Queen Mary
University of London
Science and Engineering

School of Electronic Engineering and Computer Science
QMUL-BUPT Joint Programme

EBU6475 Microprocessor System Design
EBU5476 Microprocessors for Embedded Computing

Introduction to Embedded Systems Design

Last updated: 26 February, 2022

arm
University Program Education Kits

Introduction

- What is an Embedded System?
  - Application-specific computer system
  - Built into a larger system
- Why add a computer to the larger system?
  - Better performance
  - More functions and features
  - Lower cost
  - More dependability

- Economics
  - Microcontrollers (used for embedded computers) are high-volume, so recurring cost is low
  - Nonrecurring cost dominated by software development
- Networks
  - Often embedded system will use multiple processors communicating across a network to lower parts and assembly costs and improve reliability

Options for Building Embedded Systems

	Implementation	Design Cost	Unit Cost	Upgrades & Bug Fixes	Size	Weight	Power	System Speed
Dedicated Hardware	Discrete Logic	low	mid	hard	large	high	?	very fast
	ASIC	high ($500K/ mask set)	very low	hard	tiny - 1 die	very low	low	extremely fast
	Programmable logic – FPGA, PLD	low	mid	easy	small	low	medium to high	very fast
Software Running on Generic Hardware	Microprocessor + memory + peripherals	low to mid	mid	easy	small to med.	low to moderate	medium	moderate
	Microcontroller (int. memory & peripherals)	low	mid to low	easy	small	low	medium	slow to moderate
	Embedded PC	low	high	easy	medium	moderate to high	medium to high	fast

Example:
Embedded System: Bike Computer

- Functions
  - Speed and distance measurement
- Constraints
  - Size
  - Cost
  - Power and Energy
  - Weight
- Inputs
  - Wheel rotation indicator
  - Mode key
- Output
  - Liquid Crystal Display
- Low performance MCU
  - 8-bit, 10 MIPS

Motor Control Unit

- Functions
  - Motor control
  - System communications
  - Current monitoring
  - Rotation speed detection
- Constraints
  - Reliability in harsh environment
  - Cost
  - Weight

- Many Inputs and Outputs
  - Discrete sensors & actuators
  - Network interface to rest of car
- High Performance MCU
  - 32-bit, 256 KB flash memory, 80 MHz

Benefits of Embedded Computer Systems

- Greater performance and efficiency
  - Software makes it possible to provide sophisticated control
- Lower costs
  - Less expensive components can be used
  - Manufacturing costs reduced
  - Operating costs reduced
  - Maintenance costs reduced
- More features
  - Many not possible or practical with other approaches
- Better dependability
  - Adaptive system which can compensate for failures
  - Better diagnostics to improve repair time

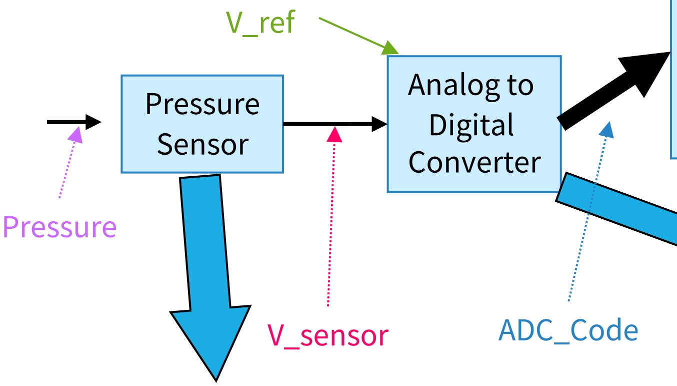
Embedded System Functions

- Closed-loop control system 不需要人的干预
  - Monitor a process, adjust an output to maintain desired set point (temperature, speed, direction, etc.)
- Sequencing
  - Step through different stages based on environment and system
- Signal processing
  - Remove noise, select desired signal features
- Communications and networking
  - Exchange information reliably and quickly

Attributes of Embedded Systems

- Interfacing with larger system and environment
  - Analog signals for reading sensors
    - Typically use a voltage to represent a physical value
  - Power electronics for driving motors, solenoids
  - Digital interfaces for communicating with other digital devices
    - Simple - switches 开关：0-1
    - Complex - displays

Example: Analog Sensor - Depth Gauge



V_ref

Pressure Sensor

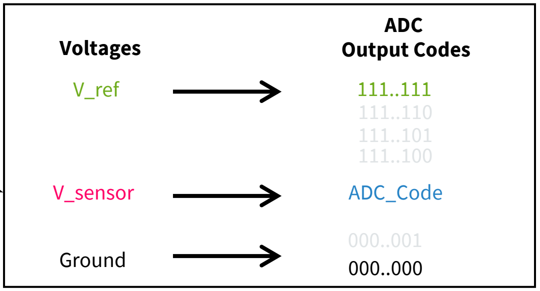
Analog to Digital Converter

Pressure

V_sensor

ADC_Code

```
// Your software
ADC_Code = adc_read();
V_sensor = ADC_code * V_ref / ADC_MASK;
Pressure_kPa = 250 * (V_sensor / V_supply + 0.04);
Depth_ft = 33 * (Pressure_kPa – Atmos_Press_kPa) / 101.3;
```



Voltages

ADC Output Codes

V_ref → 111..111

111..110

111..101

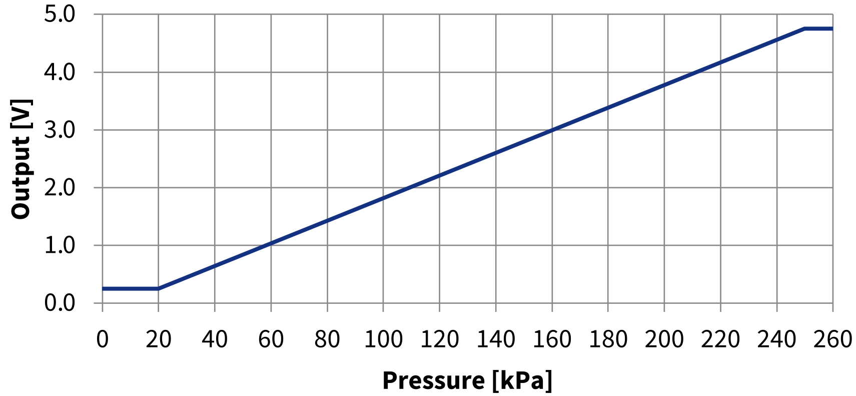
111..100

V_sensor → ADC_Code

000..001

Ground → 000..000

Typical Absolute Pressure vs. Ouput



- Sensor detects pressure and generates a proportional output voltage V_sensor **analog**
- ADC generates a proportional digital integer (code) based on V_sensor and V_ref **digital**
- Code can convert that integer to a something more useful
  - first a float representing the voltage,
  - then another float representing pressure,
  - finally another float representing depth

Microcontroller vs. Microprocessor

- Both have a CPU core to execute instructions
- Microcontroller has peripherals for concurrent embedded interfacing and control
  - Analog
  - Non-logic level signals
  - Timing
  - Clock generators
  - Communications
  - Reliability and safety

https://zhuanlan.zhihu.com/p/106860696

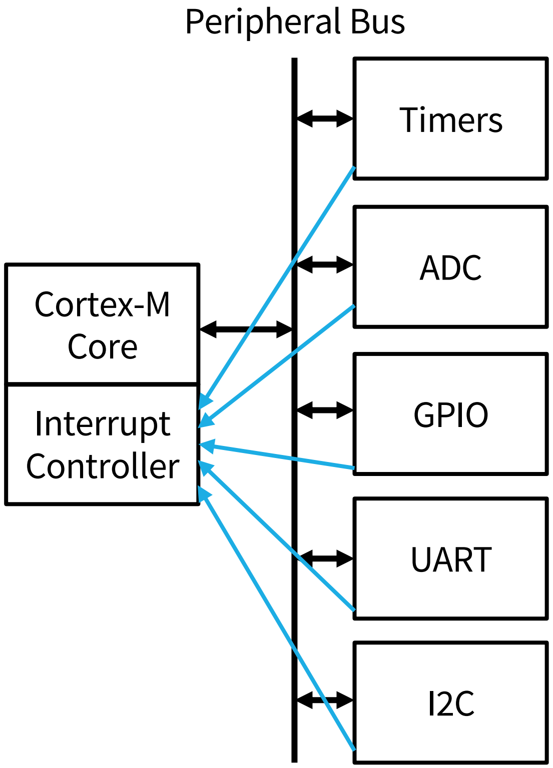
Attributes of Embedded Systems

- Concurrent, reactive behaviors
  - Continuously react to inputs from the environment by generating corresponding outputs
  - Must respond to sequences and combinations of events
  - Real-time systems have deadlines on responses
  - Typically must perform multiple separate activities concurrently
  - Typical applications include vehicle control, consumer electronics, remote sensing, smart household aplliances etc.

real-time OS
RTOS

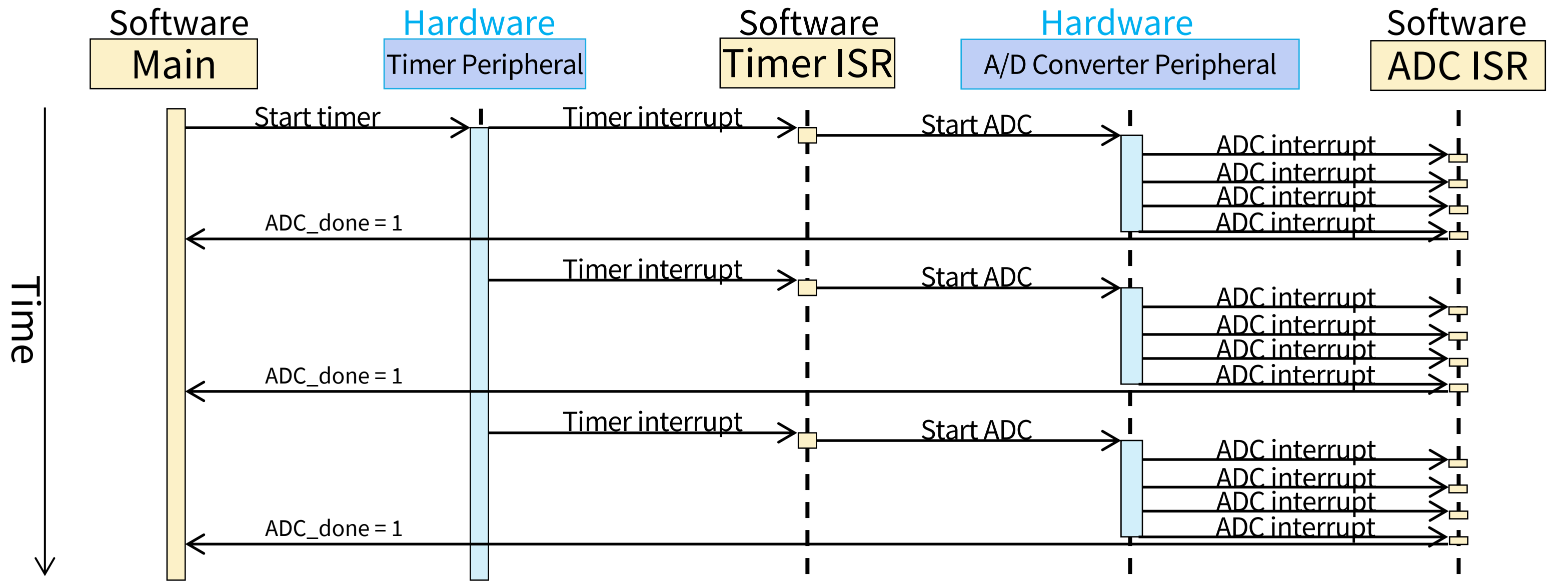
MCU Hardware & Software for Concurrency

- CPU executes instructions from one or more thread of execution
- Specialized hardware peripherals add dedicated concurrent processing
  - Watchdog timer
  - Analog interfacing
  - Timers
  - Communications with other devices
  - Detecting external signal events
  - LCD driver
- Peripherals use interrupts to notify CPU of events



Concurrent Hardware & Software Operation

ISR: interrupt service routine



- Embedded systems rely on both MCU hardware peripherals and software to get everything done on time

Attributes of Embedded Systems

- Fault handling
  - Many systems must operate independently for long periods of time, requiring system to handle likely faults without crashing
  - Often fault-handling code is larger and more complex than the normal-case code
- Diagnostics
  - Help service personnel determine problem quickly

Constraints

- Cost
  - Competitive markets penalize products which don’t deliver adequate value for the cost
- Size and weight limits
  - Mobile (aviation, automotive) and portable (e.g. handheld) systems
- Power and energy limits
  - Battery capacity
- Environment
  - Temperatures may range from -40°C to 125°C, or even more
  - Cooling limits

Impact of Constraints

- Microcontrollers used (rather than microprocessors)
  - Include peripherals to interface with other devices, respond efficiently
  - On-chip RAM, ROM reduce circuit board complexity and cost
- Programming language
  - Programmed in C rather than Java (smaller and faster code, so less expensive MCU)
  - Some performance-critical code may be in assembly language
- Operating system
  - Typically no OS, but instead simple scheduler (or even just interrupts + main code (foreground/background system)
  - If OS is used, likely to be a lean RTOS

Overview of this module

- Introductory Course:

- Building an Embedded System with an MCU
  - Microcontroller concepts
  - Processor core architecture and interrupt system
  - C as implemented in assembly language
  - Peripherals and interfacing

Why Are We…?

- Using C instead of Java (or Python, or your other favourite language)?
  - C is the de facto standard for embedded systems because of:
    - Precise control over what the processor is doing.
    - Modest requirements for ROM, RAM, and MIPS, so much cheaper system
    - Predictable behavior, no OS needed

- Learning assembly language?
  - The compiler translates C into assembly language. To understand whether the compiler is doing a reasonable job, you need to understand what it has produced.
  - Sometimes we may need to improve performance by writing assembly versions of functions.