ME 102B Final Project Report

Group 3: El Piñatatron

Constance Angelopoulos, Troy Keslinke, Romy Mastel, Anthony Moody

Device Opportunity

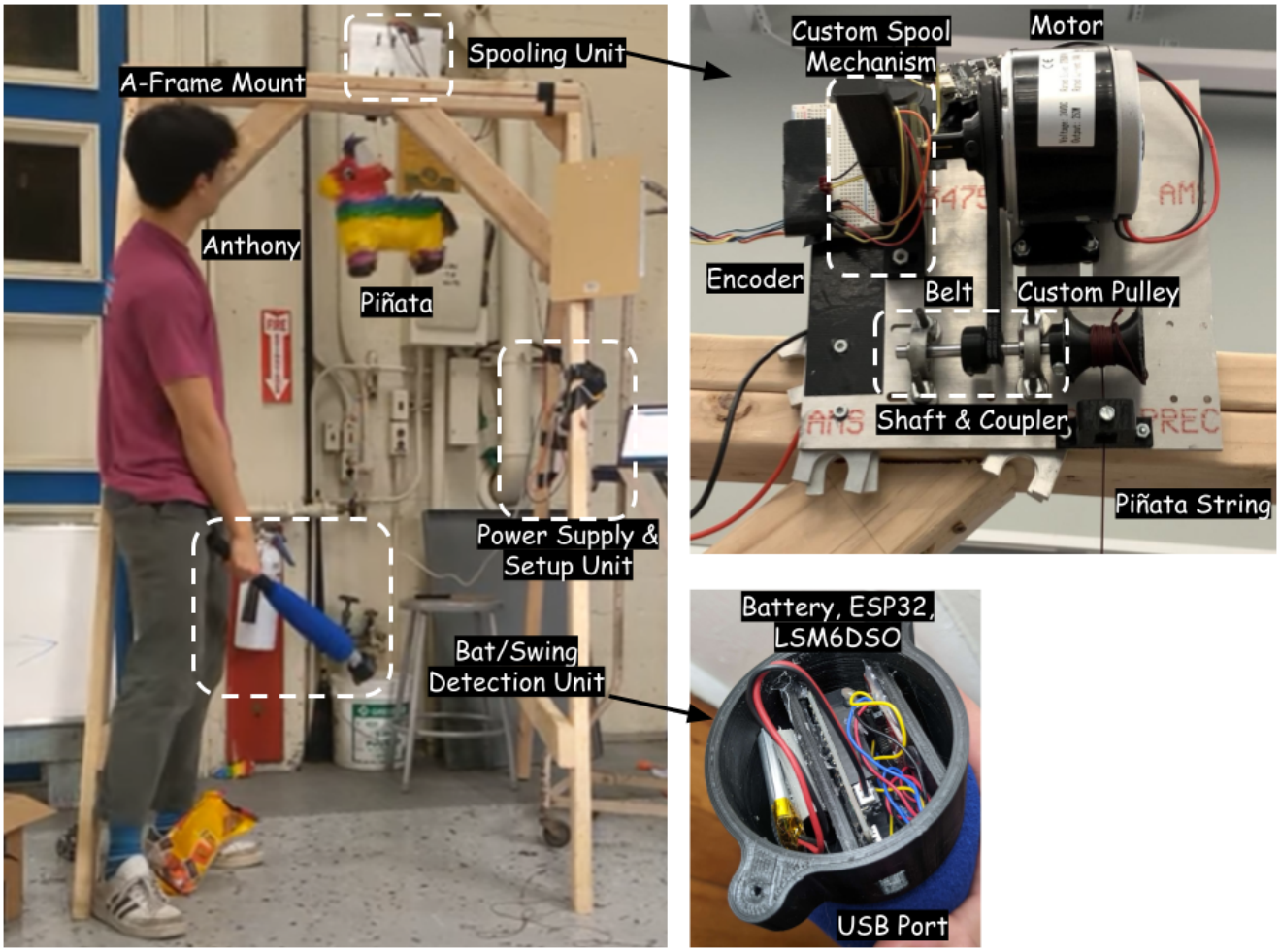
Maneuvering a pinata in a way that is fun for varying age groups and skills can be challenging; a deft operator may make the piñata unreachable for a toddler, and a too-slow reaction can result in a piñata that is decimated in a single swing from an overzealous uncle. Additionally, piñatas require at least one guest to maneuver, preventing some party guests from enjoying the activity.

Device High-Level Strategy Overview

Using a threshold readout from an accelerometer on the users' bat, the pinata can 'dodge' a swing, or be quickly reeled upward on a motorized pulley.

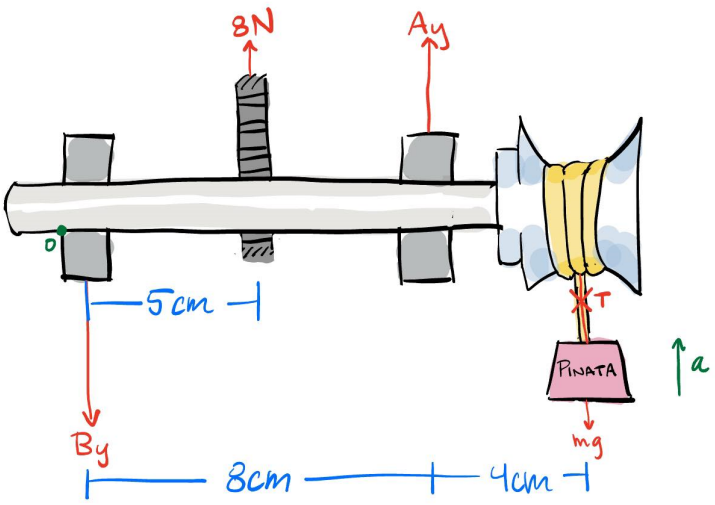
The initial scope of this project included a potentiometer affixed to the bat handle that allowed for a user to select between three different 'difficulty' settings, with difficulty being determined by two attributes: accelerometer threshold (or, the reaction speed of a dodge in response to a swing), and 'teasing' state height modulation frequency (or, how often the piñata moves in between swings).

Device Diagram



Function-Critical Decisions

1. Swing Detection: The system detects a swing as an accelerometer readout that exceeds a defined threshold.
2. Spooling Mechanism:



Desired Acceleration Calculations

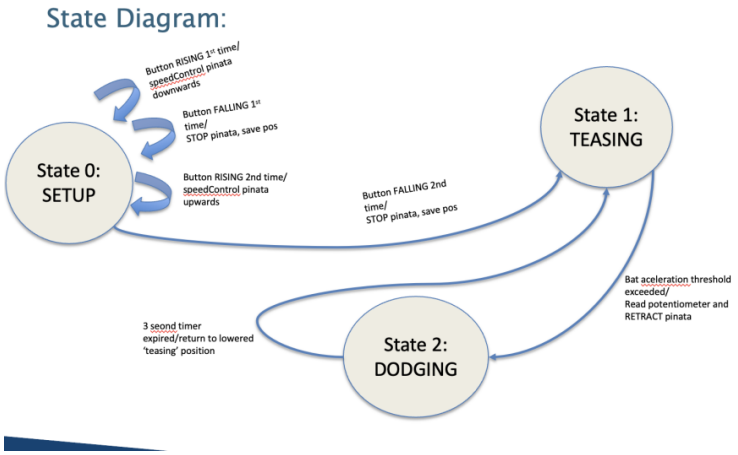
High end of bat speed for 6 year old: 20mph = 8.94m/s Swinging from 1 meter away
Pinata length of 12in = .3 meters Weight = 2lb -----> Mass = .907kg
If aiming at middle of pinata:
D = rt ----> 1m = 8.94m/s -----> t = .11186s h = .5at^2 ------> 2(.3/2)/(.112^2) = a = 23.9m/s

Bearing Calculations

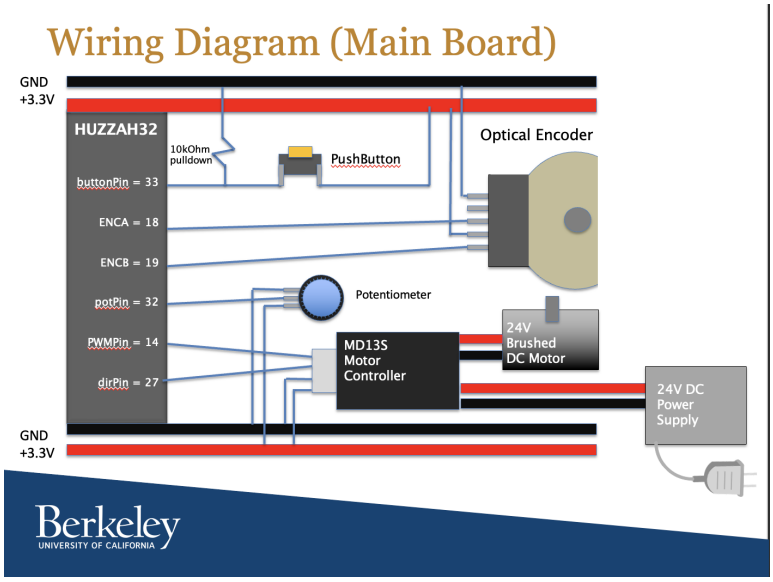
T - mg = ma ------> T = m(g+a) = .907(9.8 + 23.9) = 30.6N
0 = Net(Fy) = 8N - T + Ay - By = Ay - By - 22.6N
0 = Net(Mo) = 5(8N) + 8Ay - 12T ----> (367.2N -40N)/8 = Ay = 40.9N
By = 40.9N - 22.6N ------> By = 18.3N

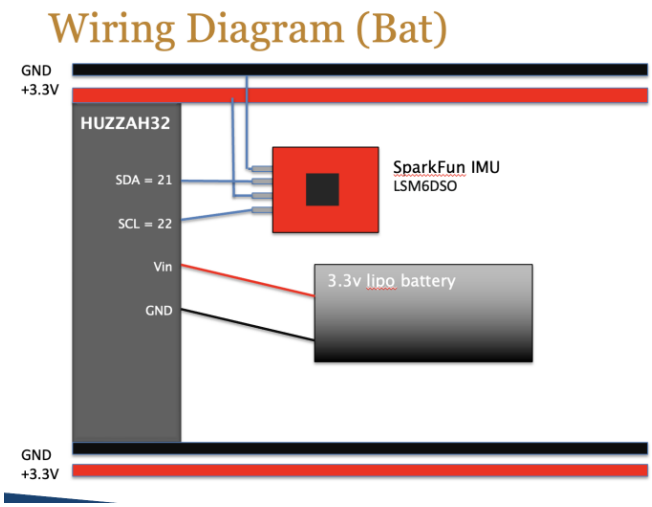
3. Bat-To-Motor Communication: The motor's ESP32 communicates with the bat's ESP32 through
4. Mounting: The spooling unit is mounted on a wooden A-frame, inspired by a playground swing set. This gives the device both the height and stability necessary to operate.

State Transition Diagram



Circuit Diagram





Reflection

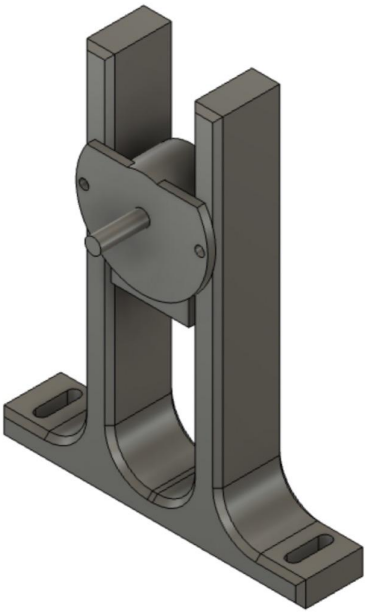
A strategy that worked well for our team was delegating specific tasks to each member based on their strengths, and recording these tasks in a running agenda document. Our group would have benefitted from a more strict internal timeline.

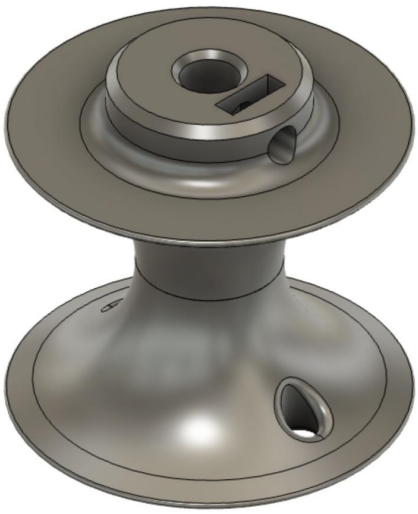
Appendix

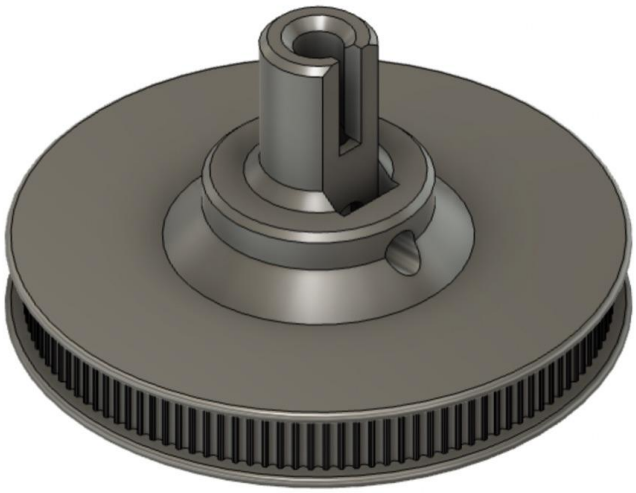
Bill of Materials

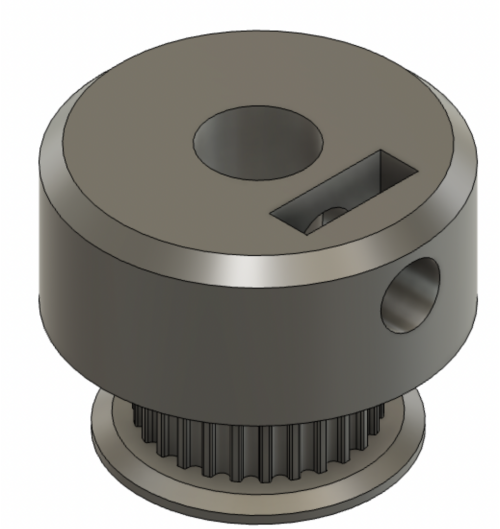
Total Cost	$554.37									
Item	**Comments**	**Status**	**Subgroup**	**Listed Product Name**	**Part #**	**Link**	**Vendor**	**$/Item**	**Order Qty**	**Cost**
Foam Bat		Arrived	Parts	CeleMoon Super Safe Soft 22 Inch Kids Foam Ba	N/A	https://www.amazon.com/CELEMOON-In	Amazon	$17.99	1	$17.99
Tiny Breadboard		Arrived	Parts	Tiny Premium Breadboard	65	https://www.adafruit.com/product/65#tech	Adafruit	$3.95	1	$3.95
IMU		Arrived	Parts	BMI088 Shuttle Board 3.0	262-SB3.0BMI088	https://www.mouser.com/ProductDetail/Bc	Mouser	$33.35	1	$33.35
3.7 V Battery	Already in our kit	Arrived	Parts	LITHIUM ION POLY BATT 3.7V 290MA	1832-1032-ND	https://www.digikey.com/en/products/deta	Digikey	$5.95	0	$0.00
1/4"-20 Bolt	Uswed wood screw instead	Rejected	Parts	Black-Oxide Alloy Steel Socket Head Screw	91251A542	https://www.mcmaster.com/91251A542/	McMaster	$11.02	1	$11.02
1/4"-20 Nut		Rejected	Parts	Medium-Strength Steel Hex Nut	95462A029	https://www.mcmaster.com/95462A029/	McMaster	$8.95	1	$8.95
3D Printer PLA	For 3D Printing Bat Housing	Arrived	Materials	SUNLU 3D Printer Filament PLA Plus 1.75mm, PL	N/A	https://www.amazon.com/PLA-Filament-S	Amazon	$16.79	0	$0.00
Pulley	3D Printed Instead	Rejected	Parts	Pulley for Wire Rope-for Lifting	3099T14	https://www.mcmaster.com/3099T14/	McMaster	$15.25	2	$30.50
Long Frame Rails (Sides)		Rejected	Parts	Single Four Slot Rail, Silver,	47065T123	https://www.mcmaster.com/47065T101-47	McMaster	$48.91	2	$97.82
End-Feed Nut with Flange		Rejected	Parts	End-Feed Nut with Flanged-Button Head	47065T142	https://www.mcmaster.com/47065T142/	McMaster	$3.30	7	$23.10
Ball Bearing		Arrived	Parts	Open, Trade Number 6001, for 12 mm Shaft Diam	5972K95	https://www.mcmaster.com/5972K95	McMaster	$22.09	2	$44.18
Short Frame Rails (Base)		Rejected	Parts	Single Four Slot Rail, Silver, 1" High x 1" Wide, Sc	47065T411	https://www.mcmaster.com/47065T101-47	McMaster	$8.97	8	$71.76
Medium Frame Rails (Top)		Rejected	Parts	Single Four Slot Rail, Silver, 1" High x 1" Wide, Sc	47065T413	https://www.mcmaster.com/47065T101-47	McMaster	$29.23	1	$29.23
Silver Gusset Bracket		Rejected	Parts	Silver Gusset Bracket, 1" Long for 1" High Rail T-S	47065T663	https://www.mcmaster.com/47065T663/	McMaster	$9.97	8	$79.76
2X4s	For the Frame	Arrived	Parts	2x4 Stud	N/A	N/A	Home Depot	$3.67	9	$33.03
Wood Screws	For Frame	Arrived	Parts	2.5" Wood Screws	N/A	N/A	Home Depot	$11.97	1	$11.97
Wood Glue		Arrived	Materials	Wood Glue	N/A	N/A	Home Depot	$4.97	1	$4.97
Turnigy Brushless Motor	Anthony already has it	Arrived	Parts	Turnigy 80-100-A 180Kv Brushless Outrunner	TR80-100-A	https://hobbyking.com/en_us/turnigy-80-1	HobbyKing	$99.99	0	$0.00
YaeTek Motor	New Motor	Arrived	Parts	YaeTek 24V Electric Motor Brushed 250W 2650RF	N/A	https://www.amazon.com/gp/product/B07I	Amazon	$32.88	1	$32.88
Multipurpose 6061 Aluminum	.5ft option, For Mounting Bracket, Waterjet	Arrived	Materials	Multipurpose 6061 Aluminum, 1/4" Thick x 2-1/4" \	8975K599	https://www.mcmaster.com/8975K599-89	McMaster	$4.91	1	$4.91
Cable Rope		Rejected	Parts	Wet-Environment Rope	3837T16	https://www.mcmaster.com/3837T16	McMaster	$0.05	100	$5.00
Cord		Arrived	Parts	Micro Cord	N/A	N/A	ACE	$10.00	1	$10.00
Pinata	Can be switched to any pinata	Arrived	Parts	GIFTEXPRESS 16-Inch Pink Unicorn Pinata for K	N/A	https://www.amazon.com/Donkey-Pinata-	Amazon	$17.99	1	$17.99
Solderless breadboard	Provided in ME102B kit	Arrived	Parts		N/A	N/A		$0.00	1	$0.00
ESP32 Microcontroller	Provided in ME102B kit	Arrived	Parts		N/A	N/A		$0.00	2	$0.00
Jumper wires	Provided in ME102B kit	Arrived	Parts		N/A	N/A		$0.00	1	$0.00
MicroUSB cable	Provided in ME102B kit	Arrived	Parts		N/A	N/A		$0.00	1	$0.00
Pushbutton switch	Provided in ME102B kit	Arrived	Parts		N/A	N/A		$0.00	1	$0.00
Potentiometer	Provided in ME102B kit	Arrived	Parts		N/A	N/A		$0.00	1	$0.00
Encoder and connector	Provided in ME102B kit	Arrived	Parts		N/A	N/A		$0.00	1	$0.00
LSM6DSO	Provided in ME102B kit	Arrived	Parts		N/A	N/A		$0.00	1	$0.00
24V Power supply	Borrowed from Tom :)	Arrived	Parts		N/A	N/A		$0.00	1	$0.00

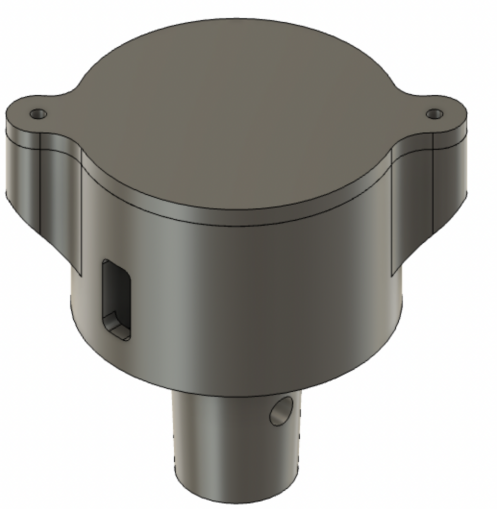
CAD Models

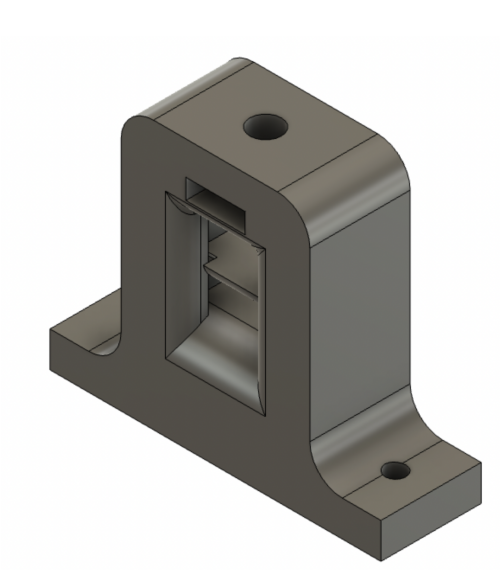


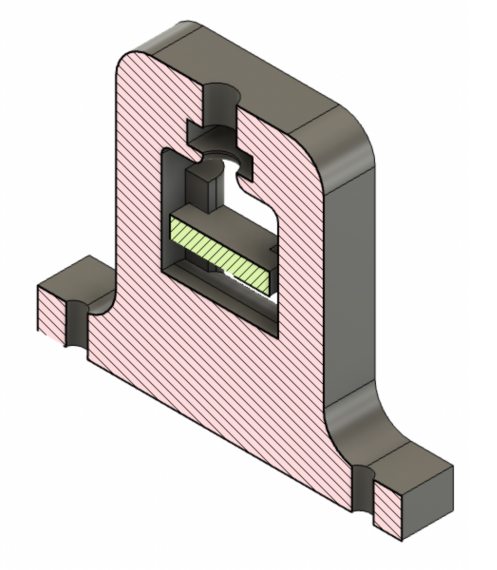












Code

1. **Motor Microcontroller Code**

```
// define pins for button, dc motor pwm pin, dc motor direction pin, encoder pins
#define buttonPin 33
#define ENCA 18 // red
#define ENCB 19 // yellow
#define dirPin 27
#define POT 32
float kp = .5;

int state = 0; // must define here because my wireless communication code references this
variable

int potReading = 0;

// wireless communcation - credit to ESPNOW guy on youtube
#include <esp_now.h>
#include <WiFi.h>

typedef struct struct_message {
    double a;

} struct_message;
struct_message myData;
float lastDodgeTime = millis();

// callback function that will be executed when data is received
void OnDataRecv(const uint8_t * mac, const uint8_t *incomingData, int len) {
  memcpy(&myData, incomingData, sizeof(myData));

  if (myData.a > 2){
    Serial.print("dodging state activated");
    lastDodgeTime = millis();
    state = 2;
  }
}
void setupWirelessCommuncation(){
  WiFi.mode(WIFI_STA);

  // Init ESP-NOW
  if (esp_now_init() != ESP_OK) {
    Serial.println("Error initializing ESP-NOW");
    return;
  }
```

```
  // Once ESPNow is successfully Init, we will register for recv CB to
  // get recv packer info
  esp_now_register_recv_cb(OnDataRecv);
}
//


int numButtonPresses = 0;

// motor properties
const int PWM = 14;
int setupMotorSpeed = 30; // pwm value for setup phase
const int freq = 20000;
const int ledChannel = 0;
const int resolution = 8;
bool messageDisplayed = false;
int highestPosition = 0;
int lowestPosition = 0;
int lowestTease = lowestPosition;
int highestTease = highestPosition;
float currentTime = millis();
float lastTeaseTime = 0;
float waitPeriod = 3000;
float teasePosition;

//setupEncoder
int aState;
int aLastState;
int lastPrint = millis();
  int printWaitTime = 1000;


// PID control variables
volatile int motorPosition = 0;
long prevT = 0;
float eprev = 0;
float eintegral = 0;

bool buttonPressed = false;
bool buttonNeedsChecking = false;
int lastDebounceTime = 0;
int lastButtonState = 0;
int debounceDelay = 50;
```

```
bool atTop = false;

void IRAM_ATTR dodgeFlag(){
  state = 2;
}
void IRAM_ATTR buttonFlag(){
buttonNeedsChecking = true;
lastButtonState = digitalRead(buttonPin);

}

void checkButton(){
  if ((millis()-lastDebounceTime) > debounceDelay){
    buttonNeedsChecking = false;
      if (digitalRead(buttonPin) == lastButtonState){ // if the current button state is the same as the
state when the interrupt was triggered
        numButtonPresses++;
        Serial.print(numButtonPresses);

    }
    lastDebounceTime = millis();
  }

}
void setupledc(){
  ledcSetup(ledChannel, freq, resolution);
  ledcAttachPin(PWM, ledChannel);
}
void printMotorPosition(){

  if ((millis()-lastPrint) > printWaitTime){
    Serial.println(motorPosition);
    lastPrint = millis();
  }
}
void setupMotor(){

  setupledc();
  setupEncoder();
```

```
//  pinMode(PWM,OUTPUT); this was causing ledc not to output pwm signal
  pinMode(dirPin,OUTPUT);
}

void IRAM_ATTR readEncoder(){
  aState = digitalRead(ENCA);
  if (aState != aLastState){
    if (digitalRead(ENCB) != aState){
      motorPosition--;

    }
    else {
      motorPosition++;
    }

  }
  aLastState = aState;
}

void setupEncoder(){
  pinMode(ENCA,INPUT);
  pinMode(ENCB,INPUT);
  aLastState = digitalRead(ENCA);
  attachInterrupt(digitalPinToInterrupt(ENCA),readEncoder,CHANGE);
}




void setup() {
  Serial.begin(115200);
  setupMotor();
  setupButton();
  setupWirelessCommuncation();
  pinMode(POT, INPUT);


}
void displayButtonPresses(){ // not working lol
  bool buttonpressdisplayed = false;
  int lastNumButtonPresses = 0;
  if (lastNumButtonPresses != numButtonPresses){
    buttonpressdisplayed = false;
  }
```

```
  if (!buttonpressdisplayed){
    Serial.print("num of button presses: ");
    Serial.println(numButtonPresses);
    buttonpressdisplayed = true;
   lastNumButtonPresses = numButtonPresses;
  }
}


void loop() {

if (buttonNeedsChecking){
  checkButton();
}
  switch (state) {
    case 0:

      if (!messageDisplayed){
        Serial.println("case 0, setup");

        messageDisplayed = true;
      }
      if (numButtonPresses == 1){
        setupMotorSpeed = 30;
        speedControl(1);
      }

      if (numButtonPresses == 2){
          ledcWrite(ledChannel,0);
          lowestPosition = motorPosition;
      }
      if (numButtonPresses == 3){
        setupMotorSpeed=50  ;
        speedControl(0); // run speed control with reverse direction
      }
      if (numButtonPresses == 4){
        ledcWrite(ledChannel,0);
        highestPosition = motorPosition;
        Serial.print("Highest Position: "); Serial.println(highestPosition);
        Serial.print("Lowest Position: "); Serial.println(lowestPosition);
        messageDisplayed = false;
        state = 1; // start teasing state

      } break;
```

```
    case 1:
      if (!messageDisplayed){
        Serial.println("case 1, teasing");
        messageDisplayed = true;
        setupTease();
      }
      tease();
      break;

  case 2:
      Serial.println("state 2: DODGING");
      motorPIDWrapper(highestPosition);
      if (motorPosition > (highestPosition-50)){
        if (millis()-lastDodgeTime > 3000){
          state = 1;
        }
      }
      break;

  }

}
void setupTease(){
  lowestTease = lowestPosition;
  highestTease = lowestTease + .65*(highestPosition-lowestPosition); // highest position it will go
in tease state is half way between lowestPosition and HighestPosition (i know i could've just
averaged them, but with the .5 in front i can easily change it to change the range at which it will
move around)
  Serial.print("lowest position:" ); Serial.println(lowestPosition);
  Serial.print("highest position:" ); Serial.println(highestPosition);
  Serial.print("lowest tease:" ); Serial.println(lowestTease);
  Serial.print("highest tease:" ); Serial.println(highestTease);
  teasePosition = motorPosition;
  lastTeaseTime = millis();

}
void tease(){
    currentTime = millis();
  motorPIDWrapper(teasePosition);
  if ((currentTime - lastTeaseTime) > waitPeriod){
    if (atTop==true){
      teasePosition = random((lowestTease+.5*(highestTease-lowestTease)), highestTease);
        kp = .3; //makes the motion slow
        atTop = false;
```

```
    }
    else if (atTop == false){
        kp = 1;
        teasePosition = 0.9*highestPosition;
        atTop = true;
    }
    lastTeaseTime = millis();
    waitPeriod = random(250,3000);// random amount of time between 0.25 and 3 second; (millis)
  }
}

void speedControl(int setupdir){ // dir is 0 or 1
  if (setupdir == 1){
      digitalWrite(dirPin, HIGH);
    }
    else if (setupdir == 0){
      digitalWrite(dirPin, LOW);

    }
    else{
      Serial.print("dirPin Error");
    }
    ledcWrite(ledChannel,setupMotorSpeed);


}

void setupButton(){
  pinMode(buttonPin, INPUT);
  attachInterrupt(buttonPin,buttonFlag,CHANGE);
}



// PID CONTROL FUNCTIONS

float computeU(int target){

  potReading = analogRead(POT);
  if (potReading < 4095/2){
     kp = 0.9; //easy mode
  }

  else{
```

```
    kp = 1.5; //hard mode
  }


  //float kp = .5; Original Code
  float kd = 0.025;
  float ki = 0.5;

  long currT = micros();
  float deltaT = ((float) (currT - prevT))/( 1.0e6 );
  prevT = currT;

  int pos = 0;
  noInterrupts();
  pos = motorPosition;
  interrupts();

  int e = target - pos;

  float dedt = (e-eprev)/(deltaT); // d(e)/(dt)

  eintegral = eintegral + e*deltaT;

  float u = kp*e + kd*dedt + ki*eintegral;
  eprev = e;
  return u;
}


void controlMotor(int dir, int pwmVal, int pwmPin){

  if(dir == 1){
    digitalWrite(dirPin,HIGH);
  }
  else if(dir == -1){
    digitalWrite(dirPin,LOW);
  }
  else{
    pwmVal = 0;
    Serial.println("else statement triggered setMotor");
  }
  ledcWrite(ledChannel,pwmVal);

}
```

```
void motorPIDWrapper(float target){ //accepts float as target position, then PID controls motor to that position
  float pwr = computeU(target);
  int dir = 1;
  if(pwr<0){
    dir = -1;
  }
  pwr = fabs(pwr);
  if(state == 2){
    if(pwr > 100){
      pwr = 100;
    }
  }
  else if(dir == 1){ //we want the power to be higher in upwards direction
    if(pwr > 10){
      pwr = 10;
    }
  }
  else if(dir == -1){
    if(pwr > 50){
      pwr = 50;
    }
  }


  int PWR = pwr; // convert to integer to send to controlMotor function

  controlMotor(dir,PWR,PWM);
}
```

2. **Bat Microcontroller Code**

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

// all IMU Code is licenced from from opensource.org/licenses/MIT
#include "SparkFunLSM6DSO.h"
#include "Wire.h"
#include "Math.h"
double AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
double totalAccel;

LSM6DSO myIMU; //Default constructor is I2C, addr 0x6B

// end IMU stuff

// REPLACE WITH YOUR RECEIVER MAC Address
uint8_t broadcastAddress[] = {0x7C, 0x9E, 0xBD, 0xD8, 0x5B, 0xFC};


// Structure example to send data
// Must match the receiver structure
typedef struct struct_message {
  double a;
} struct_message;

// Create a struct_message called myData
struct_message myData;

esp_now_peer_info_t peerInfo;

// callback when data is sent
void OnDataSent(const uint8_t *mac_addr, esp_now_send_status_t status) {
  Serial.print("\r\nLast Packet Send Status:\t");
  Serial.println(status == ESP_NOW_SEND_SUCCESS ? "Delivery Success" : "Delivery Fail");
}

void setupIMU(){
  Serial.begin(115200);
  delay(500);

  Wire.begin();
  delay(10);
  if( myIMU.begin() )
    Serial.println("Ready.");
```

```
  else {
    Serial.println("Could not connect to IMU.");
    Serial.println("Freezing");
  }

  if( myIMU.initialize(BASIC_SETTINGS) )
    Serial.println("Loaded Settings.");
}

void readIMU(){
  //Get all parameters

  AcX = myIMU.readFloatAccelX();
  AcY = myIMU.readFloatAccelY();
  AcZ = myIMU.readFloatAccelZ();
  GyX = myIMU.readFloatGyroX();
  GyY = myIMU.readFloatGyroY();
  GyZ = myIMU.readFloatGyroZ();
  totalAccel = sqrt(AcX*AcX + AcY*AcY + AcZ*AcZ);

}
void setup() {
  // Init Serial Monitor
  Serial.begin(115200);


  // setup imu
  setupIMU();

  // Set device as a Wi-Fi Station
  WiFi.mode(WIFI_STA);

  // Init ESP-NOW
  if (esp_now_init() != ESP_OK) {
    Serial.println("Error initializing ESP-NOW");
    return;
  }

  // Once ESPNow is successfully Init, we will register for Send CB to
  // get the status of Trasnmitted packet
  esp_now_register_send_cb(OnDataSent);

  // Register peer
  memcpy(peerInfo.peer_addr, broadcastAddress, 6);
```

```
  peerInfo.channel = 0;
  peerInfo.encrypt = false;

  // Add peer
  if (esp_now_add_peer(&peerInfo) != ESP_OK){
    Serial.println("Failed to add peer");
    return;
  }
}

void loop() {
  readIMU();
  // Set values to send

  // Send message via ESP-NOW
  esp_err_t result = esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));

  if (result == ESP_OK) {
    Serial.println("Sent with success");
  }
  else {
    Serial.println("Error sending the data");
  }
  Serial.println(myData.a);
//  delay(500);
}
```

Additional Graphics

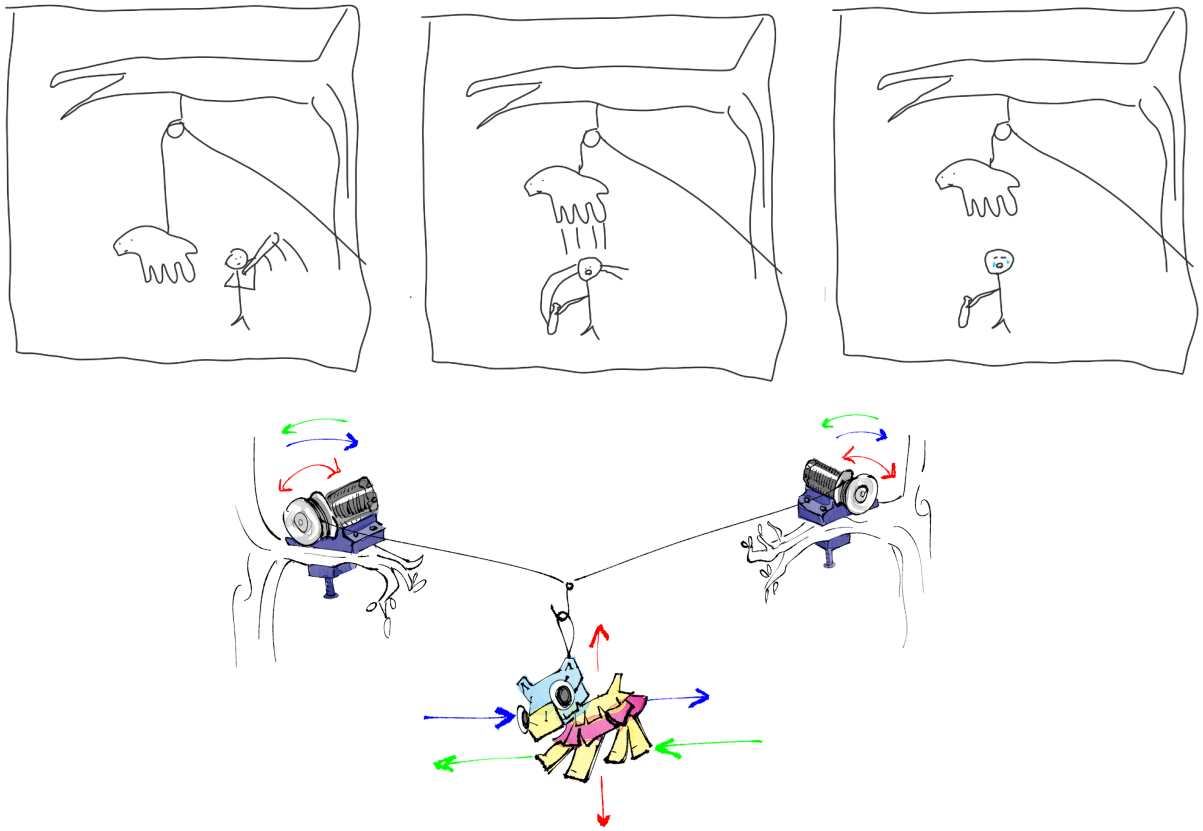


Fig. 1.1, 1.2, 1.3: Initial low-fidelity design sketches.