Arduino Project- Afraz

Description:

The device I will build is a no-smoking alarm. It will use a gas/smoke sensor to check for smoke. If there is smoke, a specific song will play on a piezo buzzer for a comedic touch. Also, either a red or green LED will be on, with respect to if there is smoke or not.

Equipment Used:

- Arduino Nano Microcontroller
- USB Cable Mini
- Gas sensor
- Piezo Buzzer
- Red LED
- Green LED
- Male to Male Jumper Wires
- Breadboard
- Wires
- 330 Ohm Resistors
- Masking Tape
- Shoe Box

Copy of Sketch:

```
#define NOTE_B0  31
#define NOTE_C1  33
#define NOTE_CS1 35
#define NOTE_D1  37
#define NOTE_DS1 39
#define NOTE_E1  41
#define NOTE_F1  44
#define NOTE_FS1 46
#define NOTE_G1  49
#define NOTE_GS1 52
```

```
#define NOTE_A1  55
#define NOTE_AS1 58
#define NOTE_B1  62
#define NOTE_C2  65
#define NOTE_CS2 69
#define NOTE_D2  73
#define NOTE_DS2 78
#define NOTE_E2  82
#define NOTE_F2  87
#define NOTE_FS2 93
#define NOTE_G2  98
#define NOTE_GS2 104
#define NOTE_A2  110
#define NOTE_AS2 117
#define NOTE_B2  123
#define NOTE_C3  131
#define NOTE_CS3 139
#define NOTE_D3  147
#define NOTE_DS3 156
#define NOTE_E3  165
#define NOTE_F3  175
#define NOTE_FS3 185
#define NOTE_G3  196
#define NOTE_GS3 208
#define NOTE_A3  220
#define NOTE_AS3 233
#define NOTE_B3  247
#define NOTE_C4  262
#define NOTE_CS4 277
#define NOTE_D4  294
#define NOTE_DS4 311
#define NOTE_E4  330
#define NOTE_F4  349
#define NOTE_FS4 370
#define NOTE_G4  392
#define NOTE_GS4 415
```

```
#define NOTE_A4  440
#define NOTE_AS4 466
#define NOTE_B4  494
#define NOTE_C5  523
#define NOTE_CS5 554
#define NOTE_D5  587
#define NOTE_DS5 622
#define NOTE_E5  659
#define NOTE_F5  698
#define NOTE_FS5 740
#define NOTE_G5  784
#define NOTE_GS5 831
#define NOTE_A5  880
#define NOTE_AS5 932
#define NOTE_B5  988
#define NOTE_C6  1047
#define NOTE_CS6 1109
#define NOTE_D6  1175
#define NOTE_DS6 1245
#define NOTE_E6  1319
#define NOTE_F6  1397
#define NOTE_FS6 1480
#define NOTE_G6  1568
#define NOTE_GS6 1661
#define NOTE_A6  1760
#define NOTE_AS6 1865
#define NOTE_B6  1976
#define NOTE_C7  2093
#define NOTE_CS7 2217
#define NOTE_D7  2349
#define NOTE_DS7 2489
#define NOTE_E7  2637
#define NOTE_F7  2794
#define NOTE_FS7 2960
#define NOTE_G7  3136
#define NOTE_GS7 3322
```

```
#define NOTE_A7  3520
#define NOTE_AS7 3729
#define NOTE_B7  3951
#define NOTE_C8  4186
#define NOTE_CS8 4435
#define NOTE_D8  4699
#define NOTE_DS8 4978"
int melody[] = {
  NOTE_A4, 0, NOTE_A4, NOTE_A4,
  NOTE_C5, 0, NOTE_AS4, NOTE_A4,
  NOTE_G4,0, NOTE_G4, NOTE_AS5,
  NOTE_A5, NOTE_AS5, NOTE_A5, NOTE_AS5,
  NOTE_G4,0, NOTE_G4, NOTE_AS5,
  NOTE_A5, NOTE_AS5, NOTE_A5, NOTE_AS5,
  NOTE_AS4, NOTE_AS4, NOTE_AS4, NOTE_AS4,
  NOTE_AS4, NOTE_AS4, NOTE_AS4, NOTE_AS4,
  NOTE_AS4, NOTE_AS4, NOTE_AS4, NOTE_AS4,
  NOTE_AS4, NOTE_AS4, NOTE_AS4, NOTE_AS4,
  NOTE_AS4, NOTE_AS4, NOTE_AS4, NOTE_AS4,
  NOTE_D5, NOTE_D5, NOTE_D5, NOTE_D5,
  NOTE_C5, NOTE_C5, NOTE_C5, NOTE_C5,
  NOTE_F5, NOTE_F5, NOTE_F5, NOTE_F5,
  NOTE_G5, NOTE_G5, NOTE_G5, NOTE_G5,
  NOTE_G5, NOTE_G5, NOTE_G5, NOTE_G5,
  NOTE_G5, NOTE_G5, NOTE_G5, NOTE_G5,
  NOTE_C5, NOTE_AS4, NOTE_A4, NOTE_F4,
  NOTE_G4, 0, NOTE_G4, NOTE_D5,
  NOTE_C5, 0, NOTE_AS4, 0,
  NOTE_A4, 0, NOTE_A4, NOTE_A4,
  NOTE_C5, 0, NOTE_AS4, NOTE_A4,
  NOTE_G4,0, NOTE_G4, NOTE_AS5,
  NOTE_A5, NOTE_AS5, NOTE_A5, NOTE_AS5,
  NOTE_G4,0, NOTE_G4, NOTE_AS5,
  NOTE_A5, NOTE_AS5, NOTE_A5, NOTE_AS5,
  NOTE_G4, 0, NOTE_G4, NOTE_D5,
  NOTE_C5, 0, NOTE_AS4, 0,
```

```
  NOTE_A4, 0, NOTE_A4, NOTE_A4,
  NOTE_C5, 0, NOTE_AS4, NOTE_A4,
  NOTE_G4,0, NOTE_G4, NOTE_AS5,
  NOTE_A5, NOTE_AS5, NOTE_A5, NOTE_AS5,
  NOTE_G4,0, NOTE_G4, NOTE_AS5,
  NOTE_A5, NOTE_AS5, NOTE_A5, NOTE_AS5
};
// note durations: 4 = quarter note, 8 = eighth note, etc.:
int noteDurations[] = {
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
```

```
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  4,4,4,4,
  };

int redLed = 12;
int greenLed = 11;
int buzzer = 10;
int smokeA0 = A5;
// My threshold value
int sensorThres = 280;

void setup() {
  pinMode(redLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(smokeA0, INPUT);
  Serial.begin(9600);
}

void loop() {
  int analogSensor = analogRead(smokeA0);

  Serial.print("Pin A0: ");
  Serial.println(analogSensor);
  // Checks if it has reached the threshold value
  if (analogSensor > sensorThres)
  {
    digitalWrite(redLed, HIGH);
    digitalWrite(greenLed, LOW);
    tone(buzzer, 1000, 200);
    for (int thisNote = 0; thisNote < 112; thisNote++) {
    int noteDuration = 750 / noteDurations[thisNote];
```

```
    tone(7, melody[thisNote], noteDuration);
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);

    noTone(7);
    }
  }
  else
  {
    digitalWrite(redLed, LOW);
    digitalWrite(greenLed, HIGH);
    noTone(buzzer);
  }
  delay(100);
}
```

Explanation of Sketch:

Also see attached flowchart.

There are 2 major aspects of this sketch, the musical tunes and the smoke detection. At the start of the code, there are variables of musical note names assigned to their respective frequencies. This creates a temporary library of musical notes, ready for use. After this, a variable called “melody” is created. This contains the order of notes used for the song. Next up is a variable called “noteDurations”. This will show how long the note should be played for. After this, variables are assigned to pins for the LEDs, buzzer and gas sensor. Another variable is also created for adjusting the threshold of the sensor. Next, the void setup command is used. The LED and buzzer pins will be outputs and the gas sensor will be the input. After this, the void loop command is used. Firstly, it will check the gas level of the gas sensor. Then, an “if” statement is used to check if the gas level is over the sensor threshold. If it is, the red LED will turn on and play the music. If the gas level is lower than the sensor threshold, the green LED will turn on and no song will play.

Sketch Source:

https://create.arduino.cc/projecthub/Sparkbuzzer/astronomia-coffin-dance-meme-music-using-arduino-uno-f8f720

https://create.arduino.cc/projecthub/Aritro/smoke-detection-using-mq-2-gas-sensor-79c54a

Demo:

See attached video to the assignment.