**IOT PROJECT**

**Integrating a Heartbeat Sensor with a Piezoelectric Buzzer**

**PRITHVI MENON**

**VIT VELLORE**

**INTRODUCTION**

There is little need for an Internet of Things (IoT) based system for visually impaired people who want to monitor their own heartbeat in the comfort of their own homes. People with visual impairments have relied on medical professionals for years, including traveling to a hospital to have their heartbeats monitored. As though for those who do not have vision impairments, this issue has never occurred. The issue with many portable gadgets that measure heart rate or pulse, but not in a way that is inclusive like attending to the It has been established that the requirements of the visually impaired are costly constructs. And, that is why, we have decided to take on this challenge of helping the visually impaired by designing an IoT construct system that makes use of simple sensors that are ultimately cost effective and proven to be the efficient yet sophisticated ways to solve this extravagantly ignored conundrum of today’s technology-based world.

**COMPONENTS USED**

* Arduino UNO
* KY-039 sensor (1 unit)
* Piezoelectric Buzzer (1 unit)
* Jumper wires (as required)
* Breadboard (1 unit)
* Arduino IDE (Pre-installed in a PC/MAC/Laptop)
* USB Power Cable (1 unit)

**CODE**

// Pulse Monitor Test Script

int sensorPin = 0;

double alpha = 0.75;

double change = 0.0;

double minval = 0.0;

float bpthres1=70.0;

float bpthres2=90.0;

const int buzzer=9;

float bpthres3=120.0;

void setup ()

{

Serial.begin (9600);

}

void loop ()

{

static double oldValue = 0;

static double oldChange = 0;

int rawValue = analogRead (sensorPin);

double value = alpha \* oldValue + (1 - alpha) \* rawValue;

if(bpthres1<=value<=bpthres2)

{

tone(buzzer,1000);

delay(1000);

noTone(buzzer);

delay(1000);

Serial.print (rawValue);

Serial.print (",");

Serial.println (value);

oldValue = value;

delay (1000);}

if(bpthres2<=value<=bpthres3)

{

tone(buzzer,1000);

delay(100);

noTone(buzzer);

delay(100);

tone(buzzer,1000);

delay(100);

noTone(buzzer);

delay(1000);

Serial.print (rawValue);

Serial.print (",");

Serial.println (value);

oldValue = value;

delay (1000);}

if(bpthres1<=value<=bpthres2)

{

tone(buzzer,1000);

delay(100);

noTone(buzzer);

delay(100);

tone(buzzer,1000);

delay(100);

noTone(buzzer);

delay(100);

tone(buzzer,1000);

delay(100);

noTone(buzzer);

delay(1000);

Serial.print (rawValue);

Serial.print (",");

Serial.println (value);

oldValue = value;

delay (1000);}

}

**CONSTRUCTION OF THE CONSTRUCT**

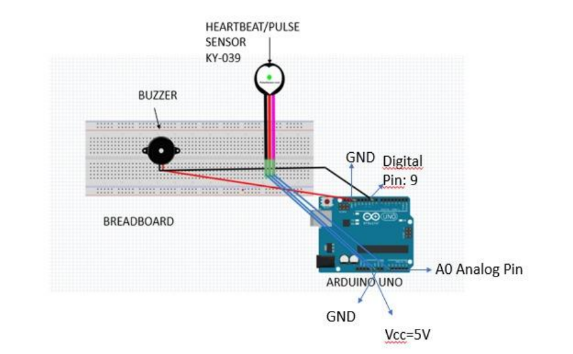
* Arduino UNO is connected to a computer via the USB
* The rightmost pin of the KY-039 sensor (when facing forward to you) is connected to the first analog pin on the Arduino UNO (I.e., Pin A0)
* The middle of the KY-039 sensor is connected to the GND pin on the Arduino UNO.
* The leftmost pin of the KY-039 sensor is connected to the 5V VCC pin on the Arduino UNO
* The previous 3 steps are all carried on a breadboard which are connected via the jumper wires.
* The positive terminal of the buzzer is connected to any digital pin on the Arduino UNO. In this case, we’ve connected the positive terminal to Pin 9.
* The negative terminal of the buzzer is connected to the GND pin on the Arduino UNO.
* The previous 2 steps are also carried forward on the breadboard with the help of jumper wires.
* The above code section is uploaded to the Arduino UNO by connecting the Arduino board to the PC/laptop/MAC via the USB cable from the Arduino IDE which has to be pre-installed in the PC/laptop
* Keep in mind before you upload the code, verify the code using the inbuild compiler within the IDE and check whether the board and port selections are accurate to the model of the Arduino Board that you’ve decided on utilizing for this construct.
* After the code is uploaded, the Arduino UNO immediately works on giving the desired output which is only possible when you give and input by placing your finger between the IR LED and Phototransistor.
* Your heartbeat/pulse can be viewed from the Serial Monitor or even the serial plotter for a graphical representation of the output that is expected to be derived from the working of the aforesaid construct.
* The whole point was to tackle the issue of the visually impaired not having access to or the freedom to check their own heartbeat and so, when they place their finger between the IR LED and phototransistor, the buzzer will buzz a specific number of times to let them know their pulse. It can be viewed in the below buzzing representation table.

**BUZZING REPRESENTATION TABLE**

The number of buzzes for each set of pulses are as follows.

|  |  |
| --- | --- |
| Buzzes | Pulse range |
| 1 | 71 and less |
| 2 | 71-90 |
| 3 | 91 and above |

**CIRCUIT DIAGRAM**



**CONCLUSION OF REPORT**

As a team we have learned the basics of an Arduino UNO. That includes connecting a

healthcare-based sensor like the KY-039 and the use of output-based devices like a piezo electric

buzzer. We have also learned to use the serial monitor and serial plotter to observe the outputs

both analog and digital that are collected from sensors that are mentioned above. This project has

also enlightened all the members of the team on the issues that are faced by the visually implored

on a daily basis as well the terms where they have to struggle a lot more to get access of their

own heartbeat in a more complex way. And this project is believed to help people out there in an

effective way.