



NORTHERN ARIZONA
UNIVERSITY

College of Engineering, Forestry & Natural Sciences

SCHOOL OF INFORMATICS, COMPUTING AND CYBER SYSTEMS



Augmented Mobility Project - User Manual



Table of contents

Table of contents	2
Introduction	4
Installation	5
Basics	5
Attaching the Wheelchair	5
Detaching the Wheelchair	5
Batteries	5
Arduino MEGA	6
Sensors	6
Head Switches	7
Joystick	8
Configuration and Use	9
Flow Chart	9
Modes/Using Inputs	9
Joystick	10
Force Feedback Joystick	11
Head Switches	11
Emergency Stop	11
Motor	11
Batteries	12
PVC Pipes	12
Maintenance	13
Batteries	13
PVC Pipes	13
Head Switches	13
Arduino Code	13
Troubleshooting	24
Wires	24
Force Feedback	24
Head Switches	24
One Faulty Head Switch	24
Two Faulty Head Switches	25



PVC	26
For Head Switches	26
Connections With Other PVC Pipes	26
Sensors	26
Wiring	26
False Zero Readings	26
Complete Sensor Failure	27
Batteries	27
Status of Planned Features	28
Kristen Shryack	28
Yutian Deng	29
Mohammed Alfatouh	29
Adam Douglass	30
Mengyuan Zhang	30
Conclusion	32
Appendices with Schematics	33
Flow Diagram	33
Various Relay Schematic	34



Introduction

Our power assist device focuses on bringing independent mobility for the first time to two students, one in middle school and one in high school through a power assist attachment that can be controlled by the student. Any wheelchair can lock into it. The project should be sufficient for future students as well as the two we are aware of. We expect the chair to be used as a training mechanism for assisted mobility. The electric chair is be able to move based on the child's input. Here are some aspects to our project:

- 3 different ways to drive the device
- 3 different types of sensors to guide the force feedback joystick
- Different safety features to protect the child/stop the device
- Modes to change input being used

The three inputs being utilized are a joystick, force feedback joystick, and head switches. A force feedback joystick is ideal in the learning aspect because it can offer quick forces in the opposite direction of obstacles. Head switches can be used by students who are currently familiar with the device or may not be able to use a force feedback joystick. The joystick is most commonly on current electric chairs, but it cannot give feedback. Moving to this input would be an important step for the child.

The project's sensors detect objects in the way and can help the child to avoid them. Inputs can be determined by switches on the back. When a certain combination is detected, it will determine what input will be used and the level of help that the chair will provide.

We are pleased that you have chosen Team Augmented Mobility Project for your students' needs. We understand that there is a strong need to have independent mobility, so this project should help students achieve that goal. We provided a device that is customizable based on each student's needs. The student may use a joystick or head switches to direct the device. The purpose of this user manual is to help you successfully use and maintain the project moving forward. Our aim is to make sure that you are able to benefit from our product for many years to come!



Installation

Basics

This device is not intended for the rain or any wet environment. It is also intended for environments that have wide areas for easy turning and movement to drive through. Although the device is heavy, it is resilient and can be moved without hesitation.

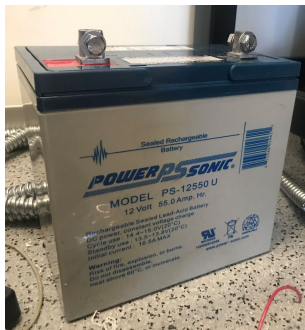
Attaching the Wheelchair

In order to attach a push wheelchair to the device, one must pull the wheelchair into the appropriate position for the wheelchair on the device (making sure that both wheels are inside of the metal grooves for them). Once it is in position, be sure that the wheelchair is rolled all the way back, as far as it can be rolled back. Once this is complete, there are two handles that are directly in front of the motor-powered wheels. These handles are rolled up inside of the handle holders. Take out one of the handles (the handles have hooks on the end). Attach the hook onto an appropriate part of the push wheelchair on the other side of the handle you are working with. Tighten the connection by pushing down on the metal plate on the back of the handle (not the one on the side). Repeat this process with the handle on the other side of the device.

Detaching the Wheelchair

When detaching the push wheelchair from the device, unhook the handles from the places you attached them onto the wheelchair. After this is complete, press the metal plate on the side of the handle's location on the base. The handles should roll up into their storage compartments. Then you will be free to move the push wheelchair off of the base.

Batteries



Be sure to charge the device's batteries by plugging the device's charging cable (located at the back of the device) into an ordinary wall socket. The device should be fully charged by the end of a day. For troubleshooting, please refer to the troubleshooting portion of the user manual.

Figure 1: Rechargeable Batteries



Arduino MEGA

The Arduino MEGA (shown on the right) can be considered the brain of the project. All processes and modes are determined through it. There should also be a motor driver shield (shown on the left) attached to it. They should always be secured together.

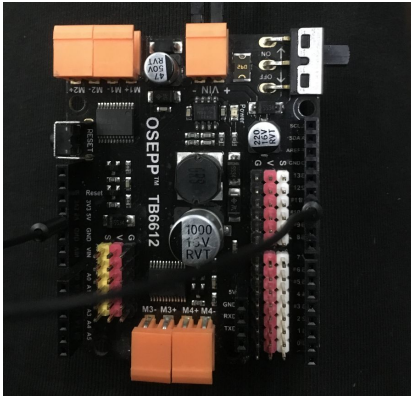


Figure 2: Motor Driver

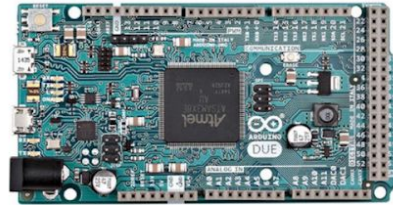


Figure 3: Arduino MEGA

Sensors

Sensors can stay in their current position at all times. If they need to be replaced, the infrared and bumper can be unscrewed and screwed back in. The ultrasonic sensors are attached with velcro. If one stops working, please visit the troubleshooting section. Sensors are located at the corners of the project and the bumper is at the front.






- ▲ Infrared Proximity Sensor
- Ultrasonic Sensor
- Bumper Sensor

Figure 4: Location of Sensors



Table 1: Type of Sensors

Sensors Image	Sensors Name	Product Description
	Infrared Sensor	Infrared proximity sensors work by sending out a beam of infrared light and detects waits to receive that infrared light, measuring the amount of time to attain a return signal.
	Sonar Sensor	Sonar sensors operate by sending out sound waves at high, unheard frequencies and measuring the amount of time it takes to get a return signal.
	Bumper Sensor	A bumper sensor works when an object is short and is not detected by the other sensors.

Head Switches



Figure 5: Head Switch

The head switches attach to the PVC pipes via velcro. This way they can be moved for the user. The PVC pipes also rotate up and down if height needs to be adjusted. They attach into an adapter near the the Arduino MEGA. The switch easily installs with the cord clicking into it and that cord attaching to the adapter.



Figure 6: Head Switch



Figure 7: Normal PVC Pipes



Figure 8: PVC Pipes Rotated Up

Joystick



The joystick should also not need any adjusting. As long as there are not any loose wires, it is installed to the device.

Figure 9: Joystick/Force Feedback Joystick



Configuration and Use

Flow Chart

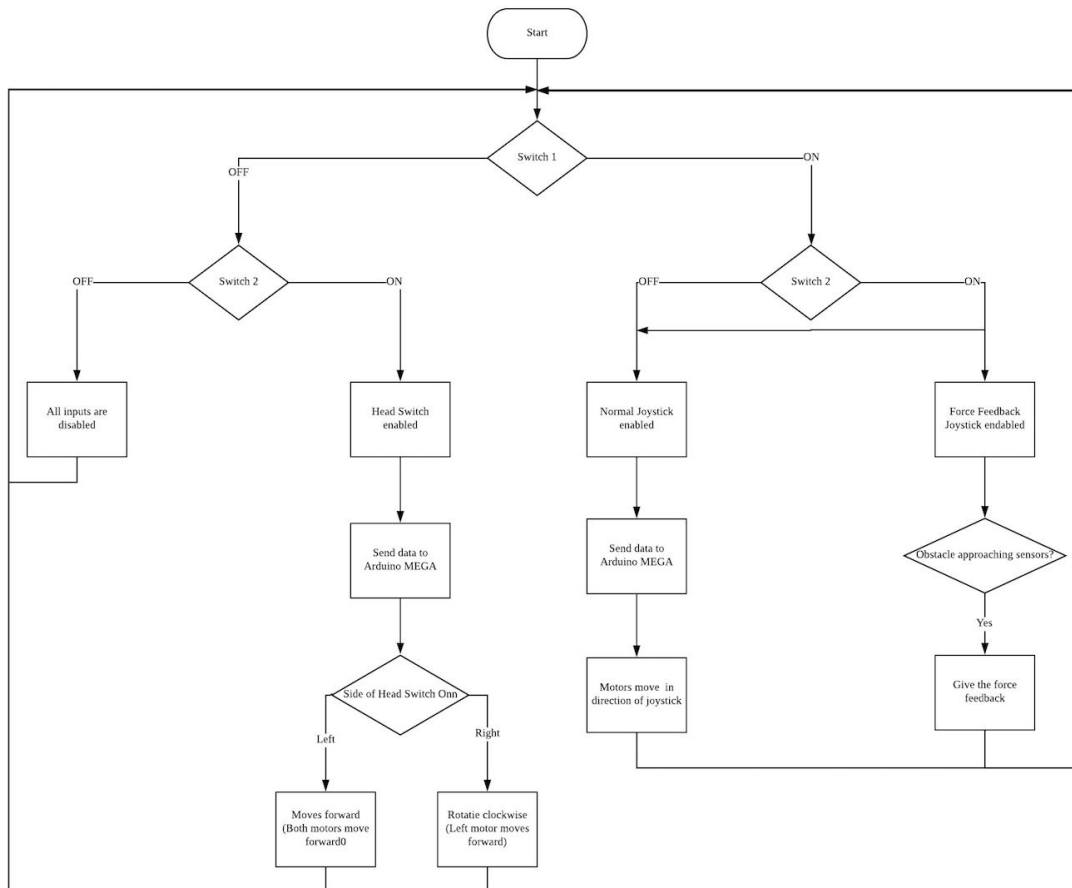


Figure 10: Flow Chart for the Four Modes

Modes/Using Inputs

For the switches located on the PVC pipe, the following table can help you determine which to turn on. The switch is OFF or '0' if it is flipped in the downwards direction. Attention that the left switch is the first '0' and right one is the second. The infrared and ultrasonic sensors are also disabled in any mode other than the force feedback mode. A quick reference can be found on the PVC pipe.



Table 2: Modes Control by Using Switches

Combination	Input
00	All inputs are disabled (OFF Mode)
01	Head switches enabled, Joystick is disabled
10	Joystick enabled, Head switches are disabled
11	Force feedback joystick enabled, Head switches are disabled



Figure 11: Switches on PVC pipes



Figure 12: Location for Switches

Joystick

To determine which mode enables the force feedback joystick as an input, visit the Modes/Using Inputs section. The joystick should be placed in the user's lap to drive the device. On the joystick, all buttons and throttles are disabled. For correct placement, the throttle should be on the left and closer to the user. The joystick can be moved in any direction to move the entire device in that direction.



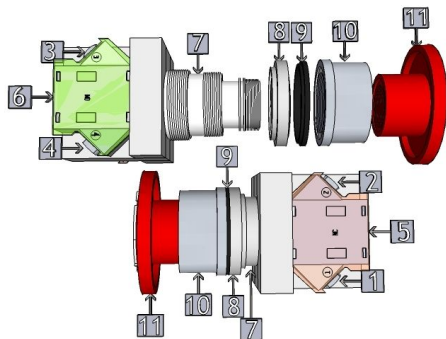
Force Feedback Joystick

To determine which mode enables the force feedback joystick as an input, visit the Modes/Using Inputs section. It provides force on the handle to quickly adjust the user away from the obstacle. Use the input exactly like the joystick mode, but expect slight guidance when there are close encounters to objects or people. There are three different types of sensors around the chair and base. The sensors will give the data to the Arduino to turn on the force feedback joystick.

Head Switches

To determine which mode enables head switches as an input, visit the Modes/Using Inputs section. While sitting in the device, the left head switch is used to move forward. The right one will move the device clockwise. Move the switch until it clicks to enable it as it only has an on/off state.

Emergency Stop



The emergency stop is on the tail end of the base. It provides a push down to stop the wheelchair when the device needs to be stopped. It will be locked when pushed down to keep the chair stationary. Rotate the switch to restore the device's normal behavior.

Figure 13 : Emergency Stop Structure Diagram

Motor

Various Relay and Motor Drivers are connected together to ensure the current is small enough to protect the board for two motors. The motors are in the base to control the two wheels on the base.



Batteries

Two 12V batteries are being used in the project. The batteries can be recharged several times. They are heavy and are inside of the chair base. A user can also change it, but be cautious of the electrical connections.

PVC Pipes

Velcro is attached to the front side of PVC pipes; this is being used for holding the head switches on each side. The height can also be changed by rotating the PVC upwards because we used the a unique method to connect them together. This method includes a cut slit for the screw to slide and lock into. Three PVC pipes on the bottom are supporting the rest of the PVC; these tie into the main metal frame of the project's base.



Figure 14: Side of PVC Pipes



Figure 15: Front of PVC Pipes



Maintenance

In order to operate the device, one must make sure that the emergency stop switch is not activated and that the mode switches are in the desired arrangement so that the mode of operation will match that which the user desires, whether that be the head switch mode, joystick mode, or forced-feedback joystick mode.

Before and after transporting the device, make sure that each component that is connected with velcro to the device (the housed wires, the sensors on the back of the machine, and the head switches) are connected to their appropriate velcro on the base itself.

Batteries

Batteries if being used consistently will last 2-3 years or 500-1000 charges and can be replaced by disconnecting the relay board and by pulling it out. Disconnection happens by removing the bolt that the wires are connected to and sliding the ring out. The device will need two batteries in order to operate. Be sure to reconnect all of the wires that are connected to the relay when installing a new battery.

PVC Pipes

PVC pipes can be replaced by unscrewing each area where the pipe meets metal. Similar designs can be replicated with 4 “L-shaped” pipe connectors, 2 “t-shaped” pipe connectors and varying pipe lengths. The PVC pipes are all 1.25” in width (except for the PVC couplers).

Head Switches

If head switches no longer work and need to be replaced, they can easily be removed from the adapter near the Arduino MEGA. Pull the grey cord from the black adapter and replace with a new head switch. (Please see troubleshooting for additional details)

Arduino Code

The following code is what has been uploaded to the Arduino MEGA. This code is not intended to be changed or altered. However, if a future team is to inherit this project, this is beneficial to them to get a deeper look into how the project works. This code was tested on on Arduino 1.8.5. This version should work as well as future ones. However if there are errors compiling, switch back to this version.

<https://www.arduino.cc/en/Main/Software>



//forward and back is blue(+5V),green(data),yellow(-5V) thin wires

//left and right are brown(+5V),red(data),orange(-5V) thin wires

//uses potentiometers

int green = 0;

int white = 0;

int PWM_1 = 6; //set trigger pins

int PWM_2 = 7;

int PWM_3 = 8;

int PWM_4 = 9;

int DIR_1 = 2;

int DIR_2 = 3;

int DIR_3 = 4;

int DIR_4 = 5;

#define Trig 22

#define Echo_1 23

#define Echo_2 24

#define Echo_3 25

#define Echo_4 26

#define InfraredPin_1 A15

#define InfraredPin_2 [A14](#)

#define InfraredPin_3 [A13](#)

#define InfraredPin_4 [A12](#)

const int moveright = A3;

const int moveforward = A4;

int modeSwitch_1 = A7;

int modeSwitch_2 = A8;

int emergencyswitchOn = A9;

float cm1, cm2, cm3, cm4; //distance

float temp_1, temp_2, temp_3, temp_4; //

uint16_t value_1 = analogRead (InfraredPin_1);

uint16_t value_2 = analogRead (InfraredPin_2);

uint16_t value_3 = analogRead (InfraredPin_3);

uint16_t value_4 = analogRead (InfraredPin_4);

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

pinMode(PWM_1, OUTPUT);

pinMode(PWM_2, OUTPUT);

pinMode(PWM_3, OUTPUT);

pinMode(PWM_4, OUTPUT);

pinMode(DIR_1, OUTPUT);

pinMode(DIR_2, OUTPUT);

pinMode(DIR_3, OUTPUT);

pinMode(DIR_4, OUTPUT);

pinMode(modeSwitch_1, INPUT);

pinMode(modeSwitch_2, INPUT);



```
pinMode(emergencyswitchOn, INPUT);
pinMode(50, OUTPUT);
pinMode(51, OUTPUT);
pinMode(52, OUTPUT);
pinMode(53, OUTPUT);
delay(50);
}

void loop() {
  // put your main code here, to run repeatedly:
  //joystick
  green = analogRead(A1) - 450;
  white = analogRead(A0) - 450;
  //green is y axis
  //white is x axis
  //int ON_1 = analogRead(emergencyswitchOn);
  int ON_2 = digitalRead(modeSwitch_1);
  int ON_3 = digitalRead(modeSwitch_2);
  delay(200);

  Serial.println(ON_2);
  Serial.println(ON_3);

  //Force feedback mode
  if (ON_2 == 1 && ON_3 == 1) //need correction
  {
    //ultrasonic
    digitalWrite(Trig, LOW);
    delayMicroseconds(2);
    digitalWrite(Trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trig, LOW);
    temp_1 = float(pulseIn(Echo_1, HIGH)); //time sound wave goes back
    temp_2 = float(pulseIn(Echo_2, HIGH));
    temp_3 = float(pulseIn(Echo_3, HIGH));
    temp_4 = float(pulseIn(Echo_4, HIGH));
    cm1 = (temp_1 * 17) / 1000; //distance
    cm2 = (temp_2 * 17) / 1000;
    cm3 = (temp_3 * 17) / 1000;
    cm4 = (temp_4 * 17) / 1000;
    Serial.print("Echo =");
    Serial.print(temp_1);
    Serial.print(" || Distance = ");
    Serial.print(cm1);
    Serial.println("cm");
    Serial.print("Echo =");
    Serial.print(temp_2);
    Serial.print(" || Distance = ");
```



```
Serial.print(cm2);
Serial.println("cm");
Serial.print("Echo =");
Serial.print(temp_3);
Serial.print(" || Distance = ");
Serial.print(cm3);
Serial.println("cm");
Serial.print("Echo =");
Serial.print(temp_4);
Serial.print(" || Distance = ");
Serial.print(cm1);
Serial.println("cm4");
DIR_1 = HIGH;
DIR_2 = HIGH;
DIR_3 = HIGH;
DIR_4 = HIGH;

//infrared
if (value_1 < 30)
{
  value_1 = 30;
}
else if (value_2 < 30)
{
  value_2 = 30;
}
else if (value_3 < 30)
{
  value_3 = 30;
}
else if (value_4 < 30)
{
  value_4 = 30;
}
uint16_t range_1 = ((67870.0 / (value_1 - 3.0)) - 40.0);
Serial.println (value_1);
Serial.print (range_1);
Serial.println (" mm");
Serial.println ();
uint16_t range_2 = ((67870.0 / (value_2 - 3.0)) - 40.0);
Serial.println (value_2);
Serial.print (range_2);
Serial.println (" mm");
Serial.println ();
uint16_t range_3 = ((67870.0 / (value_3 - 3.0)) - 40.0);
Serial.println (value_3);
Serial.print (range_3);
Serial.println (" mm");
```




```
Serial.println ();
uint16_t range_4 = ((67870.0 / (value_4 - 3.0)) - 40.0);
Serial.println (value_4);
Serial.print (range_4);
Serial.println (" mm");
Serial.println ();

//motor
//front too close
if (cm1 < 30 || cm2 < 30)
{
  analogWrite(50, 200);
  delay(50);
  analogWrite(50, 0);
  delay(50);
  //go forward
  if (white >= -150 && white <= 150 && green < -150)
  {
    digitalWrite(PWM_1, HIGH);
    digitalWrite(PWM_2, LOW);
    digitalWrite(PWM_3, HIGH);
    digitalWrite(PWM_4, LOW);
    delay(100);
  }
  //go back
  else if (white >= -150 && white <= 150 && green > 150)
  {
    digitalWrite(PWM_1, HIGH);
    digitalWrite(PWM_2, HIGH);
    digitalWrite(PWM_3, HIGH);
    digitalWrite(PWM_4, HIGH);
    delay(50);
  }
  //turn left
  else if (green >= -150 && green <= 150 && white < -150)
  {
    digitalWrite(PWM_1, HIGH);
    digitalWrite(PWM_2, LOW);
    digitalWrite(PWM_3, LOW);
    digitalWrite(PWM_4, LOW);
    delay(50);
  }
  //turn right
  else if (green >= -150 && green <= 150 && white > 150)
  {
    digitalWrite(PWM_1, LOW);
    digitalWrite(PWM_2, LOW);
    digitalWrite(PWM_3, HIGH);
```



```
digitalWrite(PWM_4, LOW);
delay(50);
}
}
//left too close
else if (range_1 < 300)
{
analogWrite(51, 200);
delay(50);
analogWrite(51, 0);
delay(50);
//go forward
if (white >= -150 && white <= 150 && green < -150)
{
digitalWrite(PWM_1, HIGH);
digitalWrite(PWM_2, LOW);
digitalWrite(PWM_3, HIGH);
digitalWrite(PWM_4, LOW);

}
//go back
else if (white >= -150 && white <= 150 && green > 150)
{
digitalWrite(PWM_1, HIGH);
digitalWrite(PWM_2, HIGH);
digitalWrite(PWM_3, HIGH);
digitalWrite(PWM_4, HIGH);

}
//turn left
else if (green >= -150 && green <= 150 && white < -150)
{
digitalWrite(PWM_1, HIGH);
digitalWrite(PWM_2, LOW);
digitalWrite(PWM_3, LOW);
digitalWrite(PWM_4, LOW);

}
//turn right
else if (green >= -150 && green <= 150 && white > 150)
{
digitalWrite(PWM_1, LOW);
digitalWrite(PWM_2, LOW);
digitalWrite(PWM_3, HIGH);
digitalWrite(PWM_4, LOW);

}
}
```



```
//right too close
else if (range_2 < 300)
{
  analogWrite(52, 200);
  delay(50);
  analogWrite(52, 0);
  delay(50);
  //go forward
  if (white >= -150 && white <= 150 && green < -150)
  {
    digitalWrite(PWM_1, HIGH);
    digitalWrite(PWM_2, LOW);
    digitalWrite(PWM_3, HIGH);
    digitalWrite(PWM_4, LOW);

  }
  //go back
  else if (white >= -150 && white <= 150 && green > 150)
  {
    digitalWrite(PWM_1, HIGH);
    digitalWrite(PWM_2, HIGH);
    digitalWrite(PWM_3, HIGH);
    digitalWrite(PWM_4, HIGH);

  }
  //turn left
  else if (green >= -150 && green <= 150 && white < -150)
  {
    digitalWrite(PWM_1, HIGH);
    digitalWrite(PWM_2, LOW);
    digitalWrite(PWM_3, LOW);
    digitalWrite(PWM_4, LOW);

  }
  //turn right
  else if (green >= -150 && green <= 150 && white > 150)
  {
    digitalWrite(PWM_1, LOW);
    digitalWrite(PWM_2, LOW);
    digitalWrite(PWM_3, HIGH);
    digitalWrite(PWM_4, LOW);

  }
}
//back too close
else if (cm3 < 30 || range_3 < 300
        || cm4 < 30 || range_4 < 300 )
{
```



```
analogWrite(53, 200);
delay(50);
analogWrite(53, 0);
delay(50);
//go forward
if (white >= -150 && white <= 150 && green < -150)
{
    digitalWrite(PWM_1, HIGH);
    digitalWrite(PWM_2, LOW);
    digitalWrite(PWM_3, HIGH);
    digitalWrite(PWM_4, LOW);

}
//go back
else if (white >= -150 && white <= 150 && green > 150)
{
    digitalWrite(PWM_1, HIGH);
    digitalWrite(PWM_2, HIGH);
    digitalWrite(PWM_3, HIGH);
    digitalWrite(PWM_4, HIGH);

}
//turn left
else if (green >= -150 && green <= 150 && white < -150)
{
    digitalWrite(PWM_1, HIGH);
    digitalWrite(PWM_2, LOW);
    digitalWrite(PWM_3, LOW);
    digitalWrite(PWM_4, LOW);

}
//turn right
else if (green >= -150 && green <= 150 && white > 150)
{
    digitalWrite(PWM_1, LOW);
    digitalWrite(PWM_2, LOW);
    digitalWrite(PWM_3, HIGH);
    digitalWrite(PWM_4, LOW);

}
}
}

else
{

//go forward
if (white >= -150 && white <= 150 && green < -150)
{
```



```
digitalWrite(PWM_1, HIGH);
digitalWrite(PWM_2, LOW);
digitalWrite(PWM_3, HIGH);
digitalWrite(PWM_4, LOW);
delay(50);
}
//go back
else if (white >= -150 && white <= 150 && green > 150)
{
digitalWrite(PWM_1, HIGH);
digitalWrite(PWM_2, HIGH);
digitalWrite(PWM_3, HIGH);
digitalWrite(PWM_4, HIGH);
delay(50);
}
//turn left
else if (green >= -150 && green <= 150 && white < -150)
{
digitalWrite(PWM_1, HIGH);
digitalWrite(PWM_2, LOW);
digitalWrite(PWM_3, LOW);
digitalWrite(PWM_4, LOW);
delay(50);
}
//turn right
else if (green >= -150 && green <= 150 && white > 150)
{
digitalWrite(PWM_1, LOW);
digitalWrite(PWM_2, LOW);
digitalWrite(PWM_3, HIGH);
digitalWrite(PWM_4, LOW);
delay(50);
}
}
}

//headswitch mode
else if (ON_2 == 0 && ON_3 == 1)
{

int analogTurn = analogRead(moveright);
int analogForward = analogRead(moveforward);
DIR_1 = HIGH;
DIR_3 = HIGH;

Serial.println(analogTurn);
Serial.println(analogForward);
```



```
if (analogRead(moveright) == 0)
{
  digitalWrite(PWM_3, HIGH);
  digitalWrite(PWM_1, LOW);
  delay(10);
}

else
{
  if (analogRead(moveforward) == 0)
  {
    digitalWrite(PWM_1, HIGH);
    digitalWrite(PWM_3, HIGH);
    delay(10);
  }

  else
  {
    digitalWrite(PWM_1, LOW);
    digitalWrite(PWM_3, LOW);
  }
}

}

//standard joystick mode
else if (ON_2 == 1 && ON_3 == 0)
{
  Serial.print(white);
  Serial.print(",");
  Serial.println(green);
  DIR_1 = HIGH;
  DIR_2 = HIGH;
  DIR_3 = HIGH;
  DIR_4 = HIGH;

  //go forward
  if (white >= -150 && white <= 150 && green < -150)
  {
    digitalWrite(PWM_1, HIGH);
    digitalWrite(PWM_2, LOW);
    digitalWrite(PWM_3, HIGH);
    digitalWrite(PWM_4, LOW);
    delay(100);
  }
  //go back
  else if (white >= -150 && white <= 150 && green > 150)
  {
    digitalWrite(PWM_1, HIGH);
```



```
digitalWrite(PWM_2, HIGH);
digitalWrite(PWM_3, HIGH);
digitalWrite(PWM_4, HIGH);
delay(100);
}
//turn left
else if (green >= -150 && green <= 150 && white < -150)
{
  digitalWrite(PWM_1, HIGH);
  digitalWrite(PWM_2, LOW);
  digitalWrite(PWM_3, LOW);
  digitalWrite(PWM_4, LOW);
  delay(100);
}
//turn right
else if (green >= -150 && green <= 150 && white > 150)
{
  digitalWrite(PWM_1, LOW);
  digitalWrite(PWM_2, LOW);
  digitalWrite(PWM_3, HIGH);
  digitalWrite(PWM_4, LOW);
  delay(100);
}

else
{
  digitalWrite(PWM_1, LOW);
  digitalWrite(PWM_2, LOW);
  digitalWrite(PWM_3, LOW);
  digitalWrite(PWM_4, LOW);
  delay(10);
}
}

//off mode
else
{
  DIR_1 = HIGH;
  DIR_2 = HIGH;
  DIR_3 = HIGH;
  DIR_4 = HIGH;
  digitalWrite(PWM_1, LOW);
  digitalWrite(PWM_2, LOW);
  digitalWrite(PWM_3, LOW);
  digitalWrite(PWM_4, LOW);
}

delay(100);
```



}

Troubleshooting

Please reference the Arduino code to determine which pins the inputs and the sensors utilize if they detach.

Wires

There are a few different situations in which you may have issues with the wires. First of all, if some of the wires that are inside of the tape-wrapped conduit have fallen from where they were suspended with velcro, locate the part of the tape-wrapped conduit that has velcro on it, locate the appropriate velcro on the PVC pipe in the very center of the base, and connect the two. Once this is done, it should not fall off again unless it has been interfered with.

Force Feedback

For the connection between the force feedback joystick and Arduino, the direction forward and back is blue (+5V), green (data), yellow (-5V) thin wires, and the direction left and right are brown (+5V), red (data), orange (-5V) thin wires. In its Arduino code, white is x-axis, and green is y-axis. The quadrants are written on the joystick as well.

Head Switches

One Faulty Head Switch

If a head switch will not properly attach to the velcro on the PVC, try changing the angle at which it is attached. It was found that a perpendicular connection was not always successful, but one at forty-five degrees stayed strong every time. If a head switch no longer moves the device forward or rotates the device clockwise, but the other head switch works, disconnect the head switches from the PVC and from the wires that plug into them initially, switch their positions, reattach them to the wires and to the PVC, and then try both head switches again. If the problem has switched (lack of rotation in place of lack of propulsion), this means that the head switch is not working and needs to be replaced. If the problem stays the same, this means that the head switch is working well, but the wiring on the problem side is faulty. If this is the case, then contact a member of the Electrical Engineering Department at NAU, and the head



switch mode should not be used until this problem is resolved. If both head switches are working, as opposed to before, leave each head switch on its current side.

Two Faulty Head Switches

In the case that both head switches are not rotating / propelling the device while the mode switches are in head switch mode (01; the left switch is off and the right switch is on), detach one of the head switches from the device, unplug the wire from it, and plug in a new head switch. If the new head switch is working, this means that the old head switch was dead. If the new one is not working, this means that the wiring is faulty (or that the new head switch is dead also, which should be very unlikely). If this is the case, contact a member of the Electrical Engineering Department at NAU, and have them assist you with the faulty wiring problem. If this is the case, avoid the head switch mode until the problem is resolved. Once the problem on this one side has been determined, switch the new head switch to the other side (detach the old head switch from the PVC and disconnect the wire from it, then attach that wire to the new head switch), and repeat this process.

Also, make sure the head switches are fully inserted into the adapter and the adapter is fully inserted into the ring attachment. An example can be viewed below.



Figure 16: Incorrect Wire Connection



Figure 17: Correct Wire Connection



PVC

As a general note, the PVC pipes are all 1.25" in width (except for the PVC couplers).

For Head Switches

If the PVC with the velcro attached to it is too low for the push-wheelchair to get under, simply lift the PVC so that the wheelchair can wheel under it. Once the wheelchair is under it, you can lower the PVC until it rests on the wheelchair.

Connections With Other PVC Pipes

If a PVC to PVC connection happens to weaken and disconnect unexpectedly, take the PVC that has fallen off and compare it with the other PVC pipe it was connected to. Take some PVC glue and use the brush on the inside to apply a conservative amount (wipe the brush against the inside ring of the glue container; it initially has far too much glue on it) of the PVC glue to both the outside of the smaller pipe (only a ring of glue on its edge) and the inside of the larger pipe (also only a ring of glue on its edge). Then take the smaller pipe and insert it into the larger pipe, making sure that it is rotated in the way that you want it to be rotated.

Sensors

Wiring

In case a sensor's wire is disconnected in the future, a key is located below that details what the function of each colored wire is for each sensor in the *front* of the device (sensor wires that had to be extended in order to reach the Arduino):

Table 3: Color of Sensor Wires

Sensor Type	Ground	Power	Output	Echopin	Trigpin
Ultrasonic	Red	Green	N/A	Blue	Yellow
Infrared	Blue	Red	Yellow	N/A	N/A
Bumper	Blue	Yellow	Red	N/A	N/A

False Zero Readings

Occasionally the ultrasonic sensors will return a value of zero, which is incorrect. If this is happening, it is because the environment in which you are in does not return sound well enough



for the sensor to receive it (the ultrasonic sensor works by sending out a sound wave and then listening for that same sound wave). If this occurs, simply ignore the zero readings and rely upon either the infrared sensors or take additional measurements with the ultrasonic sensors.

Complete Sensor Failure

In the case of a complete sensor failure, the device will be able to operate with some forced-feedback (a sensor that has completely failed will always return a value of 0 for the distance from the sensor), however, the forced-feedback will not be reliable if there are multiple failed sensors located in similar locations on the device. You will likely notice a failed sensor when the device is in the forced-feedback mode (when the mode switches are both on), an obstacle is approached, and no forced-feedback is provided on the joystick, as opposed to what it should normally do. The best thing to do in this situation is to keep a record of which directions the device can no longer detect obstacles from, and then notify a colleague in the Electrical Engineering department at NAU about the issue. A future team or someone in the department will then be able to assist you in replacing the dead / disconnected sensors.

Batteries

If the device is no longer turning on, this is likely due to a lack of charge on the batteries. In order to determine if this is the problem, please acquire a digital multimeter or a voltmeter and measure the voltage across the leads of the battery (attach one end of your meter to the positive terminal and one end to the negative terminal). Be sure to unplug the device from the wall socket BEFORE you do this in order to avoid any electric shock. If either battery's voltage falls beneath the absolute value of 10 volts, it should be replaced. When replacing the battery, be sure to acquire a battery that is rated for 12 volts and near 50 amp hours (45 to 55 will do just fine). The batteries that are currently inside of the device were purchased at Batteries Plus on Rt. 66 in Flagstaff, AZ. Also, be sure to recycle the dead battery at whichever battery store the replacement battery is purchased at (take the dead battery into the store).



Status of Planned Features

Kristen Shryack

Table 4: WBS for Kristen

Person Primarily Responsible: Kristen Shryack				
ID	Activity/Task	Description	Deliverables	Other People
1 Headswitch activity				
1.1	Purchasing	Order duplicate headswitch	Place order	Place order with Alexandra
			Receive parts	
1.2	Program Arduino	Get readings from headswitch	Breadboard circuit	
			Test headswitch	
1.3	Integration	Connect Arduino to motor	Test headswitches with motor	Work with Mengyuan
2 Force Feedback Joystick				
2.1	Purchasing	Wait on order to be delivered	Receive parts	
2.2	Program Arduino	Update current code	Test FFJ	
			Add modes to FFJ	Work with Adam to integrate sensors
2.3	Integration	Connect Arduino to motor	Test FFJ	Work with Mengyuan
3	Finalizing circuit	Integration	Integrate separate code onto one Arduino	Work with all members to bring everything together
		Solder circuit		
		Test		

Kristen purchased and received the above said parts as well as some unexpected parts. She got readings from the joystick and the head switches. The head switches were integrated to be controlled by the motor as well as the joystick. Both can drive the device with the Arduino. Force feedback integration is still being worked on. She also soldered the joystick/sensors and added heat shrink to soldered areas. Testing results showed the device able to move based on inputs. force feedback was supposed to guide the child in a different direction. This became difficult to program since the force feedback could overcorrect. We also did not want to depend on the sensors too heavily. Instead, the program was designed to give quick jolts with a delay. It would still guide the student away from the obstacle, but it would not be constant guidance. The difficult part in force feedback was the amount of voltage that it needed. She measured it to need 24 V originally and it needed a separate power supply than the joystick. Since the batteries supplied 12 V, she tried it and the force feedback was still strong. She was able to control their movement with a motor driver that could handle the higher voltage and current. The motor drivers also took in multiple inputs and could control both of the joysticks motors. My testing with all three modes of inputs was with separate motors and a power supply. When it worked, she was able to hand it off to Yutian to attach to the base's batteries and motors. Testing the force feedback with sensors was not accomplished because the sensors aspect was not finished. She attached the velcro to the PVC pipes and created the first iteration of PVC design. I'll admit, it was not the best, and Adam was able to bring it to the current design. She also soldered all of the sensors and attached them to the device. She originally wanted to screw all of them in, but the ultrasonic ended up being attached with velcro. It was because it did not have a hole to screw in, and the velcro was surprisingly stable. She also wired and attached the mode switches and the emergency stop to the device.



Yutian Deng

Table 5: WBS for Yutian

Person Primarily Responsible: Yutian Deng

ID	Activity/Task	Description	Deliverables	Other People
1 Modes and levels programming				
1.1	Headswitch	Coding and testing with the mot	Headswitch that could control the base	Work with hKristen
1.2	Standard Joystick	Coding and testing with the mot	Joystick that could control the base	Work with Kristen
1.3	Assistance FFJ	Coding and testing with the mot	FFJ that could control the base	Work with Kristen
1.4	Minimal Assistance FFJ	Coding and testing with the mot	FFJ that could control the base	Work with Kristen
2 Mechanical				
2.1	Inputs Setting	Fix inputs on the base	Inputs on the base	Work with Adam
2.2	Connection To the Input	Connect inputs to Arduino	Test mode switch	Work with Adam

Kristen and Yutian have programmed the head switch, standard joystick, and force feedback joystick, in which the head switch and the standard joystick have been tested with the motors and work well, but when using the standard joystick, it goes very fast. He also programmed the emergency stop switch and switches to change the mode so they could work and are now placed on the base (glued or taped). The head switches could be placed on the base with velcro and the joystick could be put on legs while using them.

Mohammed Alfatouh

Table 6: WBS for Mohammed

Person Primarily Responsible :Mol Mohammed Alfatouh

ID	Activity/Task	Description	Deliverables	Other People
1 Base				
1.1	Turn on	Turn on the power	Hotwire the base to turn on the power	Yutian
1.2	Battery	Changing the Battery	Order new Battery	Adam
1.3	Change key switch	Changing the key switch for the base.		
2 power supply				
2.1	converter	Connection with Battery	order received	
2.2	Arduino	Connection with converter		
2.3	Testing	Test it all together	Testing the Arduion with converter	Yutian

Mohammed and Yutian hotwired the base to turn on the power, he tried to connect the motor to the battery to drive the motor and it worked well. He used lower gauge wire for the motor instead of smaller wire because when he used smaller wire the right motor did not work, so the lower gauge wire works well now. He and Adam received dead batteries, so we changed it with two new batteries (12V) and put them into the base. He and Yutian realized the key switch is not important to change it because we made the connection from motor to the relays and motor drivers, since we did that the key switch is not needed. He received the converter to make



connection between the battery and Arduino to feeding power to Arduino (5V) but our Arduino Due can only offer (3.3V), so he and his teammates decided to change the Arduino Due to the Arduino MEGA because it has enough pins to use it and to drive the trigger of the ultrasonic sensor (output 5V), so the Arduino MEGA can connect directly with battery (the power converter is not needed).

Adam Douglass

Table 7: WBS for Adam

Person Primarily Responsible: Adam Douglass				
ID	Activity/Task	Description	Deliverables	Other People
1	Sensor Testing			
1.1	Infrared Testing	Test output VS distance	Data / graphs	
1.2	Ultrasonic Testing	Test output VS distance	Data / graphs	
2	Sensor Integration			
2.1	IR Integration	Connect to the four corners	Stable Operational	Yutian
2.2	Ult. Son. Integration	Connect to the four corners	Stable Operational	Yutian
3	Training Levels	Communicate data	Data / graphs	Kristen

Adam found that Yutian had already done good work on the infrared and ultrasonic sensor programs, and that those programs automatically displayed the distance of each obstacle when it was detected through the serial monitor. Thus, he decided to work mainly on the wire housing, PVC support, sensor wiring, sensor mounting, PVC rotatability, wire stability (once housed), and wire connections (clamped and stripped with the help of all teammates). He was the main worker on these tasks, but other teammates also helped out (Kristen with sensor mounting, most other teammates with wood cutting, Mohammed with wire housing and PVC cutting). He also went out on numerous Home Depot runs in order to acquire the appropriate materials that were needed for the project at the time. He also tested some of the sensors after they were mounted on the device in order to make sure that the mounting process did not dislodge any wires.

Mengyuan Zhang

Table 8: WBS for Mengyuan

Person Primarily Responsible: Mengyuan Zhang				
ID	Activity/Task	Description	Deliverables	Other People
1	Base Building			
1.1	Purchasing	Order the parts (Battery, etc)	Identify parts Place order Receive parts	Place order with Dr. Winfree
1.2	Base Building	Change the everything wrong and connect the circuit	Change wrong parts Build circuit	Work with all members
1.3	Checking	Make sure the base can work correctly	Use joystick to test the base	Work with all members
2	Joystick Activity			
2.1	Programming Joystick	Get reading from joystick	Use Arduino to create a code	
2.2	Integration	Connect joystick to the base	Test the joystick	
3	Base Reading			
3.1	Programming Base	Control the base moving	Use Arduino to create a code Move the base correctly	
3.2	Testing	Test the all parts can go right	Test the joystick, force feedback, sensors, etc	Work with all members



Mengyuan decided to work on the base building, joystick programming, and testing. When she got the parts, she started working on the base building. She helped other members to take out the base to see the circuit and found that there are lots of problems need to be solved. For example, she did not know how the key could work. Then, by checking and considering, she found that is not important for the whole project. Besides, the batteries were about to be changed to the new one and got working well. Then, she worked on the shelves. By thinking about more possible different materials, all team members decided to use the PVC pipes to support the head switches on the chair. At first, she wanted to put the PVC on the front of the chair, but it was unstable. So, after Adam's suggestion, the PVC pipes were put on the middle of the base and used bolts and L-shaped cuts for the structure to be held. After that, she also helped in the circuit building on the base and made the wires work together harmoniously. Although, she wrote a code for the joystick before, but all of the members decided to use another one, which both has joystick and force feedback joystick on it. So, she gave more help to the team members that were working on the new joystick program for the two different modes.



Conclusion

Team AMP has worked to make this device user friendly and beneficial to you and your students. We hope you have many years with this project and that your students have a great joy driving it. We were happy to work on this as it is local and will help your students develop and grow. Although we are graduating, many of us are attached to this project and want to see it do well. Don't hesitate to contact us if you have any questions or concerns. We want it to operate efficiently and optimally with current and future generations. With school ending, there is no promise as to how often we will check our student emails. Please contact us at our personal emails, so that we can offer any further guidance or help. With best wishes from Team AMP, Kristen Shryack, Yutian Deng, Adam Douglass, Mohammed Alfatouh, and Mengyuan Zhang, we hope you enjoy the new additions to this device.

Kristen Shryack - kris10kalyn@gmail.com

Adam Douglass - douglassadam1@gmail.com

Mengyuan Zhang - zmy_951016@163.com

Yutian Deng - dyt9605@163.com

Mohammed Alfatouh - m7medd92@gmail.com



Appendices with Schematics

Flow Diagram

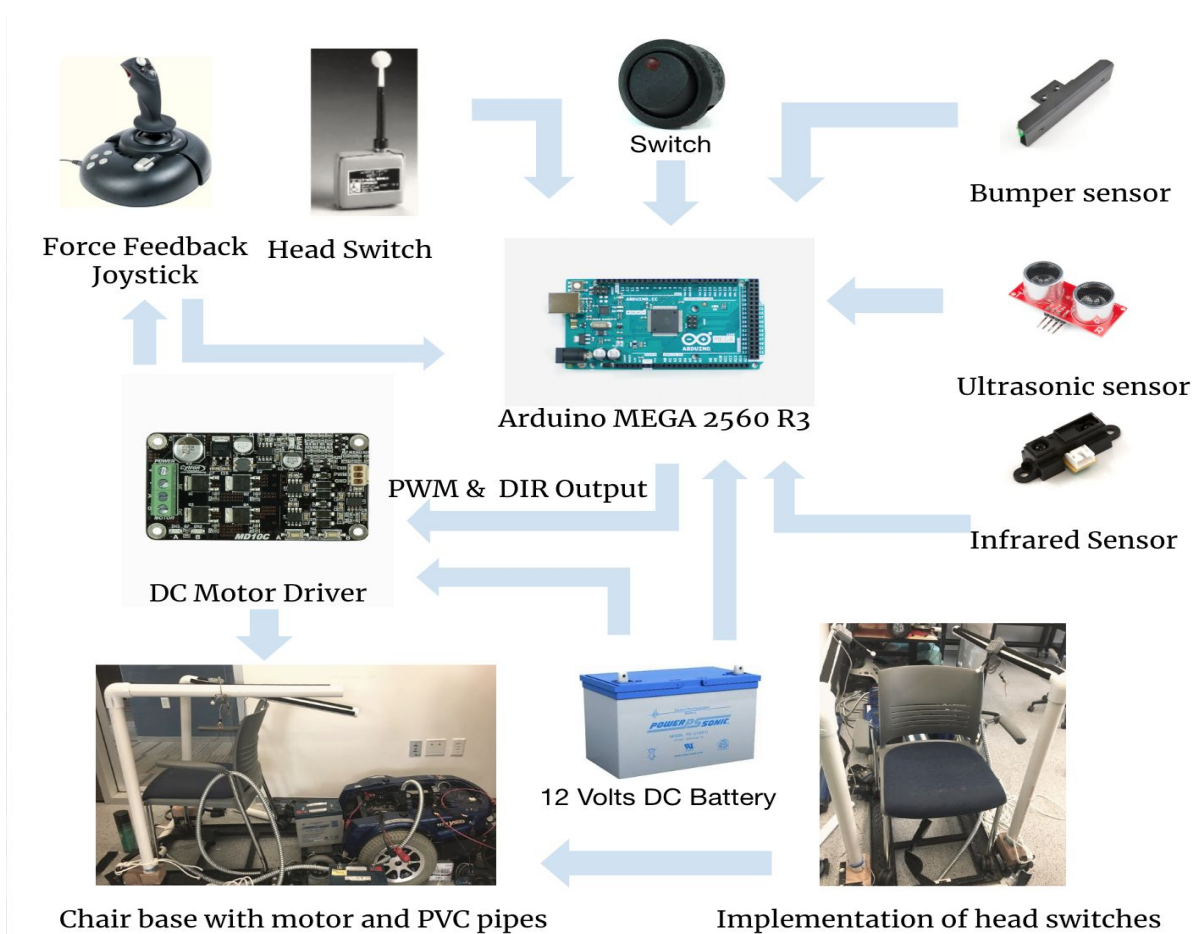


Figure 18: Overview of the Project



Various Relay Schematic

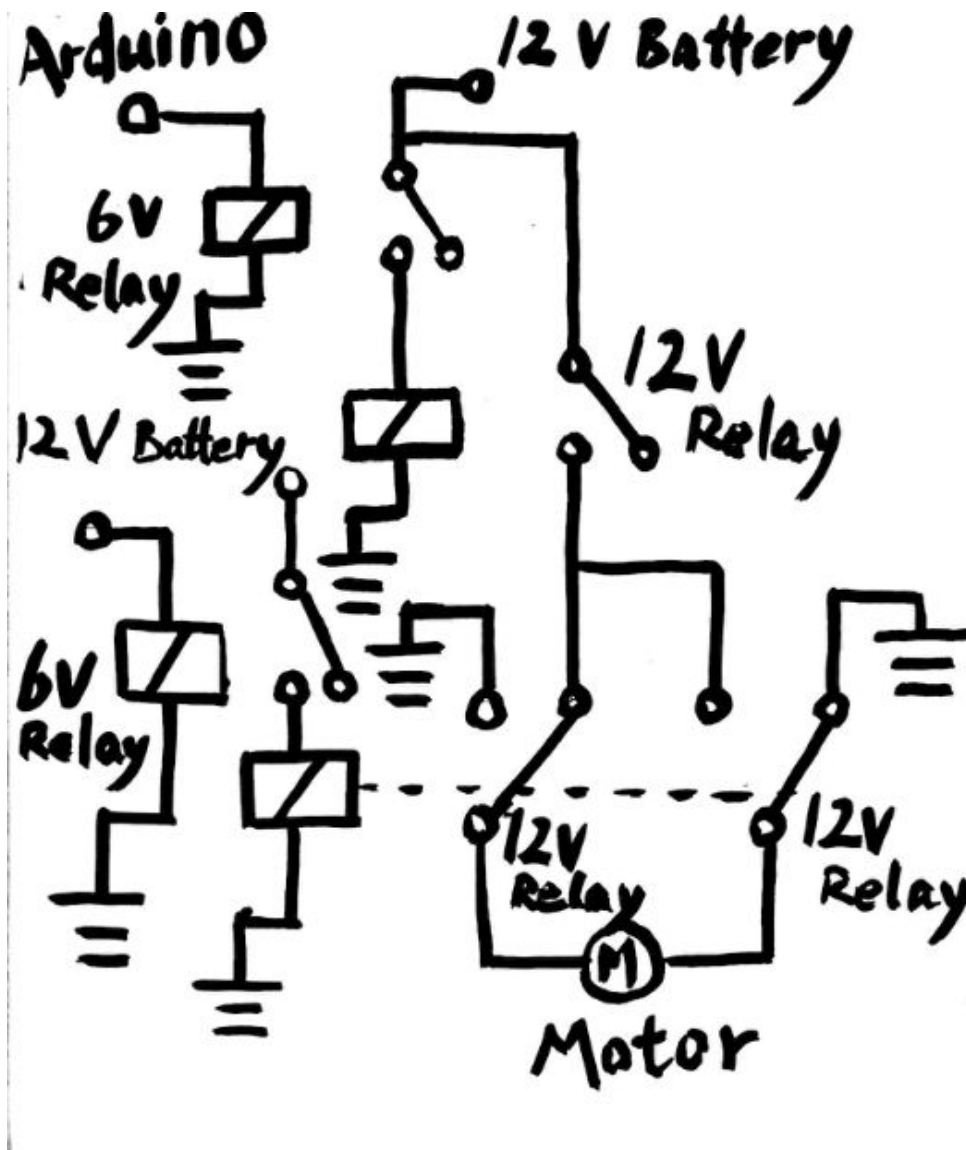


Figure 19: Schematic of Relay