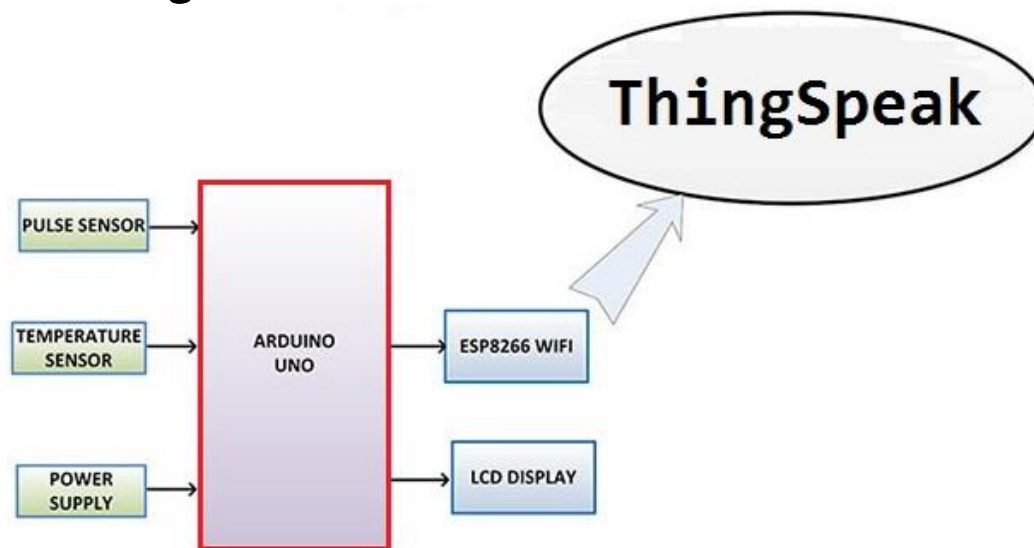


Patient Health Monitoring System using ESP8266 & Arduino

ABSTRACT:

To monitor various parameters of the patient using the internet of things. In the patient monitoring system based on the Internet of things project, the real-time parameters of a patient's health are sent to the cloud using Internet connectivity. These parameters are sent to a remote Internet location so that user can view these details from anywhere in the world

Block Diagram:

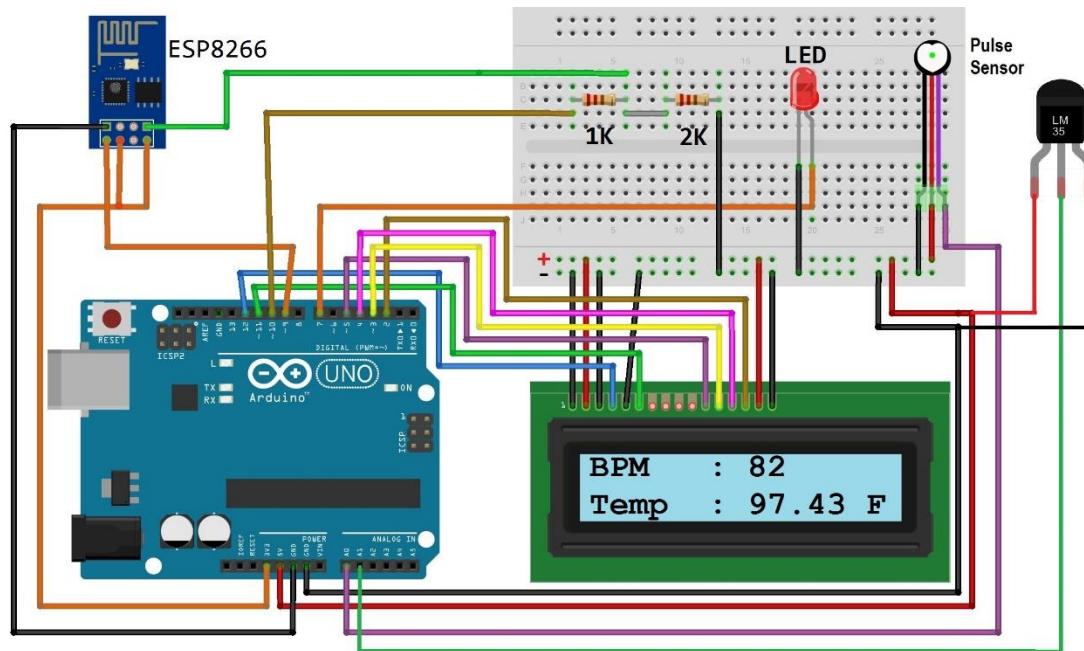


This is a simple block diagram that explains the **IoT Based Patient Health Monitoring System using ESP8266 & Arduino**. Pulse Sensor and LM35 Temperature Sensors measure BPM & Environmental Temperature respectively. The Arduino processes the code and displays it to 16*2 LCD Display. **ESP8266 Wi-Fi module** connects to Wi-Fi and sends the data to IoT device server. The IoT server used here is Thingspeak. Finally, the data can be monitored from any part of the world by logging into the Thingspeak channel.

S.N.	Components Name	Description	
1	Arduino Board	Arduino UNO/Nano or any other Board	1
2	ESP8266-01	ESP8266-01 <u>Wifi</u> Module	1
3	LCD Display	JHD162A 16X2 LCD Display	1
4	Potentiometer	10K	1
5	Pulse Sensor	Pulse Sensor from pulsesensor.com	1
6	Temperature Sensor	LM35 Analog Temperature Sensor	1
7	Resistor	2K	1
8	Resistor	1K	1
8	LED	5mm LED any Color	1
9	Connecting Wires	Jumper Wires	10
10	Breadboard	-	1

Circuit Diagram & Connections:

For designing IoT Based Patient Health Monitoring System using ESP8266 & Arduino, assemble the circuit as shown in the figure below.

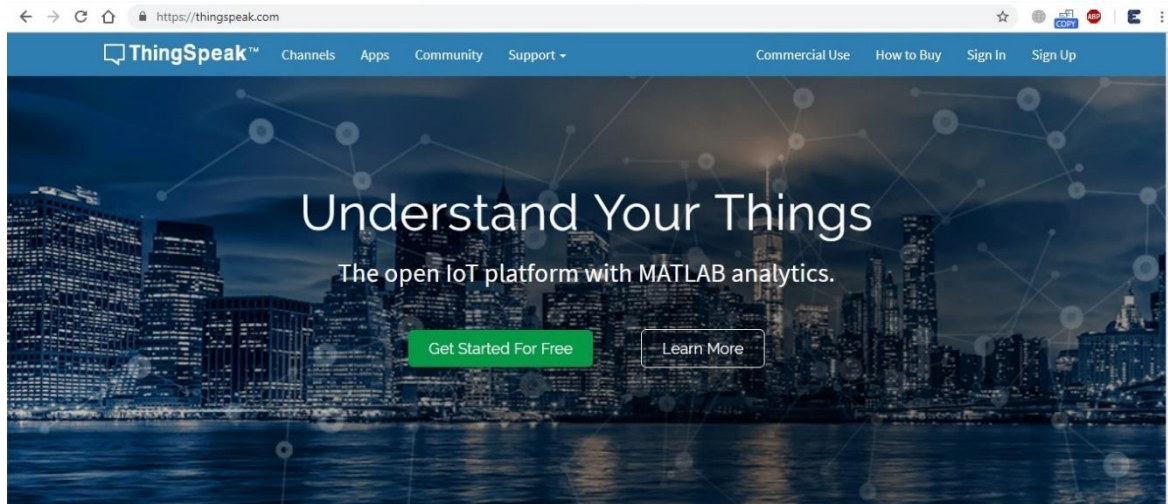


1. Connect Pulse Sensor output pin to A0 of Arduino and other two pins to VCC & GND.
2. Connect LM35 Temperature Sensor output pin to A1 of Arduino and other two pins to VCC & GND.
3. Connect the LED to Digital Pin 7 of Arduino via a 220-ohm resistor.
4. Connect Pin 1,3,5,16 of LCD to GND.
5. Connect Pin 2,15 of LCD to VCC.
6. Connect Pin 4,6,11,12,13,14 of LCD to Digital Pin12,11,5,4,3,2 of Arduino.
7. The RX pin of ESP8266 works on 3.3V and it will not communicate with the Arduino when we will connect it directly to the Arduino. So, we will have to make a voltage divider for it which will convert the 5V into 3.3V. This can be done by connecting the 2.2K & 1K resistor. Thus the RX pin of the ESP8266 is connected to pin 10 of Arduino through the resistors.
8. Connect the TX pin of the ESP8266 to pin 9 of the Arduino.

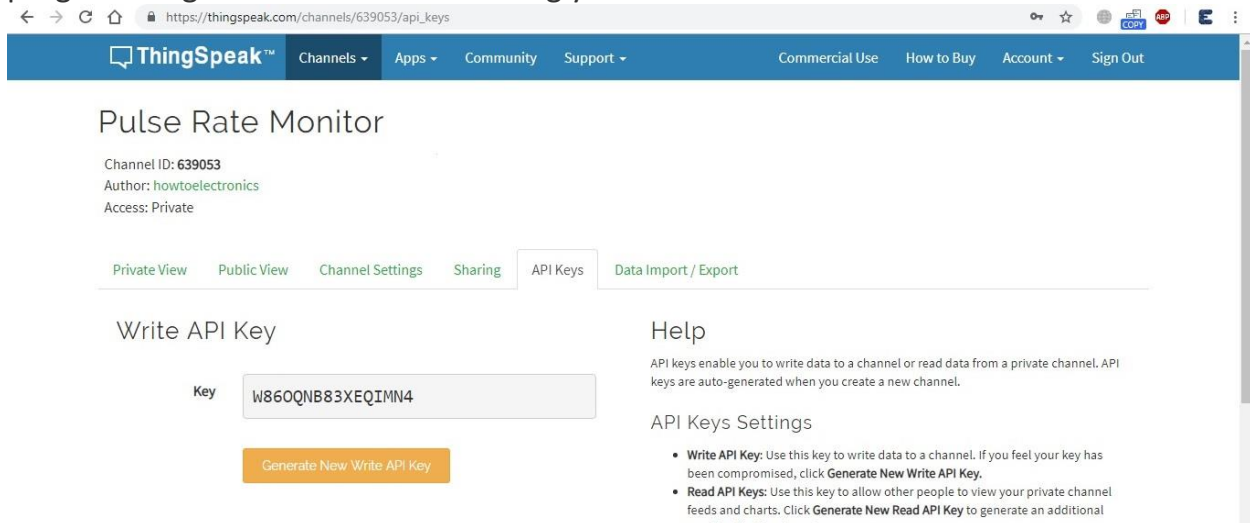
9. Setting the ThingSpeak:

ThingSpeak provides a very good tool for IoT based projects. By using the ThingSpeak site, we can monitor our data and control our system over the Internet, using the

Channels and web pages provided by ThingSpeak. So first you need to sign up for ThingSpeak. So visit <https://thingspeak.com> and create an account.



Then create a new channel and set up what you want. The tutorial in the video below. Follow the video for more clarification. Then create the **API keys**. This key is required for programming modifications and setting your data.



Then upload the code to the Arduino UNO by assembling the circuit shown above. Open the serial monitor and it will automatically connect to Wi-Fi and set up everything.

Now click on channels so that you can see the online data streaming, i.e IoT Based Patient Health Monitoring System using ESP8266 & Arduino as shown in the figure here.

OUTPUT:

