**Body Temperature Sensing System**

The DS18B20 temperature sensor was used to measure the body temperature which is connected to the Node-MCU. The VCC of the DS18B20 temperature sensor is connected to the +3V terminal on the Node-MCU. The signal pin is connected to the digital pin (D2) with additional voltage of 3V is supplied to this pin through a 4.7KΩ resistor and the last pin, the GND pin is connected to the ground pin of the Node-MCU.

To get the sensor’s raw data and to convert it Degree Celsius ( ̊ C) of the body temperature, the following codes are used:

|  |
| --- |
| *//Including all the needed libraries*  #include <OneWire.h>  #include <DallasTemperature.h>  *//Requesting for the temperature in ̊ C.*  sensors.requestTemperatures();  *//Receiving the temperature values and storing into a variable.*  float temperatureC = sensors.getTempCByIndex(0);  *//Printing the temperature as the output.*  Serial.print(temperatureC );  Serial.println("ºC"); |

The above codes are not the complete codes yet the crucial functions codes to receive the temperature values.

**Pulse Sensing System**

To detect the pulse of the patient, this system was included with a Pulse Detecting Module. The VCC of the pulse sensor is connected to the +3V terminal on the Node-MCU. the GND pin is connected to the ground pin of the Node-MCU. The signal pin is connected to the analog pin (A0). For this system the module’s connection are as below:

* VCC → 3V of the Node-MCU
* GND → Ground of the Node-MCU
* Signal → Pin A0 of the Node-MCU

To obtain the beat per minute (BPM) of the patient, the following codes were used. There were no additional libraries that was used in these codes.

|  |
| --- |
| void readPulse() {  Signal = analogRead(pulsePin);  sampleCounter += 2;  int N = sampleCounter - lastBeatTime;  detectSetHighLow();  if (N > 250) {  if ( (Signal > thresh) && (Pulse == false) && (N > (IBI / 5) \* 3) )  pulseDetected();  }  if (Signal < thresh && Pulse == true) {  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 = true;  }  }  void detectSetHighLow() {  if (Signal < thresh && N > (IBI / 5) \* 3) {  if (Signal < T) {  T = Signal;  }  }  if (Signal > thresh && Signal > P) {  P = Signal;  }  }  void pulseDetected() {  Pulse = true;  IBI = sampleCounter - lastBeatTime;  lastBeatTime = sampleCounter;  if (firstBeat) {  firstBeat = false;  return;  }  if (secondBeat) {  secondBeat = false;  for (int i = 0; i <= 9; i++) {  rate[i] = IBI;  }  }  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;  delay(1);  } |

The above code are the main function to detect if there are any pulse and calculate the BPM of the patient.