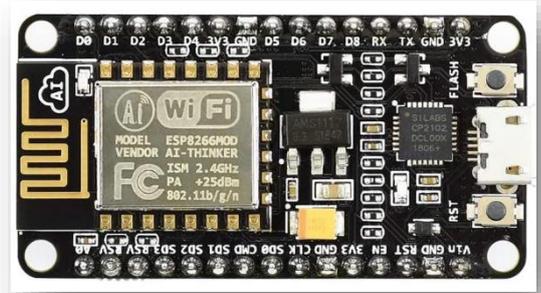


## Dein Modul NodeMCU V2 Amica mit ESP8266 12E

Dieses Modul erlaubt dir vielfältige Experimente zum Thema **Microcontroller** in Verbindung mit Sensoren und Aktoren, WLAN, Webserver, Webbrowser, HTML und Datenerfassung und -übertragung.

### Treiber- und Softwareinstallation

Installiere unbedingt die aktuelle Version der **Arduino IDE** von <https://www.arduino.cc/en/Main/Software>.



Solltest du keine Administratorrechte auf deinem PC haben, lade das „Windws ZIP file“ herunter und entpacke es. Im Ordner „Arduino IDE“ findest du die „arduino.exe“ zu Starten der Arduino IDE.

Starte die **Arduino IDE** und öffne dort im Menü **Datei** die **Voreinstellungen**.

- Schalte dort **Codefaltung aktivieren** und **Zeilennummern anzeigen** ein.

- Kopiere hinter **Zusätzliche Boardverwalter URLs**:

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)

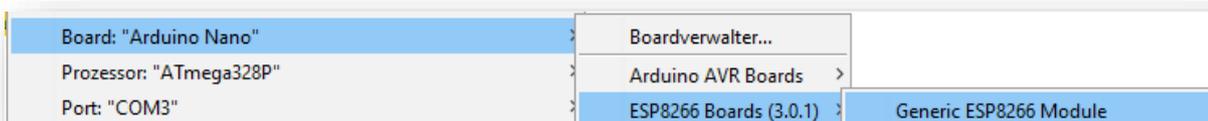
- Klicke auf **OK**.

Öffne das Menü **Werkzeuge – Board: - Boardverwalter** und gib oben in das Eingabefeld **ESP8266** ein und warte bis darunter das entsprechende Paket **esp8266 by ESP8266 Community** erscheint.

Klicke dann auf **Installieren**.



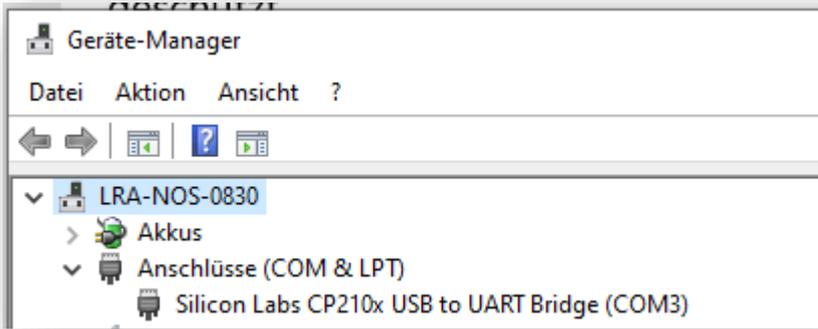
Das Menü **Werkzeuge – Board:** ist neu strukturiert und du kannst unter **ESP8266 Boards** das **Generic ESP8266 Module** auswählen.



Verbinde dein **NodeMCU-Board** per USB-Kabel mit deinem Computer und achte auf die Reaktion deines Computers.

Neben der Arduino IDE wurde auch der Treiber "Silicon Labs CP210x USB to UART Bridge" installiert, der eine serielle Verbindung zwischen deinem Computer und dem NodeMCU-Board ermöglicht.

Dieser wird im **Gerätemanager** deines PC angezeigt. Mit der Tastenkombination **[Windows]+[X]** öffnet sich ein Menü, in dem auch der **Gerätemanager** zu finden ist.



Die dort angezeigte COM-Schnittstelle, hier COM3, muss in der **Arduino IDE** unter **Werkzeuge – Port:** eingestellt werden. Damit steht unserem ersten Programm auf diesem Board nichts mehr im Wege.

### „Hello world“-Programm auf dem Amica

Dein erstes kleines Programm soll ein Lebenszeichen der **Amica** an seine Außenwelt senden.

Lade [dieses Programm](#) auf deinen Amica hoch und finde die blinkende **onboard LED** auf dem **ESP8266-Modul**.

```
blink$
1 byte LED1 = 02; // GPIO2 => D4 - onboard LED auf dem ESP8266-Modul
2
3 void setup() {
4   pinMode(LED1, OUTPUT);
5 }
6
7 void loop() {
8   digitalWrite(LED1, HIGH); delay(500);
9   digitalWrite(LED1, LOW); delay(500);
10 }
```

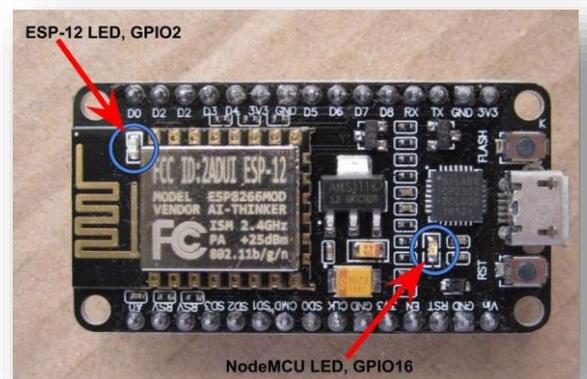
Ändere das Programm und lade es auf deinen Amica. Finde die blinkende **onboard LED** auf dem **NodeMCU-Board**.

```
blink$
1 byte LED1 = 16; // GPIO16 => D0 - onboard LED auf dem NodeMCU-Modul
```

Ändere in der Zeile 8 die Dauer für den H-Pegel am Ausgang von 500 ms auf 1000 ms.

Beantworte die Frage:

Leuchtet die onboard LED bei H-Pegel oder bei L-Pegel?



Unsere Programme sollten wir immer mit einer kurzen Info beginnen:

1. Zeile 1 - Für welches MCU-Board wurde dieses Programm geschrieben.
2. Zeile 2 – Welches Board ist unter „Werkzeuge“ einzustellen.
3. Die Angaben zu den verwendeten Ein- und Ausgängen sollten immer den Namen des GPIO-Ports **und** die entsprechende Bezeichnung auf dem NodeMCU-Board enthalten.
4. Über den **Seriellen Monitor** sollte der Name des hochgeladen Programms auf dem angeschlossenen PC ausgegeben werden.

```

blinkx2$
1 // NodeMCU Amica V2 Module mit ESP8266 12E
2 // Board: Generic ESP8266 Module
3
4 byte LED1 = 2; // GPIO2 => D4 - onboard LED auf ESP8266 12E
5 byte LED2 = 16; // GPIO16 => D0 - onboard LED auf NodeMCU Amica
6
7 void setup() {
8     Serial.begin(115200);
9     Serial.println("\nStart: blinkx2.ino");
10    pinMode(LED1, OUTPUT);
11    pinMode(LED2, OUTPUT);
12 }
13
14 void loop() {
15    digitalWrite(LED1, HIGH);
16    digitalWrite(LED2, HIGH); delay(200);
17    digitalWrite(LED2, LOW); delay(200);
18    digitalWrite(LED1, LOW);
19    digitalWrite(LED2, HIGH); delay(200);
20    digitalWrite(LED2, LOW); delay(200);
21 }

```

Lade [diese Programm](#) in die Arduino IDE und dann auf das MCU-Board hoch.

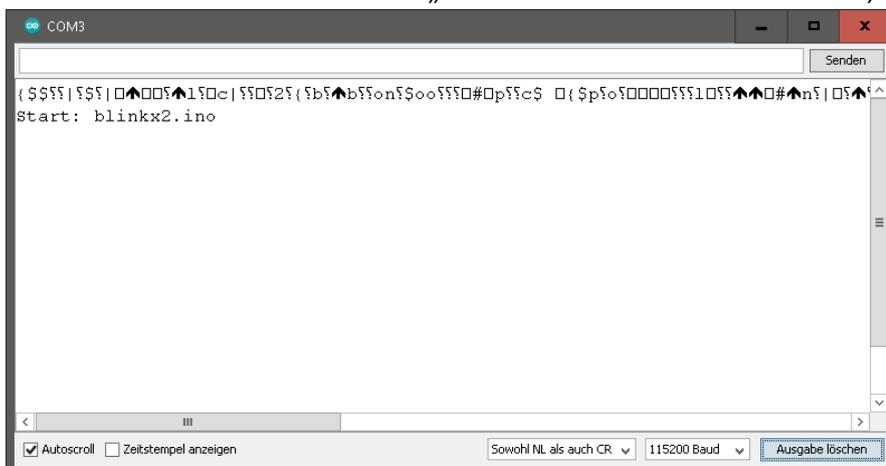


Starte den Seriellen Monitor durch Klicken auf das Symbol „Serieller Monitor“ rechts oben, **oder** über das Menü **Werkzeuge – Serieller Monitor**, **oder** mittels der Tastenkombination **[Strg]+[Umschalt]+[M]**.

Das sich öffnende Fenster trägt den Namen der COM-Schnittstelle, hier COM3.

Unten rechts ist die Baudrate auszuwählen, die durch **Serial.begin(115200)**; festgelegt wurde.

Danach ist durch Drücken die Taste „RST“ auf dem MCU-Board ein Reset, sprich ein Neustart des Amica, auszulösen.



Eventueller Kauderwelsch nach dem Reset ist zu ignorieren.

In der neuen Zeile, beginnend mit Start:, steht der Name des ausgeführten Programms.