Block 01

Course intro, intro to PIC, development environment

2023

EmLab @ FI MUNI

Financováno Evropskou unií
NextGenerationEU

Národní plán obnovy

MŠMT
MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY

Seminar on Digital System Architecture

Course organization

- course intro (this week)
- MCU architecture, instruction sets, GPIO
- intro to C in MCUs, advanced GPIO
- interrupts and timers
- free week for you (I'll be abroad)

Grading

- ends in colloquium
- mandatory homework (not for each lesson)
- graded OK/NOK, deadline usually to the next week
- you need all homeworks graded OK to pass
    - there will be possibility of correction at the end of the semester
- final project in last two weeks + exams period
    - can choose your own idea
    - a final report and defense will be required
    - you can either use our kit or create your own PCB
    - if you go for latter, you can have it manufactured

The PIC microcontroller

PIC18 microcontroller

- family of higher end 8bit MCUs
- based on the PIC core
  - PIC started as Peripheral Interface Controller for a larger CPU
  - later changed to Programmable Intelligent Computer
  - a lot of weird quirks because of this
- pipelined, RISC CPU with 75 instructions
- 16bit instructions with 8bit data path
- (modified) Harvard architecture
- we will work with PIC18F44K22 specifically

MCU vs MPU

Microcontroller

- all components in one chip
- integrated peripherals
- fixed memory and I/O
- reduced complexity, low cost
- critical applications
- small size

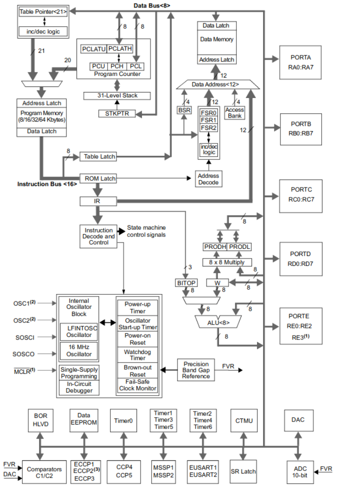
Microprocessor

- only the processor
- peripherals via external bus
- modular memory and I/O
- complex and high cost
- high performance
- large size

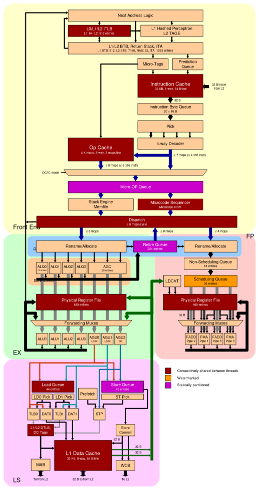
Nowadays both can be used for general purpose applications.

MCU vs MPU

PIC18 diagram (source: PIC18F44K22 datasheet)



AMD Zen2 core diagram (source: WikiChips)



Why 8bit MCU today

- simpler than modern 32bit MCUs
- easier to learn low level on
- often still powerful enough and cheaper
- extremely low level power consumption

Alternatives

- multiple vendors provide 8bit MCUs
- Microchip - PIC line, 8051 line
- Atmel - AVR line, now under Microchip
- STM - STM8 line
- and many more smaller vendors
- nowadays often a small ARM Cortex-M0 replaces 8bit MCUs

The datasheet

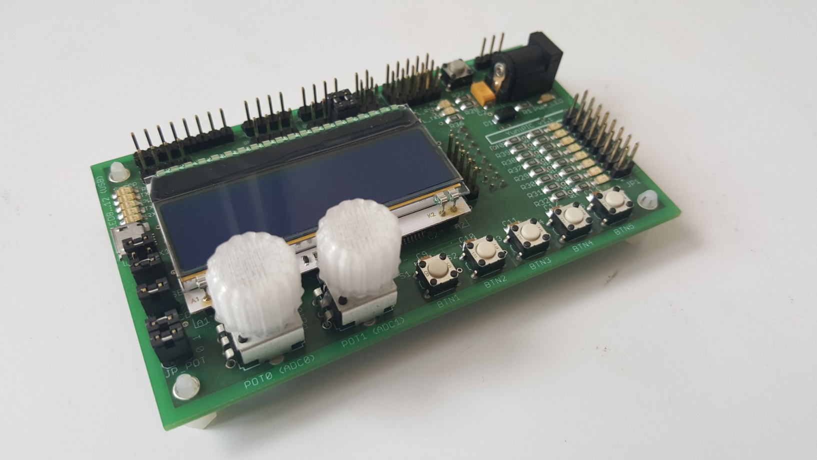
- the most important document
- we need to learn how to efficiently use it
- contains all the important info
  - electrical, mechanical specifications
  - peripherals description
  - memory and register map
  - instruction set
- you can find the datasheet here or in study materials
- you will spend a lot of time reading it
- there are also application notes and other technical documentation

Our dev board

Yunipic

- our custom development kit based on PIC18F44K22
- mostly a simple breakout
- includes the necessary power circuitry
- simple character based display
- USB $\longleftrightarrow$ UART bridge
- buttons, LEDs, potentiometers...
- schematic of the kit can be found in study materials

Yunipic


POT0 (ADC0)
POT1 (ADC1)
BTN1
BTN2
BTN3
BTN4
BTN5
JP_POT
JP1
LED8..12 (USB)

Schematics

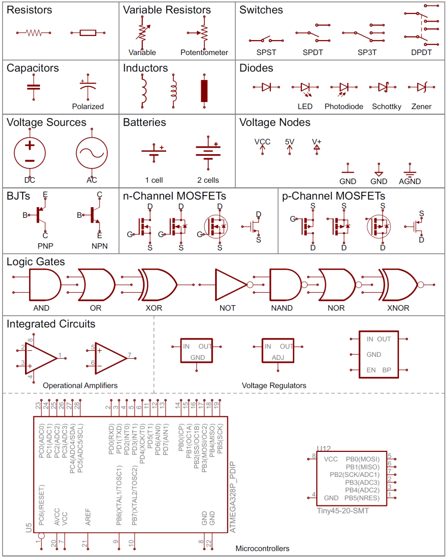
- graphical representation of electrical design
- should include all components and their connections
- also usually include notes

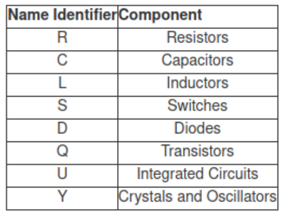
Reading a schematic

- all components have somewhat standardized naming conventions and symbols
- each "wire" is usually called a net and can be labeled (named)
- nets can be grouped into a bus, which can also be labeled
- labeling of certain nets is also standardized, mainly power nets
- there are few "direction" conventions:
    - signals progress left to right - input on left, output on right
    - power/voltage goes top to bottom - positive on top, negative on bottom
- great overview at SparkFun

Common symbols and markings

- can be found online, EU and US standards





Name Identifier	Component
R	Resistors
C	Capacitors
L	Inductors
S	Switches
D	Diodes
Q	Transistors
U	Integrated Circuits
Y	Crystals and Oscillators

Drawing a schematic

- tightly connected to electronics design
- atleast minimal electronics knowledge is needed
- tools for drawing schematics integrated as part of EDA tools
  - Electronic Design Automation - software tools to aid in design of electronics
  - free - EAGLE, KiCAD, EasyEDA
  - paid - Altium Designer, Cadence, Mentor PADS
- to produce a readable schematic, follow guidelines from previous slides

The IDE

MPLAB X IDE

- multiplatform NetBeans based IDE
- integrates the whole toolchain
- plenty other features
  - data visualization
  - call graphs
  - simulation
  - visual configuration
  - code gen

Why IDE

- all tools integrated in one
- less probability of having to deal with toolchain problems
- more time to focus on the actual course
- you are welcome to not use the IDE, but there will be no support
- homework and project has to be importable and compileable in the IDE

Creating a project

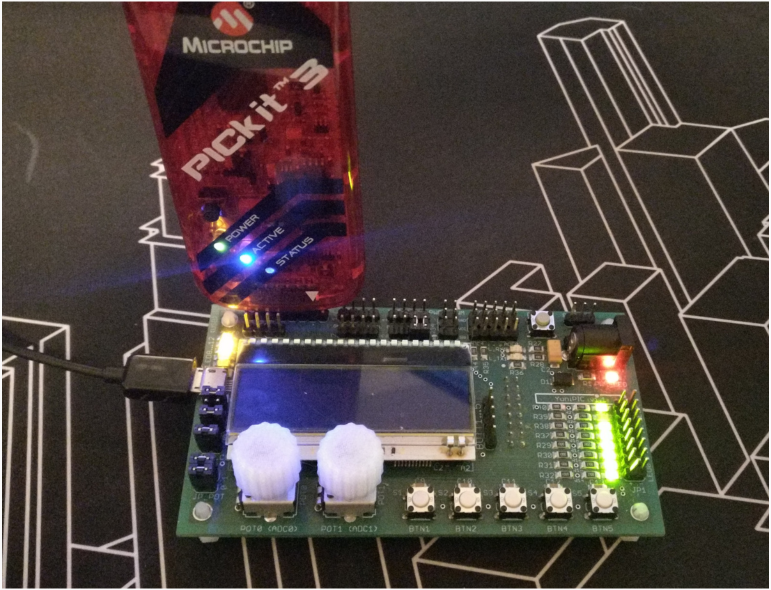
Shown only in seminar.

Uploading the code

- we need a debug probe to connect to the MCU
- we use PICkit 3
  - older generation programmer/debugger
  - supports most PIC devices
  - full support in IDE
- IDE + PICkit → debugging just like on desktop[1]

[1] apart from weird quirks and limitations

Connecting PICKit to Yunipic



Homework

Mandatory

None.

Optional

- compare Yunipic schematic to some other dev kit, eg. Arduino, FRDM-K66F, any STM32 Nucleo board
- try to note the differences in both hardware differences and in the schematic ”look and feel”
- try sketching a a high level schematic of a normal desktop computer