ME333 Final Project

March 6, 2014

1 Overview

You will create a motion controller running on the PIC32. It will use an inner and outer loop structure, with the inner loop controlling current and the outer loop controlling position. The completed project will allow the motor to track trajectories specified from a PC (in degrees).

On the PC, the user requests things from the PIC using a menu (started by running pic_menu.m). The PIC handles these requests in menu.c (a file that you need not modify or even read). The PIC must perform different functions depending on the request. The code in menu.c keeps track of the state of the PIC by updating the variable *core_state*. Your code will need to check core_state and perform the appropriate action (for instance in PWM mode your current loop will set PWM directly, whereas in IDLE mode it will set the H-bridge to coast mode.

2 Quickstart

We have provided you with many files to make this project work. The pic folder contains all of the .c and .h files that will run on the PIC. The other directories contain MATLAB code that makes the user interface work.

This project is fairly free form. We provide the code that communicates with the PIC and reads the sensors and keeps track of the state of the PIC, you will do almost everything else. We provide you with stubs for several functions that you will need to implement (in both motion.c and current.c). These functions act as an interface between the code we provide you and your own code. Aside from this minimal interface, everything else is up to you. You will need to create additional functions and define additional variables in order to complete the project.

You will need to read through motion.h, current.h, motion.c and current.c to find out what to do. Pay close attention to the comments in the .h and .c files and make sure your implementation of these stub functions matches the specification: otherwise the matlab menu system may not work. The entire project can be completed by modifying motion.c and current.c. The TODO:

comments inside motion.c and current.c indicate tasks that you will need to complete to finish the project.

First you should get your test environment set up and get a feel for what the matlab menu will do.

1. Copy your makefile to the ./pic directory. Build the pic code and load it.
2. Open matlab. Navigate to the project directory. Run startup.m.
3. Run pic_menu.m to launch the menu. Play around with the commands. Most will not work (that's where you come in!)

2.1 States

The PIC must perform several different tasks depending on context, such as direct current control, current loop tuning, and trajectory tracking. We therefore define different states (stored in core_state) so that we can perform the proper task given what the user has requested. The PIC has 5 states, all defined in core.h menu.c manages the transitions between states in response to user input. All you will have to do is understand what each of the states are and perform the correct action based on the current state.

1. IDLE - When the PIC is in IDLE mode, the H-bridge is set to off mode (that is the switches are high impedance so the motor is coasting).
2. PWM - In PWM mode the user is controlling the PWM duty cycle directly.
3. TUNE - In TUNE mode, the current loop is tracking a 100Hz -200mA to 200mA square wave
4. TRACK - In TRACK mode, the motion loop is tracking a trajectory (which is specified as a list of angles, in degrees). When it gets To the end of the trajectory, it will simply stay at the last position in the trajectory.
5. HOLD - In HOLD mode, the PIC is holding a specified angle (called the hold angle). Note that this hold angle should be stored separately from the trajectory.

3 Files

3.1 To Read

3.1.1 core.h

Contains many functions that you will need to call from your own code. Lets you read from the ADC, read the encoder, and keeps track of the state. You will be querying the variable core_state in your control loops to determine the mode in which the PIC is operating and perform the appropriate action (see **??**).

3.1.2 current.h

Contains code related to current control. You will need to implement the functions defined in this header file to interface between your current control loop and the menu. Code in your motion controller will also call some of these functions to set the amount of current going through the motor..

3.1.3 motion.h

Contains code related to motion control. You will need to implement the functions defined in this header file to interface between your motion control loop and the menu.

3.1.4 streaming.h

You will need to call streaming_record, from your control loop to output data to the PC. The other functions you do not need to use. Streaming record simply stores the values you pass it into an array, they are written to the PC by streaming_write, which is called from menu.c and does not run in an interrupt.

3.2 To Modify

You only need to modify two files, current.c and motion.c.

3.2.1 current.c

Implements the current control loop. Read through this carefully, implement all the TODO items.

3.3 motion.c

Implements the motion control loop. Read through this carefully, implement all the TODO items.

3.4 PIC code

3.4.1 main.c

Mostly boilerplate. Calls the initialization functions and then runs the menu communication code.

3.4.2 menu.c

Runs the code that communicates with the PIC. Based on the request, this code sets the core_state variable to the appropriate value, calls the functions of motion.h and current.h to modify the behavior of the current and motion control loops, and writes data that was recorded with streaming_record(...) to the PC (only when the PC requests it).

3.4.3 streaming.c

Handles recording data from interrupts and writing the data to the PC from mainline code (i.e. code that is not in an interrupt).

3.4.4 dee_emulation_pic32.c, dee_emulation_pic32.h

Microchip provided library that makes it easy to read to/ write from the flash program memory. Used for storing the gains on the PIC.

3.5 Matlab

All files in the *comm*, *menu*, *ui*, as well as pic_menu.m and startup.m are matlab files that make the menu work. You will not need to modify any of these files.