Computer Aided Design and Prototyping

(ME444 – SPRING 2024)

Thrill on the Hill Mini

Design Report

Group No. C6

Mariah Alsinan	Senior - ME
Everett Lamberson	Junior – ME
Chad Matulenko	Junior – ME
Heather Mello	Junior - MDE

School of Mechanical Engineering
Purdue University

Revision History

S. No.	Date	Revision ID	Revision Details (Page No., Paragraph, Line No. etc.)	Author
1.	4/28/2024	Revision A	Initial Release	MA, EL, CM, HM

***

Keeping track of the revisions you've made to a frequently updated technical or business document is important. A document revision history table will save you a lot of headaches when it is time to send out your document for a review.

If the document is just a few pages long, then asking your reviewers to read the "whole document" and give you a feedback is not a big deal. No one will complain about that. But imagine you've finished the fourth draft of a 2,000-page document with 250 tables and 360 figures!

You need to keep a track of the changes made and record them systematically. A Revision History Table is a good way to do so…

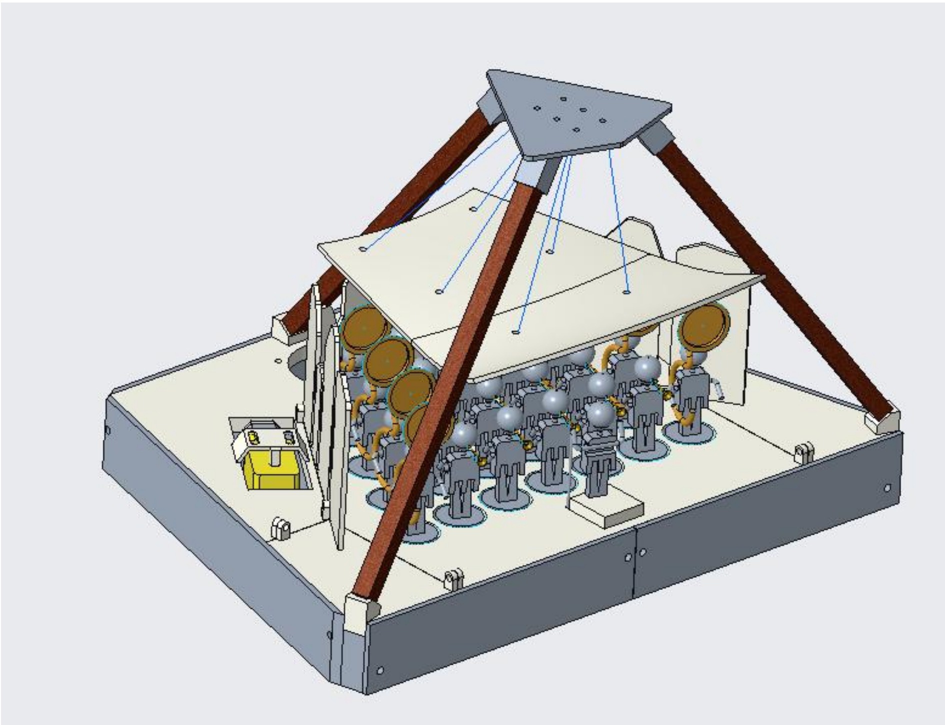
If you are interested to learn more - https://www.linkedin.com/pulse/how-design-document-revision-history-template-ugur-akinci

Executive Summary

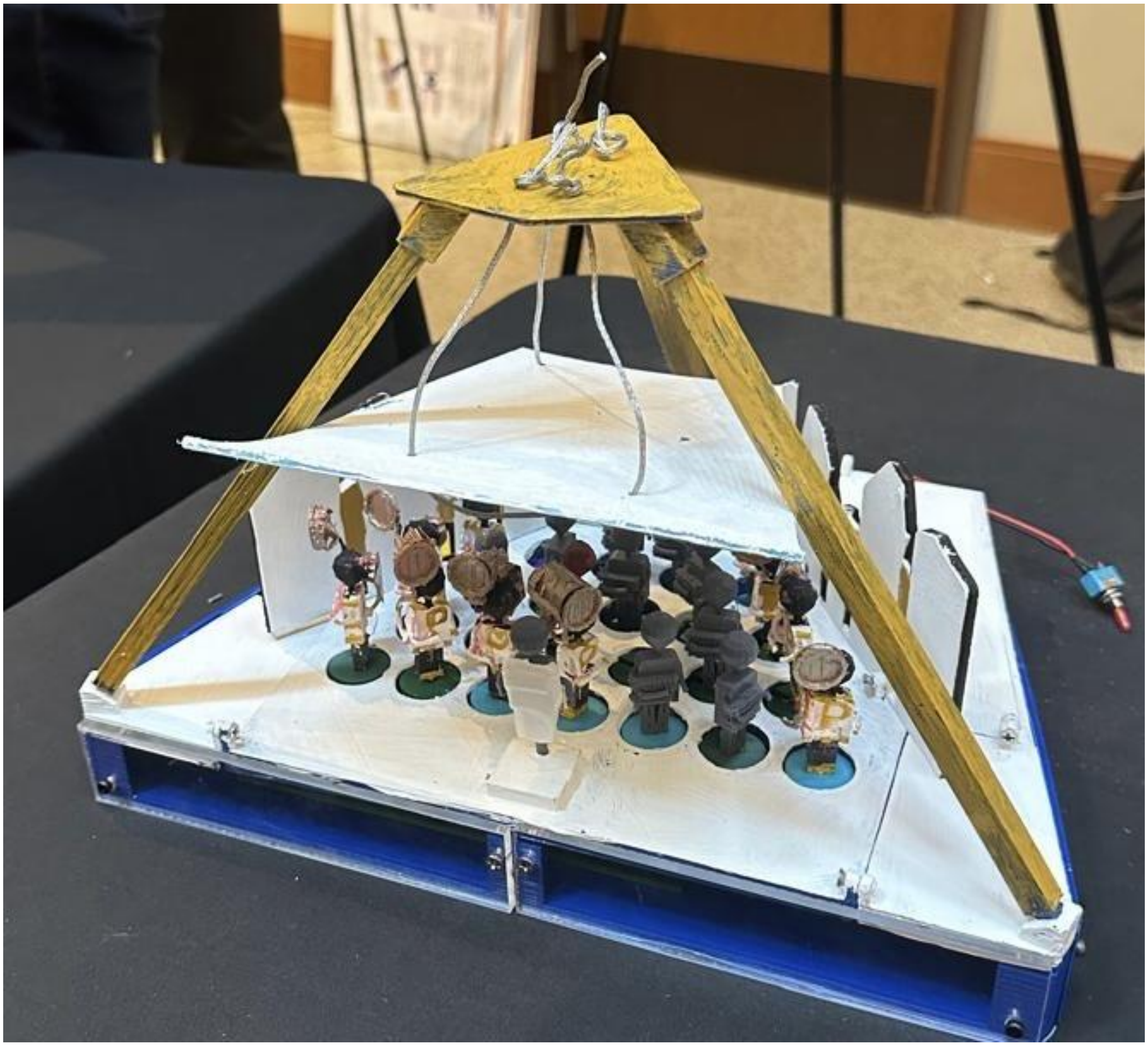
The goal of this project was to create a mechanically functioning diorama of the Purdue University "All-American" Marching Band during a concert known as "Thrill on the Hill". The market potential for this product is very high, as there are limited marching band toys, and none that are Purdue themed or aimed at the growing Purdue fan-base. The two main mechanical functions are the rotations of the band members, known as "swaggering," and the conducting motion of the drum major. The swaggering is achieved through a modified four-link system, where one link completes rotations, and one link is constrained to move through a 60-degree angle. The conducting motion is achieved through a rod moving in a track, which moves back and forth under the stage. For electronics, there was a brushed DC motor connected to a motor controller, connected to an ESP32 Arduino board to control the speed of the motor, along with a 9V battery and a toggle switch. Most parts were 3D printed and painted, with the stage walls and front panels being laser cut out of wood and clear acrylic, respectively. The main design flaw with this toy is that there was a great amount of torque required to move the gears under the stage due to oversized gears and binding of gears.

Final toy assembly can be found at the following file location:

$/ME444/Group 29/Action_Toy/CAD/rough_slayter_stage.asm



CAD Rendering of the completed toy



Prototype presented at the ME 444 toy fair

Table of Contents

Contents

1. Introduction 6
   1.1 Project Backgrounds 6
   Design Requirements & Constraints 7
2. Concept Generation and Selection 8
3. Detail Design 9
   3.1 Sub-system 1: Rotators 9
   3.2 Sub-system 2: Drum Major Conducting 10
   3.3 Sub-system 3: Electronic Systems 11
   3.4 System Integration 12
4. Prototyping and Testing 13
5. Results and discussion 13
6. Conclusion 14
7. References 14

1. Introduction

1.1 Project Backgrounds

Every home football gameday, the Purdue "All-American" Marching Band performs a concert known as "Thrill on the Hill" 90 minutes before kickoff at the Slayter Hill. This toy, "The Thrill on the Hill Mini", is a diorama representation of this concert. The movements of this toy include band members moving their instruments back and forth with the rhythm of Hail Purdue, known as swaggering, and a drum major conducting the band. The toy is battery powered and turns on using a switch. Another goal within this project was to have the band members' movements synced with the music of "Hail Purdue", with the music also turning on with the switch.

The toy uses a modified four-arm linkage system to turn complete revolutions of one link into 60-degree oscillations in another link, detailed in Section 3.1. The drum major's arms are moved up and down via a rod connected to a track under the stage. The track is translated linearly using a linkage to a rotating disk. This subsystem is detailed in Section 3.2. These two subsystems are connected through an idler gear and a gear ratio reduction.

The target age group for this toy would be anyone who is old enough to have an interest in marching band, so likely middle-school aged children (11-14) and up. Although this toy would have a moderately niche target group, there are very few, if any, automated marching band toys on the market today. Most available marching band toys are similar to the ones seen below in Figures 1.1 and 1.2. These toys are either action figures or simple dioramas, neither of which are automated nor contain electrical components. This toy, the Thrill on the Hill Mini, is innovative and has high market potential because of the fact that it is automated and other similar toys do not exist as well as the fact that it could be used to target the very large Purdue sports fan-base.

Figure 1.1: Marching Band Figurines

Figure 1.2: Marching Band Figurine Diorama

Design Requirements & Constraints

The objective was to design a mechanical or mechatronic toy that has two or more non-trivial functions. The power source of the toy was left ambiguous, with suggestions and past examples being operated using batteries, wind-up springs, and hand cranks. The virtual prototype must have models including both designed and purchased parts and must simulate the toy's real actions using the mechanism's motion simulation. These designed parts mainly be made using rapid prototype processes, including 3D printing and laser cutting. The overall size of the toy must be 10x10x10 in^3, although this requirement is less strict than the other constraints. Lastly, any purchased parts must be purchased with the ME credit card, with a limit of $70. Parts owned by a group member at the start of the semester and donated to the project, as well as parts from the guided project, may be used.

2. Concept Generation and Selection

Another alternate concept generated for the toy was called “Runner’s Rivalry”. This toy would have been a competition-based track runner game inside of a stadium. The main features of the toy are the runners who are attached to gears that move along a track below the running lanes, poles or string like attachments that would give the appearance of the “runner” moving, a sensor or alternate mechanism that would activate when the winner crosses the finish line, as well as a cam mechanism that would give the fans/crowd movement. The gears attached to the “runner” would be on a single track or set of tracks depending on how the movement of “runners” are operated (lever or button). The strings or pole mechanism are inspired by automata toys and would be attached to attached to either the “runners” arms or legs to give the “runners” more movement besides the main function of moving along the racing lanes with the gears. The sensor or alternate mechanism (i.e. Clip) would be activated or released once the first “runner” crosses the finish line which would then release victory flags at the end of the lane or potentially trigger a movement pattern of the mechanism controlling the crowd. The cam mechanism would consist of different cam and follower types for different sections of the crowd so that the movement patterns would be varied as well as give the crowd a wave effect as the “runners” pass by in the case that cam and follower are attached to the gears and track and not on their own rotation.

Ultimately, “Runner’s Rivalry” was not selected due to how complicated the running mechanisms were, as well as how it relied on sensors that would further complicate the design. Another reason why “Thrill on the Hill Mini” was selected instead of the alternative was because it is related to Purdue and is therefore a better fit for a project completed at Purdue.

Another concept that was generated was a single-player basketball game, where the user would pull a “basketball” (marble) down on a spring-loaded launcher and try to score. Like the “Runner’s Rivalry” this toy would also rely on sensors, as a crowd on the sidelines would start cheering and moving up and down whenever the basketball made it into the basket. The basketball would have likely been sensed with either a light or pressure sensor, causing a motor to spin. This motor would be attached to a rod supporting cams that would move the audience up and down. Similarly to “Runner’s Rivalry”, this idea was not selected due to how the sensors would further complicate the design. Additionally, it was determined that this was not a very creative idea as a similar toy exists in the past toy archives.

3. Detail Design

3.1 Sub-system 1: Rotators

The main sub-system in this product is the rotation feature. This was designed with the principle of a four-link system. The constraints were that one link would have complete rotations (ie. connected to a rotating gear), and another link would complete a total of 60 degrees of rotation every half cycle, and then return back 60 degrees, centered about a vertical line. In order to achieve this, the center of the driving circle was placed on a horizontal line, coincident with the midline between horizontal lines placed at the apex and low points of the 60-degree arc swept by the second link. This was done to ensure that the figurines completed symmetric 60-degree sweeps, which had the side effect of making the backwards and forwards stroke have different angular velocities. The difference in angular velocities was not very noticeable however and was only noticeable on the kinematic analysis. These basic constraints were used to define a motion skeleton, and other parameters were adjusted to create the rotating sub-system. These individual rotators were connected across rows by links at the pin. These rows were connected by a gear train of 1:1 driving and idler gears. It was chosen to have gears with a pitch diameter of ¾" for all gears in this train. The idler locations were placed on the perpendicular bisectors of the line between consecutive driving gears, tangent to the pitch circle of the driving gears.

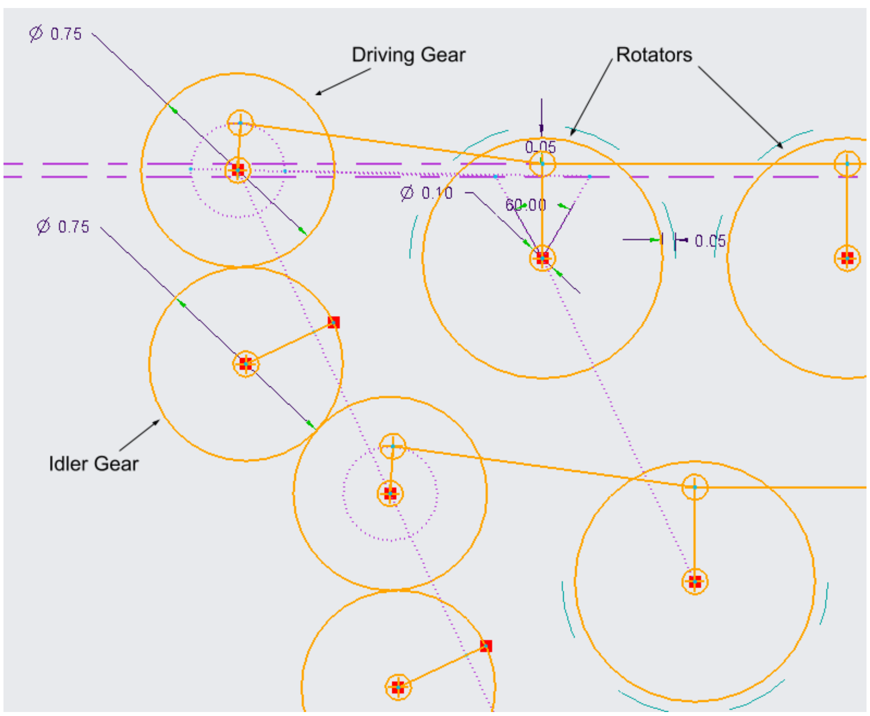


Figure 3.1.1: Motion skeleton of the rotator sub-system

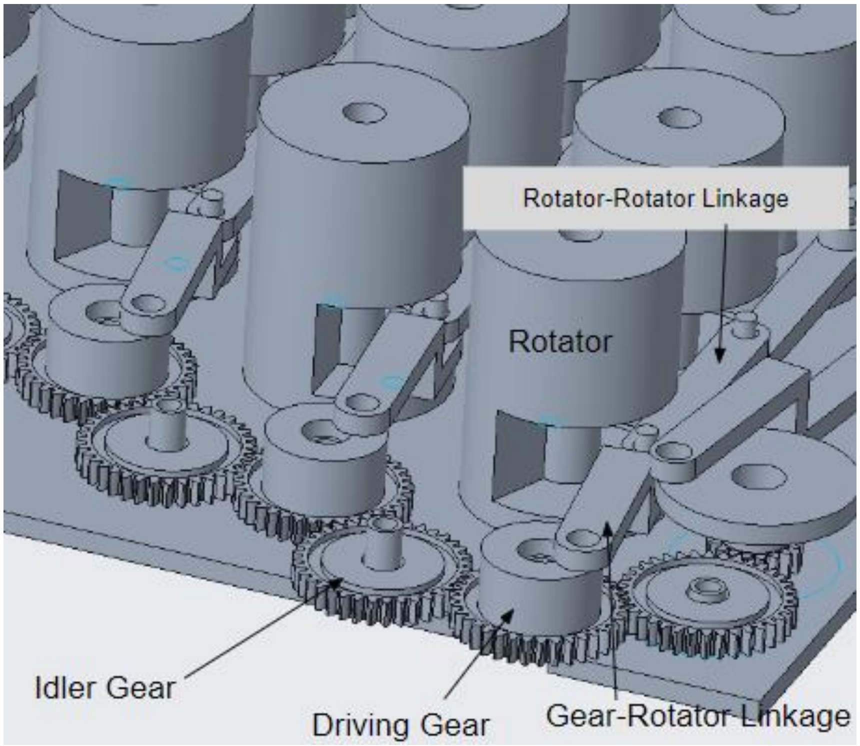


Figure 3.1.2: CAD Model of the implementation of the motion skeleton for the rotator sub-system

3.2 Sub-system 2: Drum Major Conducting

The second mechanical subsystem was the conducting motion of the drum major figurine. This was achieved through a track beneath the stage, which connected to the drum major's arm. The track was designed to have a ¼" lift to the drum major's arms and complete cycles twice as fast as the rotators to be consistent with the real relative speeds of the marching band members and drum major conducting in the real world. The track moves within a linear slot, to turn the circular motion input into the linear motion required.

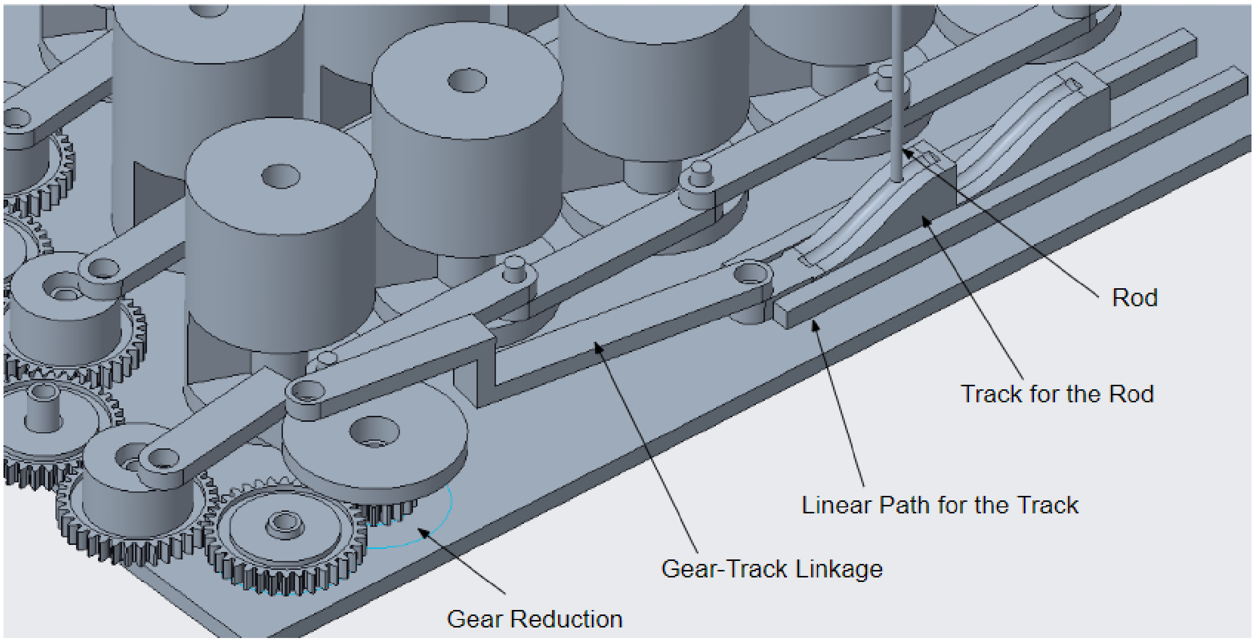


Figure 3.2.1: Annotated Sub-system 2

3.3 Sub-system 3: Electronic Systems

An Esp32s Microcontroller was used to control the Motor Driver, DC Motors, and audio amplifier. The Esps32 Microcontroller is a single board computer with a wireless microcontroller chip that can be integrated with Arduino and has Wi-Fi and Bluetooth capabilities. The L298N Dual H Bridge Motor Driver was used to drive the DC Motor, referred to as motor A. The enable pin (Pin A) located on the L298N was used to control the speed of the motors using pulse-width modulation, which generates variable voltage for the motors. The logic pins control the direction of the motor, allowing it to go forward or reverse. For this design, the motor simply needed to rotate clockwise at a constant speed to drive the mechanisms inside of the toy. Various speeds were tested to sync the marching band's movement to the beat of "Hail Purdue", but the motor was eventually set to maximum speed. Maximum speed was used because the motor could not provide the amount of torque required to spin all the rotating pieces and the slider.

Along with the motor and motor driver, another component used in the electronic subsystem was the DY-SV5W Audio Amplifier. This component, used alongside 3.5 mm audio jack, or a speaker, was selected with the intention to play "Hail Purdue" along with the movement. Unfortunately, there was a wiring issue that was not diagnosed in time and the speaker did not end up working successfully. However, the code written, which can be found in Appendix E, was successful. The serial monitor indicated that approximately every 1.5 minutes, Track 1 (Hail

Purdue) would start playing. A circuit diagram for the motors and audio amplifier can be seen in Appendix D.

3.4 System Integration

A top-down design was used to design the assembly of the marching band members and drum majors. Figure 3.1.1 shows the motion skeleton used to drive the gear-linkage system used to produce a back and forth rotation of each band member on their mini cylindrical stage. After designing the motion skeleton, which contains gears and linkages, the individual parts were designed using the skeleton model. The drum major control and rotation subsystems were linked through an idler gear and a 2:1 gear reduction as shown in Figure 3.2.1

The figurines shown below in Figure 3.4.1 were added to the system using the bottom-up assembly after the stage and mechanism system was created.

Figure 3.4.1: CAD model of a trumpet figurine (left) and the prototype result (right)

4. Prototyping and Testing

Both laser printing on wood and 3D printing with PLA were used as prototyping methods in our design. Laser cutting was used to design the stage wall shown as a drawing in Appendix A. Initially, the stage and underlying mechanism parts were 3D printed, then the figurines were designed to fit in the hole of the stage. To properly design the figurines with concepts of scale and fit, the figurines were chosen to have a total length of less than 2 inches. Moreover, since the size of the hole where the figurines stand is about ½ inches, this was taken into account to design a slightly smaller part where the figurine could stand.

Little to no problems were faced during the build and test stage. The main problem we faced was with the small size of the figurines and the musical instruments that made it slightly difficult to remove the supports from after 3D printing. This problem was solved by utilizing small tools such as screwdrivers to carefully remove the supports from the figurines. Some supports were left on the figurines and not removed as it was determined by the team that this issue does not largely impact the creativity of the design.

The bill of materials (BOM) shown in Appendix B shows the big number of parts the team used to develop the model. The BOM mainly consists of gears and roof/stage, figurines and musical instruments. In terms of budget, the team utilized the motor from the midterm project, and individually purchased the SD card, the audio amplifier and the speaker for less than $35.

5. Results and discussion

The physical prototype was designed and built as expected in a real marching band stage performance. The mechanical aspect of the project worked as expected, where the gears and linkages correctly and accurately rotated the figurines standing on the stage in 60 degrees and –60 degrees motion. The motor worked perfectly in automating the rotating motion in the toy by turning on the switch. However, there were issues with the aspect of playing the “Hail Purdue” song with the speakers after trials to automate the process with the code.

If the team had an extra prototyping cycle, multiple changes could be made to the mechanical and electrical designs of the toy. One thing is that the figurines could be made so that their hands are stuck to the body to lower the amount of supports printed. 3D printing would also be tried with the double extruder printer instead of the Einstart/Pulse printers. This is because the double extruder printer could print the supports with a dissolvable material.

Another improvement that the team could make to the design is to improve its aesthetics by painting the remaining figurines and perhaps adding a logo to the toy. All in all, more time would be given to the testing of the prototype to ensure a better outcome.

From this toy design experience, there are a lot of learning outcomes, such as paying attention to detail while designing the figurines and ensuring that the motion skeleton of the mechanism functions as expected. We learned more about 3D printing, and how supports need to be taken into account during the design part of the process.

6. Conclusion

The prototype was overall well-designed in terms of the measurements, and the way it is supposed to function. This is clearly seen in the figure of the whole assembly in the beginning of the document. There were some problems with the removal of the extra supports from the figurines due to their small size and due to the complexity of the musical instruments. Some figurines had broken hands, which was not a big issue as the figurines could stand and rotate perfectly on the stage.

The electrical sub-system, which consisted of a motor, an audio amplifier circuit and a speaker functioned partially. The motor was able to rotate the gear system that eventually guided the back and forth movement of the linkages connected to each figurine. The audio amplifier system produced a signal that the circuit electrical input is transferred from the input to the output. However, there were difficulties with getting the speaker to play “Hail Purdue” song.

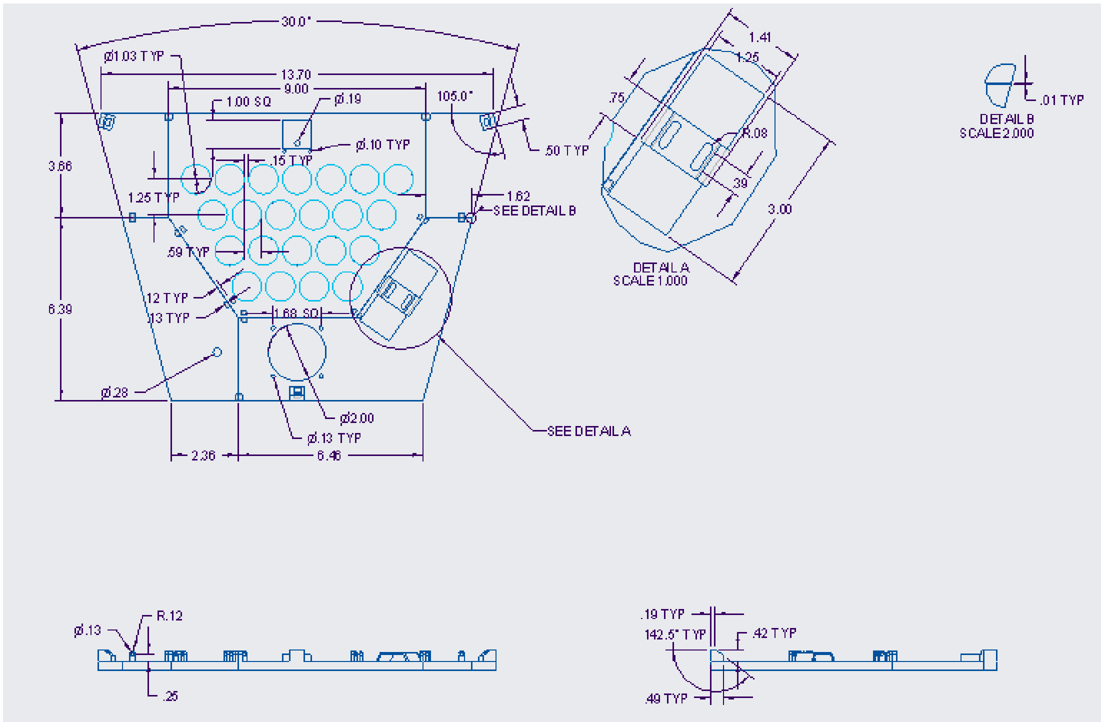
For future work, the team aims to resolve those minor issues of the figurines and the electrical circuit. The figurine hands could be made so that it is attached to the body so that extra support is not required. The double extruder 3D printer could be used instead of the Pulse/Einstart printers to make the removal of the supports easier, especially for the musical instruments that are too detailed and were difficult to print without excess support. The issue with the speaker system could be eliminated with more trials and more research.

7. References

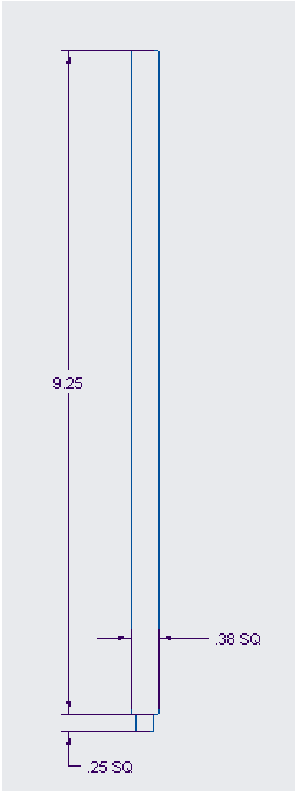
N/A

Appendices

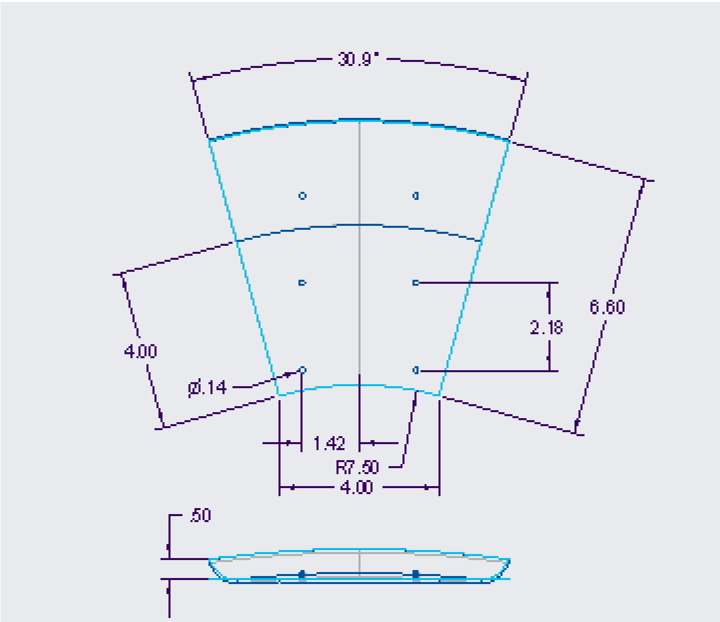
Appendix A: Part Drawings



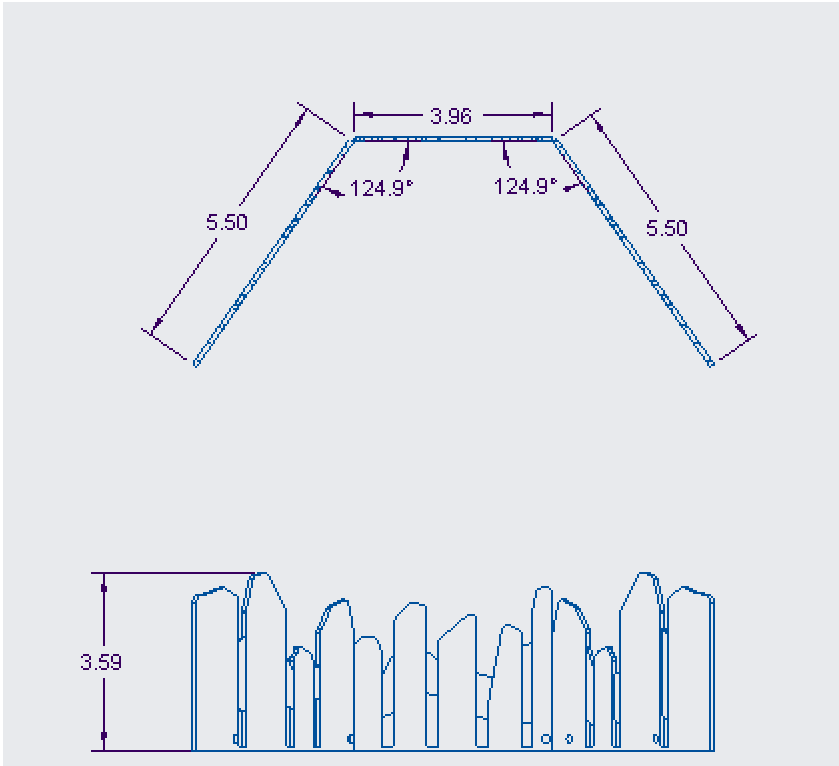
Stage Floor



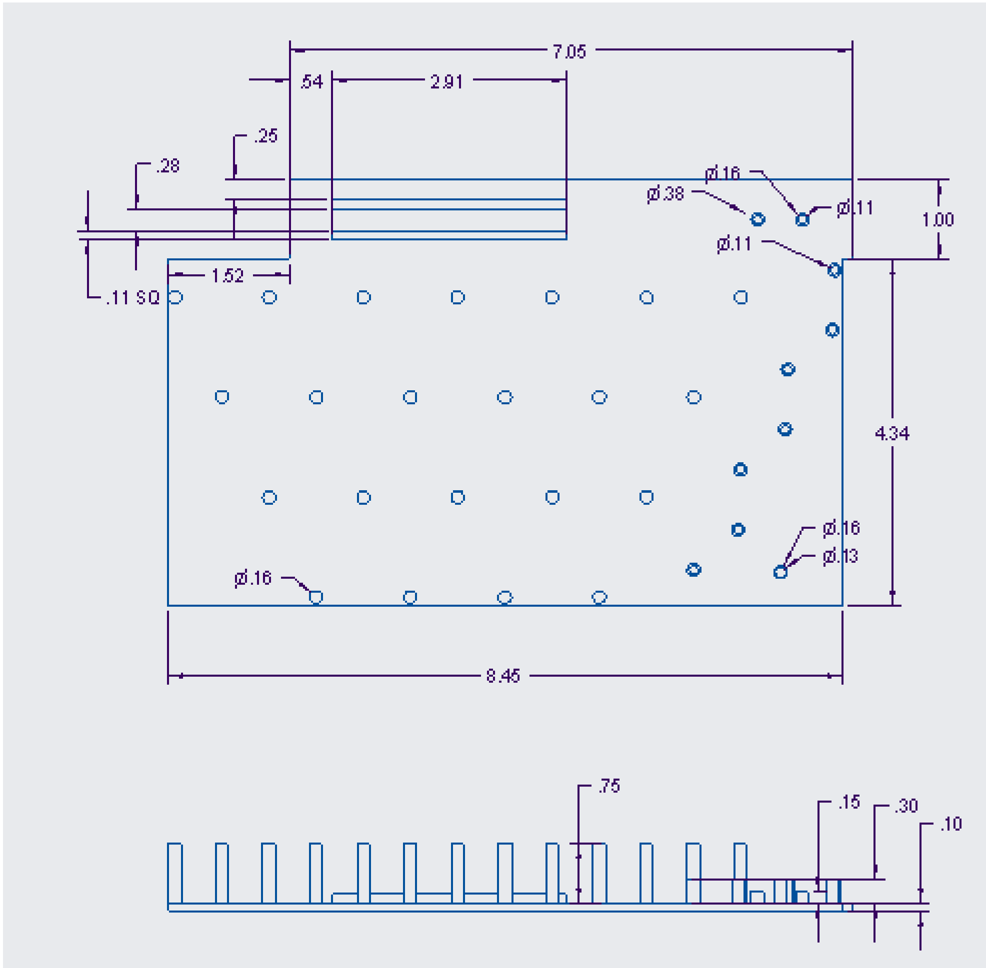
Roof Suspension Leg



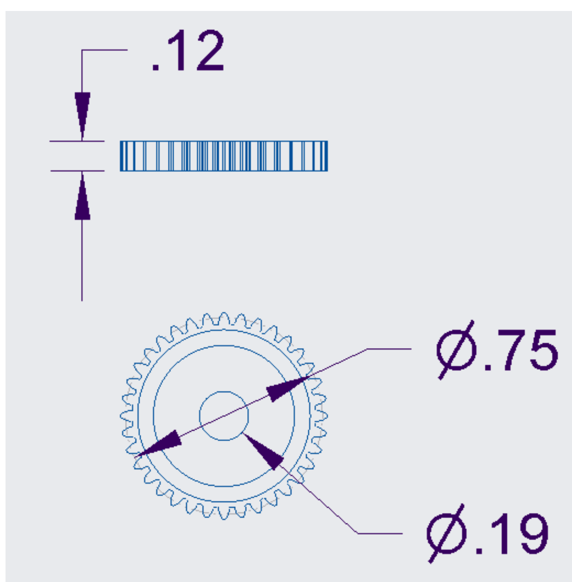
Suspended Roof



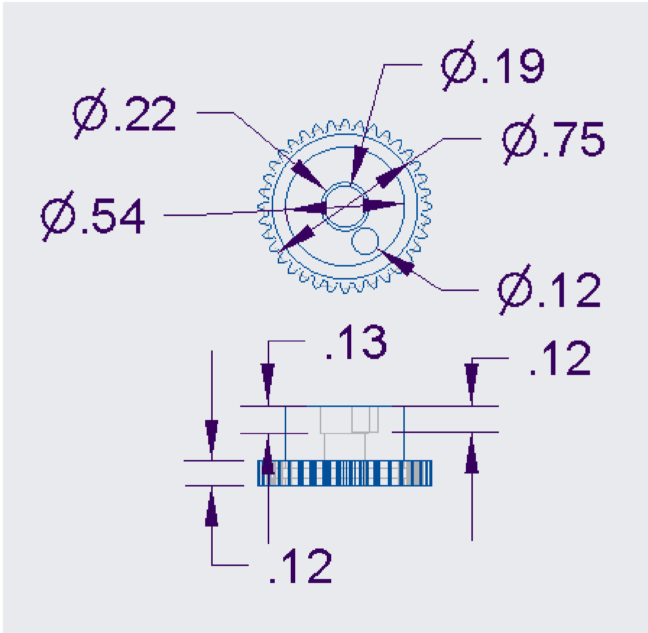
Stage Wall



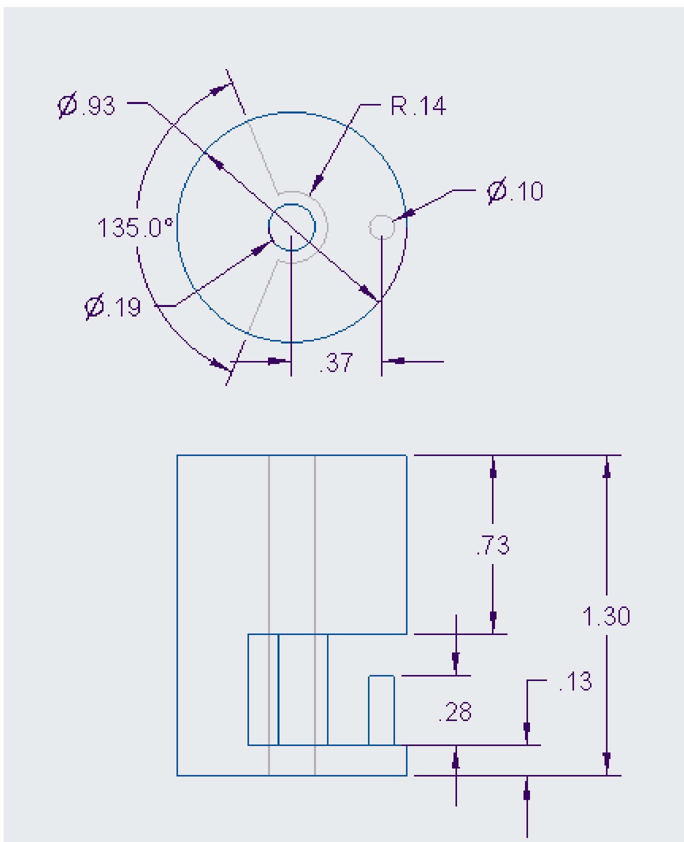
Mechanism Base



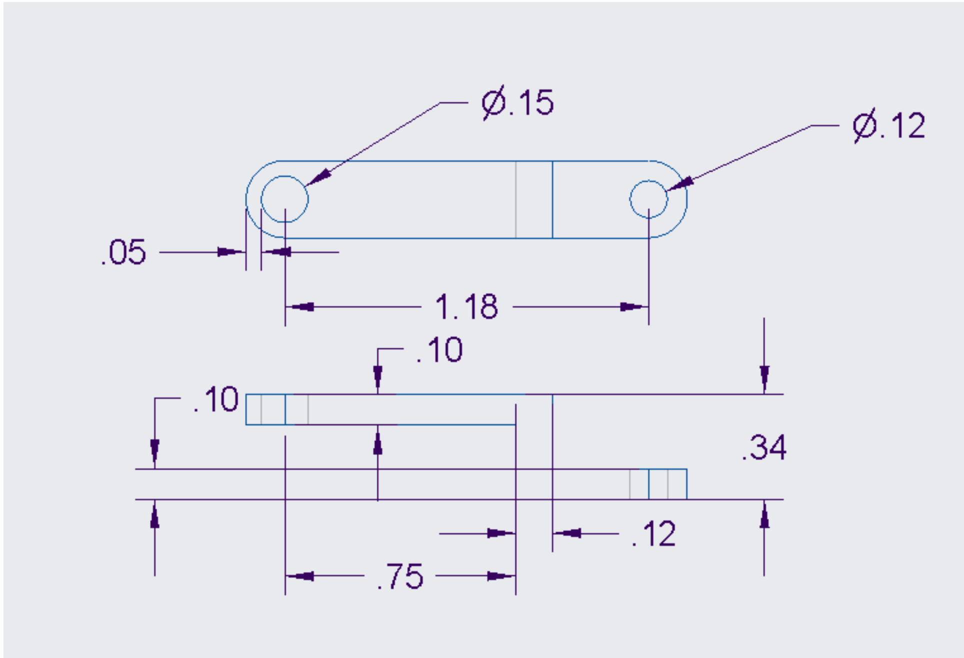
Idler Gear



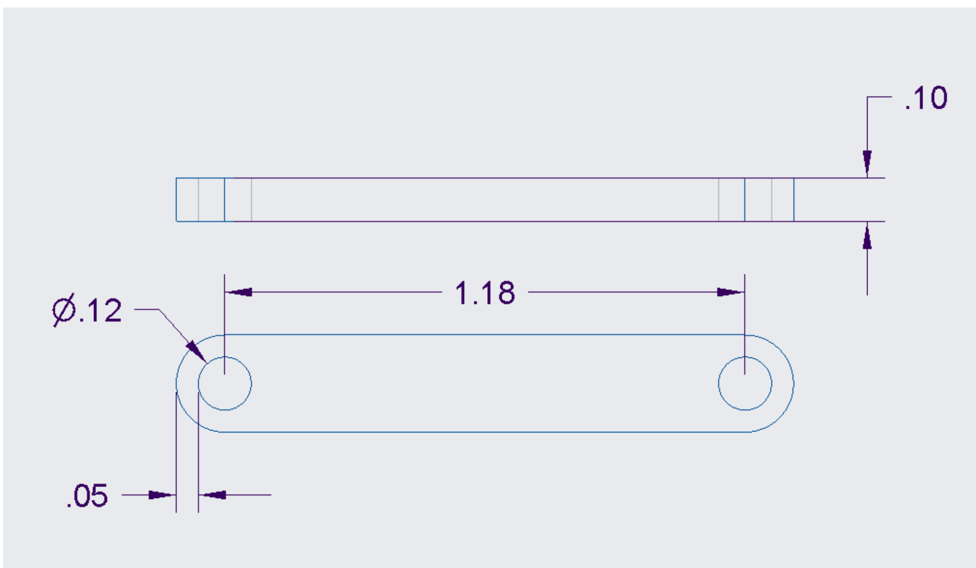
Driving Gear



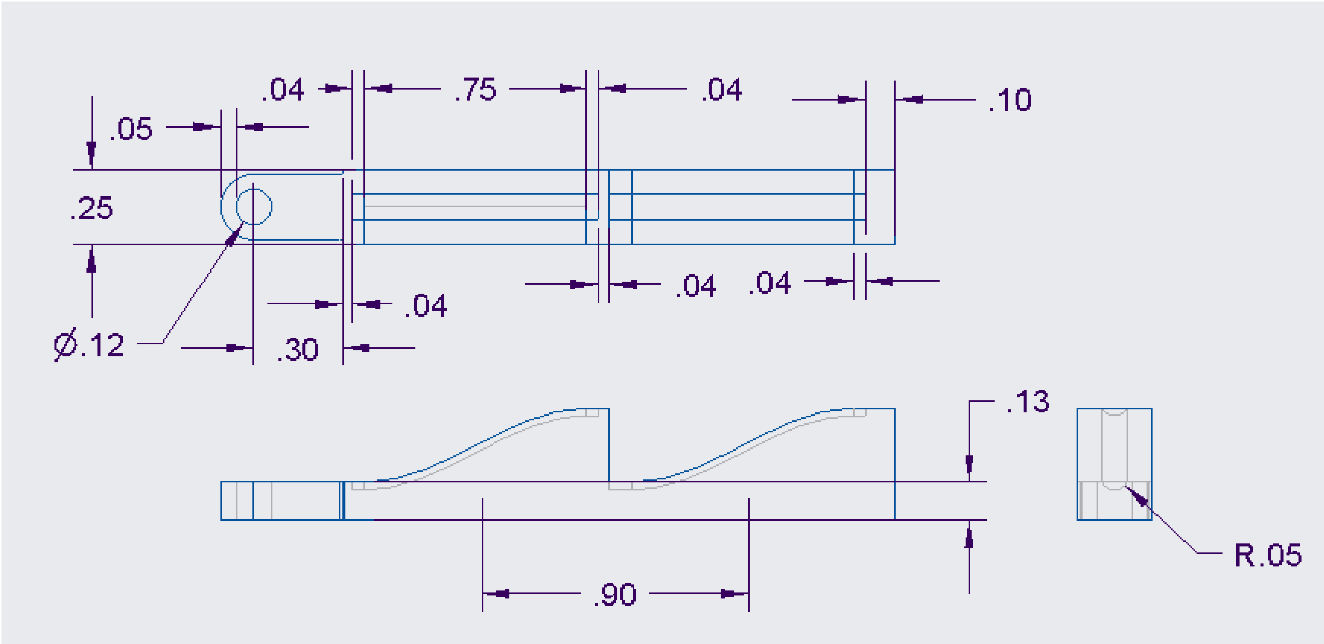
Rotator



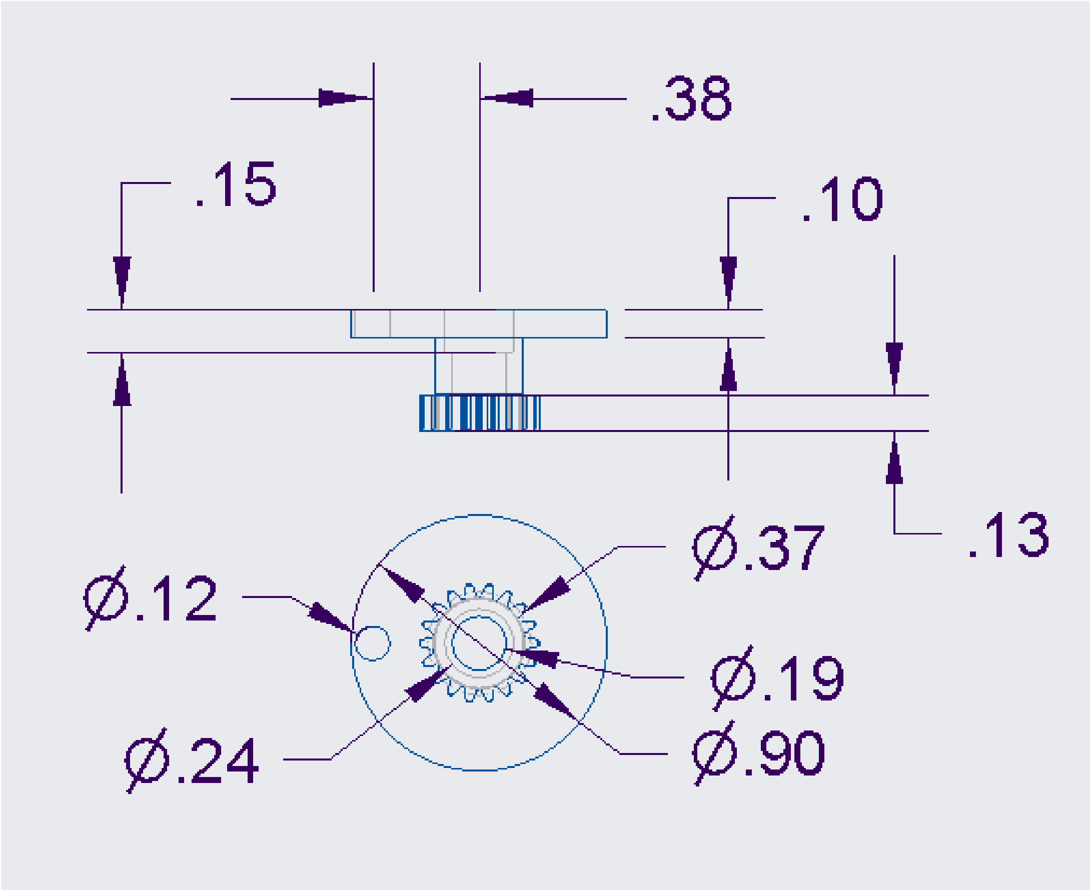
Driving gear to rotator linkage



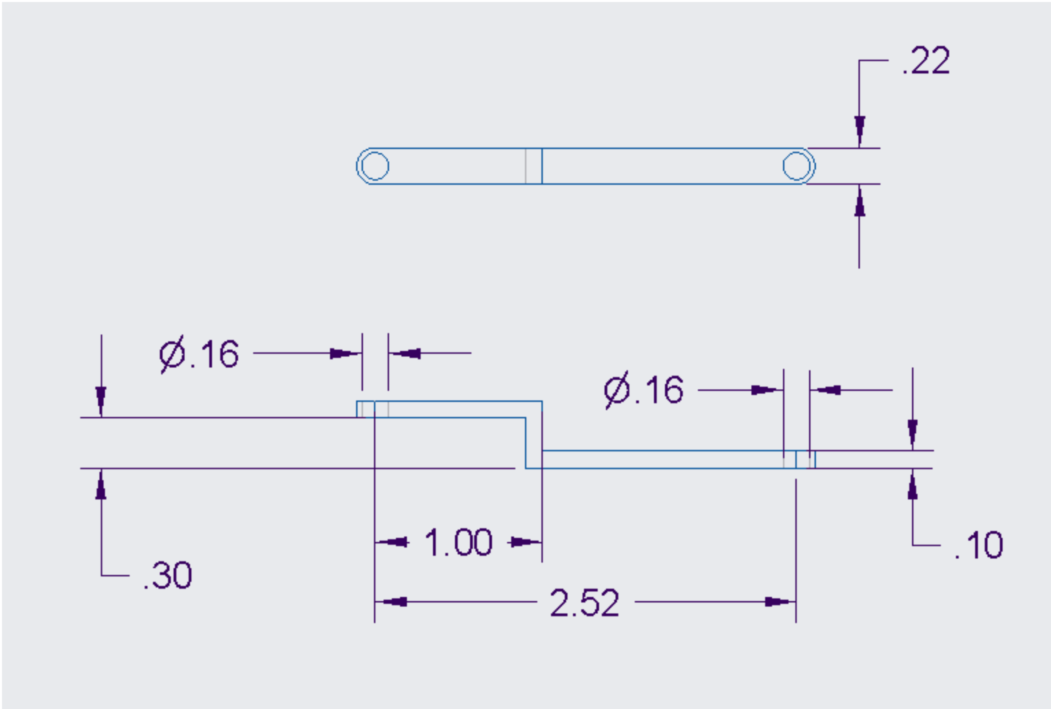
Rotator to rotator linkage



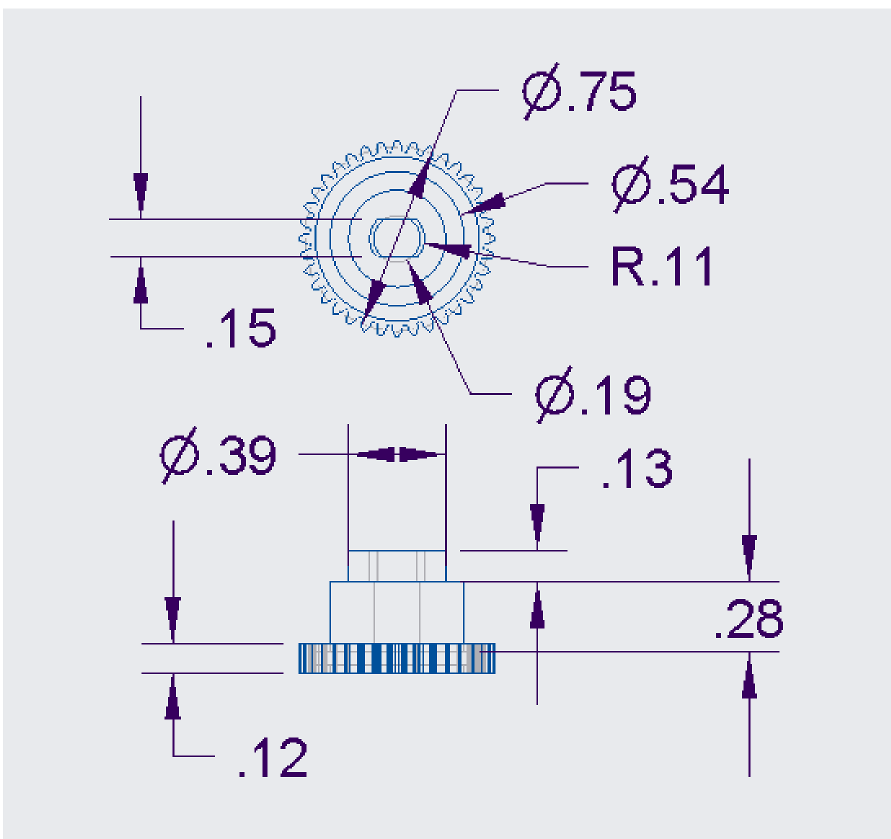
Drum major rod track



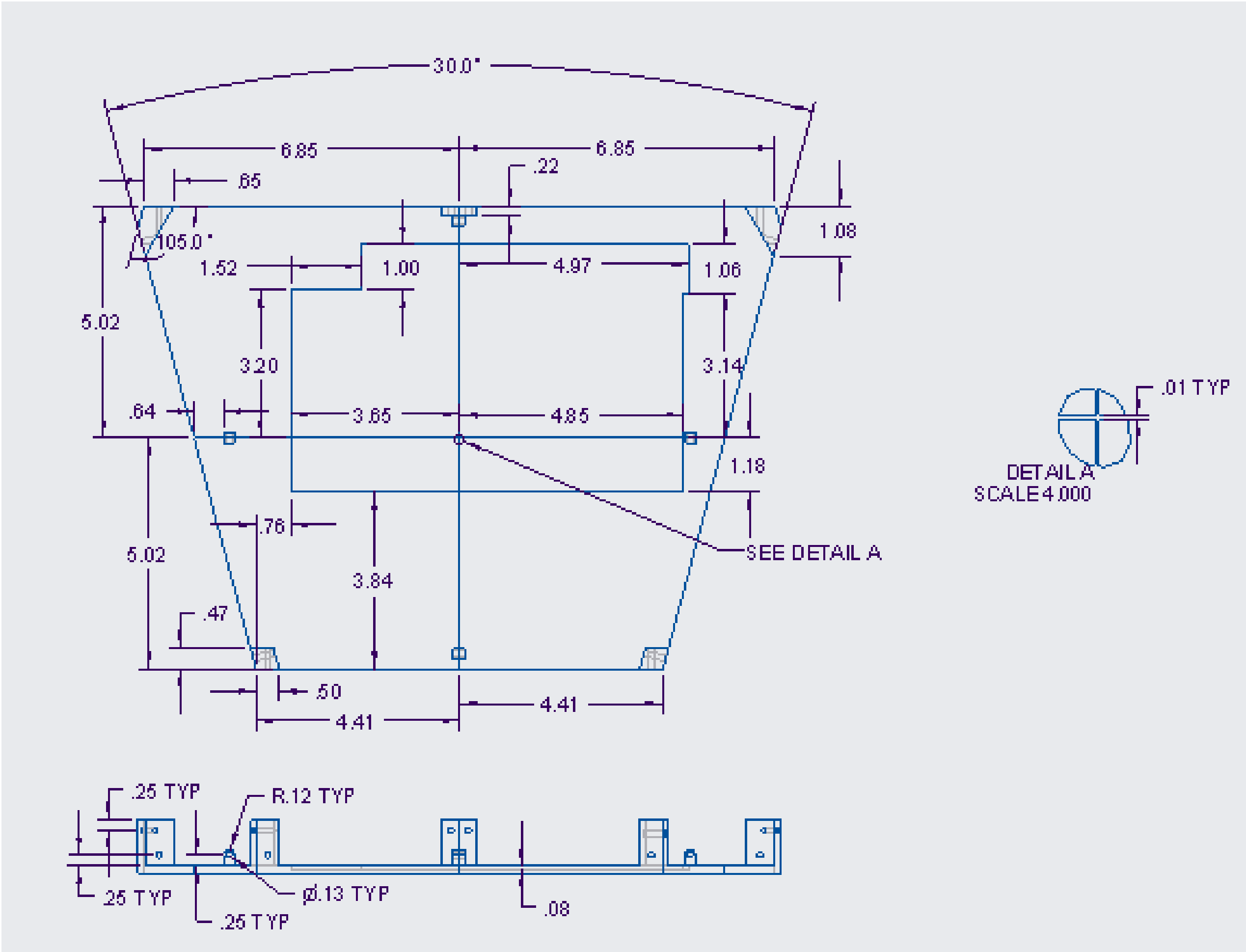
Drum major control gear



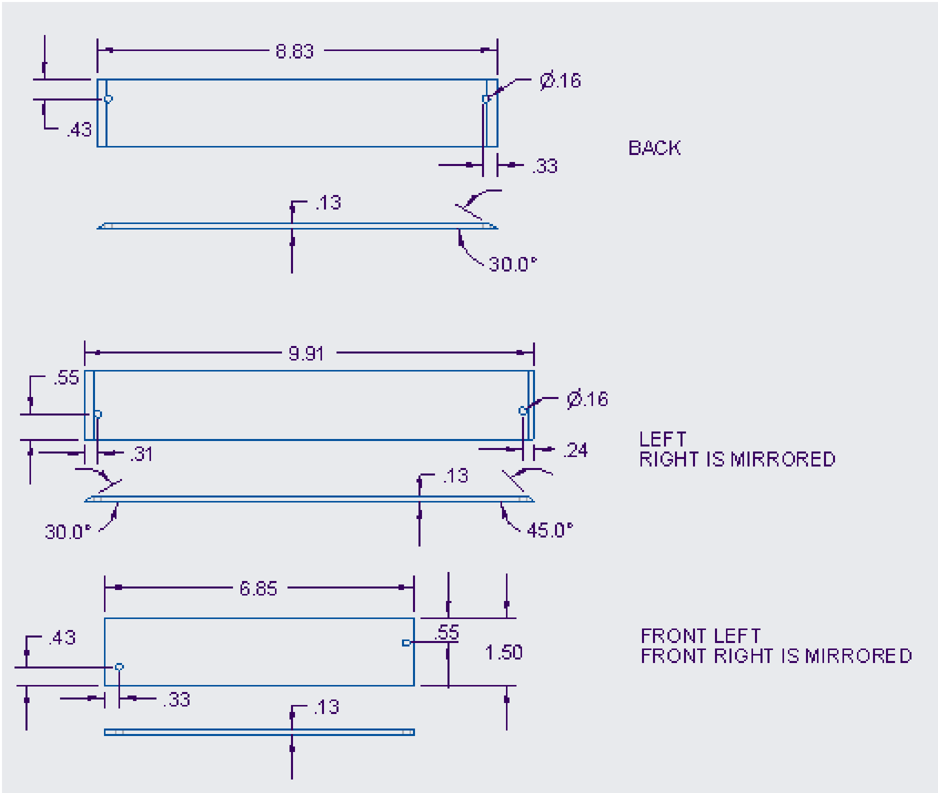
Drum major control gear to track linkage



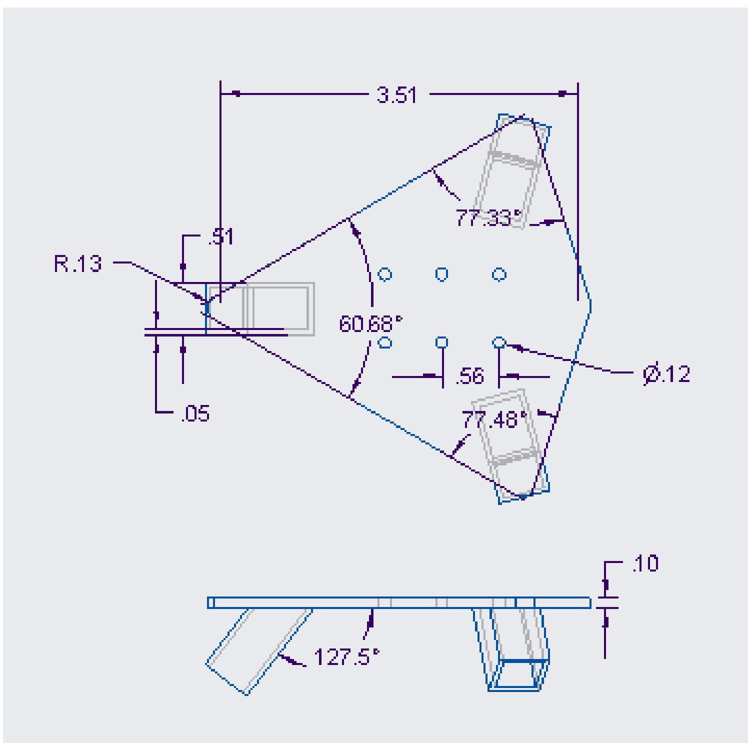
Motor driven gear



Bottom Piece



Mechanism Walls



Tripod Leg Connector

Appendix B: Bill of Materials

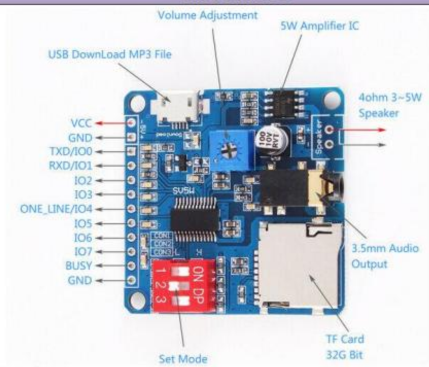
QTY.	Description	File Name
1	Stage Floor	base.prt
3	Roof Suspension Legs	tripod_leg.prt
1	Suspended Roof	slayter_roof.prt
1	Stage Wall	wall.prt
1	Mechanism Base	figure_base.prt
4	Idler Gears	gear_generic.prt
4	Driving Gears	gear_rowlink.prt
22	Rotators	figurine_rotator.prt
4	Driving gear to rotator linkage	first_link.prt
18	Rotator to rotator linkage	rowlink.prt
1	Drum major rod track	drum_major_track_2.prt
1	Drum major control gear	dm_control_gear.prt
1	Drum major control gear to track linkage	drum_major_link.prt
1	Motor driven gear	motor_drive_gear.prt
1	Motor	drive_motor.prt
1	Bottom piece	bottom.prt
1	Mechanism wall, back	bottomwall_back.prt
1	Mechanism wall, left	bottomwall_left.prt
1	Mechanism wall, right	bottomwall_right.prt
1	Mechanism wall, front right	bottomwall_frontright.prt
1	Mechanism wall, front left	bottomwall_frontleft.prt
1	Tripod Leg Connector	top.prt
14	Trumpet Figurines	trumpet_figurine.asm
8	Sousaphone Figurines	toobah_figurine.asm
1	Drum Major Figurine	drum_major_assembly.asm

2	Drum Major Control Rods	rod.prt
6	~ 3” Suspension Wires	N/A (see datum curves in main assembly file)
13	#6-32 x 3/8” machine screws	N/A
21	#6-32 x 5/8” machine screws	N/A

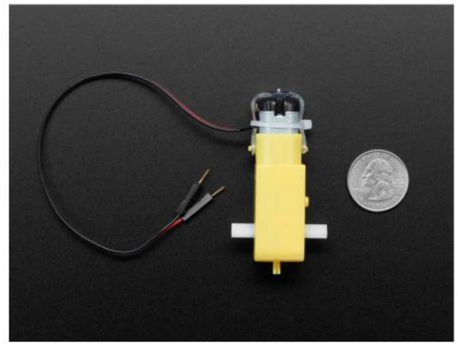
Appendix C: Manufacturer Datasheets for COTS parts

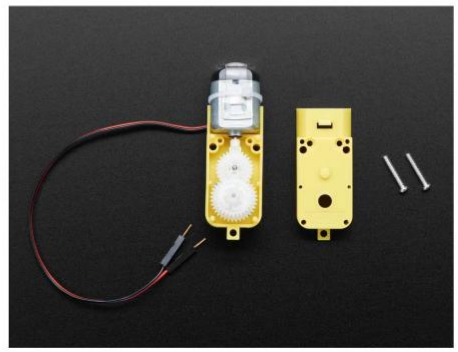
DY-SV5W Audio Amplifier Module & Geared DC motor

Pin Definition



No.	Pin Name	Instruction
1	5V+	Work Voltage Positive Pole
2	5V-	Work Voltage Negative Pole
3	TXD/IO0	IO trigger mode is input IO0;UART mode is TX.
4	RXD/IO1	IO trigger mode is input IO1;UART mode is RX.
5	IO2	IO trigger mode input IO2.
6	IO3	IO trigger mode input IO3.
7	IO4/ONE_LINE	IO mode input IO4;One_Line mode data receiver pin.
8	IO5	IO trigger mode input IO5.
9	IO6	IO trigger mode input IO6.
10	IO7	IO trigger mode input IO7.
11	BUSY	Output low level signal(0V) when playing and output high(3.3V) after playing.
12	GND	Ground



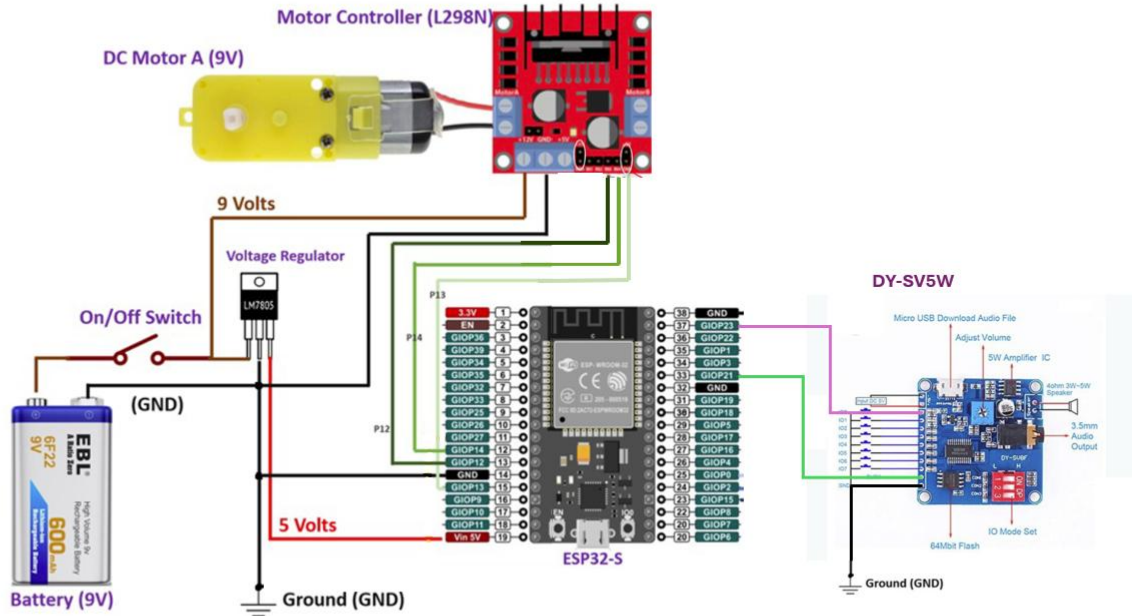


TECHNICAL DETAILS

- Rated Voltage: 3~6V
- Continuous No-Load Current: 150mA +/- 10%
- Min. Operating Speed (3V): 90+/- 10% RPM
- Min. Operating Speed (6V): 200+/- 10% RPM
- Torque: 0.15Nm ~0.60Nm
- Stall Torque (6V): 0.8kg.cm
- Gear Ratio: 1:48
- Body Dimensions: 70 x 22 x 18mm
- Wires Length: 200mm & 28 AWG
- Weight: 30.6g

Product Weight: 30.6g / 1.1oz

Appendix D: Circuit Diagram



Appendix E: Arduino Code

```
#define RXD2 22
#define TXD2 23
//variables always required to work DY-SV5W
const int busyPin = 21;//change to any suitable digital pin
byte commandLength;
byte command[6];
int16_t checkSum = 0;

#include <ESP32Servo.h>

// Motor setup: constants won't change
int servoPin = 27;     //specify pin numbers according to the circuit diagram
int enablePinA = 15;
int in1PinA = 2;
int in2PinA = 0;
int enablePinB = 13;
int in1PinB = 12;
int in2PinB = 14;

//my variables for demo sketch
int trackNum = 1;
int playStatus; //0 playing 1 stopped 2 waiting to start next track
unsigned long lastCheckTime;
unsigned long currentMillis;
int busyPinstate;

void playTrack(int soundTrack) {
  //select track
  Serial.print("soundTrack: ");
  Serial.println(soundTrack);
  command[0] = 0xAA;//first byte says it's a command
  command[1] = 0x07;
  command[2] = 0x02;
  command[3] = (soundTrack>>8) & 0xFF;//snh...track HIGH bit
  command[4] = soundTrack & 0xFF;//SNL... track low bit
  checkSum = 0;
  for (int q = 0; q < 5; q++) {
    checkSum +=  command[q];
    //Serial.print(command[q]);
    //Serial.print(",");
  }
  //Serial.println(" ");
  //Serial.println(checkSum,BIN);
```

```
  //Serial.println(checkSum);
  //command[5] = lowByte(checkSum);//SM check bit... low bit of the sum of all previous values
  command[5] = (checkSum & 0xFF);//SM check bit... low bit of the sum of all previous values
 // Serial.println(command[5],HEX);
 // Serial.println(command[5],BIN);
 // Serial.println(command[5]);
  commandLength = 6;
  sendCommand();
}

//plays whatever track has been paused or 1st track if nothing selected
//May need to be selected after putting into random mode
void play() {
  command[0] = 0xAA;//first byte says it's a command
  command[1] = 0x02;
  command[2] = 0x00;
  command[3] = 0xAC;
  commandLength = 4;
  sendCommand();
}
//selects random mode
void randomMode() {
  command[0] = 0xAA;//first byte says it's a command
  command[1] = 0x18;
  command[2] = 0x01;
  command[3] = 0x03;//random
  checkSum = 0;
  for (int q = 0; q < 4; q++) {
    checkSum +=  command[q];
  }
  command[4] = (checkSum & 0xFF);//SM check bit... low bit of the sum of all previous values
  commandLength = 5;
  sendCommand();

  //play() needs to be selected if you want the random tracks to start playing instantly
  play();
}

//sets the device volume...0 - 30
void playbackVolume(int vol) {
  if (vol > 30) { //check within limits
    vol = 30;
  }
  command[0] = 0xAA;//first byte says it's a command
  command[1] = 0x13;
```

```
  command[2] = 0x01;
  command[3] = vol;//volume
  checkSum = 0;
  for (int q = 0; q < 4; q++) {
    checkSum +=  command[q];
  }
  command[4] = (checkSum & 0xFF);//SM check bit... low bit of the sum of all previous values
  commandLength = 5;
  sendCommand();
}

//sends the command
void sendCommand() {
  int q;
  for (q = 0; q < commandLength; q++) {
    Serial2.write(command[q]);
    Serial.print(" ");
    Serial.print(command[q], HEX);
  }
  Serial.println(" End");
}

void setup() {
  Serial.begin(115200);
  Serial2.begin(9600, SERIAL_8N1, RXD2, TXD2);
  Serial.println("ESP32DYSV5Wv7");

  pinMode(busyPin, INPUT);//pin to read from DY-SV5W buyPin

  playbackVolume(17);//sets volume to lvl 17
  //Motor setup:
  Serial.begin(9600);         //starts serial communication, so that the Arduino can send out commands through
the USB connection. The value 9600 is called the 'baud rate' of the connection. This is how fast the data is to
be sent.
  // servo1.attach(servoPin);    //set up the servo pin as pin 27
  // servo1.write(0);            //initialize the servo motor at 0 degree.
  pinMode(in1PinA, OUTPUT);    //set in1PinA(2) as an output pin
  pinMode(in2PinA, OUTPUT);    //set in2PinA(0) as an output pin
  pinMode(enablePinA, OUTPUT); //set enablePinA(15) as an output pin
  pinMode(in1PinB, OUTPUT);    //set in1PinB(12) as an output pin
  pinMode(in2PinB, OUTPUT);    //set in2PinB(14) as an output pin
  pinMode(enablePinB, OUTPUT); //set enablePinB(13) as an output pin
  }
void loop() {
  //Motor:
```

```
 Serial.println("go forward");
 int speed = 200;
 analogWrite(enablePinA, speed);
// digitalWrite(enablePinA, HIGH);
 digitalWrite(in1PinA, HIGH);
 digitalWrite(in2PinA, LOW);

 //Speaker:
 currentMillis = millis();//stores time in millis() for non blocking timings
 busyPinstate = digitalRead(busyPin);
 if (busyPinstate > 0) { //nothing playing


   if (playStatus < 1) {
     Serial.print(busyPinstate);
     playStatus = 1;
     lastCheckTime = currentMillis;//set system ready for 2 sec delay
   }
 }
 switch (playStatus) {
   Serial.print(" P: ");
 Serial.println(playStatus);
   case 1:
     if (currentMillis - lastCheckTime >= 2000) { //2000 = 2 seconds of silence between tracks
       playStatus = 2;
       //Just selecting a new track
       //this could have been something like
       //trackNum = random(1,5);
       trackNum = 1;//fixed track number
       Serial.print("trackNum: ");
       Serial.println(trackNum);
       playTrack(trackNum);
       // trackNum++;
       // if(trackNum > 4){
       //   trackNum = 1;
      }
     break;
   case 2:
     if(busyPinstate < 1){
       playStatus = 0;
       Serial.print(" P: ");
       Serial.println(playStatus);
     }
     break;
   default:
```

```
      break;
  }
}
```