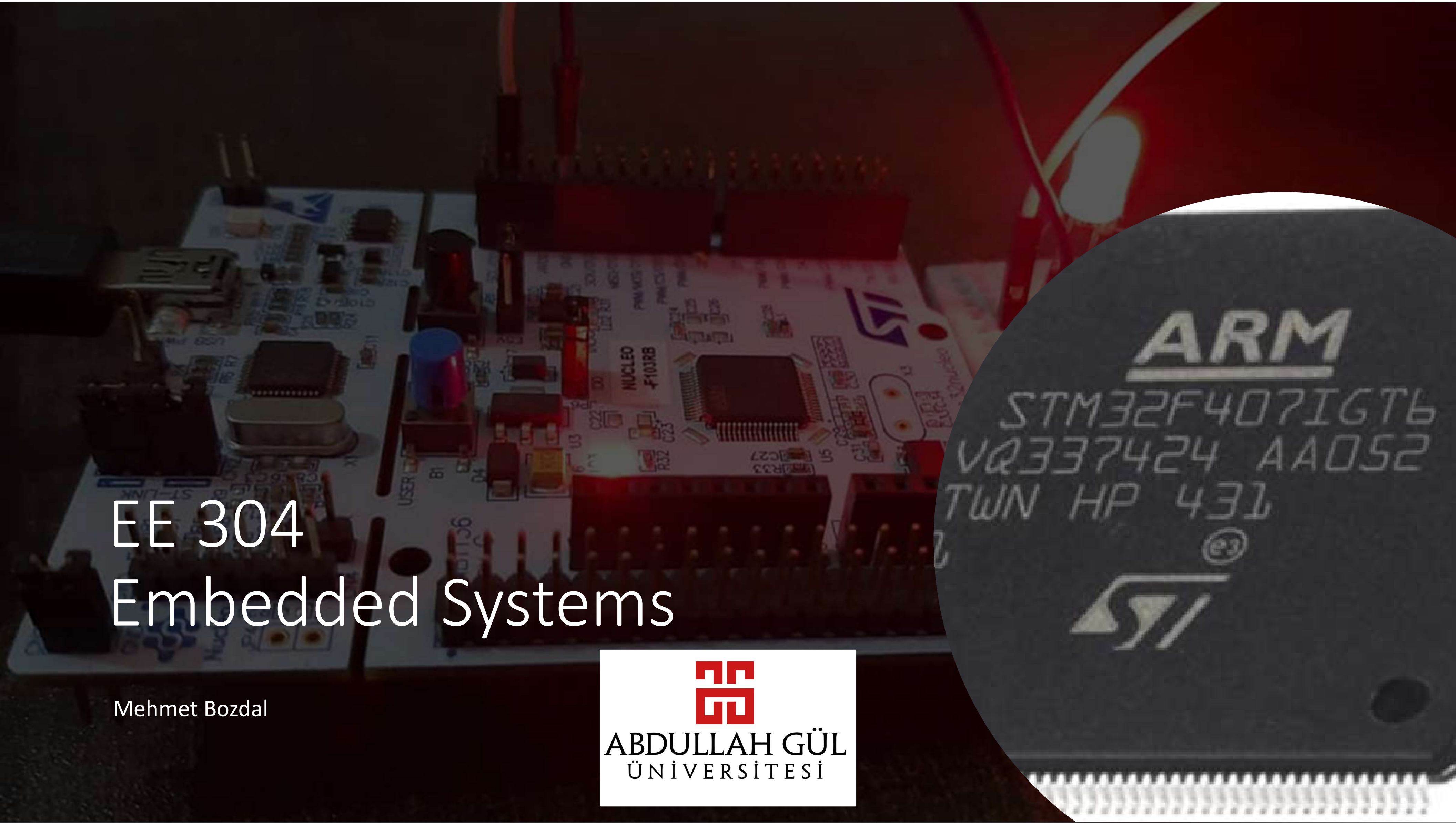


EE 304
Embedded Systems

Mehmet Bozdal

ABDULLAH GÜL
ÜNİVERSİTESİ

About me

- BSc: Electrical & Electronics Engineering @Cukurova University
- Master: Embedded Systems Engineering @University of Leeds
- PhD: Securing CAN @Cranfield University

- Email: mehmet.bozdal [at] agu . edu . tr
- Office: Factory Building F0A9
- MS Teams:

Organization of the Course

- Lecture: Friday 14:00 -16:45 @F0C14
- Laboratory: Thursday 13:00 -15:45 @EE LAB

Evaluation:

- Laboratory: %20
- Midterm Exam: %15
- Final Exam: %30
- Project: %40
  - If average of your final exam and midterm score is less than 40, your project score will be zero.
- Bonus: %5

Course Outline

- Introduction to Embedded Systems
- Microcontrollers and Microprocessors
- ARM Architecture and Assembly Language
- Timers and Interrupts
- GPIO and Digital Interfacing
- Analog Interfacing and ADC
- Communication Protocols
- Final Projects
  - Line follower or alternatives

Tools

Hardware

- EasyMx Pro Development Board
- STM3F1xx microcontroller

Software

- MDK-Arm Keil uVision (https://www.keil.com/download/product)

Resources

- https://github.com/mbozdal/embedded_systems - (Course resources)
- Embedded Systems with ARM Cortex-M Microcontrollers in Assembly Language and C - Yfieng Zhu
- Discovering the STM32 Microcontroller - Geoffrey Brown and Bryce Himebaugh
- Embedded Systems: Introduction to ARM Cortex M Microcontrollers - Jonathan Valvano

Agenda

- Embedded Systems Basics

  Definition and Significance

- Examples and Applications

  Real-world Use Cases

- Characteristics

  Key Attributes and Challenges

- Microcontrollers

  Understanding their Role

- Peripherals

  Input/Output Interfaces

- Programming Basics

  Embedded Programming Fundamentals

- STM32 Overview

  Introduction and Features

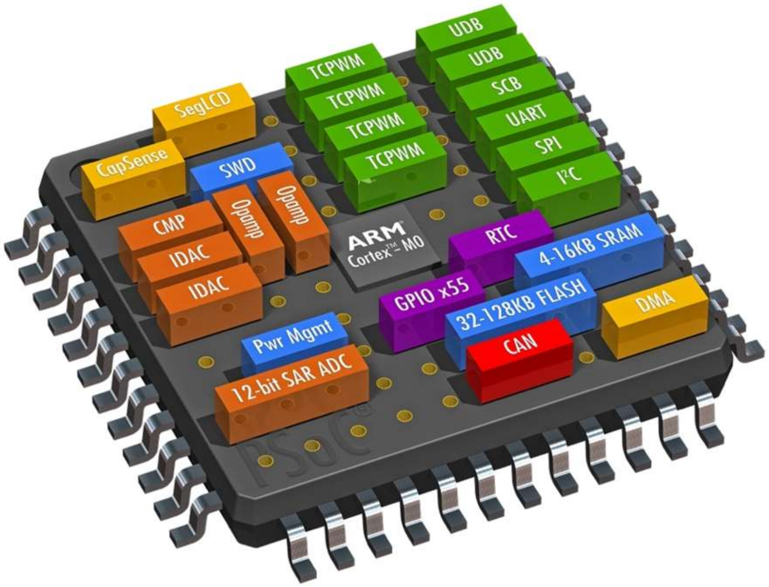
What is an embedded system?

It is a computerised system that **performs a specific task**.

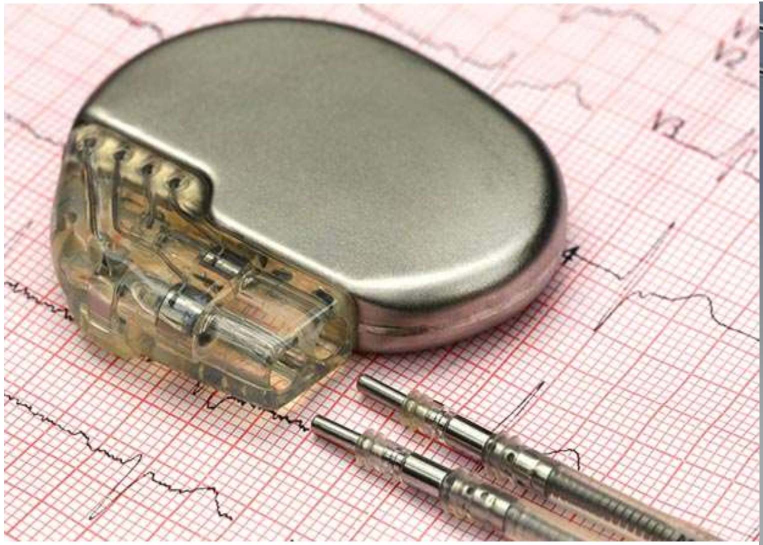
The system is consist of **hardware** and **software** components.

It is usually **embedded** or **hidden**.

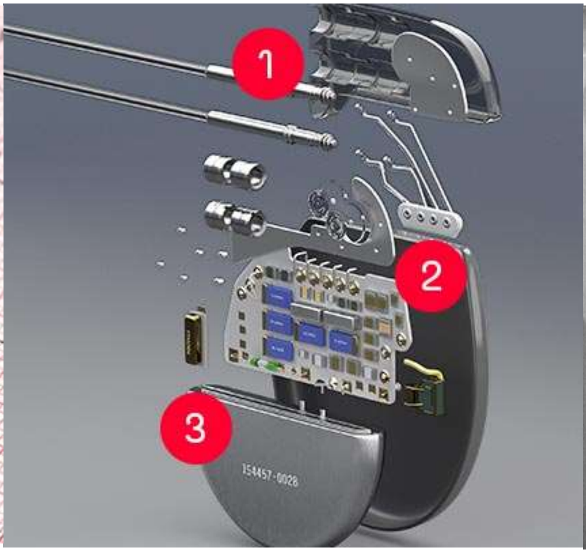
Generally, it is **small** in size.



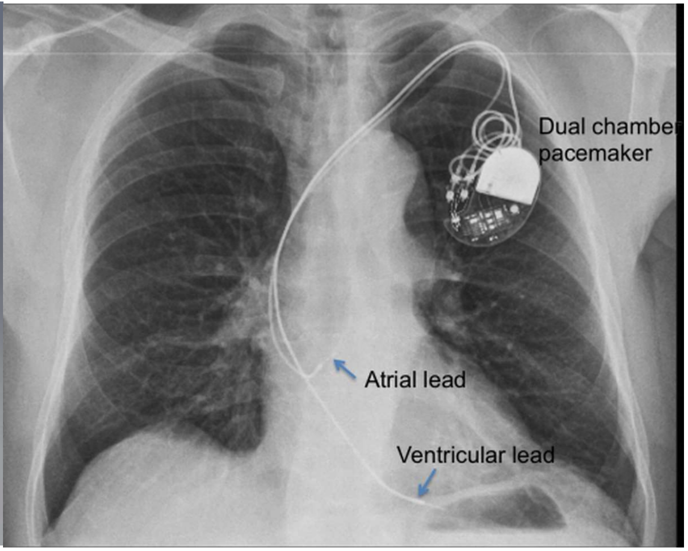
Embedded Systems Examples



a) Pacemaker [1]



b) inside of a pacemaker [1]



c) pacemaker embedded inside the body[2]

[1]https://www.bhf.org.uk/informationsupport/heart-matters-magazine/medical/how-does-a-pacemaker-work
[2] https://johnsonfrancis.org/professional/dual-chamber-pacemaker-chest-x-ray/

Embedded Systems Examples

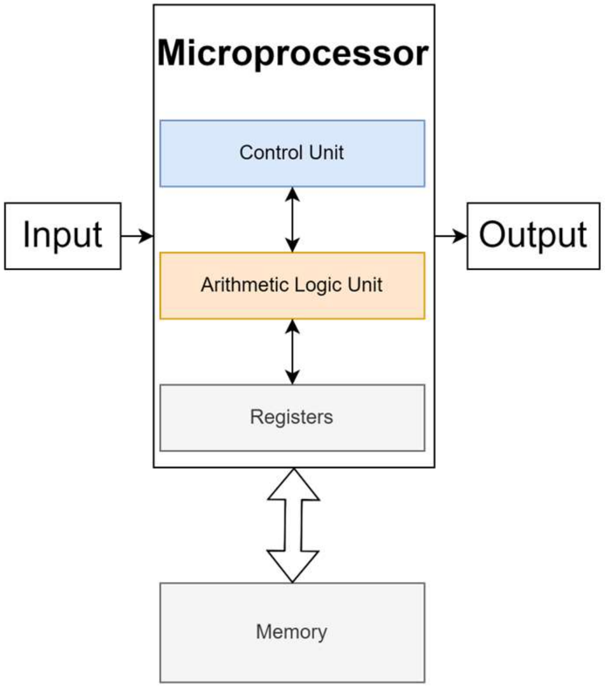
Application Areas

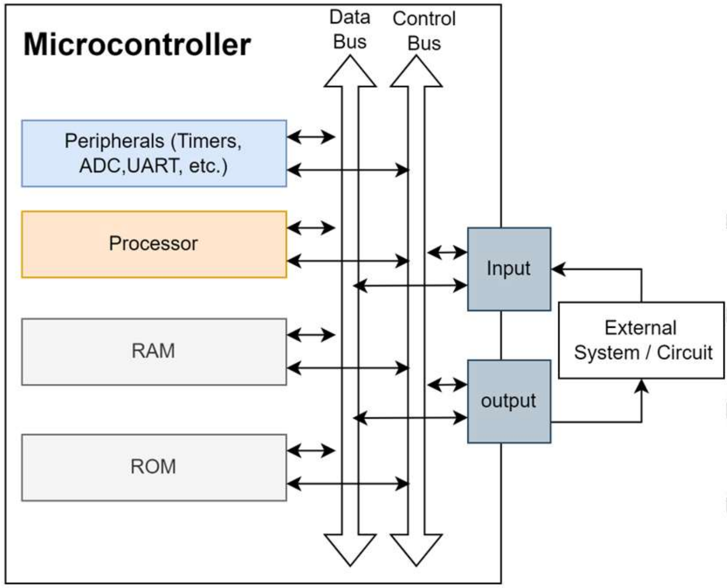
- Industry
- Military
- Consumer electronics
- Medicine
- Internet of Things

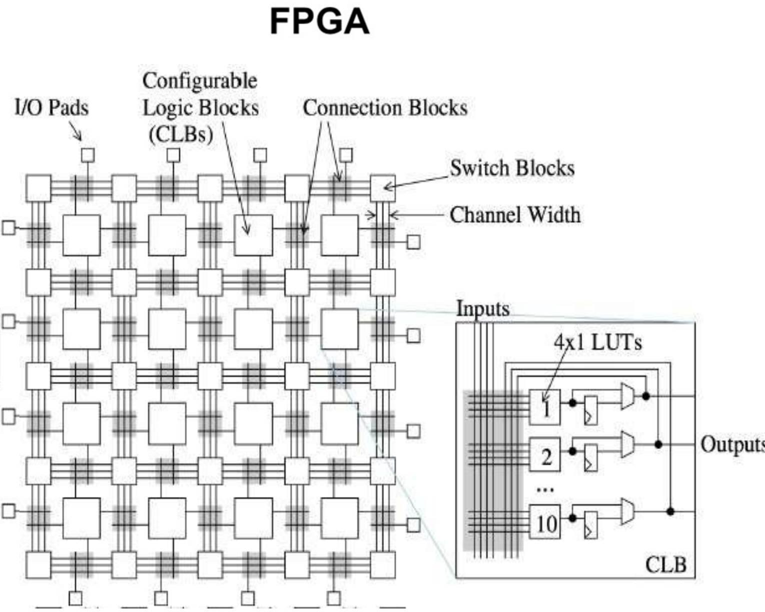
Characteristics of Embedded Systems

- They have **a dedicated task**, not a general purpose computer.
- Interacts with environment via input/output ports.
- <u>Low</u> power, <u>small</u> size and weight.
- Generally high volume, low cost.
- Time specific (i.e. runs in real time).
- Extremely reliable (some are safety critical).
- Minimal user interface.
- Not visible (can you count how many inside your car?).

Common Hardware Design Architectures

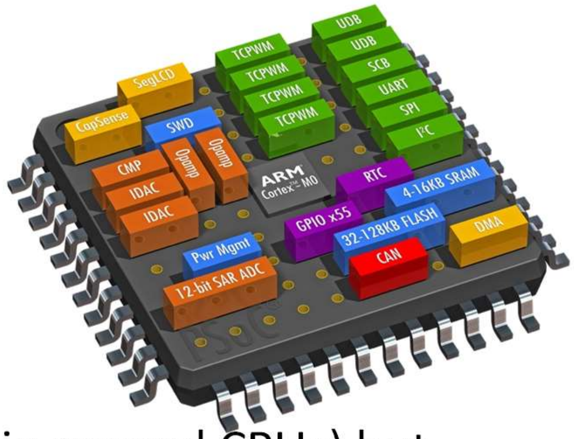



Microcontroller
Data Bus
Control Bus
Peripherals (Timers, ADC,UART, etc.)
Processor
RAM
ROM
Input
output
External System / Circuit



What is a “Microcontroller”?

- A **Microcontroller (MCU)** is an integrated circuit including all parts of complete computer.
- It includes:
  - – **CPU**
  - – Built-in **oscillator** for clock source
  - – **Flash memory** (in the order of KBytes/MBytes), to hold the program
  - – **RAM**, in the order of KBytes/MBytes
  - – Several **I/O** peripherals for both generic and specific purposes



- In its PINs, a **microcontroller** does not provides the BUS (as in normal CPUs) but the **I/O peripherals**.

What are the typical peripherals?

- Digital I/O
- Analog lines
  - Analog-to-Digital (ADC)
  - Digital-to-Analog (DAC)
- Timers
- Special digital lines (Pulse-Width-Modulation)
- Communication interfaces for other devices:
  - Universal Serial Bus
  - Universal Asynchronous Receiver-Transmitter
  - Serial Peripheral Interface
  - Controller Area Network
  - Ethernet
  - Wi-Fi, Bluetooth, etc.

Where are microcontrollers employed?

Special-purpose applications/equipment, such as:

- Measurement equipment;
- Automotive industry
- Household Appliances (TV sets, set-top-boxes, DVD, washing machines, microwave ovens, etc.);
- Previous-generation cell phones (smartphones!!)
- Industrial automation and robotics

How are microcontrollers programmed?

- Control it via set of instructions (**software**). A specific term exists for MCU software: **firmware**
- Generally, they run the software in **bare metal**, i.e. without an operating system
- In some cases, they host a very small **operating system** (e.g. FreeRTOS)
- When the system is programmed in bare metal, the developer has to take care also of programming I/O peripherals
- Usually they are programmed in **C/++** or **assembly** through a development tool running in a host computer which includes:
  - A compiler
  - A hardware tool to transfer the code into the flash memory(programmer)
  - An in-circuit debugger (optional)
- When the firmware is written in C, the MCU, at power-up, runs the program directly from the **main()** function.

Microcontrollers: manufacturers and families

There are many manufacturers of microcontrollers:

- Microchip
- Atmel
- Freescale
- STMicroelectronics
- Intel

A specific microcontroller (the specific chip) is identified by:

- The **core**, that is the **CPU**: 8-bit, 16-bit, 32-bit, 64-bit.
- The core usually denotes also the **family**
- The amount of **flash memory** and **RAM**
- The **peripherals** which are included in the chip

The MCU we will use

- STM32F1x family by STMicroelectronics.
- Several peripherals (digital I/O, ADC, timers, SPI, I2C, CAN, USB, Ethernet).
- 32-bit ARM-Cortex CPU.
- CPU clock from 72 to 240 MHz.
- Flash memory from 512K to 2M.
- RAM from 512K to 2M.

Reading List

- M68HC05 Family - Understanding Small Microcontrollers
- https://mbozdal.github.io/blog/2023/intro-embedded/
- https://mbozdal.github.io/blog/2023/microcontroller/

Q&A

Any questions?