

IOT XMAS WORKSHOP

10.12.2021

1 Setup

1.1 Arduino IDE – Installation

Zunächst muss die verwendete IDE, welche wir für Entwicklung verwenden, installiert werden. Die Arduino IDE steht unter <https://www.arduino.cc/> für alle gängigen Betriebssysteme zum download zur Verfügung.

1.2 Benötigte Bibliotheken

Um den Programmcode für unsere Christbaumkugel zu schreiben, werden einige externe Bibliotheken benötigt, welche mittels „include“ eingebunden werden müssen.

Folgende Bibliotheken sind bereits durch Arduino selbst oder durch das „esp32“-Paket mitgeliefert:

- BLEDevice: Klasse für ein BLE Device. Die Grundlage für die BLE Kommunikation.
- BLEServer: Klasse für einen BLE Server. Hierauf kann sich später das Smartphone verbinden.
- BLEUtils: Allgemeine Funktionen für BLE.

Um die LED-Stripes anzusteuern, wird zusätzlich die „NeoPixel“-Bibliothek von Adafruit benötigt. Um sie zu installieren, muss diese zunächst aus den Git-Repository von Adafruit heruntergeladen werden: https://github.com/adafruit/Adafruit_NeoPixel. Das ZIP-Paket kann nun an einem beliebigen Ort auf dem Computer ablegen werden. Als nächstes muss das Paket in die Arduino IDE eingebunden werden. Dies funktioniert wie folgt: „Sketch -> Bibliothek einbinden -> ZIP-Bibliothek hinzufügen...“ anschließend das heruntergeladene Paket auswählen.

Das Paket liefert zusätzlich zur Bibliothek einige Beispiele, die in der IDE unter „Datei -> Beispiele -> Adafruit NeoPixel“ zu finden sind.

1.3 Arduino IDE – ESP32 Setup

Um den ESP32 über euren Computer zu programmieren, müsst ihr es per Mikro-USB anschließen. Das Board wird über einen virtuellen COM Port angesprochen. Falls der COM-Port nicht sichtbar ist, muss zusätzlich noch der passende Treiber (CP210x Windows Drivers) installiert werden. Dieser ist hier zu finden:

<https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>

Zunächst ist es wichtig, alle benötigten Boardkonfigurationen und Bibliotheken einzubinden, um den ESP32 über die IDE nutzen zu können. Dazu geht man unter „Datei -> Voreinstellungen“ und fügt eine „zusätzliche Boardverwalter-URL“ ein: https://dl.espressif.com/dl/package_esp32_index.json

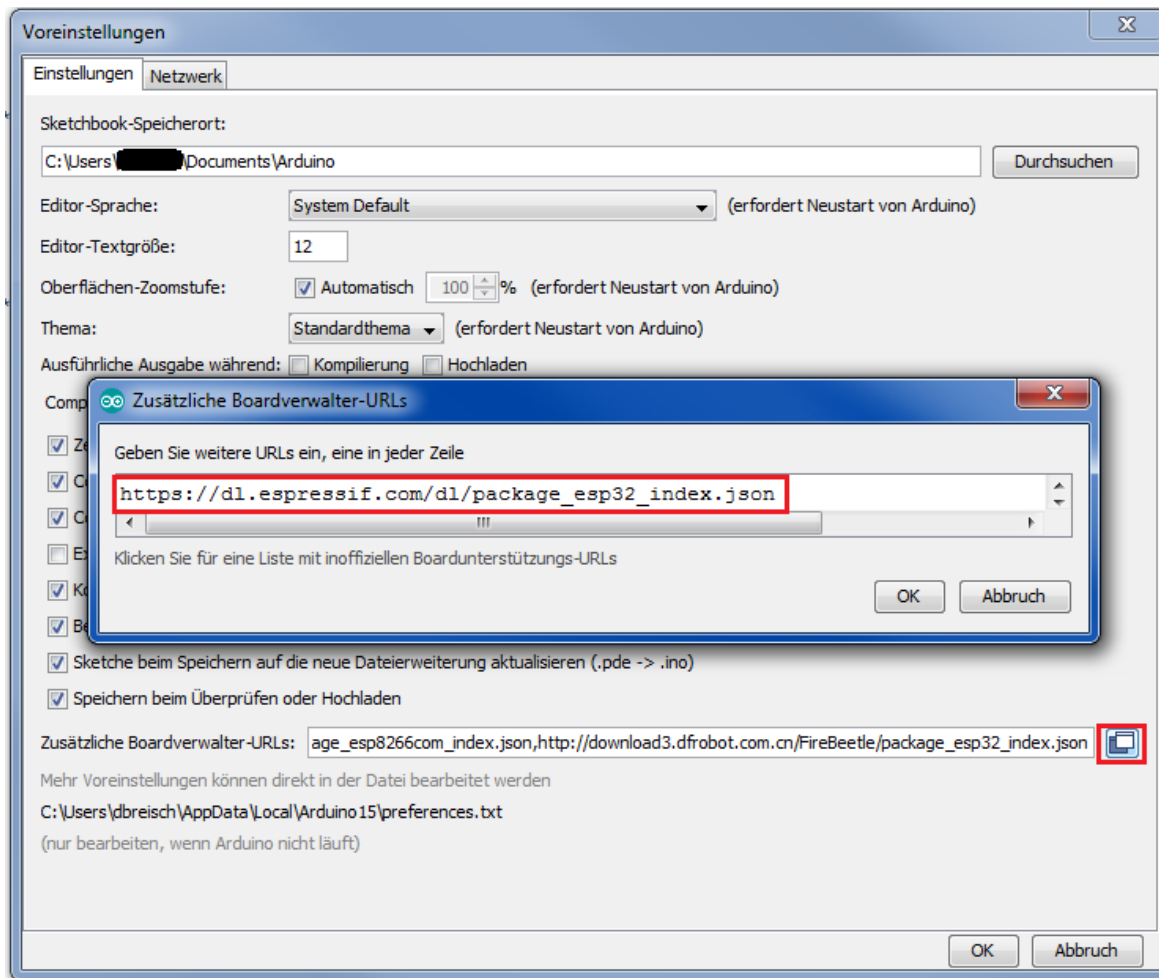


Abbildung 1 Boardverwalter-URL konfigurieren

Nun kann unter „Werkzeuge -> Board -> Boardverwalter“ das „esp32“ Paket installiert werden.

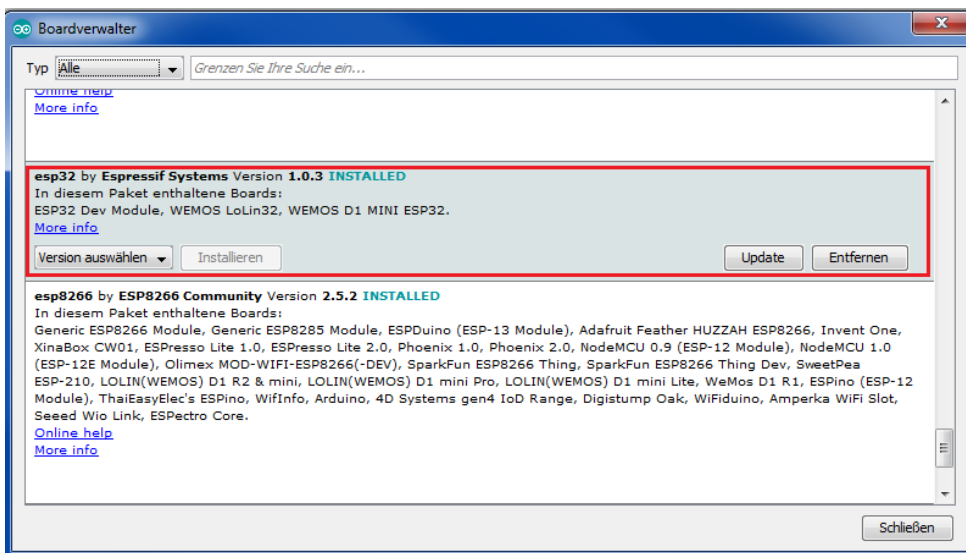


Abbildung 2 ESP32 - Pakete installieren

Damit nun die IDE weiß für welches Board tatsächlich entwickelt wird, gibt man abschließend unter „Werkzeuge“ die korrekte Konfiguration an. Wichtig hierbei ist zudem das Auswählen des richtigen virtuelle COM Ports, an dem das Board angeschlossen ist.

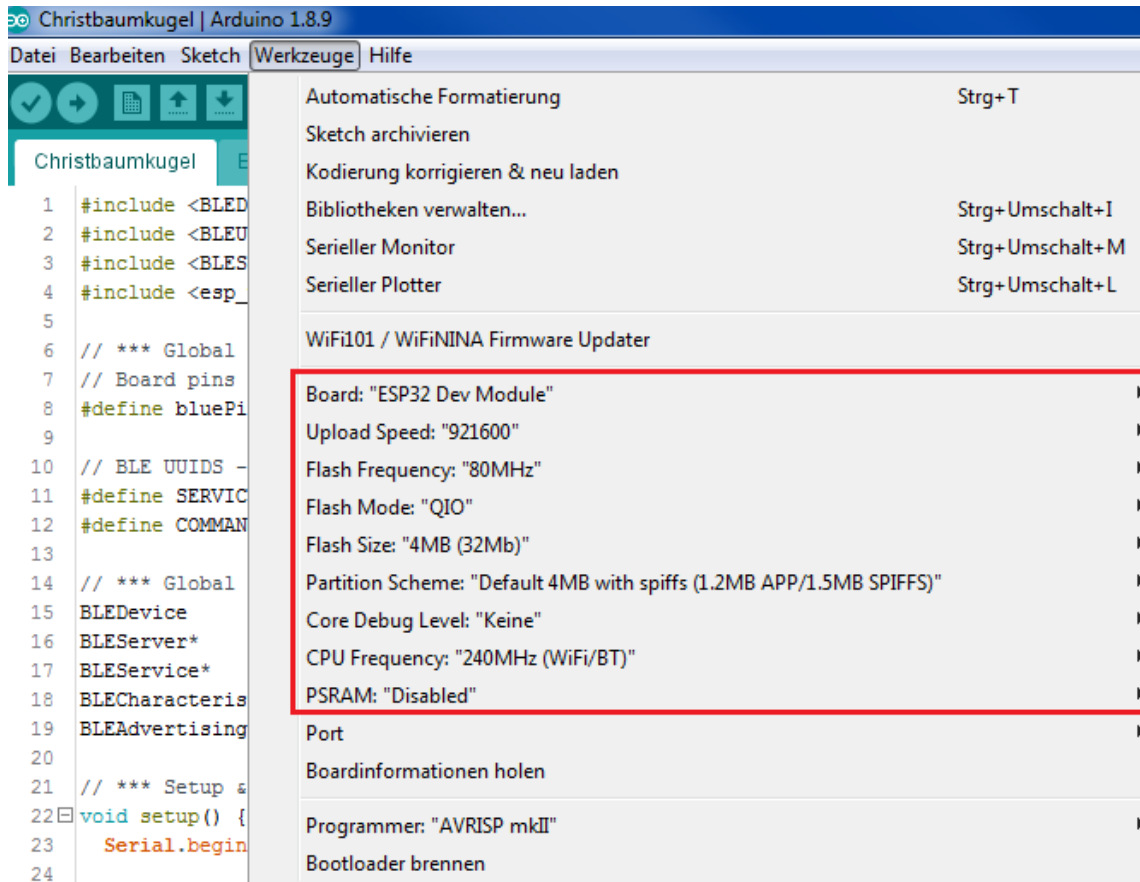


Abbildung 3 Boradkonfiguration

Will man den Sketch auf das Board flashen, klickt man auf den Pfeil „Hochladen“ (oben links in der IDE). Daraufhin wird der Sketch kompiliert und das Flashen beginnt. Will man lediglich kompilieren, klickt man auf den Haken „Überprüfen“.

Während des Flashvorgangs wird am unteren Bildschirm der aktuelle Vorgang in der Konsole ausgegeben. Sobald dort „Connecting...“ steht, muss in einigen Fällen der „Boot“ Knopf des Boards gedrückt werden, bis die Übertragung startet. Danach kann der Knopf losgelassen werden.

Für Linux Nutzer:

Unter Linux muss zunächst die Versionskontrollsoftware Git und das Python-Modul Serial mittels

```
sudo apt install git python-serial
```

installiert werden.

1.4 Sag Hallo, zur Welt!

Sobald dein Xmas Tech Paket angekommen ist, kannst du überprüfen, ob dein Setup vollständig eingerichtet ist.

Zum Testen der Konfiguration kann ein kleiner „Hello-World“ Sketch implementiert werden. Hierzu muss über „Datei -> Neu“ ein neuer Sketch angelegt werden. Mithilfe des folgenden Sketch kann eine serielle Ausgabe geschrieben werden, welche im Seriellen Monitor verfolgt werden kann. Dabei ist darauf zu achten, dass die Baudrate im Seriellen Monitor mit der im Sketch hinterlegten Baudrate (115200) übereinstimmt.

```
void setup() {  
  Serial.begin(115200);  
  Serial.println("Hello ESP32 World!");  
}  
  
void loop() {  
  Serial.println("Hello");  
  delay(500);  
}
```

Abbildung 4 Hello-World Beispiel

Eine praktische Debugmöglichkeit bietet der „Serielle Monitor“, der in der rechten oberen Ecke zu finden ist. Hier werden alle seriellen Ausgaben des Controllers angezeigt, die der Entwickler erstellt hat.

Der Serial.begin()- Befehl aktiviert die serielle Schnittstelle mit der wir über den Computer kommunizieren können. Der Wert "115200" ist die Übertragungsrate. Man sagt auch "115200 Baud". Wichtig ist, dass ihre diese auch im seriellen Monitor auswählt.

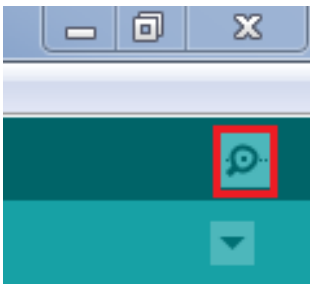


Abbildung 5 Serieller Monitor

1.5 Genau die richtige Debugging App für unsere Christbaumkugel – nRFConnect

Um die Bluetooth-Verbindung von Smartphone aus zu testen kann die App nRFConnect genutzt werden. Die App ist für Android sowie iOS kostenlos verfügbar:

- <https://apps.apple.com/us/app/nrf-connect/id1054362403>
- https://play.google.com/store/apps/details?id=no.nordicsemi.android.mcp&hl=en_US

Kontakt

T-Systems on site Services GmbH
Fasanenweg 5
70771 Leinfelden-Echterdingen

Christian Pitters
E-Mail: Christian.pitters@t-systems.com