Tooling for embedded development

By Malthe Sennels (that's me)

AALBORG UNIVERSITY
DENMARK

AAU SATLAB

The plan for the evening

- Arduino IDE
  - Whats happening underneath
    - Compiling, Linking, bitstream
  - Bootloaders and what is needed to create a devboard
- PlatformIO
  - Why even bother?
  - How i learned to stop worrying & love Vscode
  - Creating libraries
  - Debugging
    - Examples with nucleo boards

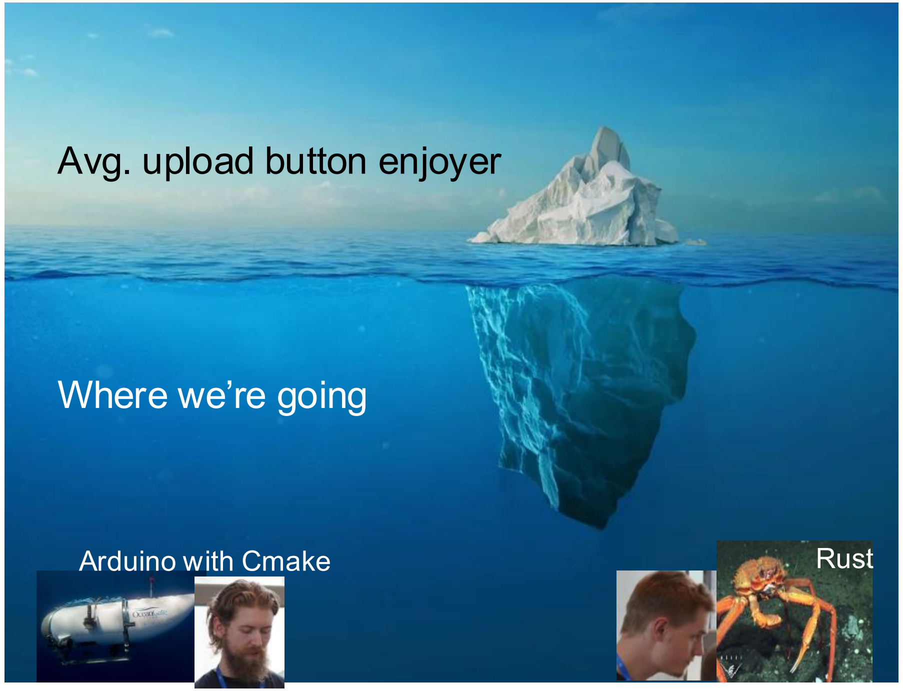
Prerequisits

- You know what an Arduino is (Important)
- You know that Arduino IDE has an upload button

AAU SATLAB

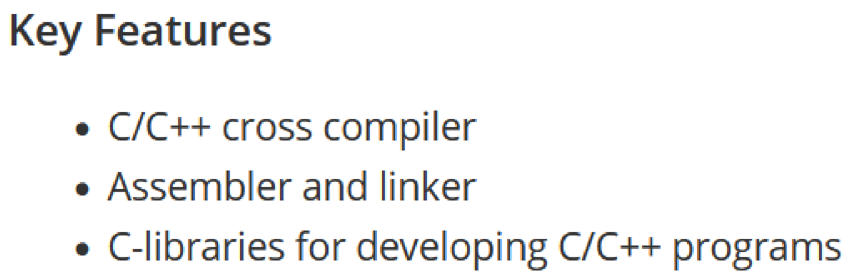
Whats actually happening

- Preprocessor
  - ”converts” into c++
    - Adds Arduino.h
    - Adds function prototypes
- AVR-GCC Toolchain
  - Compiling
  - Linking
  - Creating a binary file
- AVRdude
  - Serializing the binary via UART
- Bootloader
  - Reconstructing the binary in flash



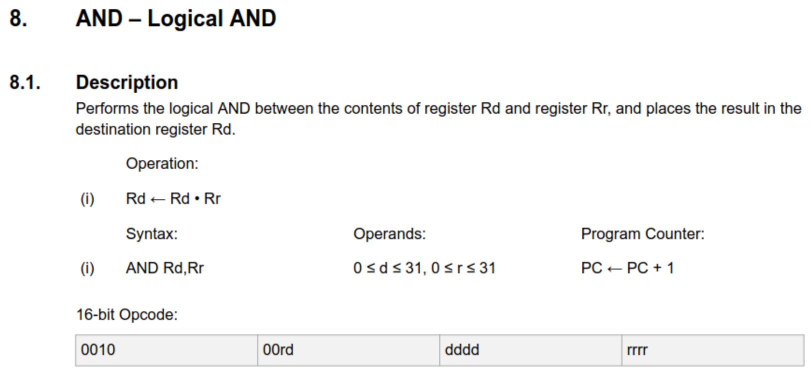
AVR-GCC compiler

- AVR-GCC 4.3>
  - Maybe newer
- Developed by atmel
  - Used in Atmel studio



Key Features

- C/C++ cross compiler
- Assembler and linker
- C-libraries for developing C/C++ programs



8. AND – Logical AND

8.1. Description

Performs the logical AND between the contents of register Rd and register Rr, and places the result in the destination register Rd.

Operation:

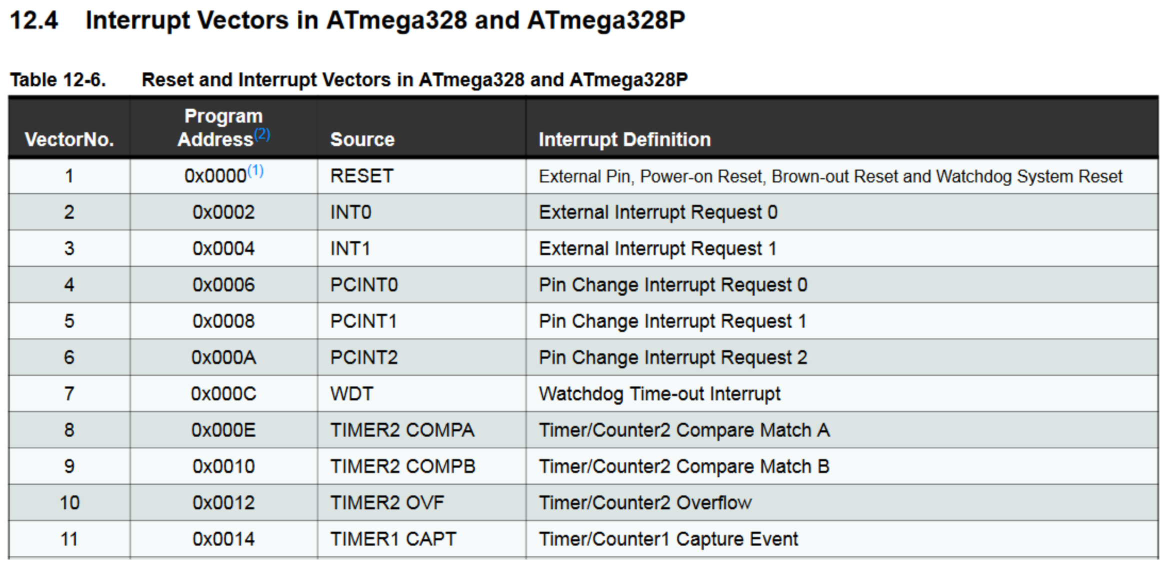
(i) Rd ← Rd • Rr

	Syntax:	Operands:	Program Counter:
(i)	AND Rd,Rr	0 ≤ d ≤ 31, 0 ≤ r ≤ 31	PC ← PC + 1

16-bit Opcode:

0010	00rd	dddd	rrrr

The binary file



12.4 Interrupt Vectors in ATmega328 and ATmega328P

Table 12-6. Reset and Interrupt Vectors in ATmega328 and ATmega328P

VectorNo.	Program Address[2]	Source	Interrupt Definition
1	0x0000[1]	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 1
4	0x0006	PCINT0	Pin Change Interrupt Request 0
5	0x0008	PCINT1	Pin Change Interrupt Request 1
6	0x000A	PCINT2	Pin Change Interrupt Request 2
7	0x000C	WDT	Watchdog Time-out Interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x0012	TIMER2 OVF	Timer/Counter2 Overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 Capture Event

AAU SATLAB

AVRdude

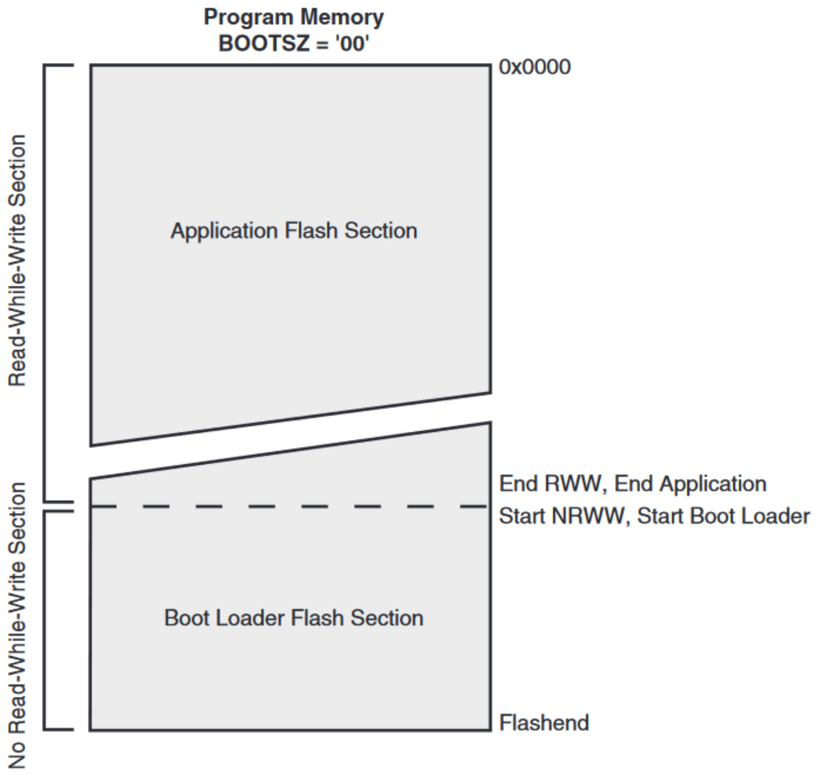
AVR Downloader Uploader

- The tool to get our binary file from the PC to the board of choice
- Stk500v2/stk500
  - https://github.com/Optiboot/optiboot/wiki/HowOptibootWorks

https://github.com/avrdudes/avrdude

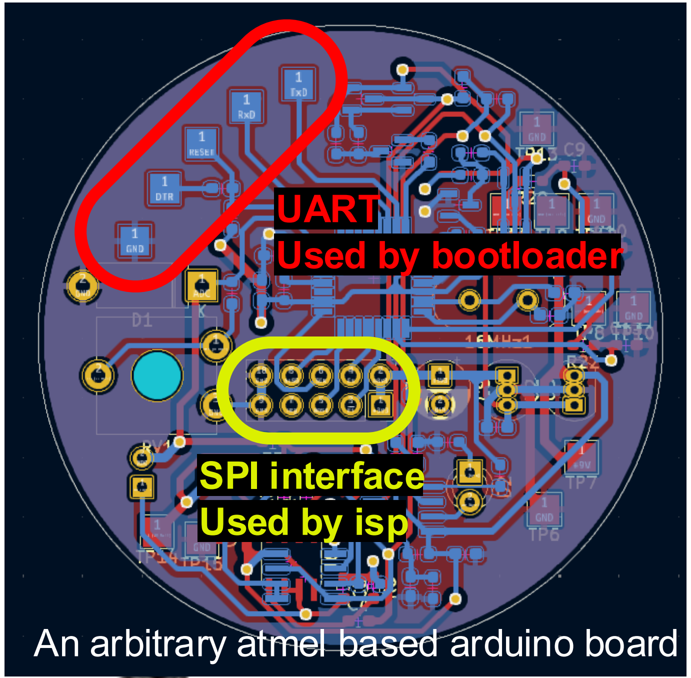
What about the processor?

- Does a μC work right out the box
  - No :(typically for arduino it needs a bootloader.
- Sometimes we're lucky
  - ESP32, ESP8266, STM32…

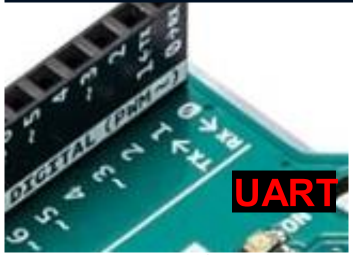

Program Memory
BOOTSZ = '00'
0x0000
Read-While-Write Section
Application Flash Section
No Read-While-Write Section
End RWW, End Application
Start NRWW, Start Boot Loader
Boot Loader Flash Section
Flashend

Bootloader by examples

- You want to build your own arduinoboard
  - Example ATmega168p
- You pick a bootloader you like (And works for your μC)
- You cry for 3 days while getting it to work
- https://github.com/Optiboot/optiboot
- Why UART?
  - https://github.com/mihaigalos/miniboot



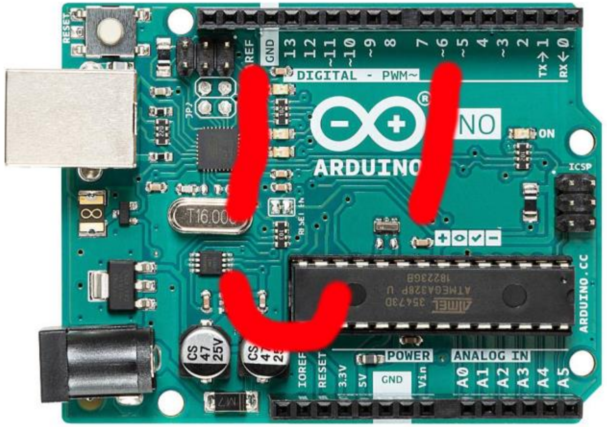
An arbitrary atmel based arduino board



AAU SATLAB

Why we love Arduino IDE

- Easy to learn – Easy to master
- Different boards work
- Fast setup and making prototypes
  - Compared to: STM32CubeIDE, Atmel-studio…

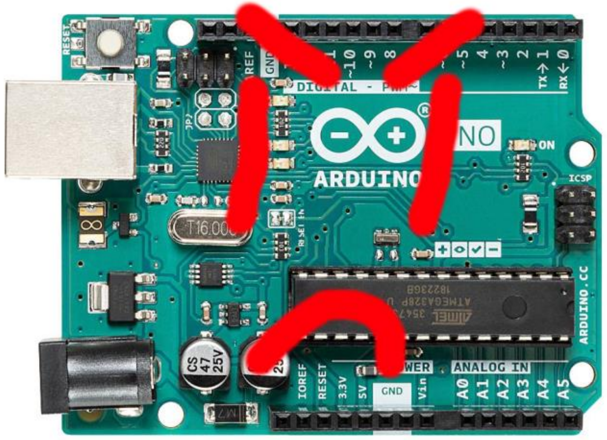


hate

Why we ~~love~~ Arduino IDE

- Usually one page with a lot of (messy) code
- We're locked by the button
- What about projects with multiple arduinos in different configurations
- Custom boards…
- Libraries are kinda 'sketchy'

- Its good for smaller projects
  - But we can do better than that 😎

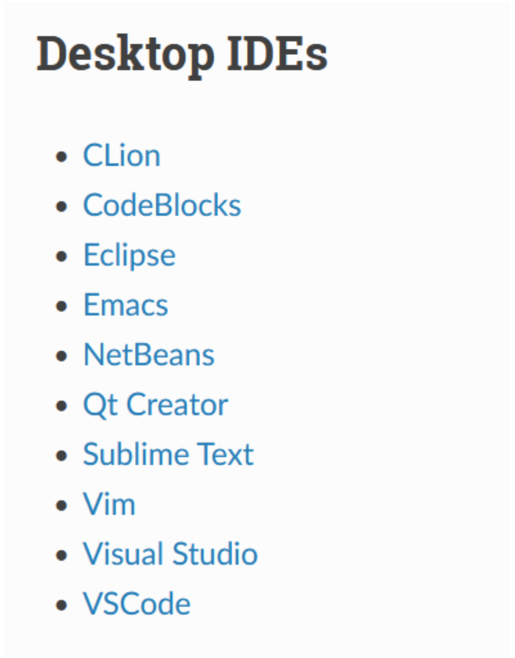


AAU SATLAB

PlatformIO

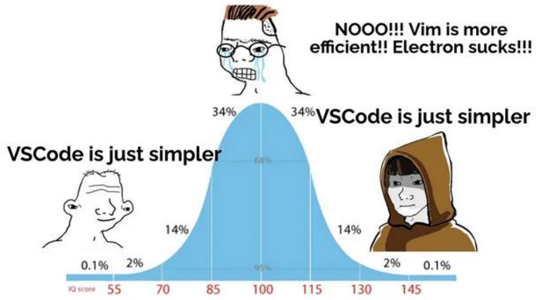
- Remote flashing (& local OTA)
- Arduino is just a standard library now
  - And we can see how it works
- Very easy board handling and libraries
- **VScode**
  - Jumping files
  - LSP

- For the connoisseurs (experienced)
  - Unit testing
  - Support for multiple in circuit debuggers
  - Simulated μC'ers



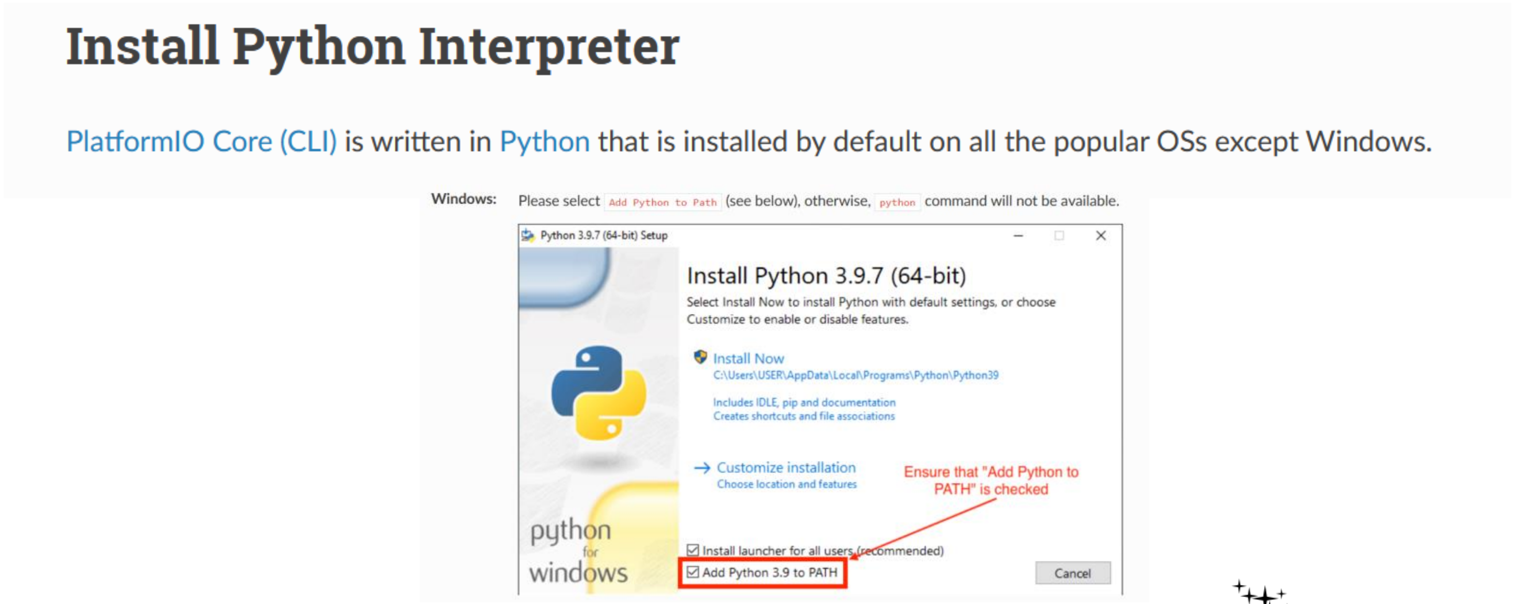
Desktop IDEs

- CLion
- CodeBlocks
- Eclipse
- Emacs
- NetBeans
- Qt Creator
- Sublime Text
- Vim
- Visual Studio
- VSCode



Install party

- Extensions Vscode – Remeber to install python and add to path



Install Python Interpreter

PlatformIO Core (CLI) is written in Python that is installed by default on all the popular OSs except Windows.

Windows: Please select `Add Python to Path` (see below), otherwise, `python` command will not be available.

Python 3.9.7 (64-bit) Setup
Install Python 3.9.7 (64-bit)
Select Install Now to install Python with default settings, or choose Customize to enable or disable features.
Install Now
C:\Users\USER\AppData\Local\Programs\Python\Python39
Includes IDLE, pip and documentation
Creates shortcuts and file associations
Customize installation
Choose location and features
Ensure that "Add Python to PATH" is checked
☑ Install launcher for all users (recommended)
☑ Add Python 3.9 to PATH
Cancel
python for windows

AAU SATLAB

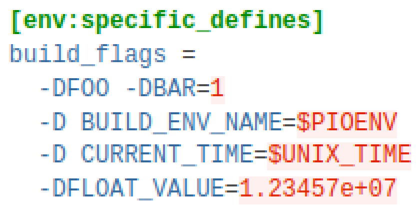
PlatformIO by examples

- A small cute 2.sem project
- Libraries
  - https://docs.platformio.org/en/latest/librarymanager/index.html
  - Internal package manager – Like Arduino IDE
  - Github links
  - Your own libraries

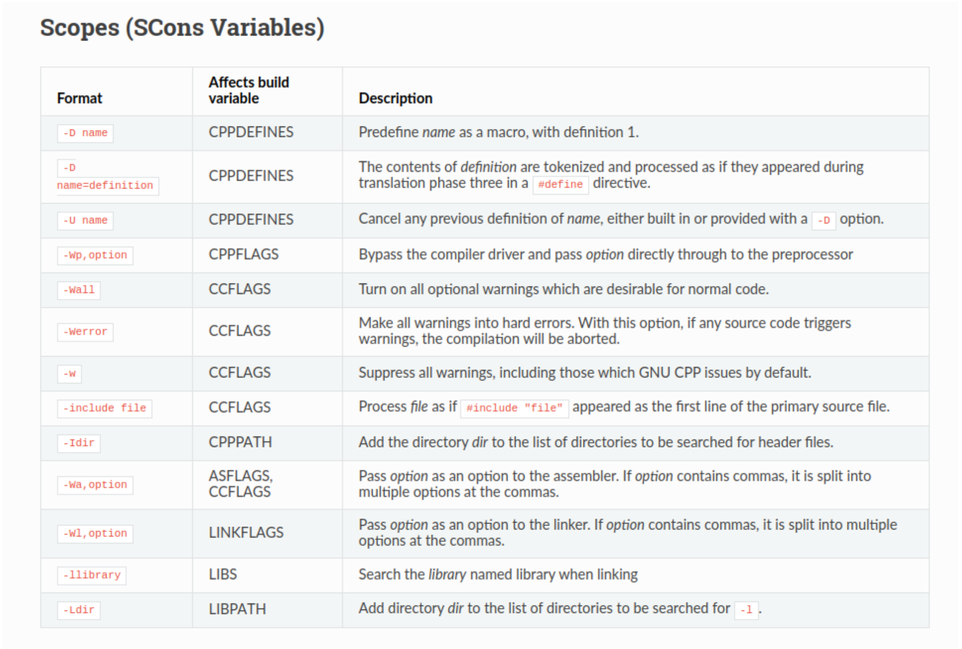
PlatformIO.ini

- A way to setup projects

- [env:some_name]
- upload_port
- monitor_port
- Build flags
- Mcu.cpu
- Fuses
  - `board_fuses.hfuse = 0xDA`
  - `board_fuses.lfuse = 0xFF`
  - `board_fuses.efuse = 0xFD`



```
[env:specific_defines]
build_flags =
  -DFOO -DBAR=1
  -D BUILD_ENV_NAME=$PIOENV
  -D CURRENT_TIME=$UNIX_TIME
  -DFLOAT_VALUE=1.23457e+07
```



Scopes (SCons Variables)

Format	Affects build variable	Description
`-D name`	CPPDEFINES	Predefine *name* as a macro, with definition 1.
`-D name=definition`	CPPDEFINES	The contents of *definition* are tokenized and processed as if they appeared during translation phase three in a `#define` directive.
`-U name`	CPPDEFINES	Cancel any previous definition of *name*, either built in or provided with a `-D` option.
`-Wp,option`	CPPFLAGS	Bypass the compiler driver and pass *option* directly through to the preprocessor
`-Wall`	CCFLAGS	Turn on all optional warnings which are desirable for normal code.
`-Werror`	CCFLAGS	Make all warnings into hard errors. With this option, if any source code triggers warnings, the compilation will be aborted.
`-w`	CCFLAGS	Suppress all warnings, including those which GNU CPP issues by default.
`-include file`	CCFLAGS	Process *file* as if `#include "file"` appeared as the first line of the primary source file.
`-Idir`	CPPPATH	Add the directory *dir* to the list of directories to be searched for header files.
`-Wa,option`	ASFLAGS, CCFLAGS	Pass *option* as an option to the assembler. If *option* contains commas, it is split into multiple options at the commas.
`-Wl,option`	LINKFLAGS	Pass *option* as an option to the linker. If *option* contains commas, it is split into multiple options at the commas.
`-llibrary`	LIBS	Search the *library* named library when linking
`-Ldir`	LIBPATH	Add directory *dir* to the list of directories to be searched for `-l`.

Exercise: blinky

”Its not a satlab presentation without some questionable exercises”

- We’re working with the nucleo

- What is needed
  - Framework: Arduino
  - Board: STM32F401re

Creating a library

- Header files
  - function prototype
  - Classes
- cpp files
  - Actual code working
- G-wagon analogy
  - Interfaces
- Example

Creating libraries

- Prototypes can be declared multiple times
- Definitions can only be declared once
- Include guards

Exercise: creating a library

- <your library name> (folder)
  - src (folder)
    - <your library name>.h
    - <your library name>.cpp

- Find a fun functionality to implement

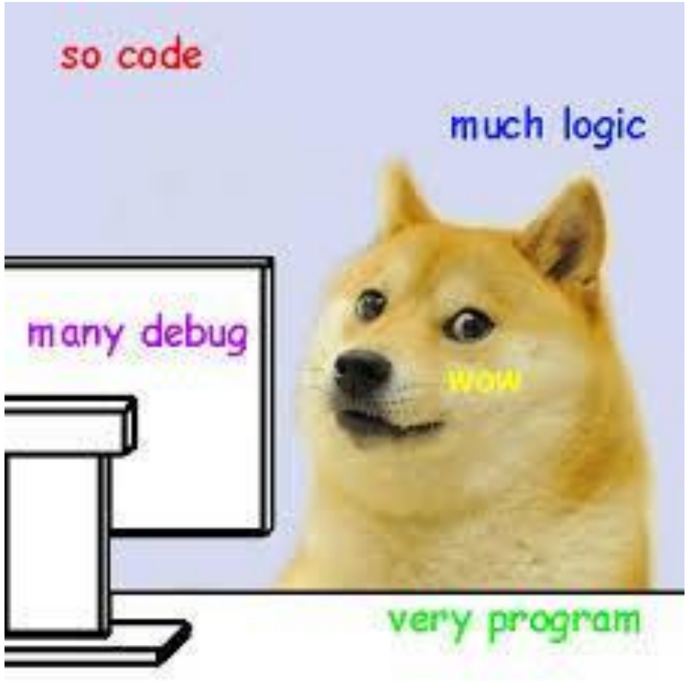
- If you have extra time add JDN-krnl from github and look around

Remote IO

- Development from away
  - Good for when you visit your parents and still want to work on semester-projects
  - Bad for your social life
- Supported
  - Local OTA like ESP-OTA
  - Remote IO broker – Flash and use serial monitor from everywhere

Extra VScode spice

- Serial monitor
  - Arduplot
  - Custom filters using python
  - ESP-32/8266 dumps
- Comments
- Autocompletion
  - For, if, while, switch
- LSP, on the fly syntax checking



AAU SATLAB

Virtual & physical debuggers for Arduino

- Quick recap
  - A debugger lets us step through one line of code seeing every variable in the system at a time.
- Ever just wanted to be able to see inside the arduino.
  - https://github.com/buserror/simavr
  - "A lean and mean Atmel AVR simulator for linux”

- Also real debuggers
    - Example: STM32 ST-Link onboard

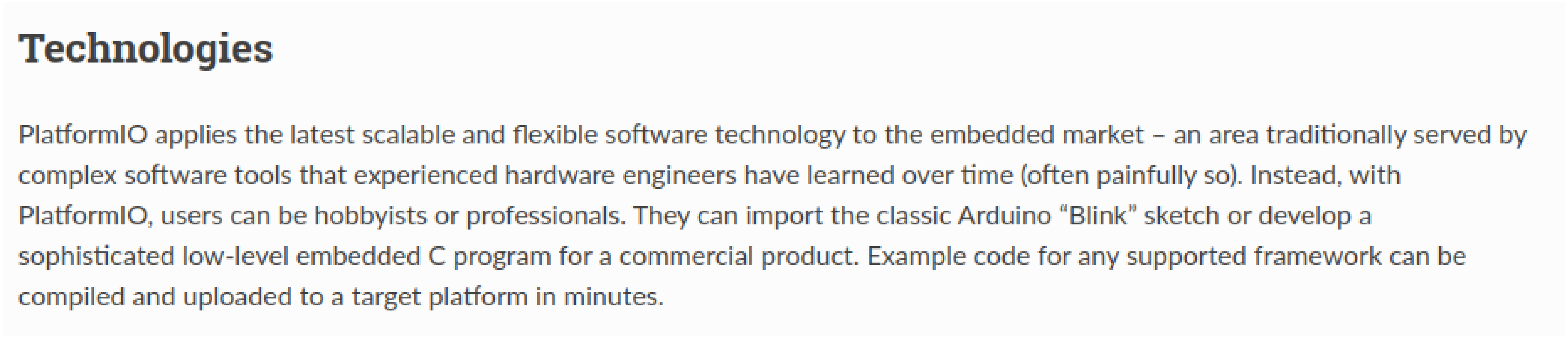
Exercise: Debugging!!

- We love debugging - ST-LINK

- Create a project (or use one from earlier)
  - In platformIO.ini
    - Set upload_port and debug_port

- https://os.mbed.com/teams/ST/wiki/ST-Link-Driver

- Hint: https://docs.platformio.org/en/latest/boards/ststm32/nucleo_f401re.html
  - kortlink.dk/2r49x



Technologies

PlatformIO applies the latest scalable and flexible software technology to the embedded market – an area traditionally served by complex software tools that experienced hardware engineers have learned over time (often painfully so). Instead, with PlatformIO, users can be hobbyists or professionals. They can import the classic Arduino "Blink" sketch or develop a sophisticated low-level embedded C program for a commercial product. Example code for any supported framework can be compiled and uploaded to a target platform in minutes.