# Arduino codes

1. Square FSR

const int FSR\_PIN = A0;

// Pin connected to FSR/resistor divider

// Measure the voltage at 5V and resistance of your 3.3k resistor, and enter

// their value's below:

const float VCC = 4.98;

// Measured voltage of Ardunio 5V line

const float R\_DIV = 3230.0;

// Measured resistance of 3.3k resistor

void setup()

{

  Serial.begin(9600);

  pinMode(FSR\_PIN, INPUT);

}

const int threshold = 40823.3;

//value for force in Grams( grams is defined later) roughly 90lbs

int countAboveThreshold = 0;

// counts how many long the threshold value is being match detected

const int iterationsToHold = 600;

// 20min/2sec (loopdelay)= 600. if the threshold is >= threshold then signal send

const int loopDelay = 2000;

void loop()

{

  int fsrADC = analogRead(FSR\_PIN);

// If the FSR has no pressure, the resistance will be

// near infinite. So the voltage should be near 0.

  if (fsrADC != 0) // If the analog reading is non-zero

  {

// Use ADC reading to calculate voltage:

    float fsrV = fsrADC \* VCC / 1023.0;

// Use voltage and static resistor value to

// calculate FSR resistance:

    float fsrR = R\_DIV \* (VCC / fsrV - 1.0);

    Serial.println("Resistance: " + String(fsrR) + " ohms");

// Guesstimate force based on slopes in figure 3 of

// FSR datasheet:

    float force;

    float fsrG = 1.0 / fsrR;

// Calculate conductance

// Break parabolic curve down into two linear slopes:

    if (fsrR <= 600)

      force = (fsrG - 0.00075) / 0.00000032639;

    else

      force =  fsrG / 0.000000642857;

    Serial.println("Force: " + String(force) + " g");

    Serial.println();

 if (force >= threshold) {

      countAboveThreshold += 1;

    } else {

      countAboveThreshold = 0;

    }

    if (countAboveThreshold == iterationsToHold) {

      Serial.println("person is resting on grate#X ");

//Placeholder action to communicate with GSM sensor FONA-mini cellular SMS breakout

    }

    delay(loopDelay);

  }

  else

  {

// No pressure detected

  }

}

Graphical user interface, application

Description automatically generated

1. Circular Soft Potentiometer

float floatMap(float x, float in\_min, float in\_max, float out\_min, float out\_max) {

  return (x - in\_min) \* (out\_max - out\_min) / (in\_max - in\_min) + out\_min;

}

// the setup routine runs once when you press reset:

void setup() {

  // initialize serial communication at 9600 bits per second:

  Serial.begin(9600);

}

// the loop routine runs over and over again forever:

void loop() {

  // read the input on analog pin A0:

  int analogValue = analogRead(A0);

  // Rescale to potentiometer's voltage (from 0V to 5V):

  float voltage = floatMap(analogValue, 0, 1023, 0, 5);

  float degree = floatMap(analogValue, 0, 1023, 0, 360);

  float resistor = floatMap(analogValue, 0, 1023, 0, 10000);

  // print out the value you read:

  Serial.print("Position Degree: ");

  Serial.print(degree);

  Serial.print(", Resistor: ");

  Serial.println(resistor);

  delay(1000);

}

}

图形用户界面, 文本, 应用程序

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

1. long flex/bend sensor

#define sensorPin  A0 // Flex Sensor is connected to this pin

#define PWMPin  6 // LED is attached to this Pin

float VCC = 5; // Arduino is powered with 5V VCC

float R2 = 10000; // 10K resistor is

float sensorMinResistance = 16700; // Value of the Sensor when its flat

float sensorMaxResistance = 18200; // Value of the Sensor when its bent at 90\*

void setup() {

  Serial.begin(9600); // Initialize the serial with 9600 baud

  pinMode(sensorPin, INPUT); // Sensor pin as input

}

void loop() {

  int ADCRaw = analogRead(sensorPin);

  float ADCVoltage = (ADCRaw \* VCC) / 1023; // get the voltage e.g (512 \* 5) / 1023 = 2.5V

  float Resistance = R2 \* (VCC / ADCVoltage - 1); // Calculate Resistance Value

  uint8\_t ReadValue = map(Resistance, sensorMinResistance, sensorMaxResistance, 0, 255); // map the values 16700 to 0  18200 to 255

  analogWrite(PWMPin, ReadValue); // Generate PWM Signal

  // Print Debug Information

  Serial.print(Resistance);

  Serial.print("  ");

  Serial.println(ReadValue);

  delay(100);

}

Graphical user interface, application

Description automatically generated