ESP8266
Design document

July 29
2015

This document describes the embedded software design of the ESP8266 Wifi module.

Version 1.0

Revision History

Date	Version	Description	Author	Role
10/9/2015	1.0	Document creation	Ibrahim Mostafa	Junior ES Engineer

Table of Contents

Table of Figures

1 Introduction

1.1 Purpose

The purpose of this document is to describe the detailed design of the ESP8266 module and how it works.

1.2 Definitions, Acronyms, and Abbreviations

ACK	*Acknowledgement*
AP	*Access Point*

1.3 References

Item	Name	link
[1]	ESP8266ATCommandsSet	
[2]	4A-AT-Espressif AT Instruction Set_020	
[3]	4B-AT-Espressif AT Command Examples_v0.3	
[4]	NURDspace on ESP8266	https://nurdspace.nl/ESP8266

1.4 Overview

ESP8266 offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor.

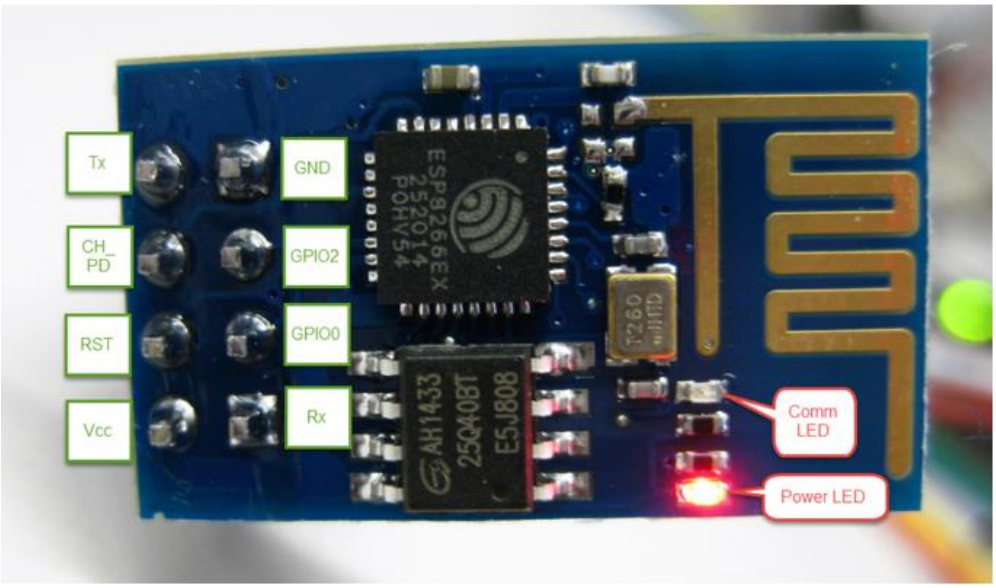
1.5 Folders and files structure

ESP8266 module was implemented by three files: ESP8266.c, ESP8266_cfg.h and nRF2401.h .

1.6 Features

- 802.11 b/g/n protocol
- Wi-Fi Direct (P2P), soft-AP
- Integrated TCP/IP protocol stack
- Integrated TR switch, balun, LNA, power amplifier and matching network
- Integrated PLL, regulators, and power management units
- Integrated temperature sensor
- Supports antenna diversity
- Integrated low power 32-bit CPU could be used as application processor
- SDIO 2.0, SPI, UART

1.7 Connection



The ESP8266 requires 3.3V power–do not power it with 5 volts!
The ESP8266 needs to communicate via serial at 3.3V and does not have 5V tolerant inputs, so you need level conversion to communicate with a 5V microcontroller like most Arduinos use.

Connect RST to Vcc (3.3 volt), and keep GPIO2 and GPIO0 floating and connect them to 0 when downloading new firmware for the module from the manufacturing company.
Connect 3.3v to CH_PD as chip select.

NOTE: The module work with 3.3v only (for Vcc and TX and RX) and the module need around 1 Amper so you should use Power Supply 3.3v .

1.8 Application:

- Smart power plugs
- Home automation
- Mesh network
- Industrial wireless control
- Baby monitors
- IP Cameras
- Sensor networks
- Wearable electronics
- Wi-Fi location-aware devices
- Security ID tags

2 Detailed Design

The module works with AT commands via UART protocol.

2.1 Initialization Module:

Using Some Commands to define the Mode as you can make it Access point or Station (client) or both of them, make it Multiple Connection, get Module Version and get IP of Module.

Basic	
Instruction	**Description**
AT	Test AT startup
AT+RST	Restart module
AT+GMR	View version info
AT+GSLP	Enter deep-sleep mode
ATE	AT commands echo or not

2.2 Join Access point and Set IP

There are commands helps to join AP or create AP and to Set Station or AP IP.

WIFI	
Instruction	**Description**
AT+CWMODE	WIFI mode（station/softAP/station+softAP）
AT+CWJAP	Connect to AP
AT+CWLAP	Lists available APs
AT+CWQAP	Disconnect from AP
AT+CWSAP	Set parameters under AP mode
AT+CWLIF	Get station's ip which is connected to ESP8266 softAP
AT+CWDHCP	Enable/Disable DHCP
AT+CIPSTAMAC	Set mac address of ESP8266 station

2.3 Open Server and Send to Server

There are Commands helping in Open server, define time out of connection and to send data. When sending data make a connection then send request command then send data.

When Sending Data, Multiple connection must be used.

TCP/IP	
Instruction	**Description**
AT+ CIPSTATUS	Get connection status
AT+CIPSTART	Establish TCP connection or register UDP port
AT+CIPSEND	Send data
AT+CIPCLOSE	Close TCP/UDP connection
AT+CIFSR	Get local IP address
AT+CIPMUX	Set multiple connections mode
AT+CIPSERVER	Configure as server
AT+CIPMODE	Set transmission mode
AT+CIPSTO	Set timeout when ESP8266 runs as TCP server

3 Driver Functions

3.1 Internal Functions

3.1.1 EF_BOOLEAN_ESP8266_GetCharArray

Format	EF_BOOLEAN_ESP8266_GetCharArray(U8_t UART_Number ,U8_t * ReturnedArray, U16_t NewLine_Numbers)
Description	This function used to receive from UART array until defined given NewLine Number or until number of definied tries or until reach to the Max Size of RX Buffer Array.
Argument	UART_Number : Number ,max. digits of it should be equaled MAX_DIGIT_FOR_ITOA ReturnedArray : pointer to the returned Array NewLine_Numbers : Max New Line receive character to stop when reaching it.
Return value	BOOLEAN to check for Errors

3.1.2 EF_BOOLEAN_ESP8266_ReadUntilExpectedKeyword

Format	EF_BOOLEAN_ESP8266_ReadUntilExpectedKeyword (U8_t* ExpectedKeyword, U8_t ExpectedKeyword_Length, U16_t TimeOut)
Description	EF_BOOLEAN_ESP8266_ReadUntilExpectedKeyword This function used to receive Array by UART, and stop until Expected keyword received or when timeout occured
Argument	ExpectedKeyword : Expected array to stop when find it or when time out ExpectedKeyword_Length : length of ExpectedKeyword Array.

	TimeOut : mili seconds (Ex: if timeout 5000 means 5sec), using to extract from this function if ExpectedArray was not found
Return value	TRUE: means OK, found this Expected word in Receiving progress. FALSE : means time is out and not found the ExpectedArray

3.1.3 itoa_Convert

Format	itoa_Convert (U32_t Number , U8_t * NumberASCII_ptr ,U8_t* NumberOFDigits_ptr)
Description	This function used to convert integer number to ASCII
Argument	U16_t Number : integer Number U8_t * NumberASCII_ptr : pointer to the ASCII number after Conversion NumberOFDigits_ptr: pointer to size NumberASCII_ptr array
Return value	None.

3.2 Global Functions

3.2.1 EF_B_Wavecom_InitModule

Format	EF_B_Wavecom_InitModule (void);
Description	This function used to initialise ESP8266 Wifi, init UART and Timer , Send some Init Commands
Argument	NONE
Return value	BOOLEAN to check for Errors

3.2.2 EF_BOOLEAN_ESP8266_JoinAP

Format	EF_BOOLEAN_ESP8266_JoinAP (U8_t* AcessPoint, U8_t AcessPoint_Length, U8_t* Password, U8_t Password_Length);
Description	This function used to connect to Given Access Point
Argument	AcessPoint: pointer to Acess Point name AcessPoint_Length: length of AcessPoint array Password: pointer to Password array Password_Length: password array length
Return value	return TRUE if ok or FALSE if not Expected Answer

3.2.3 EF_BOOLEAN_ESP8266_SetIp

Format	EF_BOOLEAN_ESP8266_SetIp (U8_t* IP_ptr, U8_t IP_Length);

Description	This function used to Set Ip for Wifi Module
Argument	IP_ptr: pointer to IP IP_Length: IP Length
Return value	BOOLEAN to check for Errors

3.2.4 EF_BOOLEAN_ESP8266_CreateAcessPoint

Format	EF_BOOLEAN_ESP8266_CreateAcessPoint (U8_t* AP_ptr, U8_t AP_Length);
Description	This function used to Set Ip for Wifi Module
Argument	AP_ptr : pointer to AP name which is wanted to create AP_Length : AP Length
Return value	BOOLEAN to check for Errors

3.2.5 EF_BOOLEAN_ESP8266_SendToServer

Format	EF_BOOLEAN_ESP8266_SendToServer (U8_t* IP_ptr, U8_t IP_Length, U8_t* Data_ptr, U8_t DataLength);
Description	Send data to Server , (to Server Ip), and if errors repeat all steps for number of attemps = MAX_ATTEMPS
Argument	IP_ptr : Tx IP IP_Length : length of Ip array Data_ptr : pointer to Data wanted to send DataLength : Data Length
Return value	BOOLEAN to check for Errors

3.2.6 EF_BOOLEAN_ESP8266_GetRxData

Format	EF_BOOLEAN_ESP8266_GetRxData (U32_t UartBase , U8_t* RxData_ptr);
Description	This function used to parcing to get the rx Data and to print any data and to reinit if reset
Argument	UartBase RxData_ptr: pointer to received data 0 if byte is not received 1 if normal byte received 2 if data completed 3 if hardware reset occured
Return value	the data located in this register