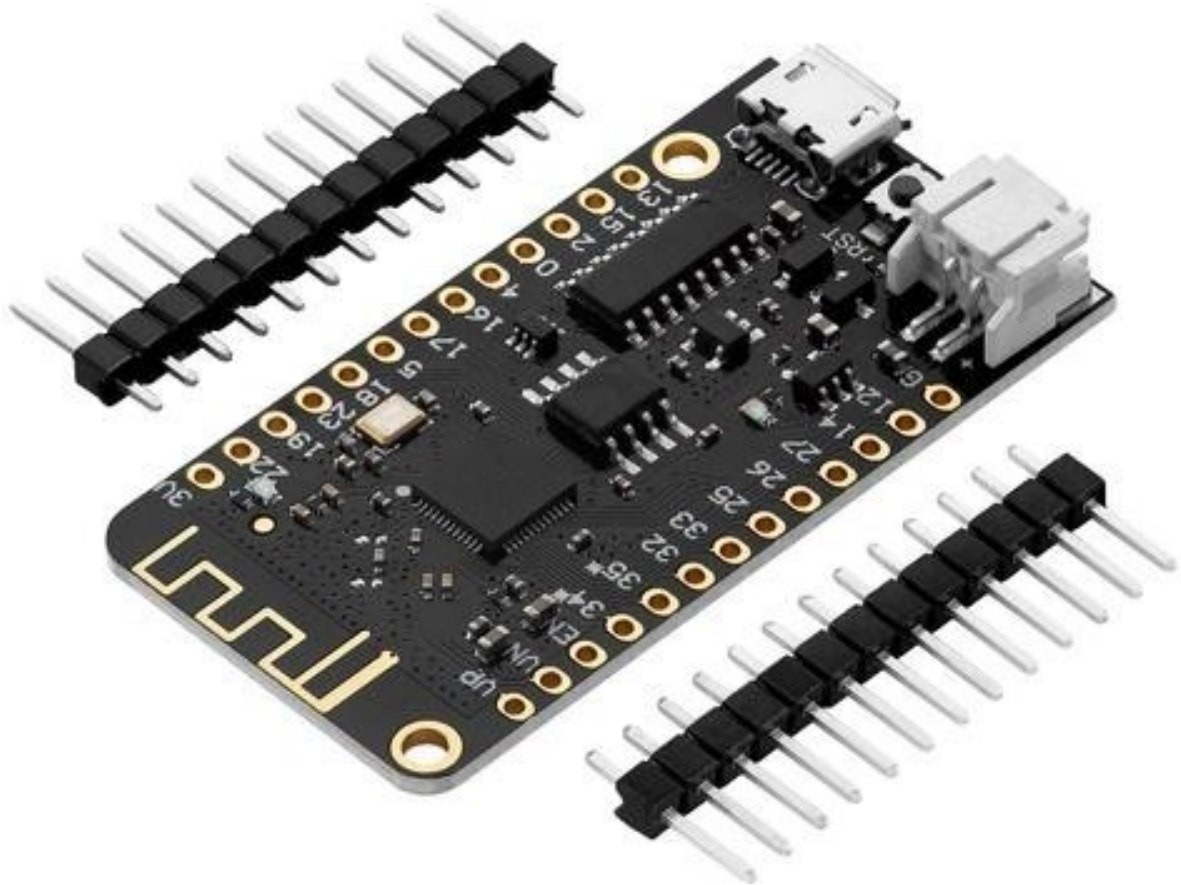


AZ-Delivery

Herzlich willkommen!

Vielen Dank, dass Sie sich für unser *AZ-Delivery ESP32 Lolin32* entschieden haben. Auf den folgenden Seiten erfahren Sie, wie Sie dieses handliche Gerät benutzen und einrichten können.

Viel Spaß!



Az-Delivery

Inhaltsübersicht

Einführung	3
Spezifikationen	4
Die Pinbelegung	5
Einrichten der Arduino IDE	6
Installieren des ESP32-Boards in der Arduino IDE	10
Beispiele skizzieren	12
Beispiel mit DS18B20	12
Beispiel mit BME280	18

Einführung

Lolin32 ist das ESP32-Entwicklungsboard. Wie andere ESP32-Boards verfügt es über einen 32-Bit-Dual-Core-Mikrocontroller, der mit 240 MHz arbeitet und über 4 MB Flash-Speicher verfügt. Es kann in einer Reihe von Sprachen programmiert werden, wie z.B. microPython, LUA, mod. C/C++ in der Arduino IDE programmiert werden, genau wie jeder andere Mikrocontroller. Es kann über Lithium-Polymer-Akkus mit einem maximalen Ladestrom von 500 mA oder über Micro-USB mit Strom versorgt werden. Die mit dem Board gelieferte Firmware akzeptiert standardmäßig AT-Befehle. Es ist mit WiFi und Bluetooth ausgestattet. Darüber hinaus enthält es einen FTDI FT231x, der USB in eine serielle Schnittstelle umwandelt und einem Computer die Programmierung und Kommunikation mit dem Mikrocontroller ermöglicht.

Dieses Modul verwendet nicht das WROOM32-Modul mit eingebautem Flash-Speicher, sondern hat den ESP32-Chip, den 4 MByte großen Flash-Speicher und die Antenne direkt auf der Platine eingebaut. Dadurch konnte die Breite um ein Rastermaß reduziert werden. So kann dieses Modul direkt auf einem einfachen Breadboard verwendet werden.

Ein weiterer Vorteil dieses Moduls ist der eingebaute Laderegler für 3,7V LiPo-Akkus. Das Modul hat eine JST 2,0 mm Buchse zum Anschluss eines Akkus. Allerdings hat das Modul auch Nachteile. Es gibt keinen 5V-Anschluss und die Stromversorgung kann nur über USB oder einen Akku erfolgen. Außerdem gibt es weniger Anschlüsse. Es gibt 19 Ein- und

Az-Delivery

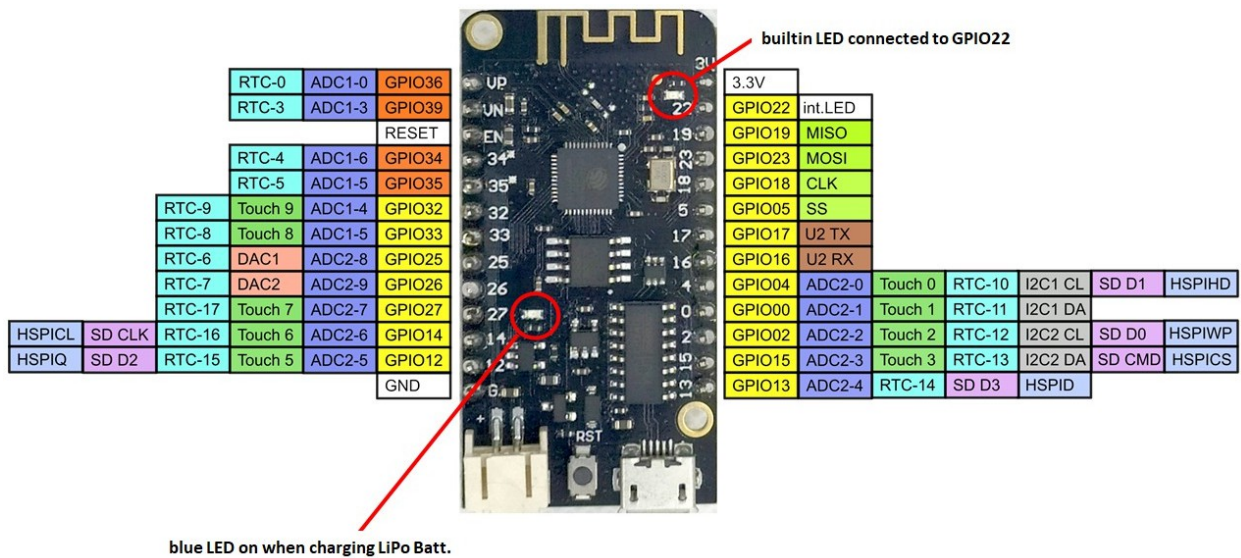
Ausgänge und 4 Eingänge. Eine auf dem Modul eingebaute LED kann über GPIO22 gesteuert werden.

Spezifikationen

Stromversorgung	2,2 V bis 3,6 V
Betriebsspannung	3.3V
Logischer Pegel	3.3V
Durchmesser der Befestigungslöcher	2 Löcher, 3mm Durchmesser
Integrierter 802.11 BGN WiFi-Transceiver	
Integriertes Dual-Mode-Bluetooth	
4MB Flash-Speicher	
Maximale serielle Baudrate	256000bps
Abmessungen	50x25mm (1.97x0.98in)

Die Pinbelegung

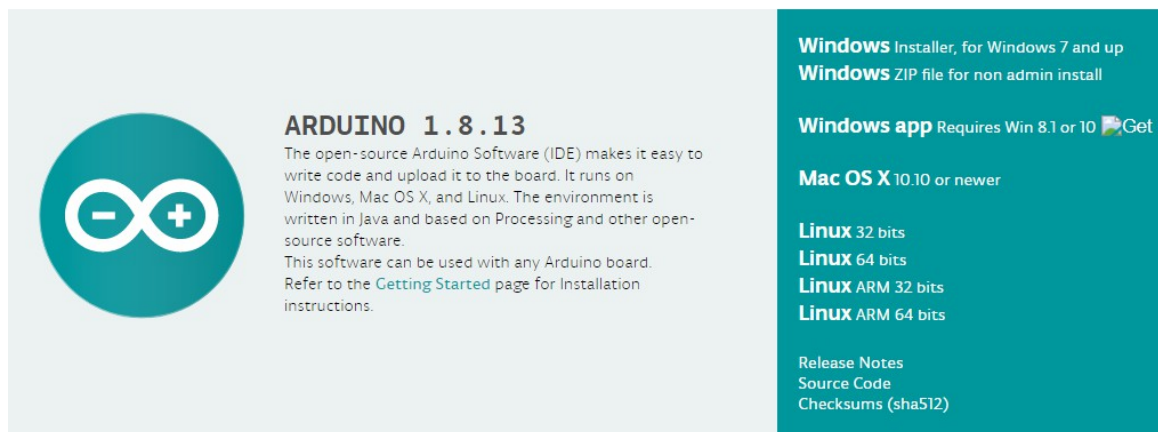
Der *ESP32 Lolin32* hat 26 Pins. Die Pinbelegung ist in der folgenden Abbildung dargestellt:




Einrichten der Arduino IDE

Wenn die Arduino IDE nicht installiert ist, folgen Sie dem [Link](#) und laden Sie die Installationsdatei für das Betriebssystem Ihrer Wahl herunter. Die für dieses eBook verwendete Arduino-IDE-Version ist **1.8.13**.

Download the Arduino IDE



The screenshot shows the Arduino IDE download page for version 1.8.13. On the left, there is a teal circular logo with a white infinity symbol containing a minus sign on the left and a plus sign on the right. To the right of the logo, the text reads: **ARDUINO 1.8.13**. Below this, it states: "The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions." On the right side of the page, there is a teal sidebar with the following options: **Windows** Installer, for Windows 7 and up; **Windows** ZIP file for non admin install; **Windows app** Requires Win 8.1 or 10  **Get**; **Mac OS X** 10.10 or newer; **Linux** 32 bits; **Linux** 64 bits; **Linux** ARM 32 bits; **Linux** ARM 64 bits; **Release Notes**; **Source Code**; **Checksums (sha512)**.

Windows-Benutzer doppelklicken auf die heruntergeladene .exe-Datei und folgen den Anweisungen im Installationsfenster.

AZ-Delivery

Für Linux-Benutzer laden Sie eine Datei mit der Erweiterung `.tar.xz` herunter, die entpackt werden muss. Nach dem Entpacken wechseln Sie in das entpackte Verzeichnis und öffnen das Terminal in diesem Verzeichnis. Zwei `.sh`-Skripte müssen ausgeführt werden, das erste heißt `arduino-linux-setup.sh` und das zweite heißt `install.sh`.

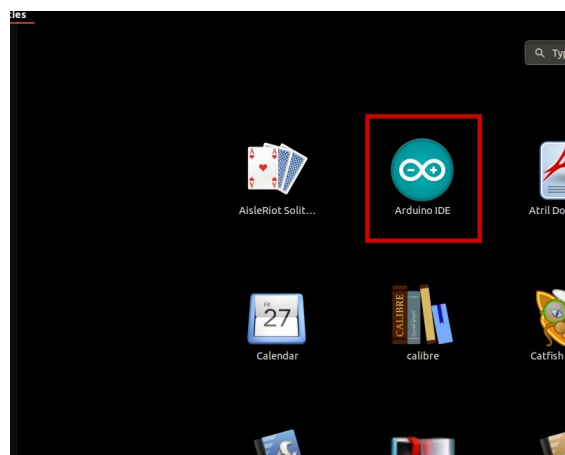
Um das erste Skript im Terminal auszuführen, öffnen Sie das Terminal im extrahierten Verzeichnis und führen Sie den folgenden Befehl aus:

```
sh arduino-linux-setup.sh benutzer_name
```

`user_name` - ist der Name eines Superusers im Linux-Betriebssystem. Beim Starten des Befehls muss ein Passwort für den Superuser eingegeben werden. Warten Sie ein paar Minuten, bis das Skript alles abgeschlossen hat.

Das zweite Skript mit dem Namen `install.sh` muss nach der Installation des ersten Skripts verwendet werden. Führen Sie den folgenden Befehl im Terminal (extrahiertes Verzeichnis) aus: `sh install.sh`

Nach der Installation dieser Skripte gehen Sie zu den *Alle Apps*, wo die *Arduino IDE* ist installiert.



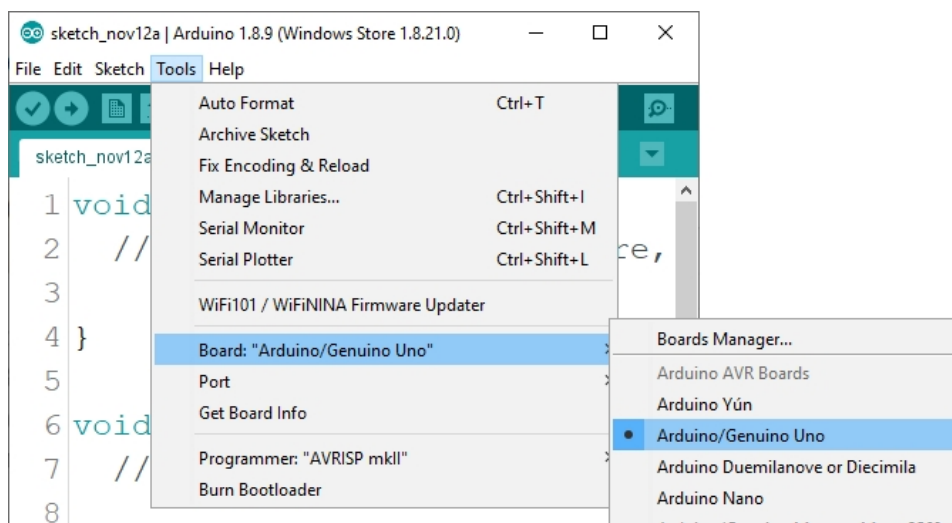
Az-Delivery

Auf fast allen Betriebssystemen ist ein Texteditor vorinstalliert (z. B. *Windows* mit *Notepad*, *Linux Ubuntu* mit *Gedit*, *Linux Raspbian* mit *Leafpad* usw.). Alle diese Texteditoren sind für den Zweck des Ebooks vollkommen ausreichend.

Als Nächstes müssen Sie überprüfen, ob Ihr PC ein Atmega328p-Board erkennen kann. Öffnen Sie die frisch installierte Arduino IDE und gehen Sie zu:

Tools > Board > {Ihr Boardname hier}

{Ihr Boardname hier} sollte der *Arduino/Genuino Uno* sein, wie er auf dem folgenden Bild zu sehen ist:



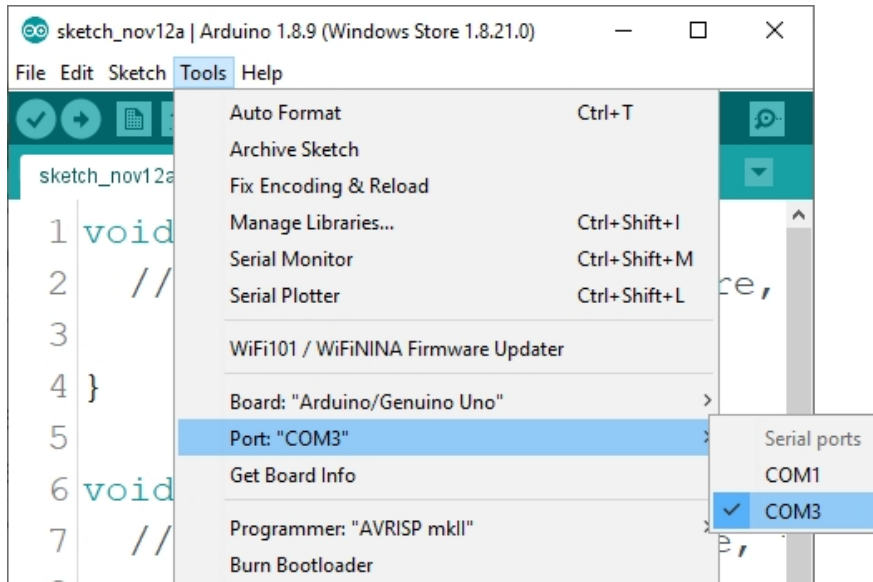
Der Port, an dem das Atmega328p-Board angeschlossen ist, muss ausgewählt werden. Gehe zu: *Tools > Port > {Portname hierher}* und wenn das Atmega328p-Board mit dem USB-Anschluss verbunden ist, kann der Name des Anschlusses im Dropdown-Menü auf dem vorherigen

Az-Delivery

Bild angezeigt werden.

Az-Delivery

Wenn die Arduino IDE unter Windows verwendet wird, lauten die Portnamen wie folgt:



Für Linux-Benutzer lautet der Anschlussname zum Beispiel `/dev/ttyUSBx`, wobei `x`

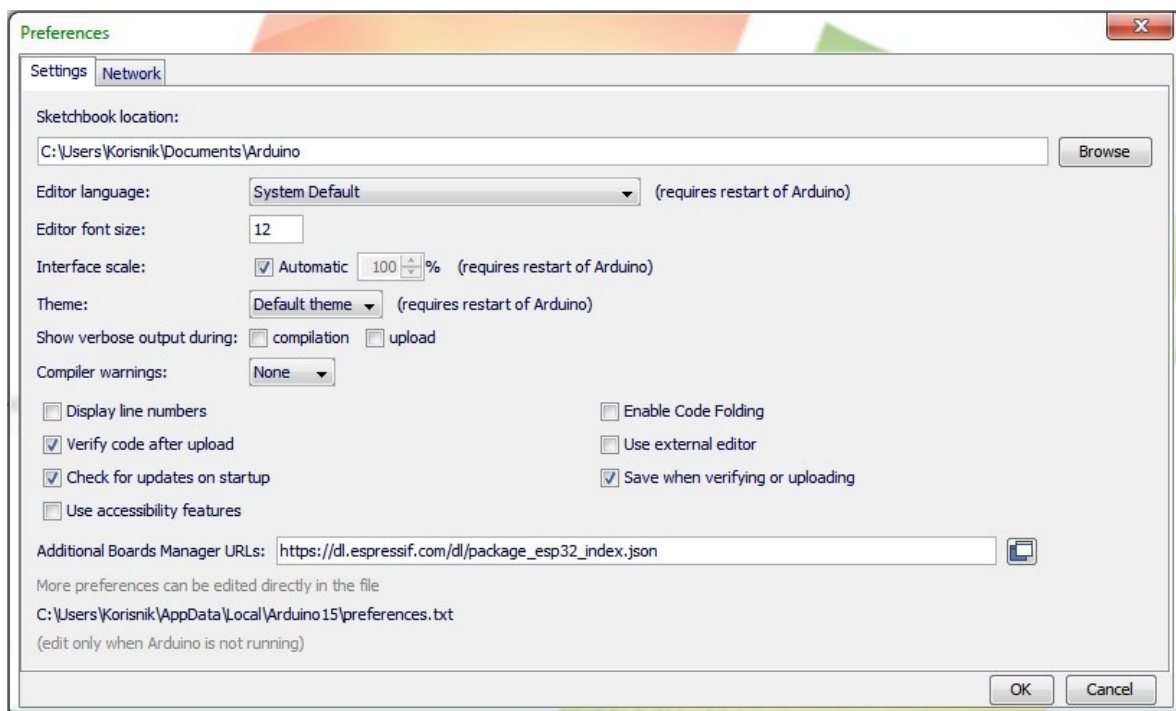
steht für eine ganze Zahl zwischen 0 und 9.

Installieren des ESP32-Boards in der Arduino IDE

Der USB-Treiber kann [hier](#) heruntergeladen werden.

Um das ESP32 Add-on in der Arduino IDE zu installieren, öffnen Sie zuerst die Arduino IDE und gehen dann zu

Datei > Voreinstellungen:



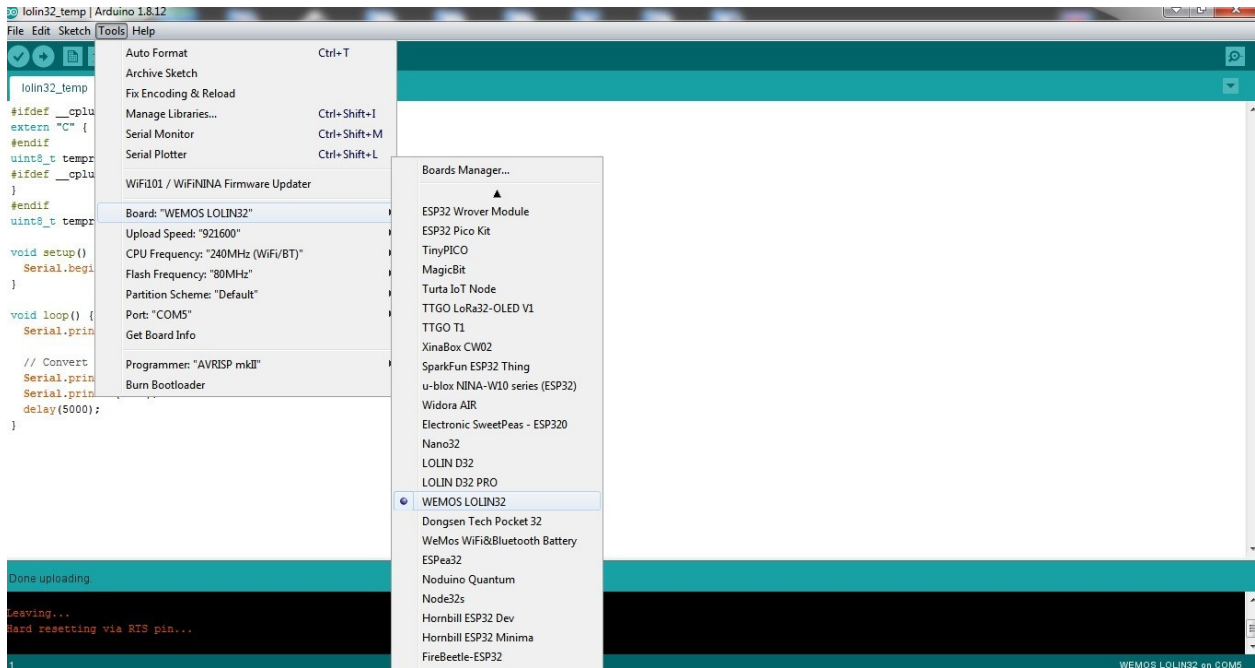
Unter *Zusätzliche Gremien Leiter URLs* Feld eingeben. die nächste URL:

https://dl.espressif.com/dl/package_esp32_index.json

Gehen Sie dann zu *Tools > Board > Boards Manager* und suchen Sie nach ESP32 und drücken Sie den Installationsknopf für *esp32 von Espressif Systems*. Nach ein paar Sekunden ist der ESP32 installiert.

Az-Delivery

Nun kann das Brett im Menü Bretter ausgewählt werden:



Das WEMOS LOLIN32-Board wird in diesem Projekt verwendet.

Jetzt ist das ESP32-Board bereit für das Hochladen von Code.

Beispiele skizzieren

Der ESP32 Lolin32 benötigt ein Micro-USB-Kabel, damit ein Computer den Mikrocontroller programmieren und mit ihm kommunizieren kann.

Beispiel mit DS18B20

```
// Messen Sie die Temperatur Ihres Pools oder Teichs mit dem DS18B20
// und melden Sie es in Ihrem lokalen WiFi

// Server
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include <ESPmDNS.h>
const char* SSID = " Ihre SSID ";
const char* PASSWORD = " Ihr Passwort ";
WebServer Server(80);

//Sensor
#include <OneWire.h>
#include <DallasTemperature.h>

// Datenleitung wird an Port 22 der Lolin32 angeschlossen, 4k7 Pull-Up-
Widerstand nicht vergessen
#define ONE_WIRE_BUS 22

// Einrichten einer OneWire-Instanz zur Kommunikation mit beliebigen
OneWire-Geräten OneWire oneWire(ONE_WIRE_BUS);
```

AZ-Delivery

```
// Übergeben Sie unsere oneWire-Referenz an Dallas Temperature.
DallasTemperature sensors(&oneWire);

void setup() {

// Server
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.begin(SSID, PASSWORD);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.printf("Verbunden mit %s.\n",
  SSID);
  Serial.printf("IP-Adresse: %s.\n", WiFi.localIP().toString().c_str());

  if (MDNS.begin("Lolin32")) {
    Serial.println("MDNS-Responder gestartet.");
  }

  server.on("/", handleRoot);

  server.onNotFound(handleNotFound);

  server.begin();
  Serial.println("HTTP-Server gestartet.");
```

Az-Delivery

```
// Sensor
Serial.println("Dallas DS18B20 Temperatur ");

// Starten Sie die
Bibliothek sensors.begin();
}

void loop() {
// Sensor

// Aufruf von sensors.requestTemperatures() zur Ausgabe einer globalen
Temperatur
// Anfrage an alle Geräte auf dem Bus
Serial.print("Temperaturen abfragen...");
sensors.requestTemperatures(); // Senden Sie den Befehl zum Abrufen der
Temperaturen Serial.println("DONE");

// Nachdem wir die Temperaturen erhalten haben, können wir sie hier
ausdrucken.
// Wir verwenden die Funktion ByIndex und holen als Beispiel nur die
Temperatur des ersten Sensors ab.
Serial.print("Die Temperatur für das Gerät 1 (Index 0) ist:
"); float tempC = sensors.getTempCByIndex(0);
Serial.println(tempC);
delay(5000);

// Server
server.handleClient();
server.on("/Sensor", HTTP_GET, []() {
String message = "<!doctype html>\n";
message += "<html>\n";
message += " <head>\n";
message +=      <title>ESP32Thermometer / Hygrometer</title>\n";
message += " </head>\n";
message += " <body>\n";
message +=      <h1>Temperatur: ";
message += String(sensors.getTempCByIndex(0), 2);
message += " &#8451;</h1>\n";
```

Az-Delivery

```
message += " </body>\n";
```

Az-Delivery

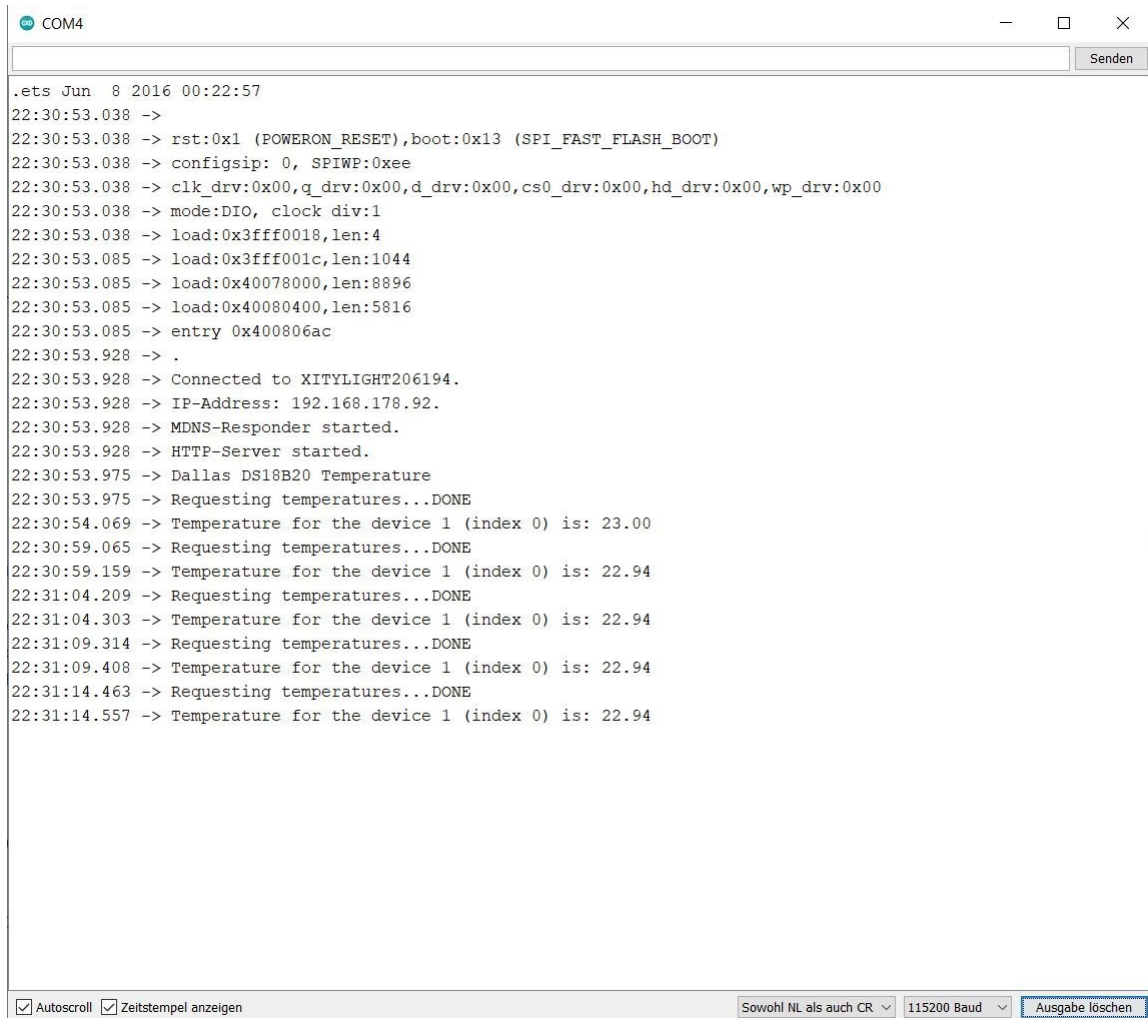
```
        message += "</html>\n";
        server.send(200, "text/html", message);
    });
}

void handleRoot() {
    server.send(200, "text/plain", "Hier ist Ihr Lolin32!");
}

void handleNotFound() {
    String message = "path ";
    message += server.uri();
    message += " not found.\n";
    server.send(404, "text/plain", Nachricht);
}
```

Az-Delivery

Die Ausgabe sollte wie in der folgenden Abbildung aussehen:



The screenshot shows a serial terminal window titled 'COM4'. The terminal displays the following output:

```
.ets Jun  8 2016 00:22:57
22:30:53.038 ->
22:30:53.038 -> rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
22:30:53.038 -> configsip: 0, SPIWP:0xee
22:30:53.038 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
22:30:53.038 -> mode:DIO, clock div:1
22:30:53.038 -> load:0x3fff0018,len:4
22:30:53.085 -> load:0x3fff001c,len:1044
22:30:53.085 -> load:0x40078000,len:8896
22:30:53.085 -> load:0x40080400,len:5816
22:30:53.085 -> entry 0x400806ac
22:30:53.928 -> .
22:30:53.928 -> Connected to XITYLIGHT206194.
22:30:53.928 -> IP-Address: 192.168.178.92.
22:30:53.928 -> MDNS-Responder started.
22:30:53.928 -> HTTP-Server started.
22:30:53.975 -> Dallas DS18B20 Temperature
22:30:53.975 -> Requesting temperatures...DONE
22:30:54.069 -> Temperature for the device 1 (index 0) is: 23.00
22:30:59.065 -> Requesting temperatures...DONE
22:30:59.159 -> Temperature for the device 1 (index 0) is: 22.94
22:31:04.209 -> Requesting temperatures...DONE
22:31:04.303 -> Temperature for the device 1 (index 0) is: 22.94
22:31:09.314 -> Requesting temperatures...DONE
22:31:09.408 -> Temperature for the device 1 (index 0) is: 22.94
22:31:14.463 -> Requesting temperatures...DONE
22:31:14.557 -> Temperature for the device 1 (index 0) is: 22.94
```

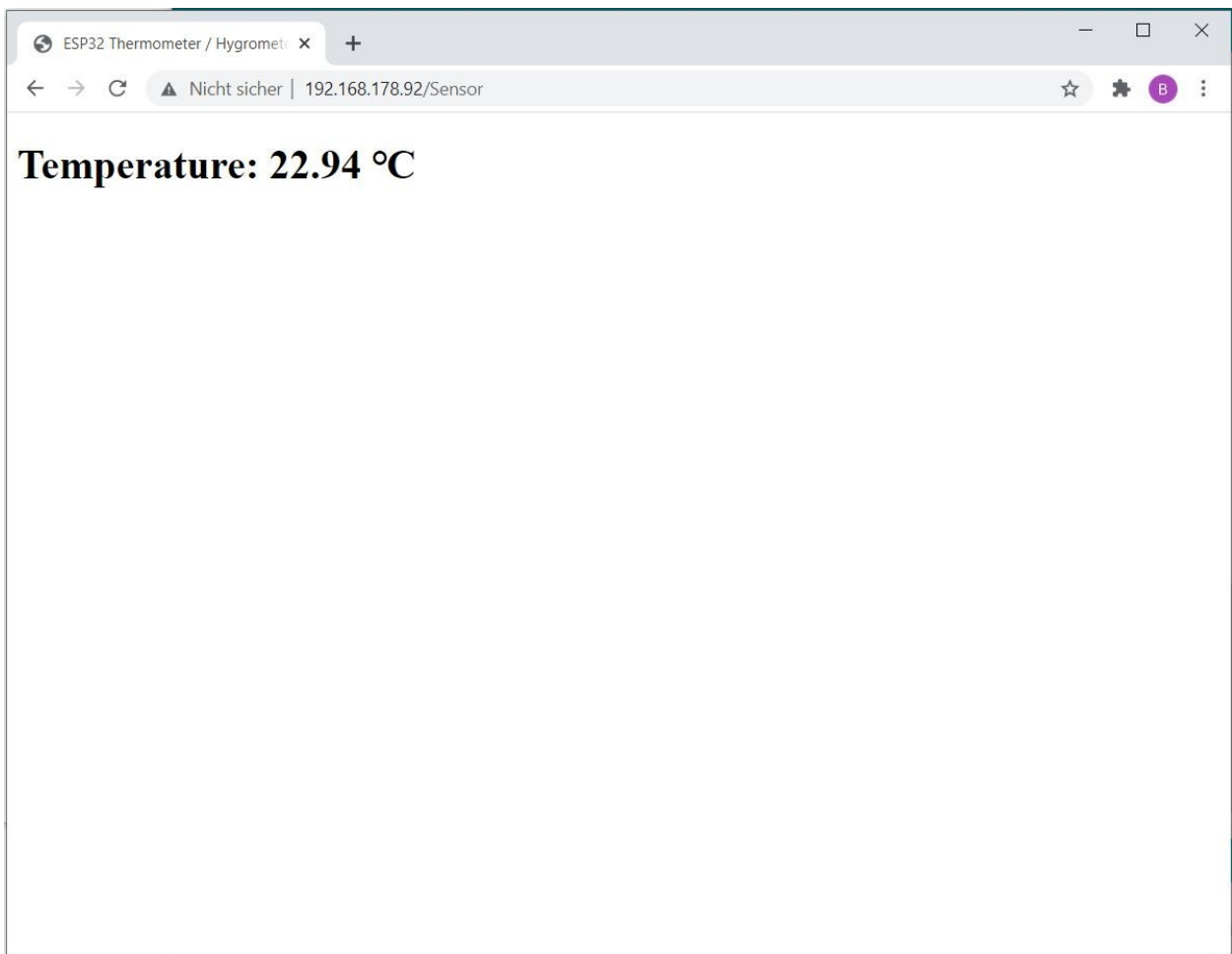
The terminal window includes a 'Senden' button at the top right and a status bar at the bottom with the following options: Autoscroll, Zeitstempel anzeigen, Sowohl NL als auch CR, 115200 Baud, and a button labeled 'Ausgabe löschen'.

Hinweis: Stellen Sie die Baudrate auf 115200 ein.

Az-Delivery

WiFi funktioniert auch, wenn das Micro-USB-Kabel nicht angeschlossen ist. Geben Sie einfach die IP-Adresse in Ihrem Browser ein, gefolgt von /Sensor.

Wenn keine Verbindung zum Computer besteht, erhält der Serial Monitor der Arduino IDE keine Informationen.



Az-Delivery

Beispiel mit BME280

```
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>

#define BME_SCK          18//verwendet diese Pins für
Hardware-SPI #define BME_MISO 19
#define BME_MOSI 23
#define BME_CS 5

Adafruit_BME280 bme; // I2C
//Adafruit_BME280 bme(BME_CS); // Hardware SPI
//Adafruit_BME280 bme(BME_CS, BME_MOSI, BME_MISO, BME_SCK); // Software SPI

unsigned long delayTime;

void setup() {
  Wire.begin();
  Serial.begin(9600);
  while(!Serial); // Zeit, die serielle Schnittstelle zum Laufen zu
  bringen
  Serial.println(F("BME280 test"));
  // Standardeinstellungen
  unsigned status;
```

Az-Delivery

```
Wire.begin(26,27); //Fügen Sie diese Zeile für I2C mit
SDA=26 und SCL=27 ein

status = bme.begin(); //Standard ist 0x77;
andernfalls bme.begin(0x76)

if (!status) {

    Serial.println("Konnte keinen gültigen BME280-Sensor finden,
Verkabelung, Adresse, Sensor-ID prüfen!");

    Serial.print("SensorID war: 0x"); Serial.println(bme.sensorID(),16);

    Serial.print(" ID von 0xFF bedeutet wahrscheinlich eine falsche
Adresse, einen BMP 180 oder BMP 085\n");

    Serial.print(" ID von 0x56-0x58 steht für einen BMP
280,\n"); Serial.print(" ID von 0x60 steht für einen BME
280.\n"); Serial.print(" ID von 0x61 steht für einen BME
680.\n"); while (1) delay(10);

}

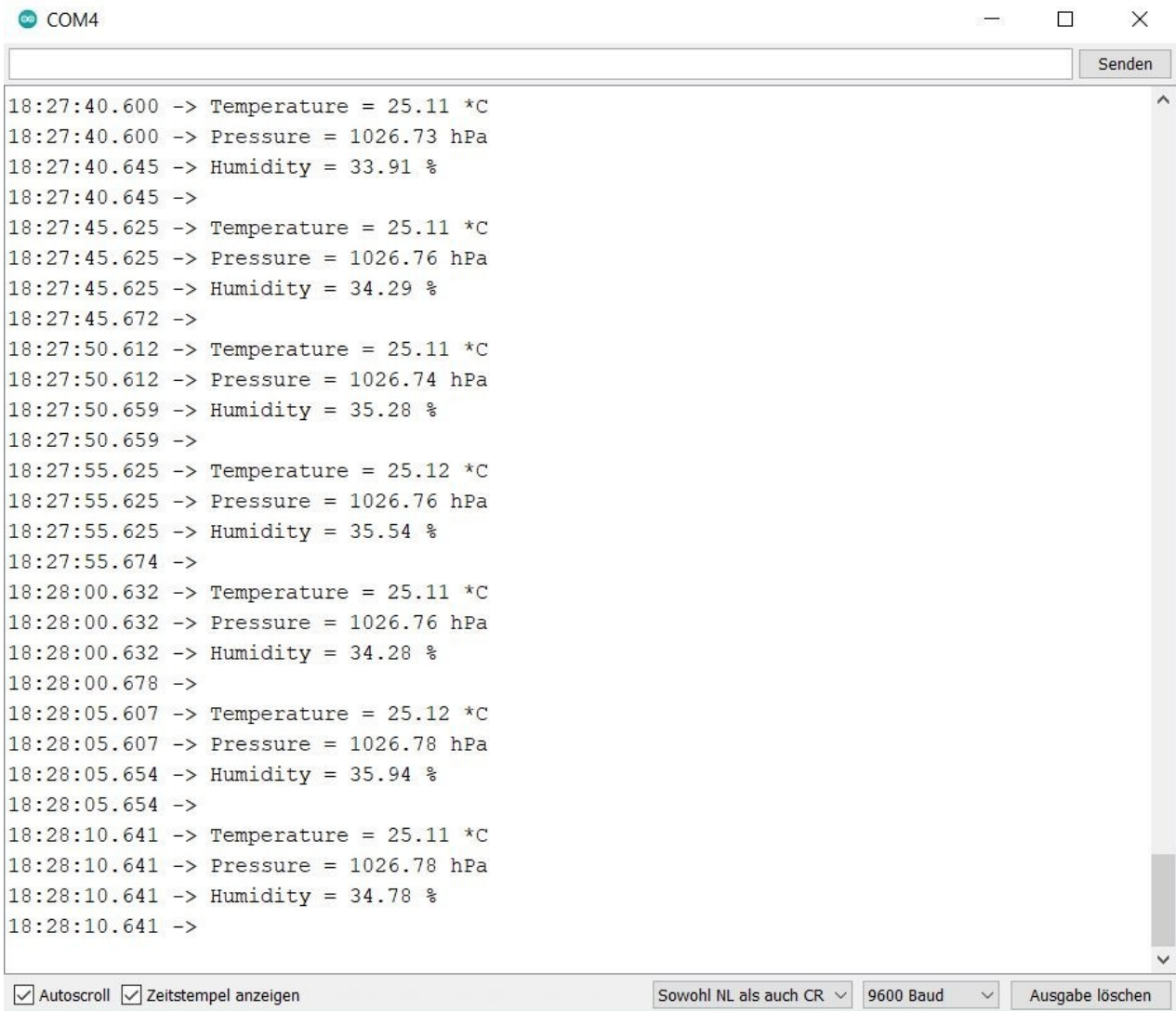
Serial.println("-- Standardtest --");
delayTime = 5000;
Serial.println();
}

void loop() {
    printValues();
    delay(delayTime);
}
```

Az-Delivery

```
void printValues() { Serial.print("Temperatur  
= "); Serial.print(bme.readTemperature());  
Serial.println(" *C"); Serial.print("Druck  
= "); Serial.print(bme.readPressure() /  
100.0F); Serial.println(" hPa");  
Serial.print("Feuchte = ");  
Serial.print(bme.readHumidity());  
Serial.println(" %");  
Serial.println();  
}
```

Az-Delivery



The screenshot shows a terminal window titled 'COM4' with a 'Senden' button in the top right corner. The window contains a list of sensor readings with timestamps. The data is as follows:

Timestamp	Temperature (*C)	Pressure (hPa)	Humidity (%)
18:27:40.600	25.11	1026.73	33.91
18:27:40.645			33.91
18:27:45.625	25.11	1026.76	34.29
18:27:45.672			
18:27:50.612	25.11	1026.74	35.28
18:27:50.659			35.28
18:27:55.625	25.12	1026.76	35.54
18:27:55.674			
18:28:00.632	25.11	1026.76	34.28
18:28:00.678			
18:28:05.607	25.12	1026.78	35.94
18:28:05.654			
18:28:10.641	25.11	1026.78	34.78
18:28:10.641			

At the bottom of the window, there are several control options: Autoscroll, Zeitstempel anzeigen, a dropdown menu set to 'Sowohl NL als auch CR', a dropdown menu set to '9600 Baud', and a button labeled 'Ausgabe löschen'.

Hinweis: Dies funktioniert für I2C auf Pins 26 und 27 und könnte auch (mit geringfügigen Änderungen) für SPI verwendet werden. Skizze und Bibliotheken sind von Adafruit (US).

AZ-Delivery

Jetzt ist es an der Zeit, zu lernen und eigene Projekte zu erstellen. Das können Sie mit Hilfe vieler Beispielskripte und anderer Anleitungen tun, die Sie im Internet finden können.

Wenn Sie auf der Suche nach hochwertiger Mikroelektronik und Zubehör sind, sind Sie bei der AZ-Delivery Vertriebs GmbH an der richtigen Adresse. Sie erhalten zahlreiche Anwendungsbeispiele, vollständige Installationsanleitungen, eBooks, Bibliotheken und Unterstützung durch unsere technischen Experten.

<https://az-delivery.de> Viel

Spaß!

Impressum

<https://az-delivery.de/pages/about-us>