

FPGA TECH SOLUTION

SOLUTION AHEAD

## Wi-Fi Trainer

# Wi-Fi Trainer User Manual

Download all Source code from following link

<https://github.com/fpgatechsolution/Spartan6-starter-kit>

---

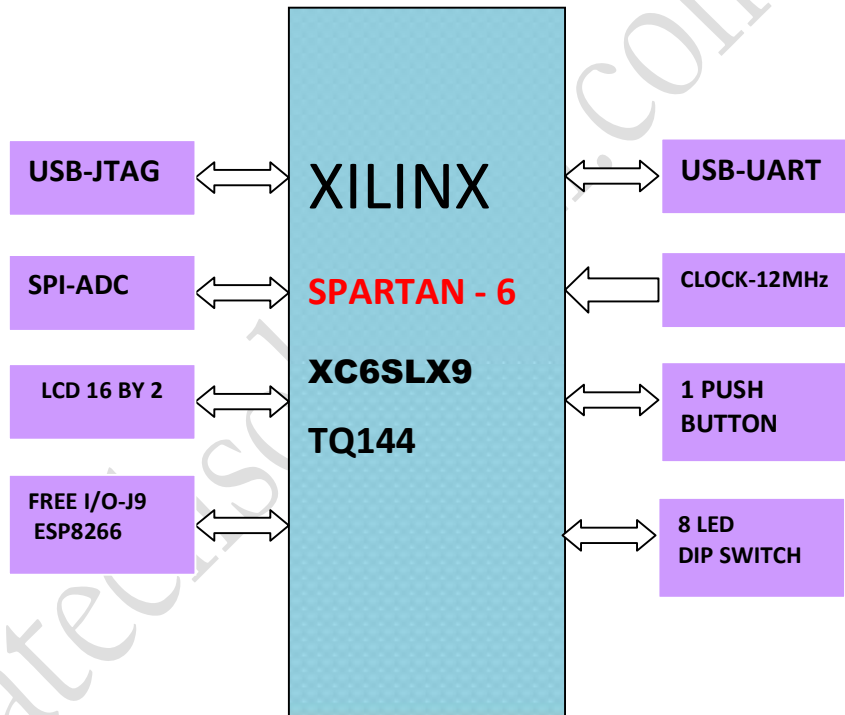
**info@fpgasolution.com,**  
**Mobile: 9665889991**  
**WWW.FPGASOLUTION.COM**

# Introduction

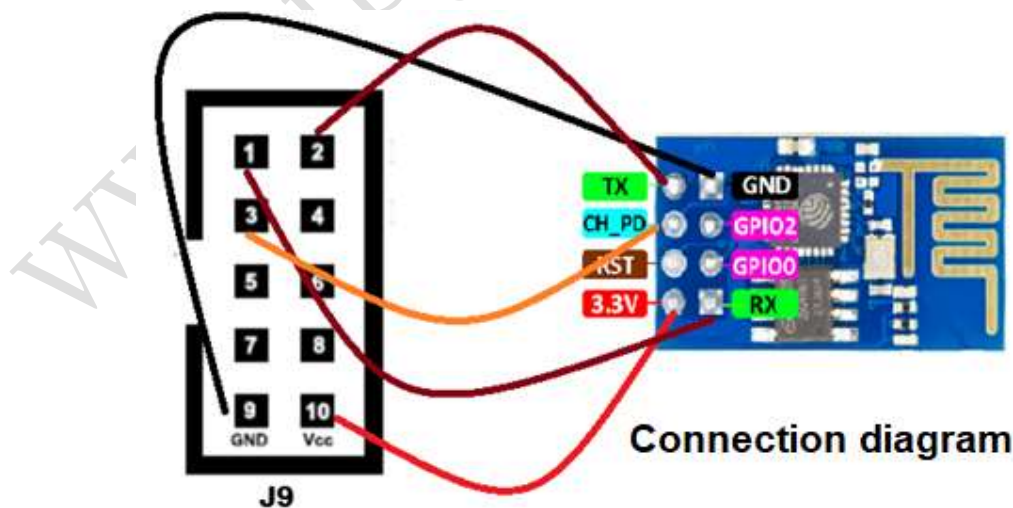
With this manual let's understand how to interface ESP8266 module with [spartan6-starter-kit](#) and send ADC value to web-server (ESP8266) with all [source code](#).

## Key components used:

- XC6SLX9\_TQ144
- OSCILLATOR – 12MHz
- On board USB JTAG
- USB to serial
- 10bit SPI ADC
- 8 LED
- 1 PUSH BUTTONS
- Free I/O J9
- LCD 16By2



For connection of esp8266 with FPGA board see following figure



Connection diagram

**AT Commands required for web server**

- 1) Ensure AT commands are received correctly (the AT seems not to be case sensitive but the rest of any command is case sensitive):

Command : AT

Response : OK

- 2) Enable the module to act as both a "Station" and an "Access Point"

Command : AT+CWMODE=3

Response : OK

- 3) List surrounding WiFi networks.

Command : AT+CWLAP

Response : You should get a response like:

+CWLAP:(3,"Cherry",-25,"e0:2c:b2:c6:91:ab",6,40)

+CWLAP:(3,"FPGATECHSOLUTION",-37,"62:f0:34:72:6f:6e",11,123,0)

- 4) Join a suitable WiFi access point:

Command: AT+CWJAP="<access\_point\_name>","<password>"

Response:

WIFI CONNECTED

WIFI GOT IP

For example, with the above list of access points you might use:

AT+CWJAP= "FPGATECHSOLUTION","FPGATECH"

- 5) Check if the module has been allocated a IP address

Command : AT+CIFSR

Response :You should get your current IP address in response like below

+CIFSR:APIP,"192.168.4.1"

+CIFSR:APMAC,"2e:3a:e8:0e:f1:87"

+CIFSR:STAIP,"192.168.43.212"

+CIFSR:STAMAC,"2c:3a:e8:0e:f1:87"

Note: STAIP is important for us as we are using the same address from the remote location to access the ESP8266. Please note down the STAIP.

- 6) You can enable the module to accept TCP connections (i.e. act as a server) in the following manner. Enable multiple connections by sending command.

**Command : AT+CIPMUX=1**

**Response : OK**

- 7) Set the module to listen (first parameter, mode is set to 1 ) for a connection on a specific port (in this case 80)

**Command : AT+CIPSERVER=1,80**

**Response : OK**

- 8) ESP8266 is ready to accept the connection.

Now open the web browser preferably Mozilla or Google Chrome. then type the web address of ESP8266 (i.e. STAIP which is already noted) in address line.



**Response:** Once you enter STAIP you get following kind of response on serial monitor +IPD,1,422:GET / HTTP/1.1

**Host: 192.168.43.212**

**Connection: keep-alive**

**Upgrade-Insecure-Requests: 1**

**User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.163 Safari/537.36**

**Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/png,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.9**

**Accept-Encoding: gzip, deflate**

**Accept-Language: en-US,en;q=0.9**

- 9) You can send data in response with the CIPSEND command as used previously, e.g. ( 0 is the channel, 79 is the length of the data):

**Command : AT+CIPSEND = 0, 79**

**Response :** The module will display the prompt: (>) to type the data to be send

- 10) Then you can send the data, e.g.: text or HTTP packet >

**< <H1> ADC OUTPUT IS 0004 0010 0405 0021 VISIT US**

**WWW.FPGATECHSOLUTION.COM </H1> >**

**Response :**

**busy s...**

**Recv 79 bytes**

**SEND OK**

In the browser you should get the following output.

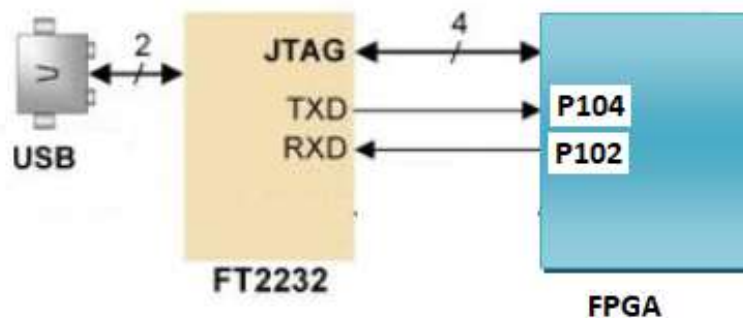
← → × 192.168.43.212

ADC OUTPUT IS 0004 0010 0405 0021 VISIT US WWW.FPGATECHSOLUTION.COM

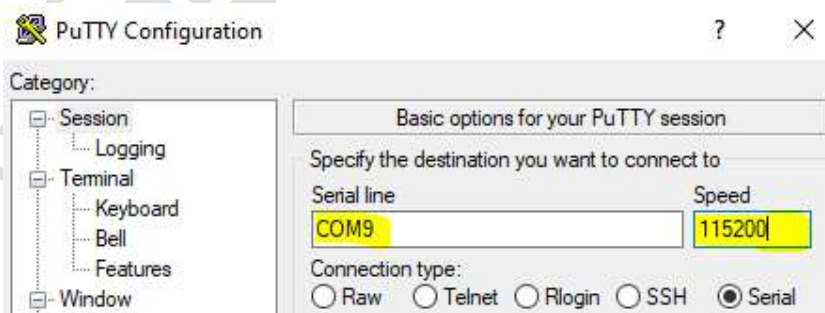
channel channel channel channel

1 2 3 4

The **SPARTAN6 STARTER KIT** board have USB interface using device FT232HL from FTDI. This act as USB to UART converter so that Communication with FPGA can accomplished by USB port.



Now connect USB cable to [spartan6-starter-kit](#) and open UART terminal with 115200 baud rate

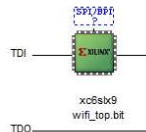




On this serial terminal we can see AT commands sent to ESP8266 and also response for those AT commands

Program wifi\_top.bit file in [spartan6-starter-kit](#) then all things will done automatically

Detail source code can be download from [here](#).



Program Succeeded

Serial terminal output we can see AT commands sent to ESP8266 and also response for those AT commands

```

ready
WIFI_DISCONNECT
AT+CWMODE=3

OK
AT+CWLAP
+CWLAP:(4,"Rajput Home",-82,"e4:be:ed:3d:23:21",3,93,0)
+CWLAP:(0,"BADG-Z29kc2dpZnRhYmhpamVldA",-83,"dc:e8:38:06:d0:0a",6,107,0)
+CWLAP:(3,"FPGATECHSOLUTION",-37,"62:f0:34:72:6f:6e",11,125,0)

OK
AT+CWJAP="FPGATECHSOLUTION","FPGATECH"
WIFI_CONNECTED
WIFI_GOT_IP

OK
AT+CIFSR
+CIFSR:APIP,"192.168.4.1"
+CIFSR:APMAC,"2e:3a:e8:0e:f1:87"
+CIFSR:STAPIP,"192.168.43.212"
+CIFSR:STAMAC,"2c:3a:e8:0e:f1:87"

OK
AT+CIPMUX=1

OK
AT+CIPSERVER=1,80

OK
0,CONNECT
1,CONNECT
AT+CIPSEND=0,79
|
OK
>
+IPD,1,422:GET / HTTP/1.1
Host: 192.168.43.212
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/80.0.3987.163 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/a
png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9

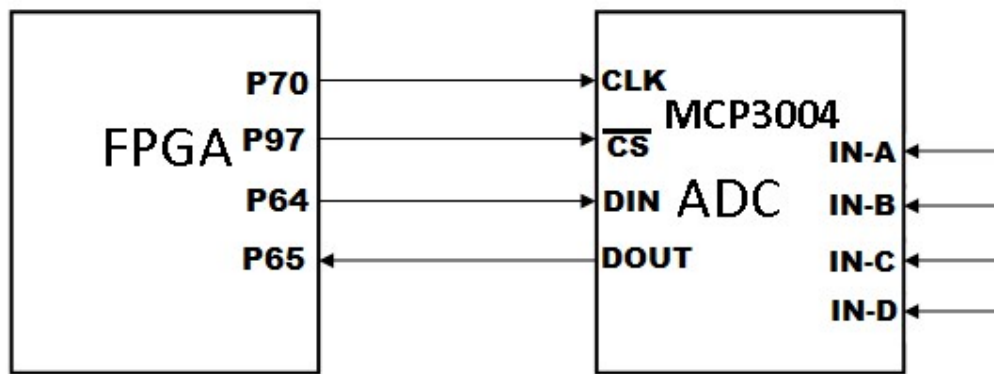
Recv 79 bytes

SEND OK

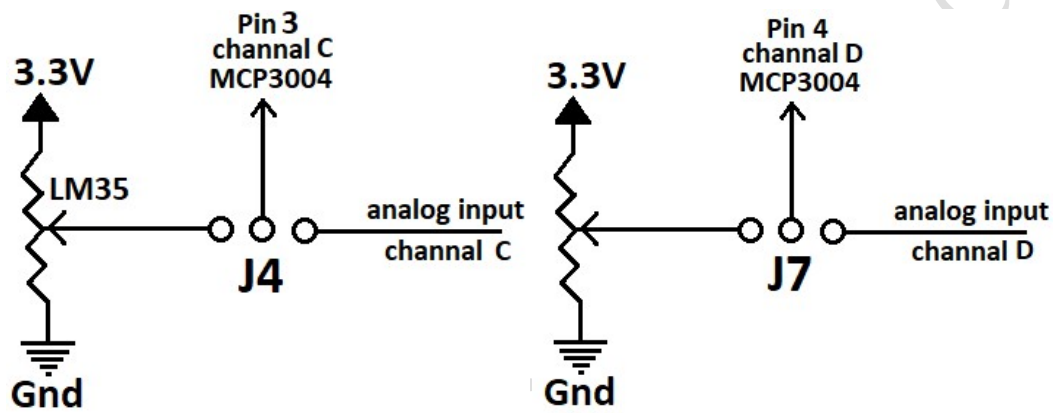
```

## ADC Interface

The **SPARTAN6 STARTER KIT** board includes an ADC MCP3004. The ADC has 4 analog input channels. The channels are selected by setting the address pins of ADC. The analog input to all channels is given by external circuit through relimate pins. The other controlling signals of ADC are interfaced with FPGA board as shown in following figure. VREF is connected to 3.3V, so analog voltage input rang of all channel is 0 to 3.3V.

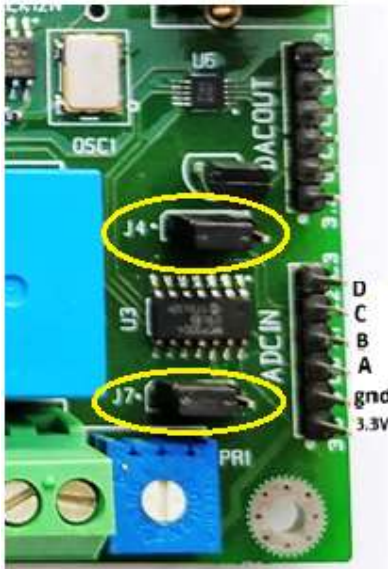


Interfacing of ADC with FPGA



Now as per following image J4 & J7 jumper connected, analog input given from temperature sensor Im35 and pot PR1 respectively





**Please do not apply  
voltage more than 3.3v**

For more information on ADC interfacing visit this link [how to test ADC](#)

