

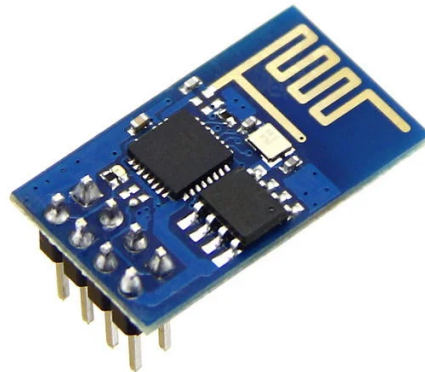


PIC18F4550 Interface with WiFi ESP8266 module

Overview of ESP8266

The ESP8266 module is low-cost standalone wireless transceiver that can be used for end-point IoT developments.

ESP8266 module enables internet connectivity to embedded applications. It uses TCP/UDP communication protocol to connect with the server/client.



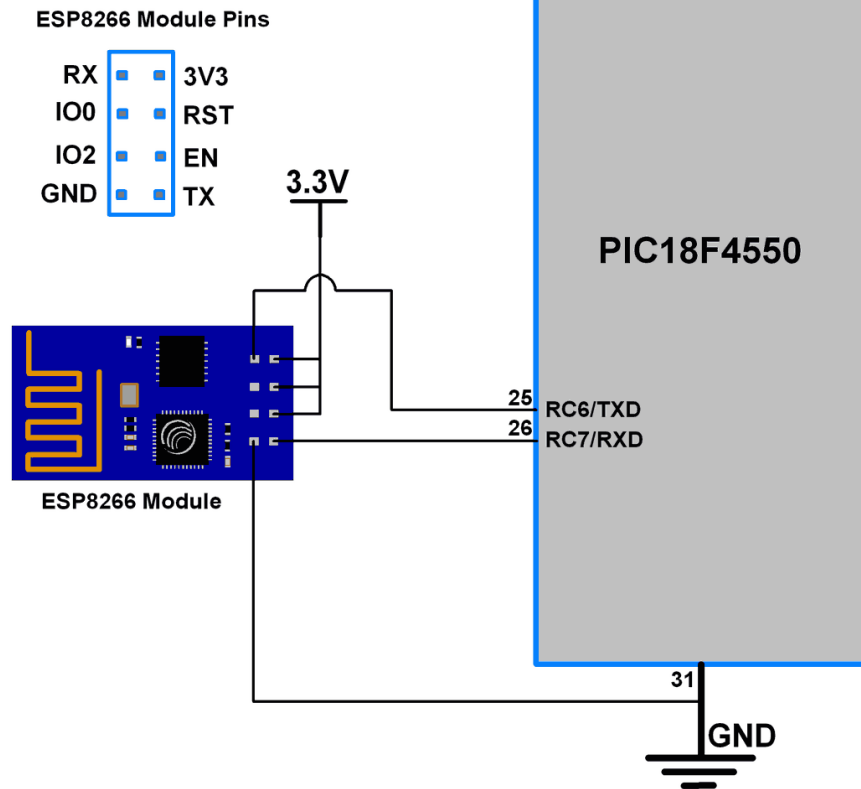
ESP8266-01 Wi-Fi Module

To communicate with the ESP8266 module, the microcontroller needs to use a set of AT commands. The microcontroller communicates with the ESP8266-01 module using UART having a specified Baud rate (Default 115200).

To know more about the ESP8266 Module and its firmware refer to **ESP8266 Module** (<https://www.electronicwings.com/sensors-modules/esp8266-wifi-module>)

Now let's interface the ESP8266 Module with PIC18F4550

Connection Diagram of ESP8266 with PIC18F4550

[Platforms \(/explore\)](#)[Projects \(/projects\)](#)[Contests \(/contests\)](#)

PIC18F4550 Interface with ESP8266 Module

TCP Client using ESP8266

Let's program PIC18F4550 to configure the ESP8266 module as TCP Client and Receive/Send data from/to Server using WIFI.

Here, we are using the Thingspeak server for TCP Client demo purposes.

Thingspeak is an open IOT platform where anyone can visualize and analyze live data from their sensor devices. Also, we can perform data analysis on data posted by remote devices with Matlab code in Thingspeak. To learn more about Thingspeak refer link

https://thingspeak.com/pages/learn_more

(https://thingspeak.com/pages/learn_more)

Just sign up and create a channel. We have below the channel and write key on Thingspeak for data send and receive.

- **channel ID** is = 119922
- **Write Key** is = C7JFHZY54GLCJY38

Note: Do not forget to tick the **Make Public** field in the **channel setting** option on your Thingspeak channel. It makes channels available to use as public. This allows any user to access channel data without any username & password.

For TCP RECEIVE method use below AT command steps shown in the screenshot of RealTerm Serial Terminal.

+ Project (/publish/project)



Use below screenshot consists of AT commands (Green) and Responses (Yellow).

```

RealTerm: Serial Capture Program 2.0.0.70

ATE0
OK
AT+CWMODE=3
OK
AT+CIPMUX=0
OK
AT+CIPMODE=0
OK
AT+CWJAP="EW-WIFI","mh163687"
WIFI DISCONNECT
WIFI CONNECTED
WIFI GOT IP
OK
AT+CIPSTART="TCP","api.thingspeak.com",80
CONNECT
OK
AT+CIPSEND=37
OK
> GET /channels/11922/feeds/last.txt
Recv 37 bytes
SEND OK
+IPD,66:<"created_at": "2017-07-07T13:36:33Z", "entry_id": 1117, "field1": "1">CLOSED
  
```

For the TCP SEND method use below AT command steps shown in the screenshot of RealTerm Serial Terminal.

```

RealTerm: Serial Capture Program 2.0.0.70

ATE0
OK
AT+CWMODE=3
OK
AT+CIPMUX=0
OK
AT+CIPMODE=0
OK
AT+CWJAP="EW-WIFI","mh163687"
WIFI DISCONNECT
WIFI CONNECTED
WIFI GOT IP
OK
AT+CIPSTART="TCP","api.thingspeak.com",80
CONNECT
OK
AT+CIPSEND=47
OK
> GET /update?api_key=C7JFHZY54GLCJY38&field1=1
Recv 47 bytes
SEND OK
+IPD,4:1117CLOSED
  
```

In the below program of TCP Client, do the following

For TCP Client RECEIVE demo

```

#define RECEIVE_DEMO          /* Define Receive demo */
// #define SEND_DEMO          /* Define Send demo */
  
```

For TCP Client SEND demo



```
/* Define Required fields shown below */
#define DOMAIN           "api.thingspeak.com"
#define PORT             "80"
#define API_WRITE_KEY    "thingspeak Write Key"
#define CHANNEL_ID       "thingspeak Channel ID"

#define SSID             "WiFi SSID"
#define PASSWORD          "WiFi Password"
```

In the below program, we are using response-based functions to get status if things deviate from normal.

Code for TCP Client using ESP8266 & PIC

```

/*
 * TCP Client using ESP8266 & PIC
 * http://www.electronicwings.com
 */

#include <string.h>
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <stdbool.h>
#include <pic18f4550.h>
#include "USART_Header_File.h"
#include "Configuration_header_file.h"

#define DEFAULT_BUFFER_SIZE 160
#define DEFAULT_TIMEOUT 10000

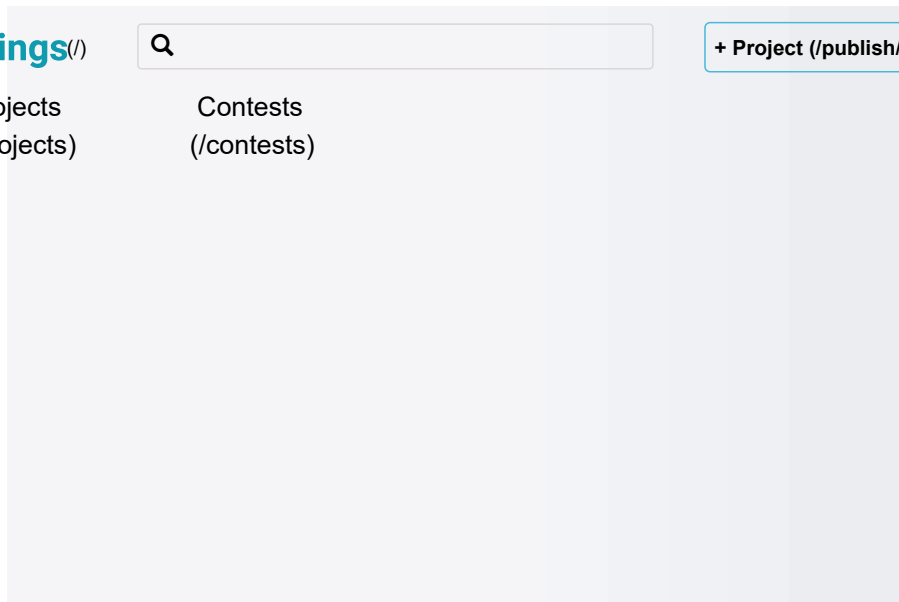
/* Connection Mode */
#define SINGLE 0
#define MULTIPLE 1

/* Application Mode */

```

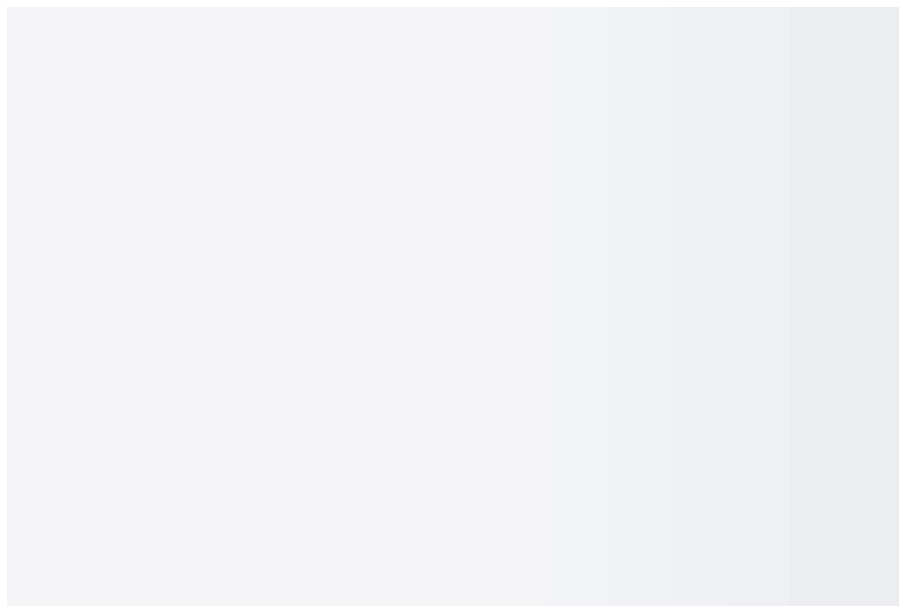
ESP8266 Response

At the client end, we need to check ESP8266 responses. We can check it on the serial terminal of PC/Laptop. Connect ESP8266 module transmit pin (TX) to the receive pin (RX) of PIC18F4550 Microcontroller and to the receive pin (RX) of usb to serial converter as shown in below figure. connect usb to serial converter to PC/Laptop. Open the serial terminal on PC/Laptop to see the ESP8266 responses for the AT command sent from PIC18F4550 microcontroller.

[Platforms \(/explore\)](#)[Projects \(/projects\)](#)[Contests \(/contests\)](#)

PIC18F4550 Interface with ESP8266 along with PC

Now for **TCP SEND** commands (sent from PIC18F4550 Microcontroller), we can see the below response from ESP8266 on the serial terminal for the Thingspeak server.



In response to **TCP SEND** we get the data entry no. as shown in the above figure i.e. 1131, 1132, and so on.

For **TCP RECEIVE** commands (sent from PIC18F4550 Microcontroller), we can see the below response from ESP8266 on the serial terminal for the Thingspeak server.

[Platforms](#)
[\(/explore\)](#)[Projects](#)
[\(/projects\)](#)[Contests](#)
[\(/contests\)](#)

In response to **TCP RECEIVE** we get the last entry data for field1 on Thingspeak as shown in the above figure.

Note: here we are retrieving the last entry data on field1 of thingspeak server hence we get the last updated data of field1 from the server as shown in the above figure i.e. "field1": "11". In the program, we used "GET /channels/119922/feeds/last.txt" to receive the last updated data only.

Updates at Thingspeak server on TCP SEND

For **TCP SEND** we can see the output at the server end. Here we are using thingspeak server and sending the incremented count at field1 on the server. We get incremented count at field1 of thingspeak server as shown in the below figure.

EW

ElectronicWings

(/)

Q

MOUSER

+ Project Electronics project

(https://www.mouser.in?utm_source=electronicswing&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)

Platform

(/explore)

Project

(/projects)

Contest

(/contests)

Components Used

ESP8266 WiFi Module
ESP8266 is a system on chip (SoC) which provide...

X 1

(https://www.mouser.in/ProductDetail/Olimex-Ltd/MOD-WIFI-ESP8266?qs=%2Fha2pyFaduiM2FizGGE3eS5tzBwP6H7%2FU4oZoQX%252BUf3gTFtvjctAktEO9xN01fVd&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)

Datasheet (/components/esp8266-wifi-module/1/datasheet)

PIC18f4550
PIC18f4550

X 1

(https://www.mouser.in/ProductDetail/Microchip-Technology/PIC18F4550-I-P?qs=oKK8NaWdAJs8nLDXBGwMXw%3D%3D&utm_source=electronicswings&utm_medium=display&utm_campaign=mouser-componentslisting&utm_content=0x0)

Datasheet (/components/pic18f4550/1/datasheet)

<https://www.electronicwings.com/pic/pic18f4550-interface-with-esp8266-module>

7/10



Platforms
(/explore)

Projects
(/projects)

Contests
(/contests)



ESP8266 Datasheet

Dow (/api/download/platf
nloa orm-attachment/535)
d



ESP8266 AT Commands

Dow (/api/download/platf
nloa orm-attachment/536)
d



ESP8266 Getting Started Guide

Dow (/api/download/platf
nloa orm-attachment/537)
d



PIC18F4550 ESP8266 WiFi Project file

Dow (/api/download/platf
nloa orm-attachment/538)
d

Comments



Comment

gaurav

(/users/gaurav/profile)

2018-01-11 02:41:27

hello.....value frequency oscillator?....u have used in this project

Reply Like

lokeshc

(/users/lokeshc/profile)

2018-01-11 10:15:59

@gaurav: here it seems that internal frequency is used which is 8MHz

OSCCON = 0x72;

Reply Like

Alvaro

(/users/Alvaro/profile)

2018-09-04 19:50:43

Hello, very nice tutorial. Congrats!!!

I have a question, why you always put the last Cahr of the AT command array as '0', i.e
_atCommand[19] = 0;

I saw that in SendATandExpectResponse function you add the LF and CR as the
esp8266 AT commands specs tells

Saludos

Reply Like 1

rgok175

(/users/rgok175/profile)



Platforms
(/explore)

Projects
(/projects)

Contests
(/contests)

file ?

Reply Like

lokeshc

(/users/lokeshc/profile)

2019-03-16 10:00:28

It is header file which has all function definition & declaration for pic controllers.

It is for xc compiler.

Reply Like

hichamoufettoul

(/users/hichamoufettoul/profile)

2020-04-20 19:01:58

hi i hope you fine . thank you about your tutorial is very interessting . my question is .

how can i send informations of sensor of temperarrent and current .voltage by using

pic18f4550 and esp8266 to my web site and thank you

Reply Like 2

peet

(/users/peet/profile)

2020-04-22 22:28:26

I am also interested in your request. I tried to connect to my own website but without success.

Reply Like 1

hichamoufettoul

(/users/hichamoufettoul/profile)

2020-04-23 00:16:39

okay i will search about this question if i find the answer i will send it to you

Reply Like 2

peet

(/users/peet/profile)

2020-04-23 01:03:33

Thank you!

Reply Like 1

zaidimran24

(/users/zaidimran24/profile)

2021-06-02 15:11:07

hi can you tell me if u had find the answer to this query of yours?

Reply Like 2

XimenaZacarias

(/users/XimenaZacarias/profile)

2022-10-25 05:08:33

I'm interested too, thanks!

Reply Like

About Us (/about)

Business Offering (/business-services)

Host Platform (/launch-platform)

Contact Us (/contactus)

Connect On:

Facebook(<https://www.facebook.com/electronicwings>)

LinkedIn(<https://www.linkedin.com/company/electronicwin>)

Youtube(<https://www.youtube.com/channel/UCNdqkukBtk4>)

Instagram(https://www.instagram.com/electronicwings_co
igshid=1cip10jjttko)

Terms of Service (/terms-of-service)

Cookies Policy (/cookie-policy)

ElectronicWings

© 2023

