MEC302: Embedded Computer Systems

Lecture 1 – Introduction to embedded computer systems

Dr. Timur Saifutdinov

Assistant Professor at EEE, SAT

Email: Timur.Saifutdinov@xjtlu.edu.cn

Coursework allocation

Classwork:

- 12 Lectures: W2-W14, Mondays 17:00 – 18:30 @ SIP-EE101;
- 2 Tutorials: W7, W12, Thursdays 14:00 – 15:00 @ SIP-EE101;
- 2 Lab sessions: W8, Thursday 9:00 – 11:30 @SD546, SD554, SC375;
  13:00 – 15:30 @SD546, SD554, CBG13.

Assessment:

- 1 Homework assignment: W7-W9, Thursday (15%);
- 1 Programming assignment: W8-W11, Thursday (15%);
- Final exam: during examination period, 2 hours (70%).

!!!No opportunity for resit exam – University policy!!!

Syllabus

- W2 – L1: Introduction to embedded computer systems
- W3-W4: Theme I – Modeling Dynamic Behaviors
  - L2: Continuous Dynamics
  - L3: Discrete Dynamics and Hybrid Systems
- W5-W10: Theme II – Design of Embedded Computer Systems
  - L4: Sensors and Actuators
  - L5: Embedded Processors
  - L6: Memory Architectures
  - L7: Basics of Embedded C Programming
  - L8: Input, Output and Peripheral Devices
  - L9: Multitasking and Scheduling
- W11-W13: Theme III – Analysis and Verification of Embedded Systems
  - L10: Invariants and Temporal Logic
  - L11: Model checking
- W14 – L12: Summary lecture and Q&A session

Module learning outcomes

By the end of the module, a responsible student will be able to:

A. Explain what is meant by an embedded computer system;
B. Describe different types of embedded processors and their applications;
C. Characterize how parallelism and concurrency relates to embedded systems (timing, pipelines and parallel resources);
D. Explain memory architectures and their importance in embedded system design;
E. Describe the design issues facing an embedded system designer with relation to input/output hardware and software;
F. Be able to model and analyze continuous and discrete systems;
G. Program the main functions of embedded systems (input/output, timers, interrupts).

Textbooks

- Lee, E.A. and Seshia, S.A., 2016. **Introduction to embedded systems: A cyber-physical systems approach.** Mit Press.
- Furber, S.B., 2000. **ARM system-on-chip architecture**. Pearson Education.

Introduction to embedded computer systems

网络物理系统

What is an **embedded system**?

- It is a **Cyber-Physical System** (**CPS**), which behavior is determined by both cyber (computational) and physical parts of the system.

Then, what is an **Embedded Computer System (ECS)**?

- A computerized **CPS**, where intensive computation is performed by a computer or multiple computers (usually, for efficient control of an object – physical part).

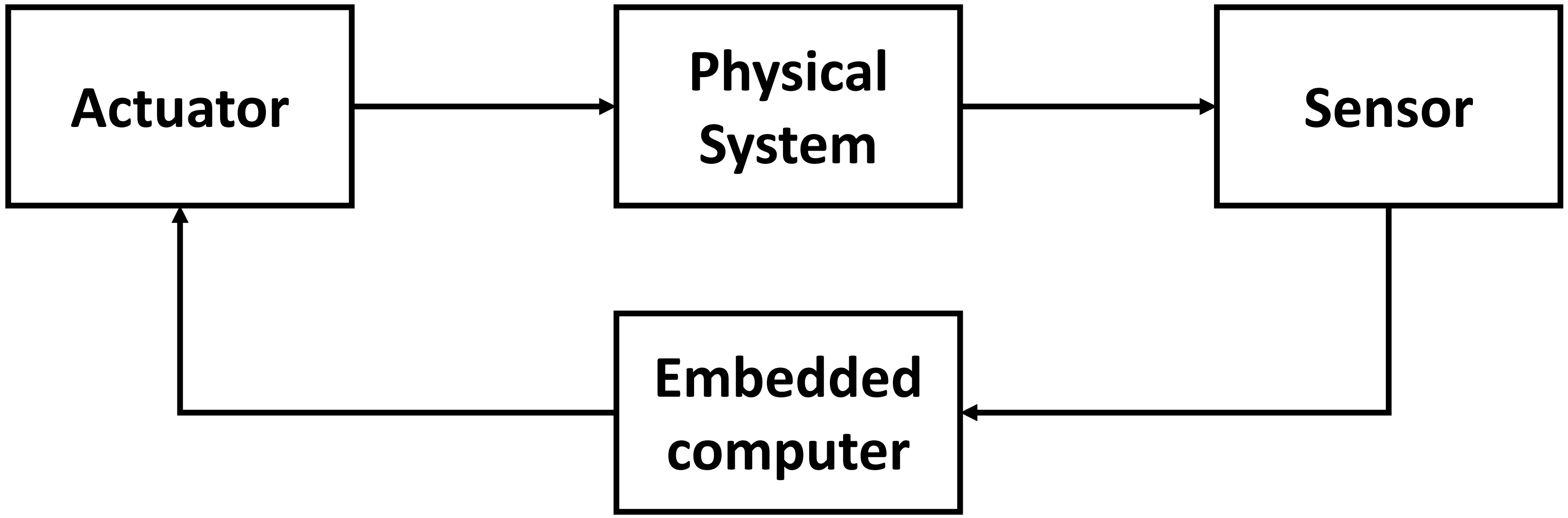
Why do you need to learn **ECS**?

- You are future **engineers**, whose job is to **design, build and maintain** machines, engines, electrical equipment, etc., using scientific principles [1];
- Most of the appliances around us are controlled by computers/controllers and it is important understand how they operate and what are the challenges they introduce.

[1] – Cambridge dictionary

Basic structure of embedded computer systems

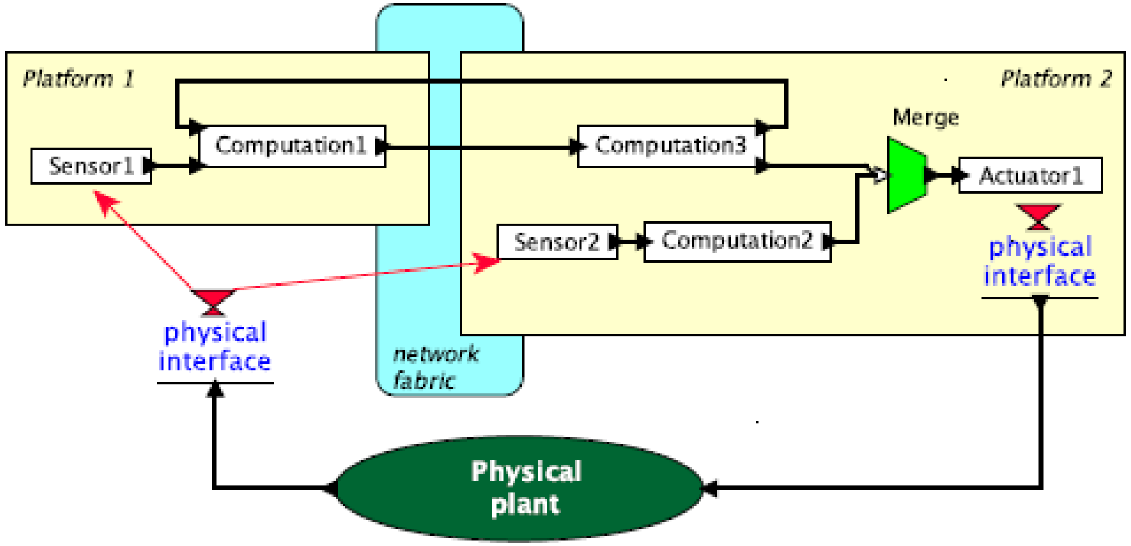
- **Physical system** – object under control;
- **Sensor** – determines physical state of an object;
- **Embedded Computer** – determines control commands;
- **Actuator** – implement control actions.



Special terms in embedded systems

Consider an abstract example of a **CPS** [2]:

- **(Physical) plant** – a physical system or an object;
- **Platform** – a computation system;
- **Network (fabric)** – communication mechanisms.



The **platform(s)** implement a **control law** by means of a **feedback control loop**.

[2] Lee, E.A. and Seshia, S.A., 2016. Introduction to embedded systems: A cyber-physical systems approach. Mit Press.

Examples of embedded computer systems

家用电器

- Domestic appliances (e.g., TV, (dish)washer, vacuum robot, etc.);
- Engine management systems (e.g., cars, ships, planes, etc.);
- City traffic system – traffic lights adjust to traffic intensity;
- Flight anti-crash system – prevent a flight to crash into obstacles (i.e. soft wall);
- Robots (e.g., arm robots, automated guided vehicles, military/disaster robots);
- Heart surgery tools – robotically controlled knife, which movement is synchronous with the heart beating.

Motivating example from the design perspectives

Quadrotor aircraft (eg, STARMAC) [3]

Main challenges:

- Control four rotors at once;
- Autonomous operation (s.t. environment):
  - Wind, obstacles, battery charge, etc.;
- Design and safety.

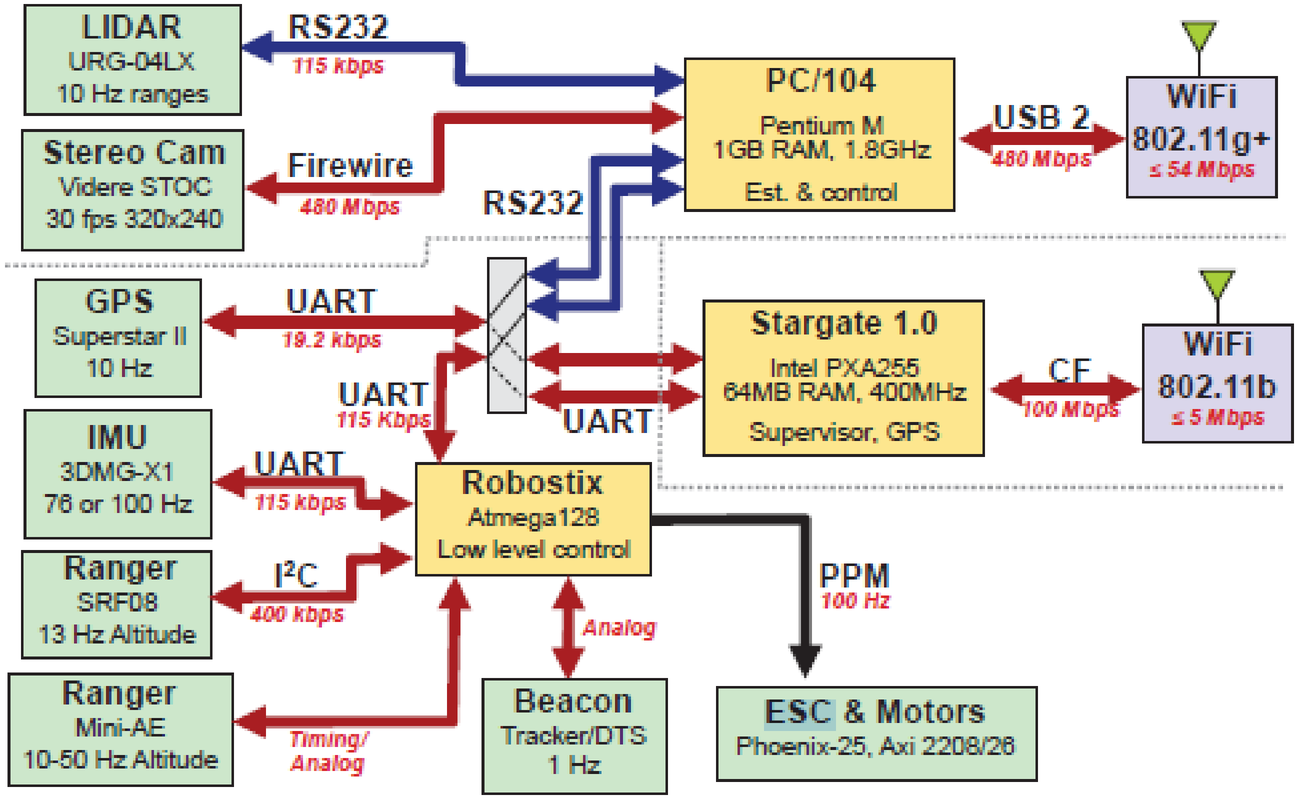
Solution:

- Detailed modeling of the dynamics of the environment, and a clear understanding of the interaction between the dynamics of the embedded system and its environment.

[3] – Hoffmann, G., D. G. Rajnarqan, S. L. Waslander, D. Dostal, J. S. Jang, and C. J. Tomlin, 2004: The Stanford testbed of autonomous rotorcraft for multi agent control (starmac). In Digital Avionics Systems Conference (DASC). doi:10.1109/DASC.2004. 1390847.

STARMAC quadrotor architecture

- 8 sensors to determine:
  - Location: GPS
  - Orientation: IMU
  - Obstacles: LIDAR and Rangers
- 3 platforms implement control:
  - Low-level control (stability, security): Robostix
    - Atmega128 – simple uController
  - Supervisory control:
    - Stargate 1.0 (Intel PXA255) – ARM comp.
  - System estimation and control
    - PC/104(Pentium M)– multi-purpose comp.
- Network fabric:
  - Numerous communication interfaces and protocols each for a purpose

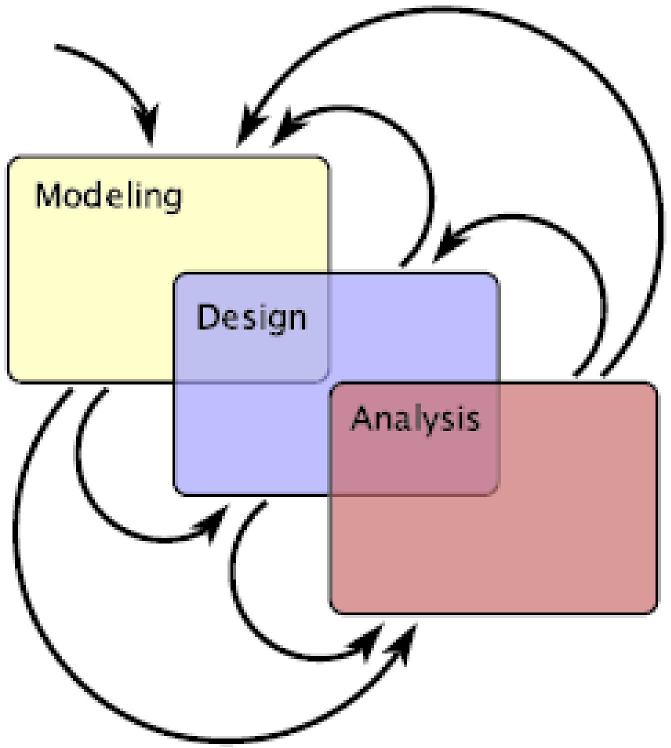


LIDAR – light detection and ranging; GPS – global positioning system; IMU – inertial measurement unit (i.e., gyroscope); Ranger – distance measurement.

Embedded system design

Creating embedded systems requires an iterative process of:

- **Modeling** – problem description and system/object representation to reflect its properties;
- **Design** – system structure and specification (ie, hardware and software);
- **Analysis** – if the system perform necessary tasks and satisfy all requirements (ie, safety).



Normally, the process begins with modeling, where the goal is to understand the problem and to develop solution strategies.

Summary

- An **Embedded Computer System (ECS)** is a computerized cyber-physical system, where (intensive) computation is performed by a computer or multiple computers;
- Engineers need to be aware of challenges and scientific principles to design, build and maintain **ECS;**
- Design of **ECS** requires understanding the joint dynamics of computers, software, networks, and physical processes – interdisciplinary module;
- Development of an efficient **ECS** requires iterative approach of modelling, design and analysis.

The end!

See you next lecture (Feb 27)