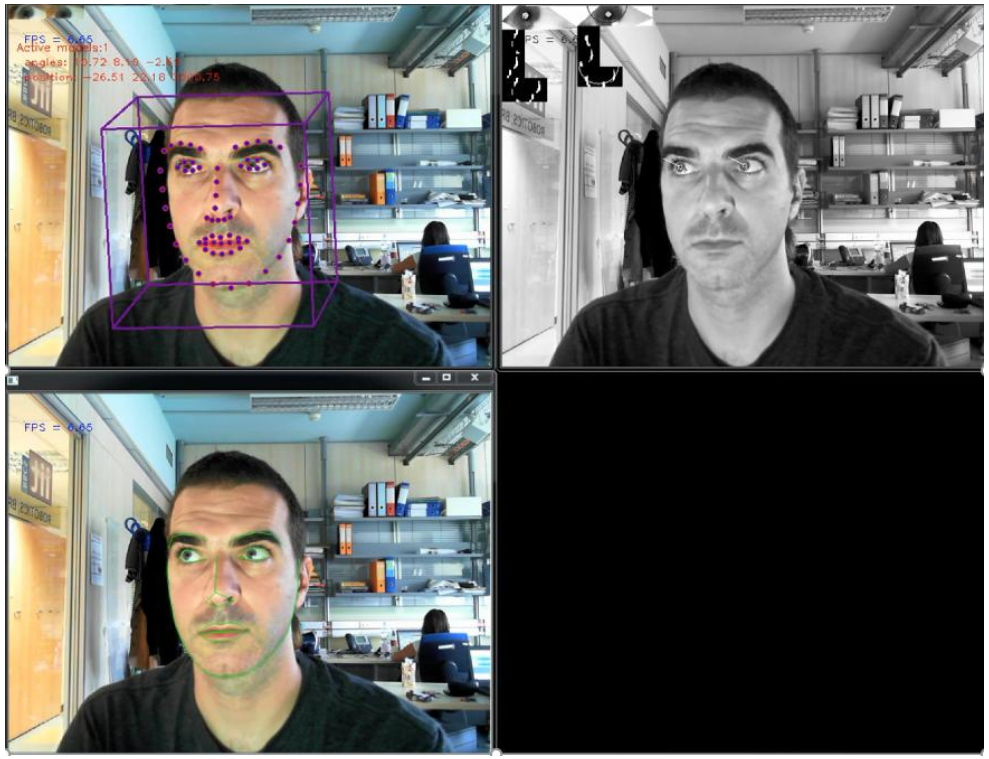
Welcome to Embedded Programming

About me



Oskar Palinko

Associate Professor at SDU Robotics, MMMI

ospa@mmmi.sdu.dk

Ø27-600a-3 in the TEK building

How to reach me:

- Email (preferred over itslearning)
- Message on itslearning (response time might be longer)
- Find me at my office (appointment encouraged, drop-ins welcome if not busy)

About Morten

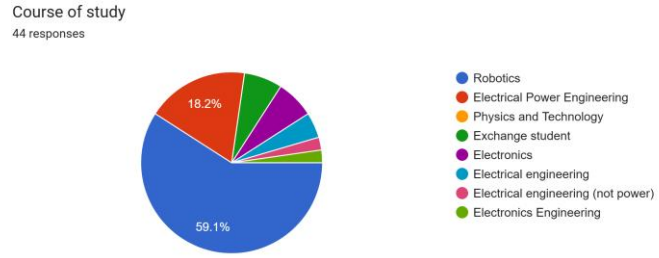
Morten Frederik Bindslev-Jensen
Teaching Assistant for Embedded Programming
MSc student in Robotek
mobin16@student.sdu.dk

In charge of labs starting today.
Contact Morten first with any technical/programming issue.

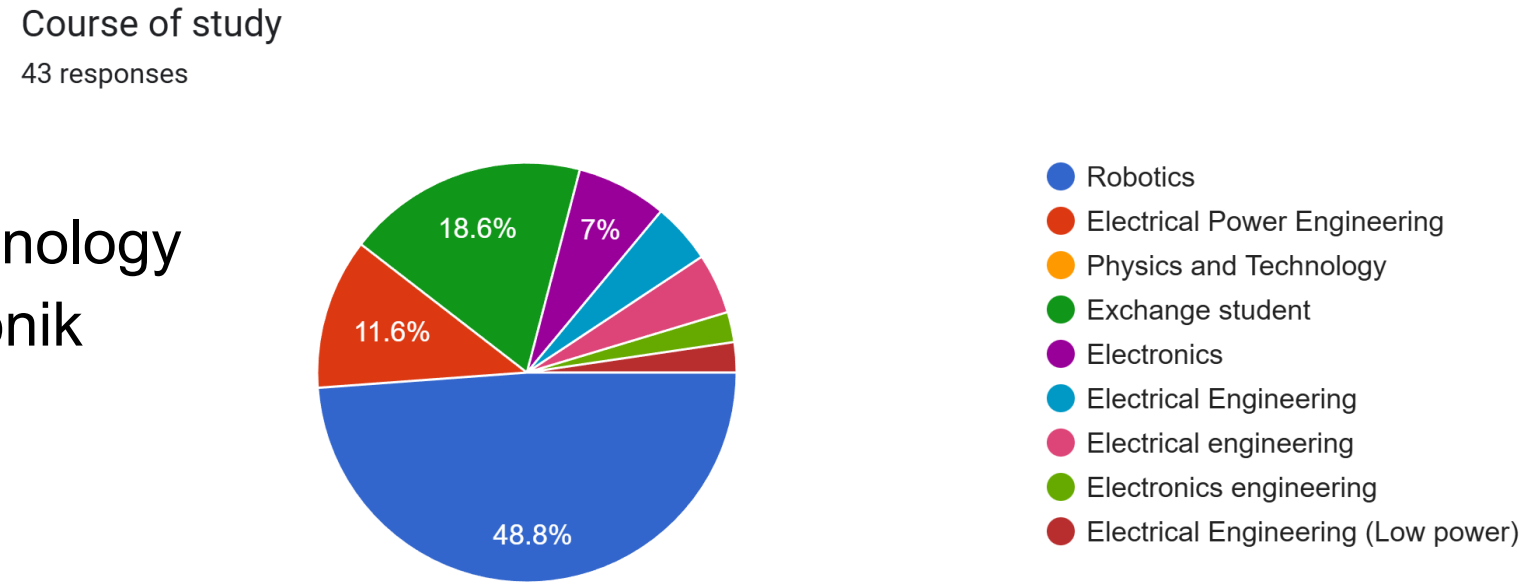
SDU

About you

Last year:



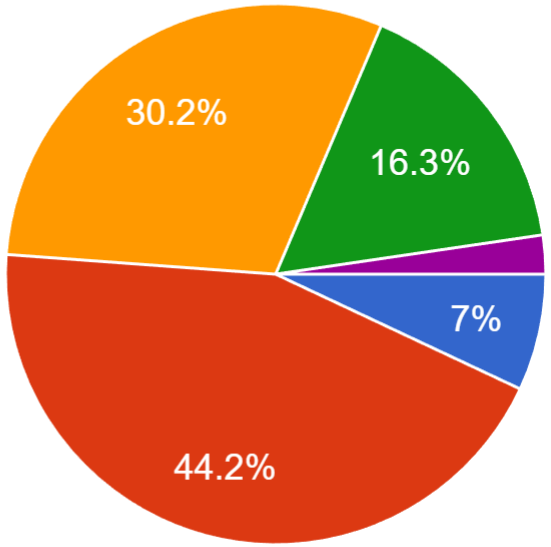
- 4th semester
- Bachelor in Robot Technology
- Diplomingeniør i elektronik
- Exchange students
- Others?



About you – responses to online form

How comfortable are you with programming/coding in C?

43 responses



- Super-comfortable
- Very
- So-so
- Less-so
- I hate programming-coding and anything related to it

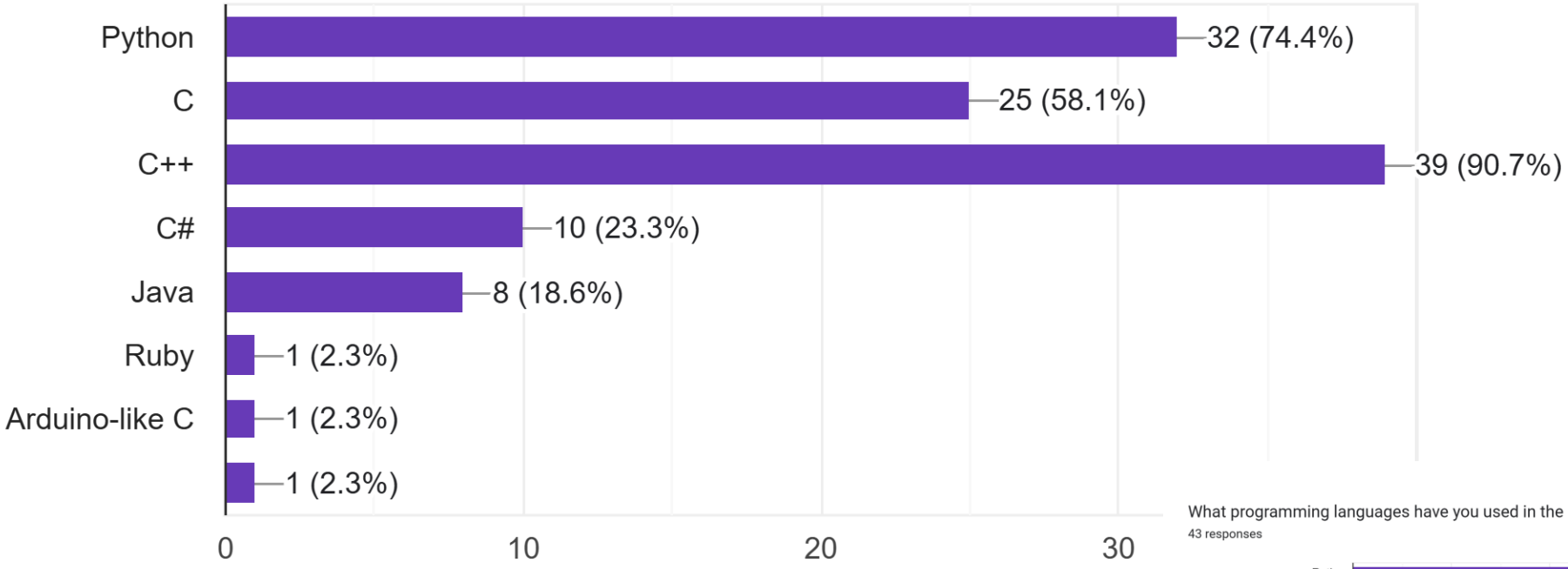
How comfortable are you with programming/coding in C?

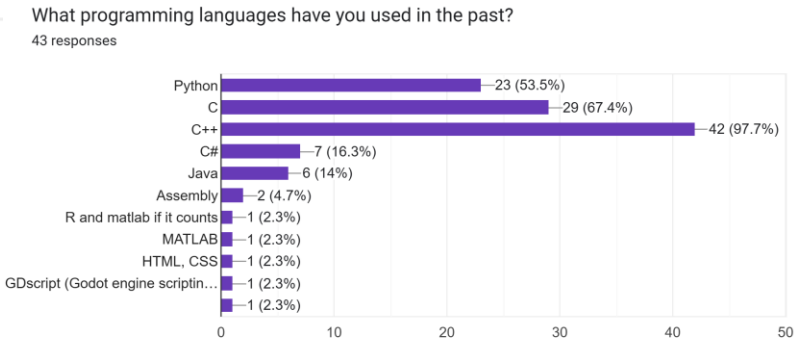
44 responses

About you – responses to online form

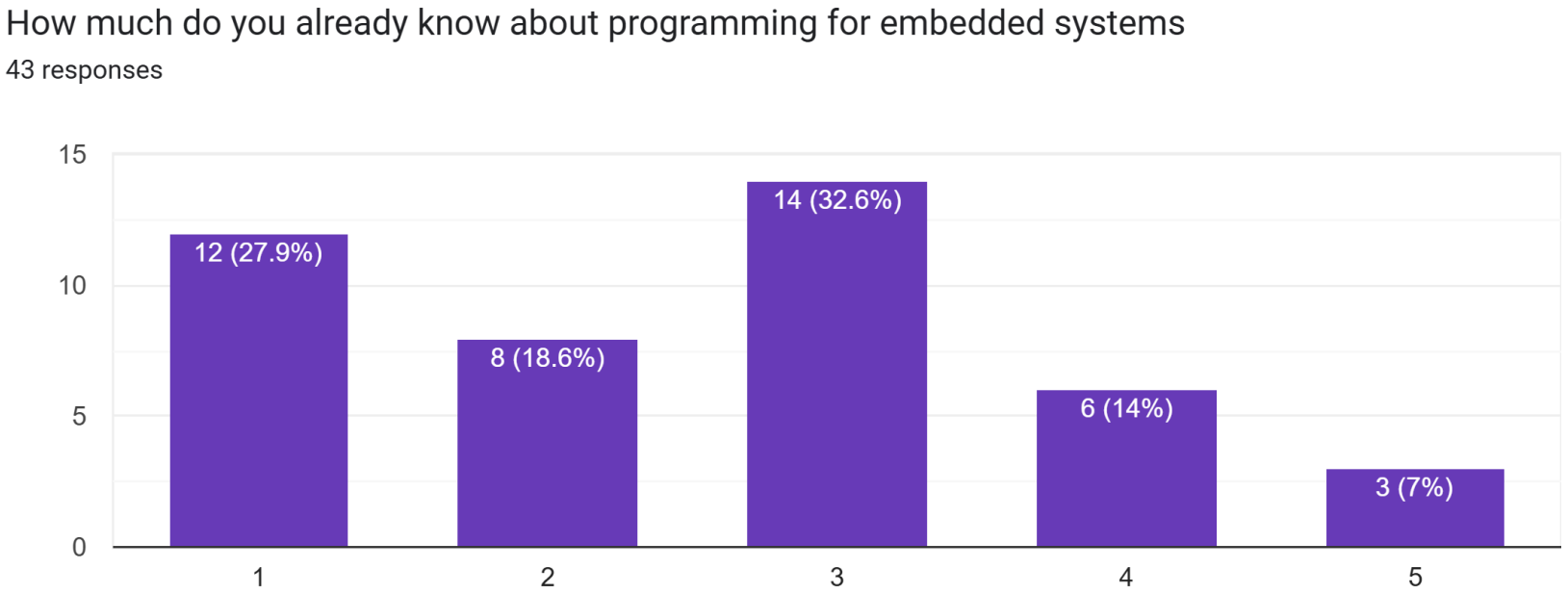
What programming languages have you used in the past?

43 responses

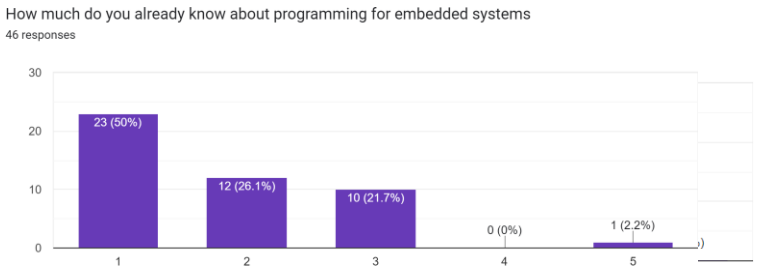




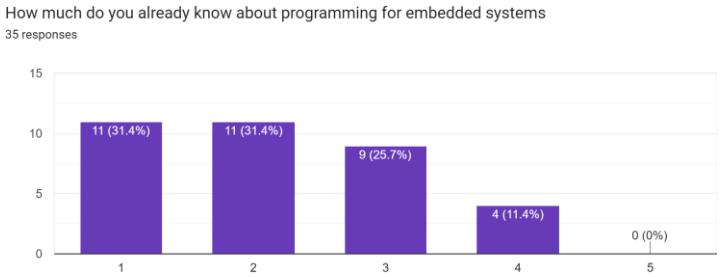
About you – responses to online form



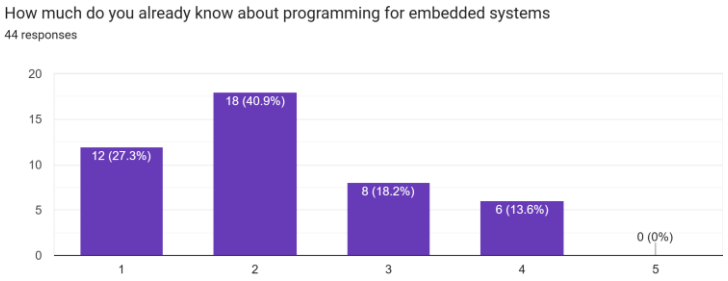
2023



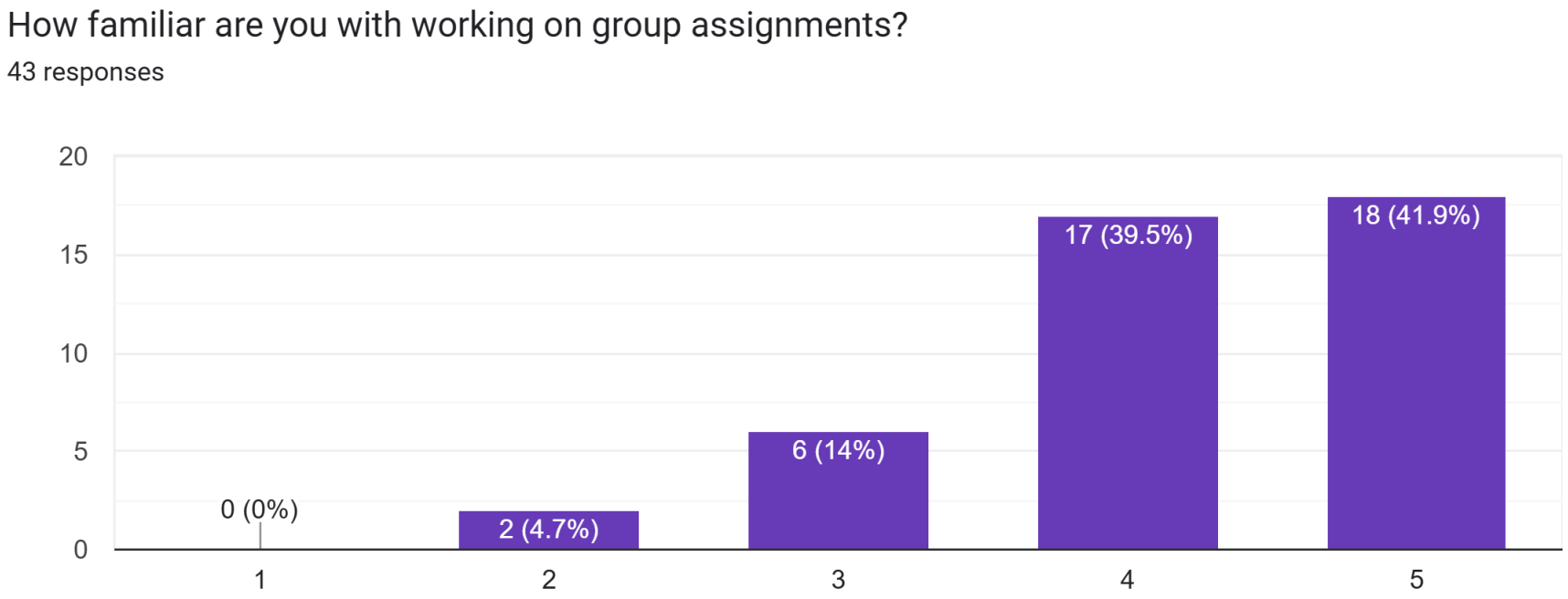
2024

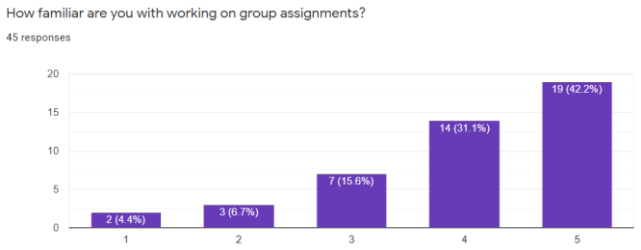


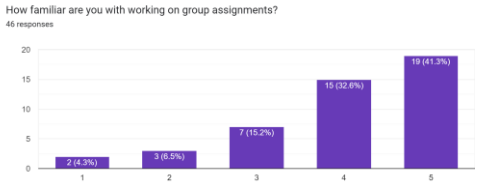
2025

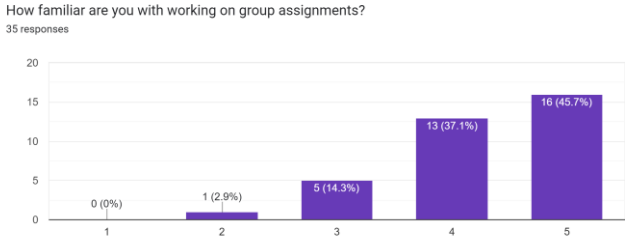


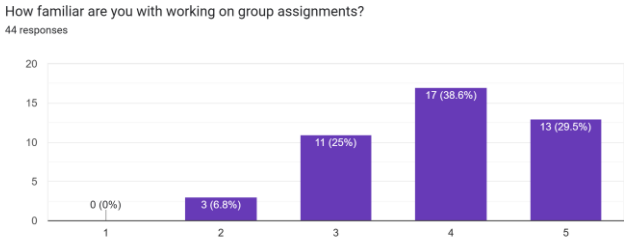
About you – responses to online form



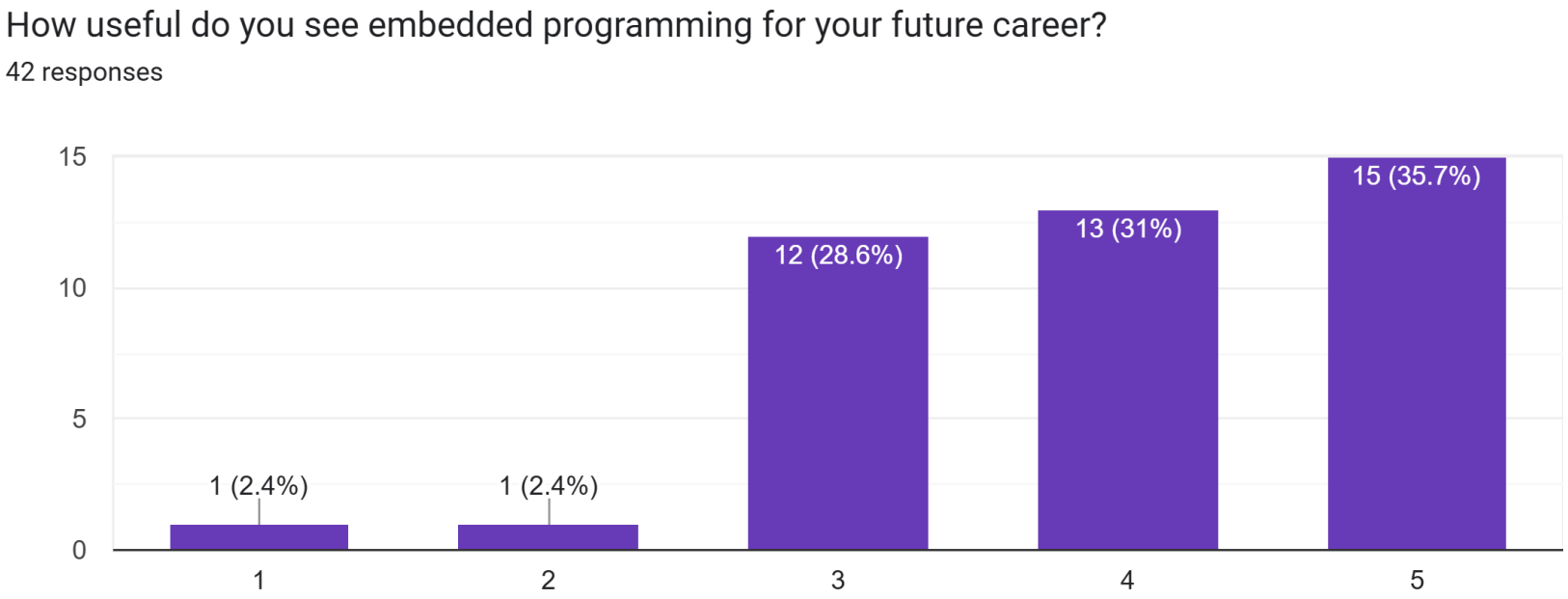


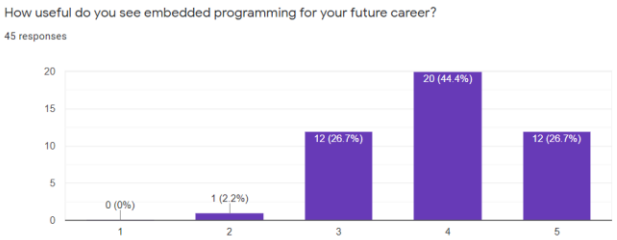


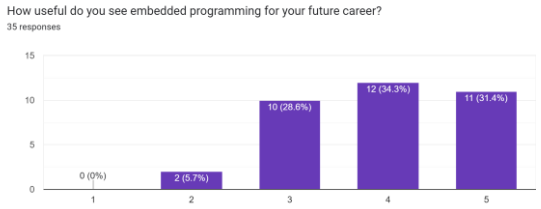




About you – responses to online form







About you – expecting to learn

I hope to learn how to design, implement, test, and debug reliable embedded software on real hardware, so I can build complete working systems in a structured, industry relevant way

Mostly how it's usefull for power electronics

Setting up fast reaction hardware on devices with constraints like missiles and drones

Implementation oriented programming that focuses on actually being solution oriented to real world problems.

I just want to get better at programming in general, this course does that.

Probably debugging and the use and programming of peripheral devices

I hope to get a deeper understanding of micro controllers and their capabilities. Maybe working with some rtos

Different use cases of embedded systems and the details of their usage.

the bridge between hardware and software

Good ways and practices to make sure my boards work

We have had a course in microcontrollers on 3rd semester, so I am hoping to learn more about how to approach embedded programming in larger systems.

In the previous C/C++ and Python programming courses i took, they were mostly related with software engineering. Maybe with this course i can also observe my code's outcome physically with the help of a hardware

I hope to get a better understanding of the architecture of a microcontroller to better optimize my code. For my career I would like to be better at actually making different embedded systems

SDU

About you – expecting to learn

What embedded systems are, How they work and when to use them. Also how to program them and make them perform the tasks I want.

More hands on programming

Experience

programming in C for embedded systems

Proper design techniques for embedded systems and coding

To wrangle microncontrollers like a cowboy wrangles snakes in his boots.

Getting comfortable with C programming and implementing it confidently in projects.

No idea haven't given it any thought

How to control, program and use small systems to control sensors and so on.

C

How it overall works

Coding different tasks

Using micro controllers

Real time control of hardware and reading sensor data.

About you – additional comments

I look forward to code :)

I hope you can teach us some cool stuff

No

From the course description it sounds very interesting

We have learned C programming, but we have not used it for dedicated GUI programming. I don’t know if this is expected, just a thought since you recommended the book “an introduction to C and GUI programming”.

I don't really have any expectations going in other than this course being relavent for our semesterprojekt

SDU

About you – self assessment

- Operating systems: Windows vs. Linux
- Integrated development environments?
- C or C++ experience, how much?
- Quality of C/C++ knowledge?

- Anything else?

SDU

About the course

- In English

About the course - prerequisite

- Completed 2 years of Electronic/Electrical Engineering study
- Basic knowledge in digital and analog electronics, computer science and programming

SDU

Learning objectives - knowledge

- C programming
- real-time systems, timing in embedded systems
- general input and output
- selected I/O-devices
- compiler, optimizing and preprocessor
- linking with assembler modules, incl. parameter transfer
- Interrupts, reentrance and recursion
- processes, threads, tasks and scheduling
- Inter-process communication, message boxes and queues
- Semaphores, mutexes, synchronization and critical sectors
- Deadlocks and memory administration

Learning objectives - skills

- Having completed this course, the student is able to:
  - explain the special demands that embedded systems place on software, and how to meet those demands by using the programming language: "C".

Learning objectives - competences

- Having completed this course, the student is able to:
  - analyze specifications of I/O-devices, and design high performance, hardware-near programs for these
  - evaluate real-time conditions in an embedded system
  - explain principles and algorithms for central parts of operating systems
  - design functions to a real-time operating systems (RTOS)

Assignments

- Introduced at the end of each class
- Programming problems to be solved as homework
- Groups of 3 students
- First assignment needs to be completed in 2 weeks
- Next assignments to be completed in 1 week each
- Everyone should do all assignments but only 3 randomly selected groups need to submit reports by the start of next class
- Each group needs to submit one assignment during the semester
- At the beginning of the next class one of the three groups is randomly selected to present their solution to the assignment (slides, working hardware)
- Need to submit slides before the deadline

SDU

Final assignment

- At the end of the semester
- Larger problem to be solved
- Poster needs to be submitted by all groups
- Presented as a poster at the last class in this semester
- The in-semester and final project poster need to be submitted as a prerequisite for attending the oral exam

SDU

Exam

- 7-point grading scale
- External censor
- Oral examination: individual, 16min
- The portfolios will be included as basis for the assessment
- Portfolios consist of reports on two assignments: one weekly and the final one
- One of four question areas is drawn randomly (state machines, run to complete scheduling, real-time operating systems, final assignment)
- Can start with a short presentation 3min max
- Questions start from drawn topic but can include other topics as well
- Final assignment usually comes up in some form

SDU

Embedded Programming part of two courses for some

- For robotek students it is a separate course
  - All robotek students take exam in EMP (this course)
- Exchange students all take exam in EMP
- For electronics students: Embedded Systems
  - Operating Systems – taught by Thor
  - At the exam you will be randomly selected to OS or EMP
  - You will be notified about this selection 1 day before the exam

SDU

End of semester student feedback from previous years

- This is not an “Intro to C” class
- “It is stupid that three groups prepare but only one presents” – on purpose
- “You didn’t tell us how exactly to solve the problem / where we can find that information in the datasheets” – on purpose – goal: to learn to find information by yourself and to learn how to read datasheets
- “You give us an assignment but only tell us how to solve it properly next week” – on purpose – there is not only one solution to an assignment – a way to introduce new topics
- “Reading from slides” – partially on purpose
- “I get to read loads of datasheets and I learned to help myself.”
- “Nice labs and assignments to force us into unknown territory.”
- “I found that the class was a bit hard as I had this class as an elective, but I also really learned a lot and I do actually appreciate that everything is not served on a silver platter regarding datasheets etc.”

SDU

Information contents of slides?

- Lots of text
  - Pro: helps you in studying for the exam
  - Con: The teacher might read from them
- Little text
  - Pro: The teacher can’t read it
  - Con: You have less information for preparing for the exam

SDU

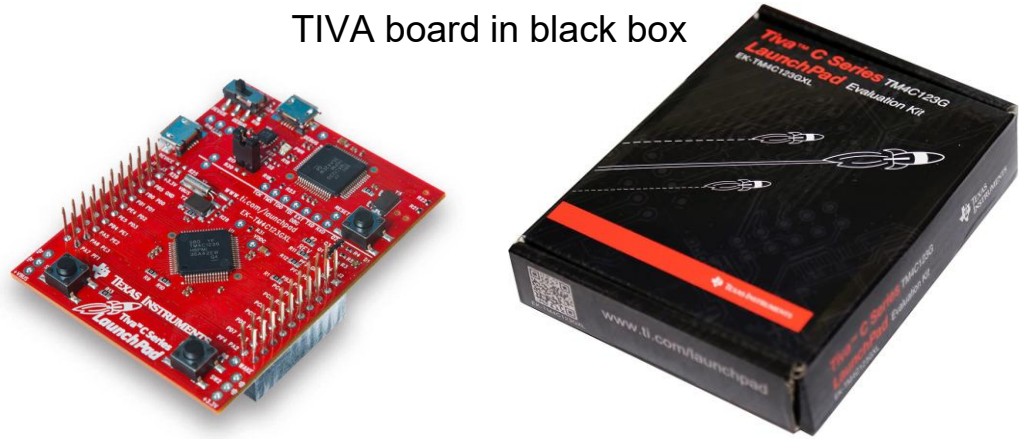
Lesson structure

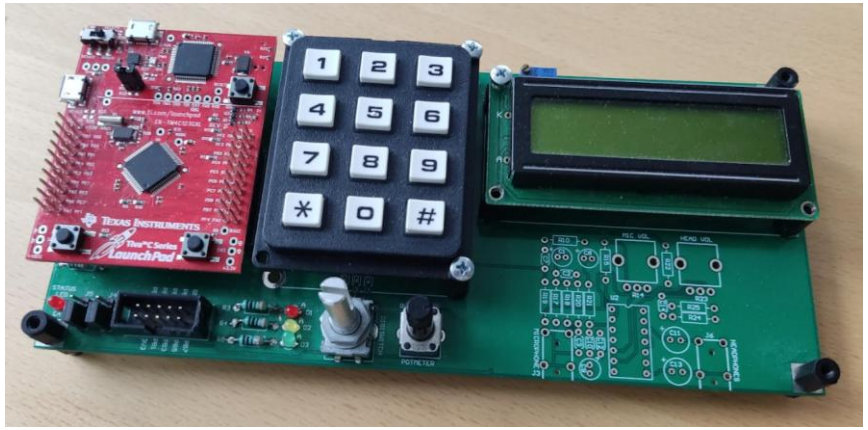
Student presentation of last week’s assignment
Lecture(s) of the day
Introduction to next week’s assignment
Lab work

Hardware platform

- TM4C123G LaunchPad Evaluation Board
  - Tiva TM4C123GH6PMI Microprocessor
    - ARM Cortex-M4 CPU core
- EMP Expansion Board
- LEDs
- Pushbuttons
- LCD Display
- Matrix keyboard
- Encoder
- Analogue input
- Etc.

TIVA board in black box





EMP board in brown box

SDU

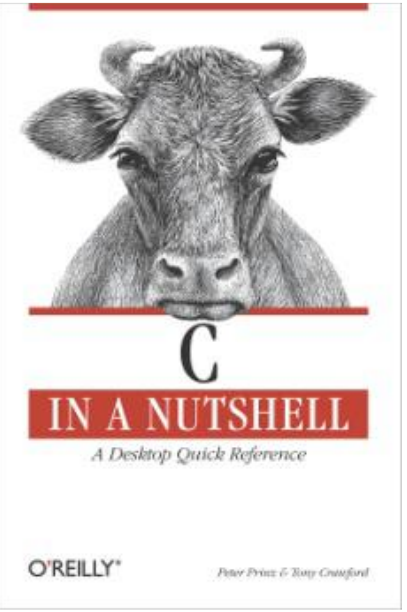
Software resources

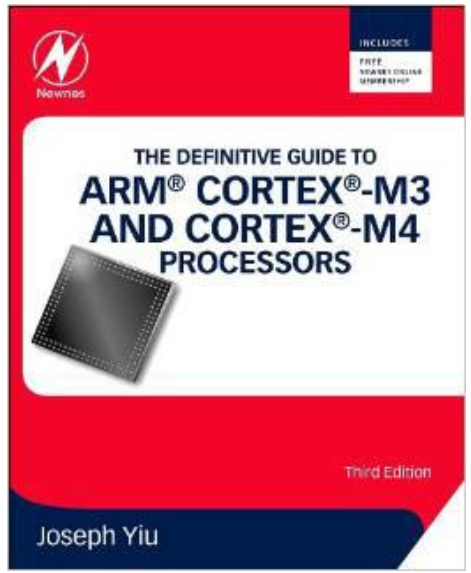
- Tool
  - Code Composer Studio IDE
    - Source code editor
    - Compiler and assembler
    - Downloader
    - Debugger

SDU

Literature

- Suggested but not mandatory books:
  - Peter Prinz, Tony Crawford, C in a nutshell.
    - ISBN: 0-596-00697-7.
  - Joseph Yiu, The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors, Third Edition.
    - ISBN: 0-124-08082-0
  - Simon Long, An Introduction to C & GUI Programming.
- Datasheets
  - TivaTM4C123GH6PMI
  - ARM Cortex M4
  - Find them on itslearning – Resources - Datasheets
- Online e.g. ti.com





Preliminary course outline

- **The plan is indicative, subject to change!**

Date	Lecture	Subject(s)	Lab	Assignment due
Feb 2	1	Introduction; ARM Cortex-M4	Lab1: Setting up the development env.	
Feb 9	2	EMP coding standard; Bit manipulation	Lab2: Bit manipulation	
Feb 16	3	State machines; The compiler chain; EMP-Board	Lab3: State machines	PF1 – Assignment 1
Feb 23	4	The task model; The pre-processor	Lab4: A clock radio task model	PF1 – Assignment 2
Mar 2	5	Queues and semaphores; Debugging	Lab5: Debug with a serial connection	PF1 – Assignment 3
Mar 9	6	Run to complete scheduler	Lab6: RTCS, Run to compile scheduler	PF1 – Assignment 4
Mar 16	7	More debugging; C: printf()	Lab7: RTCS, Debugger	PF1 – Assignment 5
March 23	8	FreeRTOS	Lab8: FreeRTOS	PF1 – Assignment 6
March 30 April 6		Easter holiday		
Apr 13	9	More queues; Assembler in C	Lab9: FreeRTOS (continued)	PF1 – Assignment 7
Apr 20	10	Re-entrance		PF1 – Assignment 8
Apr 27	11	Work on the final assignment, consultations		
May 4	12	Poster session		PF2- Final Assignment

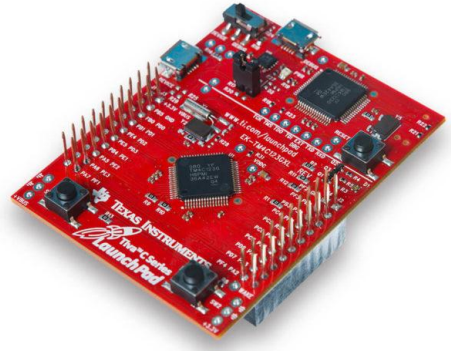
SDU

Get your Tiva C Series TM4C123G LaunchPad Evaluation Board from the library

- Everyone should have their own Tiva board (black box)

- If issues with the board:
  - Return to library, tell them it's faulty
  - Write down on a piece of paper what is wrong with the board
  - Get another board
  - If no boards left (unlikely), let me know

- Do not get the EMP board yet (brown box)



SDU

Task

- Create groups of 3 people
- Sign up here: https://tinyurl.com/sduemp2026