Robottoman

A Technical Report submitted to the Department of Electrical and Computer Engineering

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Daniel Hanson

Spring, 2020.

Technical Project Team Members

Robert Fusek

Matthew McDonnell

Omid Khan

Zachary Struble

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Harry Powell, Department of Electrical and Computer Engineering

Statement of work:

I, Zachary Struble, was a leader in the software development for our project. I worked a lot on the Robottoman App. I designed the front end interface that the user controls the Robottoman with. The code was done on android studio using android phones. I also worked on the Bluetooth connection. I had to create a socket by which the cell phone could communicate with the HC-05 module. I also helped write the C code in code composer. This C code was what was on our MSP430 controlling which wheels were being powered. It used UART in order to take input from the HC-05 and outputted to various output pins.

I, Omid Khan, played as assist for software development in our project. I studied thoroughly on which aspects of bluetooth and its compatibility with available phones which resulted in me choosing the HC-05 chip and an Android phone for our project. During early development, I found and utilized a public app which verified our chip worked, before our personal app was created. I also worked the embedded code on the project. The embedded code revolved around receiving information from the HC-05 chip and powering pins based on it.

I, Daniel Hansom, assisted with the design of the PCB and helped debug the board. I researched and bought several key components, as well as working with Rob to make sure our board included all the necessary components needed for using spy-bi-wire to program the microcontroller. I helped solder components onto the board, and helped in testing it using a multimeter and Virtualbench. When things did not work as expected, I spent a significant amount of time reading the data sheets and becoming familiar with each component we were using so that I could inform my teammates so that we could all better troubleshoot the problems.

I, Matthew McDonnell, designed and built the chassis upon which the ottoman sat. I also chose the wheels and motors to ensure the Robottoman would move at the right speed. I also assisted the other team members in their tasks. I helped solder components onto the board. I helped test the board as well. On the software side, I helped code and test the code on both the launchpad and the PCB. During the design phase I helped choose which battery and microcontroller we would be using.

I, Robert Fusek, took the lead in the electrical engineering aspect of our project. My responsibilities included choosing hardware components, building PCBs, and debugging all hardware related problems. Specific examples of my tasks include choosing voltage regulators and motors based on the voltages and currents I calculated, soldering the components, connecting all hardware aspects of the project together (motors, MSP, PCB), and debugging issues which surfaced with our motors and spy-bi-wire.

Table of Contents

Table of Figures

Abstract

The Robottoman is a device to automate the process of getting up and manually moving an ottoman. It involved the design and construction of a wheeled ottoman that upon receiving signals from an android phone app, and moves accordingly. The use cases include both those who cannot physically move their ottoman, as well as those who value the convenience of avoiding the tedious task of retrieving the ottoman after sitting down. The core functionality of this project required mobile app development, printed circuit board design, embedded coding, and a general understanding of the fundamentals of electrical engineering.

Background

As people get older and their physical strength diminishes, it can become difficult for them to move furniture around their house. Our project would allow the elderly and infirmed a way to regain some of their autonomy in their own home. It would allow them to rearrange their living space as they see fit without having to rely on others to help them. Similar projects have been done at other universities, but no commercial product exists. Additionally, the other robottomans were not built to solve the problem statement we aim to. Our system will be implemented on an ottoman but will be designed such that it can be used on other furniture. A mobile app will be developed as well giving full remote control of the furniture movement in real-time. The movement will include multi directional capability which we deem as an essential part of our project, as most times many who want to move their furniture might not have the room or patience to navigate it through turning. This project will draw on experience from many previous courses. The PCB will use knowledge from Fundamentals I, II, and III. Embedded will help with programming our microcontroller. The app and it's interface with the Robottoman will be aided by our computer science curricula.

Constraints

Design Constraints

The primary use of an ottoman is a footrest, so when we chose the wheels and motors, we had to make sure that the Robottoman would be able to support the weight of someone's feet. We made sure to make it sturdier than necessary because some people are heavier than others and the Robottoman is for everyone.

During the construction of the chassis, we had to drill larger holes in the Mecanum wheels so that they would fit on the motors. As the holes were drilled, the 3mm set socket screws fell out and were lost, which prevented us from being able to secure the wheels in time for the demo. No hardware store in the Charlottesville area carries them. Glue was considered as an alternative; however, it was not used as it would not have set in time for the demo and there was no guarantee it would even work. Another constraint in the building of the chassis was the weight of the entire Robottoman. The Robottoman needed to be light enough so that the motors would be able to move it fast enough. Because of this we

chose a cloth ottoman that only weighed 3.3 pounds. We also chose a minimal design for the chassis to reduce extra weight due to wood while maintaining structural integrity.

To power the Robottoman we had to select the right motors that would move what we predicted the weight of the Robottoman to be so we chose 12V DC motors and a 12V battery to power the entire Robottoman. This ensured that the Robottoman would move at the desired speed of 1ft/sec.

Economic and Cost Constraints

During the planning phase, we considered ordering an extra set of wheels, but they were $96 for a set of 4, so we only purchased one set. This meant we did not have the right screws when we lost ours. Other than this minor issue, there were no cost constraints in the construction of the Robottoman. All the other parts were very affordable.

External Standards

The primary external industry standards that came into play during the course of our project were Bluetooth, ISO, and IPC standards. Bluetooth is a wireless technology operating on short-wavelength frequencies between 2.400 and 2.485 GHz range. The Institute of Electrical and Electronics Engineers (IEEE) first standardized bluetooth as IEEE 802.15.1, with the Bluetooth Special Interest Group (Bluetooth SIG) becoming the supervising standards organization for Bluetooth technologies. Bluetooth SIG is a not-for-profit corporation founded in 1998. Our project uses bluetooth to communicate between the Android mobile application and the HC-05 bluetooth enable breakout board. The HC-05 uses bluetooth V2.0 Serial Port Profile (SPP) protocol standards. This essentially replaces and RS-232 or UART serial communication interface. We used it as it easily integrates with our MSP430 device. It allows data to be sent just as though there were and RX and TX wire between the two devices, but with a much higher range (<100 meters). IPC is an acronym for the Institute for Printed Circuits, and even though this is still the name for the standards, the organization that publishes them is now called Association Connecting Electronics Industries. IPC is built from over 4000 member companies that design and construct printed circuit boards. The standards set out by the IPC were used by National Instruments software to assist in the design of our board in Ultiboard, as well as the company that manufactured them in that they automatically fixed some parts of the design to adhere to the industry standards. We used a 12 volt duracell lead acid battery which adheres to the International Standards Organization (ISO) 9001 standard for quality management systems. This essentially guarantees that our battery was manufactured in such a way that we can know it is reliable. That is, it satisfies the criteria of the ISO 9001 as confirmed by a third party company. This is important as we must be able to trust that our battery is safe and will perform as expected.

Tools Employed

We used Android studio/Java for the remote control app. We knew Java from our Computer Science classes, but had to learn Android Studio. For the embedded coding portion, we used Code Composer/C. We learned how to use these tools in our Introduction to Embedded Computing Systems class. We used Multisim/Ultiboard for the PCB design. We learned how to do this in the Electrical Engineering Fundamentals classes. To precisely cut wood into 10.75 in. x 10.75 in. and 11 in. x 11 in. squares, we used a water jet cutter. We had to learn this from our friend in the Mechanical Engineering department. To screw in motor mounts and chassis and drill through holes for wires we used a drill. We knew how to use a drill already.

Ethical, Social, and Economic Concerns

Environmental Impact

The first prototype of the Robottoman used a 12V battery with lead in it. Lead is toxic to humans so any Robottomans that need to be thrown away need to be disposed of in the proper methods for lead products. Another possible environmental issue is the chance of a fire. During construction when the terminals of the battery were accidentally connected, a wire caught fire. This is especially dangerous to the environment because of the lead in the battery as well.

Sustainability

We believe our project is sustainable. The materials used to create the frame of the project are based on wood and metal therefore our technology itself is made to last. A case where our technology would not be sustainable is if a potential customer has a habit of changing their furniture. The battery is also rechargeable, so users will not have to dispose of batteries each time the Robottoman inevitably loses charge.

Health and Safety

Another concern is the danger of Robottoman collisions, so the max speed was limited to keep Robottoman collisions from doing actual harm to people or things. We are also worried the user may lose connection to the Robottoman. In this case, we will have built in safeties that stop the Robottoman should the user lose connection. There will also be a lock button that the user can press to prevent the ottoman from being moved. We also took special considerations regarding the batteries to maintain safe operating conditions. We also took care to not purchase and use treated wood as the chemicals used to treat it are toxic when breathed in during the cutting and drilling process. When choosing screws and constructing the chassis, we made sure that none of the screws stuck out to prevent users from cutting themselves. The lead battery is also a health concern because lead is toxic to humans. Users of the Robottoman should check the battery before use.

Manufacturability

We see our project as very manufacturable. Since all of the hardware engineering of the Robottoman is incorporated in the chassis, the robottman could either be sold with installation already embedded into the furniture or as just the chassis.

Ethical Issues

We do not think the robottoman has any ethical issues attached to it as it does not take anyone's job away, does not cause harm, and is meant to solve an issue which many encounter frequently.

The only ethical issues that could come up would be found in the supply chain if the Robottoman is ever put into mass production. We would have to do our due diligence to make sure we do not exploit any underage or underpaid workers. We want the Robottoman to be a fully ethical.

Intellectual Property Issues

Patent USD350856S: Ottoman

This is the patent of an Ottoman. Our design differs from this because it is made of cloth, is a square, and has wheels. It also can be controlled from your phone. This patent is purely illustrational as it only shows a specific design of ottoman. The Robottoman differs not only in function but in appearance as well. This means that we differ enough to preserve patentability.

Patent US2008016502A1: Remote Control System

This is the patent of a remote control system using bluetooth. Our control system differs from this because it is done through a mobile phone instead of a built controller which is an integral claim to the patent. The design of the controller is illustrated in detail, but we use the phone, which means we adequately are different This also does not include the object being moved which separates the Robottoman from this prior art.

Patent US9215924B2: Mobile Furniture System

This patent is for a desk on casters. This enables the user to more easily move the desk. This differs from the Robottoman system because it is a desk instead of an ottoman. Also, the Robottoman is remote controlled and utilizes Mecanum wheels. This patent claims that the “work surface” is integral to the mobile furniture system. Since the Robottoman does not have a work surface, it is separate from this prior art.

If the Robottoman were to go to market, it would be patentable as there are no patents that exist for automated or remote control furniture. What separates the Robottoman from all the patents is the unique combination of remote controls, ottomans, and wheels.

Detailed Technical Description of Project

Software Design Description

The code for the Robbotoman is designed so that the user is manipulating which motors are being powered in which direction. In order to do that there are 3 primary sections for the software development for the Robottoman. The first is the App user interface on the Android phone. The second is the bluetooth code for connecting and sending bits to the HC-05. And the final section of software is the C code on the MSP430.

The user interface design for the Robottoman app was a simple controller the user could employ to move the Robottoman around his home. Here is a diagram of the user interface below.

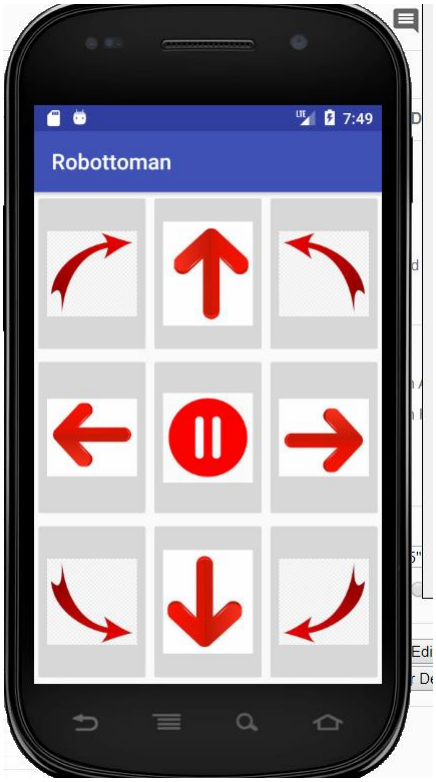


Figure 1 App Front End

The app was designed to allow the user to send 7 different commands to the Robottoman. One command was to stop all of the wheels. This was the pause button at the center of the screen. The four cardinal directions were also represented by the four arrows pointing up down left and right. In the four corners are arrows pointing in circular directions. These are for rotating the Robottoman. The Robottoman can be rotated either clockwise or counter clockwise. The top right and bottom left buttons both send the Robottoman clockwise and the Top Right and Bottom Left buttons both send the project counter clockwise. We chose to have two buttons for each rotation for visual appeal and flexibility for

our users. We wanted our display to be symmetrical and having multiple buttons also gives the user a choice. They can hit whichever one is easier for them to reach.

When the app is powered up the Phone must connect to the HC-05 module to allow a bluetooth connection between the phone and the Robottoman. The HC-05 is a very convenient module in that it takes in bits and automatically forwards it out of its TXD output pin. The bluetooth connection is established by pairing the Android phone with the device in the settings screen. The device comes up as HC-05. Although the user is able to rename it. Once the connection is established the Robottoman app can be activated and used. Upon creation the app runs code to create a socket with the HC-05. A block diagram of the process is shown below:

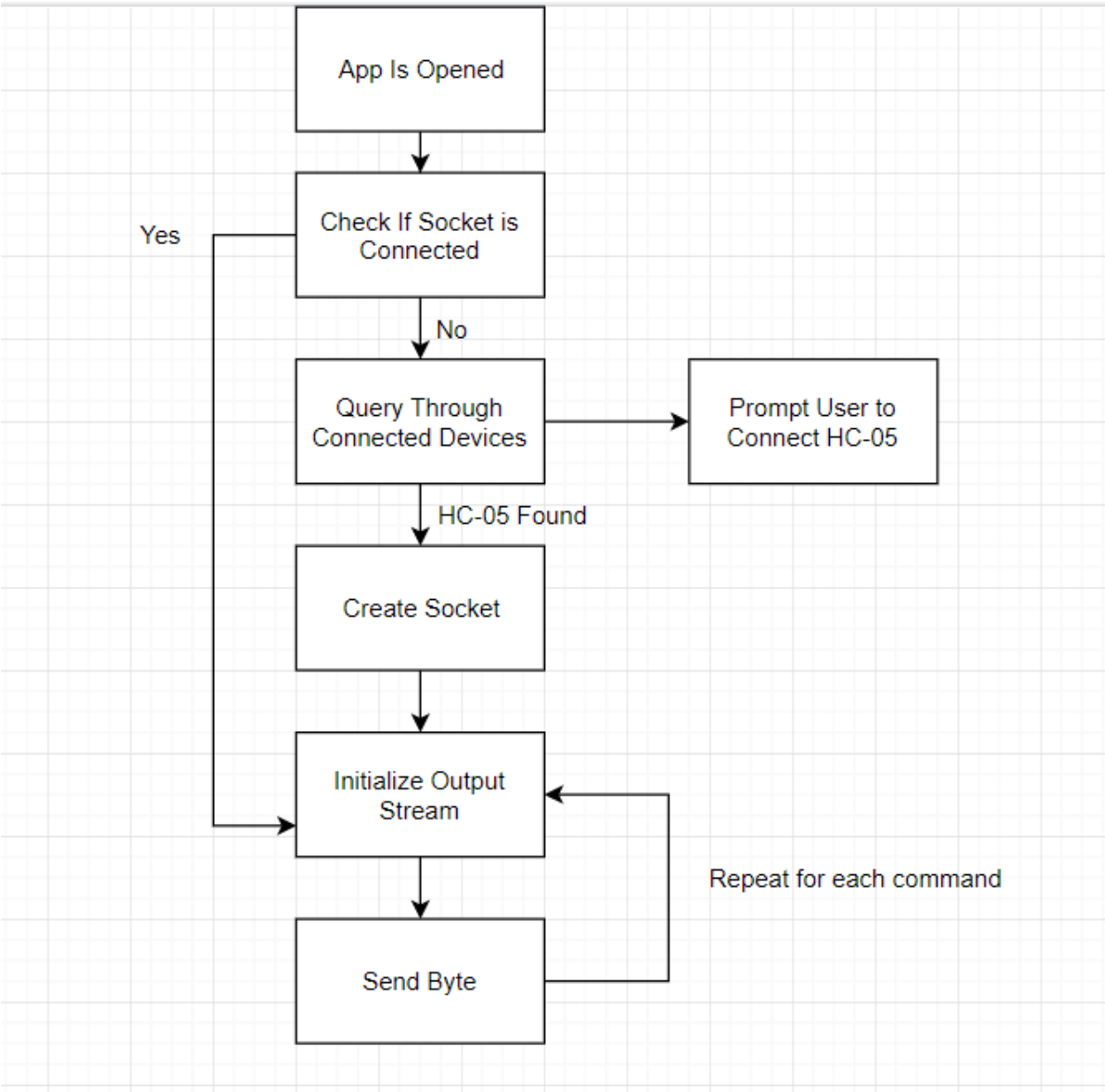


Figure 2 App Logic Flowchart

The first thing that the bluetooth code does is check to make sure that the socket is not already open and connected. If a second socket is opened then it will not be sending data over the correct socket to the HC-05. Once that check is complete the app then sets up bluetooth parameters. Accessing built in android libraries for sending and receiving data as well as interacting with the phone's list of paired devices. The list of devices paired with the Android is now traveresed, and if the HC-05 is found on there, the socket creation code will run. To create the socket we use the HC-05 UUID (Unique User Identifier) Sending out a socket with this identification number establishes a connection with our specific HC-05. After the socket is made we initialize an Output stream from the socket. Once the socket is connected and we have established an output stream we can send any byte that we want. Now the application waits for the user to press a button. Once a button is pressed a button specific function is called and a byte corresponding to that direction is sent out. A table displaying all of the different buttons and their corresponding directions and bytes is shown below.

Button	**Command**	**Byte To Send**
Top Left	**Rotate Clockwise**	**5**
Top Middle	**Forward**	**1**
Top Right	**Rotate Counter Clockwise**	**6**
Middle Left	**Left**	**3**
Middle Middle	**Stop**	**7**
Middle Right	**Right**	**4**
Bottom Left	**Rotate Counter Clockwise**	**6**
Bottom Middle	**Backward**	**2**
Bottom Right	**Rotate Clockwise**	**5**

Once the byte is sent it is picked up by the HC-05. The HC-05 is connected to the MSP430 by 4 pins. VCC for power, Ground for Ground, TXD for sending data to the MSP, and RXD for receiving data from the msp. A picture of the HC-05 to MSP430 is shown below.

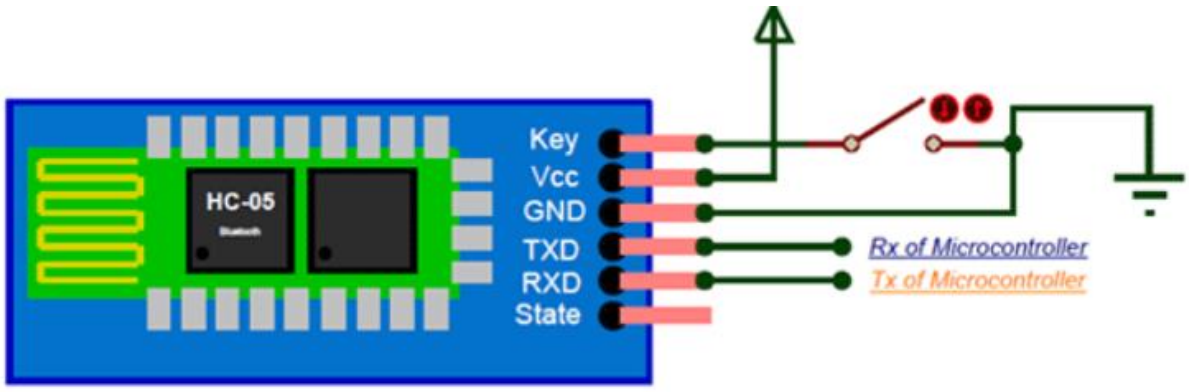


Figure 3 HC-05 Diagram

The VCC HC-05 wire is connected to pin 1 on the msp430. This gives 3.3V to the HC-05, powering it. The Ground pin connects to pin 20 on the HC-05. This is the msp430's ground pin. So they share the same voltage source and ground. The TXD connects to pin 3 on the msp430 and the RXD connects to pin 4. These are the UART inputs. Pins commonly used when the msp430 is receiving input that is more complicated than simply high to low. The HC-05 automatically forwards the byte it receives from the Robottoman App so there is no code to be written on the module.

Once the HC-05 sends the byte over the UART then the msp40 interprets it. The logic of the C code is displayed on this block diagram below.

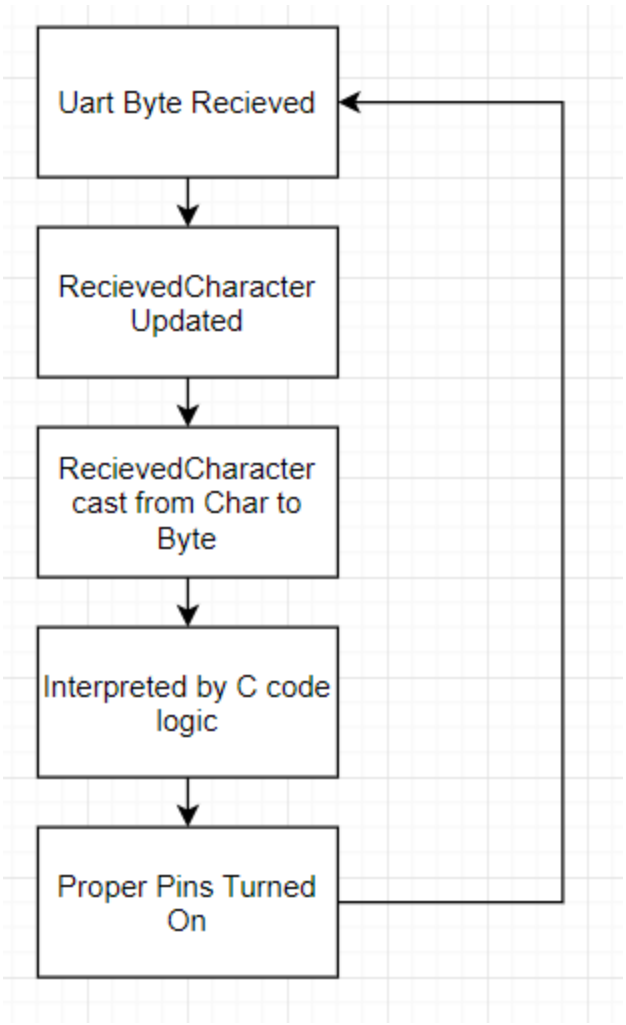


Figure 4 MSP430 Logic Flowchart

As shown above, the input to pin three triggers an interrupt and the value of char RecievedCharacter is changed. When RecievedCharacter is changed code runs that interprets RecievedCharacter to power the proper pins for the direction we want the robottoman to go. There are 10 pins that determine how the Robottoman will move. They are listed in the table below.

Command	Pin Number	Pin Name
Front Left Forward	2	A1
Front Left Back	5	A2

Front Right Forward	6	B1
Front RIght Back	7	B2
Standby	8	Standby
Back Left Forward	9	C1
Back Left Back	10	C2
PWM	11	PWM
Back Right Forward	14	D1
Back Right Back	15	PWn

The A1-D1 and A2-D2 pins send the A-D wheels either forward or backwards. PIn 8 is the standby pin. This pin connects to all four motors and must be constantly high for the motors to run. The final pin is P11 which is the PWM. The PWM controls the current that the motor drivers will send out by altering its duty cycle. We are doing a duty cycle of 100% so our PWM is always high. Here is a diagram of which pins are becoming high and low for each direction.

Command	**A1**	**A2**	**B1**	**B2**	**C1**	**C2**	**D2**	**D1**	**Stand by**	**PWN**
Forward	**HIGH**	**LOW**	**HIGH**	**LOW**	**HIGH**	**LOW**	**HIGH**	**LOW**	**HIGH**	**HIGH**
Back	**LOW**	**HIGH**	**LOW**	**HIGH**	**HIGH**	**HIGH**	**LOW**	**HIGH**	**HIGH**	**HIGH**
Left	**HIGH**	**LOW**	**LOW**	**HIGH**	**HIGH**	**LOW**	**LOW**	**HIGH**	**HIGH**	**HIGH**
RiIght	**LOW**	**HIGH**	**HIGH**	**LOW**	**LOW**	**HIGH**	**HIGH**	**LOW**	**HIGH**	**HIGH**
CW	**HIGH**	**LOW**	**LOW**	**HIGH**	**LOW**	**LOW**	**LOW**	**LOW**	**HIGH**	**HIGH**

CCW	**LOW**	**HIGH**	**HIGH**	**LOW**	**LOW**	**LOW**	**LOW**	**LOW**	**HIGH**	**HIGH**
STOP	**LOW**	**LOW**	**LOW**	**LOW**	**LOW**	**LOW**	**LOW**	**LOW**	**HIGH**	**HIGH**

Once the proppers pins are high the motor drivers take over. This sums up the software present on the Robottoman. Below is a flowchart of the entire software design.

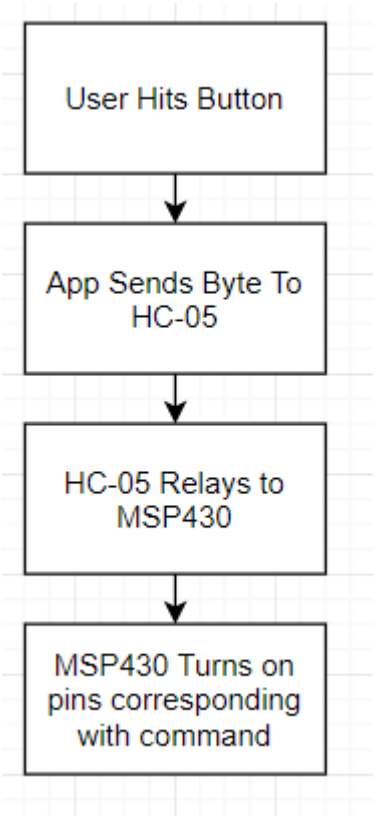


Figure 5 WHole Software Logic Block Diagram

Hardware Design Description

The main electrical Engineering component of this project was to build a PCB that can deliver power to our HC-05 module, our MSP430, and our motors, all while successfully transferring signals between them. Our multisim schematic is shown below:

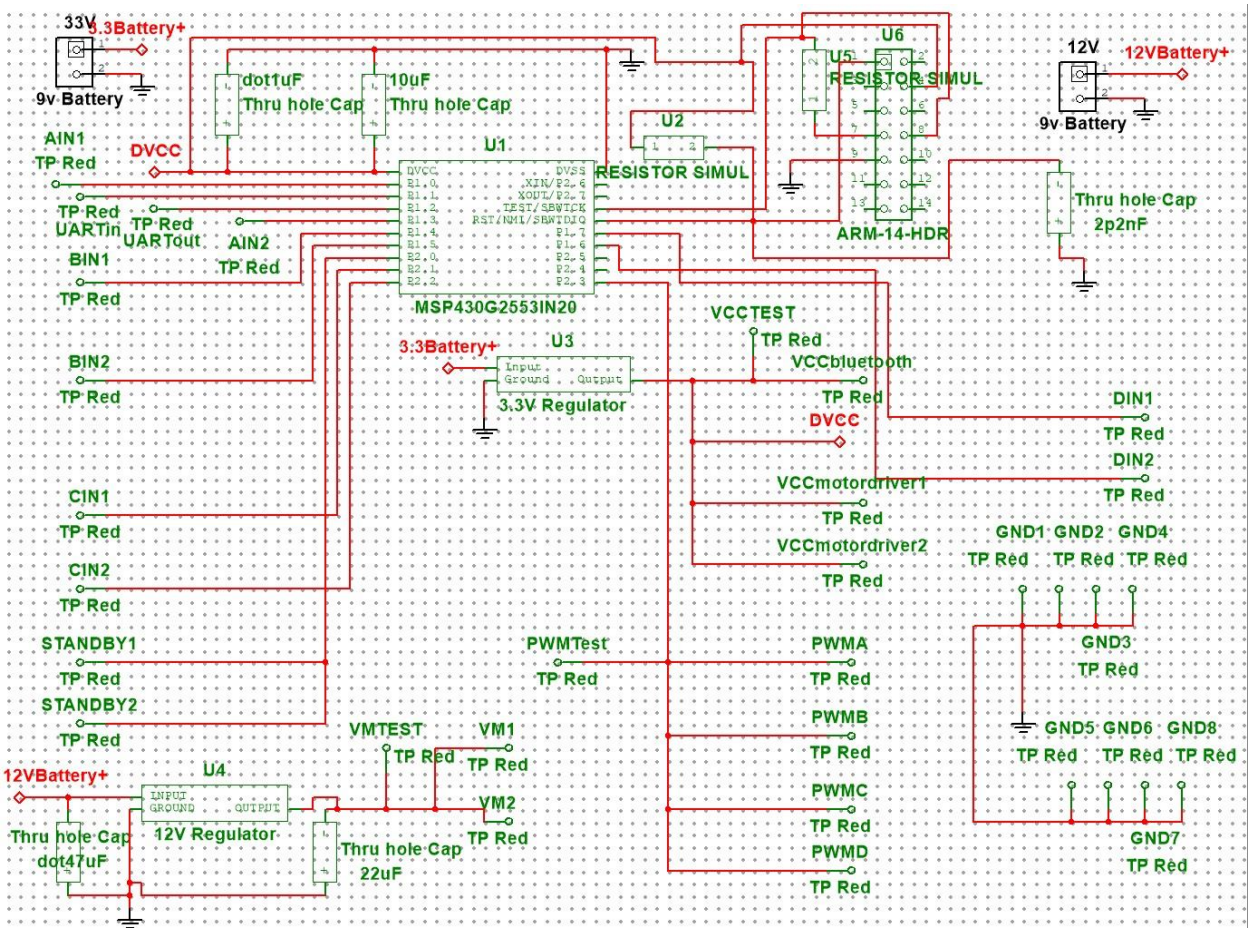


Figure 6 Multisim Schematic

This schematic shows all of the components we needed to use to make the Robottoman work. It all starts with the power supply. Our MSP430 and our HC-05 need 3.3V to safely operate, so we needed a voltage regulator to supply that consistent power. The 3.3V battery goes through U3, the 3.3V regulator in order to power VCCbluetooth and DVCC. The 3.3V from the regulator also go to our motor drivers. That power is being transferred via external connections using female to female wires off of VCCmotordriver1 and VCCmotordriver2.

The MSP430 is U1 in this schematic. It is the centerpiece of our board, as it is providing power to our motor drivers inputs to control the motors. The output pins used to control the motors are AIN1, AIN2, BIN1, BIN2, CIN1, CIN2, DIN1, DIN2, PWMA, PWMB, PWMC, PWMD, STANDBY1, and STANDBY2. There are four different letter input one and input twos because the voltage difference across each of those controls each motor. A is the front left motor, B is the front right motor, C is the back left motor and D is the Back right motor. There is a different PWM for every wheel so that we can control speed. For this project we are just setting all of the duty cycles to 100% so that they all go the same speed. The Standby simply allows the motor driver to function, so there needs to be two of them because we have two separate motor driver chips.Below is a diagram of our motor drivers.

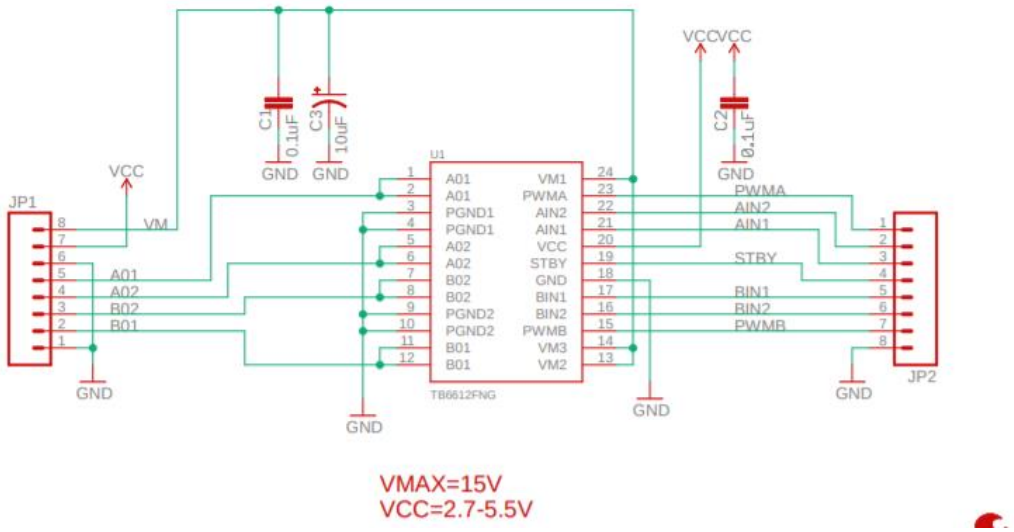


Figure 7 Motor Driver Schematic

These motor drivers have the capability to power two wheels each. We have one motor driver powering wheels A and B, and ne motor driver powering wheels C and D. The max voltage to put into the motor driver is 15 V, but we are using 12V. The VCC range is 2.7V o 5.5V, but since most components are powered to 3.3v and that is inside the motor drivers operating range we went with that. The motor driver works by creating a voltage difference across AO1 and AO2 that is proportional to the voltage difference across AIN1 and AIN2. SO for example, if the front left wheel, wheel A, should be going forward. The AIN1 pin should be powered to 3.3V and the AIN2 pin should read 0V. THis creates a 3.3V voltage difference across AIN1 and AIN2. Since the PWMA should be high at 3.3V and the STandby should be high at 3.3V the AO1 and AO2 should have a 12V difference across them. AO1 is then connected to the positive lead of our motor while AO2 is connected to the negative lead of our wire. We then hook up the BOUT pins to motor B and do the same for the C and D pins on the second motor driver. Once everything is hooked up we can properly turn our motors on and off in the desired direction with the simple touch of a button.

The next schematic is our ultiboard design. On the left side of our board are our power sources. We wanted them on the same side so that the batteries could be secured near each other on the chassis. In between the two battery inputs are our HC-05 pins. These ins provide power to the HC-05 and also connect the RXD and TXD pins to the MSP430's UART inputs. Across the bottom of the board are our A1 through C2 pins. These pins come directly from our MSP430 and are all male test pins to connect to our motor drivers through female to female wires. Te DIN1 and DIN2 are all the way on the other side of the board unfortunately because we ran out of pins on the first side. We were using 16 of the 20 MSP430 pins so we had to utilize both sides.

We had plenty of extra space on our board and didn't have to make many if any design trade offs. If anything we could have made our board smaller and more compact, but that would have just made soldering and debugging a lot harder.

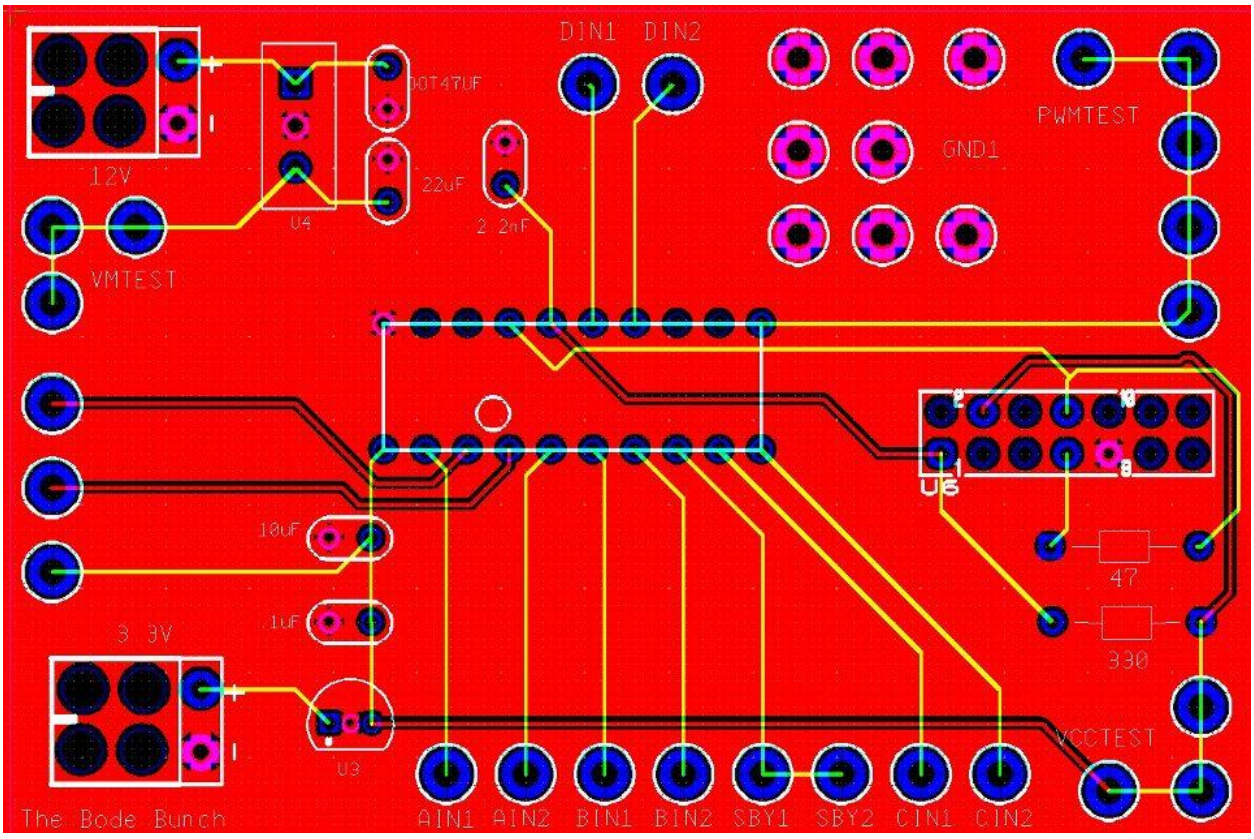


Figure 8 Ultiboard Schematic

The following picture is our physical PCB board. Our spacious design allowed for a relatively smooth soldering process. One change that we needed to make when actually constructing our board was to not use the 12V voltage regulator. The 12V voltage regulator was supposed to provide 12V voltage for our motor drivers to send out to the motors. We did not use this regulator because the motor driver chips already had voltage regulators. Instead of using that battery input we just added on additional wires from the battery to the motor drivers voltage regulator. This was actually great news because we could reduce the amount of batteries we needed from two to one. This created fewer items cluttering our chassis. We now had plenty of space.

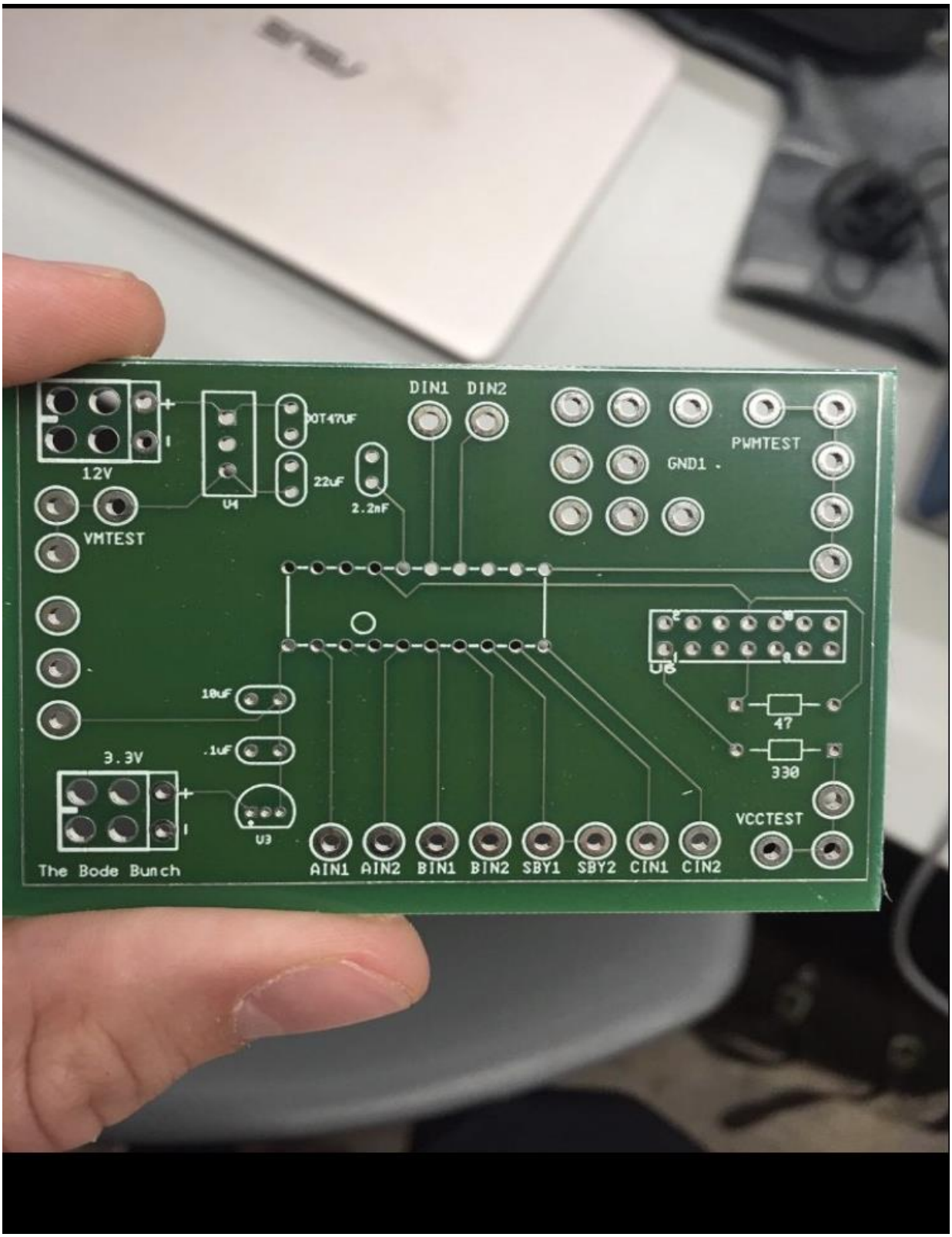


Figure 9 PCB Picture

Our next image is a mapping of the mecanum wheel directions. The mecanum wheels are designed so that the Robottoman can move in multiple directions without having to turn the wheels. This is done by powering different wheels either forwards or backwards to send the Robottoman in the desired direction. The mecanum wheels are able to do this because the spokes are at 45 degree angles from each other. This was a very cool and convenient way to have the Robottoman be able to get anywhere in a user's living room. It moves left by having the right most wheels spin in opposite directions facing outwards while the leftmost wheels spin in opposite directions going inwards. For or rotations we elected to use the turn of rear access, but place the access on the front of the Robottoman. We chose this because it would provide the most intuitive way to turn since the average user is driving the Robottoman towards them, rather than away from them as is the case for a typical remote controlled rolling device.

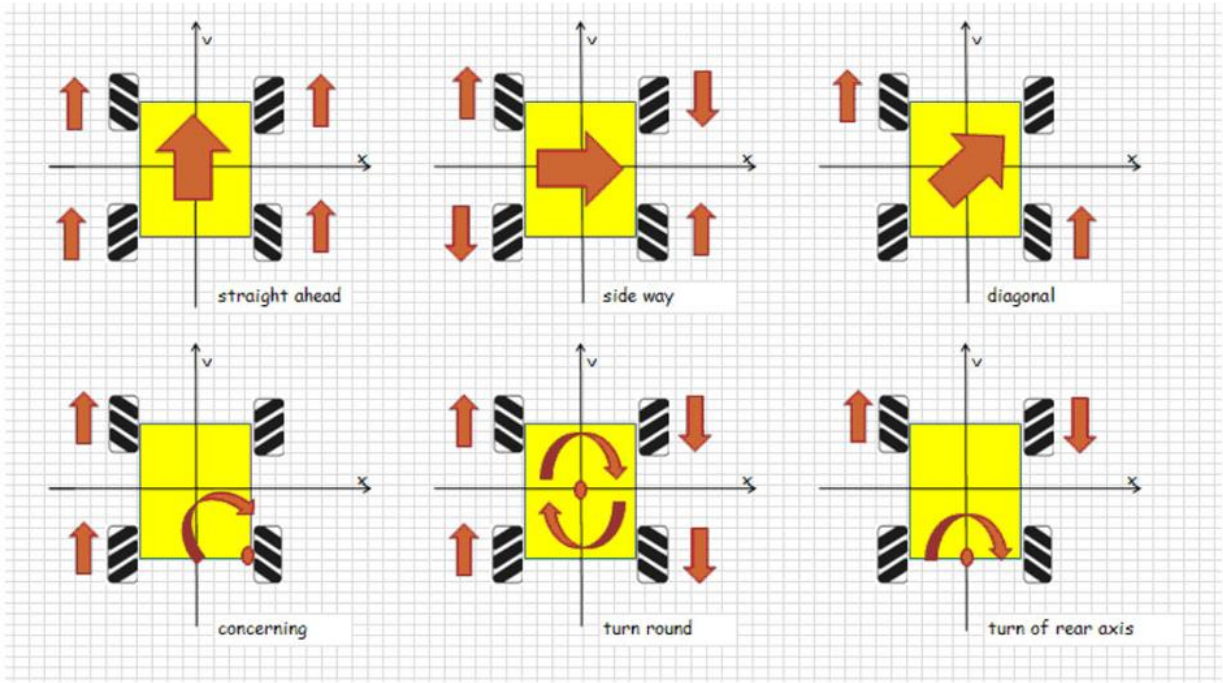


Figure 10 Mecanum Wheel Directions

In order to attach the Robottoman to the chassis we came up with a clever design. The ottoman we chose has a hollow inside with a lid that can be taken off to reveal an inside compartment. Our ottoman is shown in the picture below.

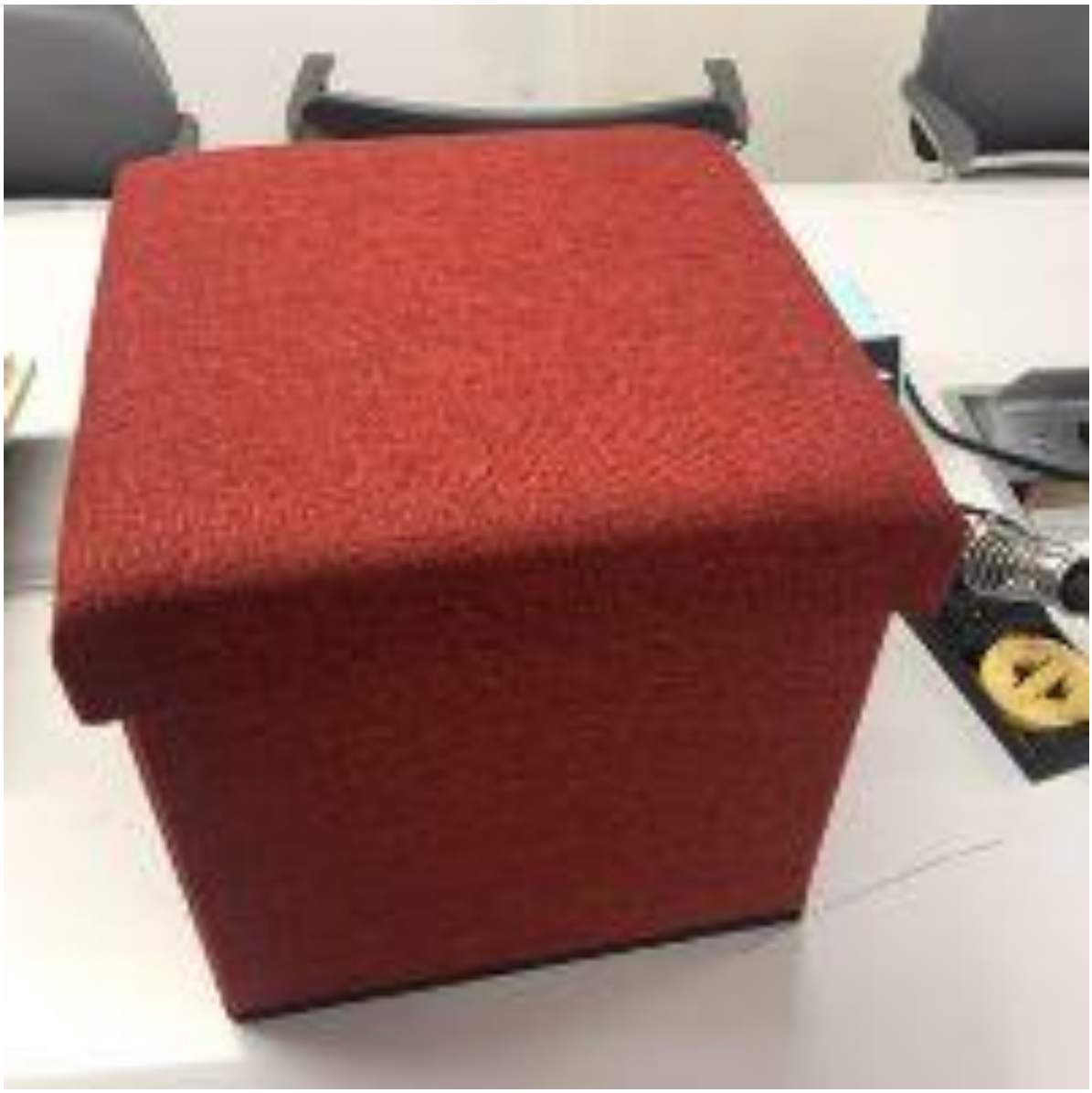


Figure 11 Ottoman

Since the lid fits perfectly on the bottom side as well we flipped the ottoman over and placed the lid on the now top facing bottom side. Since there is now no bottom the ottoman can be placed on top of the chassis to contain all of the electronics within the ottoman. The chassis we constructed is two wooden squares drilled together. The top piece of wood is slightly smaller than the bottom piece so that the base of the ottoman perfectly first around the top piece. This secures the Robottoman onto the chassis while keeping its original form. We can also take the ottoman off of the chassis with relative ease, allowing for a change of batteries or for an official Robottoman repair man to have easy access to the circuitry. The chassis overlapping wood design is shown below.

Figure 12 Chassis Design

The DC motors are mounted on the base of the Robottoman and the wires from the motor drivers go through small holes drilled in the center of the chassis. This protects the rest of the circuitry while connecting the motors to their sources of power.

The wheels are placed directly on the motors. When the user presses a command that activates the motor, it spins and the wheels spin with it. The mecanum wheels that we ordered were only one sixth of a centimeter in diameter, but our motors were one third of an inch in diameter. We had to drill new holes where the one sixth holes were to allow the motors to fit through the wheels. The picture below is the mounting position of our motors.

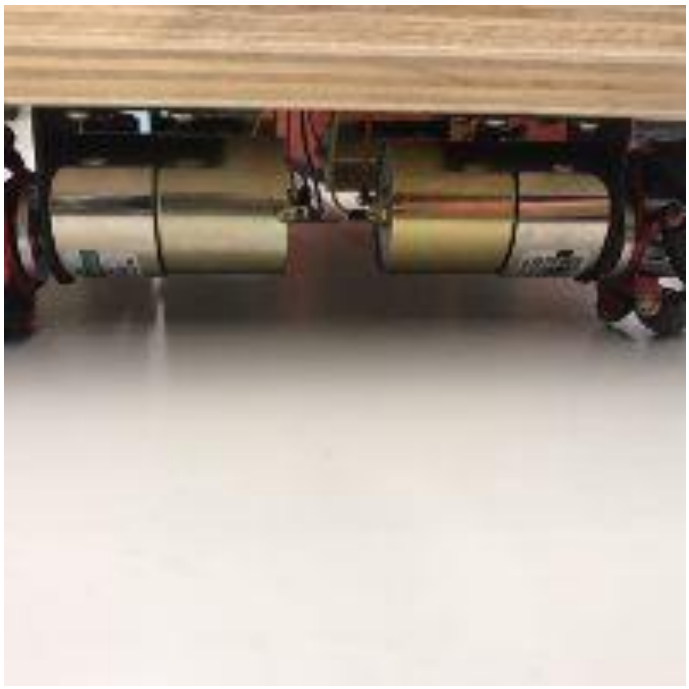


Figure 13 Motor Mounts

The next figure is a side view of the positioning of the mecanum wheels on the chassis.

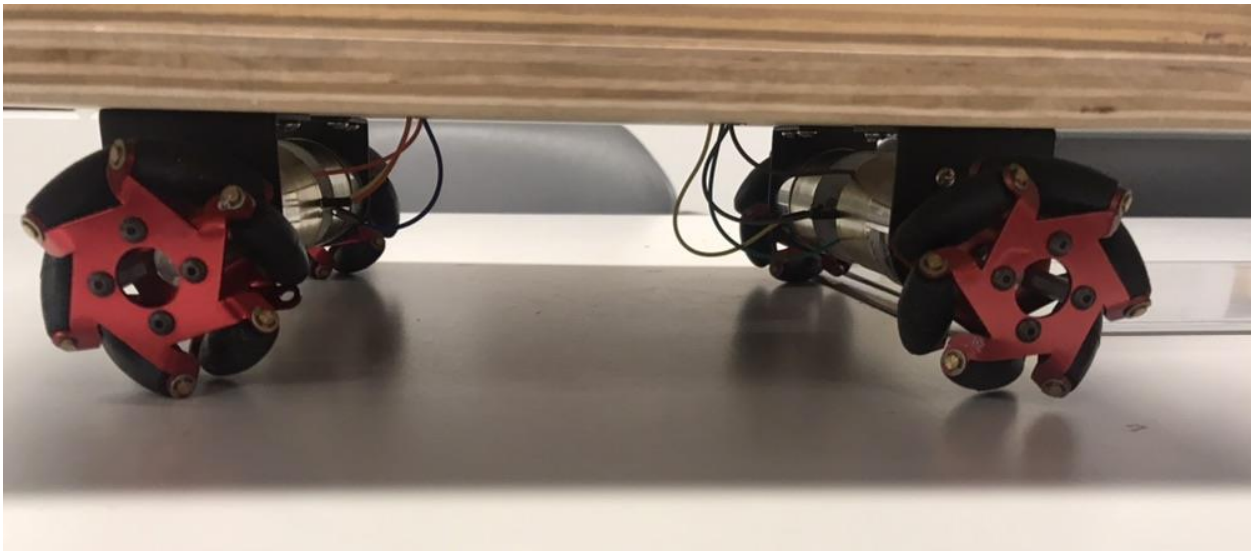
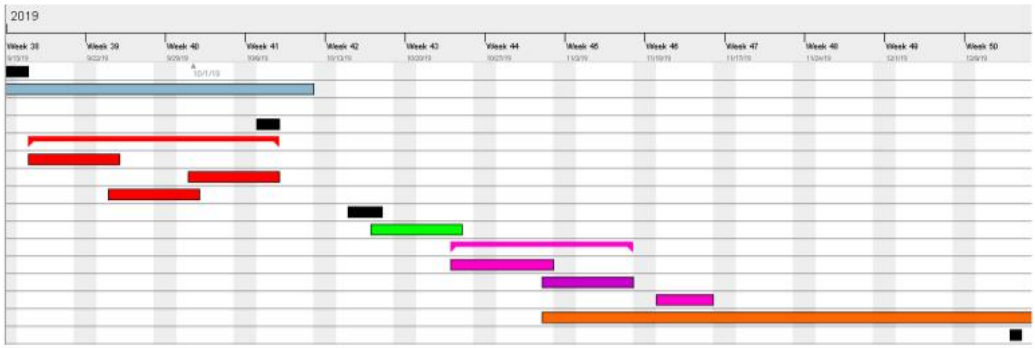


Figure 14 Wheel Positions

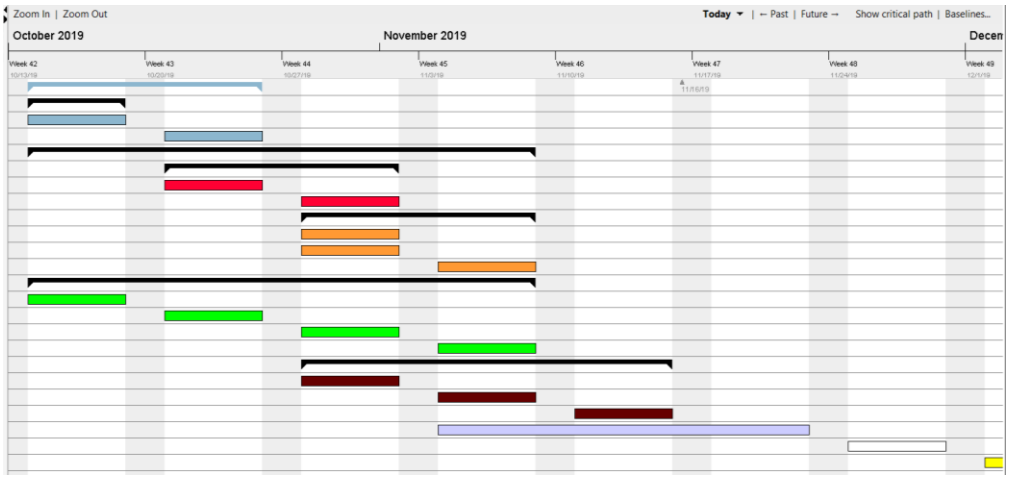
Project Timeline

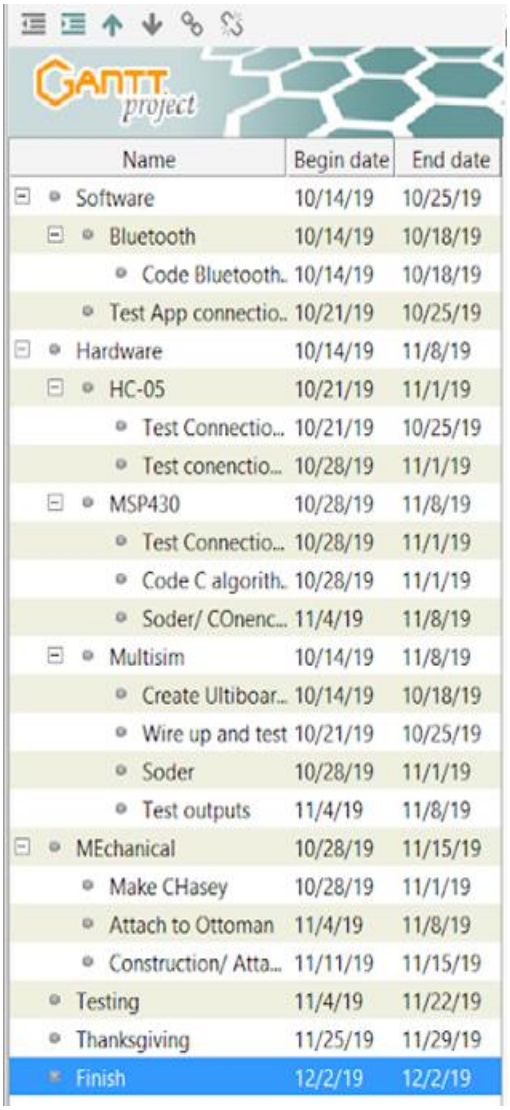


Name	Begin date	End date
Project Proposal	9/12/19	9/16/19
Research	9/12/19	10/11/19
Poster presentation	9/12/19	9/12/19
Reading Days	10/7/19	10/8/19
Software Design	9/17/19	10/8/19
Set up Application UI	9/17/19	9/24/19
MSP430 Programming	10/1/19	10/8/19
Set up bluetooth chip	9/24/19	10/1/19
Midterm Design Review	10/15/19	10/17/19
Header Board	10/17/19	10/24/19
Build	10/24/19	11/8/19
Mobility Design	10/24/19	11/1/19
Connect Power Source to W...	11/1/19	11/8/19
Mechanical Testing	11/11/19	11/15/19
Integration and testing	11/1/19	12/13/19
Final Project Demonstration	12/12/19	12/12/19

Figure 15 Original Gant

Above is our original Gantt chart.There several issues with this original plan. The first problem was a lack of overlapping processes. We needed to have components of hardware in order to properly test our code and it was overall not efficient to not be focussing on the big picture while working. Also our chart did not have anywhere near enough detail to properly guide us through the process of successfully designing building documenting and demonstrating the Robottoman. Below is the Gant chart we later created to properly map out creating the Robottoman.





Name	Begin date	End date
Software	10/14/19	10/25/19
Bluetooth	10/14/19	10/18/19
Code Bluetooth..	10/14/19	10/18/19
Test App connectio..	10/21/19	10/25/19
Hardware	10/14/19	11/8/19
HC-05	10/21/19	11/1/19
Test Connectio...	10/21/19	10/25/19
Test conenctio...	10/28/19	11/1/19
MSP430	10/28/19	11/8/19
Test Connectio...	10/28/19	11/1/19
Code C algorith..	10/28/19	11/1/19
Soder/ COnenc...	11/4/19	11/8/19
Multisim	10/14/19	11/8/19
Create Ultiboar...	10/14/19	10/18/19
Wire up and test	10/21/19	10/25/19
Soder	10/28/19	11/1/19
Test outputs	11/4/19	11/8/19
MEchanical	10/28/19	11/15/19
Make CHasey	10/28/19	11/1/19
Attach to Ottoman	11/4/19	11/8/19
Construction/ Atta...	11/11/19	11/15/19
Testing	11/4/19	11/22/19
Thanksgiving	11/25/19	11/29/19
Finish	12/2/19	12/2/19

Figure 16 Updated Gant

The new gantt chart provided a much better parallelization of tasks. We fell behind early in the semester so we had to push things further into the fall than we originally wanted to. However, we were able to stay faithful to this chart.

We, the Robottoman design team, were able to effectively distribute tasks in order to make sure there were enough people working on different things and everyone was playing to their strengths. Zach was the lead software developer throughout the project. He was assisted in writing code by Matt and Omid. Omid took a leadership role in designing the physical bluetooth components he was assisted by Zach and Daniel. Robert was the chief electrical engineer and took a massive role in board design and soldering. He was assisted by Daniel, Matt, and Zach. Matt took a leadership role in parts ordering and chassis design. He was assisted by Omid and Rob.

Overall our team was able to work together and stick together to successfully design and assemble the Robottoman. We are very happy to have worked on this fun and interesting project.

Test Plan

Our testing plan was robust and comprehensive. We methodically worked our way down the Robottoman's phone to wheels pipeline to make sure everything was working properly. In order to properly perform useful test on the Robottoman we had to break everything down into four testing modules.

The first testing module was the Android phone. The Android code stored on that phone was required to do 3 things. Those things were to connect to the HC-05 bluetooth module, display our front end design to our users, and to properly send a bit to the HC-05. For the first two requirements, connection and display, the testing was pretty straight forward. To test that the front end display was working we just opened up the app and looked at it. Since we saw the design on the phone screen we knew we had succeeded at that step. For bluetooth connection we could test this by observing the blinking LED display of the HC-05. The HC-05 has four different light display options. The first if off. THis means that the HC-05 is off. The second is blinking rapidly, many times per second. This means that the device is powered up but is not connected to a cell phone. The third setting is blinking once. This means that the phone is paired to the HC-05, but no socket has been created yet. The final LED setting is double blinking. THi means that the socket has been established and the HC-05 is ready to receive/already is receiving data. In order to test this functionality we connected the HC-05 to pins on a launchpad. We opened our app and connected, watching the blinking LED on the HC-05 turn from rapid fire, to single blink, to double blink. The final phone based testing that we had to do was test that we can send a byte of data through the UART to our MSP430. For this we wrote a simple test.c program to load up onto the MSP-430 launchpad. The launchpad has built in LEDs that turn on if pin 2 or pin 14 is powered. In the C code I toggle the LED pins if we receive data from the UART. Once we saw the LEDs turning on and off when we hit buttons on the phone screen we knew we were sending data.

The second module to test in the Robottoman was the MSP430 operation and C code. The first thing we did was use the launchpad to make sure that we knew what voltage would be enough to power the

msp. It was 3.3V. Next since we already were sending data successfully we had to write code to interpret those bytes. We tested this by using the Code composer debugger. We set breakpoints in the program to pause the run when data was received, and we could then see on the code composer debugger screen what the received character variable was equal to. This way we know that the byte the HC-05 is sending to the msp430's UART pins was being properly interpreted for our design. The next step after checking the variable values on the debugger is to check our msp430 output. We did this by using the digital multimeter on the output pins of the msp430. For each byte that was able to be sent we had to check every pin to make sure all of the pins that needed to be powered were powered and all of the pins that needed to not be powered were off. We needed to read 3.3V to prove a pin was on and 0V to prove a pin was off. We needed to make sure the active and inactive pins met the following table.

Command	A1	A2	B1	B2	C1	C2	D2	D1	Stand by	PWN
Forward	**HIGH**	**LOW**	**HIGH**	**LOW**	**HIGH**	**LOW**	**HIGH**	**LOW**	**HIGH**	**HIGH**
Back	**LOW**	**HIGH**	**LOW**	**HIGH**	**HIGH**	**HIGH**	**LOW**	**HIGH**	**HIGH**	**HIGH**
Left	**HIGH**	**LOW**	**LOW**	**HIGH**	**HIGH**	**LOW**	**LOW**	**HIGH**	**HIGH**	**HIGH**
RiIght	**LOW**	**HIGH**	**HIGH**	**LOW**	**LOW**	**HIGH**	**HIGH**	**LOW**	**HIGH**	**HIGH**
CW	**HIGH**	**LOW**	**LOW**	**HIGH**	**LOW**	**LOW**	**LOW**	**LOW**	**HIGH**	**HIGH**
CCW	**LOW**	**HIGH**	**HIGH**	**LOW**	**LOW**	**LOW**	**LOW**	**LOW**	**HIGH**	**HIGH**
STOP	**LOW**	**LOW**	**LOW**	**LOW**	**LOW**	**LOW**	**LOW**	**LOW**	**HIGH**	**HIGH**

The third testing zone was our PCB board. This was potentially the most important zone because it was all about powering everything and transferring the commands from the MSP430 to the motor drivers. Step one of our test plan for this section was to test that power was properly dissipated over our regulator. We were using a 12V battery to power our board, but we had to use a 3.3V source to power the HC-05, the MSP430 and the two motor driver chips. At first our regulator did not work properly. We were getting a 5V output on our test pin instead of a 3.3V output. After reviewing the datasheet we realized that the schematic was from the bottom view and not the top view. After we corrected that mistake we were able to get a steady 3.3V reading. Then we tested the power to the HC-05 and the MSP430. We tested these with the digital multimeter and got the proper 3.3V after fixing the regulator. Next we had to check that the MSP430 was being powered and turned on. To test this we loaded code

onto the MSP430 that would set all of the pins to high. We powered up out board and checked the pins. But everything was 0V. We figured out that our rest pin was grounded because of a misread schematic so after ungrounding it we were able to properly power the MSP430. We confirmed this be measuring the output voltage on the A1-D2 pins as well as Standby and PWM. We then put our Robottoman C file on the MSP430 and tested out the phone turning on various pins with various commands. The last part to test on our motors was our motor drivers. The motor drivers operate. When the wheel should be going forward there should be a 12V difference across Aout1 and Aout2, and while the wheel is going back there should be a negative 12V difference. We tested if according to the following chart.

Direction	A difference	B difference	C difference	D difference
Stop	0	0	0	0
Forwards	12	12	12	12
backwards	-12	-12	-12	-12
Right	12	-12	-12	12
Left	-12	12	12	-12
Clockwise	12	-12	0	0
CounterClockwise	-12	12	0	0

After the the motor drivers were giving us the right voltage value for each new command we knew the pcb was complete.

The final area to test was the chassis and motors section. The first thing to test was whether a 12V difference was abe to turn our motors. Fortunately it was and our motors would spin in the proper directions for each command. Next we had to test the weight bearing of the wooden chassis. We made sure that the wheels could support the weight of the all of our electronics. Once those two things were complete and all the wiring was done it was time for field testing. We then had to test moving the Robottoman in our 6 possible patterns.

That completes our testing plan that we executed. We felt that we had a very comprehensive testing process throughout our design lifecycle. This allowed us to be very sure of what we had working and what we did not have working. It also gave us very clear goals to achieve.

Final Results

In conclusion our project was able to accomplish the vast majority of goals we set out to achieve. The first major goal was to create an App user interface. That was completed very early on. The second goal was to be able to communicate with the bluetooth module with our app. We also completed that. Then our next goal was to write C code to interpret the data from the HC-05 and use it to execute logic controlled by the user. That was created as well. The fourth goal was to create a PCS board that can power out motor drivers, MSP430 and our HC-05, as well as connect these devices. That was also a resounding success. The final step was to wire up our motors, attach our wheels and be able to control the Robottoman. That is where we ran into two problems. The first was that our wheels were damaged when we drilled holes into them for the motors to go through. Several axles were bent and important screws for securing them to the mmotors were crushed or lost.This presented a huge challenge when it came to putting the wheels on our motors. We were able to slip them on but had no way to secure them. We were not able to find any m3 metric set screws, which was what we needed to secure the wheels. Because of this issue, we are able to flawlessly maneuver forwards and backwards, but our right and left functionality is eliminated until we can find these screws. We are still enabling a user to move and ottoman with his or her phone, but unfortunately not all of the directions are working yet. We are sad because all of our engineering design works and we can power the proper motors for left or right, but we ran out of time to secure the wheels before the capstone fair.

Below are our original grade expectations.

Letter Grade		Criteria
A	-	Device has the capability to move in the direction specified by user input
	-	Device can move based on remote input
	-	Device gets connected to remote input from external device
	-	Team sets up mobile app for external input
B	-	Device does not perform one of the tasks required for an “A”
C	-	Device does not perform two of the tasks required for an “A”
D	-	Device does not perform three of the tasks required for an “A”

We have set up our app, move the device based on remote input, connect to the device from a mobile phone and we can change the direction of the Robottoman with user input. We succeeded in all of our objectives so feel comfortable and confident in saying that we request an A grade for this section.

Costs

Here is a table of all of the parts we need to purchase for the Robottoman and how much they cost

Part	Cost
Wood	$6.00
12 V voltage regulator	$1.53
12 V Battery	$21.93
HC-05	$10.48
Pololu TB6612FNG x 2	$3.33
Mecanum wheels set	$96.00
DC motors x4	$12.02
PCB	$30.00
Msp430g2553 PDIP	$2.20
TOTAL	$173.49

These listed costs come out to $173.49. This cost can definitely come down when mass producing. Specific costs that would become less expensive are the PCB, wood, and batteries. We can also design our own mecanum wheels and develop our own motor driver and HC-05 chips that would become

substantially cheaper. Ideally we would reduce purchasing at consumer prices and could negotiate deals to get large quantities of our parts for a reduced cost.

Future Work

In this section you should offer suggestions as to how the project might be improved or expanded upon if a future group of students wished to create a new project based upon yours. You should consider difficulties that were not foreseen at the beginning, and offer advice on pitfalls to watch for.

Some enhancements which would improve our project would be adding suspension, making recharging the battery easier, and getting wheels which support a larger amount of weight. We believe that our Robottoman proves as a great proof of concept for automated furniture, but adding the three previously stated enhancements would bring it closer to a more viable solution. Suspension would improve the Robottoman in two ways: concealing the hardware under the furniture until it is called upon and providing assistance to situations where the furniture might not be on a flat surface. In our version of the Robottoman the battery is within the furniture and difficult to rewire correctly. Ideally the battery would be in an easy to reach place and easily detachable. The wheels we picked for our ottoman also supported enough weight for the scale of our project. If one were to recreate this project they would have to be liberal on the size of the wheels as one could not truly factor in the amount of weight on top of the furniture, as people in all sizes would use the device.

Difficulties which our project suffered from were mostly due to mechanical and electrical components. Sometimes some of our electrical components would “blow out” without us knowing so we would have to replace them after verifying they didn’t work with a multimeter. Another time our motors did not turn on without us letting it warm up first. These types of problems were easily solvable but hard to spot and therefore cost our team some time in debugging.

References

[1] “MSP430G2553 | MSP430G2x/i2x | MSP430 ultra-low-power MCUs | Description & parametrics.” [Online]. Available: http://www.ti.com/product/MSP430G2553. [Accessed: 16-Dec-2019].

[2] “Metal DC Geared Motor - 12V 100RPM 42kg.cm” [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/DFRobot%20PDFs/FIT0492-B_Web.pdf. [Accessed: 16-Dec-2019].

[3] “HC-05” [Online]. Available: http://www.electronicaestudio.com/docs/istd016A.pdf. [Accessed: 16-Dec-2019].

[4] “SparkFun Motor Driver - Dual TB6612FNG” [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/Sparkfun%20PDFs/ROB-14450_Web.pdf. [Accessed: 16-Dec-2019].

[5] “IPC--Association Connecting Electronics Industries”[Online]. Available: http://www.ipc.org/. [Accessed: 16-Dec-2019].

[6] “Bluetooth Technology Website”[Online]. Available: https://www.bluetooth.com/. [Accessed: 16-Dec-2019].

[7] “ISO 9000 Family”[Online]. Available: https://www.iso.org/iso-9001-quality-management.html. [Accessed: 16-Dec-2019].

Appendix

In this section you should include helpful information that does not fit into the above categories but will be helpful in understanding and assessing your work.

Figure 17 The Bode Bunch