SIMBA

Sitara Integrated Motor Boosterpack Assignment

UTDesign Spring 2023

UTDesign Team 1607
Isaac Brooks (CE)
Jenna Evans (CE)
Coleman Gamble (EE)
Mijwad Kabir (CE)
Victor Lopez (EE)
Shruthi Subramanium (CE)

Corporate Sponsor:
Randy Rosales
Anita Pratti

Special Thanks To:
James Lockridge

Faculty Advisor:
Dr. Neal Skinner

Instructor:
Dr. Neal Skinner

Final Report for Team 1607's ECE Senior Design Project.

Department of Electrical and Computer Engineering
University of Texas at Dallas
May 12, 2023

Abstract

The Sitara AM263x series is a line of high-performance microcontrollers (MCUs) from Texas Instruments (TI) designed for a variety of automotive and industrial projects, including advanced motor control. Our goal is to utilize the existing DRV8323RH boosterpack evaluation module (EVM) in combination with the Sitara AM263x LaunchPad to spin a brushless DC motor (BLDC). We have worked with our TI corporate sponsors to research various modes of motor operation, program the software development kit, and flexibly design the system to work with various encoders and BLDC motors. Our system's end product will provide high-precision configurable motor control for a weighted robotic arm using different positional feedback sensors. The Sitara MCU on the AM263 LaunchPad will process the feedback signals and adjust the motor driver inputs accordingly for this high-torque application.

Acknowledgements

We would like to express our thanks to some individuals and organizations for their guidance and help with the project. First, we would like to thank our professor and mentor, Dr. Skinner, for enthusiastically advising us each week and working with us to ensure we would have our deliverables on time. Next, we would like to thank Anita Pratti and Randy Rosales for being our corporate sponsors; Randy has been very attentive and helpful throughout the year, and he has been patient and understanding when we needed to make adjustments to our project. We would also like to give a special thanks to James Lockridge, who has taken many hours out of his time to help us in the lab, provided us with equipment when needed, and continuously given us valuable insight and support. Finally, we would like to thank the University of Texas at Dallas and Texas Instruments for the opportunity to work on this project.

Table of Contents

List of Tables

List of Figures

1. Introduction

This project features three main components: the Sitara AM263x LaunchPad (LP), the DRV8323RH, and a BLDC motor. The line of Sitara MCUs is relatively new to the industry and was made to meet the needs of various complex real-time processes. While general purpose development boards exist, there is a lack of documentation and application-specific example projects for customers to use as a reference design. This results in customers being unable to smoothly use the software development kit (SDK) or design a new product using the AM263x.

The DRV8323RH boosterpack uses the DRV8323RH gate driver to drive a 3 phase BLDC motor. This boosterpack supports both sensored and sensorless projects, and the team chose to use hall sensors for initial testing and incorporated encoder signals for higher resolution in the final design. The boosterpack meshes with the LaunchPad and is configurable via header pins, despite being originally made compatible with an older microcontroller (MSP5529) and not with the Sitara AM263x.

To resolve these issues, Texas instruments asked the team to create or configure an existing boosterpack to function as a motor controller that is compatible with the AM263x. The DRV8323RH is completely compatible with the header functions on the Sitara LaunchPad, and the design is motor agnostic, meaning that any 16-24V BLDC motor with hall sensors can be used. To aid customers and fellow engineers, this project will feature an open source GitHub release to support TI's Sitara products' documentation. This process will further function as a stepping stone for eventually creating a fully-integrated design with the MCU.

2. Review of Conceptual and Preliminary Design

2.1 Problem Analysis

2.1.1 Summary of Specifications

Our system will raise a weighted arm with desired speed and accuracy using the following specifications:

- Control a high-torque BLDC motor with a varying load using the Sitara AM263x LaunchPad in collaboration with a motor driver board
- Ability to scale driver section for ranges of motor classes
- Fully programmable for different encoders
  - HDSL/EnDAT libraries/APIs available in Software Development Kit (SDK)
  - eQEP
- Acquire torque, shaft position, and shaft velocity values to control motor commutation through the SDK

The final system will allow us to present a compact alternative to existing solutions through a public domain GitHub release.

2.1.2 Main Features of Design Problems

In regards to the AM263x MCU, there are many features we took into consideration that contributed to future design adjustments throughout the project. There are 324 ball pins in the MCU chip alone, significantly increasing PCB design difficulty when attempting an independent, fully integrated motor control system. Combining the MCU with the motor driver, external MOSFETs, and other passive components onto a single board would require time and experience that the team does not have. Therefore, utilizing the AM263x LaunchPad with the DRV8323RH boosterpack as described in later sections would be a viable solution to this design problem, serving as a stepping stone for further development. Additionally, the Sitara line of MCUs are relatively new to the market with minimal software documentation, so programming the board for motor control will require many levels of troubleshooting.

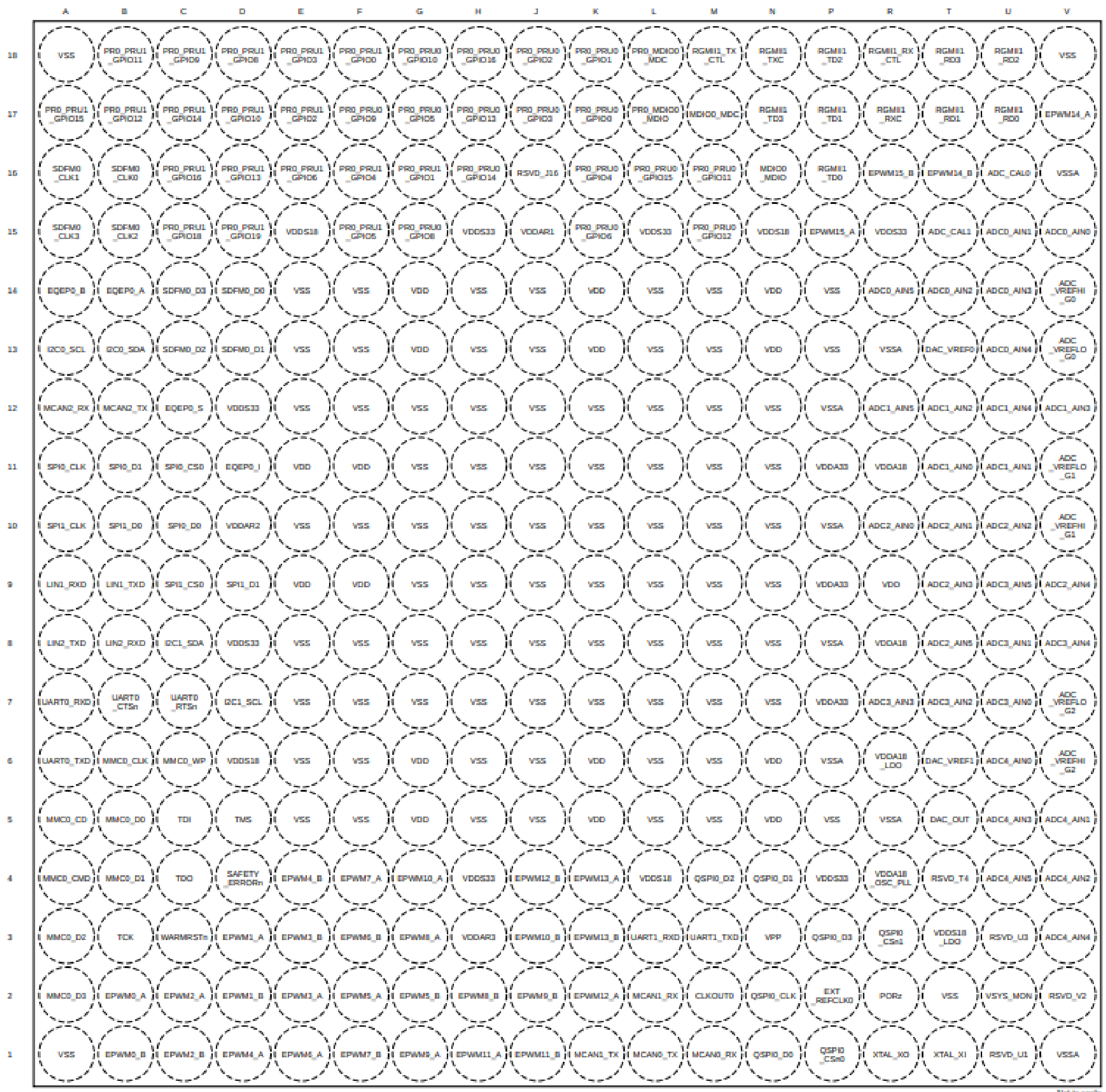


Figure 2.1. AM2634 ZCZ Package Pin Diagram

Another design problem involves making the system motor agnostic, requiring an external FET driver topology in which the FETs are separated from the motor driver chip. This allows the designer to select FETs based on current and voltage ratings that align with the desired motor to avoid under current and under voltage faults. However, routing the FETs externally will take up more board area and increase the design complexity of an integrated motor control board. Again, this can be temporarily solved by using a DRV83xx boosterpack EVM. Another factor to consider is that high torque BLDC motors often require larger power supplies that are not standard in the UTDesign lab. Additionally, the higher current and voltage ratings for features of the MCU, DRV, and motor may require larger wire sizes or copper traces. Thicker wires can handle more current, and shorter wires will reduce voltage drop and inductance.

2.1.3 Technical Approach

The project can be split into hardware and software aspects to technically approach the design. This will encompass configuring the AM263x LaunchPad to work with a motor driver system.

2.1.3a Hardware

The hardware approach is based on the feasibility of designing an integrated MCU with a motor driver system. This would involve advanced research into PCB design considerations, presenting the issues discussed in the previous section. Thus, the team will attempt an integrated design while existing boards can be used as a backup plan. This alternative solution will utilize the necessary functions from the provided AM263x LaunchPad to simplify the approach while still demonstrating the capabilities of the Sitara MCU. For the motor driver portion, acquiring an existing DRV83xx boosterpack EVM to test compatibility of components with the overall system prior to a custom design will streamline troubleshooting issues and minimize errors. Finally, since scaling the system for different motor classes is a requirement, our approach was to select a high torque motor in the middle range of motor classes.

2.1.3b Software

The SDK for the Sitara line of MCUs hosts a large range of examples and API descriptions to reference when programming the launchpad. Thus, our software approach is to utilize the provided examples that individually demonstrate the necessary SOC and board peripherals for our project; this includes the ADC, EPWM, eQEP, GPIO peripherals. After running the examples separately, we will make adjustments in the code to integrate them into a single program.

2.2 Summary of Requirements

The main performance requirements for the project include using an external gate driver with MOSFETs, a BLDC motor rated for 24V at the most, hall sensors, and an encoder with at least 1024 bits of resolution. If the motor is 24V, the maximum voltage the system should be able to handle due to voltage spikes in the transient is approximately double, or 50V. The DRV83232RH has a maximum threshold of 65V, which fits this requirement.

2.3 Decision Analysis

2.3.1 Solution Alternatives

The alternative solutions for the MCU, sensing, and driver topologies are discussed in this section.

2.3.1a MCU

The AM243x is an alternative to the AM263x. Both are part of the Sitara MCU family, but the AM243x uses intermediary current feedback with shunt resistors, while the AM263x uses direct current feedback. The table below shows a further comparison.

AM243x	AM263x
• Superse device • Use: Multi-axis controller • Industrial Connectivity: at least 2 instances of ICSS • Faser, more cores • System Memory: depending on which 243x, could get DDR4 controller and need external memory to do processing that can't be done with just embedded SRAM • Sort of bridges together supervisory and embedded to pair to multi-axis devices • More of a standalone controller / amplifier set with attached motors and feedback sensors - a distributed design • Target: 3-axis motors	• Subset Device • Use: Single-axis controller • Industrial Connectivity: 1 instance of ICSS • Slightly slower, less cores • No DDR controller • Smaller in terms of # of IO, cores, and control accelerators on the chip - tailored for instance where single MCU tethered to single motor axis - more likely embedded into an actuator (actuator/controller combo) • Target: industrial and automotive single-axis design
Same peripherals, but scaled based on the number of cores	

Table 2.1. Major Differences/Similarities of the Sitara MCUs

2.3.1b Sensing

The commutation method is one of the most important aspects of controlling a motor. It refers to the way the system processes the location of the motor shaft. Although there are sensored and sensorless ways to help the motor define its position, the team chose to use sensors for the sake of reliability and complexity. There are four main methods of commutation. The first method is using a trapezoidal signal with hall sensors, and the second method is to use a sinusoidal signal with hall sensors. The difference between the trapezoidal/sinusoidal methods is that for the former, the MOSFETs switch on and off and create a trapezoidal signal, whereas for the latter, the curve of the switching time of the MOSFETs is used to create a sinusoidal signal. The third method is a sinusoidal signal with hall sensors with an encoder or resolver. Encoders provide better resolution than hall sensors, but there are two kinds of encoders to consider: absolute and incremental/relative. The absolute encoder gives the absolute position of the motor and is more complex and expensive, while the incremental encoder only gives the relative position and needs hall sensors to initialize the loop with an absolute position. The final method is the sensorless option, field oriented control (FOC). This method uses an encoder to read in the position and then uses a PI controller to determine the location after.

For current sensing, the team chose to utilize the built-in ADC in the AM263x LP that is best suited for direct current feedback.

2.3.1c Motor Driver

In the figure below, there are three gate driver topologies: the external FET, integrated FET: multi-chip module (MCM), and integrated FET: monolithic. In the external FET topology, the gate driver and FETs are separate packages and can be chosen independently. In the integrated FET MCM topology, the driver and MOSFETs are on the same package but not the same die. The monolithic topology contains the driver and FETs on the same package and die.

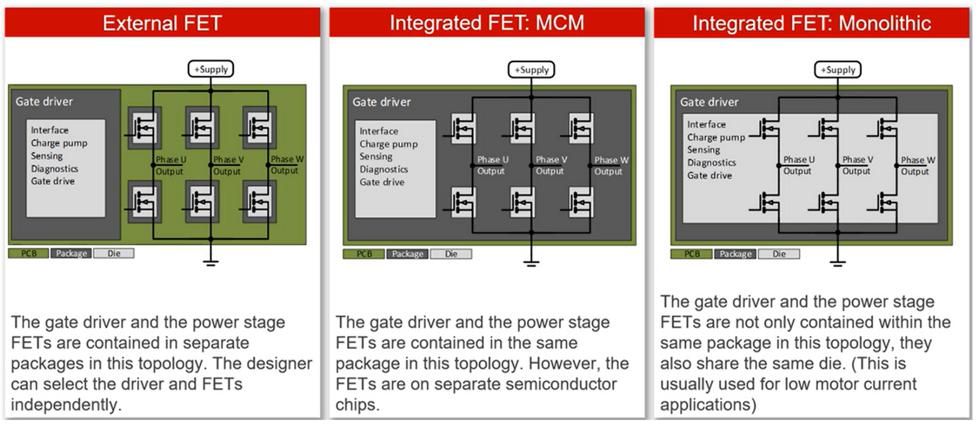


Figure 2.2. Driver Topologies

When choosing an existing DRV boosterpack EVM from TI, we considered a few options. The DRV8323RH is an external FET driver with a maximum voltage of 60V. Some alternatives to the DRV8323RH include the DRV8316 and the DRV8353. The DRV8316 is an integrated driver, while the DRV8353 works very similarly to the DRV8323; an external FET driver with a maximum voltage is approximately 90V.

2.3.2 Effectiveness Criteria

Our team compiled the most important requirements for our project and weighted them as shown below:

1. Complexity: 25%
2. High Torque Suitability: 25%
3. Cost: 20%
4. Voltage and Current Rating: 15%
5. Ease of Use: 15%

We created the table below to score our project on a scale of 1 to 5, where 1 is the lowest score and 5 is the highest. The table is to show where we could improve our project if more time was given.

ID	Criteria	Weight	Score	Weighted Score
1	Complexity	25%	4.5	22.5%
2	High Torque Suitability	25%	4.5	22.5%
3	Cost	20%	3	12%
4	Voltage and Current Rating	15%	4	12%
5	Ease of Use	15%	4.5	13.5%
Total				82.5%

Table 2.2. Overall Project Assessment

The motor boosterpack meets all of our requirements but could improve on cost. With some light research, we can see that there are more inexpensive choices for a motor control, but aren't as configurable as our project. Cost could be improved if we were able to use the processor alone to run all of our commands instead of the MCU, but the scope of a project like that would not be suitable for a timeline of one academic year. We will review the rest of the components in this report (MCU, DRV8323, motor) using the same benchmarks.

ID	Criteria	Weight	Score	Weighted Score
1	Complexity	25%	3	15%
2	High Torque Suitability	25%	4.5	22.5%
3	Cost	20%	3.5	14%
4	Voltage and Current Rating	15%	4	12%
5	Ease of Use	15%	2.5	7.5%
Total				71%

Table 2.3. Sitara MCU Assessment

Overall the Sitara Microcontroller unit is very complex and needs a lot of familiarity to use efficiently, but this is because it is a new MCU from TI and there exists little documentation and reference designs/example programs. It is somewhat cost effective for the flexibility of its uses, but it is on the higher price end for an MCU.

ID	Criteria	Weight	Score	Weighted Score
1	Complexity	25%	5	25%
2	High Torque Suitability	25%	4.5	22.5%
3	Cost	20%	4.5	18%
4	Voltage and Current Rating	15%	4.5	13.5%
5	Ease of Use	15%	4.5	13.5%
Total				92.5%

Table 2.4. DRV8323 Assessment

The DRV8323 performs its job almost perfectly and was a great fit for our project. It was simple to use and highly configurable. It was also a very cost effective part. There were some issues with learning how to use the part initially, but after consulting with our sponsor, the team had very few problems.

ID	Criteria	Weight	Score	Weighted Score
1	Complexity	25%	4	20%
2	High Torque Suitability	25%	5	25%
3	Cost	20%	4	16%
4	Voltage and Current Rating	15%	4.5	13.5%
5	Ease of Use	15%	4.5	13.5%
Total				88%

Table 2.5. Motor Assessment

The motor our team purchased was somewhat complex because it is only compatible with encoders that do not have an index pulse. Otherwise, it is a great motor that provides high torque for a good price. The voltage and current rating of the motor matches the specifications, given that the DRV8323RH can accommodate for up to 60V.

2.3.3 Decision Analysis

There were many decisions the team had to make in order to maximize system performance without increasing complexity to be able to meet project deadlines.

2.3.3a MCU

The AM263x MCU is a single-axis controller with fewer cores than the AM243x, making it less complex and more cost effective. Additionally, it is more commonly integrated on the same PCB as motor drive circuits, and it is sometimes even integrated into motor assembly itself. This leads to its target industry of industrial and automotive single-axis design, corresponding to our intended use case of high-torque robotic applications. Thus, the team chose the AM263x Sitara MCU for the project.

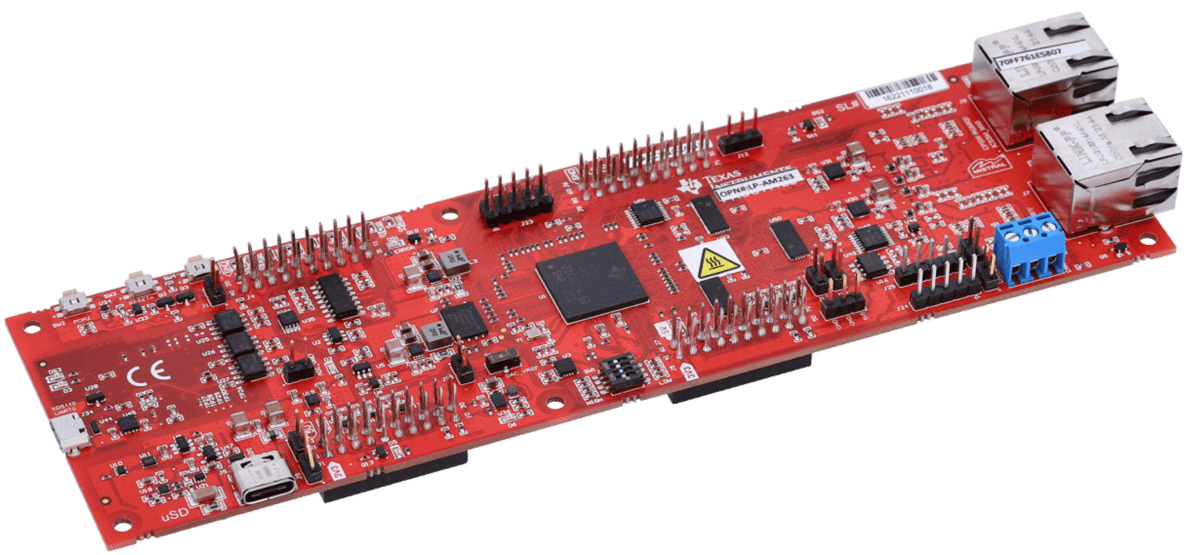


Figure 2.3. AM263x LaunchPad

2.3.3b Sensing

The commutation method chosen was the sinusoidal signal with a hall sensor and encoder. This method provides the best resolution and motor control capabilities without being the most complex option. The trapezoidal signal with hall sensors is the simplest method to use, but it would not yield the best results for our application; however, using a trapezoidal signal to initially test the system will simplify unexpected troubleshooting issues. Furthermore, using an encoder is best for high torque applications and motor control, though it is more difficult to use than the hall sensors alone. While FOC is best suited for high torque applications, the team decided that it would add an unnecessary level of complexity, and utilizing sensored control is more reliable.

2.3.3c Motor Driver

With the external gate driver topology, the driver and FETs can be selected independently to increase voltage ratings and decrease $R_{ds,on}$. This will increase design flexibility and scalability to achieve the system goal of being motor agnostic. The gate driver chosen was the DRV8323RH, which comes with current sensing amplifiers (CSA), charge pumps, and more. Additionally, this topology is better for power dissipation, since the driver and FETs are not on the same package, and it is best suited for the high voltage and current requirements of this project. The main disadvantage of using an integrated design like the DRV8316 is that the MOSFETs cannot be switched out, and the system would be limited to a certain voltage or current. This is good for low current applications and may be cost effective and simpler when designing a PCB compared to the external topology; however, it is not suitable given the project requirements and specifications. The DRV8353 would have worked equally well, but the system does not require the high power and voltage ratings it provides. Therefore, the DRV8323RH was deemed the most appropriate for the design.

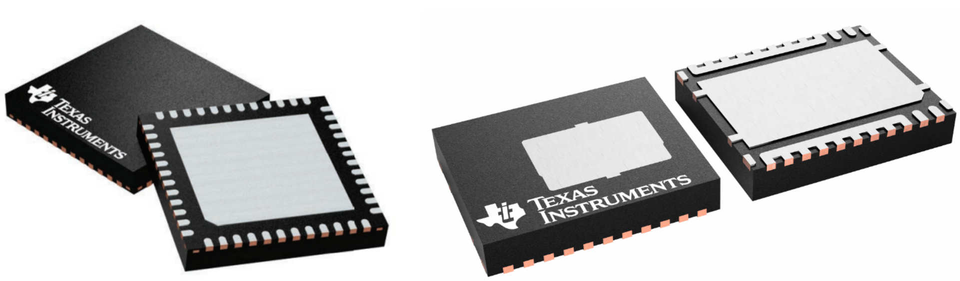


Figure 2.4. DRV8323RH (left) and CSD88584Q5DC (right)

3. Basic Solution Description

The objective of our project during the first semester was to build a fully integrated PCB with the Sitara AM2634 chip and the DRV8323RH chip.

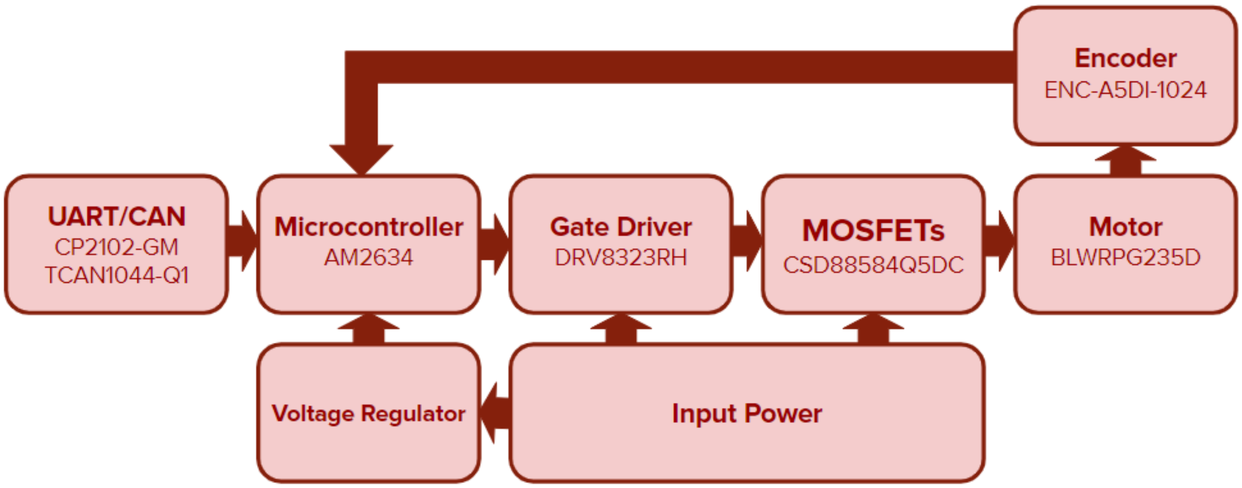


Figure 3.1. Process Flow Diagram

3.1 MCU

The Sitara AM263x is a high performance microcontroller that is relatively new to the industry. This MCU was made to handle the real-time processing needs in automotive and industrial applications and to be compatible with many different types of devices. The AM2634 will be powered with 5V from a voltage regulator, and it will handle PC communication through a USB to UART/CAN connector for receiving programmed instructions from the SDK. With the received instructions, it will send a PWM signal to the motor driver system to control the phases of the motor with the selected commutation method. Furthermore, it will receive current sensing feedback from the motor driver system to adjust the phases as needed and will be fed the encoder signals through its eQEP (Enhance Quadrature Encoder Pulse) pins for positional feedback.

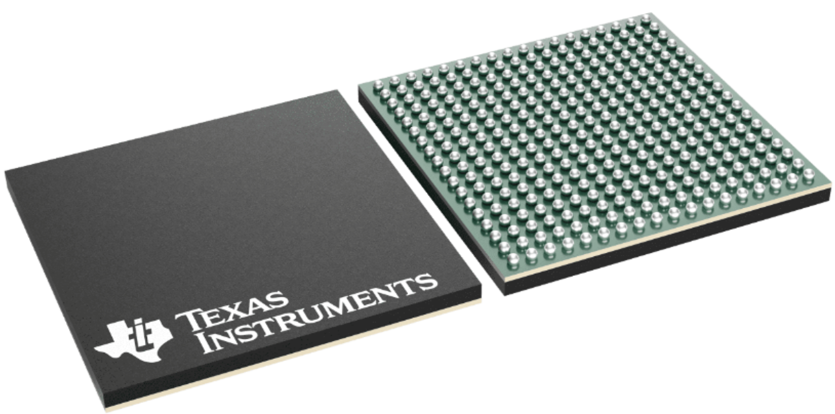


Figure 3.2. AM2634 chip

3.2 Motor Driver System

The DRV8323RH is an integrated motor driver created for 3 phase applications, such as spinning a BLDC motor with either sensored or sensorless control. The driver contains 3 half bridge gate drivers, each with a high side and low side CSD88584Q5DC MOSFET. The driver chip also has a charge pump, bidirectional current sensing capabilities, configurable 1x/3x/6x PWM modes, and an optional buck regulator. The driver can operate on a range of 6-60V from a single power supply, with an absolute max voltage rating of 65V and a 15A drive current. The BOOSTXL-DRV8323RH is the boosterpack containing this chip. It has 40 header pins that are compatible with the AM263x LP for initial testing (see table below). In addition, the boosterpack has header pins to power hall sensors in a motor.

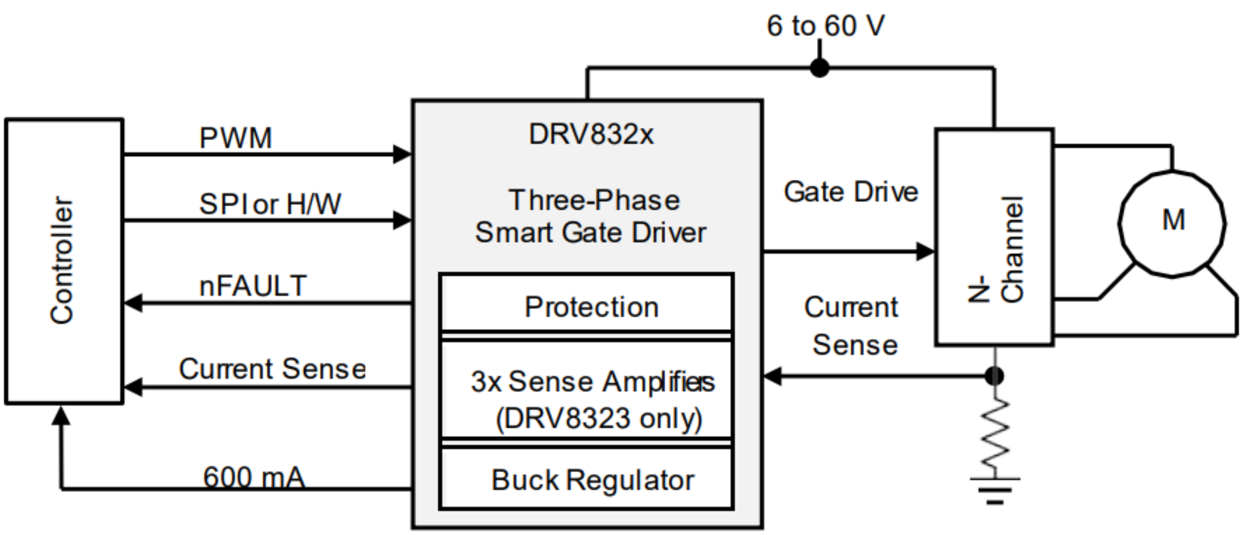


Figure 3.3. DRV832x Simplified Schematic

DRV8323RH Pin Number and Function		AM263x Pin Number and Function	
J3-1	3.3V	J5-41	3V3
J3-2	**No function**	J7-61	5V
J3-3	VSENVM	J5-42	ADC2_AIN3
J3-4	GND	J7-62	GND
J3-5	**No function**	J5-43	
J3-6	VSENA	J7-63	ADC2_AIN1
J3-7	**No function**	J5-44	
J3-8	VSENB	J7-64	ADC3_AIN1
J3-9	ENABLE	J5-45	GPIO74
J3-10	VSENC	J7-65	ADC4_AIN1
J3-11	POT	J5-46	ADC3_AIN3

J3-12	ISENC	J7-66	ADC0_AIN2
J3-13	SCLK	J5-47	SPI1_CLK
J3-14	ISENB	J7-67	ADC1_AIN2
J3-15	NFAULT	J5-48	GPIO93
J3-16	ISENA	J7-68	ADC2_AIN2
J3-17	**No function**	J5-49	
J3-18	IDRIVE	J7-69	GPIO129
J3-19	**No function**	J5-50	
J3-20	VDS	J7-70	GPIO128
J4-1	INHA	J8-80	EPWM13_B
J4-2	GND	J6-60	GND
J4-3	INLA	J8-79	EPWM13_A
J4-4	HALLA	J6-59	GPIO44
J4-5	INHB	J8-78	EPWM3_B
J4-6	HALLB	J6-58	GPIO15
J4-7	INLB	J8-77	EPWM3_A
J4-8	**No function**	J6-57	
J4-9	INHC (1x direction)	J8-76	GPIO62
J4-10	**No function**	J6-56	
J4-11	INLC (1x brake)	J8-75	GPIO 61
J4-12	SDI	J6-55	GPIO17
J4-13	MODE	J8-74	GPIO 124
J4-14	SDO	J6-54	GPIO18
J4-15	LED	J8-73	GPIO125
J4-16	HALLC	J6-53	GPIO67
J4-17	EVM ID	J8-72	GPIO119
J4-18	nSCS/GAIN	J6-52	GPIO66
J4-19	EVM ID	J8-71	GPIO120
J4-20	CAL	J6-51	GPIO65

Table 3.1. DRV8323RH to LP-AM263x Pin Mapping Functions

3.3 Motor

The BLWRPG235D-24V-4000-R47 is a 24V motor from Anaheim motors. It is a double shafted motor with a miniature differential encoder mounted on one shaft. The gearbox ratio is 47:1, and the rated torque and peak torque after considering the gearbox is 1390 and 4166 oz-in, respectively. The motor efficiency is estimated to be around 73%. The motor will receive the required voltages and currents from the motor driver system to function, while the attached incremental encoder will read the position and feed the information back to the MCU. The internal hall sensors will initialize the position for the encoder.

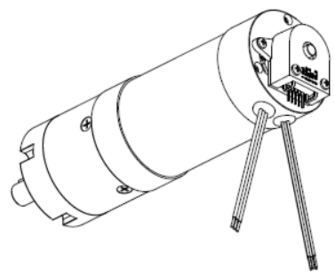


Figure 3.4. BLWRPG235D Anaheim Motor with Encoder

The figure below demonstrates a high-level diagram of the system design with the wires color-coded to their respective signal types.

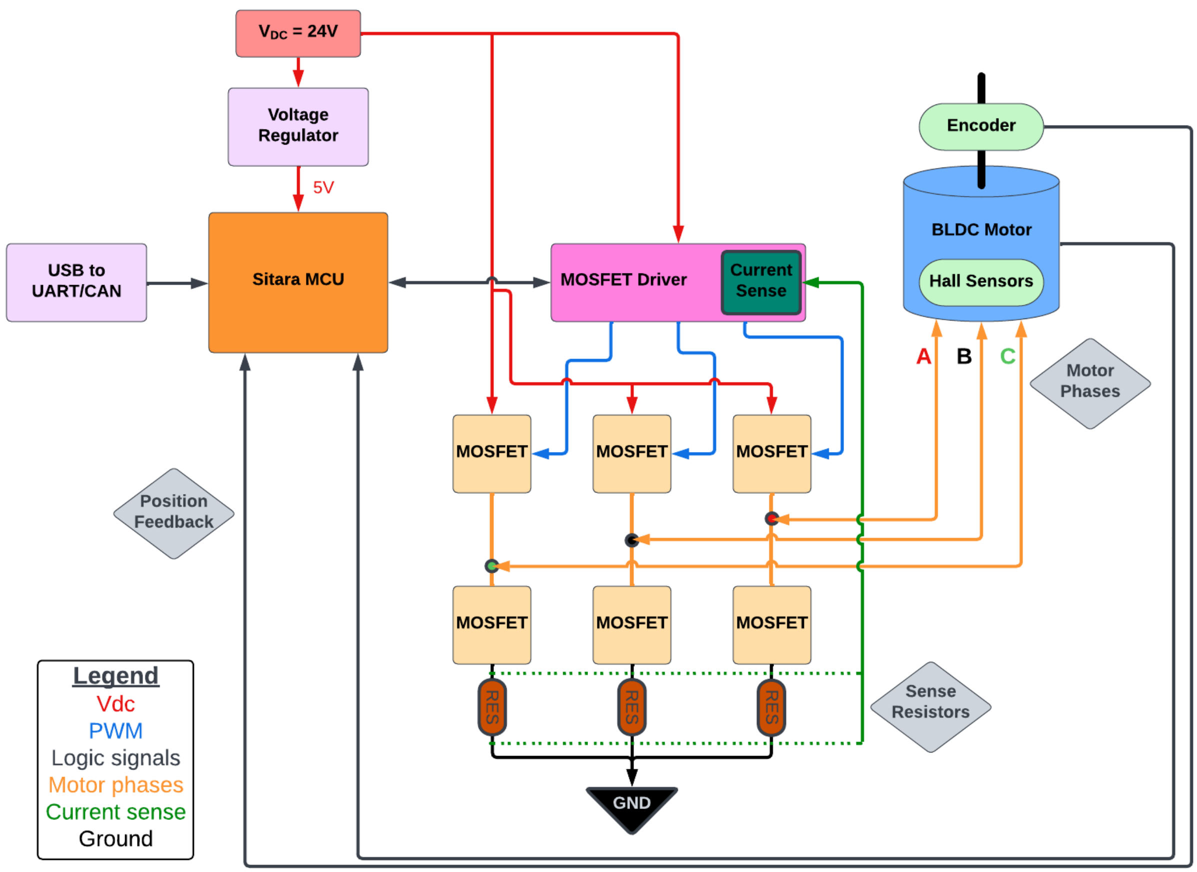


Figure 3.5. High-level Design Diagram

3.4 SDK

Figure 3.6 below shows the architecture diagram of the SDK components. The hardware of the TI Sitara MCU platform consists of the Arm R5F single- to quad-core ARM Cortex R5F cores for real-time compute, Arm M4F single-core for safety applications, device-specific SOC peripherals, and board-specific EVM peripherals. The OS Kernel supports FreeRTOS and NoRTOS options, and includes the Driver Porting Layer to abstract driver interfaces and the POSIX layer to abstract RTOS kernel functionality. The drivers expose the functionality of the SOC and board peripherals in a consistent manner across the MCU+ devices, while the protocol stacks and middleware provide additional functionality on top of the drivers. The SDK also comes with examples and demos to aid in writing applications, as well as tools to assist with development.

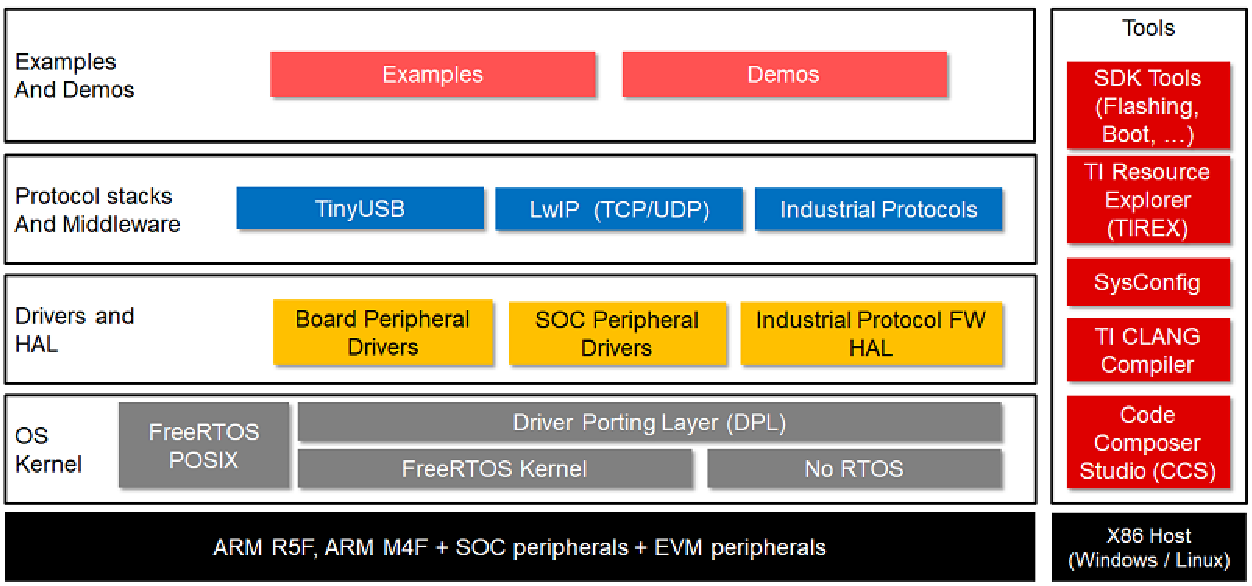


Figure 3.6. SDK Components

A typical MCU+ SDK project will consist of generated sources, main.c, projectName.c, example.syscfg, linker.cmd, and makefile_ccs_bootimage_gen files.

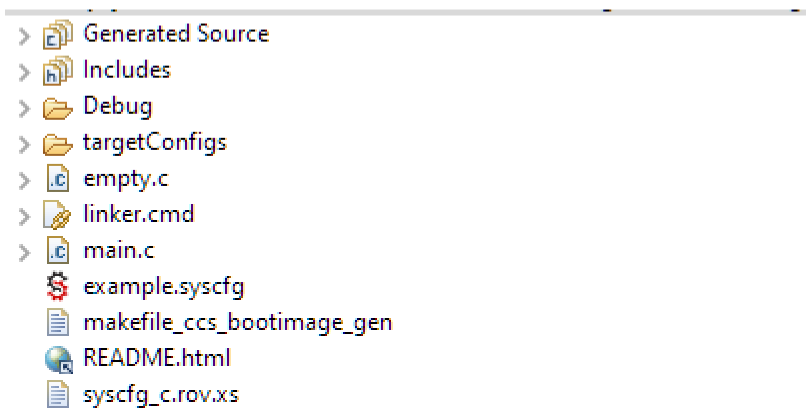


Figure 3.7. SDK Files for the Empty Project Example in the SDK

The Generated Source files provide the configurations specified by the SysConfig file and the device and driver related APIs to be called by the application. This includes board-related initialization, board-related drivers, initialization of the cache, memory protection unit, hardware interrupts, etc.

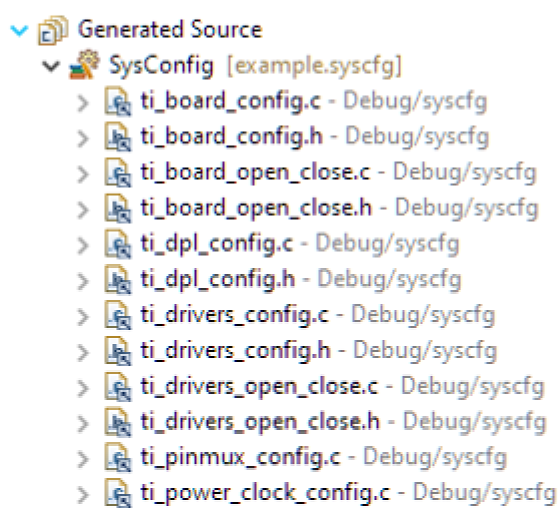


Figure 3.8. Generated Sources Example in the SDK

The main.c file contains the main() function that #includes the stdlib.h, ti_driversconfig.h, and "ti_board_config.h" header files. This function has calls to the application function and calls to device initialization and deinitialization.

The projectName.c file #includes the stdio.h, <kernel/dpl/DebugP.h>, ti_drivers_config.h, ti_drivers_open_close.h, and ti_board_open_close.h, header files. This source file contains the application function calls to open the drivers. It also has the projectName_main() function.

The example.syscfg file opens the SysConfig GUI which is used to configure the settings for drivers, middleware, protocol stacks, and more. An example screenshot of the GUI taken from the AM263x MCU+ Academy is shown in the figure below.

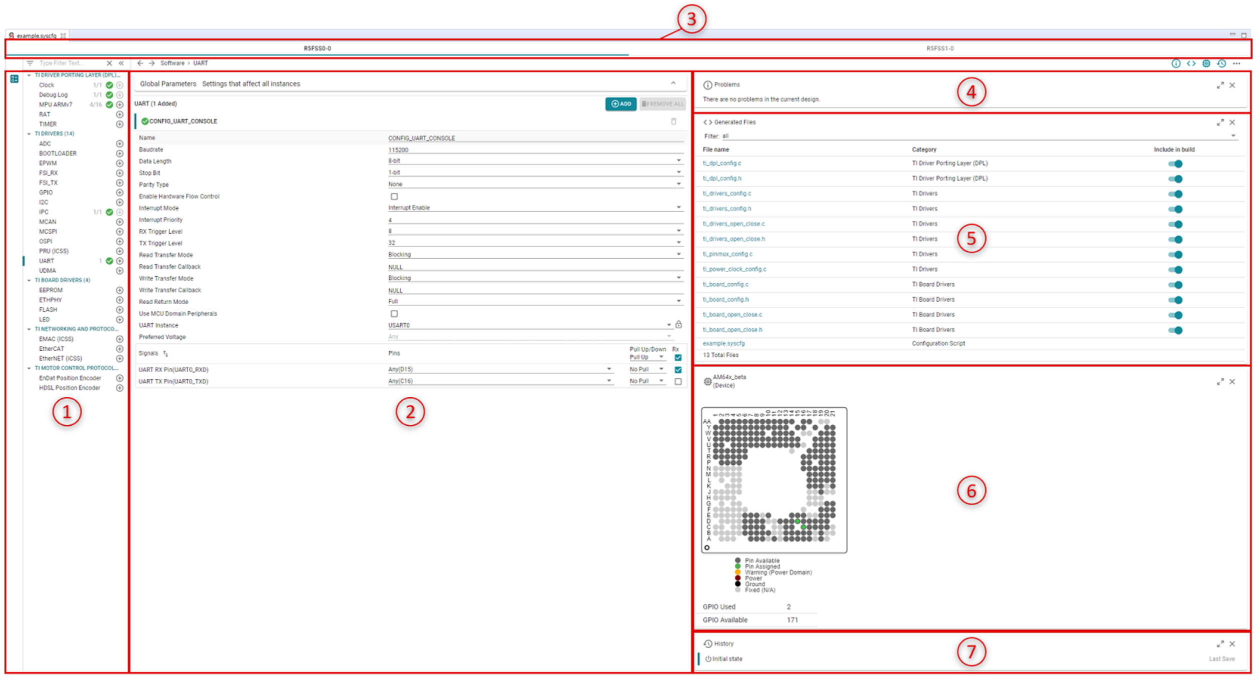


Figure 3.9. SysConfig GUI

1: Configurable Modules - Contains a list of software modules such as drivers, middleware, and protocol stacks that can be configured in the SysConfig GUI.
2: Module Configuration View - This view is where you can add an instance of a module and configure it.
3: Core Selection Tabs (multi-core projects only) - These tabs allow you to switch between SysConfig views for cores in a multi-core/system project.
4: Problems - This view highlights any problems detected by the SysConfig solver, such as pinmux conflicts.
5: Generated Files - This view contains the source code generated by SysConfig.
6: Device View - This view contains the device pin layout and displays the number of GPIO pins used in the design.
7: History - This view logs all of the changes done in the SysConfig GUI.

The program files used in the creation of this project are based on the example program files from the SOC and Board Peripheral Drivers Examples sections of the SDK. Each of these follow the basic outline of SDK projects as described above.

4. Performance Optimization and Design of System Components

4.1 Design Criteria

4.1.1 MCU - AM2634

The only alternative considered to the AM263x is the AM243x. Both are part of the Sitara MCU family, but the AM243x uses intermediary current feedback with shunt resistors, while the AM263x uses direct current feedback. Direct current feedback is less complex while still allowing the MCU to maintain a high performance. Furthermore, the AM263x has fewer cores contributing to the simplicity of the chip when compared to the AM243x. To test the system with the DRV boosterpack, the LaunchPad header pins must be compatible with the boosterpack header pin functions. The process for this will be further detailed in the next section.

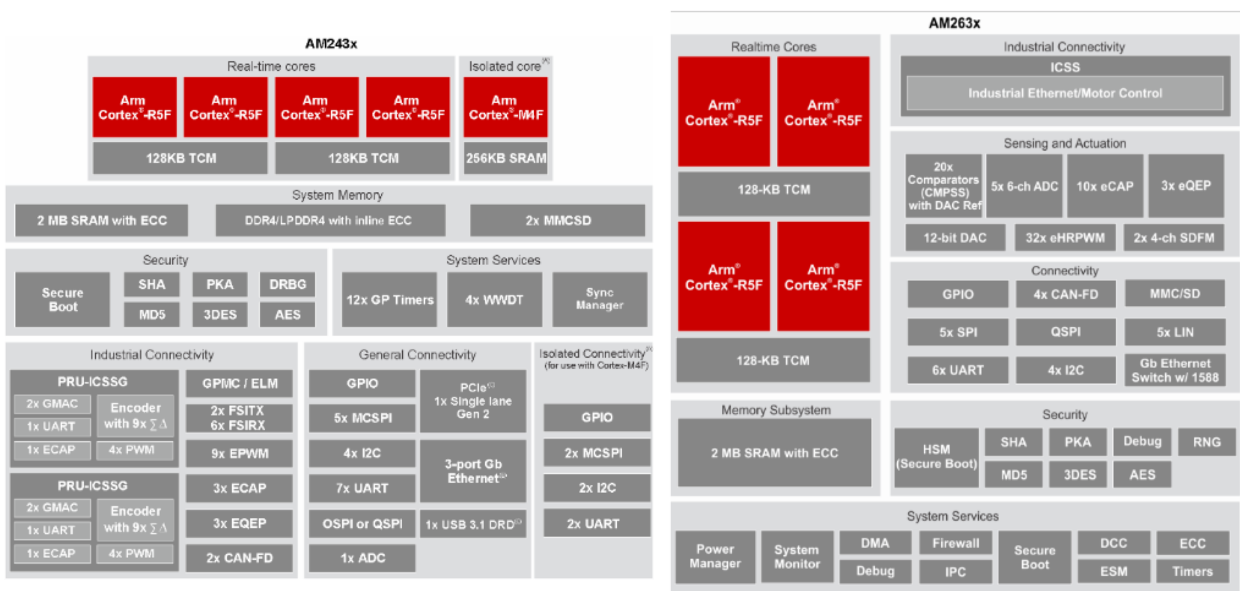


Figure 4.1 AM243x Peripherals (left), AM263x Peripherals (right)

4.1.2 Motor Driver - DRV8323RH + External FETs

The motor driver that our team chose is the DRV8323RH variant. As discussed in Sections 2.3.1c and 2.3.3c, there were a few other driver options. We chose the DRV8323 over the DRV8353 because the latter had higher voltage capabilities that were not needed. With a 24V motor, we expect to account for transients up to 2x that amount, or 48V spikes. Our selected motor driver supports up to 65V, comfortably accommodating any voltage spikes. The DRV8353 supports up to 95V which we would only need to use if the motor operated at approximately

36-48V. Our other option was the DRV8316, which is best suited for low voltage and low current applications and cannot support our motor.

The DRV8323 also has a SPI variant (S), but we chose to use the hardware variant (RH) for easier troubleshooting.

Driver (DRV8323R)	**MIN**	**MAX**
Power supply voltage (V_{VM})	6V	60V
Input voltage	0V	5.5V
High- & low-side average gate drive current	0mA	25mA
External load current	0mA	30mA
Reference voltage input (V_{REF})	3V	5.5V

Table 4.1. DRV8323R Motor Driver Performance Requirements

MOSFETs (CSD88584Q5DC)	**MIN**	**MAX**
Gate drive voltage (Vgs)	4.5V	16V
Switching frequency	5kHz	50kHz
Output current	-	50A
Voltage supply	-	36V

Table 4.2. CSD88584Q5DC MOSFET Performance Requirements

As mentioned briefly in section 3.2, the DRV832x EVM can be configured in either 1x, 3x, or 6x PWM modes to optimize performance and allow for more configurable control with more PWM signals. The figure below is a screenshot from the DRV8323x gate driver datasheet for implementing synchronous 1x PWM mode for a three-phase BLDC motor to be controlled using one PWM sourced from the MCU. The INLA, INHB, and INLB pins are used as state logic inputs that manage the half-bridge output states. The INHC input controls the direction of the motor through the 6-step trapezoidal commutation table. The INLC input functions as the motor brake by turning off the high-side MOSFETs and turning on the low-side MOSFETs.

LOGIC AND HALL INPUTS							GATE DRIVE OUTPUTS[1]						
STATE	INHC = 0			INHC = 1			PHASE A		PHASE B		PHASE C		DESCRIPTION
	INLA	INHB	INLB	INLA	INHB	INLB	GHA	GLA	GHB	GLB	GHC	GLC	
Stop	0	0	0	0	0	0	L	L	L	L	L	L	Stop
Align	1	1	1	1	1	1	PWM	!PWM	L	H	L	H	Align
1	1	1	0	0	0	1	L	L	PWM	!PWM	L	H	B → C
2	1	0	0	0	1	1	PWM	!PWM	L	L	L	H	A → C
3	1	0	1	0	1	0	PWM	!PWM	L	H	L	L	A → B
4	0	0	1	1	1	0	L	L	L	H	PWM	!PWM	C → B
5	0	1	1	1	0	0	L	H	L	L	PWM	!PWM	C → A
6	0	1	0	1	0	1	L	H	PWM	!PWM	L	L	B → A

Figure 4.2. Synchronous 1x PWM Mode

In the 3x PWM mode truth table shown below, each half-bridge supports low/high output states and is controlled by the INHx pin. The INLx pin is used to put the half-bridge in the Hi-Z state, while the corresponding INHx and INLx signals control the output states listed in the figure.

INLx	INHx	GLx	GHx	SHx
0	X	L	L	Hi-Z
1	0	H	L	L
1	1	L	H	H

Figure 4.3. 3x PWM Mode Truth Table

To best optimize system performance for motor configurability, 1x PWM mode will be used for initial testing and troubleshooting, while 3x PWM mode will be implemented in the final design.

4.1.3 Motor - BLWRPG235D-24V-4000-R47

The BLWRPG235D fit our max 24V requirement and had a higher torque when compared to motors outside the Anaheim Automation website. Anaheim itself has several other motors, including those with planetary or spur motor gearboxes. Depending on the gearbox ratio, the torque will differ. In addition, the team wanted a motor with either an internal encoder or an attached encoder. We chose this motor because it had all the required specifications, was priced reasonably, and was compatible with several different Anaheim encoders.

An alternative motor is the BLWRPG173D-24V-4000-R116 from Anaheim. This 24V brushless DC motor also fits the required specifications, but the torque and efficiency are a little lower than our motor of choice. The peak torque is 2083 oz-in, which would still be suitable for a high torque application, and we estimated the efficiency to be approximately 70%. Due to the lower torque, this motor is also somewhat cheaper than the BLWRPG235D, and there are also different encoders that are compatible.

Parameter	Target	MAX	Unit
Voltage	24	-	V
Current	11	16	A
Motor Efficiency	73	-	%
Output Torque	2928	4166	oz-in
	20.7	29.4	N-m
	15.3	21.7	lb-ft
Output Power	184	-	W

Table 4.3. Motor + Encoder Performance Requirements

The table below shows a comprehensive range of target parameters to optimize performance of the system.

	Target	MAX	Unit
Voltage	24	40	V
Current	11	50	A
Temperature	60	100	℃

Table 4.4. Comprehensive Range of Performance Requirements

4.2 Components

4.2.1 Microcontroller

The AM2634 MCU is itself a component in the system, but requires a few other components to aid in using it properly. The components added in the tables below are for supporting voltage regulation for bucking the input power down to 5V, PC communication, and LED indication for faults.

Component	Quantity
TPS51396A Buck Converter	1
0.68µH Inductor	1

5.1kΩ Resistor	1
20kΩ Resistor	1
90kΩ Resistor	1
100kΩ Resistor	1
330kΩ Resistor	1

Table 4.5. Voltage Regulation Components for the MCU

Component	**Quantity**
TCAN1044DRQ1 CAN Chip	PC Communication
1x2 Female Connector	1

Table 4.6. PC Communication Components for the MCU

Component	**Quantity**
Red LED	1
Green LED	1
Yellow LED	2
1kΩ Resistor	4

Table 4.7. LED Indication Components for the MCU

4.2.2 Motor Driver System

The DRV8323RH motor driver in combination with the three CSD88584Q5DC MOSFETs require many passive components to configure for the necessary current and voltage specifications.

Component	**Quantity**
0.01µF Capacitor	1
0.1µF Capacitor	4
1µF Capacitor	2
0.047µF Capacitor	1

4.7µF Capacitor	1
2.2µF Capacitor	1
22µF Capacitor	1
330µF Capacitor	1
1000pF Capacitor	2
2200pF Capacitor	1
Schottky Diode	1
0Ω Resisor	1
1kΩ Resistor	1
56Ω Resistor	1
330Ω Resistor	2
1.5kΩ Resistor	1
4.99kΩ Resistor	1
8.45kΩ Resistor	1
10kΩ Resistor	1
28kΩ Resistor	1
35.7kΩ Resistor	1
47kΩ Resistor	1
75kΩ Resistor	1
83kΩ Resistor	1

Table 4.8. Passive Components for the Motor Driver System

4.2.3 Motor

The BLWRPG235D comes with internal hall sensors and is compatible with several different attachable encoders. Anaheim Automation recommended using the 1024DI encoder. This is an incremental encoder with an index pulse that is compatible with the BLWRPG23 motor series. This particular motor utilizes a delta winding, where the beginning of each phase is tied to the end lead of the next phase. This kind of winding is good for high voltage applications.

The BLWRPG17 series uses a wye/star winding, where either all the ends or all the beginnings of the phase leads are tied at one neutral, central point. The wye connection is typically used for high current applications. It also comes with internal hall sensors, but it is only compatible with a series of encoders that does not have an index pulse. An index pulse is a feature where the encoder sends a pulse when the motor completes a full rotation, convenient for motor control and knowing the motor's position. The appropriate encoder part for this motor is the 1000DN8.

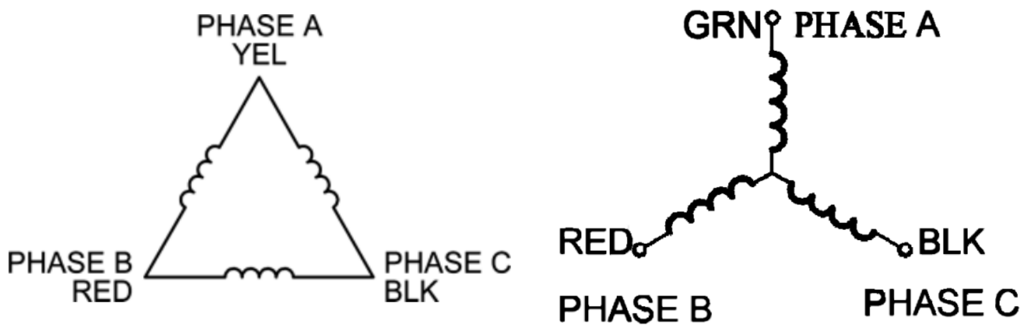


Figure 4.4. Delta Winding (left) vs Wye Winding (right)

4.3 Fabrication/Assembly Process

As stated in section 2.1.3, our technical approach consists of two phases to satisfy project requirements while staying within the time constraints of a single semester in the case of integrated PCB design infeasibility.

4.3.1 Assembly of Initial Design

Phase one of the project is using existing development boards from TI that coincide with the selected components meeting project specifications described in the previous sections. The J5, J6, J7, and J8 connectors on the LP-AM263x can be configured within the .sysconfig file in the SDK so that the BOOSTXL-DRV8323RH can mount directly to the LP, eliminating the complexity of extraneous wires; this is demonstrated in the figure below. The HALLA, HALLB, and HALLC pins on the DRV8323 boosterpack will connect to the corresponding wires coming from the motor.

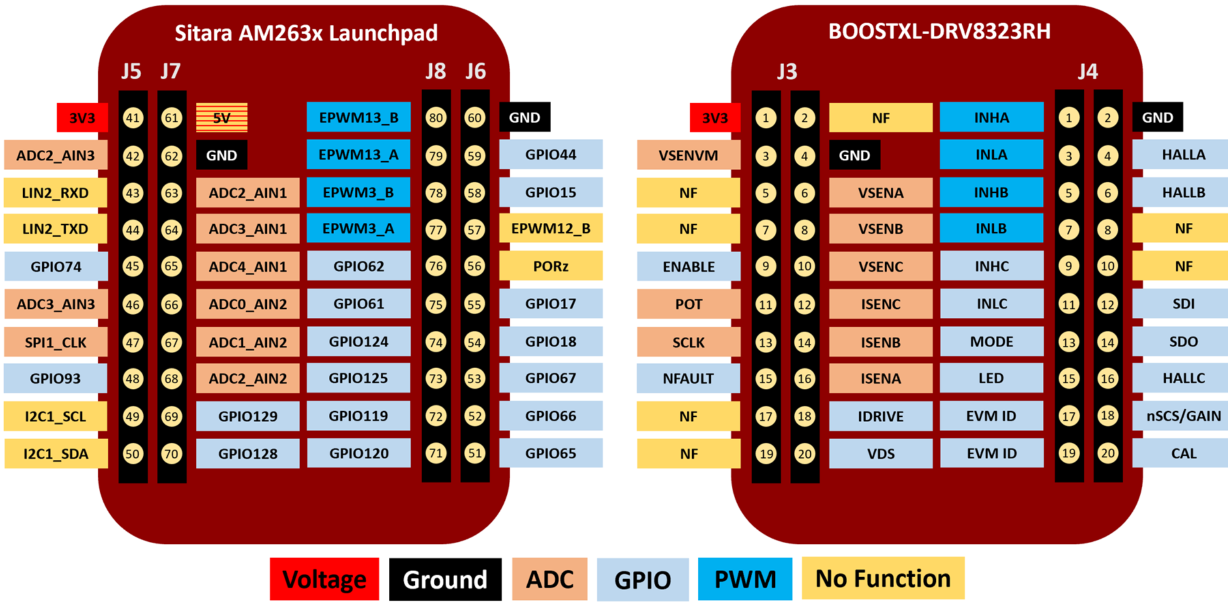


Figure 4.5. LaunchPad to DRV8323RH Boosterpack Connections

The motor will be held by the 3D printed motor mount seen in the image below with the following specifications: 60mm x 60mm face, 146mm length, and 30mm height.

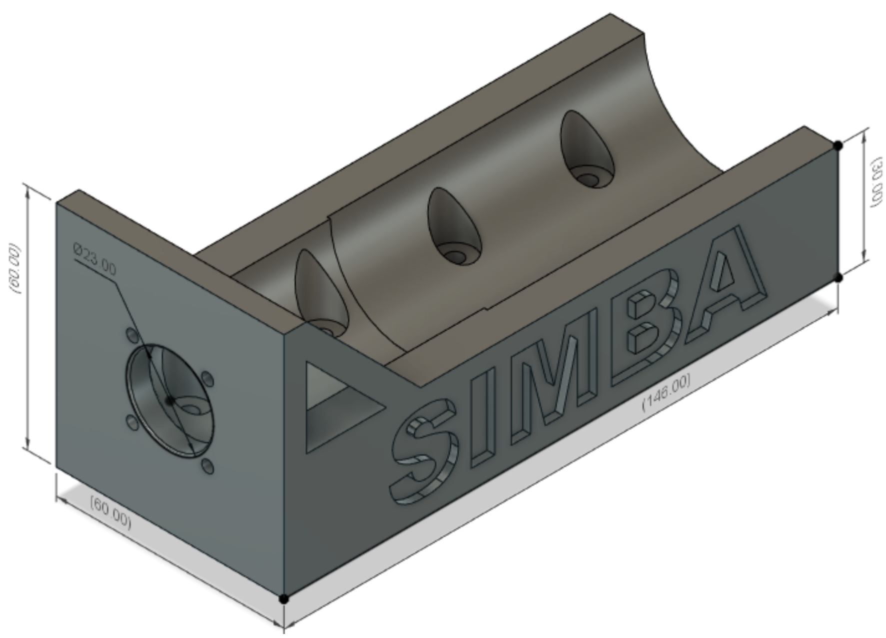


Figure 4.6. Motor Mount Design (Units in mm)

The motor shaft to extrusion arm connector in the image below shows the measurements of the physical component.

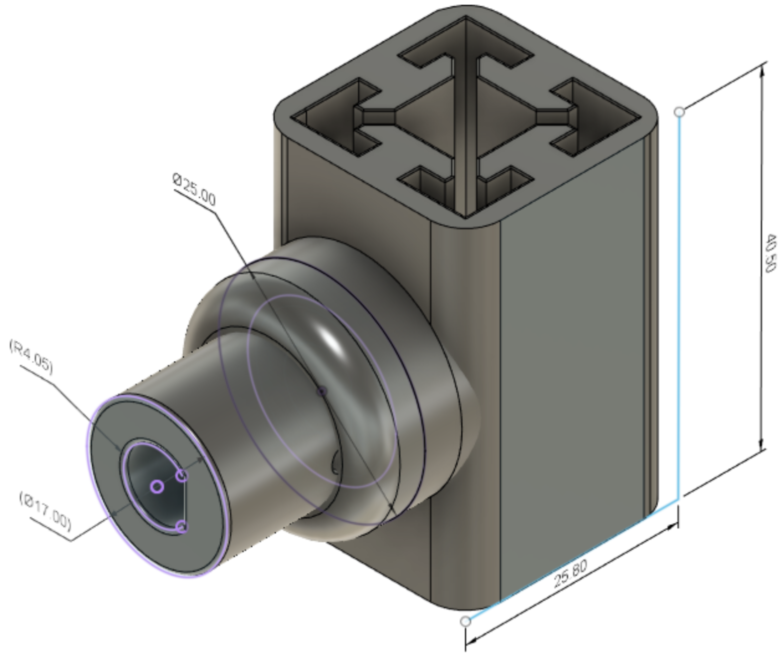


Figure 4.7. Motor Shaft to Aluminum Extrusion Arm Connector (Units in mm)

4.3.2 Fabrication of Integrated PCB Design

Phase two of the project utilizes KiCad to create the schematic designs below to develop a PCB that integrates the AM2634 MCU with the DRV8323RH and CSD88584Q5DC MOSFETs. In regards to the MOSFETs, special care must be taken with the PCB layout design and placement of the input capacitors; high-current, high dI/dT switching path; current shunt resistors; and GND return planes. Additionally, further PCB considerations for the MCU are described in the AM263x Hardware Design Guide.

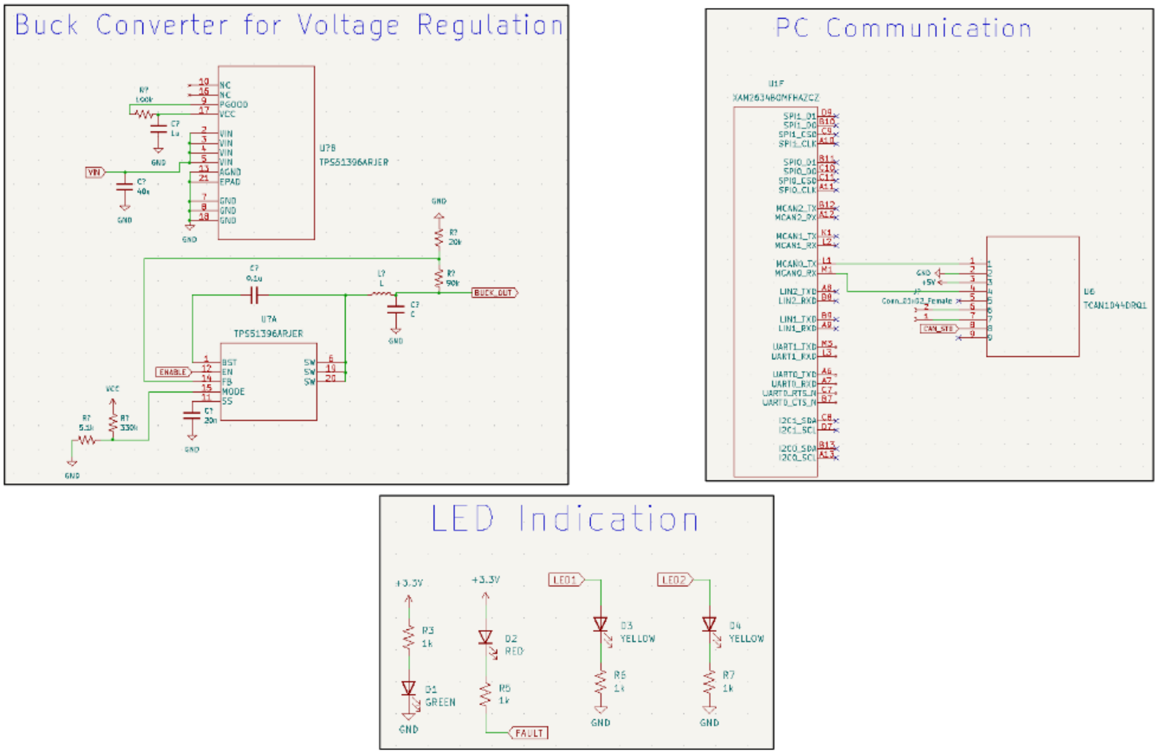


Figure 4.8. Voltage Regulation (top left), PC Communication (top right), and LED Indication (bottom) for the MCU

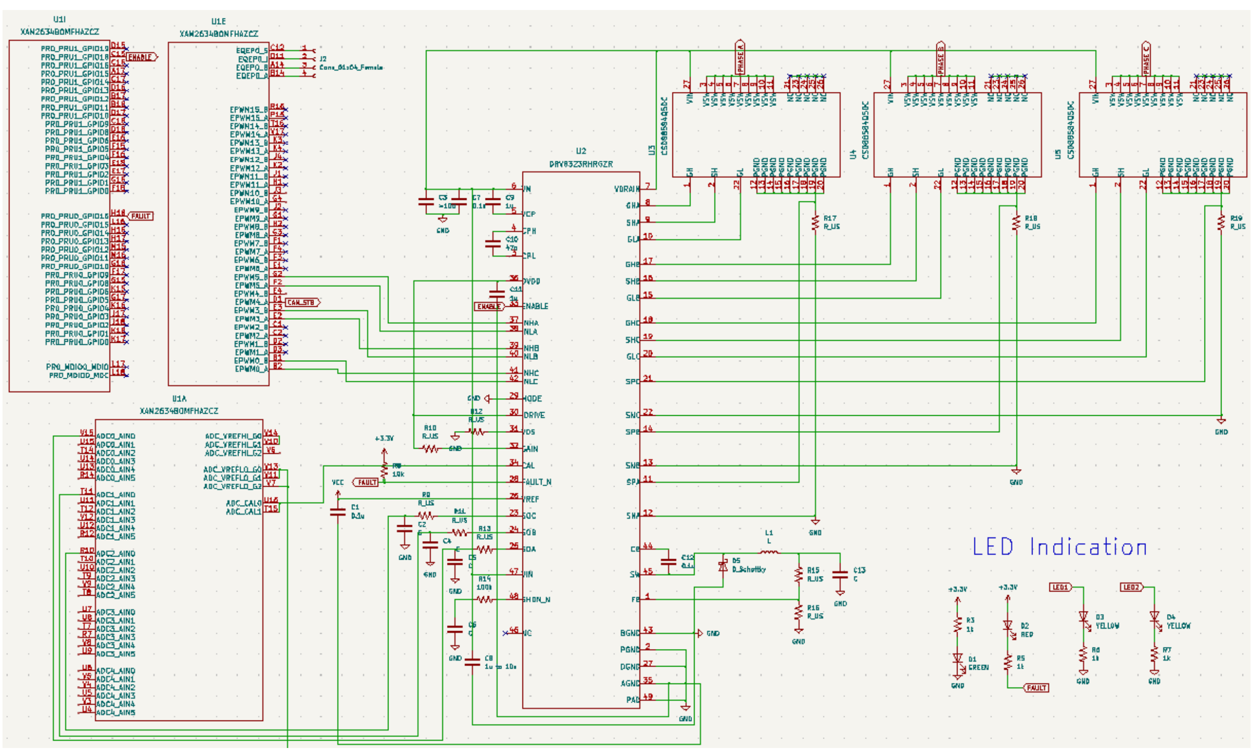


Figure 4.9. DRV8323RH + CSD88584Q5DC MOSFETs Connections to MCU

Due to the high complexity of an integrated PCB, the team was unable to properly generate the PCB design in KiCad and opted to move forward with the project implementation described in the next section.

5. Project Implementation, Operation, and Assessment

5.1 Overview of Final Implementation

5.1.1 Hardware

Due to several issues including time constraints, complexity, and component availability, our team made the decision to configure the AM263x LP to work with the BOOSTXL-DRV8323Rx EVM in order to provide highly configurable motor control for high-torque robotic applications. This will utilize the setup described in section 4.3.1 to directly mount the boards together instead of creating an integrated PCB with individual components. Furthermore, due to the complexity of configuring multiple PWM signals from the MCU, the project will continue to use the 1x PWM mode of the DRV that implements trapezoidal commutation described in sections 2.3.1b and 4.1.2 to drive the motor rather than the desired 3x PWM mode meant for sinusoidal commutation. Further description of the software implementation is described in sections 5.1.2 and 5.2.

The team originally chose a motor from the BLWRPG235 series, but the motor went out of stock. Thus, the final project is implemented with the BLWRRPG173D–24V-4000-R116, a double shafted motor with a miniature differential encoder (without index) mounted on top. The gearbox ratio is 116 with similar specifications to the older motor, but the new motor has a lower peak torque and is cheaper. Sections 4.1.3 and 4.2.3 provide more details on the two motors.

The figure below is a high-level diagram of the final project implementation, with the DRV boosterpack mounted directly onto the Sitara LP.

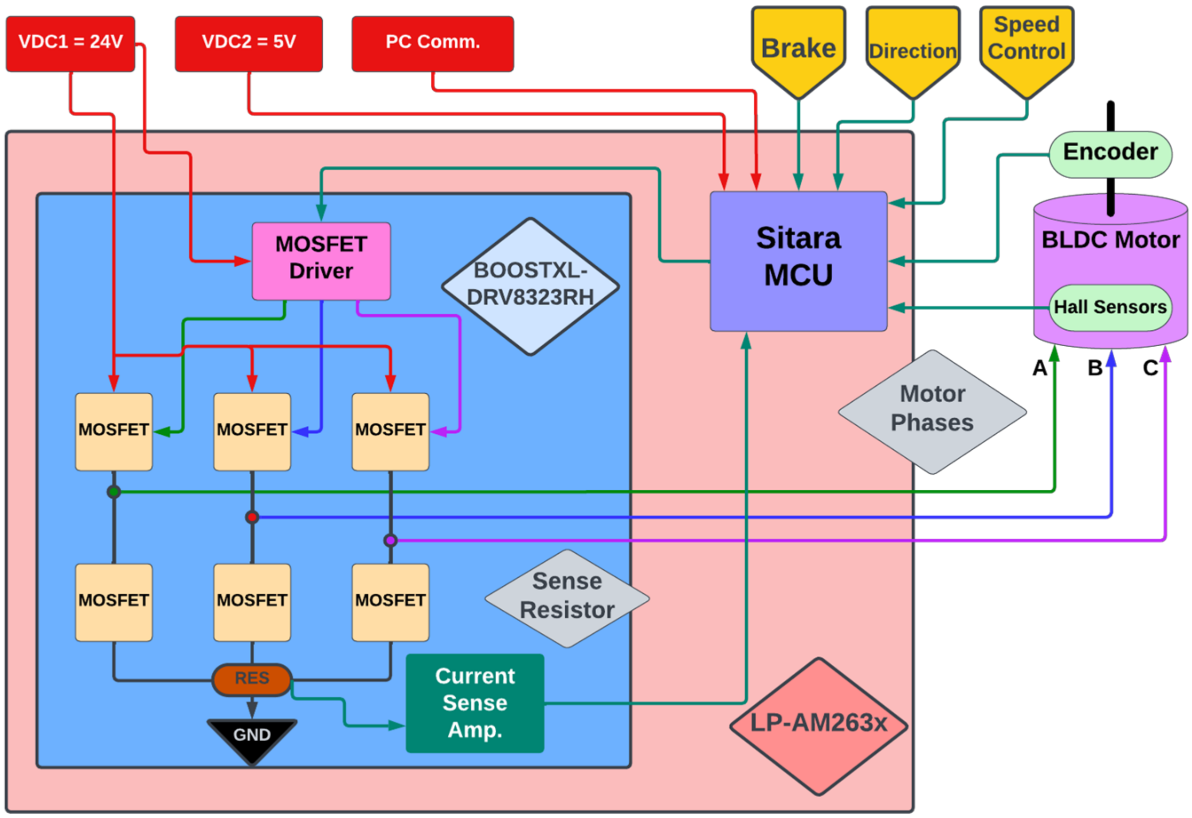


Figure 5.1. Final Boosterpack Implementation High-Level Design

Figure 5.2 shows the physical boards mounted on top of each other, with the top image being the view from above and the bottom image as the side view. Figure 5.3 is the mounted boards inside of the final build enclosure. The green, red, and black wires coming from the motor are phase A, B, and C, respectively. The blue, orange, and brown wires are connected to hall sensors A, B, and C, respectively, while yellow is the voltage supply to the hall sensors and white is ground. The wires coming from the encoder to the boards are for 5V, ground, A, A', B, and B', represented by red, black, blue, orange, yellow, and white, respectively.

Figure 5.4 and 5.5 show the mounted board and the final build to present the demos.

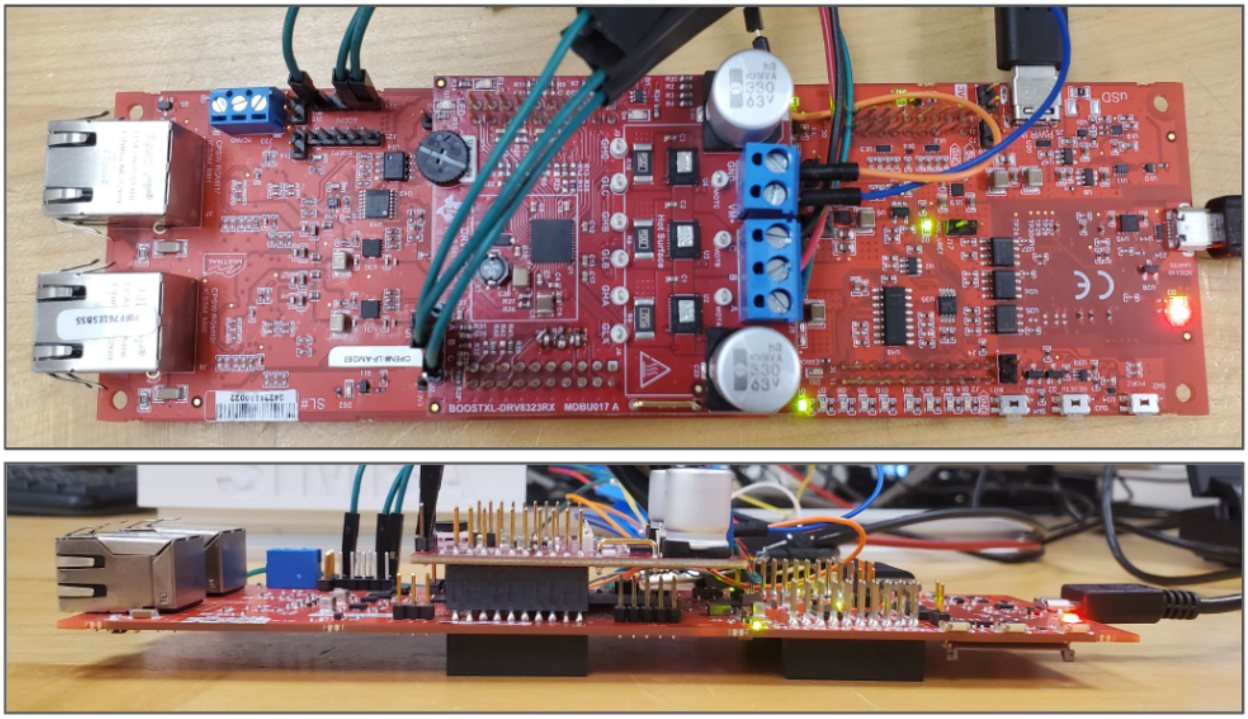


Figure 5.2. BOOSTXL-DRV8323Rx Direct Mount to LP-AM263x

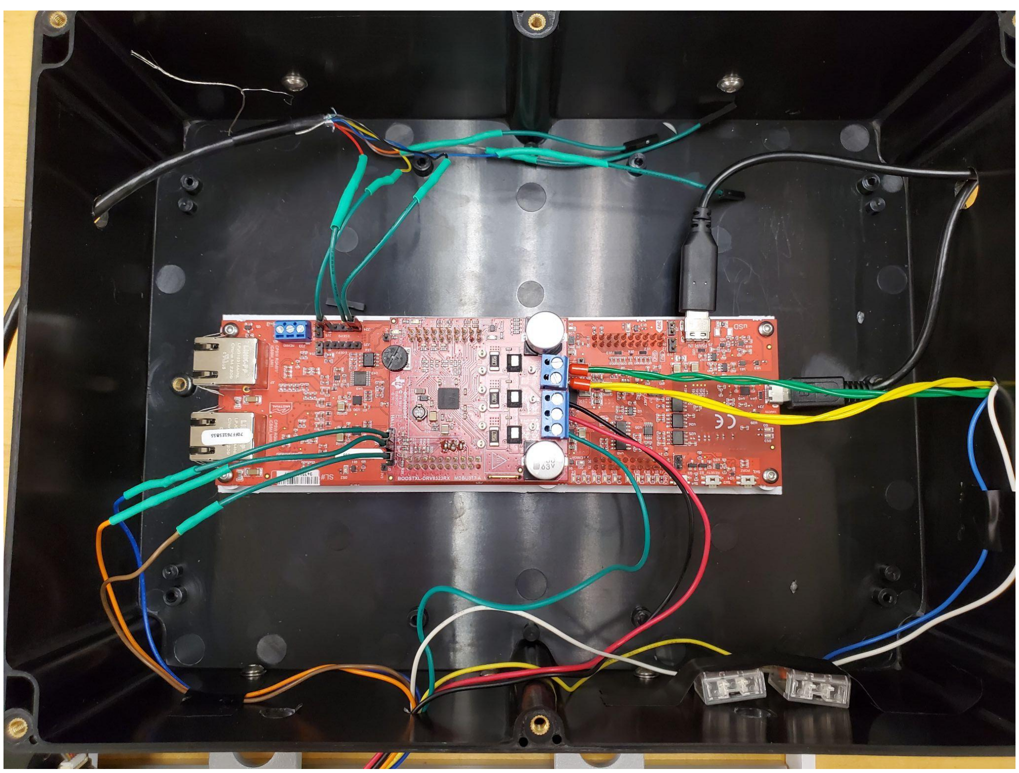


Figure 5.3. Motor + Encoder Connections to DRV

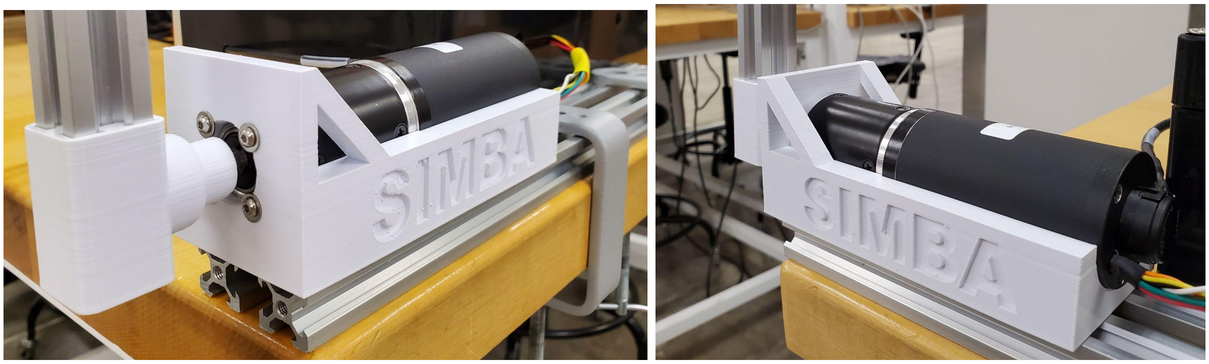


Figure 5.4. Motor Mount

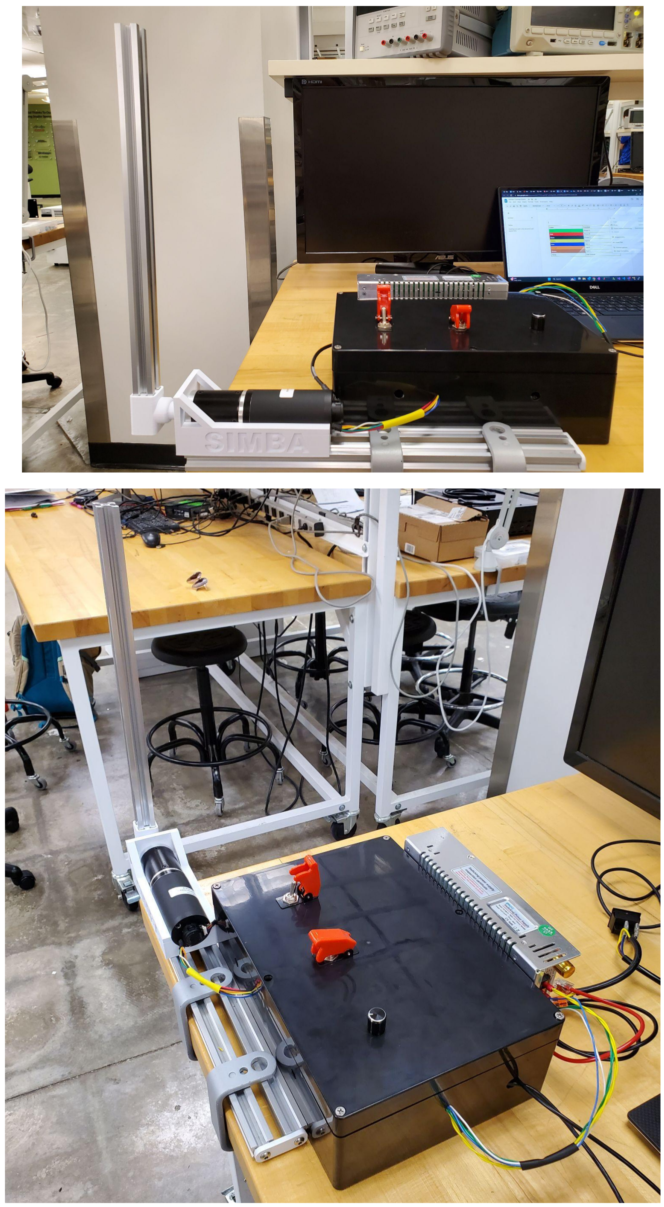


Figure 5.5. Project Case Enclosure Setup

5.1.2 Software

As described in Section 5.2.2a and 5.2.2b, the GPIO pins were configured first, and then the team used the ADC and PWM examples in conjunction to better control the motor using the potentiometer on the DRV8323RH boosterpack. The potentiometer is connected to the POT header pin on the board. When the boards are mounted, the POT pin connects directly to another header below on the AM263x LP that reads in ADC values. The pseudocode below shows the calculations done to convert the potentiometer reading into a duty cycle. Once the duty cycle is calculated, the code then creates a compare value. The compare value is a time based value that tells the PWM signal when to go high. For example, if the duty cycle is 50%, and the time of a period is 25000 clock pulses, then the compare value will be 12500. At pulse 12500, the PWM signal will go high, and will feed into the INHA pin, which is constantly reading in values and updating the phases accordingly. It signals to the board which phases are on and which phases are off. In 1x PWM mode, the PWM is fed into one phase at a time, and the DRV8323RH does the calculations to set the other 2 phases. Figures 5.7 through 5.9 show oscilloscope screenshots of the measured duty cycle.

```
count = 1000;
Init PWM
Set up GPIO Pins
while(count > 0) {
      Read POT ADC in // (in range [0, 4095])
      float duty_cycle = ADC in / 4095 // (in range [0, 1])
      compare_val = (1 - duty_cycle) * time_based_period
      Decrement count
}
```

The software flow chart for the pseudocode above is demonstrated in Figure 5.6 below.

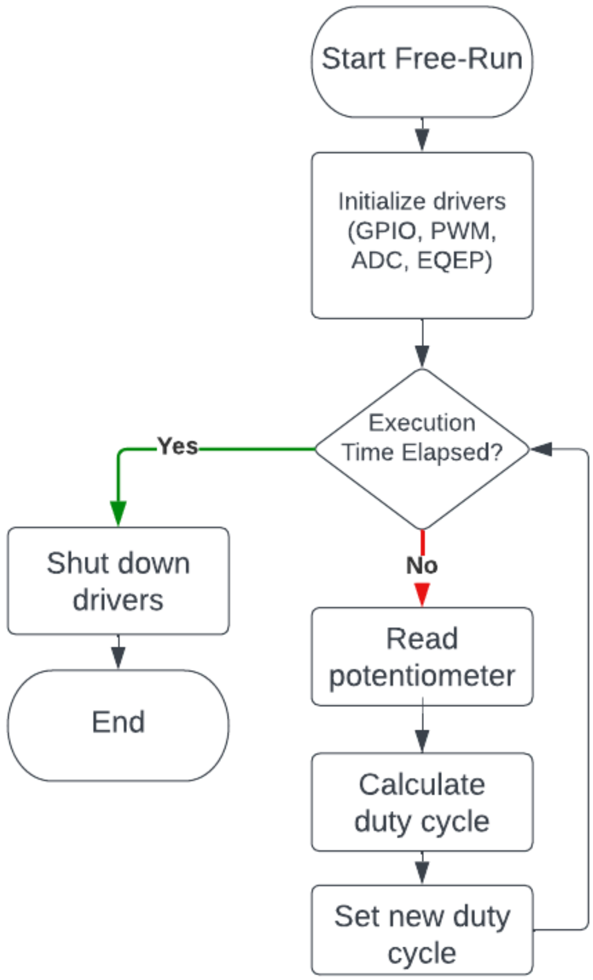


Figure 5.6. Free-Run Demo Software Flow

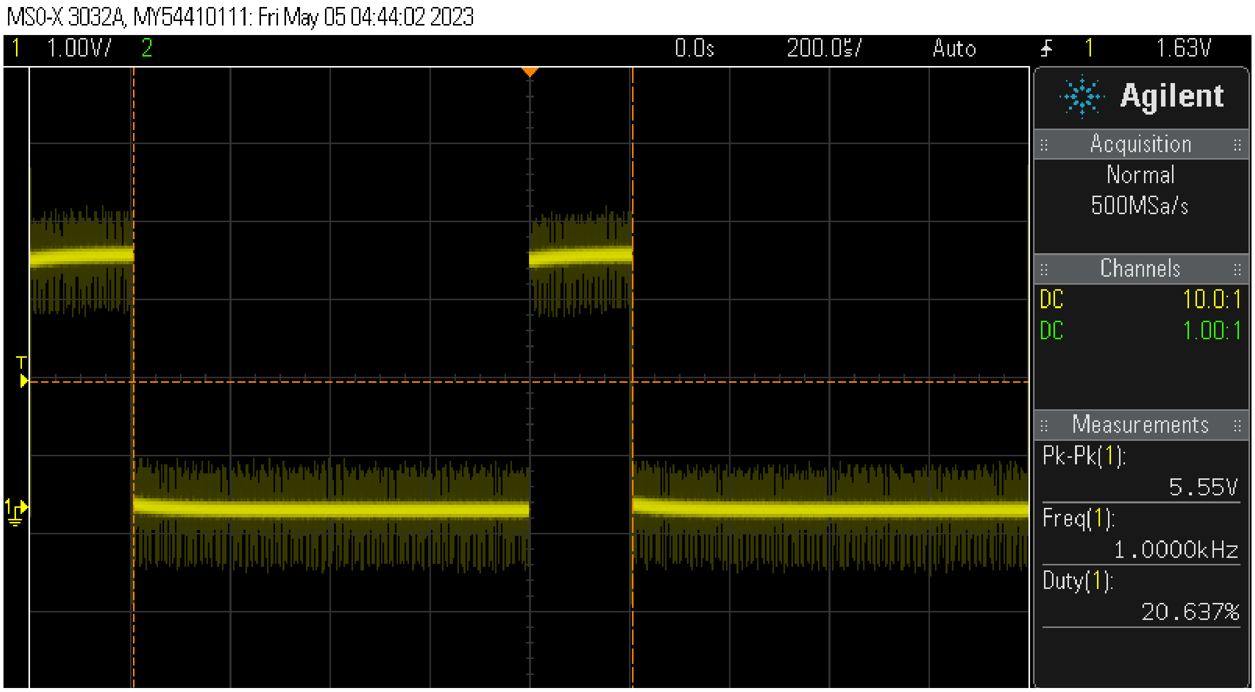


Figure 5.7. ADC Adjusting PWM 20% Duty Cycle

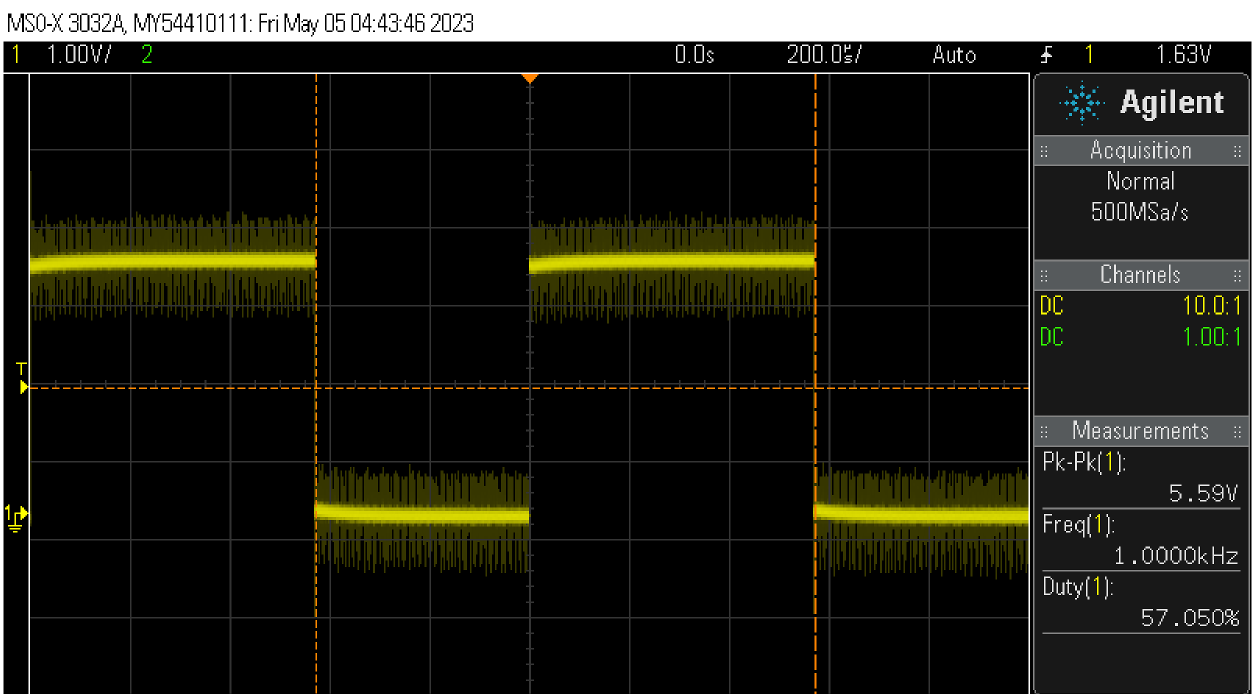


Figure 5.8. ADC Adjusting PWM 57% Duty Cycle

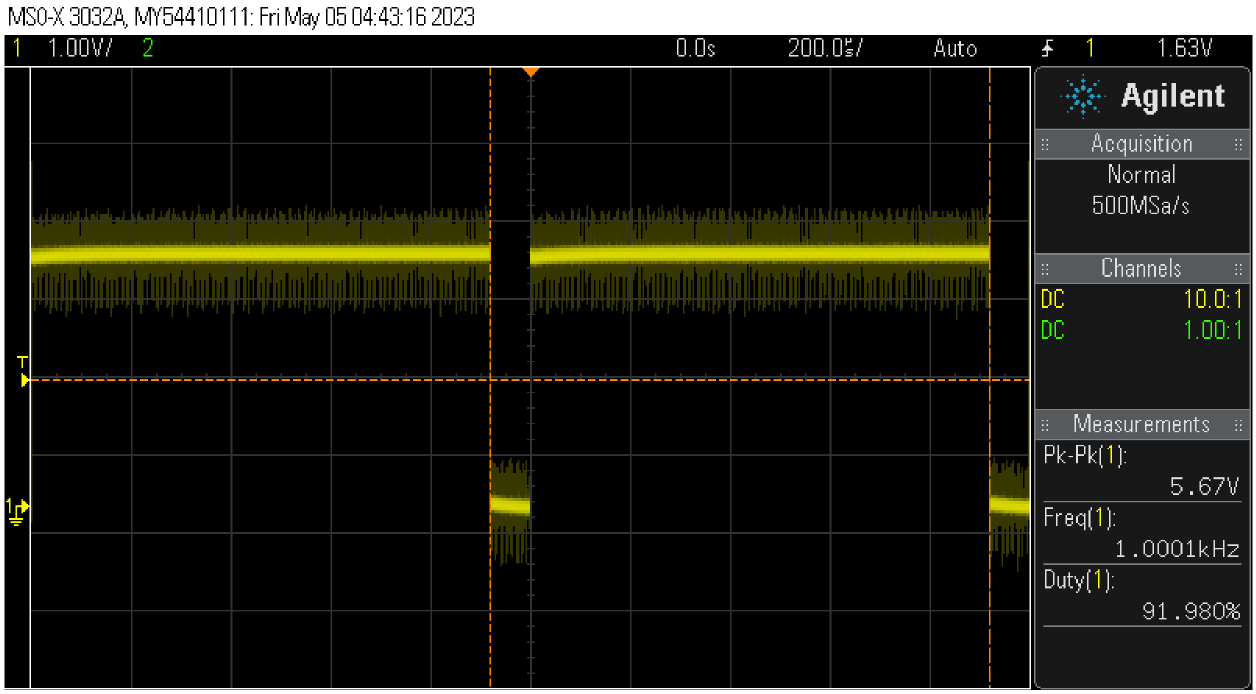


Figure 5.9. ADC Adjusting PWM 92% Duty Cycle

The encoder senses the position of the motor and sends signals to the eQEP peripheral, which counts the periods of the incoming encoder signals, on the MCU to perform positional and speed control. The encoder signals seen in Figure 5.10 are two square waves in quadrature (90° phase shifted). The direction of motion is indicated by the sign of the A-B phase difference which, in this case, is positive because B trails A. The software flow for positional control is in the Appendix section. The pseudocode is summarized below. Valid input is considered a positive or negative angle measure. After the motor reaches the desired position, there may be a small error. We calculate and report this error in the terminal.

```
Init PWM
Set up GPIO Pins
Get user input
while(input is a number) {
      Get direction from sign of angle
      Calculate number of encoder pulses from angle measure
      Set direction
      Set PWM duty cycle = 1
      while(encoder position < target position)
            Wait
      Report final encoder position
      Report error
Get user input
}
```

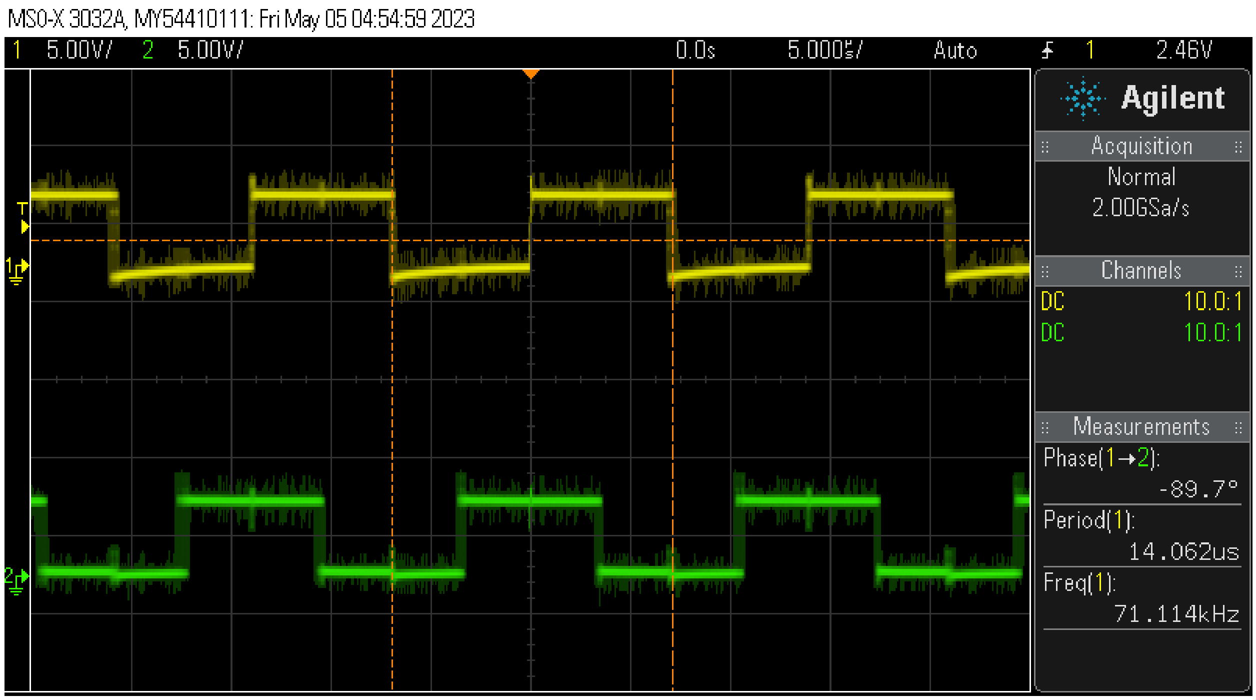


Figure 5.10. Encoder Signals

5.2 Test and Verification Plan

There are 2 major subsections of the test and verification plan: hardware and software. Our team first had to verify that the boosterpack and LaunchPad could be meshed together, and then we tested the sensors in the motor. While one part of the team worked on troubleshooting the hardware, the other half of the team worked on changing the example code in the SDK.

5.2.1 Hardware

5.2.1a Boosterpack Compatibility

While we set up the AM263x LaunchPad, we used an old LaunchPad with an MSP430 MCU with the BOOSTXL-DRV8323RH to spin the motor. Since the team has prior experience with the MSP series, the coding process was more intuitive. We first used jumper cables to connect the two boards. We connected the LaunchPad to the computer, and the 3.3V pin to pull the enable pin on the boosterpack high.

We then had to verify that the DRV8323RH boosterpack and Sitara LP were compatible. As shown in the table we created in Section 3.2, we labeled each pin with their function. In the datasheet, a pinmux shows the possible functions of each pin. Using the pinmux, the team understood which pins could stay in default mode and which pins needed to be changed. Once we configured the most basic pins on the AM263x LP, we used jumper cables to connect it to the boosterpack. The next subsection describes how we programmed the board using the SDK to mount the boosterpack onto the LaunchPad.

5.2.1b Motor Control Functionalities

Before using the hall sensors from the motor, we simulated the hall sensors using code to ensure the phases were working as expected. Once we verified that our running code could process the hall sensor signals and we could spin the motor, we connected the motor leads to an oscilloscope and turned the motor by hand to verify that the internal hall sensors were functioning correctly. Finally, we connected the motor phase screw-in terminals and hall sensor terminals to the respective leads coming from our motor. The INHC and INLC headers on the boosterpack correspond to direction and brake, respectively. We built a simple button circuit externally to switch the pins HIGH/LOW to test these functionalities. We were successfully able to brake the motor and change its direction. In addition, we created another external circuit with a potentiometer to increase/decrease the duty cycle and speed of the motor.

5.2.1c Switch Debouncing

In our final implementation, we will use switches to control the direction and brake functions of the motor. Switches often have the problem of bouncing between values more than one time when the switch is thrown. If a switch triggers an interrupt, it is not desirable for the interrupt to trigger multiple times for a single switch toggle. A debouncing circuit smoothes out the rapid changes in the switch-controlled circuit to provide a single rising/falling edge on a switch toggle

The figure below is the schematic for the switch debouncing circuit.

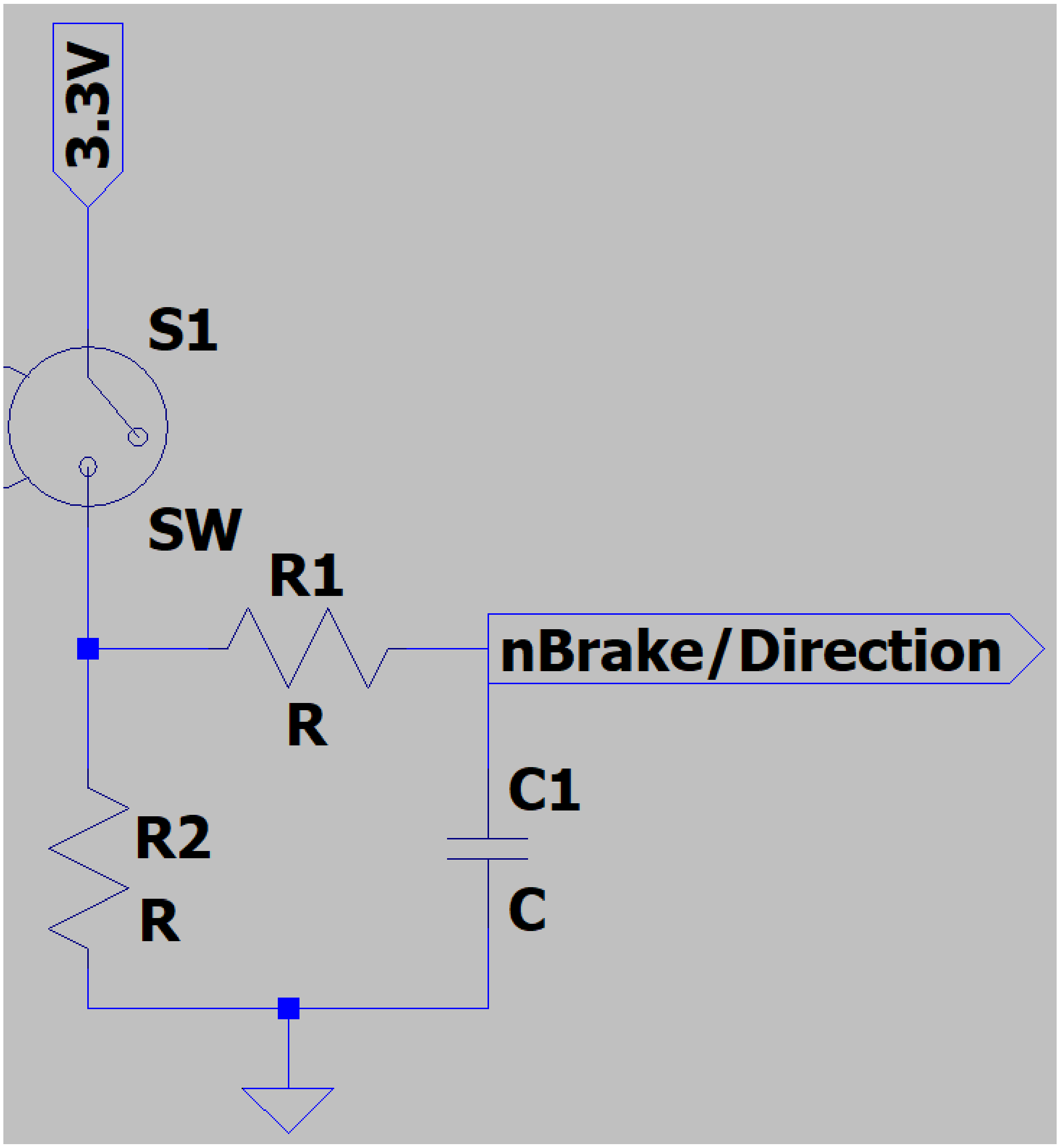


Figure 5.11. Schematic for Switch Debouncing

Using the circuit above, with R1 = 1.2kΩ, R2 = 12kΩ, and C = 4.7nF, we scope the output signal. The desired behavior is a single rising or falling edge. The following figure shows the scope output.

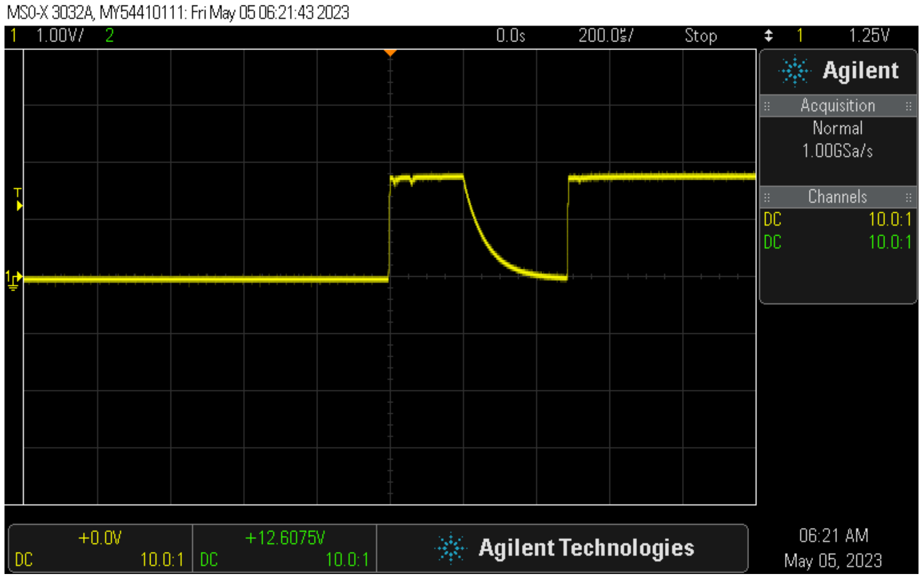


Figure 5.12. Switch Debouncing Output with R1 = 1.2kΩ, R2 = 12kΩ, and C = 4.7nF

In the figure, we see the initial rising edge, but switch bouncing causes the output signal to drop again before finally increasing. This is because the time constant of the circuit is too small. To remedy this, we increase the capacitor value from 4.7nF to 100nF and repeat the experiment on a rising edge and a falling edge. The following figures show the results.

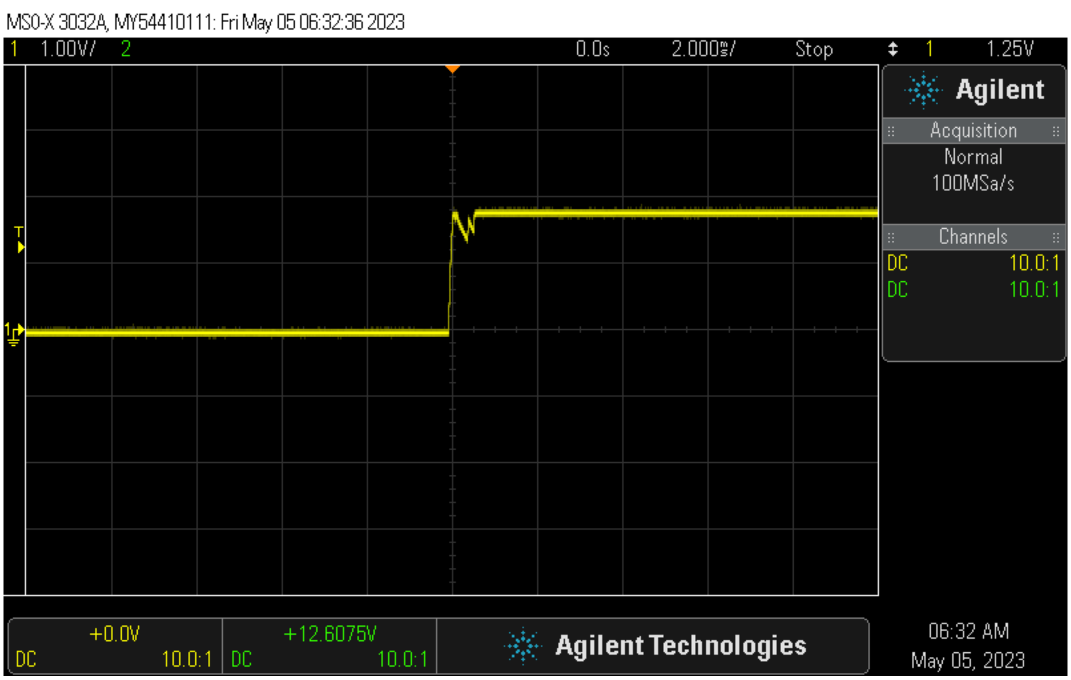


Figure 5.13. Rising Edge Switch Debouncing with R1 = 1.2kΩ, R2 = 12kΩ, and C = 100nF

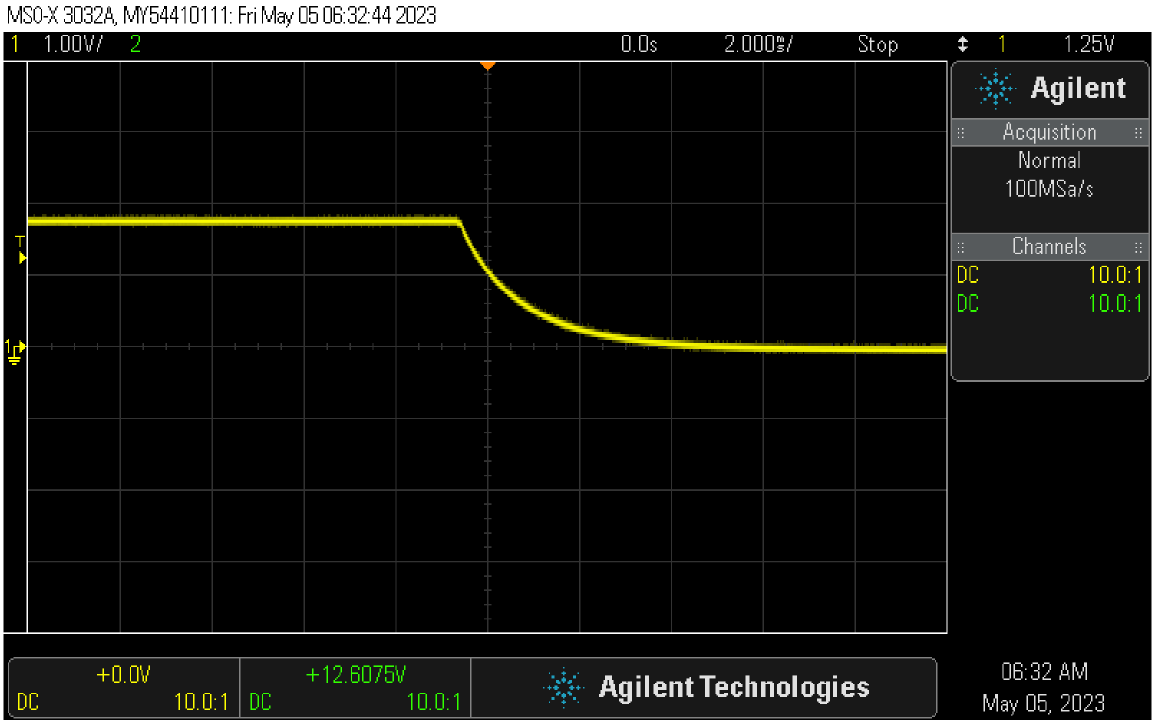


Figure 5.14. Falling Edge Switch Debouncing with R1 = 1.2kΩ, R2 = 12kΩ, and C = 100nF

In the rising edge switch debouncing output, we see a small bit of bouncing. However, small fluctuations in the output are okay as long as the fluctuations do not cross the boundary for a logic high signal. In the falling edge output, the falling edge is smooth. The large RC time constant causes a large amount of time for the signal to fall to logic 0, but this is okay for our application. In the final implementation, 3.3kΩ resistors were used instead of the previous values.

5.2.2 Software

5.2.2a GPIO Pin Configuration

To connect the AM263x LP and the boosterpack, the GPIO pins had to be configured first. Many of these pins required a change using the Sysconfig tool in the SDK to change a pin from its default function. The figure below shows an example of a GPIO pin on the Sitara LP that was configured to pull the enable pin on the DRV8323RH boosterpack high. In the GUI, the pin was given a descriptive name, told to pull high, given the exact GPIO pin number (GPIO74), and the ball number of the pin (R16) that corresponds to the IC itself. We configured several pins this way to pull the enable, brake, and direction pins on the boosterpack either high or low to verify the motor control functionalities.

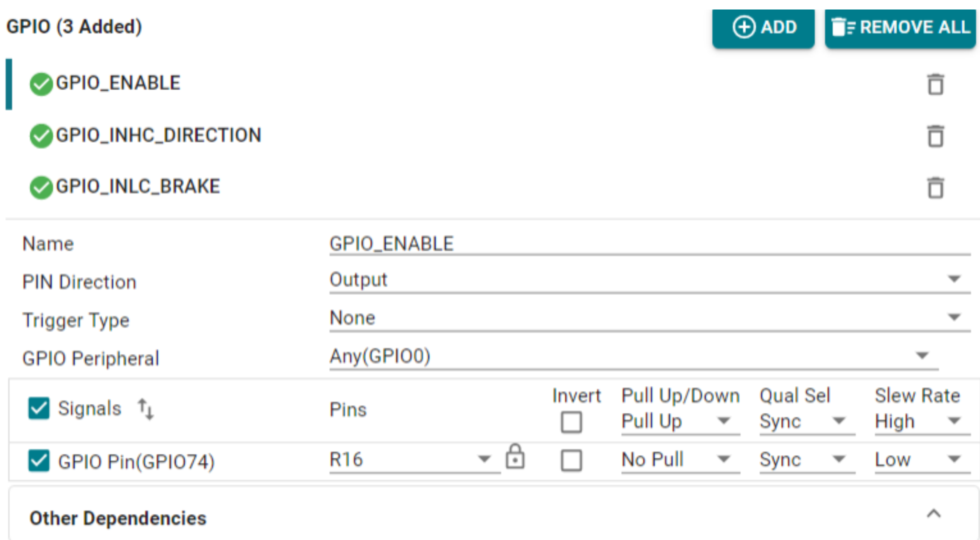


Figure 5.15. Sysconfig GPIO_ENABLE Example

5.2.2b PWM and ADC Code

We used the example PWM and ADC code in the SDK and individually tested them on the LaunchPads first to verify that each of them worked. We then put the two files together to ensure that we could read an ADC signal from the potentiometer and output the correct PWM. The PWM values were printed in the terminal, and we connected the pins to the oscilloscope to verify that the duty cycle was changing. As the potentiometer turned, we saw the duty cycle and values change accordingly. We also used a multimeter to verify that the ADC levels were outputting the correct voltage. Once these basic testing steps were completed, we started setting up the final product.

5.2.2c eQEP

Our motor encoder datasheet advertises 1000 pulses/revolution of the motor. Our motor has a 116:1 input:output gearbox, meaning 116 revolutions of the internal motor makes one full revolution of our output motor. This means that 1000*116 = 116,000 pulses for every full output revolution.

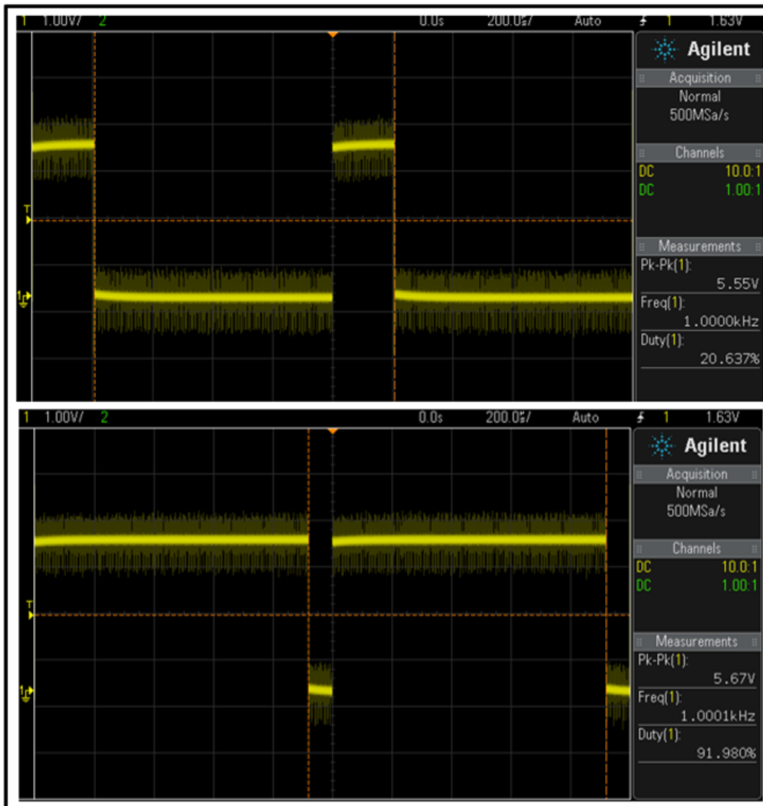
To verify this, we connected the encoder A signal to the eQEP peripheral on the AM263 Launchpad. The eQEP peripheral counts either rising/falling/rising and falling edges of a square wave input, designed for keeping track of motor encoder positions. We count the number of rising edges of the encoder signal and stop the motor once 116,000 pulses are counted, and observe whether the output motor spun for 1 full revolution. To make this observation more accurate, we observe the motor spinning at a constant rate for 30 revolutions. We expect that the number of encoder pulses would be 30 * 116,000 = 2,320,000. The error was 226 encoder pulses, which represents about 0.2% of a full rotation. This suggests that our counting measure and encoder/gearbox pair has 116,000 encoder pulse, as advertised.

5.3 Evaluation of Results

The main objective of this project was to have configurable motor control. Although we did not design one integrated PCB, we were still able to implement some rudimentary motor control concepts using the LaunchPad and Boosterpack.

One of the criteria was having high torque. We were able to demonstrate this by attaching a weighted arm to the shaft of the motor. When we pushed down on the arm, we observed a spike in the current as the motor had to draw more current to increase its performance and maintain its speed.

Another main objective of the project was to control the speed. We control the speed by using a potentiometer to adjust the PWM, which in turn adjusts the duty cycle. In the figures below, we adjusted the duty cycle from 100% to 0% and show the duty cycle waveform on the left. Then we were able to calculate the speed in rotations per second and display it in the terminal. We were able to achieve this in 1x PWM mode since our chosen DRV8323RH boosterpack can calculate all 3 phases using just one phase input at a time.




```
ADC In : 4095   Duty Cycle: 1.000
ADC In : 4095   Duty Cycle: 1.000
ADC In : 4095   Duty Cycle: 1.000
ADC In : 3983   Duty Cycle: 0.973
ADC In : 3732   Duty Cycle: 0.911
ADC In : 3529   Duty Cycle: 0.862
ADC In : 3132   Duty Cycle: 0.765
ADC In : 2971   Duty Cycle: 0.726
ADC In : 2973   Duty Cycle: 0.726
ADC In : 2968   Duty Cycle: 0.725
ADC In : 2675   Duty Cycle: 0.653
ADC In : 2411   Duty Cycle: 0.589
ADC In : 2131   Duty Cycle: 0.520
ADC In : 1926   Duty Cycle: 0.470
ADC In : 1577   Duty Cycle: 0.385
ADC In : 1359   Duty Cycle: 0.332
ADC In : 1173   Duty Cycle: 0.286
ADC In : 1007   Duty Cycle: 0.246
ADC In : 912   Duty Cycle: 0.223
ADC In : 913   Duty Cycle: 0.223
ADC In : 733   Duty Cycle: 0.179
ADC In : 601   Duty Cycle: 0.147
ADC In : 437   Duty Cycle: 0.107
ADC In : 226   Duty Cycle: 0.055
ADC In : 3   Duty Cycle: 0.001
ADC In : 0   Duty Cycle: 0.000
```

Figure 5.16. Duty Cycle Adjustment

```
EQEP delta: 715
Timer delta (10us periods): 778
Speed (RPS): 0.79

EQEP delta: 713
Timer delta (10us periods): 681
Speed (RPS): 0.90
```

Figure 5.17. Speed Calculation Display

Finally, to gain positional control, we had to use the encoder which was mounted on the back of the motor. In the figure below, the user is able to specify in code what the target angle is, and the display shows the starting position, target angle, and the error based on where the motor actually stopped. Since our encoder has such fine resolution, we were able to gain decent position control with less than 0.5% error.

```
Starting ...
RESETTING EQEP POSITION
Starting position: 0
Target angle: 1, position = 322
Ending position: 780
Error = 458 encoder counts
Error = 0.3948% revolution

RESETTING EQEP POSITION
Starting position: 0
Target angle: -90, position = 29000
Ending position: 29227
Error = 227 encoder counts
Error = 0.1957% revolution

RESETTING EQEP POSITION
Starting position: 0
Target angle: 360, position = 116000
Ending position: 116075
Error = 75 encoder counts
Error = 0.0647% revolution

RESETTING EQEP POSITION
Starting position: 0
Target angle: -2634, position = 848733
Ending position: 849281
Error = 548 encoder counts
Error = 0.4724% revolution

EPWM Duty Cycle Test Passed!!
All tests have passed!!
ADC Software Triggered Conversion Test Passed!!
All tests have passed!!
```

Figure 5.18. Encoder Position Control Output

6. Standards Used

The following standards were used in the development of this project:

- Texas Instruments C-Style Coding Guidelines
- Texas Instruments AM263x MCU+ SDK Developer Guidelines 08.06.00
- EnDat 2.2 standard for motor encoder positional feedback
- Anaheim Automation Motor and Encoder Troubleshooting Guidelines
- RS232/EIA232 for UART (Universal Asynchronous Receiver Transmitter) communication

7. Conclusion, Overview of Work Statement

Our team successfully demonstrated the abilities of the Sitara AM263x MCU by controlling a BLDC motor. We used the DRV8323RH boosterpack and the LP-AM263x in conjunction with each other to control a 24V Anaheim motor with speed and positional control in 1x PWM mode. The team programmed the LaunchPad using the SDK to configure the pins, control the speed with a potentiometer, and read in an angle from the user to turn the motor to a specified position. Additionally, we programmed the Sitara LaunchPad to brake the motor and change direction with installed safety switches for both of these functions and a knob to easily turn the potentiometer. Our team split up tasks to work efficiently throughout the year, and we upheld the highest ethical standards while doing so.

8. Future Work Recommendations

8.1 Hardware

Future work on the hardware aspect of the Sitara motor driver project would include shortening and organizing the wires connecting from the motor to the boards. Additionally, finding a compatible encoder with an index, or using a different motor with an indexed encoder, will ease the software aspect of working with the eQEP function of the MCU.

As mentioned briefly in Section 4.1.2, the DRV8323R has a software variant (DRV8323RS). Using the DRV8323RS instead of the DRV8323RH (hardware variant) will allow for simpler mode configurability within the software rather than having to solder or desolder components on the hardware.

To create the fully integrated PCB design of the MCU and motor driver system, it is recommended to become highly familiar with PCB design considerations of the individual components and best practices for working with complex designs. Utilizing aspects from the existing PCB designs from TI's EVMs (the LP-AM263x and BOOSTXL-DRV8323Rx) would aid in streamlining an integrated design.

8.2 Software

Due to time constraints and the complexity of programming the MCU, there are a few future improvements that could be made on the performance of the system. Implementing 3x PWM mode, or even 6x PWM mode, on the DRV by configuring the Sitara to output multiple PWM signals simultaneously will increase the configurability of the system. Additionally, using PWM-triggered ADC SOC will improve the efficiency of the ADC conversion for the potentiometer adjustment of the duty cycle. As mentioned in the hardware section above, utilizing an indexed encoder will ease the use of the eQEP peripheral, allowing for easier implementation of both the positional and speed control demos; instead of solely calculating the speed, the user would be able to provide a specific speed as an input for the motor to achieve. Finally, implementing current sense measurements to help determine power and efficiency, as well as increase precision.

Further future work recommendations are listed in section 11.2 of tasks pending completion, indicating stretch goals of the project.

9. Other Issues

9.1 Ethics

Integrity

Throughout the project, Team SIMBA demonstrated integrity by being transparent with the project and the scope of its work with our sponsors, reading documentation and creating original work for our project, and communicated openly amongst team members.

Reliability

Team SIMBA strives for reliability in our project and committed to it by conducting thorough testing on our project and debugging any obstacles we faced along the way.

Safety

Safety is one of Team SIMBA's highest priorities and ensured that each team member was safe and healthy throughout the entire project. We upheld our safety by researching any new materials we used, and proceeding with caution when implementing features.

Respect for Others

Each member of team SIMBA treated each other with respect in various ways. Each member was expected to listen to other member's ideas and opinions, allow everyone the chance to contribute, and talk in a polite and courteous manner.

9.2 Soft Skills

Lifelong Learning

Our team has learned a variety of skills through our project with Texas Instruments. Each member on the team has also learned the importance of consistency and making progress in small portions through a long period of time, which resulted in our final product.

Time Management

Our team learned how to organize our time to come in together and work on our project regardless of our schedules. It also helped us understand what it would be like working full time and having to attend weekly meetings and updating everyone on the progress that has been made.

Multidisciplinary Teams

Our project revolved around embedded systems which is a combination of both hardware and software work.The team consisted of six members, where 4 were Computer Engineers and the other two were Electrical Engineers. With our combined skill sets, the team had someone to bounce their ideas off of and contribute to the project.

9.3 Operating Instructions (Quick Start Guide/Manual)

9.3.1 Setup

Refer to Table 3.1, Figure 4.5, and Figure 5.2 to mount the DRV8323 boosterpack onto the AM263x LaunchPad.

Refer to the datasheet for the specific DRV8323 boosterpack to configure 1x PWM operation. We used the DRV8323RH. To configure 1x PWM operation on the DRV8323RH, replace R36, R37, and R38 with 0 Ohm resistors or a solder connection.

Connect the 5V USB-C power input on the AM263x Launchpad to a stable 5V power source. Many modern phone chargers work for this. Connect the UART0 microUSB input on the Launchpad to a computer USB port with UART capabilities.

See the MCU+ Academy for AM263x for detailed instructions for setting up the Code Composer Studio IDE, compiling programs, and running programs on the AM263x Launchpad. Specifically, the EVM Setup to ensure correct UART operation and the Download, Install and Setup CCS to create a target configuration for debugging. A brief description of the steps to run a program follows after these steps follows:

1. Open Code Composer Studio and ensure that the requisite setup steps are completed.
2. Download the projects from our Github
3. Click View->Project Explorer
4. Right click in the Project Explorer window -> Import -> CCS Projects
5. Browse to the directory where the projects are saved
6. Import the desired projects
7. Right click on a project -> Build project
8. Click View->Target Configurations
9. Launch the target configuration created in the Download, Install and Setup CCS steps
10. In the Debug window, click on the Cortex_R5_0
    a. Right click->Connect Target
    b. Shortcut: Ctrl+Alt+C
11. Click Run->Reset->CPU Reset

    a. Shortcut: Ctrl+Shift+R
12. Click Run->Load->Load program
    a. Shortcut: Ctrl+Alt+L
13. Browse for the compiled file for the desired project. You may need to look into the subfolders of the project. The name will be "<project_name>.out", then click OK
14. The first line of the program's main.c function should be highlighted
    a. Click Run to see Debug options (step over, step into, etc.)
    b. Press F8 to run the program

The following connections are necessary for both demo programs, in addition to mounting the DRV8323 boosterpack onto the AM263x Launchpad:

Connect the required positive and negative power to the motor's Hall sensors per the datasheet. Supply the motor drive voltage (~18V in our case) across the DRV8323RH J1 screw terminals. Connect the motor's phase A, B, and C inputs to the corresponding screw terminals on the DRV8323RH J5 header.

Supply power to the motor encoder according to its datasheet. In our case, the encoder requires 5V and Ground, so we connected the corresponding encoder pins to J24-4 and J24-5, respectively. Connect the encoder A signal to J24-1.

Construct two identical switch debouncing circuits as described in Figure 5-11.
Connect output of the Brake switch debouncing circuit to J1-3.
Connect output of the Direction switch debouncing circuit to J2-11.
Connect the output of the potentiometer, whose output ranges from [0V, 3.3V], to J3-26.

9.3.2 Switch Interrupt Verification

Two included projects are titled "demo_interrupt" and "demo_interrupt2". These projects are included to test and verify the hardware switches for Brake and Direction. The project "demo_interrupt" tests the operation of the switch whose output is connected to J2-11, the Direction switch. The project "demo_interrupt2" tests the operation of the switch whose output is connected to J1-3, the Brake switch.

Running either of these programs will print the number of times the switch transition has been detected, either a rising or falling edge. Verify that flipping the switch on and off each increment the printed counter value by exactly 1. If the counter value increases by more than one, there is likely extra switch bouncing causing multiple interrupts to trigger. This may require adjustments to the switch debouncing circuit, increasing the values of R1, R2, or C. See Section 5.2.1c for details on switch debouncing. We recommend running both of these programs to verify the switches are configured correctly.

9.3.3 Free-Run Operation

The project name is "demo_freerun". Upon running the program, user can turn the potentiometer on the box to adjust the speed of the motor. There are also two switches that can be flipped: The middle one changes the direction and the one on the side will cause the motor to brake. If the brake is off and the direction is switched, the motor will stop for a brief moment before switching directions. If the brake is on and the direction is switched, the motor will automatically start spinning in the new direction. The initial behavior of the motor will also depend on the current switch positions.

9.3.4 Positional Control Operation

The project name is "demo_position". The program will indefinitely prompt the user to enter how many degrees (integer values) the motor rotates. The degrees entered can be negative (rotates clockwise) or positive (rotates counterclockwise). When finished, the user should input "done" and the program will terminate

Additional documentation on this project can be found on GitHub.

10. Costs Estimate

10.1 Materials and Construction

Given a budget of $2000 to order parts plus $1000 to use in the UTDesign machine shop, we purchased items from Amazon, Mouser Electronics, and Texas Instruments throughout the course of the project.

During the Fall 2022 semester, we designed a prototype motor driver to spin a motor on a smaller scale than our intended project to demonstrate our knowledge of motor driver systems. The expense report for the first semester is shown in the table below.

Component	Part Number	Supplier	Quantity	Cost
USB C Wall Adapters	-	Amazon	3	$16.80
MOSFETs	IRFZ44NPBF	Mouser Electronics	6	$9.24
BLDC 12V Motor + Encoder	FIT0441	Mouser Electronics	1	$19.90
Motor Driver	UCC37321P	Texas Instruments	3	$3.60

Current Sense Amplifiers	INA210BIDCKR	Texas Instruments	3	$2.35
			Shipping	$12.00
			Total	**$63.89**

Table 10.1. Fall 2022 Expense Summary

The final system design was implemented during the Spring 2023 semester. The expense report for this semester is shown below.

Component	Part Number	Supplier	Quantity	Cost
BLDC 24V Motor + Encoder	BLWRPG173D-24V-4000R116-1000 DN8 -	Anaheim Automation	1	$357.00
Encoder Cable	ENC-CBL-CA-MIC6-SH-NC-1	Anaheim Automation	1	$13.00
24V, 20A Adjustable Power Supply	-	Amazon	1	$58.99
Project Case Enclosure	-	Amazon	1	$28.99
Sliding T-Slot Nuts	-	Amazon	50	$7.59
Hexagon Socket Screw Set	M2 M3 M4 M5	Amazon	880 pcs	$17.99
24V Toggle Switches	90014E	Amazon	5	$8.66
10kΩ Potentiometers	-	Amazon	3	$10.99
			Shipping	$33.98
			Total	**537.19**

Table 10.2. Spring 2023 Expense Summary

The overall project cost totalled $601.08, with about 70% of the parts budget (80% of the total budget) unused.

10.2 Estimate of Design Cost

During the first semester, a total of 616.5 man-hours were put towards project completion to meet deliverables such as the Preliminary Design Review and Critical Design Review. The second half of the project accumulated about 897 hours more, totalling at 1513.5 man-hours across both the Fall 2022 and Spring 2023 semesters. Each team member contributed between 5 to 30 hours on a weekly basis.

11. Project Management Summary

11.1 Tasks Completed

11.1.1 Fall 2022 Semester

The following table lists the tasks completed during the first semester of the project and which team member worked on the task.

Task	Member
Commutation Research	Shruthi
Current Feedback Research	Jenna
Encoder Research	Jenna
Motor Driver Research	Shruthi
High-Level System Diagram	Isaac, Coleman
PC Communications Selection	Victor
Power Filtering Selection	Mijwad
MOSFET and Driver Selection	Coleman, Shruthi
Current Sense Selection	Isaac
Motor Selection	Coleman, Shruthi
Encoder Selection	Jenna, Coleman
Procurement Requests	Isaac
Project Abstract	Isaac, Jenna, Shruthi
Schematic	Isaac, Jenna
Breadboard Prototype Testing	Mijwad, Shruthi
SDK Testing	Isaac, Mijwad
Expo Poster	Jenna, Shruthi
Expo Slide	Mijwad, Shruthi

Table 11.1 Fall 2022 Tasks + Contribution

11.1.2 Spring 2023 Semester

During the Spring 2023 Semester, the team completed about 44 tasks indicated in the table below. The Gantt charts for each semester are located in the Appendix section.

Task	Member
PC Communication Schematic	Victor
Input Power Schematic	Mijwad
Driver + MOSFETs PCB Research	Shruthi
SDFM + MMC on MCU Schematic	Coleman
EPWM + GPIO on MCU Schematic	Jenna
ADC + GPIO on MCU Schematic	Isaac
Finalized Integrated Schematic	Jenna
Through-hole Adapter PCBs	Isaac, Coleman
“Hello world” in Sitara SDK	Isaac, Jenna, Mijwad
TI Motor Driver Deep Dive Session	All
DRV8323RH Boosterpack Request	Shruthi
Test Power Traces on LP	Isaac
Blink on Sitara LP	Isaac
Setup DRV Firmware	Shruthi
PWM on LP Test	Isaac, Mijwad
DRV to LP Mapping	Coleman, Shruthi
Duty Cycle of PWM Adjustment	Isaac, Jenna
Setup Code for PWM Function	Isaac
ADC Example on LP	Mijwad
1x PWM Mode on DRV EVM	Coleman
Verification of 1x PWM Truth Table	Coleman, Shruthi
Setup Code for ADC Function	Isaac, Mijwad

Spin Motor with 1x PWM	Jenna, Coleman, Shruthi
ADC + EPWM Combo in SDK for Potentiometer Adjustment of Duty Cycle	Isaac
DRV8323RH Fault Troubleshooting	Isaac, Coleman, Mijwad, Shruthi
3x PWM Output from Sitara Attempt	Isaac, Jenna
1x PWM Mode with Sitara LP	Isaac, Jenna, Coleman
Direction + Brake Testing	Isaac, Jenna, Shruthi
Large Power Supply Troubleshooting	Mijwad, Victor
DRV to LP Pin Adjustments	Shruthi
Configuration of Sitara Pins	Isaac, Mijwad
Motor Mount Design	Coleman
3D Print Motor Mount	Coleman
Mount + Weighted Arm Attachment	Coleman
ADC Conversion Research	Isaac, Coleman
Current Sense Research	Shruthi
Code Efficiency Tests	Isaac
PWM-Triggered ADC SOC Attempt	Isaac, Mijwad
DRV to LP Direct Mount	Isaac, Mijwad
EQEP + Encoder Troubleshooting	Isaac, Jenna, Coleman, Mijwad
Final Report	Jenna, Victor, Shruthi
Final Build	Isaac, Coleman, Mijwad
Expo Poster	Shruthi
Expo Slide	Shruthi

Table 11.2. Spring 2023 Task + Contribution

11.2 Tasks Pending Completion

Below lists the stretch goals of the current project implementation to enhance presentation and tweak specific functions:

- Implement 3x PWM mode in the DRV by configuring the Sitara to output three different PWM signals simultaneously
    - This will allow for sinusoidal commutation
- Implement PWM-triggered ADC SOC to speed-up the ADC conversion for the potentiometer adjustment of the duty cycle
- Implement enhanced speed control using the eQEP peripheral

To complete the project with the initial desired end product of a fully integrated PCB design for high torque robotic applications, the following tasks must be completed:

- Finish the schematic in Figure ## with all the required passive components
- Generate the PCB design from the schematic in KiCad
    - Follow all PCB guidelines outlined in TI's documentation for hardware design
    - Run design checks in KiCad
- Get the PCB fabricated
- Troubleshoot first round PCB design
    - Make necessary changes to original schematic and PCB design
- Get finalized PCB fabricated

11.3 Time Allocated

The project ran from September 2022 to May 2023. See the Appendix for Gantt charts indicating more details of the time allocated to each task during the first and second semester.

11.4 Facilities Used

Synergy Park North Lab 1.220

11.5 Personnel

This project team consisted of Isaac Brooks, Jenna Evans, Coleman Gamble, Mijwad Kabir, Victor Lopez, and Shruthi Subramanium. Dr. Neal Skinner was our faculty advisor, who met with the team on a weekly basis and kept us up to date on the semester timeline. Our assigned corporate mentors and sponsors were Randy Rosales and Anita Pratti, who also attended our weekly meetings to provide knowledge, suggestions, and equipment on the Sitara MCU for motor driver functions. James Lockridge, an employee on the motor driver team at TI, also

provided the team with information and equipment for understanding and running the motor driver system of the project. Oddrun Mahaffey and Anuradha Goel aided in project procurement requests. We received additional aid from Gene Woten in the UTDesign lab to acquire access to lab equipment, as well as from Max Steele for help with our purchased power supply.

12. Appendices

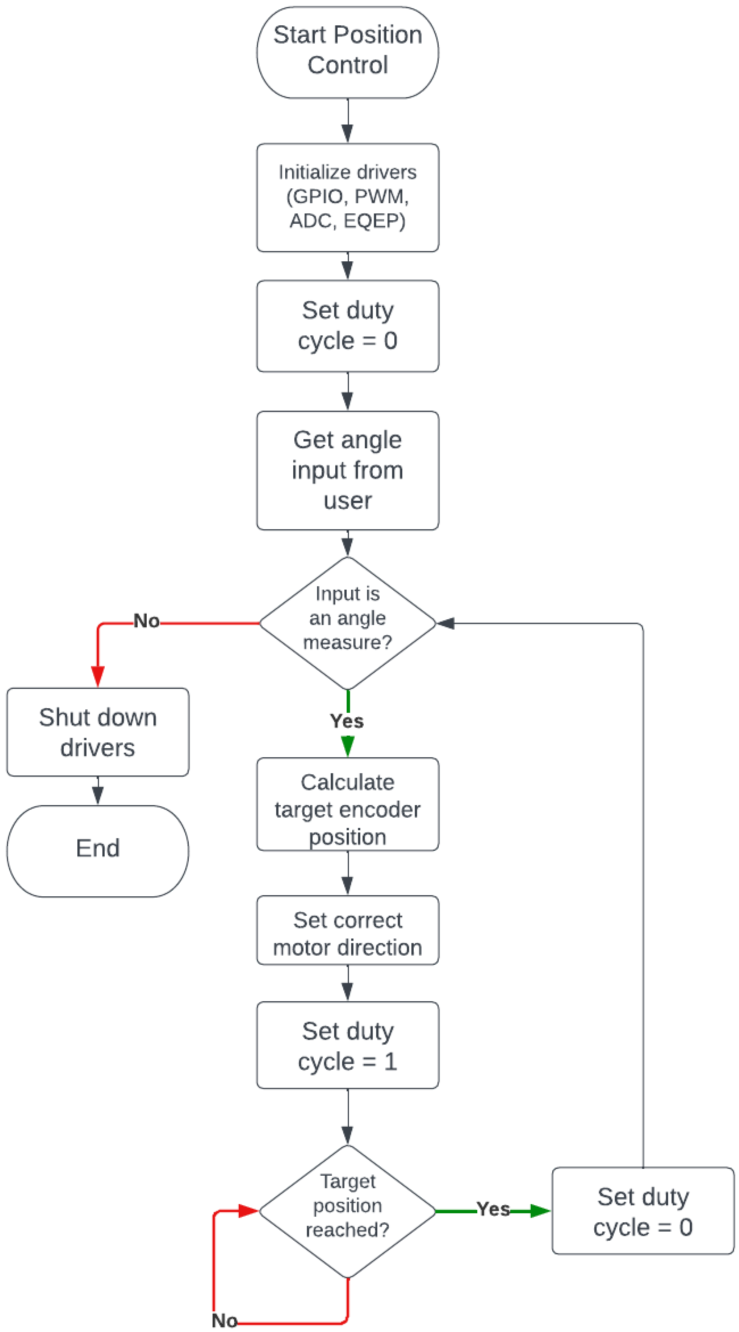


Figure 12.1. Positional Control Software Flow

	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	Kick-off Meeting	0 days	Mon 9/19/22	Mon 9/19/22		Isaac B,Jenna E,Coleman G,Mijwad K,Victor L
2	Research	14 days	Tue 9/20/22	Fri 10/7/22	1	Isaac B,Jenna E,Mijwad K,Shruthi S,Victor L,C
3	Define problem and scope	8 days	Mon 10/10/22	Wed 10/19/22	2	Coleman G,Isaac B,Jenna E,Mijwad K, Shruthi S,Victor L
4	Understand background, context, and impact	8 days	Mon 10/10/22	Wed 10/19/22	2	Coleman G,Isaac B,Jenna E,Mijwad K, Shruthi S,Victor L
5	Define design requirements	6 days	Thu 10/20/22	Thu 10/27/22	3,4	Coleman G,Isaac B,Jenna E,Mijwad K, Shruthi S,Victor L
6	Commutation Approaches Research	6 days	Thu 10/20/22	Thu 10/27/22	3,4	Shruthi S
7	Curent Feedback Approaches Research	6 days	Thu 10/20/22	Thu 10/27/22	3,4	Jenna E
8	Encoder Approaches Research	6 days	Thu 10/20/22	Thu 10/27/22	3,4	Jenna E
9	Driver Appraches Research	6 days	Thu 10/20/22	Thu 10/27/22	3,4	Shruthi S
10	High-level Block Diagram	6 days	Thu 10/20/22	Thu 10/27/22	3,4	Coleman G,Isaac B
11	Preliminary Design Review Prep	1 day	Fri 10/28/22	Sun 10/30/22	10	Coleman G,Isaac B,Jenna E,Mijwad K, Shruthi S,Victor L
12	Preliminary Design Review	0 days	Mon 10/31/22	Mon 10/31/22	11	Coleman G,Isaac B,Jenna E,Mijwad K, Shruthi S,Victor L
13	Review PDR Feedback	5 days	Mon 10/31/2	Fri 11/4/22	12	Coleman G,Isaac B,Jenna E,Mijwad K,Shruthi
14	PC Communication Selection	8 days	Mon 11/7/22	Wed 11/16/22	13	Victor L
15	Voltage Regulators / Power Filtering Selection	8 days	Mon 11/7/22	Wed 11/16/22	13	Mijwad K
16	MOSFETs & Driver Selection	8 days	Mon 11/7/22	Wed 11/16/22	13	Coleman G,Shruthi S
17	Current Sense Selection	8 days	Mon 11/7/22	Wed 11/16/2	13	Isaac B
18	Motor Selection	8 days	Mon 11/7/22	Wed 11/16/2	13	Coleman G,Shruthi S
19	Encoder Selection	8 days	Mon 11/7/22	Wed 11/16/2	13	Coleman G,Jenna E,Shruthi S
20	Order components	2 days	Thu 11/17/22	Fri 11/18/22	14,15,16,17,18,19	Isaac B
21	Fall Break	5 days	Mon 11/21/2	Fri 11/25/22		Coleman G,Isaac B,Jenna E,Mijwad K,Shruthi
22	Project Abstract	5 days	Mon 11/28/2	Fri 12/2/22	12	Isaac B,Jenna E,Shruthi S
23	Schematic	7 days	Mon 11/28/2	Tue 12/6/22	14,15,16,17,18,19	Isaac B,Jenna E
24	Breadboard Testing	8 days	Mon 11/28/2	Wed 12/7/22	20,14,15,16,17,18,	Mijwad K,Shruthi S,Victor L,Coleman G
25	SDK Testing	5 days	Mon 12/5/22	Fri 12/9/22		Isaac B,Mijwad K
26	PCB Layout	3 days	Wed 12/7/22	Fri 12/9/22	23	Isaac B,Jenna E
27	Critical Design Review Prep	2 days	Sat 12/10/22	Sun 12/11/22	23,24,25,26	Coleman G,Isaac B,Jenna E,Mijwad K, Shruthi S,Victor L
28	Expo Poster	2 days	Fri 12/9/22	Sun 12/11/22	23,24,25,26	Isaac B,Jenna E,Mijwad K
29	Critical Design Review	0 days	Mon 12/12/2	Mon 12/12/2	27	Coleman G,Isaac B,Jenna E,Mijwad K,Shruthi
30	PPT Slide	2 days	Tue 12/13/22	Wed 12/14/2	28	Coleman G,Shruthi S,Victor L
31	Design Expo	0 days	Fri 12/16/22	Fri 12/16/22	28,29,30	Coleman G,Isaac B,Jenna E,Mijwad K,Shruthi

Figure 12.2. Fall 2022 Gantt Chart Tasks

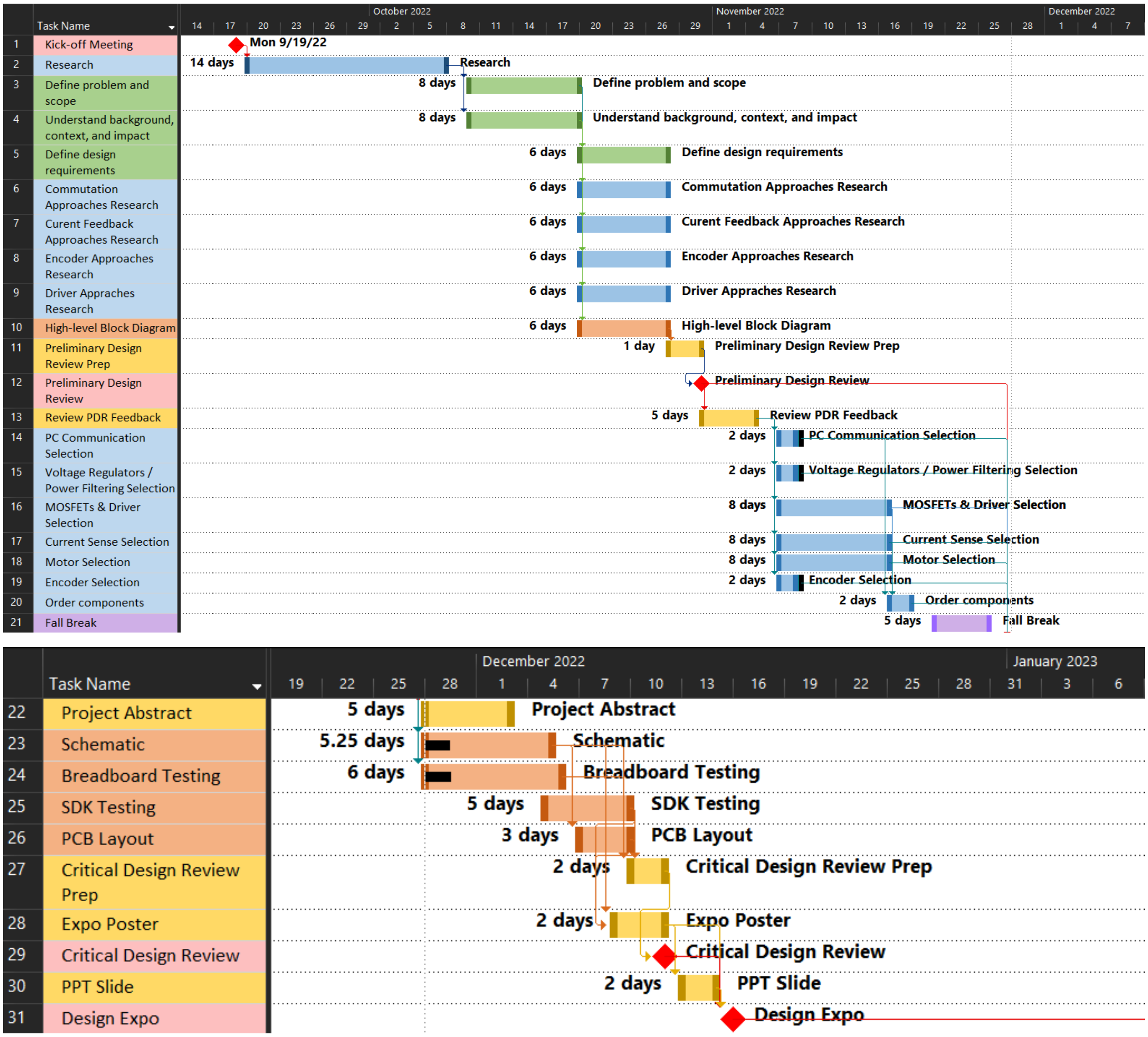


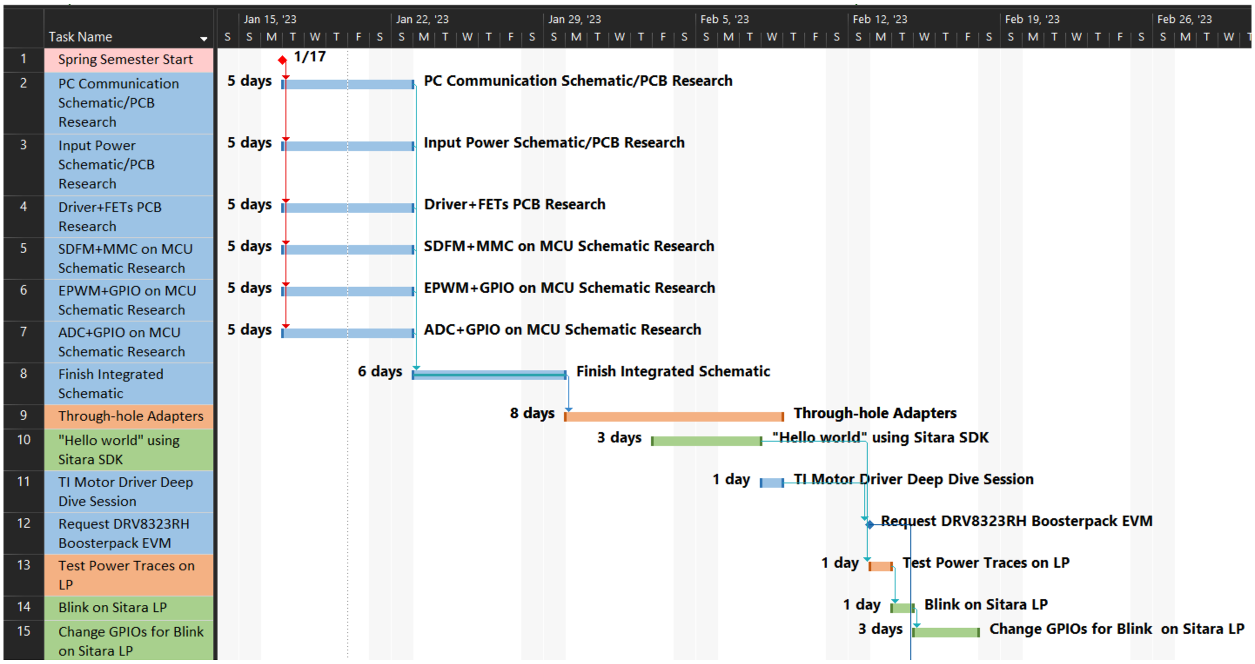
Figure 12.3. Fall 2022 Gantt Chart

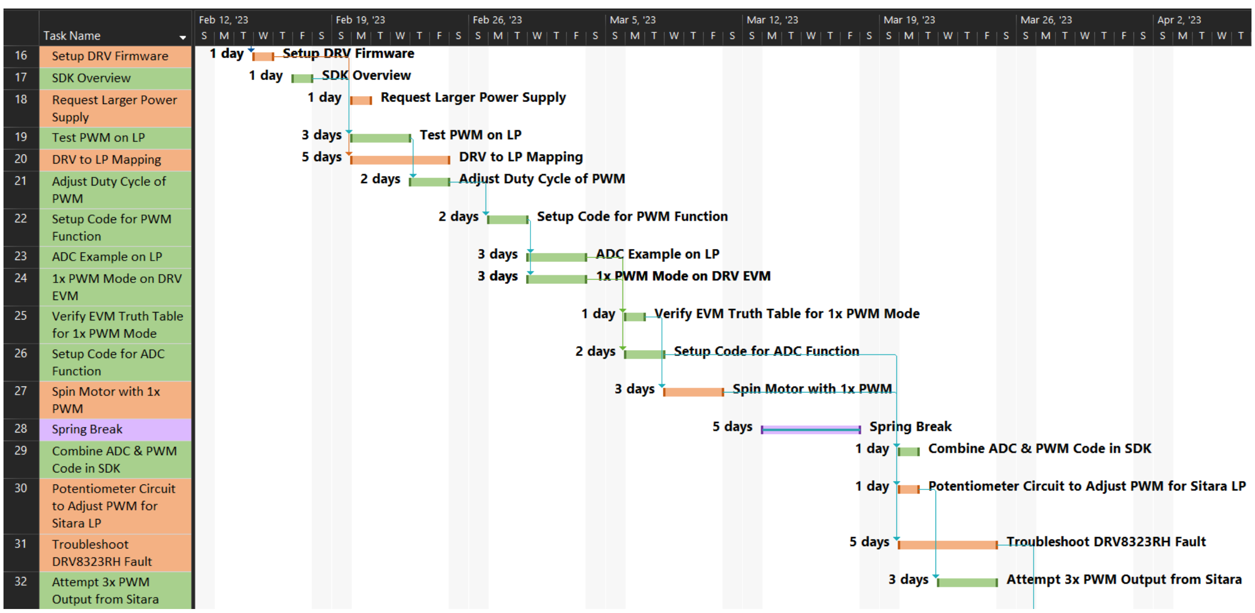
	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	Spring Semester Start	0 days	Tue 1/17/23	Tue 1/17/23		
2	PC Communication Schematic/PCB Research	5 days	Tue 1/17/23	Sun 1/22/23	1	Victor L
3	Input Power Schematic/PCB Research	5 days	Tue 1/17/23	Sun 1/22/23	1	Mijwad K
4	Driver+FETs PCB Research	5 days	Tue 1/17/23	Sun 1/22/23	1	Shruthi S
5	SDFM+MMC on MCU Schematic Research	5 days	Tue 1/17/23	Sun 1/22/23	1	Coleman G
6	EPWM+GPIO on MCU Schematic Research	5 days	Tue 1/17/23	Sun 1/22/23	1	Jenna E
7	ADC+GPIO on MCU Schematic Research	5 days	Tue 1/17/23	Sun 1/22/23	1	Isaac B
8	Finish Integrated Schematic	6 days	Mon 1/23/23	Sun 1/29/23	2,3,4,5,6,7	Jenna E
9	Through-hole Adapters	8 days	Mon 1/30/23	Wed 2/8/23	8	Coleman G,Isaac B
10	"Hello world" using Sitara SDK	3 days	Fri 2/3/23	Tue 2/7/23		
11	TI Motor Driver Deep Dive Session	1 day	Wed 2/8/23	Wed 2/8/23		Coleman G,Isaac B,Jenna E,Mijwad K, Shruthi S,Victor L
12	Request DRV8323RH Boosterpack EVM	0 days	Mon 2/13/23	Mon 2/13/23	11	Shruthi S
13	Test Power Traces on LP	1 day	Mon 2/13/23	Mon 2/13/23	10	Isaac B
14	Blink on Sitara LP	1 day	Tue 2/14/23	Tue 2/14/23	13	Isaac B
15	Change GPIOs for Blink on Sitara LP	3 days	Wed 2/15/23	Fri 2/17/23	14	Isaac B

16	Setup DRV Firmware	1 day	Wed 2/15/23	Wed 2/15/23	12	Shruthi S
17	SDK Overview	1 day	Fri 2/17/23	Fri 2/17/23		Coleman G,Isaac B,Jenna E,Mijwad K,Shr
18	Request Larger Power Supply	1 day	Mon 2/20/23	Mon 2/20/23		
19	Test PWM on LP	3 days	Mon 2/20/23	Wed 2/22/23	17	Isaac B,Mijwad K
20	DRV to LP Mapping	5 days	Mon 2/20/23	Fri 2/24/23	16	Coleman G,Shruthi S
21	Adjust Duty Cycle of PWM	2 days	Thu 2/23/23	Fri 2/24/23	19	Isaac B,Jenna E
22	Setup Code for PWM Function	2 days	Mon 2/27/23	Tue 2/28/23	21	Isaac B
23	ADC Example on LP	3 days	Wed 3/1/23	Fri 3/3/23	22	Mijwad K
24	1x PWM Mode on DRV EVM	3 days	Wed 3/1/23	Fri 3/3/23	22	Coleman G
25	Verify EVM Truth Table for 1x PWM Mode	1 day	Mon 3/6/23	Mon 3/6/23	24	Coleman G,Shruthi S
26	Setup Code for ADC Function	2 days	Mon 3/6/23	Tue 3/7/23	23	Isaac B,Mijwad K
27	Spin Motor with 1x PWM	3 days	Wed 3/8/23	Fri 3/10/23	25	Coleman G,Jenna E,Shruthi S
28	Spring Break	5 days	Mon 3/13/23	Fri 3/17/23		Coleman G,Isaac B,Jenna E,Mijwad K,Shr
29	Combine ADC & PWM Code in SDK	1 day	Mon 3/20/23	Mon 3/20/23	26	Isaac B
30	Potentiometer Circuit to Adjust PWM for Sitara LP	1 day	Mon 3/20/23	Mon 3/20/23	26	Isaac B
31	Troubleshoot DRV8323RH Fault	5 days	Mon 3/20/23	Fri 3/24/23	27	Coleman G,Isaac B,Mijwad K,Shruthi S
32	Attempt 3x PWM Output from Sitara	3 days	Wed 3/22/23	Fri 3/24/23	30	Isaac B,Jenna E

33	Acquire Replacement Boards	1 day	Mon 3/27/23	Mon 3/27/23	31	
34	1x PWM Mode w/ Sitara LP	2 days	Tue 3/28/23	Wed 3/29/23	33	Isaac B,Jenna E,Coleman G
35	Test Direction and Brake on Motor	3 days	Thu 3/30/23	Sun 4/2/23	34	Isaac B,Jenna E,Shruthi S
36	Button Circuit for Brake	1 day	Fri 3/31/23	Fri 3/31/23	33	Isaac B
37	Troubleshoot Big Power Supply	5 days	Mon 3/27/23	Fri 3/31/23		Mijwad K,Victor L
38	DRV to LP Pin Adjustments	2 days	Fri 3/31/23	Mon 4/3/23	34	Shruthi S
39	Configure Sitara Pins	3 days	Wed 4/5/23	Fri 4/7/23	38	Isaac B,Mijwad K
40	Motor Mount Design	2 days	Mon 4/3/23	Tue 4/4/23		Coleman G
41	3D Print Motor Mount	2 days	Wed 4/5/23	Thu 4/6/23	40	Coleman G
42	Attach Mount + Weighted Arm	1 day	Fri 4/7/23	Fri 4/7/23	41	Coleman G
43	Initial Demo	0 days	Mon 4/10/23	Mon 4/10/23	42	Coleman G,Isaac B,Jenna E,Mijwad K,Shr
44	ADC Conversion Research	4 days	Tue 4/11/23	Fri 4/14/23	43	Coleman G,Isaac B
45	Current Sense Research	4 days	Tue 4/11/23	Fri 4/14/23	43	Shruthi S
46	Code Efficiency Tests	5 days	Mon 4/17/23	Fri 4/21/23	44	Isaac B
47	PWM-triggered ADC SOC	3 days	Mon 4/17/23	Wed 4/19/23	44	Isaac B,Mijwad K
48	DRV to LP Direct Mount	3 days	Wed 4/19/23	Fri 4/21/23	43	Isaac B,Mijwad K
49	EQEP + Encoder Troubleshooting	5 days	Mon 4/24/23	Fri 4/28/23	48	Coleman G,Isaac B,Jenna E,Mijwad K
50	Final Report	16 days	Mon 4/24/23	Sat 5/13/23	43	Jenna E,Shruthi S
51	Final Build	5 days	Mon 5/1/23	Fri 5/5/23	49	Coleman G,Isaac B,Mijwad K
52	Expo Poster	5 days	Mon 5/1/23	Fri 5/5/23	49	Isaac B,Jenna E,Shruthi S
53	Single Expo Slide	5 days	Mon 5/1/23	Fri 5/5/23	49	Isaac B,Jenna E,Shruthi S
54	Final Tests	4 days	Mon 5/8/23	Thu 5/11/23	51	Coleman G,Isaac B,Mijwad K
55	Expo Day	0 days	Fri 5/12/23	Fri 5/12/23	52,53,54	Coleman G,Isaac B,Jenna E,Mijwad K,Shr
56	Final Report Due	0 days	Mon 5/15/23	Mon 5/15/23	55	Coleman G,Isaac B,Jenna E,Mijwad K,Shr

Figure 12.4. Spring 2023 Gantt Chart Tasks


Task Name
Jan 15, '23
Jan 22, '23
Jan 29, '23
Feb 5, '23
Feb 12, '23
Feb 19, '23
Feb 26, '23
1 Spring Semester Start
1/17
2 PC Communication Schematic/PCB Research
5 days
3 Input Power Schematic/PCB Research
5 days
4 Driver+FETs PCB Research
5 days
5 SDFM+MMC on MCU Schematic Research
5 days
6 EPWM+GPIO on MCU Schematic Research
5 days
7 ADC+GPIO on MCU Schematic Research
5 days
8 Finish Integrated Schematic
6 days
9 Through-hole Adapters
8 days
10 "Hello world" using Sitara SDK
3 days
11 TI Motor Driver Deep Dive Session
1 day
12 Request DRV8323RH Boosterpack EVM
13 Test Power Traces on LP
1 day
14 Blink on Sitara LP
1 day
15 Change GPIOs for Blink on Sitara LP
3 days


Task Name
Feb 12, '23
Feb 19, '23
Feb 26, '23
Mar 5, '23
Mar 12, '23
Mar 19, '23
Mar 26, '23
Apr 2, '23
16 Setup DRV Firmware
1 day
17 SDK Overview
1 day
18 Request Larger Power Supply
1 day
19 Test PWM on LP
3 days
20 DRV to LP Mapping
5 days
21 Adjust Duty Cycle of PWM
2 days
22 Setup Code for PWM Function
2 days
23 ADC Example on LP
3 days
24 1x PWM Mode on DRV EVM
3 days
25 Verify EVM Truth Table for 1x PWM Mode
1 day
26 Setup Code for ADC Function
2 days
27 Spin Motor with 1x PWM
3 days
28 Spring Break
5 days
29 Combine ADC & PWM Code in SDK
1 day
30 Potentiometer Circuit to Adjust PWM for Sitara LP
1 day
31 Troubleshoot DRV8323RH Fault
5 days
32 Attempt 3x PWM Output from Sitara
3 days

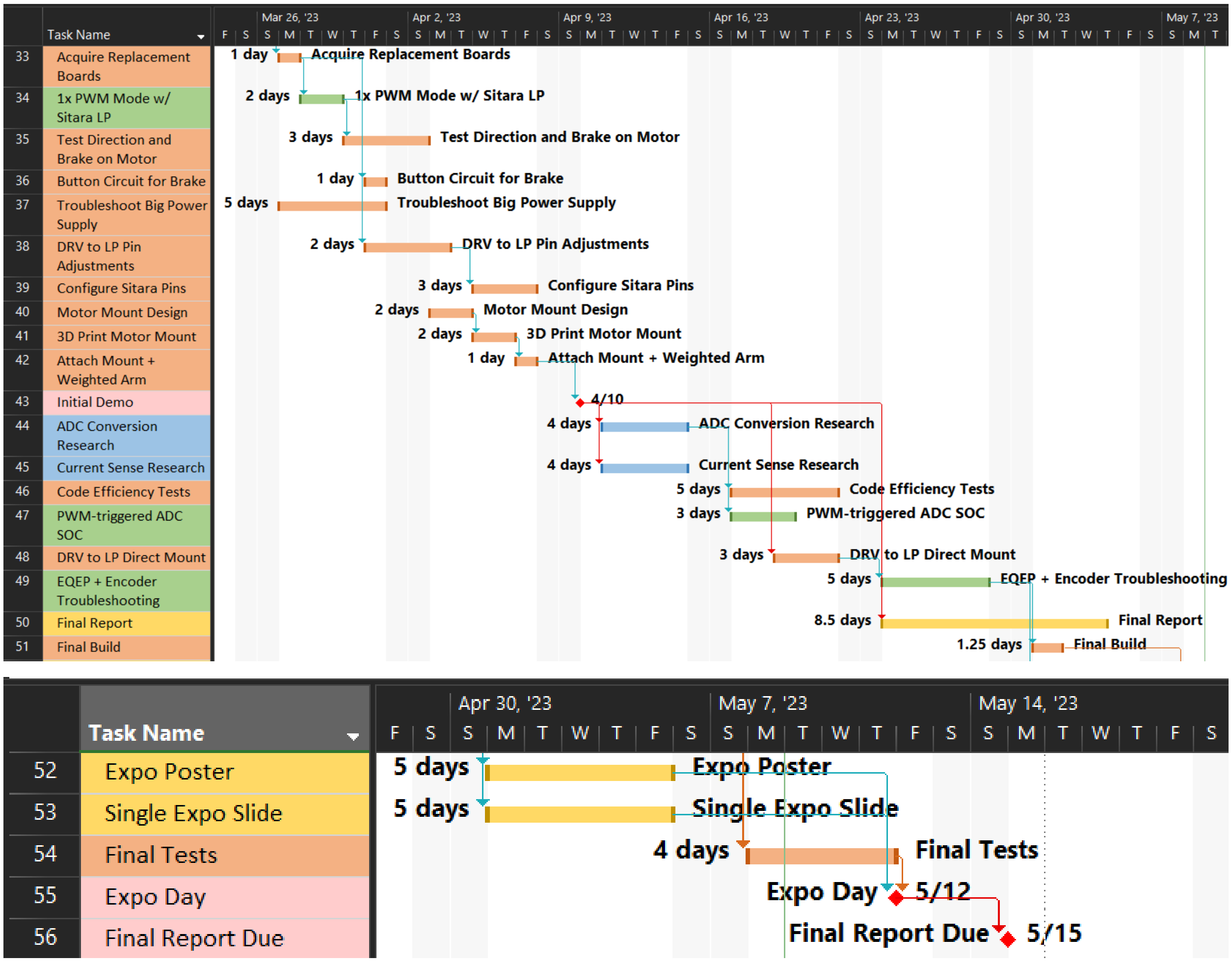


Figure 12.5. Spring 2023 Gantt Chart

12.1 Bibliography

Helpful Diagrams of Motor Controllers and Reference Designs

- DC-Input BLDC Motor
- TIDA-01516 -- Single Microcontroller 18-V/600-W BLDC Motor Control Reference Design With Bluetooth® Low Energy 5.0
  - Microcontroller interconnect
  - Diagrams and schematics
- TIDA-00643—4.4 V to 30 V, 15 A, High Performance Brushless DC Drone Propeller Controller Reference Design
  - Drone ESC Design

Helpful Youtube Links

- ESC Youtube Videos

UTDesign Student Resources

- UTDesign Student Resources
- UTDesign Forms (Reimbursement, expenses, payments)

Links from TI

- TI All Motor Drivers
- TI BLDC Motor Drivers Page
- TI Precision Labs Videos
- LP-AM263 Page
- LP-AM243 Page
- AM263 MCU+ Academy
- AM263x MCU+ SDK
- ARM Based Microcontrollers page
- DRV8316REVM

Brushless Motors

- BLDC and ESC Design Video
  - Kinda long but is a pretty good video
- How BLDC Motors and ESCs Work

Altium

- 1.5 Hour Altium Tutorial (YouTube)

Links from Randy

- ODrivePro Robot Motor Driver
- NearZero2 Fine BLDC Controller
- Robotic Actuator Video
- 600 Watt, 3d-printed, Halbach Array, Brushless DC Electric Motor : 10 Steps (with Pictures) - Instructables

Encoders and Sensors

- What is Motor Commutation
- Motor Encoders
- Motor Resolvers
- EnDat2.2 Position Encoder Protocol
- CUI Devices AMT Encoders

KiCad(Version 6)

- Kicad Documentation
- Kicad Tutorial
- Best Practices for Board Layout of Motor Drivers

12.2 Code

File “demo_freerun.zip” holds the code described in the Figure 5.6 flowchart for the Free-Run demo.

File “demo_position.zip” holds the code described in the Figure 12.1 flowchart for the Positional Control Demo.

Files “demo_interrupt.zip” and “demo_interrupt2.zip” hold the code for verifying proper function of the switch interrupts.

12.3 Design Files

Attached are the two Fusion 360 archive files which contain the two mounts we designed and 3D printed for holding our motor and attaching the metal arm.

The file “Extrusion Arm Connector.f3z” is a shaft coupler that we used to connect a 20x20mm aluminum extrusion to the 8mm in diameter motor shaft.

The file “Motor Mount v13.f3d” is the mount that we used for holding our specific motor in place. This mount was then attached to two 20x20mm aluminum extrusion poles that we connected the box and power supply to. This gave us a clean and organized design that was easy to clamp to a table for stability.