

## **Embedded and Ambient Systems**

### **Topic of exercise 1**

#### **1. Getting to know the developer card**

- A brief description of the development card and its connection to the PC.
- Starting and getting to know the development environment.

#### **2. Getting to know the views of the development environment**

A brief presentation of the different views of the development environment through the blink project:

- Opening and getting to know the sample project named blink (STK3700\_blink).
- Simplicity IDE (development)
- Debug (download, run program)
- Energy profiler (measurement of current consumption, monitoring of current consumption at function level)

#### **3. Sample program overview, API functions**

Brief description of the role of APIs (Application programming interface). BSP: Board Support Package

- CHIP\_Init(): errata correction
- BSP\_TraceProfilerSetup(): tracer initialization
- SysTick\_Config: system timer configuration
- BSP\_LedsInit(): LEDs initialization
- BSP\_LedToggle(): Switching the states of LEDs

#### **4. Debug options**

- Tracing functions and definitions
- Viewing the values of variables and registers (we only see local variables when we are inside the function)
- Task: in the blink project, follow which bits of the BSP\_LedToggle(...) register are set, and in the Registers window, see how the state of the LEDs is reflected in the bits of the register.
- Let's take a brief look at the disassembly code. Importance of disassembly.
- Sample code conversion: flashing the LEDs in a counting manner and thus watching the current draw.

## 5. Energy profiling

- Starting Energy profiler, brief introduction.
- Set the frequency using the `CMU_HFRCOBandSet(cmuHFRCOBand_**MHz)` function (possible values: 1MHz, 7MHz, 11MHz, 14MHz, 21MHz, 28MHz). The status of the LEDs should be constant (e.g. all should be switched on). The current draw can be plotted using Excel or Matlab, etc.

## 6. Startup code

- Introducing the `startup_gcc_efm32gg.s` file
  - define stack and heap
  - define interrupt vector table
  - populate initialized and uninitialized variables