CS578: Internet of Things

Smart Home Monitoring Using ESP8266 and Webserver



Dr. Manas Khatua

Assistant Professor, Dept. of CSE, IIT Guwahati

E-mail: manaskhatua@iitg.ac.in , URL: http://manaskhatua.github.io/

"Try not to become a man of success. Rather become a man of value." - Albert Einstein

System Diagram





Physical Setup





ThingSpeak cloud server accessing from a Laptop/PC/Smartphone







Router Configuration To Connect with IITG Internet

Router Configuration





- This is Tenda WiFi Router
- ESP8266 (local server) will connect to this WiFi AP
- Sensor data will be uploaded to ThingSpeak server through this WiFi AP.
- Login Tenda WiFi using given IP (192.168.0.1) and user ID (admin) and password (admin)
- Do the following:
 - Tenda WiFi SSID and Password under "Wireless" tab
 - SSID: Tenda_8060A0; Password: 12345678
 - Time and Date settings under "Tools" tab
 - You can change admin password under "Tools" tab
 - ➢ Setup Internet Connection by Advanced → Internet Connection Setup
 - Set the Static IP, Subnet Mask, Default Gateway, DNS Server, Alternate DNS Server
 - Reboot the router from "Tools" tab



Tenda				2	KAL		
	Home	Advanced	Wireless	QoS	Applications	Security	Tools
Wireless Basic Settings	Wireless Basic Se	ettings				Help	
Wireless Security	Enable Wire	less 🗹				In this section	you can
Access Control	Primary S	SID Tenda_8	060A0			settings of the	router such
Wireless Connection Status	Secondary S	SID				as the SSID (na network) and	ame of the Broadcast
Wireless Extender	Network M	ode 11b/g/n	mixed mode	\$		Channel.	
	SSID Broade	cast 💿 Enable	e 🔿 Disable			of your wireles	e public name ss network. It enda XXXXXX
	AP Isolat	tion O Enable	e 💿 Disable			(where "XXXX)	X" represents
	Char	AutoSe	lect	\$		device MAC ac	aracters in Idress.) by
	Channel Bandwi	idth 🔿 20	• 20/40			default. Please better security	change it for . Note that
	Extension Char	Auto Se	lect	\$		this field shou blank.	id not be left
	WMM Capa	able 💿 Enable	e 🔿 Disable			SSID Broadca	st:This option
	APSD Capa	able 🔿 Enable	e 💿 Disable			network name	s (SSIDs)
			OK	əl		choose to disa SSID will be hi	ble it, the dden.







	Home Advar	Wireless	QoS	Applications	Security	Tools
Status	Internet Connection Se	tup			Help	
Internet Connection Setup	Internet Connection Type	Static IP	\$		Static IP:Static	P is a
MAC Clone	IP Address	172.16.88.50		1	connection type you to specify t	e that allows he Static IP
WAN Speed	Subnet Mask	255,255,255.0			information pro your ISP or that	vided by corresponds
LAN Settings		LUCILUCILUCIO			with your existi	ng
DNS Settings	Gateway	172.16.88.254			you have a fixe	d (or static
DHCP Server	DNS Server	172.17.1.1]	IP) address, you have provided y	ir ISP will ou with the
DHCP Client List	Alternate DNS Server	172.17.1.2		(Optional)	required inform Static IP option	ation. Select and type the
	мти	1500			IP Address, Sub	net Mask
		(The default value is 1 required by your ISP.)	500. Do not modif	y it unless	the correct box	es.
					Contact your IS you are not sur	P for help if e about



Tenda						
	Home Advanc	ced Wireless	QoS	Applications	Security	Tools
Status	WAN Status				Help	
Internet Connection Setup	Connection Status	Connected			Connection S	tatus:Refers
MAC Clone	Internet Connection Type	Static IP			to the connect the router and	tion between I the device
WAN Speed	WAN IP	10.11.10.34			connected to t WAN.	the router's
LAN Settings	Subnet Mask	255.255.192.0			Internet Conr	ection Type:
DNS Settings	Gateway	10.11.0.254			This can be se > Internet Cor	t in Advanced
DHCP Server	DNS Server	172.17.1.1			Setup. DHCP a	ind PPPoE are
DHCP Client List	Alternate DNS Server	172.17.1.2			Connection T	ime:Displays

Connecting with Internet



User Authentication Required
Use IITG Credentials to Login
Password
LOGIN Forgot Password ? Reset Here

• You should be able to access Internet in your Mobile/Laptop using Tenda WiFi AP



Cloud Server Configuration to Access Web Service

Configure to use Cloud Server





- We use ThingSpeak server http://www.thingspeak.com
- First create an user account
- Then create a channel on the ThingSpeak to upload the data



nttps://tningspeak.com/channels



Name						Created	Updated
🔒 Ten	Temperature & Humidity Monitoring						2019-07-09 06:44
Private	Public	Settings	Sharing	API Keys	Data Import / Export		
	nitoring	g Four s	ensors	in Star T	Topology	2019-07-09	2019-07-09 11:30
Private	Public	Settings	Sharing	API Keys	Data Import / Export		
	■ LED Control from Web						2019-07-12 06:53
Private	Public	Settings	Sharing	API Keys	Data Import / Export		

Click **New Channel** to create a new ThingSpeak channel.

Click on the column headers of the table to sort by the entries in that column or click on a tag to show channels with that tag.

Learn to create channels, explore and transform data.

Learn more about ThingSpeak Channels.

Examples

- Arduino
- Arduino MKR1000
- ESP8266
- Raspberry Pi
- Netduino Plus

Upgrade

Need to send more data faster?

Need to use ThingSpeak for a commercial proiect?



□ ThingSpeak™	Channels -	Apps 🗸	Community	Support	c ← Commercial Use How to Buy Account ← Sign Out							
Channel ID	814887				Channel Settings							
Name	DEMO 2				Channel Name: Enter a unique name for the ThingSpeak channel.							
Description	Getting differen	Getting different sensors data			 Description: Enter a description of the ThingSpeak channel. Field#: Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields. 							
Field 1	Temperature				Metadata: Enter information about channel data, including JSON, XML, or CSV data.							
			_		Tags: Enter keywords that identify the channel. Separate tags with commas.							
Field 2	Humidity				 Link to External Site: If you have a website that contains information about your ThingSpeak channel, specify the URL. 							
Field 3	LDR sensor				Show Channel Location:							
Field 4	Pulse rate		<		 Latitude: Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072. 							
Field 5	Vibration Senso	r			 Longitude: Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275. 							
Field 6					 Elevation: Specify the elevation position meters. For example, the elevation of the city of London is 35.052. 							
Field 7					 Video URL: If you have a YouTube[™] or Vimeo[®] video that displays your channel information, specify the full path of the video URL. 							
Field 8					 Link to GitHub: If you store your ThingSpeak code on GitHub[®], specify the GitHub repository URL. 							
Metadata					Using the Channel							
Tags					You can get data into a channel from a device, website, or another ThingsSpeak channel. You can then visualize data and transform it using ThingSpeak Apps .							
				11	See Tutorial: ThingSpeak and MATLAB for an example of measuring dew point from a							



https://thingspeak.com/channels

□ ThingSpeak ™	Channels 🗸	Apps 🗸	Community	Support 🗸		Commercial Use	How to Buy	Account -	Sign Out
New Channel Search by tag					Collect data in a ThingSpeak channel from a d from another channel, or from the web.				device,
Name				Created	Updated	channel.	.nannet to create	a new i ningspea	IK
▲ Temperature & H	lumidity Mor	nitoring		2019-07-09	2019-07-09 06:44	Click on the column headers of the table to sort entries in that column or click on a tag to show			ort by the w
Private Public Settings	Sharing API k	éys Data li	mport / Export			channels w	ith that tag.		
A Monitoring Four	sensors in St	ar Topol	ogy	2019-07-09	2019-07-09 11:30	data.	eate channets, e	xpiore and transi	form
Private Public Settings	Sharing API k	eys Data l	mport / Export			Learn more	about ThingSpe	ak Channels.	
LED Control from	n Web		1	2019-07-12	2019-07-12 06:53	Examp	oles		
Private Public Settings	Sharing API k	leys Data I	mport / Export			ArduinArduin	o MKR1000		

- ESP8266
- Raspberry Pi
- Netduino Plus

Upgrade

Need to send more data faster?

Need to use ThingSpeak for a commercial proiect?

Create Channel Display





Name	Temperature	
Field	Field 1 🛟	
Update Interval	15	second(s)
Units	degree Celsius	
Data Type	🔿 Integer 🛛 💿 Decimal	2 ‡ (# of places)

- Select **Private View** of the created channel.
- Click Add Widgets
- Select the Numeric Display widget, and then set the display options.

API Key and Channel ID





- To send data to ThingSpeak, we need unique write API key and Channel ID, which will be used later in code to upload the data to ThingSpeak website
- Click on "API Keys" button to get your unique "Write API Key"
- "Channel ID" is also given on the top



IoT Network Configuration

IoT Network Configuration



- There are total five ESP8266
 - one is acting as server,
 - other four as clients in local network.

- ESP1- ESP8266 acting as local server
- ESP2- ESP8266 with Light sensor
- ESP3- ESP8266 with Pulse sensor
- ESP4- ESP8266 with vibration sensor
- ESP5- ESP8266 with temperature & humidity sensor

• **Note**: Unique ID for each ESP will be needed in programming



Sensor Configuration



ESP8266 with LDR Sensor

- Connect VCC pin of LDR sensor
- Connect GND pin of LDR sensor
- Connect DATA OUT pin of LDR sensor with A0 pin of ESP2.





ESP8266 with Pulse Sensor

with 3V3 pin of ESP2

with GND pin of ESP2

- Connect VCC pin of pulse sensor with 3V3 pin of ESP3
- Connect GND pin of pulse sensor
 - Connect SIGNAL pin of pulse sensor with A0 pin of ESP3

with GND pin of ESP3

ESP8266 with Vibration Sensor

- Connect VCC pin of vibration sensor
- Connect GND pin of vibration sensor
- Connect DATA OUT pin of vibration sensor with A0 pin of ESP4

ESP8266 with Temperature & Humidity Sensor (DHT11) with VIN pin of ESP5

- Connect VCC pin of DHT11
- Connect DATA OUT pin of DHT11 with D3 pin of ESP5
- Connect GND pin of DHT11

Dr. Manas Khatua

with VIN pin of ESP4

with GND pin of ESP4

Breadboard

ESP8266



with GND pin of ESP5



21



Arduino Tool Configuration

Configure Arduino IDE



- Download and Install Arduino IDE <u>https://www.arduino.cc/en/Main/Software</u>
- When the Arduino IDE first opens, this is what you should see:

• • •	sketch_jul11a Arduino 1.8.9	
		Ø
sketch_jul11a		-
<pre>1 void setup(2 // put you 3 4 } 5 6 void loop() 7 // put you 8 9 }</pre>) { ur setup code here, to run once: { ur main code here, to run repeatedly:	
	0	
80 MHz, Flash, Disab	led, All SSL ciphers (most compatible), 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on /dev/cu.SLAB_USBtoUA	RT2

Install ESP8266 Board in IDE



- Go to File --> Preferences
- Enter the below URL into Additional Board Manager URLs field and press the "OK" button <u>http://arduino.esp8266.com/stable/package_esp8266com_index.json</u> OR <u>https://github.com/esp8266/Arduino/releases/download/2.3.0/package_esp8266com_index.json</u>

Preferences	×
Sketchbook location:	
C: \Users\Rui Santos\Documents\Arduino	Browse
Editor language: System Default \checkmark (requires restart of Arduino)	
Editor font size: 12	
Show verbose output during: compilation upload	
Compiler warnings: None 🗸	
Display line numbers	
Enable Code Folding	
Verify code after upload	
Use external editor	
Check for updates on startup	
Update sketch files to new extension on save (.pde -> .ino)	
Save when verifying or uploading	
Additional Boards Manager URLs: http://arduino.esp8266.com/stable/package_esp8266com_index.json	
More preferences can be edited directly in the file	
C:\Users\Rui Santos\AppData\Roaming\Arduino15\preferences.txt	
(edit only when Arduino is not running)	
ОК	Cancel



- Go to Tools > Board > Board Manager
- Scroll down, select the ESP8266 board menu and install "esp8266 by ESP8266 Community"

Edison. More info
AMEL-Tech Boards by ANEL Technology Boards included in this package: SmartEverything Fox. Online help More info
esp8266 by ESP8266 Community Boards included in this package: Generic ESP8266 Module, Olimex MOD-WIFI-ESP8266(-DEV), NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Adafruit HUZZAH ESP8266 (ESP-12), SweetPea ESP-210. Online help More info



	Tools Help		
	Auto Format Archive Sketch Fix Encoding & Reload Serial Monitor Serial Plotter	Ctrl+T Ctrl+Shift+M Ctrl+Shift+L	
	WiFi101 Firmware Updater		
	Board: "NodeMCU 1.0 (ESP-12E M CPU Frequency: "80 MHz" Flash Size: "4M (3M SPIFFS)" Upload Speed: "115200" Port Get Board Info Programmer: "AVRISP mkII" Burn Bootloader	fodule)" 3	LilyPad Arduino Arduino Pro or Pro Mini Arduino NG or older Arduino Robot Control Arduino Robot Motor Arduino Gemma ESP32 Arduino ESP32 Dev Module
			Electronic SweetPeas - ESP320 Nano32 WEMOS LoLin32 ESPea32 Noduino Quantum Node32s
CU 1.0 E	ESP-12E Mod	lule)	ESP8266 Modules Generic ESP8266 Module Generic ESP8285 Module ESPDuino (ESP-13 Module) Adafruit HUZZAH ESP8266 ESPresso Lite 1.0 ESPresso Lite 2.0 Phoenix 1.0 Phoenix 2.0
		_	NodeMCU 0.9 (ESP-12 Module)

- Select the appropriate board
 - Go to Tools >Board > NodeMCU 1.0 (ESP-12E Module)
- Finally, re-open the Arduino IDE

Install Sensor Libraries



• In this demo, we use DHT11 sensor for which we will be using DHT.h header file in the code. So, this header file should be installed.

• Install Using the Library Manager

- click to Sketch menu then Include Library > Manage Libraries
- Search for "DHT" on the Search box and install the DHT library from Adafruit.

• •	Library Manag	jer	
Type All 🗘 Topic	: All 😒 [онт	
EduIntro by Arduino LLC Library used for super-fast introduction basic components (led, button, piezo, LM3 Arduino during short workshops. More info	n workshops Is intended to be used w 15, thermistor, LDR, PIR, DHT11, and se	rith Arduino UNO / MICRO / MEGA / NANO / MKR an ervo) as a way to introduce people to the basic aspe Version 0.0.7 ᅌ I	nd a set of ects of nstall
DHT sensor library by Adafruit Version Arduino library for DHT11, DHT22, etc More info	1.3.4 INSTALLED Temp & Humidity Sensors Arduino lib	brary for DHT11, DHT22, etc Temp & Humidity Sens	sors
DHT sensor library for ESPx by beegee Arduino ESP library for DHT11, DHT22, correct field separator in keywords.txt. <u>More info</u>	_tokyo etc Temp & Humidity Sensors Optim	nized libray to match ESP32 requirements. Last char	nges: Use
Grove Temperature And Humidity Sen Arduino library to control Grove Temp	sor by Seeed Studio erature And Humidity Sensor. it con	tains chip DHT11 AM2302. This temperature & bu	midity
			Close



• After installing the DHT library from Adafruit, install "Adafruit Unified Sensor" libraries.

Adafruit TSL2561 by Adafruit Unified sensor driver for Adafruit's TSL2561 breakouts Unified sensor driver for Adafruit's TSL2561 breakouts More info					
	Version 1.0.3 ᅌ	Install			
Adafruit Unified Sensor by Adafruit Version 1.0.3 INSTALLED Required for all Adafruit Unified Sensor based libraries. A unified sensor abstraction layer used More info	l by many Adafruit sensor lit	vraries.			
		Close			

- There exist other methods for installing libraries
 - Importing a .zip Library
 - Sketch --> Include Library --> Add .Zip Library
 - Manual Installation of Library
 - Download the library as .Zip --> extract it
 - Place the files in File --> Preferences --> Sketchbook location
 - Restart Arduino IDE



MCU Programming

ESP8266 with Local Server



For ESP5, write the following code in the Arduino IDE and save as **Local_Server_ESP1.ino** Install **ThingSpeak.h** library. Change the **red colored text** in code according to your setup.

```
#include <ESP8266WiFi.h>//Including ESP8266 library#include<ESP8266WebServer.h>//Including ESP8266WebServer library for web server#include<ThingSpeak.h>//Including ThingSpeak libraryIPAddress IP(192,168,4,15);//Static IP address of local serverIPAddress gateway(192,168,4,1);//Gateway of the networkIPAddress mask(255, 255, 255, 0);//Subnet mask of the networkWiFiClient client;WiFiServer server(80);unsigned long myChannelNumber = 819306;//Replace with channelID of ThingSpeak channel ID
```

const char * myWriteAPIKey = "SW7ENB9IAXJT3STP"; //Replace WriteAPIKey of channel

```
const char* softAPssid = "ESP1_Server";
const char* password = "12345678";
```

const char* wifissid = "Tenda_8060A0"; const char* pass = "12345678";

//SSID of the hotspot of ESP8266 acting as local server
//Password of the hotspot of ESP8266 acting as local server

//Replace with SSID of WIFI router providing internet access
//Password of WIFI router providing internet access



void setup() {

```
WiFi.mode(WIFI AP STA);
                                        //station mode and access point mode both at the same time
Serial.begin(9600);
                                        //Serial communication at baud rate of 9600 for debugging purpose
delay(100);
Serial.println(WiFi.getMode());
Serial.print("Configuring SoftAP....");
Serial.println(WiFi.softAPConfig(IP, gateway, mask)? "Ready" : "Failed");
delay(10);
Serial.println("Setting SoftAP...");
Serial.println(WiFi.softAP(softAPssid, password));
delay(10);
                                                                         •
Serial.println(WiFi.softAPIP());
delay(500);
WiFi.begin(wifissid, pass);
while(WiFi.status()!=WL CONNECTED) {
 Serial.print(".");
 delay(500);
Serial.print("Connected to Wifi with ssid ");
Serial.println(wifissid);
Serial.print("WiFi IP address: ");
Serial.println(WiFi.localIP());
                                        // WIFI router IP address
ThingSpeak.begin(client);
server.begin();
                                        //Start local server
```

- Two functions exist in the programme: setup () and loop ()
 - **setup():** This function runs once when FSP first boots
 - **loop():** This function reads the LDR sensor value and connects to local server then send sends data to local server



```
void loop() {
   Serial.printf("Stations connected = %d\n", WiFi.softAPgetStationNum());
   WiFiClient client = server.available(); //Waiting for the incoming data if client is ready to send
   if (!client) {return;}
   String select_fun = client.readStringUntil('\r'); //Reads the ESP8266 ID (of clients)
   if(select_fun=="5") { //If ESP5 sends the data
    String temp = client.readStringUntil('\r'); //Reads the temperature value
    String Humidity = client.readStringUntil('\r'); //Reads the humidity value
```

//Upload the temp value to ThingSpeak server as first field of channel

ThingSpeak.writeField(myChannelNumber, **1**, temp, myWriteAPIKey); delay(15000); //Wait for 15 sec after one entry //Upload the humidity value to ThingSpeak server as second field of channel

```
ThingSpeak.writeField(myChannelNumber, 2, Humidity, myWriteAPIKey);
Serial.print("Temperature: ");
Serial.print(temp);
Serial.print(" degree celsius, Humidity: ");
Serial.print(Humidity);
Serial.print("%. ");
Serial.println("Sent to ThingSpeak Server...");
```



```
//If ESP2 sends the data
if(select fun=="2") {
    String LDRval = client.readStringUntil('r');
                                                       //Reads light sensor value
                           //Upload the light sensor value to ThingSpeak server as third field of channel
    ThingSpeak.writeField(myChannelNumber, 3, LDRval, myWriteAPIKey);
    Serial.print("LDR sensor data value: ");
    Serial.println(LDRval);
    Serial.println("Sent to ThingSpeak Server...");
if(select fun=="3") {
                                                       //If ESP3 sends the data
    String pulseRate = client.readStringUntil('\r');
                                                       //Reads pulse rate
                            //Upload the pulse rate to ThingSpeak server as fourth field of channel
    ThingSpeak.writeField(myChannelNumber, 4, pulseRate, myWriteAPIKey);
    Serial.print("Pulse rate: ");
    Serial.print(pulseRate);
    Serial.println(" BPM. Sent to ThingSpeak Server..");
  if(select fun=="4"){
                                                       //If ESP4 sends the data
    String Vibval = client.readStringUntil('r');
                                                      //Reads vibration sensor data
                           //Upload the vibration sensor data value to ThingSpeak server as fifth field of channel
    ThingSpeak.writeField(myChannelNumber, 5, Vibval, myWriteAPIKey);
    Serial.print("Vibration Sensor data: ");
    Serial.print(Vibval);
    Serial.println(" Sent to ThingSpeak server..");
  delay(15000);
                           //waits for 15 secs after each transmission
```

ESP8266 with LDR Sensor



For ESP2, write the following code in the Arduino IDE and save as LDR_client.ino

```
#include<ESP8266WiFi.h> // Including ESP8266 library
char ssid[]="ESP1_Server"; //Network ssid of hotspot of local server
char pass[]="12345678"; //Password of hotspot of local server
int val;
int LDRpin = A0; //LDR Pin Connected to A0 pin
IPAddress server(192,168,4,15); // IP address of local server
WiFiClient client;
```

- Change the IP address of Local Server (i.e. ESP1)
- Change the SSID and Password of WiFi AP hosted in Local Server
- Two functions exist in the programme: setup () and loop ()
 - **setup():** This function runs once when ESP first boots
 - **loop():** This function reads the LDR sensor value and connects to local server then send sends data to local server



void setup()

{

```
Serial.begin(9600);
                                  // Serial communication at baud rate of 9600 for debugging purpose
delay(10);
WiFi.mode(WIFI STA);
                                  // ESP8266 in station mode
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, pass);
Serial.println();
while (WiFi.status() != WL CONNECTED)
  Serial.print(".");
  delay(500);
Serial.println();
Serial.println("WiFi connected");
Serial.print("LocalIP:"); Serial.println(WiFi.localIP());
Serial.println("MAC:" + WiFi.macAddress());
Serial.print("Gateway:"); Serial.println(WiFi.gatewayIP());
Serial.print("AP MAC:"); Serial.println(WiFi.BSSIDstr());
```

// MAC address of access point

}



void loop()

```
val = analogRead(LDRpin);
if(client.connect(server,80))
{
    client.print("2\r");
    Serial.print("LDR sensor value: ");
    Serial.println(val);
    String LDRval = String(val);
    LDRval += "\r";
    client.print(LDRval);
    Serial.println("Sent to Local Server..");
    delay(15000);
  }
  client.stop();
}
```

// Reads the light sensor value
// Connect to local server

// before sending data first send ESP8266 ID as 2

// Add end delimiter
// Send to local server

ESP8266 with Pulse Sensor



```
#define pulsePin A0
                                  // Pulse sensor input pin A0
#include<ESP8266WiFi.h>
                                  // Including ESP8266 library
char ssid[] = "ESP1 Server";
                                  // Replace with SSID of hotspot of local server
char pass[] = "12345678";
                                  // Replace with password of hotspot of local server
 IPAddress server(192,168,4,15); // IP address of local server
                                                                               For ESP3, write the
WiFiClient client:
                                                                               following code in the
int rate[10];
                                  // array to hold last ten IBI value
                                                                               Arduino IDE and save as
unsigned long sampleCounter = 0; // used to determine pulse timing
                                                                               Pulse client.ino
unsigned long lastBeatTime = 0; // used to find IBI
unsigned long lastTime = 0, N;
int BPM = 0;
                       // int that holds raw analog in 0. updated every 2mS
                      // int that holds time interval between beats! Must be seeded!
int IBI = 0;
                      // used to find peak in pulse wave, seeded
int P = 512;
int T = 512;
                      // used to find trough in pulse wave, seeded
int thresh = 512:
                      // used to find instant moment of heart beat, seeded
int amp = 100;
                      // used to hold amplitude of pulse waveform, seeded
int Signal;
                      // holds incoming raw data
boolean Pulse = false:
                                  // "True" when heartbeat is detected. "False" when not a "live beat".
boolean firstBeat = true;
                                  // used to seed rate array so we startup with reasonable BPM
                                  // used to seed rate array so we startup with reasonable BPM
boolean secondBeat = true;
boolean QS = false;
                                  // Becomes true when ESP8266 finds a beat
```



void setup()

```
Serial.begin(9600);
                                   // Serial communication at baud rate of 9600 for debugging purpose
delay(10);
WiFi.mode(WIFI STA);
                                   // ESP8266 in station mode
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, pass);
Serial.println();
while (WiFi.status() != WL CONNECTED)
 Serial.print(".");
 delay(500);
Serial.println();
Serial.println("WiFi connected");
Serial.print("LocalIP:"); Serial.println(WiFi.localIP());
Serial.println("MAC:" + WiFi.macAddress());
Serial.print("Gateway:"); Serial.println(WiFi.gatewayIP());
Serial.print("AP MAC:"); Serial.println(WiFi.BSSIDstr());
                                                           // MAC address of access point
```



void loop(){

```
if (QS == true){
   if (client.connect(server, 80)){
    client.print("3\r");
    String pulseRate = String(BPM);
    pulseRate +="\r";
    Serial.print("Pulse rate: ");
    Serial.print(BPM);
    Serial.println(" BPM.");
    client.print(pulseRate);
    Serial.println("Sent to local server..");
   QS = false;
   client.stop();
   delay(15000);
 else if(millis() >= (lastTime + 2)) {
   readPulse();
   lastTime = millis();
```

//if ESP8266 finds a beat
// Connect to local server
// before sending data first send ESP8266 ID as 3
// Convert into string
// Add "r" as end delimiter

// send data to local server



```
void readPulse() {
 Signal = analogRead(pulsePin);
                                         //Read pulse sensor value
                                         // Keeps track of the time in mS
 sampleCounter += 2;
 int N = sampleCounter - lastBeatTime; // Monitor the time since the last beat to avoid noise
                                         // find the peak and trough of the pulse wave
 detectSetHighLow();
                                          // Now it's time to look for the heart beat
                                          // signal surges up in value every time there is a pulse
 if(N > 250){
                                          // avoid high frequency noise
   if((Signal > thresh) \&\& (Pulse == false) \&\& (N > (IBI/5)*3))
    pulseDetected();
 if (Signal < thresh && Pulse == true) {
   Pulse = false:
   amp = P - T;
                                                      void detectSetHighLow() {
   thresh = amp / 2 + T;
                                                        if (Signal < thresh && N > (IBI/5)^* 3)
   P = thresh;
                                                                    // avoid dichrotic noise by waiting 3/5 of last IBI
   T = thresh;
                                                          if (Signal < T) {
                                                                                  // T is the trough
 if (N > 2500) {
                                                            T = Signal;
                                                                                 // Keep track of lowest point in pulse wave
   thresh = 512;
   P = 512;
   T = 512;
                                                        if (Signal > thresh && Signal > P) // thresh condition helps avoid noise
   lastBeatTime = sampleCounter;
   firstBeat = true;
                                                           P = Signal;
                                                                                  // P is the peak
   secondBeat = true;
                                                                                  // Keep track of highest point in pulse wave
```



void pulseDetected()

```
// set the pulse flag when there is a pulse
Pulse = true;
IBI = sampleCounter - lastBeatTime; // time between beats in mS
lastBeatTime = sampleCounter; //keep track of time for next pulse
                         // if it's the first time beat is found
if (firstBeat)
                                                                       BPM = 60000 / runningTotal;
                                                                      // how many beats can fit into a minute? that's BPM!
                         //clear firstBeat flag
 firstBeat = false:
                                                                      QS = true;
 return;
                                                                      if (client.connect(server, 80)) //Connects to local server
if (secondBeat)
                         // if this is second beat
                                                                        client.print("3\r");
                                                                                 //before sending the data sends ESP8266 ID as 3
  secondBeat = false; // clear secondBeat flag
                                                                        String pulseRate = String(BPM);
  for (int i = 0; i <= 9; i++)
                                                                                 // Converting integer data into string
                                                                        pulseRate +="\r";
   rate[i] = IBI;
                                                                                 // Add end Delimiter "r" in the data
                                                                        Serial.print("Pulse rate: ");
                                                                        Serial.print(BPM);
word runningTotal = 0; // clear the runningTotal variable
                                                                        Serial.println(" BPM.");
for (int i = 0; i <= 8; i++) //Shift data in the rate array
                                                                        client.print(pulseRate);
                                                                                                    //sends data to locals server
                                                                        Serial.println("Sent to local server..");
 rate[i] = rate[i + 1]; // and drop the oldest IBI value
 runningTotal += rate[i]; // add up the 9 oldest IBI value
                                                                      client.stop();
                                                                      delay(15000);
rate[9] = IBI;
                      // add the latest IBI to the rate array
                                                                                  // Wait for 15 seconds after each transmission
runningTotal += rate[9]; //add the latest IBI to runningTotal
                                                                    }
runningTotal /= 10;
                          // average the last 10 IBI values
```

ESP8266 with Vibration Sensor



For **ESP4**, write the following code in the Arduino IDE and save as **Vibration_client.ino**

<pre>#include <esp8266wifi.h></esp8266wifi.h></pre>	
#define vib A0	

```
char ssid[] = "ESP1 Server";
char pass[] = "12345678";
```

IPAddress server(192,168,4,15); // IP address of local server WiFiClient client;

// Including ESP8266 library // sensor input from A0 pin of ESP8266

//Replace with SSID of hotspot of local server // Replace with password of hotspot of local server

- Change the IP address of Local Server (i.e. ESP1) ٠
- Change the SSID and Password of WiFi AP hosted in Local Server



void setup(){

```
// Serial communication at baud rate of 9600 for debugging purpose
Serial.begin(9600);
delay(10);
                                 // Input of vibration sensor
pinMode(vib, INPUT);
WiFi.mode(WIFI STA);
                                 // ESP8266 as station mode
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, pass);
Serial.println();
while (WiFi.status() != WL CONNECTED) {
  Serial.print(".");
  delay(500);
Serial.println();
Serial.println("WiFi connected");
Serial.print("LocalIP:"); Serial.println(WiFi.localIP());
                                                         // IP address of local server
Serial.println("MAC:" + WiFi.macAddress());
Serial.print("Gateway:"); Serial.println(WiFi.gatewayIP());
Serial.print("AP MAC:"); Serial.println(WiFi.BSSIDstr()); // MAC address of access point
```

}



```
void loop(){
  int val = analogRead(vib);
                                    // Reads the sensor value
  if(client.connect(server,80))
                                    //connects to local server
   client.print("4\r");
                                     // Before sending the data sends ESP8266 ID as 4
   Serial.print("Vibration sensor value: ");
   Serial.println(val);
   String data = String(val);
                                    // Converting integer data into string type
   data += "\r";
                                    // Add end delimiter "r" in the data
   client.print(data);
                                    // sends sensor data to local server
   Serial.println("Sent to Local server..!!");
   delay(15000);
                                                 // After each transmission wait for 15 seconds
   client.stop();
```

ESP8266 with DHT11 Sensor



For ESP5, write the following code in the Arduino IDE and save as Temp_Humidity_Client.ino

#include <dht.h> #include<esp8266wifi.h> #define DHTPIN 0</esp8266wifi.h></dht.h>	//Including temperature and Humidity sensor library //Including ESP8266 library // D3 pin of ESP8266
char ssid[] = "ESP1_Server"; char pass[] = "12345678";	<pre>//Replace with ssid of hotspot of local server // Replace with password of hotspot of local server</pre>
IPAddress server(<mark>192,168,4,15</mark>); WiFiClient client;	// Static IP address of local server. Replace whatever you want.
DHT dht(DHTPIN, DHT11);	// Data of DHT11 sensor in D3 pin of ESP8266

- Change the IP address of Local Server (i.e. ESP1)
- Change the SSID and Password of WiFi AP hosted in Local Server
- Install the DHT11 library and Adafruit Unified Sensor library for DHT11 sensor



void setup() {

```
//serial communication at baud rate of 9600 for debugging purpose
Serial.begin(9600);
delay(10);
                                 // start Temperature and Humidity sensor
dht.begin();
                                 // ESP8266 mode as station mode
WiFi.mode(WIFI STA);
Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, pass);
Serial.println();
while (WiFi.status() != WL CONNECTED) {
 Serial.print(".");
 delay(500);
Serial.println();
Serial.println("WiFi connected");
Serial.print("LocalIP:"); Serial.println(WiFi.localIP());
Serial.println("MAC:" + WiFi.macAddress());
Serial.print("Gateway:"); Serial.println(WiFi.gatewayIP());
Serial.print("AP MAC:"); Serial.println(WiFi.BSSIDstr()); // MAC address of access point
```

}



```
void loop() {
  float h = dht.readHumidity();
                                          // Read Humidity value from sensor
  float t = dht.readTemperature();
                                          // Read temp value from sensor
  if(isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor");
                                                                      // Error message
    return;
  if(client.connect(server,80))
                                          // Connect to local server
    client.print("5\r");
                                          // before sending the data first send ESP8266 ID as 5
    String temp = String(t);
    temp += "r";
                                          // Add "r" as end delimiter
    client.print(temp);
                                          // send temperature to local server
    Serial.print("Temperature: ");
    Serial.print(t);
    Serial.print(" degree celsius, Humidity: ");
    Serial.print(h);
    Serial.print("%. ");
    String humidity = String(h);
    humidity += "\r";
                                          // Add "r" in data as end delimiter
    client.print(humidity);
                                          // send to Local server
    Serial.println("Sent to local server ");
    delay(15000);
                                          // delay of 15sec after each transmission
  client.stop();
```



Code Compilation and Upload

Code Compilation



temp_client | Arduino 1.8.9 **†** Ø op client 1 #include<DHT.h> //Including temperature and Humidity sensor library 2 #include<ESP8266 library Compile Button 3 4 char sstal = ESPOZOD; //Replace with ssid of hotspot of local server 5 char pass = "12345678"; // Replace with password of hotspot of local server 6 7 IPAddress server(192,168,4,15); // IP address of local server 8 WiFiClient client; 9 // D3 pin of ESP8266 10 #define DHTPIN 0 11 DHT dht(DHTPIN, DHT11); // Data of DHT11 sensor in D3 pin of ESP8266 12 13 void setup(){ Serial.begin(9600); //serial communication at baud rate of 9600 for debugging purpos 14 15 delay(10); dht.begin(); // start Temperature and Humidity sensor 16 WiFi.mode(WIFI_STA); // ESP8266 mode as station mode 17 Serial.print("Connecting to "); 18 19 Serial.println(ssid); 20 WiFi.begin(ssid,pass); Serial println(); 21 while (WiFi state () I- WI CONNECTED) 22 Done compiling. Sketch uses 276220 bytes (26%) of program storage space. Maximum is 1044464 bytes. Global variables use 27012 bytes (32%) of dynamic memory, leaving 54908 bytes for local va cfpShers (most compatible), 4M (no SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on /dev/cu.SLAB_USBtoUART2

Compilation successful message in bottom left corner.

Code Uploading



- Plug in the ESP8266 boards one by one to PC/Laptop via USB cable
- Go to Tool menu, select Board "NodeMCU 1.0 (ESP-12E Module)" and Port "COM3".
- Open the corresponding code and do uploading code in Node MCU.

Note: If COM port is not detected automatically then it is needed to install. Download port drivers from the given link and then install and then restart the system:

https://www.silabs.com/pro ducts/developmenttools/software/usb-to-uartbridge-vcp-drivers





Observe Outputs

Open Serial Monitor





				Send	
Serial Monitor of Local Server	14:39:43.602 -> Stations connected = 4 14:39:44.864 -> Vibration Sensor data: 29 Sent to ThingSpeak server 14:39:59.873 -> Stations connected = 4 14:39:59.907 -> Stations connected = 4 14:39:59.907 -> Stations connected = 4 14:39:59.945 -> Stations connected = 4 14:40:17.586 -> Temperature: 23.30 degree celcius, Humidity: 70.00%. Sent to ThingSpeak Server 14:40:32.630 -> Stations connected = 4 14:40:32.630 -> Stations connected = 4 14:40:32.665 -> Stations connected = 4 14:40:32.665 -> Stations connected = 4 14:40:32.702 -> Stations connected = 4 14:40:32.770 -> Stations connected = 4 14:40:32.4148 -> LDR sensor data value: 1024 14:40:34.148 -> Sent to ThingSpeak Server	9600 baud	CI	ear output	

۲







•••		/dev/cu.SLAB_USB	toUART8						
						Send			
<pre>{ld0 010 0 \$0 b 00 0 0{00 b00nn0\$nn000 " p00 14:19:56.258 -> 14:20:02.041 -> WiFi connected 14:20:02.074 -> LocalIP:192.168.4.115 14:20:02.074 -> LocalIP:192.168.4.115 14:20:02.109 -> Gateway:192.168.4.1 14:20:02.142 -> AP MAC:3E:71:BF:32:6E:AD 14:20:02.175 -> Vibration sensor value: 29 14:20:02.175 -> Sent to Local server!! 14:20:17.122 -> Sent to Local server!! 14:20:32.108 -> Vibration sensor value: 30 14:20:32.108 -> Vibration sensor value: 1013 14:20:32.142 -> Sent to Local server!! 14:20:47.104 -> Vibration sensor value: 30</pre>	с\$ {lpOnD	0100 #n01100	p 00nN0 100d`0 oo\$	o{000N #0\$r00n #	¥ 0 \$0 0 1\$0	00n0 00		Serial I of ESP4	Monitor 4
Autoscroll Show timestamp			Newline	9600 baud	•	Clear output			
Serial Monitor of ESP5	■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■	<pre>\$0 01 00n0 0 0=r0 9 -> 9 -> WiFi connec 3 -> LocalIP:192 3 -> MAC:3C:71:Bi 6 -> Gateway:192 1 -> AP MAC:3E:7 4 -> Temperature 6 -> Temperature 1 -> Temperature 9 -> Temperature 1 -> Temperature 2 -> Temperature 2 -> Temperature</pre>	DConnecting to ESP8 ted .168.4.116 F:32:70:77 .168.4.1 I:BF:32:6E:AD : 24.00 degree celci : 24.10 degree celci : 25.00 degree celci : 26.80 degree celci : 27.70 degree celci : 28.20 degree celci	/dev/cu.S 266 266 us, Humidity: 68. us, Humidity: 95. us, Humidity: 90. us, Humidity: 76. us, Humidity: 75.	UAB_USBt 00%. Sent 00%. Sent 00%. Sent 00%. Sent 00%. Sent	to local set to local set to local set to local set to local set to local set to local set	rver rver rver rver rver	9600 baud	Send

Results & Graphs in Web



- Open the ThingSpeak page and click on **Channels > My channels**
- Now select the channel that is created for this experiment (In this case 'Monitoring Four Sensors in Star Topology').

 nttps://tningspeak.com/c 	nanneis										
□ , ThingSpeak™	Channels -	Apps 🗸	Community	Support 🗸			Commercial Use	How to Buy	Account -	Sign Ou	
My Channel	S						Help				
New Channel	Sea	rch by tag				Q	Collect data from anoth	a in a ThingSpeak er channel, or froi	channel from a o m the web.	device,	
Name	Name				Updated		Click New (channel.	ak			
▲ Temperature & I	Temperature & Humidity Monitoring		2019-07-09	2019-07-09 06:44		Click on the column headers of the table to sort by the entries in that column or click on a tag to show					
Private Public Settings	Sharing API	Keys Data	Import / Export				channels w	with that tag.			
Monitoring Four	sensors in S	tar Topol	.ogy	2019-07-09	2019-07-09 1	1:30	data.	cate channets, e	xptore and trans	ionni	
Private Public Settings	Sharing API	Keys Data	Import / Export				Learn more	about ThingSpe	ak Channels.		
LED Control fror	n Web			2019-07-12	2019-07-12 0	6:53	Examp	oles			
Private Public Settings	Sharing API	Keys Data	Import / Export				• Arduir • Arduir	io io MKR1000			

- ESP8266
- Raspberry Pi
- Netduino Plus

Upgrade

Need to send more data faster?

Need to use ThingSpeak for a commercial project?



• click on 'Private View' to see the uploaded data

Э https://thingspeak.com/channels/819306/private_show **□** ThingSpeak[™] Community Channels -Apps -Support -Commercial Use How to Buy Account -Sign Out Private View Public View Channel Settings Sharing API Keys Data Import / Export MATLAB Visualization MATLAB Analysis Add Visualizations Add Widgets Export recent data Channel Stats Created: 28 days ago Last entry: less than a minute ago Entries: 77 Field 1 Chart 🖸 🗘 🖉 🗙 🖸 👂 🖋 🗙 Temperature **Temperature Sensor Data** 28 Temperature 28.20 26 Degree 15. Jul 22. Jul 29. Jul 5. Aug Date a few seconds ago ThingSpeak.com



• Temperature and Humidity











Light Sensor

Pulse Sensor

Vibration Sensor







Thanks!

