

# Getting started with your project

Software Project 2014

# Contents

- How to get started
- Engineer your system
- Protocol based projects
- System based projects
- Conclusions

# Get started

Follow these general steps:

1. Get documentation
2. Understand what you have to do!
3. Plan your steps, milestones and results
4. Design your architecture
5. Implement and test

# Plan your project

Define at least these items:

- The goal you want to reach
- 2-3 Milestones on the way
- Design a coarse timeframe around it
- Plan your resources (who does what)

**Important: How will you validate/verify your results?**

# Network/Application layer projects

1. Sketch your problem space, how to proceed with
  - Static typing, dynamic memory allocation
  - Constrained memory
  - Interfaces to the underlying system
2. Find reference implementations (e.g. in Java/JS/python...)
  - Do not re-invent the wheel
  - Useful for testing V&V
3. Create some reproducible test-cases
4. Create an architecture (e.g. some UML or similar diagrams...)
5. Hack it!
6. Test against
  - Predefined test cases reference implementation

# System level projects

1. Sketch your problem space, how to proceed with
  - Compile code for the target platform (cross-compiler)
  - Memories (which to use), MMU
  - Program and debug the target device
  - Interact with MCU peripherals
2. Write a very simple (but observable) program (e.g. blinky)
2. Flash your code and see its running
5. Hack RIOT
  - Start with I/O (uart, leds)
  - Then focus on task-switching
  - Leave the timer for last

# Conclusions

- Understand your project
- Don't start with 'some' implementing
- Be clear about your objectives beforehand
- Try to anticipate your problem space
- Have fun!

**And don't hesitate to ask for help!**