

Simple serial time and HTTP client

API Version 00.75

(C) 2016 Christian Zietz <czietz@gmx.net>

Allgemeine Konventionen

Die Einstellungen der seriellen Schnittstelle sind: 9600 bit/s, 8 Datenbits, kein Paritätsbit, 1 Stopbit (8N1).

Befehle an das Modul dürfen nur mit einem Wagenrücklauf (CR, \r, ASCII 13) abgeschlossen werden. Befehle sind grundsätzlich nur in Kleinbuchstaben zu verwenden. Antworten des Moduls sind grundsätzlich mit Wagenrücklauf und Zeilenvorschub (CR/LF, \r\n, ASCII 13 und ASCII 10) abgeschlossen.

vers: Versionsnummer der Firmware

Dieser Befehl fragt die Versionsnummer der Firmware ab. In diesem Dokument ist vermerkt, falls Befehle erst ab einer bestimmten Firmwareversion unterstützt werden.

Beispielaufruf:

```
vers
```

Beispielantwort:

```
00.70
1      4      0      8561729 1327328 1024      0      40000000
```

Die erste Zeile enthält die Firmware-Version. Die zweite Zeile enthält Informationen, die nur zu Debugging-Zwecken nützlich sind: Versionsnummern des NodeMCU-Interpreters (major, minor, dev), die individuelle Chip-ID des ESP8266, die ID des verbauten Flashbausteins, dessen Größe in kiB, dessen Datenübertragungsmodus und dessen Taktrate.

wifi: Liste der verfügbaren WLANs

Dieser Befehl zeigt die WLANs an, die vom ESP8266 empfangen werden.

Beispielaufruf:

```
wifi
```

Beispielantwort:

	SSID	RSSI	AUTH	CHANNEL
	NETGEAR	-72	3	6
	UPC2759999	-87	4	11
	meinWLAN	-54	3	2
	UPC247339999	-90	3	6
	UPC2758999	-77	4	11
	WLAN-69C999	-87	3	6

Angezeigt werden jeweils die SSID, die Empfangsstärke (RSSI), die Verschlüsselung (0: unverschlüsselt, 1: WEP, 2: WPA-PSK, 3: WPA2-PSK, 4: WPA/WPA2-PSK) und der verwendete WLAN-Kanal.

ssid: SSID des WLANs

Mit diesem Befehl wird die SSID des zu verwendenden WLANs festgelegt. Die WLAN-Konfiguration wird im Flash-Speicher des Moduls abgelegt und bleibt somit über einen Neustart des Moduls hinaus erhalten.

Beispielaufruf:

```
ssidmeinWLAN
```

Beispielantwort:

```
meinWLAN
```

Das Modul antwortet mit der gewählten SSID. Achtung: Das bedeutet nicht, dass die Verbindung erfolgreich war. Dies muss mit dem Befehl `stat` überprüft werden.

pass: Passwort des WLANs

Mit diesem Befehl wird das Passwort des zu verwendenden WLANs festgelegt. Das Passwort muss entweder 8 – 63 Zeichen lang sein oder bei einem unverschlüsselten WLAN leer. Die WLAN-Konfiguration wird im Flash-Speicher des Moduls abgelegt und bleibt somit über einen Neustart des Moduls hinaus erhalten.

Beispielaufruf:

```
passGeheimesPasswort
```

Beispielantwort:

```
meinWLAN
```

Das Modul antwortet mit der gewählten SSID. Achtung: Das bedeutet nicht, dass die Verbindung erfolgreich war. Dies muss mit dem Befehl `stat` überprüft werden.

stat: Status der WLAN-Verbindung

Mit diesem Befehl wird der Status der WLAN-Verbindung abgefragt.

Beispielaufruf:

```
stat
```

Beispielantwort:

```
5  
meinWLAN
```

Die erste Zeile der Antwort enthält den Verbindungsstatus: 0: unkonfiguriert, 1: im Verbindungsaufbau, 2: Fehler, Passwort falsch, 3: Fehler, Accesspoint nicht gefunden, 4: Fehler, Verbindung nicht möglich, 5: erfolgreich verbunden.

Die zweite Zeile enthält die konfigurierte SSID.

time: Formatiertes Datum und Uhrzeit

Dieser Befehl liefert Datum und Uhrzeit des Zeitservers für Menschen lesbar formatiert. Für ein abfragendes Programm ist das Format des Befehls `sntp` einfacher zu interpretieren.

Beispielaufruf:

```
time
```

Beispielantwort:

```
20 FEB 2016 14:58:42 UTC
```

sntp: Datum und Uhrzeit als Zahl

Dieser Befehl liefert Datum und Uhrzeit in maschinenlesbarer Form.

Beispielaufruf:

```
sntp
```

Beispielantwort:

```
56c87fc6
```

Die Antwort codiert die Sekunden seit dem 1. Januar 1970, 0:00:00 Uhr UTC in 8-stelliger hexadezimaler Notation. Im Fehlerfall (z.B. Zeitserver nicht erreichbar) wird 00000000 geantwortet. Das Modul verursacht danach selbsttätig, erneut eine Verbindung zum Zeitserver herzustellen.

setu: Speicherung von Benutzerdaten

Ab Firmware 00.60

Dieser Befehl erlaubt es, eine Zeile beliebige Daten zu speichern. Diese Daten werden im Flash-Speicher des Moduls abgelegt und bleiben somit über einen Neustart des Moduls hinaus erhalten. Diese Daten werden zurzeit nicht weiter genutzt, geplant ist ein Speichern der vom Benutzer gewählten Zeitzone.

Beispielaufruf:

```
setuLorem ipsum!
```

Beispielantwort:

```
OK
```

getu: Abruf von Benutzerdaten

Ab Firmware 00.60

Die mit setu gespeicherten Daten können mit diesem Befehl wieder abgerufen werden. Diese Daten werden zurzeit nicht weiter genutzt, geplant ist ein Speichern der vom Benutzer gewählten Zeitzone.

Beispielaufruf:

```
getu
```

Beispielantwort:

```
Lorem ipsum!
```

http: http-Verbindung

Ab Firmware 00.60

Dieser Befehl lädt eine Datei aus dem Web über HTTP herunter. HTTPS wird nicht unterstützt. Achtung: Das Modul unterstützt keine Flusskontrolle auf der seriellen Schnittstelle. Das empfangende Programm muss daher dafür sorgen, dass die übertragenen Daten rechtzeitig abgeholt werden, damit der Empfangspuffer nicht überläuft und Bytes verlorengehen.

Dieser Befehl ist dazu gedacht, kleine Datenmengen, z.B. Informationen über Sommerzeitumstellungen aus dem Internet herunterzuladen.

Beispielaufruf:

```
http://www.example.com/
```

Die Angabe eines Dateinamens oder Pfads (im obigen Beispiel nur /) ist verpflichtend, optional kann auch noch eine Portnummer angegeben werden:

```
http://www.example.com:8080/
```

Beispielantwort:

```
HTTP/1.1 200 OK
Date: Sat, 20 Feb 2016 15:20:41 GMT
Server: Apache/2.2.16 (Debian)
X-Frame-Options: SAMEORIGIN
Set-Cookie: wire=ngip4d8cj59id9999tbrnm8888; path=/; HttpOnly
Expires: Thu, 19 Nov 1981 08:52:00 GMT
X-XSS-Protection: 1; mode=block
Content-Length: 3462
Connection: close
Content-Type: text/html; charset=utf-8

<!DOCTYPE html>
[...]
#### END ####
```

Zurückgeliefert wird die unmodifizierte Antwort des HTTP-Servers, Header und Body.

Zusätzlich wird die Antwort mit `\r\n#### END ####\r\n` abgeschlossen. Wenn die erwartete Datenmenge nicht (z.B. über den Header `Content-Length`) bekannt ist, so ist das Abwarten dieser Kennung und des folgenden Timeouts (es werden keine Daten mehr gesendet) der einzige Weg, das Ende der Übertragung festzustellen.

brid: Transparente TCP/IP-zu-seriell-Verbindung

Ab Firmware 00.75

Mit diesem Befehl kann das Modul angewiesen werden, auf Port 23 (Telnet) auf eingehende TCP-Verbindungen zu warten. Kommt eine solche zustande, werden sämtliche über diese Verbindung übertragenen Zeichen unverändert auf der seriellen Schnittstelle ausgegeben. Umgekehrt werden auf der seriellen Schnittstelle empfangene Zeichen unverändert über die TCP-Verbindung übertragen.

Damit können z.B. Daten zwischen Rechnern übertragen werden.

Achtung: Solange die TCP-Verbindung besteht, reagiert das Modul auf keine der in dieser Anleitung genannten Befehle. Dazu muss die TCP-Verbindung erst wieder beendet werden.

Beispielaufruf:

```
brid1
```

um eingehende Verbindungen auf Port 23 zu erlauben oder

```
brid0
```

um diesen Port wieder zu schließen.

Beispielantwort:

```
10.1.2.3
```

Das Modul antwortet mit seiner IP-Adresse.