Contents

[Project requirements 1](#_Toc116929514)

[Doc: 1](#_Toc116929515)

[SRC: 1](#_Toc116929516)

[Implement the GPIO driver 1](#_Toc116929517)

[Implement Timer driver 1](#_Toc116929518)

[Implement interrupt driver 2](#_Toc116929519)

## Project requirements

- Control blinking of an LED for user-defined ON and OFF periods

- Create a function in the App layer that takes as input from the user a specific ON time and OFF time in seconds, lights a LED for the given-ON time, and dims it for the given OFF time. It’s prohibited to use predefined delay functions, only timer driver functions can be used. This function implements implicitly the PWM module using Timer Overflow.

- The COTS folder for the whole course including a folder for each layer in the layered architecture including MCAL, HAL, SERVICE, APPLICATION, and LIBRARIES

## Doc:

-A schematic/block diagram according to your understanding of the requirements.

- “Main fnc. mkv”: a screen recording of the code (main. c), run it while showing the Tiva C board simulator showing how the LED status would vary by changing the ON/OFF time in runtime.

-“screen record GPIO driver. mkv”: A screen recording of the GPIO driver, explaining its functions and Macros, and how each function will help achieve the project’s functionality.

-“ NVIC\_Screen\_rec.mkv”: a screen recording of the Interrupt driver, explaining its functions and Macros, and how each function will help to achieve the functionality of the project

-“ SysTick Screen record-part1.mkv”, “ SysTick Screen record-part2.mkv”:

a screen recording of the TIMER driver, explaining its functions and Macros, and how each function will help to achieve the functionality of the project

## SRC:

### Implement the GPIO driver

Create the GPIO driver from scratch using the sufficient interface, configure, private, and program files

Create functions in the GPIO driver that are needed to perform the required project functionality

### Implement Timer driver

Create the Timer driver from scratch using the sufficient interface, configure, private and program files

Create functions in the Timer driver that are needed to perform the required project functionality

Creating a Timer Overflow ISR that performs the required functionality

Passing the ISR as a callback function from App Layer to the Timer driver in the MCAL layer

### Implement interrupt driver

Creating the Interrupt driver from scratch using the sufficient interface, configure, private and program files

Create functions in the Interrupt driver that are needed to perform the required project functionality