DESIGN OF A THREE-PHASE BRUSHLESS DC MOTOR CONTROL SYSTEM

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Industrial Engineering

by

Peyton Ulrich

June 2021

COMMITTEE MEMBERSHIP

TITLE:	Design of a Three-Phase Brushless DC Motor Control System
AUTHOR:	Peyton Ulrich
DATE SUBMITTED:	June 2021
COMMITTEE CHAIR:	Jianbiao Pan, Ph.D. Professor of Industrial and Manufacturing Engineering
COMMITTEE MEMBER:	Xuan Wang, Ph.D. Associate Professor of Industrial and Manufacturing Engineering
COMMITTEE MEMBER:	Siyuan Xing, Ph.D. Assistant Professor of Mechanical Engineering

ABSTRACT

Design of a Three-Phase Brushless DC Motor Control System

Peyton Ulrich

In the past several decades, the Brushless DC (BLDC) motor has seen increased usage due to several distinct advantages over its brushed counterpart, including higher performance, increased reliability, and minimal maintenance requirements. However, the electronic commutation system of the BLDC motor creates the need for an accompanying electronic motor control system of increased complexity, adding to the overall cost of the BLDC motor and motor control system. As such, continued research and exploration in the area of BLDC motor control is necessary to continue to reduce the cost of BLDC motors and their corresponding motor control systems. This project focuses on the design of a motor control system for a Three-Phase Brushless DC Motor.

A printed circuit board was designed for use in Three-Phase BLDC motor control and the design process was documented within this report. Due to an international IC shortage at the time of this project, fabrication was unable to be completed, however fabrication plans and cost estimation is included herein. Preliminary software modifications were tested to the extent possible with an off-the-shelf evaluation board, and future software modifications were outlined. Description of the hardware design and software development of this system is included in this report, as well as analysis of this system for future design, fabrication, and testing.

ACKNOWLEDGMENTS

Firstly, I would like to thank my advisor, Dr. John Pan, for his constant support, advice and mentorship not only throughout the process of completing this thesis, but throughout my entire time at Cal Poly. Additionally, I am very thankful to Dr. Wang and Dr. Xing for agreeing to take time out of their busy schedules to serve on my committee. I would also like to thank Michael Derrenbacher for his advice and expertise throughout the PCB design process. Finally, I would like to thank my friends and family, and especially my parents, for supporting me throughout my five years at Cal Poly.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Since its invention in the late 19th-century, the electric motor has continued to be widely used throughout society in a variety of applications across commercial and industrial settings [30]. The electric motor, which serves the primary purpose of converting electrical energy into mechanical energy, comes in many different forms that utilize slightly different technology [41]. This report focuses particularly on DC motors, meaning motors that are powered from a direct current (DC) power source.

1.1 Background

The brushless DC motor (BLDC) differs from the conventional brushed DC motor in that it uses an electronic commutation system as opposed to a mechanical commutation system. The use of electronic commutation comes with several key advantages in terms of reliability and performance, however it also is met with key trade-offs in terms of cost and complexity.

Figure 1.1 compares the brushed and brushless DC motor through simplified pictorial representations. In both brushed and brushless DC motors, the motor can be broken down into two main components: the rotor and stator. As the names imply, the rotor is the rotating element of the motor while the stator is the stationary element of the motor. In the brushed DC motor, current is carried to coil windings of the rotor via physical brushes, generating a rotor magnetic field that varies in direction depending on the orientation of the rotor and which coil windings are presently in contact with the brushes [4]. Meanwhile, a permanent magnet affixed to the stator generates a

permanent stator magnetic field [4]. As the rotor rotates, the contacts between the brushes and commutators are alternated, resulting in the rotor magnetic field changing in direction to cause continuous rotation of the motor. In the brushless DC motor, a permanent magnet is instead affixed to the rotor, generating a permanent magnetic field at the rotor, while coil windings on the stator generate the stator magnetic field [4]. Rather than relying on the physical contacts for motor commutation, BLDC motors use an electronic commutation system that alternates the direction of the stator magnetic field. By varying which coil windings are provided with electrical current, the direction of the stator magnetic field can be changed, resulting in continuous rotation of the motor.

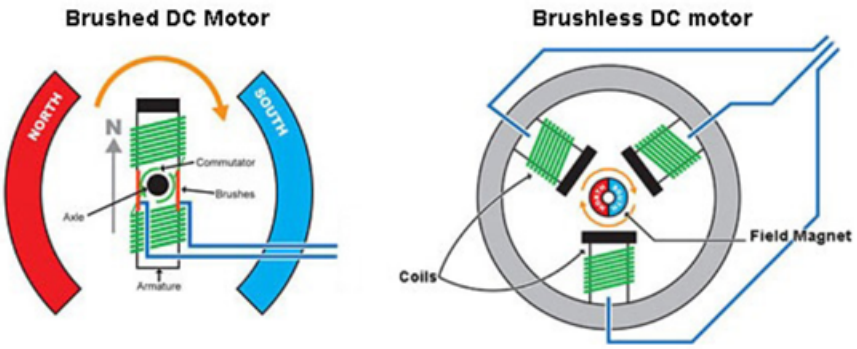


Figure 1.1: Comparison of Brushed vs. Brushless DC Motor
[10]

As the BLDC motor eliminates the need for mechanical contacts in the form of brushes, it comes with several key advantages over brushed DC motors. Namely, the need for continual maintenance and replacement of the brushes is eliminated, resulting in lower maintenance costs and longer lifetime of the BLDC motor [41]. Furthermore, replacing the mechanical contacts of the brushed DC motor with electrical switches in the BLDC motor results in a lower voltage drop and higher motor efficiency and performance. Finally, the brushes in the brushed DC motor produce friction, resulting in a less desirable torque output of the motor [41].

While BLDC motors are desirable due to lower maintenance and higher performance as compared to their brushed counterparts, they also come with key tradeoffs. To control this electronic commutation system, an electronic motor control system is needed to drive a BLDC motor as well as accompanying software. This electronic commutation system creates an additional cost component in the DC motor system, resulting in the cost of the BLDC motor and controller being higher than that of the brushed DC motor. Furthermore, this electronic commutation system results in a more complex design and implementation process for electromechanical systems using BLDC motors, further increasing the cost of such systems. A detailed comparison of the brushed and brushless DC motor, as well as the associated hardware and software components necessary for BLDC motor control follows in Section 2.1.

1.2 Statement of Problem

While BLDC motors provide key benefits over brushed DC motors, the tradeoffs include increased cost and complexity, necessitating continued research and development in the area. This project focuses on the design of a BLDC motor control system capable of controling a three-phase BLDC motor at 12 or 24 V supply voltage. In this project, focus will be put on the minimization of system complexity and ease of manufacturability, aiming to produce a low-cost system. This project is intended to serve as a resource for future design in the area of BLDC motor control.

Chapter 2

LITERATURE REVIEW

2.1 Detailed Comparison of Brushed versus Brushless DC Motors

Functionally, an electric motor is a system designed to convert electrical energy into mechanical energy [41]. To do this, electromagnetic principles are applied to produce rotation of one element of the motor, referred to as the rotor, relative to the stationary body of the motor, referred to as the stator [4]. Magnetic fields, produced either by permanent magnets or a coil, are located at each of the rotor and stator and are used to produce rotation of the motor [15]. As a result of the attractive force between the two magnetic fields, rotation of the rotor is produced relative to the stator [4]. To continually rotate the rotor, one of the magnetic fields must be continually switched such that there is a continuous attractive force driving the motor. For this reason, a system to switch the direction of one of the two magnetic fields is necessary to continually drive the motor, referred to as a commutation system [41]. This commutation system is the main difference between the brushed DC motor and brushless DC motor. In this section of the literature review, the two main types of DC motors, Brushed DC Motors, and Brushless DC Motors, will be discussed, as well as their respective advantages and disadvantages.

2.1.1 Brushed DC Motor

In the Brushed DC Motor, rotation of the motor is driven by a mechanical commutation system [41]. In this commutation system, physical contacts, referred to as "brushes," carry current to the coil windings of the rotor, creating a magnetic field

on the rotor as electrical current passes through the coils [4]. Permanent magnets are affixed to the stator of the motor, creating a permanent magnetic field on the stator [4].

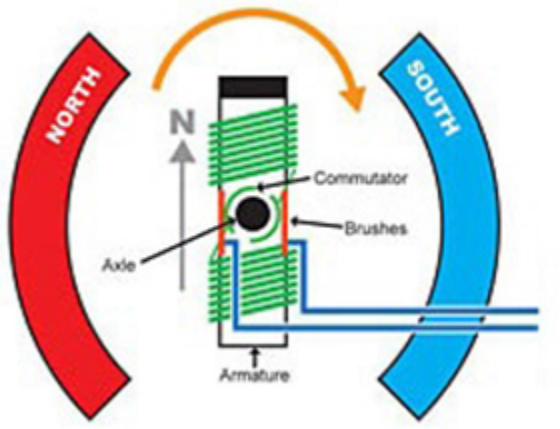


Figure 2.1: Diagram of Brushed DC Motor
[10]

Figure 2.1 depicts a simplified representation of a brushed DC motor with two brushes. The permanent magnet affixed to the stator of the motor will produce a permanent stator magnetic field, as indicated by the "North" and "South" labels on the diagram [4]. Electrical current passed through the coil winding of the rotor will produce a rotor magnetic field, as indicated by the "N" arrow on the diagram [4]. The attractive force between these magnetic fields will cause the rotor to rotate clockwise. As the rotor rotates, the direction of the rotor magnetic field will rotate as well, while the stator magnetic field remains constant [4]. After every half revolution of the rotor, the commutators will switch which brush they are in contact with, effectively reversing the direction of the rotor magnetic field by 180 degrees. This commutation produces an AC waveform at the coil windings and will allow the rotor to continuously rotate in one direction. As can be clearly seen from this image, the reversal of direction of the rotor magnetic field is dependent on the physical contact between the brushes

and commutators of the brushed DC motor, hence the classification of the brushed DC motor commutation system as "mechanical" [41].

2.1.2 Brushless DC (BLDC) Motor

Unlike Brushed DC Motors, the Brushless DC (BLDC) Motor uses an electronic commutation system to drive the rotation of the motor [39]. In this system, the magnets of the motor are "inside-out" compared to the brushed DC motor, in that the rotor utilizes one (or more) permanent magnet(s), while a set of coils on the stator produce the stator magnetic field [4].

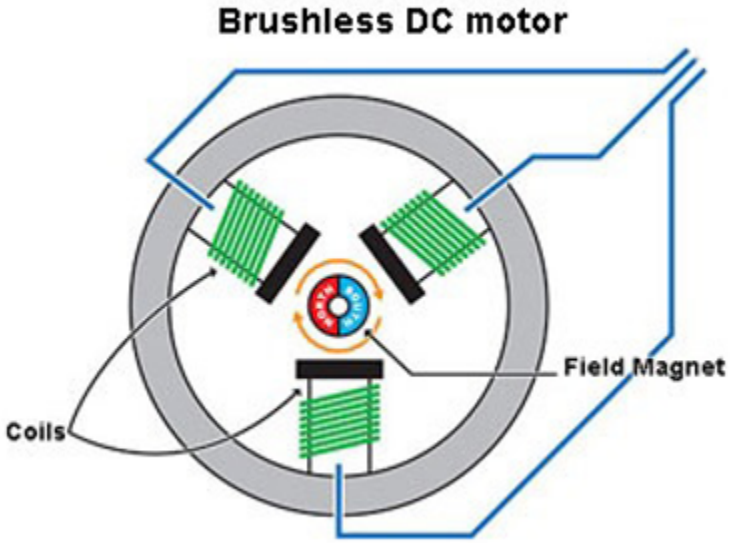


Figure 2.2: Diagram of a Three-Phase Brushless DC Motor
[10]

Figure 2.2 depicts a simplified representation of a Three-Phase BLDC motor. As can be seen from this diagram, a permanent magnet is affixed to the rotor, producing the rotor magnetic field, while electrical current driven through two (or more) of the coil windings on the stator, producing the stator magnetic field [4]. By changing

which coils are provided with electrical current, the direction of the stator magnetic field can be changed, driving continuous rotation of the rotor [4]. No mechanical contact is necessary in this system, leading to the classification of the BLDC motor commutation system as “electrical” [41]. In a BLDC motor, the brush/commutator system is replaced with an electronic motor control system that controls which coil windings of the motor are provided with electrical current [12]. Figure 2.2 depicts a ”three-phase” BLDC, meaning that the stator contains three separate coil windings that are independently supplied with electrical current.

2.1.3 Comparison of Brushed vs. Brushless DC Motor

There are several distinct advantages to the brushed DC motor system and mechanical commutation system, including simplicity and cost [41]. Brushed DC Motors require relatively simple motor control electronics and software compared to their brushless counterparts, and as such are relatively simple for design and implementation into an electromechanical system [41].

Due to the physical contact between the commutator and brushes, the brushes wear out over time, and need to be consistently maintained and/or replaced [41]. As a result, maintenance requirements are higher in brushed DC motors, and motor lifetime is reduced [41]. Additionally, due to the relatively higher voltage drop in the mechanical system (as compared to the electrical system of the BLDC), brushed motors have lower power efficiency [41].

Table 2.1: Comparison of Brushed vs. Brushless DC Motors

Brushed DC Motor	Brushless DC Motor
Simple Motor Control	Complex Motor Control
Low Design and Fabrication Cost	Increased Design and Fabrication Cost
Low Efficiency	High Efficiency
High Maintenance	Minimal Maintenance

2.2 Brushless DC Motor Control System Design

The primary disadvantage of BLDC motors, as opposed to brushed DC motors, is the added complexity of controlling the electronic commutation system, and the associated cost that comes along with this system complexity, as discussed in Section 2.1. In the brushed DC motor, a constant DC current can be applied to the brushes, and mechanical commutation system allows for continuous rotation of the motor. In the BLDC motor, this mechanical commutation sequence is replaced by an electrical commutation sequence, instead switching between coil windings to produce an AC waveform at each stator coil. This electrical commutation sequence requires accompanying hardware, typically in the form of a printed circuit board (PCB), and a motor control algorithm, typically defined in software [41].

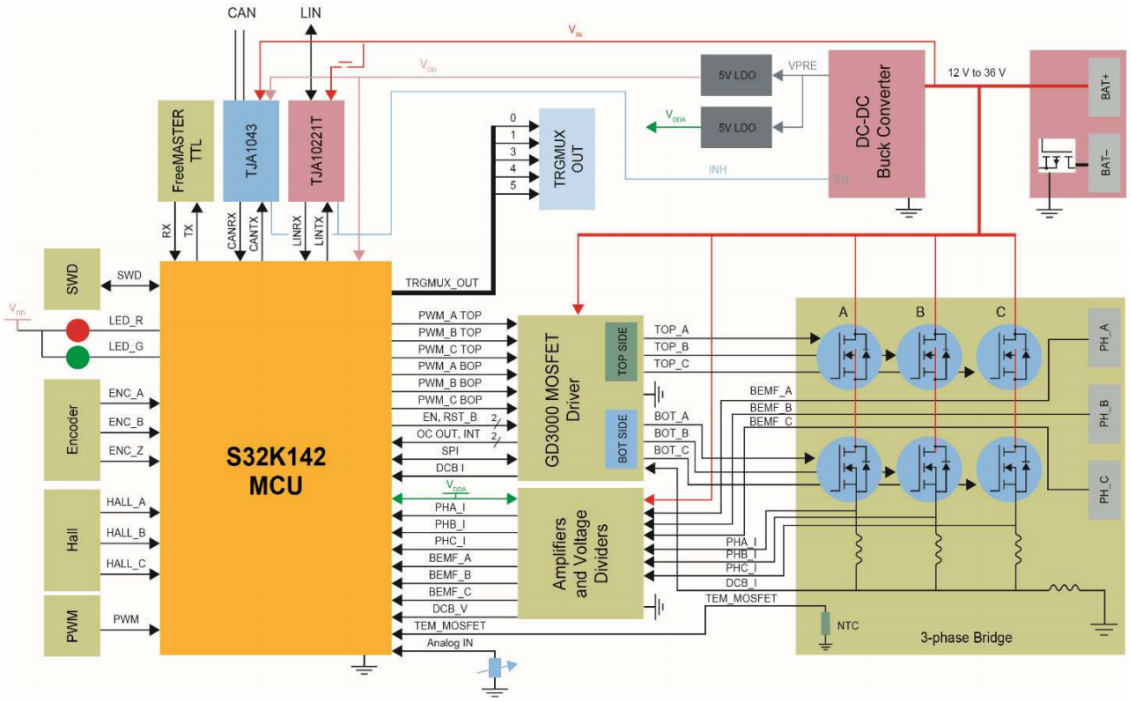


Figure 2.3: Block Diagram of NXP MCSXTE2BK142 Evaluation Board [33]

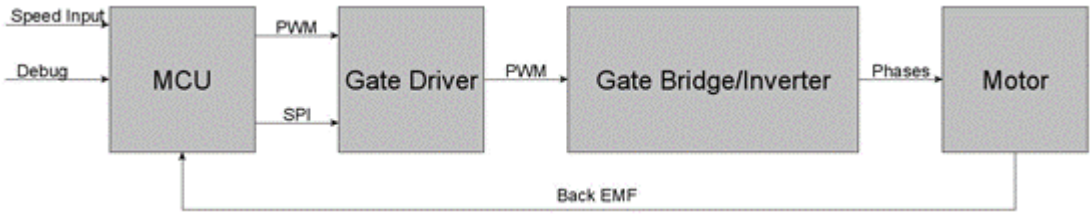


Figure 2.4: Simplified Block Diagram of a BLDC Motor Control System and Motor

Figure 2.3 shows a block diagram of the MCSXTE2BK142 Evaluation Board, designed for use in BLDC motor control applications [33]. This board can be considered representative of the general high-level architecture of the typical BLDC motor control system. Figure 2.4 further simplifies the block diagram to the three core components of the motor control system, along with the motor, as well as the key inputs and outputs from each component. This section focuses on the functionality and purpose

of each of these core components: the microcontroller (MCU), gate driver, and phase inverter circuit.

2.2.1 Microcontroller (MCU)

The microcontroller functions as the central processing element of the motor control system. The MCU processes key control inputs, including the desired motor speed and position feedback from the motor (in the form of sensored or sensorless feedback as discussed in Section 2.4), and outputs control signals to the gate driver controlling the rotation speed and direction of the motor [14]. The MCU is responsible for the processing of the motor control algorithm, using the inputs described in the previous sentence, and producing the PWM signals that control the motor.

While microcontrollers are commonly used for simple motor control systems, digital signal processors (DSP) are often used in "intelligent control systems" due to increased computing and data processing capabilities [39]. Presently, many MCU chips have been designed specifically for BLDC motor control applications, several examples of which are described by Xia [39] and summarized below. ST Microelectronics' ST72141 contains the company's patented back-EMF detection technology, for use in sensorless position feedback [39]. Other MCU's contain similar technology to enable zero-crossing detection for use in back-EMF-based sensorless feedback, such as NXP's S32K1xx series of MCU's [26]. Other MCU's, such as the Siemens C504 contains internal hardware commutation sequence that allow for the processing of position inputs without the need for hardware, reducing system development time but greatly limiting processing capabilities [39].

2.2.2 Gate Driver

The gate driver acts as an intermediate element between the microcontroller and the phase inverter circuit. The gate driver receives PWM signals from the MCU, and outputs an amplified gate drive current to the MOSFETs of the phase inverter circuit. Gate driver chips (also referred to as pre-drivers), are able to provide higher levels of current (and voltage) to the motor than a typical MCU can provide. For example, TI's DRV8343-Q1 and NXP's GD3000 are able to provide a maximum of less than 20 volts, whereas most MCU pins supply only 3.3 or 5 volts [25] [35]. Devices such as the DRV8343-Q1 or GD3000 also integrate multiple gate driver circuits into one IC, enabling the input of six separate PWM signals and simultaneously controlling the switching output of six MOSFET's [25] [35].

2.2.3 Phase Inverter Circuit

The phase inverter circuit is a circuit based around metal oxide semiconductor field effect transistors (MOSFET) designed to serve as a series of switches for each phase of the BLDC motor. In a three-phase BLDC motor control system, six MOSFET's are used to control both the high and low side of each phase. When used in switch mode, the primary goal of the MOSFET in the circuit is to "switch between the highest and lowest resistance states of the device in the shortest possible time" [7].

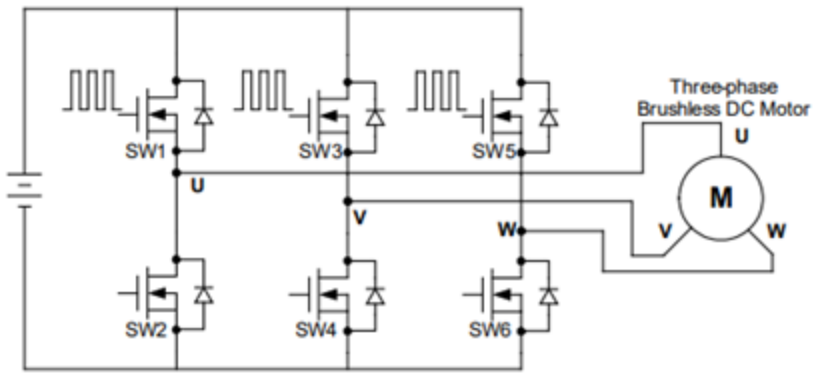


Figure 2.5: Simplified Circuit Diagram of Three-Phase Inverter [41]

Figure 2.5 shows a simplified circuit diagram of a three-phase inverter supplying power to a three-phase BLDC motor. As is seen, PWM signal is typically applied to the gates of the high side and low side MOSFET's of each of the phases. By changing the duty cycle of the PWM signal, the speed and torque of the motor can be varied [41]. When using trapezoidal control (described in Section 2.3), only two of the six MOSFET's of the circuit need to be switched "on" simultaneously [41].

2.3 Brushless DC Motor Control Algorithms

While brushed DC motors are self-commutating, meaning that the mechanical commutation system carries out commutation of the motor without the need for external control, BLDC motors require a motor control algorithm, processed by the MCU, to determine the timing and level of current supplied to each of the phases in the motor [1]. In BLDC motor control, three primary control algorithms are typically used: trapezoidal control, sinusoidal control, and field-oriented control (FOC). All three control algorithms control the commutation sequence of the BLDC motor, and

provide the framework to drive the motor. This section focuses on each of these three control algorithms.

2.3.1 Trapezoidal

Trapezoidal control is the simplest of the three motor control methods discussed. In trapezoidal control, current flows through only two phases of the motor at a time, while the third remains electrically disconnected [1].

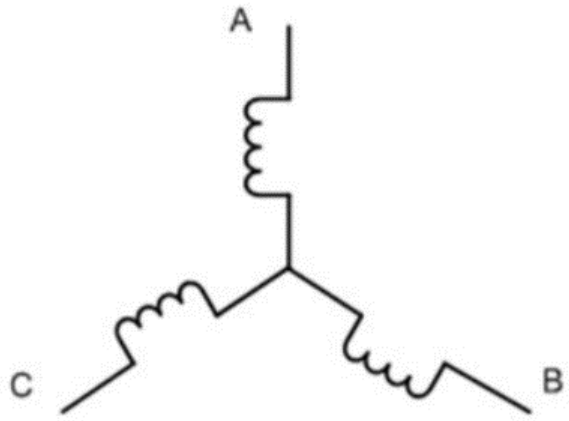


Figure 2.6: Three-Phase BLDC Motor Stator Windings
[13]

As seen in Figure 2.6, the three stator coil windings of the three-phase BLDC motor are connected at the center, meaning that the sum of the currents through each of the phases must be equal to zero. In trapezoidal control, where only two phases are used at any time, this means that the two excited phases must have currents of values +X and -X, respectively. As such, since each of the two phases must have current of equal magnitude, the stator magnetic field may only be directed in six possible directions [13]. Figure 2.7 depicts the six possible directions of the stator magnetic field.

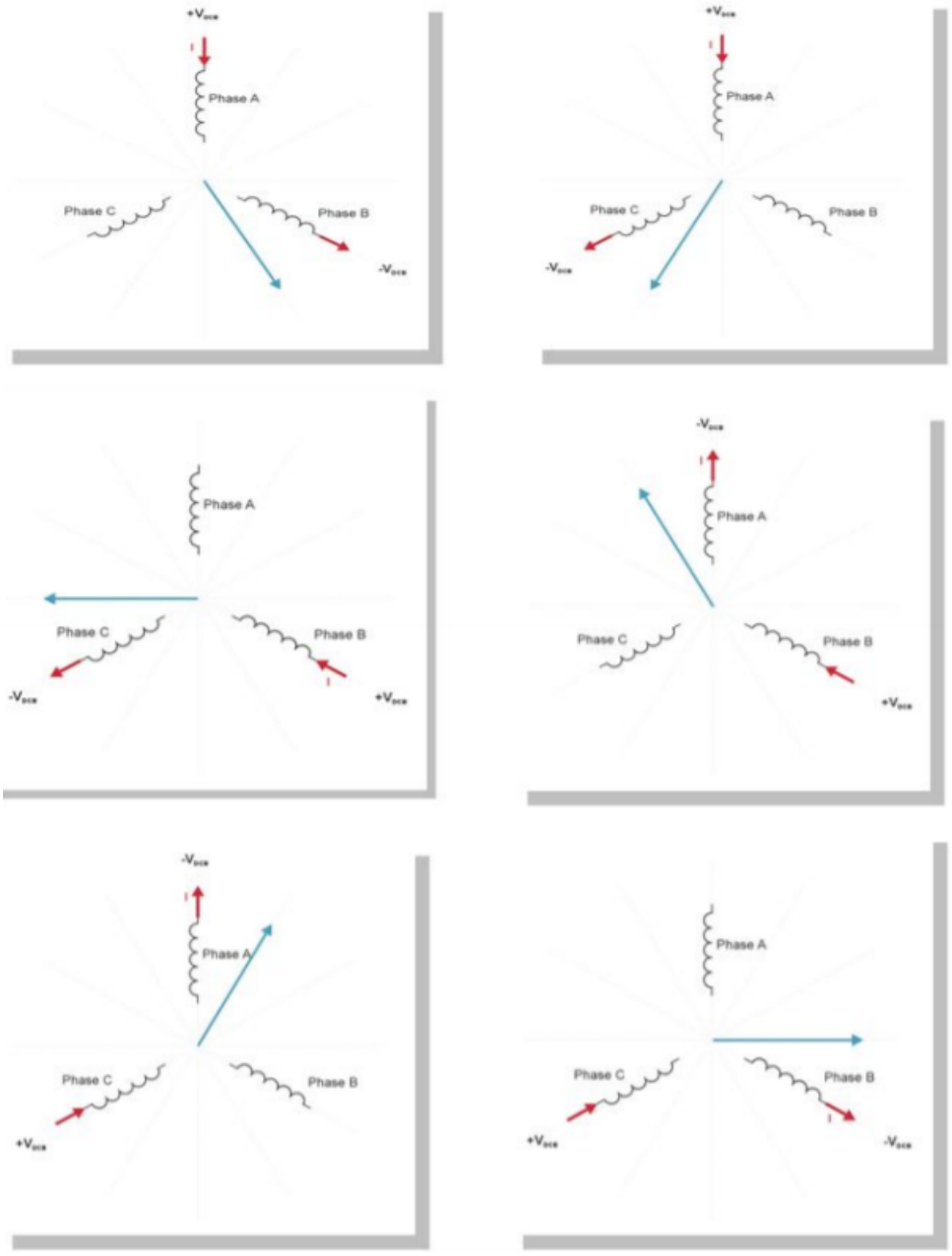


Figure 2.7: Six-Step Commutation Sequence
[13]

In Trapezoidal Control a six-step commutation pattern is followed to systematically alternate the stator magnetic field through each of these six directions, ensuring that the rotor continues to rotate [1]. Figure 2.8 depicts the six-step commutation sequence of the trapezoidal control algorithm. After each 60 degrees of rotation of the rotor, the

coil pairs that are presently excited are changed. This produces a staircase waveform at each phase, where the phase is positive for 120 degrees of rotation, zero for 60 degrees of rotation, negative for 120 degrees of rotation, and zero for 60 degrees of rotation, before repeating [1].

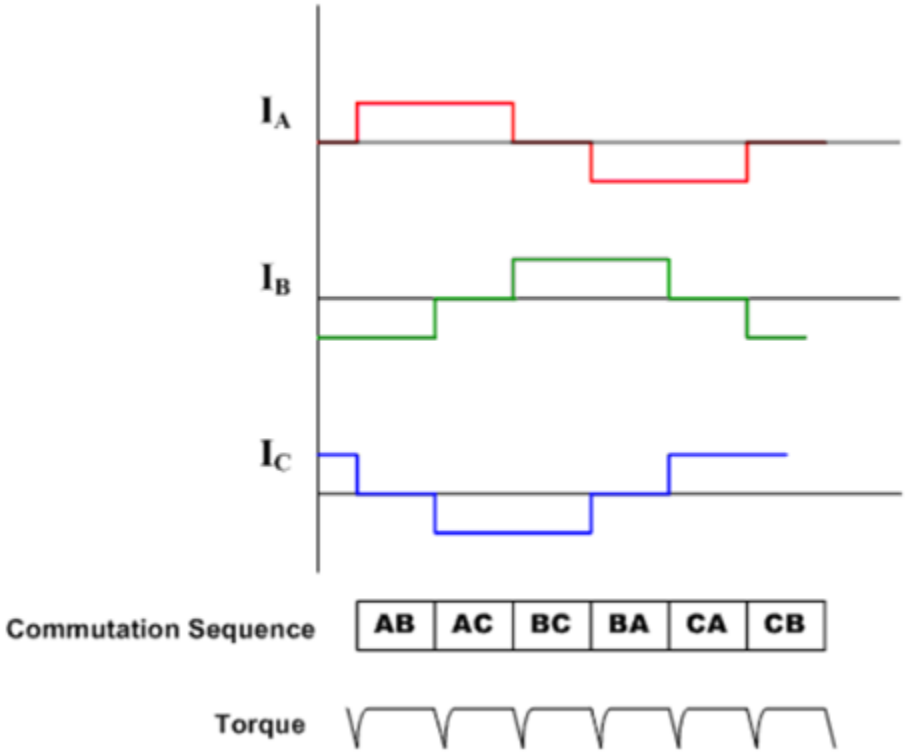


Figure 2.8: Trapezoidal Control Six-Step Commutation Sequence [1]

While the trapezoidal control algorithm is highly popular due to its simplicity, it has a key disadvantage of the torque-ripple effect demonstrated in Figure 2.8 [1]. Due to the abrupt waveform changes, and the fact that the stator magnetic field is not consistently orthogonal to the rotor magnetic field, trapezoidal control does not ensure smooth operation of the motor, nor does it maximize efficiency [22].

2.3.2 Sinusoidal and Field Oriented Control (FOC)

While trapezoidal control is advantageous for its simplicity, it is inadequate for smooth and precise BLDC motor control [1]. Instead, alternative motor control algorithms may be used in which each of the three phases are driven simultaneously, providing a smooth rotation of the stator magnetic field vector [1]. One such example of a control algorithm in which all three phases are used simultaneously is that of field oriented control (FOC) [22].

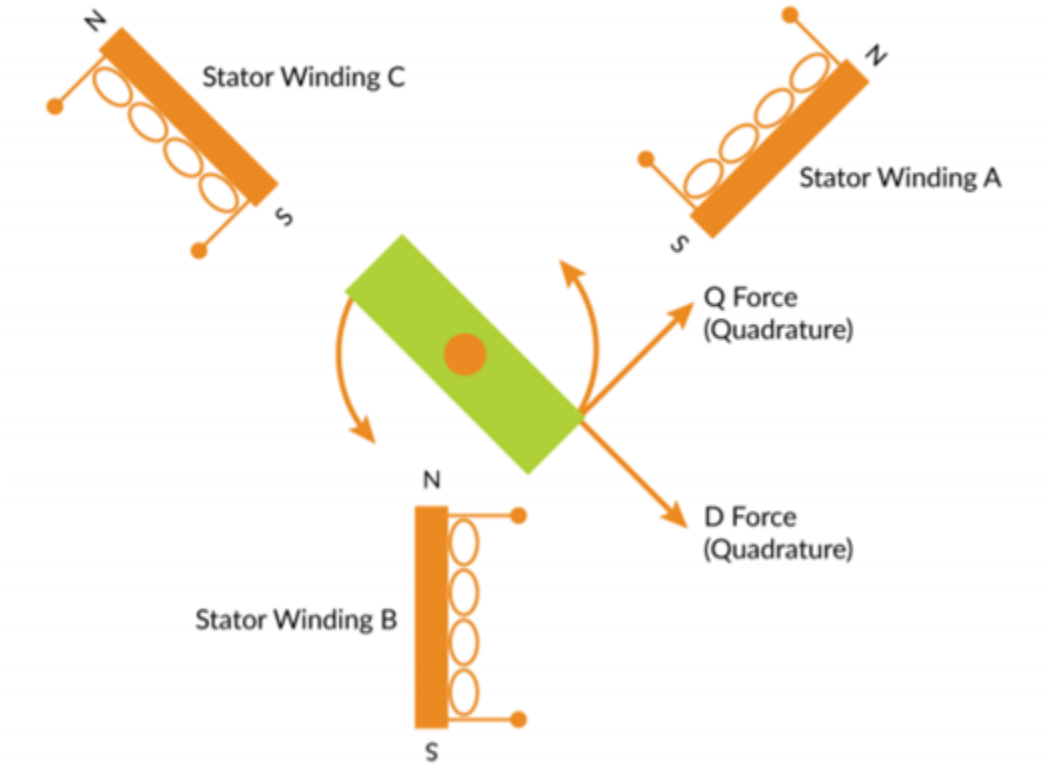


Figure 2.9: Quadrature and Direct Forces
[22]

Figure 2.9 illustrates the important concept behind the FOC algorithm: quadrature and direct forces. As depicted in the figure, direct forces are those that run parallel to the rotor pole axis, while quadrature forces run perpendicular to the rotor pole axis

[22]. If the magnetic field generated by the stator windings runs only along the direct axis, no rotation of the motor will be generated as this implies that the rotor and stator magnetic fields are aligned [22]. Thus, direct forces are not of any use for driving rotation of the motor and should be minimized [22]. This is the central principle of FOC, in which quadrature forces are maximized by simultaneously driving current through all three stator coil windings to create a stator magnetic field orthogonal to the rotor magnetic field [22]. The current driven to each of the three "phases" of the motor is varied smoothly and continuously to maintain the orthogonal position of the stator magnetic field relative to the rotor magnetic field.

Another similar motor control algorithm, sinusoidal commutation, similarly attempts to maximize quadrature forces [1], however differs from FOC in that it is dependent on time and speed whereas FOC is dependent on mathematical transforms [9]. Sinusoidal control effectively minimizes the disadvantages of trapezoidal control by providing smooth and precise control, however, breaks down at higher motor speeds when processing time becomes insufficient [3]. Field oriented control is able to overcome this problem through the transformation of forces into the direct and quadrature axis, removing the time dependency and allowing the motor to operate efficiently at high speeds [3]. The primary disadvantage of FOC, the requirement of high-performance processors due to math-intensive operations, has reduced in impact in recent years as the cost of such high-performance processors has decreased [22].

2.4 Brushless DC Motor Control Feedback Methods

For closed loop control of the BLDC motor, a method for position and speed feedback is necessary for accurate motor control. BLDC motor control feedback methods can be generally classified into two categories: sensored and sensorless [12]. Sensored

methods require the use of a sensor to measure the position and speed of the rotor. Sensorless methods do not use a motor sensor, and instead rely on electrical measurements to determine motor speed and position.

2.4.1 Sensored Methods

While there are several methods of sensored position feedback, Hall-effect sensors remain one of the most common. At a high-level, a Hall-effect sensor is a sensor able to detect when a magnetic field is applied perpendicular to the current flow of the sensor. As an effect of this, when the North pole of the rotor passes the Hall-effect sensor, its output changes to 1, and when the South pole of the rotor passes the sensor, its output changes to 0 [13]. For use in a BLDC motor, three sensors are typically placed in the air gap of the motor, at either 60 or 120 degree increments relative to each other [12] [11]. As the rotor rotates, the outputs of the three Hall-effect sensors change between 0 and 1, making it possible to determine the position of the motor in terms of the six-step commutation sequence.

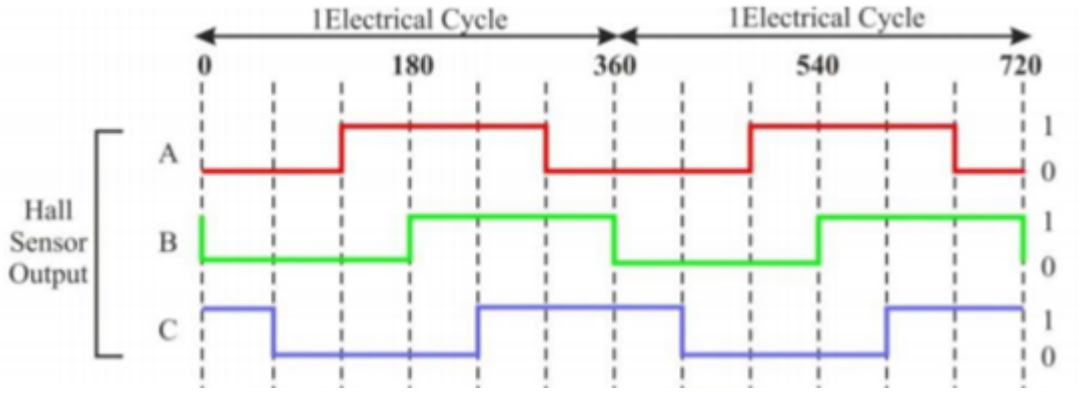


Figure 2.10: Output of Three Hall-Effect Sensors Placed At 60 Degree Increments

[12]

Figure 2.10 shows the output of three hall-effect sensors placed at 60-degree increments over two electrical cycles. Since the sensors are placed at 60-degree increments, the waveforms of the outputs trail each other by 60 degrees. Throughout one electrical cycle, the position of the rotor can be determined based on the six-step commutation sequence [12]. By determining the position of the rotor, the BLDC motor control algorithm can then determine which step of the commutation sequence the motor is in, and thus determine how to properly perform phase commutation [12].

While sensored methods are highly effective for motor control feedback and relatively simple for implementation [11], they do come with several key drawbacks. The need for the internal mounting of the sensors leads to increased size and cost of the motor, as well as increased complexity in the design of the overall system [12]. Furthermore, certain sensors can be temperature sensitive and/or require additional components, limiting the reliability and performance of the motor [12].

2.4.2 Sensorless Methods (Back EMF)

Due to the increased cost and size of BLDC motors with sensored feedback methods, it is often preferable to eliminate the need for position and speed sensors. To do this, alternative methods of motor control feedback using electrical measurements rather than sensors, referred to as sensorless methods, are used [12]. The most popular of these sensorless methods is back electromotive forces (back-EMF) [12].

Figure 2.11 depicts the general phenomenon that enables the use of back-EMF for BLDC motor control. As the rotor rotates, the back-EMF at each coil changes proportionally to the speed of the motor [12]. When the rotor magnetic field crosses either of the phases, the back-EMF of that phase changes its polarity [4]. Typically, back-EMF-based feedback methods are used in control algorithms such as trapezoidal

control, where only two of the three motor phases carry current simultaneously [12], however sensorless control methods may also be used in methods in which current is supplied to all three phases simultaneously, such as sinusoidal and field-oriented control [6].

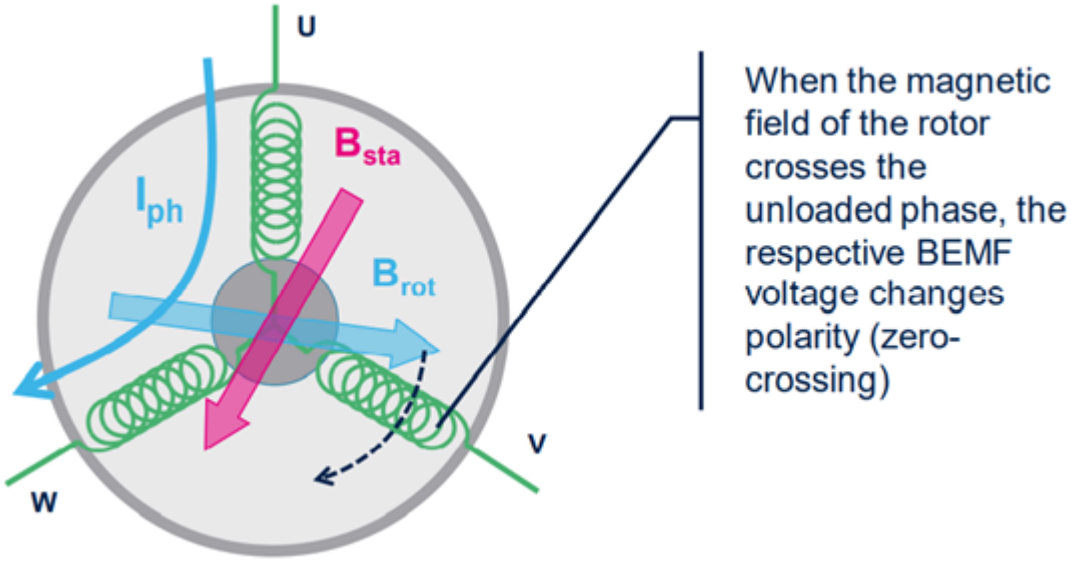


Figure 2.11: Depiction of Back-EMF Zero-Crossing Event
[4]

The driving principle in this scheme, as described by Xia [39] and illustrated in Figure 2.11, is that "if the phase current and the stator flux have the same phase, the rotor position of BLDC motor can be accurately reflected by the change of phase current". In sensorless control, an open-loop starting sequence is often required to determine the initial position of the rotor, at which point back-EMF measurements can be used to accurately estimate the position of the rotor [32]. Additionally, back-EMF-based feedback methods do not work at low speeds since back-EMF is zero at rest and proportional at speed, creating an additional need for open-loop control [21].

2.5 Printed Circuit Board (PCB) Design

The process of designing a printed circuit board (PCB) contains two primary steps: creating the schematic and creating the PCB layout. For successful completion of this project, PCB design guidelines and best-practices must be utilized at each stage of the PCB design process. Many different software programs exist for PCB computer-aided design (CAD), including Autodesk Eagle, Altium, and KiCad. This report focuses on the PCB design workflow for KiCad, the software used in this project, however a similar workflow is used in the majority of PCB CAD software. This section summarizes lessons learned from the review of literature related to PCB design guidelines, standards, and best-practices.

2.5.1 Schematic Design

The first step in designing a custom printed circuit board is to design a schematic, which is often referred to as the equivalent to an engineering drawing in mechanical design. IPC-2612 sets documentation standards for printed circuit board schematics/logic diagrams, including required information for PCB "inspection, hardware realization, software development, and design reuse" [20]. IPC defines a schematic as a diagram that "designates the electrical functions and interconnectivity to be provided by the printed board and its assembly" [20]. Given that the purpose of a PCB is to provide mechanical support to electronic components and electrical interconnections between components, a schematic is provided to depict which components will be present on the board and how they will be connected [37].

To create a schematic, component symbols are selected from a library and placed on a circuit [37]. The symbols are then interconnected by traces, representing the electrical

connection between pins [37]. A completed schematic will depict all components placed on a board, as well as the electrical interconnections between them. Additional information that may be shown on a schematic includes test point allocation, current and voltage requirements, shielding of traces, noise suppression, restrictions of heat transmission, and grounding and power requirements [20] [37].

2.5.2 PCB Layout

After creating a schematic diagram depicting the interconnections of the components, the designer must create the PCB layout, which depicts the physical layout and construction of the printed circuit board. This process can be broken down into two key steps: component placement and routing, with an important prerequisite step of footprint design and verification [37].

First, the designer must complete component placement, showing the placement of each component on the printed circuit board. Many CAD software contain automatic placement systems to attempt to optimize component placement [37]. There are several objectives in component placement, namely, to minimize trace length and consider other constraints of the components themselves [37]. Placement is done through moving, and rotating components and is typically done in more than one phase [37]. Prior to placement of the components, footprints must be defined for each component. These footprints show the physical footprint of the component on the board, and the required surface mount pads, and plated/non-plated through-holes that are required to mount the component.

Second, the designer must route all the components, which is to complete the process of drawing conductive traces between each interconnected component. Several constraints must be considered during this process, including maximum connection

lengths, shielding of a signal, or the preferred routing layer of a signal [37]. During this stage, the designer may decide to include vias, route traces at multiple layers of the board, or widen traces for optimal current flow [37]. IPC-2221 addresses common design parameters and sets standards as to the design of printed circuit boards [17].

After design of the PCB, including placement and routing, the board must be sent to a manufacturer for fabrication. Prior to being sent to the manufacturer, board designs typically undergo design verification, a process in which the design is verified versus the manufacturer's design rules [37]. The board is then typically sent in one or multiple CAD files, typically in the form of .brd or Gerber files.

2.6 PCBA Manufacturing and DFM Considerations

After the design of a printed circuit board, it must be sent to a manufacturer for fabrication and assembly. This section focuses on the main stages in the PCB manufacturing process and key design for manufacturability (DFM) considerations at each stage.

2.6.1 PCB Fabrication

A printed circuit board (PCB, also called printed wiring board or PWB), is defined by Tummala as "a composite of organic and inorganic materials with external and internal wiring, allowing electronic components to be electrically interconnected and mechanically supported" [37]. Essentially, the function of a PCB is to provide power to all components, carry signals between components, and conduct heat away from the components when necessary [37]. In industry, manufacturing of the printed circuit board itself is often referred to as "PCB Fabrication," whereas the process of placing

the components onto the board is known as "PCB Assembly." These terms will be used throughout this section.

All printed circuit boards include one or more layers of conductive materials (typically copper), interconnected by vias, and separated by an insulator epoxy-glass [37]. Printed circuit boards may be classified by several factors, including the rigidity of the board and number of conductive layers [37]. Historically, most PCB's have been made of a rigid insulating material, however recently flexible boards have become desirable in many applications.

The number of layers of the board is one of the primary contributors to the complexity of the board. Single layer boards contain only a single layer of conductive material, and are typically used in very simple applications [37]. Double-sided PCB's are the most common, including a conductive layer on both sides of the board and allowing components to be placed on both the top and bottom of the board. Multi-layer boards can range from 4-32 conductive layers and are used in applications where high component density is necessary [37].

The PCB fabrication process is completed over the course of several additive and subtractive steps. The process can be generally described as adding material one layer at a time, and then etching away the material using photoresist and imaging processes [37]. After fabrication of the layers of the board, holes are drilled through the board, through holes are plated, and solder mask and silk screen is printed onto the board [37].

There are several key attributes of the PCB affecting PCB fabrication that must be addressed during the design process. Substrate materials must be selected based upon the desired rigidity and specifications of the board. The layer structure must be considered, specifically how many layers must be used in the PCB. In general,

a minimal number of layers results in a less expensive fabrication process, however more complex circuit designs may necessitate a multi-layer board design [37]. The via technology used in the PCB must be determined during the design process, including whether any blind or buried vias may be necessary [37].

PCB manufacturers will typically list several key manufacturing constraints that should be considered during the design process. For example, PCBWay lists several key constraints in their PCB capabilities, including a minimum trace width of 0.1mm, a minimum conductive spacing of 0.1mm, and a minimum drill size of 0.2mm [29]. In an effective PCB design, these constraints should be considered to ensure manufacturability of the board. Furthermore, while a PCB manufacturer may list a specified minimum or maximum parameter as within their capabilities, in most cases this will increase the cost of the manufacturing process and in some cases will require a modified process or processing equipment. When possible, an effective design should not include design components on the limits of a manufacturers capabilities so as to reduce manufacturing cost and increase manufacturability. For example, for minimized cost PCBWay suggests a modified minimum trace width and conductor spacing of 0.15mm [29].

2.6.2 PCB Assembly

Printed circuit board assembly is described by Tummala as "the process of building functional electronic systems from individual electrical components" [37]. This process primarily involves mounting and soldering electrical components onto the finished PCB, however it can also contain other assembly methods outside of soldering [37]. The final product after assembly is referred to as a printed circuit board assembly (PCBA), printed wiring board assembly (PWBA), or printed wiring assembly (PWA) [37].

PCB assembly processes can be generally classified as either surface mount assembly/ technology (SMA/SMT) or through-hole technology (THT). During surface mount assembly, components are placed on the surface of the board, whereas in through-hole assembly the leads of the components are inserted through holes on the PCB [37]. Surface mount technology has become prevalent in industry as it helps achieve the goal of size reduction of electronic systems, as well as increasing the ability to complete the assembly process via automated methods [37]. By the late 1990's over 80% of electronics manufacturing was done by surface mount assembly [37].

Surface mount assembly typically follows a linear process on a highly automated assembly line. First, the bare PCB enters a solder paste printing machine in which solder paste is deposited onto the copper pads of the PCB by screen printing through a stencil [37]. The PCB then enters a component assembly machine known as a "pick and place" machine, in which the components are placed onto the PCB with high precision [37]. The board then passes through a reflow oven at a specified temperature profile, known as a "reflow profile," to melt the solder paste and form robust solder joints connecting the components to the PCB [37]. Many SMT assembly lines utilize automated in-process inspection systems including solder paste inspection (SPI), automated optical inspection (AOI), and x-ray inspection.

When considering the PCB assembly process during the PCB design process, there are several key items to address. SMT components can generally be considered favorable as compared to through-hole components for two primary reasons: size reduction and assembly automation. Surface mount components require space on only one side of the board, whereas through-hole components require space on both sides of the board. Additionally, plated through-holes pass through all layers of the board, complicating PCB layout and increasing the size of the system [37]. SMT components also tend to have a reduced size and pitch as compared to through-hole components,

further leading to size reduction of the system [37]. Furthermore, the SMT assembly process lends itself to automation much more than the through-hole assembly process, reducing the cost of assembly [37].

There are several reasons that through-hole components may still be preferable over SMT components. Through-hole insertion provides stronger mechanical fastening to the board than SMT assembly, which is desirable for components and systems experiencing large dynamic forces and requiring greater mechanical robustness [37]. Additionally, when large amounts of current must be conducted through the component leads, the use of through-hole components may also be necessary [37].

Beyond component selection, footprint design is also a critical step in the design process that must be addressed to result in successful PCB assembly. Component footprints must be designed such as to maximize solderability and minimize footprint area. Electronic device manufacturers typically provide recommendations on footprint design for their components. IPC standard IPC-SM-782A is particularly useful for providing industry standard footprints for common packages [18].

Chapter 3

HARDWARE DESIGN

As covered in Chapter 2, complex electronic circuitry is necessary for the successful operation of the BLDC motor. In this project, a printed circuit board was designed to function as a three-phase BLDC motor control system based off the architecture discussed in Section 2.2 and referenced in Figure 3.1 below. This section is intended to illustrate the hardware design process used in this project and to serve as a resource for the designers of future similar systems.

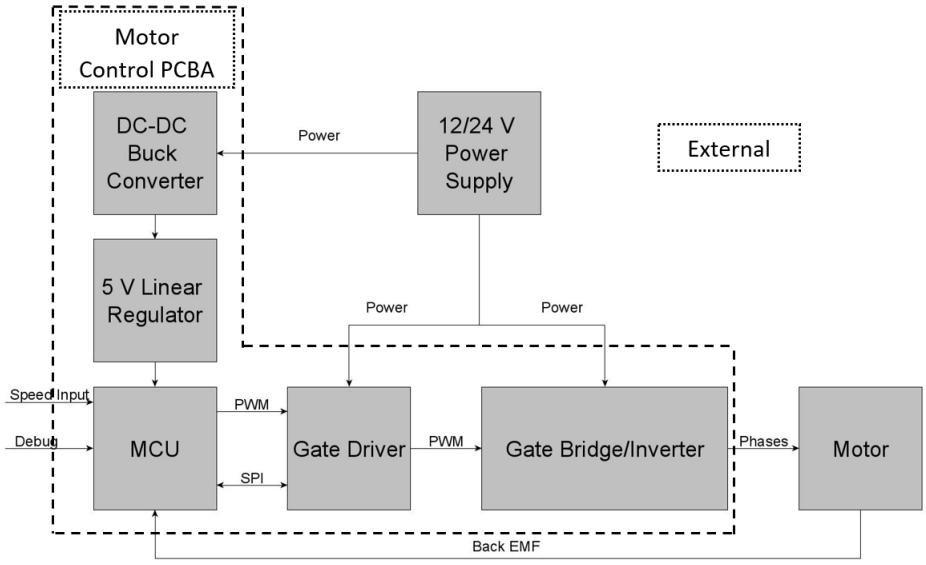


Figure 3.1: Block Diagram of 3-Phase BLDC Motor Control System and Motor

Figure 3.1 depicts the core components of the Motor Control PCBA as well as the critical external components: the power supply and motor. This system was designed to be powered from a 12 or 24 V external DC power supply. A step-down voltage converter (ie. DC-DC Buck Converter) and 5V linear dropout regulator were to be

used in series to provide a constant supply voltage of 5V to the microcontroller. A suitable microcontroller was to be selected and used to serve as the primary processing unit in the system to process the motor control algorithm and provide appropriate PWM signals to the gate driver. The gate driver is then able to use the PWM signals inputted from the MCU to provide a high current gate drive signal to the gate of the appropriate MOSFET's. A three-phase inverter sub-circuit is then used to act as a set of switches to each of the three phases of the motor. Section 3.1 details the selection of specific components for each of these functional elements.

The motor control system was designed with several functional and safety requirements in mind. The system is designed to be able to operate from a 12 or 24 volt external DC power supply and provide up to 30 amps of current to the motor. The system was designed to operate from sensorless feedback methods using back-EMF sensing. Components in the system must be of automotive-grade (ie. able to withstand ambient temperatures up to 125 degrees Celsius) to enable automotive applications of the system. It was desired that the motor speed could be controlled via PWM input or an adjustable potentiometer on the PCBA. Additionally, in the design of this system important criteria including cost and manufacturability were considered such as to work towards the project objective of designing a low-cost BLDC motor control system.

3.1 Component Selection

3.1.1 Microcontroller

To begin the design of the motor control system, suitable components were selected that met the criteria discussed above. First, NXP Semiconductor's S32K142 microcontroller was selected. This chip was selected partially due to its design as a low-cost

chip able to withstand electrically harsh environments including automotive applications [26]. According to the NXP S32K1xx series reference manual, this series of microcontrollers are best suited for a wide range of automotive applications including BLDC motor control as well as lighting, HVAC, door/window/seat controls, and park assist [26].

Finally, a key factor in the selection of this MCU was the quantity and quality of existing documentation and supporting resources for the use of this chip in BLDC motor control applications. NXP currently offers several evaluation boards utilizing the S32K1xx series of MCU, including the MCSXTE2BK142 evaluation board, which utilizes the S32K142 microcontroller for use in 3-phase BLDC motor control. This evaluation board provides the opportunity for testing during software development, as well as serves as a valuable resource in both hardware design and software development. Design reference documents are publicly available for this evaluation board including the engineering schematic, hardware user guide and BOM, as well as the corresponding software for operation of the device, as discussed in Chapter 4. By selecting an MCU with strong supporting documentation detailing the device's usage in BLDC motor control, the overall development process can be completed more rapidly.

3.1.2 Gate Driver

Second, the gate driver chip was selected. Selection of an appropriate gate driver chip can have a dramatic effect on the overall structure of the board as different gate driver chips have radically different capabilities. As discussed in Section 2.2.2, the primary purpose of the gate driver is to provide a high-current input to the gates of the MOSFET's in the three-phase inverter circuit. As such, a gate driver with suitable gate drive current capabilities must be selected given the design criteria. Furthermore,

additional functionality may be integrated within the gate driver chip, reducing the need for additional components on the PCBA. Finally, as with selecting an appropriate microcontroller, selecting a gate driver chip with comprehensive documentation and design resources will lead to a more efficient design process.

For this project, Texas Instruments' DRV8343-Q1 was selected for the reasons discussed above. This chip is designed for use in 12 V and 24 V BLDC motor control and capable of providing up to 2 A peak gate drive current. One distinct advantage of this chip is the integration of 3 current sensing amplifiers, allowing for current sensing on all 3 phases without the need for external amplifiers, as well as the integration of a 3.3 V internal regulator, potentially eliminating the need for an external regulator [35]. Finally, Texas Instruments provides a large volume of high-quality documentation for this device, including a well-organized datasheet, application specifications, and layout recommendations. Furthermore, the DRV8343S-Q1EVM evaluation board is offered to provide designers with the opportunity to test using this board, and corresponding resources including the board schematic and firmware are publicly available. This collection of design references allows for easier and more efficient integration of this chip into the motor control PCBA, which was a key factor in the selection of this chip.

In Table 3.1, the DRV8343-Q1 chip is compared with a comparable device, NXP Semiconductor's MC33GD3000 gate driver chip. Like the DRV8343-Q1, the MC33GD3000 is designed for use in BLDC motor control, and is capable of providing up to 2.5 A of peak gate drive current and operating within 12 V to 48 V systems. At an order quantity of over 1000 units, the MC33GD3000 device carries a slightly higher unit price of $4.54 compared to only $3.59 for the DRV8343-Q1 device. The MC33GD3000 comes in a QFN-56 package, a leadless package that has the advantage of minimizing footprint. The DRV8343 comes in a HTQFP-48 package with slighltly greater

footprint of 9mm x 9mm, compared to 8mm x 8mm. However, the slightly larger footprint of the DRV8343 is offset by the integration of 3 current sensing amplifiers (CSA). These integrated CSA's allow for current sensing of all 3 phases without the need for external components. The GD3000 chip, on the other hand, includes only one integrated CSA, creating the need for external operational amplifier if simultaneous current sensing of all 3 phases is desired. This was a key factor in selecting the DRV8343-Q1 chip, as reducing the overall component quantity was a design priority in this project. Finally, TI's DRV8343-Q1 includes an integrated linear voltage regulator capable of providing 3.3 V and 30 mA of power to external circuitry, whereas the GD3000 chip's internal 5 V regulator is for internal IC use only as it is capable of providing only 1 mA externally [25][35].

Table 3.1: Comparison of TI DRV8343-Q1 and NXP MC33GD3000 Gate Driver Chips[16][25][31][35]

Category	TI DRV8343-Q1	NXP MC33GD3000
Unit Price (Order Qty $\geq$ 1000)	\$2.179	\$2.04
Package	HTQFP-48	QFN-56
Package Dimensions (mm)	9 x 9	8 x 8
Qualification	AEC-Q100	AEC-Q100
Operating Temperature (°C)	$-40 < T_A < 125$	$-40 < T_A < 125$
Input Logic Level	3.3 V or 5 V	3 V or 5 V
Output Logic Level	3.3 V or 5.0 V	5 V
Integrated Current Sensing Amplifiers	3	1
System Operating Voltage	12 V or 24 V	12 V to 48 V
Maximum Gate Drive Current	2 A	2.5 A
Voltage Regulator	3.3 V and 30 mA	Internal Usage Only

3.1.3 MOSFET

Choosing suitable power MOSFET's is another critical component to the design of the motor control system. A MOSFET must be selected with a sufficiently high voltage and current rating considering its application [36]. For this design, Nexperia's

BUK762R4-60E N-channel MOSFET was selected based on its usage in NXP's reference design. This MOSFET is rated for a maximum of 60 V of drain-source voltage, allowing it to be suitable for 12 or 24 V motor control [23]. Futhermore, the device is rated for up to 120 A of drain current. Finally, this device carries an automotive grade, including a temperature rating of 175 °C [23].

3.1.4 Other Components

Additional components were selected primarily based on the needs outlined in NXP and TI product documentation and reference designs. Several factors were considered when selected specific components, including cost, availability, consistency and package. Low-cost components were selected when practical to reduce the overall manufacturing costs of the system. Attempts were made to select commonly available components to improve the ease of sourcing and assembly of the system. Efforts were made to minimize the number of unique components used throughout the PCBA by utilizing common components throughout the board when possible. Finally, common packages were selected, and primarily SMT packages were selected when possible to improve the ease and cost of assembly.

3.2 Schematic Design

After selecting suitable components to make up the motor control system, the schematic was designed to outline the interconnections between components in the system. The schematic was created using the KiCad electronic design software. KiCad was selected due to its being a free and open-source design software, as well as its strong reputation and the bulk of online resources available for learning the software, including free online tutorials.

Component symbols for most components were imported from the SnapEDA and UltraLibrarian online electronics design libraries when available, while some component symbols were made manually from datasheet specifications. All symbols were verified for accuracy from component datasheets or other documentation. The schematic was then designed in several phases, as discussed below.

First, the schematic for NXP's MCSXTE2BK142 evaluation board was used as a model for several common elements, including the MCU, phase inverter circuit, step-down voltage converter, and LDO voltage regulator. From the evaluation board reference design and MCU pinout diagram, connections to and from the MCU were established. The phase inverter circuit was then modeled based upon the example provided on the MCSXTE2BK142 evaluation board. The LM46000-Q1 step-down voltage converter and MC33375 LDO voltage regulator used in the MCSXTE2BK142 design were selected to be used in this design, and their corresponding sub-circuitry was modeled based on the NXP reference design and verified based on the application information in the devices' respective datasheets. Finally, several sub-circuits included in the NXP evaluation board design were included in the design of this system, including back-EMF sensing, temperature sensing, analog and PWM inputs, FreeMASTER control, and SWD debugging interface. Connectors were included for PWM input, FreeMASTER control, SWD debugging, and connections to the power supply and motor.

Several functional elements included in the MCSXTE2BK142 evaluation board were not included in this design, including CAN and LIN communication interfaces, and Hall and Encoder feedback methods, and the schematic was adjusted accordingly. By eliminating these elements and simplifying the overall design, the overall number of components and connections to the MCU was reduced, and cost was subsequently reduced.

Next, the TI DRV8343-Q1 gate driver was integrated into the system based on the application information provided in the DRV8343-Q1 datasheet and the schematic for TI's DRV8343S-Q1EVM evaluation board. The gate driver was connected to the supply voltage as shown in TI's design reference and with consideration for electrical specifications of the gate driver. Several key connections between the gate driver and MCU were established, including six independent PWM signals, the SPI communication lines, and current sensing amplifier outputs. The gate driver was connected to the phase inverter circuit as recommended in the application information in the DRV8343-Q1 datasheet. The three-phase inverter circuit was modified significantly to integrate the three internal current sensing amplifiers of the DRV8343-Q1 gate driver based on TI's documentation.

Overall, the schematic was completed with the focus of integration of the NXP S32K142 MCU with the TI DRV8343-Q1 gate driver for use in three-phase BLDC motor control. In this design, triple-shunt current sensing was incorporated using the DRV8343-Q1 gate driver's internal current sensing amplifiers and back-EMF sensing circuitry was included to allow for sensorless feedback and motor control. Connectors were included to support the standard SWD debugging interface as well as NXP's FREEMASTER interface. Several other functional elements were included in the design, including MOSFET temperature sensing and speed control using either PWM and analog input.

3.3 PCB Layout

After design of the schematic, the design process was continued with PCB layout in KiCad's Pcbnew module. Several key design constraints were first established for the PCB design, including a desired maximum board size equivalent to the comparable

MCSXTE2BK142 evaluation board (170mm x 160mm), as well as a total of 4 layers for the board. The board was to be designed using 2 oz copper thickness on outer layers and 1 oz on inner layers.

Figure 3.2 shows a 3D view of the completed PCBA design with component models placed on the board. This section details the design process for this final PCBA design.

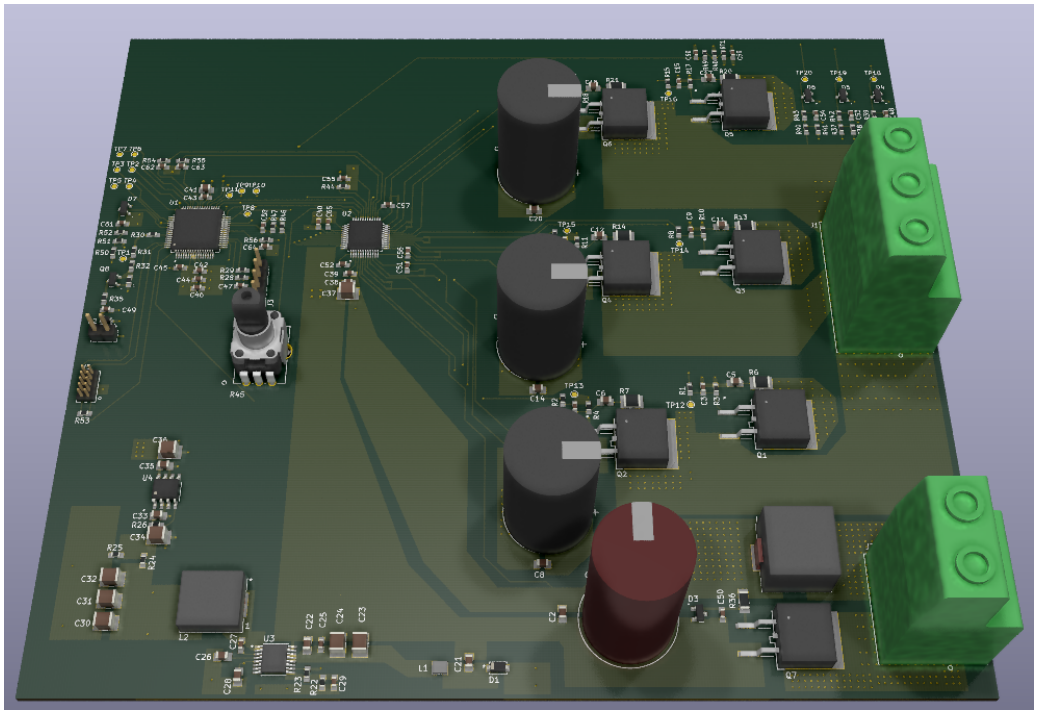


Figure 3.2: 3D Viewer of Final PCBA Design

3.3.1 Component Footprints and 3D Models

When available, component footprints were imported from SnapEDA and UltraLibrarian, and custom designed when unavailable. Regardless, all footprints were closely verified based on the manufacturer's recommendations as detailed in device datasheets. Specific attention was paid to ensure the solderability of the components by verifying pad sizes. For standard component packages such as 0603 and 0804 ca-

pacitors and resistors, relevant IPC standards were referenced for standard footprint dimensions. Additionally, attention was paid to ensure that proper component markings were present, including polarity and/or pin indicators, component outlines, and reference designators.

As shown earlier in Figure 3.2, a 3D model of the PCBA was constructed using KiCad's 3D Viewer tool. The 3D model serves as a valuable aide to verifying footprint design and component clearance. Individual 3D models were obtained for all components so that the 3D model of the PCBA could be constructed. For standard packages, KiCad's library of 3D models was used. SnapEDA and UltraLibrarian were again used for several components that were not available within KiCad's library. When 3D models were not available via an online catalog or KiCad's library, they were custom built using the SolidWorks mechanical CAD software. Figure 3.3 shows an example of a rudimentary 3D model created in the Solidworks computer-aided design software, and saved in the .step file format, which is compatible with KiCad's 3D Viewer tool

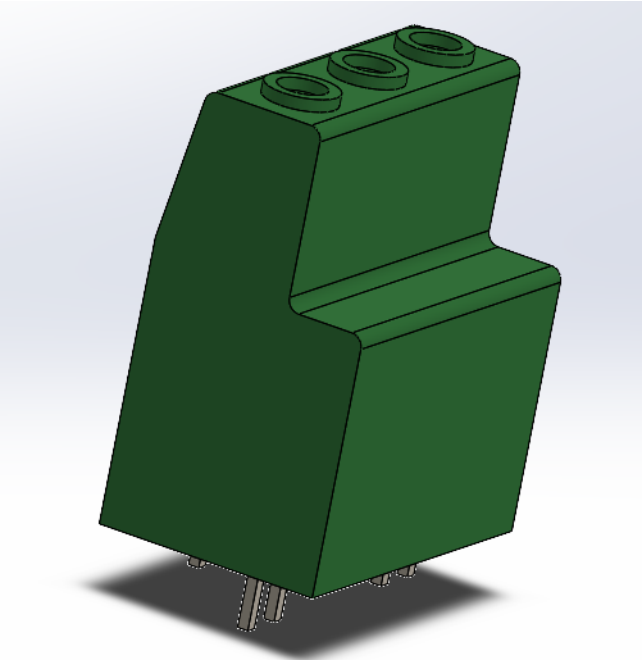


Figure 3.3: 3D Model of Phoenix Contact PN 1782909 Created in Solidworks

3.3.2 Component Placement

Components were placed on the board in a strategic order to minimize the number of iterations necessary in the layout process. Figure 3.4 is shown to provide a high-level view of the placement of the key subsystems on the PCB. This section walks through the process that resulted in this placement of components.

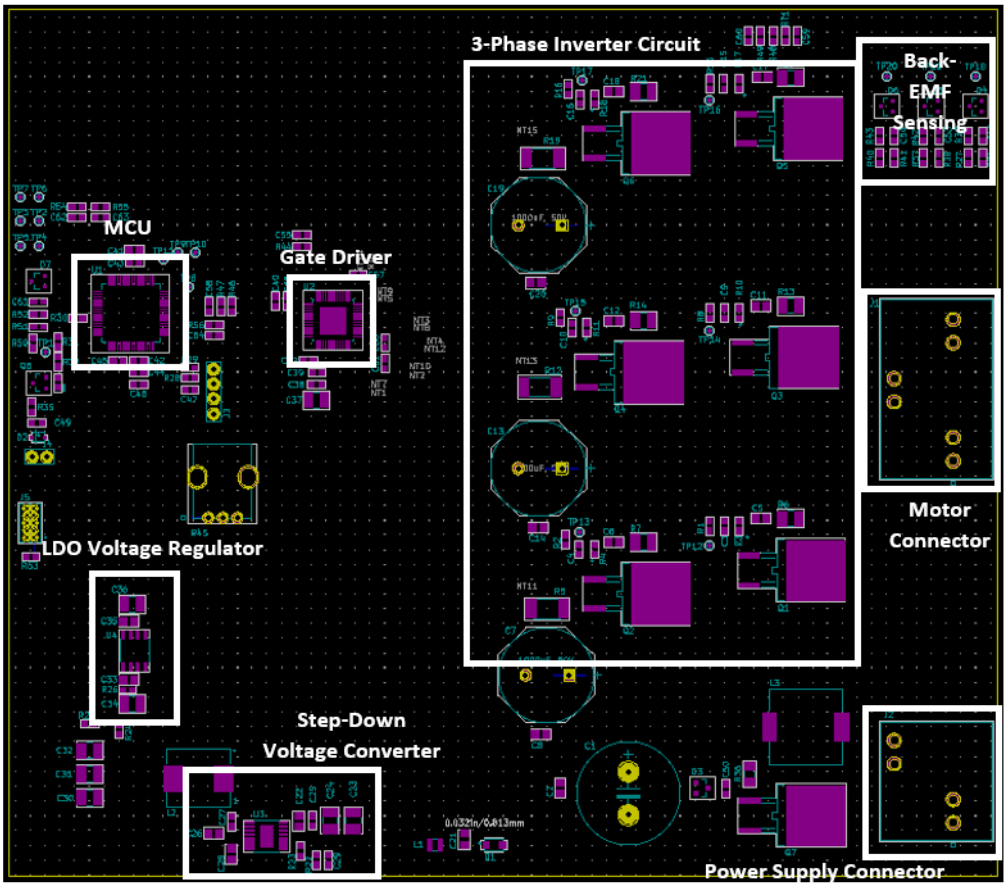


Figure 3.4: Final Placement of Components on PCB with Labels

First, focus was placed on the placement of the components making up the three-phase inverter sub-circuit since many connections in this circuit require large traces capable of carrying high current to the motor. The design of the MCSXTE2BK142 evaluation board was again used as a reference, with modifications made as necessary to incorporate current sensing at all 3 phases. This sub-circuit was placed on the right-side of the board such as to allow the motor connector to be on the edge of the board for optimal usability, as shown in Figure 3.4.

Figure 3.5 shows the placement of components for one phase of the three-phase inverter circuit, which can be considered representative of each of the phases. The high-side and low-side MOSFETs were placed side-by-side, with the low-side MOS-

FET to the left of the high-side MOSFET, allowing the drain of the low-side MOSFET to be easily connected to the source of the high-side MOSFET. The current shunt resistor (shown by R19 in Figure 3.5) was placed next to the source pad of the low-side MOSFET.

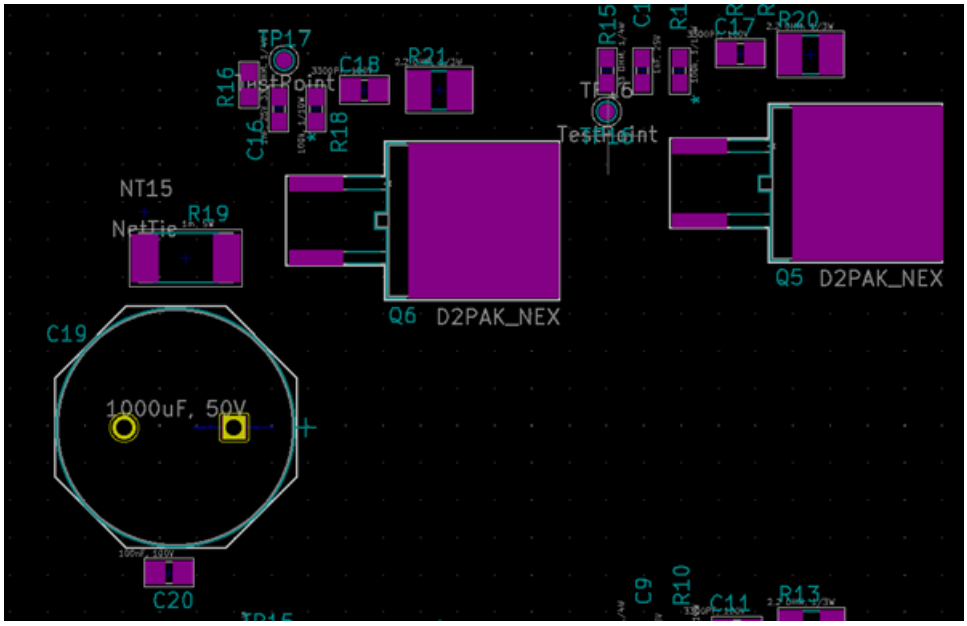


Figure 3.5: Placement of One Phase of Three-Phase Inverter Circuit

Next, the MCU and gate driver chips were placed on the board since these components both require a large number of connections and thus require strategic placement. The gate driver chip was placed to the immediate left of the 3-Phase Inverter circuit as shown in Figure 3.4 to allow for minimal trace length on connections between the gate driver and 3-phase inverter. The placement and orientation of the gate driver was made following the recommendation laid out in the DRV8343-Q1 datasheet, as shown in Figure 3.6.

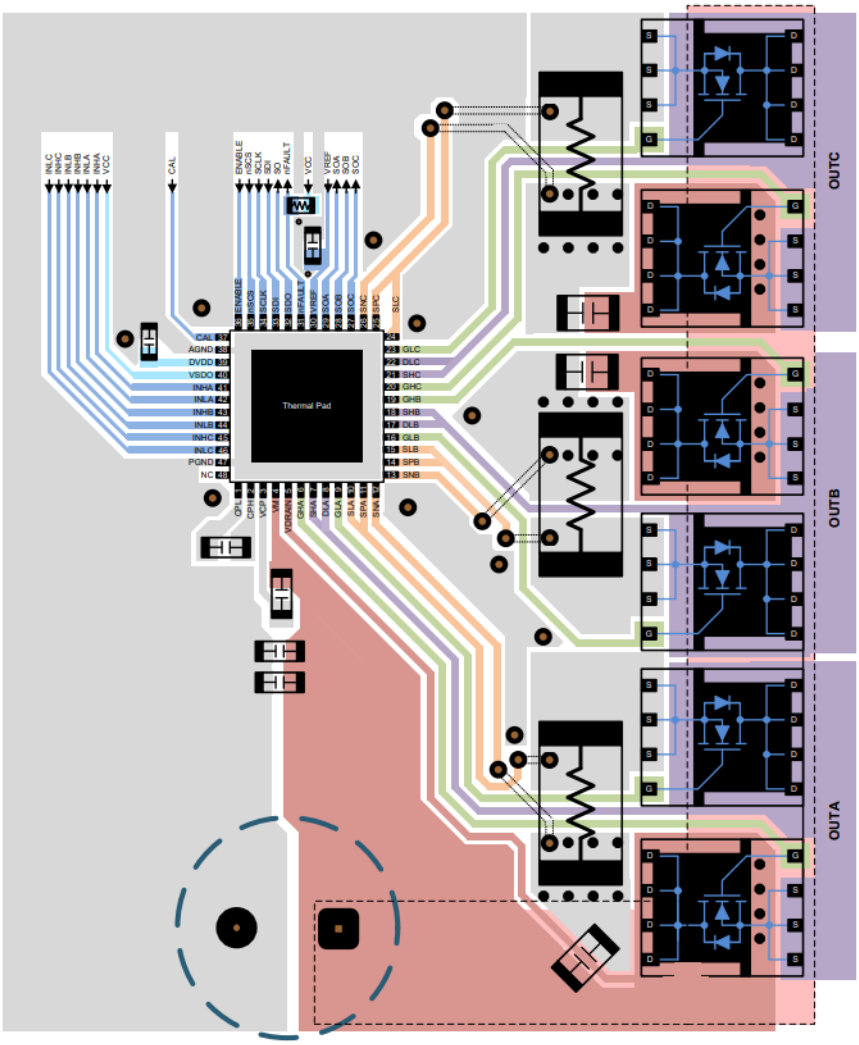


Figure 3.6: Layout Example from DRV8343-Q1 Datasheet
[35]

The MCU was placed to the left of the gate driver, allowing for short traces between the MCU and gate driver. The MCU was oriented with the attempt to place pins that would connect to the gate driver nearest to the gate driver.

Next, the components supplying power to the MCU were placed. The connector to the external power-supply was placed in the lower-right corner of the board with the intention of increasing usability by placing the connector at the edge of the board. Reverse battery protection circuitry was placed to the left of the power supply connector.

The step-down voltage converter sub-circuit and LDO voltage regulator sub-circuit were placed in series in between the power supply connector and MCU.

Remaining subsystems were placed such as to minimize the trace length, when possible. In some instances, such as the temperature sensing subsystem, special considerations were needed to be made for the location of the subsystem. In the case of the temperature sensing subsystem, the components were placed near to the MOSFETs such as to obtain an accurate measurement of the operating temperature.

Placement of certain components was done in such a way to follow recommended practices. For example, Figure 3.7 shows the placement of four decoupling capacitors at the power supply input near to the supply voltage input pins of the LM46000-Q1 step-down voltage converter, and placed in order of ascending capacitance [28].

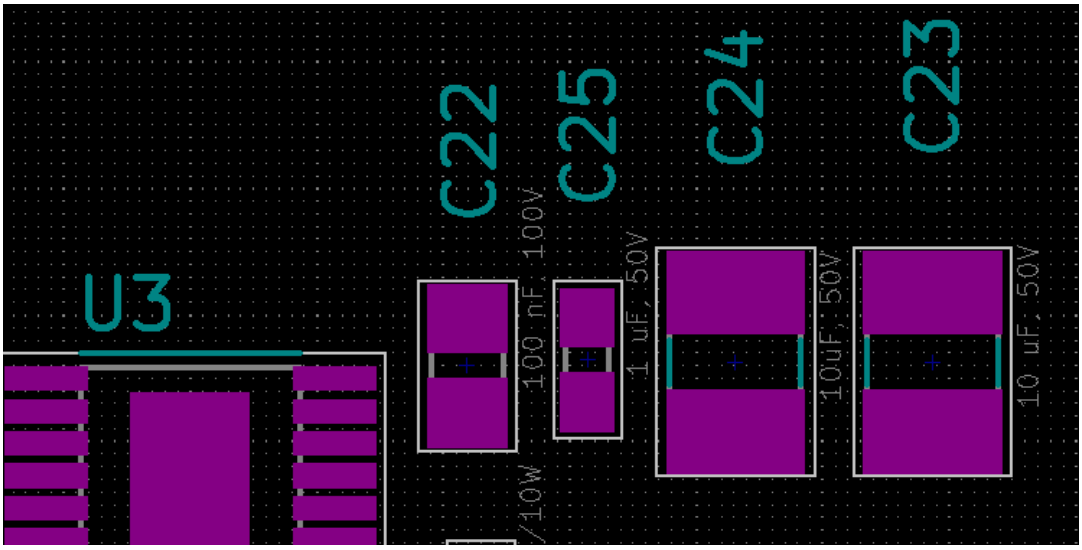


Figure 3.7: Placement of Decoupling Capacitors at LM46000-Q1 Supply Voltage Pins

3.3.3 Trace Routing

After initial placement of the components, copper traces were drawn to connect the components. As with the placement of components, traces were routed in a strategic order such that components and subsystems with a higher number of traces covering more distance were routed first, and then more simple traces were routed at the end.

First, copper traces were drawn to complete the connections between the components of the 3-phase inverter circuit as well as the connection from the 3-phase inverter circuit to the motor connector, as shown in Figure 3.8. The layout for this subsystem was designed considering the current path of the high-current signal going to and from the motor. During operation of the motor, positive current to the motor is supplied to the high-side MOSFET drain and then from its source to the connector, and when the phase is flipped 180 degrees current will exit the motor to the low-side MOSFET drain and exit at the source. As such, large copper pours were placed connecting the low-side MOSFET drain and high-side MOSFET source to the motor connector, as well as large pours placed at the high-side MOSET drain and low-side MOSFET source. Detailed discussion of the requirements considered in the sizing of these copper pours is discussed in Section 3.3.5.

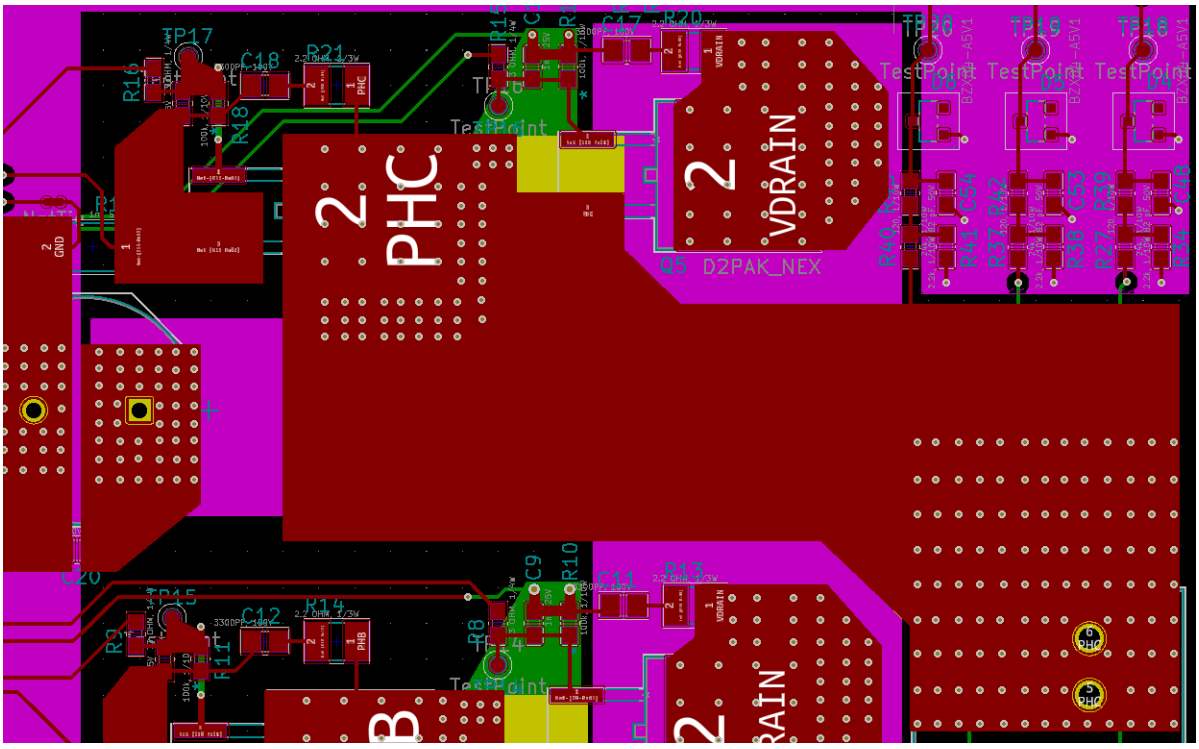


Figure 3.8: Finished Layout of One Phase of 3-Phase Inverter Circuit

Next, the power supply subsystems were routed including the path from the power supply connector to the LM46000 step-down voltage converter, and from the LM46000 to the MC33375 linear voltage regulator. During this stage, layout guidelines from TI and ON Semiconductor documentation was considered, and the MCSXTE2BK142 evaluation board layout was used as a model.

Next, copper traces were routed connecting the gate driver to the three-phase inverter circuit. During this stage, the recommended layout provided by TI and shown in Figure 3.6 was utilized, as well as general layout guidelines provided by TI. Notably, efforts were made to minimize the high-side and low-side gate driver loop lengths, and traces in each loop were routed as differential pairs. Figure 3.9 shows an example of this, demonstrating the Phase B high-side gate driver loop that runs from the GHB

pin of the gate driver, to the high-side MOSFET gate, and back ground the MOSFET source to the SHB pin of the gate driver.

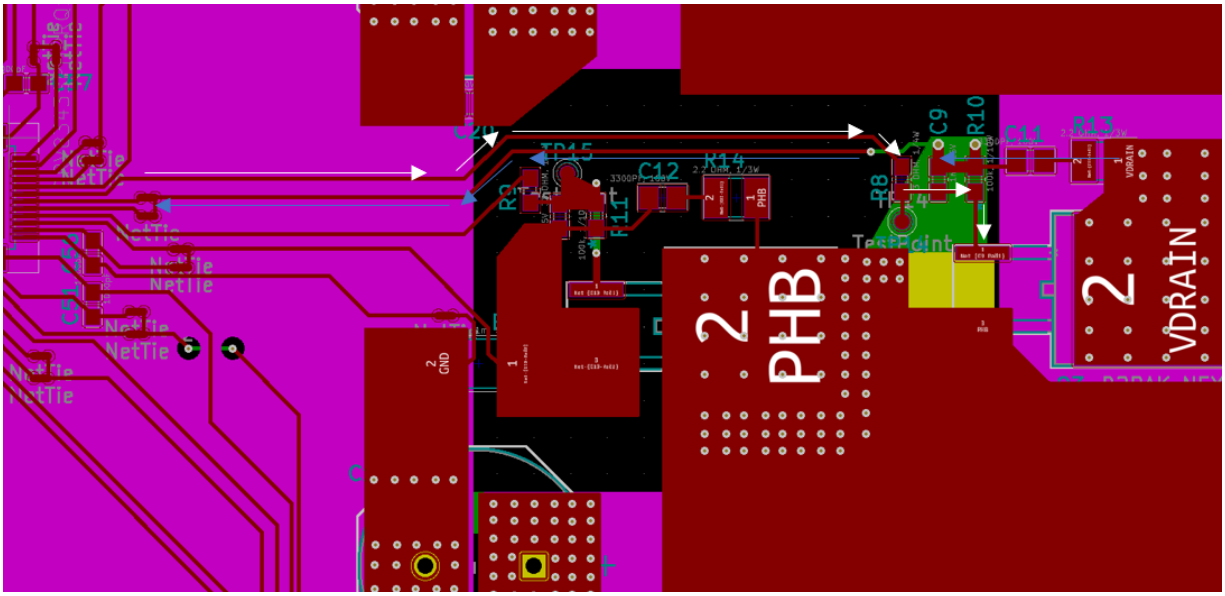


Figure 3.9: Phase B High-Side Gate Driver Loop

Next, traces were routed between the gate driver and MCU, including SPI lines, PWM, current sense amplifier outputs, and the gate driver enable signal. Again, efforts were made to minimize trace length and to maintain consistent trace length across phases. Finally, the MCU was connected to other systems, including the linear voltage regulator, analog input, SWD debugging and FreeMASTER connectors, back-EMF sensing, and temperature sensing.

3.3.4 Power and Ground Planes

Large copper pours were used in several instances, primarily for power and ground planes as well as for high-current signals. The bottom layer and "Inner 2" layer of the PCB were used to hold these power and ground planes, as discussed in this section.

In this design, two power planes were used: a 12/24 V plane (depending on external power supply voltage) supplying power to the three-phase inverter circuit and gate driver, referred to in this project as "VDRAIN" due to being connected to the drains of each of the high-side MOSFET's, and a 5 V plane supplying power to the MCU, gate driver logic supply, and other low-power subsystems, referred to as "VDD." Figure 3.10 shows the bottom layer of the PCB, in which the "VDD" plane is shown on the left connecting to the MCU and gate driver, while the "VDRAIN" plane on the right connects to the three-phase inverter circuit. The "VDRAIN" plane is also duplicated on the "Inner 2" layer to increase the current-carrying capacity, as discussed later in Section 3.3.5, and shown in Figure 3.11.

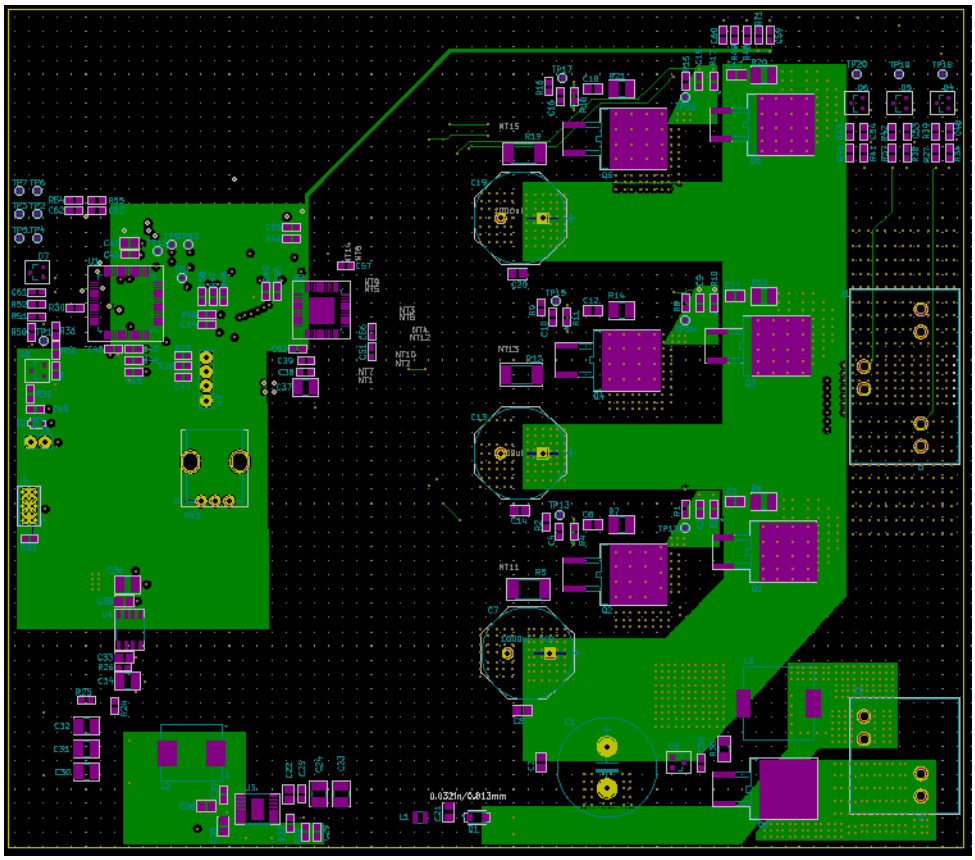


Figure 3.10: Bottom Layer of Final PCB Layout

While some resources recommend separate digital and analog ground planes for such a design as a method of separating noise, in this project the design instead used only one ground plane due to digital and analog signals being relatively isolated on the board, with most analog signals on the right-side of the board, and most digital signals on the left-side of the board. This ground plane was placed on the left-side of the "Inner 2" layer of the PCB, as shown in Figure 3.11. As mentioned earlier, the "VDRAIN" plane was duplicated onto the "Inner 2" layer, as seen on the right of Figure 3.11.

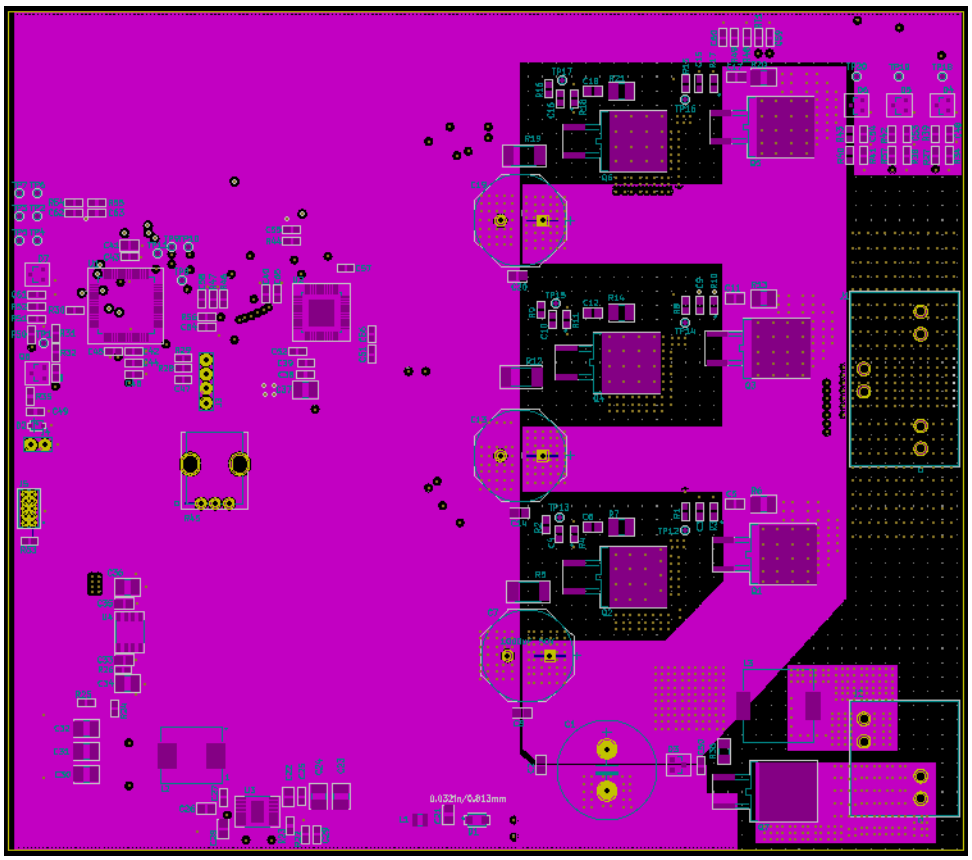


Figure 3.11: Inner 2 Layer of Final PCB Layout

After establishing ground and power planes, several modifications were made to the MCU power supply connections based on the hardware design guidelines for the S32K1xx series microcontrollers set out in NXP Application Note 5426. NXP AN5426

describes the recommended placement of decoupling capacitors and the routing connecting the S32K142 power supply pins to the power plane [24]. Figure 3.12 demonstrates these guidelines, with the decoupling capacitor placed as near as possible to the power supply pin, a via nearby the ground side of the decoupling capacitor to connect the pad directly to the ground plane, and another via nearby the power side of the capacitor connecting the pad goes directly to the VDD plane and eliminating the need for long traces connecting the pad to power.

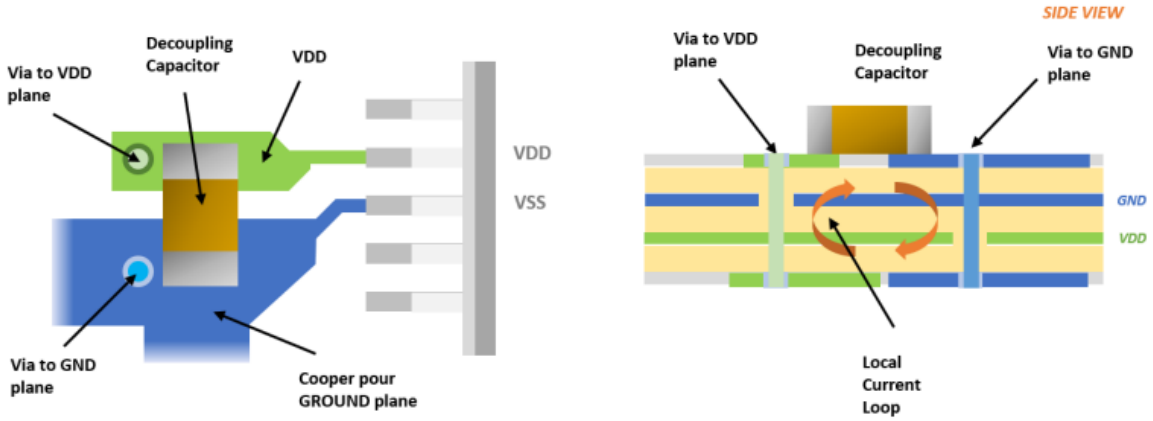


Figure 3.12: Top and Side View of Recommended NXP Power Connection [24]

After evaluation of the layout of the MCSXTE2BK142 evaluation board, power supply connections to the MCU were determined to be an area for improvement. Figure 3.13 shows the power connections to the NXPS32K142 MCU in the MCSXTE2BK142 evaluation board layout. As can be seen, a via has been placed in between the decoupling capacitor and VDD pin, mitigating the effect of the decoupling capacitor by shorting the VDD pin to the VDD plane. Furthermore, the three decoupling capacitors connected to the VDDA pins are connected to the VDDA plane through a long trace, contrary to NXP's guidelines stating that "the capacitor should not route to the power plane through a long trace" [24].

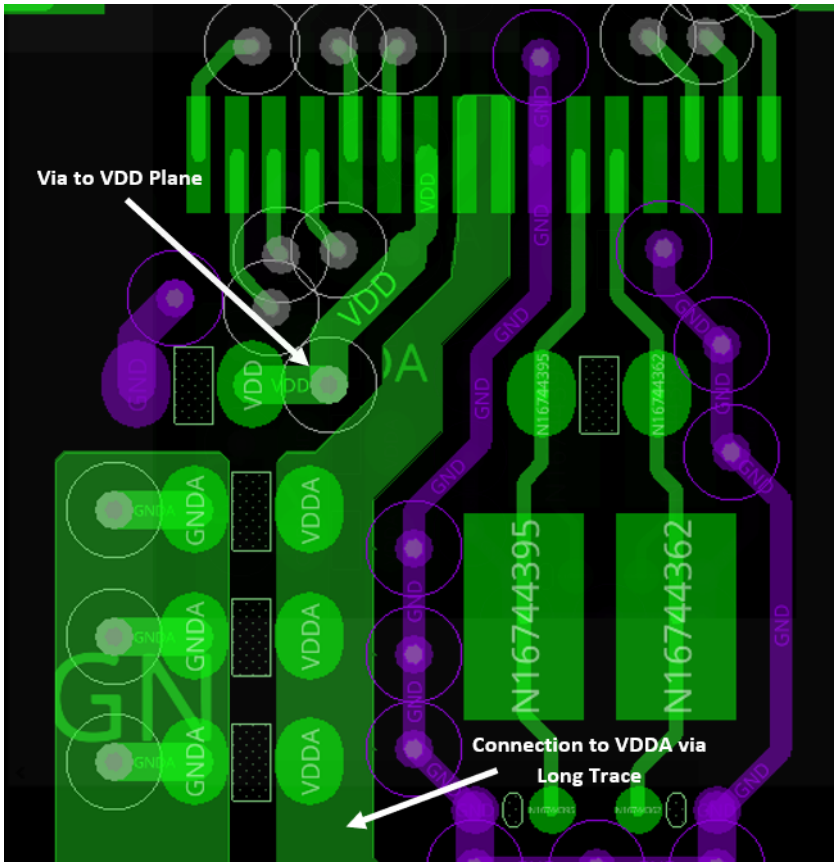


Figure 3.13: NXP Power Connections on MCSXTE2BK142 PCB Layout [40]

Figure 3.14 shows the revised power connection to the MCU in the system designed in this project. Note that since only one power plane was used in this design, rather than separate planes for digital and analog power, both the VDD and VDDA pins of the MCU are connected to the VDD plane. On the left side, the decoupling capacitor is placed in between the power supply pin and via connecting to the VDD plane, ensuring the full effect of the decoupling capacitor and consistent with NXP's guidelines shown in Figure 3.12. On the right side, a similar layout is used, and rather than connecting to the power plane via a long trace as was shown in Figure 3.13, a via is placed near the decoupling capacitors connecting directly to the VDD plane.

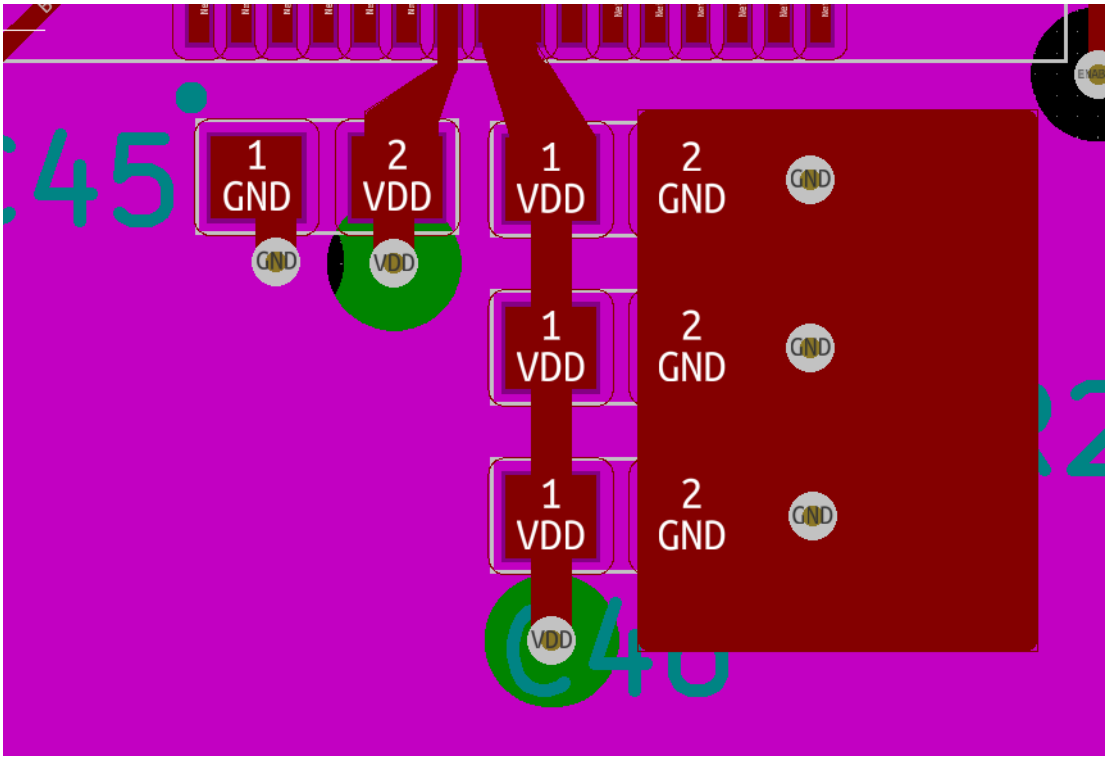


Figure 3.14: Improved Power Connections to S32K142 MCU

3.3.5 Electrical/Thermal Considerations

During the design of the PCB, several considerations were made for electrical and thermal issues. Namely, the required trace width was determined based on IPC-2152: Standard for Determining Current Carrying Capacity in Printed Board Design [19]. An allowable temperature rise of 10°C, outer layer copper thickness of 2oz and inner layer thickness of 1oz was established. Using these parameters, an approximate requirement of 1mm of trace width per 1 A of current on external traces and 2mm of trace width per 1 A of current on internal traces was established and used throughout the PCB design. High-current carrying traces were sized based on this guideline, such as the "VDRAIN" copper zone discussed earlier in Section 3.3.4. Given that the "VDRAIN" zone is required to carry up to 30 A of current, it was sized with 20 mm

thickness at the bottom layer and 20 mm thickness in the inner layer, providing 30 A of current-carrying capacity.

A similar requirement was established for the current carrying capacity of vias. A technical paper published by the PCB design service UltraCAD shows that geometrically, a via with hole diameter D_{via} and copper wall thickness T_{via} has a cross sectional area equivalent to that of a trace with width W_{trace} and thickness T_{trace} when the following equality is satisfied [8]:

$D_{via} = \frac{W_{trace}}{\pi} * \frac{T_{trace}}{T_{via}} - T_{via}$

Thus, a trace of width $1mm$ and thickness $70\mu m$ is roughly equivalent in cross-sectional area to a via with copper wall thickness $20\mu m$ and diameter $1mm$. The paper also demonstrates that the cross sectional area of a via with diameter D is equivalent to the cross sectional area of n vias with diameter $\frac{D}{n}$ [8]. In other words, a via of diameter $1mm$ has a cross sectional area equivalent to that of four vias of diameter $0.25mm$.

Combining this knowledge with the earlier established guideline of 1mm of trace width per 1 A of current, another guideline of four vias of diameter $0.25mm$ per 1 A of current was established. For example, when 10 A of current must flow between two planes on different layers, a minimum of fourty $0.25mm$ diameter vias should be used. It should also be noted that in this guideline, the $0.25mm$ diameter refers to the diameter of the plated hole, not the diameter of the via annular ring.

The inductance of traces was also calculated based on the equation shown below [34]:

$L_{ms} = 0.00508L * [ln(\frac{2L}{W+H}) + 0.5 + 0.2235 * \frac{W+H}{L}]$

With the following definitions:

L_{ms} = Inductance of microstrip (trace) in microhenries (μH)

L = Length of trace in inches

W = Width of trace in inches

H = Distance between the trace and ground plane in inches

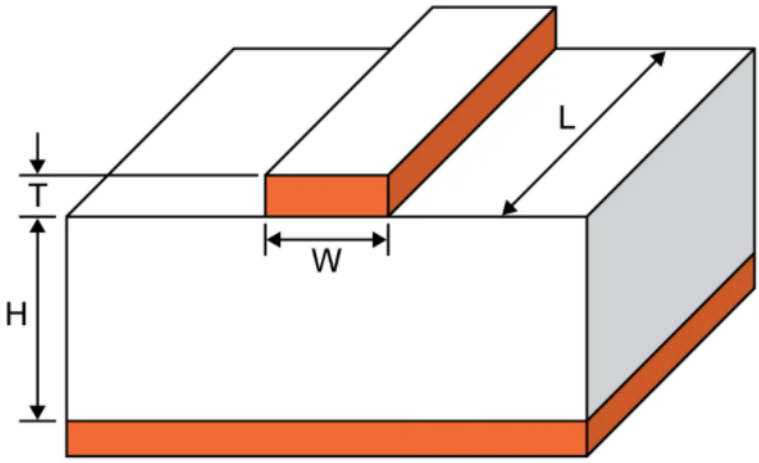


Figure 3.15: Visual of Dimensions in Trace Inductance Formula [5]

Using this equation, it can be determined that length is generally the factor with the biggest impact on trace inductance. Since trace inductance is desired to be minimized, trace length should subsequently be minimized.

3.3.6 DFM/DFA/DFT Considerations

During the design of the printed circuit board, several considerations were made for the overall manufacturability of the system. The manufacturing capabilities listed on the website of PCB manufacturer PCBWay were followed to reduce manufacturing costs and ensure manufactrability of the system [29]. Several design constraints were put in place, including a minimum trace width of .25mm, minimum hole size of .25mm, and minimum conductive spacing (clearance) of 8 mil (.008 inches). All of these design constraints were routinely checked using the "Design Rules Check" tool in KiCad.

As discussed earlier, component footprints were verified versus manufacturer recommendations to ensure solderability. SMT components were used when possible to promote automated assembly processes. Additionally, appropriate silk screen markings on the PCB were placed to enhance the ease of assembly and inspection of the PCBA, including polarity/pin indicators, component outlines, and reference designators. Reference designators were given a consistent font and were placed in intuitive and readable locations.

The potential need for future testing of the system was also considered, and many test points were placed across the board to allow for ease of system testing. Specifically, test points were added such as to allow monitoring of the gate drive current, PWM input, PWM from the MCU to the gate driver, SPI to the gate driver, and back-EMF.

Chapter 4

SOFTWARE DEVELOPMENT

In addition to the design of a printed circuit board assembly for use in three-phase BLDC motor control, this project also aimed to make efforts towards the development of software and firmware for operation of the motor control system. The MCSXTE2BK142 evaluation board was purchased and used to test and modify available software projects from NXP as described in detail later in Sections 4.1 and 4.2.1, and the integration of firmware for the TI DRV8343-Q1 gate driver was evaluated as discussed in Section 4.2.2.

4.1 Initial Testing with Off-The-Shelf Evaluation Unit

The MCSXTE2BK142 evaluation board was purchased from NXP for the purpose of software testing and modifications. Four software projects developed by NXP for three-phase BLDC motor control were downloaded from the NXP website as listed below (with project name listed in paranthesis):

- Single-Shunt Current Sensing (MCSXTE2BK142_PMSM_FOC_1Sh)
- Dual-Shunt Current Sensing (MCSXTE2BK142_PMSM_FOC_2Sh)
- Triple-Shunt Current Sensing (MCSXTE2BK142_PMSM_FOC_2Sh)
- Dual-Shunt Current Sensing with Incorporation of Analog Input On/Off Switch (XS32K142MC24_RDB_PMSM_DualShunt_SDKRTM3P0_AMMCLIB1115)

The first three projects contain nearly identical software, with the only differences being the current sensing method used in each project. The single-shunt current sensing project operates using only sensing of the the DC Bus Current. The dual-shunt current sensing project uses only sensing from Phases A and B and calculates the current of Phase C, given that the sum of the currents of all three phases must equal zero. The triple-shunt current sensing project utilizes sensing of all three phases, and samples two phases at a time and calculates the current in the third phase, with an alternating pattern of which two phases are sampled at any given time.

While a potentiometer is placed on the MCSXTE2BK142 evaluation board and is advertised as an additional method of motor speed control through an adjustable analog input signal to the MCU, this functionality is not incorporated on any of the first three projects listed. A fourth project was obtained from NXP support that incorporates the use of the potentiometer and adjustable analog input signal, however it was only incorporated to the extent of being used as an on/off switch. In this project, when the potentiometer is turned past the point in which the analog input to the MCU surpasses 2.3 V, the motor is turn on to a set speed of 1900 RPM, and turn it back below the set point will turn the motor off. Later modifications were made to fully incorporate this analog input adjustable speed control method, as described later in Section 4.2.1.

All four NXP software projects utilize the field-oriented control (FOC) algorithm for BLDC motor control. Additionally, all four software projects allow for three options for feedback: hall sensing, encoder, or sensorless. In this project, since sensorless feedback was desired for the reduction of hardware and cost, only the software utilizing sensorless feedback was tested.

To test the software with the MCSXTE2BK142 evaluation board, several items were purchased. The PeMicro Universal Debugger module was purchased to allow for con-

nection between the computer and SWD debugging interface of the MCU. Second, a USB-UART converter was purchased to allow connection between the computer and FreeMASTER interface. The "MCSXTE2BK142 Motor Control Development Board Quick Start Guide" was followed for establishing proper connection between the computer and evaluation board. The NT Dynamo Brushless DMA0204024B101 motor was used, and connected to the board's motor screw terminal connector. The Tenma 72-6630 programmable DC power supply was used and connected to the board's power supply screw terminal connector.

Code was viewed, compiled, and loaded to the board through the S32 Design Studio integrated development environment (S32 DS IDE), available for download through NXP's website. The FreeMASTER interface was then used to set and control motor parameters and control the execution of the motor. Figure 4.1 shows the operation of the motor through the FreeMASTER interface.

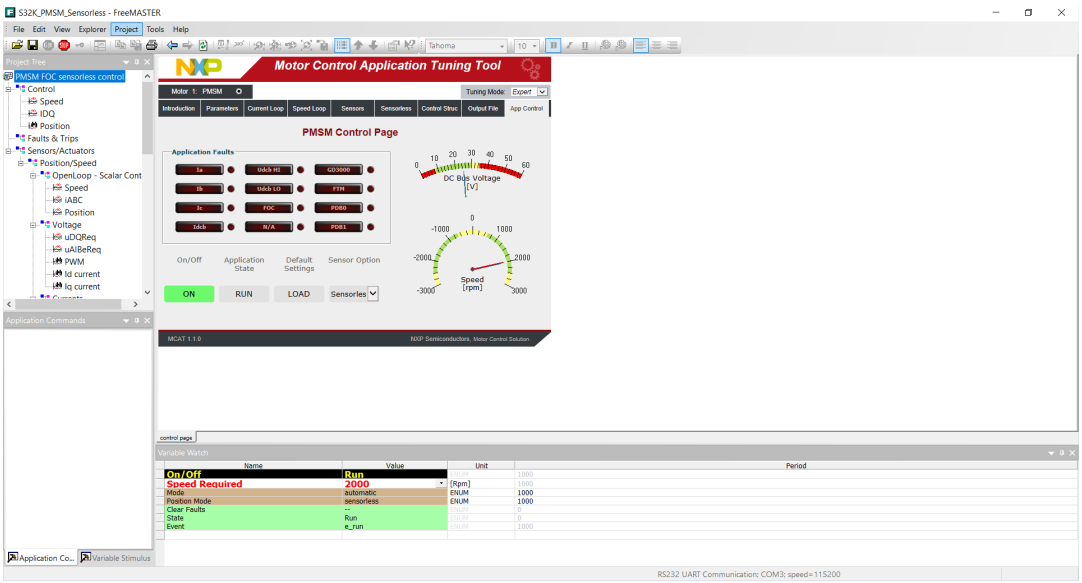


Figure 4.1: Operation of BLDC Motor Using FreeMASTER Control Interface

The dual-shunt and triple-shunt current sensing projects were tested successfully, allowing operation of the motor at speeds from from 500-2500 RPM. The single-shunt current sensing project was tested, however faults in the current sensing occurred immediately upon execution. After some investigation, it was found that improper ADC channels had been configured in the firmware, and the proper channels were configured and the current sensing faults were cleared, however after re-testing new faults occurred in the system. Due to time limitations, and single-shunt current sensing not being a primary focus of this project, efforts were placed elsewhere and problems in the single-shunt current sensing were not fully resolved. Finally, the project incorporating the analog-input signal was tested successfully, allowing the motor to be controlled with the potentiometer used as an on/off switch.

4.2 Software Modifications

4.2.1 Analog Input Speed Control

As discussed earlier, NXP's software project incoporating the potentiometer for analog input speed control was incorporated only to the extent of using the potentiometer as an on/off switch. This project was used as the basis for modifications incorporating the potentiometer for analog inputed adjustable speed control into the motor control system. In the NXP software project, the analog input signal to the MCU had been incorporated such as to read the signal through the MCU's internal ADC, and use this reading to determine whether the motor should run. If the reading is over a value of 2.3 V, the motor is set to run at 1900 RPM. If the reading drops below 2.3 V, the motor is set to turn off. In this project, a goal was to incorporate this analog input signal not only for the functionality of an on/off switch, but also for adjustable speed control between 0 RPM and a user-defined maximum RPM value.

This was successfully accomplished through modifications to the code as partially shown in Figure 4.2. A parameter "desired_max_speed" was defined, allowing the user to easily set their desired maximum motor speed in RPM. Figure 4.2 shows how the analog input is incorporated into code when the system is in the "run" state, meaning the motor is already running. If the analog input falls below 2.0 V, the motor will be shut off. If the input is above 2.0 V, the speed is adjusted based on a linear relationship between the analog input reading and the speed. If the potentiometer is fully turned in the clockwise direction, and a maximum analog input reading is received, then the motor will be set to run at the defined "desired_max_speed" value.

```
if(g_AnalogInData < 2000) //originally 2300
{
    cntrState.event = e_app_off;
    drvFOC.pospeControl.wRotElReq = 0;
}
else{
    drvFOC.pospeControl.wRotElReq = (g_AnalogInData - 2000) * (desired_max_speed * .208) / 2080;
}
```

Figure 4.2: Modified Code to Incorporate Analog Input Adjustable Speed Control in Run State

Further modifications to the code were made for the "ready" state as shown in Figure 4.3, allowing the motor to turn on to the appropriate speed when the analog input exceeds 2.0 V, based on the same linear relationship described in the paragraph above. As a result of these modifications, the user can use the potentiometer on the PCBA to adjust the speed of the motor, as well as turn the motor off. These modifications were tested and found to allow adjustable speed control within +/- 5 RPM as based on the reading in the FreeMASTER interface.

```
if(g_AnalogInData >= 2000)
{
     cntrState.event = e_app_on;
     drvFOC.pospeControl.wRotElReq = (g_AnalogInData - 2000) * (desired_max_speed * .208) / 2080;
}
```

Figure 4.3: Modified Code to Incorporate Analog Input Adjustable Speed Control in Ready State

4.2.2 Integration of TI DRV8343-Q1 Firmware

While this project initially aimed to fully modify the provided software from NXP to allow for the integration of the TI DRV8343-Q1 gate driver, this proved to be infeasible during this project due to the time intensiveness of this task. Instead, efforts were made to evaluate and summarize the functional differences in the software for use of the GD3000 gate driver (which is used on the MCSXTE2BK142 evaluation board) as compared to the DRV8343-Q1 gate driver. This was completed to provide a strong foundation for any future study continuing the work completed in this project.

To compare these differences, software for the DRV8343S-Q1EVM evaluation board was downloaded from the TI website and viewed in the Code Composer Studio IDE. It should be noted that the software for the DRV8343S-Q1EVM evaluation board uses the trapezoidal control algorithm rather than the FOC algorithm, however these differences were not investigated in detail due to the primary focus of this analysis being on the software related to the configuration and operation of the gate driver chip. Futhermore, TI provides two software projects for operation of the DRV8343S-Q1EVM evaluation board, one utilizing sensored feedback methods and one utilizing sensorless feedback. Only the sensorless feedback software project was viewed during this project.

To summarize the differences in firmware for the two gate drivers, the functionality was broken down into functional blocks: SPI, PWM, current sensing, gate driver enable, and fault detection. The SPI block contains all code necessary to configure the SPI communication protocol between the MCU and gate driver, and relevant code that utilizes the SPI lines for configuration of the gate driver. The PWM block contains all code necessary to configure the MCU pins needed to output PWM signal from the MCU to the gate driver, and code relevant to the operation of the PWM

signals. The current sensing block contains code that is relevant for the output from the internal current sensing amplifiers of the gate driver to the MCU, as well as teh corresponding code in the NXP software which allows for current sensing using the external current sensing amplifiers. The gate driver enable block shows the necesary code to configure and use the MCU to enable and disable the gate driver. The fault detection block shows the code used to configure an MCU pin for gate driver fault detection.

These differences were summarized and listed in five separate tables, one for each block, listed in Appendix D. This appendix is intended to provide documentation as to what key software modifications must be made prior to operation of the PCBA designed in this project.

4.3 Future Development and Testing

Future software development is necessary for successful operation of the system designed in this project. As described in Section 4.2.2 above, in this project the software was not modified to incorporate the code necessary for operation of the TI DRV8343-Q1 gate driver. Appendix D is intended to be a useful resource for future development in this area.

Future testing using the PCBA designed in this project should be conducted following a similar procedure as described in this section for compiling and loading software to the board. The "MCSXTE2BK142 Motor Control Development Board Quick Start Guide" is recommended to be used for understanding the connections between the board, motor, power supply and computer, and understanding the procedure to compile, load, and run code on the board.

Chapter 5

ANALYSIS AND RESULTS

5.1 Results

The original objectives of this project included the complete design, fabrication, assembly, and testing of a printed circuit board assembly for use in three-phase BLDC motor control, however due to part shortages fabrication and testing was unable to be completed during this project. During the time of this project, an international IC shortage has affected global supply chains, with a particularly large impact on the automotive industry [38].

As automotive sales in 2020 decreased due to the coronavirus pandemic, many auto manufactures temporarily closed their manufacturing plants and cut purchasing, including the purchasing from the semiconductor industry [38]. At the same time, local and national regulations including "stay at home" orders have led to an increase in purchasing of consumer electronics, leading semiconductor manufacturers to reallocate their production capacity towards the manufacture of products supplying consumer electronic industry [38]. As the automotive demand has began to rebound, the industry has been met with a semiconductor shortage as a result of this reallocation, leading to a chip shortage in the automotive industry that will cause the industry to produce a projected 1.5 to 5 million fewer vehicles in 2021 than originally planned [38]. Beyond the pandemic, severe winter weather and power loss in the state of Texas led to an NXP manufacturing facility being temporarily shut down [27], further hindering component availability.

As a result of this IC shortage, lead times for the NXP S32K142 microcontroller chip have ranged from 48 to 61 weeks, making the complete fabrication and assembly of this system infeasible in the time frame of this project. During this project, the design for a PCBA was finalized as discussed in Chapter 3. Preliminary software development efforts were then made and tested using the MCSXTE2BK142 evaluation board, including the incorporation of adjustable speed control using the analog input signal controlled by a potentiometer. Future software modifications were outlined as discussed in Sections 4.2.2 and 4.3. In lieu of complete fabrication and testing of the system, an outline of potential future fabrication and testing is discussed in Section 5.2 and a cost analysis of the system was constructed as shown in Section 5.3.

5.2 Future Fabrication and Testing

Despite not being able to complete the manufacture and assembly of the motor control system, an outline of future fabrication and testing was completed. For any future fabrication and assembly, all components for the PCBA must first be ordered. Orderable links from electronic component distributor Mouser have been listed for all components in the complete BOM for the final design of the PCBA.

Next, the PCB design must be sent to a PCB fabrication facility. It has been recommended that a PCB fabrication facility that also offers assembly services is selected due to the difficulty of assembly for this system. Due to the large number of components in the system, including several fine pitch components, it is not seen as practical to complete assembly of this system via hand soldering. It is instead recommended to select a suitable assembly service that is capable of assembling the board via automated methods or highly-skilled technicians. One such facility, PCBWay, has been referenced throughout this project and is recommended due to its wide ranging ca-

pabilities in PCB fabrication and assembly as well as its relatively low costs. Before the assembly of the system, all components must be shipped to the assembly facility, unless a "turn-key" option is available and selected in which the facility is responsible for component sourcing as part of its assembly services.

Once the final PCBA is received, testing of the overall system can be conducted. Prior to testing the system, integration of the DRV8343-Q1 gate driver into the software must be completed as discussed in Sections 4.2.2 and 4.3. Once appropriate software modifications are made, the code can be loaded onto the board and executed using the FreeMASTER interface as discussed in detail in Section 4.1. As with any new and custom designed system, problems are likely to be present in either the software or hardware design. To help the testing and debugging process, test points have been placed throughout the board as discussed in Section 3.3.6.

5.3 Cost Analysis

A cost analysis of the system was completed to estimate the overall costs of fabrication and assembly for the system at varying production volumes. Three different production volume categories were selected: prototype volume (10 unit order quantity), mid-volume (1000 unit order quantity), and high-volume (10,000 unit order quantity). Costs were then estimated for component sourcing and fabrication/assembly as shown in Table 5.1.

Component sourcing costs were estimated based on the unit cost of each component at the appropriate order quantity as listed on Mouser. A full list of component costs is included in the component cost estimation table shown in Appendix E. A substantial drop in the overall component sourcing costs was seen between prototype and mid-volume production volumes, decreasing from $85.60 per unit to $54.81 per

Table 5.1: Cost Analysis of Printed Circuit Board Assembly

Production Volume	Prototype	Mid-Volume	High-Volume
Order Quantity	10	1,000	10,000
Components	$ 85.60	$ 54.81	$ 52.75
PCB Fabrication	$ 25.70	$ 7.27	$ 7.23
Assembly	$ 31.50	$ 1.98	$ 1.81
Shipping	$ 0.80	$ 0.76	$ 1.20
Total Cost	$ 142.80	$ 64.81	$ 62.99

unit. A smaller decrease in cost was seen between mid-volume and high-volume, with high-volume component sourcing costs estimated at $52.75 per unit.

PCB Fabrication costs were estimated using the PCBWay "PCB Instant Quote" feature, which allows the user to enter key design specifications and provides an estimation of the manufacturing cost. As discussed in Chapter 3, the final PCB design was a 4-layer PCB with an area of 170mm x 150mm. 2oz copper was used on the outer layers and 1oz copper was used on the inner layers. A 170-180 degree temperature grade was selected for the FR-4 substrate material based on the design specifications for this project. A minimum hole size of 0.25mm was selected as well as a minimum clearance of 8 mil, as discussed in Section 3.3.6. Electroless nickel immersion gold (ENIG) was selected for the surface finish based on the application of this system. As with component sourcing costs, the cost of PCB fabrication is estimated to be much lower at mid-volume production as compared to prototype volume, but a less substantial decrease between mid-volume and high-volume is seen, with per unit costs estimated at $25.70, $7.27, and $7.23, respectively.

PCB Assembly costs were also estimated using the PCBWay "PCB Instant Quote" feature and selecting the operation for assembly service. It was determined that the hand-solder of this system is impractical due to the high number of components and presence of several fine-pitch components, and as such an assembly service should

be utilized. The PCBWay quoting system requires inputs including the number of unique parts on the PCBA, and number of SMT, THT, and BGA/QFP components, respectively. Assembly costs were estimated at $31.50, $1.98, and $1.81 per unit for prototype, mid-volume, and high-volume production, respectively.

Shipping costs were also estimated based on PCBWay's quoting system, coming out to $0.80, $0.76, and $1.20 per unit, for prototype, mid-volume, and high-volume production, respectively. Interestingly, shipping cost estimation is highest at high-volume, but due to the relatively small magnitude of this difference it was not investigated further.

Total costs of the system were estimated at $142.80 per unit for prototype volume, $64.81 for mid-volume, and $62.99 for high-volume. It should be noted that this cost estimate includes only the cost of component sourcing, fabrication, and assembly, and does not including the cost of engineering design or any further implementation of the system.

Chapter 6

CONCLUSION

During this project, a printed circuit board has been designed for use in three-phase brushless DC motor control. This system has been developed to be supplied by a 12 or 24 V DC power supply, and to be capable of sourcing up to 30 A of current. The design of this PCB was centered around the NXP S32K142 microcontroller chip and TI DRV8343-Q1 gate driver chip. The design for this system was based around NXP's MCSXTE2BK142 evaluation board, with the TI DRV8343-Q1 gate driver chip integrated into the system based upon documentation from TI, and a major focus put on reducing system complexity by eliminating superfluous systems.

Accompanying software for BLDC motor control with the S32K142 MCU was tested using the MCSXTE2BK142 evaluation board. Software modifications were made to enable adjustable speed control via a potentiometer controlling an analog input signal to the MCU. Adjustable speed control using the analog input was tested using the evaluation board and was determined to be accurate to within +/- 5 RPM. Preliminary efforts were made to incorporate appropriate firmware to allow use of the TI DRV8343-Q1 driver, and documentation of necessary changes was listed in Appendix D.

Fabrication and assembly of the system was unable to be completed due to an international IC shortage leading to increased lead times for the NXP S32K142 microcontroller chip. In lieu of fabrication of the system, a plan for potential future fabrication and assembly was developed, and a cost analysis of the system was completed. Sys-

tem costs were estimated at $142.80 per unit for prototype production volume, $64.81 for mid-volume production, and $62.99 for high-volume production.

Future work in continuation of this project is recommended in several areas. First, a complete incorporation of the TI gate driver into the software for this system must be completed as discussed in Section 4.3. Once part availability is resumed, components must be ordered for the printed circuit board assembly. The PCB should be sent to a reputable PCB fabrication facility, and assembly services are recommended to be contracted as well. Once the custom fabricated system is received, testing of the system can be conducted.

BIBLIOGRAPHY

[1] BLDC motor control algorithms. Technical report, Renesas. https://www.renesas.com/us/en/application/home-building/motor-control-solutions/motor-algorithms/bldc.

[2] Cal Poly Github. http://www.github.com/CalPoly.

[3] What is 'field oriented control' and what good is it? Technical report, Copley Controls Corp. https://www.maccon.de/fileadmin/redaktion/downloads/Produkte/Antriebselektronik/Copley_drives/Field-Oriented-Control.pdf.

[4] Fundamentals of motor control. Technical report, STMicroelectronics, 2020. https://www.st.com/content/ccc/resource/training/technical/product_training/group1/fd/86/4b/86/e6/c6/46/ce/Fundamentals_of_Motor_Control_2020/files/Fundamentals_of_Motor_Control_2020.pdf/_jcr_content/translations/en.Fundamentals_of_Motor_Control_2020.pdf.

[5] All About Circuits. Microstrip inductance calculator. https://www.allaboutcircuits.com/tools/microstrip-inductance-calculator/. Accessed: 2021-04-10.

[6] Allegro Microsystems. *Sensorless Sinusoidal Drive BLDC Controller*, 2020. Rev. 4.

[7] L. Balogh. Fundamentals of mosfet and igbt gate driver circuits. Technical report, Texas Instruments, 2017. https://www.ti.com/lit/ml/slua618a/slua618a.pdf?ts=1617005353256&ref_url=https%253A%252F%252F.

[8] D. Brooks and D. Graves. Current carrying capacity of vias, 2003. https://www.ultracad.com/articles/viacurrents.pdf.

[9] D. Collins. Field oriented control vs. sinusoidal commutation, May 2016. https://www.motioncontroltips.com/whats-the-difference-between-an-ec-motor-and-a-bldc-motor/.

[10] D. Collins. What's the difference between an EC motor and a BLDC motor?, July 2018. https://www.motioncontroltips.com/whats-the-difference-between-an-ec-motor-and-a-bldc-motor/.

[11] Embitel. Hall effect sensor and its role in a motor controller, 2020. https://www.embitel.com/blog/embedded-blog/hall-effect-sensor-and-its-role-in-a-motor-controller.

[12] J. C. Gamazo-Real, E. Vázquez-Sánchez, and J. Gómez-Gil. Position and speed control of brushless dc motors using sensorless techniques and application trends. *Sensors*, 10(7):6901–6947, 2010.

[13] X. Gao. Bldc motor control with hall sensors based on frdm-ke02z. Technical report, NXP Semiconductor, 2013. https://www.nxp.com/docs/en/application-note/AN4776.pdf.

[14] W. Huazhang. Design and implementation of brushless DC motor drive and control system. *Procedia Engineering*, 29:2219–2224, 2012.

[15] A. Hughes. *Electric Motors and Drives*. Newnes, 1990.

[16] T. Instruments. DRV8343-Q1. https://www.ti.com/product/DRV8343-Q1#order-quality. Accessed: 2021-05-05.

[17] IPC. *IPC-2221 A: generic standard on printed board design*, 1998.

[18] IPC. *IPC-SM-782A: surface mount design and land pattern standard*, 1999.

[19] IPC. *IPC-2152: sStandard for Determining Current Carrying Capacity in Printed Board Design*, 2009.

[20] IPC. *IPC-2612: sectional requirements for electronic diagramming documentation (schematic and logic descriptions)*, 2010.

[21] J. P. Johnson, M. Ehsani, and Y. Guzelgunler. Review of sensorless methods for brushless DC. *Conference Record of the 1999 IEEE Industry Applications Conference*, pages 143–150, 1999.

[22] C. Lewin. Field oriented control (FOC) - a deep dive. `https://www.pmdcorp.com/resources/type/articles/get/field-oriented-control-foc-a-deep-dive-article`.

[23] Nexperia. *BUK762R4-60E product data sheet*, 2016.

[24] NXP Semiconductors. *Hardware Design Guidelines for S32K1xx Microcontrollers*, 2019. Rev. 4.

[25] NXP Semiconductors. *MC33GD3000 Product Data Sheet*, 2020. Rev. 9.0.

[26] NXP Semiconductors. *S32K1xx Series Reference Manual*, 2020. Rev. 12.1.

[27] NXP Semiconductors. NXP provides update regarding impact of severe winter weather on austin, texas facilities, 2021. `https://media.nxp.com/news-releases/news-release-details/nxp-provides-update-regarding-impact-severe-winter-weather`.

[28] Optimum Design Associates. How to place a pcb bypass capacitor: 6 tips. `http://blog.optimumdesign.com/how-to-place-a-pcb-bypass-capacitor-6-tips`. Accessed: 2021-04-06.

[29] PCBWay. Pcbway pcb capabilities. https://www.pcbway.com/capabilities.html. Accessed: 2021-04-12.

[30] T. Sawa and T. Kume. Motor drive technology - history and visions for the future. *2004 IEEE 35th Annual Power Electronics Specialists Conference*, 2004.

[31] N. Semiconductors. MC33GD3000EP. https://www.nxp.com/products/power-management/motor-and-solenoid-drivers/bldc-h-bridge-stepper/3-phase-brushless-motor-pre-driver:GD3000?tab=Buy_Parametric_Tab#/. Accessed: 2021-05-05.

[32] B. Singh and S. Singh. State-of-art on permanent magnet brushless dc motor drives. *Journal of Power Electronics*, 9(1):1–17, 2009.

[33] R. Tang. *MCSXTE2BK142 Schematic*. NXP Semiconductors, 2020. Rev. A1.

[34] F. E. Terman. *Radio Engineers' Handbook*. McGraw-Hill, 1943.

[35] Texas Instruments. *DRV8343-Q1 12-V / 24-V Automotive Gate Driver Unit (GDU) with Independent Half Bridge Control and Three Integrated Current Sense Amplifiers*, 2019. Rev. A.

[36] Toshiba Electronics. *Power MOSFET Selecting MOSFETs and Consideration for Circuit Design*, 2018.

[37] R. Tummala. *Fundamentals of Microsystems Packaging*. McGraw-Hill, 2001.

[38] J. Whalen, R. Albergotti, and D. J. Lynch. Biden can't fix the chip shortage any time soon. here's why. *The Washington Post*, 2021. https://www.washingtonpost.com/technology/2021/03/01/semiconductor-shortage-halts-auto-factories/.

[39] C. Xia. *Permanent Magnet Brushless DC Motor Drives and Controls*. Wiley, 2012.

[40] J. Xiang. *Printed Wiring Board MCSXTE2BK142*. NXP Semiconductors, 2020. Rev. A.

[41] J. Zhao and Y. Yu. AN047: brushless DC motor fundamentals application note. Technical report, Monolithic Power Systems, 2011. https://www.monolithicpower.com/pub/media/document/Brushless_DC_Motor_Fundamentals.pdf.

APPENDICES

Appendix A

BILL OF MATERIALS

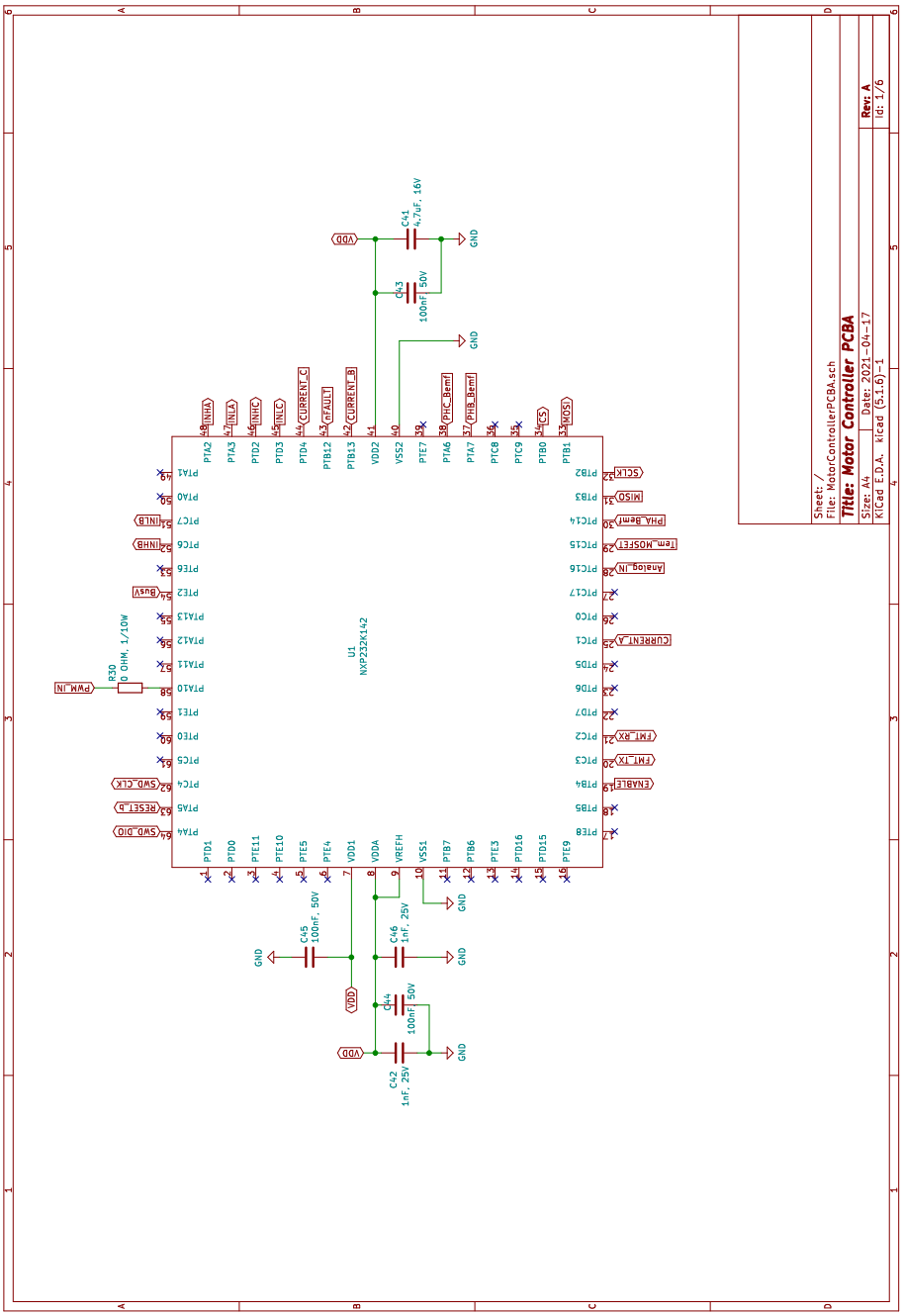
Ref Des	Qty	Value	Description	Manufacturer	Manufacturer PN	Package
C1	1	2200uF	CAP ALEL 2200uF 50V 20% AECQ200 RADIAL	PANASONIC	EEU-FC1H222C	Radial Lead
C2, C8, C14, C20, C21, C22, C33	7	0.1uF	CAP CER 0.1uF 100V 10% X7R AECQ200 0805	MURATA	GCM21BR72A104KA37L	0805
C23, C24, C30, C31, C32, C34, C36	7	10uF	CAP CER 10uF 50V 10% X7R 1210	MURATA	GRM32ER71H106KA12L	1210
C25, C47	2	1uF	CAP CER 1uF 50V 10% X5R AECQ200 0603	TDK	C1608X5R1H105K080AB	0603
C26	1	2.2uF	CAP CER 2.2uF 50V 10% X7R 0805	TAIYO YUDEN	UMK212BB7225KG-T	0805
C27	1	0.47uF	CAP CER 0.47uF 16V 10% X7R AECQ200 0603	MURATA	GCM188R71C474KA55D	0603
C28, C41	2	4.7uF	CAP CER 4.7uF 16V 10% X7R AECQ200 0805	TDK	CGA4J3X7R1C475K125AB	0805
C29	1	36pF	CAP CER 36pF 50V 5% C0G 0603	AVX	06035A360JAT2A	0603
C3, C4, C9, C10, C15, C16, C42, C46	8	1nF	CAP CER 1000pF 25V 1% C0G 0603	AVX	06033A102FAT2A	0603
C35	1	0.68uF	CAP CER 0.68uF 16V 10% X7R AECQ200 0805	TDK	CGA4J2X7R1C684K125AA	0805
C37	1	4.7uF	CAP CER 4.7uF 100V 10%, X7S, 1210	TDK	C3225X7S2A475K200AE	1210
C38	1	0.1uF	CAP CER 0.1uF 100V 10% X7R AEC-Q200 0603	TAIYO YUDEN	HMK107B7104KAHT	0603
C39	1	1uF	CAP CERM 1uF 25V 20% X5R 0603	WURTH	885012106022	0603
C40, C55	1	0.1uF	CAP CER 0.1uF 6.3V 10% X7R 0603	KEMET	C0603C104K9RACTU	0603
C43, C44, C45, C50, C58, C59	6	0.1uF	CAP CER 0.1uF 50V 10% X7R AEC-Q200 0603	AVX	06035C104K4Z2A	0603
C48, C53, C54, C60, C61	5	82pF	CAP CER 82pF 50V 10% C0G 0603	KEMET	C0603C820K5GACTU	0603
C49	1	10nF	CAP CER 0.01uF 50V 5% X7R 0603	AVX	06035C103JAT2A	0603
C5, C6, C11, C12, C17, C18	6	3300pF	CAP CER 3300pF 100V 10% X7R 0805	AVX	08051C332KAT2A	0805
C51, C56, C57	3	1000uF	DNP - CAP CER 1000pF 16V 10% X7R 0603	WURTH	DNP - 885012206034	0603
C52	1	0.047uF	CAP CER 0.047uF 100V 10% X7S 0603	TDK	C1608X7S2A473K080AB	0603
C62, C63, C64	3	2200pF	CAP CER 2200pF 16V 10% X7R 0603	WURTH	885012206036	0603
C65	1	1uF	CAP CER 1uF 6.3V 10% X7R 0603	SAMSUNG	CL10B105KQ8NNNC	0603
C7, C13, C19	3	1000uF	CAP ALEL 1000uF 50V 20% AECQ200 RADIAL	PANASONIC	EEU-FC1H102	Radial Lead

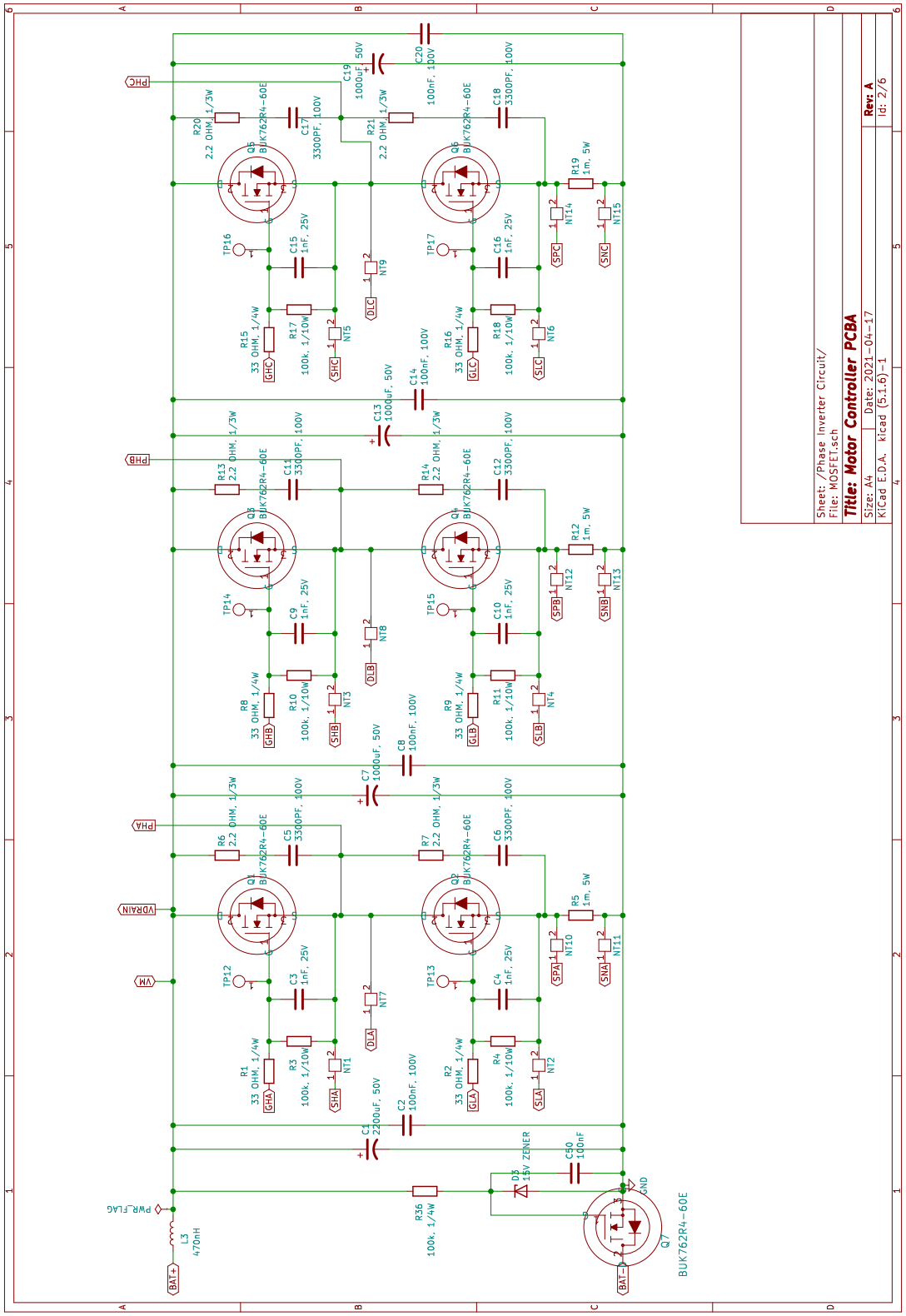
Ref Des	Qty	Value	Description	Manufacturer	Manufacturer PN	Package
D1	1	100V	DIODE SCH RECT 100V 1A AECQ101 SOD123W	NEXPERIA	PMEG10010ELR	SOD-123
D2	1	30V	DIODE ZNR 30V 300mW SOD323	NEXPERIA	BZX384-C30	SOD-323
D3	1	15V	DIODE ZNR 15V 300mW SOT-23	ON	BZX84C15LT3G	SOT-23
D4, D5, D6, D7	4	5.1V	DIODE ZNR 5.1V 250mW AECQ101 SOT23	NEXPERIA	BZX84-A5V1	SOT-23
J1	1	-	CON 1X3 TB 10.16 MM SPACING	Phoenix	1782909	TB
J2	1	-	CON 1X2 TB 10.16 MM SPACING	Phoenix	1709681	TB
J3	1	-	HDR 1X4 TH 2.54 MM SPACING	WURTH	61300411121	HDR
J4	1	-	HDR 1X2 TH 2.54 MM SPACING	WURTH	61300211121	HDR
J5	1	-	HDR 2X5 TH 50MIL CTR 219H AU 120L	AMTEC	FTSH-105-01-F-D	HDR
L1	1	1uH	IND PWR 1uH@100kHz 2.5A 20% SMD	WURTH	78438323010	
L2	1	100uH	IND PWR 100uH@100KHZ 2.0A 20% SMD	EATON	HCM1A1104-101-R	
L3	1	0.47uH	IND PWR 0.47 uH@100KHZ 50A 20% SMD	WURTH	744309047	
Q1, Q2, Q3, Q4, Q5, Q6, Q7	7	-	TRAN NMOS PWR SW 2.4mOHM 120A 60V D2PAK	NEXPERIA	BUK762R4- 60E	D2PAK
Q8	1	-	TRAN NMOS 300mA 60V SOT-23	NEXPERIA	2N7002	SOT-23
R1, R2, R8, R9, R15, R16	6	33	RES MF 33 OHM 1/4W 1% AEC-Q200 0603	VISHAY	RCS060333R0FKEA	0603
R22	1	1.0M	RES MF 1.0M 1/10W 1% 0603	WALSIN	WR06X1004FTL	0603
R23	1	169K	RES MF 169K 1/10W 1% AEC-Q200 0603	KOA SPEER	RK73H1JTTD1693F	0603
R24, R30, R31, R35	4	0	RES MF ZERO OHM 1/10W AEC-Q200 0603	PANASONIC	ERJ-U030R00V	0603
R25	1	1.3K	RES MF 1.3K 1/10W 1% AEC-Q200 0603	KOA SPEER	RK73H1JTTD1301F	0603
R26, R33, R48, R51, R53	5	10.0K	RES MF 10.0K 1/10W 1% 0603	YAGEO AMERICA	RC0603FR-0710KL	0603
R27, R37, R40, R50	4	24K	RES MF 24.0K 1/10W 1% 0603	VISHAY	CRCW060324K0FKEA	0603
R28, R29, R39, R42, R43, R47, R49, R52	8	120	RES MF 120 OHM 1/10W 1% 0603	YAGEO AMERICA	AC0603FR-07120RL	0603
R3, R4, R10, R11, R17, R18	6	100K	RES MF 100K 1/10W 1% 0603	BOURNS	CR0603-FX-1003ELF	0603
R32	1	1.0K	RES MF 1.0K 1/10W 1% 0603	YAGEO AMERICA	RC0603FR-071KL	0603
R34, R38, R41, R46	4	2.2K	RES MF 2.2K 1/10W 0.1% 0603	YAGEO AMERICA	RC0603DR-072K2L	0603
R36	1	100K	RES MF 100K 1/4W 5% 1206	ROHM	KTR18EZPJ104	1206
R44	1	10K	RES MF 10K 1/10W 5% AEC-Q200 0603	VISHAY	CRCW060310K0JNEA	0603
R5, R12, R19	3	0.001	RES PWR 0.001 OHM 3W 1% AEC-Q200 2512	VISHAY	WSLP25121L000FEA	2512
R45	1	10K	RES POT 10K 20V 20% 9MM TH	ALPS ALPINE	RK09D1130C3W	
R54, R55, R56	3	56	RES MF 56 OHM 1/10W 5% AEC-Q200 0603	VISHAY	CRCW060356R0JNEA	0603
R6, R7, R13, R14, R20, R21	6	2.2	RES MF 2.2 OHM 660mW 5% 1210 ANTISURGE	ROHM	ESR25JZPJ2R2	1210

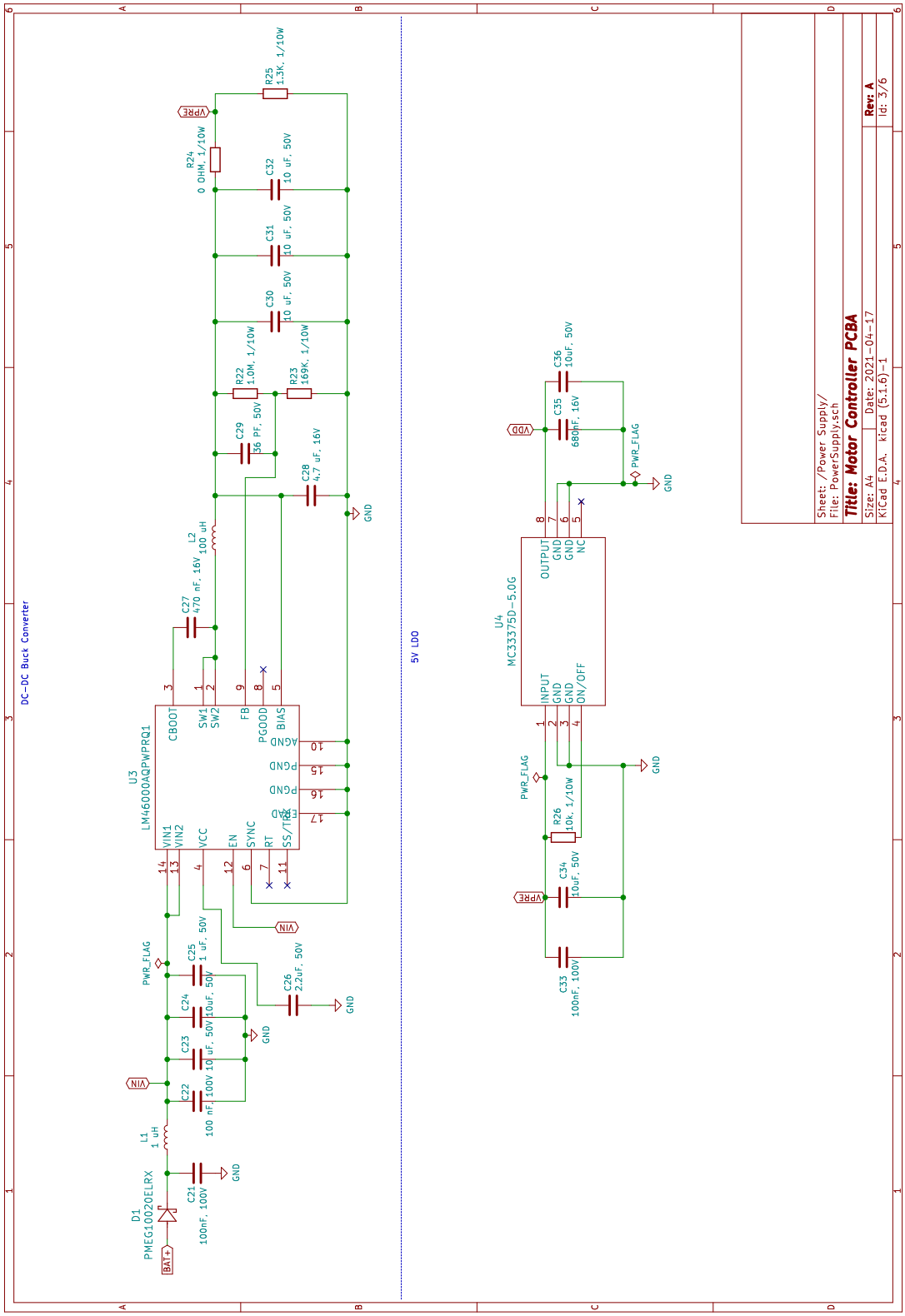
Ref Des	Qty	Value	Description	Manufacturer	Manufacturer PN	Package
RT1	1	10K	THERMISTOR NTC 10K@25DEG 125mW 5% 0603	VISHAY	NTCS0603E3103JLT	0603
U1	1	-	IC MCU 32 BIT 256K MEM 80MHz 2.7-5.5V LQFP64	NXP	FS32K142HAT0MLHT	LQFP64
U2	1	-	IC GATE DRIVER THREE PHASE 60V HTQFP-48	TI	DRV8343SPHPRQ1	HTQFP-48
U3	1	-	IC LIN BUCK DCDC SYNC 1-28V 0.5A 3.5-60V HTSSOP16	TI	LM46000AQPWPRQ1	HTSSOP-16
U4	1	-	IC VREG LDO 5V 300MA 13V SOIC8	ON	MC33375D-5.0G	SOIC-8

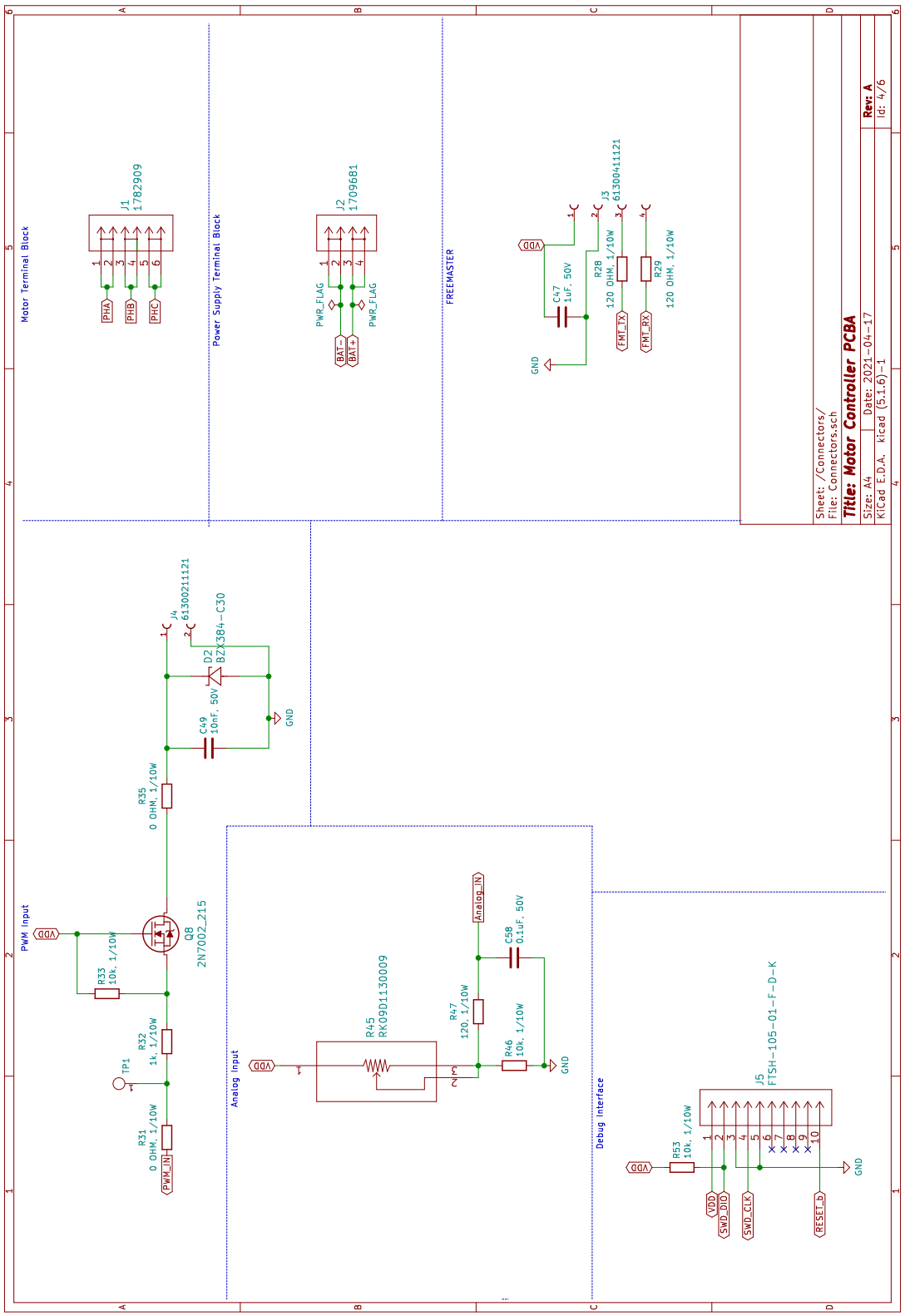
Appendix B

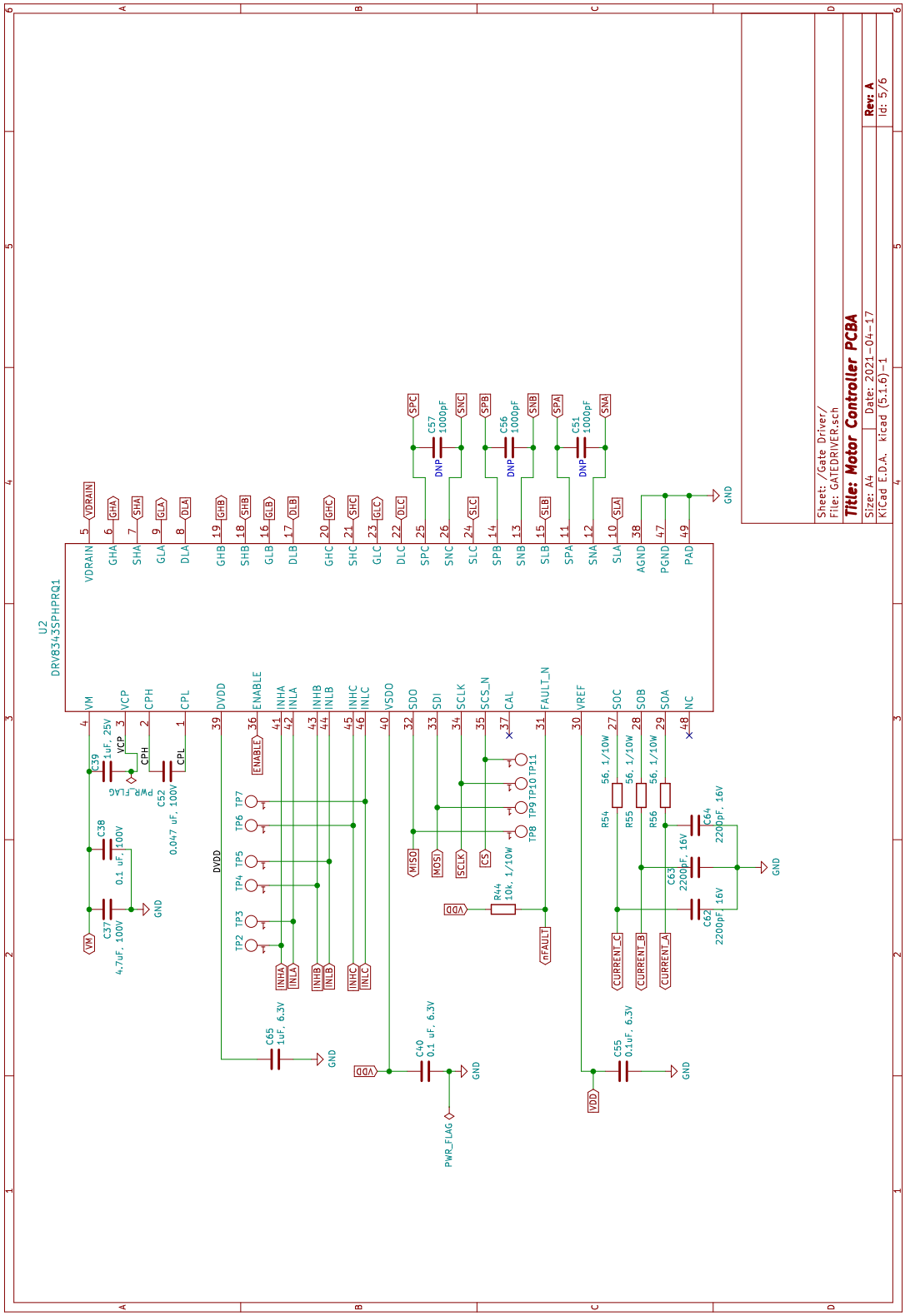
SCHEMATIC

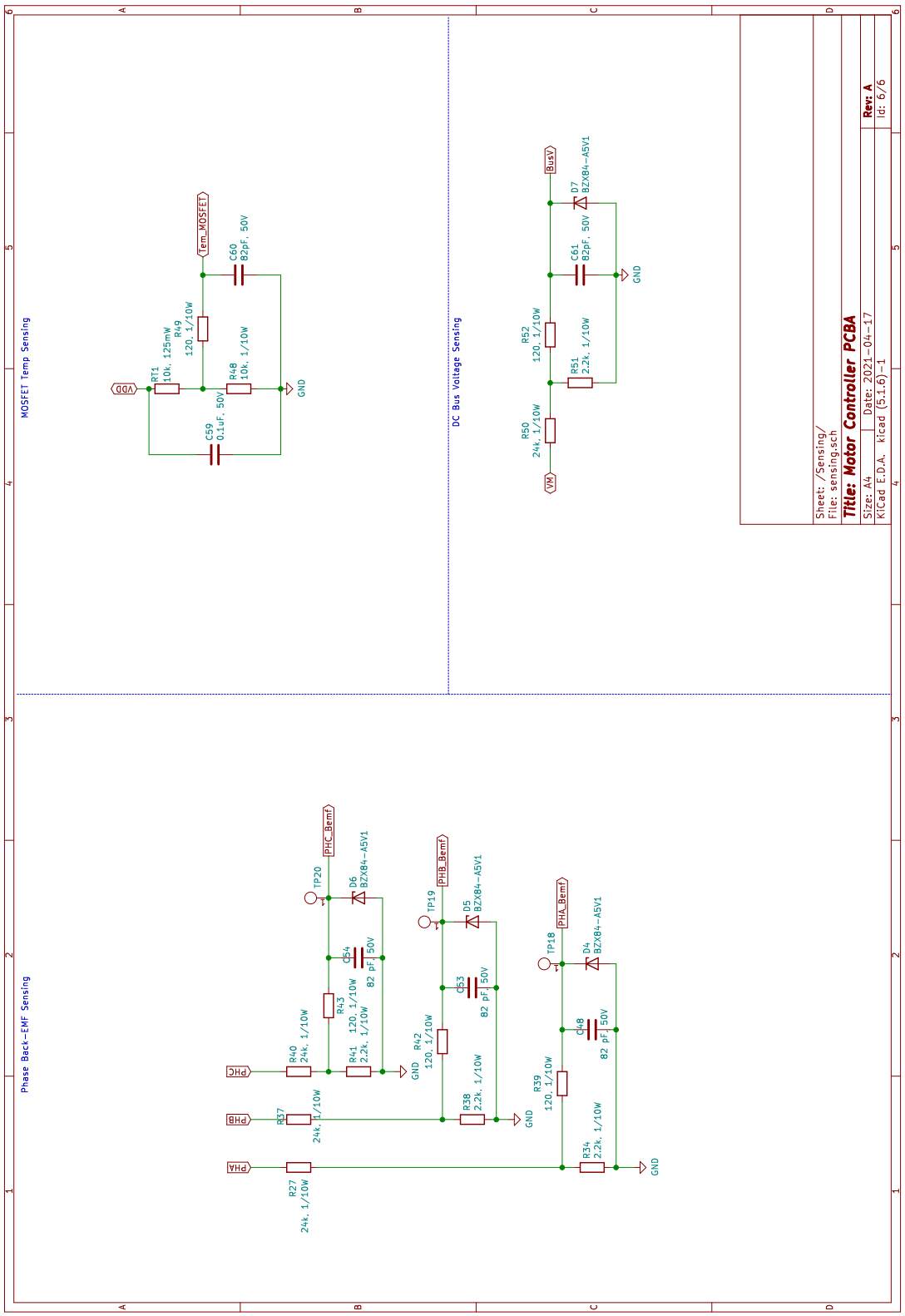



Sheet: /Phase Inverter Circuit/
File: MOSFET.sch
Title: Motor Controller PCBA
Size: A4
Date: 2021-04-17
Rev: A
KiCad E.D.A. kicad (5.1.6)-1
Id: 2/6
PHC
PHB
PHA
VDRAIN
VM
PWR_FLAG
BAT+
BAT-
GND
L3
470nH
D8
18V ZENER
Q7
BUK762R4-60E


DC-DC Buck Converter
5V LDO
U3
LM46000AQPWPRQ1
U4
MC33375D-5.0G
D1
PMEG10020ELRX
L1
1 uH
L2
100 uH
R22
1.0M, 1/10W
R23
169K, 1/10W
R24
0 OHM, 1/10W
R25
1.3K, 1/10W
R26
10k, 1/10W
C21
100nF, 100V
C22
100 nF, 100V
C23
10 uF, 50V
C24
10uF, 50V
C25
1 uF, 50V
C26
2.2uF, 50V
C27
470 nF, 16V
C28
4.7 uF, 16V
C29
36 PF, 50V
C30
10 uF, 50V
C31
10 uF, 50V
C32
10 uF, 50V
C33
100nF, 100V
C34
10uF, 50V
C35
680nF, 16V
C36
10uF, 50V
VIN
VPRE
VDD
BAT+
GND
PWR_FLAG
Sheet: /Power Supply/
File: PowerSupply.sch
Title: Motor Controller PCBA
Size: A4
Date: 2021-04-17
Rev: A
KiCad E.D.A. kicad (5.1.6)-1
Id: 3/6


Motor Terminal Block
Power Supply Terminal Block
FREEMASTER
PWM Input
Analog Input
Debug Interface
J1
1782909
PHA
PHB
PHC
J2
1709681
PWR_FLAG
BAT-
BAT+
C47
1uF, 50V
R28
120 OHM, 1/10W
J3
61300411121
FMT_TX
R29
120 OHM, 1/10W
FMT_RX
GND
VDD
J4
61300211121
D2
BZX384-C30
C49
10nF, 50V
R35
0 OHM, 1/10W
Q8
2N7002_215
R33
10k, 1/10W
R32
1k, 1/10W
R31
0 OHM, 1/10W
PWM_IN
TP1
R45
RK09D1130009
R47
120, 1/10W
R46
10k, 1/10W
C58
0.1uF, 50V
Analog_IN
R53
10k, 1/10W
J5
FTSH-105-01-F-D-K
SWD_DIO
SWD_CLK
RESET_b
Sheet: /Connectors/
File: Connectors.sch
Title: Motor Controller PCBA
Size: A4
Date: 2021-04-17
Rev: A
KiCad E.D.A. kicad (5.1.6)-1
Id: 4/6


Sheet: /Gate Driver/
File: GATEDRIVER.sch
Title: Motor Controller PCBA
Size: A4
Date: 2021-04-17
Rev: A
KiCad E.D.A. kicad (5.1.6)-1
Id: 5/6
U2
DRV8343SPHPRQ1
VDRAIN
GHA
SHA
GLA
DLA
GHB
SHB
GLB
DLB
GHC
SHC
GLC
DLC
SPC
SNC
SLC
SPB
SNB
SLB
SPA
SNA
SLA
AGND
PGND
PAD
VM
VCP
CPH
CPL
DVDD
ENABLE
INHA
INLA
INHB
INLB
INHC
INLC
VSDO
SDO
SDI
SCLK
SCS_N
CAL
FAULT_N
VREF
SOC
SOB
SOA
NC
C39
1uF, 25V
C38
0.1 uF, 100V
C37
4.7uF, 100V
C52
0.047 uF, 100V
C57
100pF
C56
100pF
C51
100pF
DNP
R54
56, 1/10W
R55
56, 1/10W
R56
56, 1/10W
C64
220pF, 16V
C63
220pF, 16V
C62
220pF, 16V
C65
1uF, 6.3V
C40
0.1 uF, 6.3V
C55
0.1uF, 6.3V
R44
10k, 1/10W
TP2
TP3
TP4
TP5
TP6
TP7
TP8
TP9
TP10
TP11
PWR_FLAG
GND
VDD
MISO
MOSI
SCLK
CS
nFAULT
CURRENT_C
CURRENT_B
CURRENT_A


MOSFET Temp Sensing
DC Bus Voltage Sensing
Phase Back-EMF Sensing
Tem_MOSFET
VDD
RT1
10k, 125mW
R49
120, 1/10W
R48
10k, 1/10W
C60
82pF, 50V
C59
0.1uF, 50V
GND
BusV
D7
BZX84-A5V1
C61
82pF, 50V
R52
120, 1/10W
R51
2.2k, 1/10W
R50
24k, 1/10W
VM
PHC_Bemf
TP20
D6
BZX84-A5V1
C54
82 pF, 50V
R43
120, 1/10W
R41
2.2k, 1/10W
R40
24k, 1/10W
PHC
PHB_Bemf
TP19
D5
BZX84-A5V1
C53
82 pF, 50V
R42
120, 1/10W
R38
2.2k, 1/10W
R37
24k, 1/10W
PHB
PHA_Bemf
TP18
D4
BZX84-A5V1
C48
82 pF, 50V
R39
120, 1/10W
R34
2.2k, 1/10W
R27
24k, 1/10W
PHA
Sheet: /Sensing/
File: sensing.sch
Title: Motor Controller PCBA
Size: A4
Date: 2021-04-17
Rev: A
KiCad E.D.A. kicad (5.1.6)-1
Id: 6/6

Appendix C

PCB LAYOUT

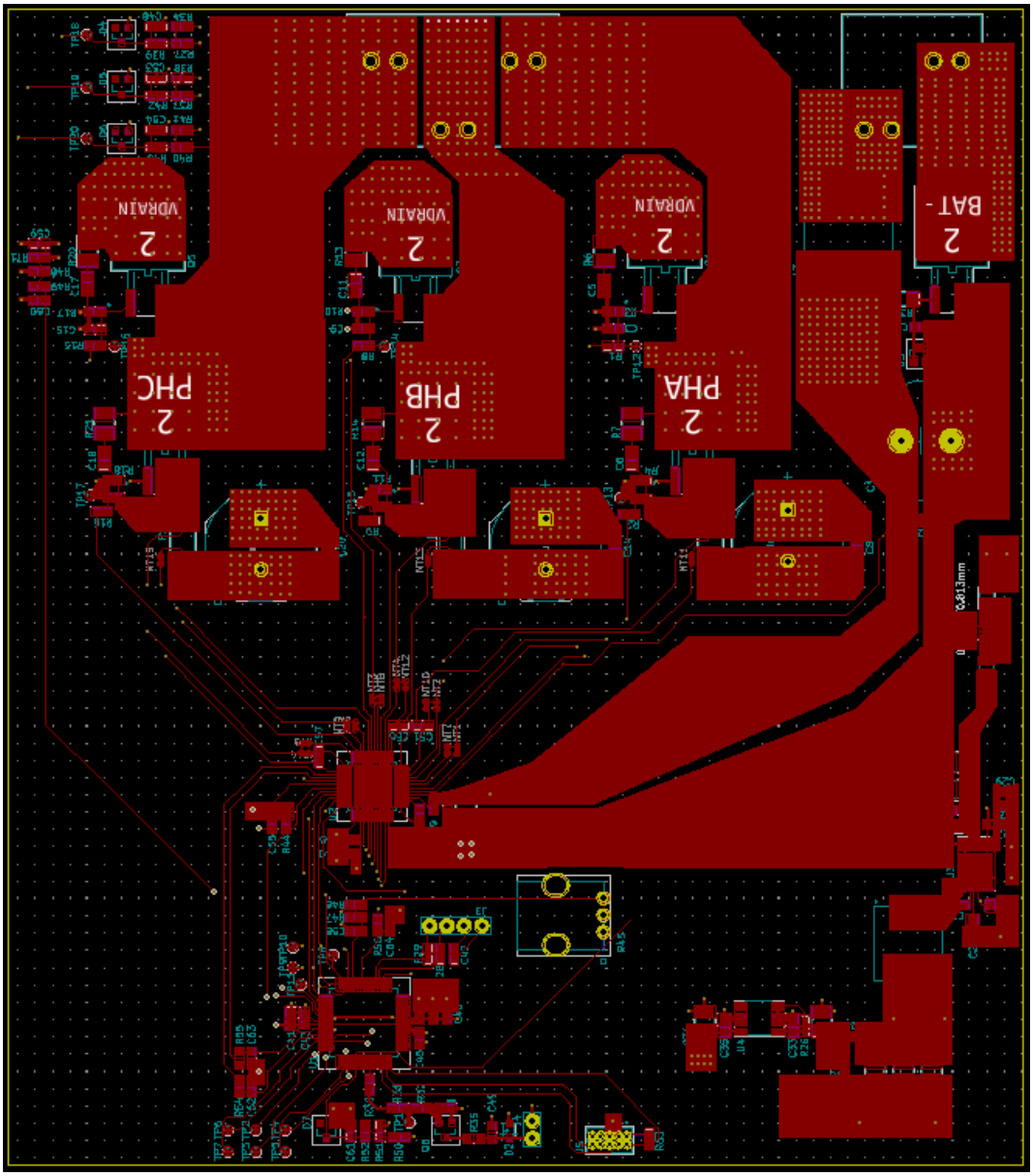


Figure C.1: Top Layer of Final PCB Layout

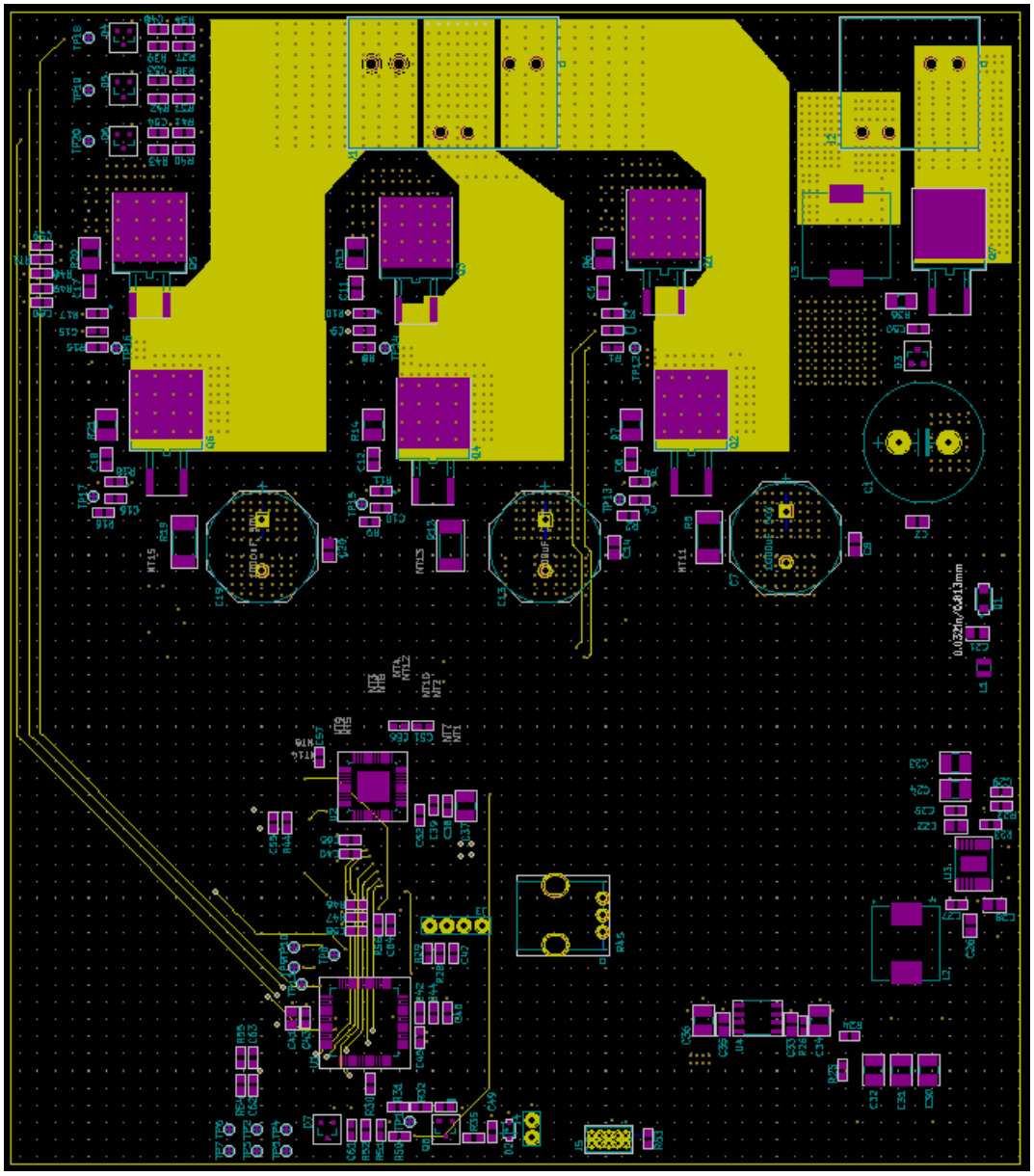


Figure C.2: Inner 1 Layer of Final PCB Layout

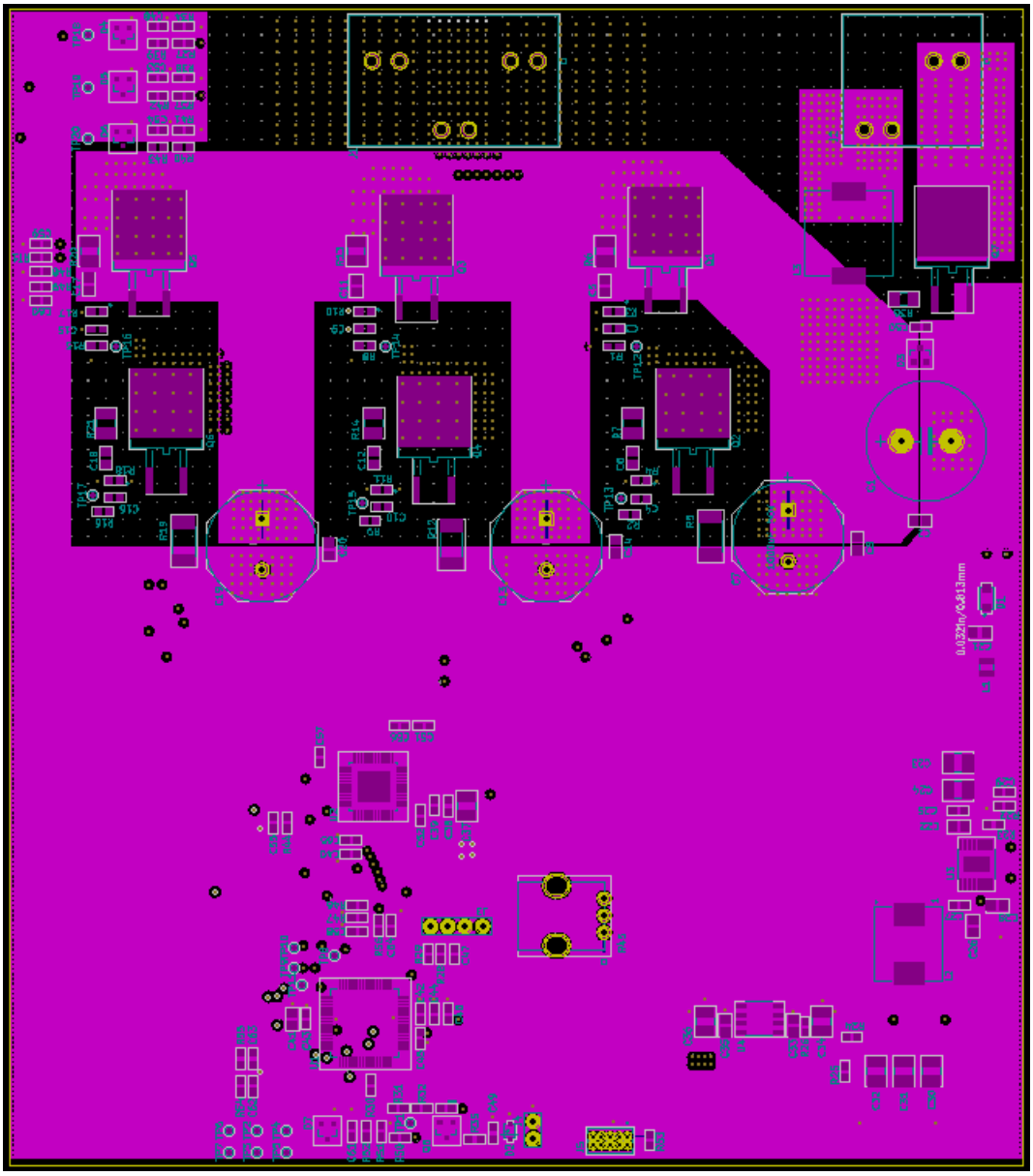


Figure C.3: Inner 2 Layer of Final PCB Layout

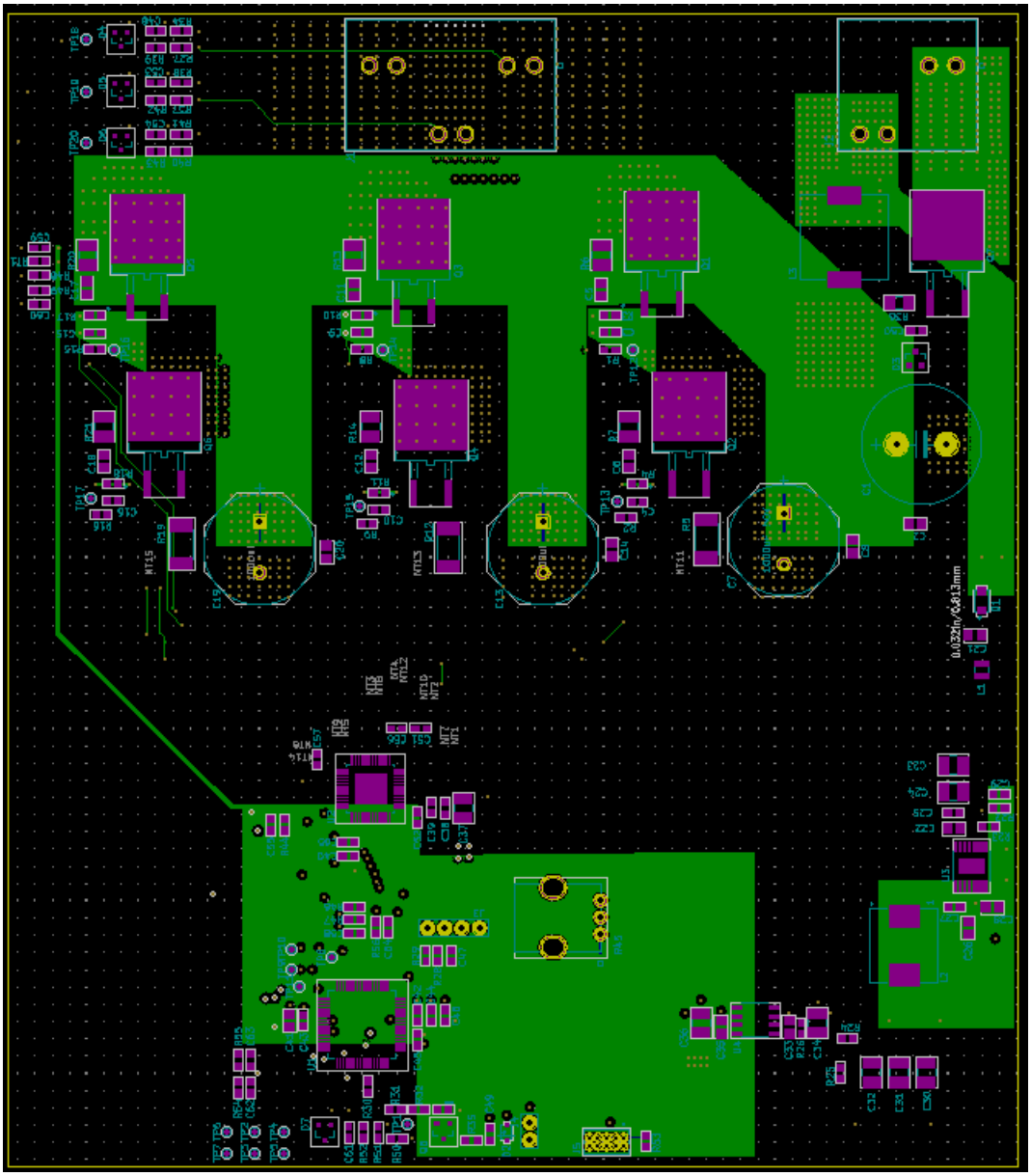


Figure C.4: Bottom Layer of Final PCB Layout

Appendix D

SUMMARY OF NXP AND TI SOFTWARE DIFFERENCES

The following tables break down the necessary software needed for interaction between the MCU and gate driver into the following six modules:

- Serial Peripheral Interface (SPI)
- Pulse Width Modulation (PWM)
- Current Sensing
- Gate Driver Enable
- Gate Driver Fault Detection (nFAULT/INT)

The tables compare the provided software from the NXP MCSXTE2BK142 and TI DRV8343S-Q1EVM evaluation boards for each of the previously listed modules. These tables are intended to serve as a reference for the necessary software modifications that must be made in order to incorporate the TI DRV8343-Q1 gate driver into the software used to operate the system designed in this project.

SPI

	NXP	TI

SPI Initialization	spi_aml.c - Line 51
	/*FUNCTION*****
	*
	* Function Name : SPI_AML_MasterInit
	* Description : Initializes the SPI as master.
	*
	*END*****/
	void SPI_AML_MasterInit(aml_instance_t instance,
	const spi_sdk_master_config_t *spiSdkMasterConfig,
	uint32_t sourceClockHz)
	{
	AML_ASSERT(instance < SPI_AML_DEV_CNT);
	AML_ASSERT(spiSdkMasterConfig != NULL);
	
	#if (SDK_VERSION == SDK_2_0)
	#if FSL_FEATURE_SOC_SPI_COUNT
	SPI_MasterInit(g_spiBases[instance], spiSdkMasterConfig, sourceClockHz);
	#elif FSL_FEATURE_SOC_DSPI_COUNT
	DSPI_MasterInit(g_dspiBases[instance], spiSdkMasterConfig, sourceClockHz);
	#endif
	#elif (SDK_VERSION == S32_SDK)
	LPSPI_DRV_MasterInit(instance, &g_lpspiState, spiSdkMasterConfig);
	AML_UNUSED(sourceClockHz);
	#endif
	}
		Init.c - Line 353
		/* USCI Initialization
		The USCI is initialized with the following setting
		3-pin, 8-bit SPI master,Clock polarity high,MSB
		*/
		void SPI_Init(void){
		/* SPI Ports Initialization * Port 3.0, 3.1, 3.2 is used for SIMO SOMI and SCLK respectively , Port 2.2 is Used for nSCS enable*/
		P3OUT &= ~(BIT0 | BIT2);
		P3DIR |= BIT0 | BIT2; /* Set SIMO, CLK as outputs */
		P3DIR &= ~BIT1; /* SOMI, slave out master in defined as input P3.1 */
		P3SEL |= BIT0 | BIT1; // P3.0,3.1 option select for UCBOSIMO and UCBOSOMI
		P3SEL |= BIT2; // P3.2 option select for UCBOCLK function
		P2OUT |= BIT2;
		P2DIR |= BIT2; // Set P2.2 to output direction for nSCS
		
		UCB0CTL1 |= UCSWRST; // **Put state machine in reset**
		UCB0CTL0 |= UCMST + UCSYNC + UCMSB; // 3-pin, 8-bit SPI master
		// Clock polarity high, MSB
		UCB0CTL1 |= UCSSEL_3; // MCLK
		UCB0BR0 = 10; // Master Clock divided by 10 used by USCI clk
		UCB0BR1 = 0; //
		UCB0CTL1 &= ~UCSWRST; // **Initialize USCI state machine**
		
		drv83xxSPISet(); // make nSCS pin of drv83xx low to start communication with master SPI;
		drv83xxSPIReset(); // make nSCS pin of drv83xx High to stop communication with master SPI;
		SPIDelay();
		}

Configure Gate Driver via SPI	N/A	drv8343.c - Line 555
		void drv83xx_Register_Write(void)
		{
		uint16_t regValue;
		// Read Register 0x00
		regValue = (Fault_Status_Reg.FSO_FAULT << 7) |
		(Fault_Status_Reg.FSO_GDF << 6) |
		(Fault_Status_Reg.FSO_CPUV << 5) |
		(Fault_Status_Reg.FSO_UVLO << 4) |
		(Fault_Status_Reg.FSO_OCP << 3) |
		(Fault_Status_Reg.FSO_OTW << 2) |
		(Fault_Status_Reg.FSO_OTSD << 1) |
		(Fault_Status_Reg.FSO_OL_SHT << 0);
		
		drv8343_SPI_Write(SPI_REG_FAULT_STAT,regValue);
		Reg_Map_Cache.FAULT_STAT = regValue;
		
		// Read Register 0x01
		regValue = (Diag_status_A_Reg.DSA_FAULT << 7) |
		(Diag_status_A_Reg.DSA_SA_OC << 6) |
		(Diag_status_A_Reg.DSA_SHT_BAT_A << 5) |
		(Diag_status_A_Reg.DSA_OL_PH_A << 4) |
		(Diag_status_A_Reg.DSA_VGS_LA << 3) |
		(Diag_status_A_Reg.DSA_VGS_HA << 2) |
		(Diag_status_A_Reg.DSA_VDS_LA << 1)|
		(Diag_status_A_Reg.DSA_VDS_HA << 0);
		drv8343_SPI_Write(SPI_REG_DIAG_STAT_A,regValue);
		Reg_Map_Cache.DIAG_STAT_A = regValue;

SPI (continued)

	NXP	TI

SPI Write	spi_aml.c - Line 327
	Note: Do to length only part of this function was included
	/*FUNCTION**********
	*
	* Function Name : SPI_AML_MasterTransferBlocking
	* Description : Performs blocking master transfer of data. The methods returns
	* when all data are sent and received.
	*
	*END**********/
	status_t SPI_AML_MasterTransfer(aml_instance_t instance,
	spi_aml_transfer_t *masterTransfer)
	{
	AML_ASSERT(instance < SPI_AML_DEV_CNT);
	AML_ASSERT(masterTransfer != NULL);
	
	#if (SDK_VERSION == SDK_2_0)
	status_t error;
	
	#if FSL_FEATURE_SOC_SPI_COUNT
	spi_transfer_t xfer;
	
	xfer.txData = masterTransfer->txBuffer;
	xfer.rxData = masterTransfer->rxBuffer;
	xfer.dataSize = masterTransfer->dataSize;
	xfer.flags = masterTransfer->configFlags;
	
	error = SPI_MasterTransferBlocking(g_spiBases[instance], &xfer);
	
	if (error == kStatus_SPI_Busy)
	{
	return kStatus_AML_SPI_Busy;
	}
	#elif FSL_FEATURE_SOC_DSPI_COUNT
	dspi_transfer_t xfer;
	
	xfer.txData = masterTransfer->txBuffer;
	xfer.rxData = masterTransfer->rxBuffer;
	xfer.dataSize = masterTransfer->dataSize;
	xfer.configFlags = masterTransfer->configFlags;
	
	error = DSPI_MasterTransferBlocking(g_dspiBases[instance], &xfer);
	
	if (error == kStatus_DSPI_Busy)
	{
	return kStatus_AML_SPI_Busy;
	}
	#endif
	if (error == kStatus_Success)
	{
	return kStatus_Success;
	}
	else if (error == kStatus_InvalidArgument)
	{
	return kStatus_InvalidArgument;
	}
	else
	{
	return kStatus_AML_SPI_Error;
	}
	#elif (SDK_VERSION == S32_SDK)
	status_t error;
		drv8343.c - Line 929
		void drv8343_SPI_Write(uint8_t regAddr, uint8_t data)
		{
		uint8_t dataLSB , dataMSB;
		dataMSB = regAddr;
		dataLSB = data;
		
		SPI_WriteTwoBytes(dataLSB , dataMSB);
		}

SPI (continued)		
	NXP	TI

Gate Driver Register Write	N/A	global.c - Line 131
		/*function
		* drv83xx_registerWrite()
		* Device specific register write funtion
		* */
		void drv83xx_registerWrite(unsigned char address, unsigned int value)
		{
		drv83xx_regToCache(address, value);
		
		/* Cache the value in the firmware */
		if (ApplicationStatus.fault != POWER_SUPPLY)
		{
		/* Write the value to the device */
		drv8343_SPI_Write(address, value);
		}
		}

SPI Read	N/A	drv8343.c - Line 921
		unsigned short drv8343_SPI_Read(uint8_t regAddr)
		{
		uint16_t data;
		uint8_t value;
		data = SPI_ReadWord(regAddr);
		value = data & 0x00FF; // DRV8343 is a 8 bit data
		return value;
		}

PWM

	NXP	TI

PWM Initialization	gd3000_init.c - Line 92
	/*******recover PTA2 & PTA3 as PWM output*******/
	PINS_DRV_SetMuxModeSel(PORTA,2, PORT_MUX_ALT2); /*configure as FTM channel PWM output*/
	PINS_DRV_SetMuxModeSel(PORTA,3, PORT_MUX_ALT2); /*configure as FTM channel PWM output*/
	
	/*******recover PTD2 & PTD3 as PWM output*******/
	PINS_DRV_SetMuxModeSel(PORTD,2, PORT_MUX_ALT2); /*configure as FTM channel PWM output*/
	PINS_DRV_SetMuxModeSel(PORTD,3, PORT_MUX_ALT2); /*configure as FTM channel PWM output*/
	
	/*******recover PTC6 & PTC7 as PWM output*******/
	PINS_DRV_SetMuxModeSel(PORTC,6, PORT_MUX_ALT4); /*configure as FTM channel PWM output*/
	PINS_DRV_SetMuxModeSel(PORTC,7, PORT_MUX_ALT4); /*configure as FTM channel PWM output*/
		Init.c - Line 313
		/* PWM Initialization Using Ports P1.3, P1.5 , P2.5 for A , B , C Phases High side , P1.2, P1.4 , P2.4 for A , B , C Phases Low side respectively */
		P1OUT &= ~(BIT2 | BIT3 | BIT4 | BIT5);
		P1DIR |= BIT2 | BIT3 | BIT4 | BIT5; // P1.2,P1.3,P1.4,P1.5 to output direction for driving PWM to drv83xx
		P1SEL |= BIT2 | BIT3 | BIT4 | BIT5; // Set P1.2,P1.3,P1.4,P1.5 for Generating PWM for A phase High side and B phase High side respectively
		P2OUT &= ~(BIT4 | BIT5);
		P2DIR |= BIT4 | BIT5; // P2.4,P2.5 to output direction for driving PWM to drv83xx
		P2SEL |= BIT4 | BIT5; // Set P2.4,P2.5 for Generating PWM for C phase High side

Configure Gate Driver PWM Mode	N/A	drv8343.c - Line 56
		/*function
		* drv83xx_set_Six_PWM_Mode(void)
		* This function when called sets the mode to six PWM mode
		* */
		
		void drv83xx_set_Six_PWM_Mode(void)
		{
		unsigned int regValue;
		regValue = drv83xx_cachetoReg(SPI_REG_DRV_CTRL_1);
		regValue &= ~DRV8343S_PWM_MODE_MASK; // Six PWM Mode
		drv83xx_registerWrite(SPI_REG_DRV_CTRL_1, regValue); // Write back the updated value to the cache and SPI
		}

PWM Commutation	ftm_pwm_driver.c - Line 634
	/*FUNCTION**********
	*
	* Function Name : FTM_DRV_FastUpdatePwmChannels
	* Description : This function will update the duty cycle of PWM output for multiple channels.
	*
	* The main differences between this function and FTM_DRV_UpdatePwmChannel is the execution speed. This
	* feature makes this function ideal for applications like motor controlling.
	* The downside is the low flexibility of the parameters (this function accept only updates in ticks).
	*
	* Implements : FTM_DRV_FastUpdatePwmChannels_Activity
	*END**********/
	status_t FTM_DRV_FastUpdatePwmChannels(uint32_t instance,
	uint8_t numberOfChannels,
	const uint8_t * channels,
	const uint16_t * duty,
	bool softwareTrigger)
	{
	FTM_Type * ftmBase = g_ftmBase[instance];
	DEV_ASSERT(instance < FTM_INSTANCE_COUNT);
	DEV_ASSERT(numberOfChannels <= FEATURE_FTM_CHANNEL_COUNT);
	uint8_t i;
	
	for (i = 0U; i < numberOfChannels; i++)
	{
	((ftmBase)->CONTROLS[channels[i]].CnV) = duty[i];
	}
	
	if (softwareTrigger)
	{
	ftmBase->SYNC |= FTM_SYNC_SWSYNC_MASK;
	}
	
	return STATUS_SUCCESS;
	}
		global.c - Line 561
		Note: The following code demonstrates only one of the six steps of the commutation sequence
		if(SensorlessTrapController.PWM_Mode == SIX_PWM_MODE)
		{
		/* Implementing Synchronous PWM i.e. to Toggle between High side and low side of a Phase with Dead Band*/
		
		switch(commState)
		{
		case 1: /* B-C */
		
		/* Reset switches for phase A (LOW-HIGH) */
		P2OUT &= ~(BIT4 | BIT5); /* Reset bits P2.4 , P2.5 */
		P2SEL &= ~(BIT4 | BIT5); /* Select P2.4 , P2.5 as I/O Function for Phase A*/
		
		/* Reset switches for phase C (HIGH)*/
		P1OUT &= ~(BIT3); /* Reset bits P1.2 , P1.3 */
		P1SEL &= ~(BIT3); /* Select P1.2 , P1.3 as I/O Function for Phase C*/
		
		P1SEL |= BIT4 | BIT5; /* Select Synchronous PWM for B phase*/
		P1OUT |= BIT2; /* Set Low side of C phase */
		SensorlessTrapController.RotationCount++;
		break;

Current Sensing

	NXP	TI

Initial CSA Reading	N/A	Init.c - Line 225
		ADC12MCTL1 = ADC12INCH_4; // channel = A4 (Read the CSA reading from Phase A)
		ADC12MCTL2 = ADC12INCH_5; // channel = A5 (Read the CSA reading from Phase B)
		ADC12MCTL3 = ADC12INCH_4; // channel = A4 (Read the CSA reading from Phase A)

Read Current Shunt	meas_s32k.c - Line 276
	Note: The following code is taken from the 3-Shunt project, in this project 2 phases are sampled at a time, and the third is calculated. Three pairs are possible (AB, AC, BC), only one pair is shown below
	Note: External current sense amplifiers are used in the NXP evaluation board, the code below shows the reading of external CSA's by the MCU and is not direct communication between the MCU and gate driver
	tBool MEAS_Get3PhCurrent(measModule_t *ptr, SWLIBS_3Syst_FLT *i, tU16 svmSector)
	{
	uint16_t PhaseA_Current;
	uint16_t PhaseB_Current;
	uint16_t PhaseC_Current;
	
	switch(drvFOC.svmSector)
	{
	case 2:
	case 3:
	// Read ADC0_CH9 value - PhaseA Current
	ADC_DRV_GetChanResult(INST_ADCONV0, 1, &PhaseA_Current);
	// Read ADC1_CH6 value - PhaseC Current
	ADC_DRV_GetChanResult(INST_ADCONV1, 1, &PhaseC_Current);
	
	ptr->measured.fltPhA.raw = MLIB_Mul(((tFloat)MLIB_Div((tFloat)(PhaseA_Current & 0x00000FFF), (tFloat)0x00000FFF)), I_MAX);
	ptr->measured.fltPhC.raw = MLIB_Mul(((tFloat)MLIB_Div((tFloat)(PhaseC_Current & 0x00000FFF), (tFloat)0x00000FFF)), I_MAX);
	
	i->fltArg1 = MLIB_Mul(MLIB_Sub(ptr->offset.fltPhA.fltOffset,ptr->measured.fltPhA.raw), 2);
	i->fltArg3 = MLIB_Mul(MLIB_Sub(ptr->offset.fltPhC.fltOffset,ptr->measured.fltPhC.raw), 2);
	i->fltArg2 = MLIB_Add(-1 * i->fltArg1, -1 * i->fltArg3);
	break;
		Global.c - Line 925
		/*function
		* ReadCurrentshunt()
		* Reads CSA value and triggers OC faults for Motor current greater than Set Limit
		* */
		void ReadCurrentShunt()
		{
		ADC12MCTL0 = ADC12INCH_4 + ADC12EOS; // channel = A4 (Read the CSA reading from Phase A for over current protection) , End of Sequence
		ADC12CTL0 |= ADC12ENC; // Enable Conversions
		ADC12CTL0 |= ADC12SC; // Start sampling of channels
		while(ADC12CTL1 & ADC12BUSY_L)
		{
		}
		;
		SensorlessTrapController.MotorPhaseCurrent = ADC12MEM0 & 0x0FFF; /* Filter the result and read only last 12 bits because MSP430F5529 has 12bit ADC*/
		
		ADC12CTL0 &= ~ADC12ENC; // Start sampling of channels
		ADC12CTL0 &= ~ADC12SC; // Start sampling of channels
		
		if(SensorlessTrapController.SPIVariant == TRUE) // If Current sense is not in phase shunt remove the offset.
		{
		if(drv83xx_is_CSA_BI_DIR_Mode()) // If the Device is set in Bidirectional CSA mode
		{
		SensorlessTrapController.MotorPhaseCurrent -= 2048; // subtracting the bias, Vref/2 1.65v is added as bias voltage to support bidirectional current sensing
		
		}
		else
		{
		SensorlessTrapController.MotorPhaseCurrent -= 3723; // subtracting the bias, 3.0v is added as bias voltage to support unidirectional current sensing
		}
		}
		else
		{
		SensorlessTrapController.MotorPhaseCurrent -= 2048; // subtracting the bias, Vref/2 1.65 is added as bias voltage to support bidirectional current sensing
		}
		SensorlessTrapController.MotorPhaseCurrent = abs(
		SensorlessTrapController.MotorPhaseCurrent);
		if((SensorlessTrapController.MotorPhaseCurrent >
		SensorlessTrapController.MotorPhaseCurrentLimit) && (ApplicationStatus.fault == NOFAULT))
		/* Motor Phase Current Limit*/
		{
		ApplicationStatus.previousstate = ApplicationStatus.currentstate;
		ApplicationStatus.currentstate = FAULT;
		ApplicationStatus.fault = OVERCURRENT;
		}
		}

Gate Driver Enable

	NXP	TI
Initialization	pin_mux.c - Line 200 { .base = PORTB, .pinPortIdx = 4u, .pullConfig = PORT_INTERNAL_PULL_NOT_ENABLED, .passiveFilter = false, .driveSelect = PORT_LOW_DRIVE_STRENGTH, .mux = PORT_MUX_AS_GPIO, .pinLock = false, .intConfig = PORT_DMA_INT_DISABLED, .clearIntFlag = false, .gpioBase = PTB, .direction = GPIO_OUTPUT_DIRECTION, .digitalFilter = false, .initValue = 0u, }, tpp.c - Line 107 /* EN1 pin. */ GPIO_AML_SetDirection(drvConfig->en1PinInstance, drvConfig->en1PinIndex, gpioDirDigitalOutput); /* EN2 pin. */ GPIO_AML_SetDirection(drvConfig->en2PinInstance, drvConfig->en2PinIndex, gpioDirDigitalOutput);	Init.c - Line 118 /*Disable the gate drivers by Default */ P1DIR |= BIT6; P1OUT &= ~BIT6;
Enable Gate Driver	tpp.c - Line 280 GPIO_AML_SetOutput(drvConfig->en1PinInstance, drvConfig->en1PinIndex); GPIO_AML_SetOutput(drvConfig->en2PinInstance, drvConfig->en2PinIndex);	mdbu_global.c - Line 268 drv83xx_setGPIO(0x01, EN_DRV, *data);

nFAULT / INT

	NXP	TI

Configure nFAULT Input Pin	pin_mux.c - Line 404
	{
	.base = PORTB,
	.pinPortIdx = 12u,
	.pullConfig = PORT_INTERNAL_PULL_NOT_ENABLED,
	.passiveFilter = false,
	.driveSelect = PORT_LOW_DRIVE_STRENGTH,
	.mux = PORT_MUX_AS_GPIO,
	.pinLock = false,
	.intConfig = PORT_DMA_RISING_EDGE,
	.clearIntFlag = true,
	.gpioBase = PTB,
	.direction = GPIO_INPUT_DIRECTION,
	.digitalFilter = true,
	},
		Init.c - Line 321
		/* Configure Port 2.7 as input for sensing faults and enable Interrupt */
		P2DIR &= ~BIT7;
		P2REN |= BIT7; // When a GPIO pin is configured as Input, Enable resistance to the pin by setting Px.REN and PX.OUT
		P2OUT |= BIT7;
		P2IE |= BIT7;
		P2IFG |= 0x00;
		P2IES |= BIT7;

Interrupt Enable	gd3000_init.c - Line 117 /* * Enable interrupt when rising edge is detected on * PTB12 for GD3000 pre-driver interrupt input, rising edge trigger */ PINS_DRV_SetPinIntSel(GD3000_INT_PORT, GD3000_INT_PIN_NUM, PORT_INT_RISING_EDGE); /* set the PORTB IRQ interrupt priority */ INT_SYS_SetPriority(GD3000_INT_IRQn,1); /* Enable PORTB IRQ interrupt */ INT_SYS_EnableIRQ(GD3000_INT_IRQn);	SEE ABOVE

Appendix E

COMPONENT COST ESTIMATION

Ref Des	Qty	Manufacturer	Manufacturer PN	Unit Price (Prototype)	Unit Price (Mid-Volume)	Unit Price (High-Volume)	Total Cost (Prototype)	Total Cost (Mid-Volume)	Total Cost (High-Volume)
C1	1	PANASONIC	EEU-FC1H222C	2.28	0.969	0.859	2.28	0.969	0.859
C2, C8, C14, C20, C21, C22, C33	7	MURATA	GCM21BR72A104KA37L	0.133	0.045	0.045	0.931	0.315	0.315
C23, C24, C30, C31, C32, C34, C36	7	MURATA	GRM32ER71H106KA12L	0.896	0.469	0.469	6.272	3.283	3.283
C25, C47	2	TDK	C1608X5R1H105K080AB	0.082	0.042	0.029	0.164	0.084	0.058
C26	1	TAIYO YUDEN	UMK212BB7225KG-T	0.155	0.085	0.069	0.155	0.085	0.069
C27	1	MURATA	GCM188R71C474KA55D	0.177	0.054	0.039	0.177	0.054	0.039
C28, C41	2	TDK	CGA4J3X7R1C475K125AB	0.155	0.084	0.084	0.31	0.168	0.168
C29	1	AVX	06035A360JAT2A	0.094	0.047	0.028	0.094	0.047	0.028
C3, C4, C9, C10, C15, C16, C42, C46	8	AVX	06033A102FAT2A	0.307	0.115	0.112	2.456	0.92	0.896
C35	1	TDK	CGA4J2X7R1C684K125AA	0.085	0.044	0.032	0.085	0.044	0.032
C37	1	TDK	C3225X7S2A475K200AE	1.11	0.725	0.725	1.11	0.725	0.725
C38	1	TAIYO YUDEN	HMK107B7104KAHT	0.078	0.036	0.026	0.078	0.036	0.026
C39	1	WURTH	885012106022	0.386	0.309	0.278	0.386	0.309	0.278
C40, C55	1	KEMET	C0603C104K9RACTU	0.06	0.03	0.023	0.06	0.03	0.023
C43, C44, C45, C50, C58, C59	6	AVX	06035C104K4Z2A	0.122	0.119	0.023	0.732	0.714	0.138
C48, C53, C54, C60, C61	5	KEMET	C0603C820K5GACTU	0.11	0.046	0.035	0.55	0.23	0.175
C49	1	AVX	06035C103JAT2A	0.039	0.018	0.011	0.039	0.018	0.011
C5, C6, C11, C12, C17, C18	6	AVX	08051C332KAT2A	0.072	0.029	0.017	0.432	0.174	0.102
C51, C56, C57	3	WURTH	DNP - 885012206034	0.022	0.018	0.015	0.066	0.054	0.045
C52	1	TDK	C1608X7S2A473K080AB	0.081	0.039	0.028	0.081	0.039	0.028
C62, C63, C64	3	WURTH	885012206036	0.022	0.017	0.014	0.066	0.051	0.042
C65	1	SAMSUNG	CL10B105KQ8NNNC	0.025	0.013	0.006	0.025	0.013	0.006
C7, C13, C19	3	PANASONIC	EEU-FC1H102	1.07	0.553	0.536	3.21	1.659	1.608
D1	1	NEXPERIA	PMEG10010ELR	0.267	0.097	0.075	0.267	0.097	0.075
D2	1	NEXPERIA	BZX384-C30	0.169	0.041	0.027	0.169	0.041	0.027
D3	1	ON	BZX84C15LT3G	0.127	0.033	0.021	0.127	0.033	0.021
D4, D5, D6, D7	4	NEXPERIA	BZX84-A5V1	0.353	0.122	0.122	1.412	0.488	0.488

Ref Des	Qty	Manufacturer	Manufacturer PN	Unit Price (Prototype)	Unit Price (Mid-Volume)	Unit Price (High-Volume)	Total Cost (Prototype)	Total Cost (Mid-Volume)	Total Cost (High-Volume)
J1	1	Phoenix	1782909	9.55	5.74	5.74	9.55	5.74	5.74
J2	1	Phoenix	1709681	9.3	6.61	6.61	9.3	6.61	6.61
J3	1	WURTH	61300411121	0.158	0.091	0.091	0.158	0.091	0.091
J4	1	WURTH	61300211121	0.11	0.063	0.063	0.11	0.063	0.063
J5	1	AMTEC	FTSH-105-01-F-D	1.33	0.826	0.701	1.33	0.826	0.701
L1	1	WURTH	78438323010	2	1.46	1.27	2	1.46	1.27
L2	1	EATON	HCM1A1104-101-R	2.05	1.34	1.26	2.05	1.34	1.26
L3	1	WURTH	744309047	2.26	1.65	1.65	2.26	1.65	1.65
Q1, Q2, Q3, Q4, Q5, Q6, Q7	7	NEXPERIA	BUK762R4- 60E	1.52	1.52	1.52	10.64	10.64	10.64
Q8	1	NEXPERIA	2N7002	0.23	0.069	0.045	0.23	0.069	0.045
R1, R2, R8, R9, R15, R16	6	VISHAY	RCS060333R0FKEA	0.131	0.015	0.012	0.786	0.09	0.072
R22	1	WALSIN	WR06X1004FTL	0.13	0.13	0.005	0.13	0.13	0.005
R23	1	KOA SPEER	RK73H1JTTD1693F	0.022	0.004	0.002	0.022	0.004	0.002
R24, R30, R31, R35	4	PANASONIC	ERJ-U030R00V	0.045	0.014	0.011	0.18	0.056	0.044
R25	1	KOA SPEER	RK73H1JTTD1301F	0.022	0.004	0.002	0.022	0.004	0.002
R26, R33, R48, R51, R53	5	YAGEO AMERICA	RC0603FR-0710KL	0.018	0.003	0.003	0.09	0.015	0.015
R27, R37, R40, R50	4	VISHAY	CRCW060324K0FKEA	0.034	0.006	0.005	0.136	0.024	0.02
R28, R29, R39, R42, R43, R47, R49, R52	8	YAGEO AMERICA	AC0603FR-07120RL	0.025	0.003	0.003	0.2	0.024	0.024
R3, R4, R10, R11, R17, R18	6	BOURNS	CR0603-FX-1003ELF	0.019	0.002	0.002	0.114	0.012	0.012
R32	1	YAGEO AMERICA	RC0603FR-071KL	0.018	0.004	0.002	0.018	0.004	0.002
R34, R38, R41, R46	4	YAGEO AMERICA	RC0603DR-072K2L	0.049	0.013	0.009	0.196	0.052	0.036
R36	1	ROHM	KTR18EZPJ104	0.124	0.019	0.017	0.124	0.019	0.017
R44	1	VISHAY	CRCW060310K0JNEA	0.028	0.005	0.005	0.028	0.005	0.005
R5, R12, R19	3	VISHAY	WSLP25121L000FEA	1.35	0.707	0.634	4.05	2.121	1.902
R45	1	ALPS ALPINE	RK09D1130C3W	0.68	0.577	0.577	0.68	0.577	0.577
R54, R55, R56	3	VISHAY	CRCW060356R0JNEA	0.032	0.006	0.003	0.096	0.018	0.009
R6, R7, R13, R14, R20, R21	6	ROHM	ESR25JZPJ2R2	0.303	0.045	0.035	1.818	0.27	0.21
RT1	1	VISHAY	NTCS0603E3103JLT	0.287	0.172	0.147	0.287	0.172	0.147
U1	1	NXP	FS32K142HAT0MLHT	7.42	5.74	5.74	7.42	5.74	5.74
U2	1	TI	DRV8343SPHPRQ1	6.7	3.59	3.59	6.7	3.59	3.59
U3	1	TI	LM46000AQPWPRQ1	2.32	2.32	2.32	2.32	2.32	2.32
U4	1	ON	MC33375D-5.0G	0.823	0.39	0.368	0.823	0.39	0.368