

```

//INVISIBLE CAT
//code by Martin Hanses

//include volume library to control volume/freq in software
#include <Volume.h>
//Library developed by Connor Nishijima (2016)
Volume vol;

//include libraries for the 9dof sensor
#include <Wire.h>
#include <SPI.h>
#include <SparkFunLSM9DS1.h>
//create object
LSM9DS1 imu;

//include libraries and such for the tof sensor
#include "Adafruit_VL53L0X.h"
Adafruit_VL53L0X lox = Adafruit_VL53L0X();

//create variables needed for finding the cat
int upperThreshold = 200;
int margin = 20;
int marginMultiplier = 2;
int lowerThreshold = upperThreshold - margin;
int pinCase = 0;
int dist = 0;

int rollMargin = 10;

int perfectPitch = 266; //Bb

#define DECLINATION -14.08 // Declination (degrees) in Montreal
//used for the 9dof sensor - not in roll, but I enjoyed calculating
this so left it in

//define all pins used
//note: the volume library uses pin 5 by default, so it does not need
defining
#define PRESSURE_PIN A0
#define VIBE_PIN1 7
#define VIBE_PIN2 6
// speaker output pin
#define BUZZER 9

```

```

void setup() {
  Serial.begin(115200);
  while(!Serial);

  pinMode(VIBE_PIN1, OUTPUT);
  pinMode(VIBE_PIN2, OUTPUT);

  //SETUP ToF Sensor
  Serial.println("Adafruit VL53L0X test");
  if (!lox.begin()) { //activate ToF sensor library
    Serial.println(F("Failed to boot VL53L0X"));
    while(1);
  }
  // power
  Serial.println(F("VL53L0X API Simple Ranging example\n\n"));

  //SETUP 9dof sensor
  if (imu.begin() == false) //"activate" library
  {
    while (1);
  }

  vol.begin(); // activate volume library
}

void loop() {

  //play audio based on parameters
  playAudio();

  //make sure the accelerometer is available, and if it is, read its
  data
  if ( imu.accelAvailable() )
  {

    imu.readAccel();
  }

  //call function to measure time of flight to gauge distance
  measureTOF();

  //vibrate as desired

```

```

vibration();

}

//vibration function
void vibration(){

    VL53L0X_RangingMeasurementData_t measure;

    //check if the hand is in the correct position
    if((dist) > lowerThreshold && dist < upperThreshold){

        //establish local roll variables based on info from the accelerometer
        float roll = atan2(imu.ay, imu.az);
        roll *= 180.0;

        //set the correct vibration motor to buzz depending on the movement
        of the hand
        if(roll > -rollMargin && roll < rollMargin){
            digitalWrite(VIBE_PIN1, HIGH);
            digitalWrite(VIBE_PIN2, HIGH);
        }
        else if(roll < -rollMargin){
            digitalWrite(VIBE_PIN1, HIGH);
            digitalWrite(VIBE_PIN2, LOW);
        } else if (roll > rollMargin)
            digitalWrite(VIBE_PIN1, LOW);
            digitalWrite(VIBE_PIN2, HIGH);

        } else { // turn that buzzing off
            digitalWrite(VIBE_PIN1, LOW);
            digitalWrite(VIBE_PIN2, LOW);
        }

    }

}

//function to measure the time of flight
void measureTOF(){
    VL53L0X_RangingMeasurementData_t measure;

    if (measure.RangeStatus != 4) { // phase failures have incorrect
data, according to library

```

```

// set dist variable to be the distance measured
    dist = measure.RangeMilliMeter;
} else {
    Serial.println("hand too far away");
}

}

void playAudio(){

// make sure hand is in the right place
if((dist) > lowerThreshold && dist < upperThreshold){
    //if so, play perfect tone at highest volume
    vol.tone(perfectPitch, 255);

    //otherwise, fade volume and lower pitch based on distance
} else if((dist < lowerThreshold) && (dist > lowerThreshold -
margin*marginMultiplier)) {

    vol.tone((perfectPitch - (lowerThreshold - dist)), (150 -
(lowerThreshold - dist)));
    }
    else if((dist > upperThreshold) && (dist < upperThreshold +
margin*marginMultiplier)) {
        vol.tone((perfectPitch - (dist - upperThreshold)), (150 - (dist
- upperThreshold)));
        }
        else {vol.noTone();}

}

```