

###ARDUINO UNO CODE###

```
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>

float pulse = 0;
float temp = 0;
SoftwareSerial ser(9,10);
String apiKey = "U5AGYVQT5JRJPSTR";

int pulsePin = A0;
int blinkPin = 7;
int fadePin = 8;
int fadeRate = 0;
LiquidCrystal lcd(13, 12, 6, 5, 4, 3);
// Volatile Variables, used in the interrupt service routine!
volatile int BPM;
volatile int Signal;
volatile int IBI = 600;
volatile boolean Pulse = false;
volatile boolean QS = false;

static boolean serialVisual = true;
volatile int rate[10];
volatile unsigned long sampleCounter = 0;
volatile unsigned long lastBeatTime = 0;
volatile int P = 512;
volatile int T = 512;
volatile int thresh = 525;
```

```
volatile int amp = 100
volatile boolean firstBeat = true;
volatile boolean secondBeat = false;

void setup()
{
  lcd.begin(16, 2);
  pinMode(blinkPin,OUTPUT);
  pinMode(fadePin,OUTPUT);
  Serial.begin(115200);
  interruptSetup();

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("IOT sensors");
  ser.begin(9600);
  ser.println("AT");
  delay(1000);
  ser.println("AT+GMR");
  delay(1000);
  ser.println("AT+CWMODE=3");
  delay(1000);
  ser.println("AT+RST");
  delay(5000);
  ser.println("AT+CIPMUX=1");
  delay(1000);
  String cmd="AT+CWJAP="IOT sensors ","temp & pulse";
  ser.println(cmd);
  delay(1000);
```

```
ser.println("AT+CIFSR");  
delay(1000);  
}
```

```
void loop()  
{  
  serialOutput();  
  if (QS == true)  
  {  
    fadeRate = 255;  
    serialOutputWhenBeatHappens();  
    QS = false;  
  }  
  ledFadeToBeat();  
  delay(20);  
  read_temp();  
  esp_8266();  
}
```

```
void ledFadeToBeat()  
{  
  fadeRate -= 15;  
  fadeRate = constrain(fadeRate,0,255);  
  analogWrite(fadePin,fadeRate);  
}
```

```
void interruptSetup()  
{  
  TCCR2A = 0x02;
```

```
TCCR2B = 0x06;
OCR2A = 0X7C;
TIMSK2 = 0x02;
sei();
}
```

```
void serialOutput()
{
  if (serialVisual == true)
  {
    arduinoSerialMonitorVisual('-', Signal);
  }
  else
  {
    sendDataToSerial('S', Signal);
  }
}
```

```
void serialOutputWhenBeatHappens()
{
  if (serialVisual == true) // Code to Make the Serial Monitor Visualizer Work
  {
    Serial.print("*** Heart-Beat Happened *** ");
    Serial.print("BPM: ");
    Serial.println(BPM);
    lcd.clear();
    lcd.print("BPM: ");
    lcd.print(BPM);
  }
}
```

```
else
{
    sendDataToSerial('B',BPM);
    sendDataToSerial('Q',IBI);
}
}
```

```
void arduinoSerialMonitorVisual(char symbol, int data )
{
    const int sensorMin = 0;
    const int sensorMax = 1024;
    int sensorReading = data; // map the sensor range to a range of 12 options:
    int range = map(sensorReading, sensorMin, sensorMax, 0, 11);
    switch (range)
    {
        case 0:
            Serial.println("");;    /////ASCII Art Madness
            break;
        case 1:
            Serial.println("---");
            break;
        case 2:
            Serial.println("-----");
            break;
        case 3:
            Serial.println("-----");
            break;
        case 4:
            Serial.println("-----");
```

```

        break;
    case 5:
        Serial.println("-----|-");
        break;
    case 6:
        Serial.println("-----|---");
        break;
    case 7:
        Serial.println("-----|-----");
        break;
    case 8:
        Serial.println("-----|-----");
        break;
    case 9:
        Serial.println("-----|-----");
        break;
    case 10:
        Serial.println("-----|-----");
        break;
    case 11:
        Serial.println("-----|-----");
        break;
    }
}

```

```

void sendDataToSerial(char symbol, int data )
{
    Serial.print(symbol);
    Serial.println(data);
}

```

```

}
ISR(TIMER2_COMPA_vect)
{
    cli();
    Signal = analogRead(pulsePin);
    sampleCounter += 2;
    int N = sampleCounter - lastBeatTime;
    if(Signal < thresh && N > (IBI/5)*3)
    {
        if (Signal < T)
        {
            T = Signal;
        }
    }
    if(Signal > thresh && Signal > P)
    {
        P = Signal;
    }

    if (N > 250)
    {
        if ( (Signal > thresh) && (Pulse == false) && (N > (IBI/5)*3) )
        {
            Pulse = true;
            digitalWrite(blinkPin,HIGH);
            IBI = sampleCounter - lastBeatTime;
            lastBeatTime = sampleCounter;
            if(secondBeat)
            {

```

```

        secondBeat = false;
        for(int i=0; i<=9; i++)
        {
            rate[i] = IBI;
        }
    }
    if(firstBeat)
    {
        firstBeat = false;
        secondBeat = true;
        sei();
        return;
    }
    word runningTotal = 0;
    for(int i=0; i<=8; i++)
    {
        rate[i] = rate[i+1];
        runningTotal += rate[i];
    }
    rate[9] = IBI;
    runningTotal += rate[9];
    runningTotal /= 10;
    BPM = 60000/runningTotal;
    QS = true;
    pulse = BPM;
}
}
if (Signal < thresh && Pulse == true)
{

```



```

    digitalWrite(blinkPin,LOW);
    Pulse = false;
    amp = P - T;
    thresh = amp/2 + T;
    P = thresh;
    T = thresh;
}
if (N > 2500)
{
    thresh = 512;
    P = 512;
    T = 512;
    lastBeatTime = sampleCounter;
    firstBeat = true;
    secondBeat = false;
}
sei();
}

void esp_8266()
{
    String cmd = "AT+CIPSTART=4,\"TCP\", \"";
    cmd += "184.106.153.149";
    cmd += "\",80";
    ser.println(cmd);
    Serial.println(cmd);
    if(ser.find("Error"))
    {
        Serial.println("AT+CIPSTART error");
    }
}

```

```

    return;
}
//api_key=LHAG4NSIYJ5UWS6U&field1=0\
String getStr = "GET /update?api_key=";
getStr += apiKey;
getStr += "&field1=";
getStr += String(temp);
getStr += "&field2=";
getStr += String(pulse);
getStr += "\r\n\r\n";
cmd = "AT+CIPSEND=4,";
cmd += String(getStr.length());

ser.println(cmd);
Serial.println(cmd);
delay(1000);
ser.print(getStr);
Serial.println(getStr);
delay(3000);
}

void read_temp()
{
    int temp_val = analogRead(A1);
    float mv = (temp_val/1024.0)*5000;
    float cel = mv/10;
    temp = (cel*9)/5 + 32;
}

```

CIRCUIT: --

