

|  |
| --- |
|  |
| **Project Report** |
|  |
| |  |  | | --- | --- | | Student 1 details: | Quang Huy Tran – n10069275 | | Student 2 details: | Arik Intenam Mir - n9637567 | | Student 3 details: | Sachin Fernando – n9879129 | | Student 4 details: | Manavdeep Singh – n9860665 | | Unit details: | EGH456 – Embedded Systems | | Unit coordinator: | Dr. Chris Lehnert |   May 28, 2020 |

Table of Contents

[Introduction 3](#_Toc42857044)

[Aim 3](#_Toc42857045)

[Problem statement 3](#_Toc42857046)

[Statement of contribution 4](#_Toc42857047)

[Design and Implementation 4](#_Toc42857048)

[Software 4](#_Toc42857049)

[Libraries and Tools 4](#_Toc42857050)

[Hardware 4](#_Toc42857051)

[Work Breakdown and approach 5](#_Toc42857052)

[Sensors & Data Filtering 5](#_Toc42857053)

[Light Sensor 6](#_Toc42857054)

[Graphical User Interface 7](#_Toc42857055)

[Results 7](#_Toc42857056)

[References 7](#_Toc42857057)

[Appendix 7](#_Toc42857058)

# Introduction

The purpose of this project is to develop a program to monitor and control via a Graphical User Interface a three-phase electric motor using a Tiva microcontroller.

## Aim

The main goal of the project was to develop a set of functionalities for controlling and monitoring a three-phase electric motor via a Graphical User Interface (GUI) using a Tiva Microcontroller.

## Problem statement

Our team was contacted by an automotive manufacture to design the real time controller and hardware interface of the electric motor for the vehicle. The task includes real-time handling of multiple tasks such as sensing and actuation of a 3-phase Brushless DC motor, sensor acquisition and filtering and system display of critical information. In terms of requirements, there are 3 main sections: Motor Control, Sensing and User Interface. Firstly, Motor Control requires to write an application programming interface for the driver that will control the motor safely within the acceleration and deceleration limits, handle any fault conditions using the provided state transition diagram. With Sensing, writing a device driver for the various sensors available on the vehicle is required to monitor critical information such as temperature, current, ambient humidity, magnitude, speed and light. All received data are filtered using sliding window technique to avoid triggering false alarms. Finally, a graphical user interface (GUI) is developed to allow the client to control motor using buttons on the first tab and view the plot of the filtered sensor information from the vehicle in a real-time.

## Statement of contribution

Each student is responsible for part of the work in the project. Details of contribution are in Table 1 below:

*Table 1. Table of contribution*

|  |  |
| --- | --- |
| Student name | Contribution |
| Quang Huy Tran |  |
| Arik Intenam Mir |  |
| Sachin Fernando |  |
| Manavdeep Singh |  |

# Design and Implementation

## Software

The software/ tool used for the development of this project are:

* TI Code Composer Studio
* To develop the software of the project.
* PuTTY
* Serial communication medium, for debugging and development.

## Libraries and Tools

* TI Driver Examples
  + For the GUI and Real Time control
* GPIO and UART
  + To implement the GUI and the sensors.
* ADC
  + To control the current and PWM signals.
* I2C
  + To read and work with ambient luminance.

## Hardware

A list of hardware used for this project is given below:

* Microcontroller – Tiva TM4C1294NCPDT Microcontroller
* Motor – DRV832x motor driver board
* Development Board – TI Tiva C Series EK-TM4C1294XL
* Touch Screen – TI BOOSTXL-K350QVG-S1 Touch Screen
* Light Sensor – TI OPT3001 Ambient Light Sensor

## Work Breakdown and approach

For completing the project in time with all the necessary functionalities working, the project was broken down into few basic work packets. These are listed below:

* Delegating roles to group members.
* Software Development
* Implementing Graphical User Interface (GUI)
* Implementing Sensors
* Implementing Motor control
* Integrating the sub-parts
* Finalizing the software development of the project
* Documentation of the project report.
* Preparing video for final video demonstration

## Sensors & Data Filtering

For monitoring the information regarding the state of the vehicle, it was required to implement sensor drivers for the available sensors on the vehicle, namely –

* Temperature
* Current
* Magnitude
* Speed
* Light

As the input of the sensors has noises, it is a requirement to identify and eradicate noises in real-time so that the noises don’t trigger false alarms. The specifications for data filtering are:

* Temperature: 3 samples at 2Hz. Uses UART connection.
* Current: 5 samples at 250Hz. Uses current sensors
* Magnitude: 5 samples at 200Hz. Setup for detecting E-stop condition threshold.
* Speed: 5 samples at 100Hz.
* Light: 5 samples at 2Hz. Uses the I2C connection.

Each of the sensors and how they were implemented are discussed briefly in the sub-sections below:

### Light Sensor

In order to implement the functionalities of the light sensors, firstly, it was required to read the light level from the ambient light sensor and then filtered using I2C connection. The light levels were read using the Opt3001 sensor and the necessary functionality for this was implemented in the .c file “lightsensor.c” The port I2C0 was used for the light sensing functionalities. The .c file EK\_TM4C1294XL was modified accordingly for enabling and configuring the pins and peripherals required for this sensor aspect of the project.

More to write here.

## Graphical User Interface

The graphical user interface (GUI) for this project was implemented using the provided TivaWare Graphics Library in TI-RTOS. The functionality for interacting with the GUI was implemented via buttons on the launchpad, where the switch SW1 was enabled for switching between the interactable options on the LCD screen and switch SW2 was enabled for selecting a certain option(desired panel from the user), and also to mention that the RESET button on the launchpad was enabled for resetting the GUI and start from an initial state whenever required.

The GUI has 8 panels in total which includes a introductory page, representing the options of choosing between “Motor Control” and “Sensors”. If the option “Motor Control” is chosen, it prompts the user to a new panel, implemented for controlling and setting the speed of the 3-phase brushless motor. And if the latter option “Sensors” is to be chosen, that prompts the user to a new panel with 5 containers, each for the 5 sensors, namely “Temperature”, “Current”, “Magnitude”, “Speed” and “Light” respectively. Selecting any of these 5 options will take the user to a further panel, where the graph functionality for each sensor are implemented.

Every panels implemented for the GUI functionality has several buffers for storing the required variables and textual data,

# Results

# References

1. Tiva TM4C1294NCPDT Microcontroller Datasheet:

<https://www.ti.com/lit/ds/symlink/tm4c1294ncpdt.pdf>

1. Opt3001 Ambient Light Sensor Datasheet:

<https://www.ti.com/lit/ds/symlink/opt3001.pdf>

1. TI-RTOS:  
   <http://processors.wiki.ti.com/index.php/TI-RTOS>

# Appendix

* Texus Instruments Code Composer Studio Version 9.3.0:

<https://www.ti.com/tool/CCSTUDIO>

* PuTTY:

<https://www.putty.org/>

* Tiva™ C Series TM4C1294 Connected LaunchPad Evaluation Kit:

: