Welcome and Introduction

Lecture 1

Microprocessor-based Systems (E155)

Prof. Josh Brake

Outline

- Course learning objectives
- Administrative details
  - Syllabus
  - Schedule
  - Important links and information
  - Labs and Checkoffs
- Final Project
- Embedded System Introduction

Learning Objectives

Embedded systems expertise

- Use an ARM-based microcontroller to interface with the real world via sensors and actuators.
- Build an embedded system project of your own design from the ground up.
- Select appropriate embedded hardware for a given task.
- Read and understand complicated datasheets at a level that enables you to incorporate them into your designs.

Debugging and systems engineering

- Effectively and efficiently debug electrical systems with measurement tools such as an oscilloscope and logic analyzer.

Communication

- Clearly communicate technical results in a professional manner both through oral presentations and written reports.

Class Norms & Expectations

- Try to stay focused and engaged – I will do my best to make the lectures engaging and mix in activities.
- Use the chat or raise hand feature to ask questions – I want to hear your feedback and questions!
- If you are able, keep your video on.
- Collaboration
  - Attempt everything on your own first, but don't be afraid to ask for help!
  - Ask questions on Slack
  - Use Zoom rooms as collaborative lab workspaces
  - Create Github issues or pull requests for any bugs in code

Important Information

- Course website: http://pages.hmc.edu/brake/class/e155/
- Email list: eng-155@g.hmc.edu
- Slack Workspace: micropsfa20.slack.com

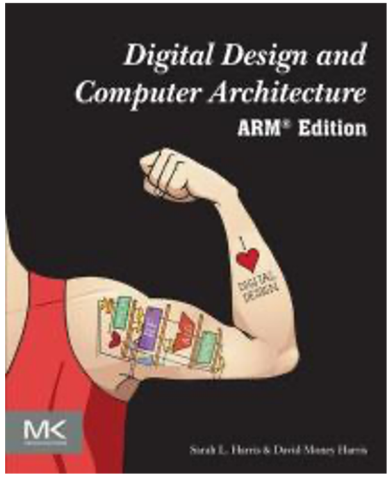
Office hours TBD based on clinic scheduling

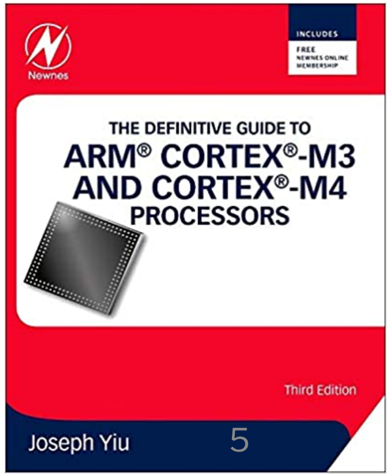
Course Texts

No official textbook, but a few helpful references:

- Sarah L. Harris and David Money Harris, *Digital Design and Computer Architecture: ARM© Edition*
- Joseph Yiu, The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors, 3rd Edition

Both books available for free as e-books through the library.



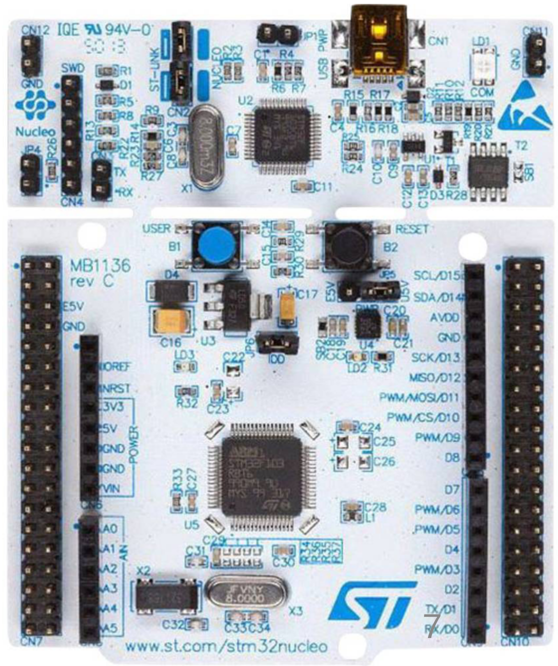


Tentative Course Schedule

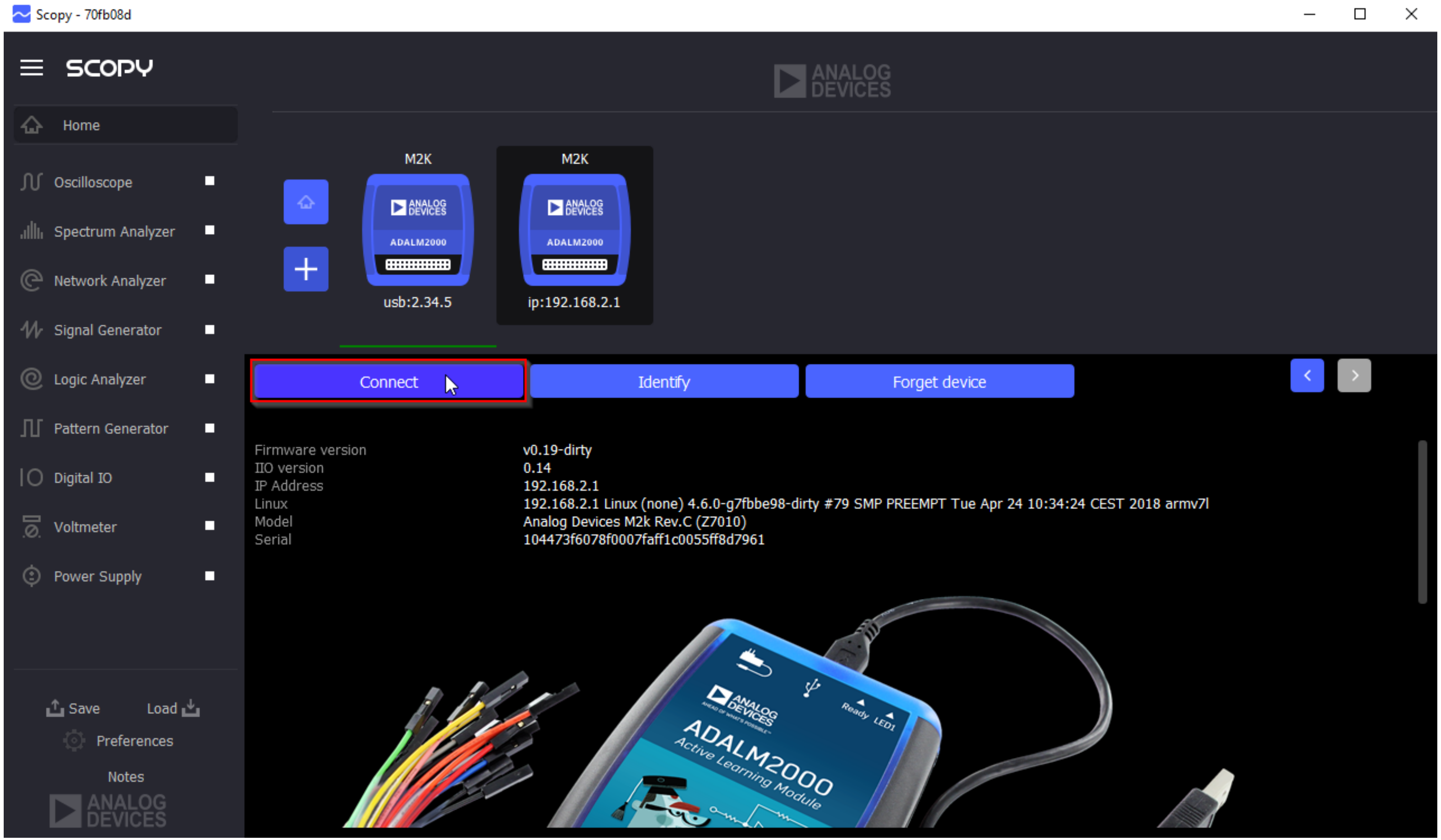
Week	Monday Lecture	Wednesday Lecture	Due
8/24	Intro and Overview	C Programming	(Lab Demos)
8/31	Toolchains	ARM Assembly Programming	Lab 1
9/7	STM32 Datasheet	Memory-mapped I/O	Lab 2
9/14	Processor Core	Peripherals	Lab 3
9/21	Interrupts	Common Digital Structures	Lab 4
9/28	Serial Interfaces	Internet of Things	Lab 5
10/5	Project Kickoff	ARM CMSIS	Lab 6
10/12	Digital Signal Processing	How to pick a micro	Lab 7
10/19	Custom Board Bringup	Motors and Speakers	Project Proposal
10/26	Graphics and Displays	Bootloaders	Proposal Debriefs
11/2	Digital Business	Advanced MCU Topics	
11/9	Patents and Intellectual Property	TBD	Project Status Report & Demo
11/16	TBD	Presentations	
11/23	Presentations	**Happy Thanksgiving! No class**	
11/30	Presentations	Presentations	Project Checkoffs & Final Demos Due

Kit

- ADALM2000
- Nucleo F401RE ARM Development Board
- Basic components kit (wires, resistors, capacitors, etc.)… and apparently no breadboard!
- External peripherals
  - LM 386 audio amplifier and speaker
  - Digital-to-analog convertor (DAC) integrated circuit
  - WiFi module (ESP8266)
  - Temperature sensor
  - Microphone



Scopy

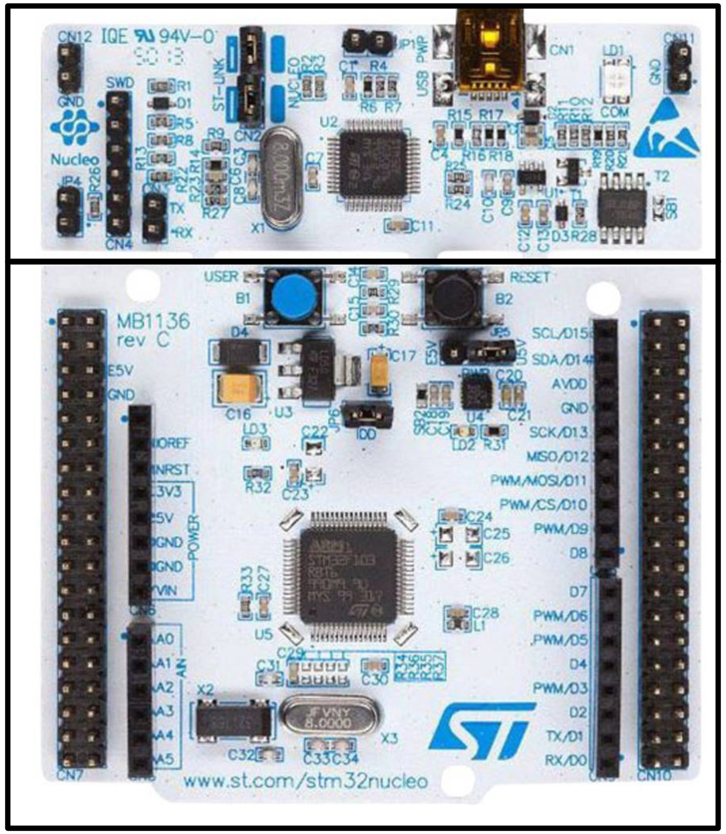


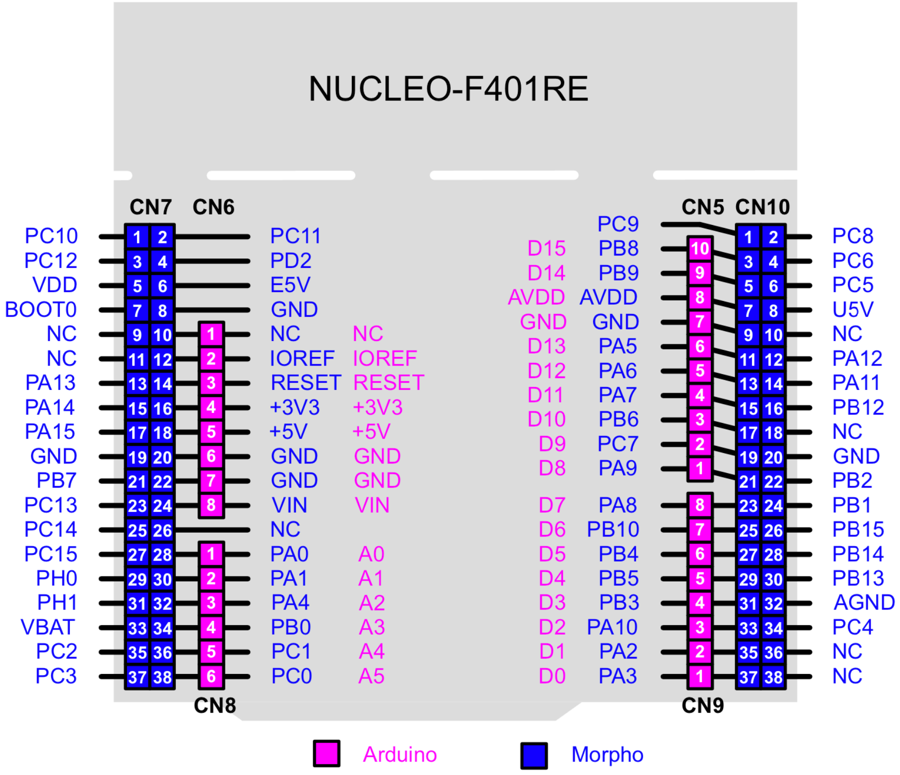
Scopy - 70fb08d
SCOPY
ANALOG DEVICES
Home
Oscilloscope
Spectrum Analyzer
Network Analyzer
Signal Generator
Logic Analyzer
Pattern Generator
Digital IO
Voltmeter
Power Supply
Save
Load
Preferences
Notes
M2K
ADALM2000
usb:2.34.5
M2K
ADALM2000
ip:192.168.2.1
Connect
Identify
Forget device
Firmware version v0.19-dirty
IIO version 0.14
IP Address 192.168.2.1
Linux 192.168.2.1 Linux (none) 4.6.0-g7fbbe98-dirty #79 SMP PREEMPT Tue Apr 24 10:34:24 CEST 2018 armv7l
Model Analog Devices M2k Rev.C (Z7010)
Serial 104473f6078f0007faff1c0055ff8d7961
ADALM2000
Active Learning Module

Nucleo-F401RE

ST-LINK

MCU and headers





Labs Overview

- Lab 1 - The GNU Toolchain
- Lab 2 - ARM Assembly Sort
- Lab 3 - Digital Audio
- Lab 4 - Serial Peripheral Interface
- Lab 5 - Pulse Width Modulation
- Lab 6 - Serial Temperature Sensor
- Lab 7 - The Internet of Things

Software Tools

Docker
(Labs 1-2)

VSCode & PlatformIO
(Labs 3-7)

Lab Checkoffs

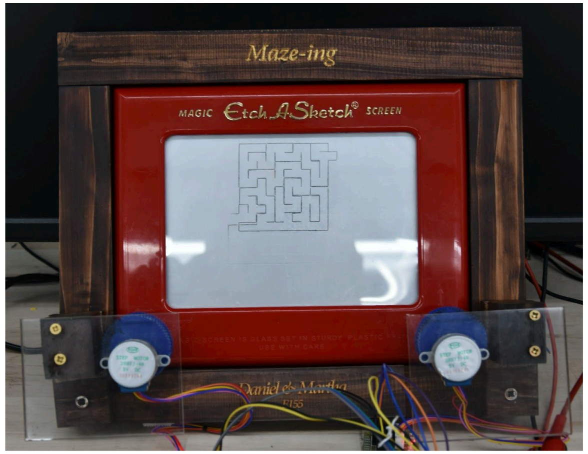
- Check the website for a link to the Google Sheet schedule and sign up by the end of this week. Try to find a time during your scheduled lab day.
- Weekly 10 min meeting on Zoom where you present your lab. Please join the waiting room a few minutes before your scheduled time.
- Submit lab writeup (.pdf) and supporting files (links to movies or photos) to Sakai ahead of time.
- Grading
  - Demonstration of the lab (screen share and/or show in front of camera)
  - Interactive grading of the report
  - Fault tolerance question – a general question about a concept related to the lab. Good interview practice!

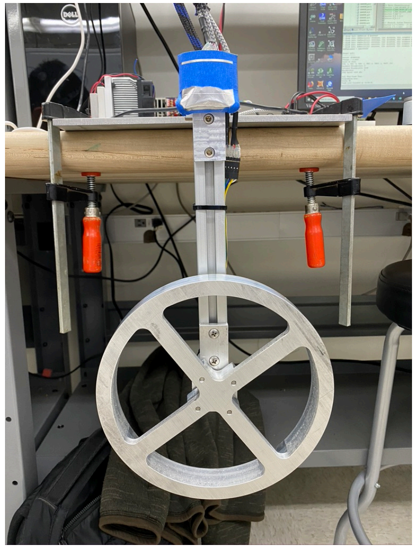
Drawing good schematics

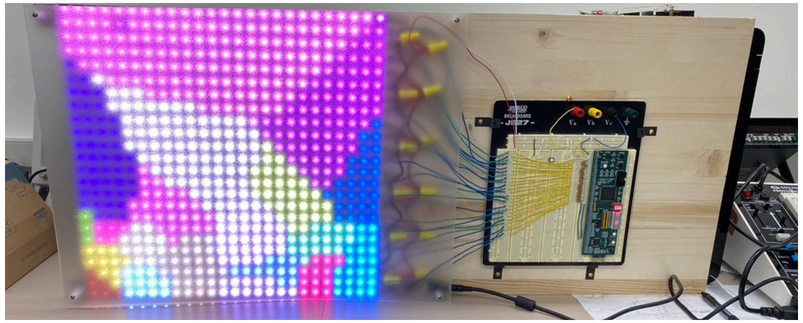
- Make sure everything is neat and clear. Someone should be able to replicate your design with just the schematic.
- Tips
  - Label pin names AND numbers.
  - Don’t need to go over the top with fancy computer drawn schematics, neat hand-drawn diagrams are fine.
  - Make sure you put values for all the components (e.g., resistor and capacitor values)

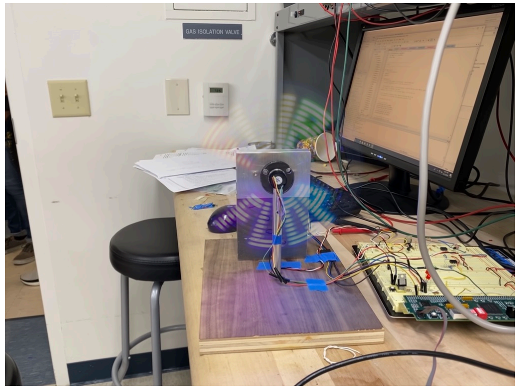
Final Project

- Start thinking about it early (i.e., now!)
- Teams of 2
- Project requirements
  - Use MCU to do something fun or useful
  - Incorporate new piece of hardware
  - Use advanced features of MCU
- Make sure it is feasible but not trivial.
- Key learning outcome is designing a system and delivering what you promise.









Embedded System Design

Embedded Systems

- What is an embedded system?
- What is a common field where embedded systems are used?
- What is an example of an embedded system you use regularly?

https://pollev.com/joshbrake155

What is an embedded system?

Embedded systems are information processing systems embedded into a larger product

Table 1.2 Distinction between PC-like and embedded system design

	Embedded	PC-/Server-like
Architectures	Frequently heterogeneous very compact	Mostly homogeneous not compact (×86 etc.)
×86 compatibility	Less relevant	Very relevant
Architecture fixed?	Sometimes not	Yes
Model of computation (MoCs)	C+multiple models (data flow, discrete events, …)	Mostly von Neumann (C, C++, Java)
Optimization objectives	Multiple (energy, size, …)	Average performance dominates
Safety-critical?	Possibly	Usually not
Real-time relevant	Frequently	Hardly
Applications	Guarantees for several concurrent apps. needed	Best effort approaches to run application
Apps. known at design time	Yes, for real-time systems	Only some (e.g., WORD)

Marwedel, Peter. *Embedded System Design.* Springer, 2018.

What are some common fields where embedded systems are used?

- Transportation
- Factory automation
- Health
- Smart buildings
- Smart grid
- Scientific experiments
- Public safety
- Structural health monitoring
- Disaster recovery
- Robotics
- Agriculture
- Military
- Telecommunication
- Consumer electronics
- …

Marwedel, Peter. *Embedded System Design.* Springer, 2018.

Challenges

Dependability

- Safety
- Security
- Confidentiality
- Reliability
- Repairability
- Availability

Resource Awareness

- Energy
- Run-time
- Code size
- Weight
- Cost

Marwedel, Peter. *Embedded System Design.* Springer, 2018.

Design Flow

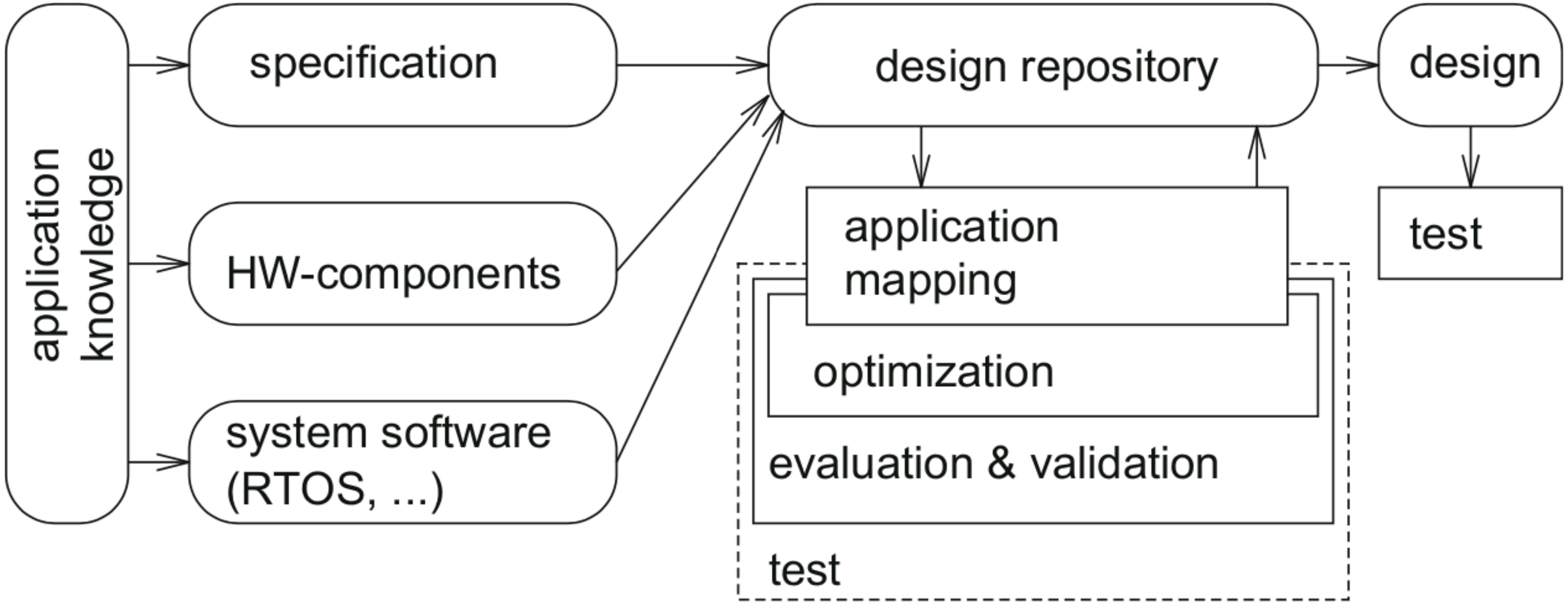


Fig. 1.8 Simplified design information flow

Marwedel, Peter. *Embedded System Design.* Springer, 2018.

Design Flow - Unfolded

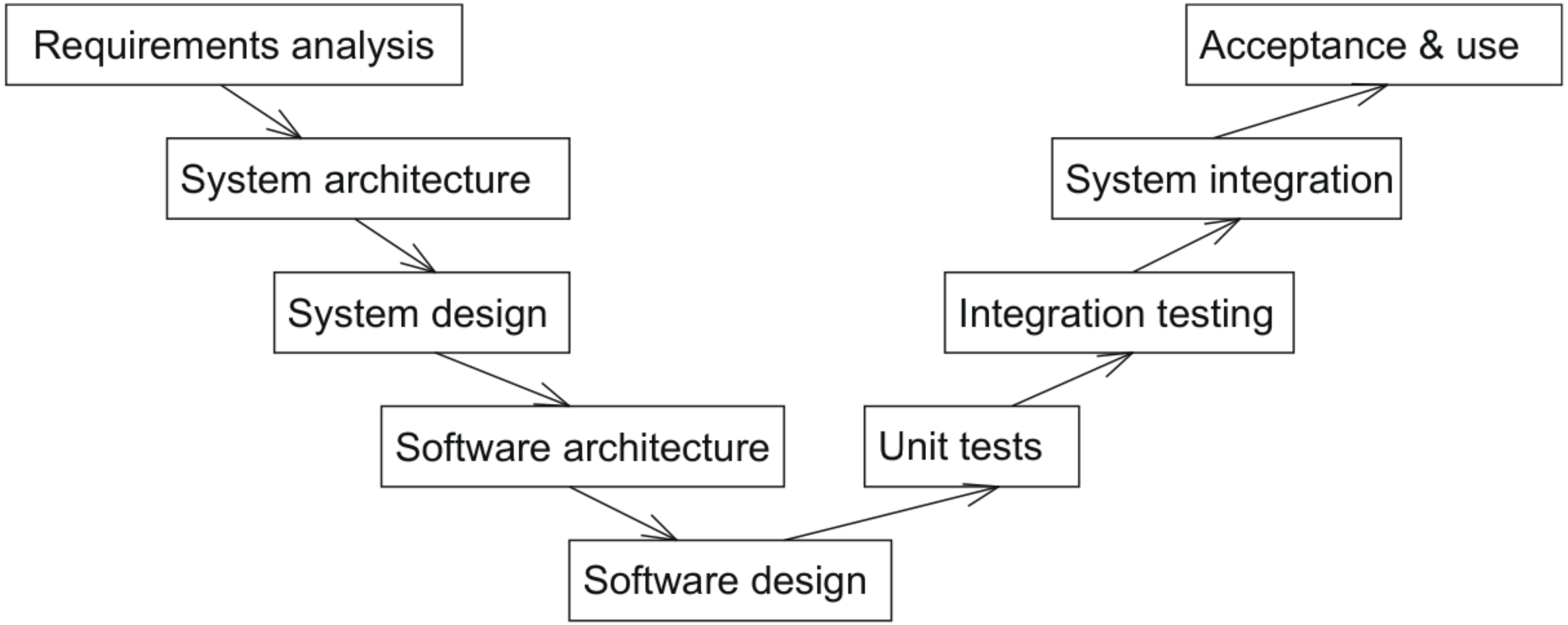
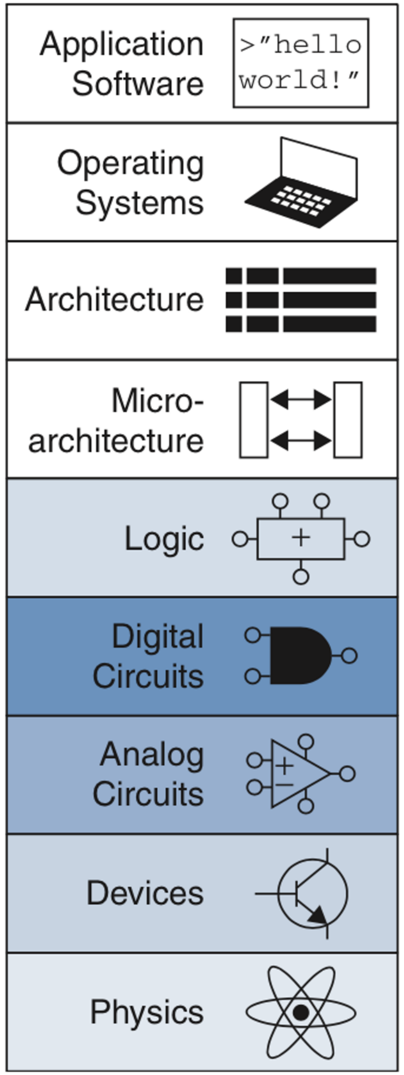


Fig. 1.10 Design flow for the V-model

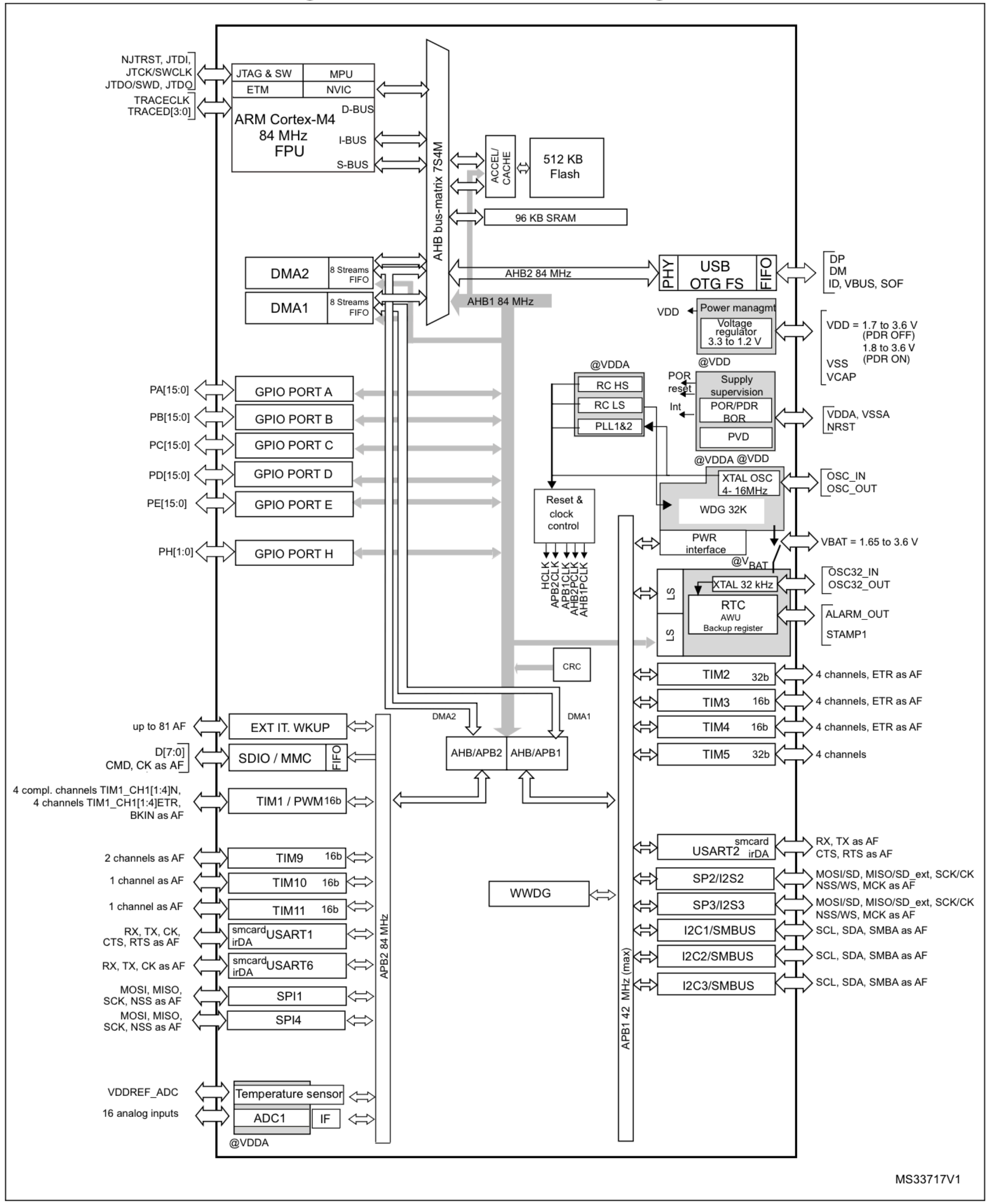
Marwedel, Peter. *Embedded System Design.* Springer, 2018.

MicroPs in Context



From DDCA ARM edition

Figure 3. STM32F401xD/xE block diagram

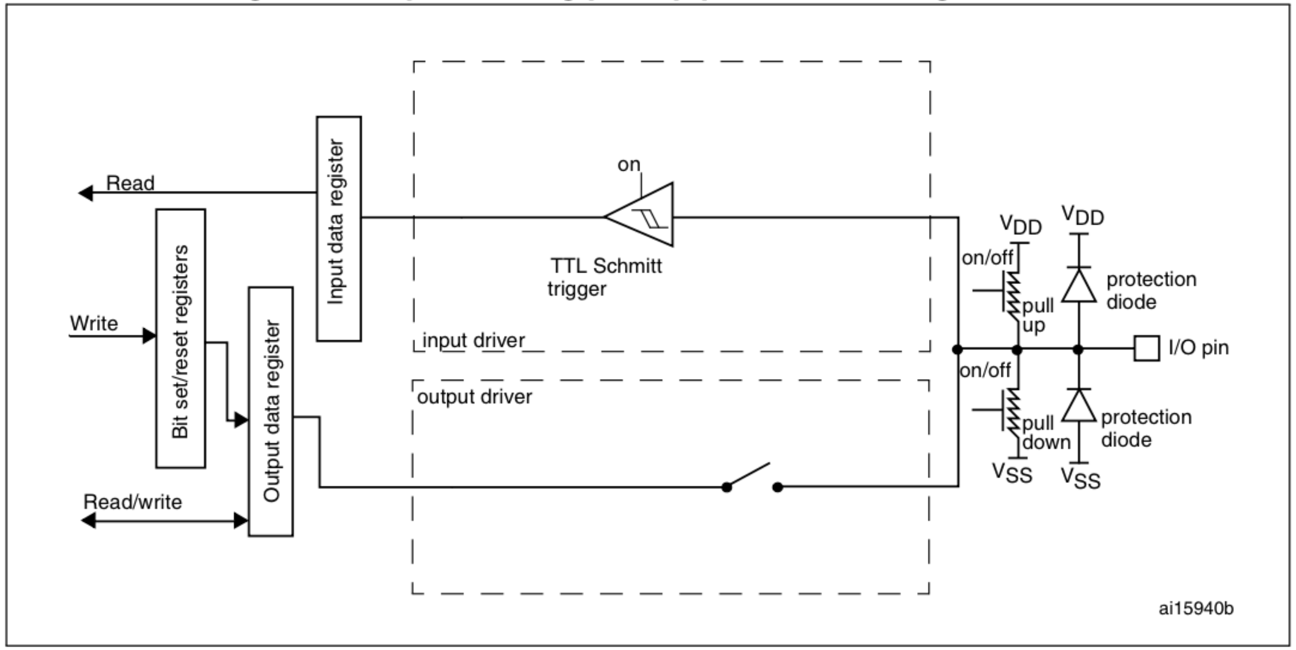


STM32F401RE Datasheet - p.14

MicroPs in Context

- Focus on learning general concepts – specifics are obsolete tomorrow!
- Some core concepts endure
  - Understanding device layout and how to read documentation
  - Memory-mapped I/O
  - Peripheral configuration flow
  - Clock configuration
  - Peripherals
    - UART
    - SPI
    - Timers
    - PWM

Figure 18. Input floating/pull up/pull down configurations



STM32F401RE - RM0368 Reference manual p.154

Survey results

- Reasons for excitement
  - Getting to build project
  - Learn more ”lower level” embedded systems programming
  - Build on E85 knowledge
  - Optimization and processor architectures
- Reasons for concern
  - Troubleshooting – Slack if you get stuck
  - Collaboration and group work – suggest Zoom rooms for when you are working
  - Time management – start labs early; suggest a multi-pass approach
  - Broken kit components
  - Final project – post ideas on Slack and brainstorm with others

For next time…

- Look at lab 1
- Set up software on your computer to make sure everything installs properly
  - Docker
  - OpenOCD
  - VSCode & PlatformIO
- Wednesday lecture - C programming!