Michigan State University
ECE480 – Senior Design
Design Team 4
October 21, 2010

Roy Dong
Micajah Worden
Andrew Kleeves
Mark Barnhill
Dave Seaton

Facilitator: *Professor Elias Strangas*
Sponsor: *Tim Adcock*

Email:msuece480dt4@gmail.com

MSP430 MOTOR CONTROL CARD FOR DRV8412 MOTOR DRIVER

Proposal

EXECUTIVE SUMMARY

Motor control algorithms are necessary for predictable behavior in the face of unpredictable conditions. Texas Instruments has requested that we redesign a pre-existing implementation of a motor control algorithm, through a C2000-microcontroller-based motor control card, to use a microcontroller from the MSP430 family. This project both has a hardware component, requiring the redesign of a pre-existing card, and a software component, requiring the implementation of a motor control algorithm specifically designed for the new MSP430 motor control card. Texas Instruments requested that we adhere to the specifications and functionality of the C2000 card as closely as possible. Our design solution proposes the use of the MSP430F5435 and an integrated hardware fabrication and assembly process.

Table of Contents

INTRODUCTION

Different direct current (DC) motors require different driving mechanisms. Our project requires the generation of driving signals for two types of motors: brushed DC motors and stepper motors.

A brushed DC motor consists of six components: field magnets, a DC power supply, an armature or rotor, an axle, a commutator, and brushes. Field magnets are stationary magnets and are used to create a magnetic field in the motor. The DC power supply is used to provide a current to the armature. The armature consists of multiple coils wound around the axle. When the DC power supply is connected an electromagnetic force is generated around the armature. The magnetic field induced by the field magnets, along with the electromagnetic force of the armature generated by the DC power supply, interacts to cause the armature and the connected axle to spin. The DC power supply can be as simple as a battery, or as complex as a Pulse-Width Modulated (PWM) driving signal, described below. The commutator is an electrical switch that switches the direction of the current between the armature and DC power supply, and therefore also switches the polarity of the electromagnetic force induced. The brushes work with the commutator to reverse the direction of current appropriately. Every time the commutator comes in contact with the brushes, the direction of the current is changed. A steady torque is produced when the current reverses directions in the coils of the armature at the correct times.

Stepper motors differ significantly from brushed DC motors. The stepper motor is a brushless synchronous motor that divides its full rotations into multiple steps. Stepper motors have electromagnets arranged around an armature. The armature in the stepper motor is a permanent magnet. The electromagnets surrounding the armature must be turned on at precisely the correct moment to cause the motor to spin. The electromagnets must be energized using some sort of control circuit, typically a microcontroller, because the timing of the signals is crucial to correct stepper motor operation. The speed of the motor is directly proportional to the average of the supply voltage.

Varying the duty cycle, the average value of the motor voltage can be varied. Varying the duty cycle to achieve a desired average voltage is known as Pulse-Width Modulation (PWM). PWM regulates the current sent to the electromagnets, and therefore the strength of the electromagnetic force generated.

Electric motors have a dynamic relationship between torque and load. Motor control becomes complicated when the load is unknown, because load affects the back electromagnetic force (emf). This back emf affects the amount of current supplied for a given driving signal, which affects the torque. Torque, in turn, also affects the back emf. To allow for precise control of the motor in the presence of dynamic loads, motor control algorithms need to be implemented. Our project handles one particular implementation of a motor control algorithm using a microcontroller family known as the MSP430, produced by Texas Instruments.

PROJECT BACKGROUND

Our sponsor, Texas Instruments (TI), is planning on introducing a new revision (rev. F) of the DRV8412 motor driver card. The DRV8412 provides a large amount of functionality for the purposes of driving a motor. The DRV8412 has on-board DACs, which calculate the equivalent DC voltage of Pulse-Width Modulated (PWM) signals, PWM amplifiers, which amplify supplied signals to levels sufficient to drive motors, and current sensors, for advanced motor control utilizing feedback. However, the DRV8412 does not have any on-board control capability.

TI wants the DRV8412 to be released with several motor control cards using a variety of microcontrollers to increase demand for the DRV8412. The new revision of the DRV8412 will utilize a 100-pin Dual In-line Memory Module (DIMM100) interface for communication between the DRV8412 and a motor control card. These motor control cards should both be able to provide simple driving signals to the motor as well as implement robust control of motor speed and torque in the presence of dynamic loads.

TI has already designed a C2000 motor control card. The C2000 microcontroller family consists of high performance 32-bit microcontrollers. These microcontrollers can be used in a variety of applications from digital motor control to power line communications. TI's C2000 motor control card uses the Piccolo C2000F28035 chip. The Piccolo C2000F28035 chip has 16 ADCs, which are connected to the DRV8412's current sensors. This C2000 motor control card can be programmed through Code Composer Studio, using both Assembly and C. TI has already implemented software for the simultaneous spinning and control of two brushed DC motors and the spinning and control of one stepper motor.

DESIGN SPECFICATIONS

The customer, Texas Instruments (TI), has requested that we redesign an existing card, the C2000 motor control card. The redesign would replace the Piccolo C2000F28035 microcontroller with a microcontroller from the MSP430 family of microcontroller. The MSP430 microcontroller family is a set of ultra-low power consumption 16-bit microcontrollers. This MSP430 motor control card should match the C2000 motor control card exactly, both in functionality and performance. This constraint creates several explicit specifications for our project. For example, the MSP430 motor control card must use a DIMM100 interface to communicate with the DRV8412 motor control card.

Once designed, we should also provide several software libraries for the MSP430 motor control card. This code should function exactly like the C2000 motor control card software libraries provided us. Functionality will include both spinning and control of brushed DC motors and stepper motors.

These software libraries must also contain appropriate software abstractions and modality, within the practices of conventional object-oriented programming. Not only must this code be extensible, it must also contain sufficient software interfaces to utilize a Graphical User Interface (GUI) when the software becomes available. This GUI must be able to display motor RPM, current, temperature, &c. We will have to follow general software programming processes closely to be able to successfully to interface with software that does not currently exist, is coded by a different company, and isn't specifically designed for our software libraries. TI also requires that we conform closely to TI coding standards.

The project also requires several deliverables. These include:

- Schematic for the MSP430 motor control card design.
- PCB layout for the MSP430 motor control card design.
- Full documentation on the MSP430 motor control card's performance test results.

- Hardware Guide for the MSP430 motor control card.
- Brushed DC Motor System Guide for the MSP430 motor control card.
- Stepper Motor System Guide for the MSP430 motor control card.
- Commented source code conforming to TI coding standards.

FAST DIAGRAM

The Functional Analysis System Technique (FAST) diagram is a methodological way to analyze design projects. The structure of the FAST diagram is intentionally simple and nontechnical. The diagram methodology is meant to ensure that the Voice of Customer does not become obscured by design constraints and engineering problems. Our diagram follows this methodology and exposes the motivations for each part of our project:

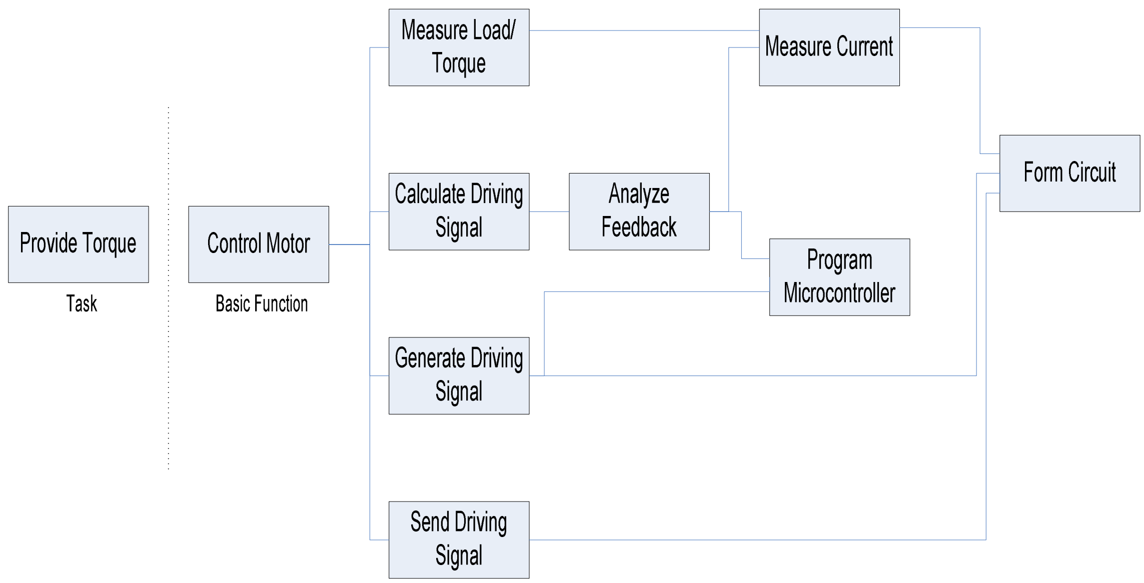

Figure 1: FAST Diagram

CONCEPTUAL DIAGRAM

Once we were given our design specifications, our first decision was to deconstruct design issue into its fundamental constituents. This provided a highly-abstracted system level model of our project, and made particular design decisions we encountered in the future more clear. Our model is pictured below:

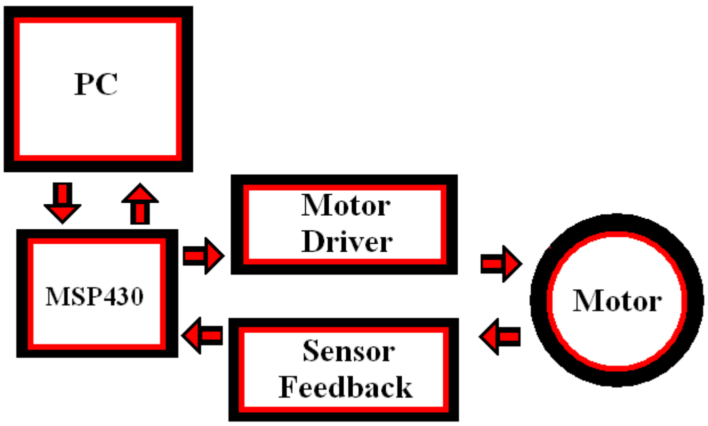

Figure 2: Block Diagram of a Motor Control System

The PC-MSP430 interface will be done through USB-based communication. This communication will use the UART protocol, as the C2000 motor control card did. The interface will be responsible for programming the MSP430 as well as providing information to the PC for a future Graphical User Interface (GUI).

This MSP430 motor control card will also have to interface with the DRV8412 motor driver card. This is pictured above as two blocks: motor driver and sensor feedback. These two different features of the DRV8412 emphasize the fact that the MSP430 motor control card is both analyzes inputs and generates output.

PROPOSED DESIGN SOLUTION

The design solution we propose is to utilize the MSP430F5435 microcontroller. This microcontroller was chosen over other MSP430 microcontrollers primarily for two reasons. First, the MSP430F5435 microcontroller has the same pin package (80-pin Low-profile Quad Flat Package (LQFP)) as the Piccolo C2000F28035 microcontroller used in the pre-existing design from TI. Choosing a microcontroller with the same pin package will minimize the redesign required, as well as ensure that our MSP430 provides sufficient functionality in terms of ability to interact with peripheral hardware.

There were two MSP430s with an 80-LQFP pin-package. The reason we chose the MSP430F5435 over MSP430F5437 was lower cost. The MSP430F5437 has less flash memory, but a close examination of the existing C2000 software as well as the specifications for functionality of the desired MSP430 motor control card led us to conclude that additional flash memory was an unnecessary cost for our final design.

Although the MSP430 we chose has the same pin package as the C2000 in the existing design, the actual footprint of the two chips are significantly different. This required us to map C2000 pins to MSP430 pins of equivalent function. Several design decisions had to be made here, as the functionality of the two chips are not exactly the same. For example, the MSP430 has 12 ADCs (two of which must be used for reference voltage) while the C2000 has 16 ADCs. The DRV8412 has an interface for 14 ADCs, so design decisions had to be made as to which signals from the DRV8412 will be sensed by the MSP430. The design decision was made based upon the pre-existing code and a close examination of the DRV8412 motor control card.

Our design specifications require us to redesign and fabricate the PCB board design we were given. We performed our redesign in PADS Layout, which was the format which TI originally designed the C2000 motor control card. This required access to PADS Layout, and familiarization with the PADS technology. Fortunately, there is a PADS Evaluation Version, which is a 30-day fully-functional trial. For our design project, the PADS Evaluation Version was sufficient.

We also propose to integrate PCB fabrication and assembly. Unfortunately, we could not use the same company that TI used in the production of the C2000 motor control card. TI's PCB fabrication company, Gorilla Circuits, has a minimum lot size of 30 PCBs, which would cost $1200. This exceeds both our needs and our budget. We also compared several other companies, including ExpressPCB and Mini Micro Stencil. Eventually, we resolved to use Hughes Circuits, Inc. Hughes Circuits is able to both fabricate PCBs as well as surface mount chips. The deliverables required for the PCB process include the necessary Gerber files, a schematic, assembly drawings, a Bill of Materials, and a footprint data sheet for every chip on the PCB. These will need to be produced as quickly as possible, so we will have time for any possible required hardware redesigns.

Once the hardware aspect of our project is complete, we also propose to design software for both simple motor spinning as well as advanced motor load/torque compensation. The latter will be accomplished with a PID regulator and current feedback. There was functionally similar code provided with the C2000; however, the MSP430 architecture is significantly different, and code from one microcontroller family cannot easily be translated into another. For example, the C2000 code makes heavy use of the IQMath library, which is a high-precision fixed-point arithmetic library. The MSP430 does not support the IQMath library; we are currently looking through MSP430 libraries to determine the best substitute that can provide the precision as well as calculation speed necessary for our digital motor control application.

The code creation will be done in C, using the Code Composer Studio (CCS) Integrated Development Environment (IDE). CCS is an IDE specifically designed for TI's microcontrollers, and features robust functionality that allow for live variable manipulation and debugging. As stated in the design specifications, the code that we write must also conform to TI's software conventions. Examples of this are available in TI ControlSUITE.

PROJECT MANAGEMENT

Our team has been tasked with doing a large multitude of tasks, both technical and nontechnical. To facilitate our success and ensure that deliverables are finished by their respective deadlines, both technical and nontechnical responsibilities were assigned. This compartmentalization of a large project in smaller tasks allowed for efficient specialization. Our team consists of Roy Dong, Micajah Worden, Andrew Kleeves, Mark Barnhill, and Dave Seaton. The following tables break down both the technical and non-technical roles performed by each team member:

Team Member	**Technical Role**
Roy Dong	Hardware/Software Interface
Micajah Worden	Hardware/PCB Design
Andrew Kleeves	Motor Control Algorithms
Mark Barnhill	Microcontroller Programming
Dave Seaton	DC Motor Operation

Figure 3: Technical Roles

Team Member	**Non-technical Role**
Roy Dong	Presentation Preparation
Micajah Worden	Lab Coordinator
Andrew Kleeves	Web Coordinator
Mark Barnhill	Documentation Preparation
Dave Seaton	Management

Figure 4: Non-Technical Roles

BUDGET

One of the constraints placed upon our team was our budget. Our proposed design solution had to spend less than $500. However, parts provided by our sponsor, Texas Instruments, do not need to be deducted from our budget. Below is our estimated cost of the materials necessary to create the PCB board[1]:

Component	**Cost**
MSP430 Microcontroller	$9.46
Resistors	$2.88
Capacitors	$8.86
ICs	$22.40
Miscellaneous	$10.71
PCB Fabrication and Assembly	TBD[1]
Total	$54.31 + PCB Cost

Figure 5: Design Team 4 Overall Budget

An in-depth Bill of Materials is available on our website, as well. This details the specific components that incur costs in each of the categories above.

In terms of development environment, no costs were incurred. We were able to use PADS Evaluation Version for our project, which was provided with a 30-day license by Mentor Graphics. Also, Code Composer Studio is a free IDE provided by Texas Instruments.

[1] At the time of writing, we do not yet have a price quote from Hughes Circuits, Inc. Several necessary deliverables, including Gerber files and a schematic, were necessary to receive an accurate price quote.

REFERENCES

HowStuffWorks. “How Electric Motors Work” Accessed 10/21/10. http://www.howstuffworks.com/motor.htm

Strangas, E.G. (2007) *Notes for an Introductory Course on Electrical Machines and Drives*.

TI.com. “C2000 High Performance 32-bit Microcontroller”. Accessed 10/21/10. http://focus.ti.com/mcu/docs/mcuprodoverview.tsp?sectionId=95&tabId=1531&familyId=916&DCMP=C2000&HQS=Other+BA+c2000_bhp

TI.com. “MSP430 Microcontroller (MCU), Low Power Mixed Signal Processors”. Accessed 10/21/10. http://focus.ti.com/mcu/docs/mcuprodoverview.tsp?sectionId=95&tabId=140&familyId=342