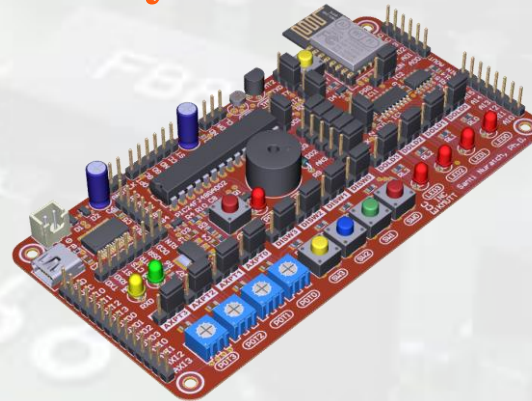
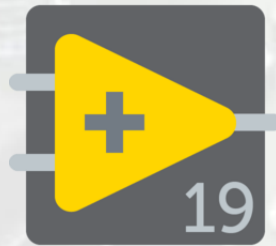
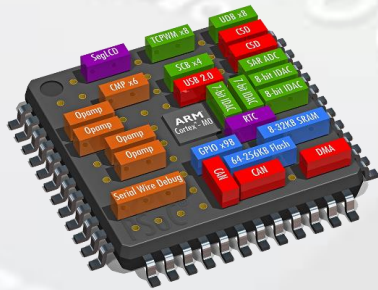


# Week #01 - Introduction

**INC 352:**

**Microprocessor and  
Embedded Systems Laboratory**



**Asst.Prof.Dr.Santi Nuratch**

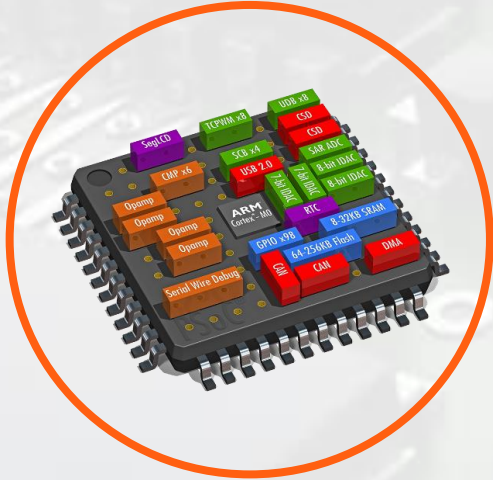
**Embedded Computing and Control Lab. @ INC-KMUTT**

santi.inc.kmutt@gmail.com

Department of Control System and Instrumentation Engineering,  
King Mongkut's University of Technology Thonburi, **KMUTT**

# There are two main modules

## Embedded Programming



Embedded C Programming Techniques

Microcontroller Peripherals

Monitoring and Control Algorithms

Protocols and Communications

## Graphical Programming



LabVIEW & Graphical Programming

Event-Driven and FSM Techniques

Monitoring and Control Algorithms

Industrial Automation Applications

Self-learning

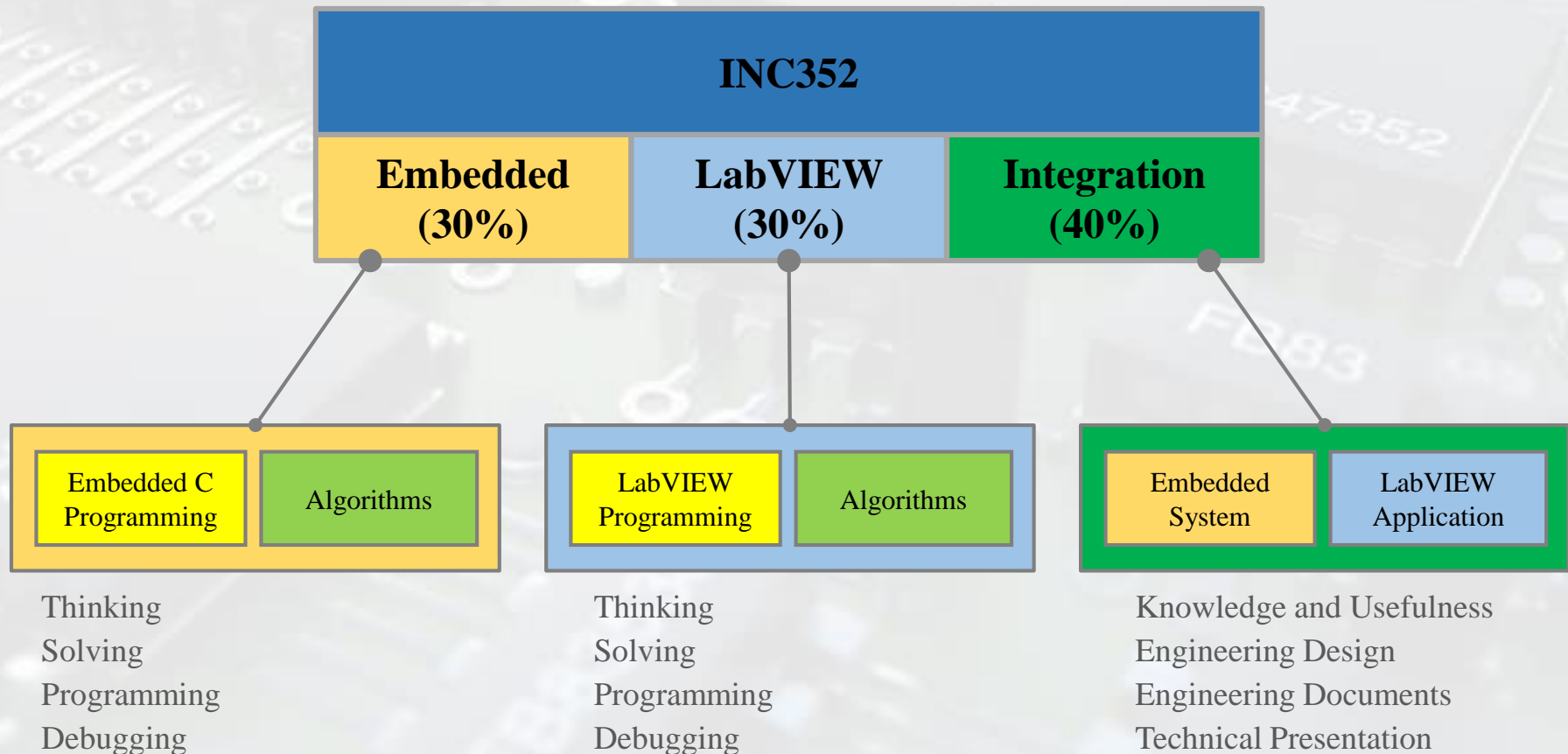
Assignments

Tests

Presentations

Teamwork

# Three main important tasks, 100%



# Evaluation (score & grade)



Grades	Scores
A	80 - 100
B+	75 - 79
B	70 - 74
C+	65 - 69
C	60 - 64
D+	55 - 59
D	50 - 54
F	0 - 49



## Weekly Quiz (60%)

- Embedded 30%
- LabVIEW 30%



## System Integration (40%)

- Document 25%
- Presentation 15%

Once a Week Is Not Enough!!



Self-Study is better than Classroom Learning

# Required Software Tools



Visual Studio Code

<https://code.visualstudio.com/>



MPLAB XC16 C Compiler

<https://www.microchip.com/mplab/compilers>



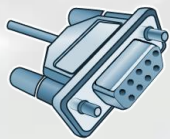
ecc-pic24-cli (Toolchain for PIC24 Development)

<https://github.com/drsanti/INC352-2020>



HTerm (Terminal for serial communication)

<http://www.der-hammer.info/pages/terminal.html>



Virtual Serial Port Driver

<https://www.virtual-serial-port.org/>



LabVIEW 2019 (32-bit)

<https://www.ni.com/>



# What is Embedded Systems?

## Electronic Devices



Most of electronic devices that we are using today are composed of microcontroller (a programmable computer). These devices are driven by the Embedded system.

# What is Embedded Systems?

## Electronic Components



Electronic components are not Embedded systems, but the Embedded systems are built from the electronic components

# What is Embedded Systems?

## General-purpose Computers



A general-purpose computers perform most common computing tasks. Personal computers, including desktops, notebooks, smartphones and tablets, are all examples of general-purpose computers.

But there's another type of computing system, the Embedded Computing Systems

Personal computers have over 10 embedded processors. Graphics accelerator, mouse, keyboard, hard-drive, CD-ROM, bus interface, network card, etc.



# What is Embedded Systems?

## Embedded Computers



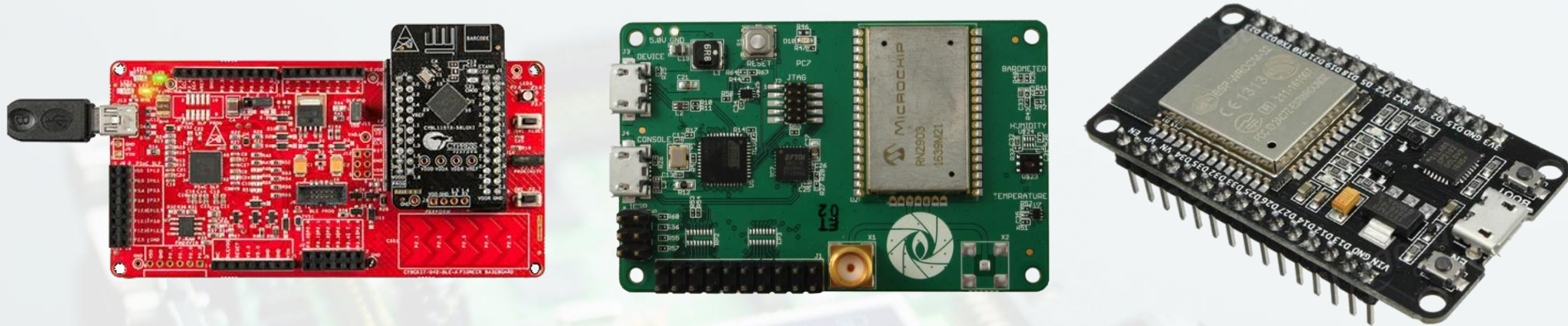
A single-board computer (SBC) is a complete computer built on a single circuit board, with microprocessor(s), memory, input/output (I/O) and other features required of a functional computer.



Industrial computers are primarily used for process control and/or data acquisition. In some cases, an industrial PC is simply used as a front-end to another control computer in a distributed processing environment.

# What is Embedded Systems?

## Single Board Microcontroller



A single-board microcontroller is a microcontroller built onto a single printed circuit board. This board provides all of the circuitry necessary for a useful control task: a microprocessor, I/O circuits, a clock generator, RAM, stored program memory and any necessary support ICs. The intention is that the board is immediately useful to an application developer, without requiring them to spend time and effort to develop controller hardware.



# What is Embedded Systems?

## A “short list” of Embedded Systems

Anti-lock brakes	Modems
Auto-focus cameras	MPEG decoders
Automatic teller machines	Network cards
Automatic toll systems	Network switches/routers
Automatic transmission	On-board navigation
Avionic systems	Pagers
Battery chargers	Photocopiers
Camcorders	Point-of-sale systems
Cell phones	Portable video games
Cell-phone base stations	Printers
Cordless phones	Satellite phones
Cruise control	Scanners
Curbside check-in systems	Smart ovens/dishwashers
Digital cameras	Speech recognizers
Disk drives	Stereo systems
Electronic card readers	Teleconferencing systems
Electronic instruments	Televisions
Electronic toys/games	Temperature controllers
Factory control	Theft tracking systems
Fax machines	TV set-top boxes
Fingerprint identifiers	VCR's, DVD players
Home security systems	Video game consoles
Life-support systems	Video phones
Medical testing systems	Washers and dryers



# Type of Embedded Systems

## Four General Embedded System Types

### General Computing

- Applications similar to desktop computing, but in an embedded package
- Video games, set-top boxes, wearable computers, automatic tellers



### Control System

- Closed-loop feedback control of real-time system
- Vehicle engine, chemical processes, nuclear power, flight control



### Signal Processing

- Computations involving large data streams
- Radar, Sonar, video compression



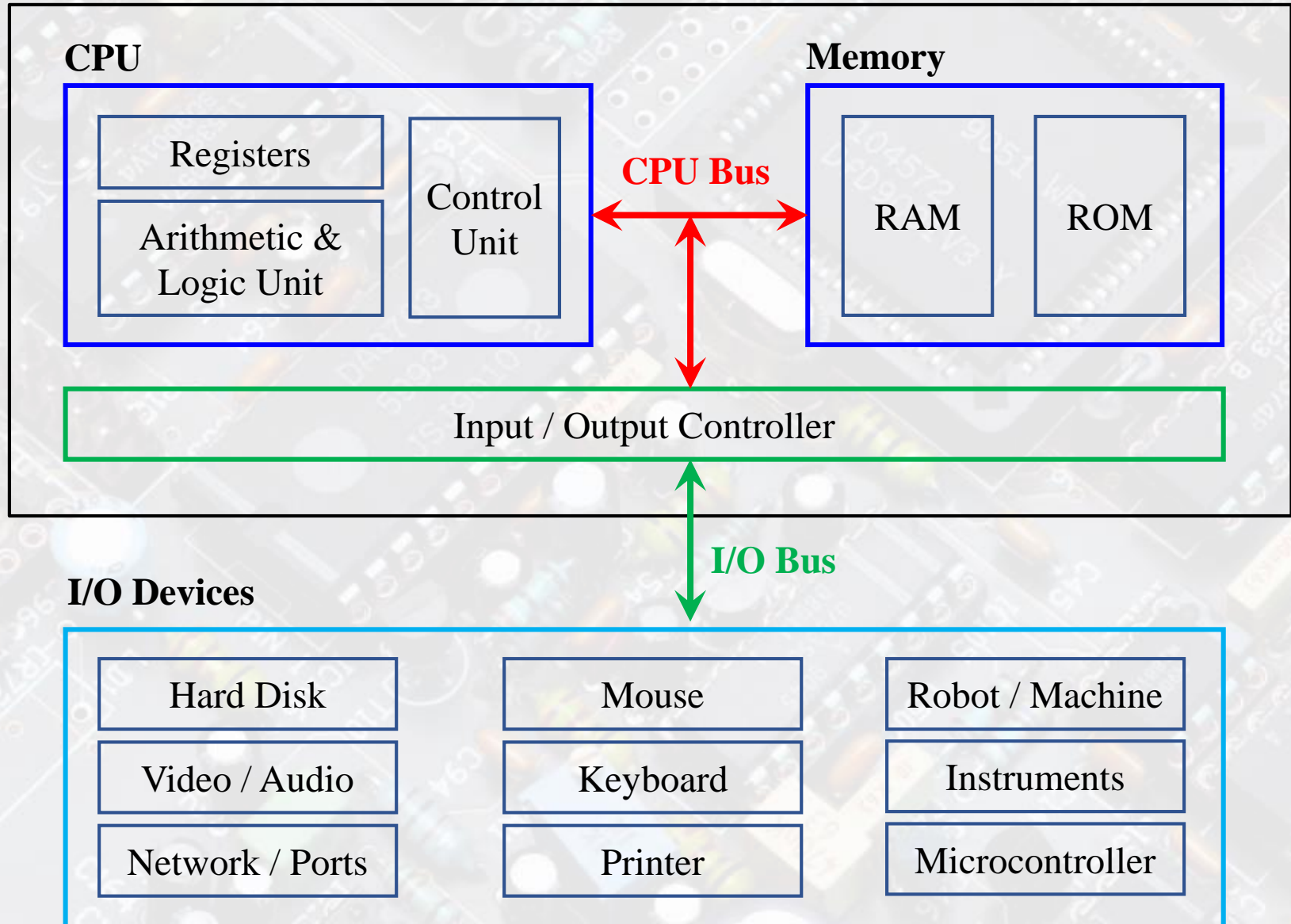
### Communication and Networking

- Switching and information transmission
- Telephone system, Internet





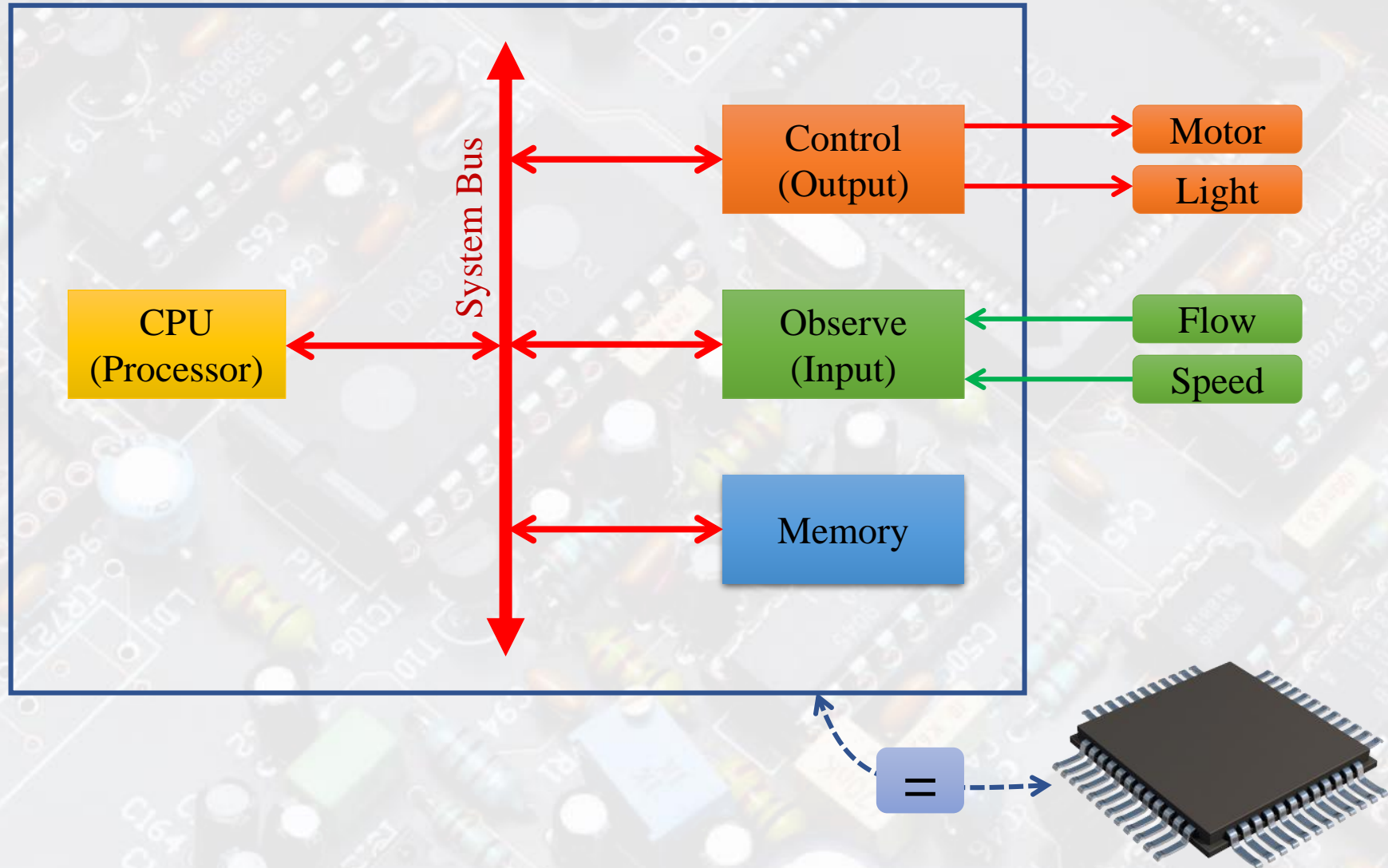
# Computer System Block Diagram



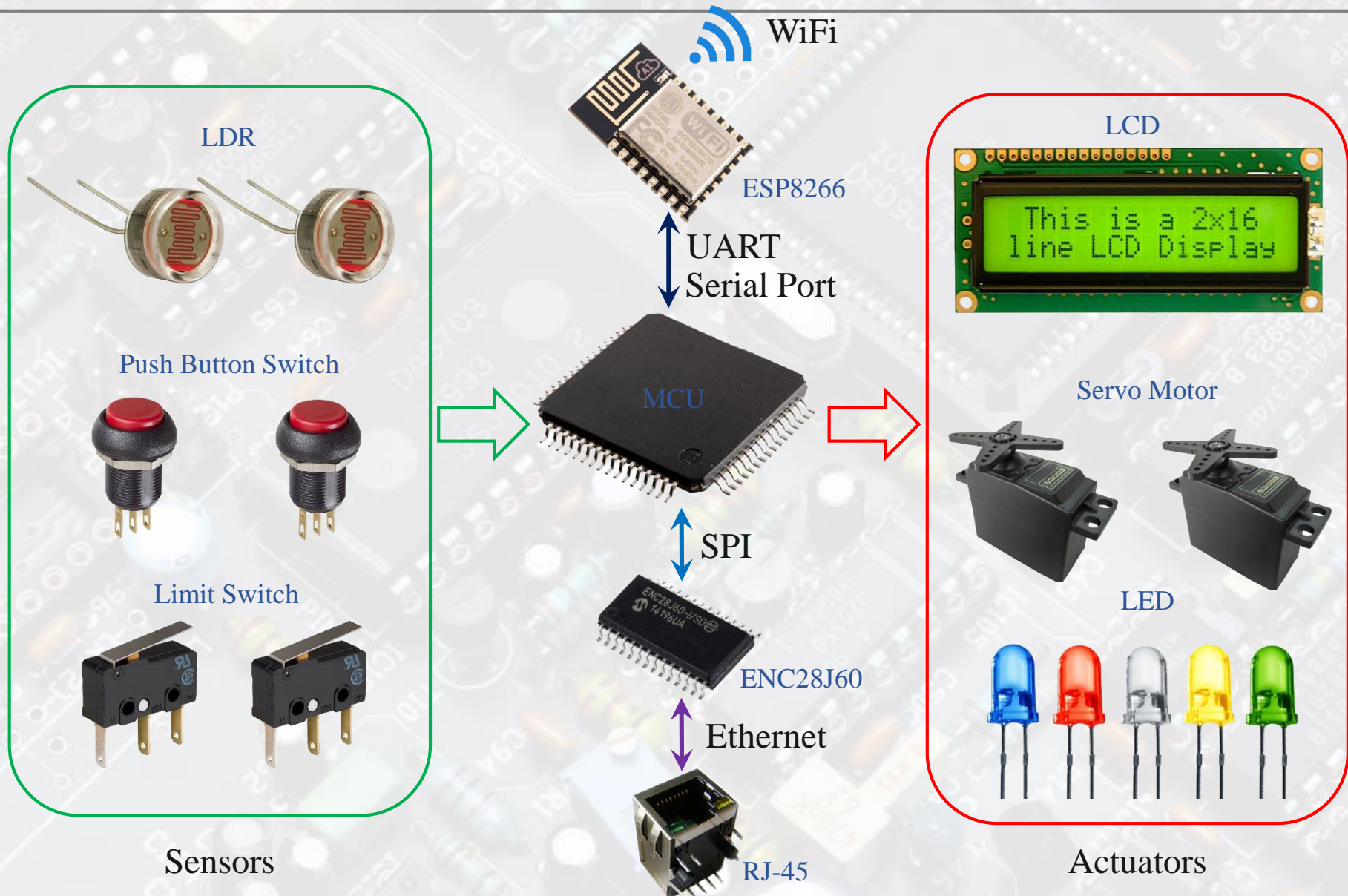


# Embedded System Block Diagram

## Simple Block Diagram of an Embedded System



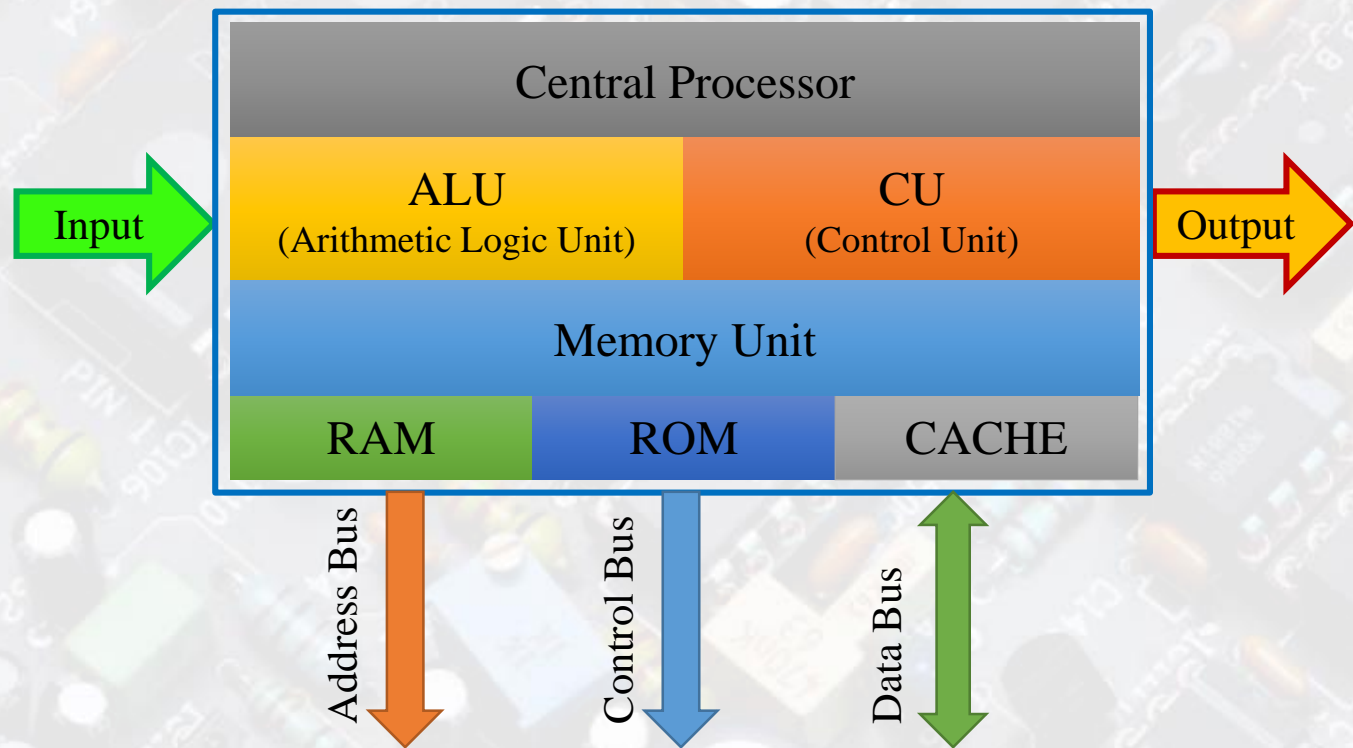
# Embedded System Block Diagram



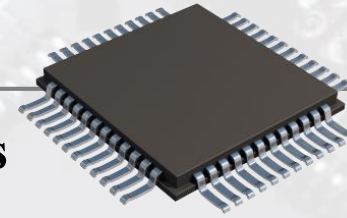


## Computer's Processors

- A processor is the logic circuitry that responds to and processes the basic instructions that drive a computer. The four primary functions of a processor are fetch, decode, execute and write/read to/from memory.

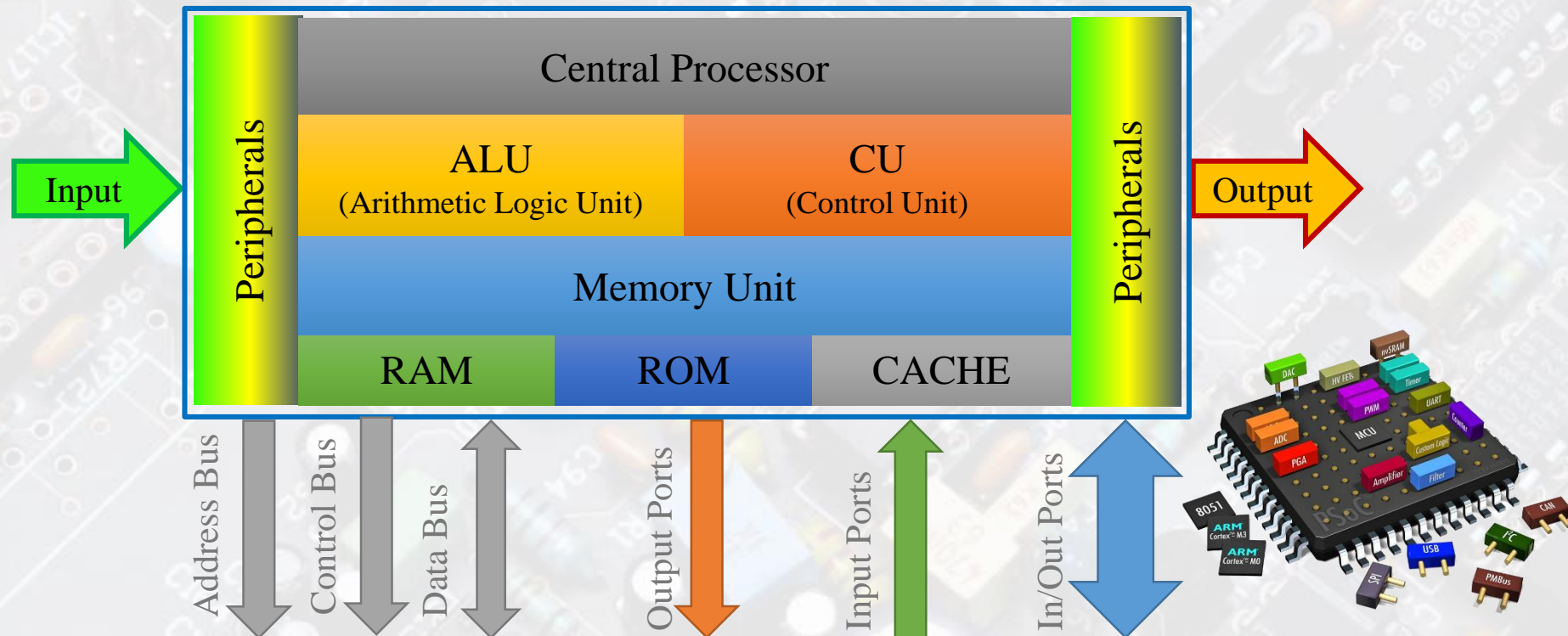






## Embedded System's Processors or Microcontrollers

- A microcontroller (MCU) is a small computer on a single integrated circuit. In modern terminology, it is similar to, but less sophisticated than, a system on a chip or SoC; an SoC may include a microcontroller as one of its components. A microcontroller contains one or more CPUs (processor cores) along with memory and programmable input/output peripherals.



## Computational micros (32- or 64-bit)

- CPU of workstations, PCs, or high-end portable devices (PDAs)
- x86, PA-RISC, PowerPC, SPARC, etc.

## Embedded general purpose micros (32-bit)

- Designed for a wide range of embedded applications
- Often scaled-down version of computational micros
- ARM, PowerPC, MIPS, x86, 68K, etc.

## Microcontrollers (4-, 8-, or 16-bit)

- Integrate processing unit, memory, I/O buses, and peripherals
- Often low-cost, high-volume devices

## Domain-specific processors (No of bits varies greatly)

- Designed for a particular application domain
- Digital signal processors, multimedia processors, graphics processors, network processors, security processors, etc.





## Small Size, Low Weight

- Hand-held electronics
- Transportation application – weight costs money



## Low Power

- Battery power for 8+ hours
- Limited cooling may limit power even if AC power available



## Harsh Environment

- Heat, Vibration, Shock
- Power fluctuations, RF interference, lightning



## Safety-Critical Operation

- **Must** function correctly
- **Must not** function *in*correctly



## Extrema cost Sensitivity

- \$.05 adds up over 1,000,000 units

## Does it really work?



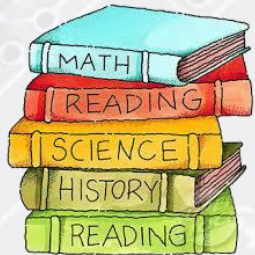
- Is the specification correct?
- Does the implementation meet the spec?
- How do we test for real-time characteristics?
- How do we test on real data?

## How do we work on the system?



- Observability, controllability?
- What is our development platform?

## Are you an Embedded Engineer?



- **Hardware** (e.g.; Digital and Electronic Circuits) is your **life**
- **Software** (e.g.; C/C++ Programming for Embedded) is your **breath**

# Embedded System Design Metrics

**Unit cost:** The monetary cost of manufacturing each copy of the system, excluding NRE cost

**NRE cost (Non-Recurring Engineering cost):** The one-time monetary cost of designing the system

**Size:** The physical space required by the system

**Performance:** The execution time or throughput of the system

**Power:** The amount of power consumed by the system

**Flexibility:** The ability to change the functionality of the system without incurring heavy NRE cost

**Time-to-prototype:** The time needed to build a working version of the system

**Time-to-market:** The time required to develop a system to the point that it can be released and sold to customers

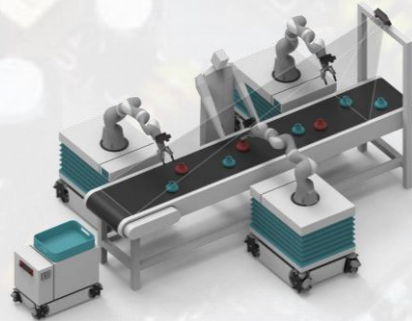
**Maintainability:** The ability to modify the system after its initial release





## What is Real-time Systems?

- Working correctly (functionally correct)
- Producing outputs **in time!**  
(i.e. **correct result(s) at the right time**)





## Hard Real-time

- System designed to meet all deadlines
- A missed deadline is a **design flaw**, For examples: ABS brake, nuclear reactor monitoring systems
- System hardware (over) designed for worst-case performance
- System software rigorously tested
- Formal proofs used to guarantee timing correctness



## Firm Real-time

- System designed to **meet all deadlines**, but occasional missed deadline is allowed ( $< 5\%$ ), For examples: multimedia systems
- System hardware designed for average case performance
- System software tested under average (ideal) conditions



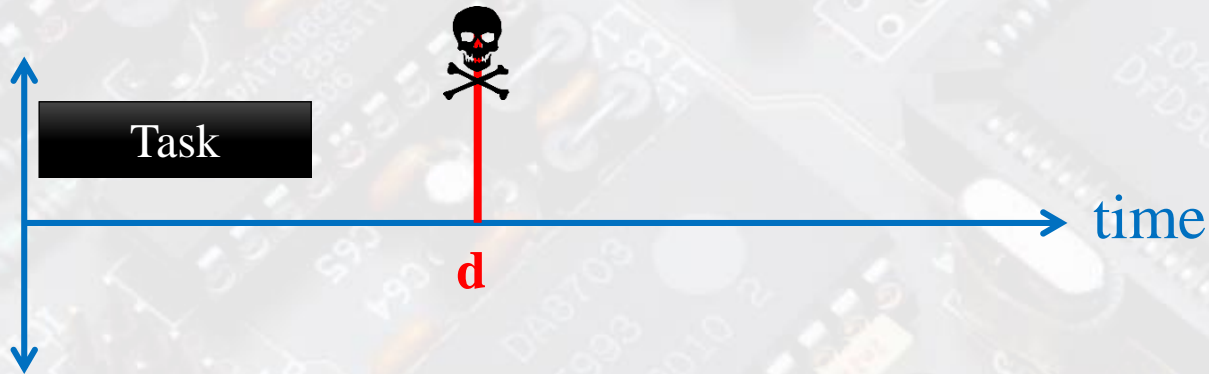
## Soft Real-time

- System designed to **meet as many deadlines** as possible, For examples: network switch or router
- System hardware designed for average case performance
- System software tested under averaged (ideal) conditions

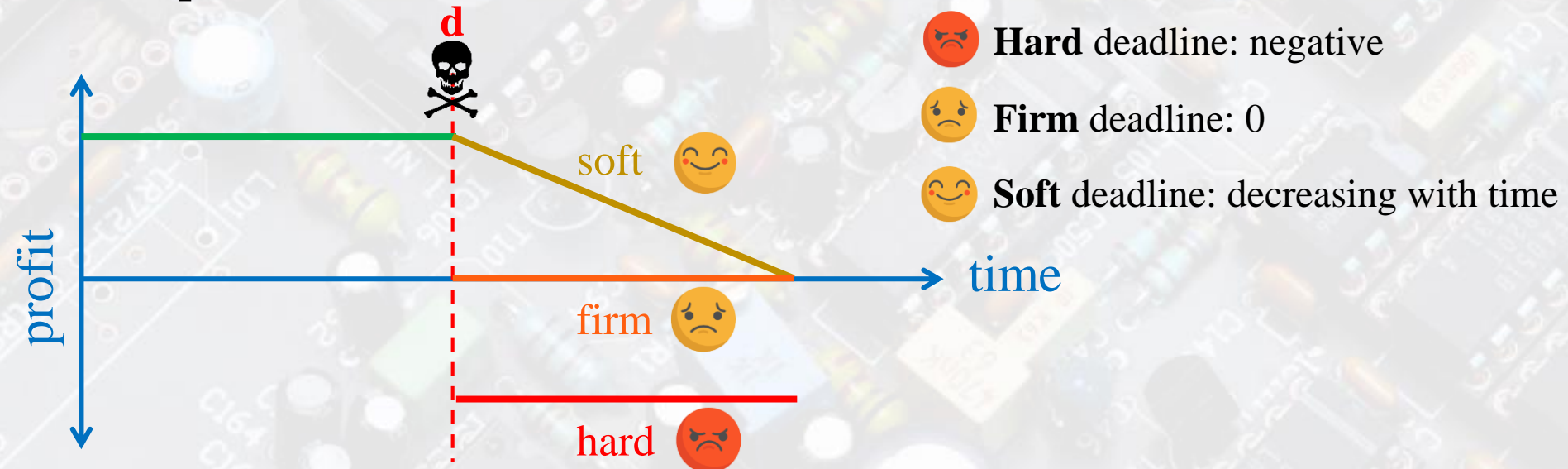


# Real-time Systems Deadlines

**Deadline:** maximum time before a task must complete

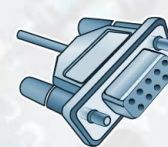


**The profit associated with execution of a task after the deadline:**

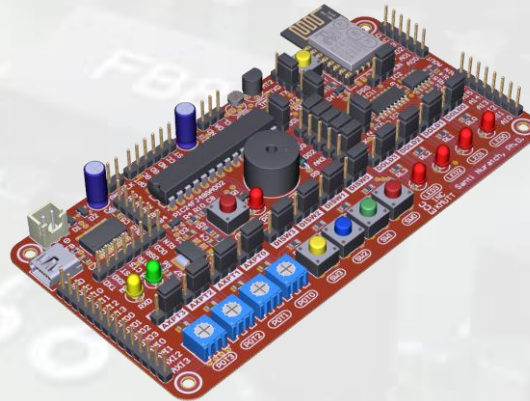
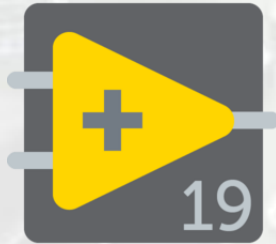
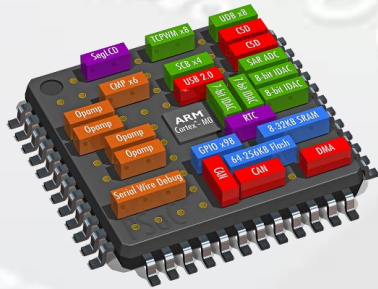


Learn by Doing  
Excel Thru Experimentation  
Lead by Example

Acquire skills and get employed  
Update skills and stay employed



# THANK YOU!



**Asst.Prof.Dr.Santi Nuratch**

**Embedded Computing and Control Lab. @ INC-KMUTT**

santi.inc.kmutt@gmail.com

Department of Control System and Instrumentation Engineering,  
King Mongkut's University of Technology Thonburi, **KMUTT**