

[Logout](#)

PROJECT SPECIFICATION

LED control

Blinking LED

CRITERIA	MEETS SPECIFICATIONS
Read project requirements	<p>Control blinking of an LED for a user-defined ON and OFF periods</p> <p>Create a function in the App layer that takes as an input from the user a specific ON time and OFF time in seconds, lights a LED for the given ON time, and dim 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.</p> <p>You should deliver a schematic/block diagram according to your understanding of the requirements.</p>

CRITERIA	MEETS SPECIFICATIONS
Prepare project folders	<p>Create a COTS folder for the whole course including a folder for each layer in the layered architecture to include MCAL, HAL, SERVICE, APPLICATION and LIBRARIES</p> <p>Provide a screenshot of your COTS folder with the stated arrangement</p>
Implement the GPIO driver	<ol style="list-style-type: none">1. Create the GPIO driver from scratch using the sufficient interface, configure, private and program files2. Create functions in GPIO driver that are needed to perform the required project functionality <p>Provide a screen recording of GPIO driver, explaining its functions and Macros, and how each function will help to achieve the functionality of the project (Maximum 3 minutes)</p>
Implement Timer driver	<ol style="list-style-type: none">1. Create the Timer driver from scratch using the sufficient interface, configure, private and program files2. Create functions in Timer driver that are needed to perform the required project functionality3. Creating a Timer Overflow ISR that performs the required functionality4. Passing the ISR as a callback function from App Layer to Timer driver in

CRITERIA	MEETS SPECIFICATIONS
	<p>MCAL layer</p> <p>Provide a screen recording of TIMER driver, explaining its functions and Macros, and how each function will help to achieve the functionality of the project (Maximum 4 minutes)</p>
Implement interrupt driver	<ol style="list-style-type: none">1. Creating the Interrupt driver from scratch using the sufficient interface, configure, private and program files2. Create functions in Interrupt driver that are needed to perform the required project functionality <p>Provide a screen recording of Interrupt driver, explaining its functions and Macros, and how each function will help to achieve the functionality of the project (Maximum 2 minutes)</p>
Test your application	<p>Deliver a screen recording of your code (main.c), and run it while showing the Tiva C board simulator showing how the LED status would vary by changing the ON/OFF time in runtime. (Maximum 3 minutes)</p>