Bedienungsanleitung

Fischertechnik Controller

CODY++

MAXI und MINI

Wintersemester 2018/2019

Inhaltsverzeichnis

[Einleitung 3](#_Toc530160985)

[Programmierung und Vorbereitung 3](#_Toc530160986)

[Beschreiben des ESP32 mit der Arduino IDE 3](#_Toc530160987)

[Benötigte Bibliotheken und Programme 4](#_Toc530160988)

[Beschreiben des ESP32 4](#_Toc530160989)

[1)Verbindung herstellen: 4](#_Toc530160990)

[2)Programmupload 7](#_Toc530160991)

[3) Cody++ hochladen 7](#_Toc530160992)

# Einleitung

In dieser Bedienungsanleitung werden die Eigenschaften und Funktionen der Fischertechnikcontroller näher erläutert.

# Programmierung und Vorbereitung

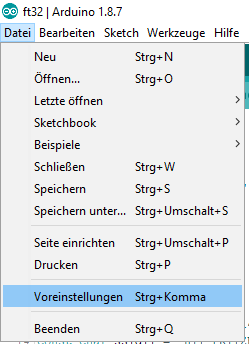
Die für Fischertechnik entwickelten Controller, die auf der Basis eines EP32 arbeiten, werden trotz unterschiedlicher Bauart vom selben Programm gesteuert.

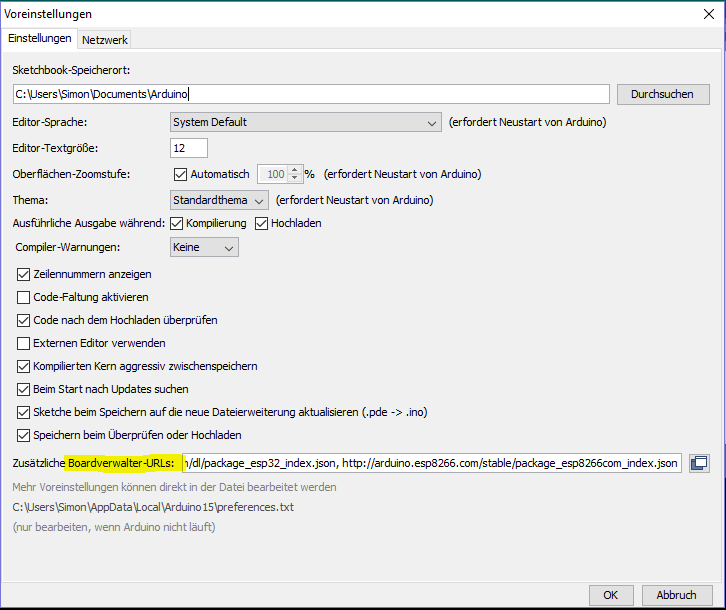
## Beschreiben des ESP32 mit der Arduino IDE

Der ESP32 kann mit Hilfe der Arduino IDE 1.8.7 auf den Betriebssystemen Windows und Linux mit dem Programmcode beschrieben bzw. das bestehende Programm editiert werden. Hierzu ist es nötig das ESP32 - Board in die Programmierumgebung der Arduino – IDE einzubinden. Die unten abgebildeten Links können in der geöffneten Programmierumgebung unter Datei (File)-> Voreinstellungen(Preferences) eingegeben werden.

Arduino IDE:

* <https://www.arduino.cc/en/Main/Software>





Boardverwalter-URL´s:

* <https://dl.espressif.com/dl/package_esp32_index.json>,

## Benötigte Bibliotheken und Programme

Für die Verwendung auf dem Controller verbauten Oled Displays ist es nötig die Arduino-IDE um die Bibliotheken von Adafruid1306 und AdafruidGFX zu erweitern.

Librarys:

* <https://github.com/adafruit/Adafruit_SSD1306>
* <https://github.com/adafruit/Adafruit-GFX-Library>
* <https://github.com/sparkfun/SparkFun_SX1509_Arduino_Library>

Diese können auch über Sketch🡪Bibliothek einbinden🡪Bibliotheken verwalten einfach gesucht und installiert werden.

Python:

* <https://www.python.org/downloads/>

Es eignet sich Python Version 2.7.15 sehr gut für den Ansatz. Ideal ist es, wenn man esptools zusätzlich von Python aus installiert, da dann alle zugehörigen packages für Python heruntergeladen werden. Die Installation erfolgt über „pip install esptool“ aus der Command Line.

## Beschreiben des ESP32

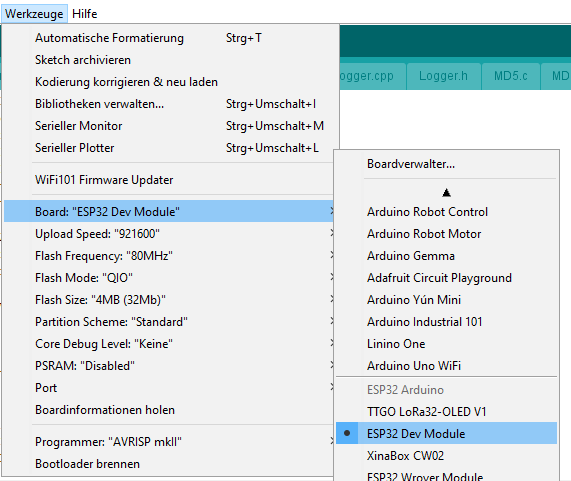
### Verbindung herstellen:

Nach erfolgreichem Einrichten der Arduino-IDE kann der Controller mit Hilfe eines USB-Kabels (USB Typ A auf USB Micro-B) beschrieben werden. Zurnächst muss der Controller an der vorhandenen Aussparung, an der sich die Mico-USB-Typ-B Buchse befindet mit dem Kabel an die USB Buchse des PCs angeschlossen werden. !! Achtung: Zum Zeitpunkt des Beschreibens darf keine Buchse des Controllers an Fischertechnik Bauteile angeschlossen sein!! Die zum Beschreiben des Controllers benötigte Spannungsversorgung erfolgt über den Computer.

Um den Controller erfolgreich zu beschreiben ist es nötig vor dem Upload der Software das korrekte Board und dessen Anschluss am PC einzustellen. Dazu muss das ESP Dev Module unter :

Werkzeuge-> Board: ->ESP Dev Module

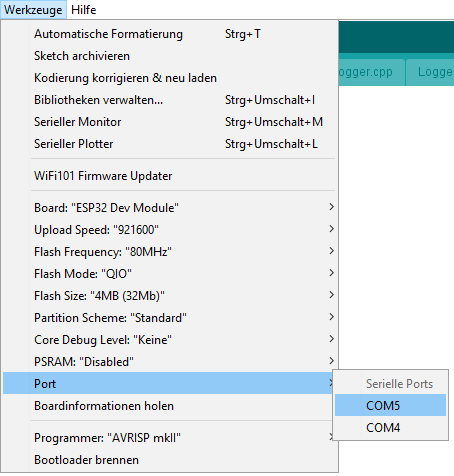
ausgewählt werden.



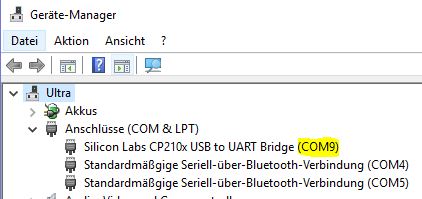
Anschließend muss der Richtige USB Port zur Kommunikation unter:

Werkzeuge-> Port->COM X (X steht für die entsprechende Portnummer am gegenwärtigen PC).

ausgewählt werden.

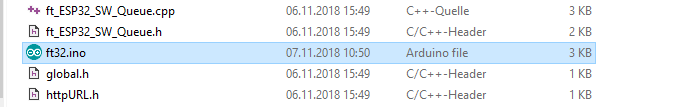


Ist der Port nicht bekannt kann dieser im Geräte-Manager (OS: MS Windows) ermittelt werden.



### 2)Programmupload

Zum Beschreiben des Controllers muss jetzt die sich im Programm befindende ft32.ino Datei in der Arduino- IDE geöffnet werden.



Falls ein anderes Netzwerk als „SSID: HIT-FRITZBOX-7490“ verwendet werden soll, muss die ft32.ino Datei an folgenden Stellen editiert werden:



In Zeile 13 muss das vorherige Passwort durch das des neuen Netzwerks ersetzt werden.

In Zeile 14 muss die neue Netzkennung (im W-Lan sichtbarer Name) eingetragen werden.

Nun kann in der Arduino-Programmierumgebung das Programm mit dem Button Hochladen auf den Controller geladen werden.

### 3) Cody++ hochladen

Dies erfolgt wie bereits hier beschrieben:

<https://github.com/hama1067/ft32#writing-to-spiffs-memory-in-linux>

Wichtig hierbei ist, dass die Version von mkspiffs.exe 0.2.3 ist. Die Version von esptool.py muss 2.5.0 oder höher sein. Sonst kommt es zu Problemen.

Nach dem Upload der Software sollte der Controller vom Computer getrennt werden und anschließend wieder verbunden werden. Durch das Trennen vom Computer erfolgt ein Reset. Anschießend kann auf dem Display des Controllers überprüft werden ob dieser erfolgreich mit dem eingerichteten Netzwerk verbunden ist oder ob dies Fehlgeschlagen ist.

Alternativ kann der Weiterentwickelte [Spiffs Manager](http://www.hit-karlsruhe.de/hit-info/info-ws18/FT_ESP32_HWK/0407Quellcode.html) verwendet werden, der sich Ports etc. selber sucht. In der Datei ist eine eigene ReadMe enthalten.