Lab 2: Setting up a Development Environment

Lab IoT

Philipp H. Kindt

Assistant Professorship for Pervasive Computing Systems (PCS)
TU Chemnitz

November 11, 2021

Overview

- Programming and debugging is realized through different hardware blocks.
- Utilizing them also requires different software tools that need to play together. Setting up this environment is non-trivial.
- The goal of this exercise is setting up the development environment based on a Linux Mint operating system.
- Other microprocessor architectures require replacing some parts of this toolchain, but the overall handling remains similar. Hence, the knowledge obtained on it will extend to programming a large variety of different microcontrollers.

What is Needed?

- A compiler is needed to generate code for the nRF52 SoC.
- For flashing and debugging, we need to set up GDB and OpenOCD.
- Our code will use functions from the nRF52 SDK, which we need to download.
- We provide code templates for all exercises of this lab, which need to be retrieved from a repository.
- To conveniently edit & compile code and to flash and debug the microcontroller, we need the Eclipse IDE and multiple plugins.

Task 2: Setting up the Environment

- Our software templates are contained in a repository, whose address will be announced.
- Detailed setup instructions are contained in the README.md file of this repository.
- Follow the steps described in these instructions to set up the environment.
- When you are done, open a terminal and change the directory to the *exercises/lab1-3* subdirectory of your copy of the lab repository.
- Type `make` to compile our example code. Then, type `./flash_openocd.sh` to flash the code onto the nRF52 SoC.
- If you see the upper-left LED on Micro:bit screen blinking, you were successful.

What Else?

The following additional software might become useful.

- For reading strings printed on the Micro:bit, you need a serial terminal.
- For Linux Mint, we recommend the terminal *Cutecom*, which is contained in the official software repositories.
- The Micro:bit will register itself as a virtual serial port on your PC/laptop. To find out its device path, type `dmesg` directly after connecting the Micro:bit to the PC via USB. You should obtain something like this:
  `cdc_acm 2-3:1.1: ttyACM0: USB ACM device`
  This means that the device path is `/dev/ttyACM0`, which can be used in Cutecom. On the Micro:bit, you can send messages to the terminal using `log_printf()`. The syntax is identical to the `printf()` function on the PC, but does not support floating point values.