The Red Legs Robot

A Major Qualifying Project
Submitted to the faculty of
WORCESTER POLYTECHNIC INSTITUTE
In partial fulfillment of the requirements for the
Degree of Bachelor of Science

May 16, 2020

James Burley
Nam Tran Ngoc
Brandon Waugh

Project Advisor:
Professor Michael A. Gennert

Abstract

The Red Legs robot was created in 2011 to be a sensitive, bipedal walking platform. However, over the past few years, Red Legs has been neglected and has fallen into a state of disrepair. The goal of this project is to design working systems to enable the Red Legs robot to carry out basic movements and set up an intuitive framework for future work. The project also aims to create comprehensive documentation along the way. To achieve this goal, the robot is examined and studied to find out precisely its current state, and the manner in which it works. The project attempts to improve the designs of existing parts and software, and make necessary changes to allow for more complex movements.

Acknowledgements

Our team would like to thank our project advisor, Professor Michael Gennert for his continuing support and guidance throughout the project. We would also like to thank Professor Stephen Bitar, of the Electrical and Computer Engineering department for his input on circuit design and manufacture. Lastly, we would like to thank our sponsor, LORD Microstrain, for their generous donation of the Attitude, Heading and Reference System (AHRS).

Table Of Contents

List Of Images

List Of Tables

Authorship

Author	**Section**	**Page No.**
Nam Tran Ngoc	Methodology: Power Distribution and Safety Mechanisms Methodology: Controller Board Design Results: Power Distribution Board Results: Main Control Board	11-12 13-20 27-28 29-32
Brandon Waugh	Current Status: Software Designs Design Goals Methodology: Sensors Results: AHRS Conclusion Recommendations for Future Work	7-8 8-10 21-26 34 36 36-37
James Burley	Introduction Background Current Status: Mechanical System Current Status: Electrical System Results: Safety Wiki Conclusion	1 2-5 5 6 34 34 36

Introduction

The goal of this project is to get Red Legs functional by redesigning and refurbishing different parts of the robot. At the beginning of the project, Red Legs was in various states of disrepair, and there was very little documentation to give insight on how the robot worked. Focus was placed on the electrical and software components of the robot. The electrical system was completely rewired, a new power distribution board along with a centralized control board was designed and inserted. A new software layer was rewritten, with documentation and support to the aforementioned hardware. By reworking Red Legs, future research groups could use Red Legs as a platform for bipedal robotics development - especially on sensitive walking and gait generation.

Background

The Red Legs robot was created in 2011 by Vadim Chernyak, a graduate student at WPI, in an effort to design a sensitive, bipedal walking platform[1]. In order to decrease mechanical stress and conserve electrical energy, he built it to be more mechanically similar to natural structural and muscular systems than other bipedal robots. Chernyak left the robot without having given it the ability to walk, but the project was taken up by two successive Major Qualifying Project (MQP) teams [2][3] which did electrical and mechanical work on the robot. The work of the latter MQP team resulted in the robot being able to stand on its own, and to perform walking motions while being suspended in the air.

However, since it was last worked on in 2017, the Red Legs robot has fallen into a state of disrepair. As seen below in figure 1, multiple sensors are displaced; circuit boards are disconnected, and there is some mechanical wear and tear on the robot. While documentation of all of this work is available, it is not nearly extensive or thorough enough to clearly explain how to bring the robot back to a working condition. This was a major setback in the project as much valuable time was spent trying to learn how the architectures of the robot were integrated - its mechanical, electrical, and mostly its software architecture. Time was also spent trying to find the exact specifications of some of the parts on the robot, and testing some of the circuit boards to determine their operation.

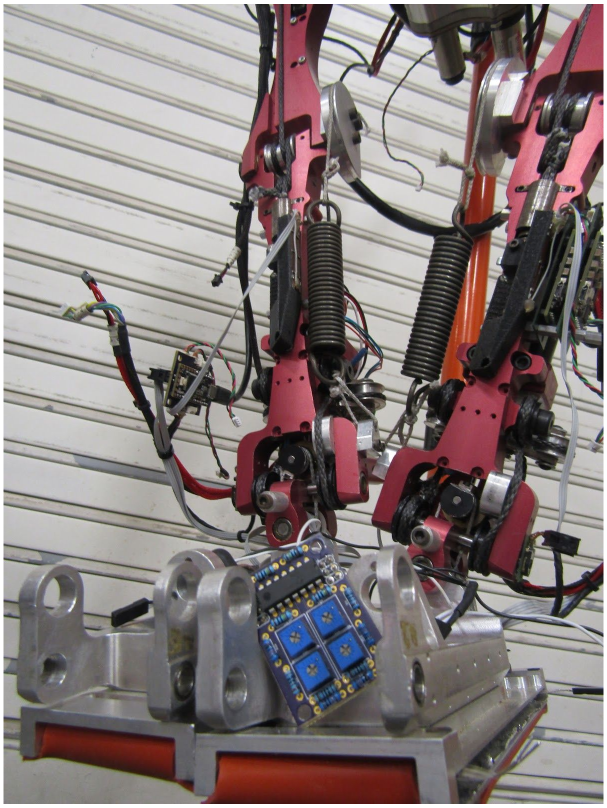


Figure 1. Electrical disarray

In view of this, the team project partially morphed as it became apparent that progress toward making the robot walk would be substantially hindered by the lack of documentation of the previous work done on the robot. Thus, rather than working solely to make the robot walk, the team decided to put a significant amount of effort into creating a platform off of which future teams could easily work.

This new project goal manifested itself in the creation of a number of different elements geared specifically toward the future. This includes clear documentation of the robot - what previous teams have done to it, as well as how this team has improved it; a Wikipedia page which contains all of the technical information on the robot presented in concise, clear form; schematics, Bill of Materials, and links to the specifications of many of the parts added during this project; kicad models, and new, improved circuit boards for power distribution and signal data analysis; a rewired system with added safety features; decisions which are designed to improve the software architecture of the robot; and a restructuring of a portion of the electrical and software systems in order to facilitate the newly decided upon architecture.

Current Status

Mechanical System

The Red Legs robot is a bipedal walking platform, with 10 degrees of freedom and 12 joints. The robot frame is built out of aluminum, with the front and back plates made out of carbon fiber reinforced wood. A 36V, 12AH Li-ion battery sits on top of the frame.

Figure 2. Red Legs

The robot joints are controlled by means of Series Elastic Actuators (SEA). Nylon cables are attached in series with the motor shafts, which are then connected to plungers that are inside the springs. Series elastics can be used for dynamic control of the joints. In the original design, potentiometers are attached to the springs to measure the spring deformation, and thus can be used to calculate the force applied on each joint. Unfortunately, the potentiometers are currently damaged and are not functional.

Electrical System

At the beginning of the project, the electrical system of the robot was in a state of disrepair. Due to the condition the robot was stored in, certain wirings were either disconnected or damaged. Some of the existing force sensors on the robot feet were missing, and the digital encoders had fallen off, rendering it impossible to measure the angle of the joints.

The power distribution board designed by O'Brien et al (2017) was examined, and was found to be unsatisfactory. The power board was designed to distribute battery power to the motors in parallel. However, there were no safety features installed. This can prove to be dangerous to both the operator and the robot. Thus a safety system was deemed to be necessary.

The existing control system was also examined. The original design utilized 4 separate Teensy development boards, each of them responsible for controlling 2 to 3 motors, alongside relevant sensors. The 4 Teensy boards are then connected to an USB Hub, which can then be connected to an external computer for gait generation and control. An improved design was implemented for this crucial portion of the robot's architecture.

Finally, there was a lack of comprehensive documentation on the electrical system. O'Brien et al designed a power distribution board, and multiple "helper" circuit boards attached to the Teensy microcontrollers - however, there were no schematics provided. Claretti (2013) worked on a motor controller system, in which no documentation was provided. It was therefore necessary to test and examine each individual board, and gain an understanding of how they worked before moving on with the project.

Another setback we faced with documentation was tracking down small components. Since one of our design goals was to make the system backward-compatible, different parts have to be measured and tracked down, such as power and motor connectors.

Software Designs

O'Brien et al. set up a Github repository as a version control system for their software. Unfortunately, the existing code only contains firmware, most of which were undocumented. O'Brien and their team mentioned an external laptop with software used to control the Teensy boards. However, this laptop was nowhere to be found, and no such code was present on the Github repository.

The previous repository can be found at https://github.com/samkhal/biped.

Design Goals

In order for the team to realize their goal of getting the robot operable, a number of design goals were set out.

Firstly, safety mechanisms needed to be installed, in order to prevent damage to the operator or the robot itself.

Secondly, there was the goal to improve upon the original electrical and software design. This goal was split into multiple objectives. The power distribution board had to be redesigned, manufactured, and installed on the robot as part of an effort to improve the power distribution system, which was lacking in safety and function. This sub goal also directly helps to move the robot closer to being able to walk.

The communication system needed to be redesigned, which included the design, manufacture, and installation of a main control board. This custom control board was an integral part of streamlining the communication system since it allows the movement of every joint to be reported directly to one centralized computer which can autonomously process that data rather than be spread over multiple microcontroller boards which then communicate with an external computer for instructions. This sub goal also directly helps to move the robot closer to being able to walk.

A new Attitude, Heading and Reference System (AHRS) had to be purchased, and integrated with the new system. The AHRS will be essential to the robot's ability to walk.

Lastly, we aimed to document the progress that was made and our process for improving the design. A large part of the project was spent on analyzing the robot and reverse engineering the electrical system. This is something that we would like future teams to avoid by documenting our past and present progress.

Overall, the implementation of the goals listed above should be able to dramatically aid students who work on the robot. The result of these goals are:

- To ensure that all parts on the robot are working well so that they will not need to be tested anytime in the near future unless the robot is accidentally damaged.
- To streamline the systems in place on the robot, in some cases drastically changing them, so that students working on the robot in the future will have an efficient, well thought out robot to work with.
- To make easily accessible, and abundantly clear the workings of the robot.

Methodology

In order for the team to complete their goal, the following plans for design and execution were laid out. After close examination of the robot itself, the team looked at all of the available documentation on the robot, such as project papers and part order forms. Next, the team verified which parts were still on the robot and if said parts functioned properly. This verification phase included an examination and testing of past teams' software, power distribution system, sensory systems, motors, joints, springs, and cables. Once the examination was completed, we set out to repair or replace anything that proved crucial to getting the robot to walk.

The next step was to restore the robot to a working condition by first working on necessary repairs on the robot's mechanical and electrical systems to bring them to a funcional state. The team attempted to simplify different parts of the robot, especially its electrical system by, among other things, reducing the number of off-the-shelf circuit boards. A new power distribution system was designed with safety and reliability as its key features. A new communication architecture was to be designed which included the design of a main control board. A safety mechanism, of which the robot was previously entirely devoid, was added to the robot. Missing or broken sensors were also researched and replaced, and all necessary sensor and board remounting was to be done as well.

Finally, the team researched, tested, and designed the methods of system operation for the robot. Hardware design was thought to be a part of the project until potentially even the end of the project. Software design began in C-term, and was to progress to such a point by the end of the project that sensory data from all of the various sensors could be read, processed, and subsequent commands could be sent to the motors which would cause the robot to walk and stand.

As the project progressed, and the team gradually became more aware of the difficulties they were up against, new project goals were approved by the project advisor as a set of realistic and useful advancements to make on the robot. These goals were to be realized through the creation of deliverables to future project teams.

Furthermore, the team created a Github wiki page, where future teams could look to find most if not all of the useful information they might need for working on the robot. This wiki consolidates much of the information from years prior thought to be of interest to future teams, as well as much of the information relating to improvements made during this project in a clear and concise manner.

Power Distribution and Safety Mechanisms

The Red Legs robot is powered by a 36V Lithium-Ion battery. Due to the nature of the battery materials, and the current draw of the motors, an improved version of the existing power distribution board was designed. The new board can accommodate up to 12 connections, 10 of which are reserved for the motors. The remaining two can be used to power the control board, and an extra peripheral if needed. Care was taken to design the board with a suitable trace width, capable of handling up to 30A of power at maximum load, as shown in Figure 3.

In addition, fuses were installed in series with each connection in case of electrical shorts, alongside a master fuse inline with the battery. The fuses are automotive blade fuses - chosen for their availability and ease of replacement. The connector fuses are chosen to handle 3A current, while the master fuse has a limit of 30A.

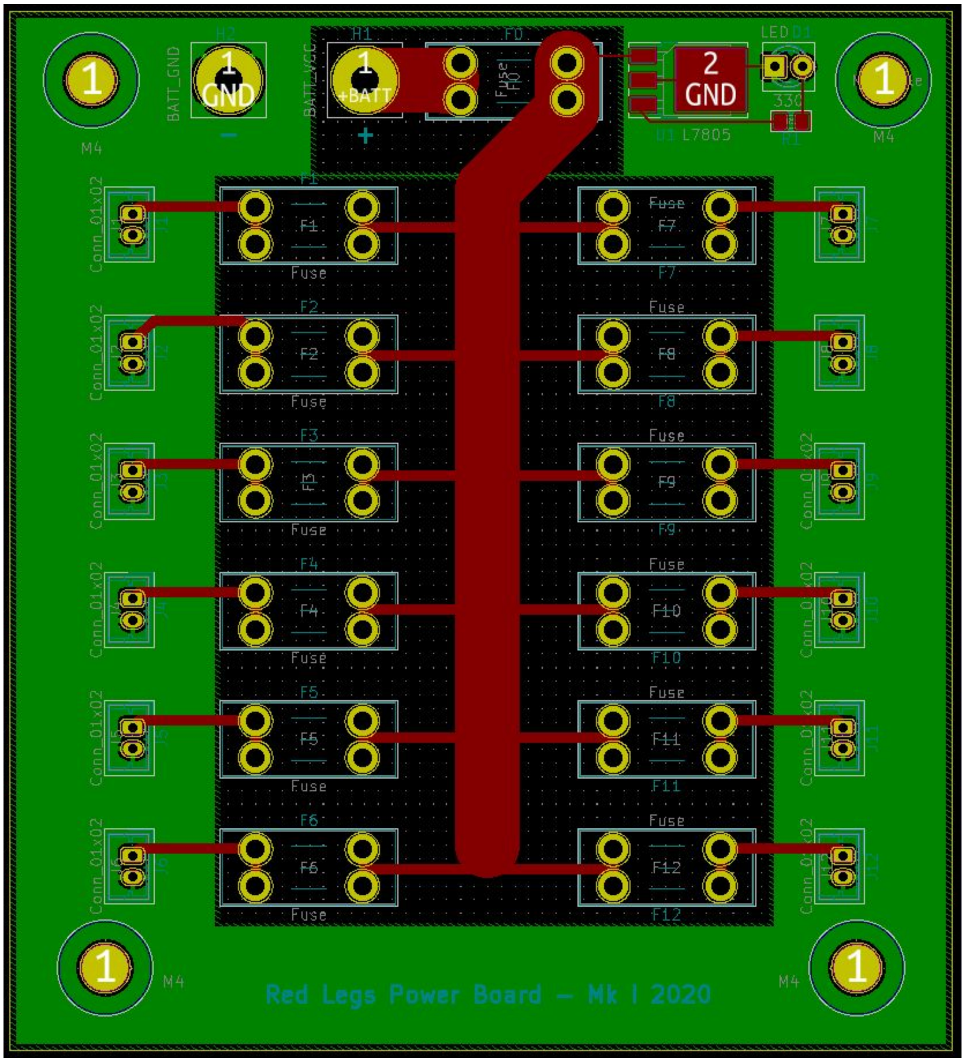


Figure 3. Power Distribution Board Layout

In order to safely operate and develop Red Legs, an emergency stop system was added. This is connected in series with the battery, and can be used to cut off power coming into the system. Due to the fact that the Emergency Stop button was rated for 10A, the team opted to connect the button to a contactor switch, which will isolate the button circuit with the main circuit.

Controller Board Design

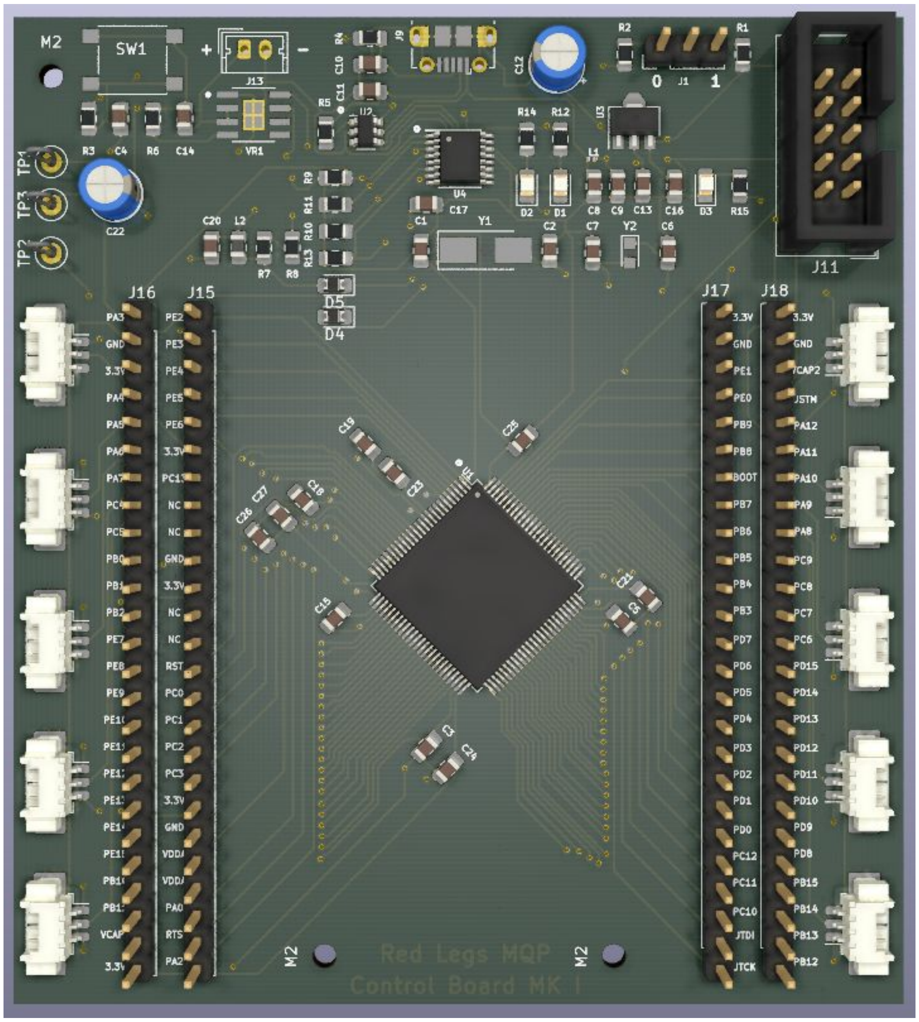


Figure 4: 3D Render of Control Board

Figure 4 shows an overview of the controller board. The main purpose of this board is to provide a centralized development environment, with one microcontroller controlling the robot peripherals and actuators. The control board can be split into 3 sections: Power regulation, Debugging and communication, and Main microcontroller.

Main microcontroller

In order to control the robot in its full capability, the heart of the control board has to satisfy the following requirements:

- Capable of controlling motors and peripherals across multiple interfaces (I2C, SPI, DMA).
- Support Real Time Operating Systems.
- Multiple communication options (UART, USART, JTAG/SWD, CAN).
- Well-documented and well-supported platform.
- Multiple GPIO channels for future expansion.
- Low power consumption for battery power.
- Capable of high speed operation.
- Reasonably priced.

From these outlined criteria, we decided to go with ST for our microcontroller vendor. The 32-bit microcontroller line (STM32) was widely used and well documented, and with the help of ST’s tool, MCU Finder, we were able to select the STM32F427VGT microcontroller, which satisfies all of our requirements.

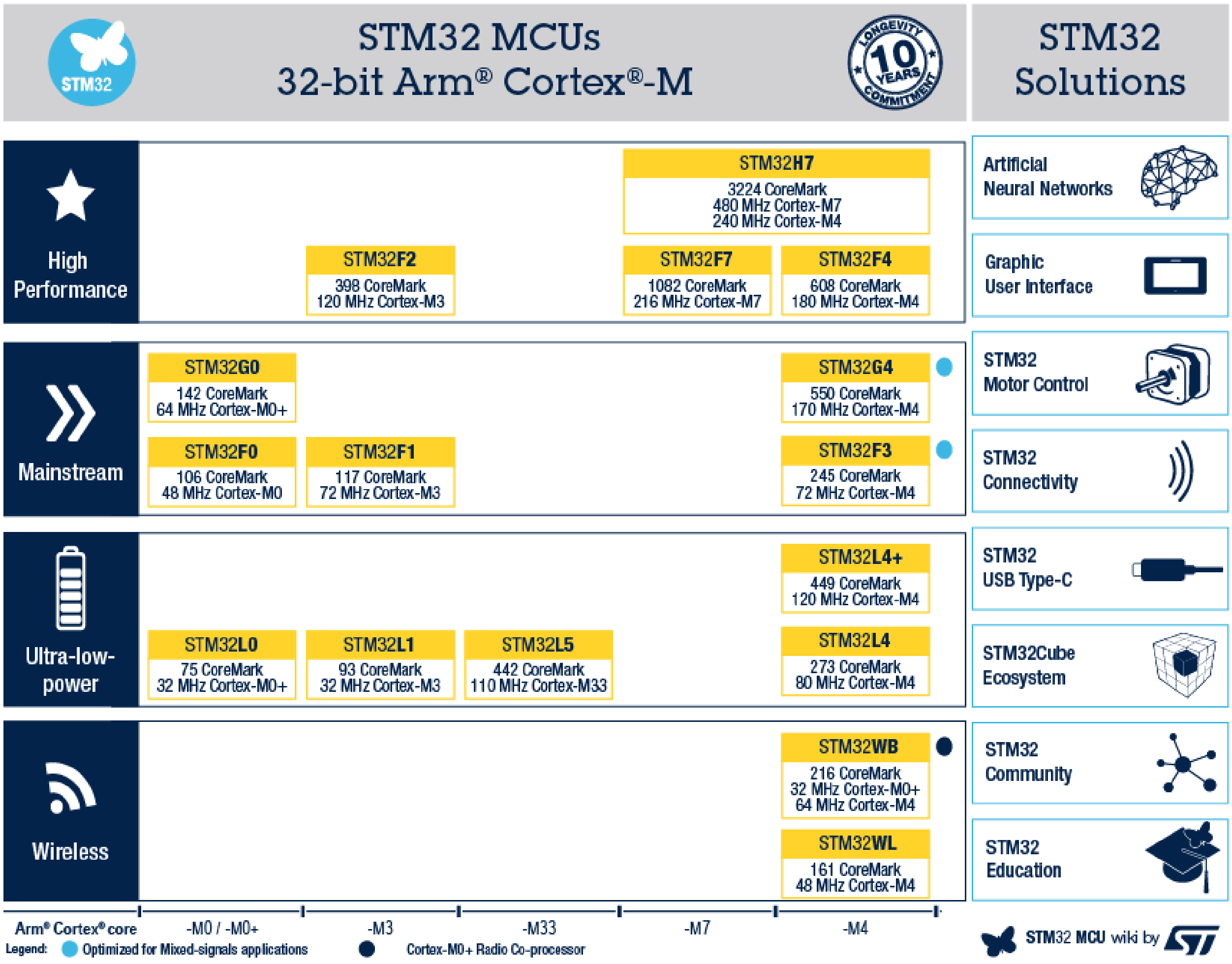


Figure 5. STM32 MCUs

Figure 5 above shows different lines of STM32 products. The on-board STM32F4 controller falls into the High Performance category

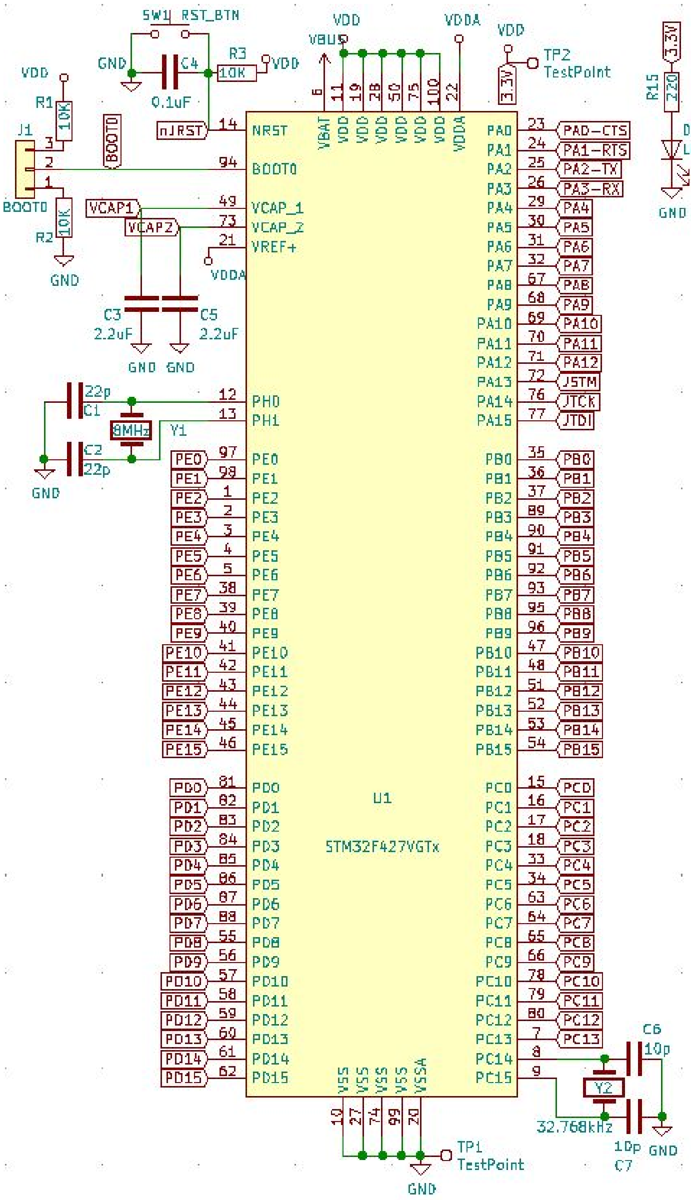


Figure 6: STM32F427 Connection diagram

Figure 6 shows the connection diagram of the STM32F427. Two notable connections are our BOOT0 pin, and our oscillators.

An external oscillator is required for precise timing. We picked commonly used oscillator values for both low frequency (32.768kHz) and high frequency (8MHz) applications. Our accompanying capacitor values were chosen based on the following equation: [4]

$$C_{CL} = \frac{C_1 * C_2}{C_1 + C_2} + C_{stray}$$

Where C_{CL} represents the oscillator's own load capacitance, C1 and C2 are the accompanying capacitors, and C_{stray} represents the stray capacitance present in the traces. We opted to use a nominal value of 4-5pF for C_{stray}. The 8Mhz oscillator possesses a load capacitance of 15pF, and the 32.768kHz oscillator has a capacitance of 12.5pF. Using the equation, we were able to choose the values of 22pF for C1 and C2, and 10pF for C6 and C7.

The BOOT0 pin on the STM32 IC is used to select the boot location in memory. The IC can be booted from user Flash, system memory, or the embedded SRAM, depending on the input voltage on the BOOT0 pin. We have connected the pin to a header jumper, enabling the user to switch boot modes at will.

Power Regulation

The board is designed to be powered from both the onboard battery and external power via a USB port. To achieve this, the team designed a two-stage power regulation system. The incoming power is first regulated to 5V, and subsequently regulated to a 3.3V signal. This 3.3V signal is used to power the onboard microcontroller and peripheral components.

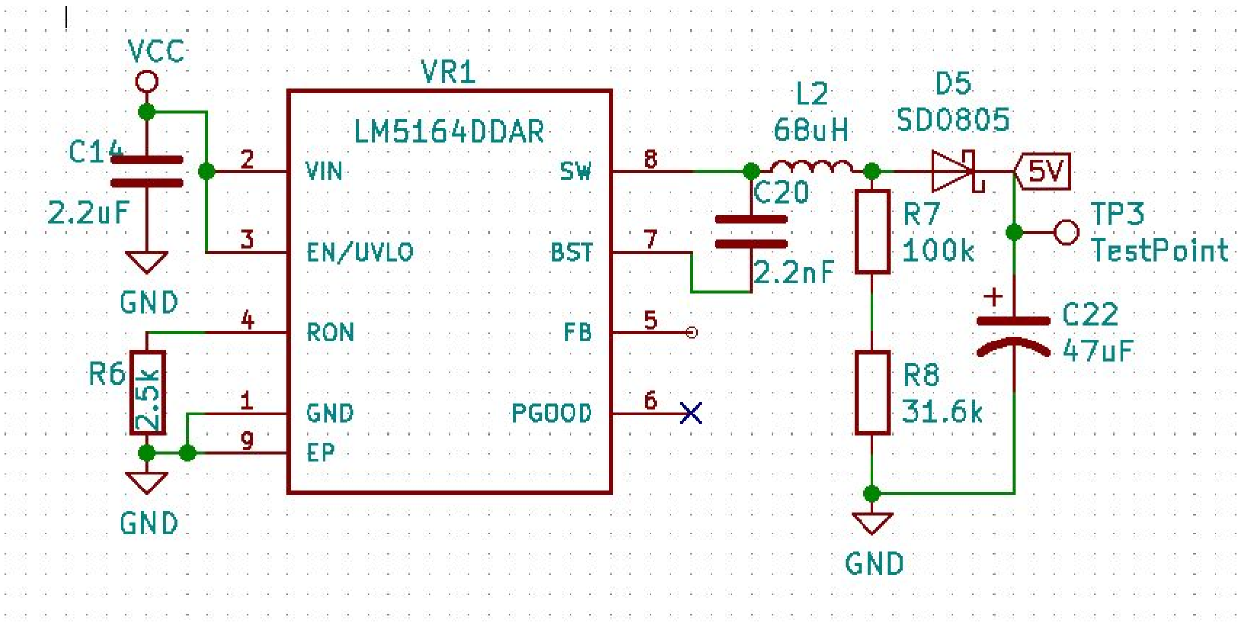


Figure 7. V-5V Voltage Regulation

Since the battery operates at 36V and we desire an output voltage of 5V, a suitable voltage converter needs to be used. We have chosen the LM5164 DC/DC Synchronous Buck Converter, due to its wide input ranges and low costs. Furthermore, as the LM5164 is a switching voltage regulator, our battery voltage is stepped down with minimal excessive heat.

In order to achieve a 5V output signal, the two resistors R7 and R8 were chosen based on the following equation [5]:

$$R_8 = \frac{1.2V}{V_{OUT}-1.2V} * R_7$$

At $R_7 = 100k\,\Omega$, and $R_8 = 31.6k\,\Omega$, we can achieve an output signal Vout:

$$V_{OUT} = (1.2V * \frac{R_7}{R_8}) + 1.2V = 4.997\,V$$

To reduce output signal noise, coupling capacitors and inductors were connected to the circuit.

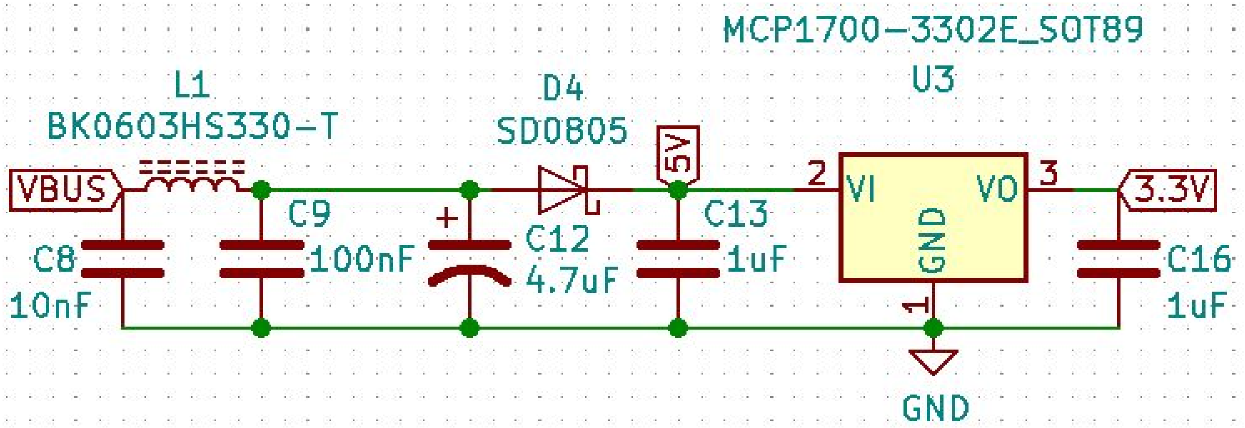


Figure 8. 36V-3.3V Voltage Regulation

At the second stage, the 5V signal is then fed through an MCP1700 IC, a linear low-dropout voltage regulator. Using this IC, the signal is dropped into a stable 3.3V output, suitable for our main microcontroller.

To ensure reverse voltage protection, two SD0805 Schottky diodes were installed - one on LM5164 output, and another on the USB VBUS line input. This prevents damage when the board is powered by both battery and USB.

Debugging and Communications

In order to communicate with the on board STM32 microcontroller, two communication interfaces were implemented.

The first one is a JTAG/SWD interface, connected to a 10-pin header. SWD stands for Serial Wire Debug, and is a popular alternative to JTAG since it uses the same protocol, with only three wires. Connections are made to the Clock signal, Data Input/Output, and Reset. A schematic is outlined as below.

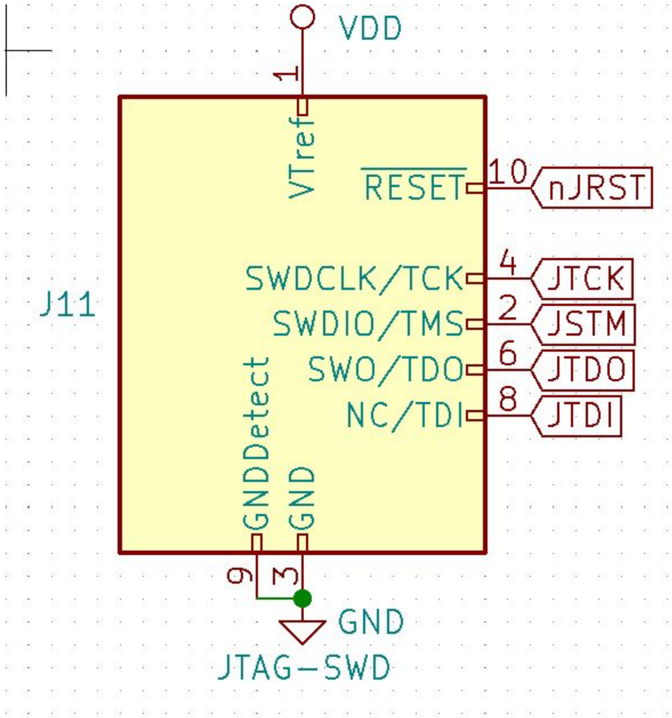


Figure 9. JTAG/SWD interface

The board can also communicate via USB using the Serial interface. Pins PA0, PA1, PA2, and PA4 are designated to the USART1 channel on the STM32F427, which are then connected to the FT230XS USB-to-Serial Interface IC. A USBLC6 IC was inserted between the data line and the Serial interface for ESD protection. Two LEDs are connected to pin 7 and pin 14 of the FT230 IC to signify data transfer.

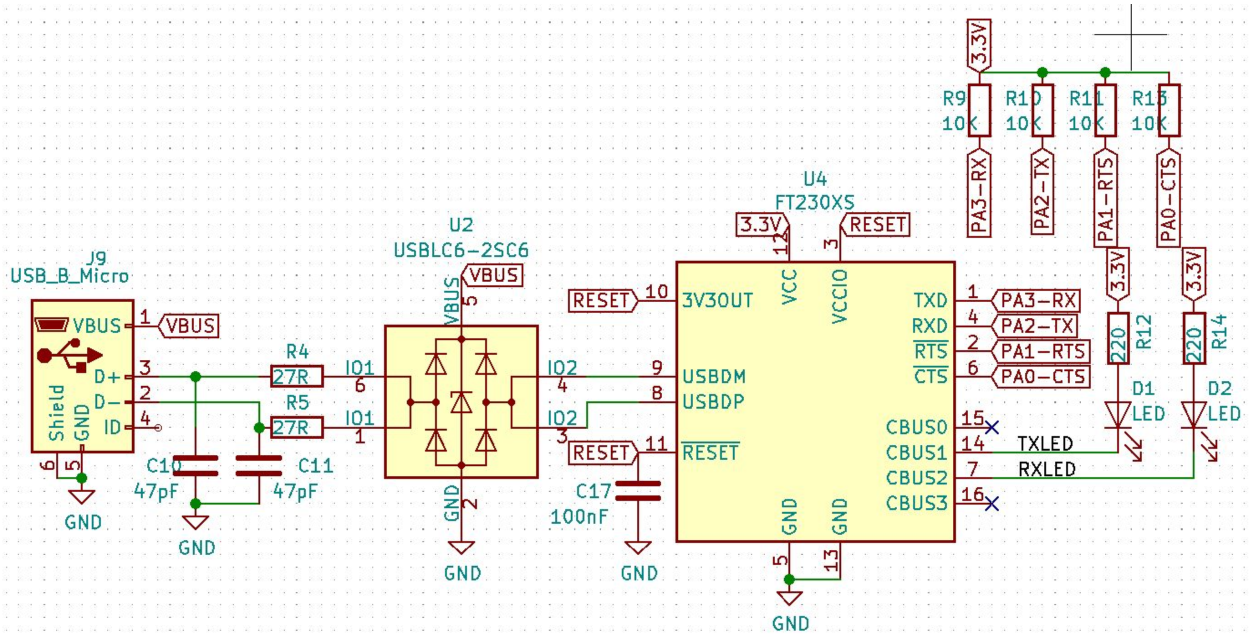


Figure 10. USB-Serial Communication with STM32F427VGT

Sensors

Attitude and Heading Reference System

The existing Attitude and Heading Reference system (AHRS) was a 3DM-GX3-25 from LORD MicroStrain. It was accompanied with a nonfunctional, custom jury-rigged USB 2.0 to mini RS232 serial connection cable. The team before us didn’t have the money in their budget to purchase one for the quoted $90. The software for the GX3 was also only supported up to Windows 8 and not supported by the development team at LORD MicroStrain.

In response, the team decided a new AHRS was required. After doing research, the team found that LORD MicroStrain had a newer version of the same sensor; the 3DM-GX5-25. As seen below in Table 1, The 3DM-GX5-25 has a very accurate measurement range making it perfect for doing the precise calculations it takes to get the robot walking.

	Accelerometer	Gyroscope	Magnetometer
Measurement Range	±8g	300°/sec (standard)	±8 Gauss
Non-Linearity	±0.02%fs	±0.02%fs	±0.3%fs
Resolution	0.02mg(+/-8g)	0.0003dps	--

Table 1. GX5-25 IMU Sensor Outputs [6]

Because LORD MicroStrain wanted to be a sponsor to the project, we were able to receive this new AHRS free of charge as a donation. The 3DM-GX5-25 is able to communicate via USB2.0 and RS232 communications. We were given the necessary cables to connect by both methods.

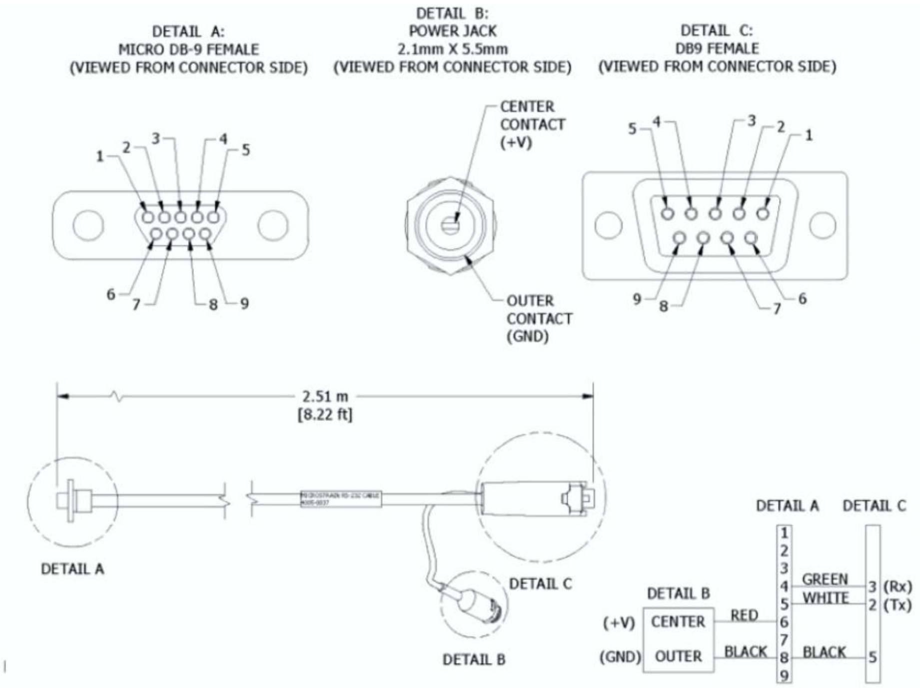


Figure 11. RS232 Communications and Power Cable [7]

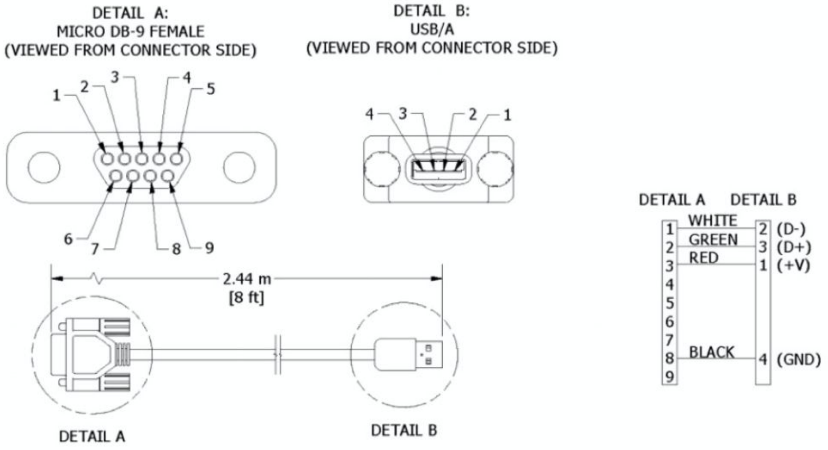
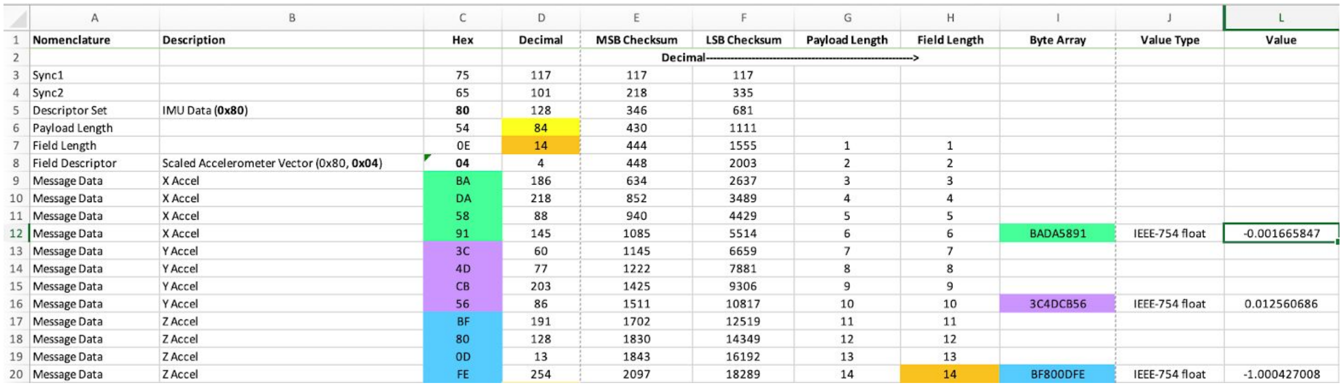


Figure 12. USB 2.0 Communications Cable [7]

Since our goal was to be able to have Red Legs function without a tether to a computer, the AHRS had to be able to work untethered as well. With the provided setup software *MIP Monitor*, the AHRS was set up to output the following values: Accelerometer Vector, Angular Rate Vector, CF Attitude (Matrix), and CF Quaternion. Once resetting the AHRS’s default setup to output the data we wanted, it would continuously stream data packets filled with information. Each data packet was 90 hex bytes long starting with the two sync bytes and ending with 2 checksum bytes. For more information on how to set up the sensor, head over to the GitHub wiki.

To test that the AHRS was properly set up, the AHRS was connected to a computer with the USB 2.0 communications cable. Then code was written to open up a computer's com port and read the incoming hex bytes. To read the data in a useful way , i.e. not just random hex bytes, they had to be parsed through in a specific way. For our values, (the incoming character array was 90 bytes long. Each set of specific data like had an field descriptor byte and then byte arrays of length 4 of associated values. Each byte array is then converted to IEEE-754 floats to retrieve the correct values. Below in figure 13, is an example of parsing the Accelerometer Vector returned from the AHRS.



	A	B	C	D	E	F	G	H	I	J	L
1	**Nomenclature**	**Description**	**Hex**	**Decimal**	**MSB Checksum**	**LSB Checksum**	**Payload Length**	**Field Length**	**Byte Array**	**Value Type**	**Value**
2					**Decimal-->**						
3	Sync1		75	117	117	117					
4	Sync2		65	101	218	335					
5	Descriptor Set	IMU Data (**0x80**)	**80**	128	346	681					
6	Payload Length		54	84	430	1111					
7	Field Length		0E	14	444	1555	1	1			
8	Field Descriptor	Scaled Accelerometer Vector (0x80, **0x04**)	**04**	4	448	2003	2	2			
9	Message Data	X Accel	BA	186	634	2637	3	3			
10	Message Data	X Accel	DA	218	852	3489	4	4			
11	Message Data	X Accel	58	88	940	4429	5	5			
12	Message Data	X Accel	91	145	1085	5514	6	6	BADA5891	IEEE-754 float	-0.001665847
13	Message Data	Y Accel	3C	60	1145	6659	7	7			
14	Message Data	Y Accel	4D	77	1222	7881	8	8			
15	Message Data	Y Accel	CB	203	1425	9306	9	9			
16	Message Data	Y Accel	56	86	1511	10817	10	10	3C4DCB56	IEEE-754 float	0.012560686
17	Message Data	Z Accel	BF	191	1702	12519	11	11			
18	Message Data	Z Accel	80	128	1830	14349	12	12			
19	Message Data	Z Accel	0D	13	1843	16192	13	13			
20	Message Data	Z Accel	FE	254	2097	18289	14	14	BF800DFE	IEEE-754 float	-1.000427008

Figure 13. Parsing the Accelerometer Vector

Once it was verified to be working properly, the next step was to get an STM32F4 series chip to communicate with it. For all testing with the AHRS, an STM32F429i discovery board, like the one below in figure 14, was used.

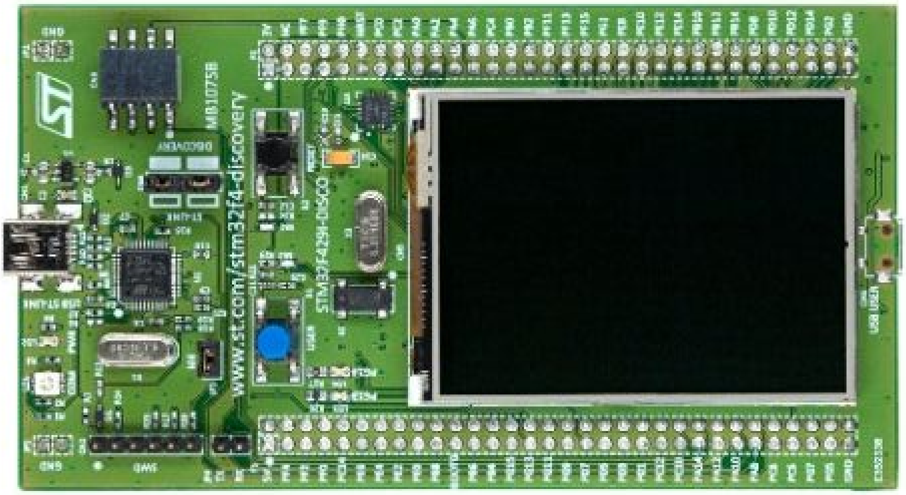
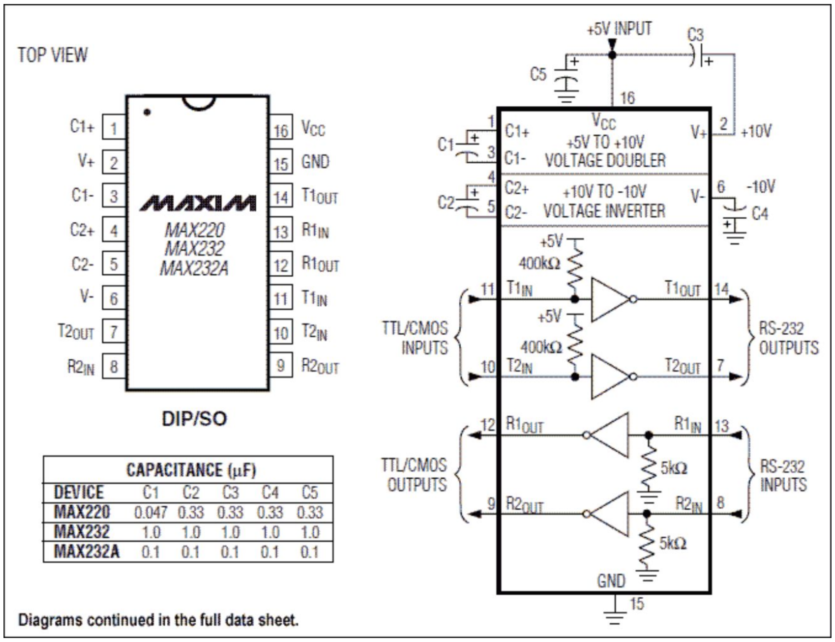


Figure 14. The STM32F429i discovery board

In order to know we were communicating properly and receiving data packets from the sensor, the received data was sent via the Universal Asynchronous Receiver/Transmitter ports on the discovery board to the computer's COM port. To receive the data properly, the signal from the AHRS first has to be run through a MAX 232 chip to convert the signal from RS232 serial to a signal suitable for TTL digital logic circuits. The code previously written to parse the data was reused for getting workable data. The code library was then expanded to be able to specifically retrieve different pieces of data outputted from the AHRS.



CAPACITANCE (µF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	0.047	0.33	0.33	0.33	0.33
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

Figure 15. MAX232 Circuit [8]

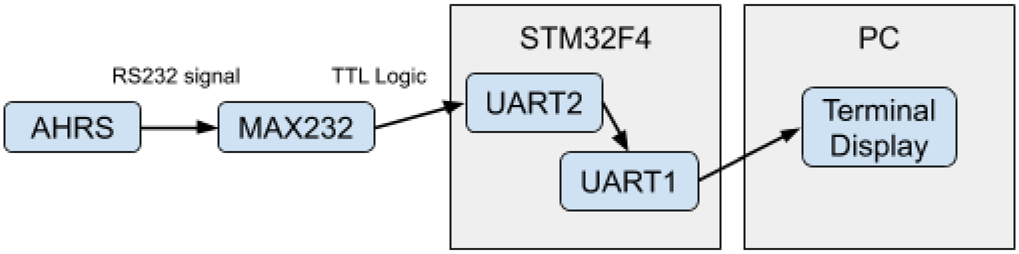


Figure 16. Connection of AHRS to Terminal display to verify data.

Results

Safety

The robot was scrutinized for weak spots in its safety, which mainly presents in the power system. Our addition of the emergency stop button and the fuses proved to be successful, as we were able to cut off current to the motors, and kept the current draw under the fuse limits.

Power Distribution Board

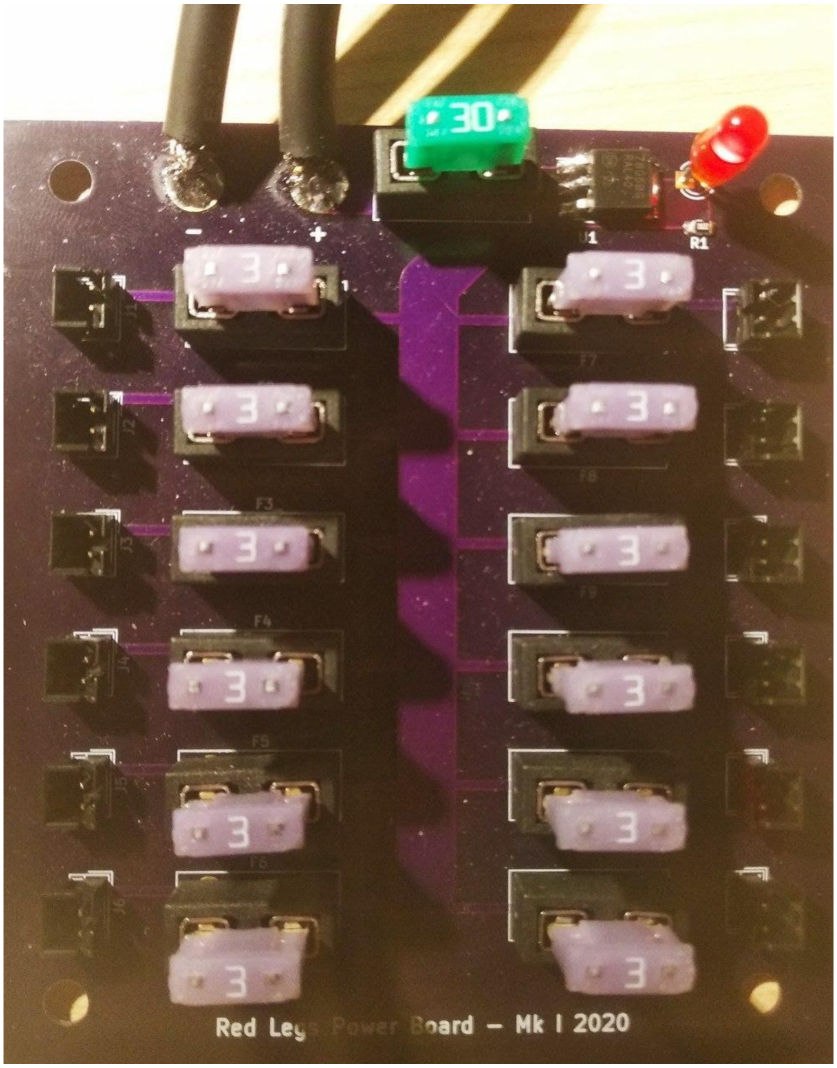


Figure 17. The Power Distribution Board

A new power distribution board has been designed with 12 individual fuses, and 1 master fuse to regulate current flow. An indicator LED has been mounted, allowing observers to visually determine the power status of the robot. The new board also has an improved layout, and a schematic and Bill of Materials are supplied.

Furthermore, as an effort to isolate and compartmentalize the circuit, we ordered a custom sized enclosure to house the contactor and the Power distribution board. While the contactor enclosure fits, the Power distribution board proved to be slightly bigger than its own box, and will need to be altered.

Due to the impact of the novel COVID-19 virus, we were unable to mount the newly added components without the necessary hardware tools.

Main Control Board

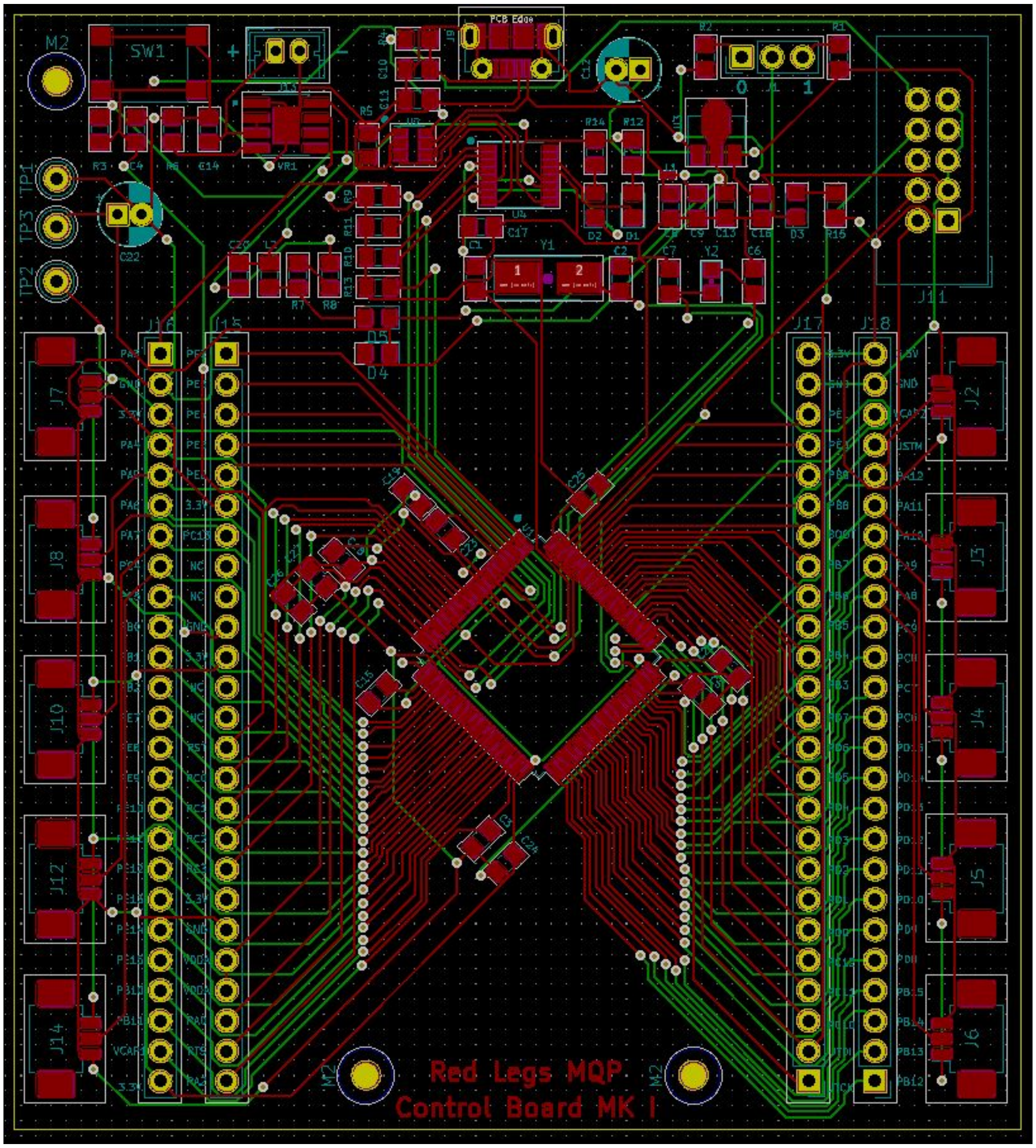


Figure 18. The Main Control Board

Surface Mount Components

The newly designed control module was printed and assembled. The components were chosen to be surface-mounted, in an effort to make the design workable on a two-layered circuit board. Furthermore, we have chosen to use component size 0805 (0.08’’ x 0.05’’, or 2mm x 1.2mm) and enlarged to allow for hand solder. This is to account for the lack of laboratory equipment - as the team were forced to work from home under quarantine.

Test Points

In order to verify our voltage regulation circuits, and ease of debugging in the future, the team added several test points to the system, positioned on the top left side of the board. The three connections that can be tested align with the 5V, 3.3V and Ground signals.

Breakout Headers

The STM32F427 microcontroller is capable of driving up to 80 General Purpose Input/Output (GPIO) pins. To account for future uses, the team designed male breakout headers on both sides of the board. Future teams can take advantage of the headers to add new components and sensors to the robot without modifying existing electrical connections.

Design Challenges

Over the course of assembling the control board, the team ran into connectivity issues. A test firmware was compiled, however it was unable to be loaded onto the control board via the JTAG/SWD interface.

To troubleshoot this, the team used a connectivity meter and a Black Magic Probe connected to the circuit board. The Black Magic Probe is an in-circuit debugging tool that supports a variety of interfaces, including JTAG/SWD. Furthermore, the Probe can connect to a GNU Project Debugger (GDB) server to monitor the state of the STM32F427 microcontroller.

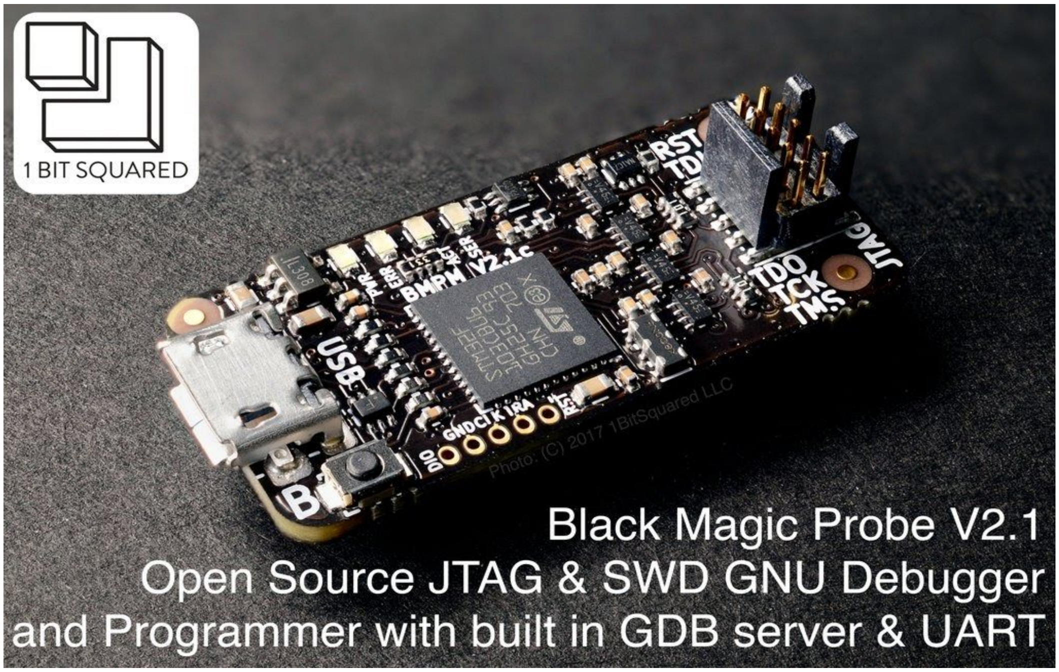


Figure 19. The Black Magic Probe V2.1

One notable problem was the fact that the microcontroller was stuck in a reset-boot loop. It was found that the V_{BATT} pin, responsible for powering the microcontroller under battery power, had been mistakenly connected to 3.3V main power at V_{DD}. This was rectified by physically cutting the trace on the first revision of the board.

Another problem that prevented the board from working properly is the fact that pin V_{DDA} was not connected to main power, while only connected to its own decoupling capacitor. A solder bridge was made between two capacitors connecting V_{DD} and V_{DDA}, allowing the microcontroller to be powered properly.

After the problems were rectified, the firmware was successfully uploaded to the board using the JTAG connector.

The first version of the board is otherwise not without faults, however. Connections to the 32.768kHz oscillators were made through vias - connections from one layer of the board to another. Vias can act as capacitive/inductive discontinuities in trace lines, which can cause signal integrity problems. A second revision of the control board can shorten the traces line to the oscillators, and keep them on the same layer.

Due to time and manufacturing constraints, the board was designed to account for 2-layer manufacturing. This causes the board to be larger than necessary, and traces to be routed close to component pads, or trace hugging. Trace hugging can cause connectivity problems during the assembly process. By redesigning the board layout to account for 4-layer manufacturing, traces can be routed in a better way, and reduce the size of the control board.

Attitude and Heading Reference System

The existing 3DM-GX3-25 Attitude and Heading and Reference System (AHRS) was replaced with a new AHRS that was donated to us by LORD MicroStrain. With the newer 3DM-GX25, we are able to get the acceleration vector, angular rate, attitude matrix, and quaternion of the robot so we know what the robot's position is at all times.

Code was written to test connection between the AHRS and a computer's COM port via USB 2.0. As previously described in the Methodology, this test involved opening a computer's COM port and reading the incoming data. The test was successful and proved it was possible to not use the provided code library from the manufacturer.

Then using an STM32F429I-Discovery development board, which utilizes a similar chip to the custom control board, tests were run trying to read data from the AHRS via RS232 communications. These tests, previously described in the Methodology section, were successful and a code library was created for such applications.

Unfortunately, the code was not tested with the actual control board because the AHRS and control board were in two different locations, due to the effect of COVID-19.

Wiring

Since the robot's wiring was dangling off of the robot in an unseemly, and potentially unsafe manner, it had to be neatened up. This entailed some rewiring of the robot, but mainly just rerouting and securing wires and cables in various places.

Wiki

The wiki was created as a way for future teams to quickly learn important information about the robot. The information put into the wiki will be useful not only in the initial process of learning how the robot works, but also as the project progresses, as the wiki contains schematics and Bill of Materials, which are essential to making design choices.

The wiki is made up of a Home page, an Overview page, a Bill of Material page, a Roadmap page, alongside technical pages concerning the mechanical, electrical and software aspects of the robot.

The idea behind the Overview page is to give students who are new to the robot a quick, easily understandable description of how the robot works, giving them an effortless, though not very detailed, grasp of its operation.

The Overview page gives a top-down view of the robot, including the mechanical, electrical and software makeup. Links to other sections of the wiki are included.

The Bill of Materials lists the quantity and footprint of each component placed on the main control board, and the power distribution board. Many of the listed components are also paired with a live link so that students working on the robot in the future are only a click away from the most detailed description of these parts that they could possibly wish for.

The Roadmap page outlines a series of improvements which can be made to the robot in the future. These are changes that have not been made to the final revision of the board due to a lack of resources, time, or otherwise unforeseen circumstances.

A link to the Wiki can be found at: https://github.com/bmwaugh/red_legs20/wiki/

Conclusion

Good progress was made on the robot over the course of this project. Much of the work already done by previous teams was improved upon and polished. This includes new circuit board design which increases both the safety of those who are near the robot and the efficiency of its communication architecture.

An emergency stop switch was added to the robot to further add to the safety of the robot, and a new AHRS sensor was bought and tested as well.

Thorough documentation was created in a user friendly format which will make the project much easier to get off the ground in the future for any teams who work on the robot. This documentation includes BOMs and schematics, as well as live links to the websites where various components were purchased.

Recommendation for Future Work

These recommendations include a list of problems and possible solutions directly related to the problems that the team has faced.

The new control PCB and power distribution PCB should be properly mounted with a containment unit built around the power distribution board. This containment unit should be able to fully enclose the power board to keep it safe from the elements as well as keep the other electronics on board from accidently coming into contact with it. Keeping in mind that the robot should remain balanced, modifications to the pelvis wings will probably be necessary to mount the boards.

The shaft encoders should be replaced and then properly mounted. As before, they were hot glued on and were unreliable. Then the encoders should be implemented into the existing code.

The series elastic actuators remain nonfunctional. Our team never got the chance to look into these in depth. As the previous team suggested, a redesign of the series elastic may mean that an entire redesign of the system is necessary.

Like many other things on the robot, the foot sensors need to be repaired and reattached. Some of the circuits created by past MQPs were missing. It should also be noted that adding sensors to be able to sense force in the horizontal direction could prove useful when trying to use Red Legs for sensitive walking.

Materials and Budget

The group was allotted $750.

Item	Quantity	Price per Unit	Total Cost
30 Gauge Flexible Silicone Rubber Electric Wire 6 Colors 32.8 feet Each 30 AWG	1	$13.88	$13.88
Fuse holder	25	$0.8512	$21.28
3A 58V fuse	25	$0.63	$15.75
30A 58V fuse	5	$0.65	$3.25
Male power connector	50	$0.165	$8.25
Female power connector	50	$0.06	$3.00
MC7805BDTRKG Voltage regulator	5	$0.56	$2.80
50058-8000 (cut strip) wire crimps	100	$0.038	$3.80
28 AWG 5 color wire set 33 ft each	1	$13.99	$13.99
3 pin receptacle 51021-0500	15	$0.197	$2.96
3 pin receptacle 51021-0300	30	$0.156	$4.68
5 pin connector 53261-0371	15	$0.929	$13.94
3 pin connector 53261-0371	30	$0.805	$24.15
DC Power Contactor	1	$28.92	$28.92
Emergency Stop	1	$15.99	$15.99
Fuse	13	$1.03	$13.39

Capacitor - 22p	2	$0.038	$0.76
Capacitor - 47pF	2	$0.063	$0.126
Capacitor - 4.7uF	1	$0.29	$0.29
Capacitor - 2.2uF	3	$0.10	$0.30
Capacitor - 0.1uF	11	$0.049	$0.539
Capacitor - 2.2nF	1	$0.027	$0.027
Capacitor - 47uF	1	$0.62	$0.62
Capacitor - 1uF	3	$0.114	$0.342
Capacitor - 10pF	2	$0.063	$0.126
Capacitor - 10nF	1	$0.063	$0.063
LED	3	$0.224	$0.672
Diode - SD0805	2	$0.322	$0.644
Connector - pin header	5	$0.126	$0.630
Molex PicoBlade Connector	10	$1.18	$11.80
Connector (JTAG-SWD)	1	$0.987	$0.987
Molex micro-B USB connector	1	$0.88	$0.88
Inductor - BK0603HS330-T	1	$0.14	$0.14
Inductor - 68uH	1	$0.19	$0.19
Resistor - 10KΩ	7	$0.059	$0.413
Resistor - 220Ω	3	$0.059	$0.177
Resistor - 27Ω	2	$0.02	$0.04
Resistor - 2.5kΩ	1	$0.65	$0.65

Resistor - 100kΩ	1	$0.039	$0.039
Resistor - 31.6kΩ	1	$0.081	$0.081
Button switch	1	$0.78	$0.78
IC - STM32F427VGTx	1	$11.69	$11.69
IC - USBLC6-2SC6	1	$0.45	$0.45
IC - MCP1700-3302E_SOT89	1	$0.37	$0.37
IC - FT230XS	1	$2.04	$2.04
IC - LM5164DDAR	1	$3.56	$3.56
Crystal - 8MHz	1	$0.81	$0.81
Crystal - 32.768kHz	1	$1.10	$1.10

Table 2. Purchased components for the project

In total, the team spent $232.37 on developing the circuit boards and improving the electrical system. An extra $252 was spent on manufacturing circuit boards.

References

[1] Vadim Chernyak. The Design and Realization of a Sensitive Walking Platform. WPI, 2012.

[2] Ennio Claretti. The Design and Realization of a Sensitive Walking Platform System Design. WPI, 2013.

[3] Alexander Preston O'Brien, Ari Benjamin Goodman, Georgios Ardamerinos, Logan William Tutt, Samuel Khalandovsky. Control of a Bipedal Robot. WPI, 2017.

[4] Texas Instruments, "Crystal Selection Guide," AN100 datasheet, Nov. 2013.

[5] Texas Instruments, "LM5164 100-V Input, 1-A synchronous buck DC/DC converter with ultra-low IQ," LM5164 datasheet, Sep. 2018 [Revised Jan. 2019].

[6] LORD Sensing, "3DM-GX5-25 Datasheet," 8400-0093 datasheet, 2019.

[7] LORD Sensing, "3DM-GX5-25 User Manual," 8500-0012 user manual, 2019.

[8] Texas Instruments, "MAX323x Dual EIA-232 Drivers/Receivers," SLLS047M datasheet, 2019.