Low Price drives

Capstone final report and User's Manual

04/16/2021

Marzouq Kraidees: mhk44@nau.edu , 646-683-5657

Fatemah Alawadhi: faa275@nau.edu ,928-219-2556

Mubarak Alluwimi: Ma3345@nau.edu , 928-605-2225

Xingyu liu: xl274@nau.edu ,928-266-4327

Client: Venkata Yaramsu: Venkata.yaramasu@nau.edu

GTA: Han Peng: hp263@nau.edu

EE 486C
Team 6
Low Price Drive
04/16/2021

Table of content

Introduction

Client background

Venkata Yaramasu received his Ph.D. degree in electrical engineering from Ryerson University, Toronto, Canada, in 2014. During 2014–2015, he worked as a Postdoctoral Research Fellow at the Laboratory for Electric Drive Applications and Research (LEDAR) and Center for Urban Energy (CUE), Ryerson University. He is currently working as an Assistant Professor of Electrical Engineering in the School of Informatics, Computing, and Cyber Systems (SICCS), Northern Arizona University, Flagstaff, Arizona, USA. His research interests include renewable energy, high-power converters, variable-speed drives, electric vehicles, energy storage, smart grid, and model predictive control.

Dr. Yaramasu worked closely with Rockwell Automation, Toronto Hydro, Hydro One, Natural Sciences and Engineering Research Council of Canada (NSERC), Wind Energy Strategic Network (WESNet), and Connect Canada, and completed 8 industrial projects in Power Electronics, Electric Drives, and Renewable Energy. He is currently working on the Salt River Project (SRP) to analyze the impact of zero net energy homes and electric vehicles on the distribution grids in Arizona and develop fast-charging stations for electric vehicles. During his Ph.D. dissertation research, he developed several novel high-power converters and control schemes for megawatt-level wind energy conversion systems. During 2011-12, he actively participated in NSERC/Rockwell Industrial Research Chair Program and worked closely with Rockwell Automation, Canada on current source converters. During 2009-2012, he also associated with Toronto Hydro and Hydro One to solve urban energy issues by developing novel control techniques for grid-connected converters, flywheel energy storage systems, and active power filters. He has published more than 70 peer-reviewed technical papers including 27 journal papers. He published a book entitled “Model Predictive Control of Wind Energy Conversion Systems” with the Wiley-IEEE Press in 2016. He also authored/co-authored 10 book chapters on Renewable Energy and Model Predictive Control. He has produced 12 technical reports for the power industry.

Dr. Yaramasu received six Best Student Paper Awards and two first prizes in National Level Technical Quiz Competitions during his undergraduate studies in India. He is a recipient of a Best Paper Award from the IET Electric Power Applications in 2020 and a Second Prize Paper Award from the IEEE Journal of Emerging and Selected Topics in Power Electronics (JESTPE) in 2015. During his Ph.D. studies at Ryerson University, he received Best Poster Awards from the Electrical and Computer Engineering (ECE) Department and Faculty of Engineering and Architectural Science (FEAS) in 2013, Student Research Awards from the Toronto Hydro, Hydro One, and Connect Canada in 2010, 2012, and 2013, Research Excellence Awards from the ECE Department in 2012, 2013, and 2014, a Best Poster Award at the NSERC–WESNet Annual Meeting 2010, and a Best Teaching Assistant Award from the FEAS in 2010.

The problem

Our project is to design a new instrumentation platform to perform electric drives class labs. The purpose of this project is to help NAU students in electric drive courses to understand this class better and to help other universities that are not able to buy the dSpace to use our product as an economical solution. NAU students use DSpace to do the labs in the electric drive class now. The major problem for DSpace is the cost. The hardware part costs $4000 per unit and the software part upgrades costing $300 per unit to update. It is too expensive for the university and the students both. Arduino is a cheap solution that can realize the same functions as dSpace. The hardware part of our platform is based on Arduino, and the software part of our platform is based on Matlab/Simulink. Our product can realize the same measuring and controlling functions as the DSpace solution based on only one Arduino Mega, and our product can support Matlab/Simulink workflow.

Design process

Overall design process

Our project topic is “Low-Cost Drive”, so the main problem that we are trying to solve is to come up with a cheaper alternative platform. In this project, we came up with many different solutions to the problems that we explained. Our solution for the cost problems is to use a cheaper platform, which is using an Arduino control board instead of dSPACE. Also, in our project, we verified that we can implement the Arduino platform to the dSPACE based experiments. In addition, to solve the hardware problems, we can use the PCBs to connect all the components together which can be more reliable, cheaper, and easy to fix. Furthermore, we are going to use less space and lower the cost of our project by using smaller and cheaper components in our circuit boards Our project is based on three main subsystems which are Arduino, MATLAB/Simulink, and PCB. Using the Arduino platform is to control the output signals from the circuit boards (PCB) to operate the DC motor, which can be an alternative for the dSPACE hardware. In addition, we use the MATLAB/Simulink platform as the software part of our project, and we use it to control the speed of the DC motor and measure the current. Also, we use the printed circuit boards (PCBs) to connect all the components together. We do this by connecting the inverter board to the Arduino board and DC motor, and from the Arduino board to the Simulink to operate the DC motor.

Functional decomposition

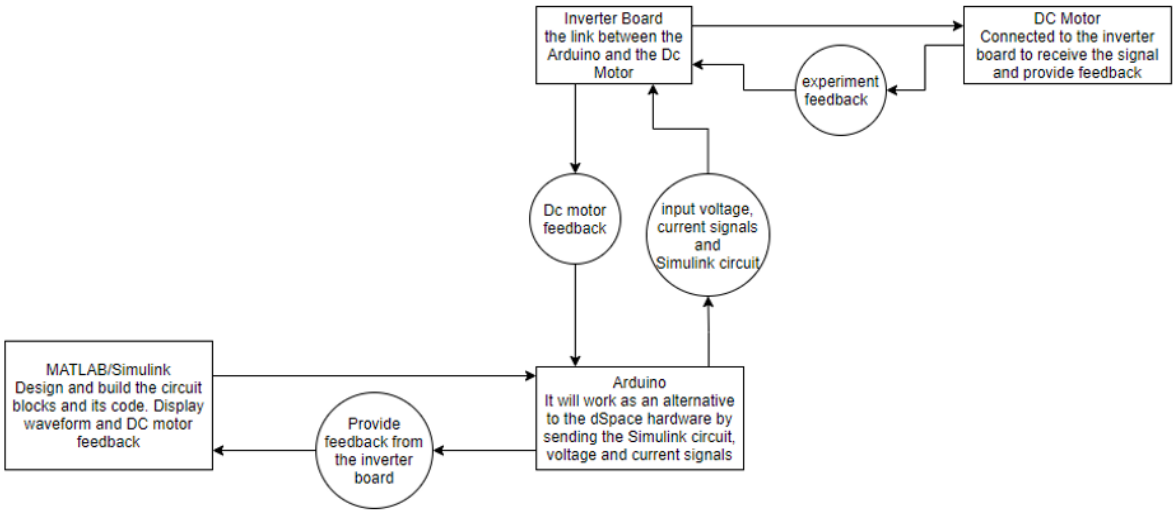


Figure 1: System diagram of the project process.

Prototype

Our prototype consists of three main parts that will be useful for our project. These three prototypes are like puzzle pieces that will give us the final result. The three prototypes used in our project are Arduino, inverter board, and MATLAB Simulink. The team demonstrated the three prototypes and explained how they would be useful to the project. By selecting these three prototypes, the team considered many aspects of the project's final result of the project.

Our team chose these three prototypes because it is going to be important for the final result of our project. The Arduino will be a significant part of implementing the DSPACE experiments; we will use the Arduino to convert DSPACE labs into Arduino Due. The main idea is to find a cheaper method to make these DSPACE experiments work. Using the Arduino, we can convert the experiments by connecting the DR37-M2 with the Arduino, and with that part, we can connect the 37 pin d sub from the inverter to the Arduino. We demonstrated the original project, which uses dSpace instead of the Arduino because our client Dr. Yaramasu's plan for this project uses the Arduino next semester in Spring 2021 with the new design of the inverter board. Also, he did not give us permission to use the Arduino in this path of the project. The dSpace will be replaced with the Arduino next semester. The Arduino will do the exact functions as the dSpace. Therefore, we implemented one of the experiments when we got the results, and the measurements were exactly like our calculations and our expectations. The inverter board is a great part that can be a link between the Arduino and the motor. We are going to use the DC motor in our experiments to run the DSPACE experiments and the Arduino implementation of dSPACE. Using the inverter, we can run the motor and compare both Arduino and DSPACE results by using a converter on the board that can show us both results. MATLAB Simulink is important for our project because we can program everything through it. Using MATLAB Simulink, we can give commands for speed measurement, set PWM pins, and use Arduino packages to convert DSPACE experiments.

These prototypes will be used for our final project's result, which is the big picture, and the team tried to cover many points to make it work. The inverter board will be smaller, and we will make the Arduino fit under the board, and Simulink will be used for programming. Arduino is an alternative to the dSpace DS1104 hardware, where the Simulink circuit will be implemented in the Arduino. The Arduino will be connected to the inverter board, where the inverter board is the link between the Arduino and the DC motor. Signals will be sent from the Arduino to the inverter board and from the inverter board to the DC motor. Therefore, the Arduino will receive the feedback and the output from the inverter board. We connect the Arduino to the MATLAB/Simulink, where the results show as plots and all the measurements.

Arduino will help the project lower the project's cost because the dSpace is costly hardware and software, where the Arduino is the best solution to do the same functions as the dSpace. Moreover, the Arduino will help the students in their labs not to face any difficulties with identifying the Arduino in any computer in the lab room. Also, it will help the student to work in a small place and be more flexible. The inverter board is the link between the dSpace and the DC motor. It provides voltage and current to the DC motor. There are many parts to it, like two-phase inputs, voltage input, and current input. It is understanding the board because the next stage of the project is to shrink the board, making it one-quarter of its current size. The Matlab prototype fits with the design's functional aspects because it helps build the different circuit blocks required in the electric drive. This prototype uses the Mat-lab/ Simulink to model the circuit while still simulating the circuit design. Using MATLAB Simulink, we can give commands for speed measurement, set PWM pins, and use Arduino packages to convert DSPACE experiments. This prototype is expected to learn the whole concept of how to use these prototypes and make them work in order to create the final product. We will use the circuit test function on Matlab Simulink to detect possible errors in the circuit and timely correct and modify them so as to avoid and reduce the risk to the whole project. Simultaneously, these risks are concentrated in the prototype by making multiple computations to ensure that all the voltage and current metrics have been fulfilled.

In the process of completing the experiment, we encountered some significant challenges. In the aspect of the Arduino, the biggest challenge we had in the first place that we couldn't use the Arduino to demonstrate; the challenges we had with the dSpace was making the software control desk of the dSpace work because it takes time, and also sometimes we get an error message which we have to fix the settings and restart the computer. If the prototype Arduino filed, the problem would be in the wire connections or how we implemented an experiment; therefore, we will have to check the connections and check the circuit diagram in MATLAB/Simulink. According to dSpace, sometimes we try to open the control desk software to upload the circuit diagram, but the control desk wouldn't start; the software setting should be identified to the hardware as DS1104. The prototype sometimes takes longer than we expected because of the computer and the software of the dSpace however, for the Arduino, at first, it would take time to connect it with the DR37-M2 connectors and make sure all the wire connections were correct. Corresponding to the Arduino, we would work on it ahead of time to get more familiar with the Arduino part and know how to connect the Arduino with the MATLAB/Simulink to control the DC motor so that it won't be a high risk for the next semester. This prototype's results will affect the project because it is one of the significant prototypes we develop in the project. It is the only solution we chose to help the project be a low cost instead of 4000$ for the hardware. It will put the project at high risk if the Arduino did not work and the results were incorrect. We would fail the project and start to find other solutions to replace the Arduino with the same price or lower. The Arduino is much cheaper and more friendly for the student to use.

In the MATLAB part, the major challenge was to make the Simulink block diagrams work because it takes some time to learn the whole concept and make the Adriano packages work. Therefore, there was a challenge when integrating the circuit with the MATLAB simulator. If there was a failure in the prototype, the PWM signal could not have been simulated; hence, the motor's speed would be difficult to find. The prototype took about 20 hours to complete. We expected it to take such a long time that all the Arduino measurements had to be taken using the MATLAB/ Simulink. Taking these measurements depended on computational simulations that take time. We think the prototype design was done correctly. If we would do it again, we would have learned about MATLAB Simulink better because it will be easier to do all the experiments and convert them into Arduino. The prototype took some time because we were learning the concept of converting DSPACE to Arduino experiments, so we had to do two experiments and get the results, and we expected it to take some time. The projects' perception is going to be great because we spent a lot of time learning difficult material, and once we learn everything, we will do great in the future. Our concern with the prototype that might negatively impact the project plan is MATLAB Simulink errors; other than that, there are no concerns.

The three prototypes are Arduino, inverter board, and MATLAB Simulink. These prototypes will help us get the most out of the project. The main question we've been asking ourselves is, what is the most valuable thing that will fit this project into the future. During our meeting, we discussed including a DC motor in our prototype, but we agreed that MATLAB

Simulink would be a better choice because Simulink is associated with a DC motor, and it has more options. The team integrated our ideas well and came up with the three prototypes.

Final Design

System Architecture

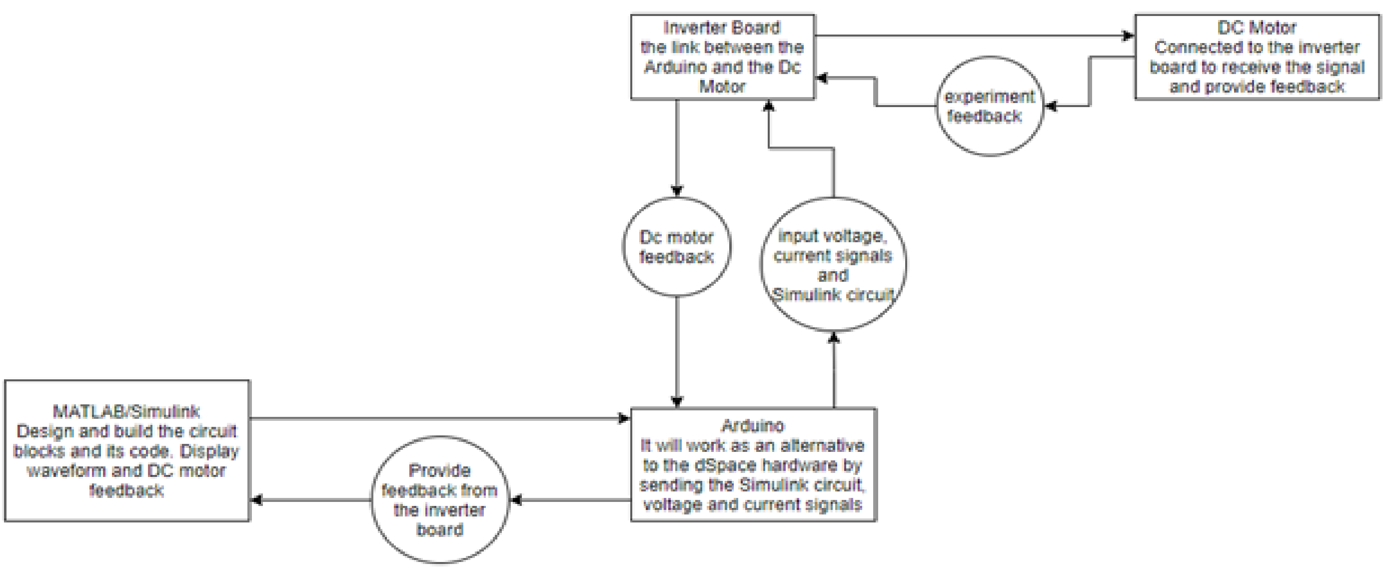


Figure 2: System Architecture

The system architecture shows the subsystems of the project. Subsystems are a breakdown and a full understanding of the project. It allows us to separate the project between members of the team, so it makes the prototyping phase easier. We created our system architecture using all the subsystems that we identified. It is like the system diagram that is shown before this figure. We have four main subsystems which are the Arduino, DC motor, MATLAB/Simulink, and the inverter board.

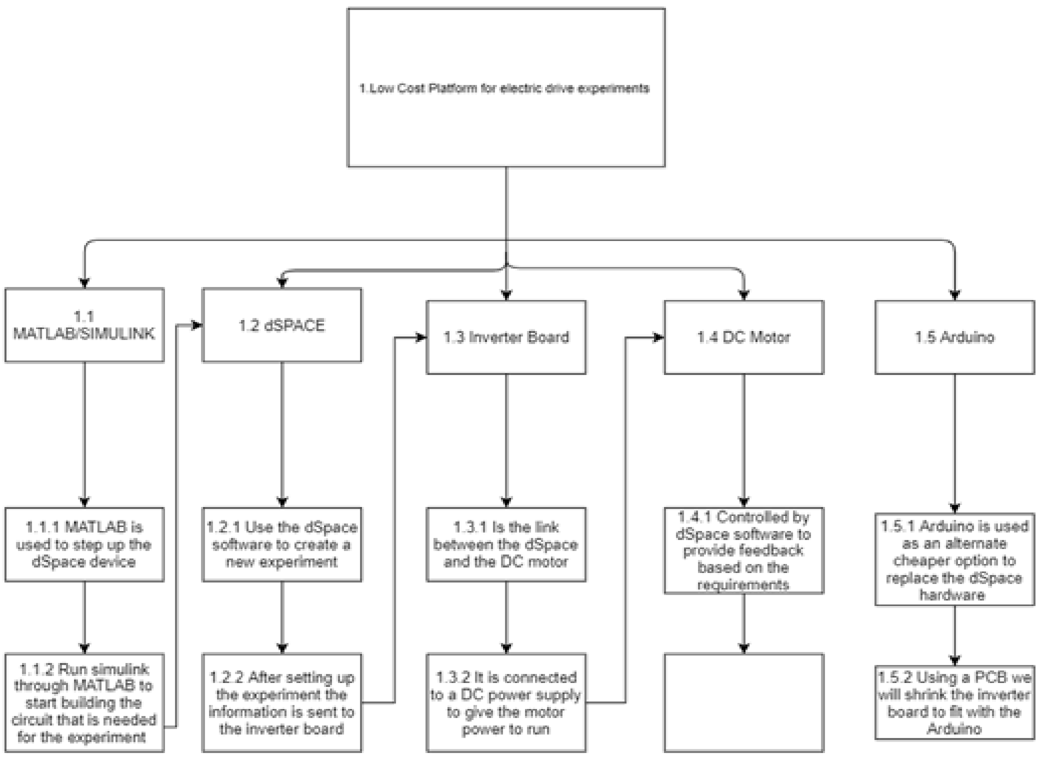


Figure 3: System diagram

This system diagram shown above demonstrates how the equipment works and how the labs are supposed to run. First, the dSpace device is set up by MATLAB. Then we use MATLAB/Simulink to generate the circuit and code. The circuit and code that are generated by using MATLAB/Simulink are sent to the dSpace hardware where it uses its software to create a new experiment interface. After the creation of the interface, the experiment is sent to the inverter board. The inverter board is connected to the DC motor which provides voltage and current to the motor. The motor will provide feedback based on the lab requirements it might be current or voltage. This the basic system diagram of how the project works. We will be replacing the dSpace with the Arduino.

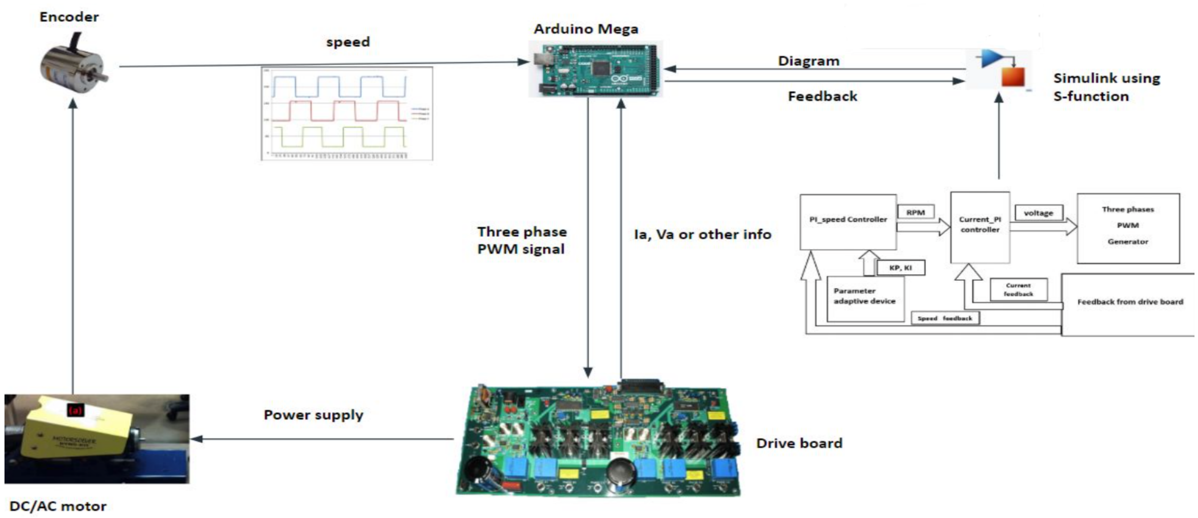


Figure 4: Diagram of the final connections between the Hardware in the lab.

Major components:

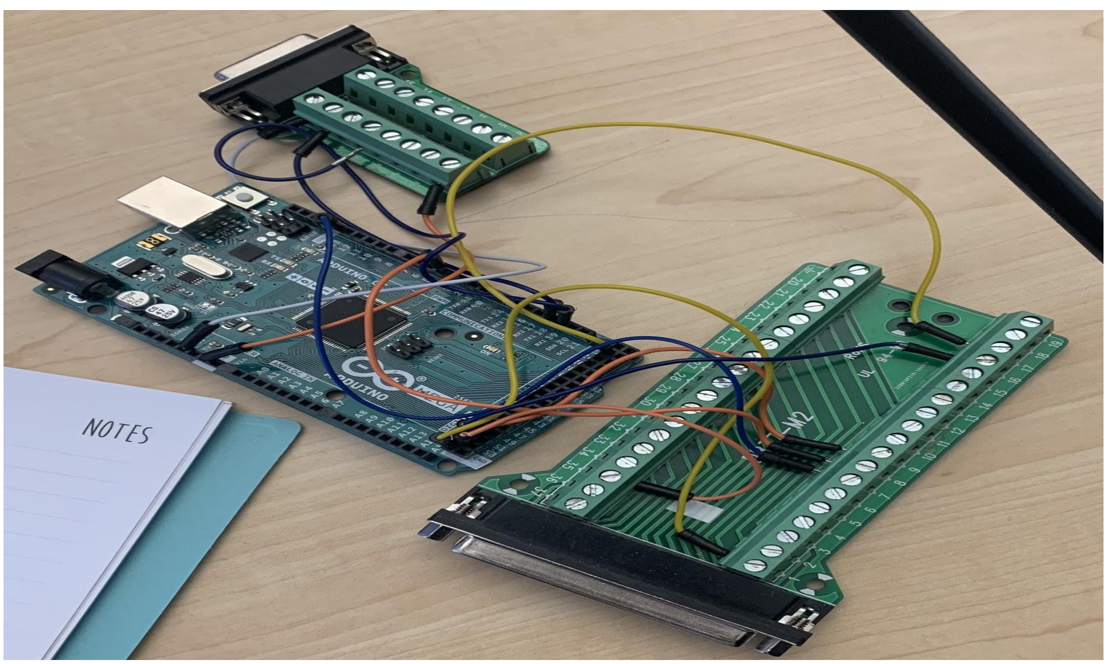


Figure 5: Arduino MEGA 2560

The first component is the Arduino. The Arduino is going to be a vital component of the project. It will be replacing the dSpace hardware and software. It is a considerably cheaper alternative to the dSpace. It is also a user-friendly device with many different capabilities when programmed correctly. The labs will be converted from dSpace to Arduino usage using MATLAB. The Arduino is going to send the input into the inverter board and receive the output feedback from it. It will have a 37 pin d sub to connect it to the inverter. Like the dSpace, the Arduino is going to be the receiving input from the MATLAB/Simulink generated circuit and code. Although we must program the Arduino using MATLAB to make it do the functionalities of the dSpace. The programming using MATLAB is going to help us replace the dSpace software. Arduino program packages are available in MATLAB to help us with programming it to implement the labs. Each lab requires a different output and our goal is to program the Arduino to provide that output. We will be using the DC and analog pins on the Arduino to perform the lab experiments.

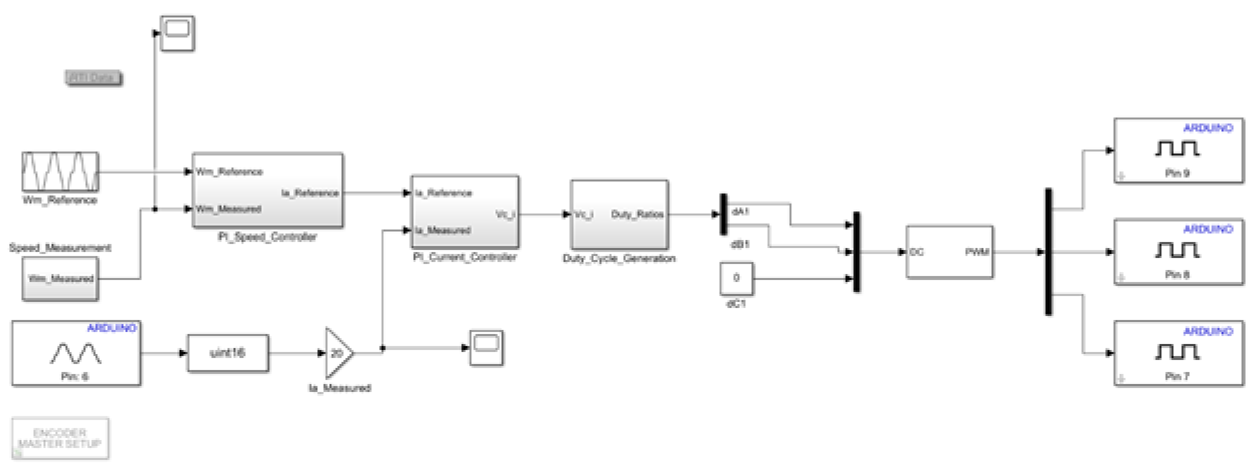


Figure 6: MATLAB/Simulink. Example of the lab 1 circuit diagram.

The second component is Matlab/Simulink. The Matlab/Simulink has three functions in this project. First is building up the code for each circuit design experiment so it can be implemented into the dSPACE software control desk or create the Arduino. Also, after replacing the dSPACE with the Arduino, the motor's speed and waveforms' measurements will be displayed by the Matlab program and instead of the control desk software. Therefore, Matlab/Simulink will control the Arduino by converting the Arduino code and performing the labs. Matlab/Simulink will be the only software used in the project, which will help the project be cheaper than before and easier for the students. Thus, the software part of our platform is based on Matlab/Simulink.

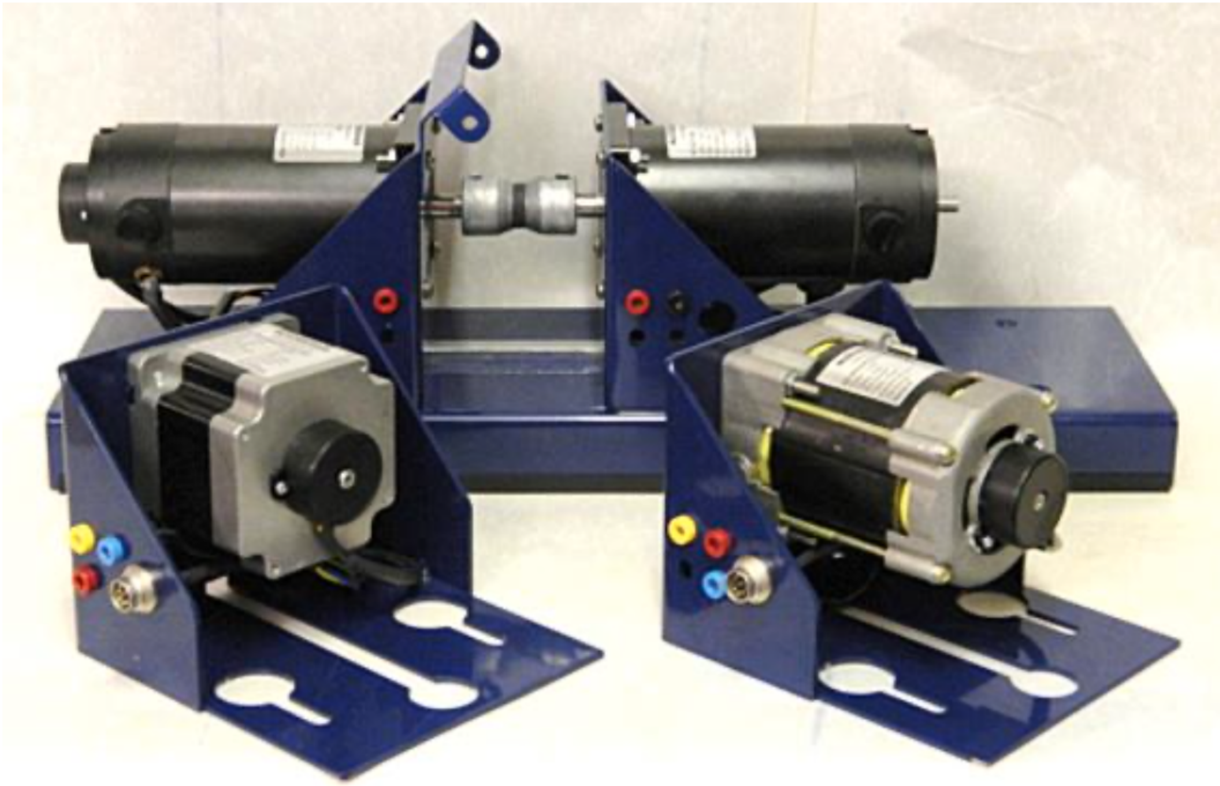


Figure 7: The Motorsolver kit

The third component is Motorsolver. The Motorsolver kit has a DC motor, DC generator, DC brushless motor, AC induction motor from the kit, and an encoder that most machines fit in. The motor is 250 watt, where it requires a 40 to 48 V supply. Therefore, the generator is fixed on the base and easy to test machines, and the DC motor and DC generator are removable and can be replaced easily. The DC motor in this project will be connected to the inverter board. Therefore, the DC motor will receive the signals from the Arduino through the inverter board and send back the feedback to the Arduino from the link between the Arduino and the DC Motor.

EE 486C
Team 6
Low Price Drive
04/16/2021

Results

After designing our project we started assembling it. We started working on different parts of the project. We started testing our project. We started by identifying the requirements for our project. After that, we used our requirements that we identified to start our testing process. We choose four parts of our project to do our testing. We ended up having three major tests.

Instructions: List all of your requirements, and use a numbering system.

Type of Test	Status	Req #	Requirement
		1	Arduino integration
		1.1	Establish the connection between the arduino and Computer.*
Integrate		1.1.1	Connect the Arduino into the inverter board and the DC Generator encoder.
Inspect		1.1.2	Get feedback from the encoder and display the results on Simulink. *
UTM		1.2	Display the current and voltage from the inverter board connected to the Arduino.
Inspect		1.2.1	Required the DB37-M2 connector to connect the Arduino in the inverter board and firgure out the PWM pins in both inverter board and DB37 connector. *
		1.2.2	Place an order for the inverter board component for next year group because it's a continues project for next group.
UTS		1.3	Display the required units based on what the lab is asking for i.e Voltage,Speed, and current
		1.3.1	Compare the results from the Arduino with the dSPACE results we got from last semester.

Figure 8: The requirement spreadsheet shows the requirement testing results.

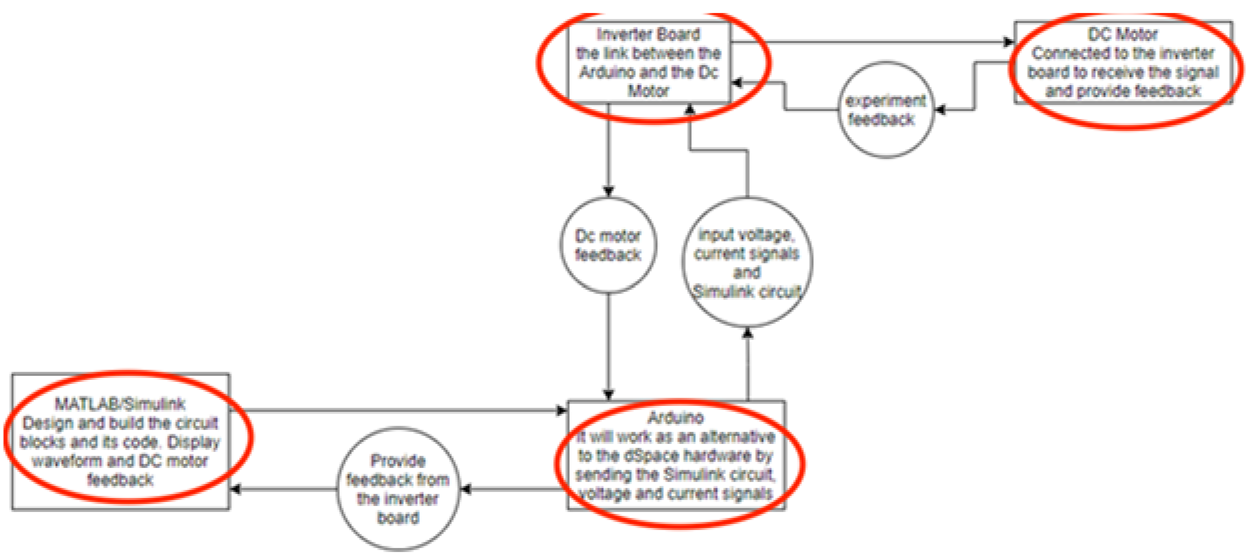


Figure 9: System architecture. The red circles represent the parts tested.

EE 486C
Team 6
Low Price Drive
04/16/2021

On our excel spreadsheet, we chose three requirements that we deemed important to our client. The first requirement is establishing a connection between the Arduino and the computer which is the MATLAB/SIMULINK software. It is a vital part of our project because not being able to connect the Arduino to the computer means that we can perform the experiments. the labs are performed using the SIMULINK software for circuit design and code generation. Our client made it clear to us that we must perform the labs using the Arduino and the SIMULINK software. failing to meet that requirement means that we cannot move on with our project and start implementing the labs. The second requirement is to connect the Arduino to the dc motor encoder. This is also an important step in our project because it will provide us with feedback from the motor showing the encoder's position and the speed of the motor. Most of the labs require feedback from the dc motor's encoder hence why this an essential part of our project. like the first requirement, our client urged us to focus on this requirement for it being such a huge part of our project. The final requirement that we have identified is connecting the Arduino to the inverter board using a DB37-M2 connector also figuring out the pulse-width modulation pins in the inverter board so that we can use connector pins and attach wires to them connecting it to the Arduino. It is important for the Arduino to be connected to the inverter board because it will enable it to send PWM and digital signals to the board. Those signals are the signals that should control the DC motor. Going back to the two previous requirements, our client stressed the importance of connecting the inverter board to the Arduino for us to be able to finish all the labs. failing to meet these requirements will result in the failure of completing this project.

The first type of test we wanted to perform a step-by-step unit test. The parts that are involved in this test are the dc motor, Arduino, and the inverter board. For this test, we wanted to input different voltages from the Arduino to the inverter board so that we can change the speed of the motor. The second type of test is the integration test. The part that is involved in the Arduino. We want to connect the Arduino to the inverter board, and the dc motor encoder. The third type of test is the matrix unit test. The parts that are involved are the Arduino and the computer. We wanted to establish a connection between the Arduino and the computer software SIMULINK.

The first major test we did was we wanted to establish a connection between the Arduino and Simulink. We start with a few simple circuits to gain an understanding of how to create a circuit on Simulink and generate its code to be sent to the Arduino. We started with turning on a single LED light using the Arduino and then we moved on to turn on the LED light by using a button. Both tests were successful. After completing those simple tests, we wanted to see how we can build a circuit to the PWM signals from the Arduino. The PWM signals are important for our project because most labs contain them. For us to control the speed of the motor we must be sending PWM signals to the inverter board so that it can deliver them to the dc motor. On the inverter board there are three phases that control the motor, phase A, B, and C. for our testing we connected phase A to Arduino digital pin 9 phase B to pin 8, and phase C to pin 7. All of them are set to be outputs so that we can connect them to the oscilloscope to see if the signals are being sent. We did 6 total tests. For each test, we have a different PWM signal that is being sent to the digital outputs and the oscilloscope should display a digital graph showing the PWM signal. All our 6 tests passed, and we established the PWM part of our labs.

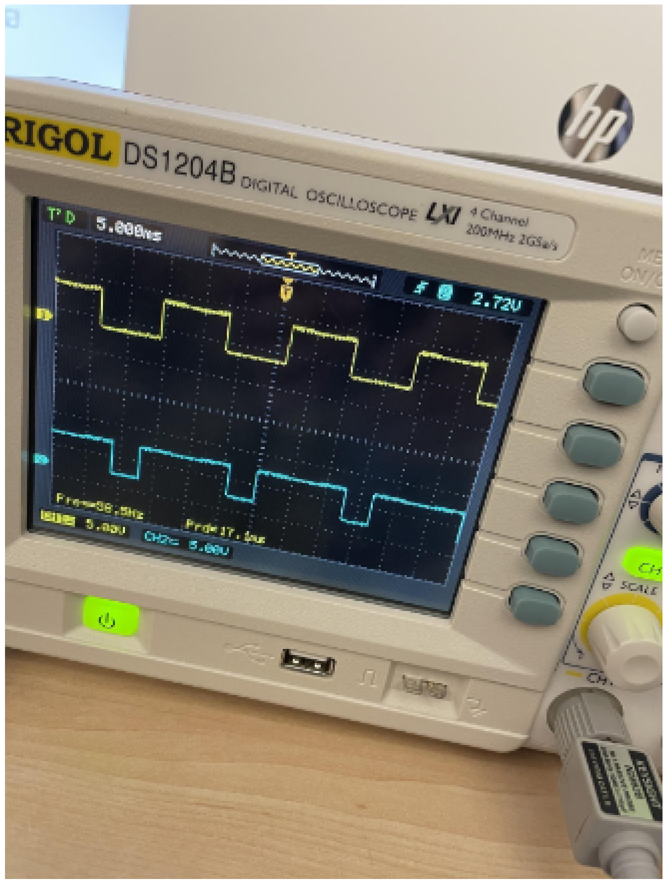


Figure 10: One of the test results that we ran. Phase

A is 0.5V and Phase B is 0.25V and Phase C is 0

The second major test that we did was connecting the Arduino with the inverter board and the dc motor encoder. The first connection we wanted to make is connecting the Arduino to the inverter board. We used a DB37-M2 connector to establish that connection. We started off by figuring out where each PWM pin on the inverter board can be found. Searching the internet for the user manual of the inverter board we successfully found and connected the pins on the DB37-M2 connectors. Then we connected the encoder to the Arduino using a DB15-M2 connector. Like the one before we had to look up its user manual so that we can identify each wire function. We successfully completed the connection. Finally, we had to connect a PNC cable from the board to the Arduino so that we can get a measured current of the DC motor. We failed to do so and that derailed our project progress.

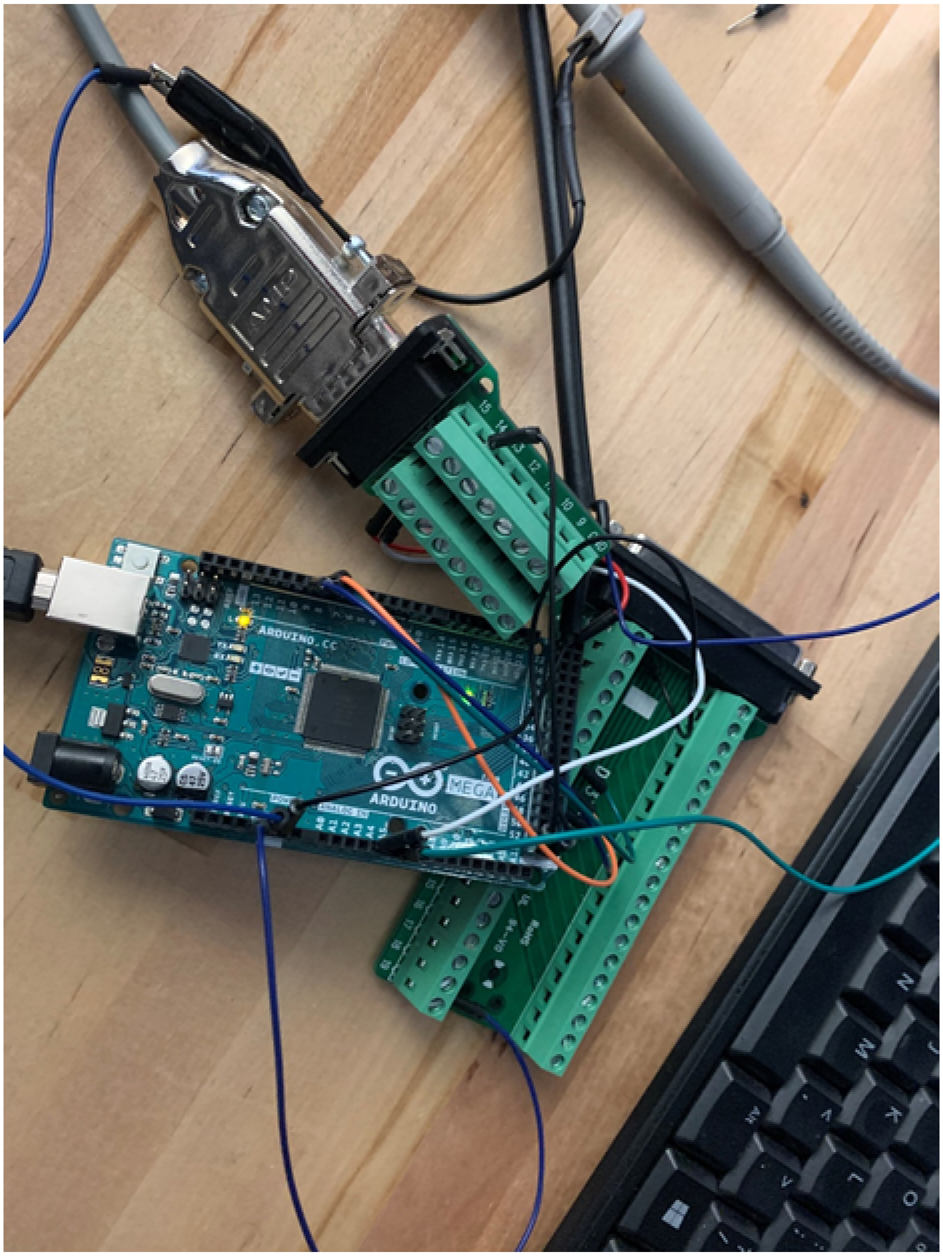


Figure 11: The two connectors connected to the Arduino with the correct wire from the user manuals.

The last major test we tried to do is the input of different voltages in the inverter board to change the speed of the DC motor. A few of the labs require us to change the speed of the motor while it is running. It asks us to do so by sending in different voltage inputs into the board and with that the speed will increase or decrease depending on the voltage being sent in. We did not finish the prerequisites for this test, so we failed to attempt it. We did not have enough time to reach that level in our project so we could not perform the tests. You can refer to appendix iii for the flow chart for this test.

After performing these tests the results went as we expected them to go. We were unable to complete the entire project but the tests that we performed were successful and up to expectations. We met most of our requirements but not all of them. From what we built the project performed well under the test conditions.

Conclusion

In this project, we designed a low-cost instrumentation platform to perform electric drives class labs. Since this project was a low-cost solution therefore it helped students to perform electric drive labs using more economical products. To perform the electric drive labs, usually dSpace is used but since the hardware is very costly i.e., 4000$ per unit and 300$ to update, and not every university can afford it. To overcome this issue, an Arduino based solution was proposed in this project using which the electric drives labs can be performed. The hardware part of our platform was based on Arduino, and the software part of our platform was based on Matlab/Simulink. Using this platform, we redesigned and performed a total of 8 labs. It is observed that the Arduino can effectively perform these labs, hence it can be an alternate choice to replace the dSpace. The DC motor was controlled by the Arduino and using its encoder, the feedback was received to display on the computer screen. The circuit of the lab 1 was redesigned and performed using the proposed framework. However, due to the shortage of time, we did not rectify and troubleshoot all the errors. Furthermore, some more time was required to understand all the working blocks of the Arduino and the interconnections between them. Therefore, during the testing of the lab 1, we faced some errors related to these which needed to be fixed. Finally, it can be concluded that using this low-cost solution, the electric drives labs now is more accessible for students and universities.

EE 486C
Team 6
Low Price Drive
04/16/2021

User Manual

Introduction

We are pleased that you have chosen Low Price drives for your business needs. There is a strong need for an Electrical Drives Instrumentation Platform because current instrumentation is too expensive. We provided for you here a powerful and cheap instrumentation platform to help you to teach electrical drives course in a better way. Besides, that has been custom-designed to meet your needs. Some of the key highlights include:
1.Replace expensive dSpaces;
2.Support Arduino-based Matlab /Simulink workflows;
3.Low cost;
4.As small as a possible model;
5.Easy to reproduce.

The purpose of this user manual successfully uses and maintains the Electrical Drives Instrumentation Platform in your actual business context going forward. Our aim is to make sure that you are able to get the profit and the benefit of our product for many years to come!

The idea of our project is to create a low-cost platform for electric drive experimentations. Our client Dr. Yaramasu is currently teaching an electric drives course and in this course, they use an expensive platform called ‘dSPACE’ to perform lab experiments and these hardware labs are quite expensive which cost around $4000. Our objective is to get familiar with these experiments and learn about them and convert these experiments using a cheaper alternative, so students can access these labs everywhere and whenever they want. The Arduino board is a great alternative and we will use it to implement the dSPACE experiment into the board with the help of MATLAB Simulink software. Simulink software is the main design program for this project and we will convert the provided dSPACE blocks into a design that works with the Arduino, installing the Arduino support package will help us with this process. The final model will have the complete setup using the PCB on which all the boards will connect together in a single chip and the Arduino will link with the PCB where the current and voltage controllers will present as well.

Our product has four main subsystems. The first subsystem is hardware synthesis and software design. We designed the Arduino code to perform experiments that could do the same thing as dSpace, and then converted the Arduino code into Simulink blocks because our product needed to support MATLAB /Simulink. The second subsystem is hardware connectivity and software testing. We hooked up all these conversion cables and wires that we had to buy because they allowed us to connect each needle to the Arduino, and we tested our code to see if it met all the lab requirements. The third subsystem is to execute experiments. After we build our product, we use our product to check if there are any differences between the lab concept and our project design. We check whether people can use our products to successfully implement electrical drive classes in the laboratory. The final subsystem is the PCB design. Because there are many unnecessary parts in the current inverter board, we must first design a new inverter board. However, due to lack of time, we use Dr. Yaramasu's PCB board.

Our products meet the needs of our customers and we have created a skilled solution to solve the problems of the electrical drive class laboratory. We built our project at a low cost. We do our projects on a per-lab basis, which means we test feasibility at each lab. We use our own products to complete each part of the lab before we release the final product to the public. Because our products can be successfully implemented in every part of the laboratory, our products will be available to the public. We also compared our solution to the dSpace based solution to evaluate and analyze our results. In addition, our product still has potential because it is an open-source design.

Installation

1. Hardware components

The basic components of an electric drive system shown in figure 12. The input command shown in the figure is the process and the calculations that are developed by the software MATLAB/Simulink. HiRel / Vishay electric drives inverter board #75771 functions as the power processing unit (PPU), and it contains sensors and transistors. The most important part is Arduino MEGA 2560 which will be alternate for the dSPACE which will be the controller and implementation on your MATLAB/Simulink design. The Motorsolver Dyno-Kit will be the motor and load, which means some of the motors have encoders to determine the motor's speed and direction.

To work on the lab you have to find a computer that contains the dSPACE PC board (DS1104) and collect all the components from the lab cabinet as shown in figure 13. Place the items near your computer but don't connect them together yet. Check the motor you have labeled as "DC Generator" and the other motor is the "DC Motor".

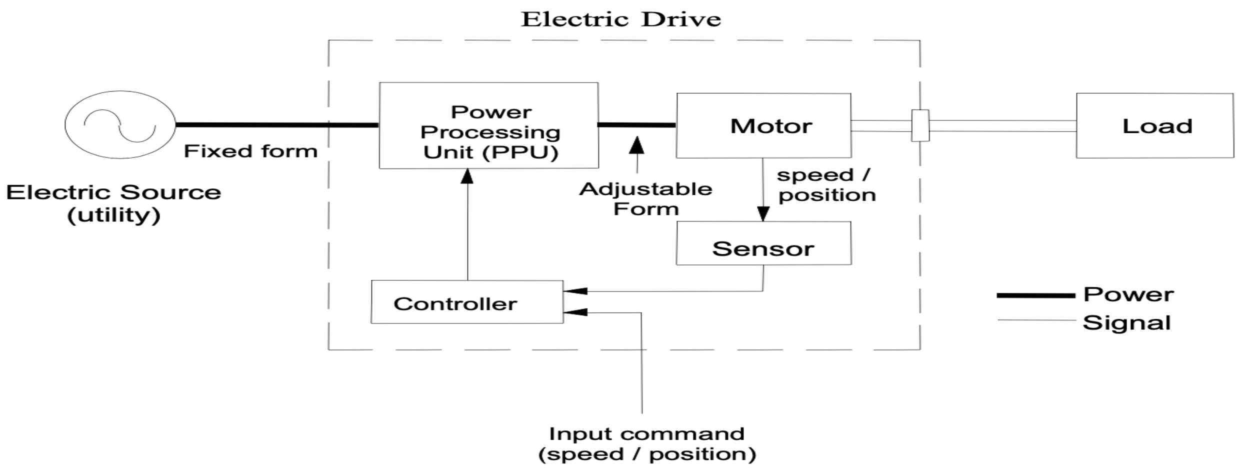


Figure 12: Basic components of an electric drive system

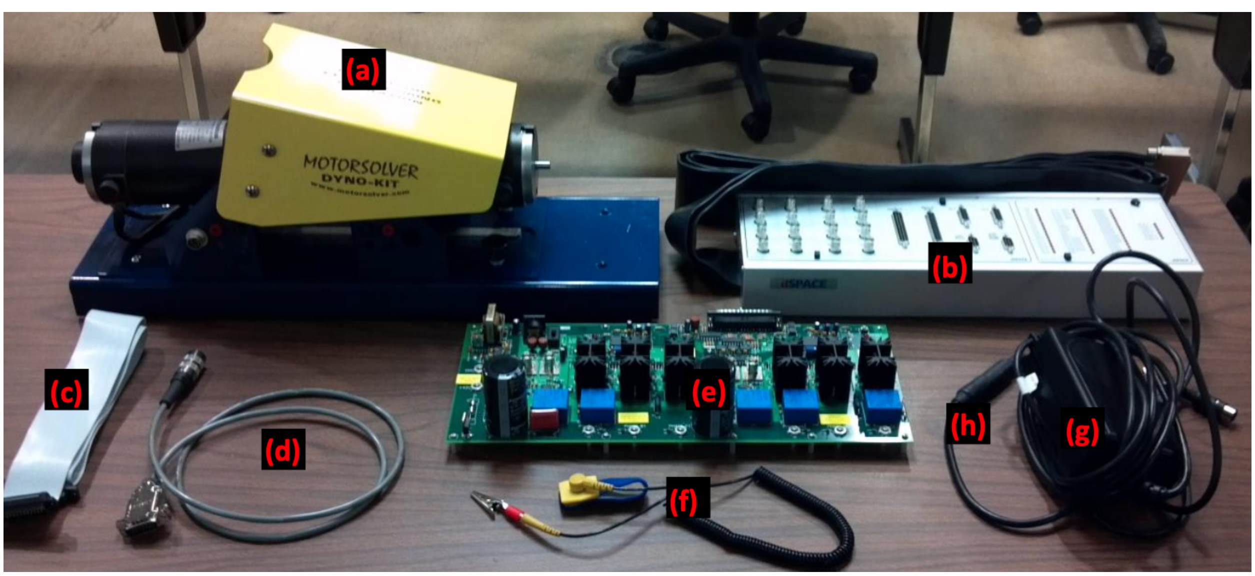


Figure 13: items of the electric drives lab in the lab's cabinet. (a) MotorSolver Dyno-Kit, (b) dSPACE CP1104 connector / LED panel, (c) 37-pin DSUB cable, (d) 8-pin encoder cable, (e) HiRel / Vishay electric drives inverter board #75771, (f) grounding wrist strap, and (g) +/- 12V analog power supply connected to (h) a pin conversion cable.

1.2 Hardware connections

Before connecting any component, use the high-power DC power supply you will find on your workstation, set its DC output to 40 V, and then turn off the power supply. One of the team members must be grounded. The grounded team member may use the banana cables to connect the + port on the DC power supply to the +42V port on the left side of the inverter board. Next, connect the DC power supply's '-' port to the GND connection at the left side of the inverter board. Do NOT connect the ground port of the DC power supply to the inverter board. Do NOT turn on the DC power supply at this time.

Connect the +/- 12 V analog power supply to the inverter board's analog power connection. Then turn on the analog power toggle switch. If the green LED next to the toggle switch becomes lit, means the input analog power is working. If the LED does not turn double-check the connections. If not the connection turns off the toggle switch and checks the fuses located below the -/+ 12V connection. Remove the fuses carefully and measure the resistance by using the multimeter. Notify your TA if you need a new fuse.

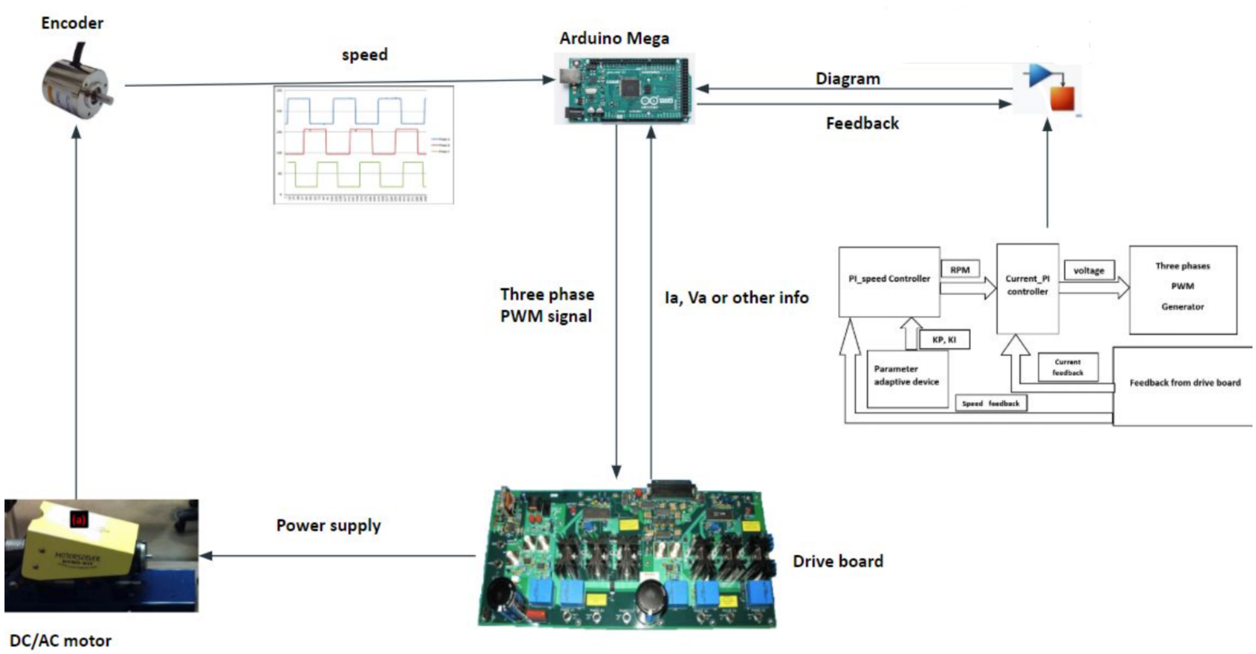


Figure 14: Diagram shows the hardware connections for the lab.

Therefore, the grounded team member should connect the DSUB 37-pin connector between the DB 37 connector and the 37- pin port in the inverter board. Then connect the pin 6,7,8,9 on the connector to the Arduino Digital pin 7,8,9,10 respectively.

Connect the encoder cable 15-pin to the DB15 connector to the MotorSolver Kit’s encoder. Using the 15-pin connector to connect between the Motot’s encoder and the 8- pin encoder cable. Using pin 1,2,14 to connect the Arduino pin 5V, 4, GND. The encoder will provide the controller the location and speed of the motor. Then the grounded team member should bring the BNC-BNC cable located on the side of the lab cabinet to connect the CP1104 analog to digital ADC-5 port to the inverter board’s CURR A1 port. The inverter board phase A1 is the current sensor that will give information about the current being drawn by the motor.

The connections to the phase A1 and phase B1 voltages at the top of the inverter board. These voltages will power the motor’s drive. Therefore, the grounded team member may connect phase A1 output voltage on the inverter board to the black connector on the DC Generator which will be used as the motor instead of a generator. And connect phase B1 to the red connector in the DC Generator. (see the Final connection will be as shown in appendix.)

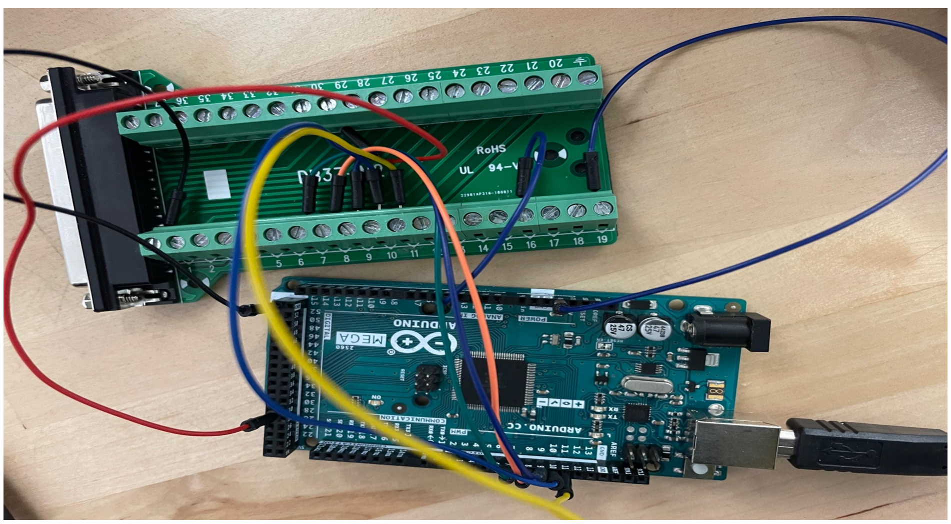


Figure 15: The connection between Arduino and DB 37 female connector.

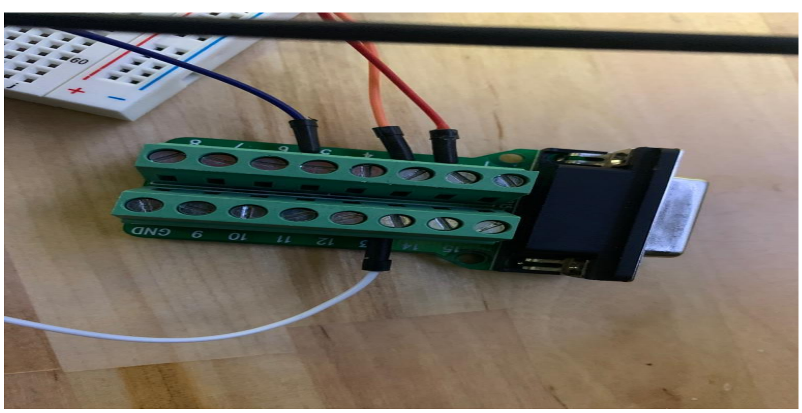


Figure 16: Connect the DB 15 female connector with the Arduino.

2. Configuration and use

Our project had multiple hardware and software parrots to it. This section will be about how to configure and use the lab material. The example for this section is lab 1. The first part is about the MATLAB and SIMULINK circuit building and code generation. For this lab, electric drive control algorithms will be modeled in MATLAB Simulink and then executed on the Arduino. The MATLAB Simulink model is provided for this lab. Create a folder called Exp1 on the lab computer's desktop, and download the file "Exp1.mdl". Double-click on the downloaded file, and it should automatically open the 32-bit version of Matlab Simulink. If not, find and start the 32-bit version using the Search capability of theWindows Start menu. Set MATLAB's current/working directory to your Exp1 folder, as displayed and set in the upper-left region of the MATLAB command window.

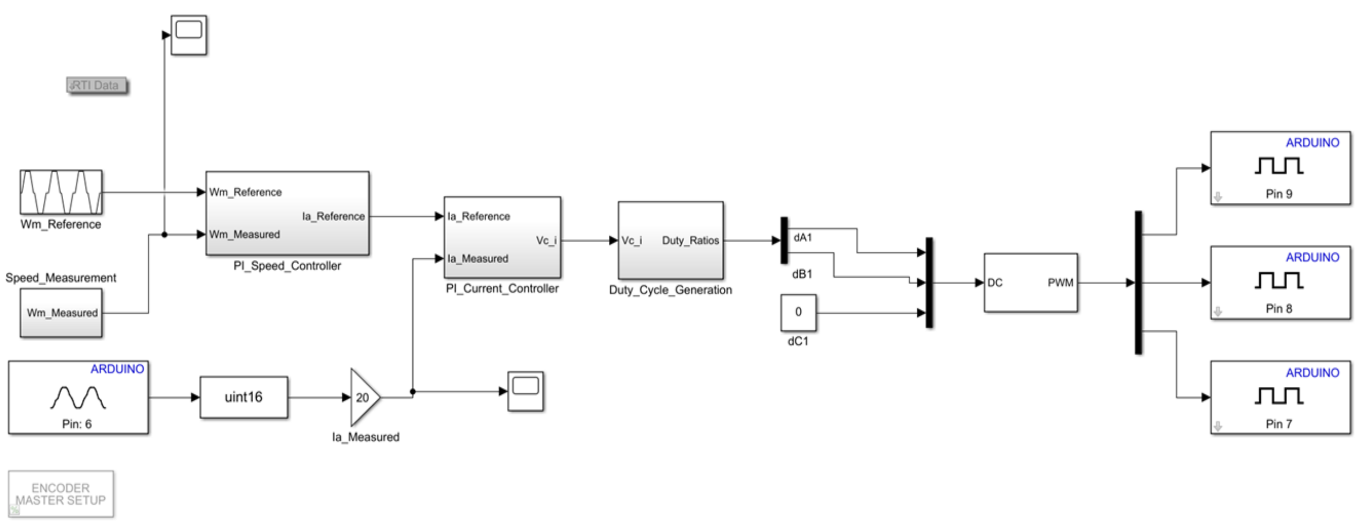


Figure 17: Lab 1 circuit assembled on Simulink

Analyze the MATLAB Simulink model of the control system. Double-click on the"Wm_Reference" block in MATLAB Simulink to find the maximum and minimum values of the reference or desired motor speed in this case, in radians per second. The "Speed_Measurement"block estimates the motor's speed, using data from the MotorSolver kit's encoder. The motor's measured speed is then compared to its desired speed, and a proportional-integral (PI) speed controller uses the error between these two values to calculate an updated value for the motor's armature current, Ia. Ensure that the Simulink window (the one with the block diagram) is active, and enter Ctrl-B to generate C code that will implement the modeled control system in real-time. After several seconds, MATLAB's command-line window should pop up and display informative messages as the C code is generated, compiled, and linked. For the hardware part of the configuration connect the encoder cable 15-pin to the DB15 connector to the MotorSolver Kit's encoder. Using the 15-pin connector to connect between the Motot's encoder and the 8- pin encoder cable. Secondly, a grounded member of the team should connect the DSUB 37-pin connector between the DB 37 connector and the 37- pin port in the inverter board. After setting up both the software and hardware parts of the lab connect the inverter board to the DC voltage supply. Start running the lab and the results will show on the computer screen.

Maintenance

For this project, there is a couple of things that are needed to maintain efficiency and for it to properly work without any issues:

- When transporting the hardware be careful with them and not drop them hardware so that it won't be comprised to any damage.
- Make sure all wire connections from the Arduino to the motor's encoder and the inverter board are connected without having any loose wire connections to get the best results.
- When putting away the hardware connections put them away with care or else the hardware connections will be loose or disconnected.
- Be sure to use the correct Matlab "Simulink" software folder for the lab that you are performing.
- Must be using the latest MATLAB version.
- Store the hardware in a safe place to avoid damage to the hardware.

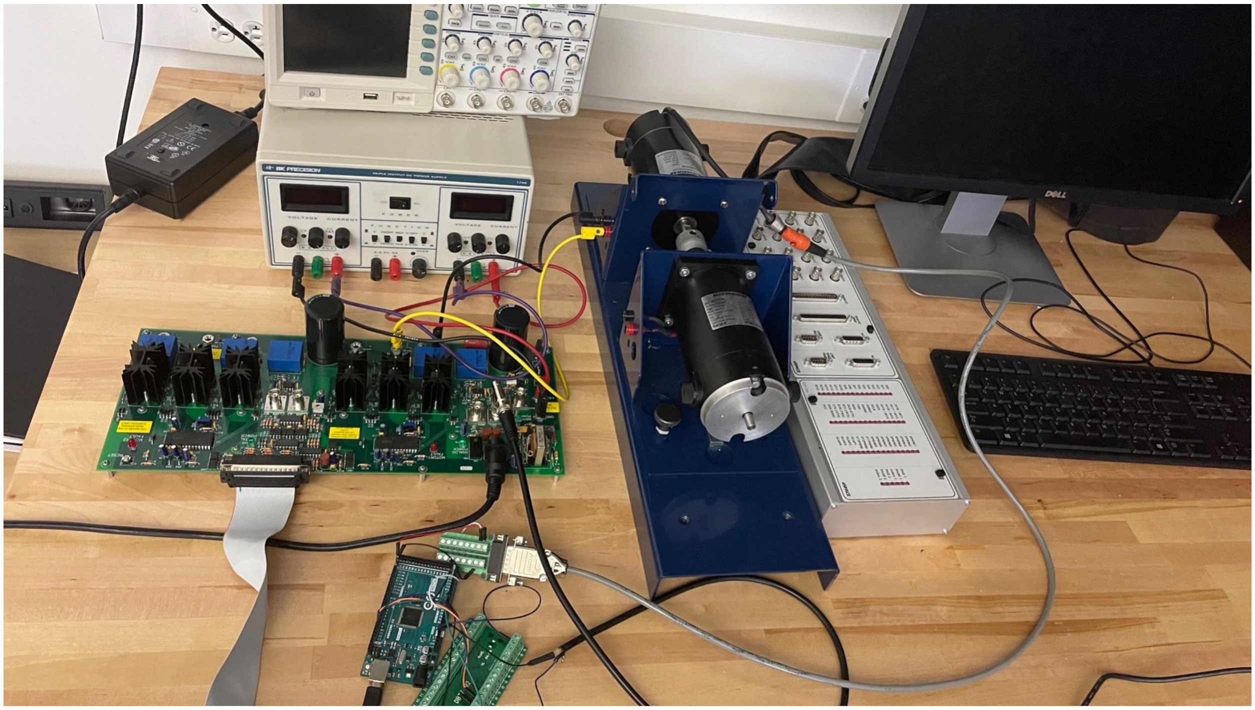
Troubleshooting operation

The goal of this project as stated earlier is to replace the dSpace with the Arduino. With this is in mind the troubleshooting process is very different when it comes to the Arduino. There are many wires and parts that may easily be affected by different circumstances compared to the dSpace. This project includes a lot of wiring which means one loose wire derails the entire experiment. One way of troubleshooting is to connect the wires to an oscilloscope to check each wire is receiving any signals. Before connecting the hardware to each other check the wire connections on the Arduino to have an error-free experiment. Any loose wire can affect the results and may cause mistakes on the lab report due to wire. So it is important to always make sure all the connections are correct and firmly connected.

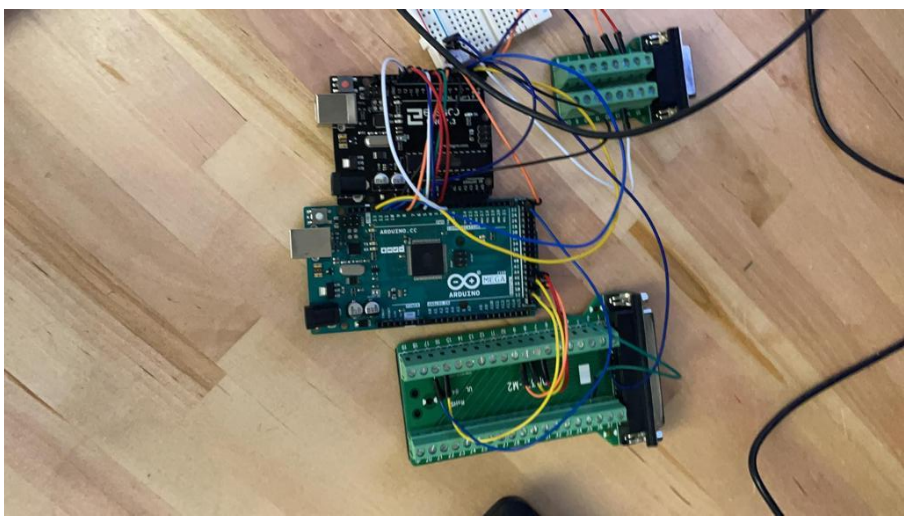
Conclusion

In this user manual, we have introduced our project and what is the goal of it, how to install the project and start using it in the electric drives lab including all the hardware connections to the Arduino, the configuration of each component using lab 1 as an example, the maintenance of the lab material so that it can in great shape and do not have any issues, and finally we have included some troubleshooting ways for the project just in case any issues arise from the hardware.

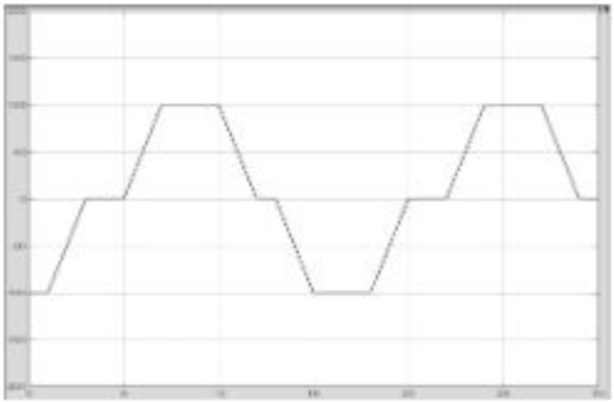
Appendices



The final hardware connections



Encoder and inverter board connections to the Arduino



Lab 1 expected speed results from dSPACE