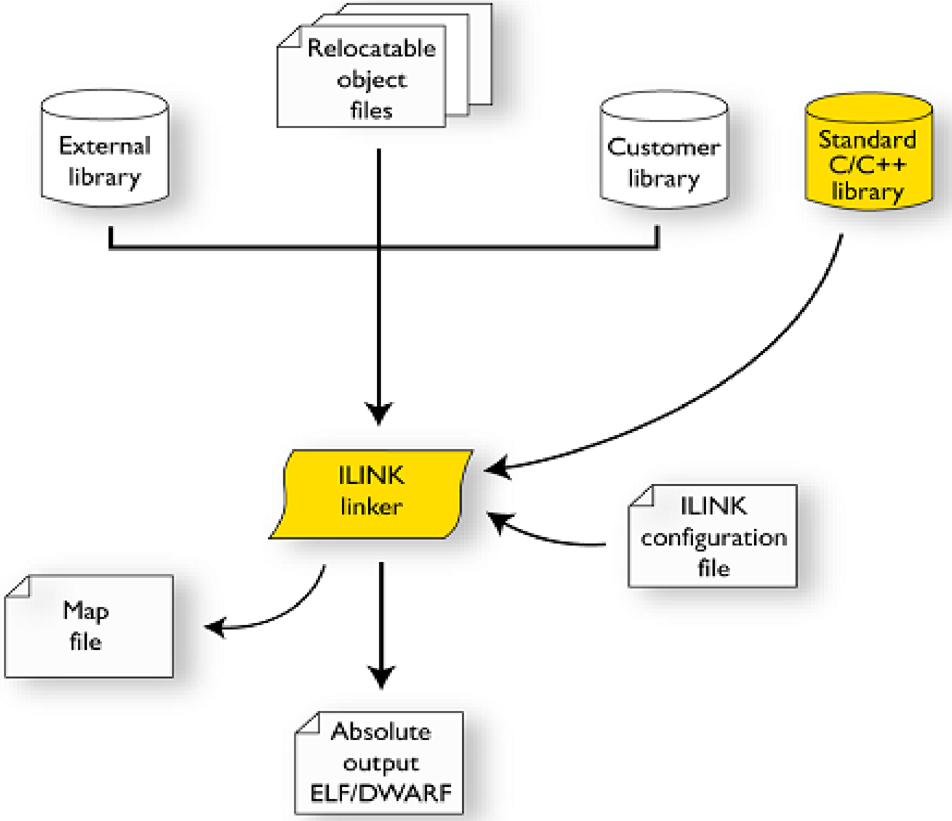


Table of contents

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

Software Flow

Pooling

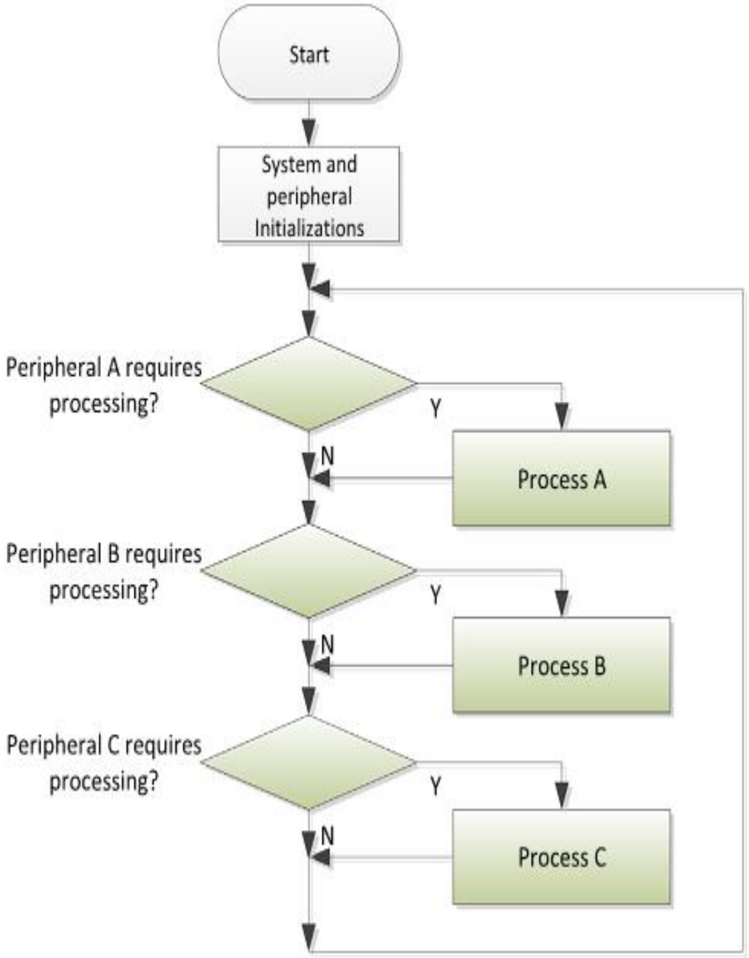
- Continuously checking the status of a peripheral; e.g. read data from an input keyboard.
- Polling is relatively straightforward in design and programming with the sacrifice of system performance.

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.

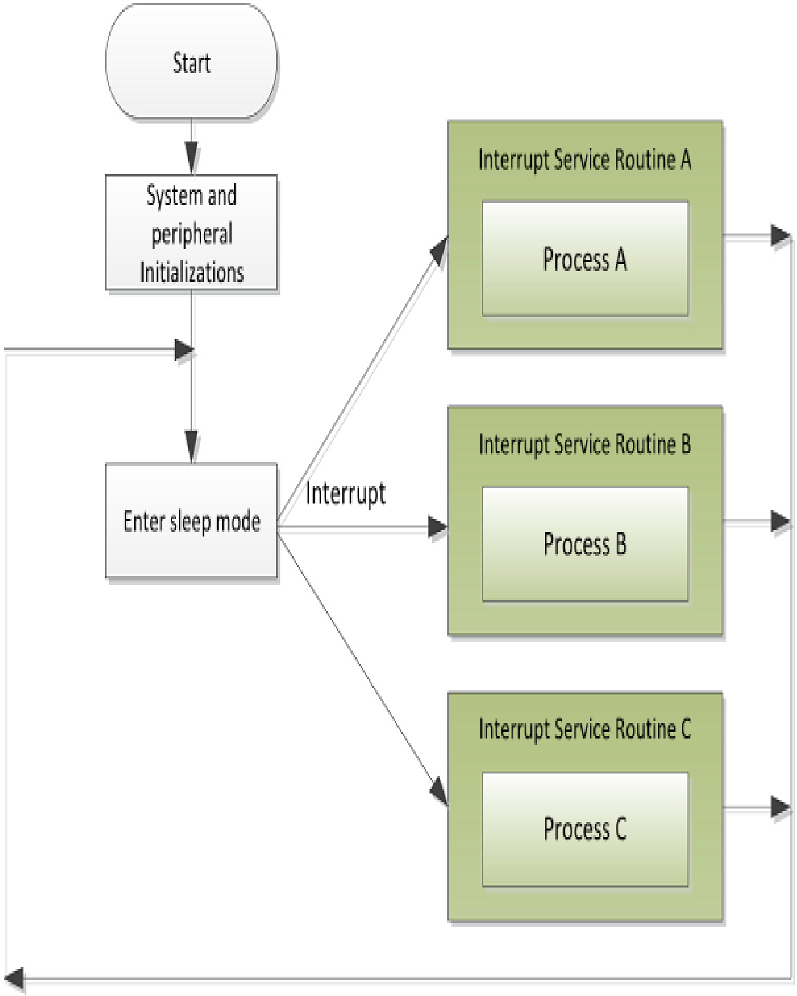
Software Flow

Pooling



Start
System and peripheral Initializations
Peripheral A requires processing?
Y
N
Process A
Peripheral B requires processing?
Y
N
Process B
Peripheral C requires processing?
Y
N
Process C

Interrupt



Start
System and peripheral Initializations
Enter sleep mode
Interrupt
Interrupt Service Routine A
Process A
Interrupt Service Routine B
Process B
Interrupt Service Routine C
Process C

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.

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.

Table of contents

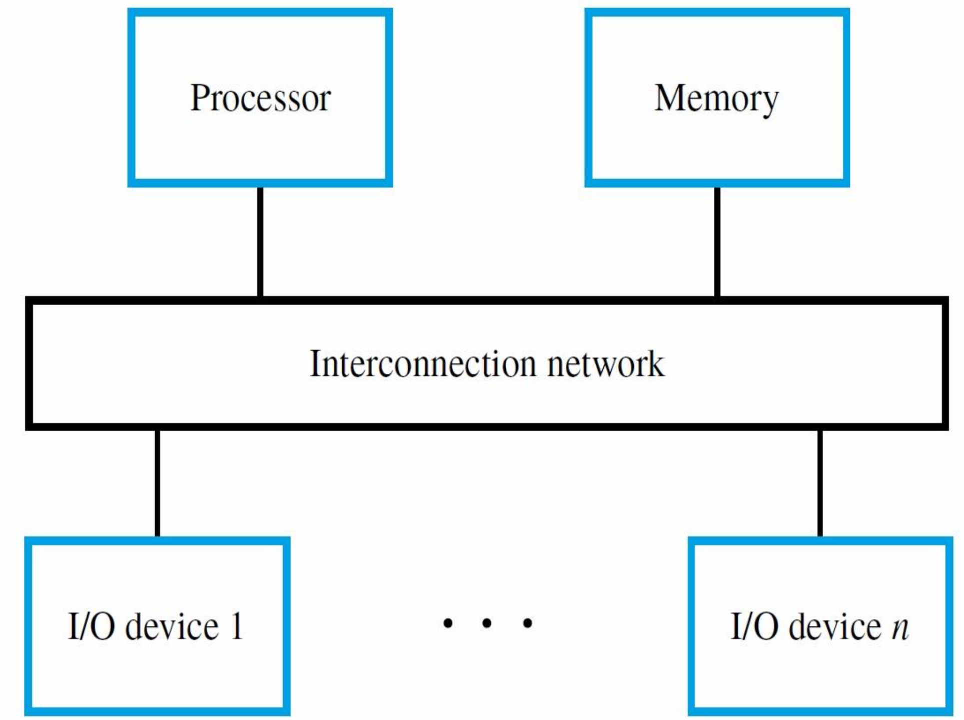
1. Embedded Software Overview
2. Embedded Software Development Flow
3. Software Flow
4. Input/output Basic
5. Summary

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 imagine; 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 !

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.