

Die NanoESP Library

[Originalartikel](#)

[Backup](#)

```
<html> <div class=„entry-content“><div class=„entry-content-inner“><div class=„content-
headline“><h1 class=„entry-headline“>Die neue NanoESP-Library mit MQTT, HTTP und Blynk-
Support</h1></div><p class=„post-meta“> Autor: <a href=„http://iot.fkainka.de/author/fk“
title=„Beitr&#228;ge von fk“ rel=„author“>fk</a> Montag, der 19. September 2016 <a
href=„http://iot.fkainka.de/library#comments“>26 Kommentare</a></p><div class=„post-info“><p
class=„post-category“><a href=„http://iot.fkainka.de/category/board“ rel=„category
tag“>Board</a>, <a href=„http://iot.fkainka.de/category/calendar-2016“ rel=„category
tag“>Calendar 2016</a>, <a href=„http://iot.fkainka.de/category/calendar-2016/days-16“
rel=„category tag“>Days-16</a>, <a href=„http://iot.fkainka.de/category/board/support“
rel=„category tag“>Support</a></p><p class=„post-tags“><a
href=„http://iot.fkainka.de/tag/library“ rel=„tag“>Library</a></p></div><p>Im Kontext des neues
IoT Adventskalenders 2016 wurde die Library zum NanoESP um einige Funktionen erweitert. Neben
den klassischen Befehlen zum Steuern des Moduls, gibt es in dieser Version auch zwei Sub-Libraries,
die das &#220;bermitteln von Webseiten erleichtern oder die Verwendung des beliebten MQTT(<a
href=„http://mqtt.org/“>http://mqtt.org/</a>) IoT-Protokolls erm&#246;glichen.</p><h1><a
href=„https://github.com/FKainka/NanoESP/archive/v1.1.zip“><img class=„wp-image-499 alignleft“
src=„http://iot.fkainka.de/wp-content/uploads/2015/09/download-151556_640.png“
alt=„download-151556_640“ width=„40“ height=„48“
srcset=„http://iot.fkainka.de/wp-content/uploads/2015/09/download-151556_640.png 536w,
http://iot.fkainka.de/wp-content/uploads/2015/09/download-151556_640-251x300.png 251w“
sizes=„(max-width: 40px) 100vw, 40px“/></a></h1><h1><a
href=„https://github.com/FKainka/NanoESP/archive/v1.1.zip“>Download</a></h1><p>Link auf
Github:&#160;<a
href=„https://github.com/FKainka/NanoESP“>https://github.com/FKainka/NanoESP</a></p><p>Unte
r den Beispielen befindet sich au&#223;erdem ein Programm, welches Blynk-Projekte (<a
href=„http://www.blynk.cc/“>http://www.blynk.cc/</a>) erm&#246;glicht. Damit Sie das Programm
verwenden k&#246;nnen, ben&#246;tigen Sie eine zus&#228;tzliche Library:</p><p><a
href=„https://github.com/blynk/blynk-library/releases/tag/v0.3.4“>https://github.com/blynk/blynk
-library/releases/tag/v0.3.4</a></p><p>Auf dieser Seite m&#246;chte ich die einzelnen Funktionen
und Methoden etwas detaillierter erl&#228;utern sowie auf etwaige Neuerungen eingehen. Die Basis
der Library bildet die SoftwareSerial-Library, die bereits in Arduino integriert ist. Aufgrund dessen
k&#246;nnen Befehle wie find() und findUntil() auch mit dieser Library genutzt
werden.</p><h2>Inhalt</h2><p><a href=„http://iot.fkainka.de/library#_Toc462047614“>Basics
(NanoESP)<br/></a><a href=„http://iot.fkainka.de/library#_Toc462047615“>Webserver
(NanoESP_HTTP)<br/></a><a href=„http://iot.fkainka.de/library#_Toc462047616“>MQTT
(NanoESP_MQTT)<br/></a>&#160;&#8211;&#160;<a
href=„http://iot.fkainka.de/library#_Toc462047617“>Datentyp
mqtt_msg.<br/></a>&#160;&#8211;&#160;<a
href=„http://iot.fkainka.de/library#_Toc462047618“>Abonnieren mit Funktions-
Verkn&#252;pfung:</a></p><h1><a name=„_Toc462047614“/>Basics (NanoESP)</h1><table
width=„787“><tbody><tr><td width=„295“><strong><u>Funktion</u></strong></td><td
width=„492“><strong><u>Erl&#228;uterung</u></strong></td></tr><tr><td
width=„295“>boolean <strong>init</strong>(boolean vDebug=false);</td><td
width=„492“>Initialisiert das Board und stellt default Werte ein (Transfermode = 0, Multiple
```

Connections = 1)
 Wenn der Debug-Parameter true ist, wird bei einem fehlerhaften Befehl Zusatzinformationen über den Seriellen Monitor ausgegeben</td></tr><tr><td width=„295“> String sendCom(String command);</td><td width=„492“>Sendet ein AT-Kommando an das ESP-Modul</td></tr><tr><td width=„295“>boolean sendCom(String command, char respond[]);</td><td width=„492“>s.o.; Zusätzlich wird überprüft, ob das Modul eine Antwort sendet, die dem Respond-Parameter entspricht.</td></tr><tr><td width=„295“>boolean setMultipleConnections();</td><td width=„492“>Erlaubt Mehrfachverbindungen (AT+CIPMUX=1)</td></tr><tr><td width=„295“>boolean setTransferMode();</td><td width=„492“>Setzt den Transfermodus auf transparent (AT+CIPMODE=0)</td></tr><tr><td width=„295“>boolean reset();</td><td width=„492“>Resetet das Board und wartet, bis es wieder bereit ist</td></tr><tr><td width=„295“/><td width=„492“/></tr><tr><td width=„295“>boolean configWifiMode(int modus);</td><td width=„492“>Stellt den WLAN Modus ein (STATION, ACCESSPOINT, DUAL)</td></tr><tr><td width=„295“>boolean configWifi(int modus, String ssid, String password);</td><td width=„492“>Stellt das WLAN ein mit Modus, SSID und Passwort</td></tr><tr><td width=„295“>boolean configWifiStation(String ssid, String password);</td><td width=„492“>Stellt eine Verbindung zu einem WLAN-Router her</td></tr><tr><td width=„295“>boolean configWifiAP(String ssid, String password);</td><td width=„492“>Stellt einen eignen AccesPoint zur Verfügung (Passwort darf leer sein, sonst min sechs Zeichen)</td></tr><tr><td width=„295“>boolean configWifiAP(String ssid, String password, int channel, int crypt);</td><td width=„492“>S.o.<p>Zusätzliche Parameter: WLAN-Kanal, Verschlüsselungsmodus</p></td></tr><tr><td width=„295“>boolean disconnectWifi();</td><td width=„492“>Trennt die WLAN-Verbindung</td></tr><tr><td width=„295“/><td width=„492“/></tr><tr><td width=„295“>bool wifiConnected();</td><td width=„492“>Überprüft, ob automatisch eine WLAN-Verbindung zu einer Station hergestellt wurde (Zeitersparnis gegenüber neue Verbindung bei bekannter Station)</td></tr><tr><td width=„295“>bool getIpMac(String &ip, String &mac);</td><td width=„492“>Liefert IP und MAC-Adresse des Boards. Parameter werden als Referenz übergeben</td></tr><tr><td width=„295“>String getIp();</td><td width=„492“>Ermittelt die IP/IPs des Boards (AT+CIFSR)</td></tr><tr><td width=„295“>boolean newConnection(int id, String type, String ip , unsigned int port);</td><td width=„492“>Baut eine neue Verbindung (TCP oder UDP) auf.<p>Parameter: Verbindungs-ID (0-4), Typ (TCP/UDP), IP (Ziel IP oder Adresse), Port</p></td></tr><tr><td width=„295“>boolean closeConnection(int id) ;</td><td width=„492“>Trennt die Verbindung mit der angegebenen Verbindungs-ID</td></tr><tr><td width=„295“>boolean startUdpServer(int id, String ip , unsigned int port, unsigned int rcvport, int mode=0);</td><td width=„492“>Öffnet eine UDP-Verbindung<p>Der Empfangs-Port kann ein anderer sein als der Sende-Port. Modus 0: Die Ziel IP ändert sich nicht; Modus 1: die Ziel IP ändert sich einmal, wenn das Board eine Nachricht von einer anderen IP erhält; Modus 2: Das Board ändert die Ziel IP immer, wenn es eine neue Nachricht von einer anderen IP erhalten hat.</p></td></tr><tr><td width=„295“>boolean endUdpServer(int id);</td><td width=„492“>Beendet die UDP-Verbindung mit der angegeben ID</td></tr><tr><td width=„295“>boolean startTcpServer(unsigned int port) ;</td><td width=„492“>Startet einen TCP-Server unter dem angegeben Port. Es kann nur ein TCP-Server aktiviert sein</td></tr><tr><td width=„295“>boolean endTcpServer();</td><td width=„492“>Beendet den TCP-Server</td></tr><tr><td width=„295“>boolean sendData(int id, String msg);</td><td width=„492“>Sendet Text über die angegebene

Verbindung</td></tr><tr><td width=„295“>boolean sendDataClose(int id, String msg);</td><td width=„492“>Sendet Text und schließt danach die Verbindung</td></tr><tr><td width=„295“>bool sendRaw(int id, unsigned char data[], int LenChar);</td><td width=„492“>Sendet binäre Daten (kein String)</td></tr><tr><td width=„295“>int getId();</td><td width=„492“>Gibt die Verbindungs-ID bei Datenempfang an. Wenn keine Daten empfangen wurden ist der Wert -1</td></tr><tr><td width=„295“>bool recvData(int &id,int &len);</td><td width=„492“>Wird true, wenn Daten empfangen wurden. ID und Länge werden per Referenz zurückgegeben</td></tr><tr><td width=„295“>int ping(String adress);</td><td width=„492“>Pingt den Server unter der angegebenen Adresse an. Die Antwort-Zeit in ms wird zurückgegeben. Ist die Zeit 0 wurde keine Antwort empfangen.</td></tr><tr><td width=„295“></td><td width=„492“></td></tr><tr><td width=„295“>void serialDebug();</td><td width=„492“>Stellt eine direkte Verbindung zwischen Software und Hardware Serieller Schnittstelle her. Kann zum Testen von AT-Kommandos verwendet werden.</td></tr></tbody></table><h1>Webserver (NanoESP_HTTP)</h1><table width=„791“><tbody><tr><td width=„258“><u>Funktion</u></td><td width=„533“><u>Erläuterung</u></td></tr><tr><td width=„258“>bool recvRequest(int &id, String &method, String &ressource, String ¶meter);</td><td width=„533“>Gibt true zurück, wenn ein http-Request empfangen wurde. ID, Methode (POST oder GET) sowie angeforderte URL und etwaige Befehle werden per Referenz zurückgegeben.</td></tr><tr><td width=„258“>bool recvHTTP(int id, int len, String &method, String &ressource, String ¶meter);</td><td width=„533“>s.o.</td></tr><tr><td width=„258“>bool sendFromFlash(int client, const char *website, int len);</td><td width=„533“>Sendet eine als Progmem-Variable gespeicherte Webseite direkt an den angeben Client (Verbindungs-ID)</td></tr><tr><td width=„258“>bool sendStreamHeader(int connectionId);</td><td width=„533“>Sendet die Antwort auf einen Event-Stream-Request. Server-Send-Events können zum schnellen Datenaustausch vom Server zum Client genutzt werden (<http://www.html5rocks.com/en/tutorials/eventsource/basics/>)</td></tr><tr><td width=„258“>bool sendRequest(int id, char method[5], String address);</td><td width=„533“>Sendet einen http-Request an einen Server
 (ID, Methode (GET/POST), URL)</td></tr><tr><td width=„258“>bool sendRequest(int id, char method[5], String address, String parameter);</td><td width=„533“>Sendet einen http-Request an einen Server inklusive Sub-Parameter</td></tr></tbody></table><h1>MQTT (NanoESP_MQTT)</h1><table width=„791“><tbody><tr><td width=„330“><u>Funktion</u></td><td width=„461“><u>Erläuterung</u></td></tr><tr><td width=„330“>bool connect(int id, String brooker, unsigned int port, String deviceId);</td><td width=„461“>Baut eine Verbindung zu einem MQTT-Server auf. Minimale Parameter: Verbindungs-ID, Broker-Adresse und Geräte-ID (darf nur einmal auf dem Broker vorhanden sein)<p>(Defaultwerte: Clean Session = true, keepAliveTime = 120s)</p></td></tr><tr><td width=„330“>bool connect(<u>s.o</u>, mqtt_msg * lastWill);</td><td width=„461“>Erweitere Parameter der Connect-Funktion:<p>Parameter: lastWill-Nachricht</p></td></tr><tr><td width=„330“>bool connect(<u>s.o</u> , bool cleanSession, byte keepAliveTime);</td><td width=„461“>Erweitere Parameter der Connect-Funktion:<p>Parameter: cleanSession, keepAliveTime</p></td></tr><tr><td width=„330“>bool connect(<u>s.o</u> , bool cleanSession, byte keepAliveTime, mqtt_msg * lastWill);</td><td width=„461“>Erweitere Parameter der Connect-Funktion:<p>Parameter: cleanSession, keepAliveTime, lastWill-Nachricht</p></td></tr><tr><td width=„330“></td><td width=„461“></td></tr></tbody></table>

[illegible]

Parameter vom Typ String haben, der die Nachricht des Topics enth<#228;lt. Es k<#246;nnen maximal 5 dieser Funktions-Verkn<#252;pfun gen definiert werden.

Beispiel:

```
mqtt.subscribe(0, test/song, 2, gotnewSong)
void gotnewSong(String value)
{
  Serial.println(New Song: + value);
}
```

[<like href=„http://iot.fkainka.de/library“ send=„true“ layout=„button_count“ width=„200“ show_faces=„true“/>Tweet <plusone size=„medium“ href=„http://iot.fkainka.de/library“/></div></div></div><div class=„entry-content“><div class=„entry-content-inner“/></div> </html>](http://iot.fkainka.de/library)

From:

<https://schnipsl.qgelm.de/> - Qgelm

Permanent link:

<https://schnipsl.qgelm.de/doku.php?id=wallabag:die-nanoesp-library>

Last update: **2021/12/06 15:24**

