Team Members: Prithika Ganesh, Christopher Lambert, Sunni Lin, Sarah Stantial, Susan Zhang

Task Machine Competition Project Details

Sleepy Rhett

Introduction:

A task machine is a simple machine that undoes the action it performs. Our task machine will act as an "alarm" — a 5-second countdown until school starts — that will run whenever activated by a user.

The setting is a cozy bed with a sleeping "student", Rhett, with an alarm clock located on a bedside table. When the machine is turned on by a switch, the alarm screen will display itself as "Rhett's Alarm for School." Then, the user will activate the alarm by covering the light sensor that acts as the "off button." This causes the alarm to display a five second countdown after which an annoying alarm sounds and a red LED lights, rousing a disgruntled Rhett from his sleep. Drowsy, he uses his tail to cover the snooze button to deactivate the alarm — sometimes even growling or barking in annoyance. Once the alarm is turned off by Rhett it will display "Have Fun at School" and will be ready to be set off again by a human.

Therefore, the task that this machine undoes is deactivating an alarm clock every time it is triggered.

Methodology:

The phototransistor will sense the light level passing through and alter the current flowing to the Arduino, which we will code to tell the alarm when to sound and when to stop. When the alarm goes off, a motor will be used to move the tail from rest to over the phototransistor and then back to rest again — with a fun addition of occasional growling or barking from Rhett emitted from a speaker each time he is woken by the alarm. The bed, bedside table, and alarm clock will be machined from wood and Rhett will be 3D printed; everything will also be hollow to allow for the wires to pass through the different parts. For finishing, the wood and Rhett will be painted, the bed will be covered in fabric, and the tail will be covered in cotton balls for texture and opaqueness.

Main Processes of Machine:

Mechanical Process:

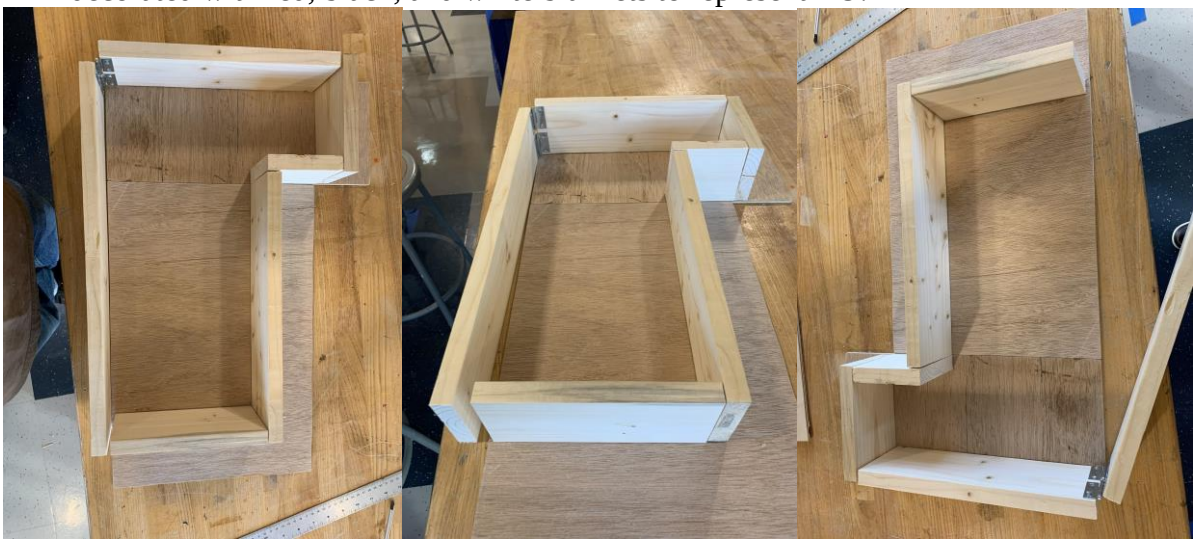
This part of the project focuses on the physical, moving components of the machine. We started by building a basic frame for the bed and bedside table. The pictures below depict this before it was permanently glued. The longest side of the bed has a hinge to allow access to the cavity beneath the bed in which the electronic components are stored. A base of thin wood was added for the bottom, and the same wood was used for the top of the bed. The bed is held together by wood glue.

The alarm clock was constructed out of thin wood and consists of five sides, with the bottom being open for electrical wires to enter. Holes were cut and drilled into faces for electronic components to come out of. The side facing the bed holds the LCD, the top holds the LED and buzzer, and one side holds the switch while the other holds the phototransistor.

Rhett was 3D printed in parts from the CAD assembly included. The head was printed in two parts then glued together, as was the body. The rear of the body includes a slot allowing for the tail to rotate off of the motor.

The motor causes the tail to wag back and forth. There is a wooden component for the tail that consists of a dowel inserted in a small wooden block. The block has a hole in the bottom that fits over the head of the motor snugly.

For detail, a head and footboard were added to the ends of the bed, and it was decorated with red, black, and white blankets to represent BU.



Electrical Process:

The electrical components are all located under the bed, inside the hollow body of Rhett, or in the alarm clock. There is an arduino programmed with the code that activates the alarm clock and a breadboard that allows for more connections and components. These are attached to the other components by jumper wires, allowing them to extend to the alarm. The phototransistor effectively senses light and triggers the alarm to either start or turn off, prompts the motor to turn in either the clockwise or counterclockwise direction, and activates the other associated electronic functions.

Results:

The final product is successful in achieving the goal of a task machine. It snoozes an alarm that is activated by an outside user, essentially undoing the action of triggering the alarm. The project is also successful in other respects, such as the appearance, creativity, and personality of the machine. Rhett looks to be sleeping peacefully on a comfortable bed, decorated with pillows and cozy blankets. The task machine is an original design and idea, and does not simply copy an existing product. Additionally, it incorporates school spirit in its inclusion of Rhett, the BU logo, and BU colors. Lastly, it contains personality when it performs the tasks which is shown by Rhett's barking and growling when the alarm is triggered at various times. The alarm display itself incorporates a unique feature, printing different messages depending on if it's before, during, or after the alarm has gone off.

Appendix:

References

Arduino - DC Motor, TutorialsPoint, 2021, https://www.tutorialspoint.com/arduino/arduino_dc_motor.htm.

"Arduino - Light Sensor: Arduino Tutorial." *Arduino Getting Started*, https://arduinogetstarted.com/tutorials/arduino-light-sensor.

"DFPlayer_Mini_SKU_DFR0299." *DFRobot*, https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299#target_6.

"Interface Buzzer with Arduino Uno." *Arduino Project Hub*, https://create.arduino.cc/projecthub/akshayjoseph666/interface-buzzer-with-arduino-uno-694059?ref=user&ref_id=600499&offset=3.

Team, The Arduino. "Simple Keyboard Using the Tone() Function." *Arduino*, https://www.arduino.cc/en/Tutorial/BuiltInExamples/toneKeyboard.

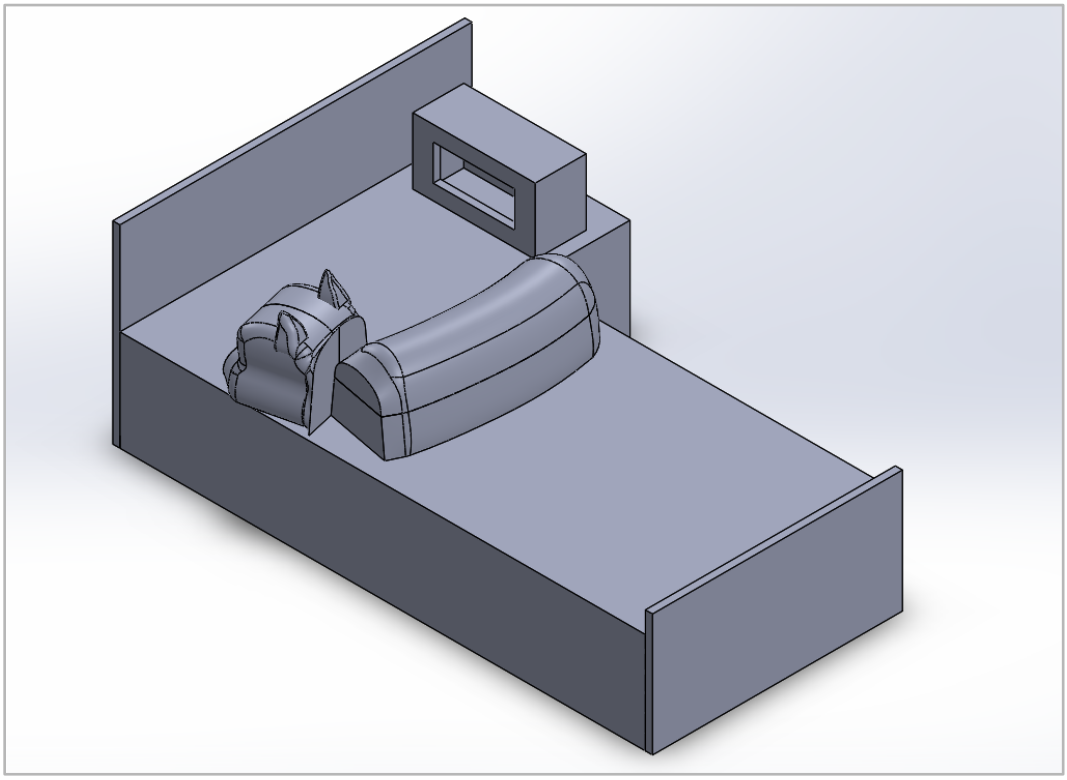
TFK-280SA-22125 Motor - Arduino. TT MOTOR (HK) INDUSTRAIL CO.,LTD, https://www.arduino.cc/documents/datasheets/DCmotor.PDF.

Bill of Materials

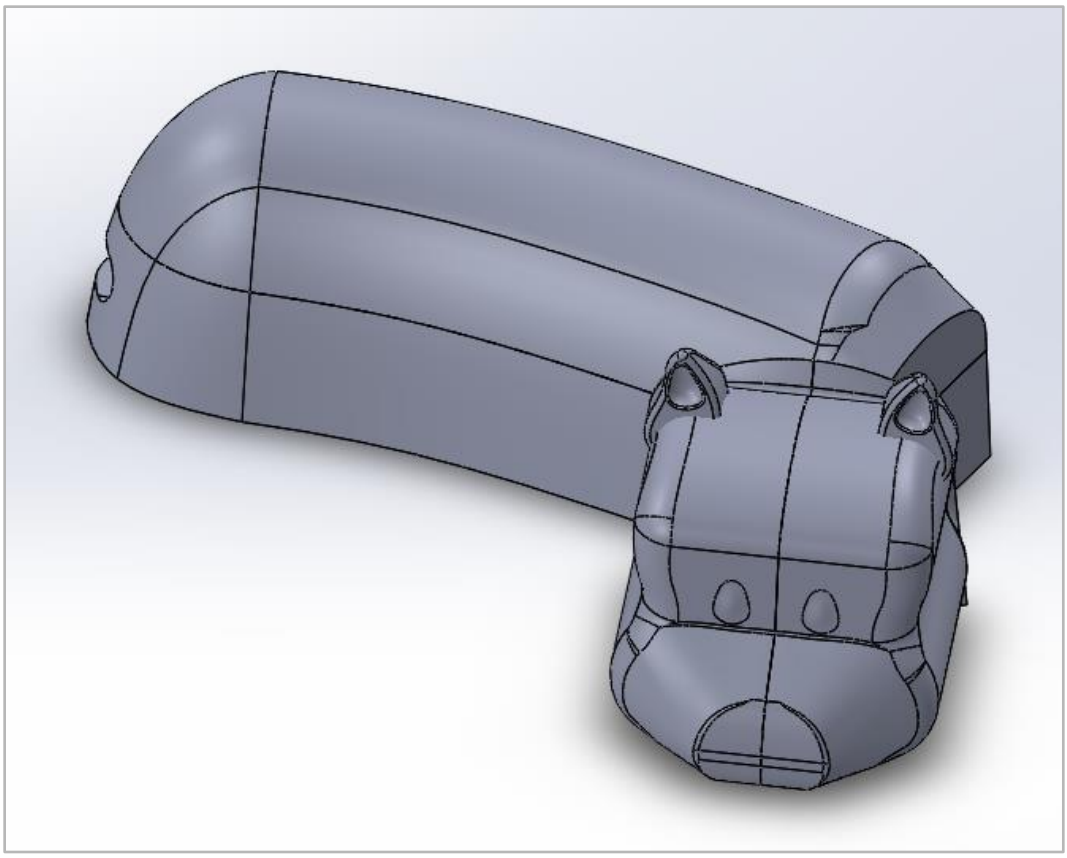
Sub-Assembly	Part Name	Quantity	Function	Finish	Dimensions (L x W x H mm)	Cost Estimate per Quantity ($)
Electronics	Arduino Uno	1	Control overall system with code	Metal	68.6 x 53.4	23.00
	Phototransistor	1	Senses light/darkness levels, which triggers alarm clock on/off	Plastic	5 D x 5.3 H	0.95
	LCD	1	Displays user info; clock countdown & default message	Metal	80 x 35 x 10	9.95
	Speaker Buzzer	1	Makes buzzing alarm sound when	Plastic	20 D x 10 H	0.68
	Red LED	1	Indicator Light, turning on when alarm is triggered	Plastc	5 D x 8 H	0.35
	9V Battery	1	Power system	Metal	45 x 25 x 15	4.00
	Switch	1	Turn power source on/off (as not to waste battery)	Plastic	20 x 13 x 23	0.54
	Piezo Speaker	1	Makes barking sounds	Plastic	28 D x 13 H	2.09
	DFPlayer Mini MP3 Player For Arduino	1	Stores barking sound and relays to speaker	Plastic	20 x 20 x 7	6.99
Bed Assembly	Bed	1	Hold circuitry	Wood	500 x 200 x 85	N/A
	Hinge	1	Open bed frame	Metal		5.30
	Bedside table	1	Hold circuitry	Wood	155 x 50 x 85	N/A
	Alarm Clock	1	Hold LCD and phototransistor	Wood	110 L x 50 W x 60 H	N/A
Appearance	Fabric	3	Decoration	Cloth	N/A	1.00
Rhett Assembly	Body	1	Aesthetic	Plastic		N/A
	Tail	1	Cover phototransistor	Cotton/wood	150 L x 20 W	N/A
	Motor	1	Rotate tail	Metal	30 D x 45 H	2.15

CAD Assembly:

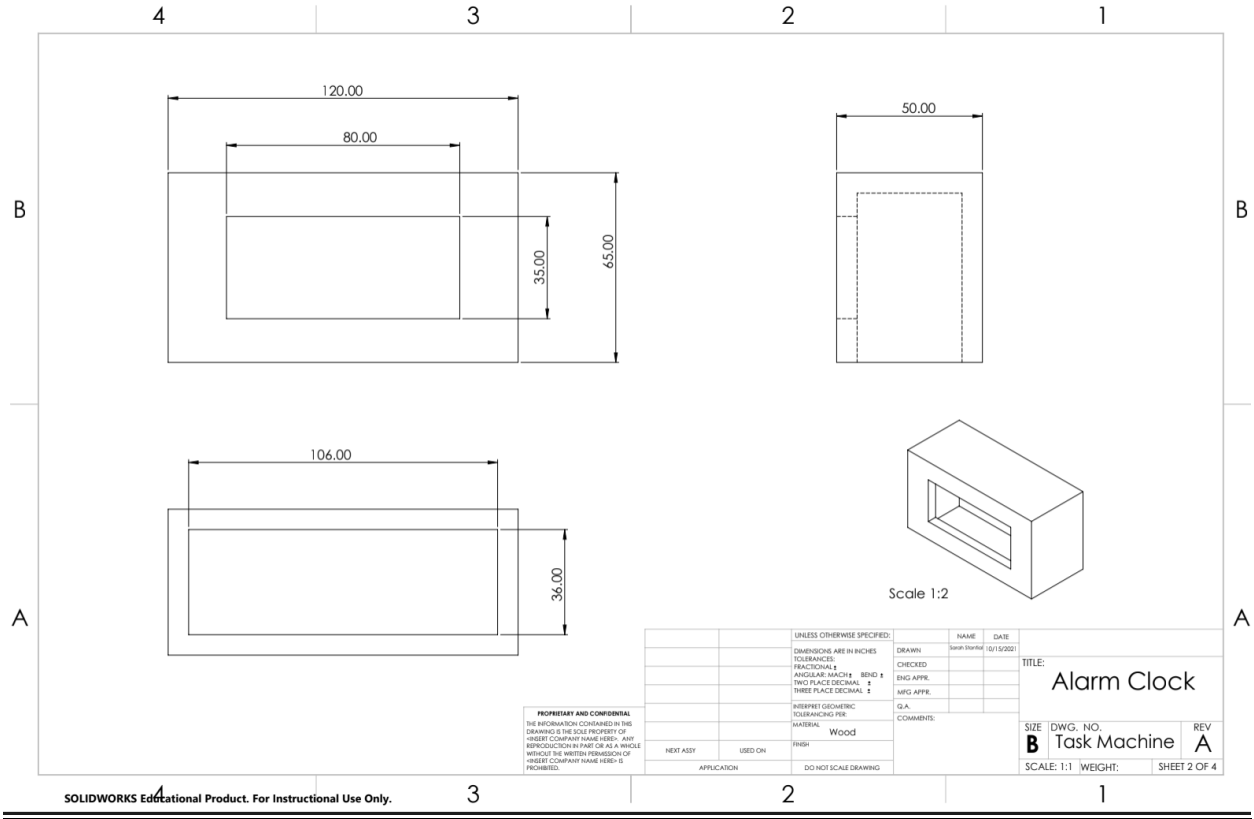
Full Assembly:

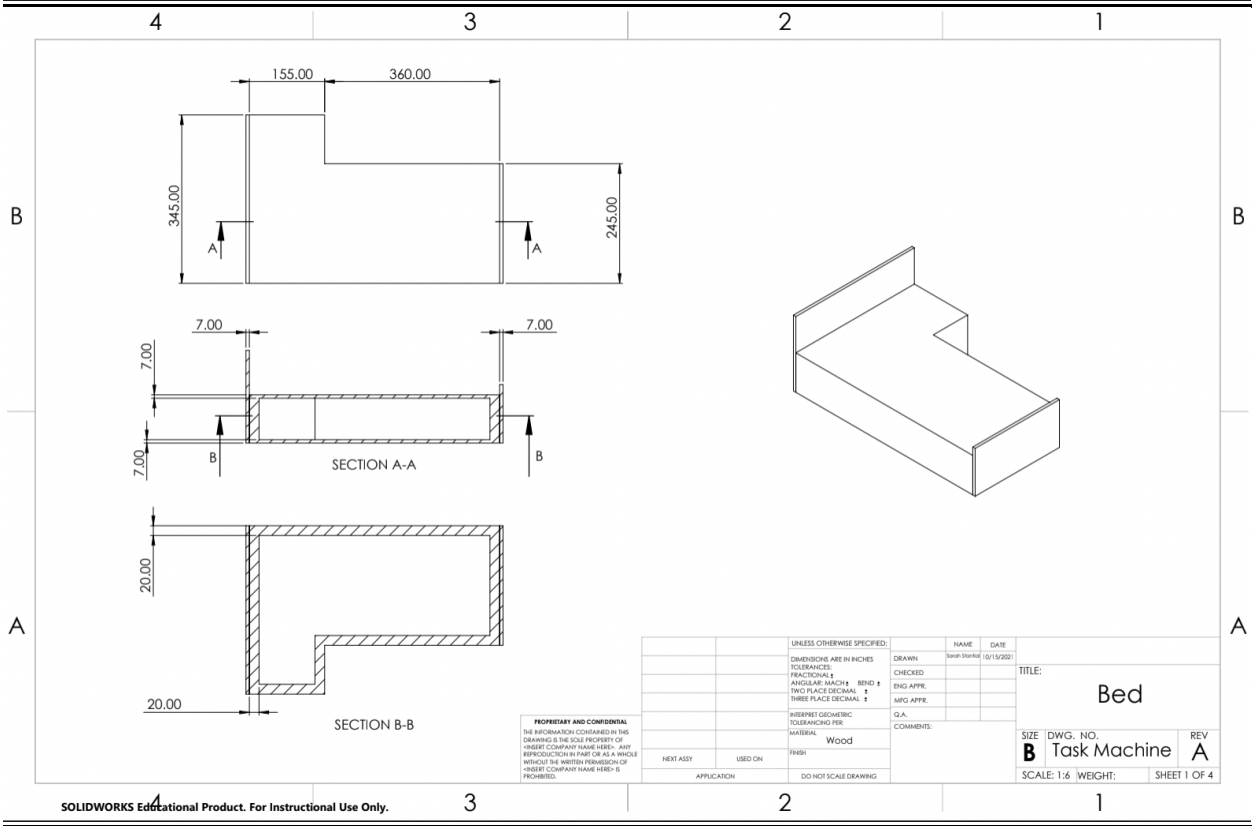


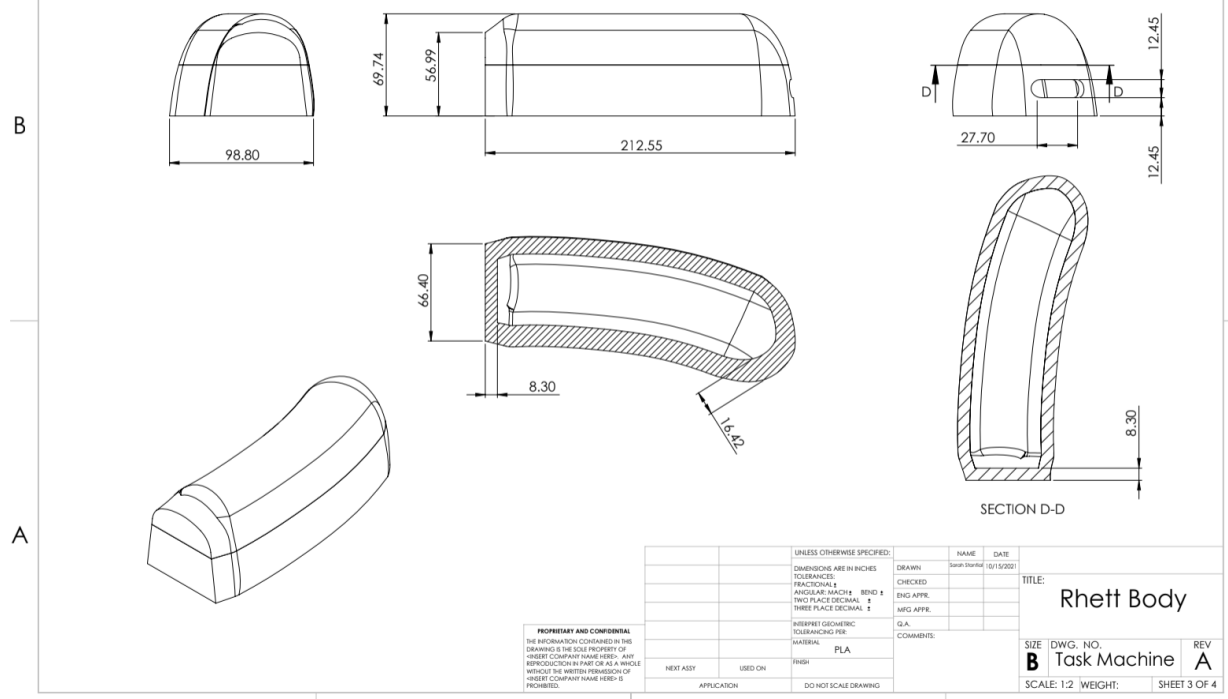
Rhett Assembly



Parts Drawing:







		UNLESS OTHERWISE SPECIFIED:		NAME	DATE
		DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN	Sarah Stanfiel	10/15/2021
			CHECKED		
			ENG APPR.		
			MFG APPR.		
		INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.		
		MATERIAL PLA	COMMENTS:		
		FINISH			
NEXT ASSY	USED ON				
APPLICATION		DO NOT SCALE DRAWING			

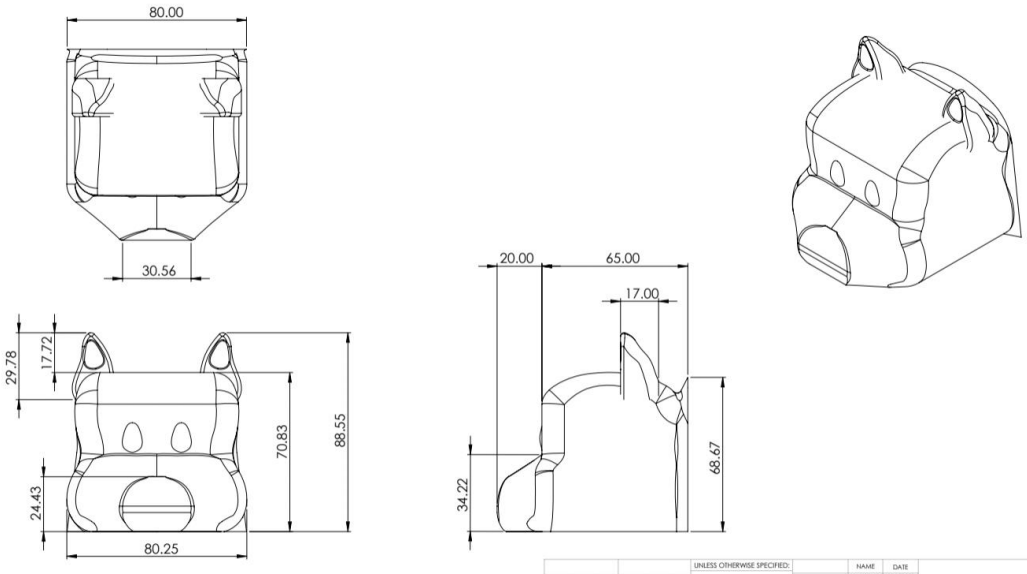
TITLE: Rhett Body

SIZE	DWG. NO.	REV
B	Task Machine	A

SCALE: 1:2 WEIGHT: SHEET 3 OF 4

PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

SOLIDWORKS Educational Product. For Instructional Use Only.



		UNLESS OTHERWISE SPECIFIED:		NAME	DATE
		DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL ± ANGULAR: MACH ± BEND ± TWO PLACE DECIMAL ± THREE PLACE DECIMAL ±	DRAWN	Sarah Stanfiel	10/15/2021
			CHECKED		
			ENG APPR.		
			MFG APPR.		
		INTERPRET GEOMETRIC TOLERANCING PER:	Q.A.		
		MATERIAL PLA	COMMENTS:		
		FINISH			
NEXT ASSY	USED ON				
APPLICATION		DO NOT SCALE DRAWING			

TITLE: Rhett Head

SIZE	DWG. NO.	REV
B	Task Machine	A

SCALE: 1:1.3 WEIGHT: SHEET 4 OF 4

PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

SOLIDWORKS Educational Product. For Instructional Use Only.

Arduino Code:

```
//#include <LiquidCrystal.h>
#include "pitches.h"
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "Arduino.h"
#include "SoftwareSerial.h"
#include "DFRobotDFPlayerMini.h"

SoftwareSerial mySoftwareSerial(5, 6); // RX, TX
DFRobotDFPlayerMini myDFPlayer;
//void printDetail(uint8_t type, int value);

LiquidCrystal_I2C lcd(0x20, 16, 2);

char timeline[16];

const int lightSensorPin = A0;
int motor1pin1 = 2;
int motor1pin2 = 3;
int motorspeed = 9;

int indicatorLCD = 4; //number tbd
int buzzerPin = 11;
int barkPin = 10;
int LCDcount = 0; //To count number of times alarm went off for correct LCD
Display
int growlcount = 0;

//variables
int count_light;  //counts how many times the phototransistor reading has
been triggered
int trigger_count; //to count the 5 seconds of light blockage
int alarm_sound_count;
bool alarm_on;
bool tail_trigger;
int senseCount = 0;

void setup()
{
  // put your setup code here, to run once:
  pinMode(indicatorLCD, OUTPUT);

  pinMode(motor1pin1, OUTPUT);
  pinMode(motor1pin2, OUTPUT);
  pinMode(motorspeed, OUTPUT);

  pinMode(barkPin, OUTPUT);

  count_light = 0; //counts the number of times light sensor has been
triggered so know when to activate alarm
  trigger_count = 0; //counts the # of seconds of light blockage
  alarm_sound_count = 0;
  alarm_on = false;
  tail_trigger = false;
```

```
  //digitalWrite(motorPin, LOW);

  lcd.init();
  lcd.backlight();

  Serial.begin(9600);

  mySoftwareSerial.begin(9600);
  //Serial.begin(115200);

  Serial.println();
  Serial.println(F("DFRobot DFPlayer Mini Demo"));
  Serial.println(F("Initializing DFPlayer ... (May take 3~5 seconds)"));

  if (!myDFPlayer.begin(mySoftwareSerial)) {  //Use softwareSerial to
communicate with mp3.
    Serial.println(F("Unable to begin:"));
    Serial.println(F("1.Please recheck the connection!"));
    Serial.println(F("2.Please insert the SD card!"));
    while(true);
  }
  Serial.println(F("DFPlayer Mini online."));

  myDFPlayer.volume(30);  //Set volume value. From 0 to 30
  myDFPlayer.play(1);

}

void loop() {
  // put your main code here, to run repeatedly:
  //Serial.print("Count light = ");
  //Serial.print(count_light);

  //int count_consecutive_darkness = 0;

  //light sensor
  int lightSensorReading = analogRead(lightSensorPin);

  //printing for fun/testing
  Serial.println(lightSensorReading);
  delay(100); //0.1 seconds


  if (lightSensorReading < 50) //possibly change amount based on testing
  {
    Serial.println("It is dark.");
    trigger_count++;
  }
  else
  {
    Serial.println("It is light.");
    trigger_count = 0;
  }

  if (trigger_count == 5)
  {
```

```
      count_light++;

      if((count_light%2) == 1)
      {
        //triggers timer method/on method
        //light up indicator light
        alarm_on = on_method();
      }
      else if ((alarm_on && count_light%2) == 0)
      {
        //Rhett off
        alarm_on = off_method();
        tail_trigger = tail_off();
      }
  }

  //currently trapped in forever while loop
  if (alarm_on)
  {
    //trigger buzzer
    tone(buzzerPin, 200); //pin#, frequency
    delay(500);
    Serial.print (lightSensorReading);
    noTone(buzzerPin);
    delay(500);
    alarm_sound_count++;

    if (alarm_sound_count >= 5 && tail_trigger == false)
    {
      Serial.print("THE TAIL IS ON IF STATEMENT");
//     // myDFPlayer.next();  //Play the first mp3
//       static unsigned long timer = millis();
//
//    if (millis() - timer > 3000) {
//      timer = millis();
//      myDFPlayer.next();  //Play next mp3 every 3 second.
//    }
      tail_trigger = tail_on();
    }

  }
  else
  {
    noTone(buzzerPin);
    lcd.setCursor(0, 0);
    if (LCDcount == 0)
    {
      lcd.print("  Rhett's Alarm   ");
      lcd.setCursor(0, 1);
      sprintf(timeline,"   for school!");
      lcd.print(timeline);
    }
    else
    {
      lcd.print("  Have fun at   ");
      lcd.setCursor(0, 1);
      sprintf(timeline,"   school!");
```

```
      lcd.print(timeline);
      LCDcount = 0;
    }
  }
}

bool on_method(){
  digitalWrite(indicatorLCD, HIGH); //turning indicator light on, signals
human to move hand away

  //start 5 sec timer
  int timer = 5;

  lcd.clear();
  for(int i=0; i<5; i++){
    //LCD print
    lcd.setCursor(0, 0);
    lcd.print("School starts ");
    lcd.setCursor(0, 1);
    sprintf(timeline,"in %0.2d secs     ", timer);
    lcd.print(timeline);

    timer--;
    delay(1000);
  }
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("School starts ");
  lcd.setCursor(0, 1);
  sprintf(timeline,"NOW!!!!!!");
  lcd.print(timeline);

  //speakers triggered
  return true;
}

 bool off_method(){
  alarm_sound_count = 0;
  digitalWrite(indicatorLCD, LOW);
  return false;
}

bool tail_on(){
//  myDFPlayer.next();
//  delay(1000);
//

    // myDFPlayer.next();  //Play the first mp3
       static unsigned long timer = millis();


    if (millis() - timer > 3000) {
      timer = millis();
      myDFPlayer.next();  //Play next mp3 every 3 second.
    }

  //trigger Rhett movement
```

```
  Serial.print("TAIL IS ON");

  int lightSensorReading = analogRead(lightSensorPin);

  analogWrite(motorspeed, 250); //motor 1, speed range from 0-255
  digitalWrite(motor1pin1, LOW);
  digitalWrite(motor1pin2, HIGH);
  tone(buzzerPin, 200); //pin#, frequency
  delay(500);
  Serial.print (lightSensorReading);
  noTone(buzzerPin);
  delay(500);
  alarm_sound_count++;
  //delay(150);
  digitalWrite(motor1pin1, LOW);
  digitalWrite(motor1pin2, LOW);

    return true;
}

bool tail_off(){

  Serial.print("TAIL IS OFF");

  analogWrite(motorspeed,175);
  digitalWrite(motor1pin1, HIGH);
  digitalWrite(motor1pin2, LOW);
  delay(150);
  digitalWrite(motor1pin1, LOW);
  digitalWrite(motor1pin2, LOW);

  return false;
}

void playTone(uint16_t tone1, uint16_t duration) {
  if(tone1 < 50 || tone1 > 15000) return;  // these do not play on a piezo
  for (long i = 0; i < duration * 1000L; i += tone1 * 2) {
     digitalWrite(barkPin, HIGH);
     delayMicroseconds(tone1);
     digitalWrite(barkPin, LOW);
     delayMicroseconds(tone1);
  }
}

void ruff() {   // dog ruff
  uint16_t i;
  for(i=890; i<910; i+=2)     // "rrr"  (vary down)
     playTone(i,3);
  playTone(1664,150);         // "uuu" (hard to do)
  playTone(12200,70);         // "ff"  (long, hard to do)
}

void arf() {    // dog arf
  uint16_t i;
  playTone(890,25);          // "a"    (short)
  for(i=890; i<910; i+=2)    // "rrr"  (vary down)
     playTone(i,5);
```

```
  playTone(4545,80);          // intermediate
  playTone(12200,70);         // "ff"    (shorter, hard to do)
}
```