Revision

EE5182 Microcontrollers and Embedded Systems

Nimal Skandhakumar

nimal@sjp.ac.lk

https://academic.nimal.info/ee5182/

Course Outline

1. Introduction
2. Embedded design concepts and definitions
3. System architectures
4. Introduction to ARM core architecture
5. On-chip peripherals
6. Communication
7. Hardware components of embedded systems
8. Interfacing
9. Real time operating systems
10. RTOS examples
11. Revision
12. ~~Basic embedded programming~~
13. ~~Programmable logic controllers (PLC)~~

Introduction

Systems

- A system is a way of working, organizing or doing one or many tasks according to a fixed plan, program or set of rules.
- A system is also an arrangement in which all its units assemble and work together according to the plan or program.

Embedded Systems

- An Embedded System is one that has computer hardware with software embedded in it as one of its important components.
- Its software embeds in ROM (Read Only Memory). It does not need secondary memories as in a personal computer.
- It has Hardware
  - Processor, Timers, Interrupt controller, I/O Devices, Memories, Ports, etc.
- It has main Application Software
  - Which may perform concurrently the series of tasks or multiple tasks.
- It has Real Time Operating System (RTOS)
  - RTOS defines the way the system work. Which supervise the application software. It sets the rules during the execution of the application program. A small scale embedded system may not need an RTOS.

Embedded Systems

What's different?

- Real-time operation
- Size
- Cost
- Time
- Reliability
- Safety
- Energy
- Security

How to classify?

- Small Scale Embedded System
- Medium Scale Embedded System
- Sophisticated Embedded System

Embedded Design Concepts and Definitions

Embedded systems basics

- Embedded systems are designed for a specific task.
  - Although they use computer techniques, they cannot be used as a general purpose computer using a variety of different programmes for different task. In this way their function can be focussed onto what they need to do, and they can accordingly be made cheaper and more efficiently.
- The software for embedded systems is referred to as firmware.
  - Rather than being stored on a disc, where many programmes can be stored, the single programme for an embedded system is normally stored on chip and it is referred to as firmware.

Design Concepts

Hardware

- When using an embedded system there is a choice between the use of a microcontroller or a microprocessor.
  - ***Microcontroller based systems***
  - ***Microprocessor based systems***

Software

- One of the key elements of any embedded system is the software that is used to run the microcontroller. There is a variety of ways that this can be written:
  - ***Machine code***
  - ***Programming language***

Characteristics of Embedded Systems

- Must be dependable:
  - Reliability
  - Maintainability
  - Availability
  - Safety
  - Security
- Must be efficient:
  - Energy efficient
  - Code-size efficient (especially for systems on a chip)
  - Run-time efficient
  - Weight efficient
  - Cost efficient
- Dedicated towards a certain application
- Dedicated user interface (no mouse, keyboard and screen)
- Many ES must meet real-time constraints

System Architectures

Microprocessor

- A single chip that contains a whole CPU
  - Has the ability to fetch and execute instructions stored in memory
  - Has the ability to access external memory, external I/O and other peripherals
- Examples:
  - Intel Core or AMD Athlon in desktops/notebooks
  - ARM processor in smartphones

Microcontroller

- Essentially a microprocessor with on-chip memories and I/O devices
- Designed for specific functions
- All in one solution
  - Reduction in chip count
  - Reduced cost, power, physical size, etc.

Microcontroller Classifications

- Classification According to Number of Bits
  - 8-bit / 16-bit / 32-bit
- Classification According to Memory Devices
  - Embedded memory / External memory
- Classification According to Memory Architecture
  - Harvard Memory Architecture / Princeton Memory Architecture
- Classification According to Instruction Set
  - Complex Instruction Set Computer / Reduced Instruction set Computer

Compare Microcontrollers

Types of Microcontrollers

- 8051
- PIC
- AVR
- ARM

Comparison Criteria

- Bus width
- Communication Protocols
- Speed
- Memory
- ISA
- Memory Architecture
- Power Consumption
- Families
- Manufacturer
- Cost

Introduction to ARM core architecture

ARM processors vs. ARM architectures

- ARM architecture
  - Describes the details of instruction set, programmer's model, exception model, and memory map
  - Documented in the Architecture Reference Manual
- ARM processor
  - Developed using one of the ARM architectures
  - More implementation details, such as timing information
  - Documented in processor's Technical Reference Manual

ARM processor lines

- The Cortex-M profile
  - Processors of the M profile are optimized for cost sensitive and microcontroller applications, like automotive body electronics, smart sensors.
- The Cortex-A profile
  - It aims at high-end applications running open and complex OSs, like smartphones, tablets, netbooks, eBook readers.
- The Cortex-R profile
  - It marks processors for real time applications, like mass storage or printer controllers.
- The SecureCore profile
  - The ARM SecurCore™ processor family provides processors with security features for applications like smartcards, pay TV, eGovernement.

ARM vs. x86

- ARM processors require significantly fewer transistors than typical PC processors because of the fact that it has a RISC-based design.
- Fewer transistors minimizes power use, heat and production cost. All qualities that are preferred for battery-powered devices such as laptops, tablets, and smartphones.
- On the other hand x86 processors usually consumes a lot of energy but the are also a lot faster.
- This makes x86 based processors ideal for desktops, gaming and super computer that require speed more than energy efficiency.

On-chip Peripherals

System on a Chip (SoC)

- All the necessary components of a computer system are embedded on a single silicon die.
- A typical SoC will contain:
  - A processor
  - Onboard execution memory (SRAM)
  - Many microcontrollers may contain FLASH memory for program storage
  - Peripheral systems and interfaces connected to the processing core via a SoC bus

System on a Chip (SoC)

Advantages

- Decreased power consumption
- Increased reliability
- Smaller board space
- Can be cheaper when using ready to go components
- More tightly integrates SoCs will result in smaller electronic products that use less power, are faster, and more reliable
- Nano scale robots for fighting human diseases, curing diseases.

Disadvantages

- Extremely high design cost (for the actual chip)
- Large silicon space may be required
- Component testing may be difficult
- Prototyping may take longer
- Intellectual property (IP) issues

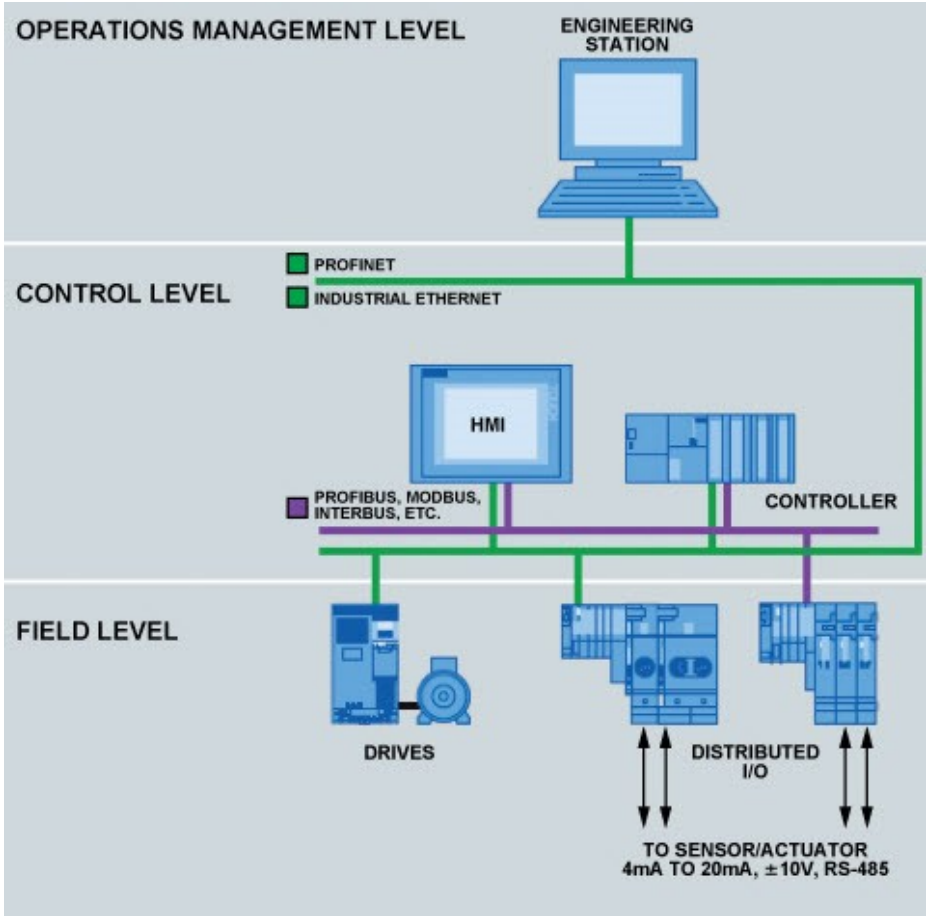
On-chip Peripherals

- On-chip flash program memory
- On-chip static RAM
- Interrupt controller
- General purpose timers/external event counters
- Watchdog timer
- Real-time clock
- Pulse width modulator
- Fast general purpose parallel I/O
- ADC
- DAC
- USB
- UARTs
- I2C
- SPI
- SSP

Communication

Hierarchical levels of communication

- **Device Level:** This lowest level consists of field devices such as sensors and actuators of processes and machines.
- **Control Level:** This level consists of controllers, distributed control units, and computer systems. The tasks of this level include configuring automation devices, loading of program data and process variables data, supervising control, historical archiving, etc.
- **Information Level:** This is the top level of the industrial automation system which gathers the information from its lower level i.e., control level.



Data communication in control systems

- Control systems need to communicate:
  - Between controllers and plants (controlled devices)
  - Between controllers and sensors
  - Between controllers and other related controllers
  - Between controllers and systems managers/monitors
- Embedded systems communication concepts:
  - Point-to-point networks
  - Shared media networks
  - Event based communication
  - State based communication

Basic communication methods

- It is essential to understand some of the basic communication methods that can be used to interconnect control systems.
- Characteristics of communication methods:
  - Simplex, Duplex & Semi Duplex
  - Serial Vs Parallel
  - Synchronous Vs Asynchronous
  - Data Throughput

Hardware Components of Embedded Systems

Hardware Components of Embedded Systems

- Electronic / Electrical / Electro-mechanical / Mechanical
- A sensor is a device that receives and responds to a signal.
  - This signal must be produced by some type of energy, such as heat, light, motion, or chemical. Once a sensor detects one or more of these signals, it converts it into an analog or digital representation of the input signal.
- A transducer is a device which converts one form of energy into another.
  - Transducers are used in all aspects of life to measure changes in the environment, to enhance everyday applications, and to learn more about the world around us.
- An actuator is a device that converts energy into motion.
  - Therefore, it is a specific type of a transducer. When the output of the transducer is converted to a readable format, the transducer is called a sensor.

Common Sensors and Transducers

Quantity being Measured	Input Device (Sensor)	Output Device (Actuator)
Light Level	Light Dependant Resistor (LDR), Photodiode, Photo-transistor, Solar Cell	Lights & Lamps, LED's & Displays, Fibre Optics
Temperature	Thermocouple, Thermistor, Thermostat, Resistive Temperature Detectors	Heater, Fan
Force/Pressure	Strain Gauge, Pressure Switch, Load Cells	Lifts & Jacks, Electromagnet, Vibration
Position	Potentiometer, Encoders, Reflective/Slotted Opto-switch, LVDT	Motor, Solenoid, Panel Meters
Speed	Tacho-generator, Reflective/Slotted Opto-coupler, Doppler Effect Sensors	AC and DC Motors, Stepper Motor, Brake
Sound	Carbon Microphone, Piezo-electric Crystal	Bell, Buzzer, Loudspeaker

Sensor Transfer Function

- $S=f(s)$
  - S = output signal;
  - s = stimulus;
  - $f(s)$ = functional relationship
- For binary sensors:
  - S = 1 if s > 0 and S = 0 if s < 0.
- The ideal functional form for an analogue measuring device is a simple proportional relationship, such as:
- $S=C+ms$
  - C = output value at a stimulus value of zero
  - m = constant of proportionality (sensitivity)

Interfacing

Interfacing

- Transfer of data among processors and memories is known as interfacing.
  - Processor ⟶ Process data
  - Memory ⟶ Storage
  - Buses ⟶ Communication

Basic protocol concepts

- **Actor**: is the processor or memory involved in data transfer.
- A protocol involves two actors: a master, and a servant (slave).
- A master initiates the data transfer (usually general-purpose processor), and the servant responds to the initiation request (usually memories and peripherals).
- **Data direction**: the direction that the transferred data moves btw. actors(receiving or sending data).
- **Addresses**: a special type of data used to indicate where regular data should go to or come from (used to address memory locations , peripherals and peripheral's registers).

Memory-Mapped I/O and Standard I/O

- They are two bus-based methods for microprocessor to communicate with peripherals.
- In memory-mapped I/O, peripherals occupy specific addresses in the existing address space.
    - e.g., Bus has 16-bit address, lower 32K addresses may correspond to memory, and upper 32k addresses may correspond to peripherals.
- In standard I/O (I/O-mapped I/O), the bus includes an additional pin (M`/IO), to include whether the access is to memory or peripheral.
    - e.g., Bus has 16-bit address, all of them for memory addressing if M`/IO=0, and all of them for I/O addressing if M`/IO=1.

Real Time Operating Systems (RTOS)

Real Time Systems

- A system is said to be Real Time if it is required to complete it's work and deliver it's services on time.
- Real Time Embedded:
  - Nuclear reactor control; Flight control; Basically any safety critical system
  - GPS; Mobile phone
- Real Time, but not Embedded:
  - Stock trading system
  - Skype
- Embedded, but not Real Time:
  - Home temperature control
  - Sprinkler system
  - Washing machine, refrigerator, etc.
  - Blood pressure meter

- Characteristics of real time systems:
  - Event-driven, reactive
  - High cost of failure
  - Concurrency/multiprogramming
  - Stand-alone/continuous operation
  - Reliability/fault-tolerance requirements
  - Predictable behavior

What's Important in Real Time Systems

- Metrics for real-time systems differ from that for time-sharing systems.

	Time-Sharing Systems	Real-Time Systems
Capacity	High throughput	Schedulability
Responsiveness	Fast average response	Ensured worst-case response
Overload	Fairness	Stability

- schedulability is the ability of tasks to meet all hard deadlines
- latency is the worst-case system response time to events
- stability in overload means the system meets critical deadlines even if all deadlines cannot be met

RTOS

Features of RTOS

- Scheduling
- Resource Allocation
- Interrupt Handling
- Other issues like kernel size

Scheduling in RTOS

- More information about the tasks are known
  - No of tasks
  - Resource Requirements
  - Release Time
  - Execution time
  - Deadlines
- Being a more deterministic system better scheduling algorithms can be devised.

RTOS Examples

- For one RTOS (µC/OS, FreeRTOS, Femto OS, Nut/OS, DuinOS)
  - Task States
  - Task Scheduling
  - Task Management
  - Interrupts
  - System Clock
  - Advantages & Disadvantages

Course Information

https://academic.nimal.info/ee5182/