Higher Level Programming

Agenda

1. MCU Expresso IDE
2. GPIO - button and LED
3. GPIO - interrupts
4. PIT - timing
5. PWM - LED brightness controlling

MCU Expresso IDE

- Allows us to:
  - Configure generated set-up code
  - Compile our project
  - Flash project to board
  - Debug
- Reset IDE layout: Toolbar -> Window -> Perspective -> Reset Perspective

MCU Expresso IDE - SDK

- Kit containing drivers, examples related to a board
- Install SDK:
    i. Chose workspace, close welcome page
    ii. Go to "Installed SDKs" panel (bottom middle tab)
    iii. Right-click on the empty space
    iv. Chose "Download and Install SDKs"
    v. Find "frdmmcxn947" SDK and install it

MCU Expresso IDE - New Project

- To create a new project:
    i. Go to "Quickstart Panel" (bottom left tab)
    ii. Click on "Create a new C/C++ project..."
    iii. Find "frdmmcxn947" SDK and click "next"
    iv. Change "Project type" to "C++ Project"
    v. Change "SDK Debug Console" to "Semihost"

Hello world

- We can print into console \o/
- use `PRINTF` function MACRO the same way as usual prinf

Pin Setup Tool

- Allows us to configure pins
- Go to Toolbar -> ConfigTools -> Pins
- Search pins in Pins panel (on the left)
- Click on the square in the first column and route it to correct peripheral
- See "Routing details" for pin configuration on the bottom

Configure Pins

1. In "Functional groups" pick "BOARD_InitLEDsPins"
2. Click on the flag on the right to turn on this group
3. See how the pins are configured in "Routing Details"
4. Do the same for "BOARD_InitBUTTONsPins" functional group

Peripherals Setup Tool

- Allows us to configure peripherals (PWM, timers)
- Go to Toolbar -> ConfigTools -> Pins

Assignment (together)

1. Look at the generated project stucture
2. Check GPIO drivers in code
3. Check pin setup in code
4. Find functions for handling GPIO

Assignment (Alone)

1. Read button state
2. Print something upon button press
3. Change led whe the button is pressed

Bonus: Interrupts

1. Update pin config to allows interrupts on button pins
2. Add GPIO interrupt peripheral using Peripheral setup tool
3. Copy and paste interrupt handler into code
4. Handle button press events using interrupts instead of polling

Bonus: PIT

1. Setup PIT peripheral to generate periodic interrupts
2. Print something to console every 3 seconds
3. Add code for PIT handling

1. Setup PWM peripheral
2. Update pin config to route LED from GPIO to PWM
3. Add code for PWM handling