

Setup-Dokument

Bevor Sie mit den Aufgaben starten, lesen Sie dieses Dokument gewissenhaft durch und installieren Sie alle nötigen Programme und Pakete. Im Lernraum befinden sich unter anderem ein ZIP-Ordner mit den benötigten Dateien zum Bearbeiten der Aufgaben, sowie eine entweder in Deutsch oder Englisch verfasste Aufgabenstellung. Achten Sie darauf, dass sie nur den ZIP-Ordner für ihren Zeitraum nutzen.

Das Semesterprojekt basiert auf dem Microcontroller-Board (vor Beginn des Projektes abzuholen) und verschiedenen Code-Dateien.

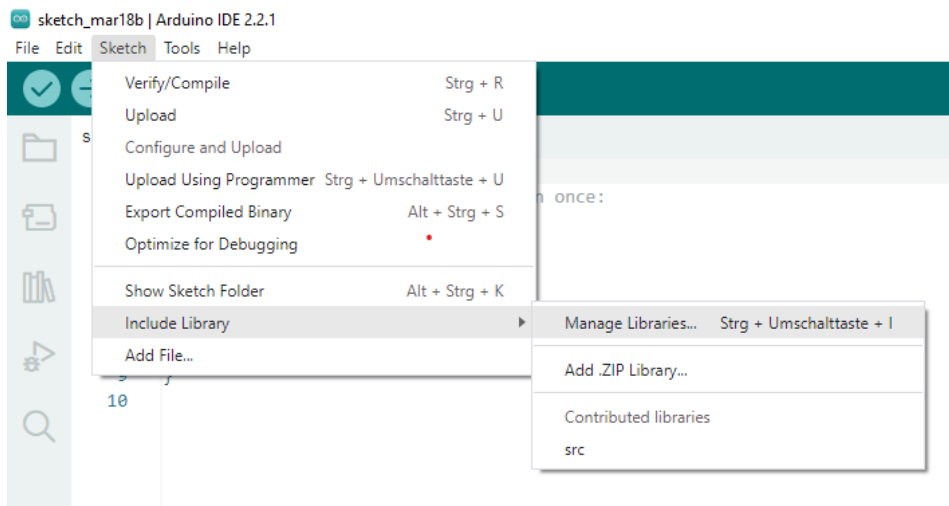
Im Folgenden wird Ihnen eine Anleitung an die Hand gelegt, wie Sie diese Programme installieren und das Set-Up gestalten, um einen reibungslosen Ablauf zu ermöglichen. Die Durchführung dieses Projekts wurde von uns mit Python 3.10 und Arduino IDE 2.3.3, sowie mit Ubuntu 24.04, Windows 11 und Windows 10 getestet.

Arduino-IDE:

Zunächst benötigen Sie die IDE des Arduinos selbst. Laden Sie bitte die aktuelle Version dieser herunter. Download-Links für verschieden Betriebssysteme finden Sie hier: <https://www.arduino.cc/en/software>

Öffnen Sie die IDE und beginnen Sie damit einen neuen Sketch für ihr Projekt zu erstellen. Dafür benötigen Sie die Datei student.ino, sowie die student.h aus dem ZIP-Verzeichnis. Um einen neuen Sketch zu erstellen, müssen Sie zunächst einen Ordner mit dem Namen student erstellen und die Dateien in diesen hinzufügen. Anschließend klicken Sie oben links in der Taskzeile auf File -> Open und öffnen die Datei in ihrem erstellten Ordner.

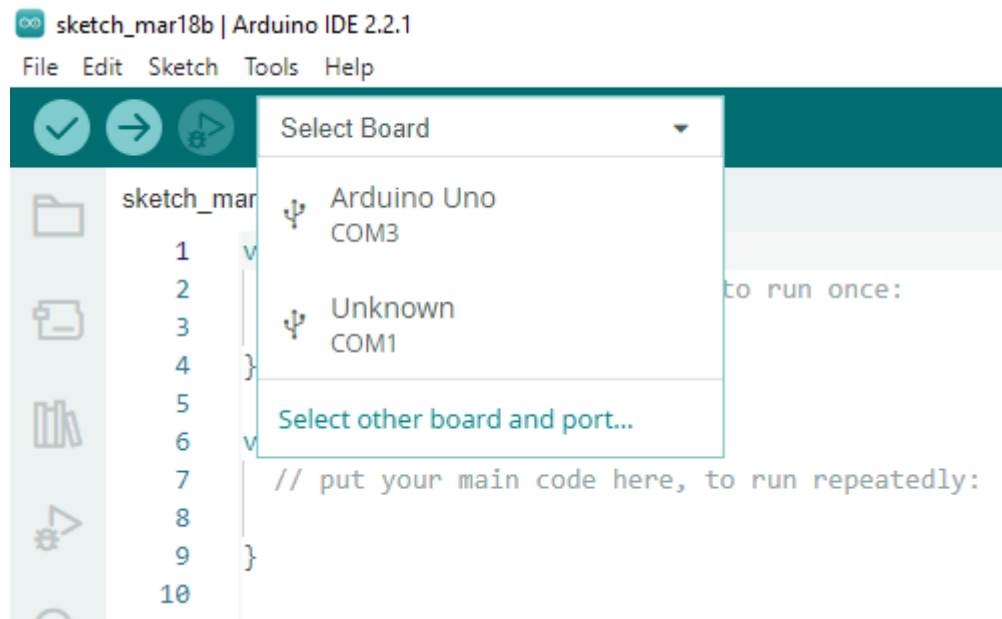
Als letzter Schritt müssen noch die benötigten Libraries hinzugefügt werden, dafür werden einmal die von Arduino selbst zur Verfügung gestellte „Servo“-Bibliothek verwendet, sowie unsere selbst erstellte Namens „SignalSource“ (ebenfalls im ZIP-Verzeichnis).



Öffnen Sie dieses Menü und suchen Sie dann in der Suchzeile nach „Servo“, wählen Sie im Dropdown-Menü Version 1.2.1 aus und installieren Sie diese.

Für unsere eigene Library, wählen Sie stattdessen den Reiter **Add .ZIP library** aus und wählen die signalSource.zip Datei aus. Diese wird nun automatisch installiert.

Im letzten Schritt zur Initialisierung des Microcontrollers muss noch das richtige Board und der richtige COM-Anschluss ausgewählt werden. Hierfür schließen Sie bitte den Microcontroller mit dem beigelegten Kabel an Ihr Gerät an. Anschließend sollte im folgenden Menü:



Ihr Board sowie der zugehörige COM-Port angezeigt werden. Wählen sie diesen dann bitte aus.

Python:

Als nächstes muss noch eine Umgebung für das Python-Skript erstellt werden. Dieses dient dazu Ihre Ergebnisse zu plotten und ein Interface für Sie bereitzustellen, um Änderungen an Signalen etc., wie in den Aufgaben von Ihnen verlangt werden durchzuführen.

Hierfür empfiehlt sich eine IDE Ihrer Wahl, dies ist jedoch nicht notwendig. Die Installation wird im Folgenden anhand des Beispiels von Visual Studio Code erläutert.

Wie Sie über die RWTH Zugriff auf dieses Programm beispielsweise erhalten und es downloaden können erfahren Sie hier: <https://help.itc.rwth-aachen.de/service/468ad37bfd1f4fe19073f4465ea3c685/article/1f561baf6041487ab08e6c9edd95f4e0/>

Als erstes müssen Sie Python installieren. Auf Windows kann das über die Website: <https://www.python.org/downloads/> erfolgen. Details zu anderen Betriebssystemen finden Sie hier: <https://code.visualstudio.com/docs/python/python-tutorial>

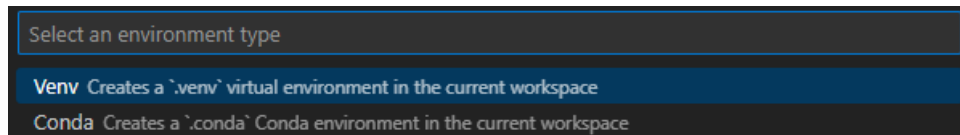
Sie könne nun die PlotData Datei in Visual Studio Code öffnen.

Falls Sie bereits Python-Pakete nutzen, wie es bei vielen Linux-Distributionen üblich ist, empfiehlt es sich eine virtuelle Umgebung zu erstellen, um keine Versionskonflikte zu erzeugen.

Virtuelle Umgebung (falls nötig und gewollt):

Falls Sie keine Umgebung erstellen wollen, fahren Sie mit dem nächsten Schritt fort.

1. Öffne mit Ctrl+Shift+P die Command-Zeile und wähle **Python:Create Environment** aus
2. Wähle nun Venv aus



3. Wählen Sie anschließend den Interpreter aus den Sie zuvor installiert haben
4. Wenn alles funktioniert hat wird ein /.venv Ordner im Workspace erscheinen

Sie können eine Virtuelle Umgebung auch über eine Konsole erzeugen. Eine Anleitung finden sie hier: <https://docs.python.org/3/library/venv.html>

Pakete installieren:

Für das Pythonskript werden verschiedene Pakete benötigt, welche bspw. Das Plotten ermöglichen. Diese müssen von Ihnen installiert werden.

Um alle Pakete und auch deren richtigen Versionen zu installieren, wird Ihnen als Hilfe eine requirements.txt Datei bereitgestellt. In der befinden sich die Informationen und es wird alles mit einem Befehl ausgeführt.

Öffnen Sie das Terminal, und führen Sie den Befehl: ***python -m pip install -r requirements.txt*** aus. (Der Befehl kann je nach Betriebssystem variieren)

Überprüfen Sie die Installation, indem Sie den Code einmal ausführen, es sollten keine Fehlermeldungen entstehen und ein Hilfetext zur Nutzung des Moduls erscheinen.

Projekthardware und Durchführung:

Verbinden Sie zunächst den Servomotor entsprechend der folgenden Tabelle mit dem Arduino.

Servomotor	Arduino
Rot mit Widerstand	+5V

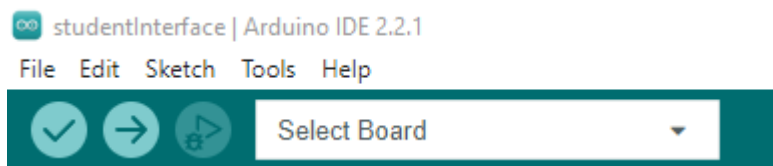
Schwarz	GND
Weiss	Pin 6
Gelb	Pin 3

Arduino Verkabelung

Verbinden Sie nun den Arduino per USB-Kabel mit Ihrem Computer.

Flashen des Arduinos

Um Ihr Programm ausführen zu können, muss dieses auf den Arduino gespielt werden (flashen), das muss vor jeder neuen Nutzung und nach jeder Änderung im C-Code durchgeführt werden. Nutzen Sie hierfür den Upload-Button in der Arduino IDE.



(Upload-Button Pfeil nach rechts)

Als zweite Schnittstelle gibt es das Pythonskript „plotData.py“, dieses kommuniziert über eine serielle Verbindung mit dem Arduino und wählt bestimmte Versuche aus, die Sie durchführen können. Gleichzeitig liest dieses Skript Messwerte und berechnete Werte aus und stellt diese mit Plotly dar. Eine Anpassung dieses Skripts ist nicht nötig, sie können es aber anpassen, um sich die Auswertung zu erleichtern. Übergeben Sie bei der Ausführung des Skripts alle nötigen Parameter. Sie müssen dabei auch den richtigen COM-Port in das Skript übergeben, welchen Sie in der Arduino IDE sehen können. Unter Windows kann das z.B. etwas wie “COM3” sein und unter Linux z.B. etwas wie /dev/ttyACM0.

```

plotData.py > ...
1  import serial
2  import time
3  import matplotlib.pyplot as plt
4  import numpy as np
5  from enum import Enum
6  import time
7  import pandas as pd
8  from struct import *
9
10 # set up the serial line
11 ser = serial.Serial('COM3', 2000000)
12 if not ser.isOpen():
13     ser.open()
14 print('com3 is open', ser.isOpen())
15 time.sleep(2)

```

siehe Zeile 11

Vergewissern Sie sich, dass der Arduino zuvor geflasht wurde. Es zeigt nach Beendigung des Runs einen Plot an, welchen Sie zum Teil für Ihre Aufgaben benötigen.