Prof. Dr. Marco Zimmerling Max Granzow Jonas Kubicki
https://nes-lab.org/

Networked and Low-Power Embedded Systems

WS 2025/26

Holiday Lab: Whac-a-Mole

Introduction

This exercise focuses on implementing a small Whac-A-Mole–style reaction game on the nRF52840DK using the four on-board LEDs and the four corresponding buttons. This simple version of the game operates in rounds. During each round, one of the four LEDs is turned on after a randomly chosen delay. As soon as the LED turns on, the player must press the matching button to *whack the mole*. The LED turns off once a button is pressed or the timeout interval of 2 s elapses. If the player manages to press the button corresponding to the correct LED within an interval of 700 ms after the LED turned on, a *hit* is registered. If they press the correct button after said interval has elapsed but before the LED turns off, a *late hit* is registered. If the player presses an incorrect button or the timeout is reached before any button was pressed, a *miss* is registered.

> ℹ To compile the program you can simply use the `$ make` command. If you want to flash the file to your nRF, you can use the `$ make flash` command.

Header Figure: https://www.flaticon.com/free-icon/whack-a-mole_1835817

Peripherals

The first task is to configure the four LEDs and buttons on the nRF52840DK.

> LED pins: P0.13, P0.14, P0.15, P0.16.
> Button pins: P0.11, P0.12, P0.24, P0.25

In order to set time constraints for the game, a timer should also be configured to pulse every 1 millisecond. Another useful peripheral is the true random number generator provided by the nRF52840.

> Information about the RNG peripheral can be found in the product specifications on page 363 chapter 6.21 RNG - Random number generator

You can fill in the TODOs in the helper.c file and then use the given functions, or create your own. Remember when you do so to also define these functions in the header file helper.h.

Game Logic

Next we want to implement the game logic. To do this, you can use the functions from the `helper.c` in the `main.c`, as well as the printf function for debugging etc. You should implement the rounds as described above. After each round, you should indicate the outcome via UART and via different blinking light patterns. You can compile your code with `$ make` and flash it like usual with `$ make flash`.

Scoring System

Implement a scoring system for the Whac-a-Mole game. The system shall output scores via UART. A penalty of 1 point is applied for a *miss*, a *late hit* neither adds nor looses points and a *hit* awards 1 point.