UNIVERSITY OF RWANDA/ ACEIOT

SMART SENSORS AND ACTUATORS

GROUP MEMBERS:

1. NIYITEGEKA Janvier
2. UWUMUREMYI Gaspard Nelly
3. KAMUHANDA Deny-Beny

TOPIC: **IoT Based Patient Monitoring System**

INTRODUCTION

The proposed system helps the patients to do some auto-check up of their health status by using medical sensors like Temperature sensor for detecting the body temperature and Pulse sensor for detecting the heart beat rate of the person without intervention of doctor physical presence. Secondly, it will help the patients to communicate their diseases to the nearby health centers through mqtt broker communication technology and allow the patients to get the advice and appointment from the doctor in real time using mqtt protocol. But, because of lacking of the NTC Thermistor to be used for measuring the body temperature as planned, DHT22 was used instead.

PART I:

In this part we are going to show you how to collect temperature data and detect heartbeat pulse from temperature and pulse sensors respectively and display the information on the serial monitor. Secondly, we are going show you how to publish the collected data on cloud or Message Queuing Telemetry Transport protocol (MQTT).In the next part we will show you how to read BPM data from pulse sensor and push them to cloud too.

COMPONENTS USED:

1. NodeMCU esp8266
2. DHT22 sensor (NTC Thermistor)
3. Pulse Sensor
4. Breadboard

BROKER USED:

1. things.ubidots.com

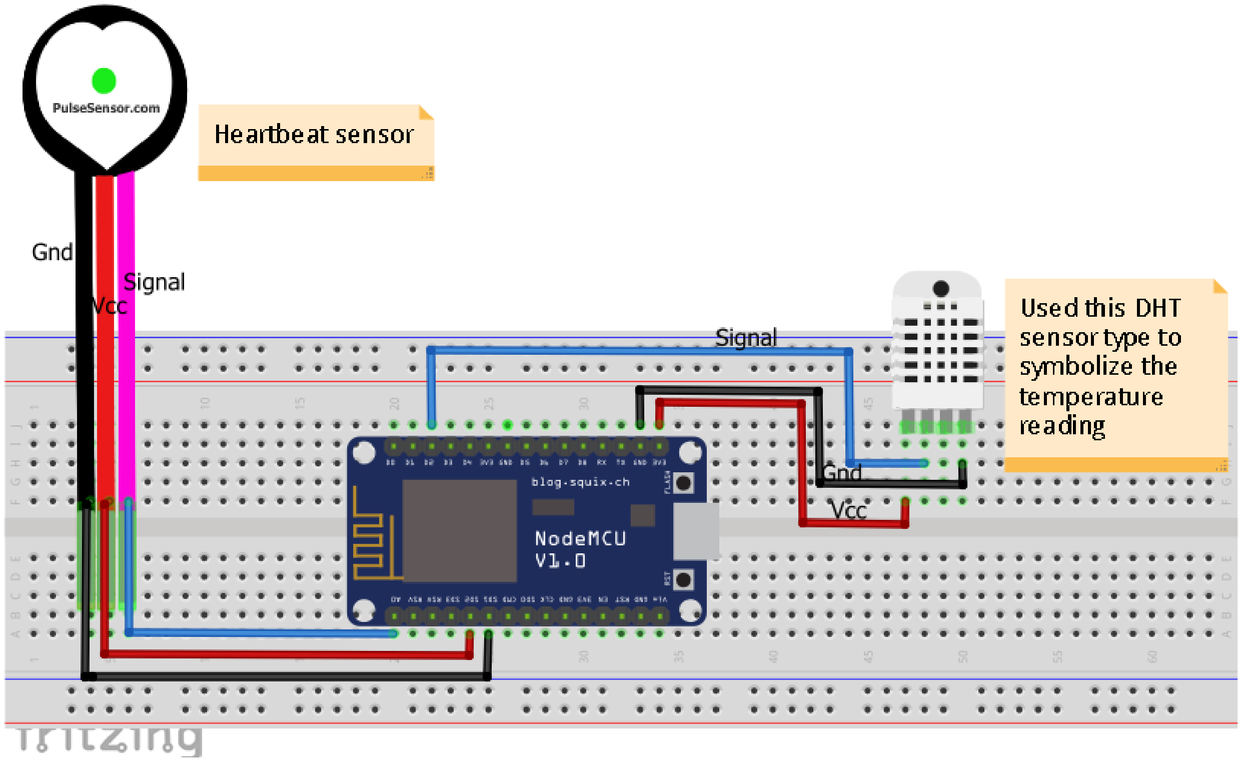
CIRCUIT CONNECTION

a) Pulse sensor-Nodemcu connection

PULSE SENSOR	**NODEMCU**
GND	GND
VCC	3.3v
Signal	A0

b) DHT22 Sensor-Nodemcu connection

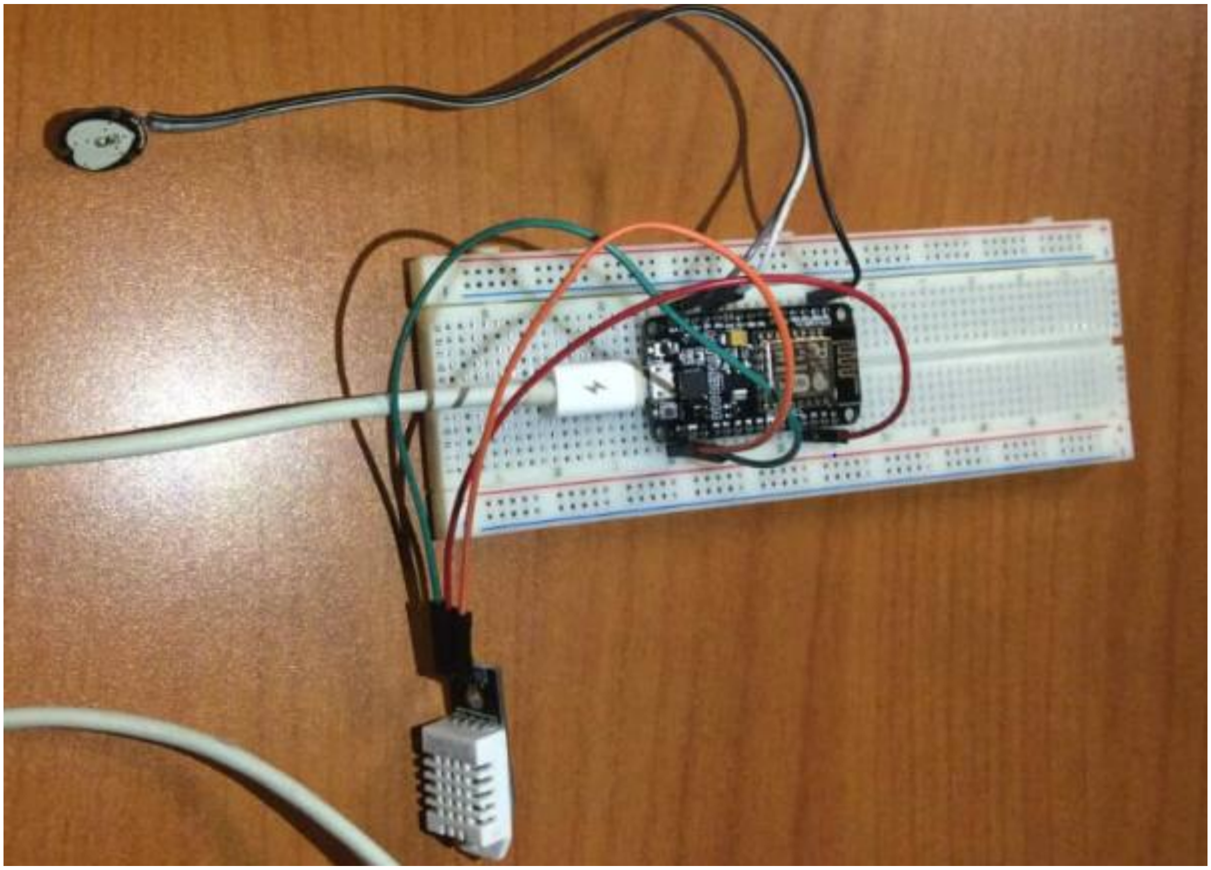
DHT22 Sensor	**NODEMCU**
GND	GND
VCC	3.3v
Signal	D2



Heartbeat sensor

Used this DHT sensor type to symbolize the temperature reading

PROJECT IMPLEMENTATION



PROGRAM CODE

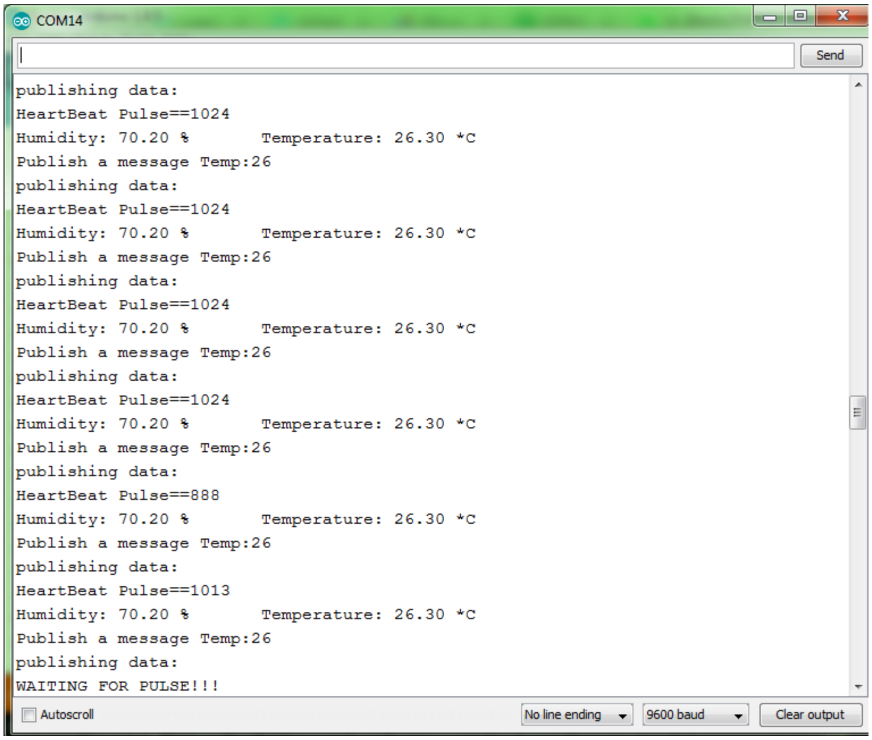
```
#include <ESP8266WiFi.h>

#include <PubSubClient.h>

#include "DHT.h"

#define DHTPIN D2

#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

//add network information

const char* ssid = "nijas2012";

const char* password = "urwombonye";
```

```
//add broker information
#define TOKEN "A1E-BgFDEzhAYKGbZJ1q1dpWDS5W36Fklw"
#define MQTT_CLIENT_NAME "Janvier"
#define Variable_label "temperature"
#define Variable_label2 "pulse-sensor"
#define device_label "meet-doctor-iot-based"
const char* mqttbroker = "things.ubidots.com";// mqtt broker
char payload[100];
char topic[50];
char topicSubscribe[150];
char sensor_data[10];
WiFiClient ubidots;
PubSubClient client(ubidots);
//declare and initialize pulse sensor variables
int PulseSensorPurplePin = A0;        // Pulse Sensor PURPLE WIRE connected to ANALOG PIN 0
int Signal;                // holds the incoming raw data. Signal value can range from 0-1024
int Threshold = 600;            // Determine which Signal to "count as a beat", and which to ingore.
//define pulse_sensor function to read pulse sensor data
void pulse_sensor(){
  Signal = analogRead(PulseSensorPurplePin);  // Read the PulseSensor's value.
  if(Signal>=Threshold){
    Serial.print("HeartBeat Pulse==");
    Serial.println(Signal);
    char csbr[16];
    itoa(Signal,csbr,10);
    //publish pulse sensor data to ubidots cloud
    sprintf(payload,"%s","");
    sprintf(payload,"{\"%s\":",Variable_label2);
    sprintf(payload,"%s{\"value\":%s}}",payload,csbr);
    client.publish(topic, payload);
    delay(10);
}else{
  Serial.println("WAITING FOR PULSE!!!");
    Signal=0;
    char csbr[16];
    itoa(Signal,csbr,10);
    //publish pulse sensor data to ubidots cloud
    sprintf(payload,"%s","");
    sprintf(payload,"{\"%s\":",Variable_label2);
    sprintf(payload,"%s{\"value\":%s}}",payload,csbr);
    client.publish(topic, payload);
    delay(10);
}

}
//Set-Up WiFi connection
void setup_wifi() {
    delay(100);
    // We start by connecting to a WiFi network
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED)
    {
    delay(500);
    Serial.print(".");
    }
    randomSeed(micros());
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}
//Connect to mqtt
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected())
  {
    Serial.print("Attempting MQTT connection...");
    if (client.connect(MQTT_CLIENT_NAME,TOKEN,""))
    {
      Serial.println("connected");
      client.subscribe(topicSubscribe);
    } else {
      Serial.print("failed, rc=");
```

```
   Serial.print(client.state());

   Serial.println(" try again in 5 seconds");

   // Wait 6 seconds before retrying

   delay(10);

  }

 }

}

//Callback Function

void callback(char* topic, byte* payload, unsigned int length)

{

  Serial.print("MESSAGE ARRIVED: [");

  Serial.print("Subscribed to Topic: ");

  Serial.print(topic);

  Serial.println(" ]");

  for(int i=0;i<length;i++){

  Serial.print((char)payload[i]);}

  Serial.println();}


  void setup() {

  Serial.begin(9600);

  Serial.println("DHTxx test!");

  dht.begin();

  client.setServer(mqttbroker, 1883);

  reconnect();

  //sprintf(topicSubscribe,"/v1.6/devices/%s/%s/Lv",device_label,Variable_label_subscribe);
```

```
}
// define loop function to execute the user logic program
void loop() {
  if (!client.connected()) {
  reconnect();
  }
  client.setCallback(callback);
  client.subscribe(topicSubscribe);
  client.loop();
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);
  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(t);
  Serial.println(" *C ");
  Serial.print("Publish a message Temp:");
  int numt=t;
  char cstr[16];
  itoa(numt,cstr,10);
  Serial.println(cstr);
 Serial.println("publishing data:");
 //publish temperature data to ubidots broker
 sprintf(topic,"%s%s","/V1.6/devices/",device_label);

 sprintf(payload,"%s","");

 sprintf(payload,"{\"%s\":",Variable_label);

 sprintf(payload,"%s{\"value\":%s}}",payload,cstr);

 client.publish(topic, payload);

 pulse_sensor();

}
```

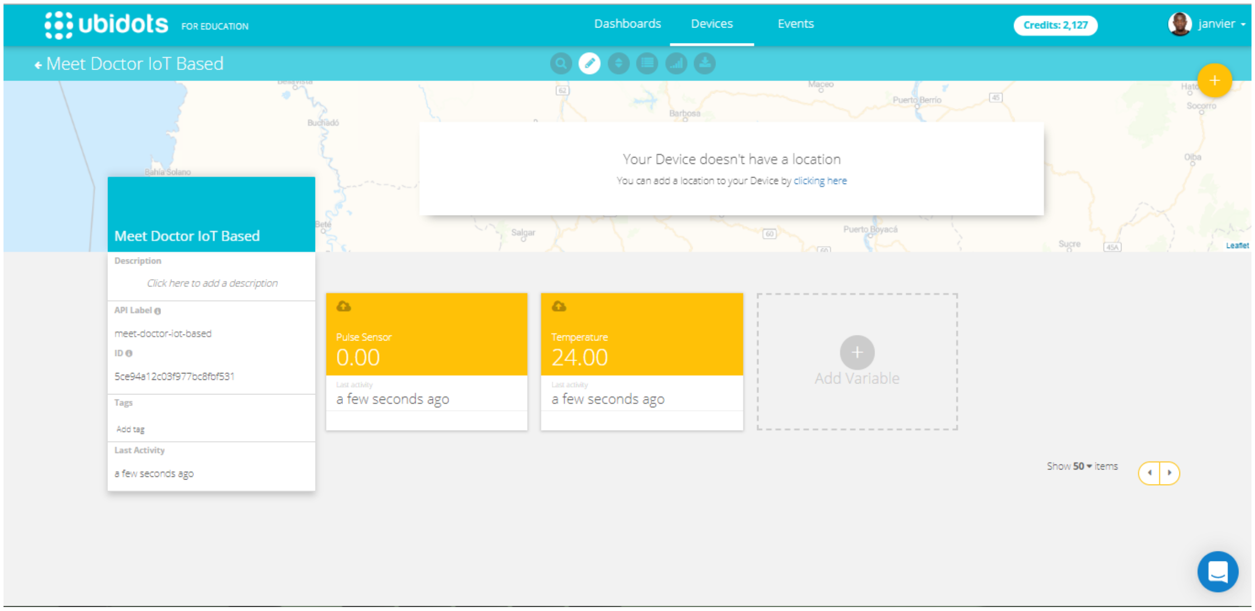
Serial monitor results



```
publishing data:
HeartBeat Pulse==1024
Humidity: 70.20 %       Temperature: 26.30 *C
Publish a message Temp:26
publishing data:
HeartBeat Pulse==1024
Humidity: 70.20 %       Temperature: 26.30 *C
Publish a message Temp:26
publishing data:
HeartBeat Pulse==1024
Humidity: 70.20 %       Temperature: 26.30 *C
Publish a message Temp:26
publishing data:
HeartBeat Pulse==1024
Humidity: 70.20 %       Temperature: 26.30 *C
Publish a message Temp:26
publishing data:
HeartBeat Pulse==888
Humidity: 70.20 %       Temperature: 26.30 *C
Publish a message Temp:26
publishing data:
HeartBeat Pulse==1013
Humidity: 70.20 %       Temperature: 26.30 *C
Publish a message Temp:26
publishing data:
WAITING FOR PULSE!!!
```

Steps for connecting Device to ubidots cloud

1) Step 1: registering the project and the sensor variables



Meet Doctor IoT Based

Your Device doesn't have a location

You can add a location to your Device by clicking here

Meet Doctor IoT Based

Description

Click here to add a description

API Label

meet-doctor-iot-based

ID

5ce94a12c03f977bc8fbf531

Tags

Add tag

Last Activity

a few seconds ago

Pulse Sensor 0.00

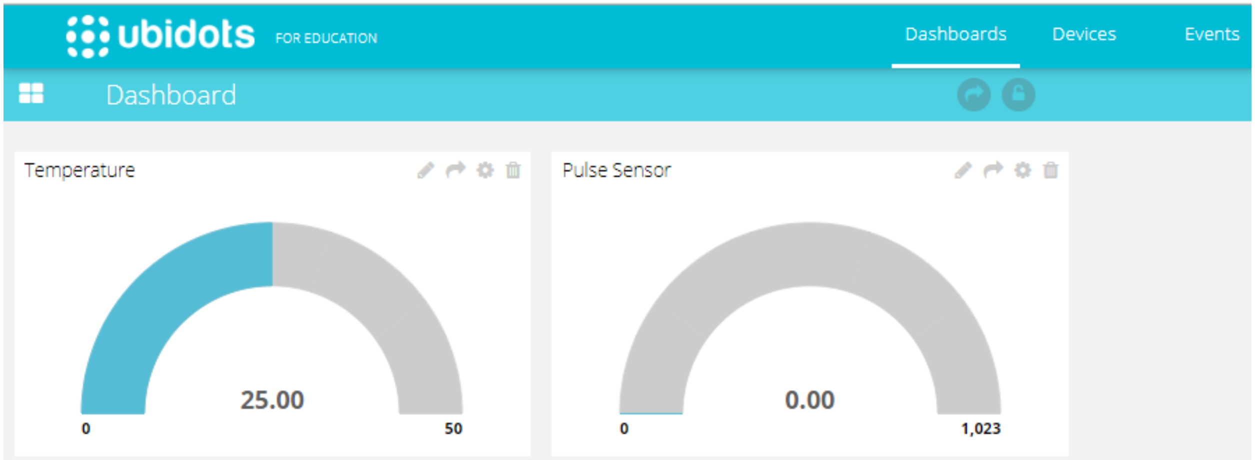
Last activity a few seconds ago

Temperature 24.00

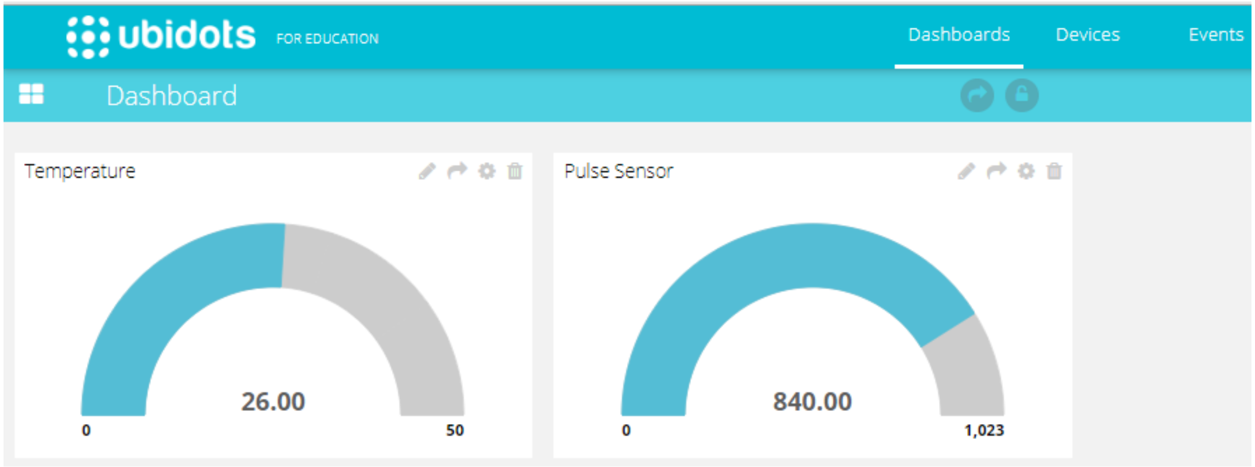
Last activity a few seconds ago

2) Step 2: creating dashboard to visualize data

i) Heartbeat Pulse not yet detected



ii) Heartbeat Pulse detected



For more information visit:

1. Youtube channel: IoT_Tech Kigali