

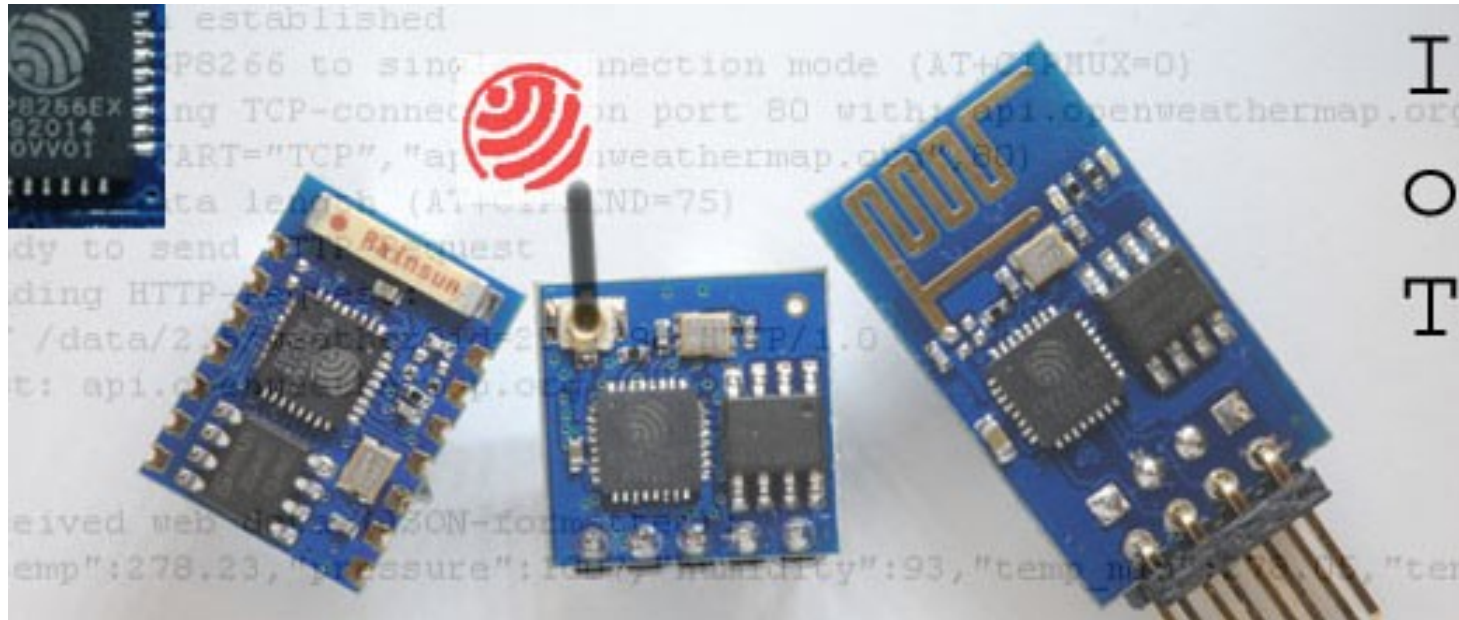
Termine im Attraktor

- 14.2. Näh-Workshop mit (Suse)
- 21.2. Workshop Portalfräse (Axel)
- 2.3. Elektronik-Stammtisch: Selbstbauprojekt Reflow Oven (Lucas & Lucas)
- 28.3. Arduino-Day
- 6.4. Elektronik-Stammtisch: Schaltplan und PCB-Software DipTrace (Axel)

- März Workshop Dosenantenne mit ESP8266
- März Anfänger-Lötworkshop
- April ESP8266 Workshop

Attraktor Webseite	blog.attraktor.org
Attraktor wiki	wiki.attraktor.org
Terminkalender	https://wiki.attraktor.org/Attraktor_e.V._Wiki:Current_events
Mailingliste	https://lists.attraktor.org/listinfo/attraktionen
Twitter	@Attraktor_org
Facebook	https://www.facebook.com/attraktoreV
Meetup-Gruppe	http://www.meetup.com/attraktor/

ESP8266



Internet of Things (IoT) für 3 Euro

Markus Ulsaß, Axel Theilmann, Attraktor Hamburg, 2.2.2015

Attraktor

Attraktor - der Makerspace in Hamburg

- seit 2010 gemeinnütziger Verein, früher City-Nord, seit April 2014 in Altona
- auf 270 Quadratmetern
- ca. 100 Mitglieder
- Veranstaltungszentrum und offene Werkstatt
- für Technikinteressierte und Kreative
- Geräte wie CNC-Fräsen, 3D-Drucker, Laser-Cutter, Drehbank, Platinenfräse
- Regelmäßige Events: Basteldonnerstag, Elektronik-Stammtisch, Näh- und Lötworkshops
- Finanzierung über Mitgliederbeiträge, Veranstaltungen und Spenden
- Wir machen gerne eine Führung durch den Space im Anschluss

Veranstaltungstermine im Kalender

https://wiki.attraktor.org/Attraktor_e.V._Wiki:Current_events

Agenda

- Grundlagen ESP8266EX, Sicherheit
- Betriebsarten, Modultypen
- Konfiguration ESP-01, Firmware flashen, AT-Modus
- Anwendungsbeispiel Arduino & ESP8266 (Abruf Wetterdaten)
- Eigene Firmware (Flashen, RemoteSwitch mit Lua/ NodeMcu, Explorer IDE)
- Firmware mit Espressif-SDK, Windows-/ Linux-Toolchain

Sicherheit

Unbekannter Hersteller (Modul: China/ Core: USA)

Ungewöhnlich niedriger Preis

Möglichkeit des Cloud-Update/ „phone home“?

Unbekannte WiFi/ TCP/IP-Implementierung

Nur teilweise offene Sourcen (SDK)

Dokumentation sehr lückenhaft (oft nur chinesisch)

Gerät ist im eigenen Netzwerk

IoT generell Zugriff von außen (theoretisch) möglich

Nicht in sicherheitsrelevanten Bereichen nutzen

Grundlagen

ESP8266EX

SoC (System on Chip)

Hersteller Espressif, China

5x5mm

Package 32 T/SLP

Pin-pitch: 0,5mm



Minimalbeschaltung: 7 Widerstände und Kondensatoren,
1 Quarz, 1 Flash-RAM (z.B. 4Mbit/ 512kByte)

Grundlagen/ WiFi

Hardware-WLAN:

802.11 b/g/n (2.4GHz, 2412 - 2484 MHz, Kanäle 1-14)

Integrierter TCP/IP protocol stack (IPv4)

Client und Access-Point gleichzeitig (3 Modi: Station, SoftAP, Station & SoftAP), WiFi-Direct (P2P)?

20(24)dBm output im 802.11b Modus/ Output programmierbar

Stromverbrauch: bis zu 250mA (ESP8266EX-IC – keine Peripherie)

DHCP, WebServer

Encryption: WEP, WPA-PSK, WPA2/ PSK (AES & TKIP)

Grundlagen/ CPU Core

CPU-Core

- Low Power Core **Xtensa LX3**
- Tensilica „Diamond Standard 106Micro Controller“ 32-bit RISC MCU (Prozessor-Design von 2004, LX3 2009)
http://ip.cadence.com/uploads/pdf/xtensalx_overview_handbook.pdf
- Tochter von Cadence Design Systems (electronic design automation - EDA/ semiconductor intellectual property - IP)
- Mit integriertem on-Chip SRAM (Harvard-Architektur: „IRam“ & „DRam“)

Anwendung und Firmware werden im on-Chip ROM und SRAM ausgeführt, geladen werden sie durch das SDIO 2.0-Interface vom externen Flash-Speicher (Quad/ Dual SPI-Flash, oft 4Mbit/ 512kByte, z.B. ESP-01). Flash-Speicher wird in den 32bit-Adressraum des Xtensa Core gemapped

Taktfrequenz: 80 Mhz (Vgl.: Arduino 16 MHz, 8 Bit)

Grundlagen/ Block Diagram

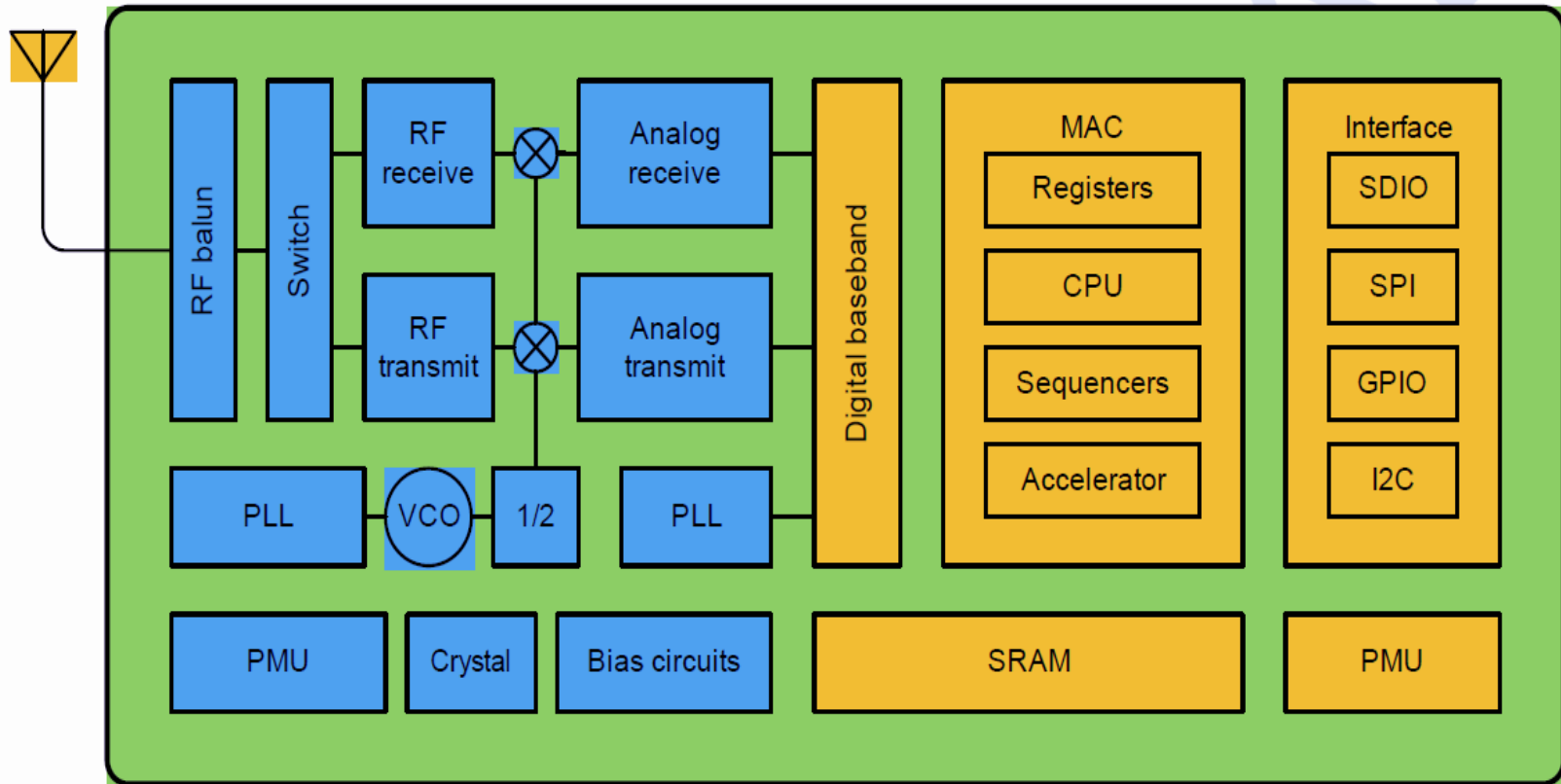


Figure 1: ESP8266EX Block Diagram

Quelle: [Datenblatt Espressif Beginner's Guide, Version 17.9.2014](#)

Grundlagen/ Schnittstellen

Spannungsversorgung: 3 – 3,6V

I_{max} GPIO: 12mA sink pro Pin, source >12mA, I_{max} ges.: 16*12mA

<http://bbs.espressif.com/viewtopic.php?t=139>

Schnittstellen:

GPIO (bis zu 16) (nicht 5V-tolerant!)

-> jeder mit internem Pull-Up/ Pull-Down, Output, Input (Sampling), High-Impedance (TriState), Input Trigger...

-> multiplexed mit UART, SPI...

SDIO 2.0 (Anbindung Flash RAM)

„1.5“ * **UART** (uart0: RXD/TXD GPIO25/26, uart1: TXD GPIO2)

I2C, SPI, PWM

10-bit **ADC** (0~1V) – Nutzung nur, wenn Transceiver nicht aktiv

JTAG debugging interface

System-level Features: sleep (Power-Down <10µA, 60µA deep-sleep (RTC/ WDT running), <1mA verbunden mit Access Point), VoIP, ASP...

Firmware update durch Cloud Server

Grundlagen/ Pinout

1.2.1. Pin Definition

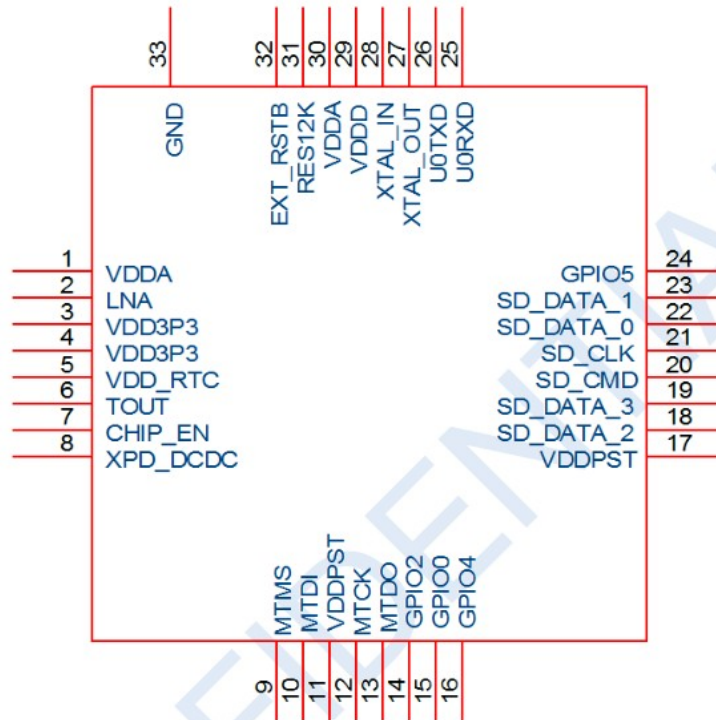


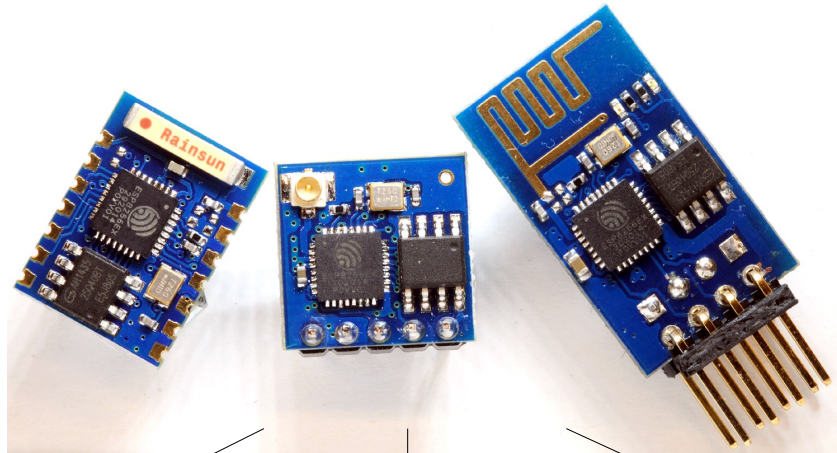
Figure 2: ESP8266EX Pin Definition Diagram

Pin	Name	Type	Function
1	VDDA	P	Analog Power 3.0 ~3.6V
2	LNA	I/O	RF Antenna Interface. Chip Output Impedance=50 Ω No matching required but we recommend that the π -type matching network is retained.
3	VDD3P3	P	Amplifier Power 3.0~3.6V
4	VDD3P3	P	Amplifier Power 3.0~3.6V
5	VDD_RTC	P	NC(1.1V)
6	TOUT	I	ADC Pin
7	CHIP_EN	I	Chip Enable. High: On, chip works properly; Low: Off, small current
8	XPD_DCDC	I/O	Deep-Sleep Wakeup : GPIO16
9	MTMS	I/O	GPIO14; HSPICLK
10	MTDI	I/O	GPIO12; HSPICQ
11	VDDPST	P	Digital I/O Power Supply (1.8V~3.3V)
12	MTCK	I/O	GPIO13; HSPID
13	MTDO	I/O	GPIO15; HSPICS
14	GPIO2	I/O	UART Tx during flash programming; GPIO2
15	GPIO0	I/O	GPIO0; SPICS2
16	GPIO4	I/O	GPIO4
17	VDDPST	P	Digital I/O Power Supply (1.8V~3.3V)
18	SDIO_DATA_2	I/O	Connect to SD_D2 (Series R 200 Ω); SPIHD; HSPHD
19	SDIO_DATA_3	I/O	Connect to SD_D3 (Series R 200 Ω); SPIWP; HSPWP
20	SDIO_CMD	I/O	Connect to SD_CMD(Series R 200 Ω); SPICS0
21	SDIO_CLK	I/O	Connect to SD_CLK (Series R 200 Ω); SPICLK
22	SDIO_DATA_0	I/O	Connect to SD_D0 (Series R 200 Ω); SPIQ
23	SDIO_DATA_1	I/O	Connect to SD_D1 (Series R 200 Ω); SPID
24	GPIO5	I/O	GPIO5
25	U0RXD	I/O	UART Rx during flash programming; GPIO3
26	U0TXD	I/O	UART Tx during flash programming; GPIO1; SPICS1
27	XTAL_OUT	I/O	Connect to crystal output, can be used to provide BT clock input
28	XTAL_IN	I/O	Connect to crystal input
29	VDDD	P	Analog Power 3.0~3.6V
30	VDDA	P	Analog Power 3.0~3.6V
31	RES12K	I	Connect to series R 12k Ω to ground
32	EXT_RSTB	I	External reset signal (Low: Active)

Note: GPIO2, GPIO0, MTDO can be configurable as 3-bit SDIO mode

Quelle: Datenblatt Espressif Beginner's Guide, Version 17.9.2014

Betriebsarten



AT-Modus

Seriell auf WiFi

+Arduino, RasPi etc.

Third-Party Firmware

Stand-alone

Lua/ NodeMcu

MQTT

Frankenstein etc.

Espressif SDK

Stand-alone

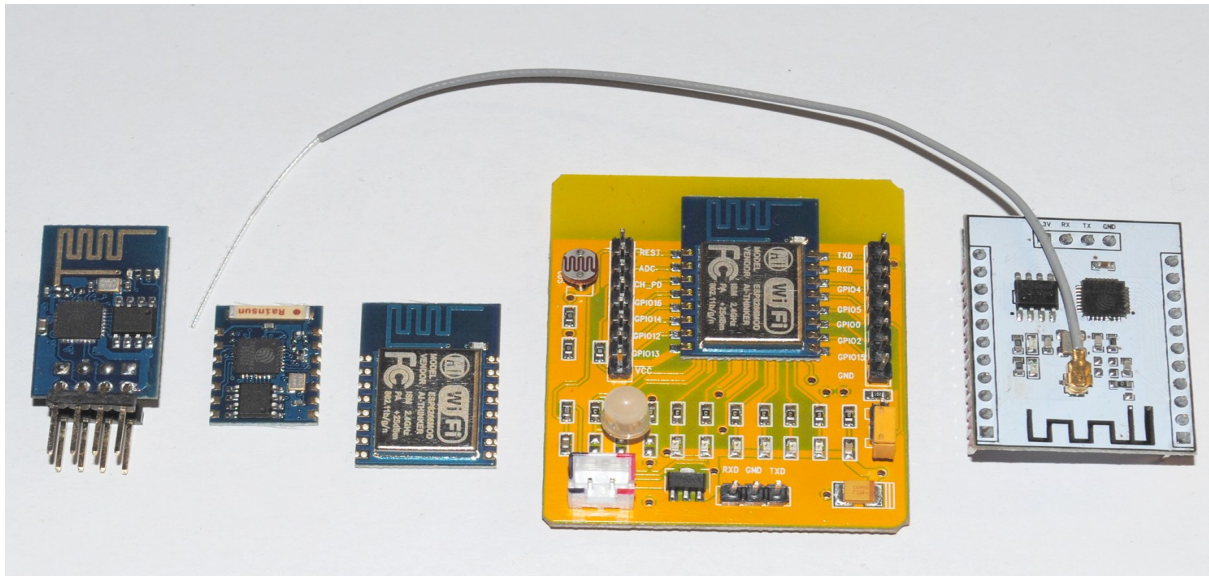
Linux/ Win IDE,

Espressif SDK

C/ C++

Modultypen

- „ESP-X“-Modul Serie (ESP-01, 02..,-12) mit und ohne externe Antenne, in verschiedenen Größen und mit unterschiedlich vielen herausgeführten GPIOs (über ebay, Aliexpress)
- Dev Boards (z.B. Olimex, ESP-201, custom made)



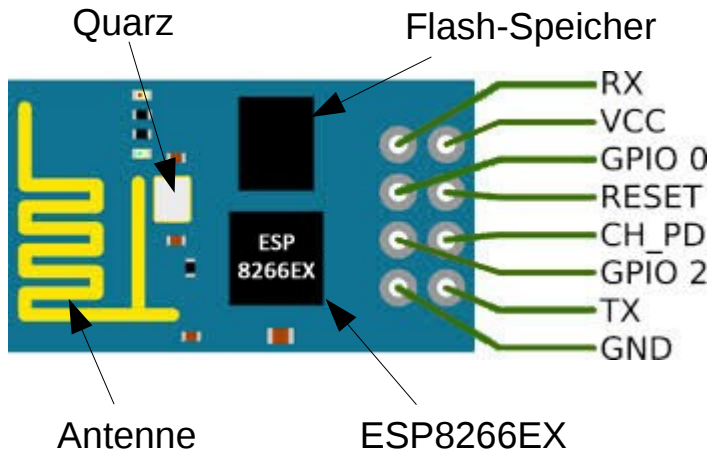
ESP-01, ESP-03, ESP-12, Dev Board, ESP-201

Modultypen/ Vor dem Start

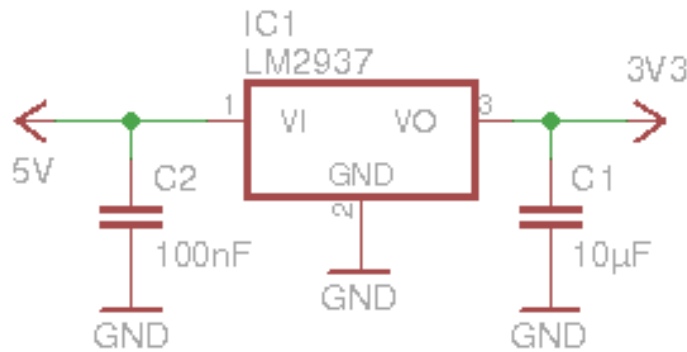
Voraussetzungen:

- ✓ Stabile Spannungsversorgung (3-3,6V, min. 500mA, Puffer-Kondensator zum Ausgleich von Spitzen/ Linearregler vs. Schaltregler)
- ✓ Eingänge sind nicht 5V-tolerant (Level-Shifter mit Widerständen, Mosfets, 4050, ADI „ADUM“-Serie etc.)
→ <http://www.mikrocontroller.net/articles/Pegelwandler>
- ✓ USB-Seriell Adapter (FTDI, CP210x, PL2303-HX...) möglichst mit 3,3V sonst 5V mit Level-Shifter (Massen verbinden)
- ✓ Terminal-Software (putty, GtkTerm, TeraTerm, Arduino Serial-Monitor...)
- ✓ Baudrate (9600/ 115200), Boot-Log Baudrate (74880)
- ✓ Je nach Modul, Beschaltung und Betriebsart müssen bestimmte GPIOs (z.B. GPIO0, GPIO15, RESET, CH_EN (CH_PD) auf einen Signalpegel gezogen werden (empfehlenswerter weise mit Widerstand)

Modultypen/ Beispiel ESP-01



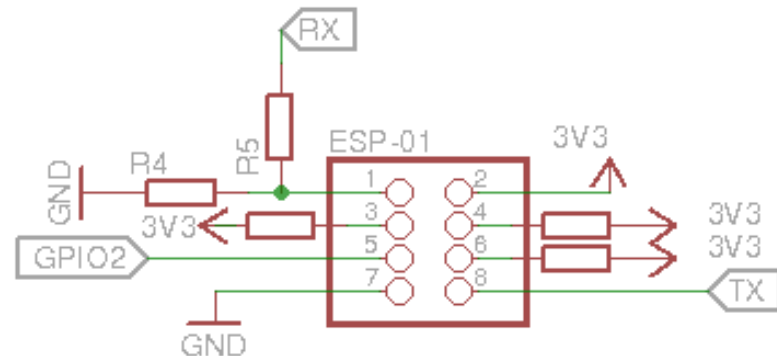
Spannungsversorgung



Anwendungs-Modus:

VCC → 3,3V (3 – 3,6V)
GND → GND
GPIO0 → VCC + R (~4,7k)
RESET → VCC + R (~4,7k)
CH_PD → VCC + R (~4,7k)
(GPIO15 → GND)

Anwendungs-Modus



AT-Firmware aufspielen

Manche Module werden ohne Firmware ausgeliefert. Zum Flashen einer Firmware muss das Modul in den Programmier-Modus gebracht werden. Dazu wird GPIO0 auf GND gelegt.

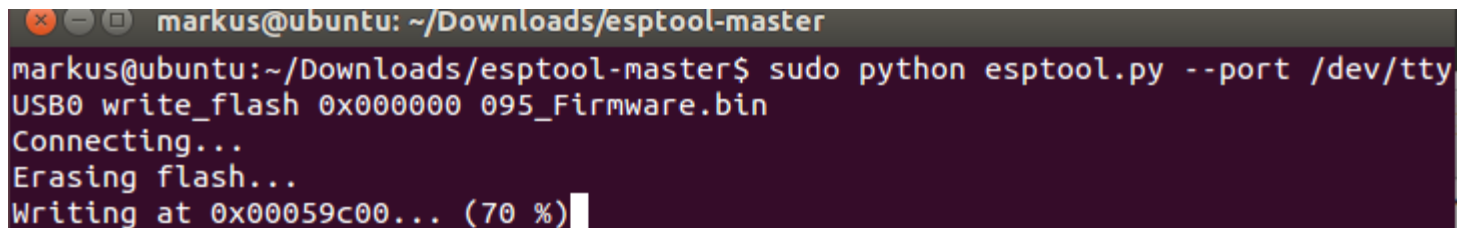
Als Flash-Tool kann man z.B. esptool.py benutzen. Dazu wird Python 2.7 und Pyserial (Windows/ Linux) benötigt.

<https://github.com/themadinventor/esptool>

Zusätzlich benötigt man ein Binary der AT-Firmware, z.B. diese:

<http://www.electrodragon.com/w/Wi07c#Firmware>

Das Binary flasht man mit Hilfe von esptool auf das Modul:

A terminal window with a dark background and light text. The title bar shows 'markus@ubuntu: ~/Downloads/esptool-master'. The command 'sudo python esptool.py --port /dev/tty' has been executed. The output shows 'USB0 write_flash 0x000000 095_Firmware.bin', followed by 'Connecting...', 'Erasing flash...', and 'Writing at 0x00059c00... (70 %)' with a cursor at the end of the line.

```
markus@ubuntu: ~/Downloads/esptool-master
markus@ubuntu:~/Downloads/esptool-master$ sudo python esptool.py --port /dev/tty
USB0 write_flash 0x000000 095_Firmware.bin
Connecting...
Erasing flash...
Writing at 0x00059c00... (70 %)█
```


AT-Firmware aufspielen

Ist der Vorgang erfolgreich, dann sieht das ganze etwa so aus:

```
markus@ubuntu: ~/Downloads/esptool-master
markus@ubuntu:~/Downloads/esptool-master$ sudo python esptool.py --port /dev/tty
USB0 write_flash 0x000000 095_Firmware.bin
Connecting...
Erasing flash...
Writing at 0x0007ec00... (100 %)

Leaving...
markus@ubuntu:~/Downloads/esptool-master$
```

Nicht vergessen: Modul nach dem Flashen vom Strom trennen, GPIO0 wieder auf Vcc und wieder mit Spannung versorgen. Dann kann man das Modul mit einem Terminal-Programm ansprechen.

```
COM15:9600baud - Tera Term VT
File Edit Setup Control Window Help
58
ready
AT

OK
AT+GMR
00200.9.5(b1)
compiled @ Dec 25 2014 21:40:28
AI-THINKER Dec 25 2014

OK
```

AT-Modus

AT (=“**ATTENTION**“) → Modul wird als „Modem“ angesprochen (Hayes Communications seit 1981 als AT-Befehlssatz)

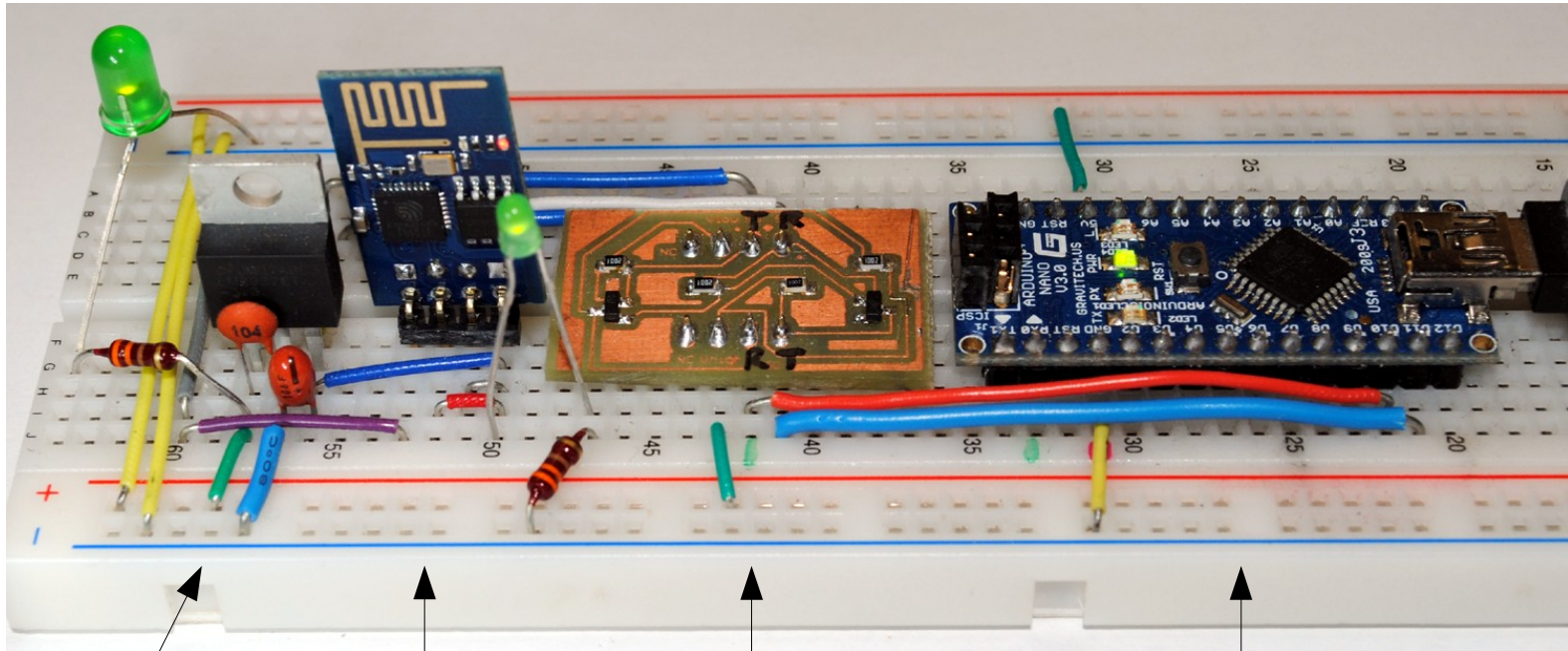
- ✓ „Richtige“ Baudrate einstellen (je nach Firmware 9600/115200)
- ✓ Terminal-Programm muss die Eingaben mit CR+LF abschließen!

Beispiel für AT-Befehle mit ESP8266:

AT Kommando	Beschreibung
AT	Test – Antwort „OK“
AT+RST	Modulreset, Bootmeldung, „ready“
AT+GMR	Abfrage der Firmware Version
AT+CWLAP	Ausgabe der verfügbaren APs
AT+CWJAP=<ssid>,<password>	Verbinden mit WiFi-Netzwerk

https://github.com/espressif/esp8266_at/wiki/AT_Description

Beispiel Arduino & ESP8266



3,3V
Spannungs-
versorgung

Modul
ESP-01

Level-Shifter

Arduino
Nano (5V)

Beispiel Arduino & ESP8266

```
COM78 - PuTTY
Sending reset to ESP8266 (AT+RST)
ESP8266 module ready
Connecting to SSID:      (AT+CWJAP=<SSID>,<PASSWORD>)
Connection established
Setting ESP8266 to single connection mode (AT+CIPMUX=0)
Establishing TCP-connection on port 80 with: api.openweathermap.org
(AT+CIPSTART="TCP","api.openweathermap.org",80)
Sending data length (AT+CIPSEND=75)
Ready to send HTTP-Request
Sending HTTP-Request: >
GET /data/2.5/weather?id=2911298 HTTP/1.0
Host: api.openweathermap.org

Received web data (JSON-formatted):
{"temp":278.23,"pressure":1007,"humidity":93,"temp_min":278.05,"temp_max":278.45}
Parsing JSON data.....
Temperature: 5.08
Pressure: 1007
Humidity: 93
```

Eigene Firmware Lua/ NodeMcu

Lua is a powerful, fast, lightweight, embeddable scripting language.

[*http://www.lua.org*](http://www.lua.org)

NodeMcu: An open-source firmware and development kit that helps you to prototype your IOT product within a few Lua script lines.

[*http://nodemcu.com/index_en.html*](http://nodemcu.com/index_en.html)

[*https://github.com/nodemcu*](https://github.com/nodemcu)

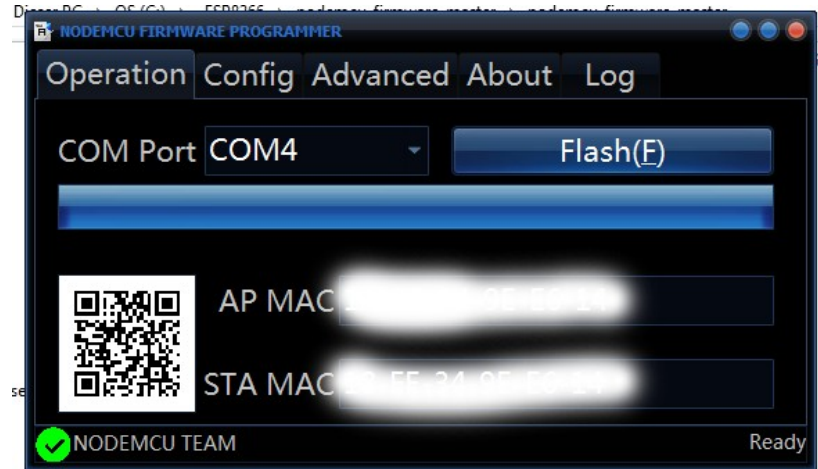
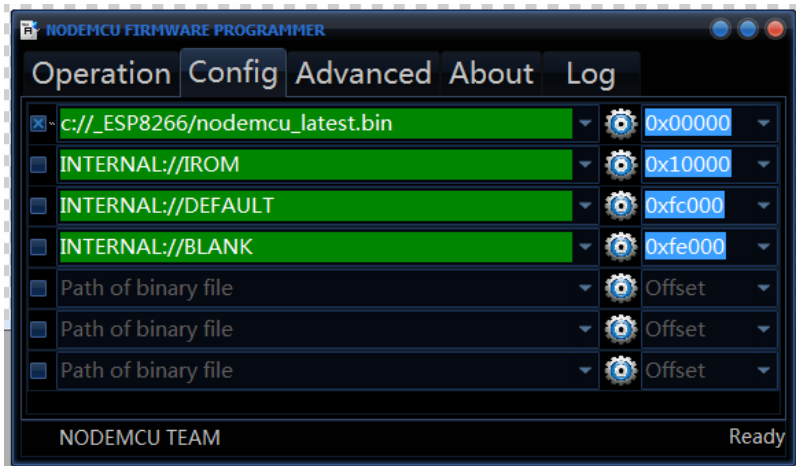
Zum Flashen der Firmware muss das Modul in den Programmier-Modus gebracht werden. Dazu wird GPIO0 auf GND gelegt. Flash-Tool z.B. esptool.py

```
markus@ubuntu:~/ESPtool/esptool-master$ sudo python esptool.py --port /dev/ttyUS
; B0 write_flash 0x000000 180902.bin
ck Connecting...
Erasing flash...
r Writing at 0x0007ef00... (100 %)
sole Leaving...
markus@ubuntu:~/ESPtool/esptool-master$
```

Eigene Firmware Lua/ NodeMcu

Firmware flashen mit NodeMcu-Flasher (Windows):

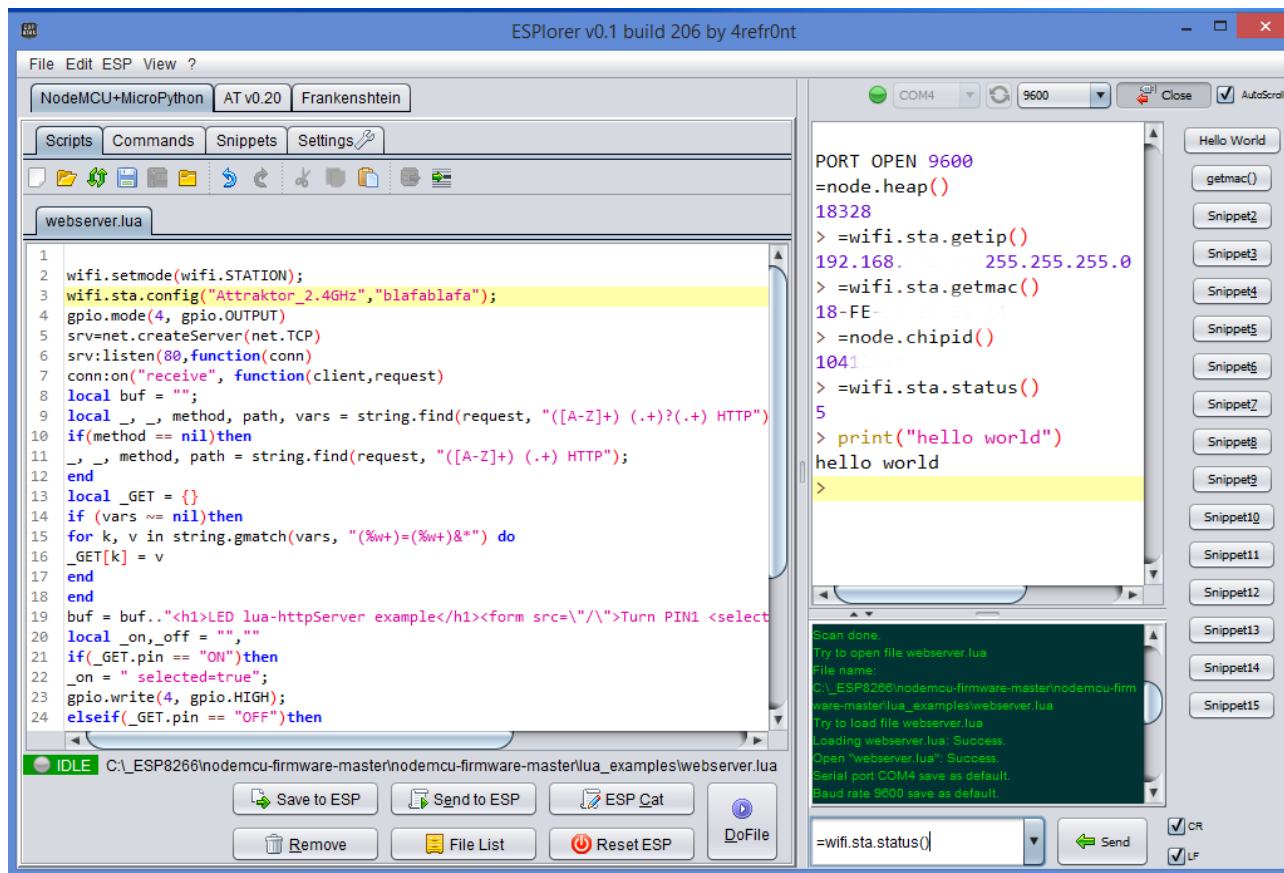
- ✓ Firmware von [NodeMcu github](#) herunterladen & entpacken
- ✓ Flasher von [NodeMcu github](#) herunterladen & entpacken
- ✓ ESP-Modul in den Programmier-Modus bringen (GPIO0 auf GND) und anschließen (USB-Seriell)
- ✓ Flasher aufrufen und in /nodemcu-firmware/prebuild/latest die Firmware auswählen
- ✓ Flashen → fertig



Lua/ NodeMcu RemoteSwitch

Mit der Lua-IDE „Esplorer“ lassen sich Skripte unter NodeMcu (und MicroPython) sowie AT-Kommandos testen und aufspielen.

<http://www.esp8266.ru/esplorer>



Firmware mit SDK/ Windows

Will man mit Hilfe des Espressif SDKs und dem Xtensa Crosscompiler eigene Programme aufspielen, kann man dies unter Windows z.B. mit Eclipse tun.

Im Forum esp8266.com gibt es dazu eine detaillierte Anleitung und schon vorbereitete Software. Am Ende hat man eine komplette Toolchain unter Windows mit Eclipse und kann eigenen C/ C++-Code programmieren und auf den ESP8266 aufspielen.

In dem Programmpaket sind bereits zahlreiche Programm-Beispiele vorhanden.

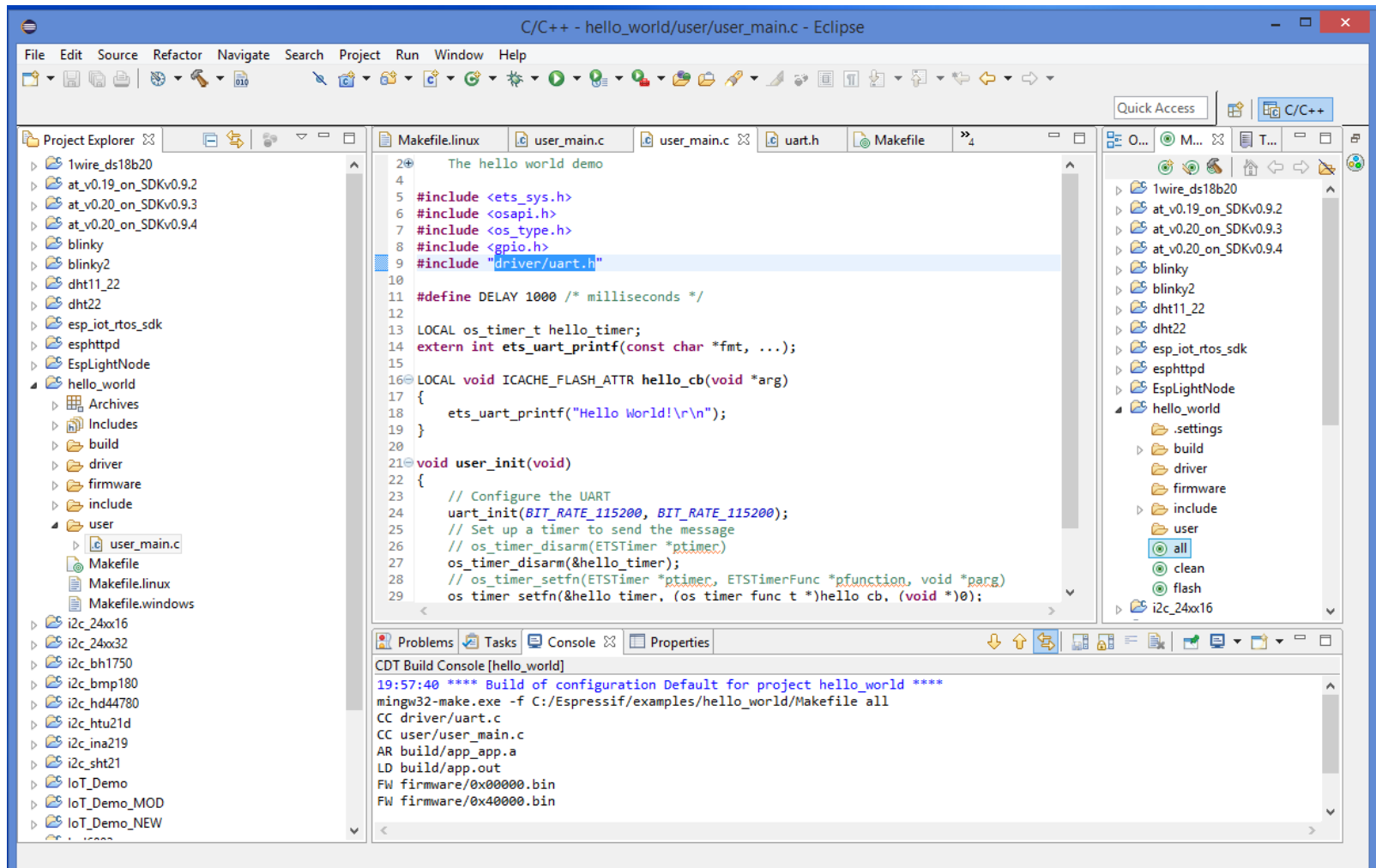
<http://www.esp8266.com/viewtopic.php?t=820>

Infos zum Espressif SDK findet man im:

Espressif IoT SDK Programming Guide

http://www.mikrocontroller.net/attachment/245197/2C-SDK-Espressif_IoT_SDK_Programming_Guide_v0.9.5.pdf

Firmware mit SDK/ Windows



Links I

Links im Text

Attraktor Kalender: https://wiki.attraktor.org/Attraktor_e.V._Wiki:Current_events

Die Shot: <http://zeptobars.ru/en/read/Espressif-ESP8266-wifi-serial-rs232-ESP8089-IoT>

Xtensa LX Handbook: http://ip.cadence.com/uploads/pdf/xtensalx_overview_handbook.pdf

Espressif Beginner's Guide:

ftp://imall.iteadstudio.com/Basic_Module/IM141118001/0B-Beginner%20Guide%20to%20ESP8266_V0.4.1.pdf

Maximale Strombelastbarkeit: <http://bbs.espressif.com/viewtopic.php?t=139>

FAQ Pegelwandler: <http://www.mikrocontroller.net/articles/Pegelwandler>

AT-Firmware: <http://www.electrodragon.com/w/Wi07c#Firmware>

AT-Befehle: https://github.com/espressif/esp8266_at/wiki/AT_Description

Lua: <http://www.lua.org/>

NodeMcu: http://nodemcu.com/index_en.html

NodeMcu github: <https://github.com/nodemcu>

Esptool: <https://github.com/themadinventor/esptool>

Explorer: <http://www.esp8266.ru/explorer>

Espressif IoT SDK Programming Guide:

http://www.mikrocontroller.net/attachment/245197/2C-SDK-Espressif_IoT_SDK_Programming_Guide_v0.9.5.pdf

Windows/ Eclipse Toolchain: <http://www.esp8266.com/viewtopic.php?t=820>

Foren

www.ESP8266.com

www.Mikrocontroller.net

bbs.Espressif.com

Links II

Websites:

Espressif: <https://espressif.com/en/products/esp8266/>

Espressif github: <https://github.com/espressif>

Nurdspace: <https://nurdspace.nl/ESP8266>

Electrodragon: <http://www.electrodragon.com/w/Wi07c>