FINAL 01 — Low Fidelity Chassis Prototype

Cardboard Chassis Prototype
Diana Fan
DESINV 22 — Summer 2019

Contents:

Product Video

Link to video: https://youtu.be/CmSdQh_b8jw

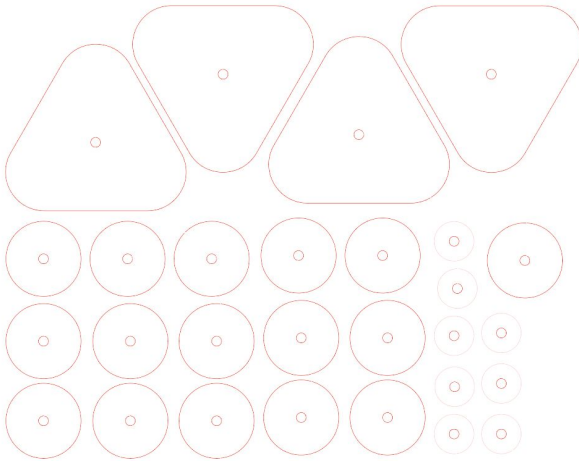**Product Shot**

Process Photos

Photo A: Illustrator file for the laser cut cardboard.

Photo B: Cardboard prototype on top of temporary stand.

Bill of Materials

Item	Description	Quantity
Cardboard	basic cardboard from a box	~12"x15"
Wooden Dowel	¼" wide	1
Paper	thicker than printer paper, thin strips ~1 cm wide	1
Wooden Blocks	various sizes	4

Tool	Description	Quantity
Laser Cutter	Universal VLS 6.60, machine in Jacobs	1
Wood Glue	provided by Invention Lab	1
Adobe Illustrator	software used to create cut files	
Hacksaw	provided by Jacobs	1

Written Response

- *What is your experience with designing a vehicle?*

   If you can count putting together Lego blocks to create a "vehicle" as experience, then that's the experience I have. Aside from that, the chassis from the last project gave my partner and I a little insight into vehicle design and we were inspired by the treads, so we decided to go with them since it seemed like they covered a good amount of surface area in proportion to our proposed weight and size of the chassis.

- *What level of difficulty is your vehicle design?*

   At surface level, the design looks very simple. I wouldn't say it's the easiest, but we didn't include features such as suspension, universal joints, or multiple bodies. Everything is contained within one storage component, and the movement is propelled essentially by gears and treads.

- *Reflect on your process: What surprises did you encounter when making your prototypes? What parts were the most frustrating? How might you approach it if you were doing it again?*

   I didn't really encounter any surprises or have any frustrations when working with the cardboard prototype, since I was working with materials I was very familiar with (cardboard, wood, paper). If I were to remake the model, I think I would make the "wheel" gears much wider, so the paper treads would represent more surface area that the future model would have.

Process and Description

My partner and I first did some research to look up examples of robot vehicles, and then we sketched out a few ideas. We ended up wanting to go with a small, simple but approachable-looking model because we wanted to keep in mind the idea that we wanted our robot to look like a real industrial product that people (maybe children) would want to buy. Our model was based on an equilateral triangular prism with two motors and treads on either end. We also wanted a design that could account for flexibility—if it lost balance and fell forwards or backwards, it would still be able to move because of the symmetry of the sides.

Since it was a simple cardboard prototype, I didn't feel the need to cut out any actual gears, and used cardboard circles as a placeholder. I created the Illustrator files, laser cut the cardboard (technically didn't need to but I felt that was faster than hand-cutting) and assembled the model, fitting paper strips over the gears as placeholder treads. Spinning the connecting dowel in the middle was supposed to simulate the dual DC motors running but in future prototypes they're obviously not going to be connected.

FINAL 02 — Updated Motorized Chassis Prototype

Plywood Motorized Chassis Prototype
Diana Fan
DESINV 22 — Summer 2019

Contents:

Product Video

Link to video: https://youtu.be/LaffAoV7N3I
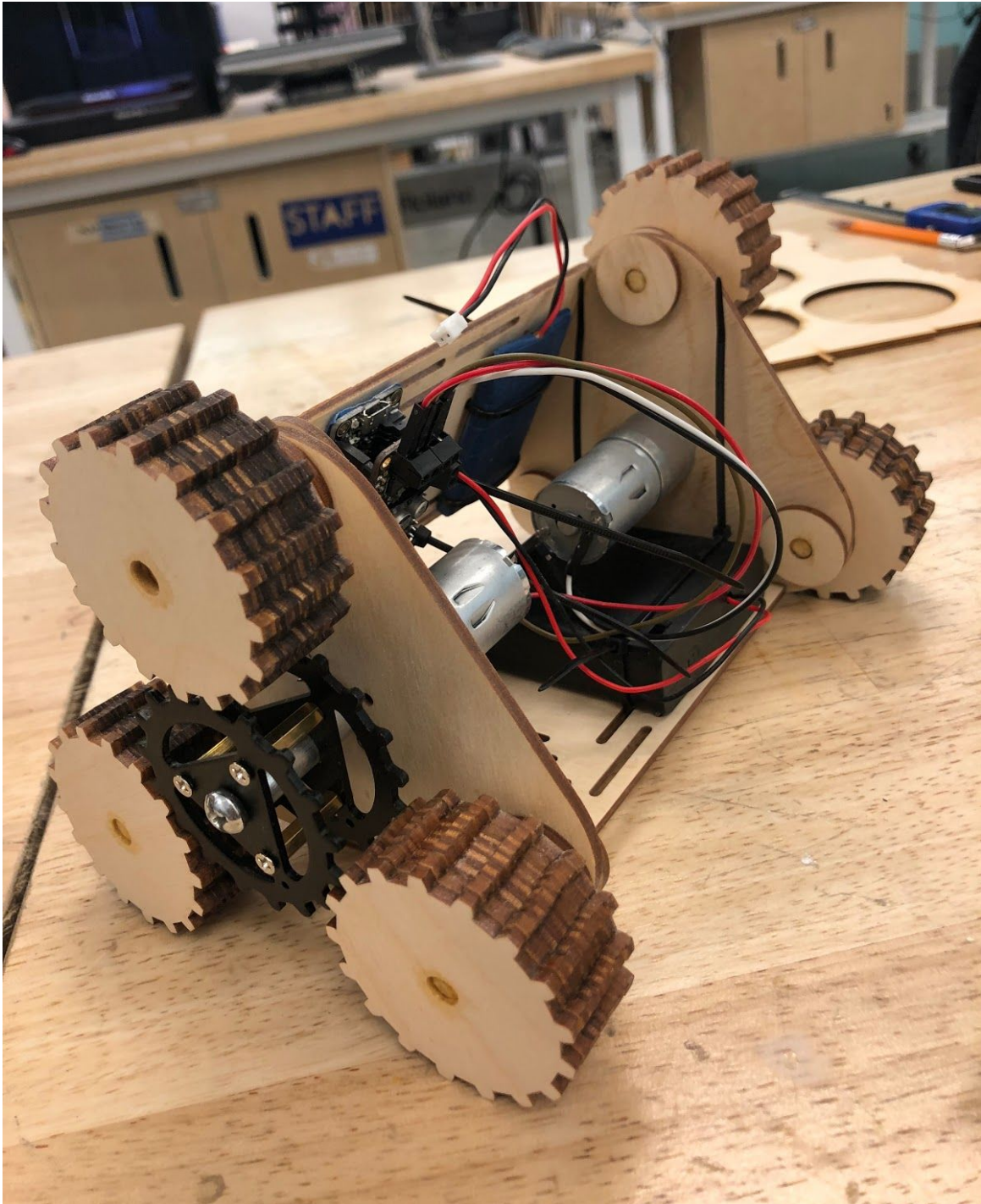

Product Shot

Photo: Updated medium fidelity prototype, programmed under the name "lightning mcqueen"

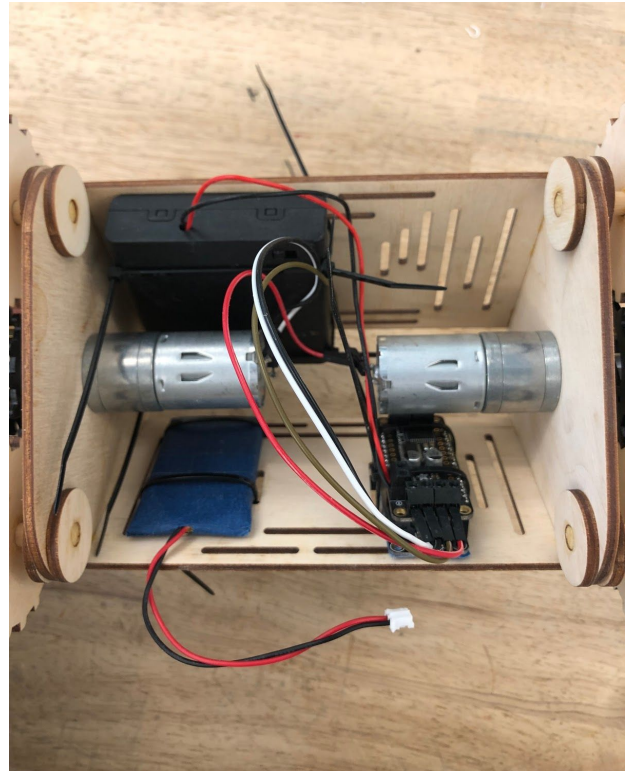## Process Photos

Photo A: Cut gears and washers out of plywood

Photo B: Electronic components affixed to the chassis

Bill of Materials

Item	Description	Quantity
Plywood	⅛" thickness, 11.5" x 24", bought from Jacobs	2
Wooden Dowel	¼" wide	1
Metal Gears	taken from metal tank chassis (Midterm 02)	2
Screws	Various sizes	
Feather 32u4 Bluefruit LE	Arduino-compatible, comes with Bluetooth control	1
Motor Featherwing	provides power to the motors, provided by Chris	1
DC Motors	attached to the tank chassis, provided by Chris	2
Lithium Ion Battery	to power the Feather with, provided by Chris	1

AA Batteries	batteries to power the motor with	4
4xAA Battery Holder	portable battery holder with an on/off switch	1
Jumper Wires	for connections	4

Tool	Description	Quantity
Laser Cutter	Universal VLS 6.60, machine in Jacobs	1
Wood Glue	provided by Jacobs	1
Adobe Illustrator	software used to create cut files	
Hacksaw	provided by Jacobs	1

Code (credit: Adafruit MiniRaceCar Sketch, edited to comment out code for temperature/ humidity sensor)

```
/**********************************************************************

 Mini Race Car kit -- Formula E Feather robot with HTU21D-F
 Temperature/Humidity sensor

 This is an example for our nRF51822 based Bluefruit LE modules

 Modified to drive a 3-wheeled BLE Robot Rover! by http://james.devi.to

 Pick one up today in the Adafruit shop!

 Adafruit invests time and resources providing this open source code,
 please support Adafruit and open-source hardware by purchasing
 products from Adafruit!

 MIT license, check LICENSE for more information
 All text above, and the splash screen below must be included in
 any redistribution
 **********************************************************************/

#include <Arduino.h>
#include <Adafruit_BLE.h>
#include <Adafruit_BluefruitLE_SPI.h>

#include "BluefruitConfig.h"

#include <Adafruit_MotorShield.h>

#include <Wire.h>
//#include "Adafruit_HTU21DF.h"
// Connect Vin to 3V DC
// Connect GND to ground
// Connect SCL to I2C clock pin
// Connect SDA to I2C data pin
```

```cpp
// Create the temp/humidity sensor object
//Adafruit_HTU21DF htu = Adafruit_HTU21DF();


// Create the motor shield object with the default I2C address
Adafruit_MotorShield AFMS = Adafruit_MotorShield();

// And connect 2 DC motors to port M3 & M4 !
Adafruit_DCMotor *L_MOTOR = AFMS.getMotor(1);
Adafruit_DCMotor *R_MOTOR = AFMS.getMotor(2);

//Name your RC here
String BROADCAST_NAME = "lightning mcqueen";

String BROADCAST_CMD = String("AT+GAPDEVNAME=" + BROADCAST_NAME);

Adafruit_BluefruitLE_SPI ble(BLUEFRUIT_SPI_CS, BLUEFRUIT_SPI_IRQ, BLUEFRUIT_SPI_RST);

// A small helper
void error(const __FlashStringHelper*err) {
  Serial.println(err);
  while (1);
}

// function prototypes over in packetparser.cpp
uint8_t readPacket(Adafruit_BLE *ble, uint16_t timeout);
float parsefloat(uint8_t *buffer);
void printHex(const uint8_t * data, const uint32_t numBytes);

// the packet buffer
extern uint8_t packetbuffer[];

char buf[60];

// Set your forward, reverse, and turning speeds
#define ForwardSpeed             255
#define ReverseSpeed             255
#define TurningSpeed             100


/**************************************************************************/
/*!
    @brief  Sets up the HW and the BLE module (this function is called
            automatically on startup)
*/
/**************************************************************************/
void setup(void) {
  Serial.begin(9600);
  //if (!htu.begin()) { //start the temp/humidity sensor
  //  Serial.println("Couldn't find sensor!");
  //  while (1);
  //}

  AFMS.begin();  // create with the default frequency 1.6KHz

  // turn on motors
  L_MOTOR->setSpeed(0);
  L_MOTOR->run(RELEASE);

  R_MOTOR->setSpeed(0);
  R_MOTOR->run(RELEASE);

  Serial.begin(115200);
  Serial.println(F("Adafruit Bluefruit Robot Controller Example"));
  Serial.println(F("---------------------------------------"));
```

```
  /* Initialize the module */
  BLEsetup();
}

void loop(void)
{
  // read new packet data
  uint8_t len = readPacket(&ble, BLE_READPACKET_TIMEOUT);

  readController();

  //for the plotter

  //ble.print("Temperature C: ");
  //ble.print(htu.readTemperature());
  //ble.print("\t");

  //ble.print("\tHumidity: ");
  //ble.println(htu.readHumidity());

}


bool isMoving = false;
unsigned long lastPress = 0;

bool readController(){
  uint8_t maxspeed;

 // Buttons
  if (packetbuffer[1] == 'B') {

    uint8_t buttnum = packetbuffer[2] - '0';
    boolean pressed = packetbuffer[3] - '0';

    if (pressed) {
      if(buttnum == 1){
        //ble.print("Temperature C: "); ble.println(htu.readTemperature());
      }

      if(buttnum == 2){
        //ble.print("Temperature F: "); ble.println((htu.readTemperature()*1.8)+32);
      }

      if(buttnum == 3){
        //ble.print("Humidity: "); ble.println(htu.readHumidity());
      }

      if(buttnum == 4){

      }

      if(buttnum == 5){
        isMoving = true;
        L_MOTOR->run(FORWARD);
        R_MOTOR->run(FORWARD);
        maxspeed = ForwardSpeed;
        ble.println("Forward");
      }

      if(buttnum == 6){
        isMoving = true;
        L_MOTOR->run(BACKWARD);
        R_MOTOR->run(BACKWARD);
```

```
        maxspeed = ReverseSpeed;
        ble.println("Backward");
      }

      if(buttnum == 8){
        isMoving = true;
        L_MOTOR->run(BACKWARD);
        R_MOTOR->run(FORWARD);
        maxspeed = ReverseSpeed;
        ble.println("Left");
      }

      if(buttnum == 7){
        isMoving = true;
        L_MOTOR->run(FORWARD);
        R_MOTOR->run(BACKWARD);
        maxspeed = ForwardSpeed;
        ble.println("Right");
      }

      lastPress = millis();

      // speed up the motors
      for (int speed=0; speed < maxspeed; speed+=5) {
        L_MOTOR->setSpeed(speed);
        R_MOTOR->setSpeed(speed);
        delay(5); // 250ms total to speed up
      }
  } else {
      isMoving = false;
      // slow down the motors
      for (int speed = maxspeed; speed >= 0; speed-=5) {
        L_MOTOR->setSpeed(speed);
        R_MOTOR->setSpeed(speed);
        delay(5); // 50ms total to slow down
      }
      L_MOTOR->run(RELEASE);
      R_MOTOR->run(RELEASE);
    }
}
}

void BLEsetup(){
  Serial.print(F("Initialising the Bluefruit LE module: "));

  if ( !ble.begin(VERBOSE_MODE) )
  {
    error(F("Couldn't find Bluefruit, make sure it's in CoMmanD mode & check wiring?"));
  }
  Serial.println( F("OK!") );

  /* Perform a factory reset to make sure everything is in a known state */
  Serial.println(F("Performing a factory reset: "));
  if (! ble.factoryReset() ){
      error(F("Couldn't factory reset"));
  }

  //Convert the name change command to a char array
  BROADCAST_CMD.toCharArray(buf, 60);

  //Change the broadcast device name here!
  if(ble.sendCommandCheckOK(buf)){
    Serial.println("name changed");
  }
  delay(250);
```

```
  //reset to take effect
  if(ble.sendCommandCheckOK("ATZ")){
    Serial.println("resetting");
  }
  delay(250);

  //Confirm name change
  ble.sendCommandCheckOK("AT+GAPDEVNAME");

  /* Disable command echo from Bluefruit */
  ble.echo(false);

  Serial.println("Requesting Bluefruit info:");
  /* Print Bluefruit information */
  ble.info();

  Serial.println(F("Please use Adafruit Bluefruit LE app to connect in Controller mode"));
  Serial.println(F("Then activate/use the sensors, color picker, game controller, etc!"));
  Serial.println();

  ble.verbose(false);  // debug info is a little annoying after this point!

  /* Wait for connection */
  while (! ble.isConnected()) {
      delay(500);
  }

  Serial.println(F("******************"));

  // Set Bluefruit to DATA mode
  Serial.println( F("Switching to DATA mode!") );
  ble.setMode(BLUEFRUIT_MODE_DATA);

  Serial.println(F("******************"));
}
```

Process and Description

As listed in the requirements, we created a medium fidelity version of our cardboard prototype, motorized it, and made it steerable forward, reverse, left, and right via Bluetooth connection.

Chassis

In order to get started, I studied the metal chassis that Chris gave us to see how the motors were attached to the body. After measuring the spacing of the tiny holes for the screws, I replicated the same holes on an Illustrator file, and designed the rest of the

chassis similarly to our cardboard prototype, which was based off of the shape of an equilateral triangle. The exception was that I had to figure out a way to attach the electronics to the chassis this time, which I solved by creating two sides with various slots and cutouts. Having multiple cutouts created a modular solution such that I could rearrange the electronics if I needed to. I also laser cut multiple gears to stack and glue together (seven ⅛" gears for "one" gear) to create wheels that would match the width of the ones I hacked (sort of without permission but I'll put them back on the original chassis I know they're expensive) from the metal chassis. The metal gears would be the ones motorized, with one on each side, and would spin the three surrounding it. I also cut out some washers to keep the makeshift gear wheels horizontally fixed.

After attaching the motors to the metal gears, and gluing the wooden gears to the dowels and to the sides of the chassis, I joined the left and right sides with the two slats of plywood with the cutouts, creating sort of a triangular prism with one open side. Then I arranged the electronics to make sure nothing would get in the way of the wheels or be tangled, and also made sure the on/off switch and plug areas were accessible. I zip-tied the parts firmly to assert that nothing would fall out when the vehicle was moving, and the mid-fidelity prototype was finished.

Controller

Figuring out how to implement Bluetooth control was difficult at first because it was hard to determine which tutorials we could reference and which libraries we needed to install. We ended up following the Adafruit tutorial for "My Mini Race Car" (https://learn.adafruit.com/my-mini-race-car), and since our vehicle wasn't required to have sensors, we commented out portions of the code that referred to the temperature/humidity sensors. After configuring to the correct board and port, we uploaded the code. I also made sure to rename our vehicle "lightning mcqueen" so we wouldn't get confused and accidentally connect to the wrong vehicles on accident for future reference. The code worked great for our vehicle and the motors worked at a nice speed—I was initially concerned that the wooden gears wouldn't be able to keep up and would end up not working.

FINAL 03 — Bluetooth, Power Distribution, and High Fidelity Progress

High Fidelity Chassis Progress
Diana Fan
DESINV 22 — Summer 2019

Contents:

Product Video

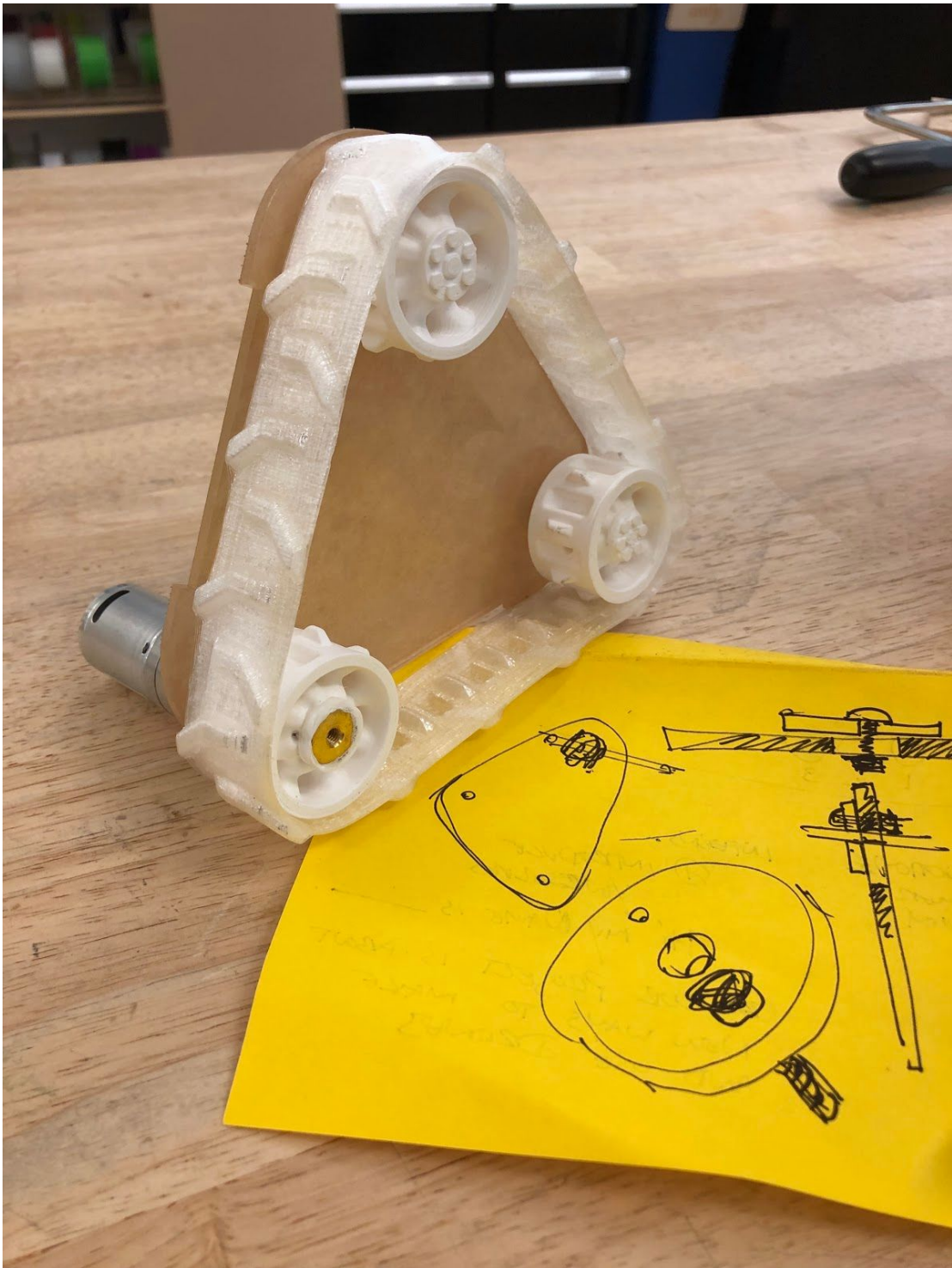Link to video: https://youtu.be/x_AxHe4RBMc

Product Shot

Photo: Progress shot of one side of the high fidelity prototype with an acrylic base, 3D printed wheels, and 3D printed gears.
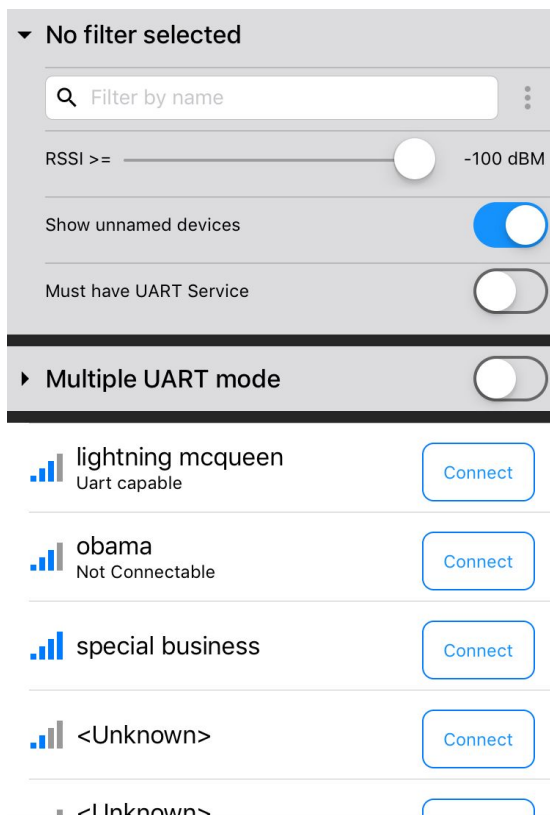
Process Photos

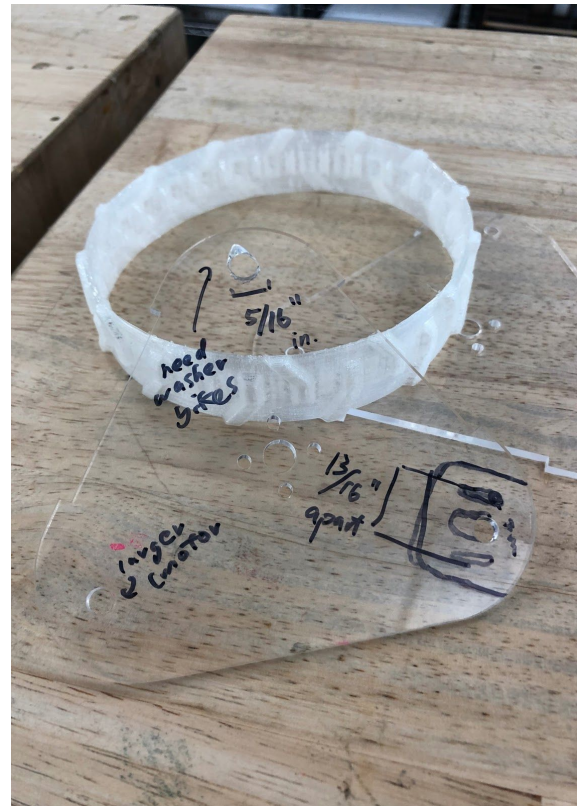Photo A: Screenshot of renamed and connectable vehicle.

Photo B: 3D printed treads along with notes on acrylic base draft.

Bill of Materials

Item	Description	Quantity
Cast Acrylic	⅛" thickness, 11.5" x 24", bought from Jacobs	1
PLA	white, 3D print material for wheels	
Nuts	to affix the wheels and screws onto acrylic base	4
Screws	to affix the wheels onto acrylic base	4
Feather 32u4 Bluefruit LE	Arduino-compatible, comes with Bluetooth control	1
Motor Featherwing	provides power to the motors, provided by Chris	1

DC Motors	provided by Chris	2
Lithium Ion Battery	to power the Feather with, provided by Chris	1
AA Batteries	batteries to power the motor with	4
4xAA Battery Holder	portable battery holder with an on/off switch	1
Jumper Wires	for connections	4

Tool	Description	Quantity
Wood Glue	provided by Jacobs	1
Adobe Illustrator	software used to create cut files	

Process and Description

We had already figured out the process for the bluetooth configuration and power distribution for the mid-fidelity prototype, so refer to FINAL 02 for documentation. This week we focused on the beginnings of bringing our medium fidelity prototype to high fidelity.

Critique of Vehicle Concepts

We didn't have any power issues within our prototype so far, or any steering issues as seen in the video, as it is able to go forward, backward, and turn left and right with no problems. However, I do have some concerns regarding traction, because I did try running the obstacle course ramp (separately outside of class time) and as expected, the plywood wheels slipped a bit, but still were able to make it up the slope. But in order to solve that problem, that is why we 3D printed treads with Ninjaflex material to have more grip on the cardboard.

Vehicle electronics

All of the electronics involved were the same as before with no changes, powered by four AA batteries and lithium ion battery hooked onto the Feather and Featherwing. We made sure that the wires did not run over boards and that plug-in points were easy to access.

Chassis Progress

We thought that having a see-through chassis would be a cool design choice so that's why my partner and I settled on choosing clear acrylic for the material of the body. We heeded Chris' warning about it being brittle so we are designing to make sure the body does not touch the ground or receive too much pressure in case it shatters. As seen from the Product Shot, we are still in the process of figuring out the rest of the body and how it will hold the electronics, but it shows how the wheels will be mounted. I 3D printed multiple sets of wheels with extras, drilled larger holes and threaded them so that we could pass them into holes in the acrylic with screws and nuts to be affixed. The treads fit with adequate stretch around the wheels as well.

FINAL 04 — Design Showcase: Vehicle Final & Obstacle Course

Finished Vehicle
Diana Fan
DESINV 22 — Summer 2019

Contents:

Product Video

Link to video: https://youtu.be/gzJ1B6Sq9xc

Product Shot
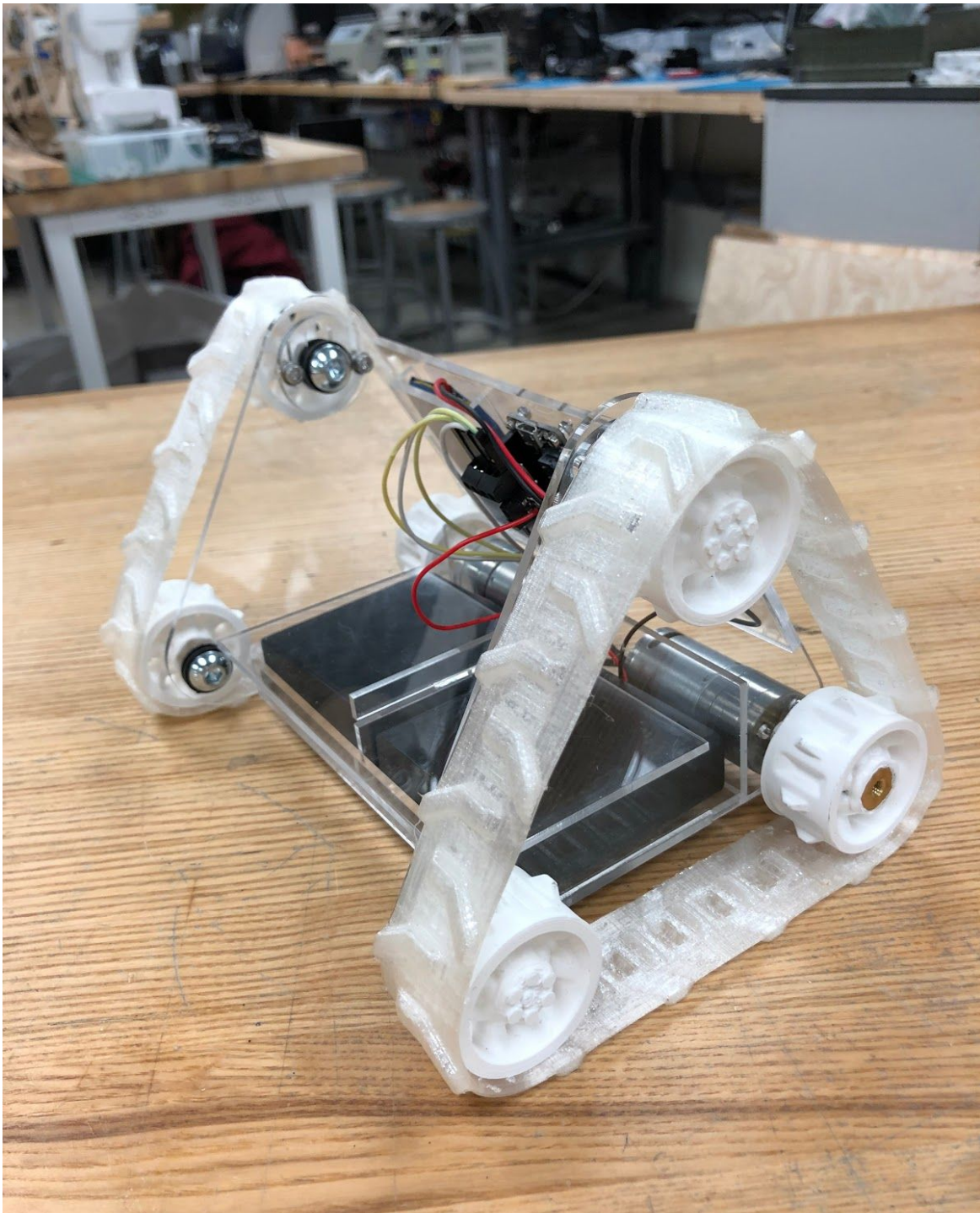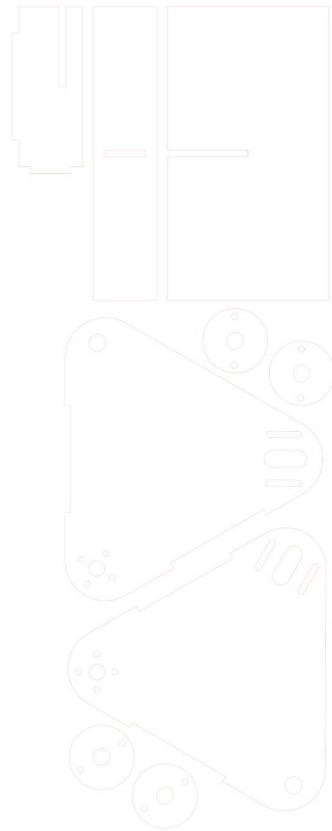
Process Photos

Photo A: Illustrator cut file for the acrylic chassis.
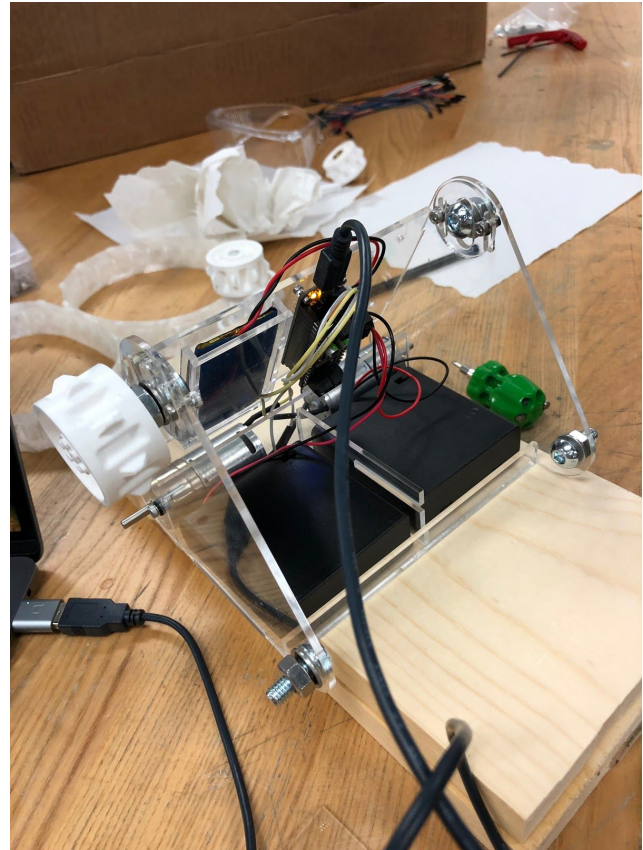
Photo B: Working out the configuration of the electronics.

Bill of Materials

Item	Description	Quantity
Cast Acrylic	⅛" thickness, 11.5" x 24", bought from Jacobs	2
PLA	white, 3D print material for wheels	
Ninjaflex	clear, 3D print material for treads	
Nuts	to affix the wheels and screws onto acrylic base	4
Large Screws	to affix the wheels onto acrylic base	4
Small Screws	4 for the adjustable wheel on top, 4 to affix the motors	8

Plastic Washers	black, 2 per non-motorized wheel	8
Feather 32u4 Bluefruit LE	Arduino-compatible, comes with Bluetooth control	1
Motor Featherwing	provides power to the motors, provided by Chris	1
DC Motors	provided by Chris	2
Lithium Ion Battery	to power the Feather with, provided by Chris	1
AA Batteries	batteries to power the motor with	8
4xAA Battery Holder	portable battery holder with an on/off switch	2
Jumper Wires	for connections	4

Tool	Description	Quantity
Laser Cutter	Universal VLS 6.60, machine in Jacobs	1
Allen Wrenches	various sizes	
Screwdrivers	various sizes	
Needle Nose Pliers	to grip with	1
Solder	provided by the Invention Lab	
Soldering Iron	provided by the Invention Lab	
Acrylic Cement	to bond pieces of acrylic with	
Teflon	used to lubricate joints	
Arduino	software for coding electronics	
Adobe Illustrator	software used to create cut files	

Process and Description

When moving from the plywood to acrylic vehicle, we wanted to note several things we had to change. One was that we needed to figure out a way of mounting the

electronics without zip ties as required in the specifications of the assignment. My partner and I decided to create a small boxes with finger joins to hold the lithium ion battery, make containers in the bottom to hold the AA battery cases, and use small screws and nuts to mount the Featherwing. In the meanwhile, I had already begun 3D printing wheels and treads (credit: https://www.thingiverse.com/thing:430305) for the final vehicle. We kept the original sizes of the wheels, but realized that they did not fit our project parts entirely—namely, the motor. Chris suggested that we drill to make the holes slightly larger and thread them so that they could pass through screws that would go through the acrylic body. We used a heat gun to heat up the brass tube and pass it through the PLA of the to-be-motorized wheels. The treads were printed in Jacobs on a Type A machine.

We also decided it would be a good idea to have the top wheel be modular, so we could slide it up and down if we needed to adjust the tension in the treads. It took us a while to also look for appropriately fitting parts, but after wiring up the electronics and setting them in the chassis, we attached the wheels. However, we ran into a major problem: the code was working fine and so were the motors, but when the tread was put on, the vehicle could only move forward a few seconds before stalling and stopping. We figured out that the wheels were being pulled down with tension, preventing them from turning. This was also because they were uneven, and not completely parallel to the ground. I fixed it by adding washers to the inside and outside of the chassis, and replaced them with plastic for a smoother torque. We also added some Teflon for insurance, and finally, our vehicle was finished!

FINAL 05 — Final Writeup and Video

Final Vehicle
Diana Fan
DESINV 22 — Summer 2019

Contents:

Product Video

Link to video: https://youtu.be/pf7ApJZG-e8

Final Product Shot (Refer to FINAL 04 Product Shot)

Process Photos (Refer to FINAL 01 - 04 Process Photos)

Bill of Materials (Refer to FINAL 04 BOM)

Reflection

Initially, I was absolutely terrified of this project because like with most assignments before, I never had any experience working with electronics, motors, and engineering. I learned that prototyping is a slow process, and it's okay to start over if an idea didn't entirely work out in your head, even though you really wanted it to. Even now there are things I want to change, like making the acrylic body lighter, or less fragile because after several runs through the course the acrylic cement wasn't holding up the joints very well and we had to re-bond them. But even though it was a lot of work and time spent in the lab, my partner and I were really satisfied with the outcome. This class was pretty eye-opening overall because as a student here, I really don't get to work with my hands a lot, and didn't really know that we had the resources here on campus to build whatever we wanted. It was super fun learning how to use the machines, and I know that this is kind of not really a reflection about the final project anymore, but I just wanted to extend my thanks to the staff for this class and in the lab, because I definitely found something fun to do this summer. And even though I'm not taking a class that requires a Makerspass this fall, I definitely still want one so I can keep creating in my own time. Thanks!

Bibliography

Special thanks to the creators of these resources:
- https://learn.adafruit.com/my-mini-race-car
- https://www.thingiverse.com/thing:430305