PROBLEM BASED LEARNING ASSIGNMENT: Controller and Hardware Interface for Electric Vehicle Motor

Date Available on Blackboard: Week 6

Date Demonstration due: Week 12 at a scheduled time for your group

Date Report due: Monday 28th May 11:59 pm Week 13

Weight: 30% (Demonstration + Presentation)

The specifications in this document is subject to change.

Problem Description

Battery-Electric and Hybrid-Electric vehicles are becoming more and more economical, efficient and environmentally friendly resulting in an increase in adoption rates. In addition, the recent advancements in autonomous electric vehicles are also resulting in the automotive industry starting to transition from petrol to electric vehicle design. As an embedded system engineer you have been contracted by an automotive manufacture who is transitioning from petrol to Battery-Electric technology. Your task is to design the real time controller and hardware interface of the electric motor for the vehicle.

Your challenge is to handle the complete sensing and actuation of a 3-phase Brushless DC (BLDC) motor common in electric vehicles[1]. This will include monitoring the state of the motor such as velocity, power and temperature and handle the start-up, braking and emergency procedures for the system. Your system design will require real-time handling of multiple tasks for sensor acquisition and filtering, motor fault handling and system display of critical information.

Your setup includes a motor testing station including the motor driver and sensor board and a development kit for the Tiva TM4C129X microcontroller. Figure 1 illustrates the setup interface of the testing station. Inputs and outputs of sensors and per-phase connections are compatible with the requirements of the microcontroller ports, with an electrical interface to achieve this.

The TM4C129X development kit is connected to the motor driver board via a DB-25 connector with all necessary I/O and device pins mapped through this connector. The mapping of the microcontroller pins (GPIO, I^2C, ADC) and the motor driver inputs and outputs is provided in a separate document. Additional documentation that will be provided includes a description of the operation of the motor and a statement of the functions of the motor driver inputs and outputs.

[1] For a concise summary on the advantages and disadvantages of BLDC vs Induction motors see https://www.tesla.com/en_AU/blog/induction-versus-dc-brushless-motors

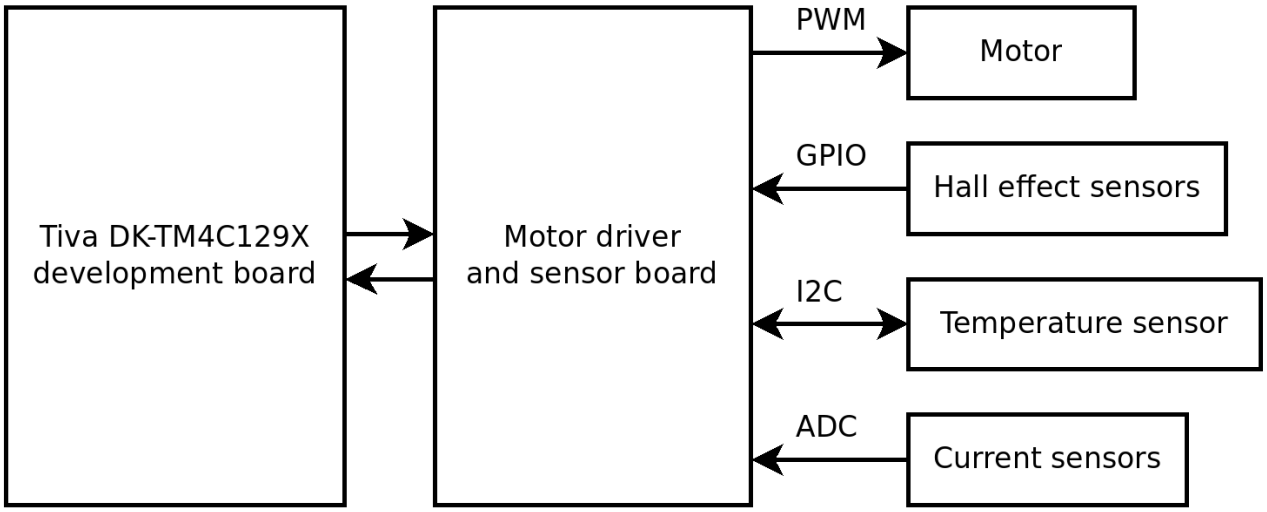

Figure 1 - Testing Station System Diagram. The system includes a microcontroller development kit, a motor driver and sensor board, brushless motor with hall effect sensors for position and velocity feedback and temperature and current sensors for fault monitoring.

Requirements

Your tasks are to:

1. Write a device driver for the motor driver to provide an application programming interface that will:
   a) Read and filter the temperature from the temperature sensor over an I^2C connection.
   b) Read and filter the current on the 24V line via the analogue signal provided by a current sensor.
   c) Measure and filter the current speed of the motor in revs/min.
   d) Control a user given desired speed of the motor in revs/min.
   e) Start and control the motor safely, handling any fault conditions using the provided state transition table. Faults can be detected using the provided I/O and sensors to the motor driver board.
   f) Start and stop the motor using a safe acceleration and deceleration specified (will be defined later in semester based on load requirements).
2. Write an application program that uses the device driver and other drivers and allows:
   a) Starting and stopping the motor using buttons.
   b) Displaying start and stopped status using an LED.
   c) Setting the current desired speed of the motor using user input.
   d) Keeping track of time and displaying calendar time.

e) Setting an upper limit of allowable current on the 24V line. If this condition is met a fault has occurred and the motor should be stopped safely.

f) Setting an upper limit of allowable temperature of the motor. If this condition is met a fault has occurred and the motor should be stopped safely.

g) Display the critical and runtime information such as the state of the motor and calendar time using the LCD touch screen display in a real-time, user friendly manner. Use your own structure for the display.

h) Display the following real time information as a moving plot over time: power usage of the motor (in watts), temperature (in Degrees Celsius) and speed (revs/min). Units can be of different magnitudes (e.g. Kilowatts), scaled or adjusted for display purposes.

Additional requirements

The following is some additional requirements to be met when designing the solution to your task.

Data Filtering

All received data is noisy in some fashion such as temperature, current and speed. In order to correctly identify faults in a real-time fashion and ensure noise does not trigger false alarms you must filter this data before displaying it. You must filter the data using the following specifications:

- Temperature – no less than 5 samples at 50 Hz.
- Current – no less than 5 samples at 1 KHz
- Speed – no less than 5 samples at 1 KHz

It is recommended that this filtering should not be done in a hardware interrupt. For example you should capture 5 independent samples of the current every 1000ms and then apply an appropriate filtering technique to estimate an accurate value of the current.

Speed Control

The motor must be controlled by specifying the revs/min. No marks will be awarded if the motor is controlled by specifying the PWM duty cycle.

Groups

Each assignment group should comprise of 4 students – preferably 2 groups of 2 students who have worked together within laboratory exercises. You must form a group within your enrolled lab class. In an exceptional circumstance, a group of 3 may be permitted. The procedure for forming groups will be announced via Blackboard shortly after the assignment has been released. This will allow

the unit coordinator to facilitate the process of placing those students who have been unable to form the required size groups in merged groups. Before week 12, a schedule of presentations/demonstrations will be placed on Blackboard using the final list of groups. Each group of 4 students will be allocated 25 minutes for demonstration and presentation.

Demonstration and Class Presentation

You are required to present and demonstrate your work to the teaching team. The groups will present as a team; however, you will be individually assessed. For the demonstration, you should prepare to show and talk about what you have done. You will also be asked questions throughout your demonstration. **Additional advanced** features will also improve your mark only if the design requirements for that system have been met.

For the presentation you should aim to present your designed system to your client which is the automotive manufacture (teaching team). Your presentation should include an overall description of your final design, how you have met specifications and showing all relevant details.

The duration is strictly 3 minutes per student for presentation (Come prepared for this) and 13 minutes for demonstration. You should prepare only one PowerPoint slide per student.

Assessment

PRESENTATION (5%) - Based on effectiveness of oral communication and quality of the one visual slide presented by each student.

ACHIEVEMENT (15%) - Based on demonstration of evidence that requirements are met.

REPORT (10%) - A suggested report format is provided at the end of this document.

Criterion Referenced Assessment Sheet – CRA sheets for the presentation, demonstration, and report will be available on Blackboard.

Late Demonstration

This **will not be permitted** except for cases permitted by QUT policy. Regardless of the state of completion of your assignment work, you are required to make a presentation as scheduled describing your work up to that time.

Penalty for Late Submission

QUT policy for late submission of assignments will apply. A late report without an approved exemption will receive 0%.

Requirements that could not be demonstrated

If you describe your work well enough – and your demonstration marks were low – you could be given partial credit to compensate for an inability to demonstrate when it was required. You can add an explanation for each failure if you have been able to figure out how you might proceed if you could do it again.

Suggested Format of the Report

The report should contain the following items. The content and style are flexible if these are separately identifiable. Approximate page guidelines are given in parentheses.

- **Cover page** – names, student ID numbers, course, and unit information. (1 page)
- **Table of Contents** – Section headings, list of figures, list of tables. (1 to 2 pages)
- **Introduction** – Problem statement, context, requirements, and statements on the individual contributions by each team member. (2 pages)
- **Design and Implementation** – Approach to design, important issues and choices and their relationships to theoretical concepts and the hardware and software platforms. Check if you have addressed:
  - Provided a graphical representation for your design?
  - Provided a Project Folders and Files diagram?
  - Did you use the GPIO module? How? Did you use the graphics library? How? Did you use Tasks, Hwi and Swi? How?
  - Did you use multiple threads/handlers? How? Did you use the ADC? How?
- **Results** – Summary of evidence of functional requirements that were demonstrated and explanation of failures as learning outcomes in terms of what could have been done differently. (2 pages)
- **References** – Including vendor supplied technical documents with a table that links each document to sub-sections of your report where they are relevant with entries: The reference number, the sub-section number, topic keyword or keywords, the pages that were found useful. (2 pages)

- **Appendix** – Mention the name of the hardware development platform and versions of the tools used such as Code Composer Studio, Tivaware, TI-RTOS, etc. (1 page)