

Assignment 2

LED Pattern

1 Goal

The goal of this assignment is to represent a set of LED patterns using object-based techniques. Students will learn how to encapsulate an abstract idea, i.e. an LED pattern, with C++ classes. Students will implement a typical "hello-world" program to display those patterns with a physical LED.

2 Setup

1. We will use the same project baseline (platform-stm32l475-disco-assignment1.tgz) as in Assignment 1. If you have not done so already, follow the instructions of the previous assignment to set up your development environments.
2. With the firmware running on your target board, type the command "**gpio on 0**" at the UART console (e.g. Tera Term). Verify the green USER LED next to the ST-Link USB connector (LD2) blinks twice.

3 Tasks

1. To-do tasks are marked with "`// Assignment 2`" in the source code.
2. Complete the implementation of the GpioPattern and GpioPatternSet classes in **Src/app/GpioOutAct/GpioOut/GpioPattern.h**.

Note – These member functions are *inline*, and therefore implemented in the header file.
3. Implement **two** LED patterns in the structure named TEST_GPIO_PATTERN_SET in **Src/app/GpioOutAct/GpioOut/GpioPattern.cpp**.
4. Implement the handlers for the command "gpio on <index> <repeat>" in **Src/app/GpioOutAct/GpioOut/GpioOutCmd.cpp**.

This command shows the indexed pattern on the USER LED (LD2). If repeat is "0", it is shown once; otherwise it is shown **5 times**. Handle the case when the index is out of range. As a reminder, the set of LED patterns is defined in the structure TEST_GPIO_PATTERN_SET.

Before adding your code, please remove or comment out the sample code marked between "`// Beginning sample code`" and "`// End sample code`".

5. Test your code with the console command:

gpio on <pattern index from 0 to 1 > <0 for non-repeating; 1 for repeating>

For example, "gpio on 1 1" shows pattern 1 for 5 times.)