Lab 3: GPIO and Interrupts

Goals

1. Understand the basic concept of interrupts
2. Use external interrupts (EXTI) to detect a signal event external to the microcontroller

Pre-Lab Assignment

- Read chapter 11. Interrupts (sections 1,2 3 and 6) from the book "Discovering the STM32 Microcontroller" Geoffrey Brown
- Watch the instruction video: "Stm32 Peripheral Drivers from Scratch : GPIO Programming: Interrupts" by Eddie Amaya, https://www.youtube.com/watch?v=A_zuzEajLE4&t=409s

Lab Demo

In the rest of the document we will use the acronym MCP to address the Manual Control Panel. The MCP contains two buttons and two LEDs, labeled B0 and B1 for the buttons and LED0 and LED1 for the LEDs. A short press has a duration between 20 and 500 ms. A long press has a duration longer than 500 ms.

The letters in front of the requirements refer to the MoSCoW method.

Functional

(M) A short press on B0 must turn on LED0
(M) A short press on B1 must turn on LED1
(M) A long press on B0 must turn off LED0
(M) A long press on B1 must turn off LED1
(M) A press shorter than a short press must be ignored

Non-functional

(M) The MCP must function properly with buttons with a bouncing time of less than 20 ms.
(M) Button presses must never be missed i.e. detecting button presses must be done without polling.

Part A:

1. Investigate what is needed to implement the requirements described above. Research and compare different alternative (possible) solutions and give a motivation why the alternative you are choosing is the best choice to implement. This includes algorithms, needed internal MCU components and so on.
2. Design all the state diagrams needed for the operation of the MCP.
3. Conduct POC's (Proof of concepts) which handle the configuration of the needed internal components of the MCU. POC's are experiments which you carry out to test critical parts of your design. Please document these POC's and use pictures / diagrams to clarify your setup and measurements. Note that a POC can be seen as a research method.
4. Write and integrate all necessary code such that all requirements are implemented. Assess, using tests or testcases, that all requirements are implemented properly.
5. Write a report in which you show all taken research steps and elaborate on the choices made, how you validated the results and so on.

Additional information/Hints

To measure a time delay you may use the SysTick as used in the previous assignment to get the elapsed time in ms. You may use the HAL_GetTick().You could research and think of alternative methods.

Part B:

Do something cool! You can come up with something on your own but make sure it is of a reasonable complexity.

Part C: Submission

The grading for this challenge will mainly be based on your documentation. The reader should be able to verify the design process and the design choices based on objective arguments, test results, measurements, etc. Use pictures and diagrams to clarify your setup / measurements and design. Fully functional code without the required documentation has not much value.

Your submission is a zip file that consists of:

1. All code
2. Proper documentation consisting of:
   2.1. Title page with date and name
   2.2. Short introduction

2.3. Your research: Show all relevant research steps and elaborate on the choices made

2.4. Your design: all diagrams with short description and well argumented design choices

2.5. How you tested your implementation including proof (how you validated the results and so on)

2.6. Reflection on what you have learned

2.7. Bibliography (sources) in APA style (https://www.bibme.org/citation-guide/apa/)

3. A short video (max 4 min) where you demonstrate and clearly explain your solution