CC2511 Week 12

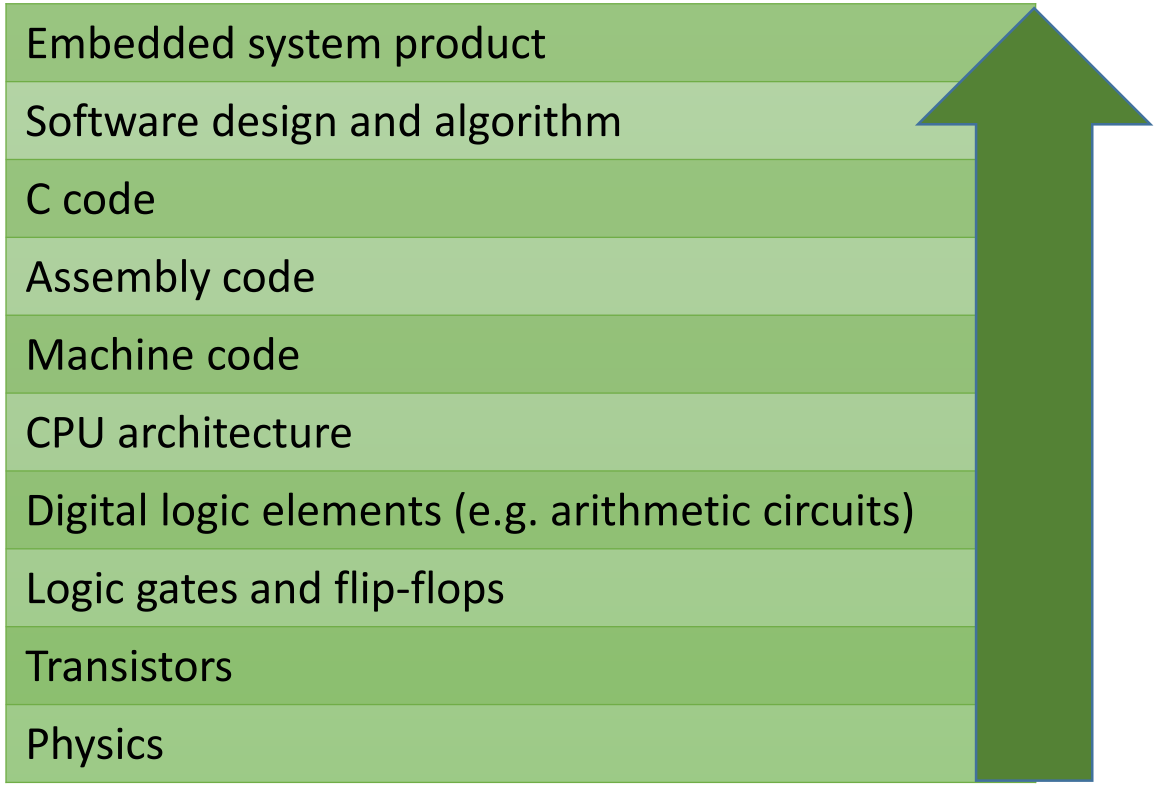
Today

- Revision and reflection
- Outlook:
  - What’s coming up in future subjects.
  - Employment in embedded systems.
- Exam preparation
  - What to expect and how to prepare.

Next week

- No lecture next week. Use the class time for assignment work.
- No assessable lab this week. The lab session is for assignment help.
- Your assignment is due on next week during the lab.

Revision: embedded systems from the bottom up

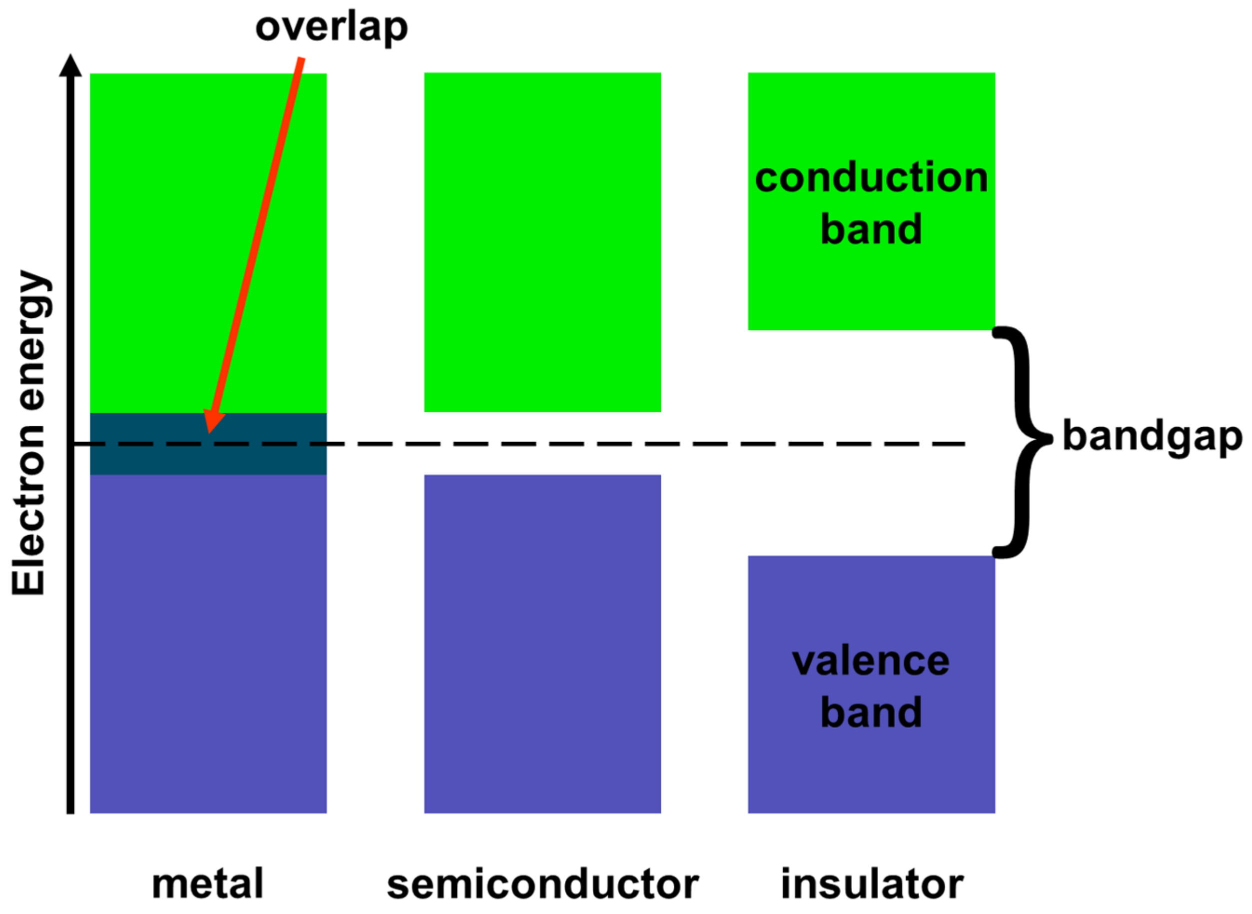


Each layer is an abstraction that we use to control the complexity. However, these abstractions are not perfect!

To be a good engineer, you need to be able to think across multiple levels and see how it all fits together.

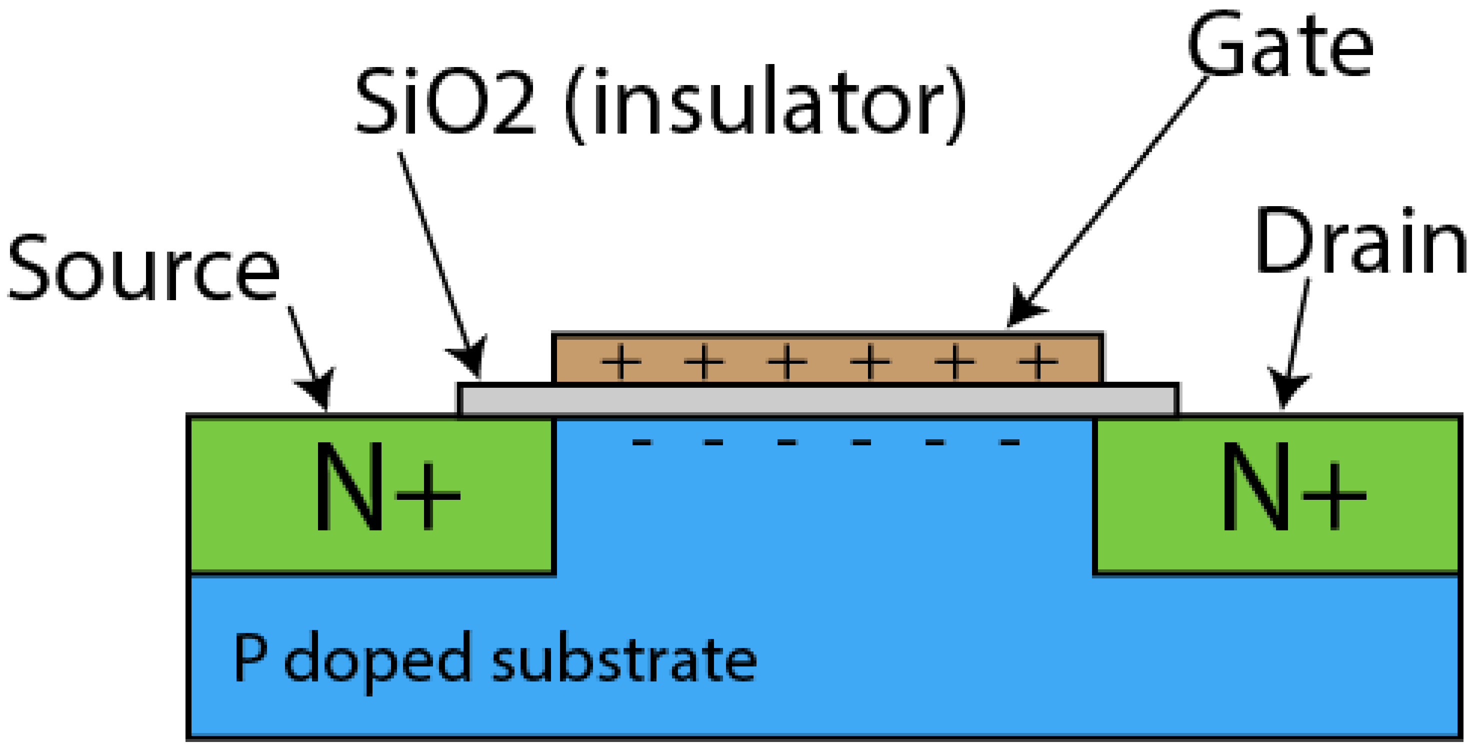
Semiconductors

Embedded system product
Software design and algorithm
C code
Assembly code
Machine code
CPU architecture
Digital logic elements (e.g. arithmetic circuits)
Logic gates and flip-flops
Transistors
Physics



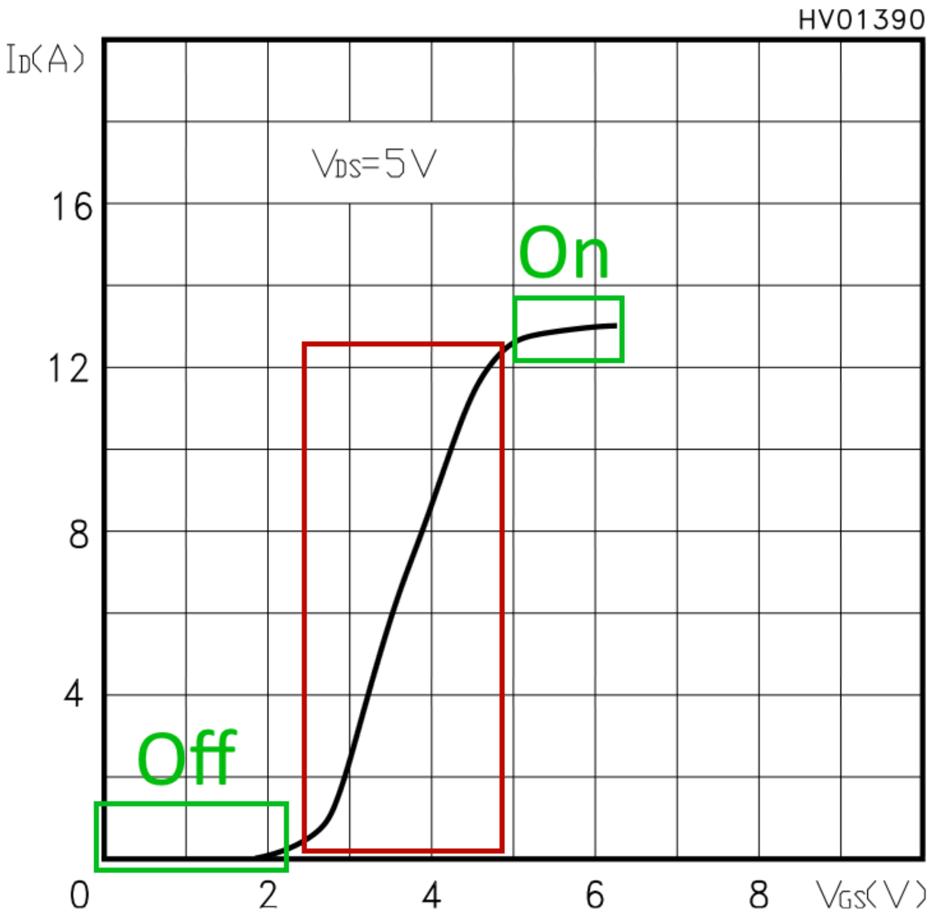
Transistors

Embedded system product
Software design and algorithm
C code
Assembly code
Machine code
CPU architecture
Digital logic elements (e.g. arithmetic circuits)
Logic gates and flip-flops
Transistors
Physics



Transistors

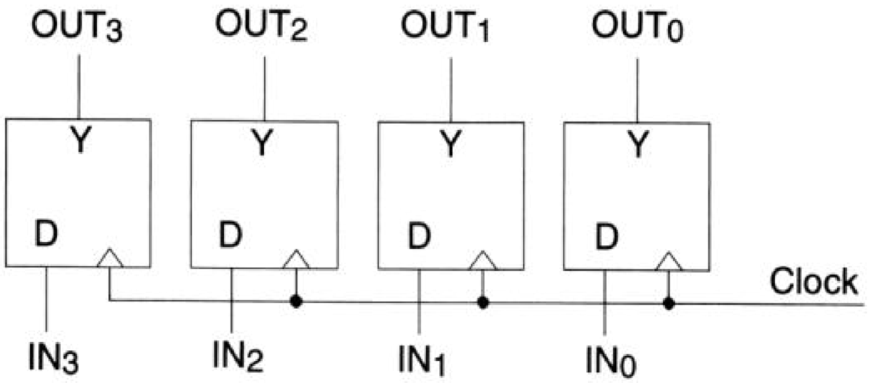
Embedded system product
Software design and algorithm
C code
Assembly code
Machine code
CPU architecture
Digital logic elements (e.g. arithmetic circuits)
Logic gates and flip-flops
Transistors
Physics

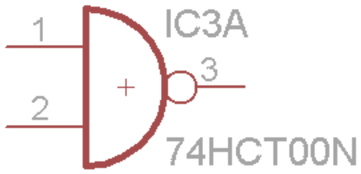


Logic gates

Embedded system product
Software design and algorithm
C code
Assembly code
Machine code
CPU architecture
Digital logic elements (e.g. arithmetic circuits)
Logic gates and flip-flops
Transistors
Physics

- Transistors are combined together to make logic gates (AND, OR, NAND, NOR, …) and flip-flops.
- Flip-flops are used to implement registers.



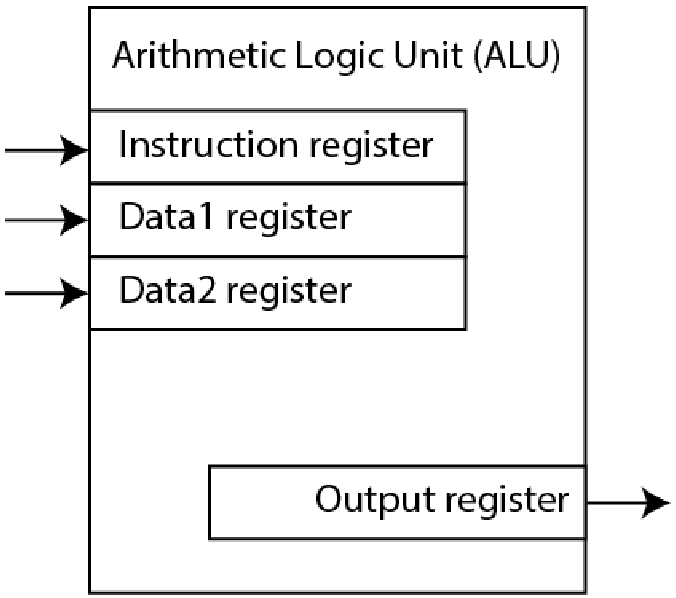


Arithmetic-Logic

Embedded system product
Software design and algorithm
C code
Assembly code
Machine code
CPU architecture
Digital logic elements (e.g. arithmetic circuits)
Logic gates and flip-flops
Transistors
Physics

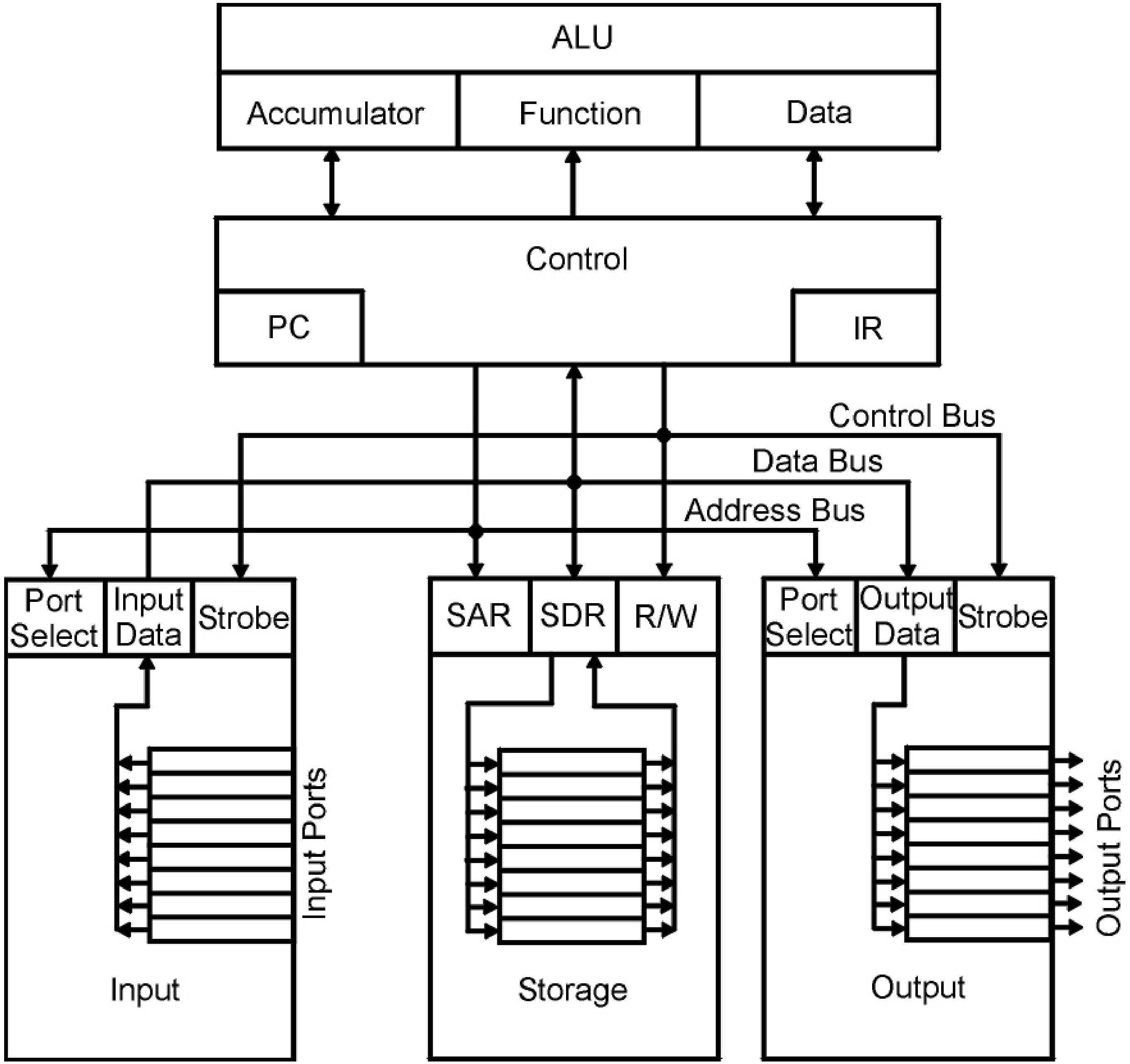
"Pseudo-VHDL":

```
-- IR = instruction register
-- OR = output register
if (clock is rising edge) then
    if (IR=1) then
        OR <= Data1 + Data2
    elsif (IR=2) then
        OR <= Data1 – Data2
    else ...
        -- other opcodes here
    end if
end if
```



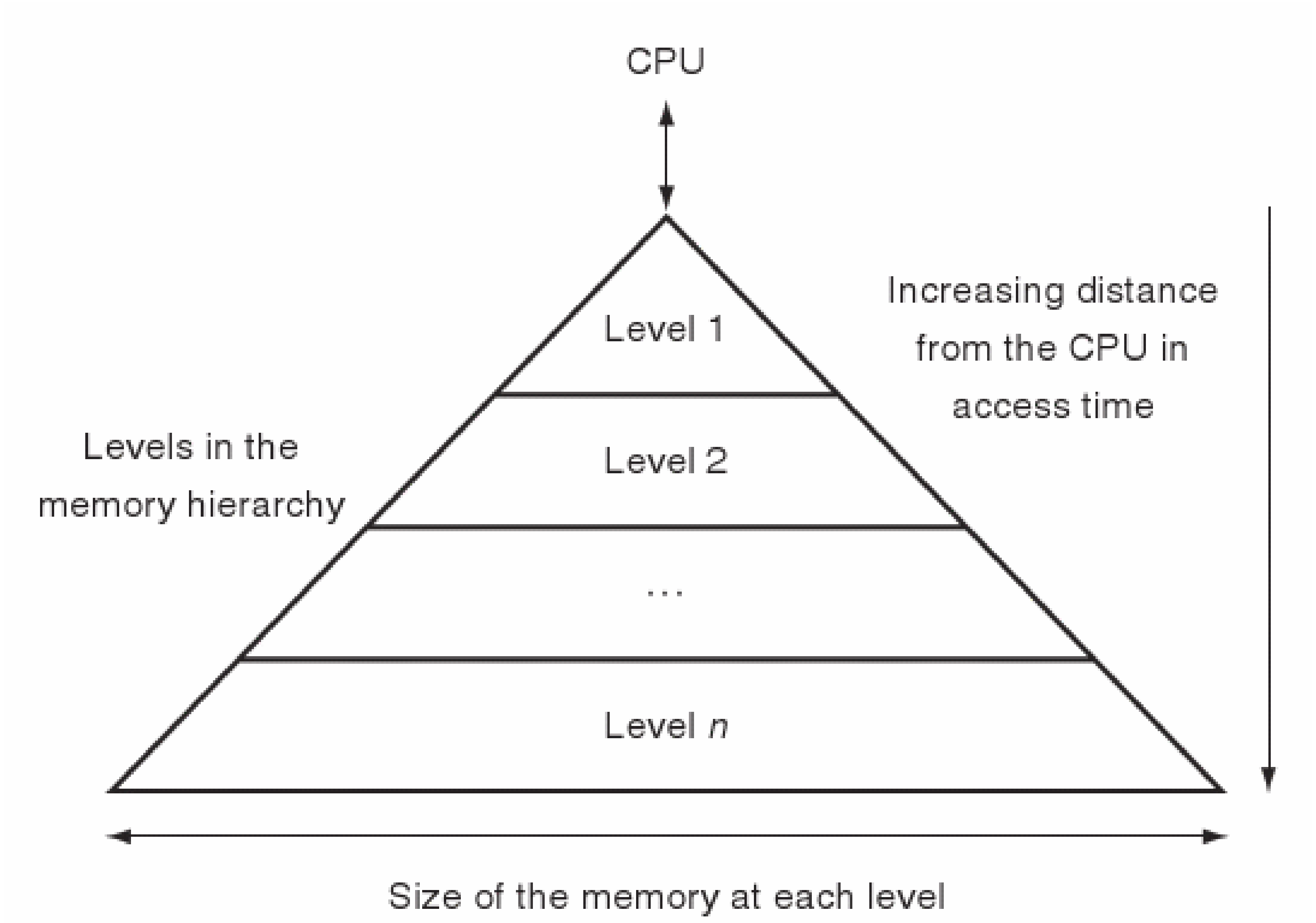
CPU architecture: toy model

Embedded system product
Software design and algorithm
C code
Assembly code
Machine code
CPU architecture
Digital logic elements (e.g. arithmetic circuits)
Logic gates and flip-flops
Transistors
Physics



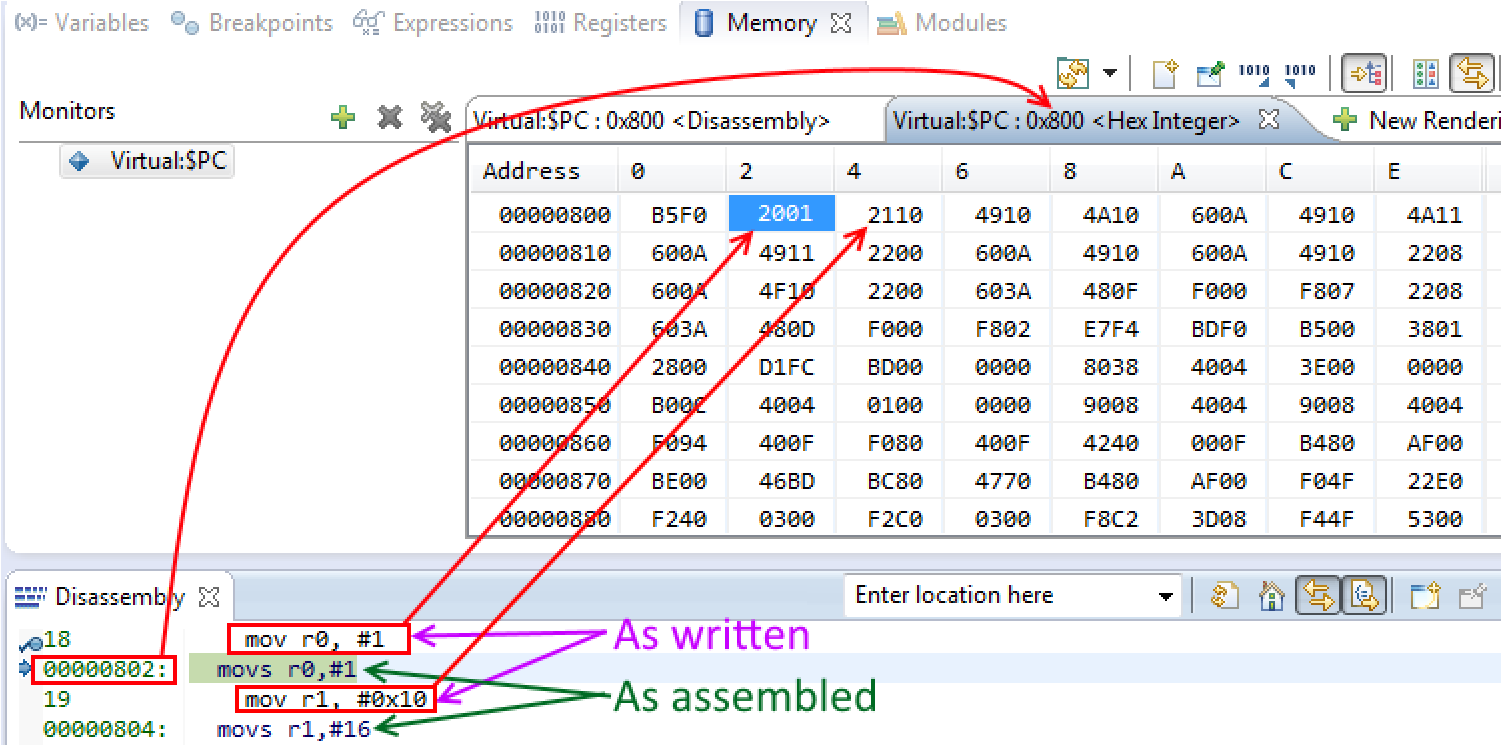
Memory hierarchy: registers-cache-memory

Embedded system product
Software design and algorithm
C code
Assembly code
Machine code
CPU architecture
Digital logic elements (e.g. arithmetic circuits)
Logic gates and flip-flops
Transistors
Physics



Assembly code

Embedded system product
Software design and algorithm
C code
Assembly code
Machine code
CPU architecture
Digital logic elements (e.g. arithmetic circuits)
Logic gates and flip-flops
Transistors
Physics



C code

Embedded system product
Software design and algorithm
C code
Assembly code
Machine code
CPU architecture
Digital logic elements (e.g. arithmetic circuits)
Logic gates and flip-flops
Transistors
Physics

- C code is generally used over assembly language because it is:
  - Usually more productive (requiring less programmer time), and
  - Easier to read and maintain.
- Assembly code is used in specialist situations such as:
  - Low level initialisation, and
  - When extremely high performance is needed.

Software design

Embedded system product
Software design and algorithm
C code
Assembly code
Machine code
CPU architecture
Digital logic elements (e.g. arithmetic circuits)
Logic gates and flip-flops
Transistors
Physics

- The software design is not independent of the lower layers!
- A good understanding of the fundamentals will help you design better software.

Overall product

Embedded system product
Software design and algorithm
C code
Assembly code
Machine code
CPU architecture
Digital logic elements (e.g. arithmetic circuits)
Logic gates and flip-flops
Transistors
Physics

- Software's purpose is to solve a problem and create a product.

Where to from here?

- The follow-on subject is **CC3501 Computer Interfacing and Control** where you'll extend your embedded systems knowledge to:
  - Multiple, cooperating CPUs
  - Communication between digital systems
  - Larger and more capable embedded processors
- You'll build PCBs in future subjects, and if you have an interest in hardware you might even develop a PCB as part of your final year thesis project.

Embedded Market Study: survey of engineers working on embedded systems

ASPENCORE

2017 Embedded Markets Study

Integrating IoT and Advanced Technology Designs, Application Development & Processing Environments

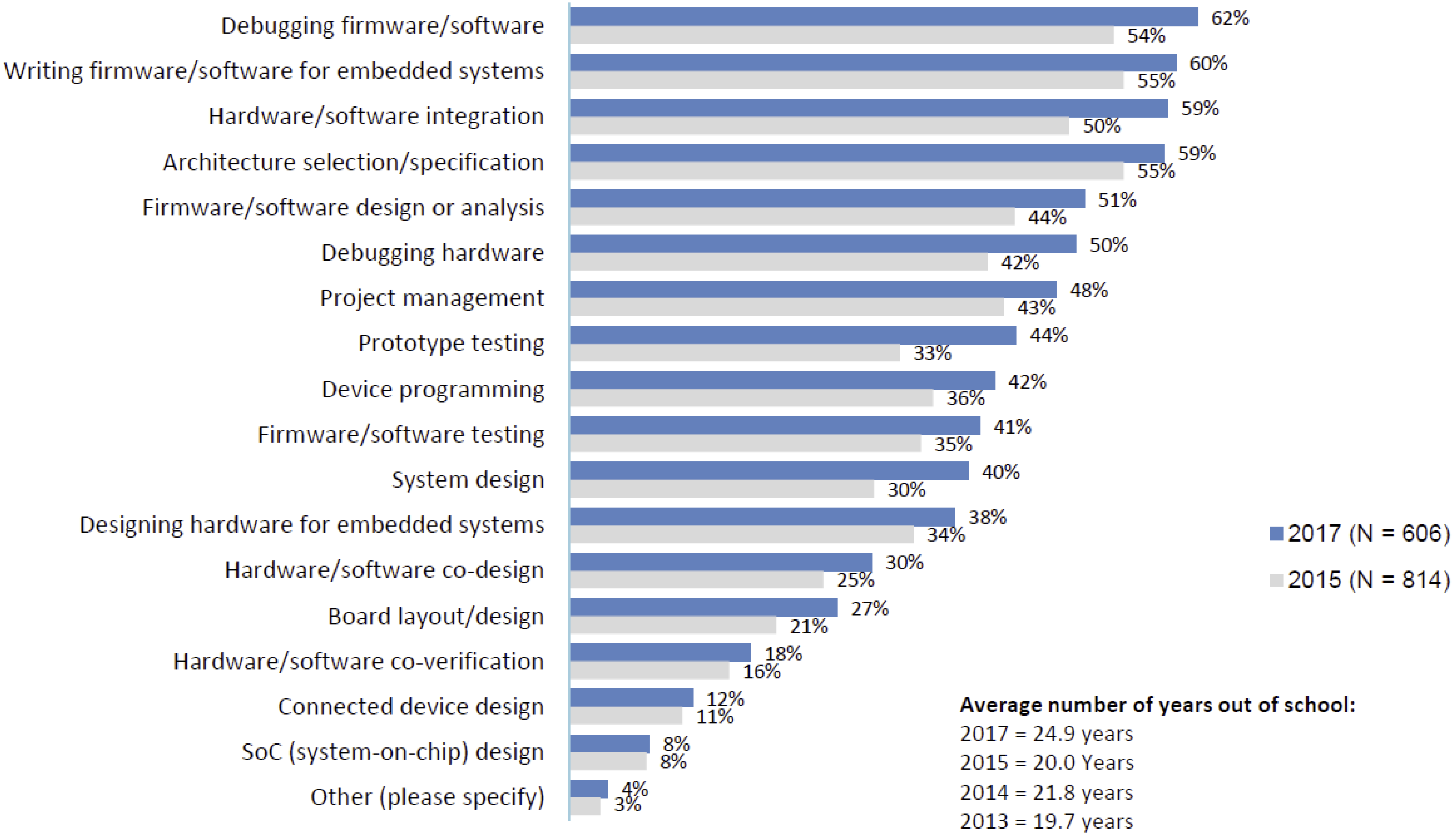
April 2017

Presented By: EETimes embedded

© 2017 Eetimes/embedded.com All Rights Reserved

ASPENCORE

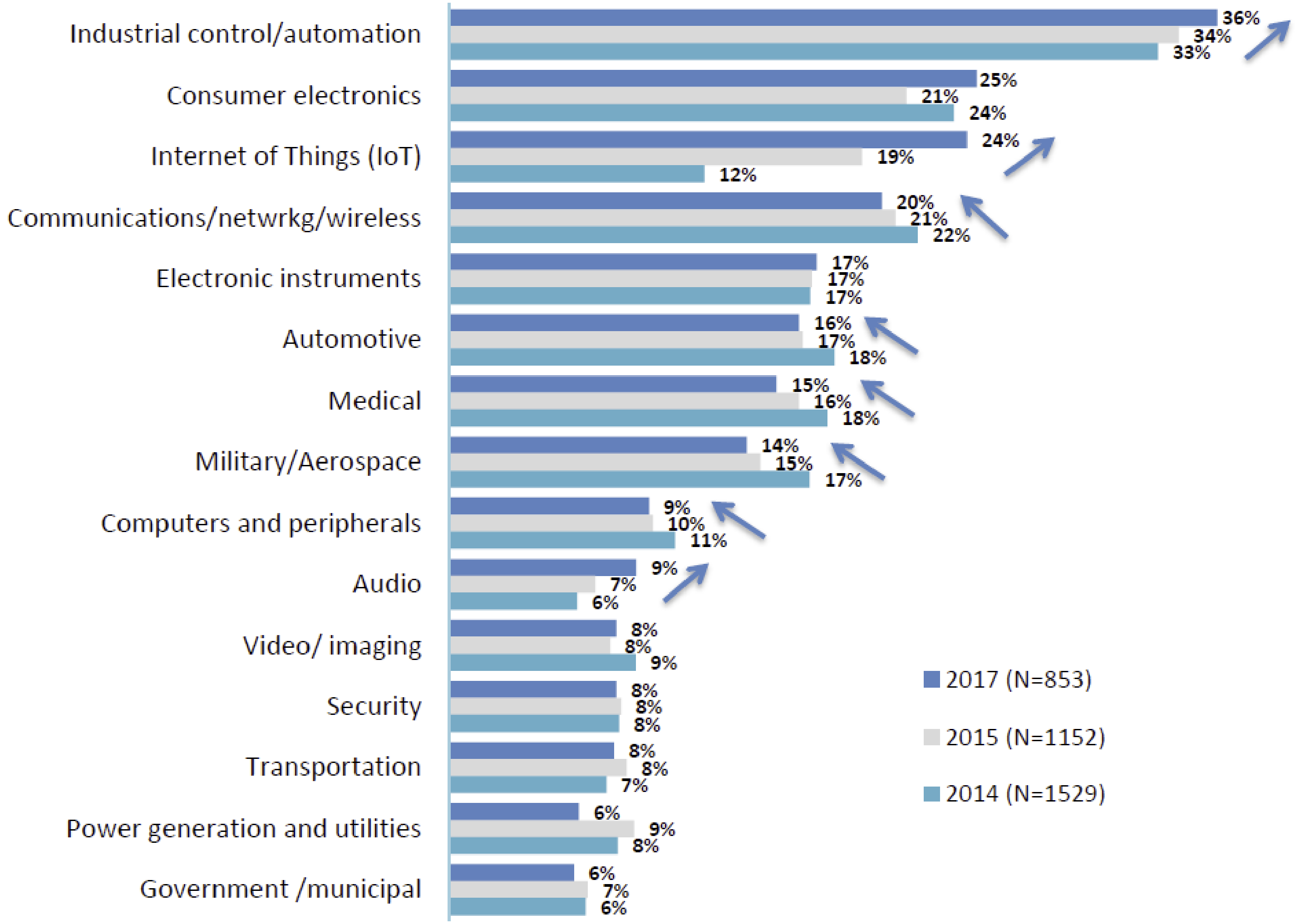
Job Functions



Average number of years out of school:
2017 = 24.9 years
2015 = 20.0 Years
2014 = 21.8 years
2013 = 19.7 years

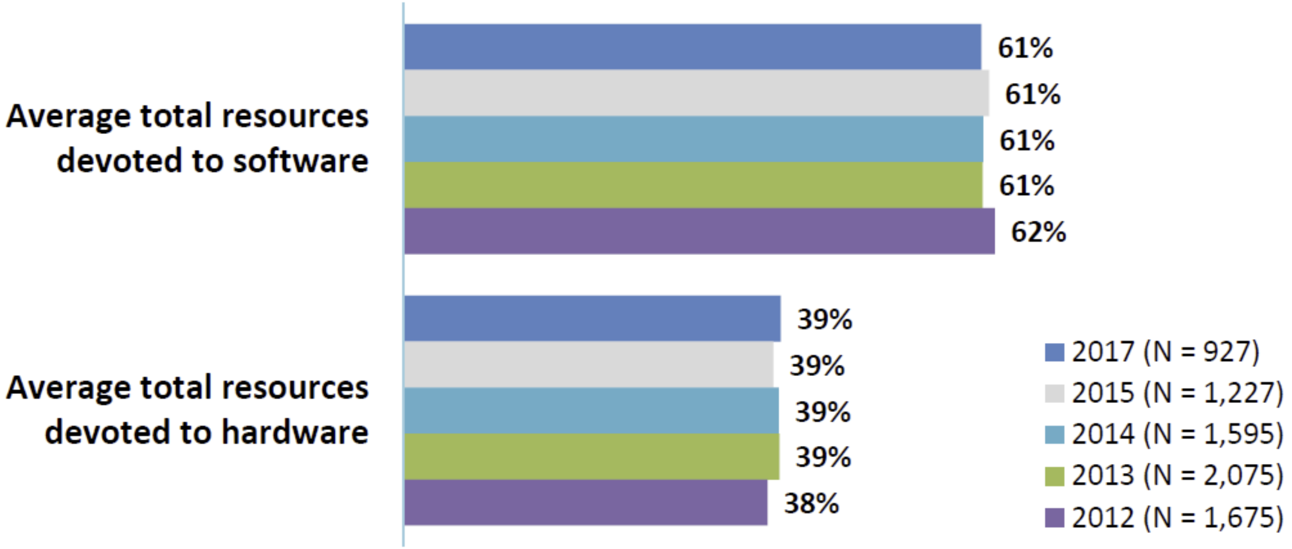
© 2017 Copyright by AspenCore. All rights reserved.

For what types of applications are your embedded projects developed?



© 2017 Copyright by AspenCore. All rights reserved.

What is your development team's ratio of total resources (including time/dollars/manpower) spent on software vs. hardware for your embedded projects?

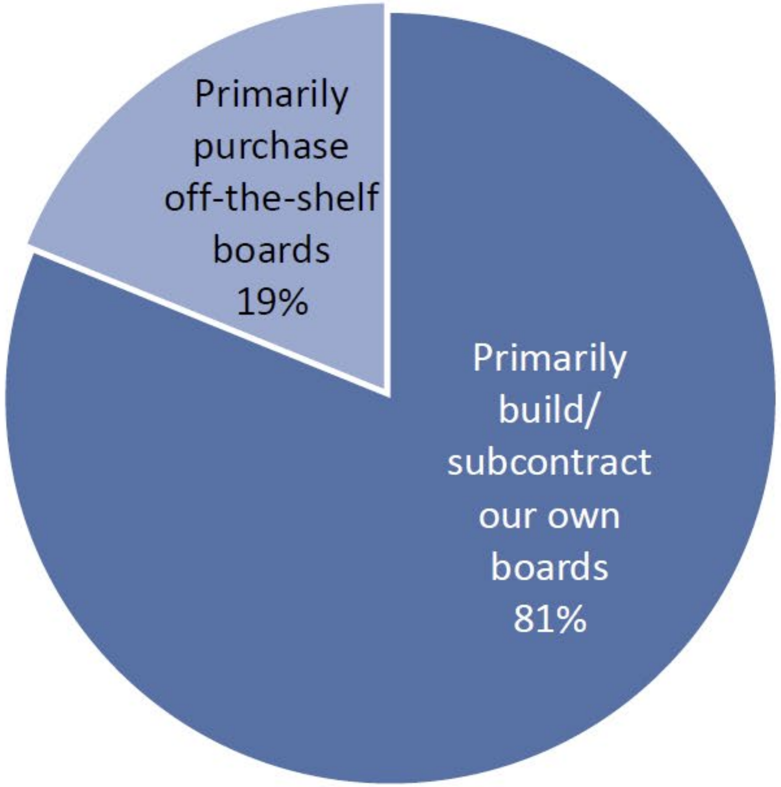


Note:
In 2017, respondents averaged working on 2.1 projects at the same time.
In 2015, respondents averaged working on 2.1 projects at the same time.
In 2014, respondents averaged working on 2.0 projects at the same time.

© 2017 Copyright by AspenCore. All rights reserved.

NEW IN 2017

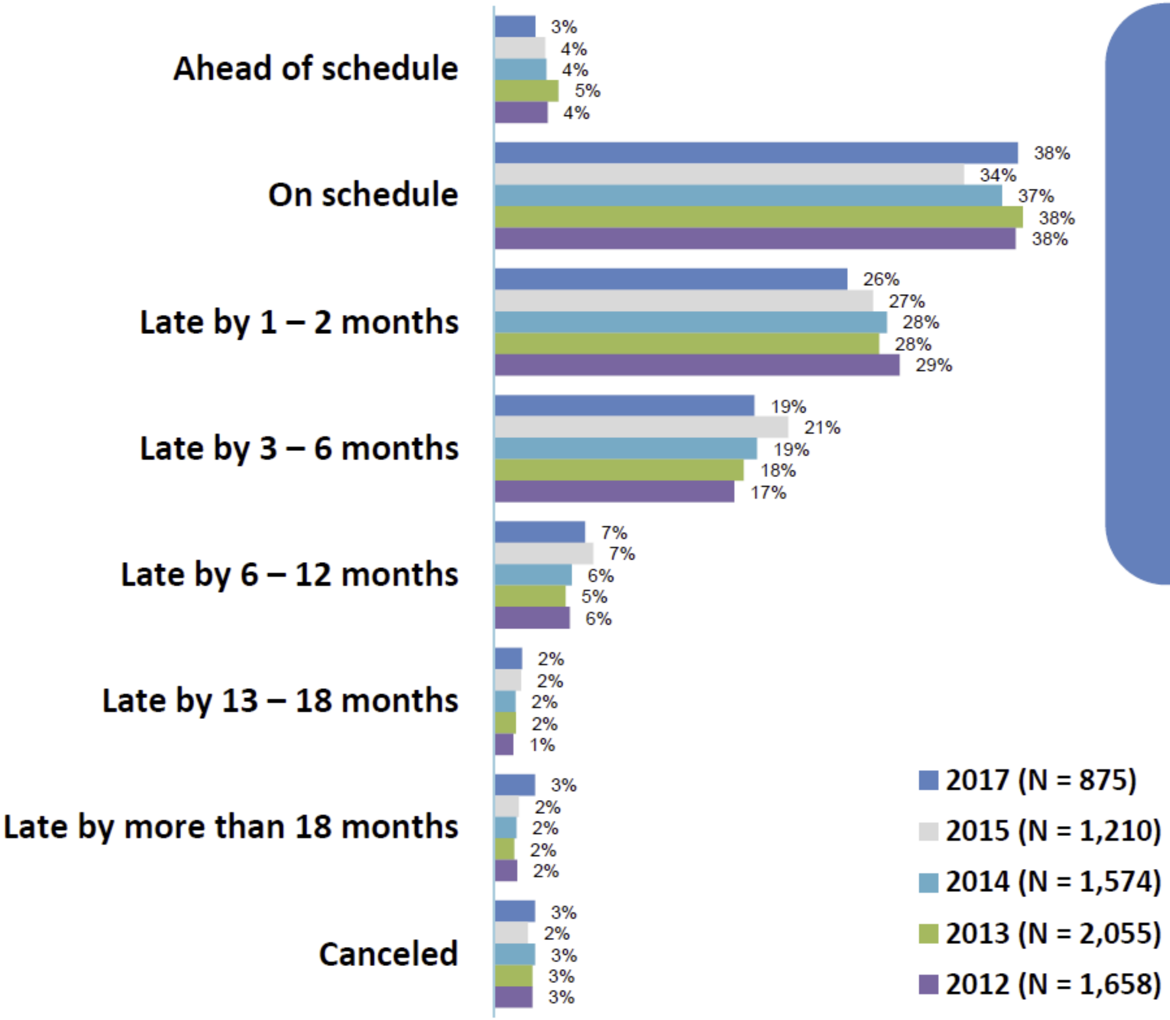
Do you primarily design or subcontract the design of custom circuit boards, or do you purchase off-the shelf boards?



2017 (N=923)

© 2017 Copyright by AspenCore. All rights reserved.

Was that project completed . . .



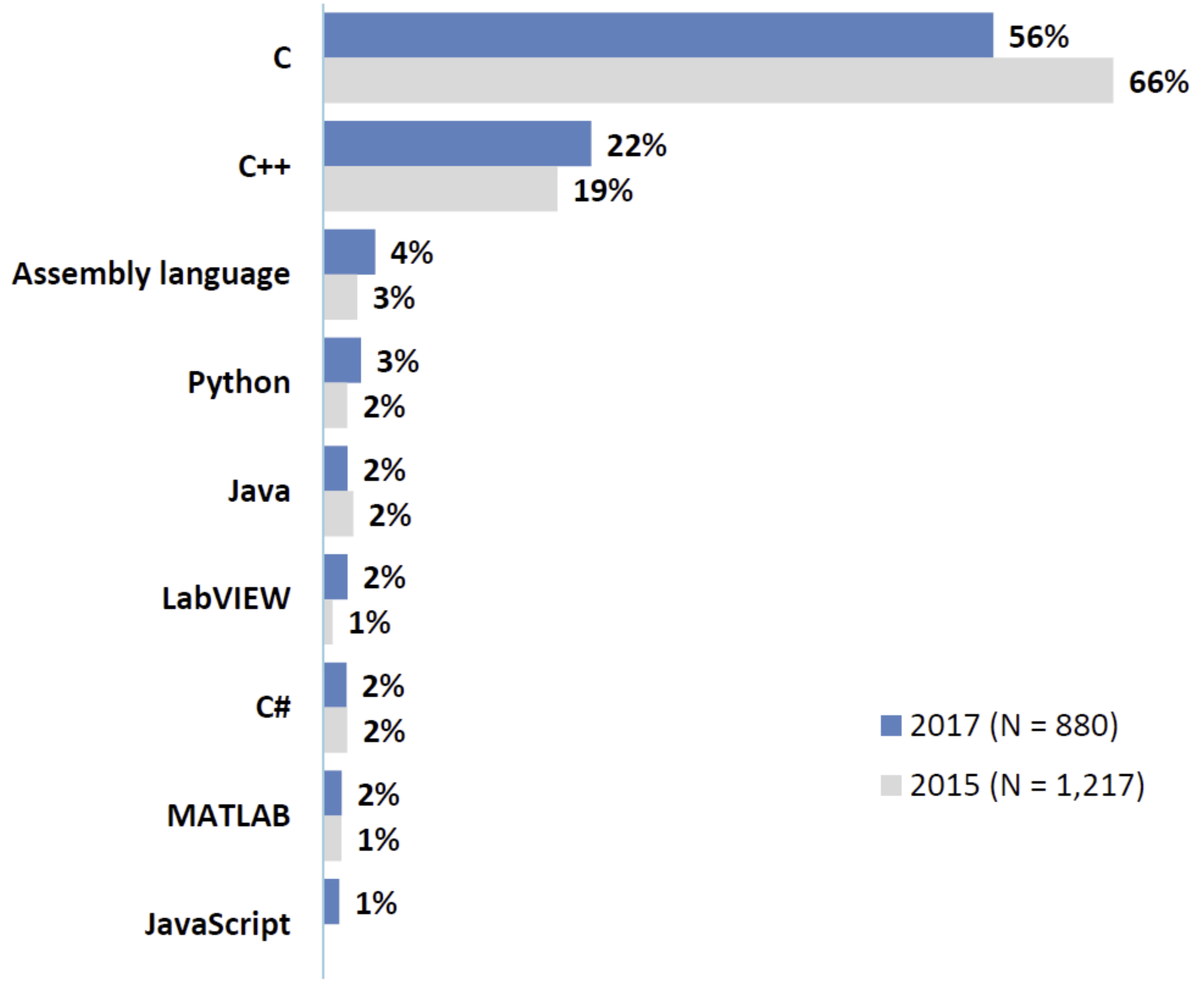
In 2017, 41% of all projects finished "ahead of" or "on" schedule, and 59% finished "late or cancelled".

In 2015, 38% of all projects finished "ahead of" or "on" schedule, and 62% finished "late or cancelled".

2017 performance has returned to the performance levels of the 2012-2014 that averaged 41%-44% "on/ahead of" schedule.

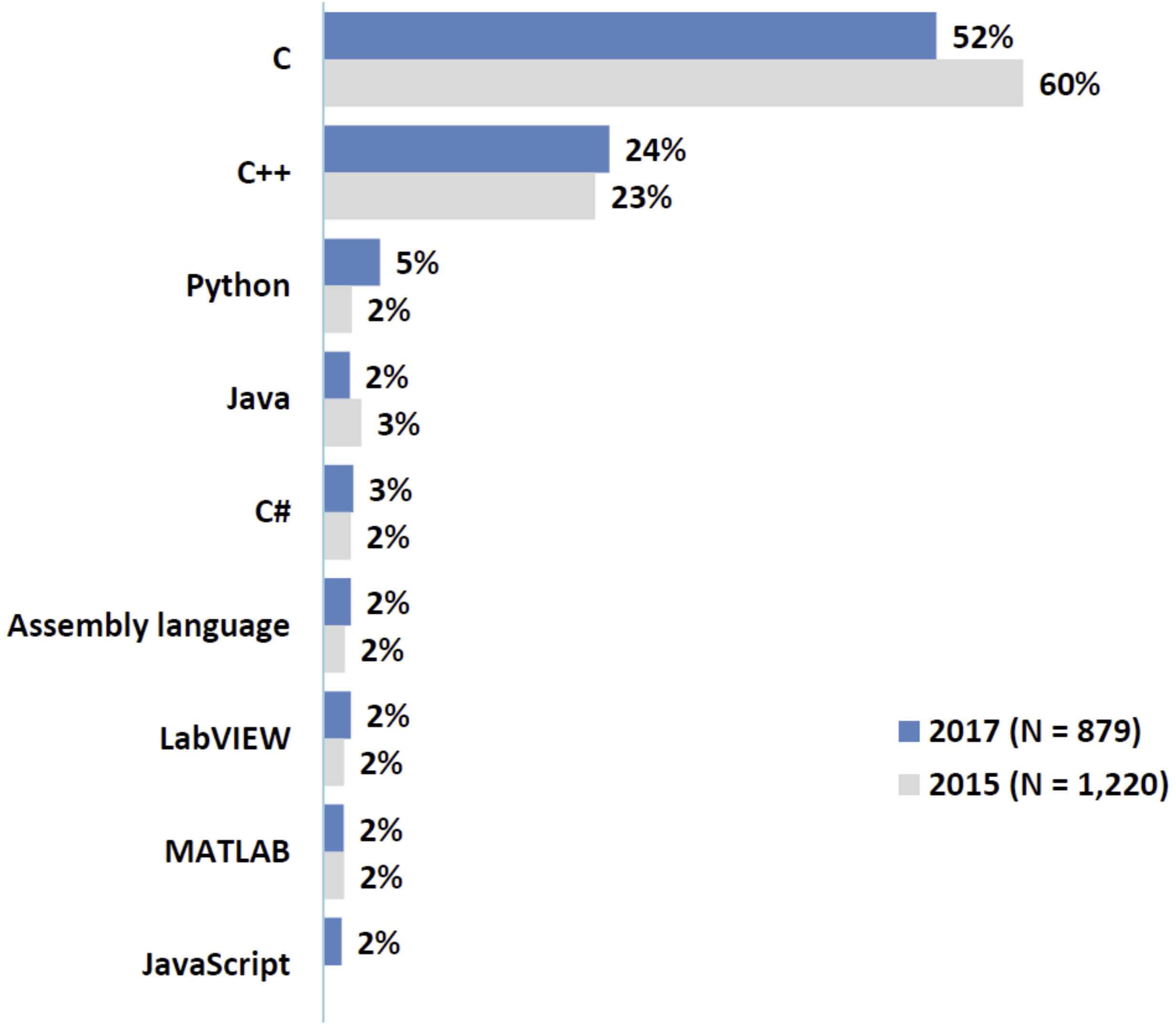
© 2017 Copyright by AspenCore. All rights reserved.

My current embedded project is programmed mostly in:



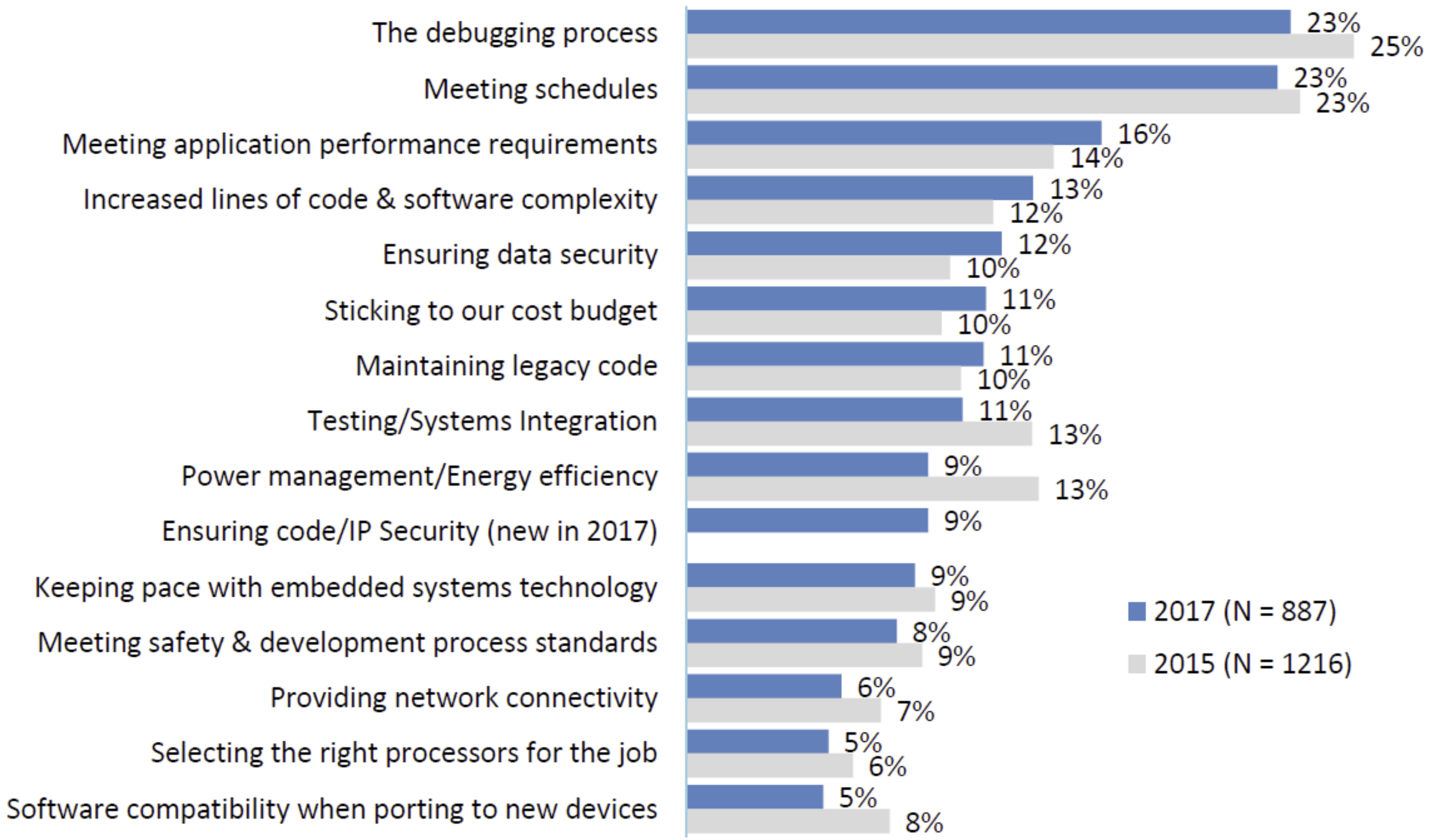
© 2017 Copyright by AspenCore. All rights reserved.

My next embedded project will likely be programmed mostly in:



© 2017 Copyright by AspenCore. All rights reserved.

Which of the following challenges are your own or your embedded design team's greatest concerns regarding your current embedded systems development?

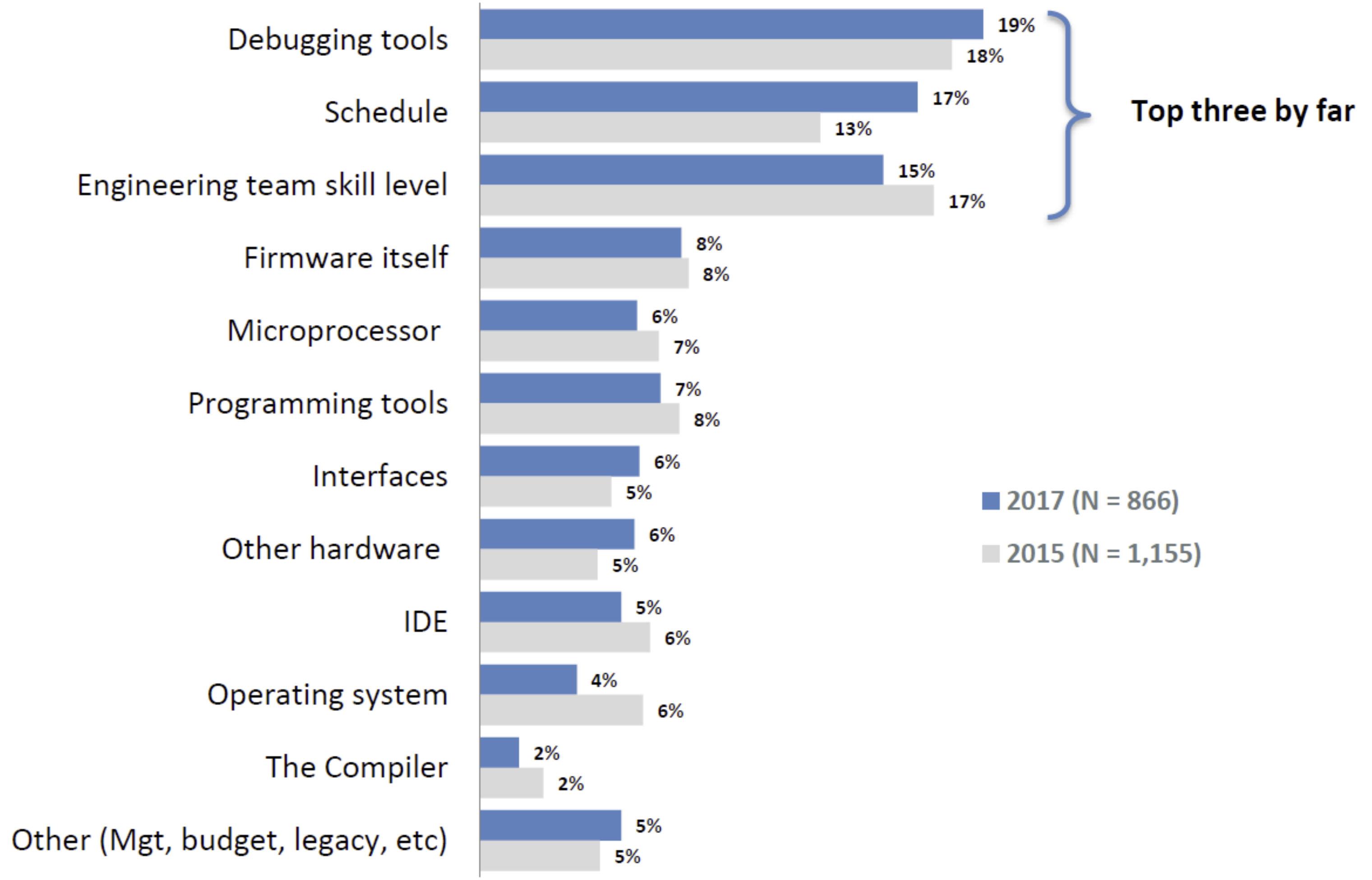


* Added in 2015

© 2017 Copyright by AspenCore. All rights reserved.

ASPENCORE

If you could improve one thing about your embedded design activities, what would it be?



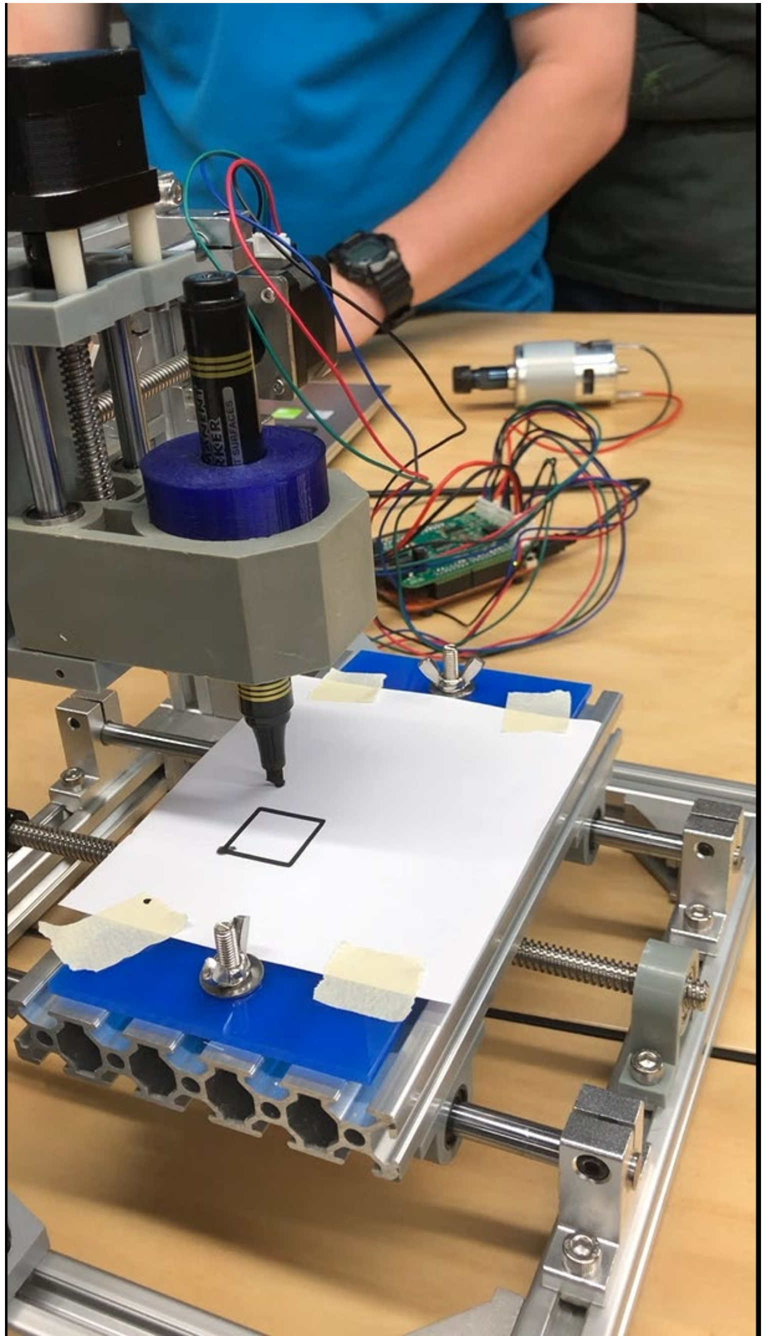
© 2017 Copyright by AspenCore. All rights reserved.

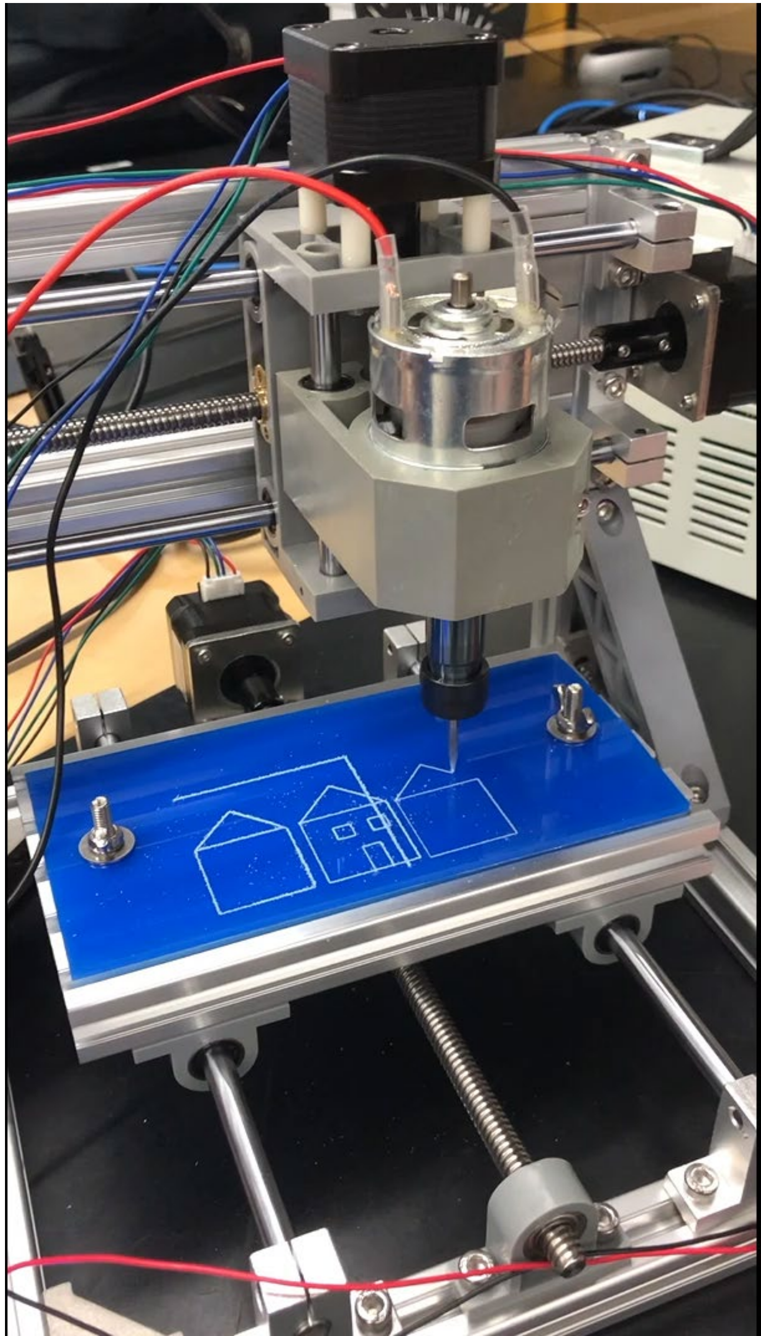
What’s next?

- Assignment. Due Friday 26 October.
  - **Demonstrate physical product** in the lab session.
  - Submit electronic copy of **your group’s software.**
  - Submit electronic copy of **your <u>individual</u> report.**
- Exam. Timetable to be posted.
  - Programming exam using the FRDM boards.

Assignment product demonstration

- Demonstrate your assignment in the final lab.
- I will ask each group to set up their product.
- I'll test your implementation and make note its functionality.





Assignment report

- **Each person writes a report.**
- **Keep it short**. Strict limit of 5 pages but a good report is probably only 2-3 pages.
- Task sheet has questions for you to respond to in the report.
- You must provide:
  - **Critical assessment of your design.** What worked well and what didn't? With the benefit of hindsight, what would you do differently? Show some insight here.
  - **A personal reflection/commentary** on your design process and your group's teamwork.

Preparation for the exam

- The exam is a programming task.
- Two hours duration + 10 minutes reading time.
- Held in the lab (E1-022 or 14-209)
- Full open book including Internet access.
- Worth 50% of your grade.

What to expect

- You will be asked to implement code to achieve particular objective(s).
- The equipment is a FRDM board.
  - That means the tasks must be related to the peripherals contained on the FRDM board: Serial communications, GPIO, PWM, ADC, …
- You may use any mix of C and assembly. It's your choice, provided that you implement the tasks required.
  - I expect most people to use C exclusively.

Previous exam

- A previous exam is published on LearnJCU under “Exam preparation”.
- This will show you the type of questions to expect.

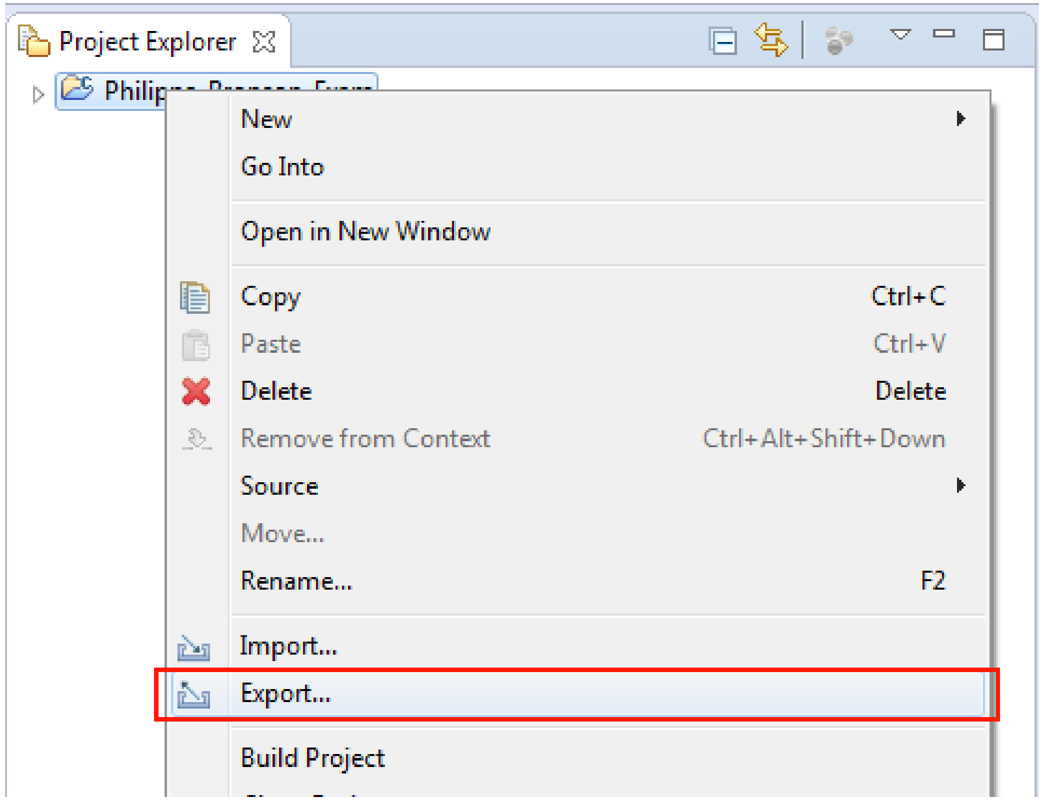
Administrative issues

- Before the exam begins:
  1. Fill in an attendance form
  2. Check that your computer works and that you can program the FRDM board.
- We will not start the timer until every person has checked that their computer is working.

Submitting your exam

The exam will instruct you to:

- **Name the project after your name: “Lastname_Firstname_Exam”**
- **To submit, export it to an archive file and email it to me.** CC yourself in the email and don’t leave the room until you see the message in your own inbox with the attachment present.



What to revise: C syntax

- You need to know C syntax well enough to be quick in using it.
- If you have been doing the labs every week then you probably already know enough C.
- You don’t want to waste time looking up basic syntax.

What to revise: common software skills

- In lectures and labs we studied common software skills that you might need to use in the exam:
  - String handling
  - Implementing mathematical operations
  - Arrays and array manipulation
  - … and others.
- You need to be able to apply these software design skills.

What to revise: common peripherals

- You need to be familiar with the microprocessor peripherals that we have studied in class.
- You need to know which peripheral to use for which task, and which Processor Expert components to use to configure those peripherals.
- The exam will only ask about peripherals used in labs or assignments.
    - You won't be asked to use the accelerometer, or the capacitive touch panel, etc.

What will not be on the exam

The exam will not ask:

- Circuit design
- Hardware considerations
- Boolean algebra
- Altium CAD software
- Version control software (e.g. Git)

How to prepare for the exam

- Make sure that you can easily implement all the software lab tasks.
- Write some non-trivial code for the major assignment. Writing real code is probably the best preparation that you can do.
  - It doesn’t matter if your teammates have already implemented feature X. Make your own implementation from scratch. You might even do a better job!
- **Prepare some good reference material to bring into the exam with you**. Consider what information you are likely to want to look up, and have it ready.

Reflection

- I hope that this subject is one of the highlights of your degree.
- You can now make physical electronics products and write software to drive them.
- Making your first circuit board is a key milestone in your career.

Student feedback

- Please complete the YourJCU student feedback survey on LearnJCU.
- **We really do listen to what you have to say**.

Thank you

- Thank you.
- Good luck with the assignment. Please come and see me at any time if you’re having problems.
- Good luck with the exam!