

PV198 - Introduction

One-chip Controllers

**Daniel Dlhopolček, Marek Vrbka, Jan Koniarik, Oldřich Pecák,
Tomáš Rohlínek, Ján Labuda, Jan Horáček, Matúš Škvarla, Ondřej Bleha,
Martin Klimeš, Adam Valt**

Faculty of Informatics, Masaryk University

1/2024

Content

Organization

Course overview

Lectures

Hardware Introduction

Hello World

MCUXpresso software package

- MCUXpresso IDE

- MCUXpresso SDK

- MCUXpresso Config Tools

Homework

Organization

- Course takes place in room A415
- 13 lessons (2 hours)
- Maximum of 2 unexcused absences
- Mandatory homework for every lesson
 - Deadline until next lesson
 - 1/0.5/0 points per homework
 - > 2 points lost = course failure
- Extra study materials:
 - <https://pv198.pages.fi.muni.cz/>
 - Visit the page before the beginning of semester!
- Exam at the end of course
 - Implement homework-difficulty level task
 - Extra open questions (from study materials, not required if you have colloquium level of completion)

Contact

You have multiple options to contact us:

- Never be afraid to ask during your seminars!
- For questions related to course organization:
Jan Labuda (jan.labuda@mail.muni.cz)
- Discussion forum in IS
- As an informal alternative, discord server:
<https://discord.gg/UUyNayBPkk>

Course overview

- We expect you to know the C language
- You will learn how to write code for micro-controllers
- You will work with real hardware

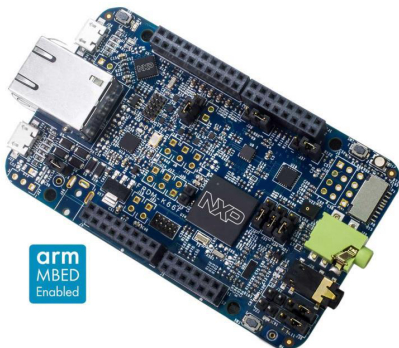
Lectures

- Basic peripherals
 2. GPIO
 3. Interrupts/Timers
 4. PWM
 5. ADC
- Communication interfaces
 6. SPI
 7. I2C
 8. UART I
 9. SPI II
 10. UART II
- Advanced peripherals
 11. LCD
 12. Ethernet

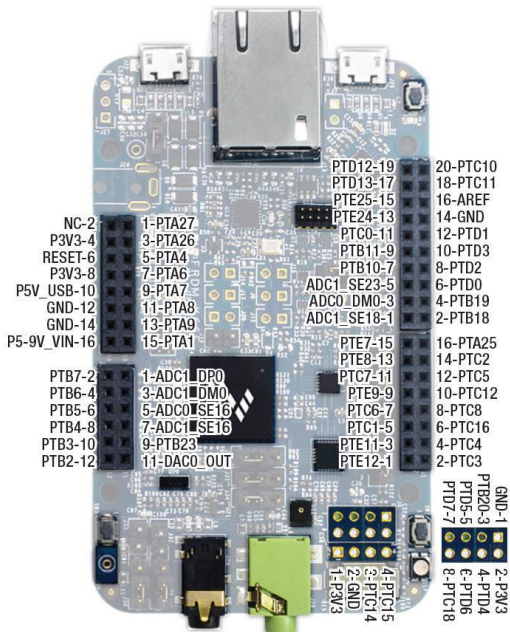
Hardware Introduction

NXP FRDM-K66F Board

- MK66FN2M0VMD18 MCU
- 180 MHz
- 2 MB Flash memory
- 256 kB SRAM
- Accelerometer, Magnetometer
- 3-coloured LED
- 2 push buttons
- USB
- Ethernet
- Audio
- ... and more



Hardware Introduction



Links

- [Product page](#): Main source of information
- [Additional information](#): Detailed information
- [Get started](#): Step-by-step guide
- [Board user guide](#): Information about the board and its peripherals
- [K66 Reference manual](#): Features and register description
- [K66 data sheet](#): Technical data about the MCU itself
- [Board schematics](#)

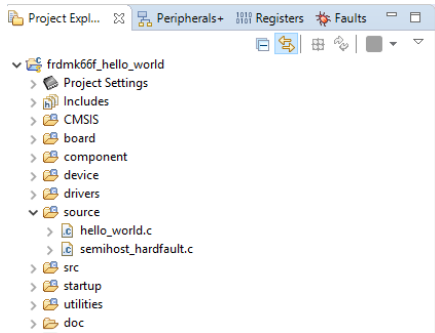
Hello World

Task 1

1. Open IDE
2. Setup SDK
3. Import "Hello World" example project

Hello World

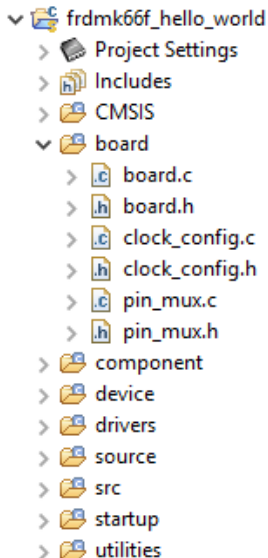
Project content:



Board folder

Folder *project/board*:

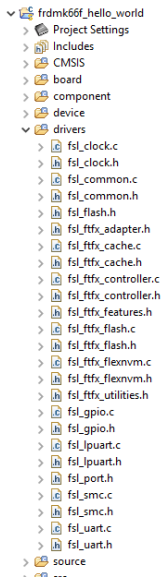
- Files generated by MCUXPresso Config Tools
 - clock_config
 - pin_mux
 - peripherals
- Board definitions
 - LEDs
 - buttons
 - debug
 - etc.



Drivers folder

Folder *project/drivers*:

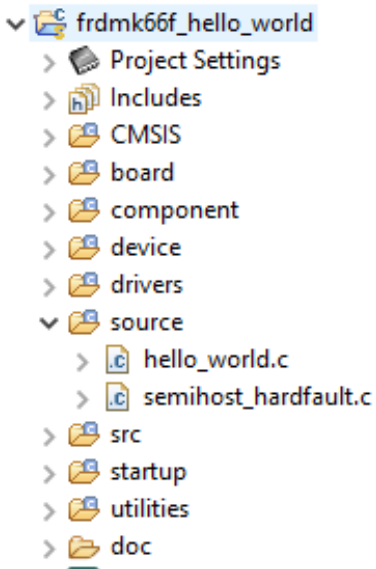
- SDK files
- Peripheral control API



Source folder

Folder *project/source*:

- Your application code
- Main function



MCUXpresso



MCUXpresso SDK

An open-source software development kit (SDK) built specifically for your processor and evaluation board selections.



MCUXpresso IDE

An easy-to-use integrated development environment (IDE) for creating, building, debugging, and optimizing your application.



MCUXpresso Config Tools

A comprehensive suite of system configuration tools, including pins, clocks, SDK builder and more.

MCUXpresso IDE

- Eclipse-based development environment for NXP®MCUs
- Free
- Code size unlimited
- Integrated configuration tools, including pins, clock and peripherals tools
- Multicore debugging
- Visualization of registers, variables, heap, stack, ...
- Can be extended with Eclipse plug-ins

MCUXpresso IDE

- Console / Terminal
- Quickstart Panel
- Project explorer – **always select your project**
- Debugging – pause, breakpoints, variables, step over, step into ...
- Config Tools – open, save, **update**

MCUXpresso SDK

- Framework equivalent
- Production-grade software with integrated RTOS(optional), integrated stacks and middleware, reference software, and more
- Open-source peripheral drivers that provide stateless, high performance, easy-to-use APIs
- MISRA-C:2004 compliant and checked with Coverity static analysis tools
- Stacks/middleware: USB, FatFs, lwIP, SDMMC, TensorFlow, ...
- [API Reference Manual](#)

MCUXpresso Config Tools

- Generate initialization C code
- Pins tool – assigns internal signals to external pins, sets, electrical properties
- Clocks tool – graphical representation of the MCU clock tree system
- Peripherals tool – configures peripherals

MCUXpresso Config Tools - Tips

- Check selected project in the combo box
- Don't forget to update the code

Automated Testing

- We have automated test in CI for homeworks, this uses GitLab CI
- Do not touch the runners or the `.gitlab-ci.yml` file unless instructed
- Intentionally sidestepping the testing system will be punished
- Tests passing do not guarantee that your program is correct, it will always be checked by your TA for code quality, untested parts and intentional sidesteps
- It is possible our tests are overly strict and a correct program will not pass, reach out to Marek Vrbka (469025@mail.muni.cz or via Discord, there is a channel `#ci-issues` for it) if you think that is the case
- Only have one MCUXpresso project in each week

Homework

- We prepared a git repository for each of you:
<https://gitlab.fi.muni.cz/pv198/2024/<xname>>
- We suggest that you make a new branch for each week, with a name: *Week_XX*
where *XX* is a number of the week aligned to two characters.
For example: *Week_01*, *Week_02*, *Week_03*, ...
- If you want to submit your homework, create a git tag with name: *Submission_XX_Y*
where *XX* is a number of the week aligned to two characters, and *Y* is number of the attempt.
There has to be only one project in the tagged commit. For example: *Submission_01_0*, *Submission_01_1*, ...

Homework

Task 2 - Just a repository tag

1. Clone your git repository
2. Create a branch *Week_01*, push it
3. Create a tag *Submission_01_0*, push it
4. Check for a result of CI job
5. Tell us if there are any issues (ask your TA to confirm that you tagged it properly)

MUNI

FACULTY

OF INFORMATICS