Contents

[**Design Brief** 2](#_Toc192753064)

[Sketches 3](#_Toc192753065)

[Chase 3](#_Toc192753066)

[Jake 4](#_Toc192753067)

[Drawings with parts list 5](#_Toc192753068)

[Chase 5](#_Toc192753069)

[Jake 6](#_Toc192753070)

[3D model 7](#_Toc192753071)

[CODE 8](#_Toc192753072)

[PROGRESS PICTURES 12](#_Toc192753073)

[SYSTEM VIDEOS 14](#_Toc192753074)

[Works Cited 15](#_Toc192753075)

# **Design Brief**

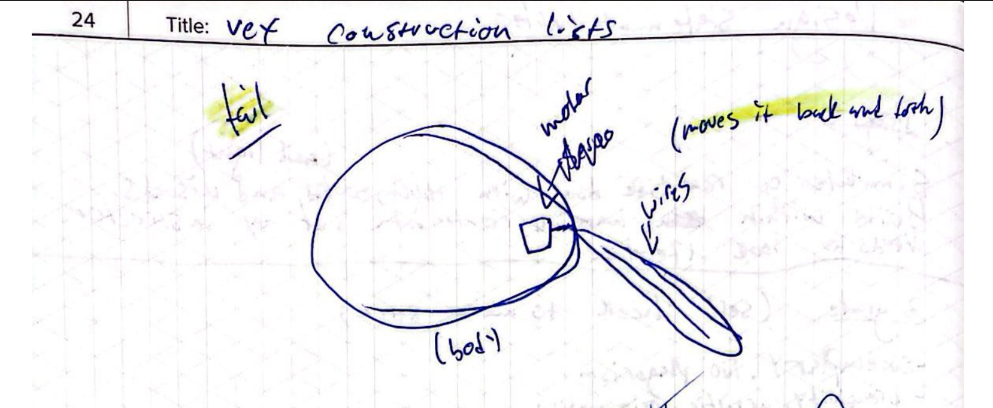
|  |  |
| --- | --- |
| Client | Small business |
| Target Consumer | Tourists |
| Designer(s) | Jake Simmons, [Chase Steenhoven](mailto:steenhovenchasea@scps.net) |
| Problem Statement (restate in your own words) | A small business is showcasing life-like animatronics. They need a dog animatronic under their constraints. The business needs moving parts and it must look good. |
| Design Statement | Using Vex robotics, we will demonstrate biomimicry of a dog. |
| Criteria  How are the listed criteria measurable? | 1. Gear system 2. At least four Moving parts 3. Life like attributes 4. Must look realistic 5. Must produce controlled sound |
| Constraints  How are the listed constraints measurable | 1. 2ft max height 2. 3ft max length 3. 1.25ft max depth 4. Completely autonomous |

# Sketches

## Chase

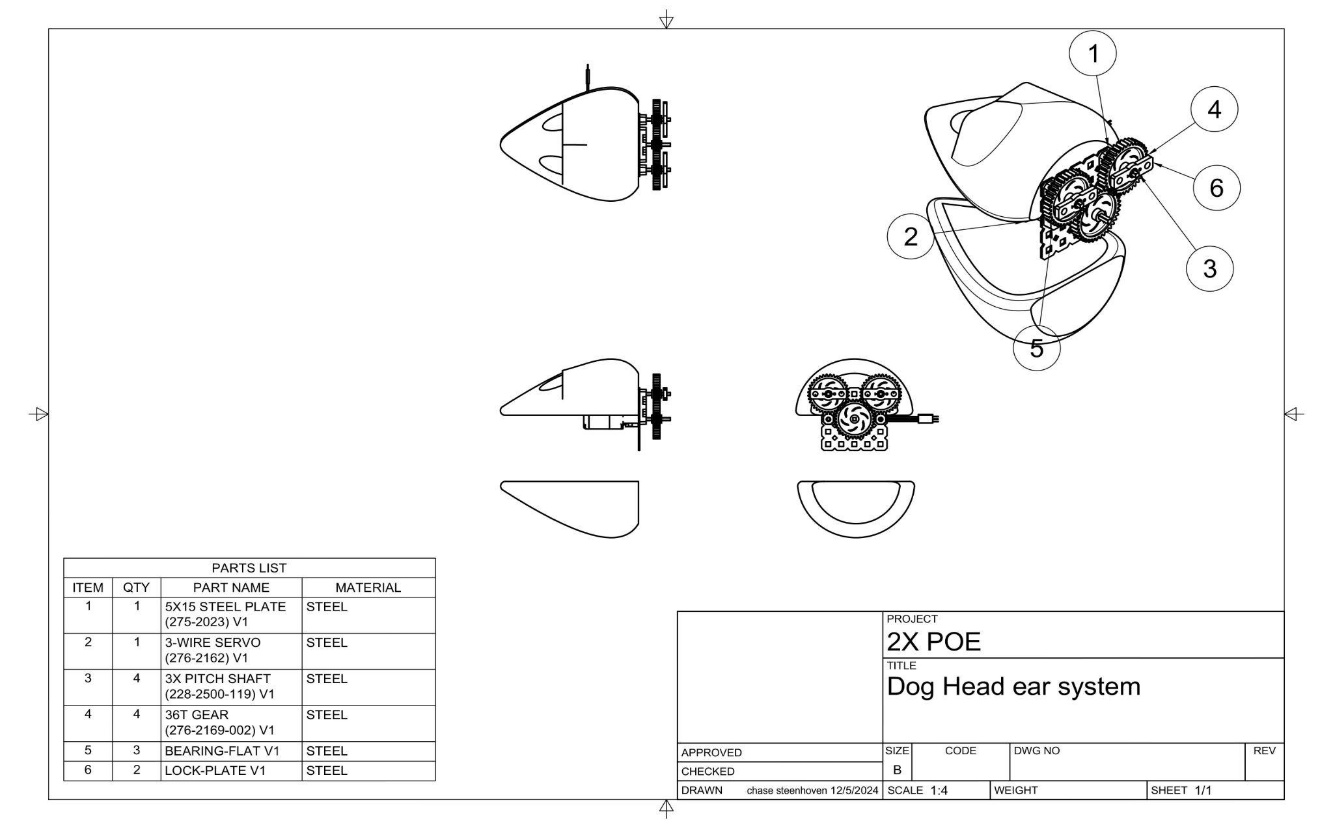


## Jake



# Drawings with parts list

## Chase



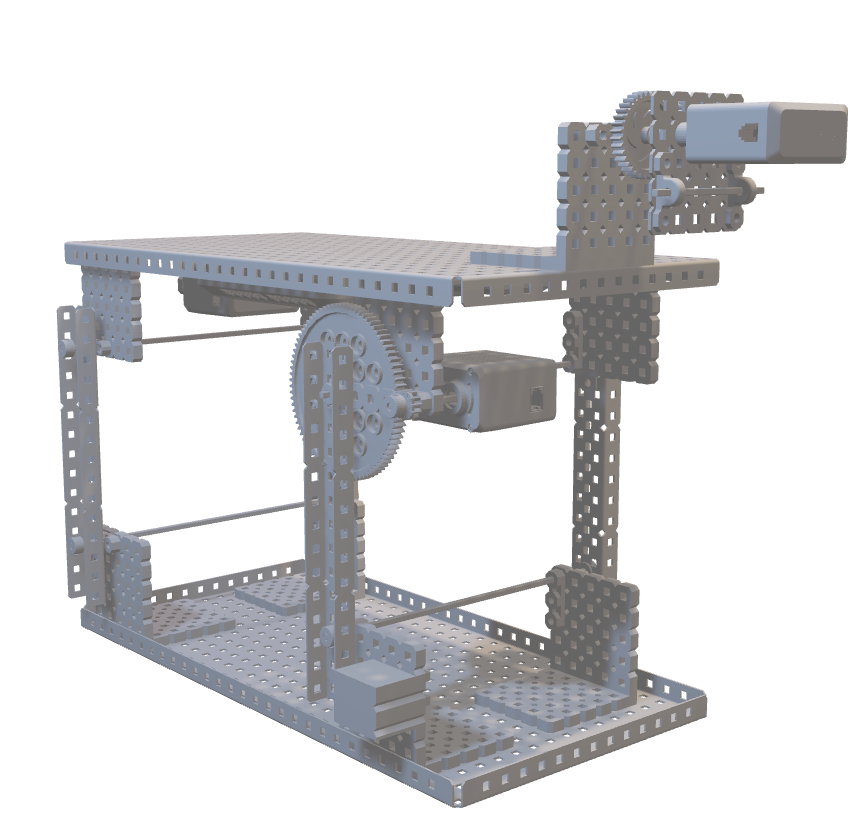
A computer screen shot of a machine

AI-generated content may be incorrect.

## Jake

A diagram of a device

AI-generated content may be incorrect.

3D model

# CODE

#region VEXcode Generated Robot Configuration

from vex import \*

import urandom

# Brain should be defined by default

brain=Brain()

# Robot configuration code

LegMotor = Motor(Ports.PORT1, GearSetting.RATIO\_18\_1, False)

TailMotor = Motor(Ports.PORT2, GearSetting.RATIO\_18\_1, False)

EarServo = Servo(brain.three\_wire\_port.a)

espHappy = DigitalIn(brain.three\_wire\_port.d)

# wait for rotation sensor to fully initialize

wait(30, MSEC)

# Make random actually random

def initializeRandomSeed():

wait(100, MSEC)

random = brain.battery.voltage(MV) + brain.battery.current(CurrentUnits.AMP) \* 100 + brain.timer.system\_high\_res()

urandom.seed(int(random))

# Set random seed

initializeRandomSeed()

def play\_vexcode\_sound(sound\_name):

# Helper to make playing sounds from the V5 in VEXcode easier and

# keeps the code cleaner by making it clear what is happening.

print("VEXPlaySound:" + sound\_name)

wait(5, MSEC)

# add a small delay to make sure we don't print in the middle of the REPL header

wait(200, MSEC)

# clear the console to make sure we don't have the REPL in the console

print("\033[2J")

#endregion VEXcode Generated Robot Configuration

# Library imports

from vex import \*

# Begin project code

#Jake

legGearRatio= 1/7 #gear ratio from motor output shaft to the legs themselves

legForwardsAngle=80 #goal anlge to consider forwards

legUpAngle=-90 #goal angle to be standing up

legBackwardsAngle=0 #goal angle to be standing backwards

tailAAngle=-45 #left tail wag angle

tailBAngle=45 #right tail wag angle

tailSpeed=50 #speed at which to wag

earUpAngle=45 #”happy” ear angle

earDownAngle=-45 #”sad”ear angle

SpinA=True #whether or not tail spins to the left

SpinATo0=False #whether or not tail has met it’s left goal

SpinB=False #whether or not tail spins to the left

SpinBTo0=Fals e#whether or not tail has met it’s right goal

#endJake

#Chase

def legInit():

LegMotor.set\_max\_torque(100,PERCENT) #max torque on legs

LegMotor.set\_velocity(10) #low speed on legs

LegMotor.spin(FORWARD) #spin legs

wait(10,SECONDS) #for 10 seconds

LegMotor.set\_position(0/legGearRatio,DEGREES) #call that position 0

LegMotor.set\_max\_torque(100,PERCENT) #max torque

LegMotor.spin\_to\_position(-70/legGearRatio,DEGREES)#go to up angle

def legSet(goalAngle): #easier function for setting legs position

LegMotor.spin\_to\_position(goalAngle/legGearRatio,DEGREES,wait=False)#spin angle

def layDown(): #easy “lay down” function

legSet(legBackwardsAngle) #spin legs to lay down angle

def standUp(): #easy “stand up” function

legSet(legUpAngle) #spin legs to stand up angle

#endChase

#jake

LegMotor.set\_max\_torque(100,PERCENT )#max leg torque

legInit() #zero legs

TailMotor.set\_position(0,DEGREES) #zero tail

TailMotor.set\_velocity(10,PERCENT) #set tail speeds

#endJake

while True:

#chase

brain.screen.clear\_screen() #clear screen on every iteration

brain.screen.set\_cursor(2,2) #set the printing line top left, with margin

brain.screen.print(espHappy.value()) #print Boolean value of petting sensor

#endChase

#Jake

if(espHappy.value()!=0): #if dog Is happy

layDown()

if (SpinA==True): #if spinngin left

EarServo.set\_position(-45,DEGREES) #ears down

TailMotor.spin\_to\_position(tailAAngle) #tail spins left

SpinB=True #start spinning right

if (SpinB==True): #if spinning right

EarServo.set\_position(0,DEGREES) #ears up

TailMotor.spin\_to\_position(tailBAngle) #spin right

SpinA=True #start spinning left

else: #if dog is sad

TailMotor.spin\_to\_position(0,DEGREES,wait=False) #zero tail

EarServo.set\_position(45,DEGREES) #ears down

standUp() #laydown

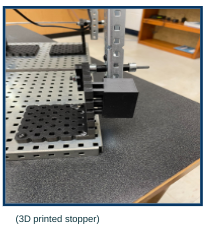
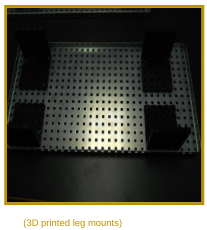
#EndJake



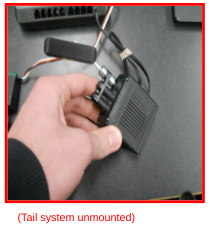
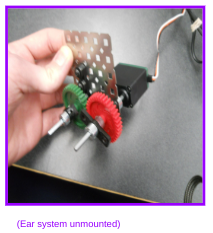
[Github code for speaker](https://github.com/CHASEEEEY127/speaker-test/blob/main/speaker-test.ino)

[Github Code for touch sensor](https://github.com/CHASEEEEY127/POE-Biomicry-CSteenhoven-JSimmons/blob/main/touhcing.ino)

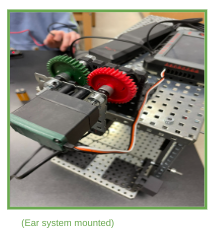
# PROGRESS PICTURES



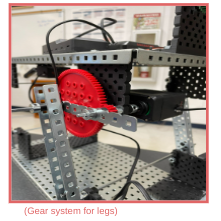
Over-Center-prevention installed leg base mounted



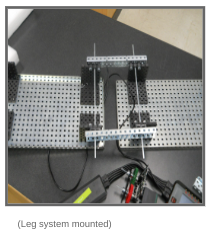
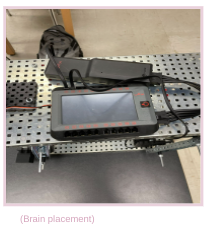
Tail Motor assebly Ear servo assembly



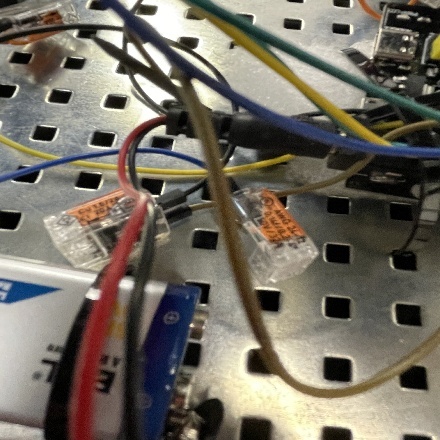
Leg motor mounted Ear servo mounted



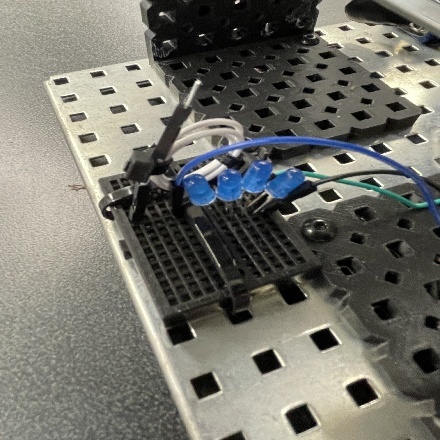
Leg gearing legs mounted



Brain wired torso half-connected



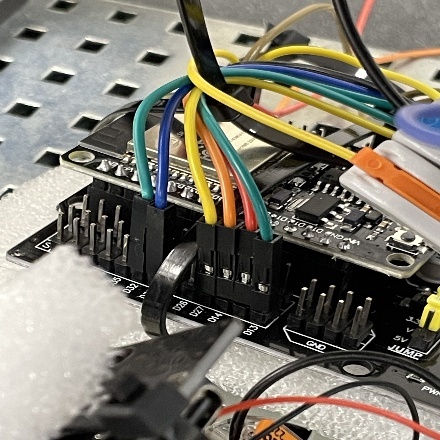
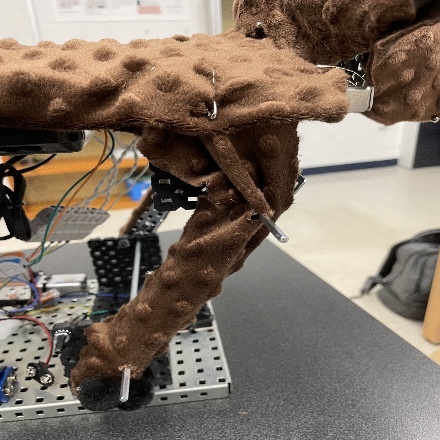
Wiring through fur petting->brain wiring

A close-up of wires and wires

AI-generated content may be incorrect.

A close up of wires

AI-generated content may be incorrect.petting wire umbilical front ball electronics

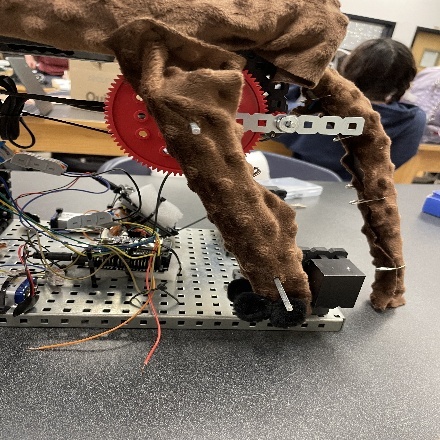
Petting->brain connection decorated ears

# 

Full sound module decorated legs

# 

Battery connection to sound fur coated tail



# 

Forward facing leg clearance decorated face

A hand holding a device with wires and wires

AI-generated content may be incorrect.

Velcro access to battery sound module insulation

# A hand holding a small white object with wires and wires AI-generated content may be incorrect.A person holding a piece of fabric AI-generated content may be incorrect.A hand holding a piece of cloth AI-generated content may be incorrect.A hand holding a green cloth AI-generated content may be incorrect.A close up of ball electronics A dog standing on a green cloth AI-generated content may be incorrect.

Ball closeup full grass cover

Ball access through grass speaker access through grass

Tail internal fluffing sound module packaging

# SYSTEM VIDEOS

* [Leg system](https://drive.google.com/file/d/1ul-jvJ_rD70a3YOX-5GQn_d8HoAwBPtA/view?usp=sharing)

* [Ears+Tail System](https://drive.google.com/file/d/1Yxg9HMDsj4w2f67lpetkK6gTAgpMzn_W/view?usp=sharing)
* [Full System Overview](https://drive.google.com/file/d/1-HWJH-ijQ3y_hkaNyKP5_yNaYcJWg3hf/view?usp=sharing)

# Works Cited

Vexcode v5. (n.d.-b). <https://codev5.vex.com/>

Fusion.online.autodesk.com. (n.d.). <https://fusion.online.autodesk.com/webapp?submit_button=Launch%2BAutodesk%2BFusion>

V5 – vex library. (n.d.). <https://kb.vex.com/hc/en-us/categories/360002333191-V5>

Onshape, a P. B. (n.d.). *Product development platform*. <https://www.onshape.com/en/>