

Lab 3. Joystick and LED Programming With CubeMX

Introduction

We have used code generated from STM32CubeMX to configure GPIOs to control LEDs and Joystick keys on the HW kits in in the `expl_014_template_for_HW_prjt` project and WS 3. In this lab, we learn the details of the setups in STM32CubeMX to generate code. We also do further programming based on the generated code.

To use STM32CubeMX, you need to install it first. The installation is straightforward. If you need instructions, please see `cec32x_devtool_14_CubeMX_installation.pdf` under **Dev tools** in **Modules** on the **CEC 320 Canvas** page.

Lab Tasks

Code generation with CubeMX

(30 points)

You can get started with the setup from scratch according to the steps provided in `expl_014_template_for_HW_prjt.ioc`

You need to define all the (five) keys of the Joystick and the Red and Green user LEDs in the same way as we have used in `expl_014_template_for_HW_prjt`. Keep the following in mind.

- The pins of the Joystick and LEDs can be found at `[[cec32x_HWKit_12_GPIOs_of_L476_F412_kits]]` `cec32x_HWKit_12_GPIOs_of_L476_F412_kits` or the files below this one.
- Label the keys as `JOY_L`, `JOY_R`, `JOY_U`, `JOY_D`, `JOY_C` for the left, right, up, down, and center keys, respectively.
- Label the LEDs as `LD_R` and `LD_G` for the red and green user LEDs.
- Make sure the LEDs are all off after the program is initialized.

Note that you need to follow the instruction in

Programming based on the generated code

(50 points total)

In Keil, write your own user C code to do the following subtasks in your `user_tasks.c` and `user_tasks.h` files. You may want to take a close look at the programming assignment for WS 3 before tackling on the tasks here.

- (10 points) Write a function to set the two LEDs to have LD_R on and LD_G off. Name the function as `void USR_Task_LD_R_on_LD_G_off(void)`. Call this function in the `main.c` before running the `while` loop. This can be done in a similar way as the added global function in `[[cec32x_ws_18_WSO3_getting_started_with_HW_programming]]` `cec32x_ws_18_WSO3_getting_started_with_HW_programming`. Make sure this works before moving forward to code the other tasks. From a debugging perspective, you should make sure the calling of this functions work when commenting out everything in the `while` loop.
- (10 points) When no key is pressed, the two user LEDs should be blinking **alternately** with a delay of 100 ms before checking the JoyStick keys again.
- (5 points) When the Up key is pressed, both LEDs should be on for a duration of 2 seconds.
- (5 points) When the Down key is pressed, both LEDs should be off for a duration of 2 seconds.
- (5 points) When the Left key is pressed, the Red LED (LD_R) should be on and the Green LED (LD_G) should be off for a duration of 2 seconds.
- (5 points) When the Right key is pressed, the Red LED should be off and the Green LED should be on for a duration of 2 seconds.
- (10 points) When the Center key is pressed, the two LEDs should blink **together** with a period of 200 ms for a duration of 2 seconds.

The demo will be graded according to the points given above. Note that you need to use the conditional compilation directives as given in `expl_014_template_for_HW_prjt` to make sure your code to work for both HW kits. Yet, you don't have to debug for the kit you do not have.

Submission of Lab Report

In addition to the live demo of your running code to the TAs, you need to submit your zipped project file and a pdf report including the following:

- (10 points) The code snippets in your entire `user_tasks.c` file. Note that you need to write the code with clear indentation and comments so that a fellow programmer can understand your code easily. Even better, you want to replace your comments with the meaningful function names as shown in `[[cec32x_ws_18_WSO3_getting_started_with_HW_programming]]` `cec32x_ws_18_WSO3_getting_started_with_HW_programming`.
- (10 points total, 5 points each) Answer the following questions in the report:
 - What board you are using? What mode is being used to drive the LEDs (e.g. Push-pull or Open-drain)?
 - Can we change the output mode above to the other mode? Why or why not?