Introduction to Embedded Systems

Prof. Hugo Vieira Neto

2020/1

Embedded Computer System

- Computer System:
  - Processor + Memory + Peripherals
- Embedded:
  - Is part of another system
  - Examples: airplanes, cars, domestic appliances, agricultural equipment, medical equipment, telecommunications equipment, etc.
- Reacts to external and internal events

Embedded Computer System

- Has specific functionalities / use
  - Embedded software (firmware) updates are rare
- Complies to several restrictions:
  - Cost (acceptable price)
  - Portability (physical dimensions)
  - Robustness (environmental conditions)
  - Power consumption (battery powered apps)

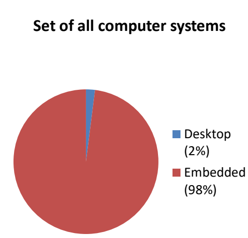
Real-Time Operation

- Correct operation does not depend solely on the results of computation, but on the time they are generated
  - Control systems (e.g.: ABS brakes)
  - Biomedical systems (e.g.: pacemakers)
  - Multimedia systems (e.g.: audio and video players, and communication devices)

Real-Time Operation

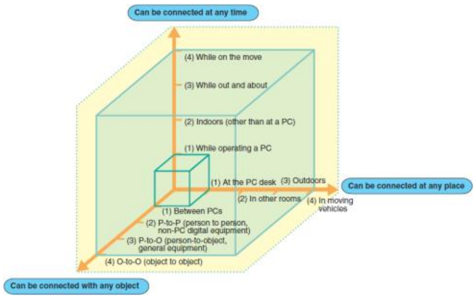
- Hard real-time
  - Failure to meet the deadlines results in drastic consequences for the system or its users
- Soft real-time
  - Failure to meet the deadlines results in system performance degradation from the viewpoint of its users

"Invisible" Computer Systems

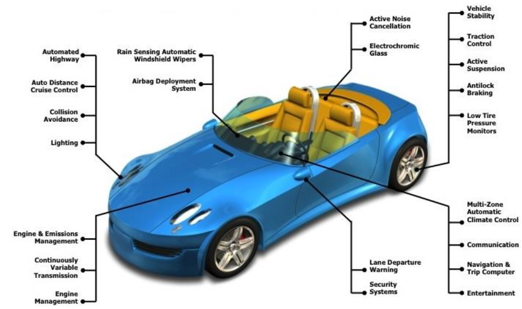


- Users normally perceive the existence of ES only when they stop working (or put their lives at risk)
- A great deal of these Embedded Systems operates in real-time

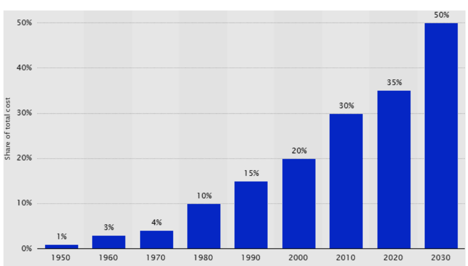
Driving Factor: Internet



Automotive Systems (Drive-by-Wire)

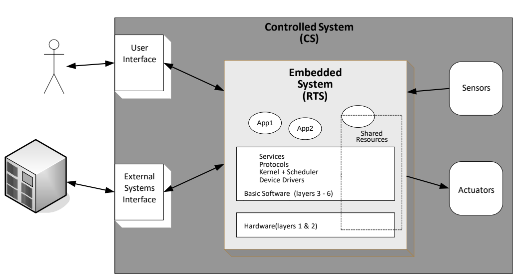


Automotive Embedded Electronics



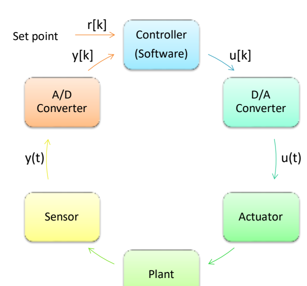
Avionic Systems (Fly-by-Wire)

Generic Example



Adapted from: Stadzisz, P. C.; Renaux, D. P. B., Software Embarcado. In: SBC – Escola Regional de Informática (ERI), 2007.

Digital Control System



PID Control Software

- Acquire sensor reading $y[k]$
- Compute $e[k] = r[k] - y[k]$, in which $r[k]$ is the desired state
- Apply control law according to known constants and dynamic parameters:
  $$u[k] = K_P.e[k] + K_I.sum + K_D.dif$$
- Update state variables (sum e dif)

Paper by Tim Wescott: *PID Without a PhD*, Wescott Design Services, 2016.

Application Videos

- Automotive – vision of the future (Texas)
- Internet of Things – vision of the future (Telit)
- Delivery (Prime Air) – present
- Domestic use (Peggy, Eight, Inirv) – present
- Illumination (Hue, Lifx, Deako) – present

Course Objectives

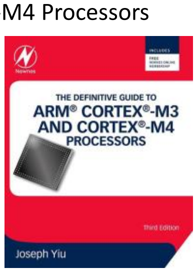
- To develop competences to specify, plan, implement, test and debug Embedded Systems, including those that operate in Real-Time.
- Platforms:
  - ARM Cortex-M3/M4 Microcontroller
  - Keil RTX 5 Real-Time Operating System (CMSIS-RTOS API v2)

Contents

- ARM Cortex-M3/M4 architecture (revision)
- ARM Cortex-M3/M4 exceptions (revision)
- Embedded Systems Modeling
- Real-Time Operating Systems
- Concurrent Programming and Scheduling
- Memory use in Embedded Systems

Main Reference

- The Definitive Guide to ARM Cortex-M3 and ARM Cortex-M4 Processors

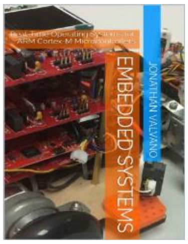


Main Reference (Portuguese)

- Sistemas Operacionais de Tempo Real e Sua Aplicação em Sistemas Embarcados

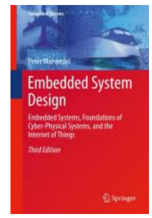
Supplementary Reference

- Embedded Systems: Real-Time Operating Systems for ARM Cortex-M Microcontrollers



Supplementary Reference

- Embedded System Design: Embedded Systems, Foundations of Cyber-Physical Systems, and the Internet of Things



Course Prerequisites

- Programming Foundations 1
- Programming Foundations 2
- Data Structures
- Digital Systems
- Computer Architecture and Organization
- Microcontrollers / Microcontroller Systems
- Operating Systems

Course Dynamics

- Heavy workload of practice (50%)
- High level of relationship between theory and practice (lab work)
  - Theory gives support to practice
  - Practice reinforces theory through experimentation and proactive attitude of students
  - Lab work themes will be subject of theoretical exams and theoretical themes will be subject of lab work demonstrations

Course Dynamics

- Weekly in class workload : 4h (2T + 2P)
- Weekly out of class workload: 4h (2T + 2P) more (at least)
- Out of class dedication is necessary **since the first week of the course**
  - Regular study of subjects of the course
  - Anticipated lab work preparation (reading manuals, planning solutions, writing code) – demands **organization**

Out of Class Workload (CCH)

- Out of class activities
  - Reading of technical documentation (manuals)
  - Specification and planning of solutions
  - Preparation of documentation and diagrams
  - Development of software and hardware
  - **Peer review** (part of lab work)

Peer Review

- Some assignments will be reviewed by three fellow students (peers)
  - Your assessment result will be validated after the delivery of your own reviews of fellow students
  - In this way everyone can learn objectively that their actions have consequences not only for themselves, but for others too
  - By reviewing work of fellow students we have the opportunity to learn with them and also teach them – all of us tend to benefit from the process and learn to "do things well"

Theory and Practice

- T & P are equally important and therefore will have the same weight in the final grade, which will be computed by geometric mean:

$$N_F = \sqrt{N_T \times N_P}$$

- In other words, to obtain the minimum passing grade, **all** theoretical and practical assessments have to be made and partial results have to be **balanced**.

Assessments

- Two theoretical assessments (written tests)

$$N_T = \sqrt{P_1 \times P_2}$$

- Three practical assessments (lab demos)

$$N_P = \sqrt[3]{L_1 \times L_2 \times L_3}$$

- Lab grades (L_1, L_2, L_3):
  - Planning and development (weight = 3)
  - Demonstration and presentation (weight = 5)
  - Individual student assessment (weight = 2)

Retakes

- Theoretical tests
  - A single written retake test with **all** the contents of the course
  - A grade below 4,0 in **any** of the written tests will make the retake test mandatory
- Practical demos
  - Presentation after the deadline is allowed, subject to a penalty of 10% in the maximum grade per week of delay

Marking Criteria

- First test: less rigorous marking
- Second test: more rigorous marking
- Retake test: absolutely rigorous marking

- Marking criteria will be presented and discussed in class

- The retake test grade will *replace* the lower grade
  - It is meant to be an exception, not the rule

Bonuses in The Final Grade

- Bonuses may be given in an individual fashion (per student, not per team), being conditioned to good grades in every course assessment
- Students are eligible to bonuses only if they have already passed without the bonuses

Attendance

- Punctuality
  - Delays in arrival will be computed, as well as early departures in every class
- Attendance will be recorded weekly in the Academic System
  - It is the student's responsibility to control his/her own attendance to class

Some Clarifications

- Passing condition
  - Final grade ≥ 6,0 **AND** attendance ≥ 75%
- Assessment criterion:
  - The student must demonstrate that he/she has acquired enough knowledge about the subject of the course in theoretical and practical assessments
- Irrelevant arguments:
  - "Professor, I am in the process of being evicted."
  - "Professor, I just need to pass this course to graduate."
  - "Professor, I work and I don't have time to study."

What is at stake?

- It is not just about your own well-being: "your grades" or "your diploma"
- It is about **our** collective well-being: **your competence** for the good working of society as a whole
- A lot of time and **public resources** (from society) are being used in **your training** – therefore, your responsibility in giving back (to society) is big

Teaching-Learning Process

- My experience of more than 20 years as a professor in higher education:
  - There is no teaching without students...
  - But there is learning without a professor.
  - Therefore, students should focus on **learning** (personal responsibility).
  - Teaching consists of helping to guide students' efforts more effectively (professor's responsibility).

Teaching-Learning Process

- There is no learning without some level of discomfort – learning means leaving the comfort zone and broadening horizons.
- But normally there is a reward (satisfaction) at the end of the process – being able to delay rewards is considered part of recipes for "success".
- Teaching: controlling the level of discomfort + maximizing the probability of reward.

Knowledge vs. Information

- Prof. Gustavo Reis' talk (TEDx):
  - https://youtube.com/watch?v=1NqMt7dU5WY

$$\text{Motivation} = \frac{1}{\text{Available Information}}$$

- The more information available (e.g. Google), the less motivation to dig into it...

The Embedded Systems Case

- Available Information→ ∞
  - Cortex-M4 Technical Reference Manual (~110 p.)
  - Cortex-M4 Devices User Guide (~280 p.)
  - Microcontroller Datasheet (~1900 p.)
  - Evaluation Kit User's Guide (~40 p.)
  - Peripheral Driver Library User's Guide (~720 p.)
  - IDE Manuals (IAR EWARM, ~1800 p.)
  - Application notes, errata, books, etc...

The Embedded Systems Case

- Therefore, motivation to dig into it → 0
  - It is desirable to have a mentor to **guide** the selection of relevant information, such that information transforms into knowledge (autonomy)
- Trust your professor to guide your efforts!

Neuroscience of Learning

1. There is no passive learning.
   - Students should participate **actively** in the classes, but being conscious that merely attending or participating does not mean to be learning.
   - Classes are teaching activities (guidance), not learning activities!

Neuroscience of Learning

2. There is no passive learning.
   - Students should do all practical activities – **if** done **individually and actively**, they will be learning activities.
   - Be careful with traps in teamwork for practical activities in which somebody takes complete leadership – if you are not the leader, you risk taking a passive attitude instead of an active one.

Neuroscience of Learning

- Take handwritten notes in paper (motor activity). Avoid computers or tablets for this.
- Prefer to read study material in paper or e-paper (Kindle, Kobo) instead of computer or tablet screens, when possible.
- Study everyday (read again lecture notes, **your own notes**, do exercises and practical activities) – preferably during the same day in which you had classes about the subject (that is, before a night of sleep).

Neuroscience of Learning

- Avoid "studying for the exams", especially if it is only on the eve of them – if you study **actively** (almost) everyday, you will learn effectively and will never need to "study for the exams".
- Prepare "cheat sheets" regularly (therefore you will be **taking handwritten notes and selecting relevant subjects** actively and objectively), but do not use them in exams.

Neuroscience of Learning

- The use of drugs (licit or illicit) impairs physical-chemical balance in the brain and therefore also impairs learning.
- Prof. Pierluigi Piazzi's talk:
  - https://youtube.com/watch?v=opMgrJyHP9k
- Prof. Pierluigi Piazzi's book: "Aprendendo Inteligência: Manual de Instruções do Cérebro para Estudantes em Geral", Aleph, 2014.

More Information

- Website: http://dainf.ct.utfpr.edu.br/~hvieir/ELx74/ELW74.html
  - Lecture notes and support material
  - Schedule for lab work demonstrations
  - References (books and articles)
  - Hardware and software documentation
  - Examples and tutorials
  - General information

Lab Work Teams

- Maximum of **two** students per team
- Reasons:
  - Greater learning efficacy
  - Limited number of benches
- Must be defined until the second week of the semester

Student Behavior

- Encouraged attitudes
  - Comparison of solutions regarding their performance, complexity, organization, elegance, etc.
  - **Suggestions** (guidance) from one lab work team to another in order to **help** solve problems (this is not about providing or sharing solutions)

Student Behavior

- Illegal attitudes:
  - Read: "Regulamento Disciplinar do Corpo Discente da UTFPR" (May / 2015)
  - Present work or parts of work done by others as your own
  - Pass on information about exams or answers at any time
  - Pass on partial or full lab work solutions
  - Copy, photograph or record exams or classes – if the professor deems it appropriate to publicize any content, he will do so through the website – ask for it!

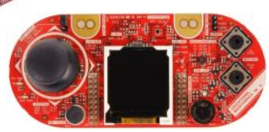
Development Kits

- There are three available models of development kit – some are newer (2018), others are in use for longer (since2012)
- Require care in handling not to be damaged
- Each team will **always use the same kit** throughout the semester
- There will be two extra kits available for use outside the class (open laboratory)

Development Kits (1)

- EK-TM4C1294XL + Educational BoosterPack MKII*

TM4C1294:
ARM Cortex-M4F



Development Environments (1)

- IAR Embedded Workbench for ARM (EWARM)
  - Has a simulator
- Keil Microcontroller Development Kit (MDK)
  - Has a simulator
- Code Composer Studio (CCS)
  - Does not have a simulator in more recent versions

- Expansions compatible with TI BoosterPacks

Development Kits (2)

- Renesas SK-S7G2

S7G2:
ARM Cortex-M4F

Development Environments (2)

- IAR Embedded Workbench for ARM (EWARM) for Renesas Synergy
  - Has a simulator
- Synergy Standalone Configurator (SSC)

- Expansions compatible with Arduino Shields and PMOD interfaces

Development Kits (3)

- LPCxpresso 1343 + LPCxpresso base board

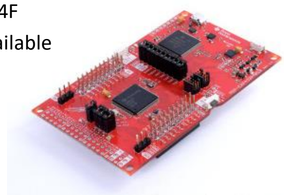
LPC1343:
ARM Cortex-M3

Development Environments (3)

- IAR Embedded Workbench for ARM (EWARM)
  - Has a simulator
  - Needs an external JTAG interface (J-LINK)
- Keil Microcontroller Development Kit (MDK)
  - Has a simulator
  - Needs an external JTAG interface (J-LINK)
- Code Red / LPCxpresso
  - Does not have a simulator
  - Uses the integrated JTAG interface

Another Option: MSP-EXP432P401R

- On sale for about U$20.00 (store.ti.com)
- Integrated JTAG interface (IAR, Keil, Code Composer)
- MSP432: ARM Cortex-M4F
- Several BoosterPacks available
- Educational BoosterPack MKII on sale for about U$30.00 (store.ti.com)
- Shipping: about U$7.00*
- Fees: about U$2.50*



Lab Work: Ideal Situation

- Each team uses its own computer and has its own kit (greater autonomy) :
  - EK-TM4C1294XL (TM4C1294) → U$20.00
  - http://www.ti.com/tool/EK-TM4C1294XL

  or
  - MSP-EXP432P401R (MSP432) → U$20.00
  - http://www.ti.com/tool/MSP-EXP432P401R

Lab Work: Ideal Situation

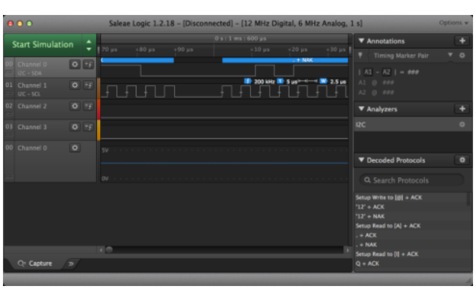
- Optional, but desirable:
  - Prof. Peron's BoosterPack

  or
  - Educational BoosterPack MKII → U$30.00
  - https://www.ti.com/tool/BOOSTXL-EDUMKII

Lab Work: Optional Tool

- 8-channel USB logic analyzer (24MHz)
- On sale for about R$40,00 (Mercado Livre)

Logic Analyzer Software



Extra-class Activity

- Read the white paper:
  - ARM Cortex-M for Beginners (Joseph Yiu)
- Get acquainted with documentation:
  - TM4C1294NCPDT microcontroller datasheet
  - EK-TM4C1294XL Manual
  - Educational BoosterPack MKII Manual
  - Prof. Peron's BoosterPack schematics
- Note: the links to the documents above are available on the website

Extra-class Activity

- Answer in writing ("cheat sheet"):
  - Which kind of information is available in each manual (microcontroller, kits)?
  - Which computational resources (memory, peripherals) are integrated to the microcontroller?
  - Which external resources are available in the development kit (not integrated to the microcontroller)?

Acquaintance with the Kit and IDE

- Objective: run the "simple_io_main_sp" project from the "EK-TM4C1294_IAR8" workspace (example-code available in the website)
  - Compiler setup
  - Linker setup
  - Debugging tools
- Explore IDE functionalities

Acquaintance with the Debugger

- Use the debugger with the simulator or kit:
  - Connection setup
  - Code upload in flash memory
  - Controlled execution (step-by-step, breakpoints)
  - Code execution in C and Disassembly
  - Memory inspection and variable inspection
  - Register inspection (CPU and peripherals)
  - Stack, I/O terminal, etc.