Final Project Report

December 14, 2017

ENGR 112 Section 504, Introduction to Engineering
Texas A&M University, College Station, TX

Team 22
Syed Abeer Hasan Zaidi
Hassan Baig
Caleb Greenstreet
Caleb Jasik

DSTR Robot “Voltron”

Abstract

We built and tested a DSTR robot, named Voltron, for this project. Our mechanical team worked diligently to make sure the wheels were aligned, motors wired correctly, and that the construction was consistent and correctly implemented. Our control team worked tirelessly to solder motors, to resolder motors, and to write the Energia code to control the robot. We successfully constructed a working DSTR robot and yet soon everything turned disastrous. There were numerous problems with the wiring and motors which required us to compensate for disparate motor speeds in the software running the robot. We also went to the ECEN lab and tested our robot to discover the root cause of this problem, and ultimately determined that it was a mechanical problem with one (or both) of our motors. During competition, our robot completed near last for both FOM1 and FOM2 due to the mechanical failure, including one of the rear motors entirely seizing up and dying on us in the last moments before testing. Despite this, we learned a great deal about teamwork, delivering on a completed, if imperfect project, and communication and delegation in a team. We learned to think on our feet, and to anticipate and deal with temporary problems as they come.

1. INTRODUCTION

1.1. *Project Overview*

The goal of this project was to create a DSTR robot we could control over Wifi via phone and laptop. Though our robot, Voltron, performed poorly on FOM1 and FOM2 due to mechanical failure, we managed to mitigate the drifting by adjusting voltage levels per motor side, and succeeded at isolating the root of the problem. On the day of competition though, our wheel seized up and prevented us from participating.

1.2. *Team Introduction*

Name	Role	Contributions
Syed Abeer Hassan Zaidi	Mechanical Subsystem	constructing the mechanical subsystem repaired a motor
Hassan Baig	Mechanical Subsystem	constructing the mechanical subsystem wired up the H-bridge
Caleb Greenstreet	Control Subsystem	constructing the control subsystem wrote the Energia code for controlling the robot
Caleb Jasik	Control Subsystem	constructing the control subsystem wrote the Matlab script for FOM2

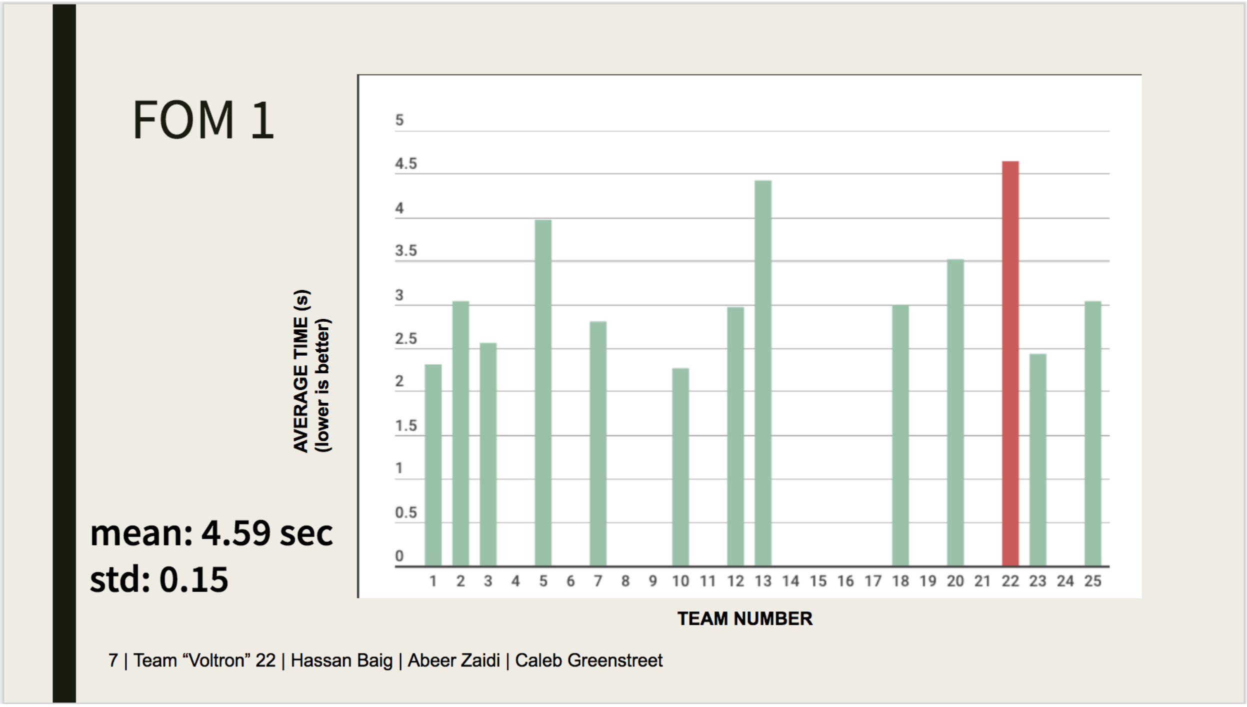
1.3. *Project Status*

In constructing the DSTR robot Voltron, we discovered that both left motors were immobile, and Voltron tends to the left. We affected repairs by resoldering both left motors. However, one of the left motors was still immobile, so we disassembled it. We found a shifted gear inside the motor, we promptly shifted the gear back into the correct position. Despite the repairs we did, The left motors moved slower than the right ones. We knew a previous TA for ENGR 112, and received his help in troubleshooting the problem. We hooked up the motors to a signal generator and ammeter, and discovered that the left side was drawing 1000mA. Twice the 500mA draw from the right side. This was done with just the wires to the motors and the motors in the circuit, without respect to the control bridge. We resoldered the wires, retested, and ultimately determined it must be a physical impedance in the motor.

2. SYSTEM PERFORMANCE

2.1. *FOM1*

Slow and steady doesn't win the race. We encountered high current draw and quickly draining batteries, which led us to be sure to charge the batteries every moment possible, and to manually account for the offset in steering. As can be viewed in Figure 2, we completed FOM1 poorly, finishing nearly in last place. We did have the most consistent timing though.



FOM 1

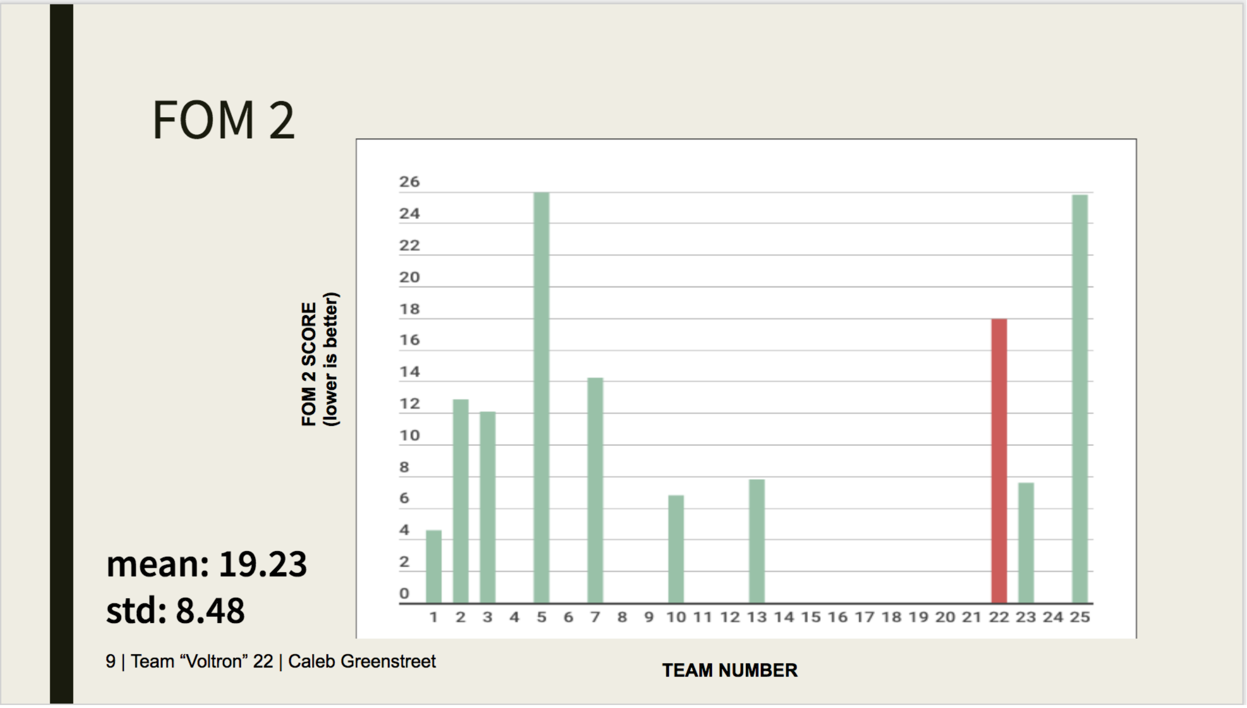
mean: 4.59 sec
std: 0.15

7 | Team "Voltron" 22 | Hassan Baig | Abeer Zaidi | Caleb Greenstreet

Figure 2. FOM1 Scores.

2.2. *FOM2*

Precision over speed was key. FOM2 was less reliant on speed, so we were able to improve our ranking moderately. We accounted for the inconsistent left and right motor speeds by making adjustments in the Matlab code, allowing our DSTR, “Voltron”, to move fairly straight despite the slow left side. Figure 3 shows our score versus the other teams in the class. While although we did not come in the top 50%, we did improve a bit, and we were the most consistent throughout testing.



FOM 2

mean: 19.23
std: 8.48

9 | Team “Voltron” 22 | Caleb Greenstreet

Figure 3. FOM2 Scores.

3. TEAM MEMBER CONTRIBUTIONS AND LESSONS LEARNED

3.1. *Hassan Baig*

What I did:

- Cutting, Drilling and assembling the DSTR
- Connected the wiring for the H-bridge
- Minor adjustments to DSTR robot throughout the project
- Worked on FOM1 and FOM2 with all group members

What I learned :

- Reliability and performance in Engineering design .
- Team Work
- Coding in Energia

As a part of of mechanical subteam, I got the opportunity to work on the stuff which I like the most. I did the drilling and cutting for most part of our DSTR project. With my other team mates, I worked on assembling our robot and connected the wire to the H-bridge. For most part of the project I worked with other team members to solve the motor related issues. I also worked on assembling the the initial frame and putting the legs together. During our FOM1 and FOM2 I worked with Caleb Jasik to adjust the speed of the right motor to make it go more straight. From this project, I learned that reliability and performance in engineering design is a key to achieve the desired results. I also learn how important is to work in a team. I also learned to code in energia.

3.2. *Syed Abeer Hasan Zaidi*

What I did:

- Measured and marked the rods and bars for cutting, drilling
- Helped build the DSTR
- Worked with Caleb Greenstreet on disassembling and fixing the motors
- Worked with Caleb Jasik to adjust the alignment of the legs using washers, to reduce the lateral offset

What I learned:

- Coding in Energia
- Working with power tools
- Working with electronic systems
- Resources offered at the EIC

I was a part of the mechanical sub team, so I did some of the drilling and cutting for the DSTR fabrication, and worked with everyone else to put together the DSTR. I also worked with Caleb Greenstreet to disassemble and fix one of the left motors. I also worked with Caleb Jasik in adding and removing washers on the legs of the DSTR, to make sure they were in line with one another. Over the course of this project I learned a lot about programming in Energia, electronic systems, working with power tools, and the resources available for us at the EIC.

3.3. *Caleb Greenstreet*

What I did:

- Measured and marked all sub assembly pieces
- Worked with Caleb Jasik to glue the picnic table together and bend the sled
- Worked with Caleb again to solder the wires to our motors and connect them to our H-bridge
- Wrote the Energia code required to run the DSTR from your phone

What I learned:

- How to read pin maps
- How to solder
- How to desolder
- Coding for microcontrollers
- Code to communicate wirelessly

Being apart of the Control Sub-Assembly I worked with my partner Caleb Jasik on a lot of the tasks. I measured and marked our the sled and picnic table, Caleb then drilled the holes in the picnic table. Caleb and I then heated and bent the sled. A lot of our time as a sub team was spent wiring the motors. We spent many hours in the EIC soldering and desoldering the connections to our motors. Lastly I wrote the energia code we used in our project to run our DSTR via phone. A lot of valuable lessons came from this project. Learning about programming microcontrollers and wireless communication will prove very valuable in my future. I feel I have gained a firm grasp of soldering after spending hours doing so. Overall this project came with a lot of mistakes but those quickly turned into learning opportunities.

3.4. *Caleb Jasik*

What I did:

- Prepared subassembly pieces to be cut and drilled
- Worked with Caleb Greenstreet to assemble picnic table and sled
- Spent hours at the EIC soldering motors to connect to the H-bridge
- Wrote code to communicate sensor data to Matlab
- Wrote and adjusted code for FOM2

- Wrote Energia code for communicating with DSTR controller

What I learned:

- Doing is more important than having a sane project
- Check the wires seven times and solder once
- Even projects constructed with ultimate care and precision can kerplode
- How to solder and write Energia code
- I love the EIC and it’s late hours

I worked with Caleb Greenstreet to complete the control subassembly, which mainly consisted of soldering motors and troubleshooting backwards wiring and resistive motors. One of the problems we faced was that our wiring was correct, and our contact points sound and heat-shrinked, but the right-side motors spun counter to each other. We tried everything, walking through the wiring, testing the motors while connected directly to a battery rather than the H-bridge. Eventually we gave up and switched the polarity! (We crossed the black and red wires). Preparing for FOM2 meant many rounds of running and tweaking the Matlab code to get the power to each pair of motors and the amount of time to run just right. One of the more useful things I learned from this team effort was that Doing is sometimes more important than “things making sense”. When we were trying to figure out the reversed motors problem, I wanted to discover the root problem, but I needed to balance that with actually creating a completed robot. Also, after we managed to solder all of the motors to the wires leading to the H-bridge, we have had to re-solder upwards of five motors to fix connection problems, or flipped connections. After testing the electrical subsystem of the motors with a signal generator and ammeter, we discovered that the left motors were drawing twice the current of the right side motors. After resoldering the connection between the motors and double-checking that they weren’t connected to the H-bridge during the test, we concluded that it must be a mechanical resistance in one of the motors. On the last day of testing, right before we wanted to record our FOM2 trial, the back left motor seized up, though we had taken great care to construct our robot correctly and effectively. I’ve learned how to program and use Matlab and Energia with TI Launchpads, and how to solder. Ultimately, I’ve learned how to work within a team better, what trade-offs to make when trying to complete a project, and a better understanding of how to communicate in a team.

4. Summary and Conclusion

Overall the performance of DSTR robot was not good as our team expected. Throughout our project we had issues with the left motors. Despite making several attempts to fix the motors , the left motors of our DSTR robot did not function properly. We were able to fix some of the problems, account for the errors with our code, and make adjustments when manually controlling the DSTR. As a result, our DSTR robot was slow and our team did not achieve the desired results in FOM1. During our FOM2 Testing our robot was slow but drove straight . So in comparison our performance in FOM2 was quite better as it was less reliant on the motor speed and more on precision in its movement. Overall we think this project was a success as we did have fairly good testing results on FOM1 and FOM2, and we were provided with a lot of learning experience from the problems we encountered. Had we noticed the problems in our motors earlier, we could have switched one of the motors on the left side, accounting for the lateral offset caused by the uneven motor speeds.

APPENDIX A, TEAM LAB NOTEBOOK

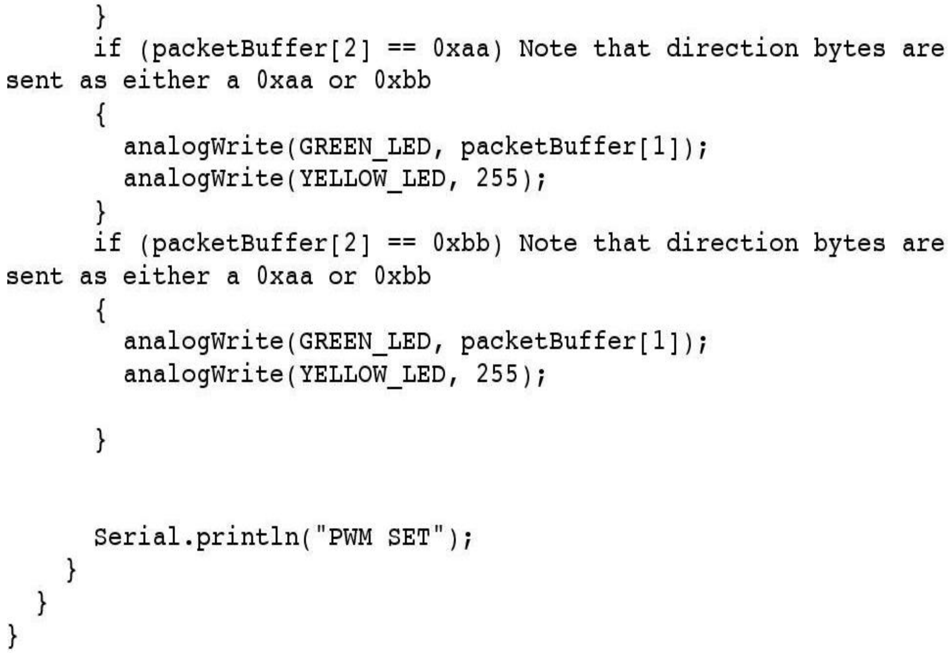
Date	Location	Purpose
Oct, 19 2017	EIC	Initial cutting and drilling
Oct, 25 2017	EIC	Completing cutting and drilling
Nov, 8 2017	EIC	Finishing Assembly of DSTR
Nov, 12 2017	EIC	Completing the wiring, and testing it for the first time
Nov, 13 2017	EIC	Solving Motor Issues
Nov 15 2017	EIC	Caleb Diagnosing Motor Issues
Dec 12, 2017	EIC	Final Presentation

APPENDIX B, DESCRIPTION OF ADDITIONAL MATERIAL

Code for controlling the DSTR robot, and code for interpreting the commands sent and executing them are included after this page for the reader.

```
#include <SPI.h>
#include <WiFi.h>

char wifi_name[] = "Insert Team Name Here"; //Change this value
to the desired broadcast name of your Wi-Fi
char wifi_password[] = "dstrdstr"; //Changing this is not
required

/* NOTE: This port is already defined in the "DSTR" app,
changing this will make the default app non-functional */
unsigned int localPort = 3553;      // local port to listen on
WiFiUDP Udp;

char packetBuffer[255]; //buffer to hold incoming packet
char  ReplyBuffer[] = "acknowledged";       // a string to send
back

void setup()
{

  // Start WiFi and create a network with wifi_name as the
network name
  // with wifi_password as the password.
  pinMode(GREEN_LED, OUTPUT);  //On the CC3200 Launchpad,
"GREEN_LED" is boosterpack pin 10 - marked P02 on the silk
screen
  pinMode(YELLOW_LED, OUTPUT); //On the CC3200 Launchpad,
"YELLOW_LED" is boosterpack pin 9 - marked P01 on the slik
screen
  pinMode(RED_LED, OUTPUT);  //On the CC3200 Launchpad,
"RED_LED" is boosterpack pin 29 - marked P64 on the silk screen
  pinMode(31, OUTPUT);     //On the CC3200 Launchpad,
boosterpack pin 31 is marked P17 on the silk screen
  analogWrite(GREEN_LED, 0);
  analogWrite(YELLOW_LED, 0);
  analogWrite(RED_LED, 0);
  analogWrite(31, 0);
  Serial.begin(115200);
  Serial.print("Starting network...");
  WiFi.beginNetwork(wifi_name, wifi_password);
  Serial.println("done.");
  Udp.begin(localPort);
  Serial.println("Please input PWM in LMOTORS|RMOTORS format!");
```

```
}

void loop()
{

  int packetSize = Udp.parsePacket(); //Waits for receipt of a
packet and stores the data into a rx buffer
  if (packetSize)
  {
    Serial.print("Received packet of size ");
    Serial.println(packetSize);
    Serial.print("From ");
    IPAddress remoteIp = Udp.remoteIP(); //Returns the IP of the
remote device connected to the launchpad
    Serial.print(remoteIp);
    Serial.print(", port ");
    Serial.println(Udp.remotePort()); //Returns the remote port
which the packet was received on
    // read the packet into packetBufffer
    int len = Udp.read(packetBuffer, 255); //Copies up to 255
bytes of data from the rx buffer into the packetBuffer array
    if (len > 0) packetBuffer[len] = 0;
    Serial.println("Contents:");
    // Serial.println(packetBuffer);
    Serial.println(packetBuffer[0], DEC); // direction 1
    Serial.println(packetBuffer[1], DEC); // speed 2
    Serial.println(packetBuffer[2], DEC);// direction 2
    Serial.println(packetBuffer[3], DEC);// speed 1

    if (len == 4)
    {

      /*
         STUDENT CODE BEGINS HERE
      */

      if (packetBuffer[0] == 0xaa) Note that direction bytes are
sent as either a 0xaa or 0xbb
      {
        analogWrite(GREEN_LED, packetBuffer[3]);
        analogWrite(YELLOW_LED, 255);
      }
      if (packetBuffer[0] == 0xbb) Note that direction bytes are
sent as either a 0xaa or 0xbb
      {
        analogWrite(GREEN_LED, 255);
        analogWrite(YELLOW_LED, packetBuffer[3]);
```



```
        }
        if (packetBuffer[2] == 0xaa) Note that direction bytes are
sent as either a 0xaa or 0xbb
        {
          analogWrite(GREEN_LED, packetBuffer[1]);
          analogWrite(YELLOW_LED, 255);
        }
        if (packetBuffer[2] == 0xbb) Note that direction bytes are
sent as either a 0xaa or 0xbb
        {
          analogWrite(GREEN_LED, packetBuffer[1]);
          analogWrite(YELLOW_LED, 255);

        }


        Serial.println("PWM SET");
      }
    }
}
```

ACKNOWLEDGMENTS

Our special thanks to Jacob Fasseler, the previous ENGR112 TA.