Introduction to Microcontrollers

Zach Metzinger (N0ZGO) | Version 2020-01

Background

- What is a microcontroller?
  - CPU
  - Memory
    - Flash (Program)
    - SRAM (Data)
  - I/O
    - GPIO
    - UART (Async. Serial), SPI, I2C
    - USB
  - Timers, DMA, and lots more!

+

+

=

Background

- What do we mean by “Embedded”?
  - Limited resources
    - Program and/or data capacity
    - Smaller set of appropriate programming languages
  - Specialized processor architectures
    - Optimized for power/cost
      - 8051, M68K, PIC, H8, ARM, etc.
  - Non-traditional I/O
    - Interfaces to the real world – sensors, relays, radios
  - Real-time constraints
    - Precisely-timed events

Examples of Embedded Platforms

Arduino / mbed

- “Bare Metal” programming
  - ARM Cortex-M or similar
- Advantages
  - Nearly instantaneous “boot”
  - Can be very low power
  - Fast, deterministic I/O
- Disadvantages
  - Generally no file system
  - More difficult to network

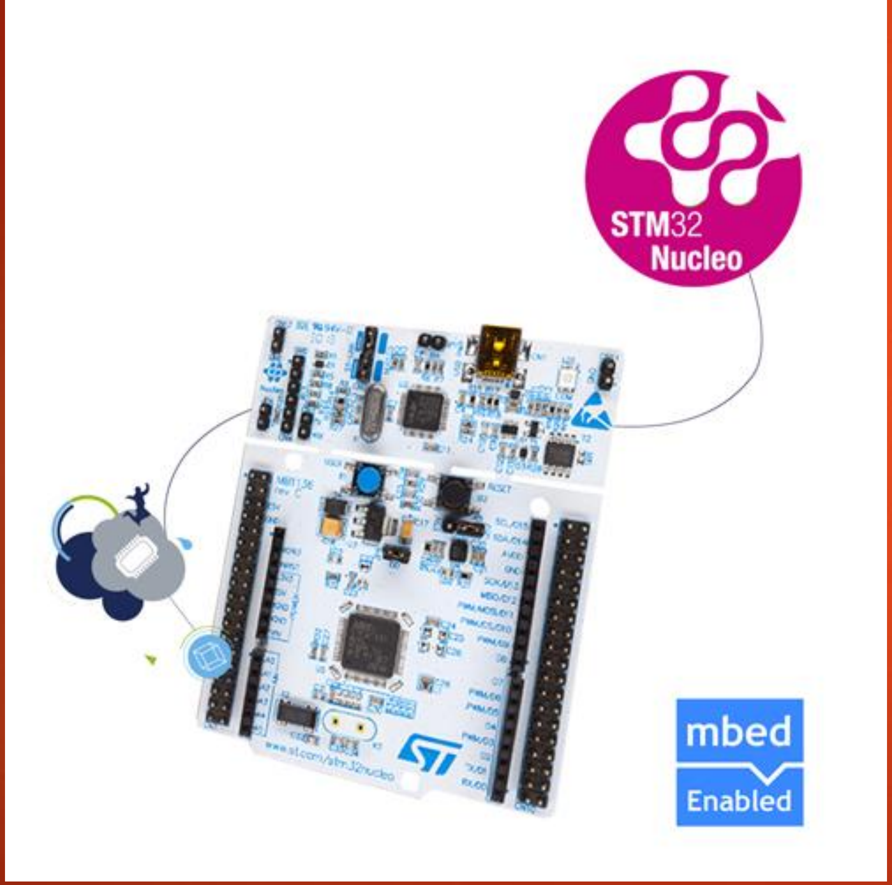
Raspberry Pi / Beaglebone

- Multi-user Linux OS
  - ARM Cortex-A or similar
- Advantages
  - Networking & file system built-in
  - Can leverage a lot of FOSS
- Disadvantages
  - Slow to “boot”
  - More difficult to develop/debug
  - Slower I/O and more jitter on I/O

Focus: ARM® Cortex™

- Advantages
  - Widely used in embedded systems
    - Mobile phones, wearable devices, vehicles
  - Free and commercial tools
    - GCC, IAR, Keil (now ARM), etc.
  - Dense instruction encoding
  - Many optimized variants
    - High performance – Cortex-A
    - Low power – Cortex-M
    - Real-time – Cortex-R

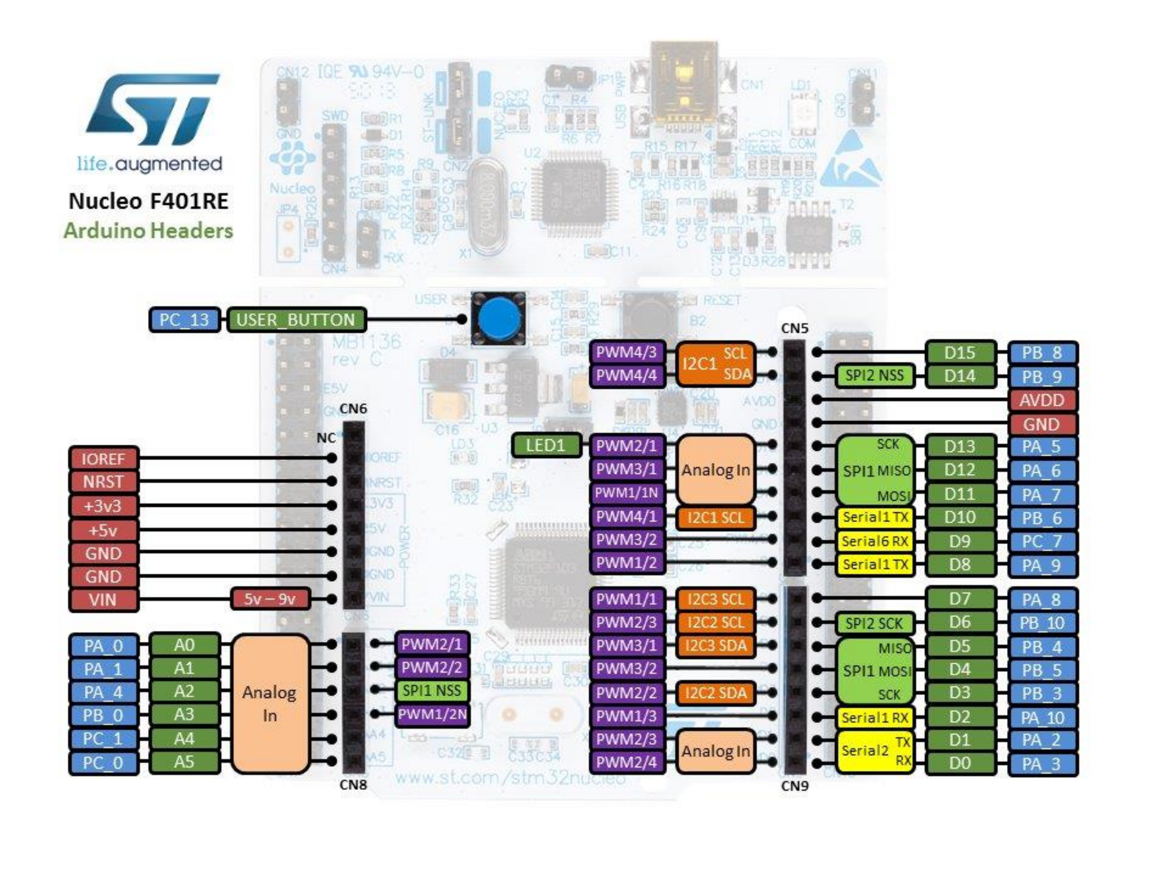
ARM® mbed™



- mbed ecosystem
  - Standard SDK based on C++
  - On-line IDE and compiler
  - Programs stored "in the cloud"
  - Local development via USB
    - Allows programming and debugging through one cable
- Many vendors have mbed boards
  - Code is generally portable between boards

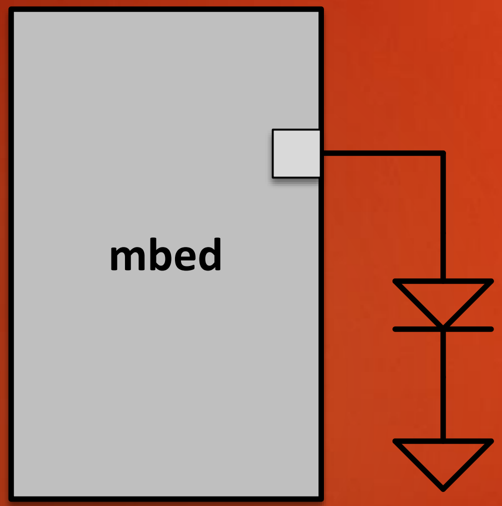
Microcontroller I/O

- The mbed board has multiple I/O pins
  - Most are digital
  - Some are analog
- Collectively called “port pins”
- Direct interfacing not always possible
  - LEDs - Current-limiting resistor
  - Motors
    - Driver transistor or H-bridge

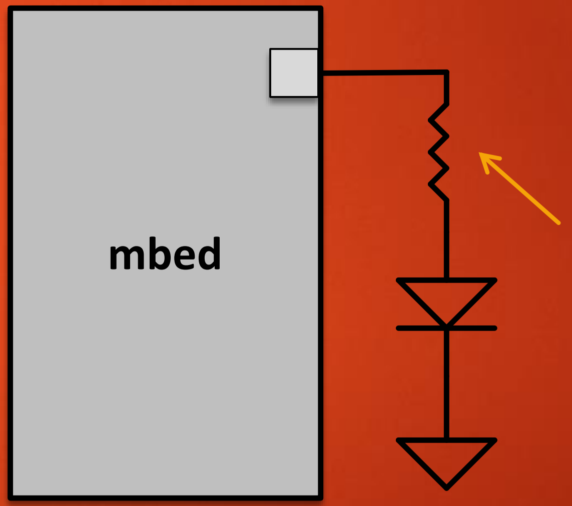


Connecting LEDs to I/O pins

- Incorrect



- Correct

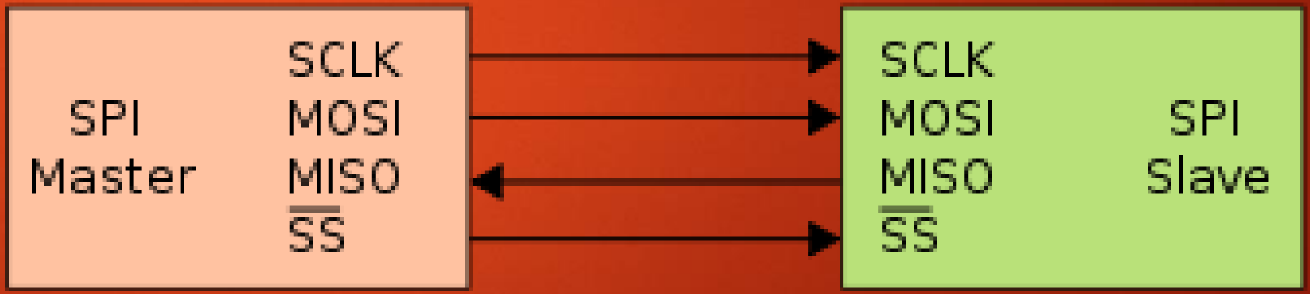


Current-limiting resistor

$R = (V_{IO} - V_{LED}) / I_{LED}$

SPI

- High-speed interface
  - 20+ Mbit/sec common
  - Displays, radios, data storage, etc.
- Point-to-point
- Four wires (usually)
  - MOSI
  - MISO
  - SCLK
  - /SS

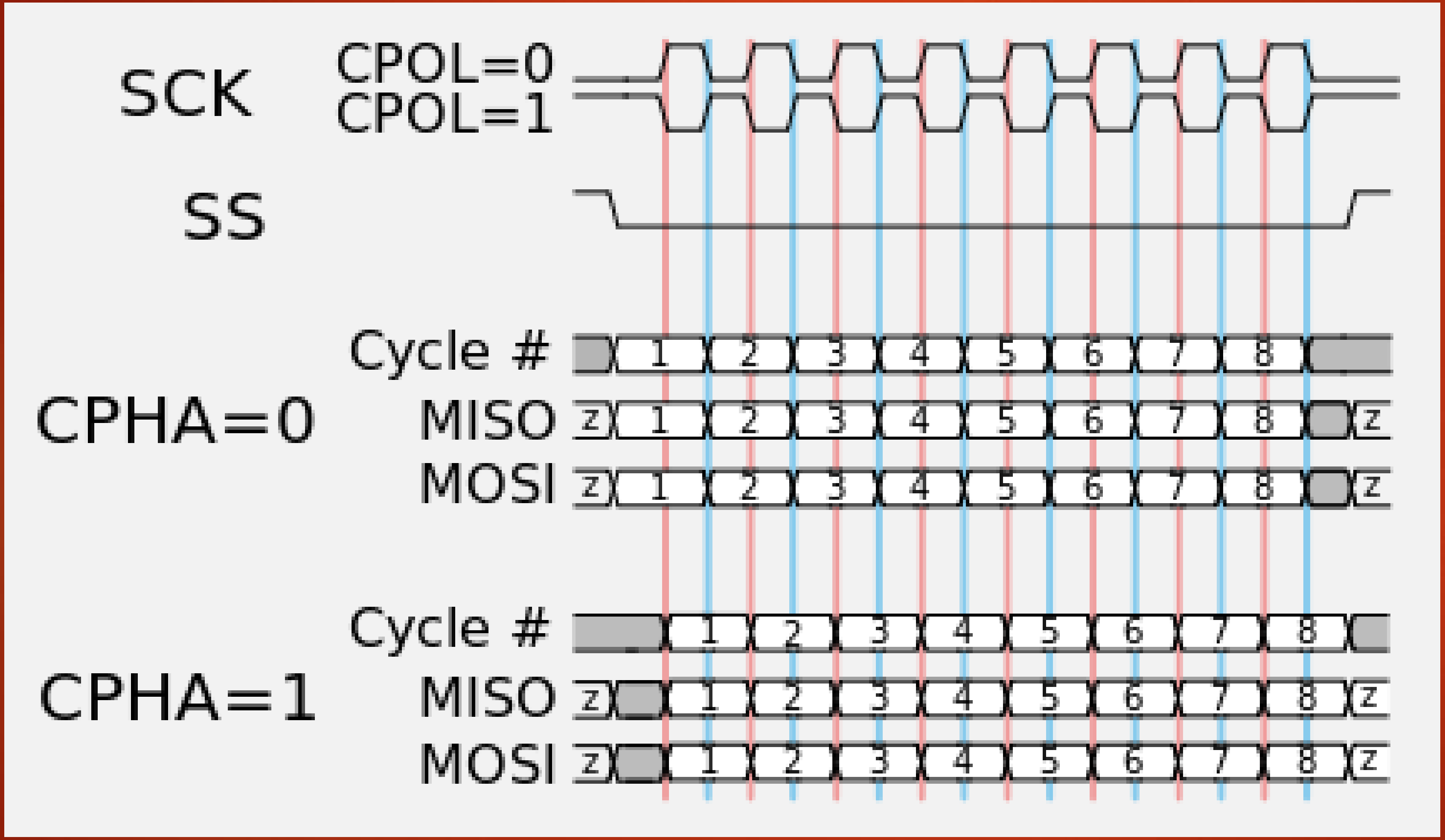


https://commons.wikimedia.org/wiki/File:SPI_single_slave.svg

SPI (cont.)

- Transaction sequence varies by device
- Generally follows a pattern – Mode 0
  - Master asserts Slave Select (/SS)
  - Master puts data on MOSI
  - Slave puts data on MISO
  - Master raises clock
  - Slave captures data on MOSI, Master captures data on MISO
  - Master lowers clock
  - Process repeats until all data is transferred
  - Master de-asserts Slave Select (/SS)

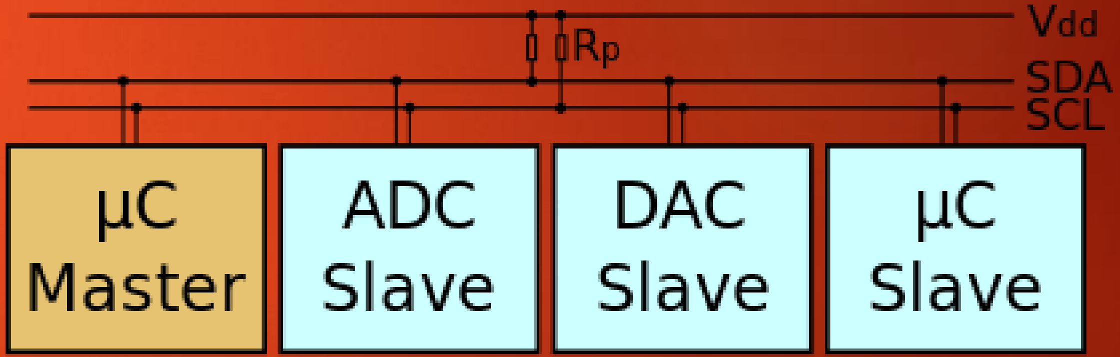
SPI (cont.)



https://commons.wikimedia.org/wiki/File:SPI_timing_diagram2.svg

I2C

- Lower-speed interface
  - 100 kHz, 400 kHz, & (rarely) 1 MHz clock rates
- Multi-drop bus
  - Each device has an address
- Two wires
  - SDA
  - SCL



https://commons.wikimedia.org/wiki/File:I2C.svg

Programming

Programming Basics

- Computers do simple operations
  - Hold numbers
    - Called “variables”
  - Perform math on numbers
    - A + B, A - B, A x B, A / B, A | B, A & B, etc.
  - Compare two or more numbers
  - Decide to continue one way or the other based on results
    - Called “conditional operations”
  - Write numbers somewhere
    - Can be used to save results of computations
    - May also be used for I/O to the real world

Programming Basics (cont.)

- You already know how to program!
  - “If I have $5 in my wallet, then I will go see a movie. Otherwise, I will stay home.”
  - if (wallet.dollars >= 5) { movie.attend(); } else { home.stay(); }
- Computer program
  - Sequence of simple operations
  - Takes input
  - Computes output
  - Terminates or starts again at “takes input”

Making Programming Easier

- Beginners should use libraries
  - A library is code which is written for you
  - Usually comes with examples
- mbed provides Software Development Kit (SDK)
  - The SDK provides many Class objects to interface to the real world
  - CLASSES! Oh no! This sounds like C++!
    - (It is.. but only the good parts)
- SDK Example programs
  - Learn by emulating what others do

mbed SDK Classes (partial)

Class Name	Description
AnalogIn AnalogOut	Read (ADC) a pin's voltage, or set (DAC) a pin's voltage
DigitalIn DigitalOut	Read or set a digital pin's level
PwmOut	Generate Pulse Width Modulation (PWM) outputs
Serial	Send or receive asynchronous serial data
Timer	Timers can be used to delay, measure, or count some event
SPI I2C	Communicate with Serial Peripheral Interface (SPI) or Inter-IC Communications (I2C) Bus

Integrated Development Environments (IDE)

- Benefits
  - Permit easy editing of code
  - Provide language and/or SDK references
  - Usually include debugging capability
- mbed has a HTML5-based IDE
  - Write your code in a web browser window
  - Compile code into binary file
  - Download binary file to local PC
  - Program your board and test it

Downloading (Programming)

- Plug in mbed board to PC
  - Should appear as 3 devices in “Device Manager”
    - CMSIS-DAP
    - Mass Storage (Disk)
    - Serial Port
- Drag ‘n drop binary file to mbed “disk”
  - Automatically loads code into microcontroller flash
- Non-volatile
  - Until explicitly erased (or reprogrammed)

Further Resource

- https://www.mbed.org
  - Documentation, hardware, and IDE for mbed
- https://www.learn-c.org/
  - Never learned C? Here's an easy site to start
- https://webpages.uncc.edu/~jmconrad/ECGR4101-2015-08/notes.html
  - Very in-depth microcontroller course with YouTube video lectures

Questions?