

Grundlagen der C-Programmierung in ISOBUS (1)

Franz Höpfinger

Inhaltsverzeichnis

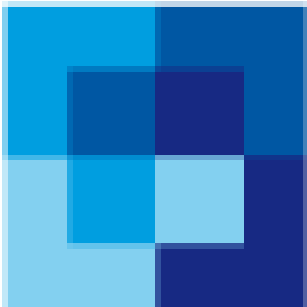
	myst: enable_extensions: [„colon_fence“] html_meta: „description lang=en“: 6 „metadata description“ „description lang=de“: „metadata description“ „description lang=fr“: „description des métadonnées“ „keywords“: „Sphinx, MyST“ „property=og:locale“: „de“	
1	Wiki 1: C-Programmierung in ISOBUS	7
1.1	Meisterschulen am Ostbahnhof, München	7
1.1.1	 Das steht im Lehrplan (Rahmenlehrplan 2025)	7
2	Atom Erweiterungen	9
3	Eagle	13
3.1	 Podcast	13
3.2	 Video	13
3.3	Abkündigung:	13
3.4	Installer:	13
3.5	probleme	13
4	 Installation und Verwendung von Gitkraken:	15
5	 Hardware	31
5.1	 Podcast	31
6	Startseite	33
6.1	 Podcast	33
7	HutschienenMoped Eingänge	35
7.1	Unterscheidung von Endschaltern für Pneumatikzylinder:	35
7.1.1	Reedschalter	35
7.1.2	Magnetsensoren mit Halbleitertechnik	35
7.2	Anschluss Endschalter an Eingang	38
7.3	Grove System	39
7.4	Reihenschaltung von Endschaltern	40
7.5	ButtonBoards	40
7.5.1	 Passende Themen-Unterseiten auf ms-muc-docs.de	40
8	HutschienenMoped Erweiterungen	41
9	HutschienenMoped Stecker	51
9.1	1. Grundlegende Verdrahtung und Anschlüsse	51

9.2	2. Isolierung und Berührungsschutz	54
9.3	3. Detailansicht: Spezielle Signalanschlüsse	55
9.4	4. Steckverbindungs-Gehäuse und Gesamtansicht	57
9.5	5. Betriebszustand	61
10	HutschienenMoped	63
10.1	 Podcast	63
10.2	 Repository & Ressourcen	63
10.3	 Anschluss & Belegung	63
10.3.1	Pinbelegung	63
10.4	 Konfiguration (4-Pol vs. 6-Pol)	63
10.4.1	Betrieb mit 4-poligem Kabel	63
10.5	 Galerie & Montage	64
10.5.1	 Hardware-Ansichten	64
10.5.2	Montageoptionen	66
10.5.3	 Passende Themen-Unterseiten auf ms-muc-docs.de	67
11	Hutschienenmoped XL	69
11.1	 Highlights	69
11.2	 Zertifizierung	69
11.3	 Konstruktion & CAD	70
11.3.1	Repository	70
11.4	 Galerie	70
11.5	i Technische Details	71
11.5.1	 Passende Themen-Unterseiten auf ms-muc-docs.de	71
12	ISOBUS Kabel klein	73
13	 Installation	79
14	IsoAgLib-ISOMAN	81
15	JTAG-S3-USB	85
15.0.1	 Passende Themen-Unterseiten auf ms-muc-docs.de	92
16	JTAG	93
16.1	JTAG Schritt_01:	93
16.1.1	USB - Schnittstelle mit ZADIG umstellen:	93
16.2	JTAG Schritt_02:	101
16.3	JTAG Schritt_03:	102
16.4	JTAG Schritt_04:	102
16.5	JTAG Schritt_05:	103
16.6	JTAG Schritt_06:	103
16.7	JTAG Schritt_07:	103

17	Joystick	105
18	Kalkulation Hardware	107
18.0.1	 Passende Themen-Unterseiten auf ms-muc-docs.de	108
19	Lego	109
19.1	 Podcast	133
19.1.1	 Passende Themen-Unterseiten auf ms-muc-docs.de	133
20	Optional	135
21	Paket	137
21.1	 Podcast	137
21.2	 Video	138
22	Selbsttest	139
23	Setting-up (LINUX)	141
24	Setting-up-II	143
25	Setting-up Installation	153
25.1	Schritt 01:	153
25.2	Schritt 02:	153
25.3	Schritt 03:	154
25.4	Schritt 04:	155
25.5	Schritt 05:	156
25.6	Schritt 06:	158
25.7	Schritt 07:	159
25.8	Schritt 08:	160
25.9	Schritt 09:	161
25.10	Schritt 10:	162
25.11	Schritt 11: (nur Info)	163
25.12	Schritt 12:	164
26	blink_on_atom	165
26.0.1	Schritt_14:	165
26.0.2	Schritt_15:	165
26.0.3	Schritt_16:	166
26.0.4	Schritt_17:	167
26.0.5	Schritt_18:	170
26.0.6	Schritt_20:	170
26.0.7	Schritt_21	171
26.0.8	Schritt_22	174
26.0.9	Programm verändern	176
27	cci_EasyExample	177

28	cci_EasyExample_CAN2IP	179
29	 Das steht im Lehrplan: C-Programmierung in ISOBUS	183
29.1	 Relevanz der C-Programmierung & ISO 11783 C-Libraries im Lehrplan 2025	183
29.2	 Lehrplan-Module & Schwerpunkte	183
29.2.1	1. Teil I – Lerneinheit 1.4: Automatisierte Funktionen (80 UE)	183
29.2.2	2. Teil II – Lerneinheit 1.1: Systemtechnik (120 UStd.)	183
29.2.3	 Passende Themen-Unterseiten auf ms-muc-docs.de	183
29.3	 Übersicht, Hauptportal & Direktlinks zu allen 6 Wikis	183

0.1 `myst: enable_extensions: [„colon_fence“]` `html_meta: „description lang=en“: „metadata description“ „description lang=de“: „metadata description“ „description lang=fr“: „description des métadonnées“ „keywords“: „Sphinx, MyST“ „property=og:locale“: „de“`



Meisterschulen am Ostbahnhof

1. Wiki 1: C-Programmierung in ISOBUS

1.1 Meisterschulen am Ostbahnhof, München

1.1.1 Das steht im Lehrplan (Rahmenlehrplan 2025)

Die **C/C++-Programmierung für eingebettete Systeme** ist die technische Grundlage für die Implementierung normgerechter ISOBUS-Kommunikationsstacks (ISO 11783) an Land- und Baumaschinen (Teil I LE 1.4 sowie Teil II LE 1.1).

-  **Das steht im Lehrplan – C-Programmierung in ISOBUS**

Willkommen bei der Dokumentation zur C-Programmierung in ISOBUS.

Diese Dokumentation ist Teil der Wissensdatenbank der Meisterschulen am Ostbahnhof München.

Nützliche Links & Hauptportal:

-  **Hauptportal Meisterschulen München**
-  **Das steht im Lehrplan (Lehrplan 2025)**
-  Interaktiver Rahmenlehrplan 2025
-  Zentraler PDF-Download-Hub
-  Super-Suche (alle Wikis)
-  PDF-Handbuch herunterladen

!!! note This project is under active development.

Contents

 [Hauptmenü](#) |  [Super-Suche \(alle Wikis\)](#) | [Schnelle Suche \(IEC 61499\)](#)

2. Atom Erweiterungen

grundsätzlich gehen am GROVE Port alle Erweiterungen von M5 Stack:

<https://shop.m5stack.com/collections/m5-sensor>

und alle Erweiterungen die den GROVE Port anbieten:

https://wiki.seeedstudio.com/Grove_System/

https://www.youtube.com/watch?v=1Rc_OiebDPo

mit folgenden werden wir arbeiten:

<https://shop.m5stack.com/collections/m5-sensor/products/mini-dual-button-unit>

<https://shop.m5stack.com/collections/m5-sensor/products/angle-unit>

<https://shop.m5stack.com/collections/m5-sensor/products/mini-button-unit>

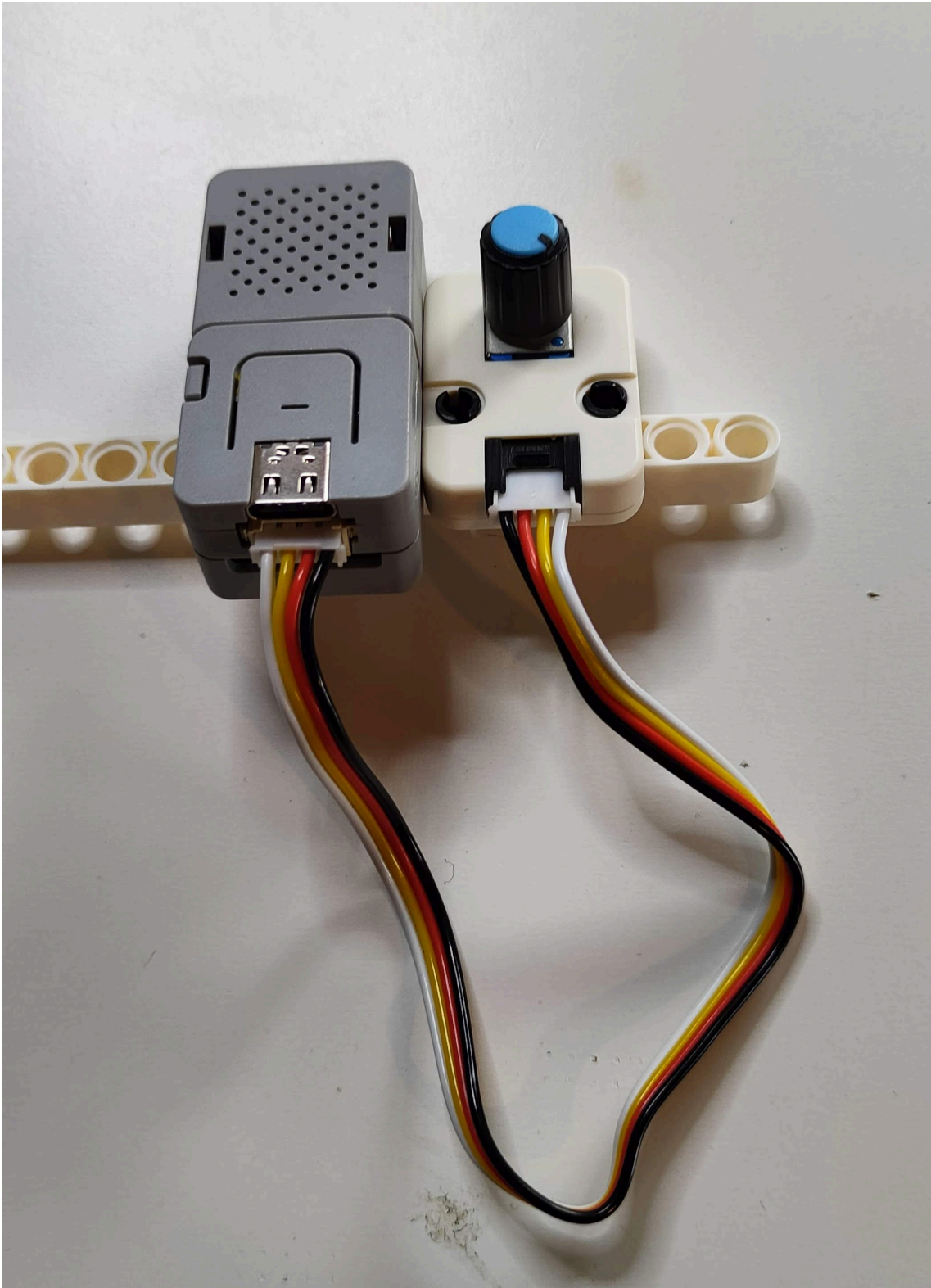
Spannende andere:

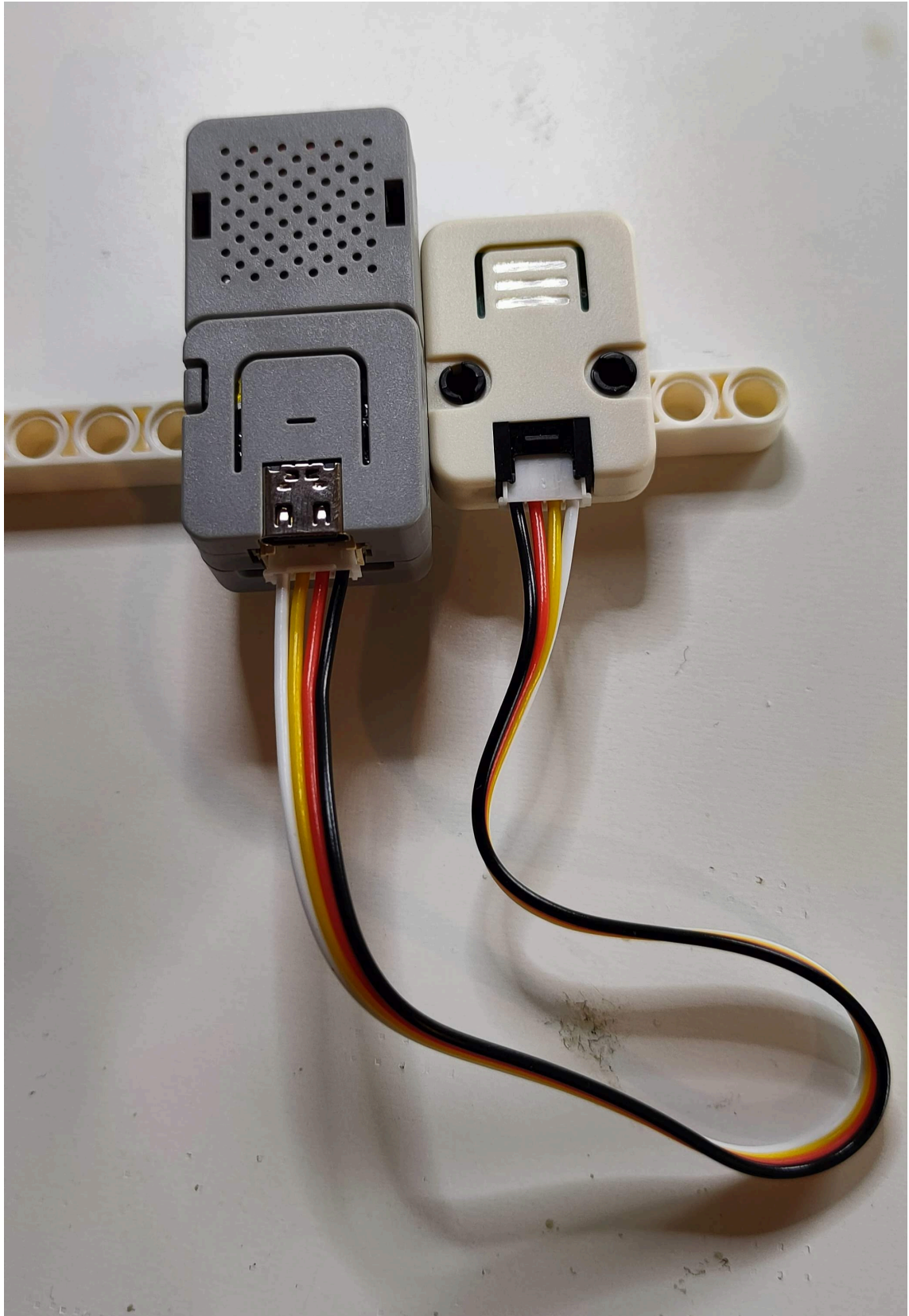
Erdfeuchte: <https://shop.m5stack.com/collections/m5-sensor/products/earth-sensor-unit>

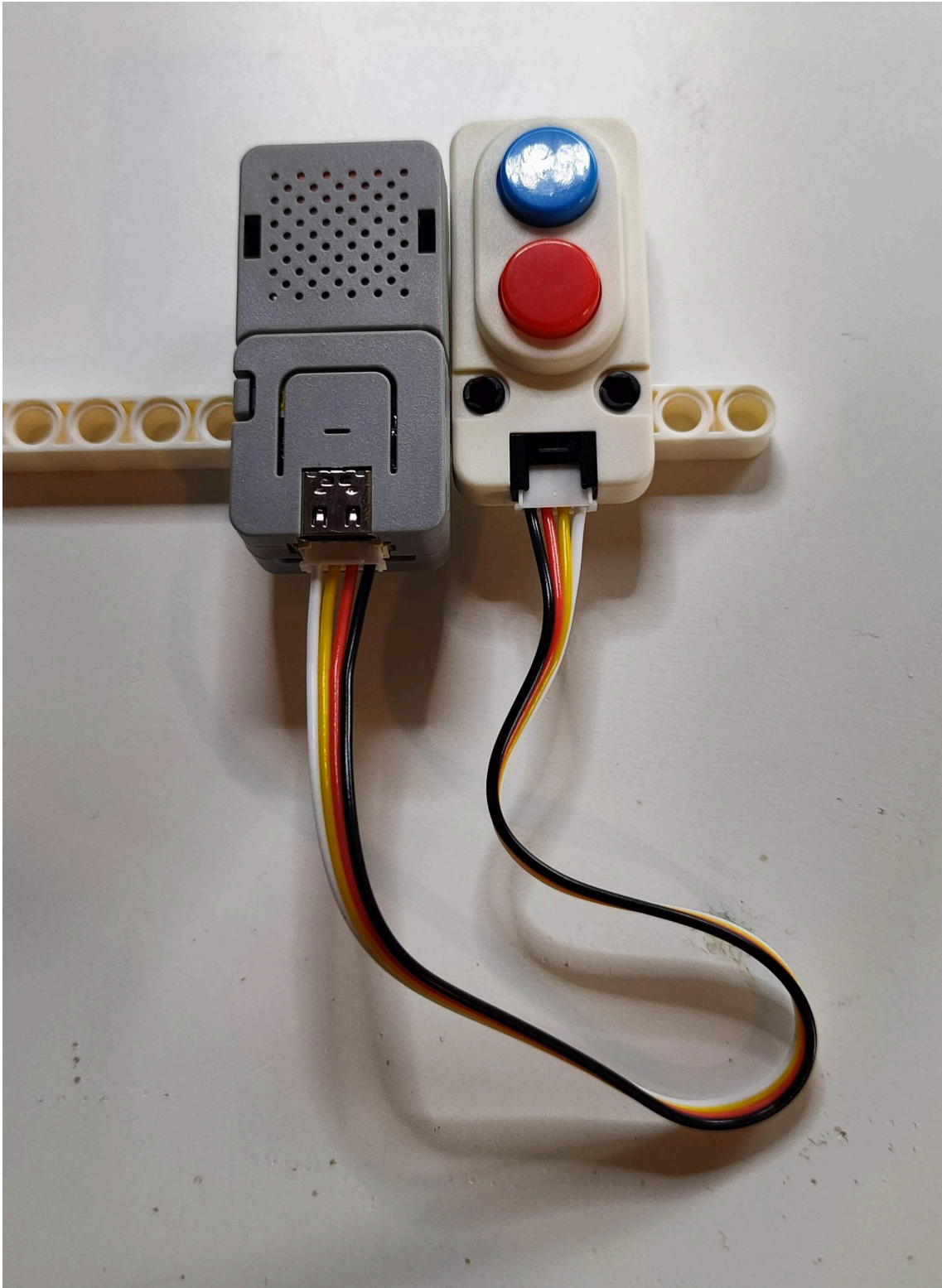
auch Erdfeuchte: https://wiki.seeedstudio.com/Grove-Capacitive_Moisture_Sensor-Corrosion-Resistant/

https://wiki.seeedstudio.com/Grove-LED_Button/ <https://wiki.seeedstudio.com/Grove-LED_Button/

<https://shop.m5stack.com/collections/m5stack-new-arrival/products/fader-unit-with-b10k-potentiometer-sk6812>







3. Eagle

3.1 Podcast

- Vom bayerischen Dorf in die Autodesk Cloud: Die faszinierende Metamorphose der Software-Legende EAGLE CAD

3.2 Video

- EAGLE CAD: A Legend

3.3 Abkündigung:

On June 7, 2026, EAGLE will no longer be available

<https://www.autodesk.com/products/fusion-360/blog/future-of-autodesk-eagle-fusion-360-electronics/>

3.4 Installer:

Autodesk_EAGLE_9.6.2_English_Win_64bit.exe

3.5 probleme

Eagle hat ein Problem unter Windows 11 bei bestimmten HP und Lenovo Laptops: dabei Öffnet Eagle kurz, und macht sofort wieder zu.

<https://www.autodesk.com/support/technical/article/caas/sfdcarticles/sfdcarticles/Eagle-crashes-seconds-after-launching-splash-screen.html>

<https://www.youtube.com/watch?v=pyK8LCPOxkk>

Die Lösung ist dabei zwei Dateien „libeay32.dll“ und „ssleay32.dll“ aus dem Paket

(https://www.totalcommander.ch/win/tools/openssl-1.0.2u-x64_86-win64.zip)[https://www.totalcommander.ch/win/tools/openssl-1.0.2u-x64_86-win64.zip]

herunterzuladen und zu ersetzen.

danach läuft Eagle wie gewohnt.

4. Installation und Verwendung von Gitkraken:

BEVOR Sie beginnen lesen Sie bitte den Abschnitt zu Github durch.

es ist wichtig dass Sie sich mit ihrer Schul-E-Mail Adresse bei Github angemeldet haben,

und Sie sollten auch die privaten Repositories sehen, z.B. dieses hier: https://github.com/Meisterschulen-am-Ostbahnhof-Munchen/blink_on_atom entfernen Sie das Leerzeichen vor „blink“

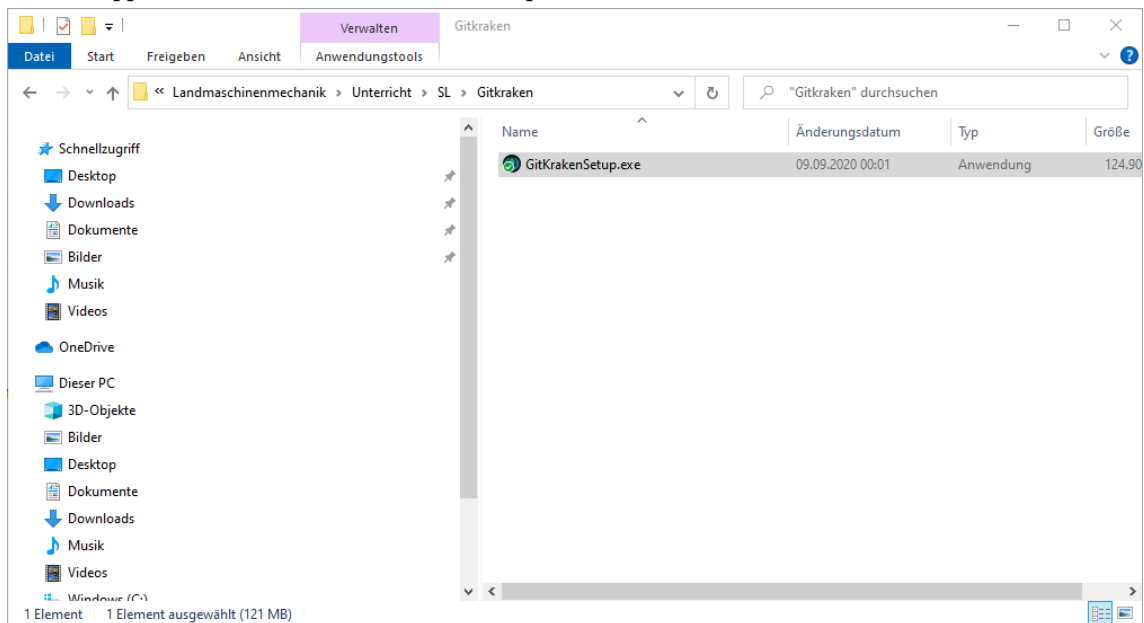
Sie finden Gitkraken unter M:\Landmaschinenmechanik\Unterricht\SL\Gitkraken

alternativ auch unter <<https://www.gitkraken.com/download/windows64>, dort finden Sie auch zahlreiche Videos zur Erklärung:

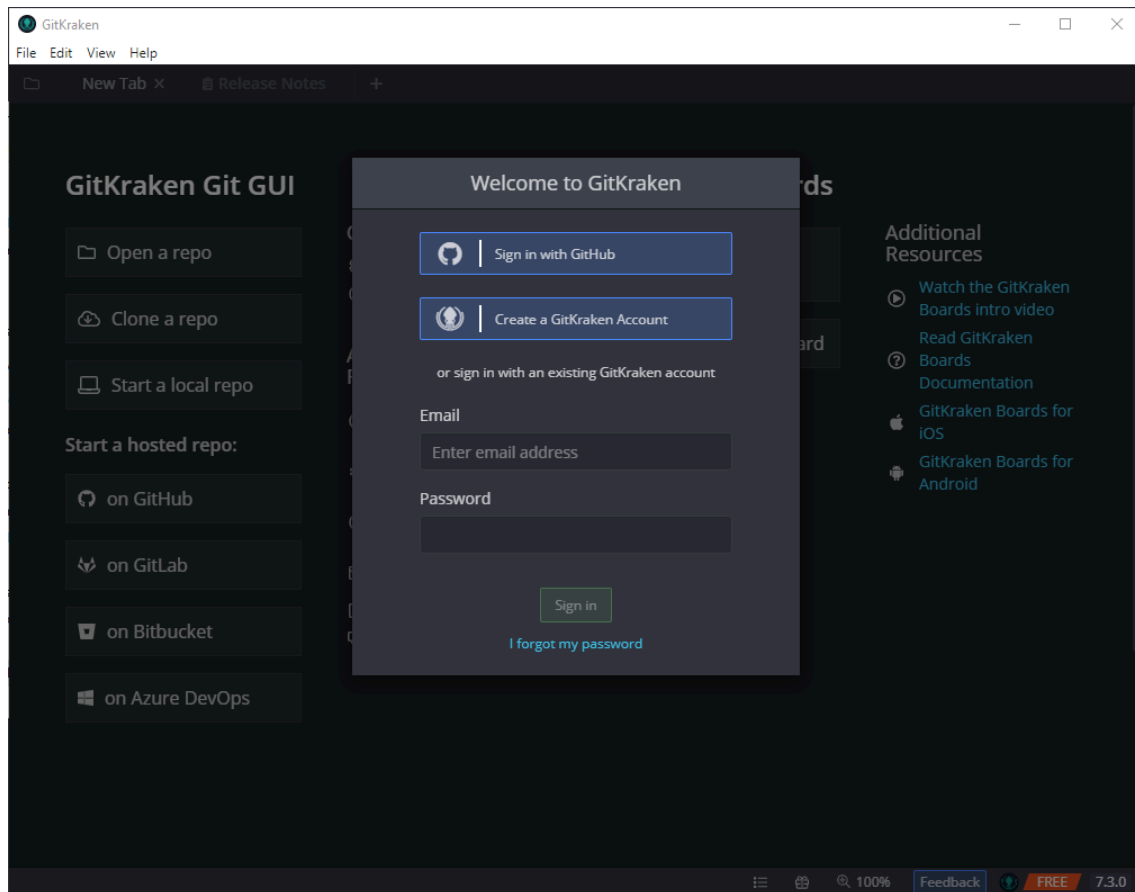
https://www.youtube.com/watch?v=ub9GfRziCtU&feature=emb_title

<https://www.youtube.com/channel/UCp06FAzrFalo3txskS1gCfA>

dort doppelklicken Sie auf die GitKrakenSetup.exe



der Startbildschirm sieht bei mir so aus:

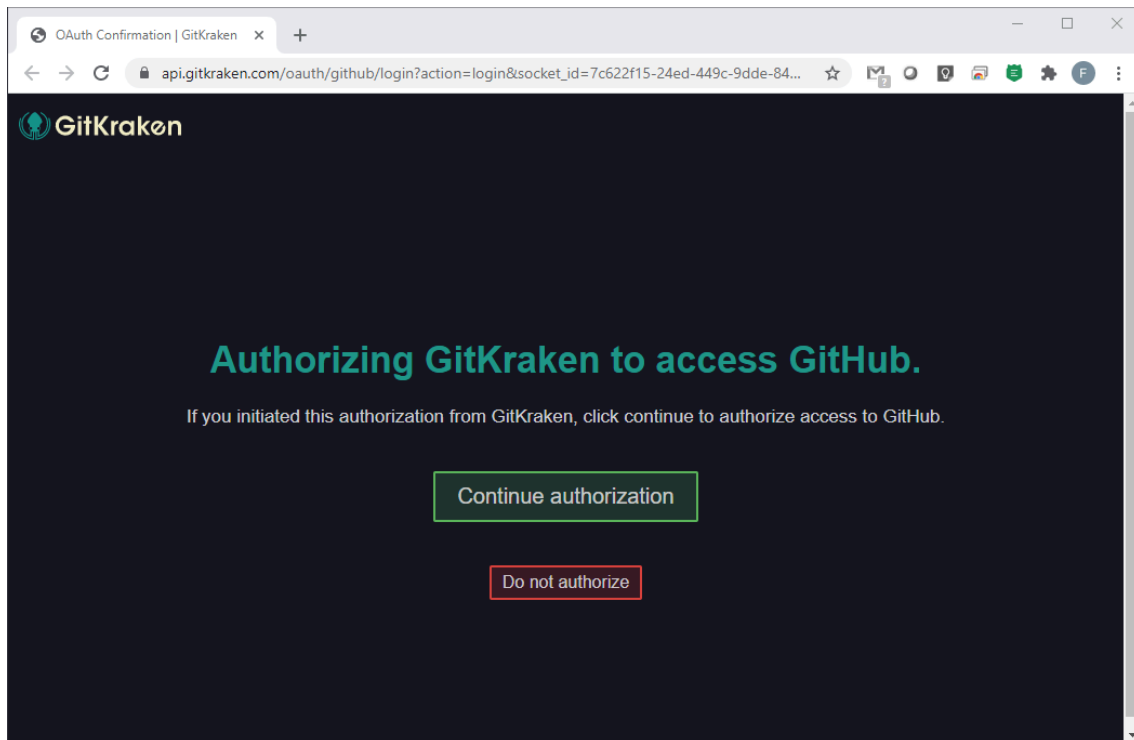


jetzt haben Sie 2 Möglichkeiten:

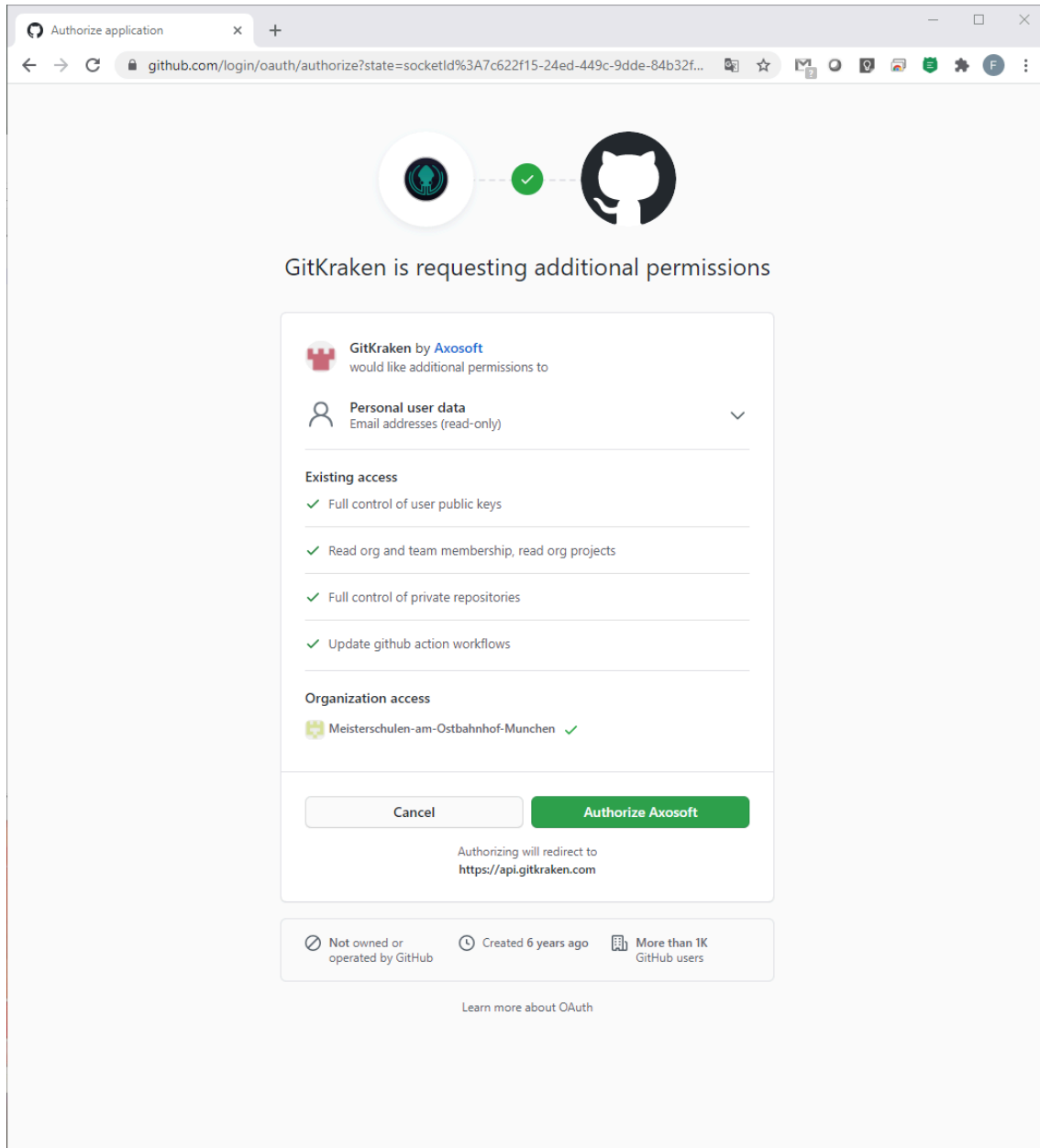
1. für ungeduldige: „Sign in with GitHub“, das beschreiben wir unten auch gleich.
2. <https://www.gitkraken.com/student-resources> hier können sie ein kostenloses GitKraken Pro bekommen, das kostet normalerweise 49\$ pro Jahr und Nase: <https://www.gitkraken.com/pricing>

so, wir beschreiben hier nur die erste Option:

wenn Sie „Sign in With Github“ klicken (sie müssen die Fehler darunter leer lassen>`\_` dann öffnet sich der Browser:



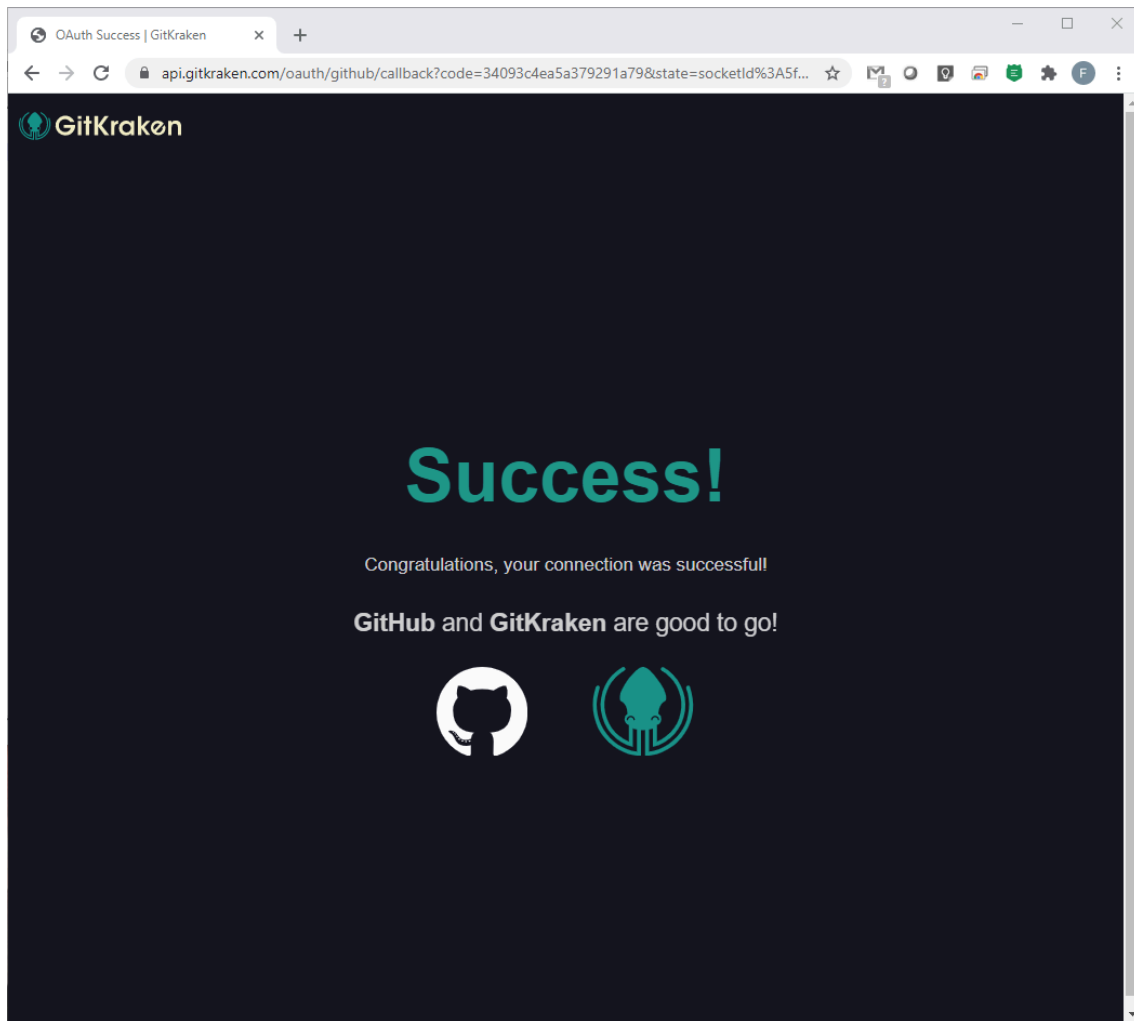
klicken Sie „Continue authorisation“
jetzt sollten Sie so etwas sehen:



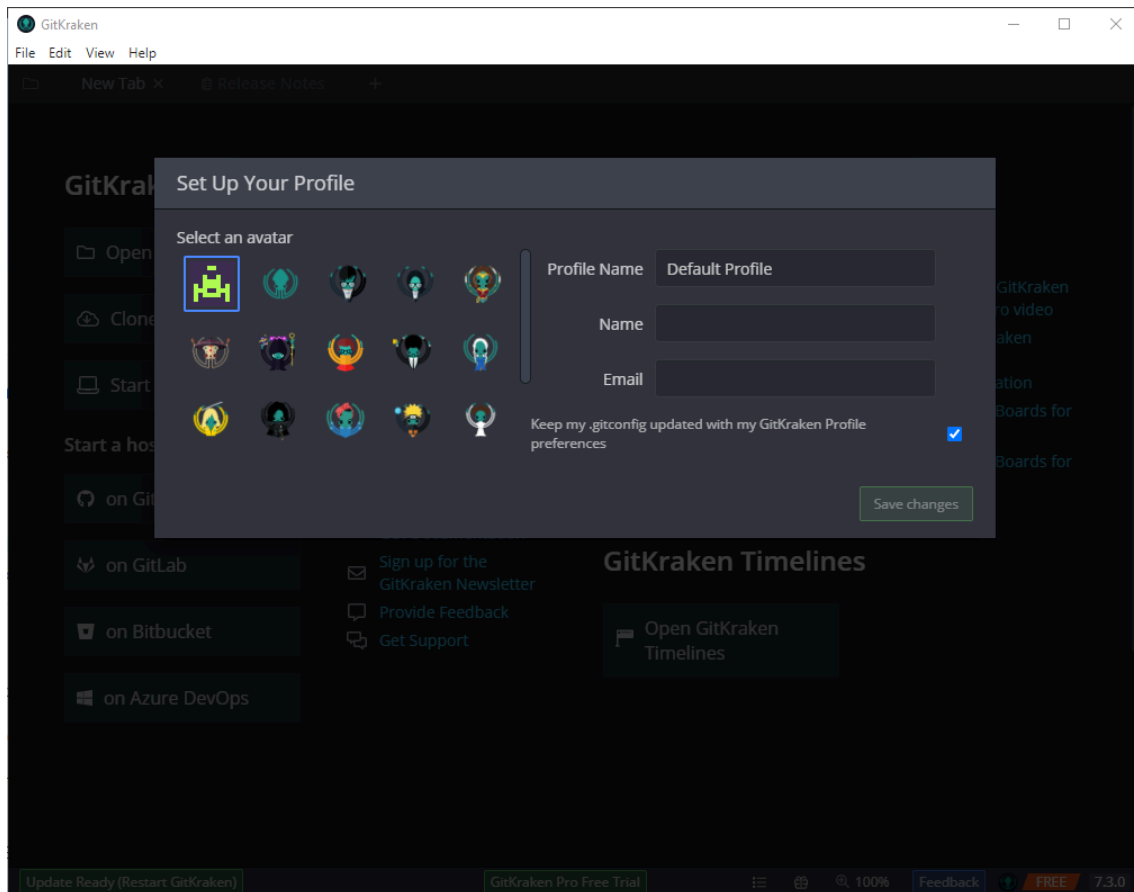
Übrigens: wenn ihnen ihr Foto nicht gefällt: in Github kann man es frei anpassen.

klicken Sie auf „Authorize Axosoft“ und machen Sie bei Meisterschulen-am-Ostbahnhof-Munchen einen haken wenn der noch nicht da wäre.

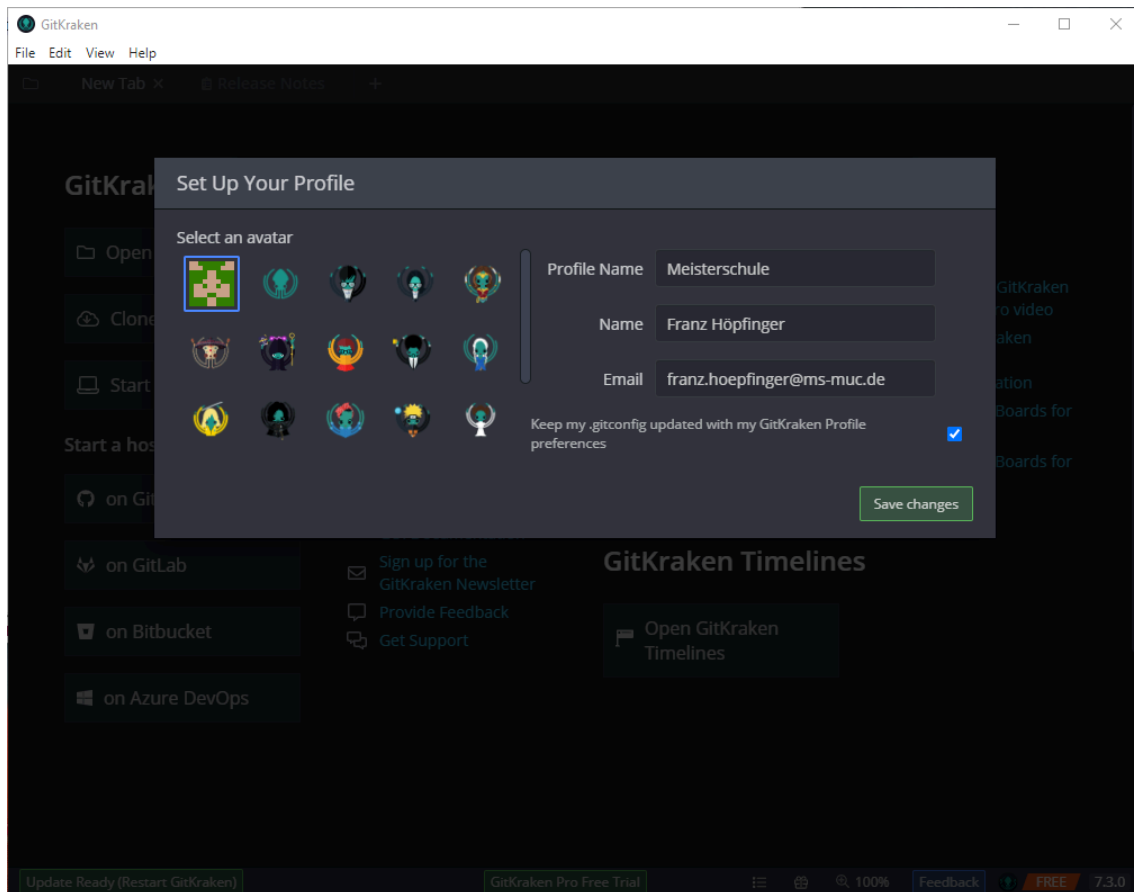
wenn dieses Fenster erscheint hat es funktioniert:



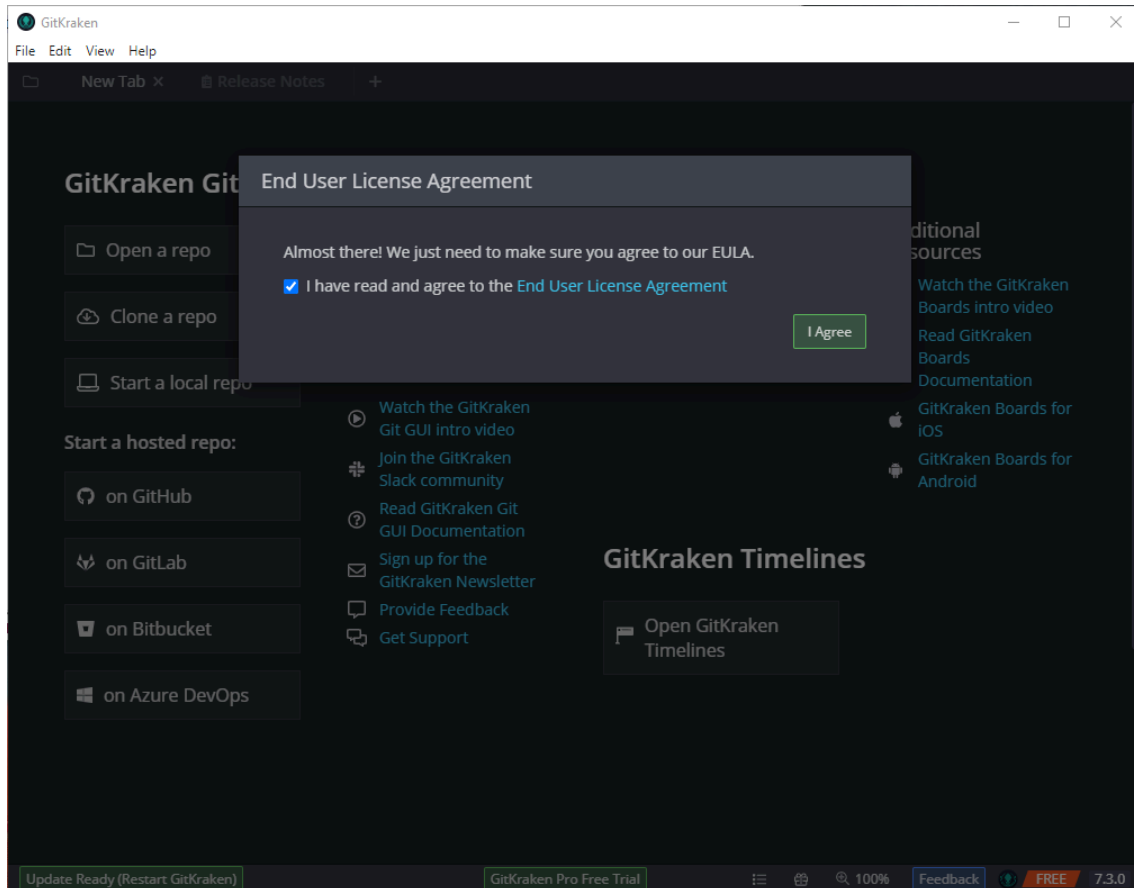
zurück in Gitkraken sollten Sie das sehen:



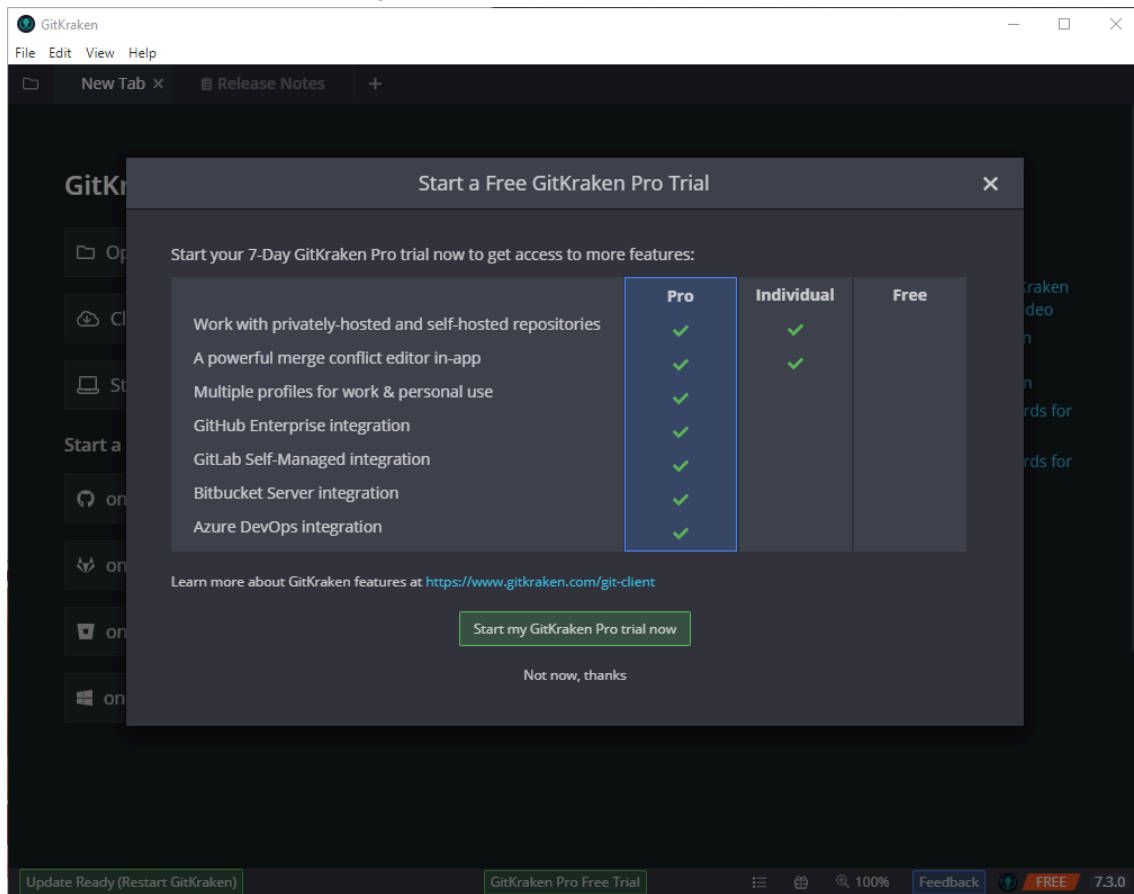
geben Sie ihren Namen usw. ein,
so zum Beispiel:



und dann klicken Sie auf Save changes.

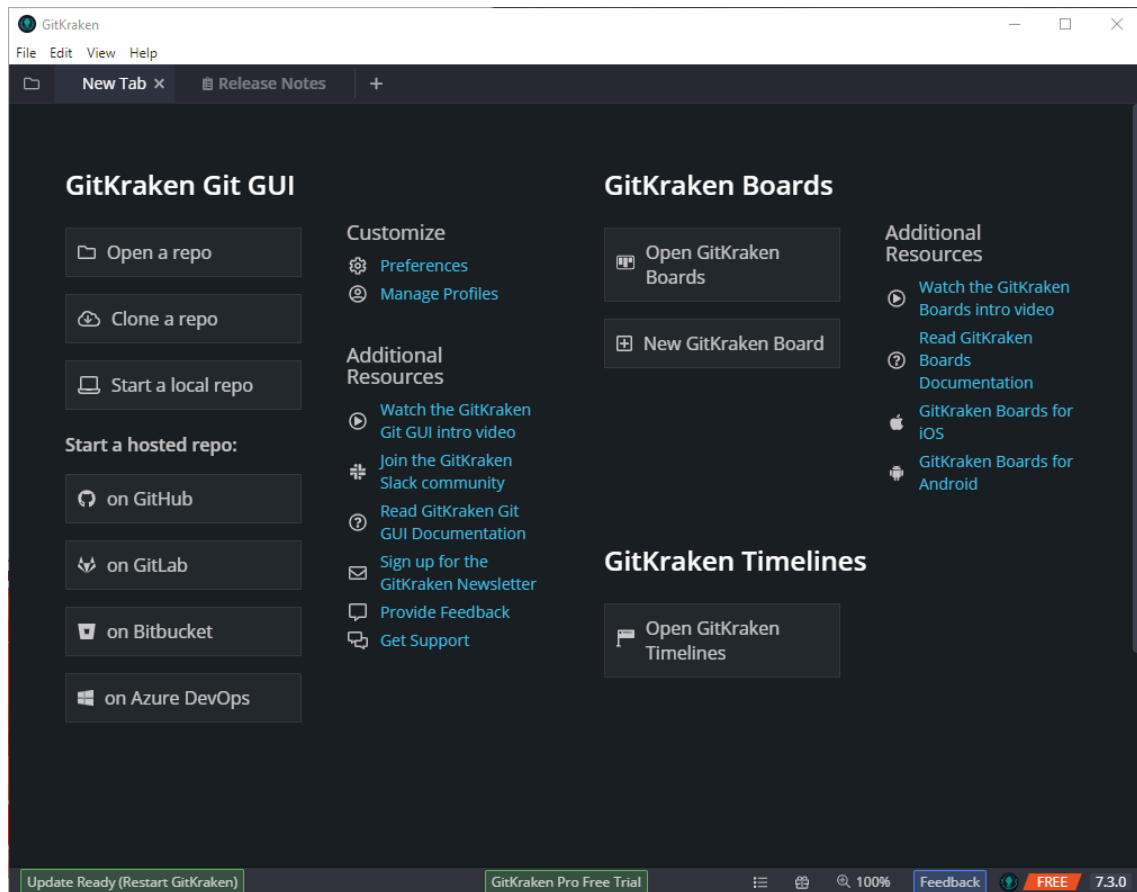


hier einen Haken setzen, und „I Agree“ drücken.

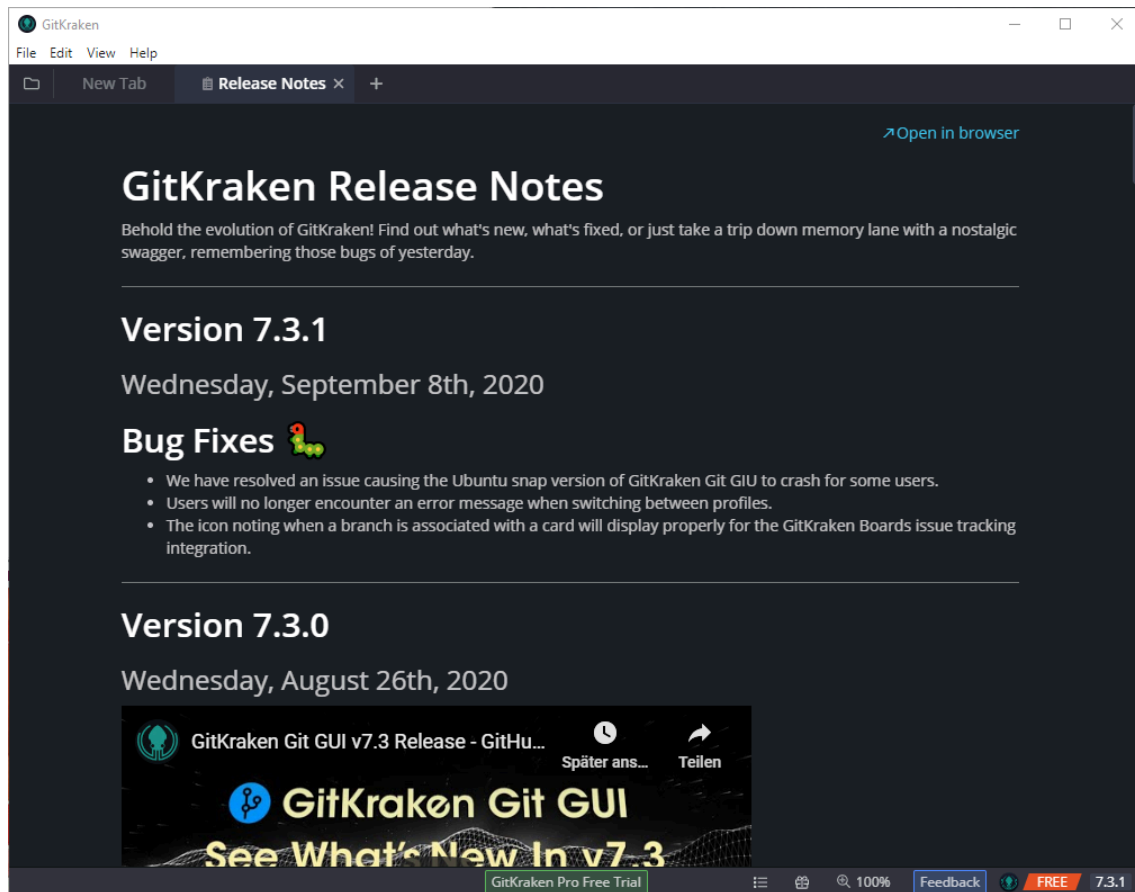


hier ist egal was Sie wählen, Gitkraken gibt ihnen automatisch ein Trial der Pro Version. deswegen wählen wir „Start my Pro trial“

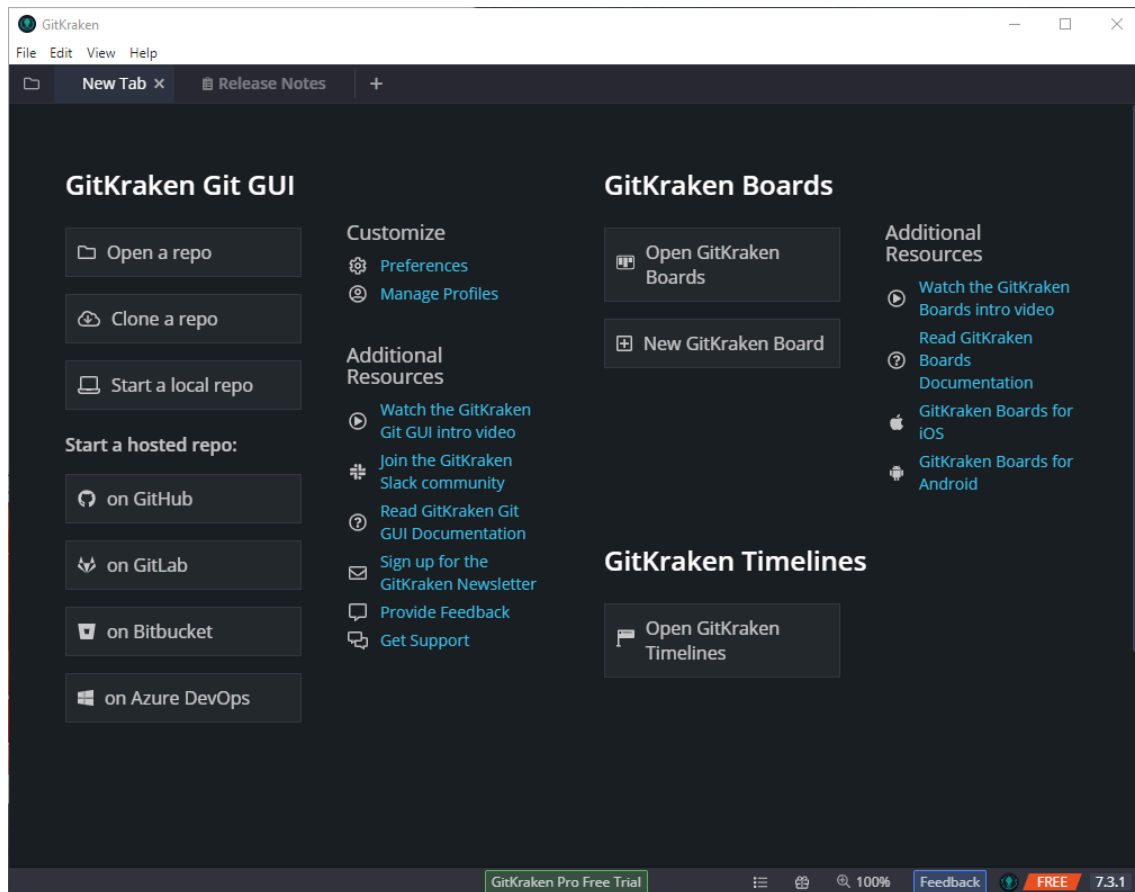
(siehe oben für eine Möglichkeit als Student eine kostenlose Pro für mehr als die 7 Tage zu bekommen>`_



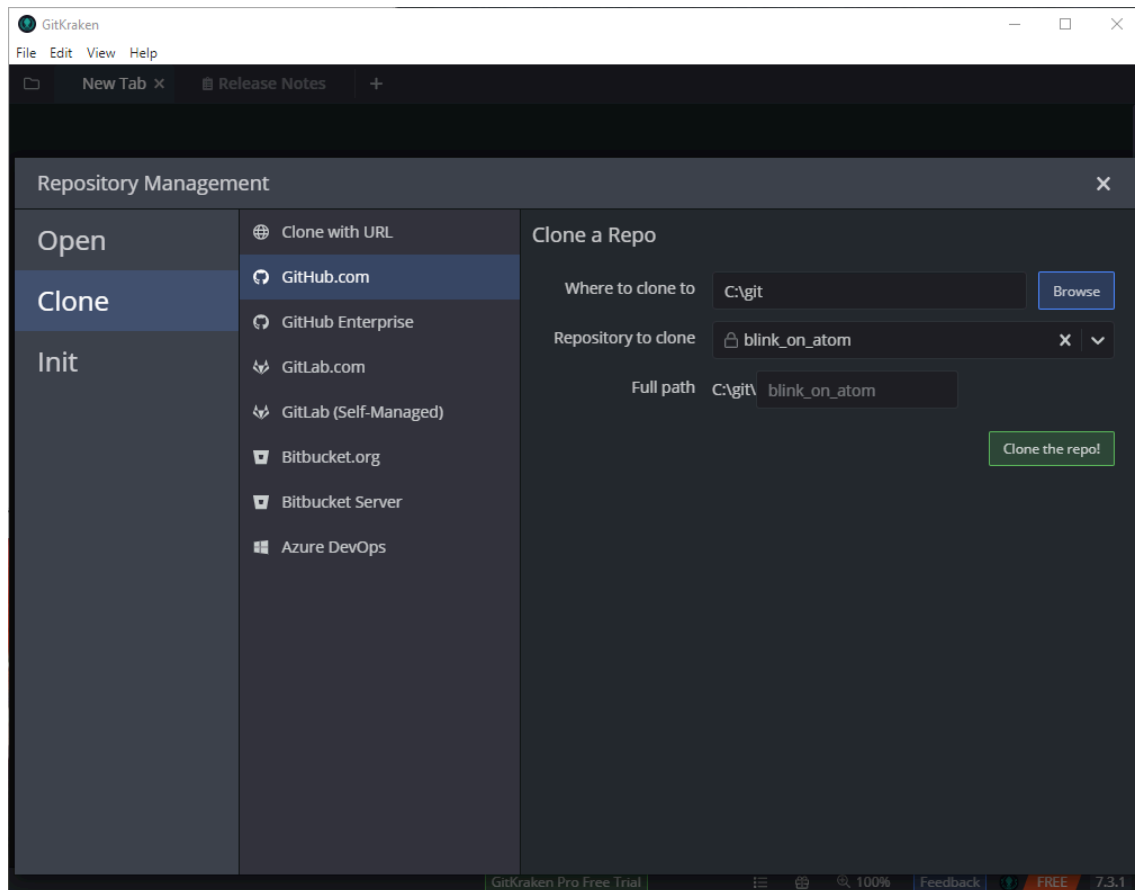
jetzt klicken Sie noch auf Update Ready,
und dann sollte es so aussehen:



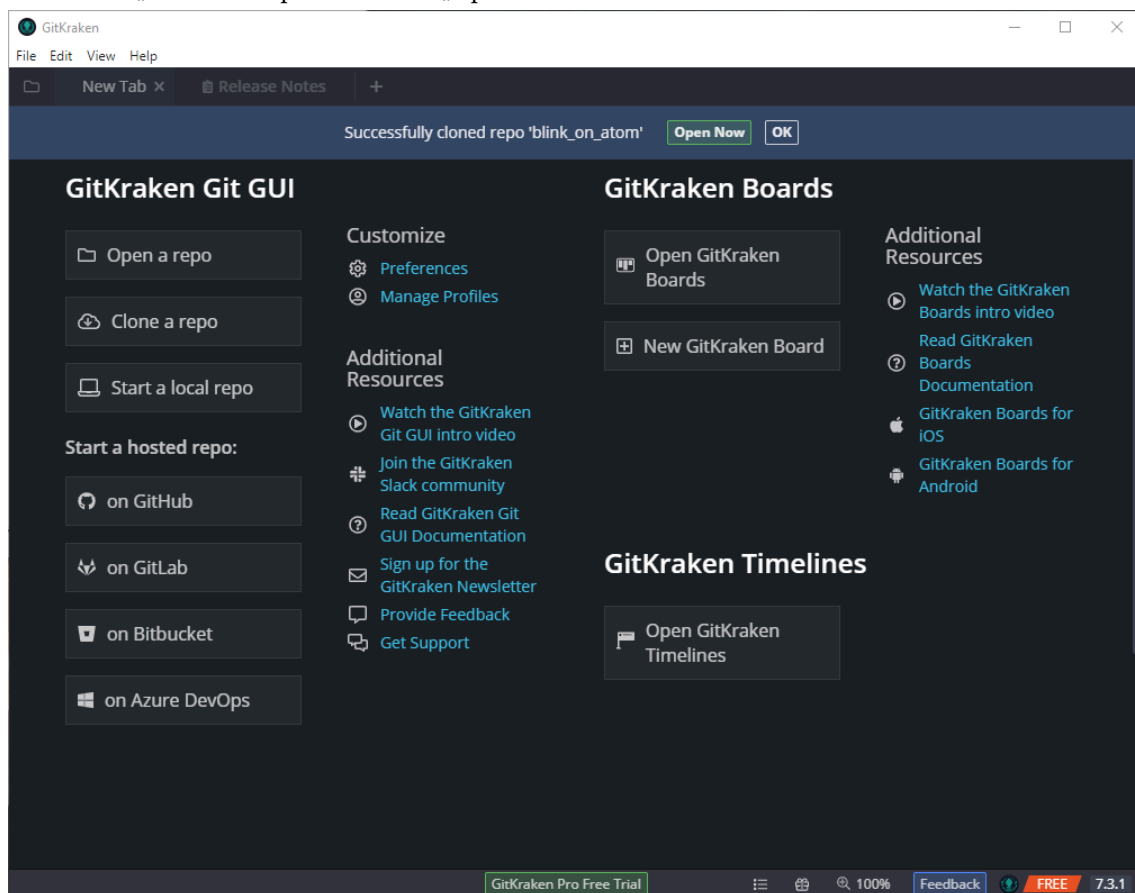
als nächstes holen wir uns 2 Repositories.
dazu brauchen wir zuerst einen Platz.
ein guter Platz ist C:\git oder D:\git wenn ihr Laptop 2 Festplatten hat.
ein Netzlaufwerk wie das E:\ Laufwerk ist eine schlechte Idee.
klicken Sie „new tab“ und „clone a Repo“



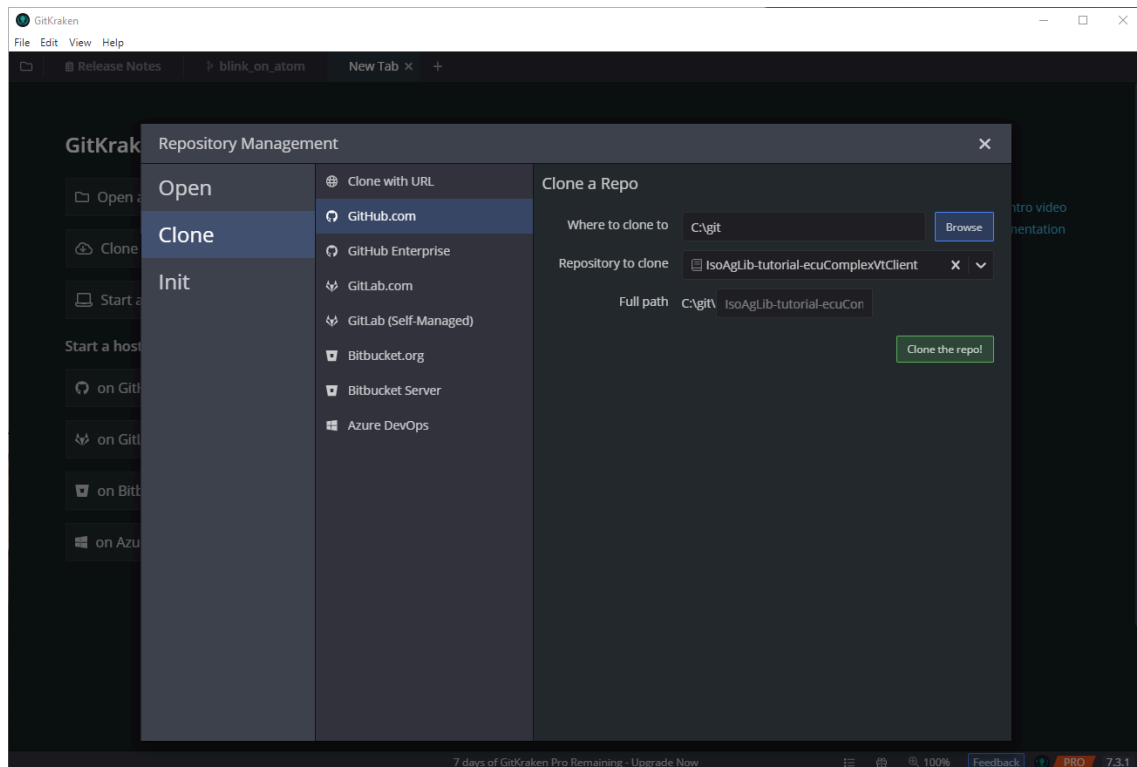
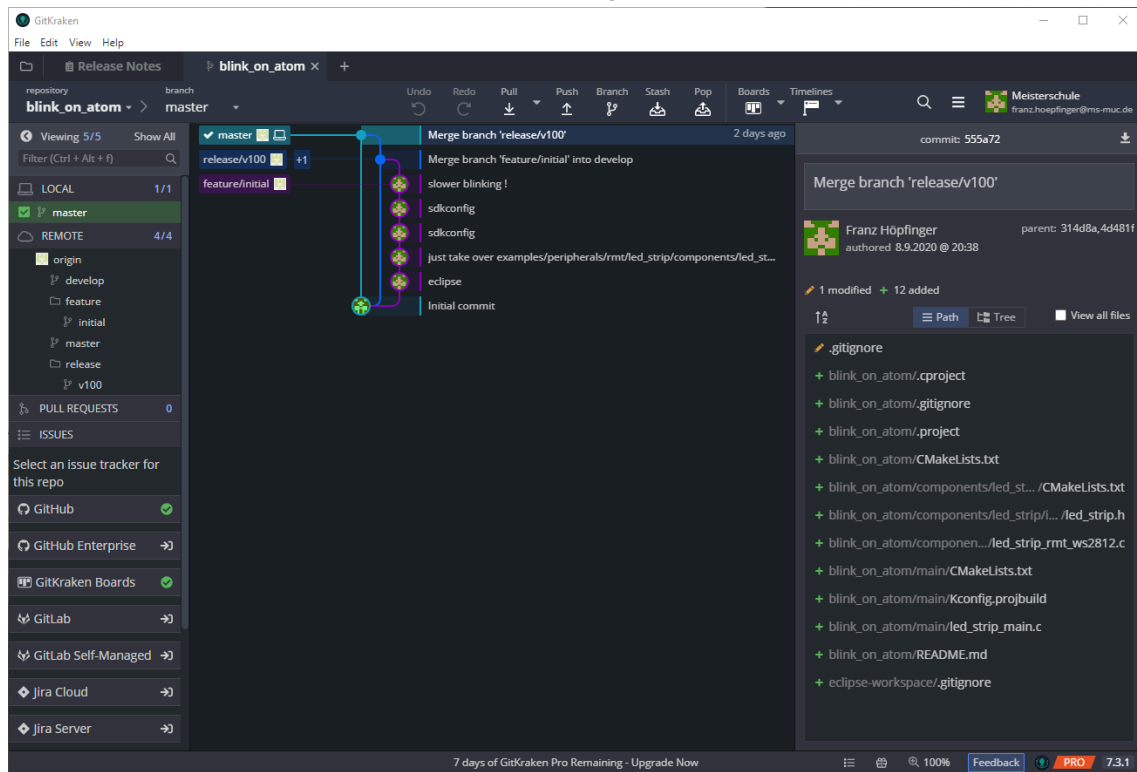
im nächsten Bild wählen Sie „GitHub.com“ und ihren vorher bestimmten Platz, und wählen das Repository „blink_on_atom“



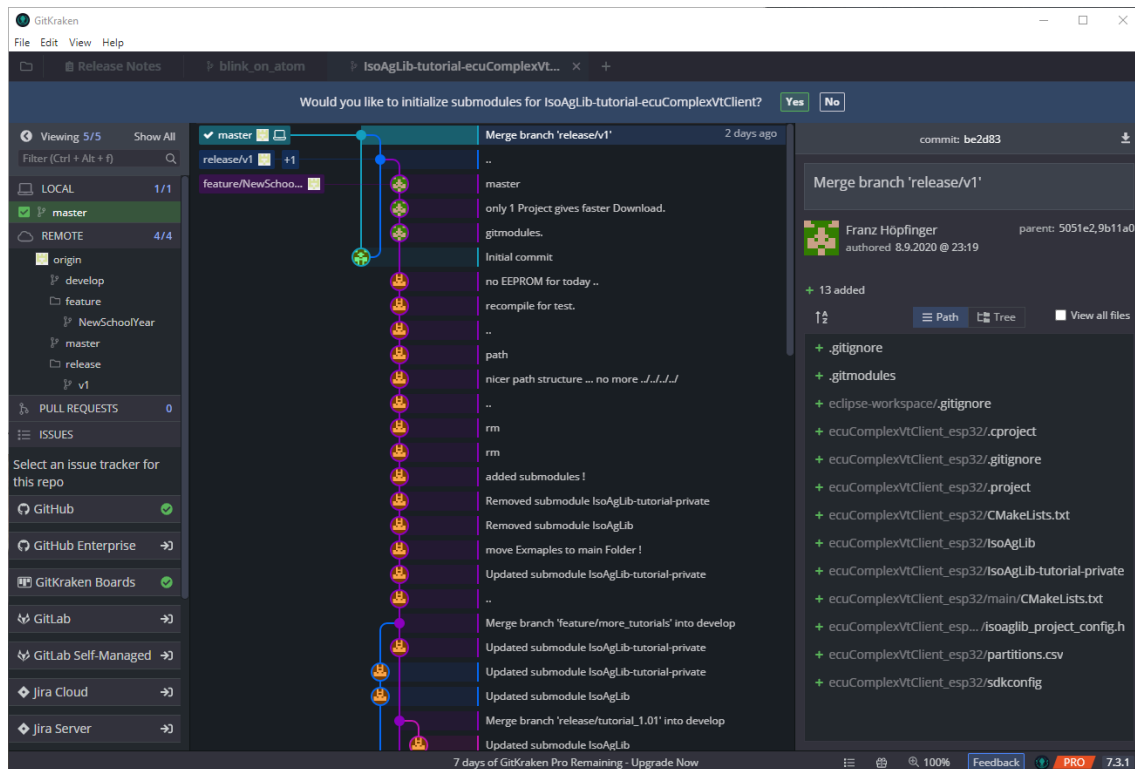
Klicken Sie „Clone the repo!“ und dann „Open Now“



jetzt klicken Sie auf das „+“ oben und wiederholen das ganze noch einmal:



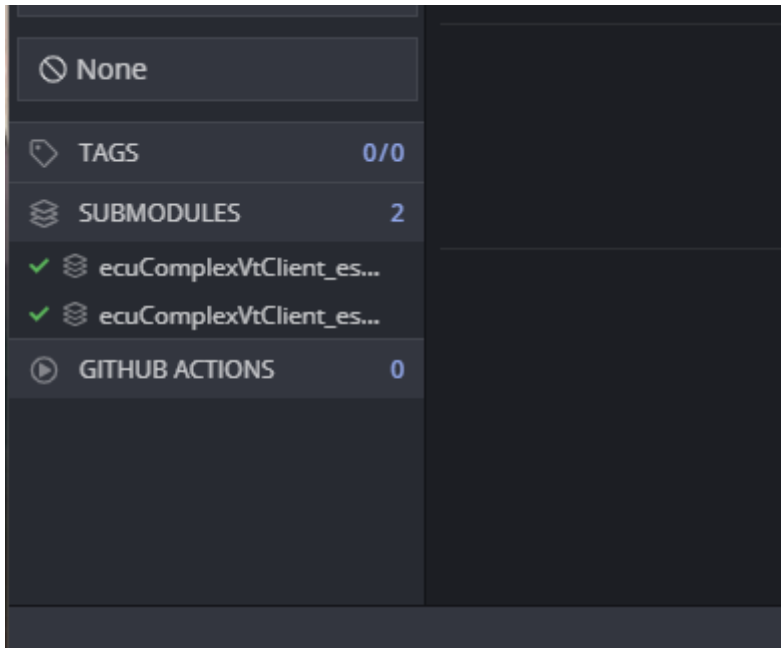
Dieses Mal wählen Sie den „ISOMAN“
Sie klicken wieder „open Now“
und bestätigen hier mit YES



links unten sollte es dann so aussehen:



nach einer Weile aber dann so:



so, für heute war es das mit Gitkraken.

wir werden noch viele Stunden damit verbringen, aber im Moment brauchen wir Gitkraken nicht mehr.
Sie können es schließen.

5. Hardware

5.1 Podcast

- Automatisierung 4.0: Warum Software die Hardware überholt und was das für deine Skills bedeutet
- SPS: Das Unsichtbare Gehirn der Industrie – Von robuster Hardware zur IT/OT-Konvergenz
- Universal Automation: So entkoppeln Sie Software und Hardware für die Zukunft der Industrie

6. Startseite

6.1 Podcast

- logiBUS®: Revolutioniert die Agrar-IT – So wird ISOBUS zum Smart Home für Landwirte

7. HutschienenMoped Eingänge

7.1 Unterscheidung von Endschaltern für Pneumatikzylinder:

<https://www.ifm.com/at/de/shared/produkte/zyindersensoren/technologie>

7.1.1 Reedschalter

Ein Reed-Schalter ist ein elektrisches Schaltelement, das aus zwei ferromagnetischen Kontakten in einem hermetisch verschlossenen Glasrohr besteht. Diese Kontakte werden durch einen dünnen, elastischen Glasstab voneinander getrennt gehalten, der mit einer speziellen Metallegierung beschichtet ist.

Die wichtigsten Eigenschaften eines Reed-Schalters sind:

1. Magnetische Aktivierung: Ein Reed-Schalter wird durch ein magnetisches Feld aktiviert. Wenn das Feld stark genug ist, bewegen sich die Kontakte und schließen einen elektrischen Stromkreis.
2. Kleine Größe: Reed-Schalter sind sehr klein und können in engen Räumen untergebracht werden.
3. Hohe Schaltgeschwindigkeit: Aufgrund ihrer geringen Größe können Reed-Schalter sehr schnell schalten, was sie ideal für Anwendungen wie Schaltkreise und Sensoren macht.
4. Hohe Zuverlässigkeit: Da Reed-Schalter hermetisch verschlossen sind, sind sie gegenüber Feuchtigkeit, Staub und anderen Umwelteinflüssen geschützt, was ihre Lebensdauer und Zuverlässigkeit erhöht.
5. Geringer Stromverbrauch: Reed-Schalter benötigen sehr wenig Strom, um zu schalten, was sie ideal für batteriebetriebene Anwendungen macht.
6. Hohe Empfindlichkeit: Reed-Schalter können bereits auf sehr schwache magnetische Felder reagieren, was sie ideal für Anwendungen wie Magnetfeldsensoren und Strommessgeräte macht.

<https://de.wikipedia.org/wiki/Reedschalter>

Eigenschaften des Reedschalters:

- Öffner: immer 2-Leiter-Technik
- Schließer: immer 2-Leiter-Technik
- Wechsler (selten): 3-Leiter
- Es ist im Datenblatt immer nur die MAX-Spannung angegeben
- Stromrichtung: egal
 - Masse-Klemme: nicht vorhanden
 - in der Regel keine LED (Sonderbauformen ausgenommen)
 - nicht kurzschlussgeschützt !!!

<https://www.ifm.com/de/de/product/MR0100>

7.1.2 Magnetsensoren mit Halbleitertechnik

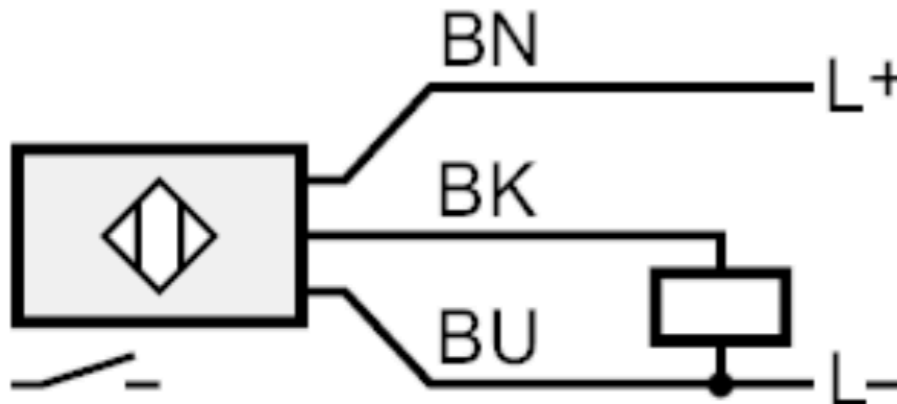
https://de.wikipedia.org/wiki/Magnetoresistiver_Effekt

Immer Angegeben: z.B. 5...36 V DC

- Minimal und Maximalspannung angegeben !!!
- Öffner: meist 3-Leiter-Technik
- Schließer: meist 3-Leiter-Technik
- Wechsler: 4-Leiter
- Stromrichtung: wichtig, siehe PNP/NPN, L+ L-
 - Masse-Klemme: nicht vorhanden
 - in der Regel keine LED (Sonderbauformen ausgenommen)
 - moderne Bauarten sind kurzschlussgeschützt, kann nicht durchbrennen

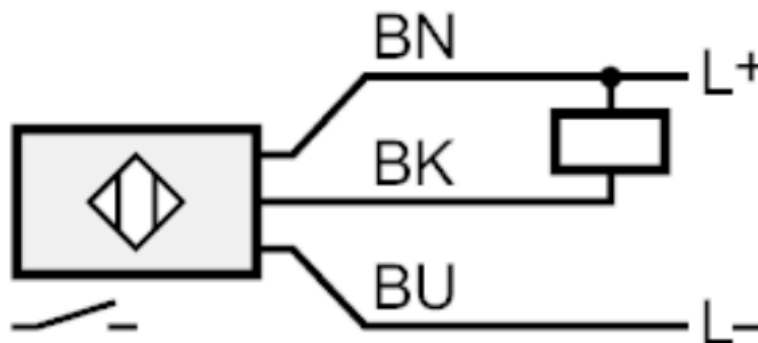
Beispiel: <https://www.ifm.com/de/de/product/MK5214> (kurzschlussfest; überlastfest)

7.1.2.1 PNP Sensor



Quelle: <https://www.ifm.com/de/de/product/MK5140>

7.1.2.2 NPN - Sensor



Quelle: <https://www.ifm.com/de/de/product/MK5309>

7.1.2.3 AMR-Zelle

Eine AMR-Zelle (Anisotropic Magnetoresistive Cell) ist eine Art von magnetoresistiver Zelle, die in der Halbleiter- und Magnetindustrie eingesetzt wird. Die AMR-Zelle besteht aus einem dünnen Schichtsystem aus ferromagnetischen und nicht-magnetischen Schichten, die auf einem Siliziumsubstrat aufgebracht sind.

Die wichtigsten Eigenschaften einer AMR-Zelle sind:

1. Magnetische Empfindlichkeit: Die AMR-Zelle ist sehr empfindlich auf magnetische Felder und kann geringe magnetische Veränderungen erkennen.
2. Geringer Stromverbrauch: Die AMR-Zelle benötigt sehr wenig Strom, um zu funktionieren.

3. Hohe Genauigkeit: Die AMR-Zelle kann sehr präzise Messungen durchführen und ist daher ideal für Anwendungen, bei denen eine hohe Genauigkeit erforderlich ist, wie beispielsweise in der Magnetfeldsensorik.
4. Hohe Geschwindigkeit: Die AMR-Zelle kann sehr schnell auf Veränderungen des magnetischen Feldes reagieren und ist daher ideal für Anwendungen, die schnelle Messungen erfordern, wie beispielsweise in der Datenverarbeitung.
5. Geringe Kosten: Die AMR-Zelle ist vergleichsweise kostengünstig und einfach zu produzieren, was sie für eine Vielzahl von Anwendungen geeignet macht.

Insgesamt ist die AMR-Zelle ein sehr vielseitiges Bauelement, das in vielen verschiedenen Anwendungen eingesetzt werden kann, von der Magnetfeldsensorik bis hin zur Datenverarbeitung und vielen anderen Bereichen.

7.1.2.4 GMR-Zelle

Eine GMR-Zelle (Giant Magnetoresistive Cell) ist eine Art von magnetoresistiver Zelle, die in der Halbleiter- und Magnetindustrie eingesetzt wird. Die GMR-Zelle besteht aus mehreren dünnen Schichten aus ferromagnetischen und nicht-magnetischen Materialien, die in einer Schichtstruktur auf einem Substrat aufgebracht sind.

Die wichtigsten Eigenschaften einer GMR-Zelle sind:

1. Hohe Empfindlichkeit: Die GMR-Zelle ist sehr empfindlich auf magnetische Felder und kann sehr geringe Veränderungen erkennen.
2. Hohe Genauigkeit: Die GMR-Zelle kann sehr präzise Messungen durchführen und ist daher ideal für Anwendungen, bei denen eine hohe Genauigkeit erforderlich ist, wie beispielsweise in der Magnetfeldsensorik.
3. Geringer Stromverbrauch: Die GMR-Zelle benötigt sehr wenig Strom, um zu funktionieren.
4. Hohe Geschwindigkeit: Die GMR-Zelle kann sehr schnell auf Veränderungen des magnetischen Feldes reagieren und ist daher ideal für Anwendungen, die schnelle Messungen erfordern, wie beispielsweise in der Datenverarbeitung.
5. Hohe Temperaturstabilität: Die GMR-Zelle ist sehr temperaturstabil und kann auch bei hohen Temperaturen eingesetzt werden.
6. Geringe Kosten: Die GMR-Zelle ist vergleichsweise kostengünstig und einfach zu produzieren, was sie für eine Vielzahl von Anwendungen geeignet macht.

Insgesamt ist die GMR-Zelle ein sehr vielseitiges Bauelement, das in vielen verschiedenen Anwendungen eingesetzt werden kann, von der Magnetfeldsensorik bis hin zur Datenverarbeitung und vielen anderen Bereichen.

<https://www.ifm.com/de/de/product/MK5117>

7.2 Anschluss Endschalter an Eingang



M5 Stack:

Eingangsspannung am Pin:

3.3V

Spannung vom Sensor:

5V

Spannungsteiler:

Startseite > Umwandlungsrechner > Kalkulator für Spannungsteiler



Kalkulator für Spannungsteiler

Mit diesem Tool können Sie die Ausgangsspannung einer Widerstandsteilerschaltung für einen gegebenen Satz von Widerstandswerten und eine gegebene Quellenspannung berechnen.

[+Mehr](#)

Eingangsspannung (V_1)
 V

Erster Widerstand (R_1)
 Ω

Zweiter Widerstand (R_2)
 Ω

Ausgabe
Ausgangsspannung (V_{out})

3.3

Unter Last [Anzeigen](#)

No Load

$$V_{out} = V_1 \frac{R_2}{R_1 + R_2} = \frac{V_1 R_2}{(R_1 + R_2)}$$

<https://www.digikey.de/de/resources/conversion-calculators/conversion-calculator-voltage-divider>
in der Praxis:

1. Widerstand 1800Ohm
2. Widerstand 3300Ohm

Weil:

<https://www.electronicplanet.ch/Widerstand/Widerstandsreihe-E24.htm>

Startseite > Umwandlungsrechner > Kalkulator für Spannungsteiler



Kalkulator für Spannungsteiler

Mit diesem Tool können Sie die Ausgangsspannung einer Widerstandsteilerschaltung für einen gegebenen Satz von Widerstandswerten und eine gegebene Quellenspannung berechnen.

[+Mehr](#)

Eingangsspannung (V_1)
 V

Erster Widerstand (R_1)
 Ω

Zweiter Widerstand (R_2)
 Ω

Ausgabe
Ausgangsspannung (V_{out})

3,2352941176

Unter Last [Anzeigen](#)

No Load

$$V_{out} = V_1 \frac{R_2}{R_1 + R_2} = \frac{V_1 R_2}{(R_1 + R_2)}$$

7.3 Grove System

<https://www.seeedstudio.com/category/Grove-c-1003.html>

<https://exp-tech.de/search?type=product&q=grove>

<https://www.distrelec.de/search?q=grove>

7.4 Reihenschaltung von Endschaltern

https://www.baumer.com/de/de/service-support/inbetriebnahme-montage/inbetriebnahme-und-montage-von-induktiven-sensoren/a/Know-how_Mounting_Inductive-sensors

7.5 ButtonBoards

<https://github.com/Meisterschulen-am-Ostbahnhof-Munchen/ButtonBoards>
Multiplexer

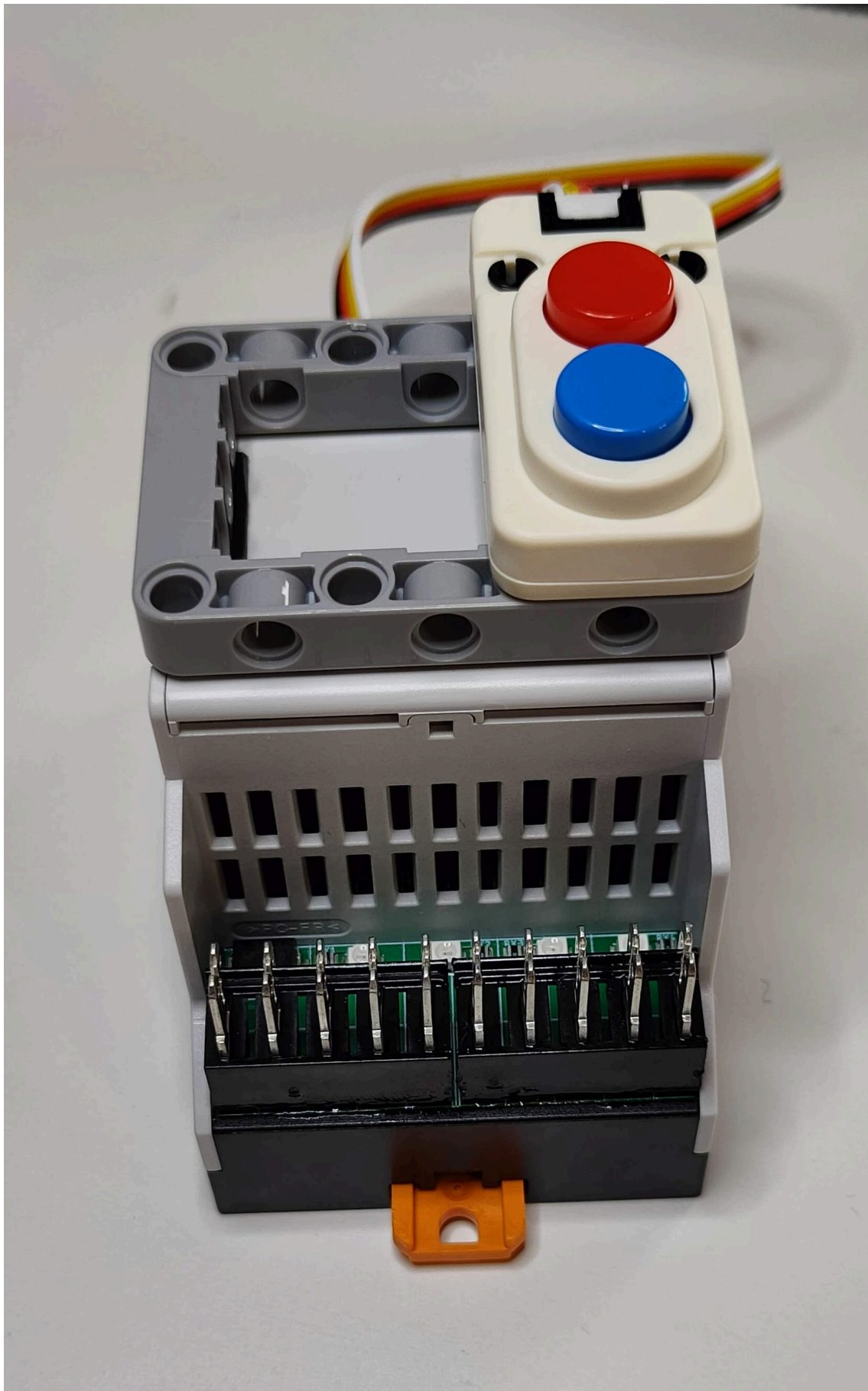
7.5.1 Passende Themen-Unterseiten auf ms-muc-docs.de

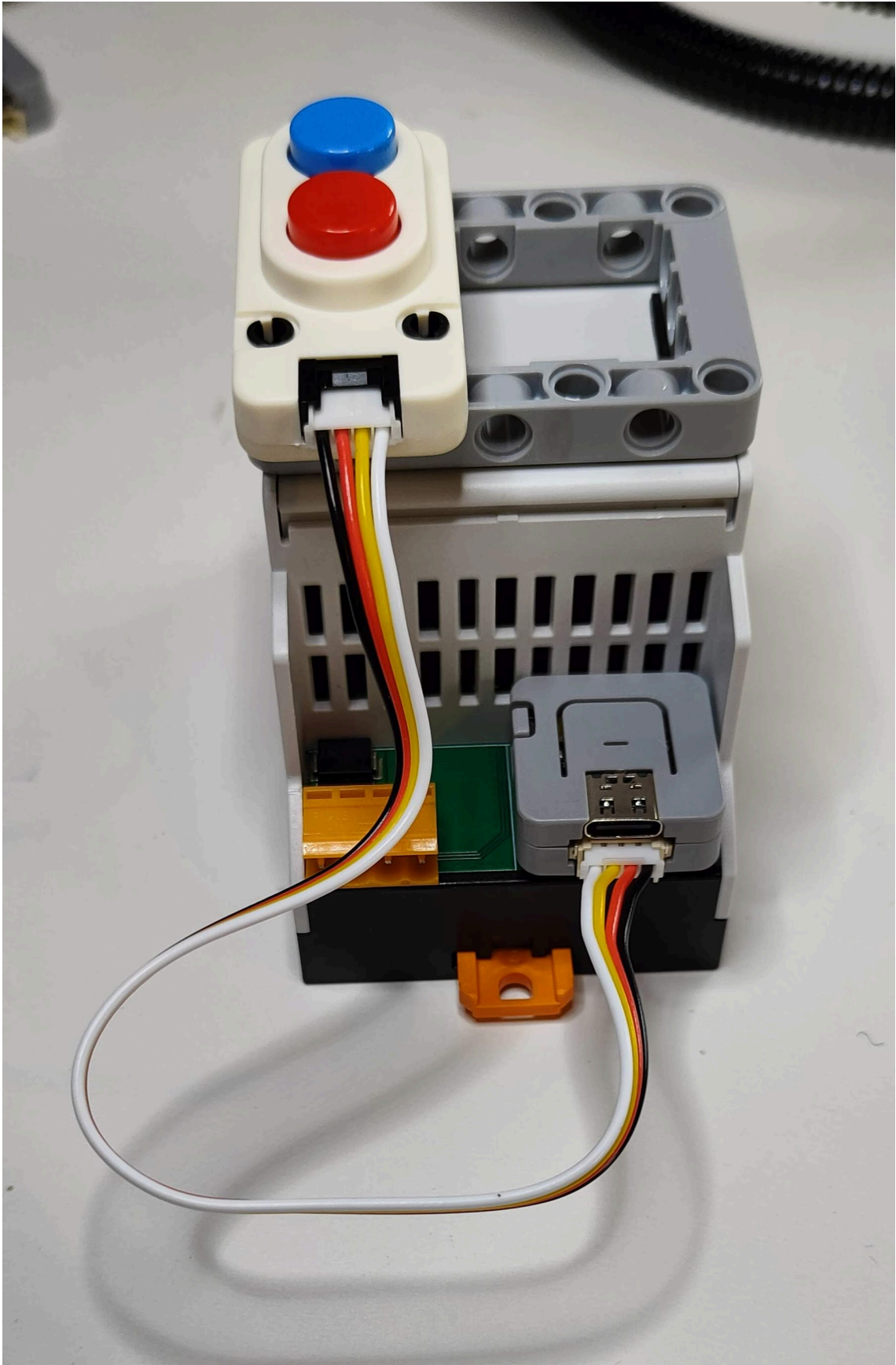
-  Gesamtwiderstand in Reihen- & Parallelschaltung auf ms-muc-docs.de
-  Belasteter & Unbelasteter Spannungsteiler auf ms-muc-docs.de
-  Bipolartransistor Grundlagen auf ms-muc-docs.de

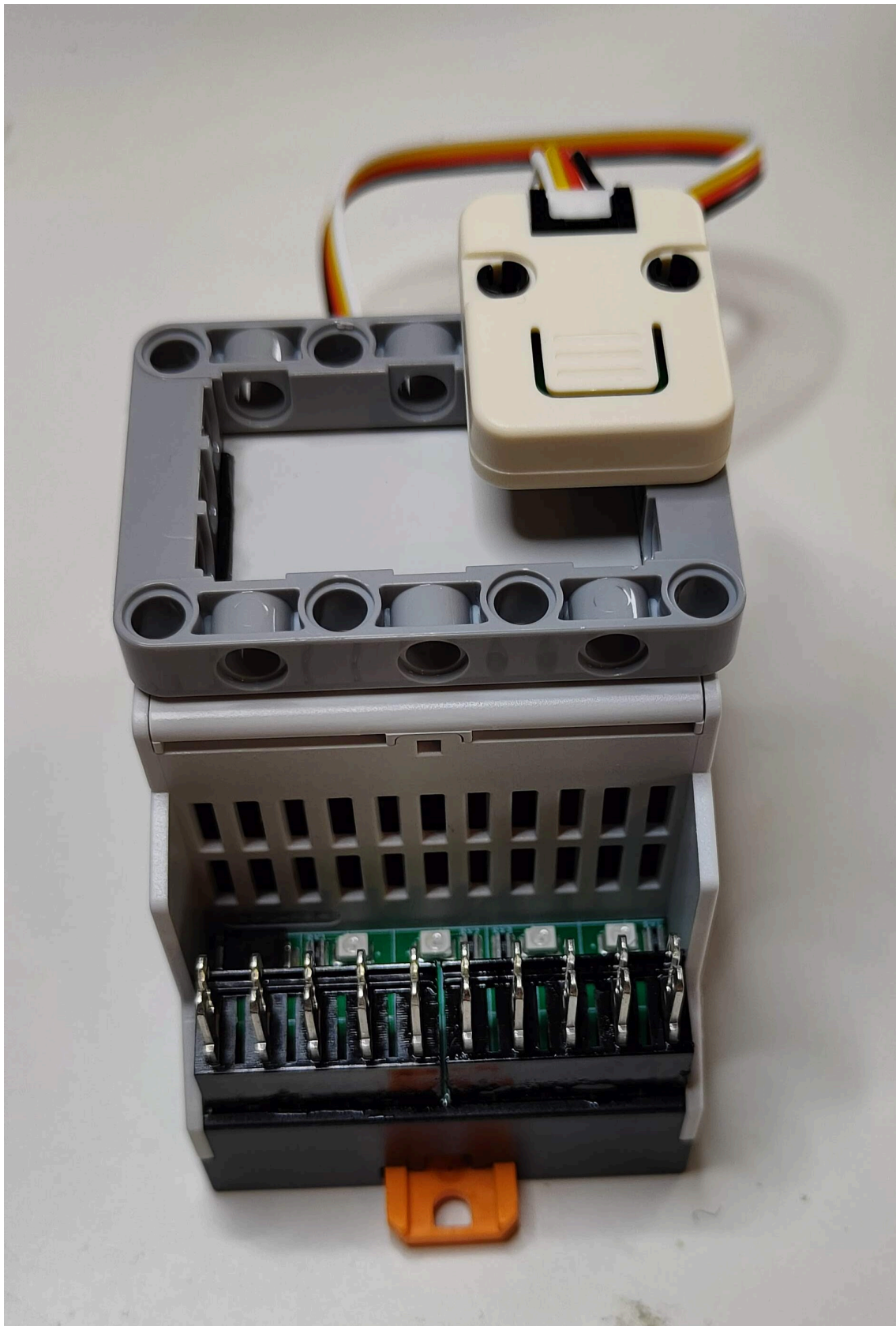
8. HutschienenMoped Erweiterungen

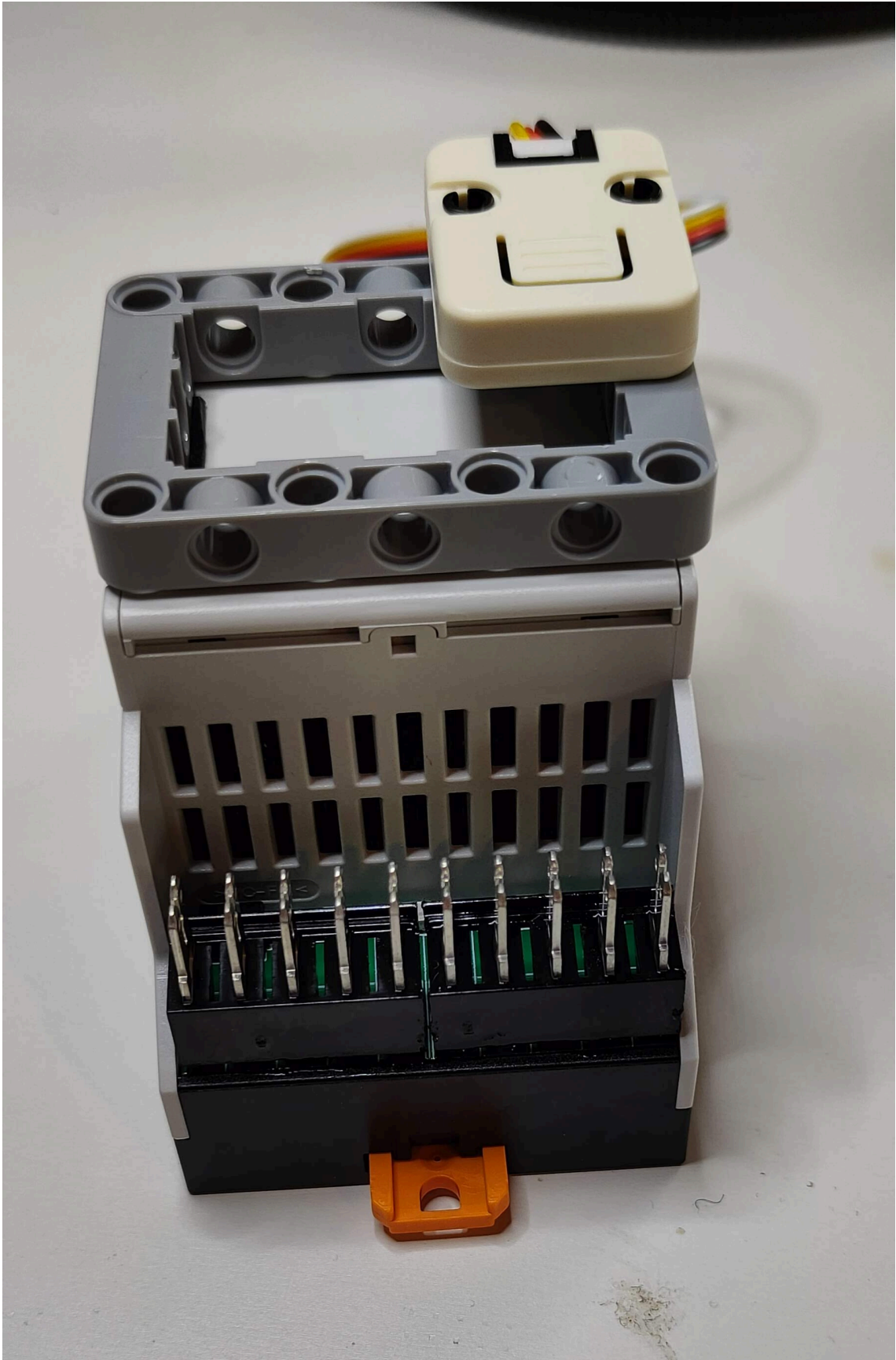
siehe auch:

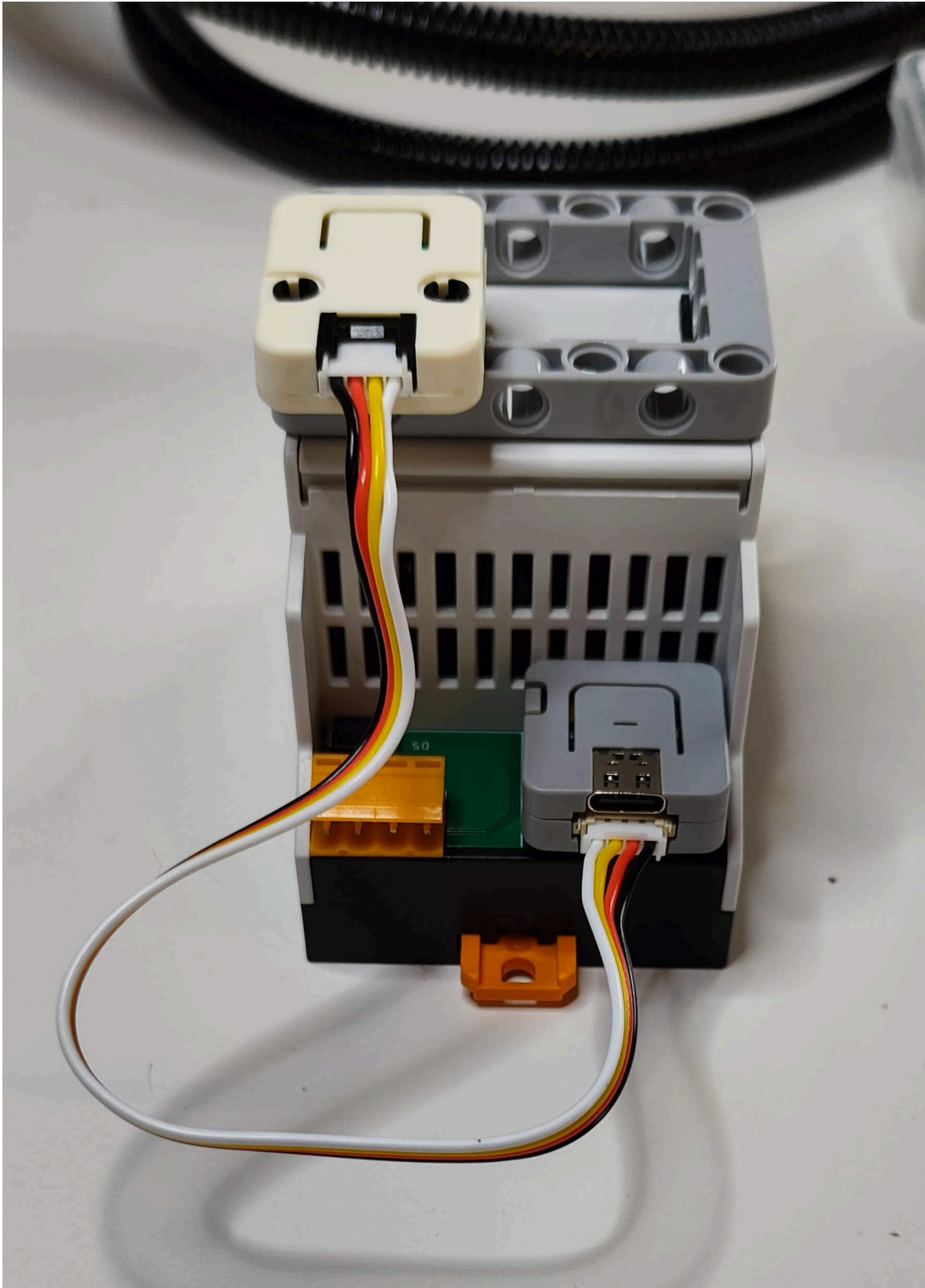
<https://install-isobus-environment-docs.readthedocs.io/de/latest/Atom-Erweiterungen.html>

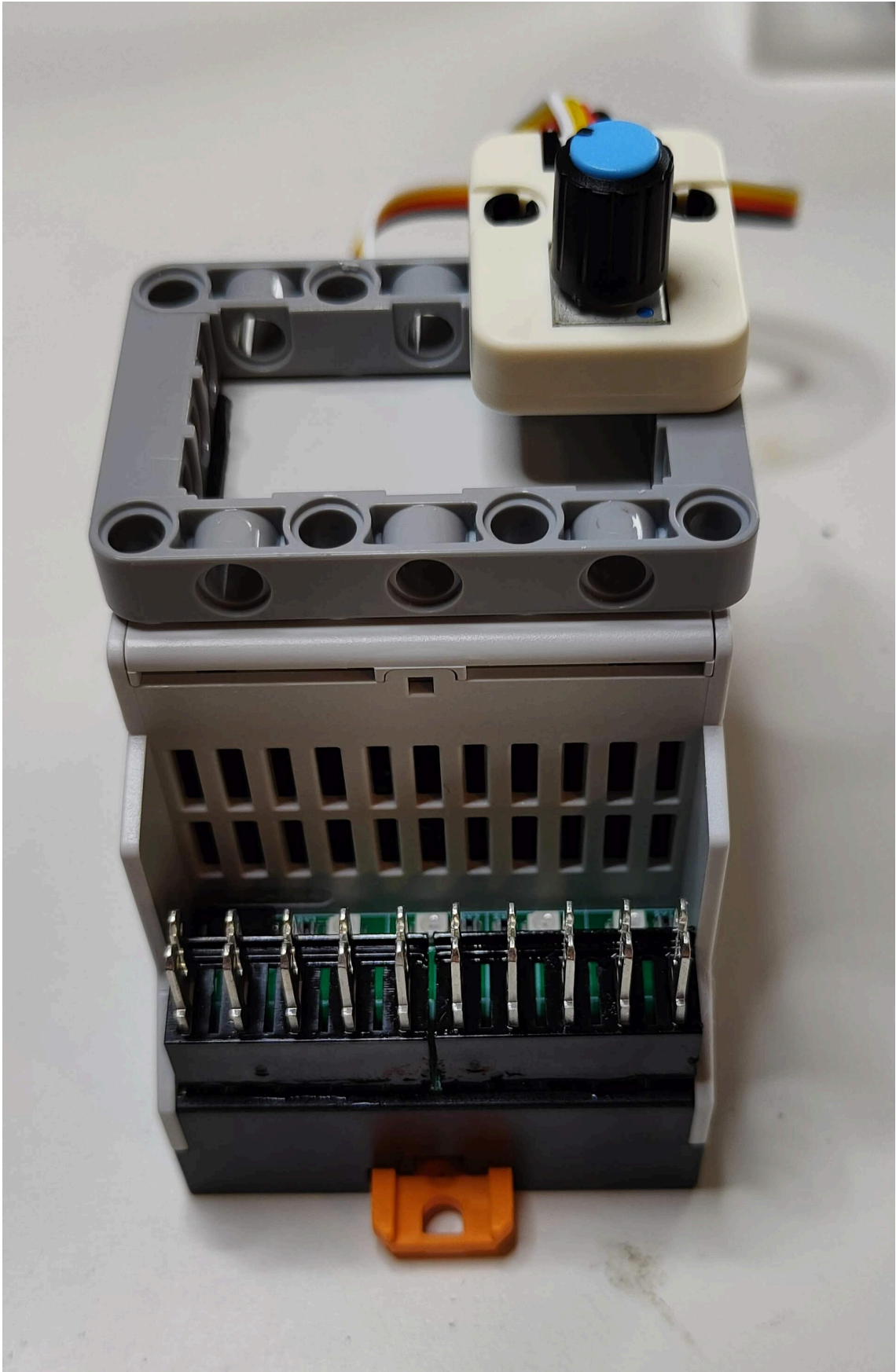


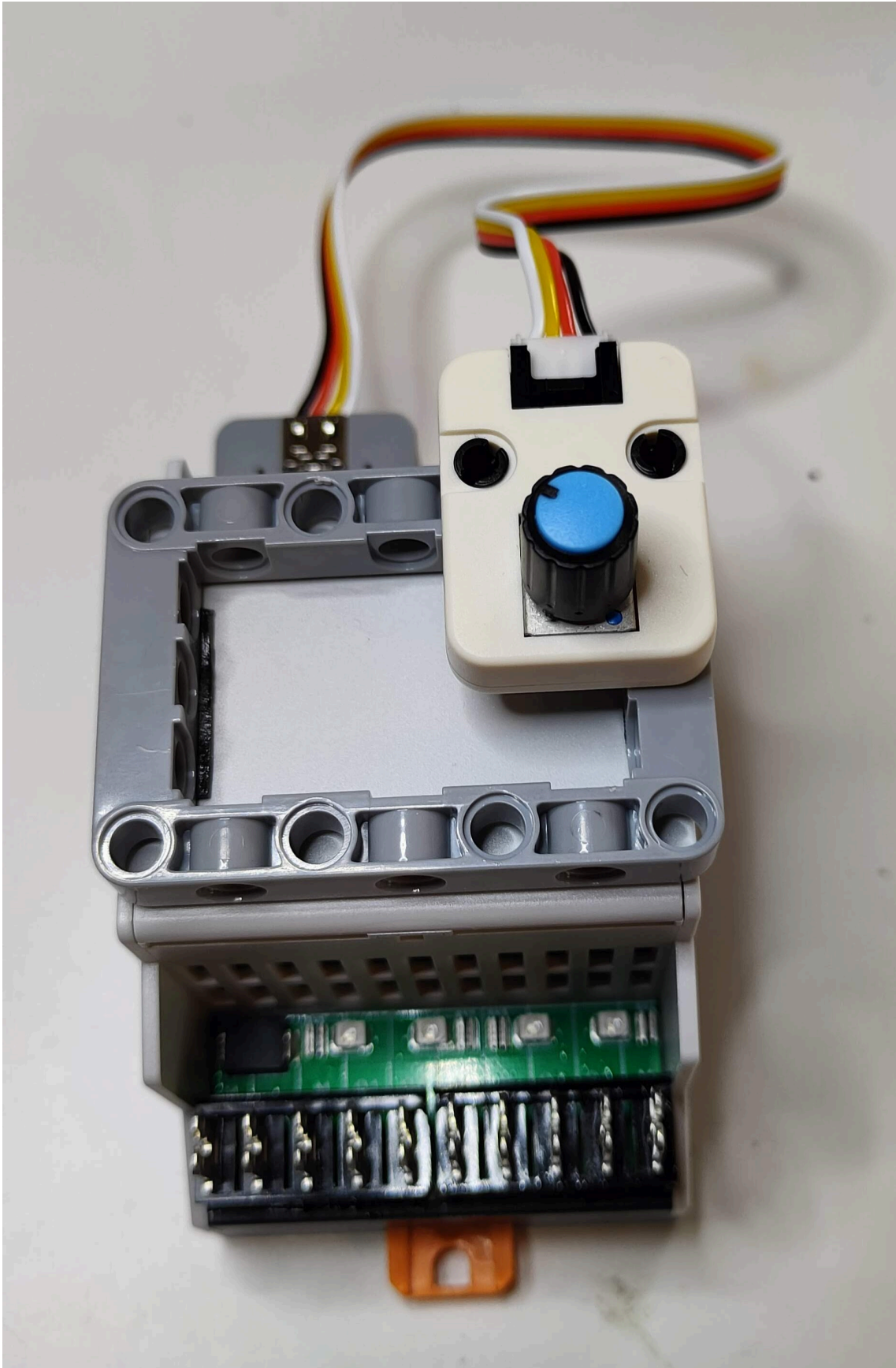


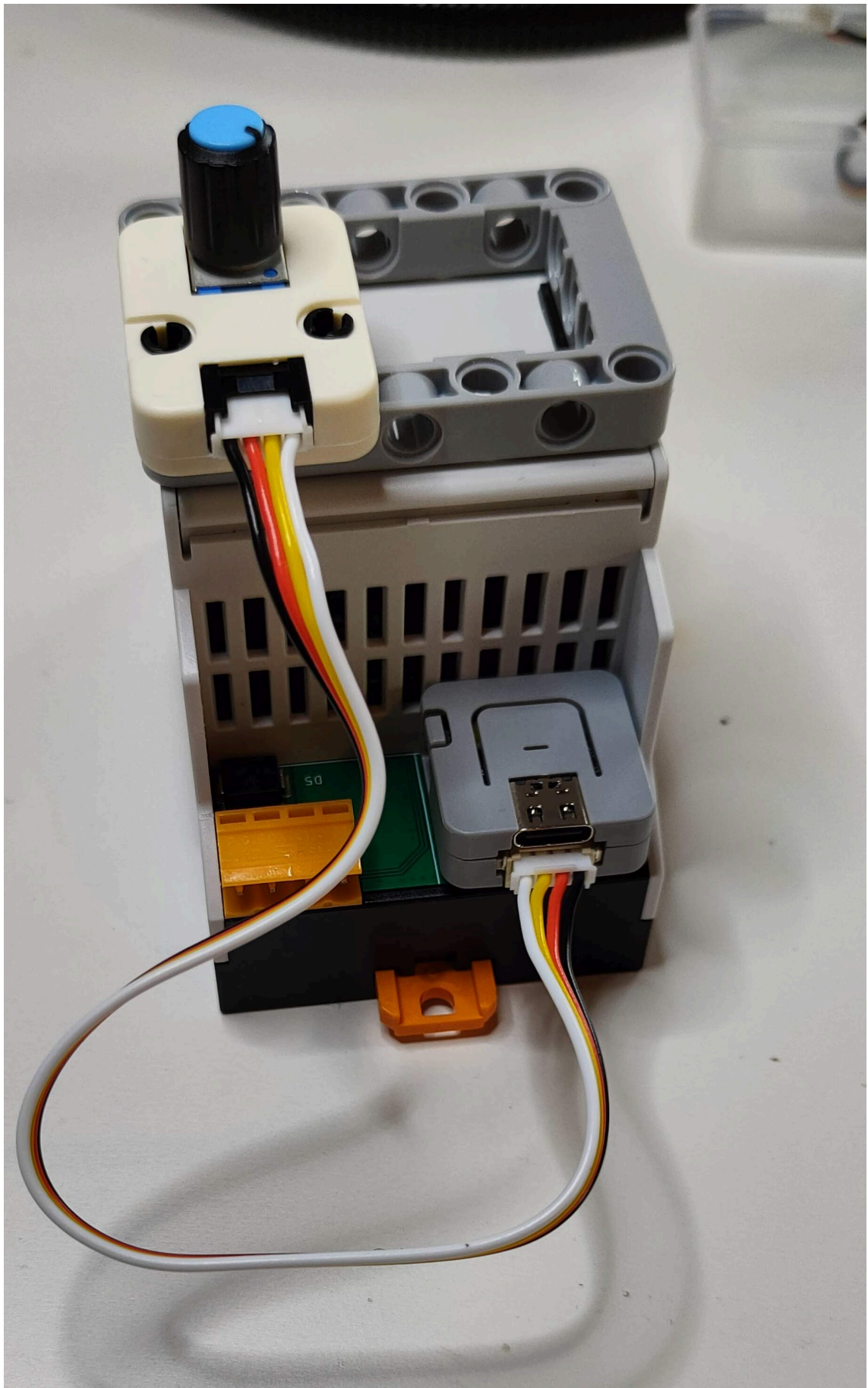












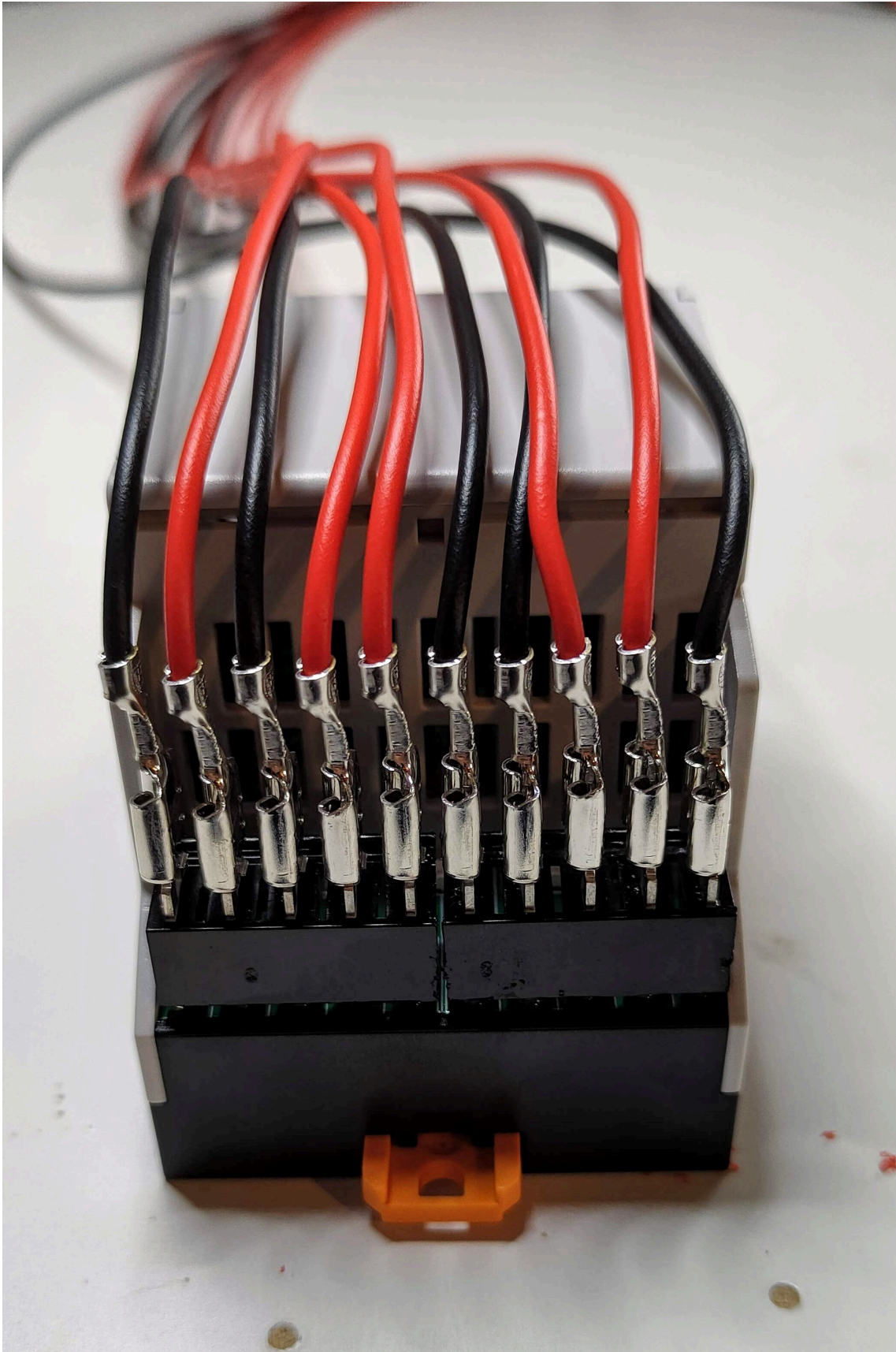
9. HutschienenMoped Stecker

Diese Fotostrecke dokumentiert die korrekte Verkabelung und die verwendeten Steckertypen für das „HutschienenMoped“-Modul.

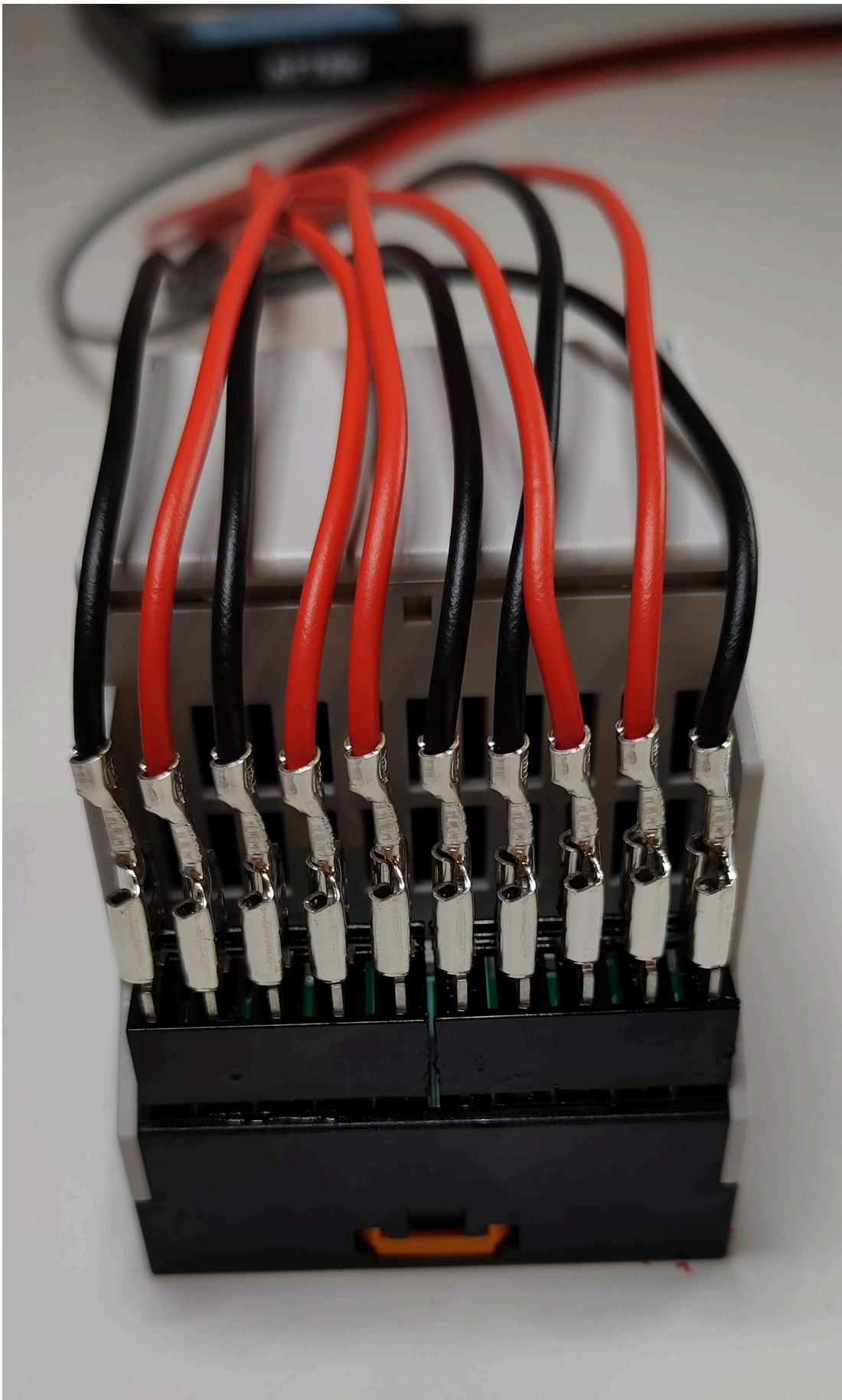
9.1 1. Grundlegende Verdrahtung und Anschlüsse

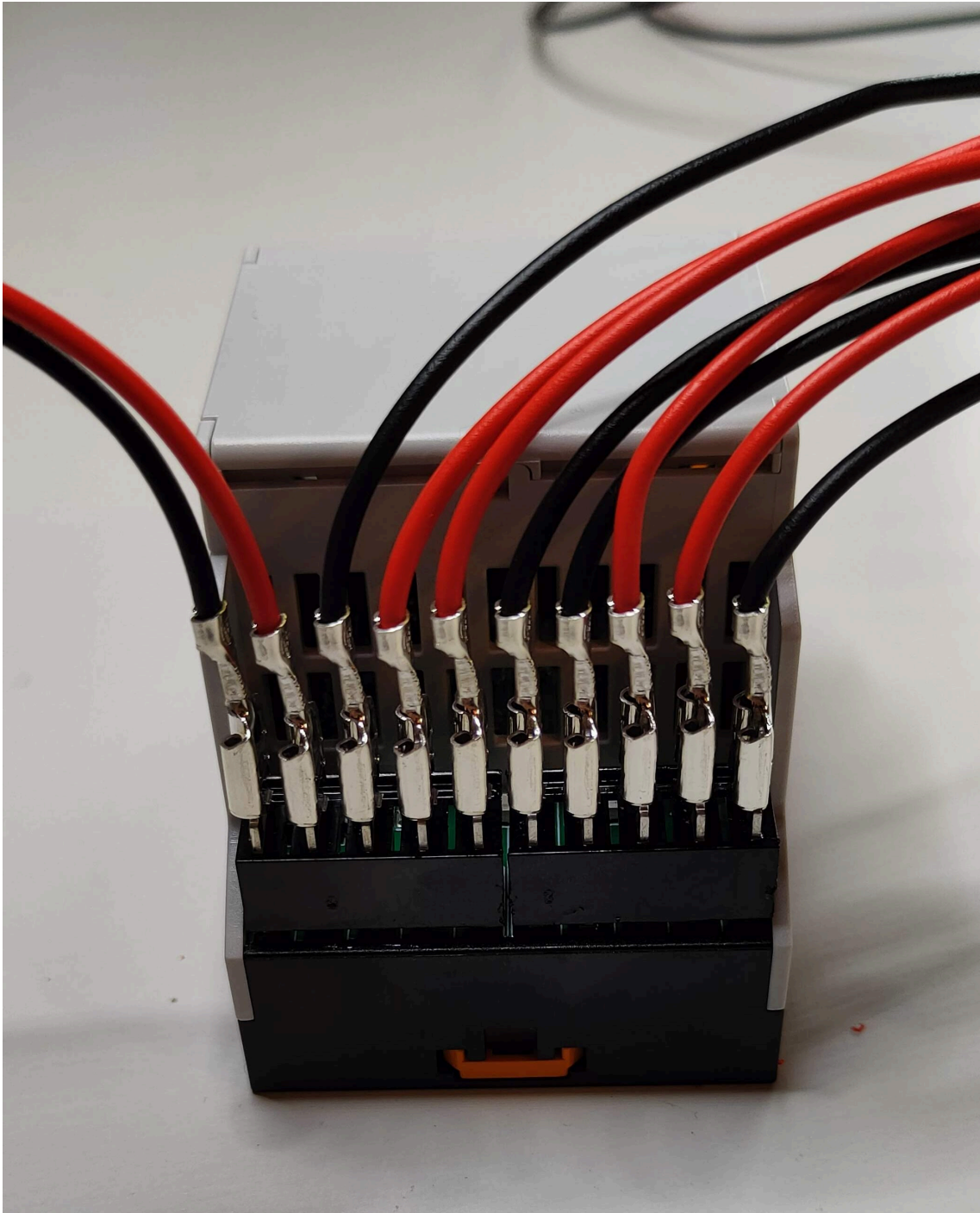
Die Basisverdrahtung des Moduls auf der Hutschiene erfolgt über rote und schwarze Leitungen (Spannungsversorgung und Masse), die über Flachsteckhülsen mit den Stiftleisten der Platine verbunden sind.

- **Steckertyp:** Standardmäßig kommen hier **6,3 mm Flachsteckhülsen** zum Einsatz.



6.3mm Flachsteckhülsen





9.2 2. Isolierung und Berührungsschutz

Um Kurzschlüsse zwischen den eng beieinanderliegenden Kontakten zu vermeiden und die Berührungssicherheit zu gewährleisten, werden die Flachsteckhülsen isoliert. Rote und schwarze Isoliertüllen (oder Schrumpfschläuche) umschließen die Crimpstellen und den hinteren Teil der Steckverbinder vollständig.



9.3 3. Detailansicht: Spezielle Signalanschlüsse

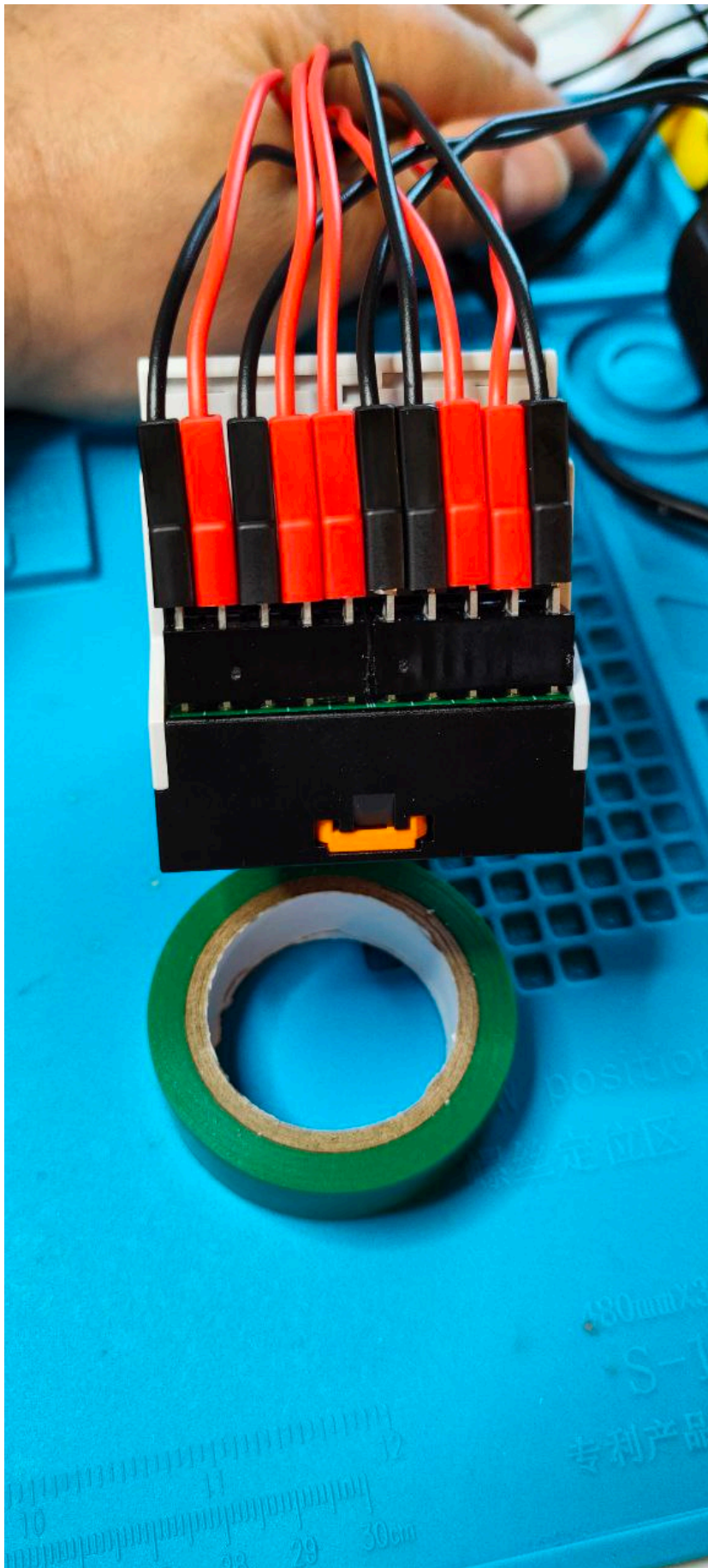
Für spezifische Anschlüsse oder bei engeren Platzverhältnissen werden abweichende Größen verwendet. Das Bild zeigt den Einsatz von **zwei 2,8 mm Flachsteckhülsen**, die nebeneinander auf einem Anschlussblock sitzen. Dies ist wichtig für Kontakte, die mechanisch nicht für die breiteren 6,3 mm Hülsen ausgelegt sind.

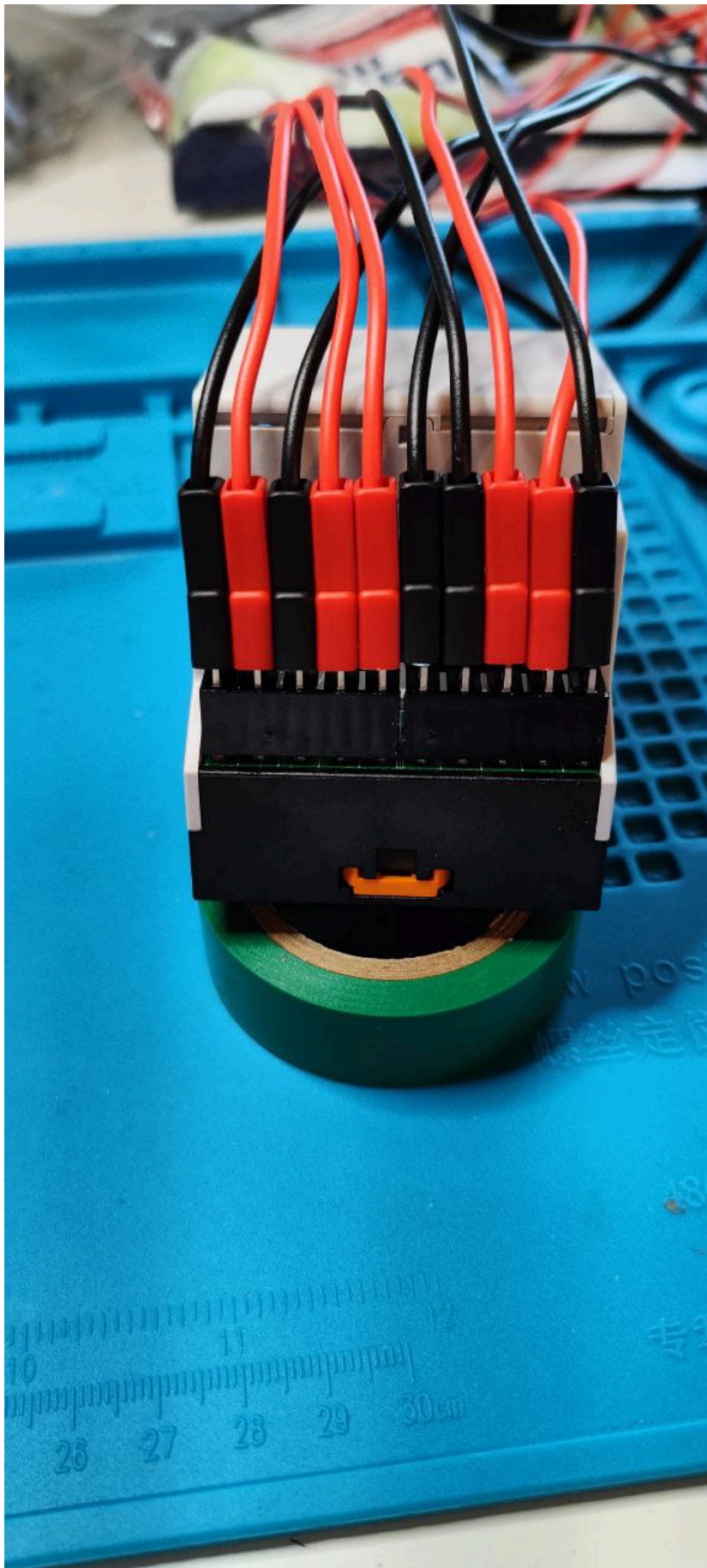


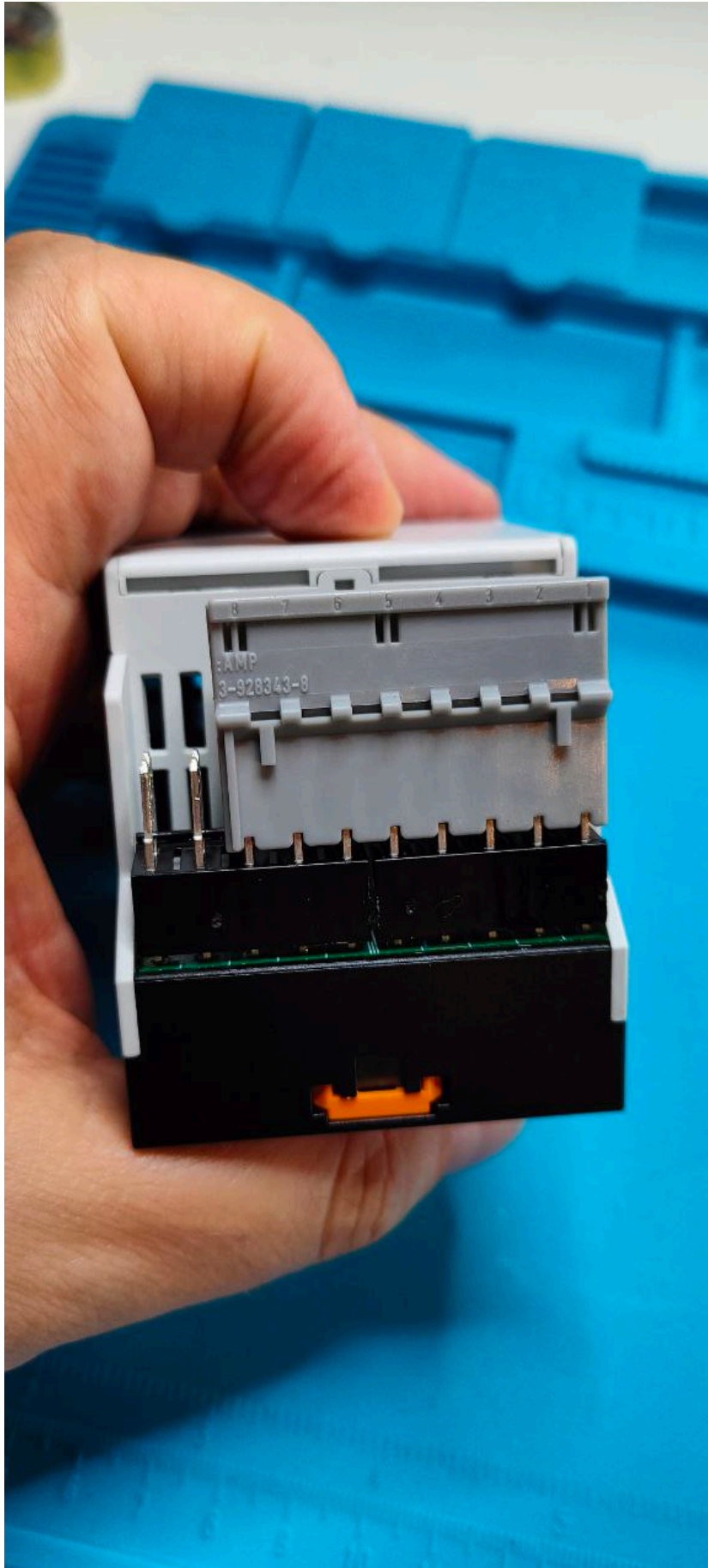
2x 2.8mm Flachsteckhülsen

9.4 4. Steckverbindungs-Gehäuse und Gesamtansicht

Die fertig isolierten Leitungen werden sauber geführt. Für eine modularere Verbindung kommt zudem ein graues Steckergehäuse der Marke AMP (TE Connectivity) mit der Teilenummer **3-328343-8** zum Einsatz, um mehrere Kontakte zu bündeln.

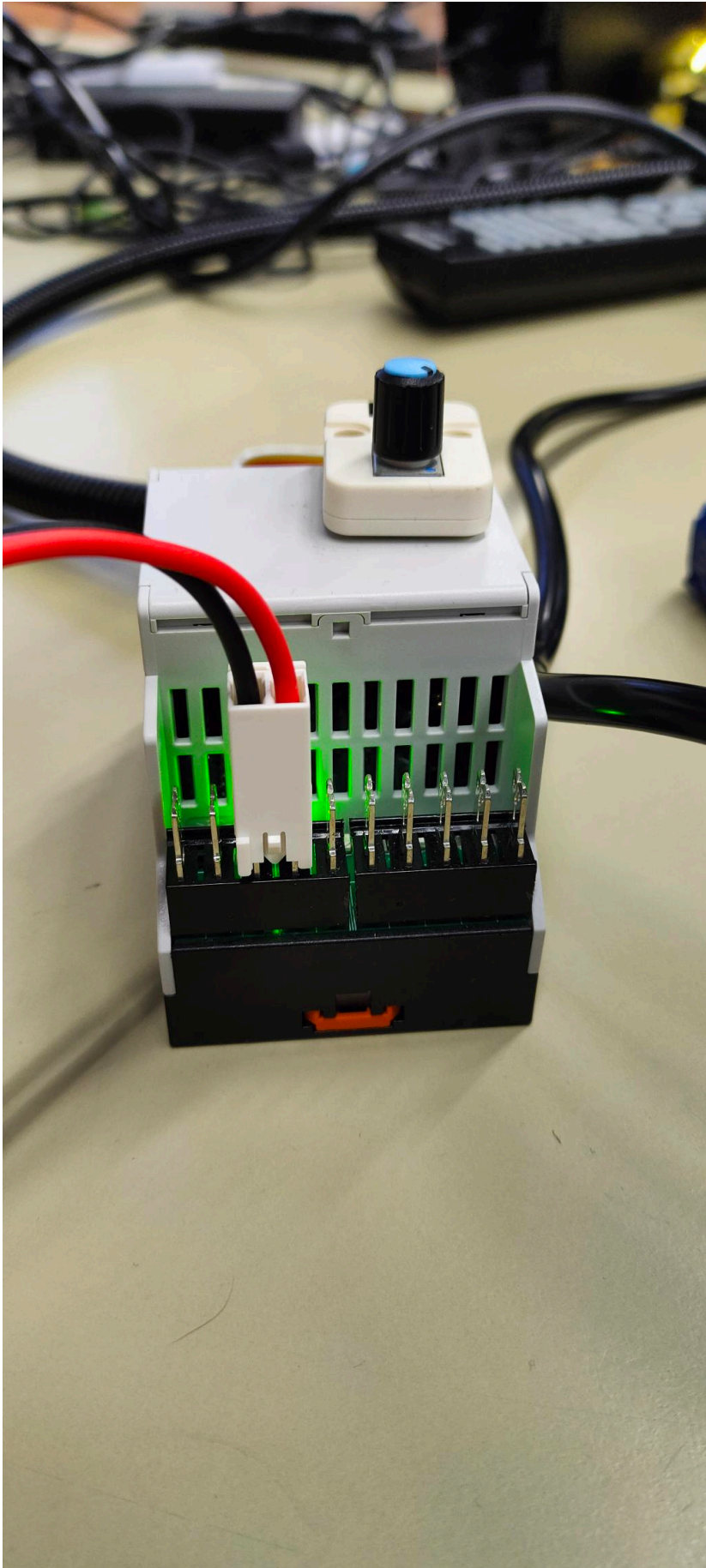






9.5 5. Betriebszustand

Das fertig verkabelte Modul im aktiven Betrieb: Die grünen LEDs leuchten im Inneren des Gehäuses, und die Peripherie ist angeschlossen.



10. HutschienenMoped

10.1 🎧 Podcast

- [Hutschienenmoped XL: ISOBUS ECU Entwicklung](#)

10.2 📁 Repository & Ressourcen

Der Quellcode und die Hardware-Dateien liegen im GitHub-Repository: 📁 [GitHub: Hutschienenmoped](#)

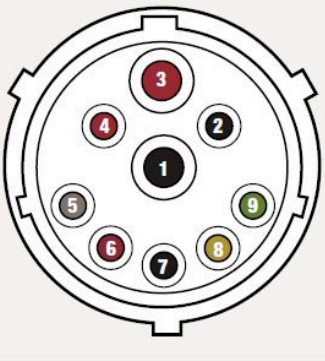
10.3 ⚡ Anschluss & Belegung

Das Hutschienenmoped verfügt über abgesicherte Ausgänge und flexible Anschlussmöglichkeiten.

- **Absicherung:** Die einzelnen Ausgänge sind durch MOSFETs abgesichert (siehe Datenblatt).
- **Stromversorgung:** Die Spannungseinspeisung benötigt zum Leitungsschutz eine externe Sicherung.

10.3.1 Pinbelegung

Pin	Bezeichnung	Farbe	Beschreibung	Spannung	Querschnitt/ Bemerkung
1	GND	schwarz	Masse	Masse	6 mm ²
2	ECU_GND	schwarz	Masse Steuereinheit	Masse	2,5 mm ²
3	PWR	rot	Leistungsstromversorgung	12 V	6 mm ²
4	ECU_PWR	rot	Stromversorgung der Steuereinheit	12 V	2,5 mm ²
5	TBC_DIS	-	Steuerung Terminierung		Brücke
6	TBC_PWR	rot	Stromversorgung für Terminierung	12 V	
7	TBC_RTN	schwarz	Masse für Terminierung	Masse	
8	CAN_H	gelb	Datenübertragung		verdrillt
9	CAN_L	grün	Datenübertragung		



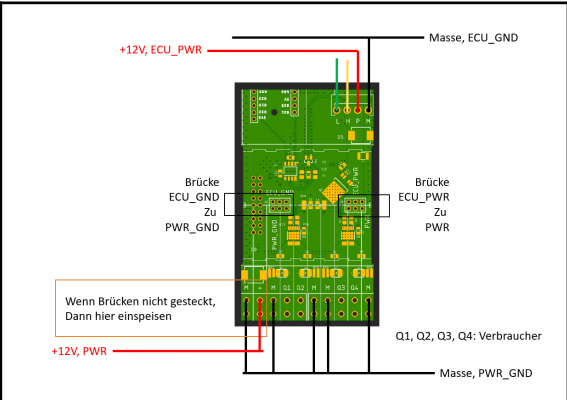
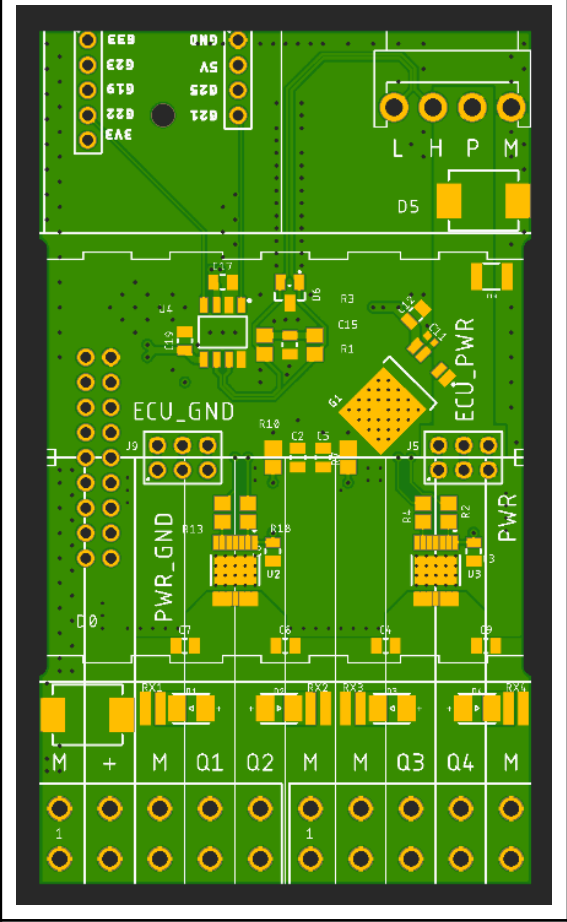
10.4 ⚙️ Konfiguration (4-Pol vs. 6-Pol)

Das Gerät kann je nach Anwendungsfall unterschiedlich betrieben werden.

10.4.1 Betrieb mit 4-poligem Kabel

Für den Betrieb mit einem 4-poligen Kabel müssen **zwei Brücken** gesteckt werden.

Ansicht	Beschreibung
---------	--------------

	Position der BrückenDetailansicht der Steckplätze.
	Schematische DarstellungVisualisierung der Brückenverbindung.

10.5 **Galerie & Montage**

Das Gehäuse ist für verschiedene Montageszenarien ausgelegt.

10.5.1 **Hardware-Ansichten**

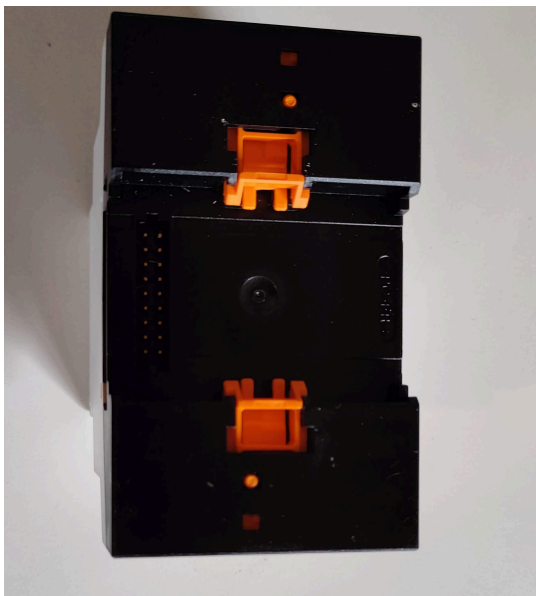
Ansicht	Beschreibung
---------	--------------



Gehäuse geschlossen Frontansicht.

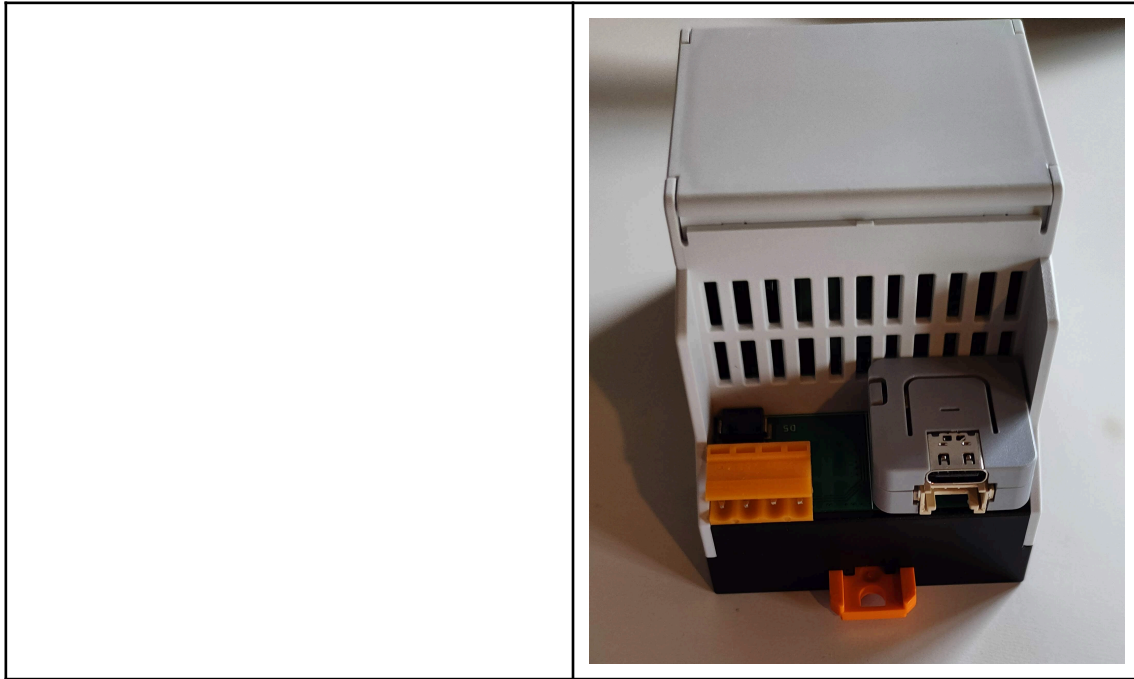
10.5.2 Montageoptionen

Tragschiene (Hutschiene)



Wandmontage





10.5.3 Passende Themen-Unterseiten auf ms-muc-docs.de

-  Leiterplatten-Entwurf & Bestückung auf ms-muc-docs.de

11. Hutschienenmoped XL

Das **Hutschienenmoped XL** ist die erweiterte Version unserer Open-Source-ECU. Es wurde speziell für die Integration in ISOBUS-Systeme entwickelt und eignet sich hervorragend für anspruchsvolle Steuerungsaufgaben in der Landtechnik.

11.1 🌟 Highlights

- **ISOBUS Kompatibel:** Volle Unterstützung für **ISO 11783**.
 - **IEC 61499:** Vorbereitet für verteilte Steuerungssysteme.
 - **Open Source:** Die Hardware ist vollständig offengelegt und zertifiziert.
 - **Erweiterbar:** Durch den modularen Aufbau und die Nutzung des ESP32 kann das System flexibel angepasst werden.
-

11.2 🏆 Zertifizierung

Das Hutschienenmoped XL ist offiziell als Open Source Hardware zertifiziert.

- **UID:** [DE000145](#)
- **Zertifizierungsstelle:** Open Source Hardware Association (OSHW)



11.3 Konstruktion & CAD

Das Hardware-Design wird in Autodesk Fusion 360 gepflegt. Hier können Sie die aktuellen 3D-Modelle und Schaltpläne einsehen:

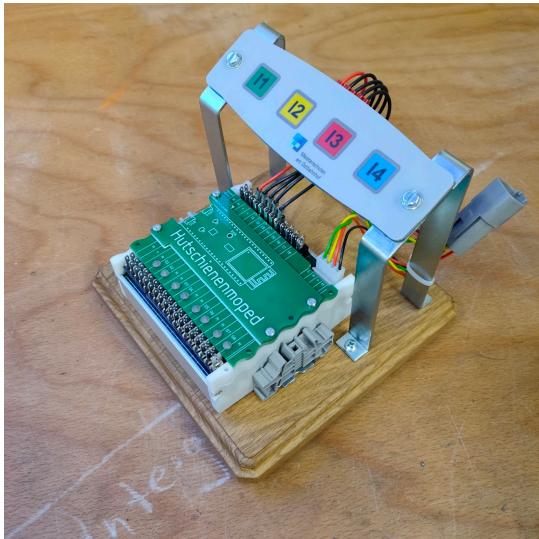
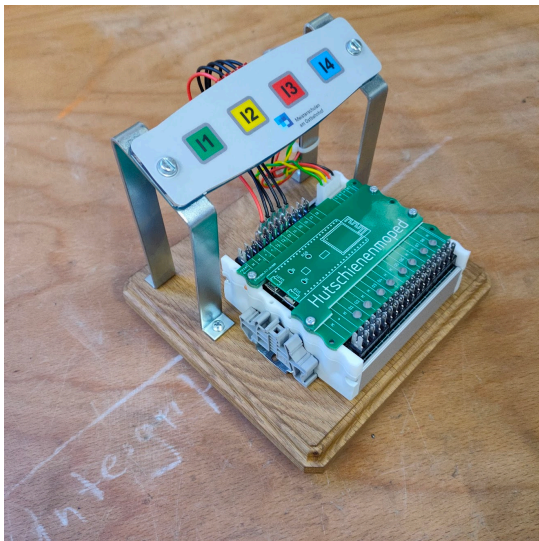
-  [Hauptplatine \(PCB\) in Fusion 360 ansehen](#)
-  [Gesamtzusammenbau \(Assembly\) in Fusion 360 ansehen](#)

11.3.1 Repository

Der Quellcode und die Hardware-Dateien liegen im GitHub-Repository:  [GitHub: Hutschienenmoped-XL](#)

11.4 Galerie

Hier einige Einblicke in die Hardware:

Ansicht	Beschreibung
	Bestückte Platine (Oben) Zeigt den ESP32 und die Anschlüsse.
	Bestückte Platine (Unten) Rückseite mit Leiterbahnenführung.

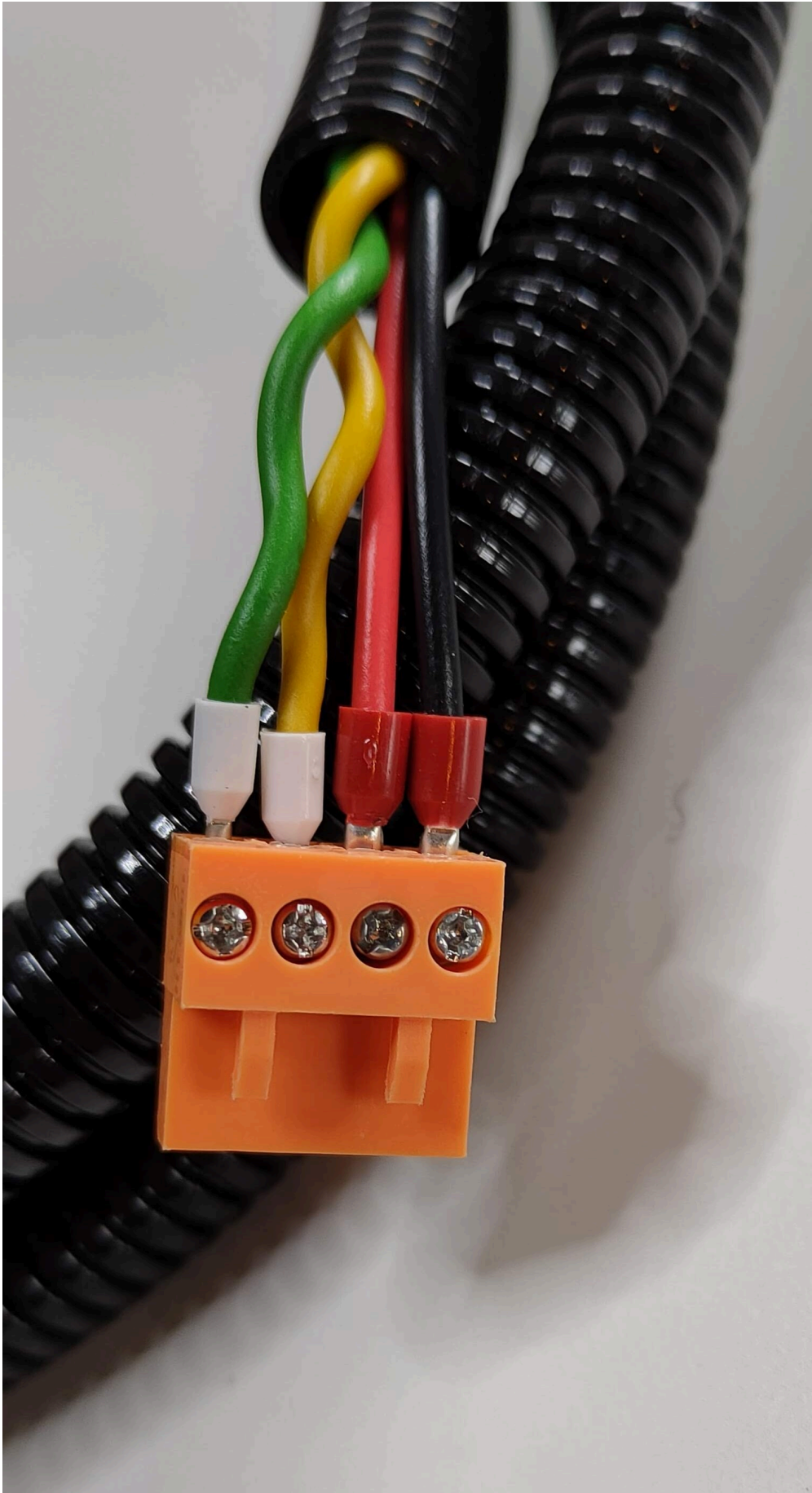
12. ISOBUS Kabel klein







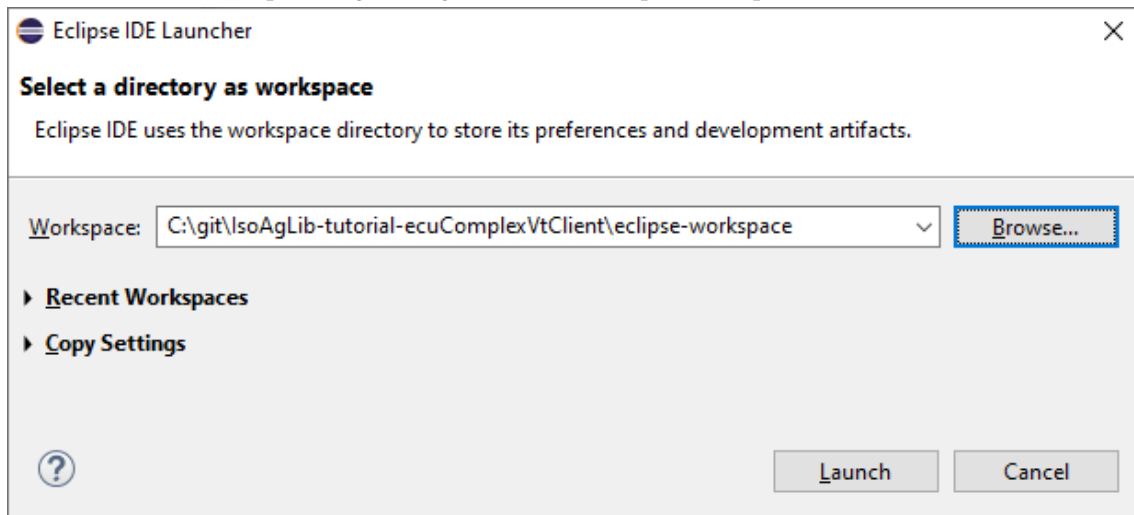




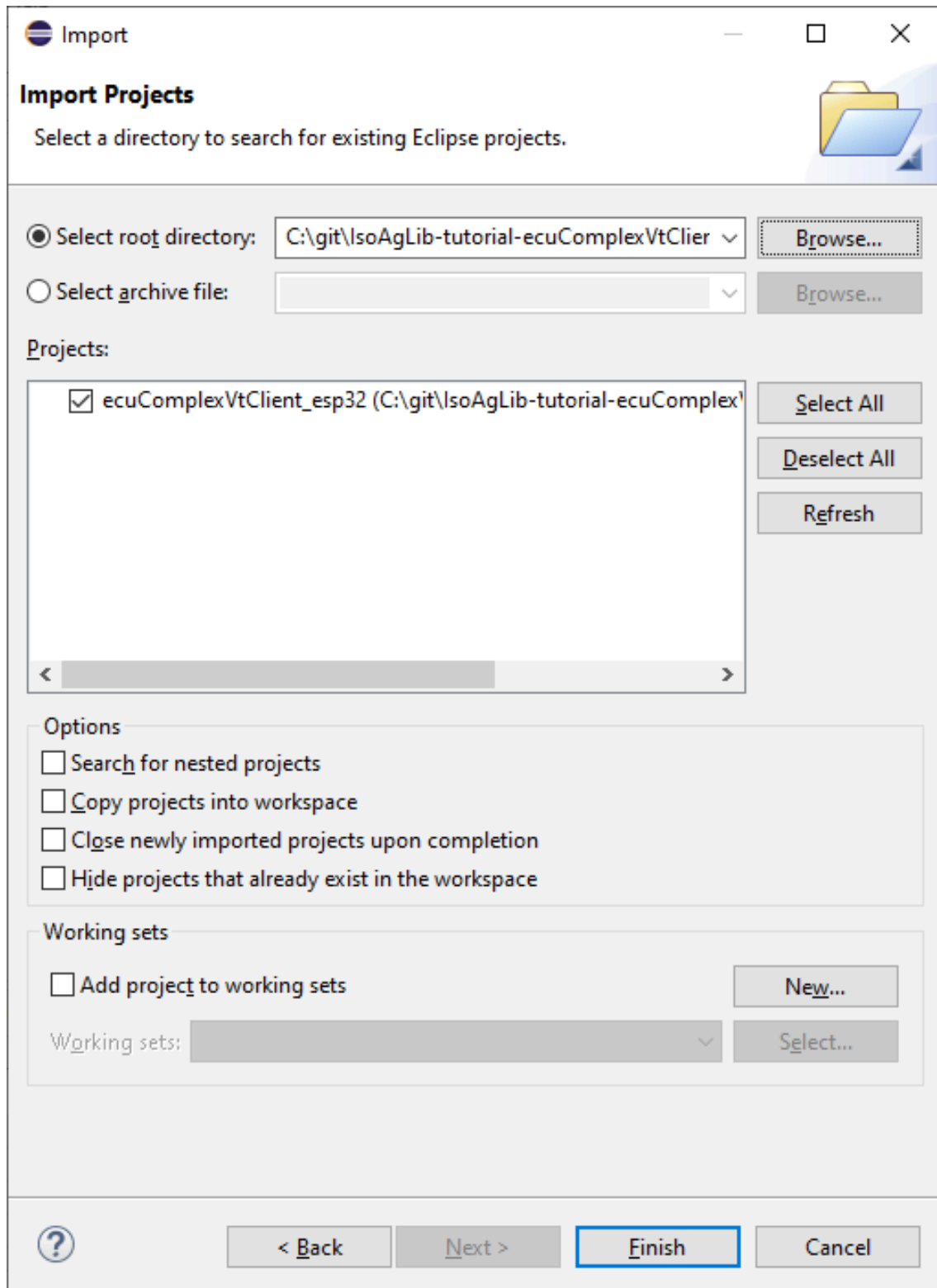
13. Installation

14. IsoAgLib-ISOMAN

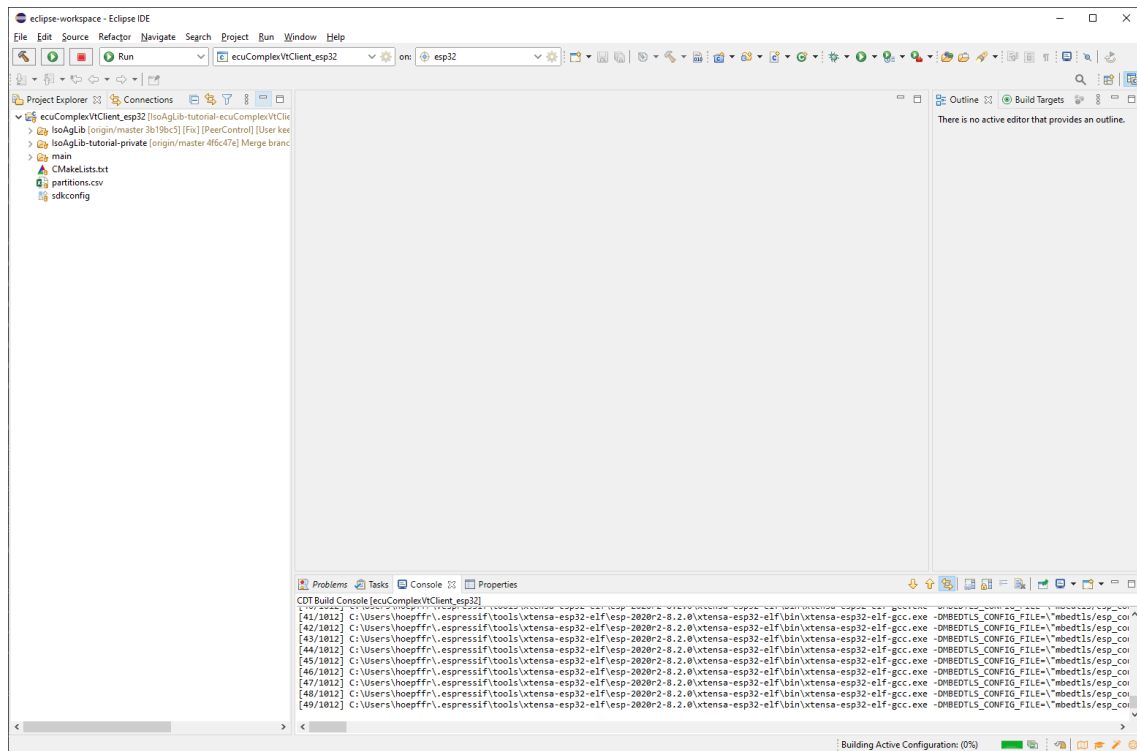
für diesen Versuch muss der Atom auf dem Sockel stecken. mit der Schraube vorsichtig sichern.
(das ist Kunststoff, da sind keine 500Nm Drehmoment nötig>`_
öffnen Sie den Workspace C:\git\IsoAgLib-ISOMAN\eclipse-workspace



Import Project wie vorher (das können Sie jetzt schon>`_



Launch Target anlegen,
Download and Configure ESP-IDF (wie vorher>`_
Dann Hammer,
dann Play:

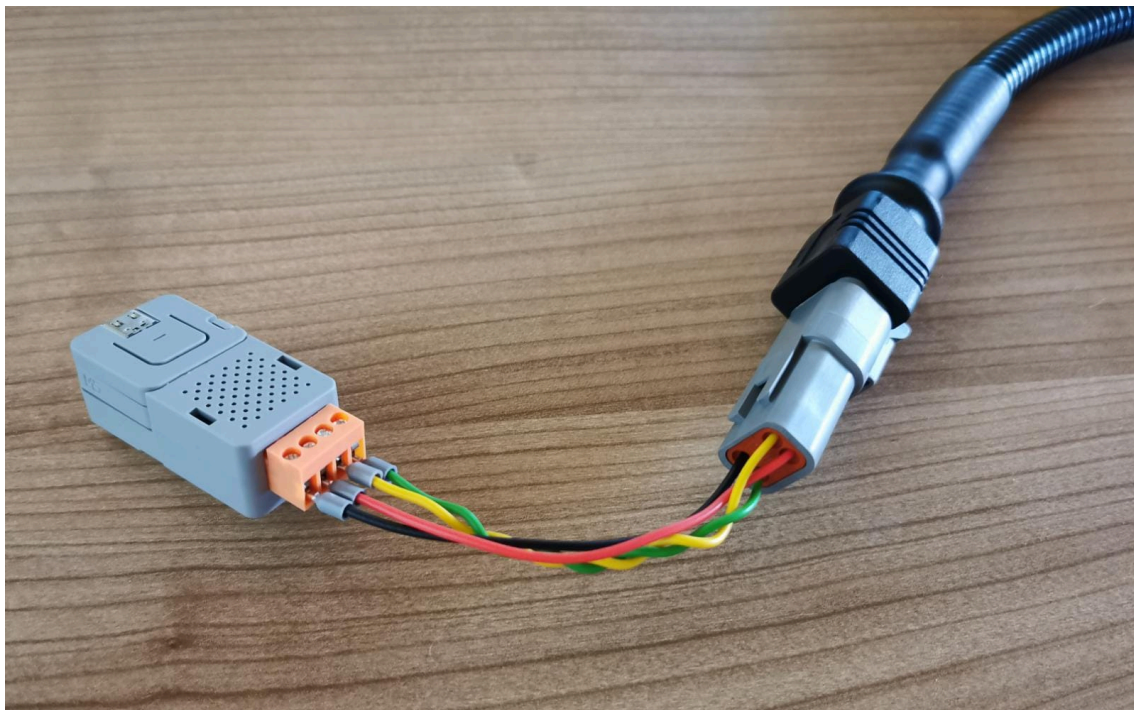


so,

jetzt steigen Sie auf den Fendt und schließen das Gerät an und schauen was passiert.

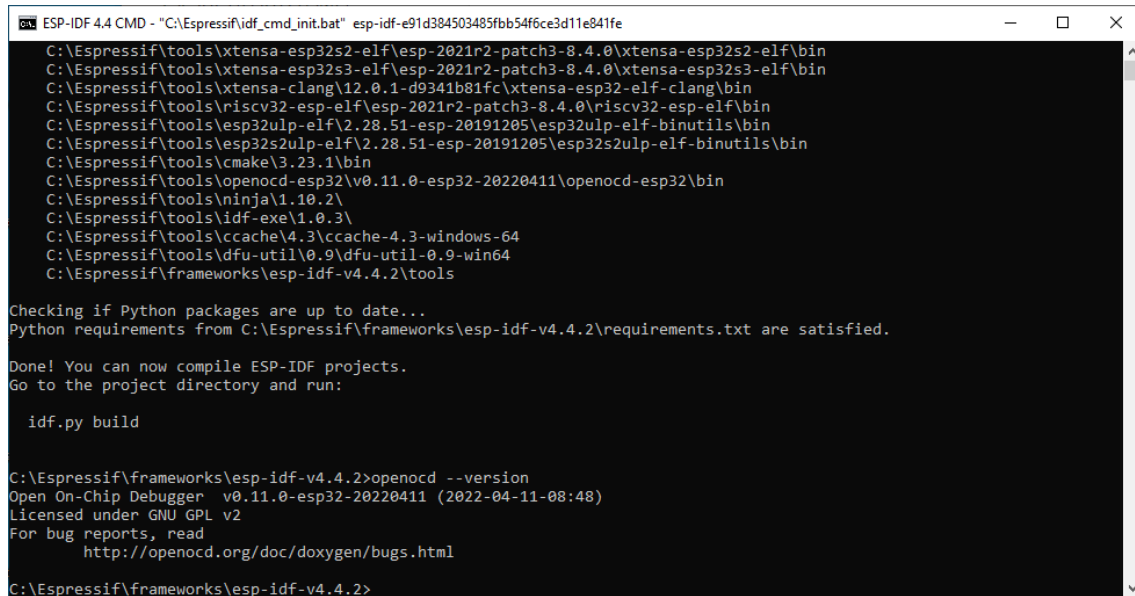
das USB Kabel müssen Sie auf dem Fendt NICHT anschließen, der Atom wird von den 12V versorgt.

so:



15. JTAG-S3-USB

<https://docs.espressif.com/projects/esp-idf/en/latest/esp32s3/api-guides/jtag-debugging/index.html>
openocd --version



```
ESP-IDF 4.4 CMD - "C:\Espressif\idf_cmd_init.bat" esp-idf-e91d384503485fbb54f6ce3d11e841fe
C:\Espressif\tools\xtensa-esp32s2-elf\esp-2021r2-patch3-8.4.0\xtensa-esp32s2-elf\bin
C:\Espressif\tools\xtensa-esp32s3-elf\esp-2021r2-patch3-8.4.0\xtensa-esp32s3-elf\bin
C:\Espressif\tools\xtensa-clang\12.0.1-d9341b81fc\xtensa-esp32-elf-clang\bin
C:\Espressif\tools\riscv32-esp-elf\esp-2021r2-patch3-8.4.0\riscv32-esp-elf\bin
C:\Espressif\tools\esp32ulp-elf\2.28.51-esp-20191205\esp32ulp-elf-binutils\bin
C:\Espressif\tools\esp32s2ulp-elf\2.28.51-esp-20191205\esp32s2ulp-elf-binutils\bin
C:\Espressif\tools\cmake\3.23.1\bin
C:\Espressif\tools\openocd-esp32\v0.11.0-esp32-20220411\openocd-esp32\bin
C:\Espressif\tools\ninja\1.10.2\
C:\Espressif\tools\idf.exe\1.0.3\
C:\Espressif\tools\ccache\4.3\ccache-4.3-windows-64
C:\Espressif\tools\dfu-util\0.9\dfu-util-0.9-win64
C:\Espressif\frameworks\esp-idf-v4.4.2\tools

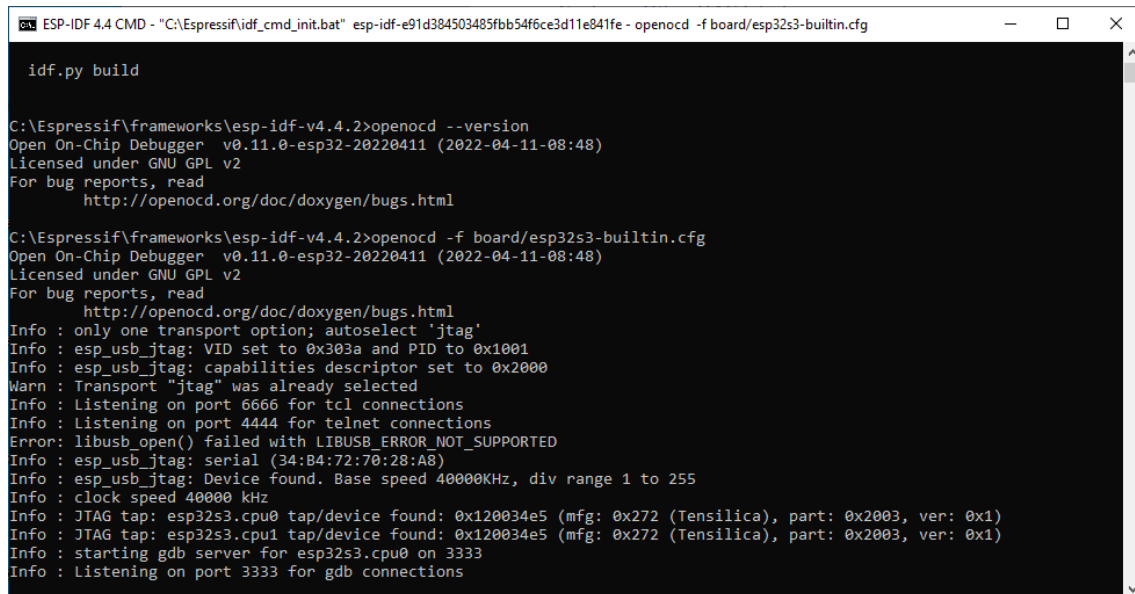
Checking if Python packages are up to date...
Python requirements from C:\Espressif\frameworks\esp-idf-v4.4.2\requirements.txt are satisfied.

Done! You can now compile ESP-IDF projects.
Go to the project directory and run:

idf.py build

C:\Espressif\frameworks\esp-idf-v4.4.2>openocd --version
Open On-Chip Debugger v0.11.0-esp32-20220411 (2022-04-11-08:48)
Licensed under GNU GPL v2
For bug reports, read
http://openocd.org/doc/doxygen/bugs.html
C:\Espressif\frameworks\esp-idf-v4.4.2>
```

openocd -f board/esp32s3-builtin.cfg -> scheint richtig
openocd -f board/esp32s3-bridge.cfg -> scheint falsch

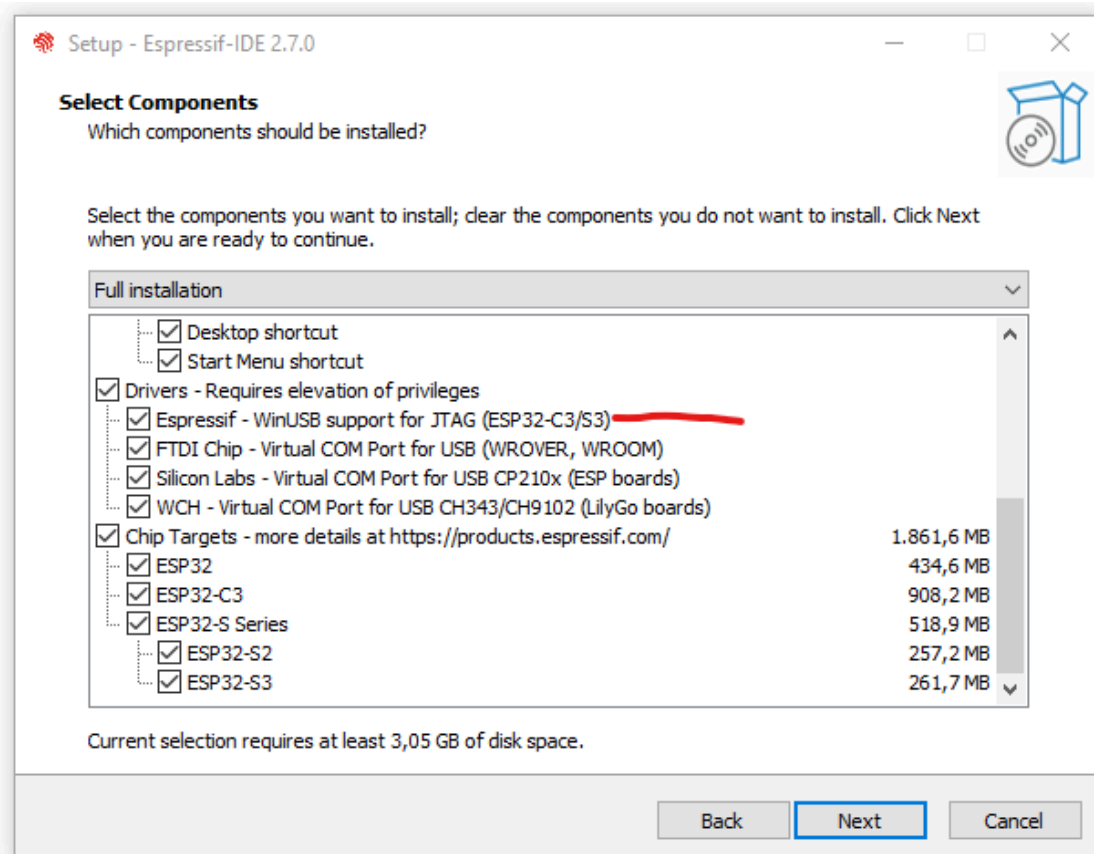


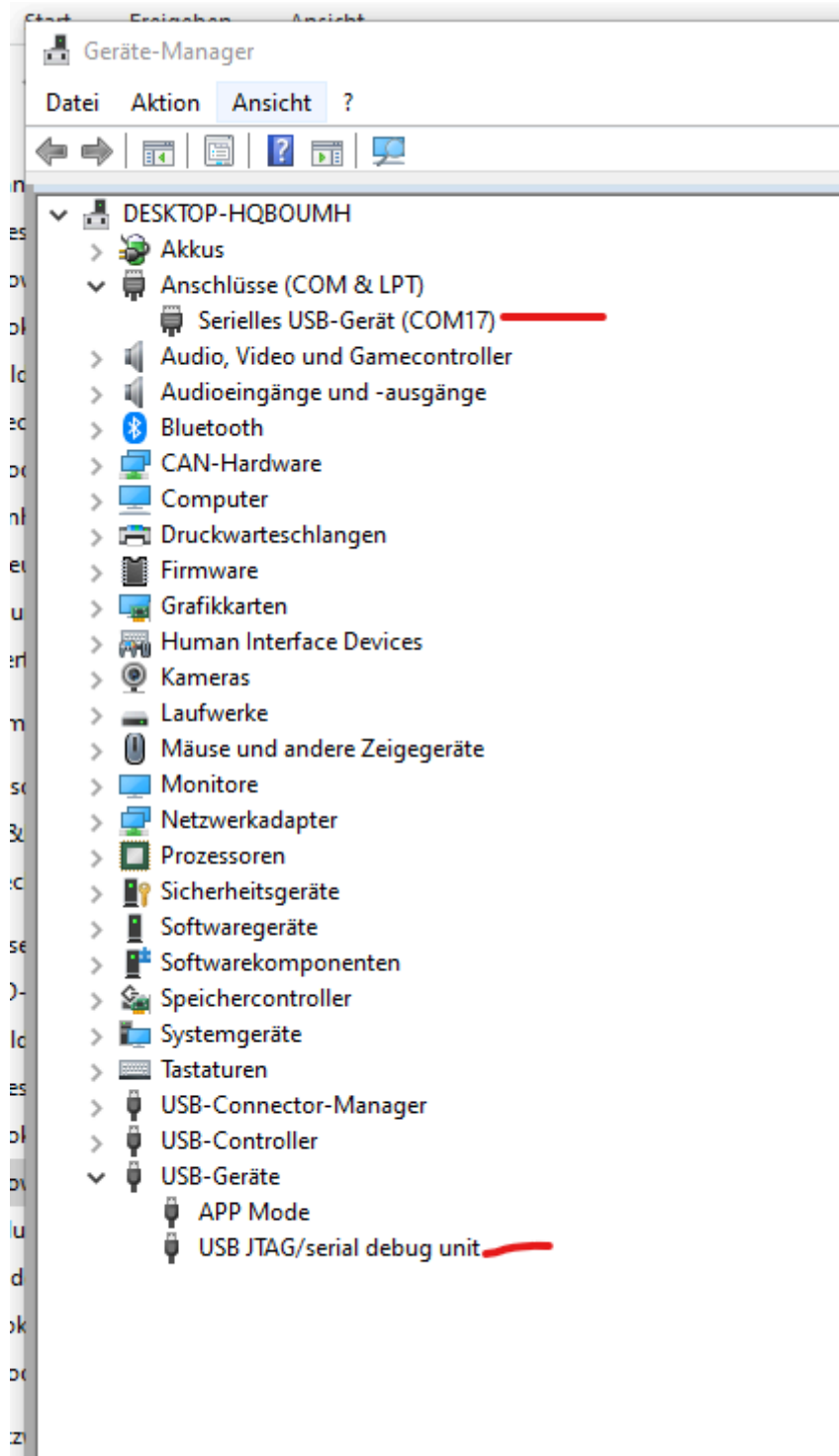
```
ESP-IDF 4.4 CMD - "C:\Espressif\idf_cmd_init.bat" esp-idf-e91d384503485fbb54f6ce3d11e841fe - openocd -f board/esp32s3-builtin.cfg

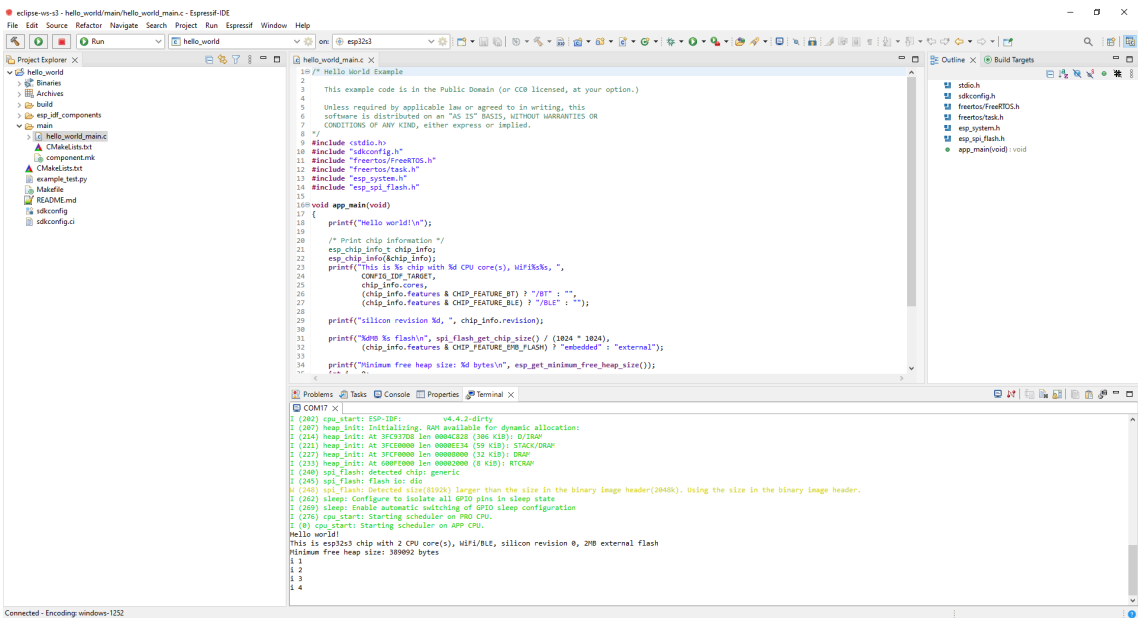
idf.py build

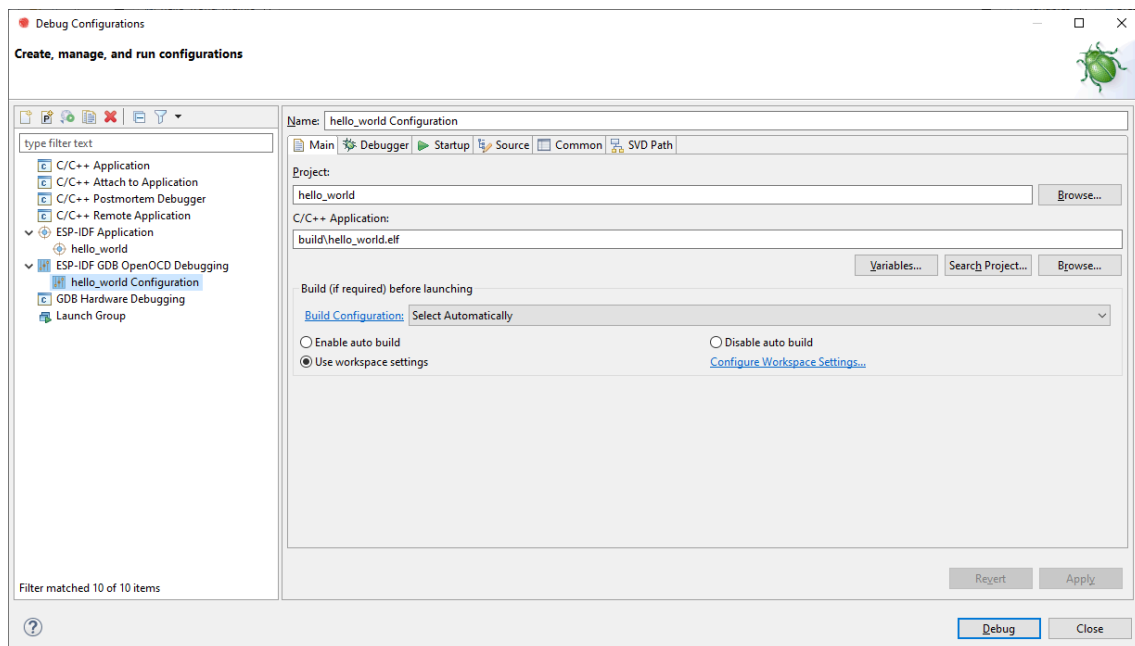
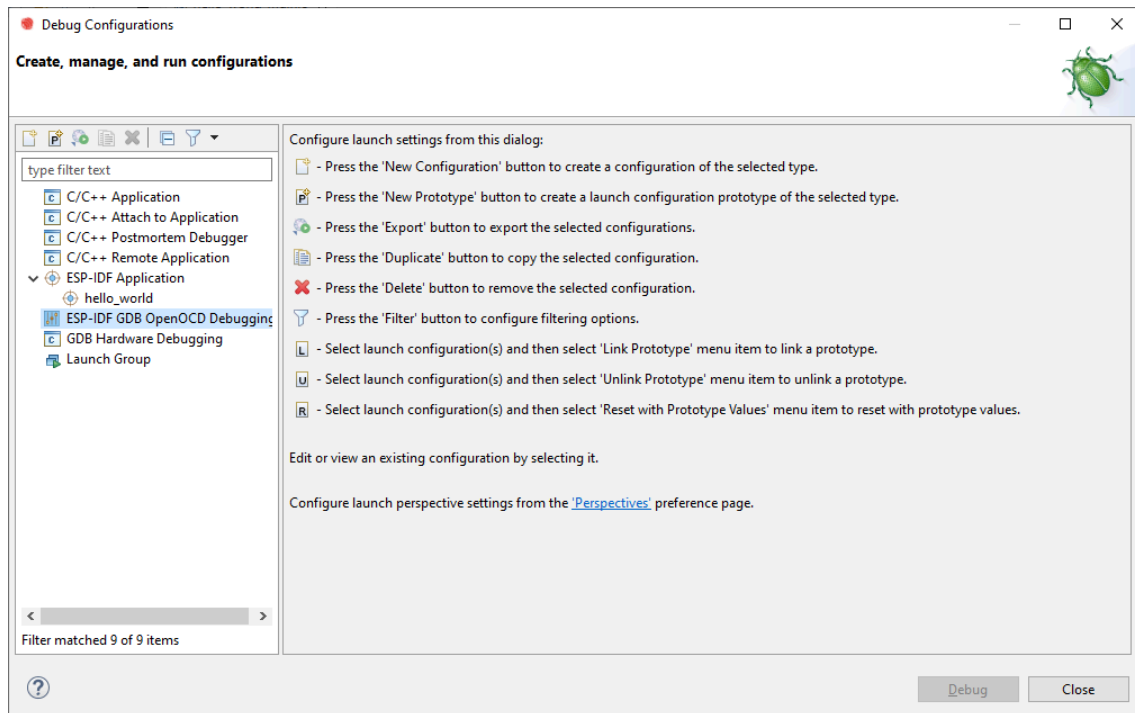
C:\Espressif\frameworks\esp-idf-v4.4.2>openocd --version
Open On-Chip Debugger v0.11.0-esp32-20220411 (2022-04-11-08:48)
Licensed under GNU GPL v2
For bug reports, read
http://openocd.org/doc/doxygen/bugs.html

C:\Espressif\frameworks\esp-idf-v4.4.2>openocd -f board/esp32s3-builtin.cfg
Open On-Chip Debugger v0.11.0-esp32-20220411 (2022-04-11-08:48)
Licensed under GNU GPL v2
For bug reports, read
http://openocd.org/doc/doxygen/bugs.html
Info : only one transport option; autoselect 'jtag'
Info : esp_usb_jtag: VID set to 0x303a and PID to 0x1001
Info : esp_usb_jtag: capabilities descriptor set to 0x2000
Warn : Transport "jtag" was already selected
Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections
Error: libusb_open() failed with LIBUSB_ERROR_NOT_SUPPORTED
Info : esp_usb_jtag: serial (34:B4:72:70:28:A8)
Info : esp_usb_jtag: Device found. Base speed 40000KHz, div range 1 to 255
Info : clock speed 40000 kHz
Info : JTAG tap: esp32s3.cpu0 tap/device found: 0x120034e5 (mfg: 0x272 (Tensilica), part: 0x2003, ver: 0x1)
Info : JTAG tap: esp32s3.cpu1 tap/device found: 0x120034e5 (mfg: 0x272 (Tensilica), part: 0x2003, ver: 0x1)
Info : starting gdb server for esp32s3.cpu0 on 3333
Info : Listening on port 3333 for gdb connections
```

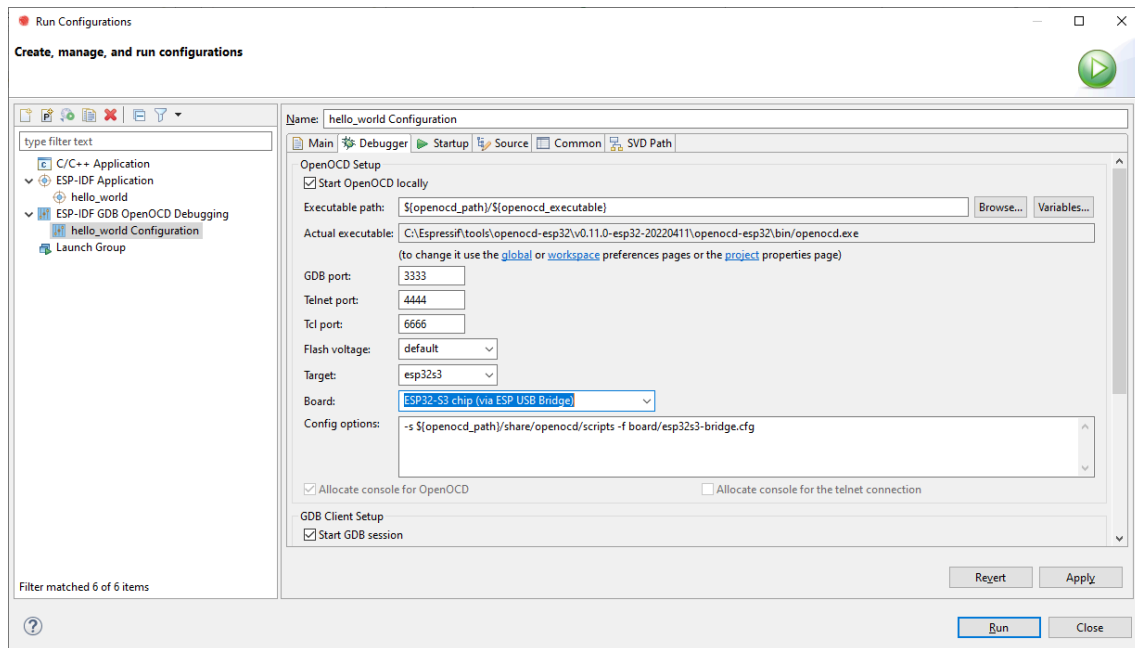





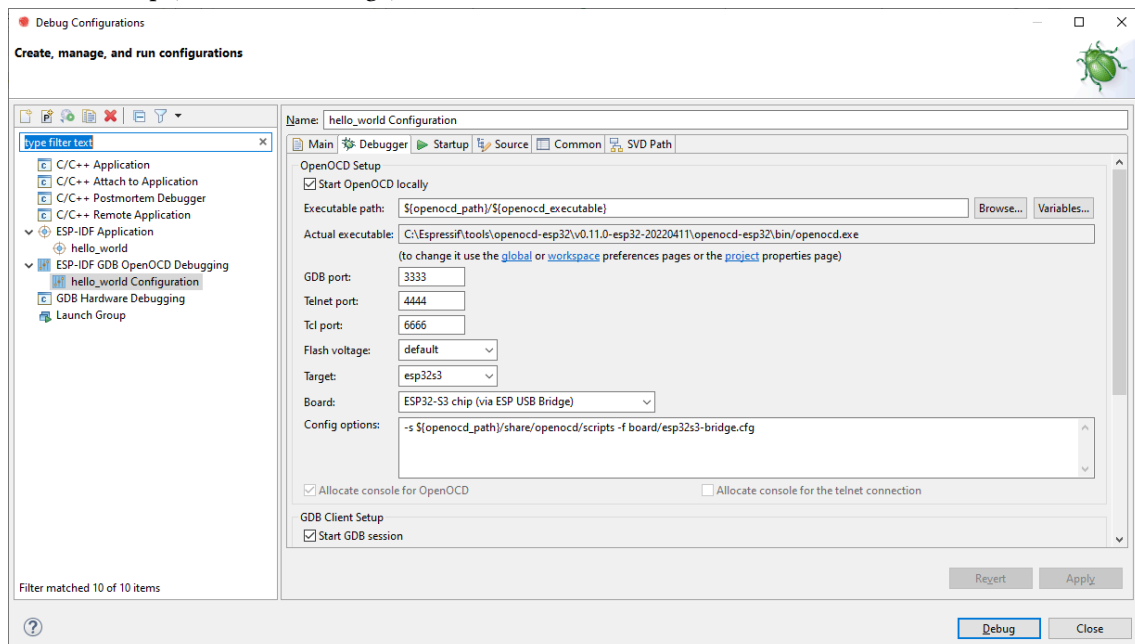




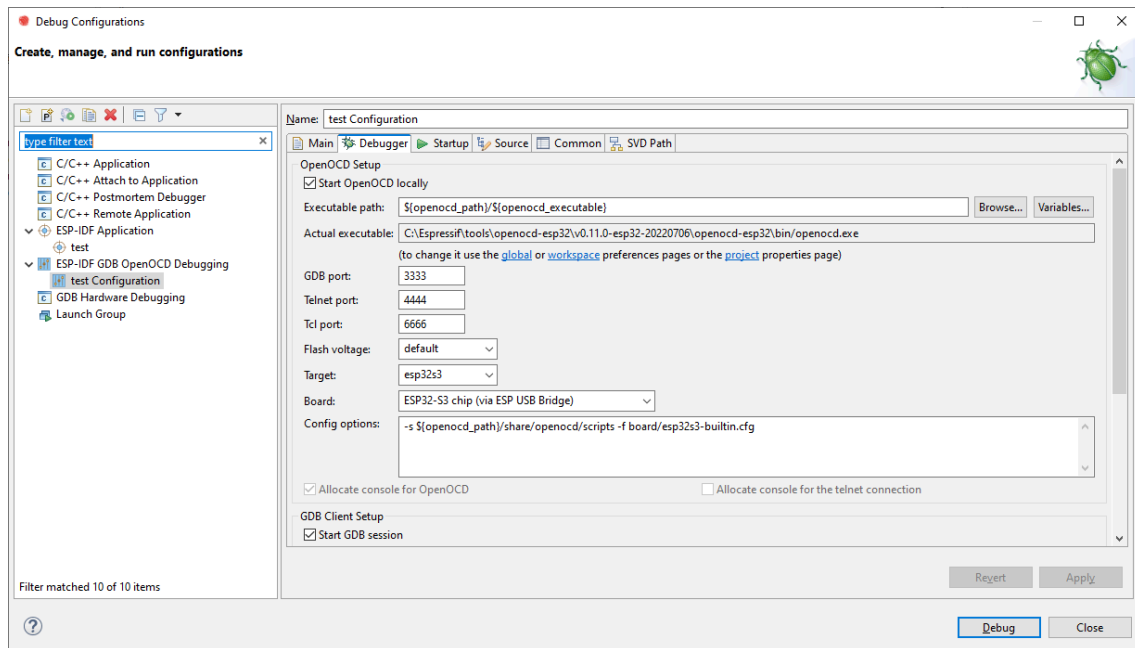
dann läuft es nicht, es muss folgende Einstellung geändert werden:



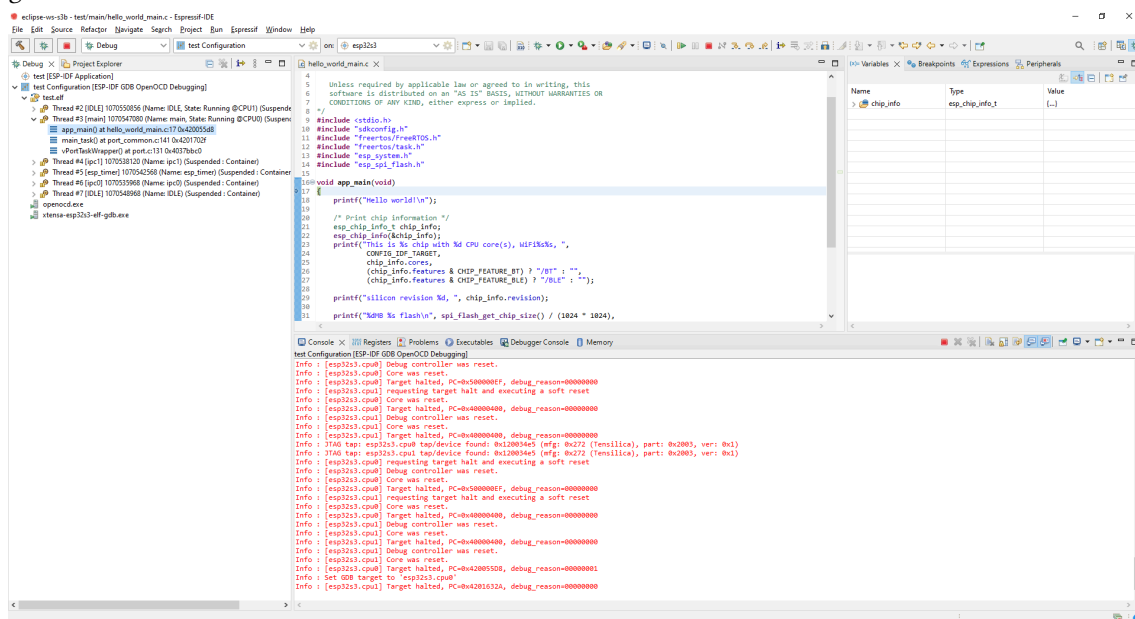
ESP32-S3 chip (via ESP USB Bridge)



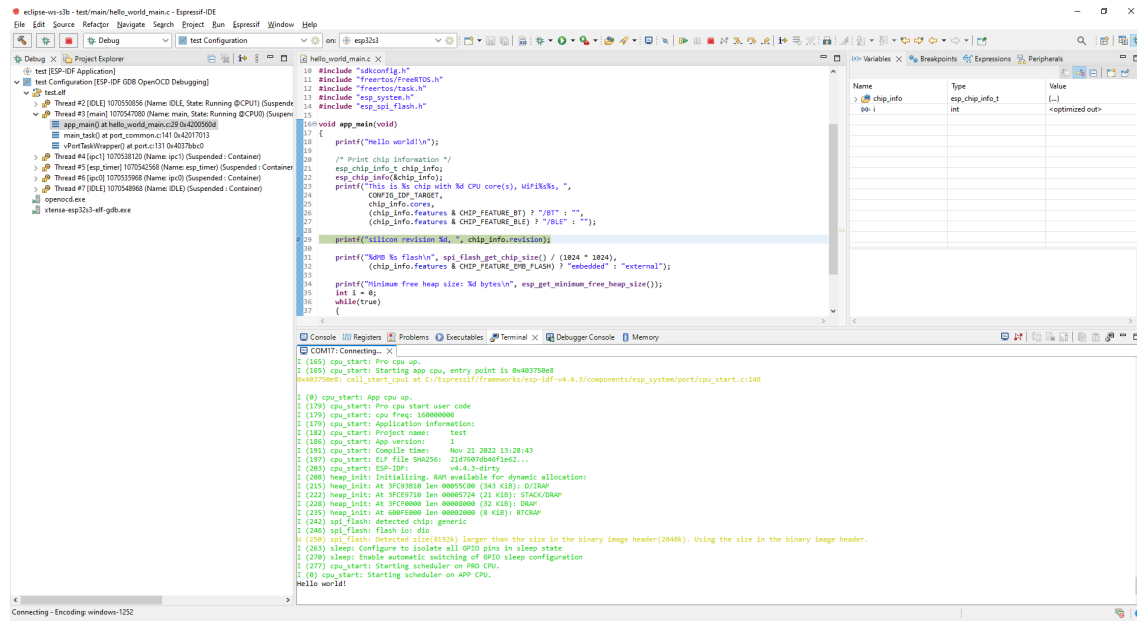
und dann noch „bridge“ durch „builtin“ ändern:



geht:



Auch hier kann die Console parallel zum Debugging laufen,
über ein USB Kabel,
cool:



15.0.1 Passende Themen-Unterseiten auf ms-muc-docs.de

-  ESP32 & ESP32-S3 DevKit auf ms-muc-docs.de

16. JTAG

16.1 JTAG Schritt_01:

jetzt richten wir JTAG ein.

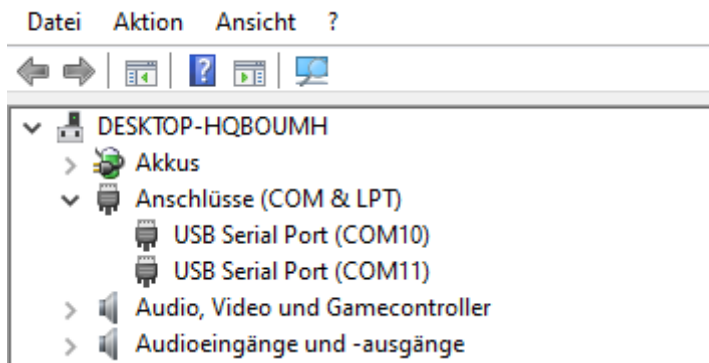
<https://docs.espressif.com/projects/esp-idf/en/v4.4.3/esp32/api-guides/jtag-debugging/configure-ft2232h-jtag.html>

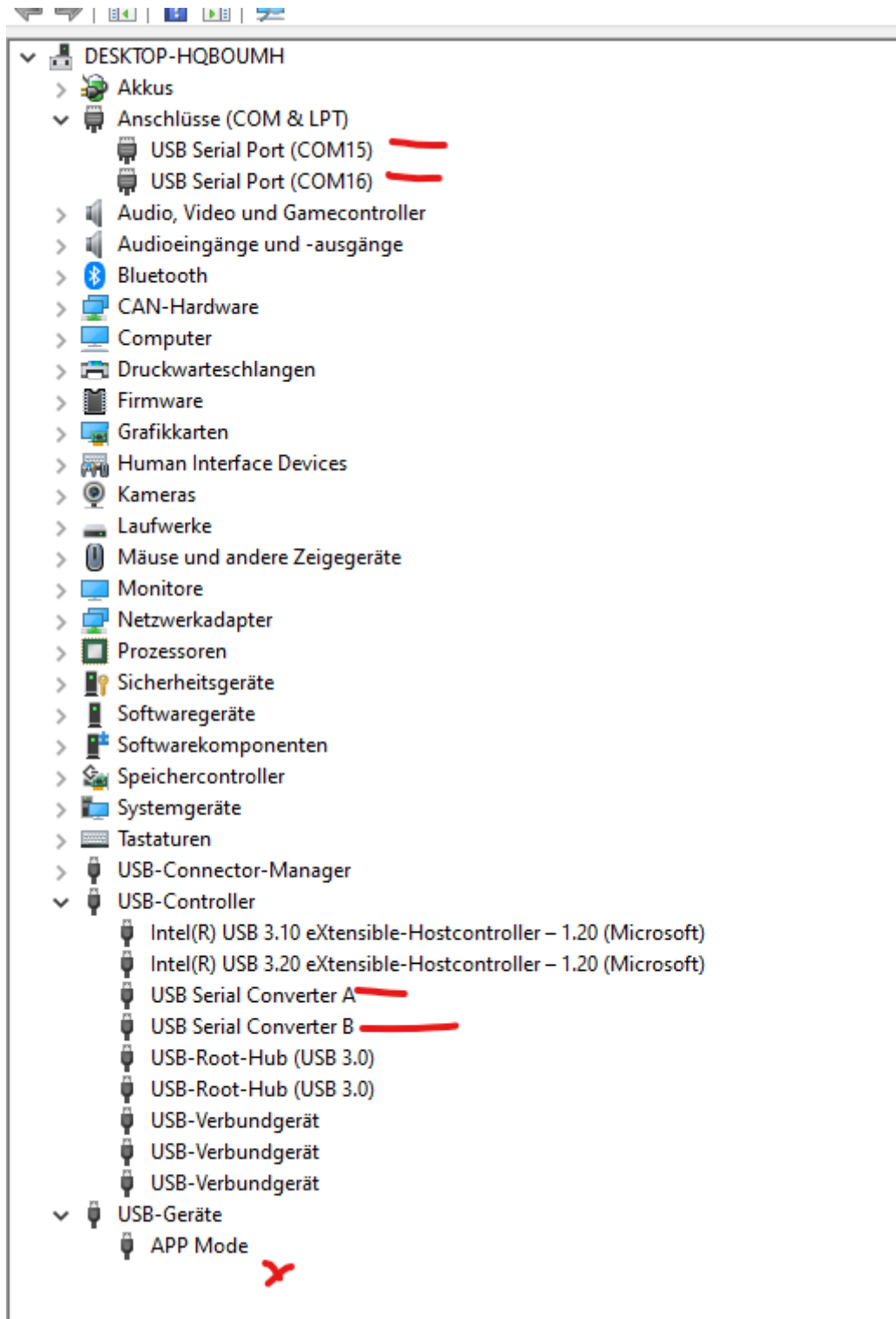
https://docs.espressif.com/projects/espressif-esp-iot-solution/en/latest/hw-reference/ESP-Prog_guide.html

<https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-guides/jtag-debugging/index.html>

16.1.1 USB - Schnittstelle mit ZADIG umstellen:

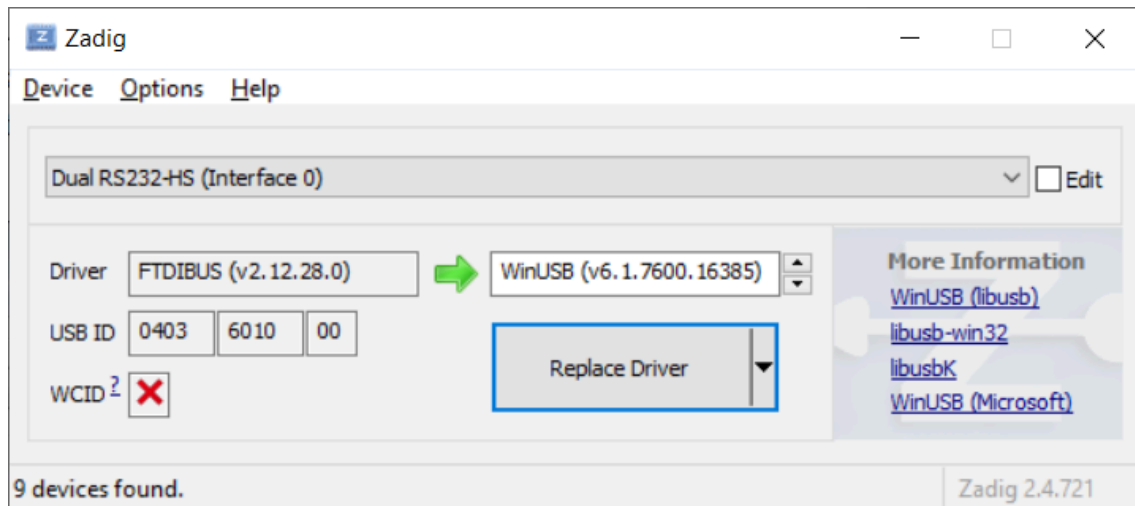
16.1.1.1 vorher:



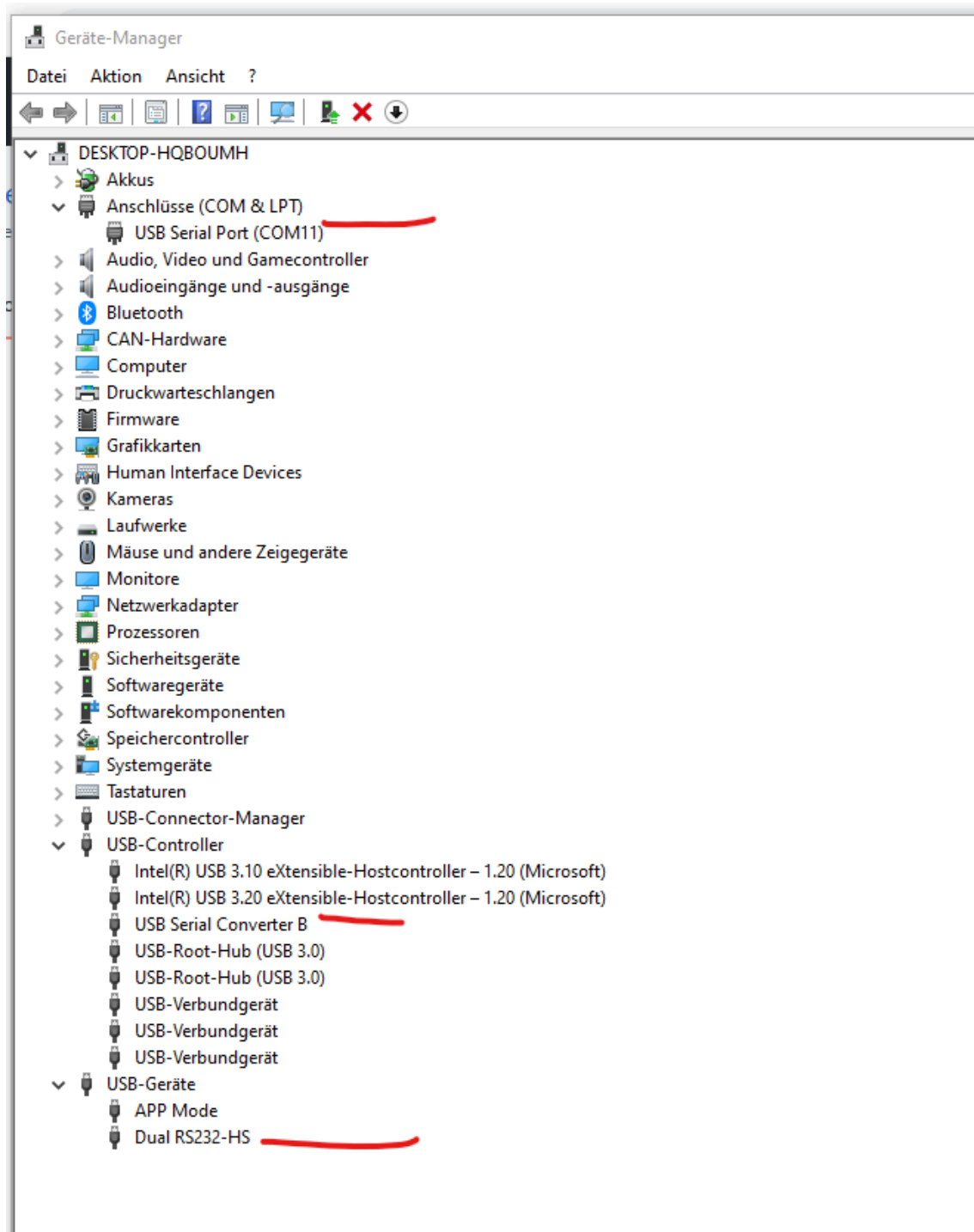


16.1.1.2 Aufruf von ZADIG:

siehe auch: <https://docs.espressif.com/projects/esp-idf/en/v4.4.3/esp32/api-guides/jtag-debugging/configure-ft2232h-jtag.html#windows>

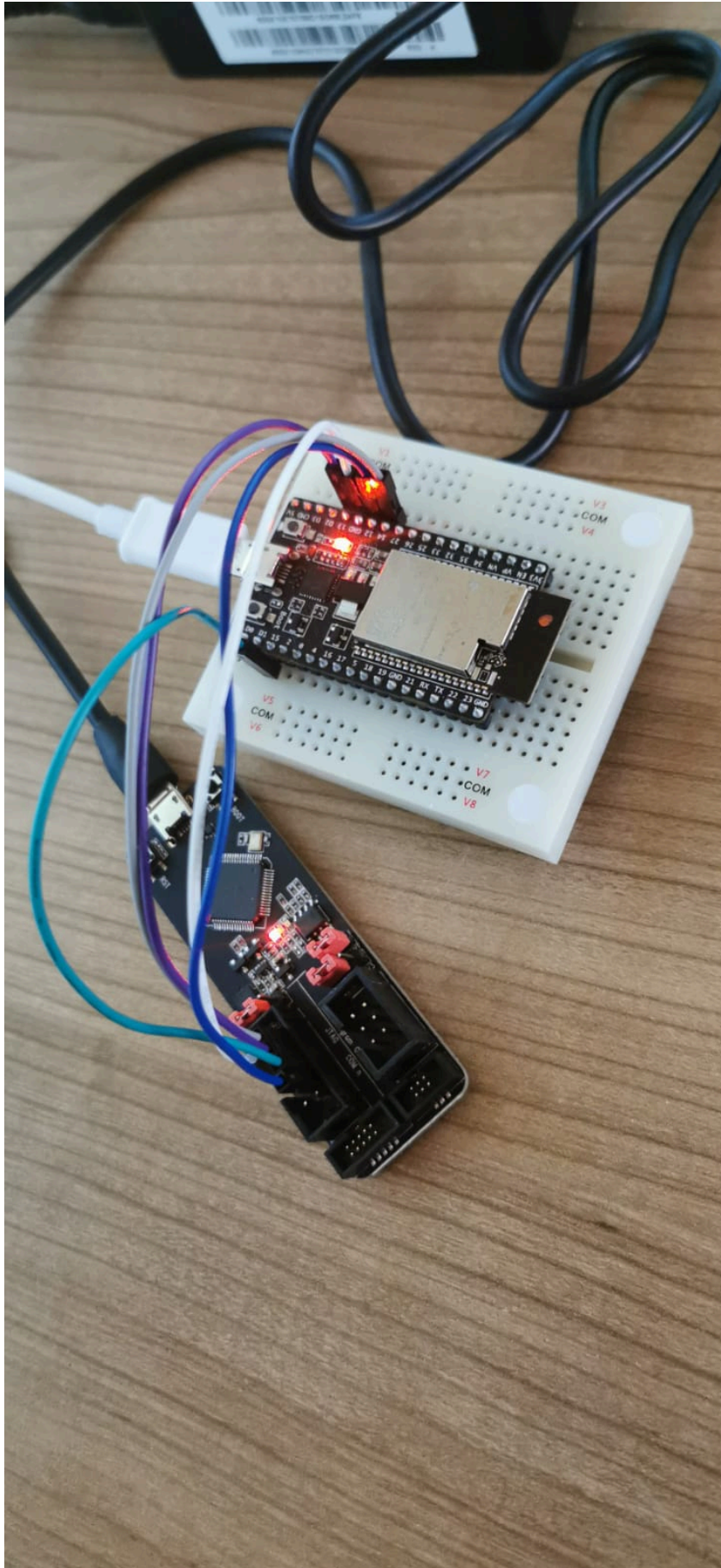


16.1.1.3 nachher:



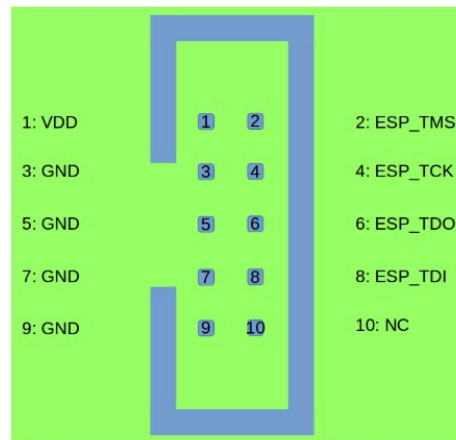
Hinweis: der Vorgang muss leider wiederholt werden wenn man den ESP-PROG an eine andere Schnittstelle anschließt.

daher idealerweise immer dieselben USB Steckplätze verwenden.

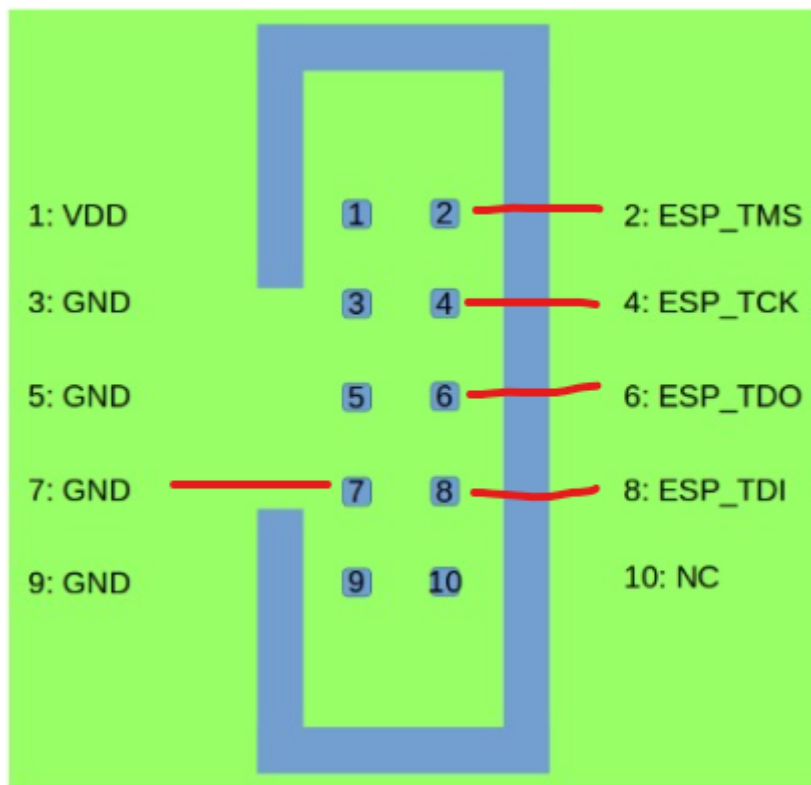


- JTAG Interface

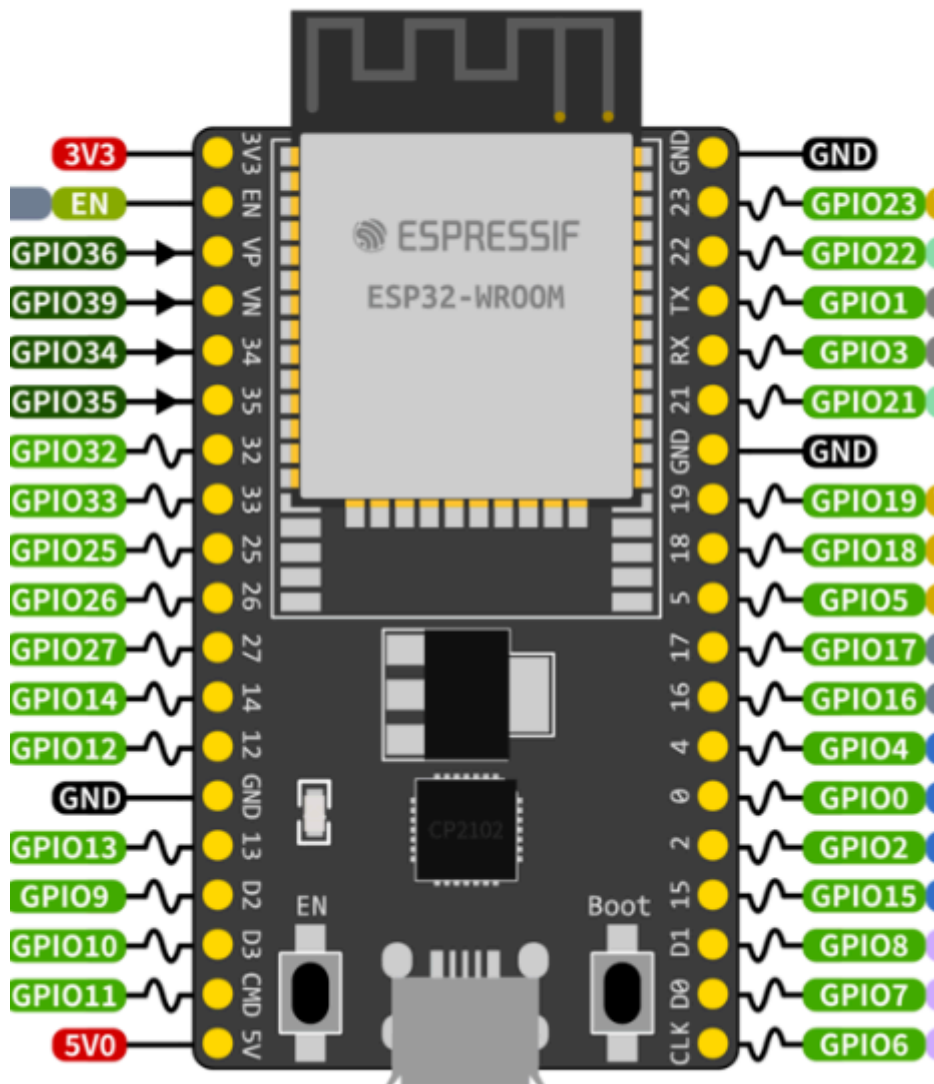
The design for the JTAG interface on the user board should follow the reference provided in the figure below.

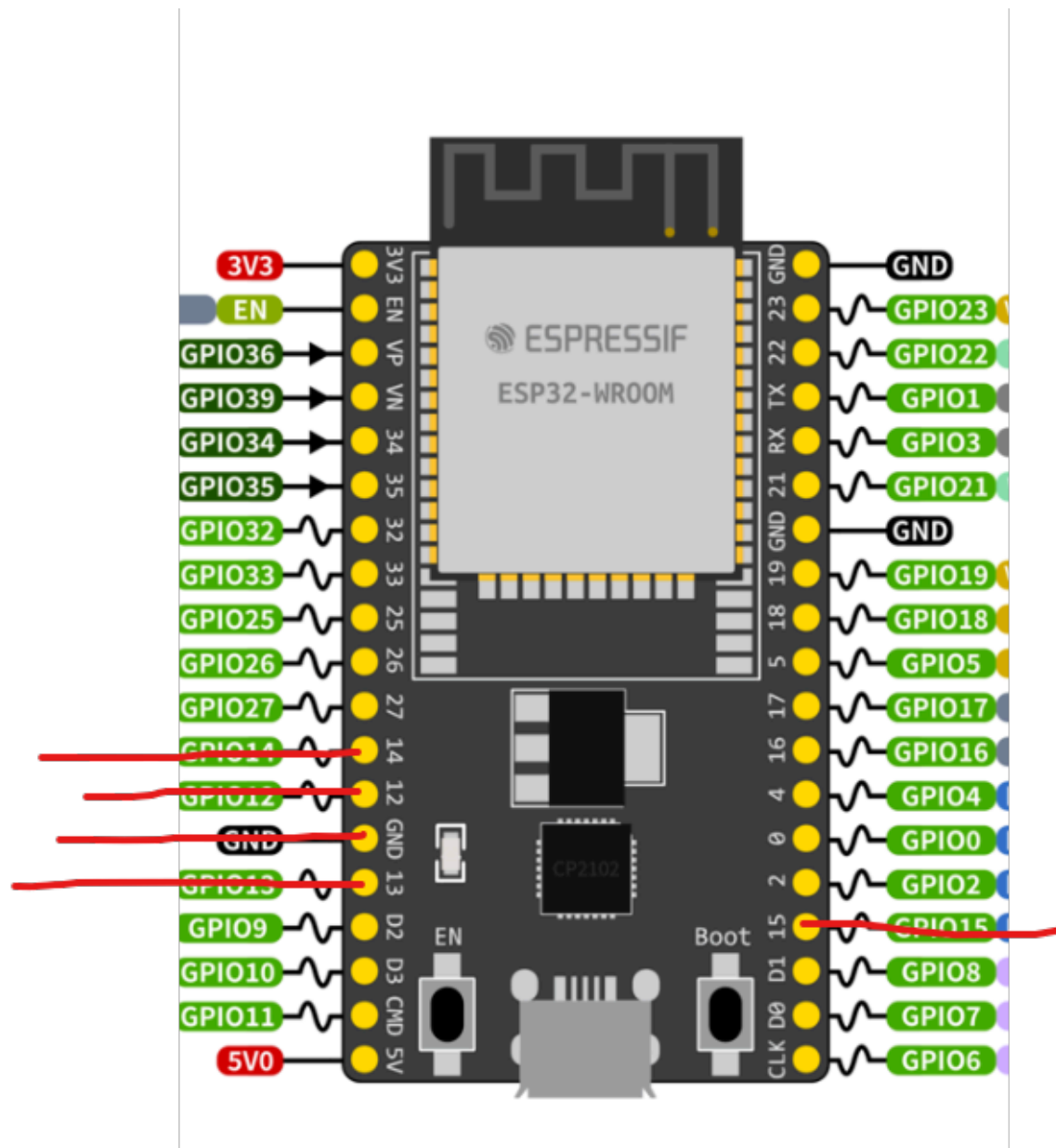


Diese Pins werden mindestens gebraucht:



	ESP32 Pin	JTAG Signal
1	CHIP_PU	TRST_N
2	MTDO / GPIO15	TDO
3	MTDI / GPIO12	TDI
4	MTCK / GPIO13	TCK
5	MTMS / GPIO14	TMS
6	GND	GND





<https://docs.espressif.com/projects/esp-idf/en/latest/api-guides/jtag-debugging/configure-other-jtag.html>

16.2 JTAG Schritt_02:

`openocd -version`

```

ESP-IDF Command Prompt (cmd.exe) - "C:\Users\hoepffr\espessif\idf_cmd_init.bat" "C:\Users\hoepffr\AppData\Local\Programs\Py...
Setting ID_PATH: C:\Users\hoepffr\Desktop\esp-idf

Adding ESP-IDF tools to PATH...
C:\Users\hoepffr\espessif\tools\xtensa-esp32-elf\esp-2019r2-8.2.0\xtensa-esp32-elf\bin
C:\Users\hoepffr\espessif\tools\esp32ulp-elf\2.28.51.20170517\esp32ulp-elf-binutils\bin
C:\Users\hoepffr\espessif\tools\cmake\3.13.4\bin
C:\Users\hoepffr\espessif\tools\openocd-esp32\v0.10.0-esp32-20190313\openocd-esp32\bin
C:\Users\hoepffr\espessif\tools\mconf\v4.6.0.0-idf-20190628\
C:\Users\hoepffr\espessif\tools\ninja\1.9.0\
C:\Users\hoepffr\espessif\tools\idf-exe\1.0.1\
C:\Users\hoepffr\espessif\tools\ccache\3.7\
C:\Users\hoepffr\espessif\python_env\idf4.0_py3.7_env\Scripts
C:\Users\hoepffr\Desktop\esp-idf\tools

Checking if Python packages are up to date...
Python requirements from C:\Users\hoepffr\Desktop\esp-idf\requirements.txt are satisfied.

Done! You can now compile ESP-IDF projects.
Go to the project directory and run:

idf.py build

C:\Users\hoepffr\Desktop\esp-idf>openocd --version
Open On-Chip Debugger v0.10.0-esp32-20190313 (2019-03-13-09:57)
Licensed under GNU GPL v2
For bug reports, read
    http://openocd.org/doc/doxygen/bugs.html
C:\Users\hoepffr\Desktop\esp-idf>

```

16.3 JTAG Schritt_03:

<https://docs.espressif.com/projects/esp-idf/en/v4.4.3/esp32/api-guides/jtag-debugging/index.html#run-openocd>

`openocd -f board/esp32-wrover-kit-3.3v.cfg`

```

ESP-IDF 4.4 CMD - "C:\Espressif\idf_cmd_init.bat" esp-idf-e91d384503485fbb54f6ce3d11e841fe - openocd -f board/esp32-wrover-kit-3.3v.cfg
C:\Espressif\frameworks\esp-idf-v4.4.2>openocd -f board/esp32-wrover-kit-3.3v.cfg
Open On-Chip Debugger v0.11.0-esp32-20220411 (2022-04-11-08:48)
Licensed under GNU GPL v2
For bug reports, read
    http://openocd.org/doc/doxygen/bugs.html
Info : Listening on port 6666 for tcl connections
Info : Listening on port 4444 for telnet connections
Info : ftdi: if you experience problems at higher adapter clocks, try the command "ftdi tdo_sample_edge falling"
Info : clock speed 20000 kHz
Info : JTAG tap: esp32.cpu0 tap/device found: 0x120034e5 (mfg: 0x272 (Tensilica), part: 0x2003, ver: 0x1)
Info : JTAG tap: esp32.cpu1 tap/device found: 0x120034e5 (mfg: 0x272 (Tensilica), part: 0x2003, ver: 0x1)
Warn : target esp32.cpu1 examination failed
Info : starting gdb server for esp32.cpu0 on 3333
Info : Listening on port 3333 for gdb connections

```

ACHTUNG !!! das starten des GDB Servers von der Kommandozeile ist nur noch erforderlich, wenn man auch an der Kommandozeile debuggt.

→ aus der ESP-IDF heraus wird openocd automatisch gestartet.

16.4 JTAG Schritt_04:

an dieser Stelle ist die Doku hoffnungslos veraltet.

wenn man die Espressif-IDE (Espressif-IDE) verwendet, dann stimmen diese Links nicht mehr.

<https://docs.espressif.com/projects/esp-idf/en/latest/api-guides/jtag-debugging/debugging-examples.html#jtag-debugging-examples-eclipse>

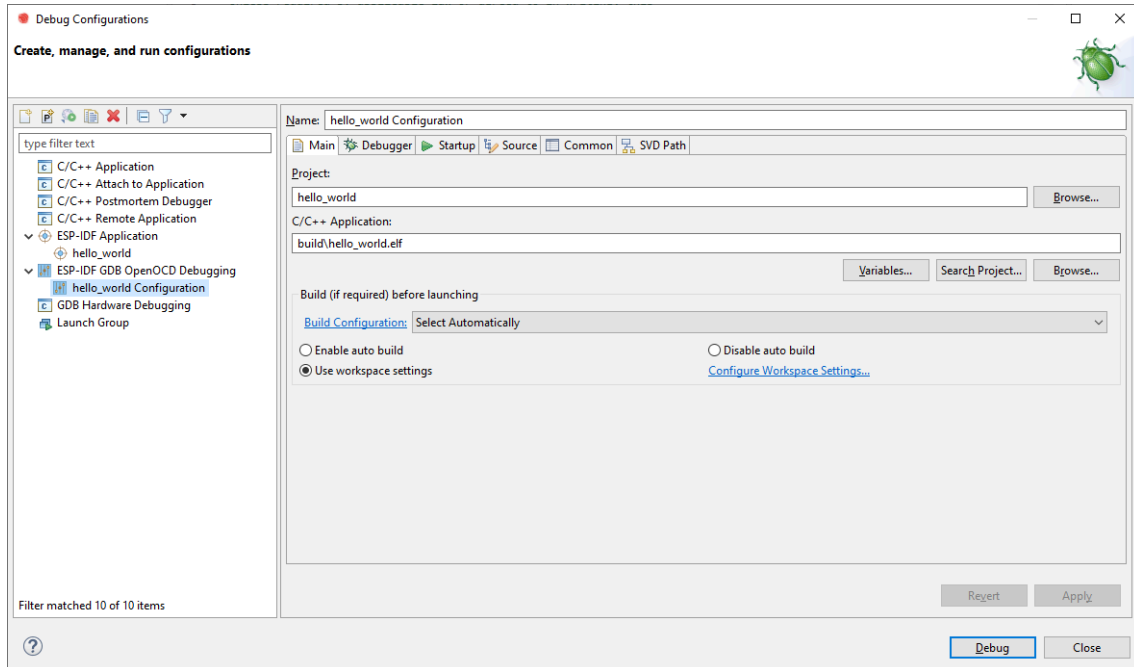
<https://docs.espressif.com/projects/esp-idf/en/latest/api-guides/jtag-debugging/using-debugger.html#jtag-debugging-using-debugger-eclipse>

Ich habe wie folgt sofort eine Debugging zustande gebracht:

1. nur ESP-IDF GDB... wählen, dann links oben auf das



Symbol klicken,
und starten.
keinerlei Einstellungen zu tätigen.



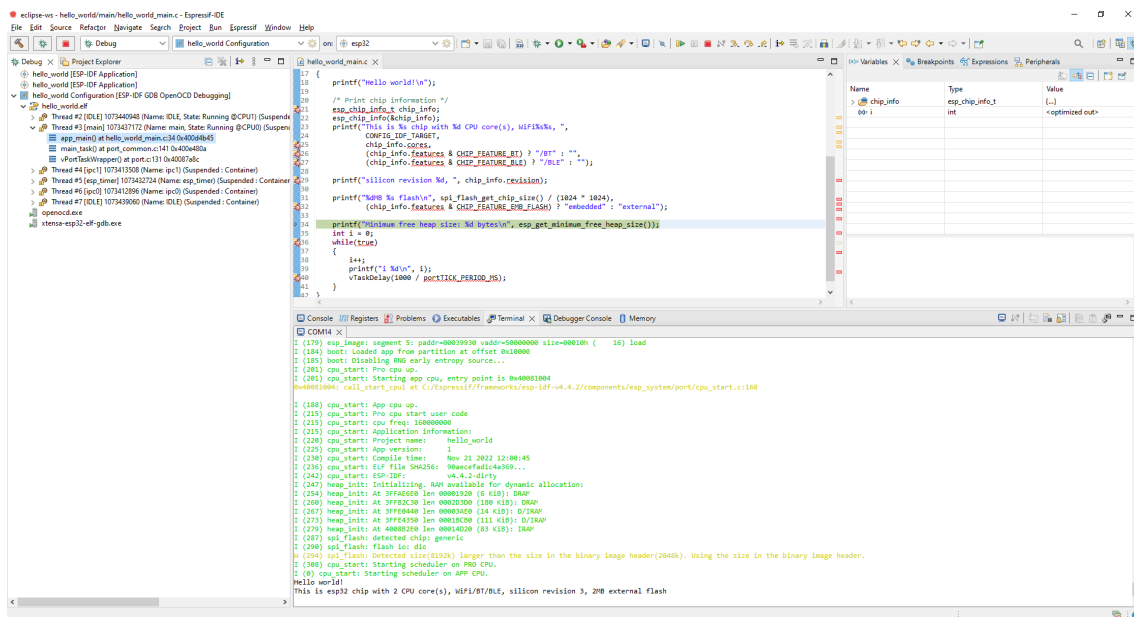
16.5 JTAG Schritt_05:

entfällt

16.6 JTAG Schritt_06:

entfällt

16.7 JTAG Schritt_07:



und ja, man kann die Console parallel zur Debugging Session laufen lassen, aber die Console muss vor dem Start der Debugging Session offen sein.

17. Joystick

Versuchen sie nun die eben erstellte ISOBUS Maschine mit dem AUX-Joystick zu steuern.
Probieren Sie dies sowohl mit dem Traktor-Monitor, als auch mit dem Zusatzbildschirm.
Notieren Sie ihre Erkenntnis, wir werden diese später im Unterricht besprechen.

18. Kalkulation Hardware

Unsere ECU basiert auf dem ATOM von M5 Stack.

Schaltplan:

https://github.com/Meisterschulen-am-Ostbahnhof-Muenchen/ISOBUS_Hardware

Stückliste ebendort.

Der Schaltplan ist in EAGLE gemacht, das bekommen Sie so.

Als Student oder Lehrer haben Sie Anspruch auf ein Äquivalent der Premium-Version von Autodesk EAGLE für den nicht kommerziellen Unterricht.

Aktivieren Sie eine kostenlose, individuelle Bildungslizenz über die Education Community unter TODO: Link

hier die Kalkulation:

1. Kalkulation ohne Nebenkosten:

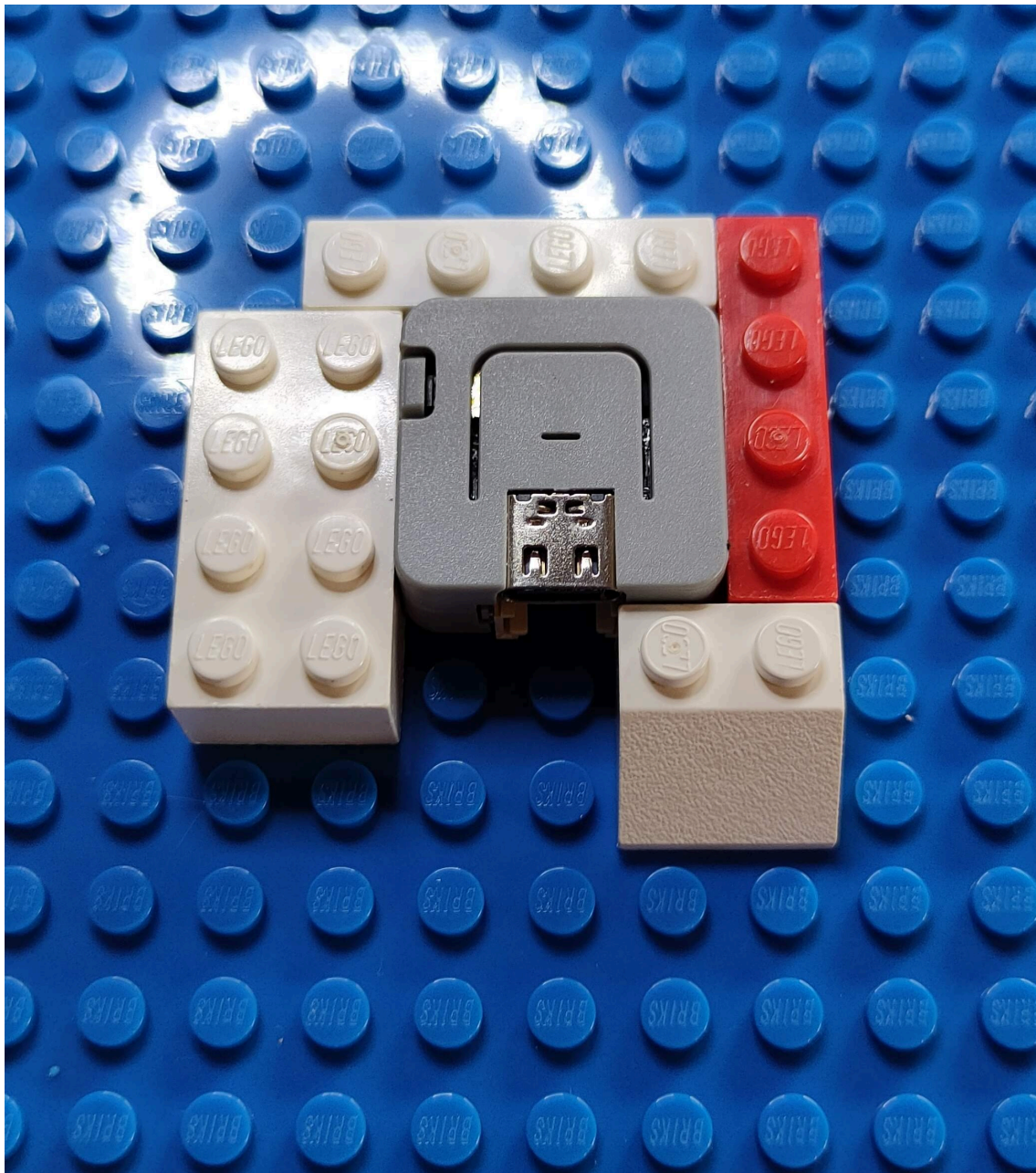
Nr.	Bezeichnung	Menge		Einzelpreis	Gesamtpreis	Link
	USB Kabel auf Type-C USB2.0 schwarz	1		0,99€	0,99€	
C008	ATOM Lite ESP32	1	\$5.95	5,10 €	5,10 €	atom-lite-esp32
A077	ATOMIC Proto Kit	1	\$3.50	3,00 €	3,00 €	atomic-proto-kit
R-78E5.0-0.5	DC-DC-WANDLER 5V 0,5A	1		2,29 €	2,29 €	919-R-78E5.0-0.5
TMK325ABJ476MM-T	1210 25VDC 47uF 20% X5R	1		1,05 €	1,05 €	963-TMK325ABJ476MM-T
TLE9250VSJ	CAN-IC-Schnittstelle	1		0,889 €	0,889 €	726-TLE9250VSJXUMA1
SS12L R3	1A 20V Schottky	1		0,39 €	0,39 €	821-SS12LR3
824501161	400W Bidir. 16V TVS	1		0,214 €	0,214 €	824501161
	Platine	1		0,2235€	0,2235€	Atomic_V1.0_2020-10-15.zip
	60.4R	2		0,005 €	0,010 €	CRCW040260R4FKED
	47 nF	1		0,008 €	0,016 €	TMF105B7473KVHF
	100 nF	2		0,006 €	0,012 €	GCM155R71C104KA55D
824098024 oder PESD2CAN	WE-TVS_SOT23-3L	1		0,076 €	0,076 €	PESD2CAN215

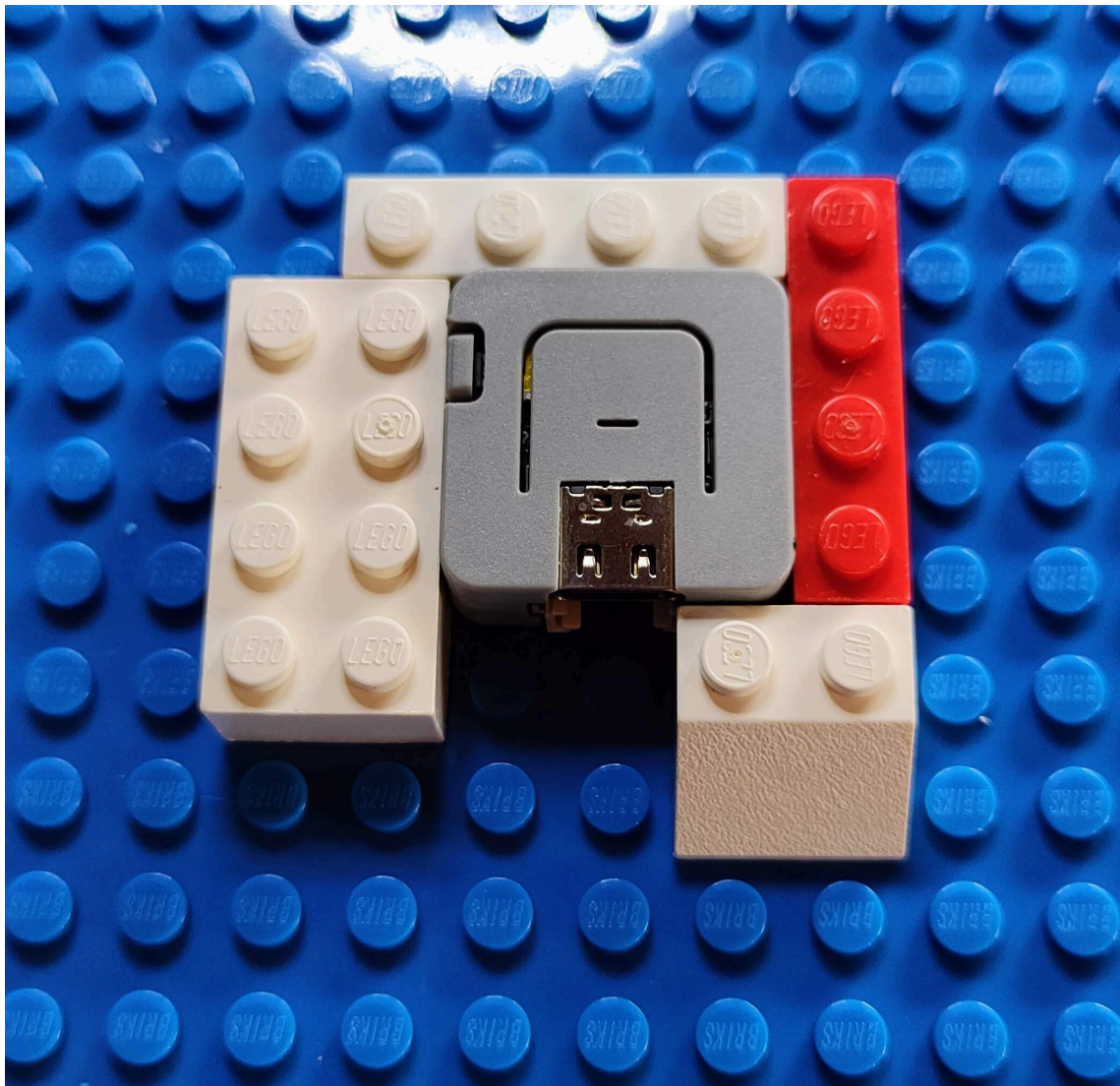
18.0.1 Passende Themen-Unterseiten auf ms-muc-docs.de

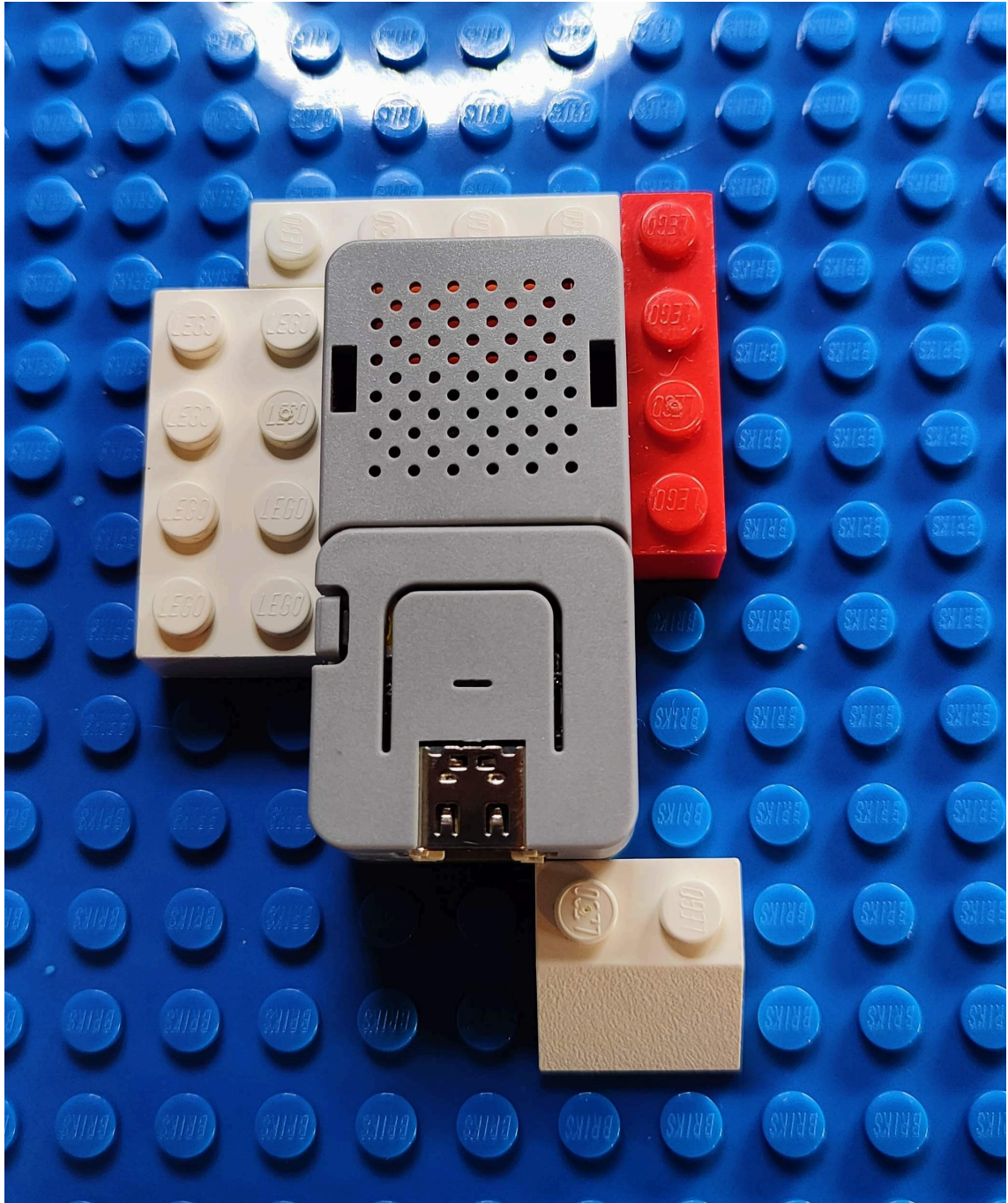
-  [ESP32 & ESP32-S3 DevKit auf \[ms-muc-docs.de\]\(https://ms-muc-docs.de\)](#)

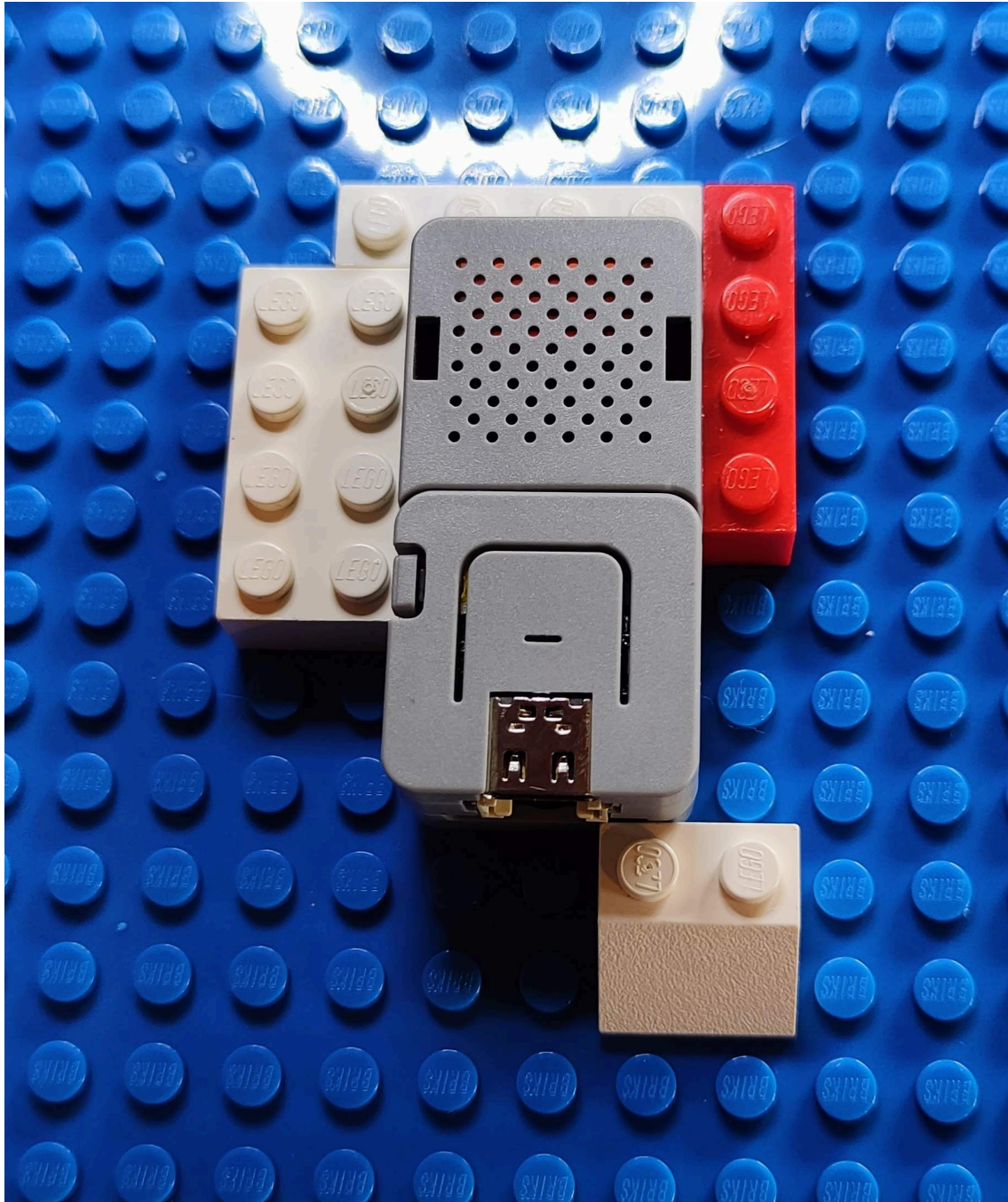
19. Lego

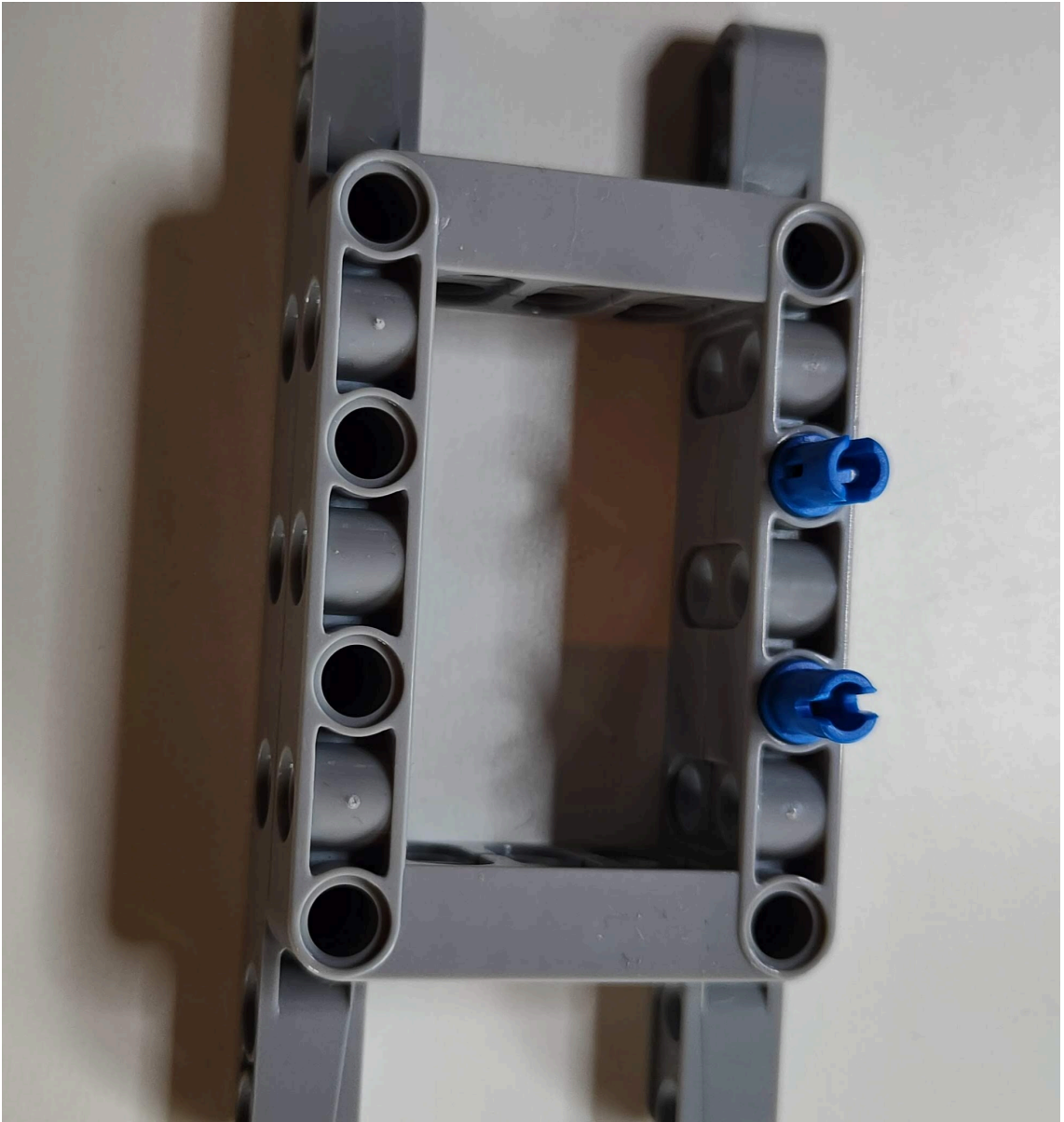
<<https://www.1000steine.de/de/gemeinschaft/forum/?entry=1&id=442294>
<<https://youtu.be/z4hOIl2aygs>

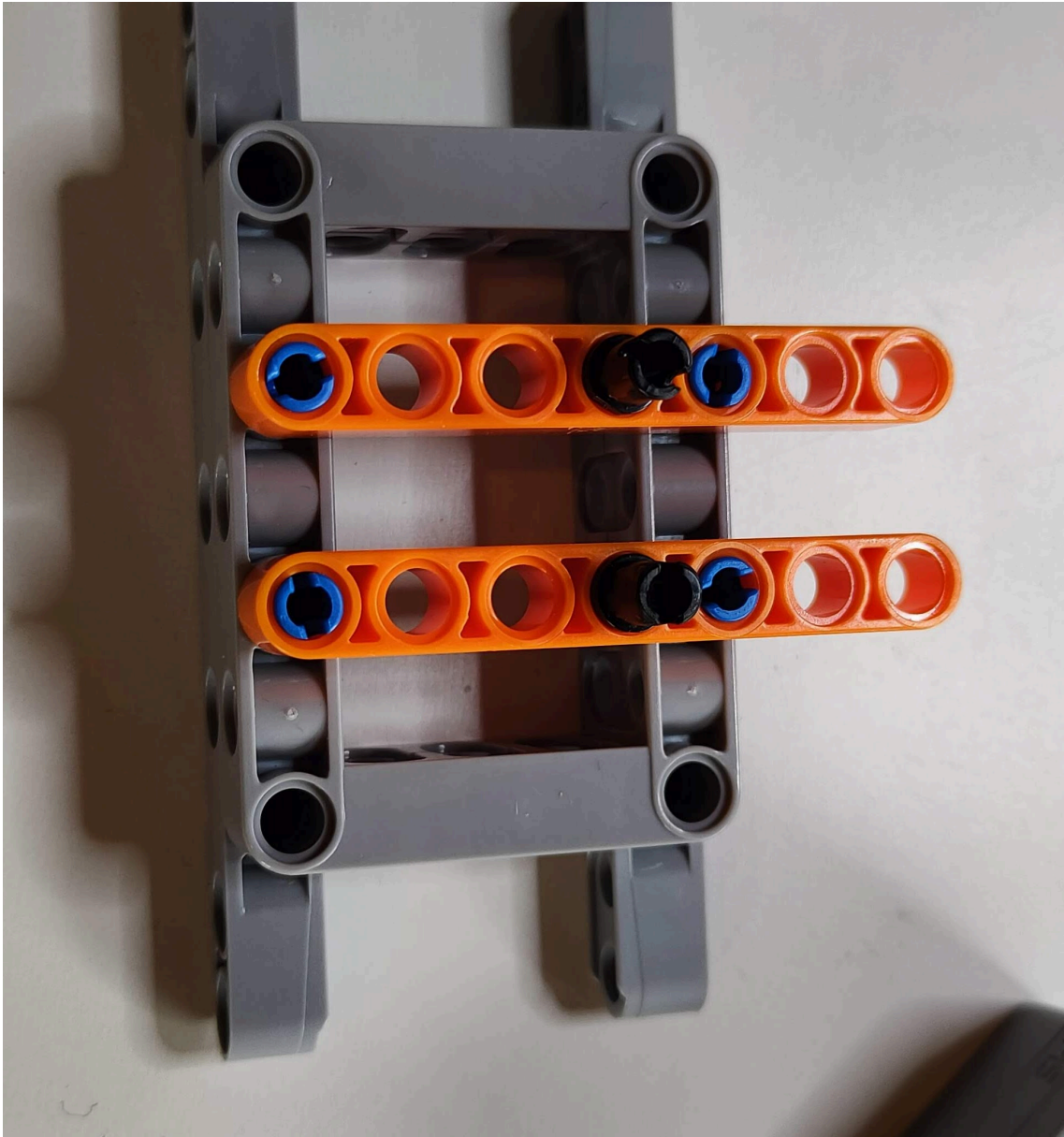




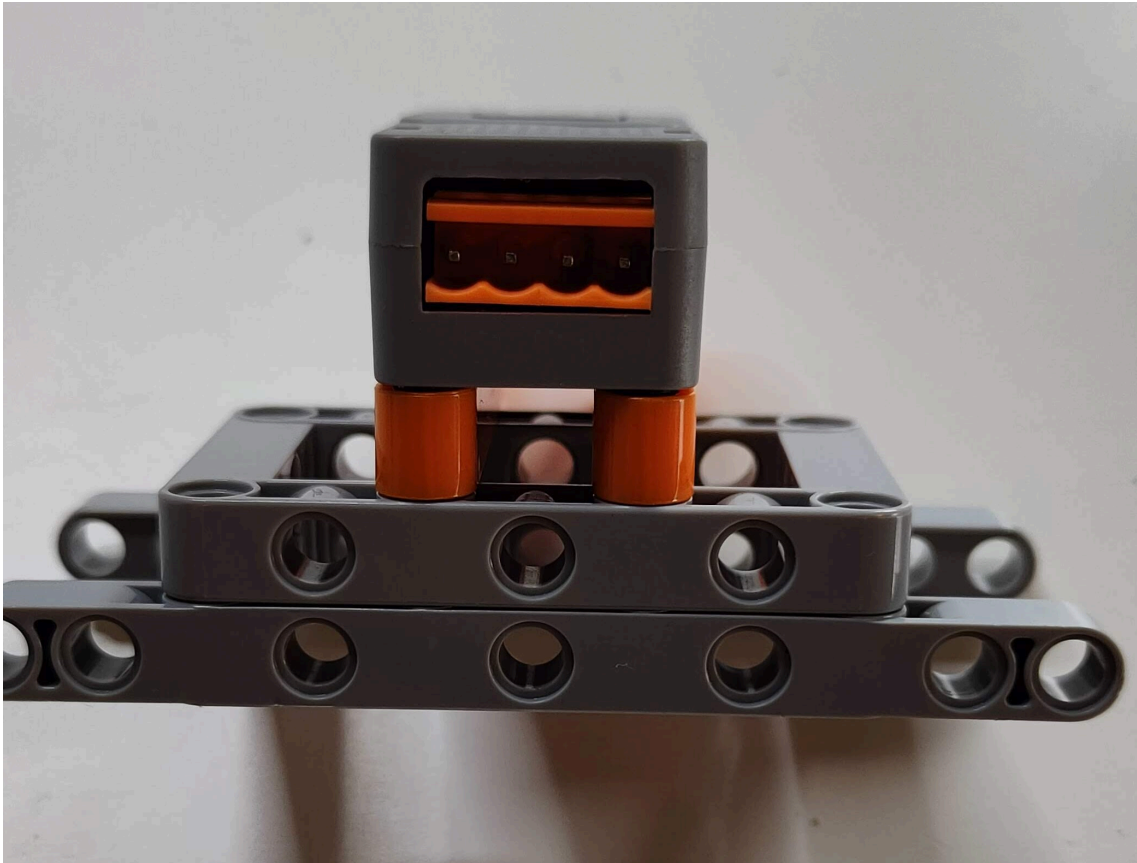


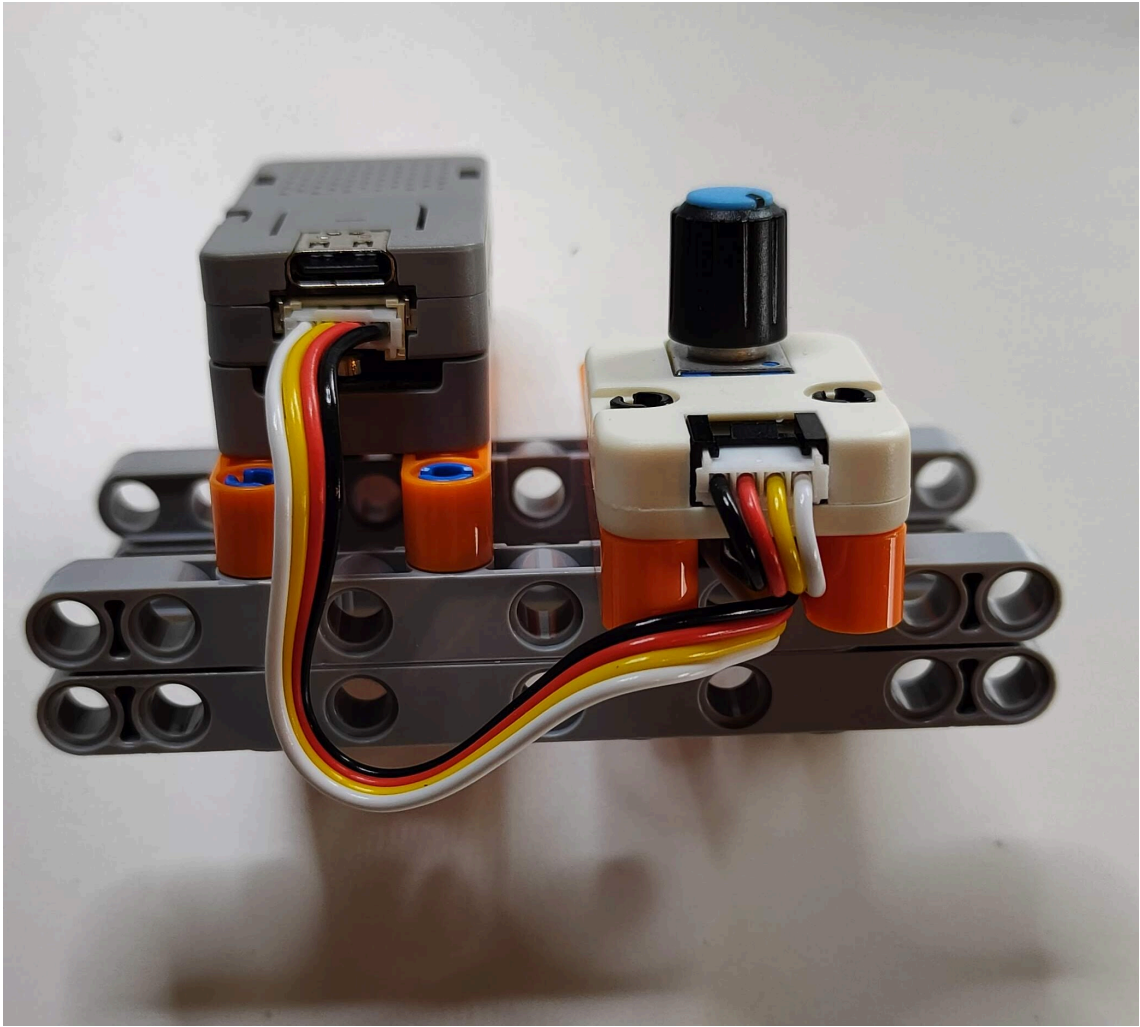


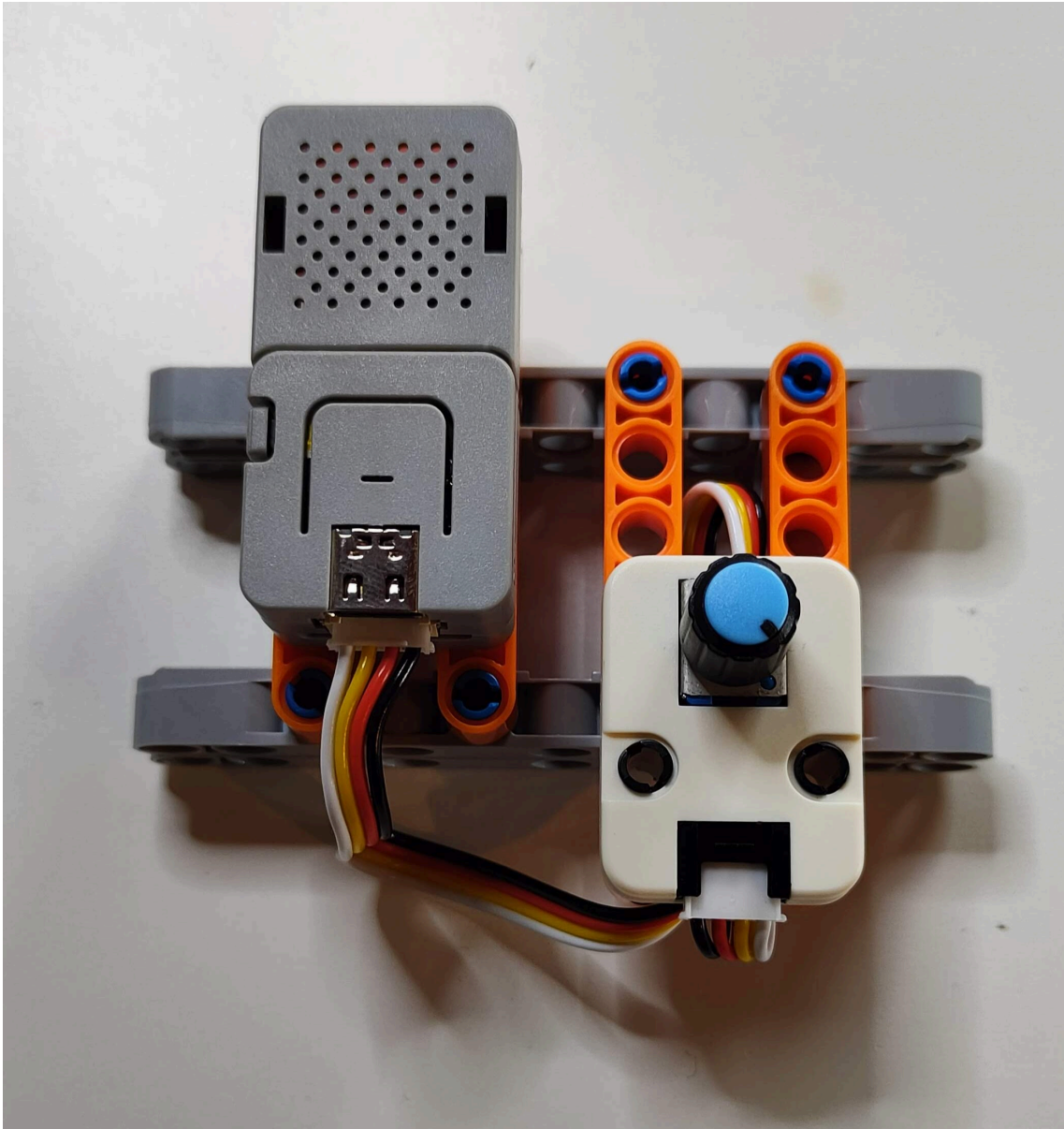


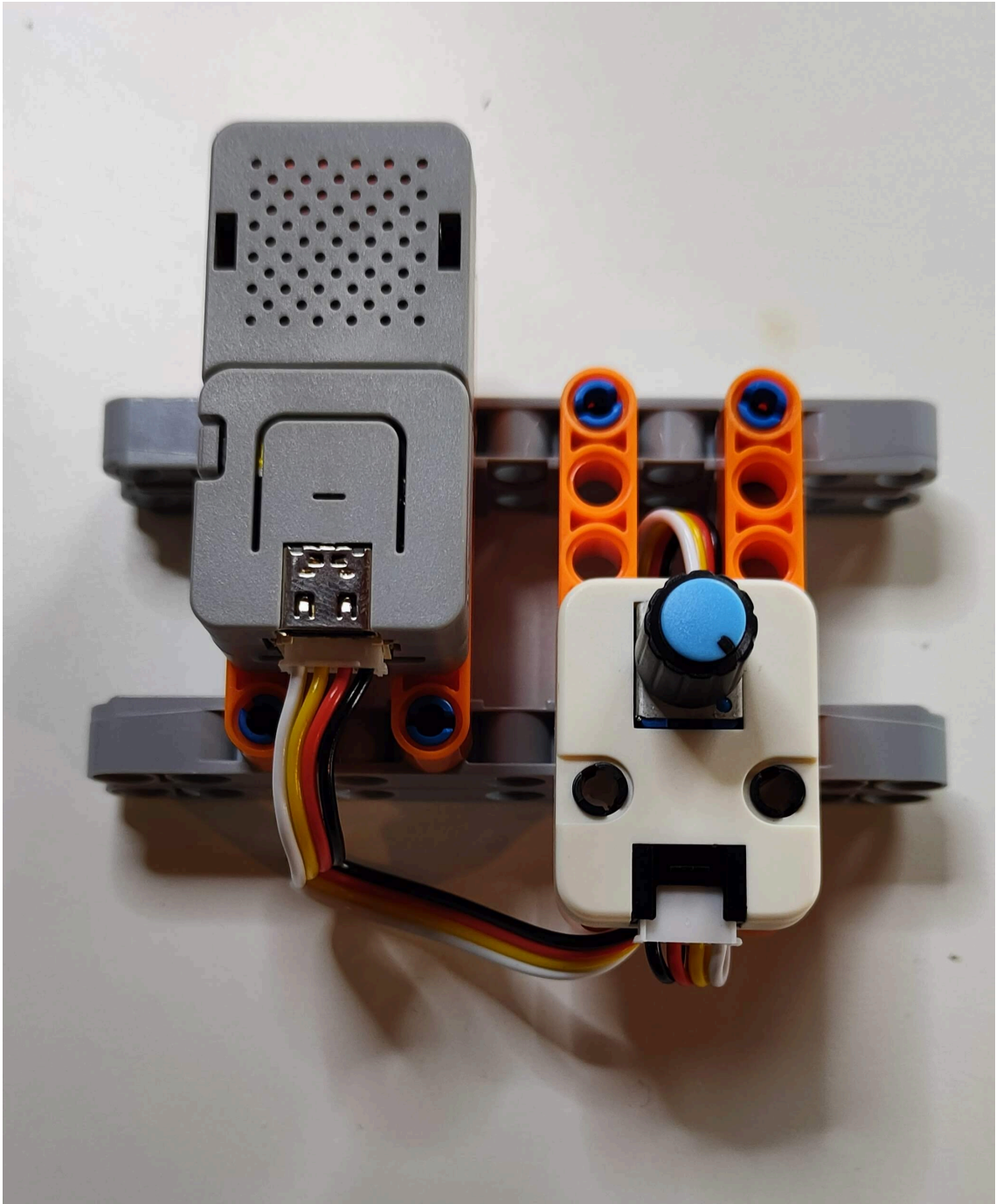


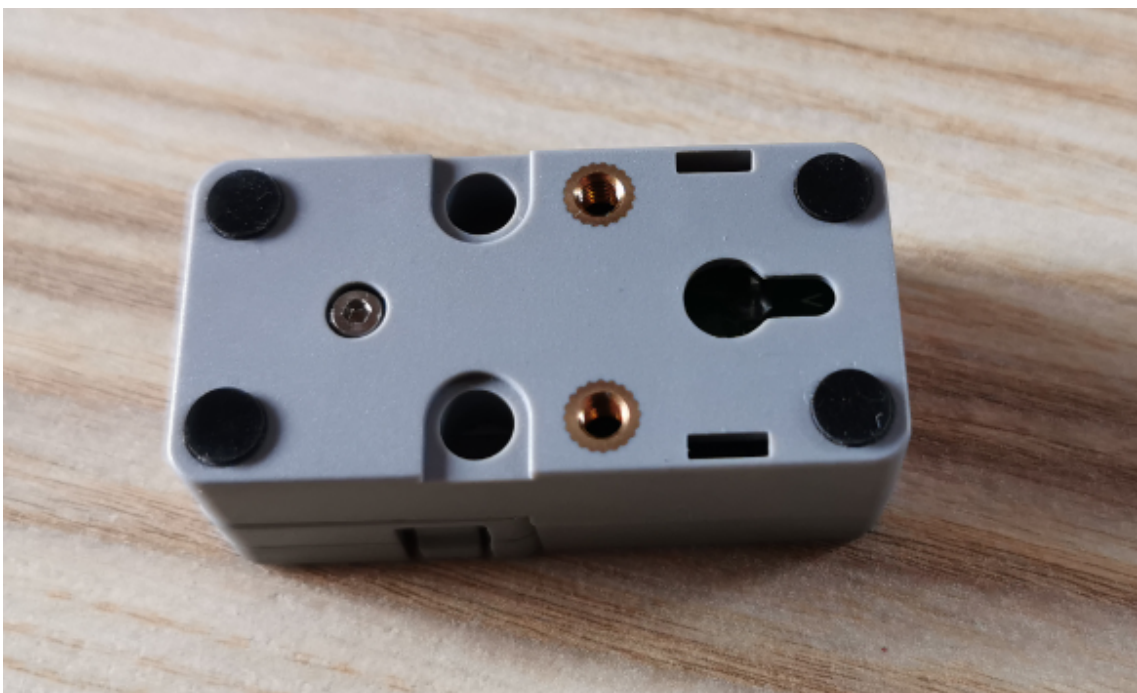
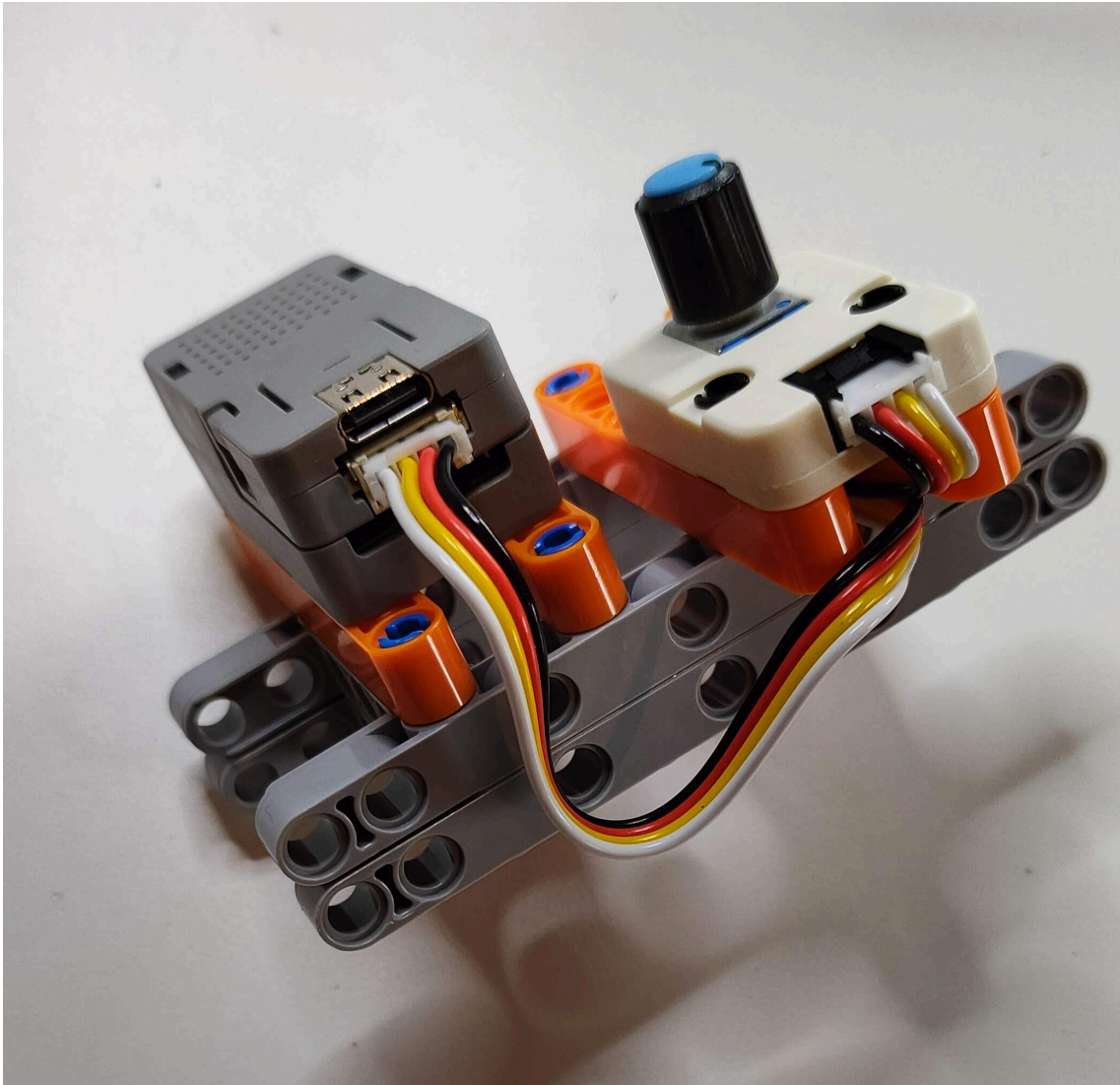


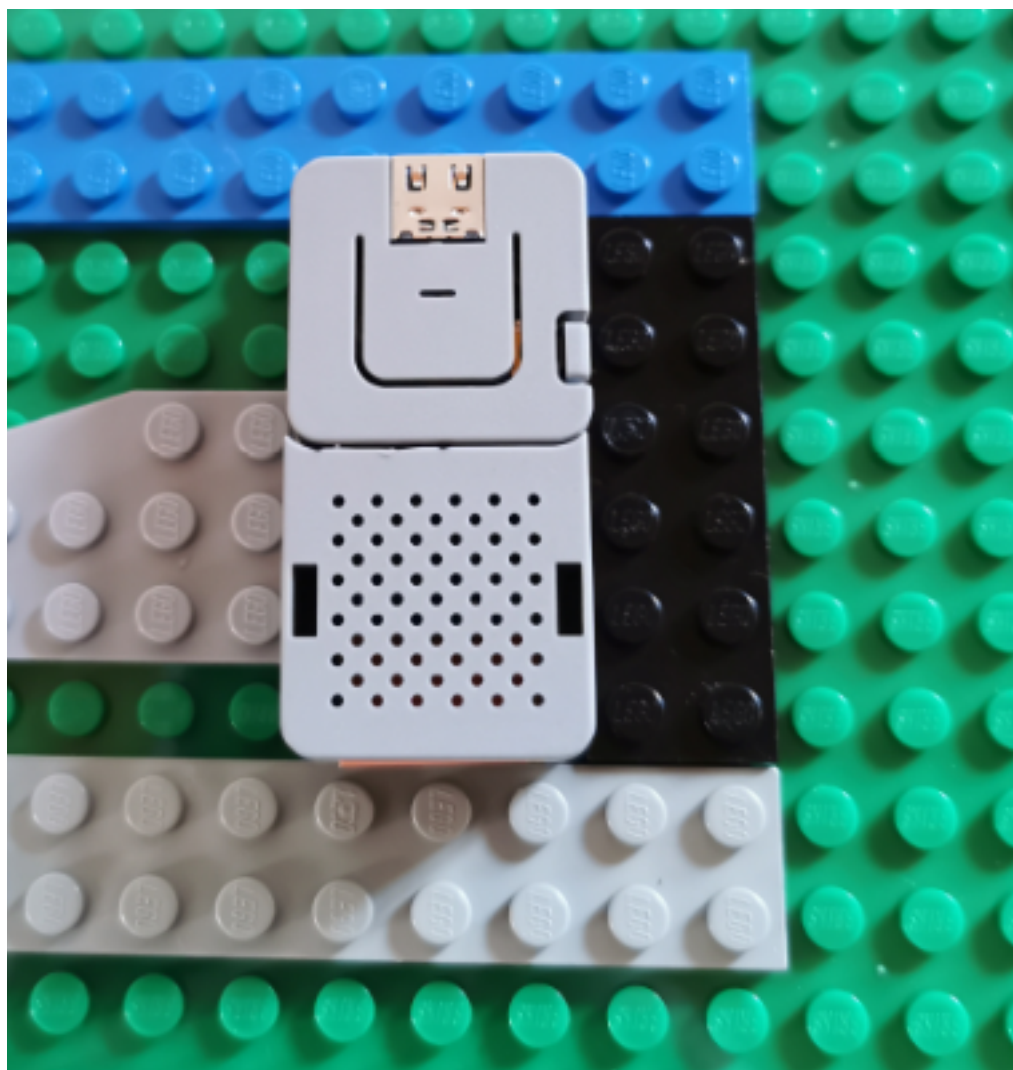












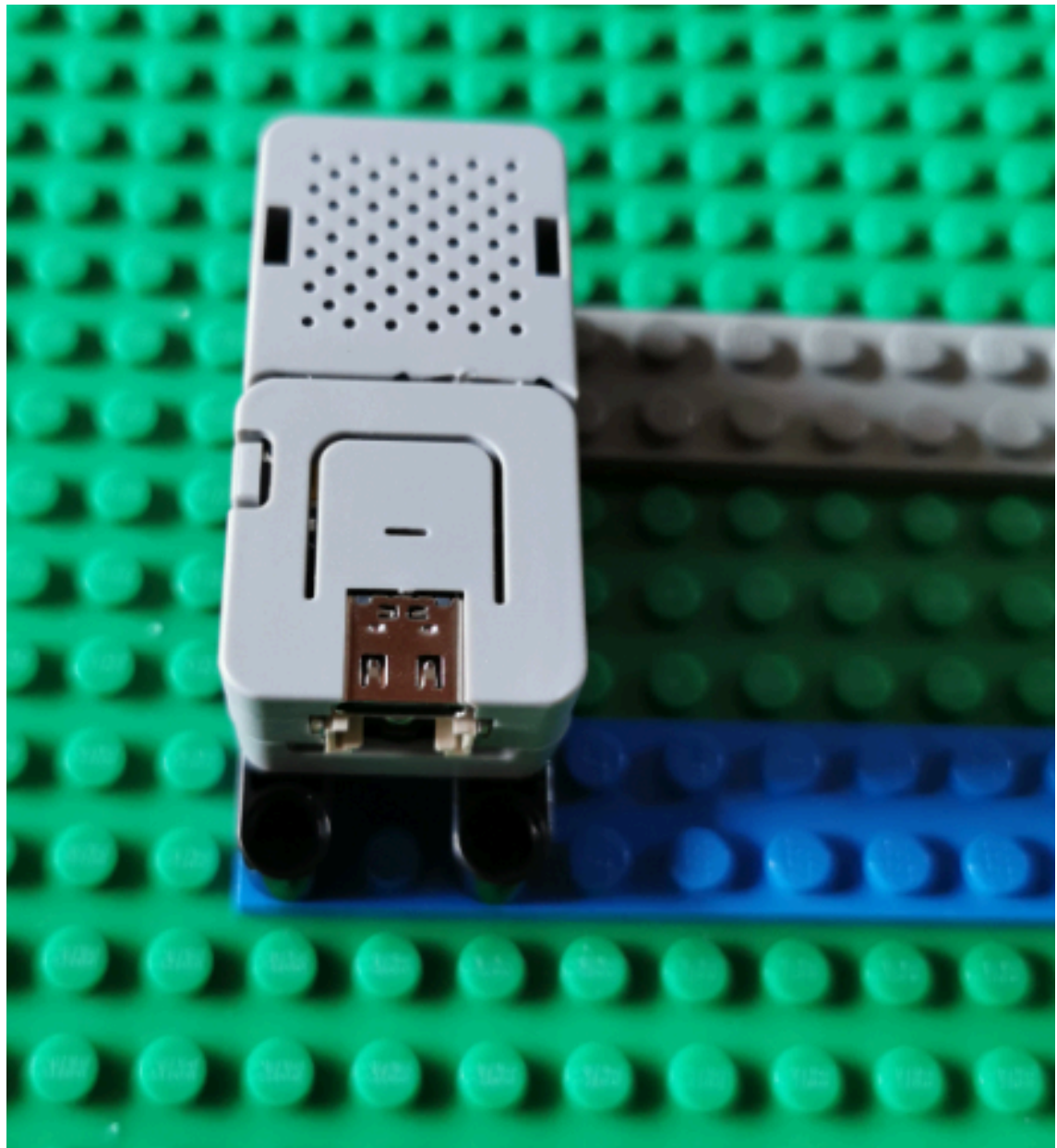




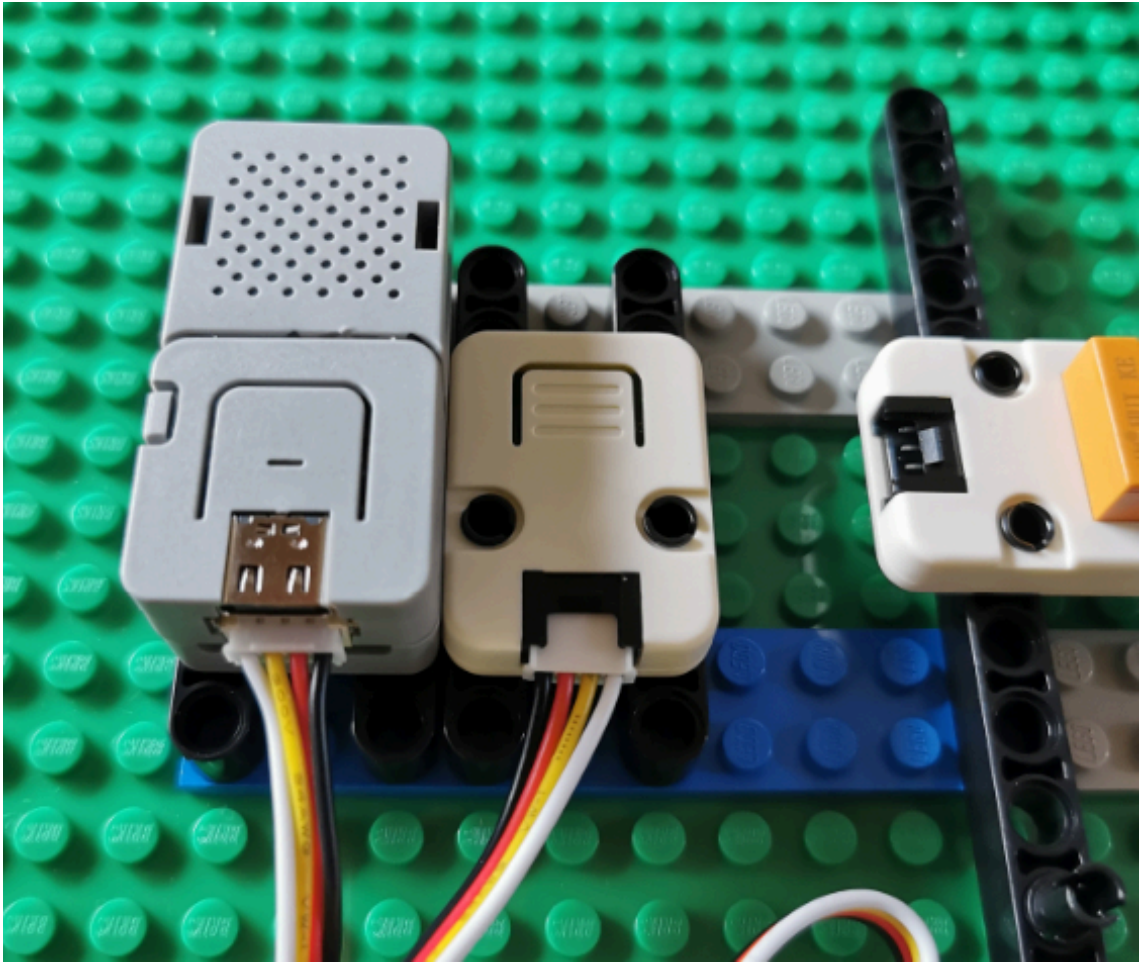


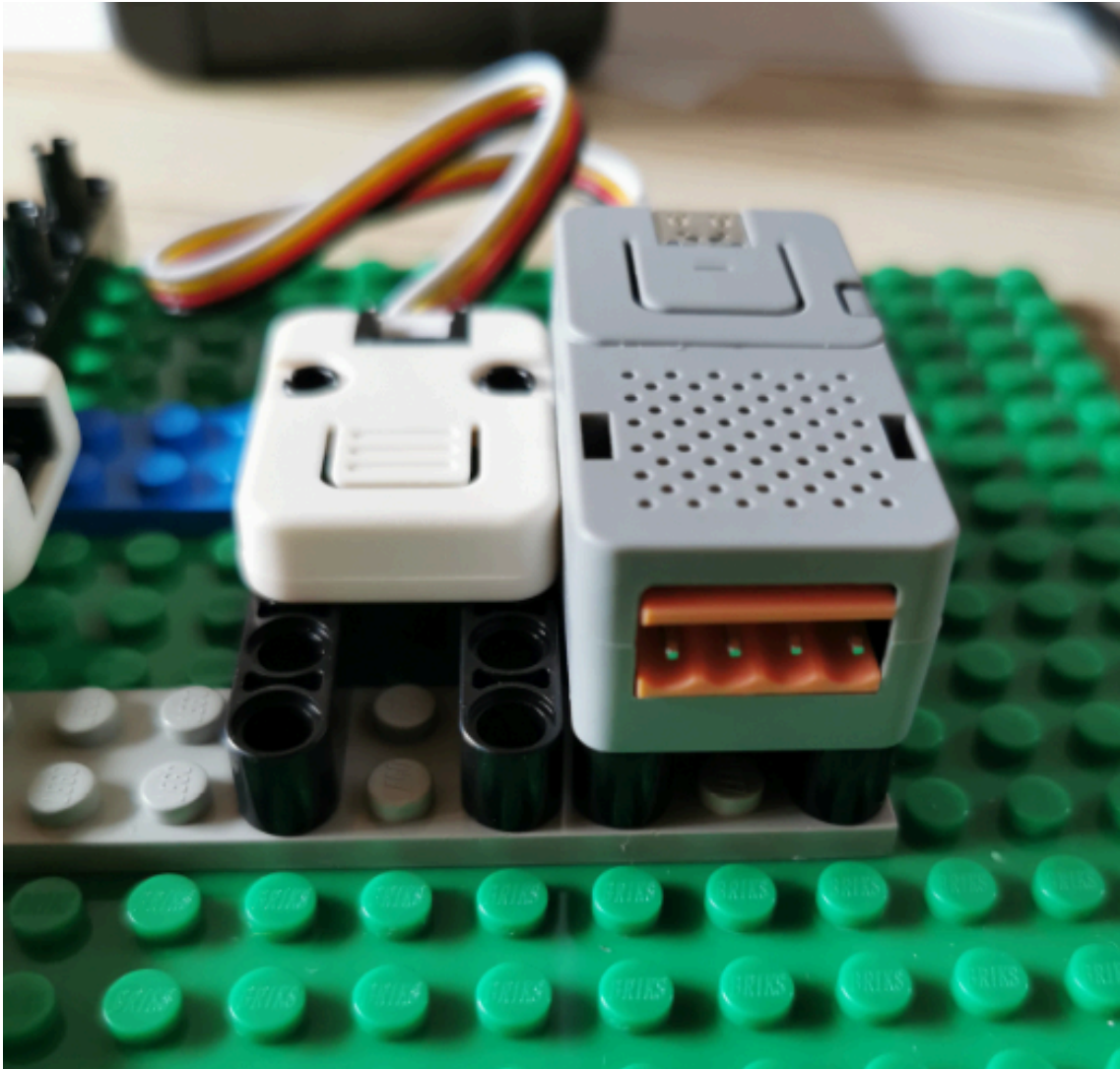




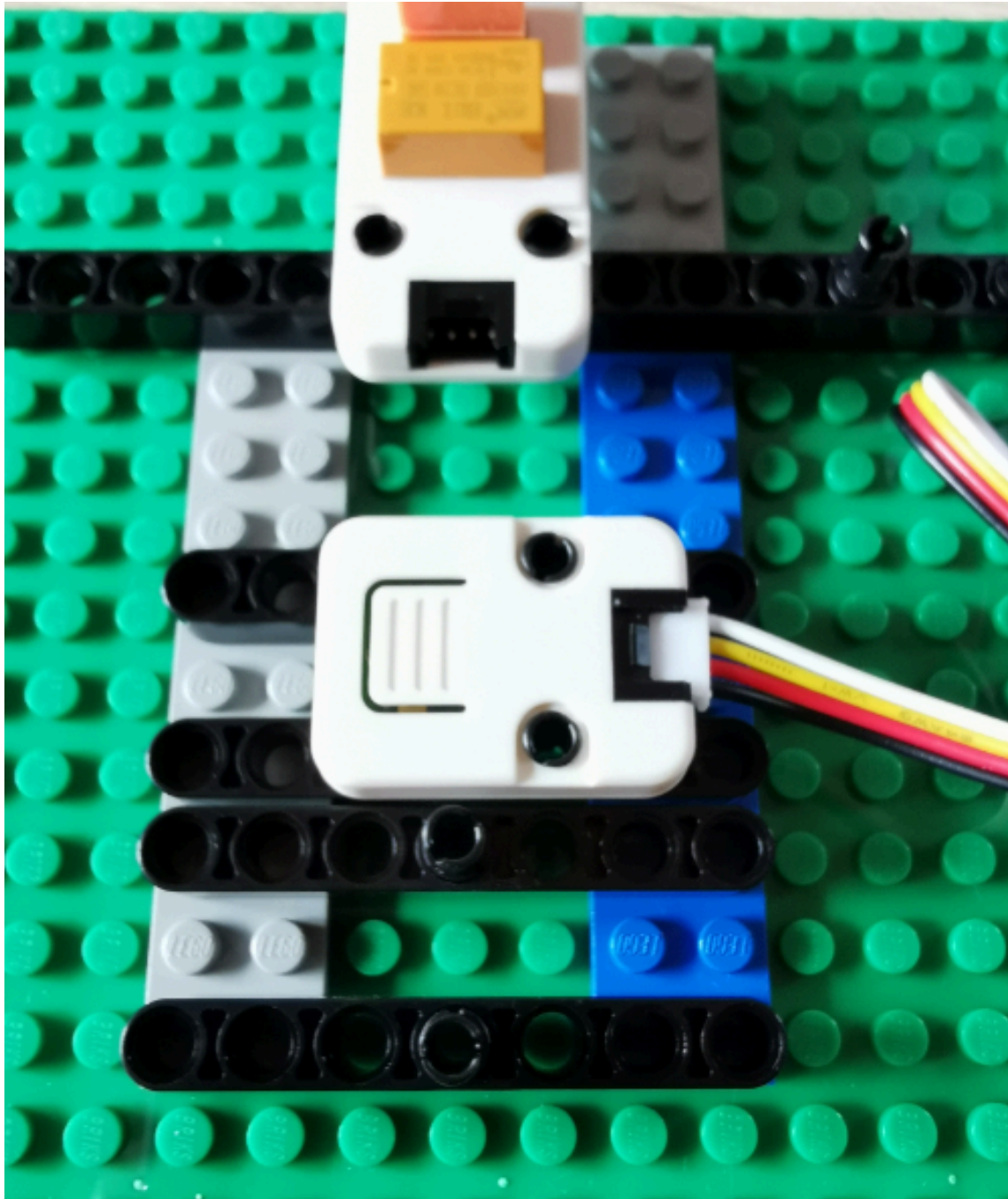


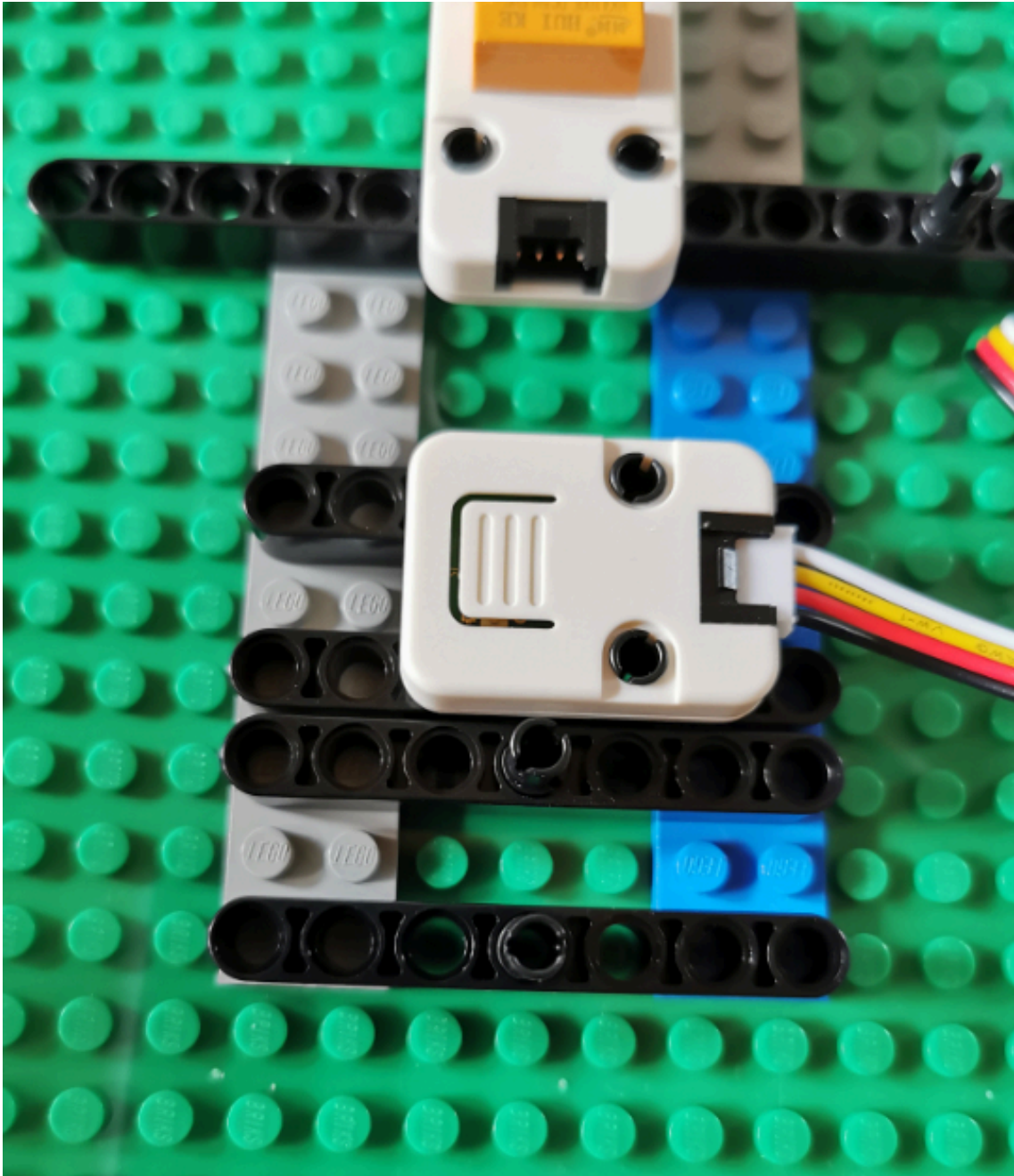












19.1 Podcast

- Eclipse 4diac FORTE: IEC 61499 verstehen – Der LEGO®-Baukasten für Ihre Industrie 4.0 Steuerung
 - DIN EN 61499-1: Der digitale Lego-Baukasten für flexible Automatisierung und smarte Zukunft
 - DIN EN 61499-1: Die Lego-Steine für flexible und ereignisgesteuerte Industriesteuerungen
 - EventFBs nach IEC 61499: Legosteine der Automatisierung – So funktionieren Ereignis-Funktionsbausteine
 - IEC 61499-1: Der Funktionsbaustein – Lego-Prinzip für moderne Steuerungssysteme in Land- und Baumaschinen
-

19.1.1 Passende Themen-Unterseiten auf ms-muc-docs.de

-  Eclipse 4diac IDE & Farb-Referenz auf ms-muc-docs.de

20. Optional

21. Paket



21.1 Podcast

- Kraftpakete im Einsatz: Das Geheimnis der Hydraulikzylinder – Von Baggern bis Hightech-Maschinen

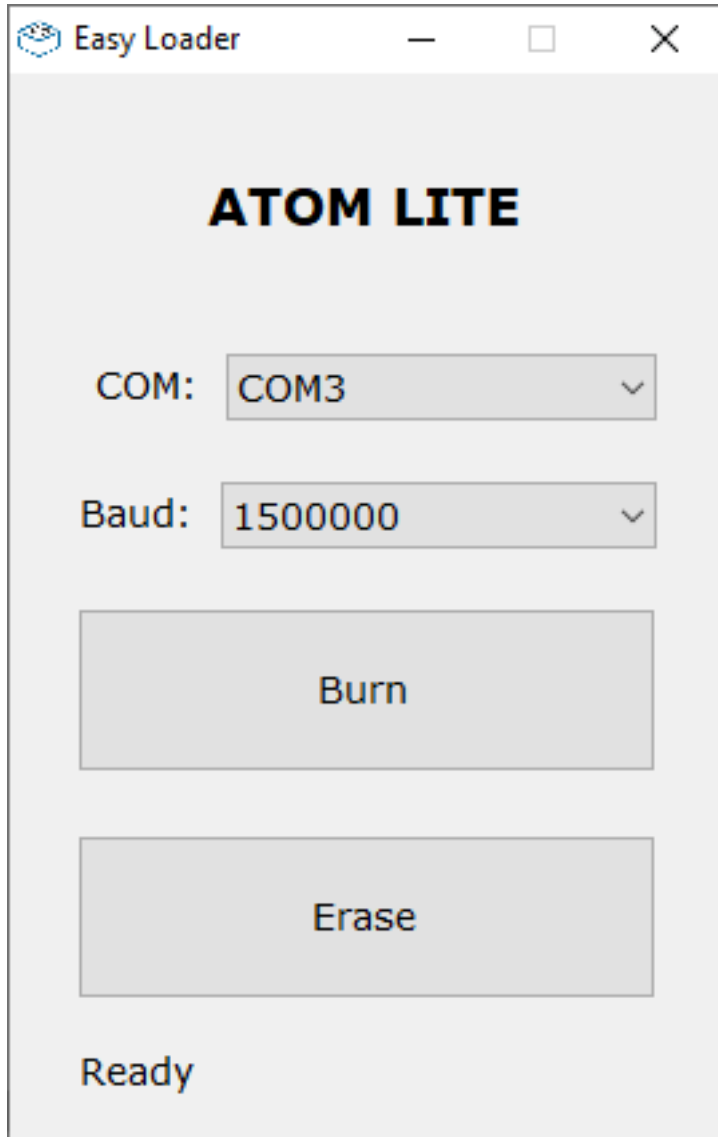
21.2 Video

- [Chip wird Kraftpaket](#)

22. Selbsttest

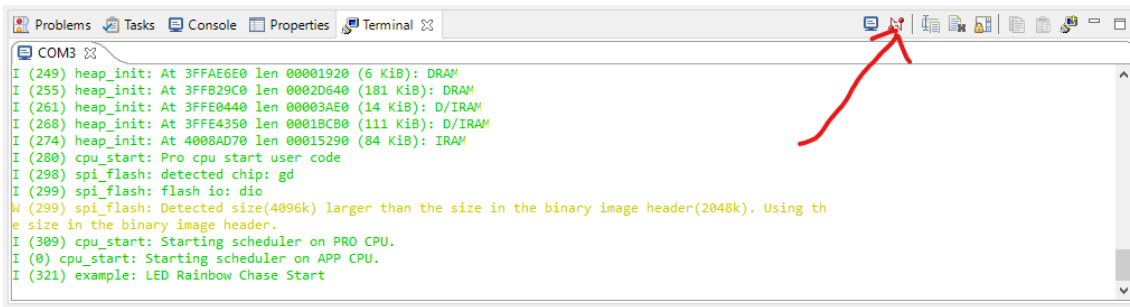
Öffnen Sie die Datei

M:\Landmaschinenmechanik\Unterricht\SL\esp-
idf\AtomSelbsttest\EasyLoader_ATOM_LITE_FactoryTest.exe



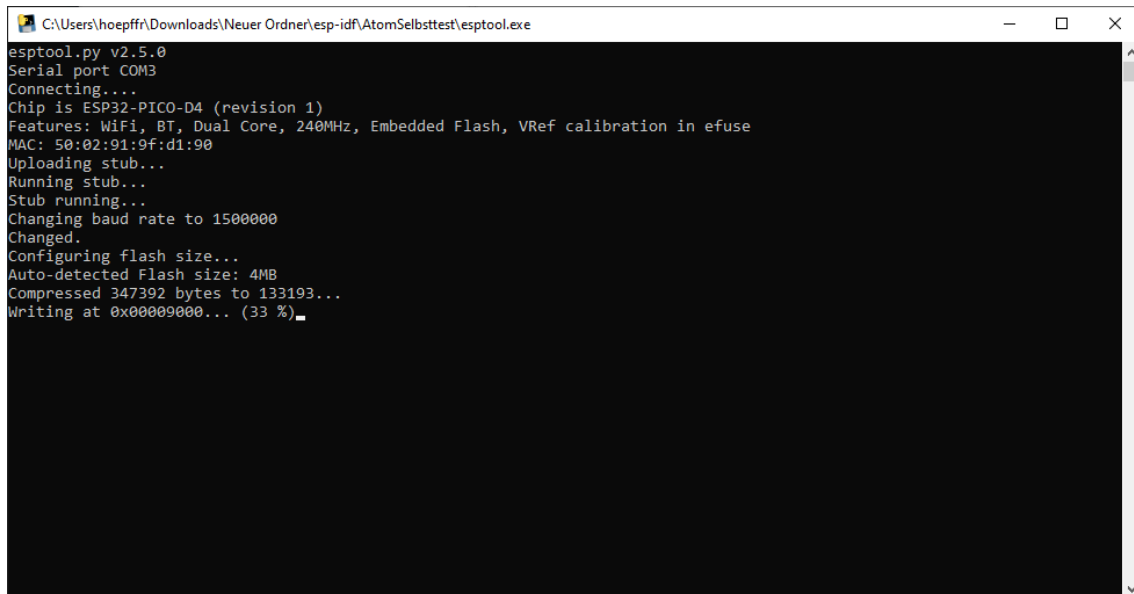
Mit Erase können Sie den ATOM immer auf Werkseinstellungen zurücksetzen,
mit BURN laden sie ein Selbsttestprogramm drauf.

Falls Sie noch mit dem Terminal verbunden sind: drücken Sie hier:



```
COM3
I (249) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (255) heap_init: At 3FFB29C0 len 0002D640 (181 KiB): DRAM
I (261) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (268) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (274) heap_init: At 4008AD70 len 00015290 (84 KiB): IRAM
I (280) cpu_start: Pro cpu start user code
I (298) spi_flash: detected chip: gd
I (299) spi_flash: flash io: dio
W (299) spi_flash: Detected size(4096k) larger than the size in the binary image header(2048k). Using th
e size in the binary image header.
I (309) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
I (321) example: LED Rainbow Chase Start
```

und versuchen Sie erneut BURN oder ERASE
sollte so aussehen:



```
C:\Users\hoepffr\Downloads\Neuer Ordner\esp-idf\AtomSelbsttest\esptool.exe
esptool.py v2.5.0
Serial port COM3
Connecting...
Chip is ESP32-PICO-D4 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, Embedded Flash, VRef calibration in efuse
MAC: 50:02:91:9f:d1:90
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 1500000
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Compressed 347392 bytes to 133193...
Writing at 0x00009000... (33 %)

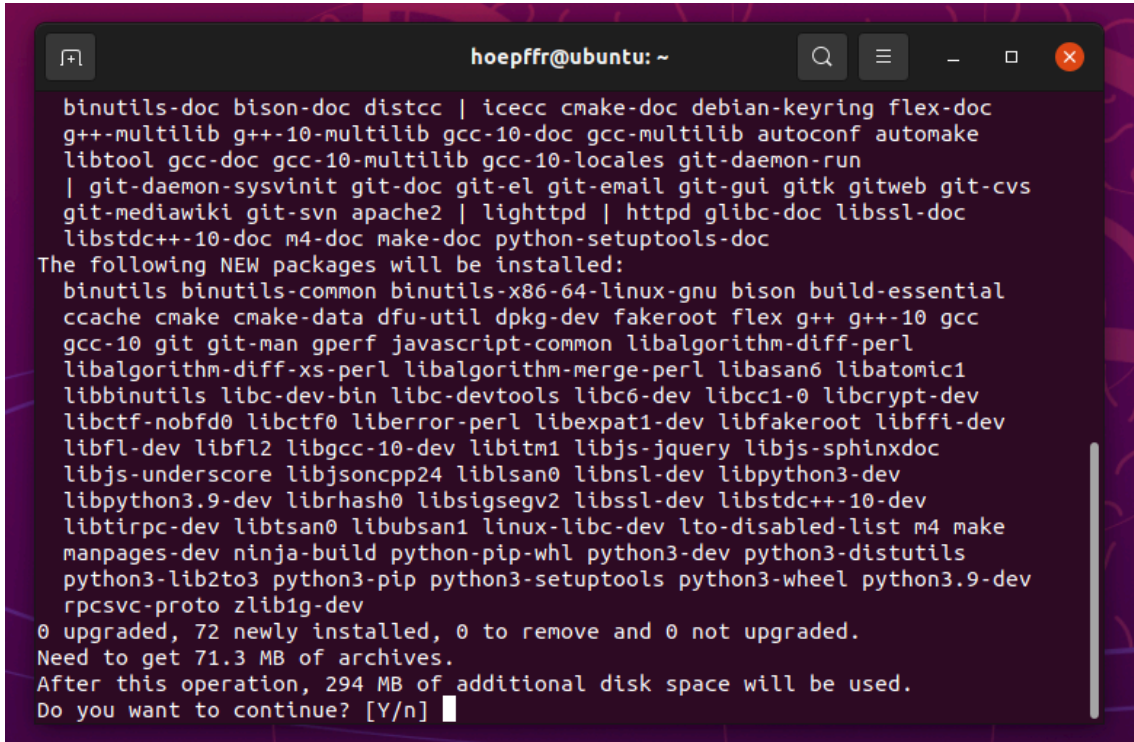
```

um den Selbsttest durchzuführen muss die Taste des ATOM betätigt werden.

23. Setting-up (LINUX)

<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/linux-setup.html>

```
sudo apt install git wget flex bison gperf python3 python3-pip python3-setuptools cmake  
ninja-build ccache libffi-dev libssl-dev dfu-util libusb-1.0-0
```



```
binutils-doc bison-doc distcc | icecc cmake-doc debian-keyring flex-doc  
g++-multilib g++-10-multilib gcc-10-doc gcc-multilib autoconf automake  
libtool gcc-doc gcc-10-multilib gcc-10-locales git-daemon-run  
| git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs  
git-mediawiki git-svn apache2 | lighttpd | httpd glibc-doc libssl-doc  
libstdc++-10-doc m4-doc make-doc python-setuptools-doc  
The following NEW packages will be installed:  
binutils binutils-common binutils-x86-64-linux-gnu bison build-essential  
ccache cmake cmake-data dfu-util dpkg-dev fakeroot flex g++ g++-10 gcc  
gcc-10 git git-man gperf javascript-common libalgorithm-diff-perl  
libalgorithm-diff-xs-perl libalgorithm-merge-perl libasan6 libatomic1  
libbinutils libc-dev-bin libc-devtools libc6-dev libcc1-0 libcrypt-dev  
libctf-nobfd0 libctf0 liberror-perl libexpat1-dev libfakeroot libffi-dev  
libfl-dev libfl2 libgcc-10-dev libitm1 libjs-jquery libjs-sphinxdoc  
libjs-underscore libjsoncpp24 liblsan0 libnsl-dev libpython3-dev  
libpython3.9-dev librtmp1 libsigsegv2 libssl-dev libstdc++-10-dev  
libtirpc-dev libtsan0 libubsan1 linux-libc-dev lto-disabled-list m4 make  
manpages-dev ninja-build python-pip-whl python3-dev python3-distutils  
python3-lib2to3 python3-pip python3-setuptools python3-wheel python3.9-dev  
rpcsvc-proto zlib1g-dev  
0 upgraded, 72 newly installed, 0 to remove and 0 not upgraded.  
Need to get 71.3 MB of archives.  
After this operation, 294 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

mit Yes bestätigen.

Nun laden Sie ESP-IDF herunter.

```
mkdir -p ~/esp
```

```
cd ~/esp
```

```
git clone --recursive https://github.com/espressif/esp-idf.git
```

ESP-IDF wird in diesen Ordner heruntergeladen: ~/esp/esp-idf

Navigiere jetzt mit `cd ~/esp/esp-idf` in den Ordner.

Mache die Install Datei mit `chmod +x install.sh` ausführbar. Danach führe die Datei aus. `./install.sh`

Fertig.

24. Setting-up-II

dieses Setup steht ab ca. Januar 2022 zur Verfügung.

<<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/index.html>

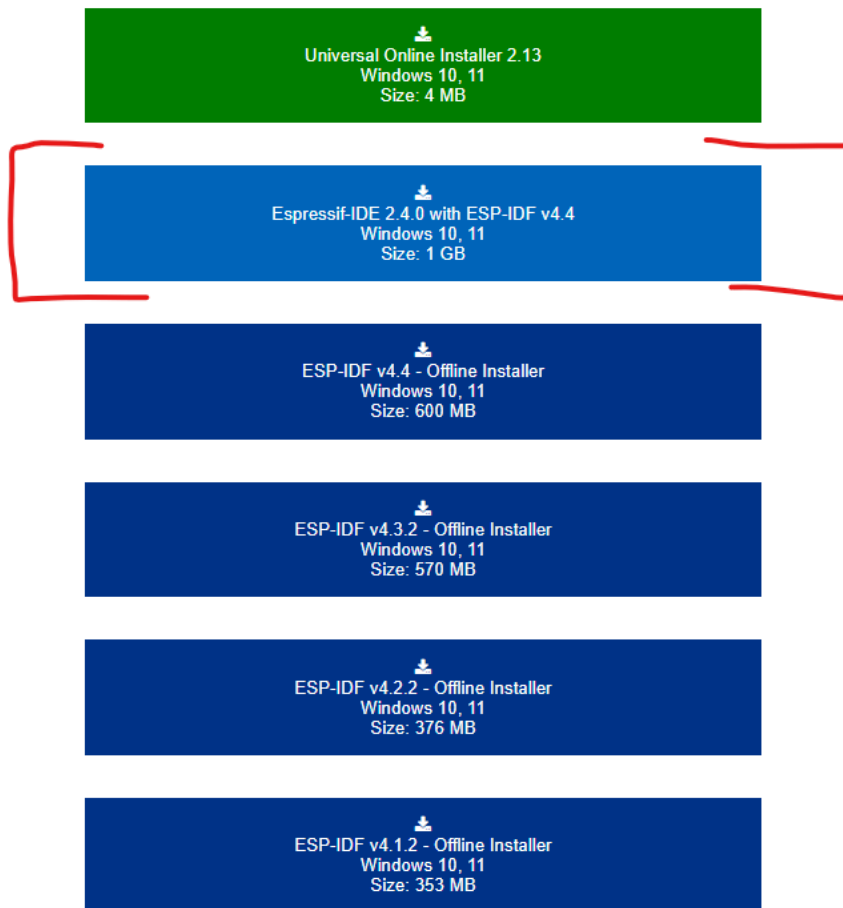
<<https://dl.espressif.com/dl/esp-idf/?idf=4.4>

Espressif-IDE 2.4.0 with ESP-IDF v4.4 Windows 10, 11 Size: 1 GB

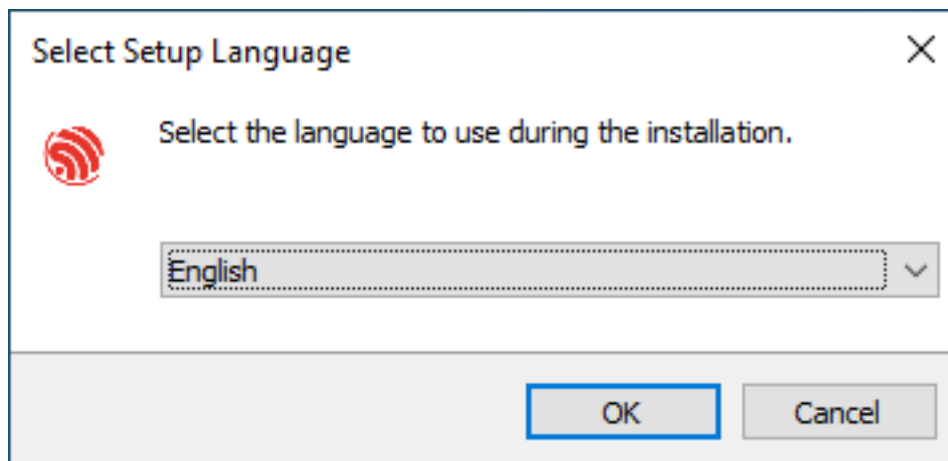
espressif-ide-setup-espressif-ide-2.4.1-with-esp-idf-4.4.exe

ESP-IDF Windows Installer Download

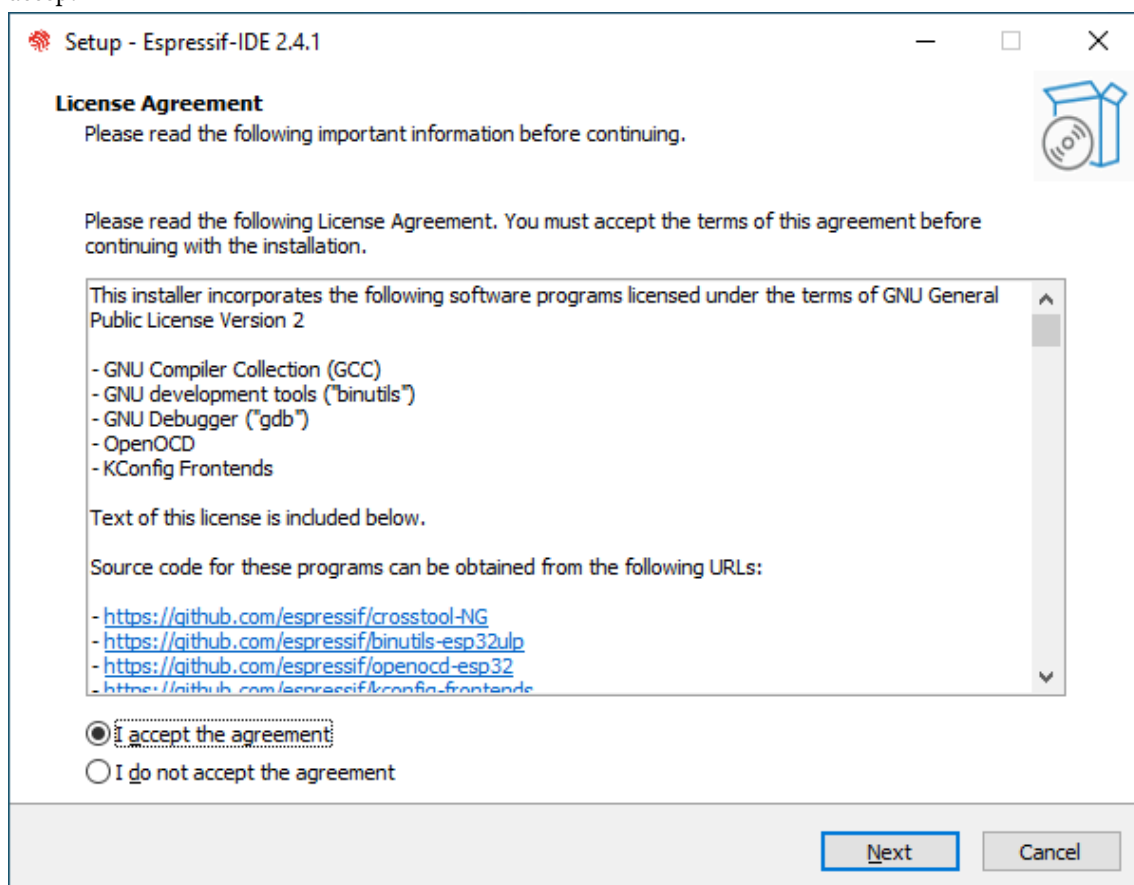
Open Source IoT Development Framework for ESP32

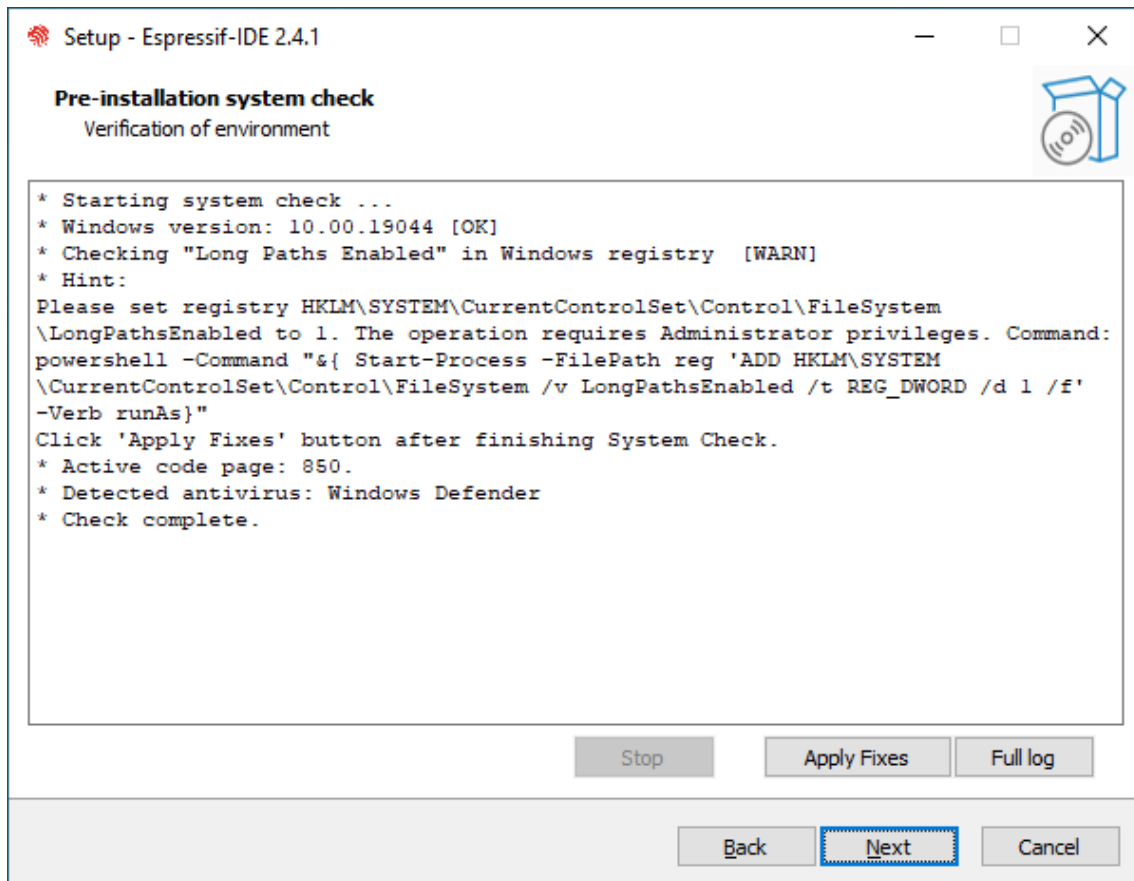


Installation instructions: [ESP-IDF documentation](#) and [Espressif Systems Youtube channel](#).

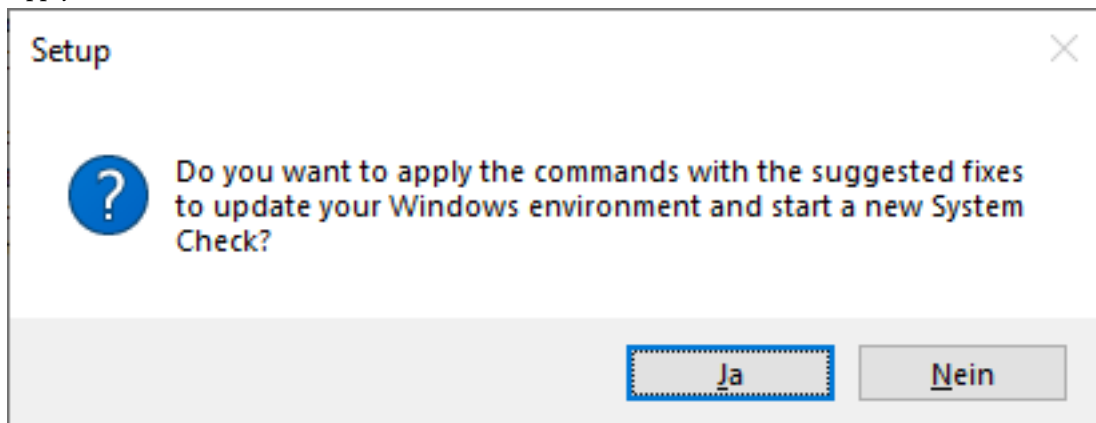


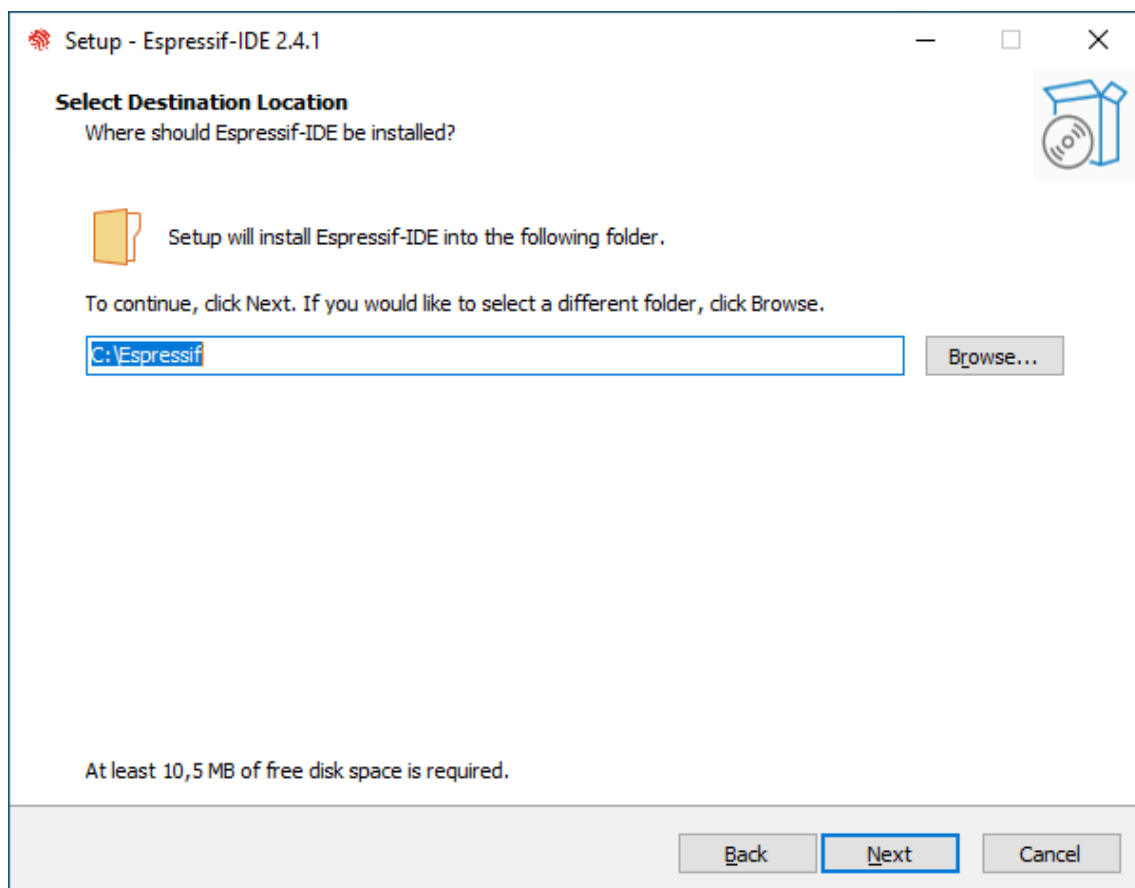
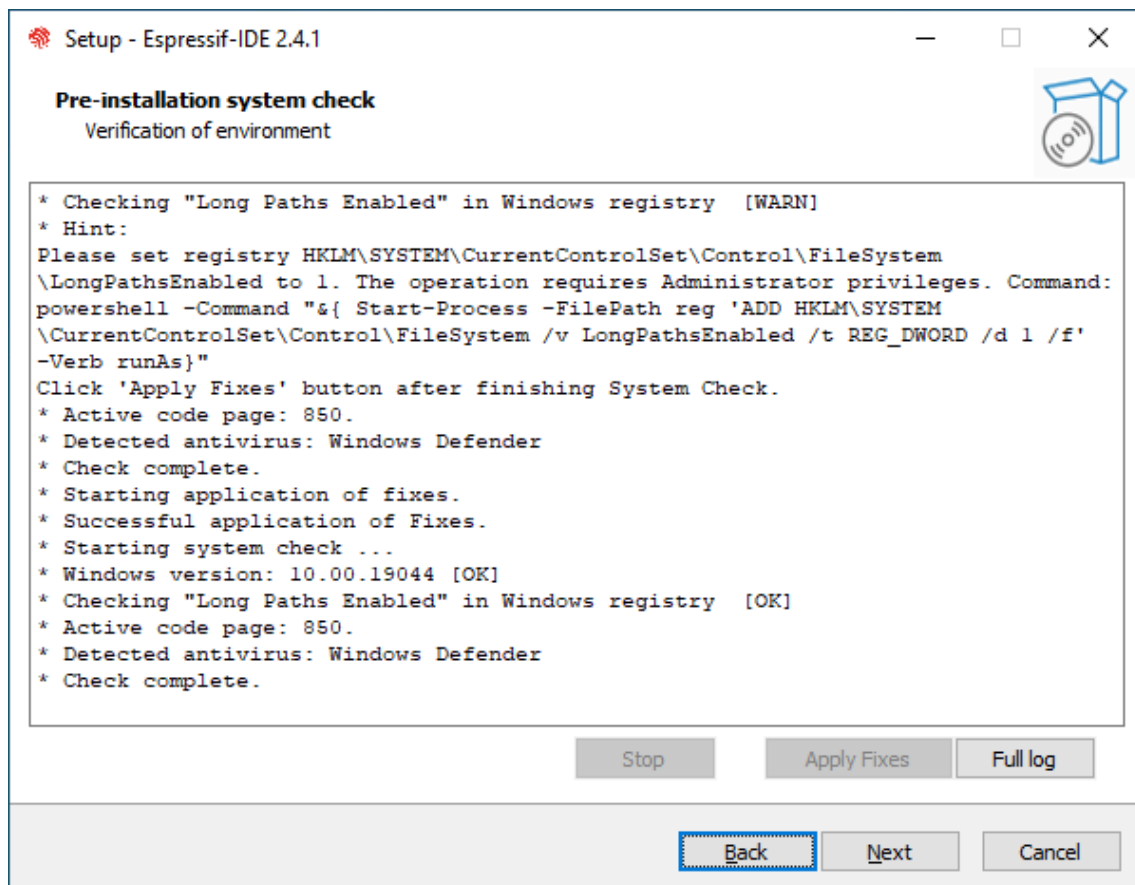
accept:





Apply Fixes





Setup - Espressif-IDE 2.4.1

Select Components

Which components should be installed?



Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

Full installation

☒ Frameworks

742,3 MB

☒ ESP-IDF v4.4

528,2 MB

☒ Development integrations

470,9 MB

☒ Espressif-IDE

470,9 MB

☒ Desktop shortcut

☒ Amazon Corretto 11 JDK

293,7 MB

☒ PowerShell

☒ Windows Terminal Dropdown Menu

☒ Desktop shortcut

☒ Start Menu shortcut

☒ Command Prompt

☒ Desktop shortcut

☒ Start Menu shortcut

☒ Drivers - Requires elevation of privileges

☒ Espressif - WinUSB support for JTAG (ESP32-C3/S3)

☒ FTDI Chip - Virtual COM Port for USB (WROVER, WROOM)

☒ Silicon Labs - Virtual COM Port for USB CP210x (ESP boards)

☒ Chip Targets - more details at <https://products.espressif.com/>

1.672,6 MB

☒ ESP32

407,0 MB

☒ ESP32-C3

473,4 MB

☒ ESP32-S Series

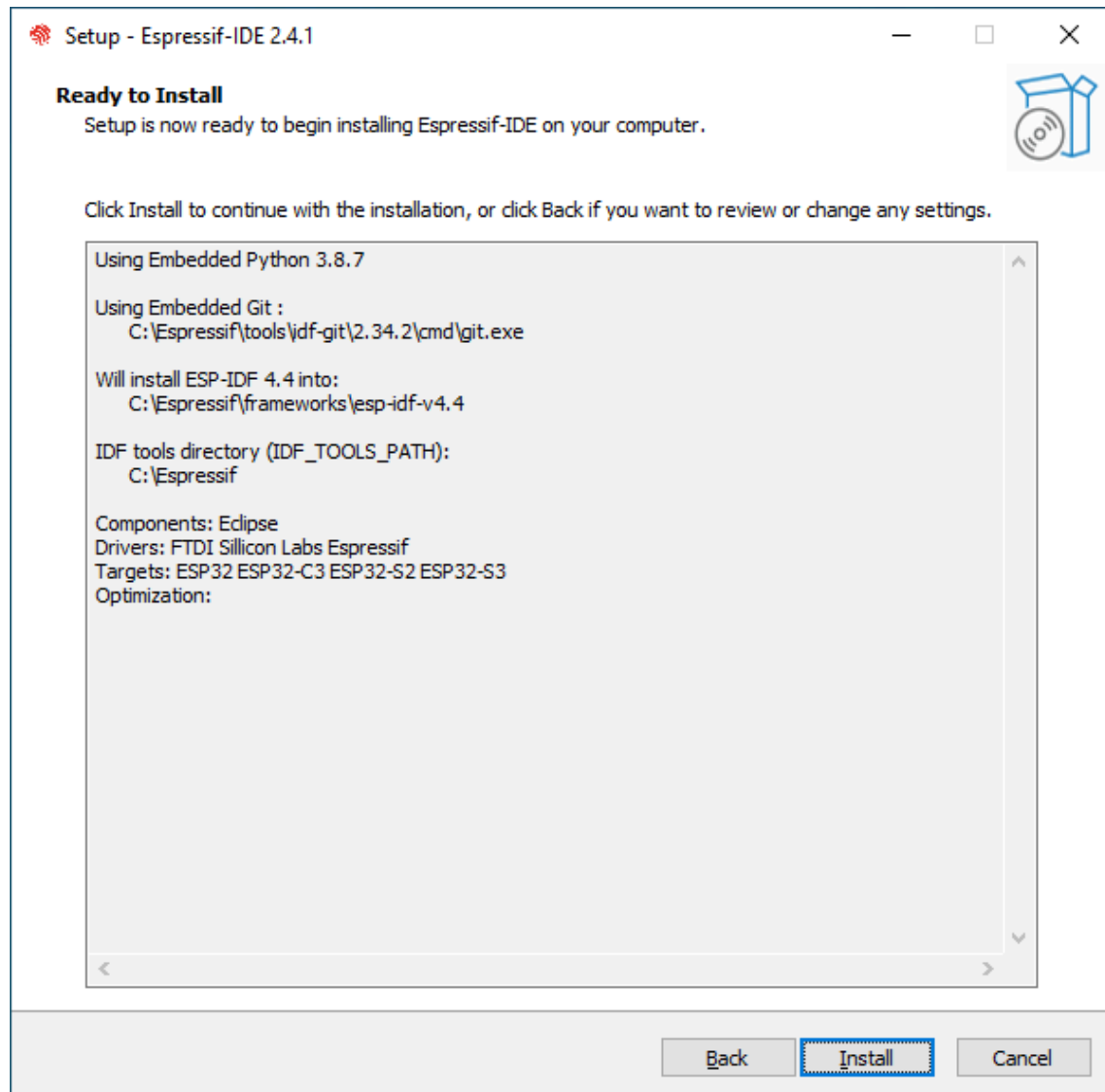
792,3 MB

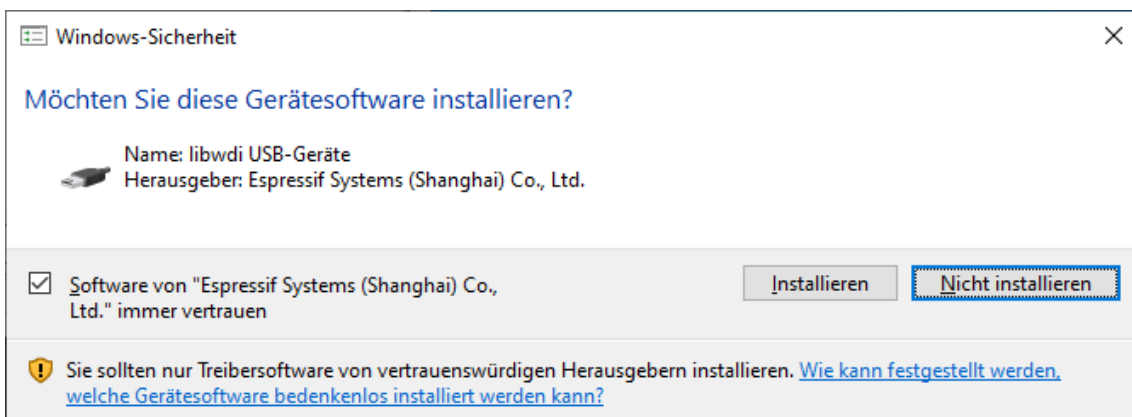
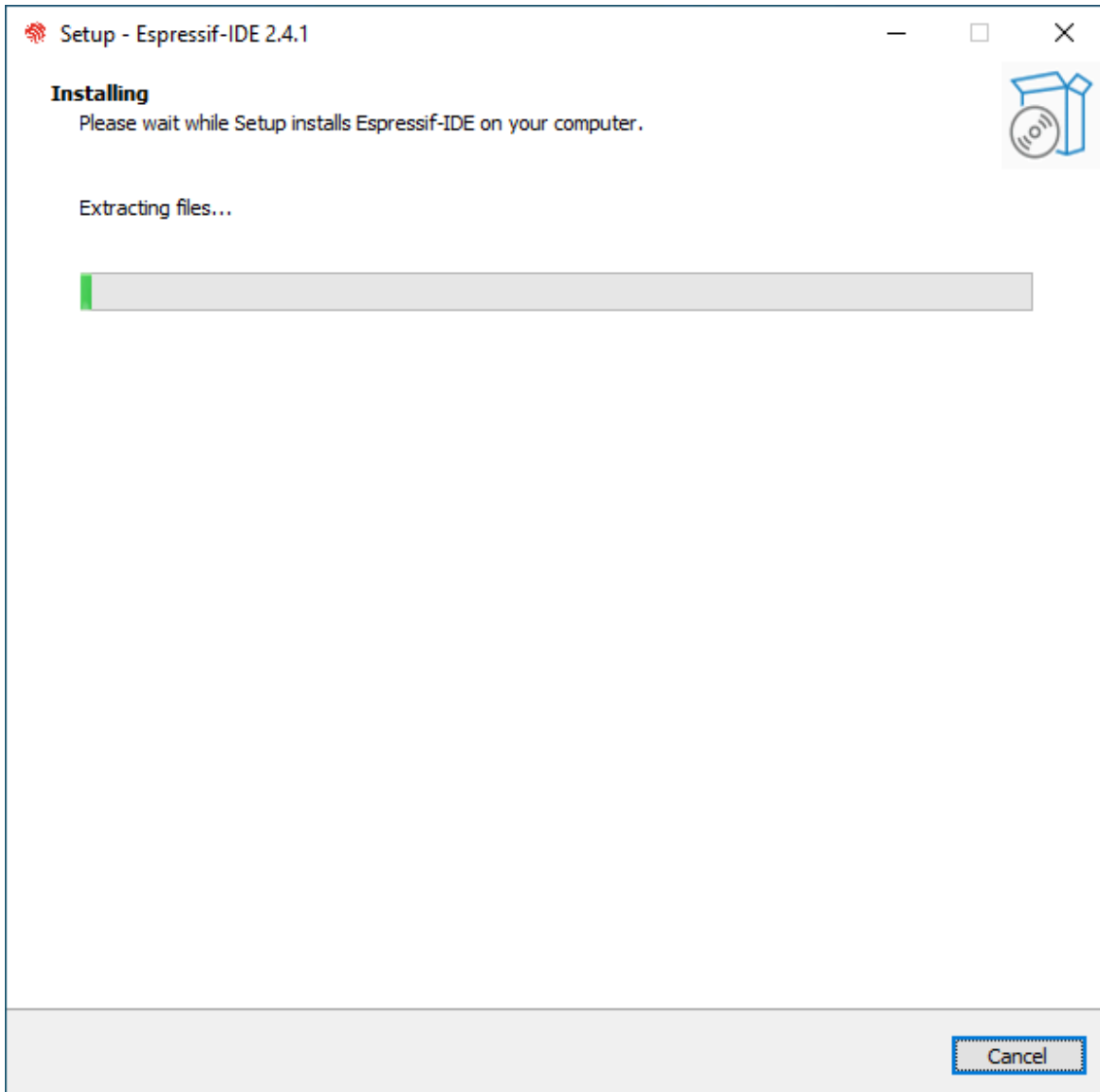
Current selection requires at least 2,83 GB of disk space.

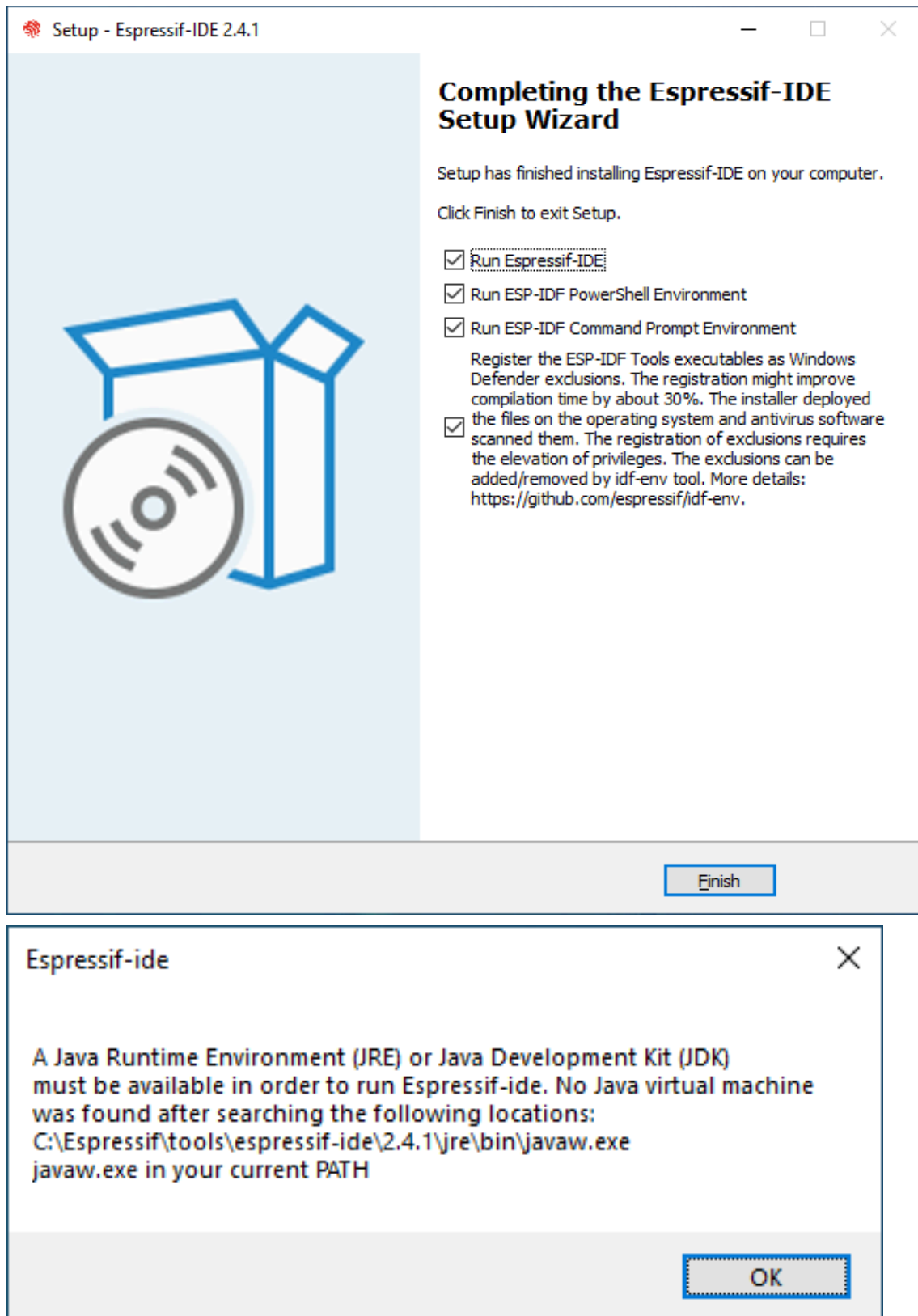
Back

Next

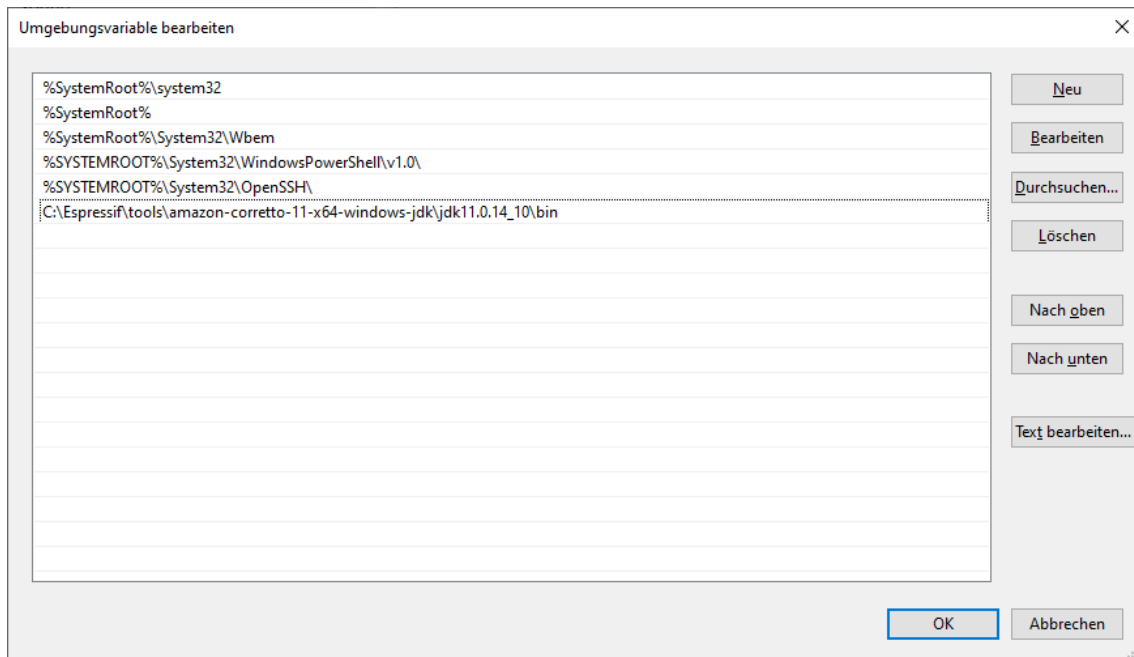
Cancel



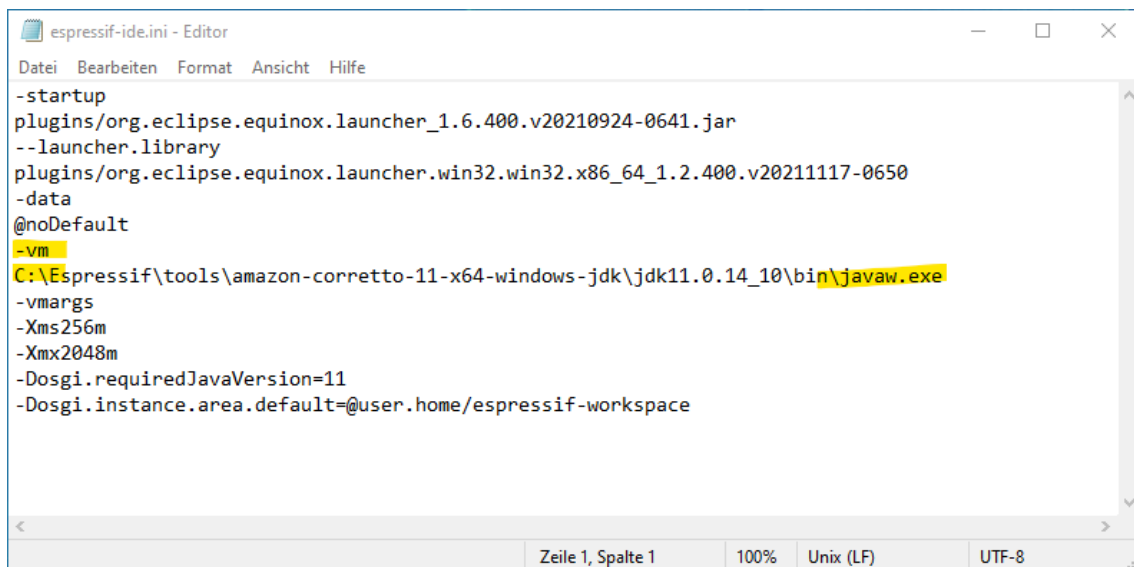




Pfad:



ODER:



25. Setting-up Installation

25.1 Schritt 01:

Hintergrundinfos gibt es hier:

<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/>

Bitte prüfen Sie VORAB ob ihr Username unter C:\Users ohne Umlaut und ohne Leerzeichen ist. ansonsten bekommen Sie schwerwiegende Probleme.

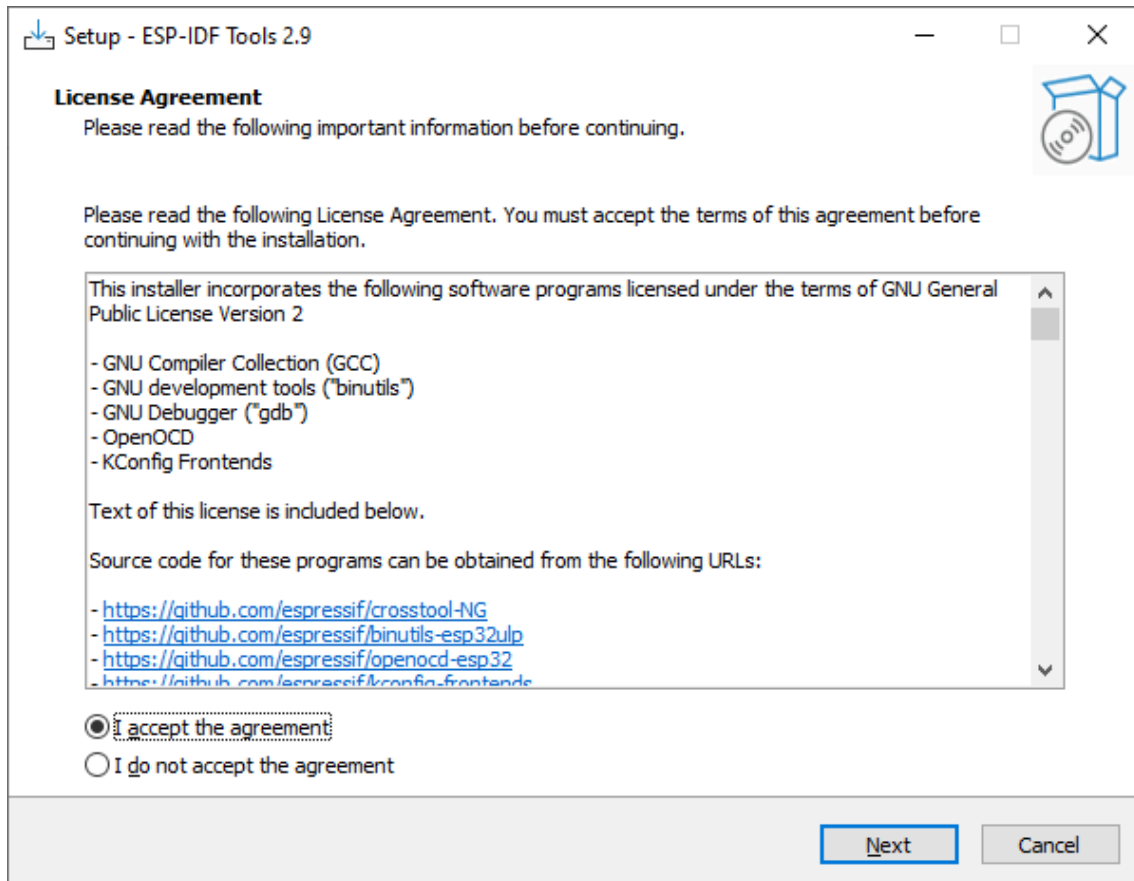
Laden Sie sich die ESP-IDF Tools herunter.

<https://dl.espressif.com/dl/esp-idf/?idf=4.4>

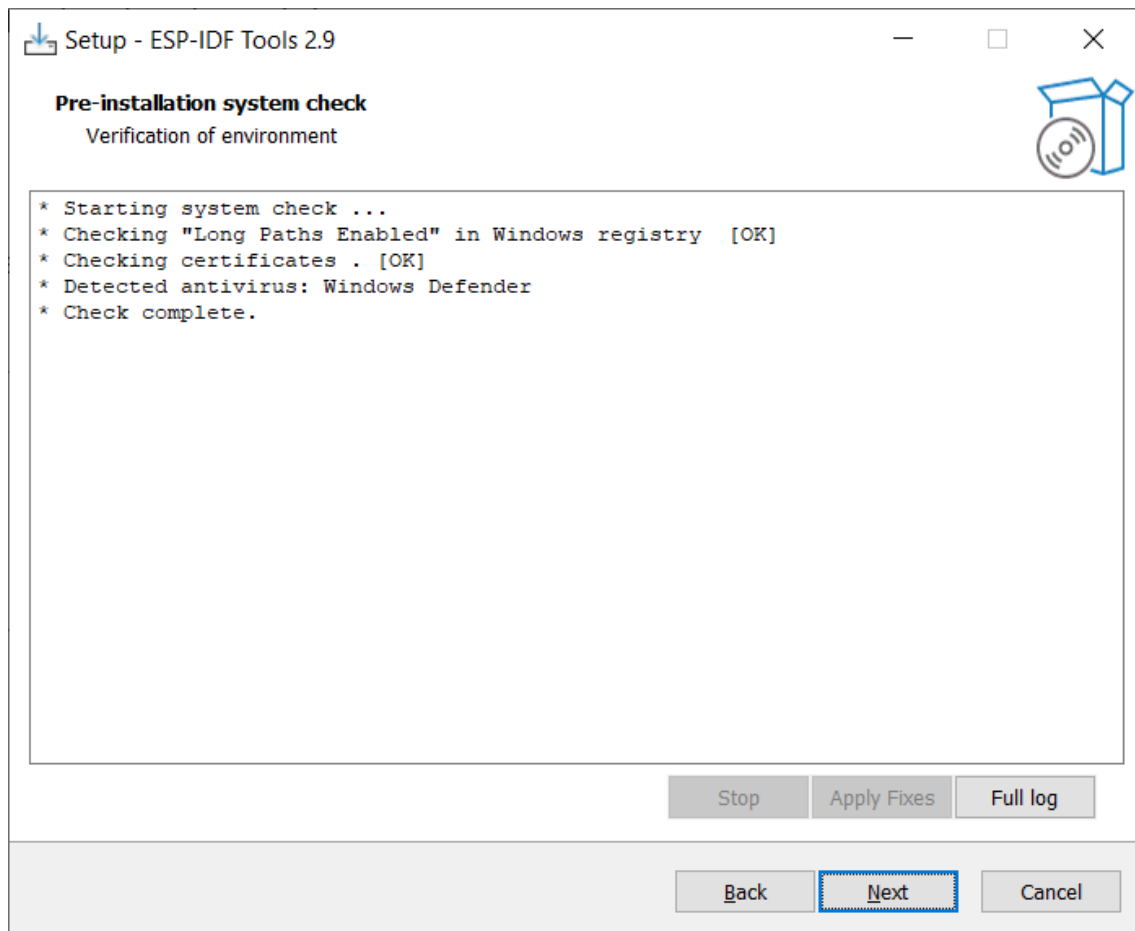
<https://dl.espressif.com/dl/esp-idf-tools-setup-online-2.9.exe>

<https://dl.espressif.com/dl/esp-idf-tools-setup-offline-2.10.exe>

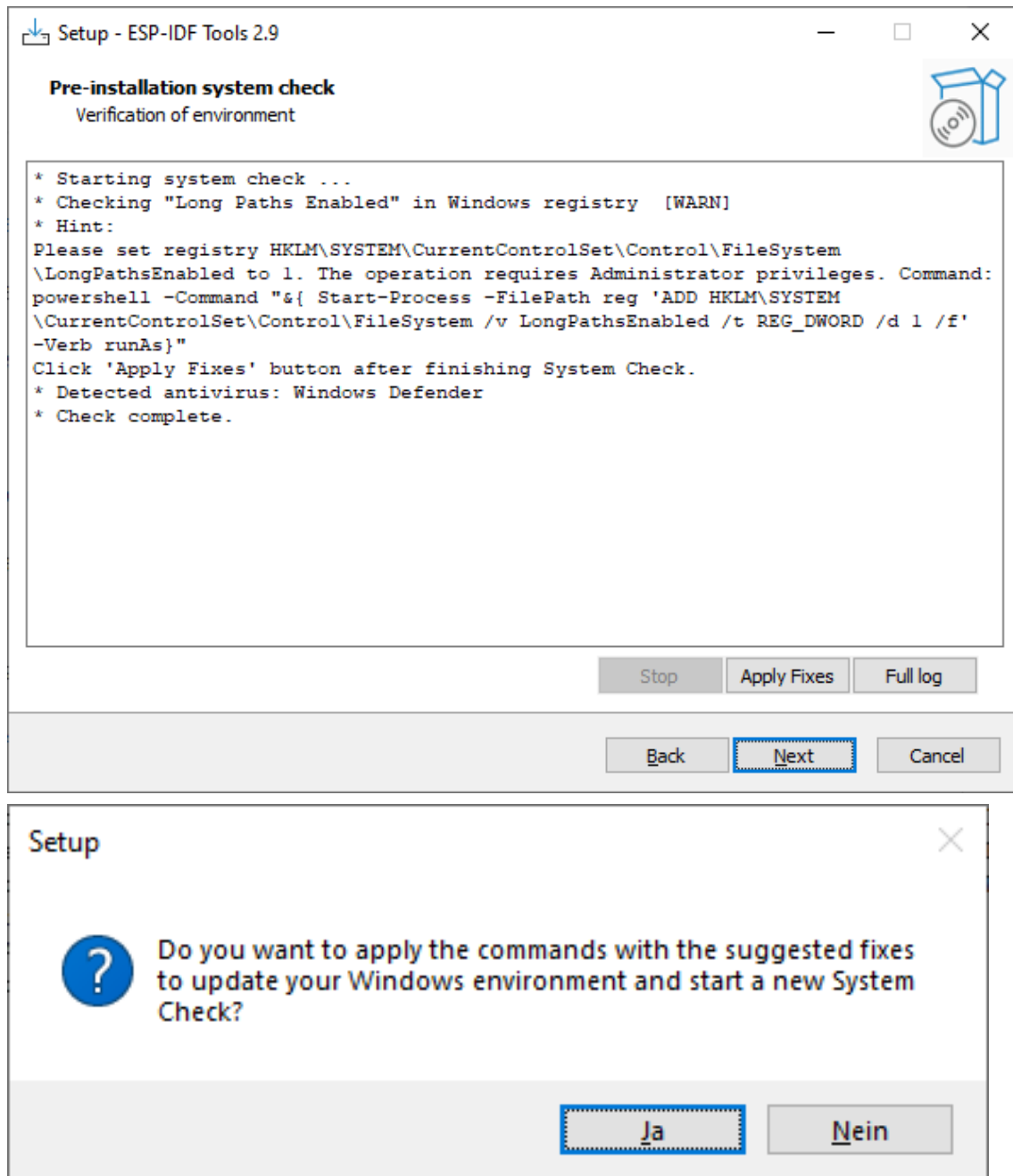
25.2 Schritt 02:



25.3 Schritt 03:



wenn das so aussieht ist es toll.
wenn nicht:

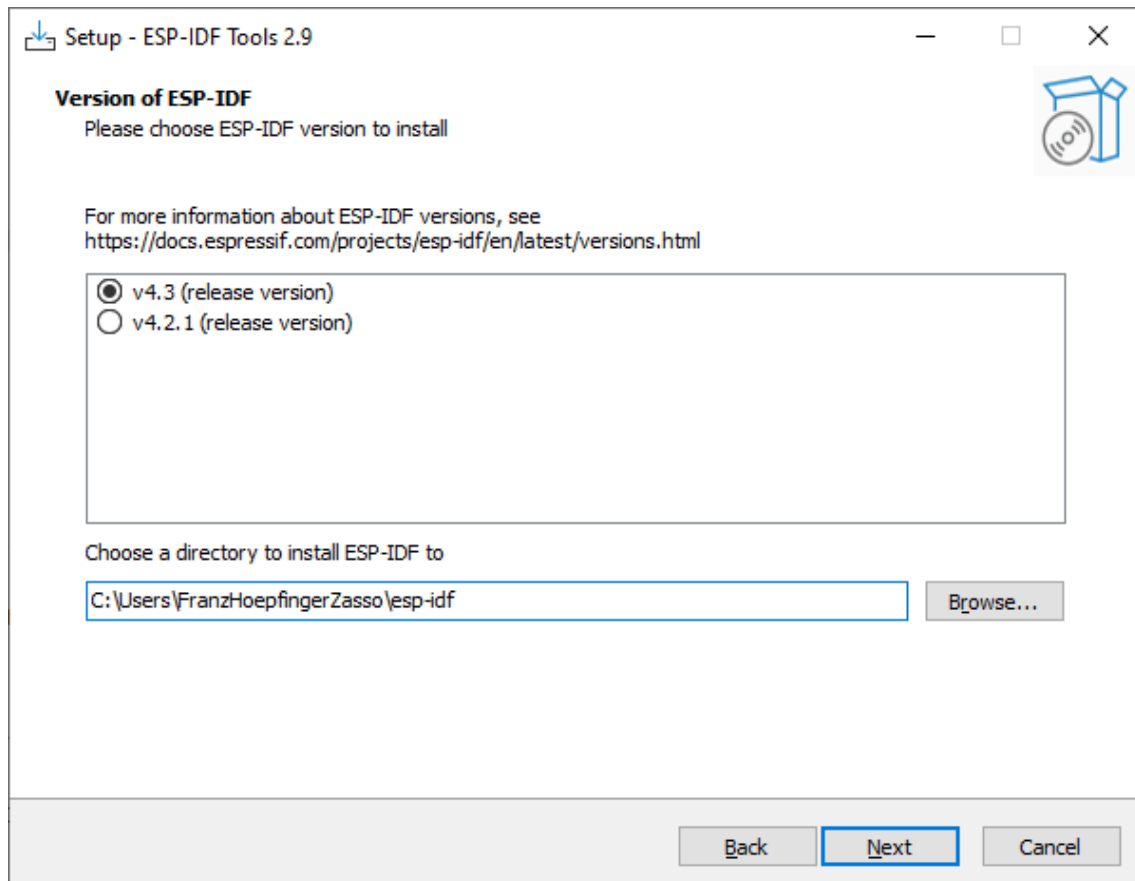


drücken Sie bitte „Apply Fixes“
und dann ja.

25.4 Schritt 04:

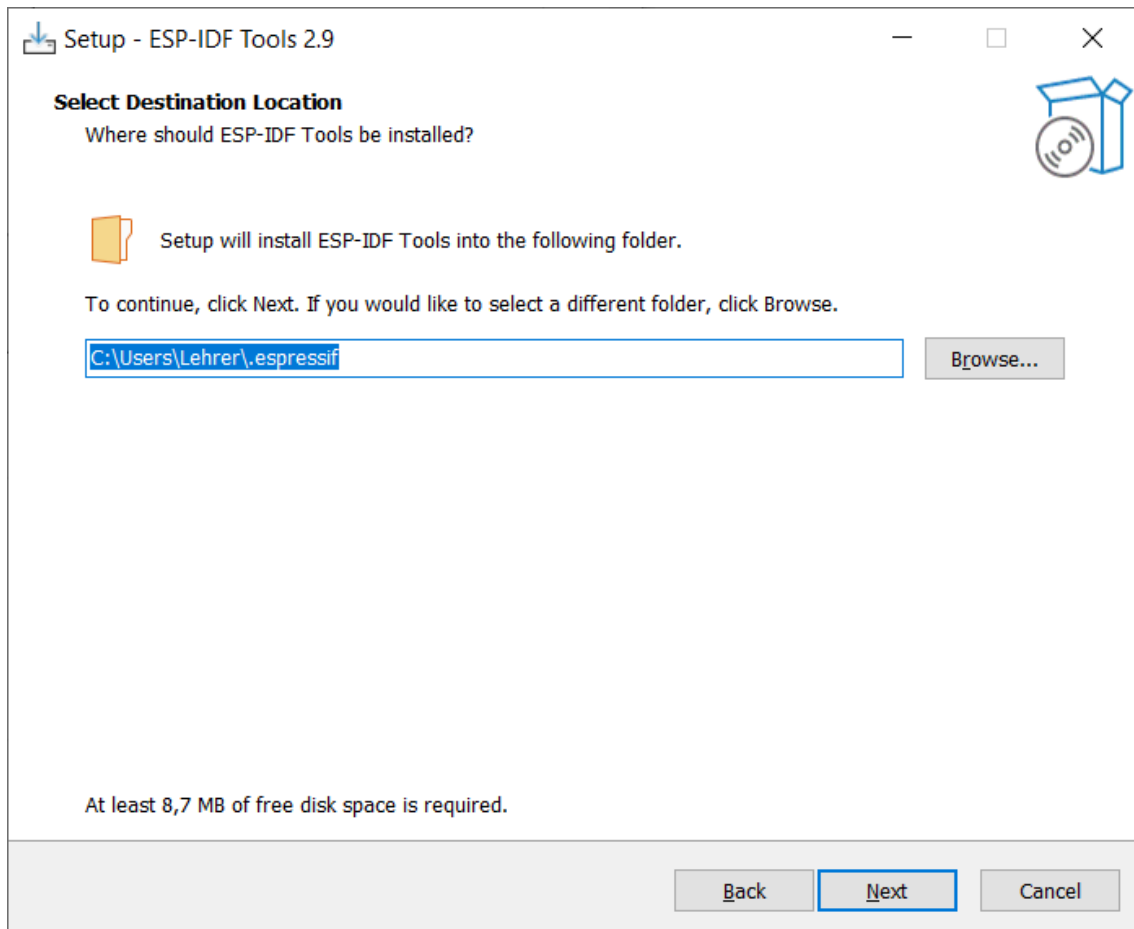
wählen Sie 4.3

hier wird ihnen der Installer „desktop“ vorschlagen, mir gefällt das nicht !

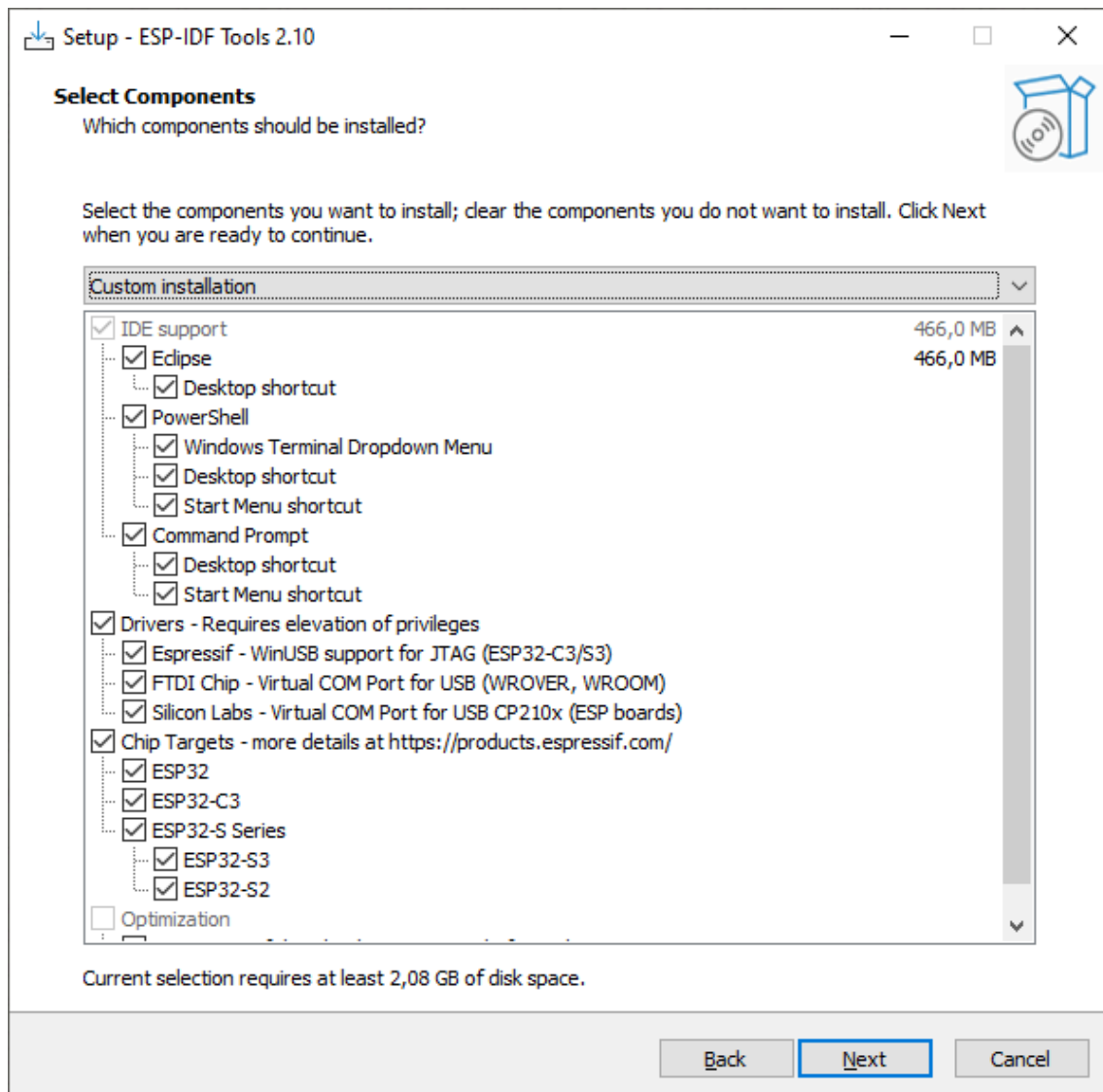


25.5 Schritt 05:

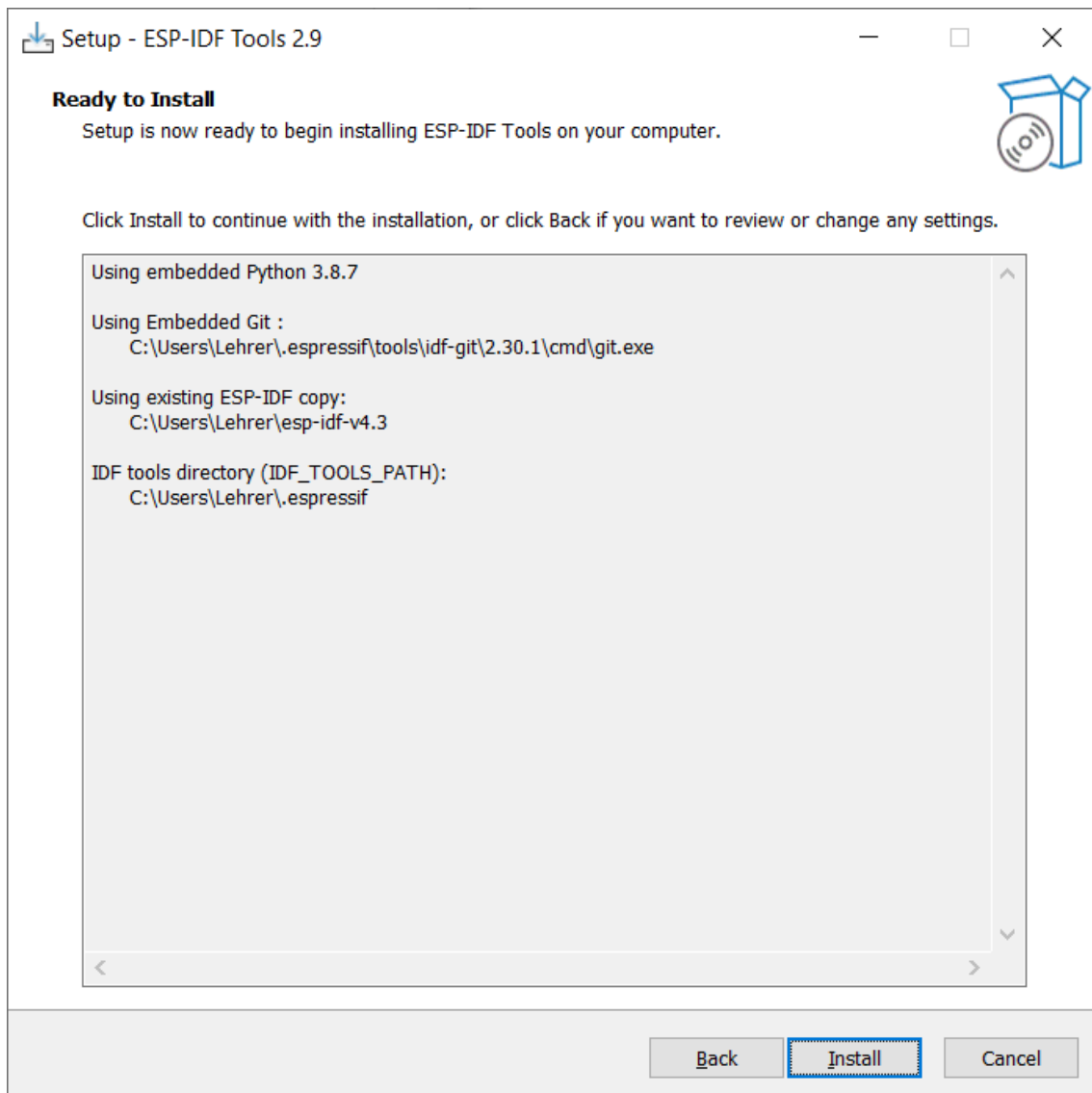
C:\Users\hoepffr\espressif
auch dieser Ordner darf keine Umlaute oder Leerzeichen haben



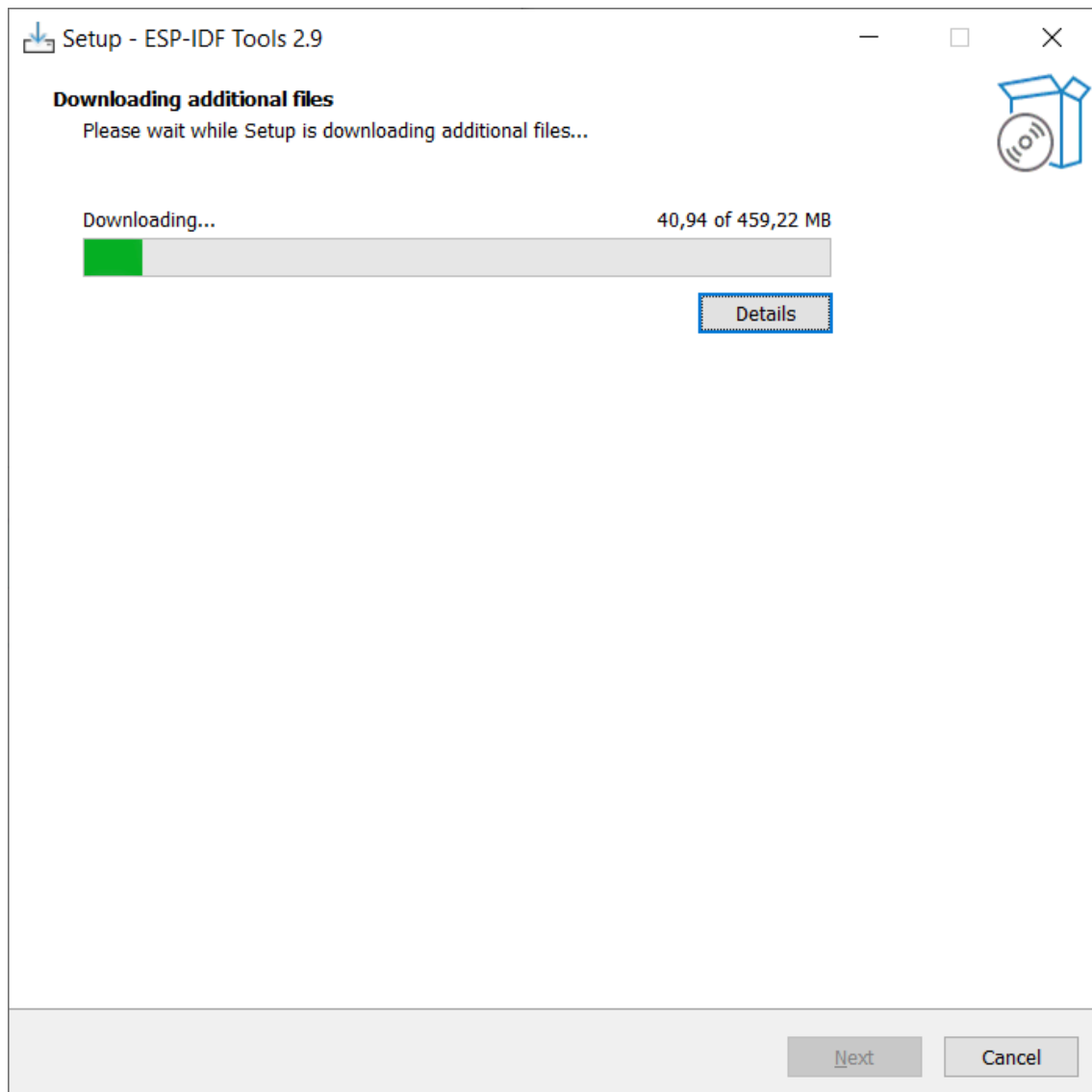
25.6 Schritt 06:



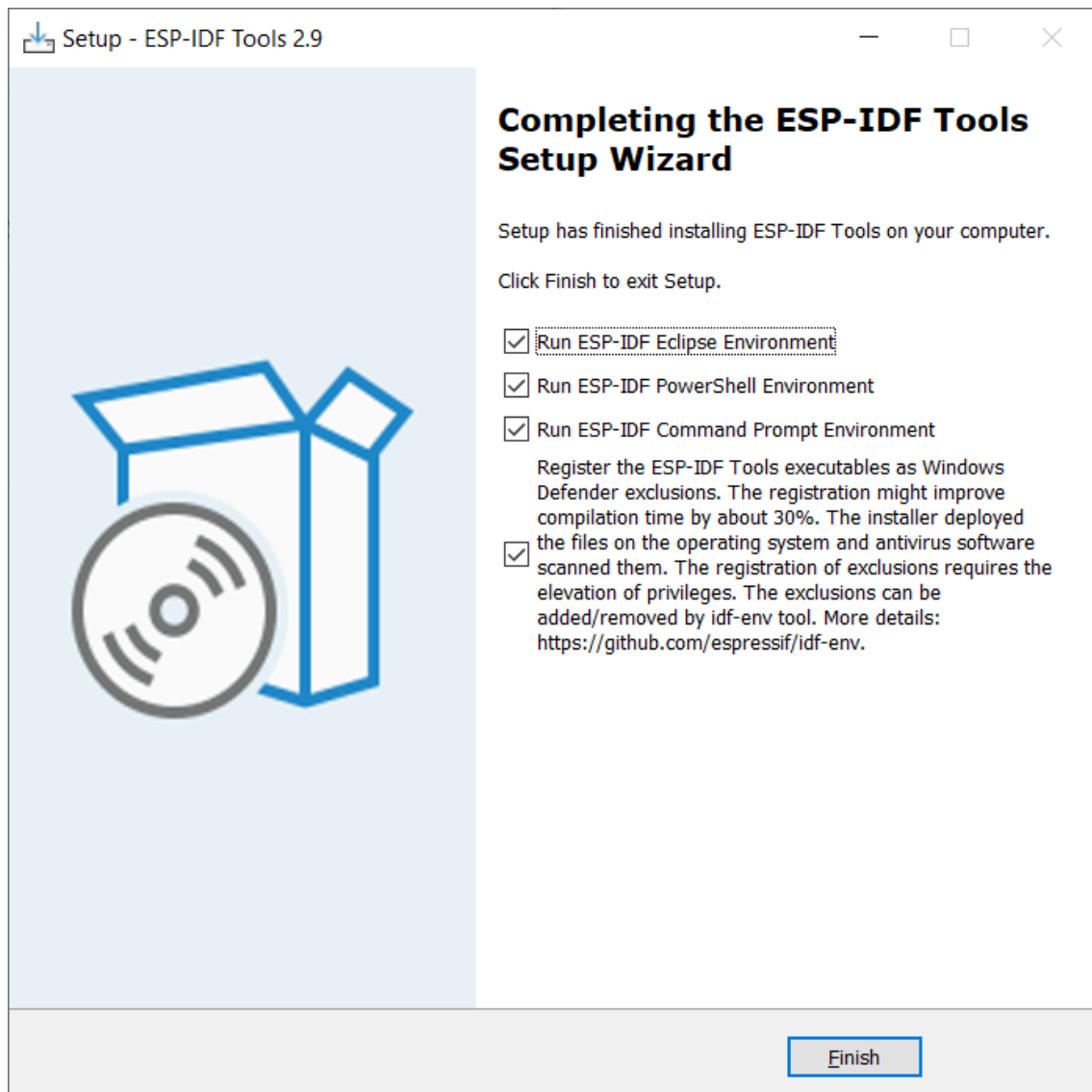
25.7 Schritt 07:



25.8 Schritt 08:



25.9 Schritt 09:



25.10 Schritt 10:

```
ESP-IDF 4.3 PowerShell
C:\Users\Lehrer\esp-idf-v4.3\components\esptool_py\esptool
C:\Users\Lehrer\esp-idf-v4.3\components\app_update
C:\Users\Lehrer\esp-idf-v4.3\components\espcoredump
C:\Users\Lehrer\esp-idf-v4.3\components\partition_table
C:\Users\Lehrer\esp-idf-v4.3\components\xtensa-esp32-elf\esp-2020r3-8.4.0\xtensa-esp32-elf\bin
C:\Users\Lehrer\esp-idf-v4.3\components\xtensa-esp32s2-elf\esp-2020r3-8.4.0\xtensa-esp32s2-elf\bin
C:\Users\Lehrer\esp-idf-v4.3\components\xtensa-esp32s3-elf\esp-2020r3-8.4.0\xtensa-esp32s3-elf\bin
C:\Users\Lehrer\esp-idf-v4.3\components\riscv32-esp-elf\1.24.0.123_64eb9ff-8.4.0\riscv32-esp-elf\bin
C:\Users\Lehrer\esp-idf-v4.3\components\esp32ulp-elf\2.28.51-esp-20191205\esp32ulp-elf-binutils\bin
C:\Users\Lehrer\esp-idf-v4.3\components\esp32s2ulp-elf\2.28.51-esp-20191205\esp32s2ulp-elf-binutils\bin
C:\Users\Lehrer\esp-idf-v4.3\components\cmake\3.16.4\bin
C:\Users\Lehrer\esp-idf-v4.3\components\openocd-esp32\v0.10.0-esp32-20210401\openocd-esp32\bin
C:\Users\Lehrer\esp-idf-v4.3\components\ninja\1.10.2\
C:\Users\Lehrer\esp-idf-v4.3\components\idf-exe\1.0.1\
C:\Users\Lehrer\esp-idf-v4.3\components\ccache\3.7\
C:\Users\Lehrer\esp-idf-v4.3\components\dfu-util\0.9\dfu-util-0.9-win64
C:\Users\Lehrer\esp-idf-v4.3\components\python_env\idf4.3_py3.8_env\Scripts
C:\Users\Lehrer\esp-idf-v4.3\tools
%PATH%
Checking if Python packages are up to date...
Python requirements from C:\Users\Lehrer\esp-idf-v4.3\requirements.txt are satisfied.

Done! You can now compile ESP-IDF projects.
Go to the project directory and run:
    idf.py build

PS C:\Users\Lehrer\esp-idf-v4.3>
```

```
ESP-IDF 4.3 CMD - "C:\Users\Lehrer\esp-idf\idf_cmd_init.bat"
Using Git in C:\Users\Lehrer\esp-idf\tools\idf-git\2.30.1\cmd
git version 2.30.1.windows.1
Setting ID_PATH: C:\Users\Lehrer\esp-idf-v4.3

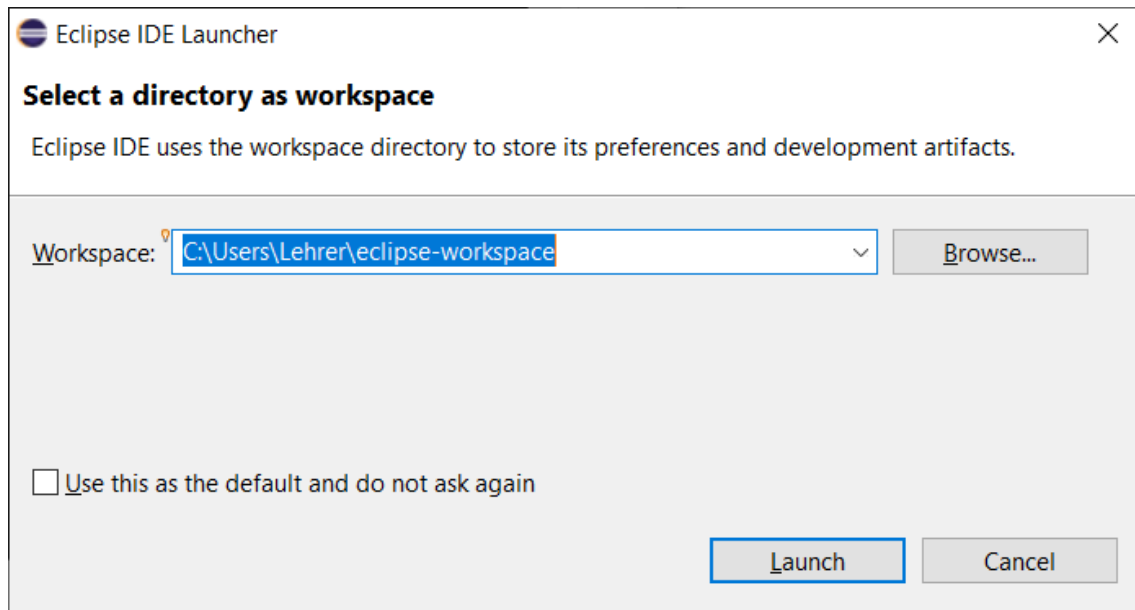
Adding ESP-IDF tools to PATH...
C:\Users\Lehrer\esp-idf-v4.3\components\xtensa-esp32-elf\esp-2020r3-8.4.0\xtensa-esp32-elf\bin
C:\Users\Lehrer\esp-idf-v4.3\components\xtensa-esp32s2-elf\esp-2020r3-8.4.0\xtensa-esp32s2-elf\bin
C:\Users\Lehrer\esp-idf-v4.3\components\xtensa-esp32s3-elf\esp-2020r3-8.4.0\xtensa-esp32s3-elf\bin
C:\Users\Lehrer\esp-idf-v4.3\components\riscv32-esp-elf\1.24.0.123_64eb9ff-8.4.0\riscv32-esp-elf\bin
C:\Users\Lehrer\esp-idf-v4.3\components\esp32ulp-elf\2.28.51-esp-20191205\esp32ulp-elf-binutils\bin
C:\Users\Lehrer\esp-idf-v4.3\components\esp32s2ulp-elf\2.28.51-esp-20191205\esp32s2ulp-elf-binutils\bin
C:\Users\Lehrer\esp-idf-v4.3\components\cmake\3.16.4\bin
C:\Users\Lehrer\esp-idf-v4.3\components\openocd-esp32\v0.10.0-esp32-20210401\openocd-esp32\bin
C:\Users\Lehrer\esp-idf-v4.3\components\ninja\1.10.2\
C:\Users\Lehrer\esp-idf-v4.3\components\idf-exe\1.0.1\
C:\Users\Lehrer\esp-idf-v4.3\components\ccache\3.7\
C:\Users\Lehrer\esp-idf-v4.3\components\dfu-util\0.9\dfu-util-0.9-win64
C:\Users\Lehrer\esp-idf-v4.3\components\python_env\idf4.3_py3.8_env\Scripts
C:\Users\Lehrer\esp-idf-v4.3\tools

Checking if Python packages are up to date...
Python requirements from C:\Users\Lehrer\esp-idf-v4.3\requirements.txt are satisfied.

Done! You can now compile ESP-IDF projects.
Go to the project directory and run:

    idf.py build

C:\Users\Lehrer\esp-idf-v4.3>
```



25.11 Schritt 11: (nur Info)

idf.py menuconfig

idf.py build

<https://docs.espressif.com/projects/esp-idf/en/latest/get-started/index.html#step-6-connect-your-device>

<https://docs.espressif.com/projects/esp-idf/en/latest/get-started/establish-serial-connection.html>

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

<https://www.ftdichip.com/Drivers/VCP.htm>

https://docs.espressif.com/projects/espressif-esp-iot-solution/en/latest/hw-reference/ESP-Prog_guide.html

idf.py flash

idf.py monitor

so, hier könnte man von der Kommandozeile aus bereits arbeiten.

```
ESP-IDF Command Prompt (cmd.exe) - "C:\Users\mail\espressif\idf_cmd_init.bat" "C:\Program Files\Python38\" "C:\Program Files\Git\cmd\"
Using Python in C:\Program Files\Python38\
Python 3.8.5
Using Git in C:\Program Files\Git\cmd\
git version 2.27.0.windows.1
Setting IDF_PATH: C:\Users\mail\esp-idf-v4.1

Adding ESP-IDF tools to PATH...
C:\Users\mail\espressif\tools\xtensa-esp32-elf\esp-2020r2-8.2.0\xtensa-esp32-elf\bin
C:\Users\mail\espressif\tools\xtensa-esp32s2-elf\esp-2020r2-8.2.0\xtensa-esp32s2-elf\bin
C:\Users\mail\espressif\tools\esp32ulp-elf\2.28.51-esp-20191205\esp32ulp-elf-binutils\bin
C:\Users\mail\espressif\tools\esp32s2ulp-elf\2.28.51-esp-20191205\esp32s2ulp-elf-binutils\bin
C:\Users\mail\espressif\tools\cmake\3.13.4\bin
C:\Users\mail\espressif\tools\openocd-esp32\v0.10.0-esp32-20191114\openocd-esp32\bin
C:\Users\mail\espressif\tools\ninja\1.9.0\
C:\Users\mail\espressif\tools\idf-exe\1.0.1\
C:\Users\mail\espressif\tools\ccache\3.7\
C:\Users\mail\espressif\python_env\idf4.1_py3.8_env\Scripts
C:\Users\mail\esp-idf-v4.1\tools

Checking if Python packages are up to date...
Python requirements from C:\Users\mail\esp-idf-v4.1\requirements.txt are satisfied.

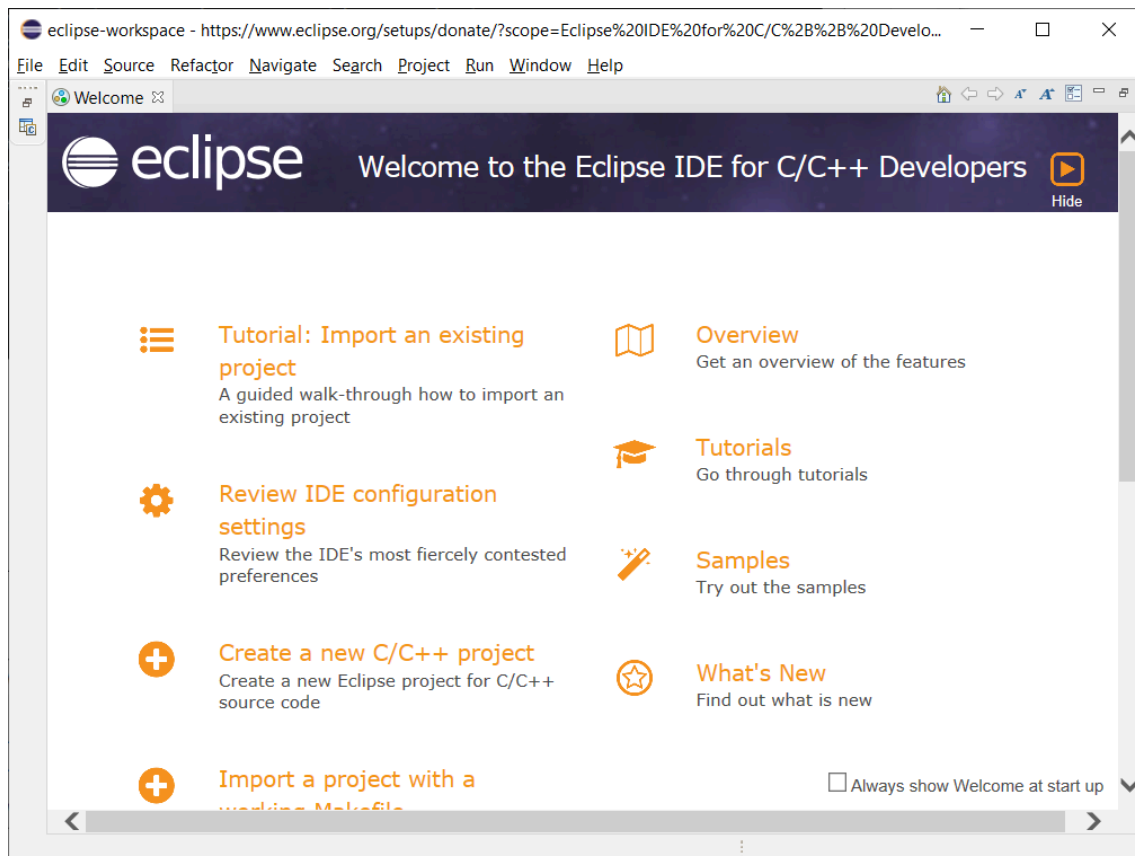
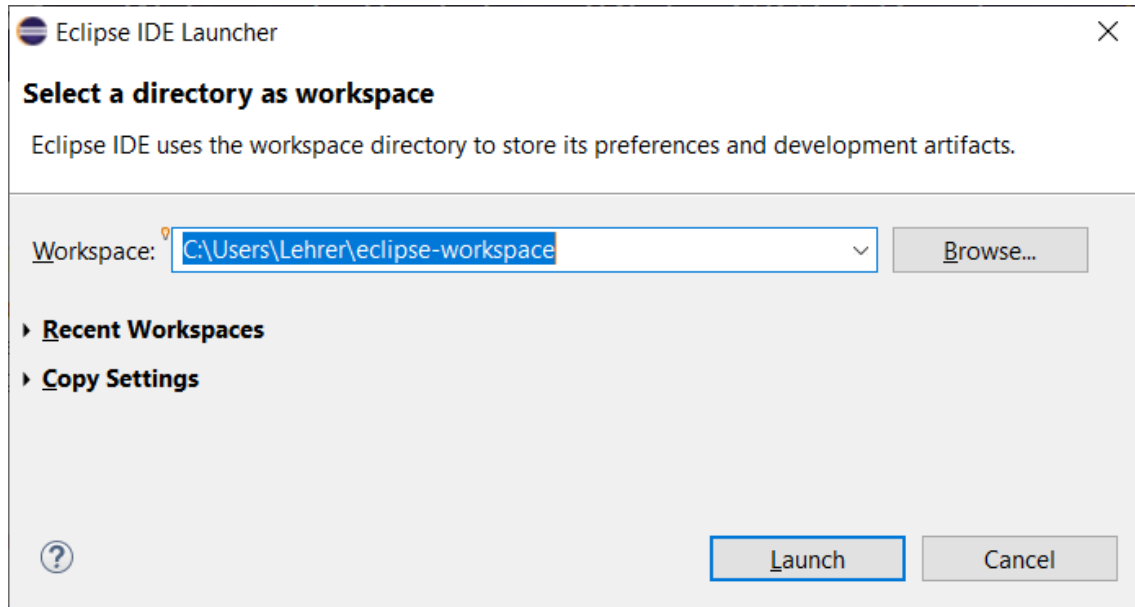
Done! You can now compile ESP-IDF projects.
Go to the project directory and run:

idf.py build

C:\Users\mail\esp-idf-v4.1>
```

25.12 Schritt 12:

Öffnen Sie nun Eclipse und erstellen Sie sich einen neuen Workspace.



erledigt.

man braucht kein JAVA, kein Python und kein Git vorab installieren.

26. blink_on_atom

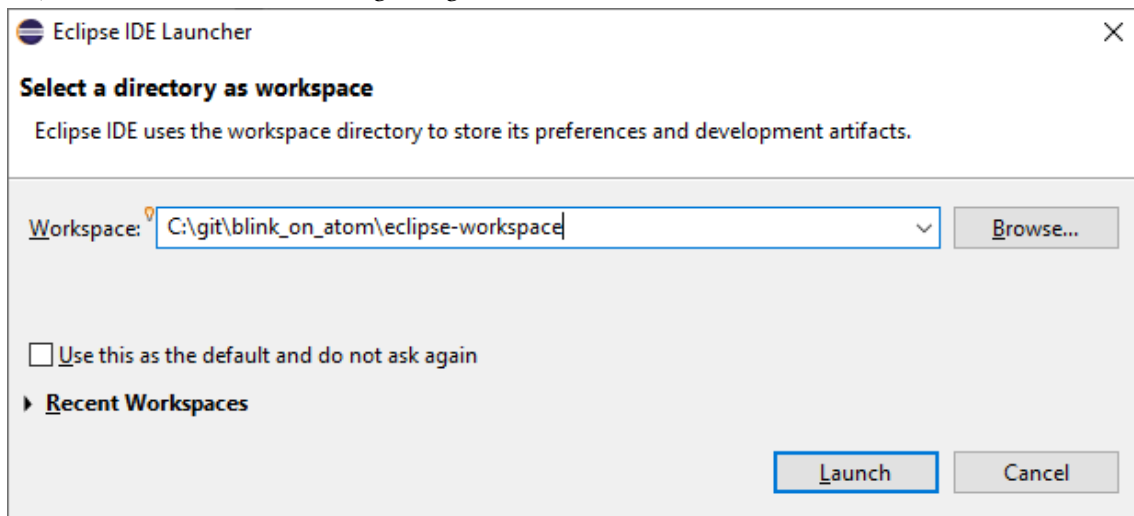
26.0.1 Schritt_14:

Jetzt kann man Eclipse wirklich starten:

Nehmen Sie für das erste Projekt diesen Workspace:

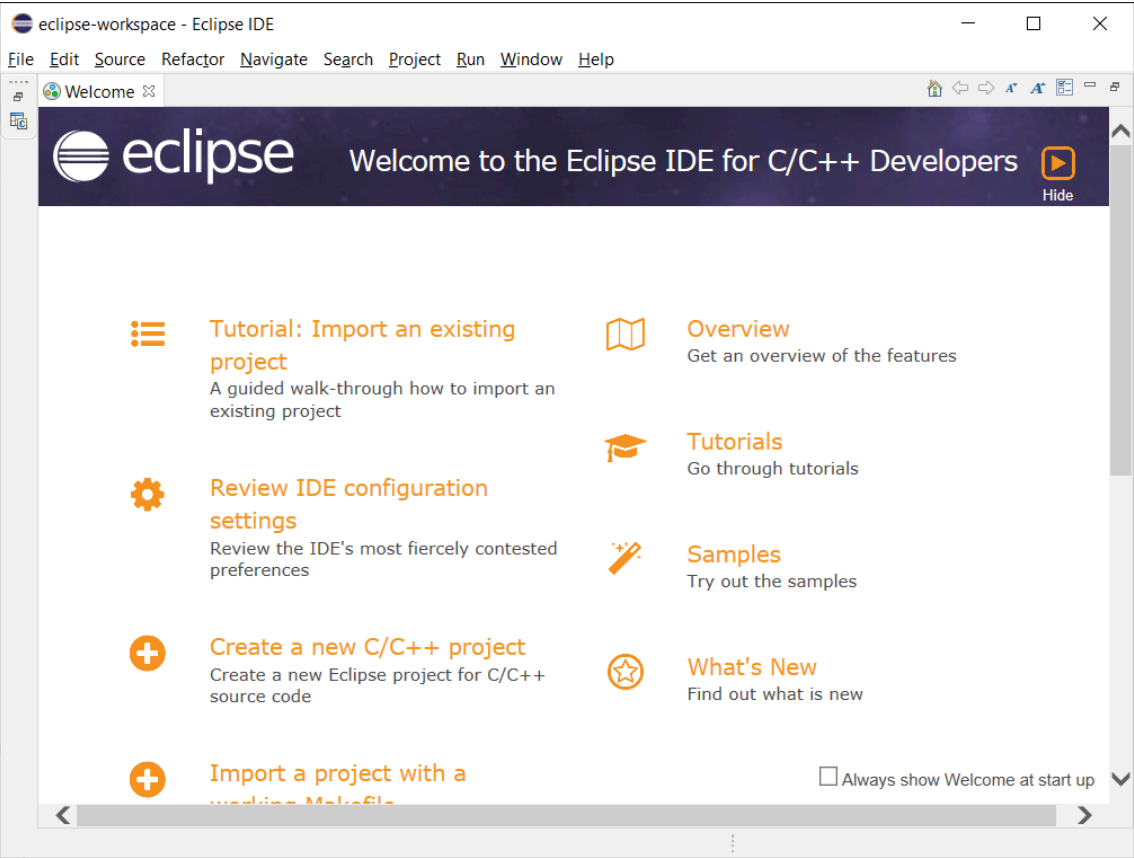
C:\git\blink_on_atom\eclipse-workspace

(außer sie haben vorhin auf D:\git ausgecheckt>`_

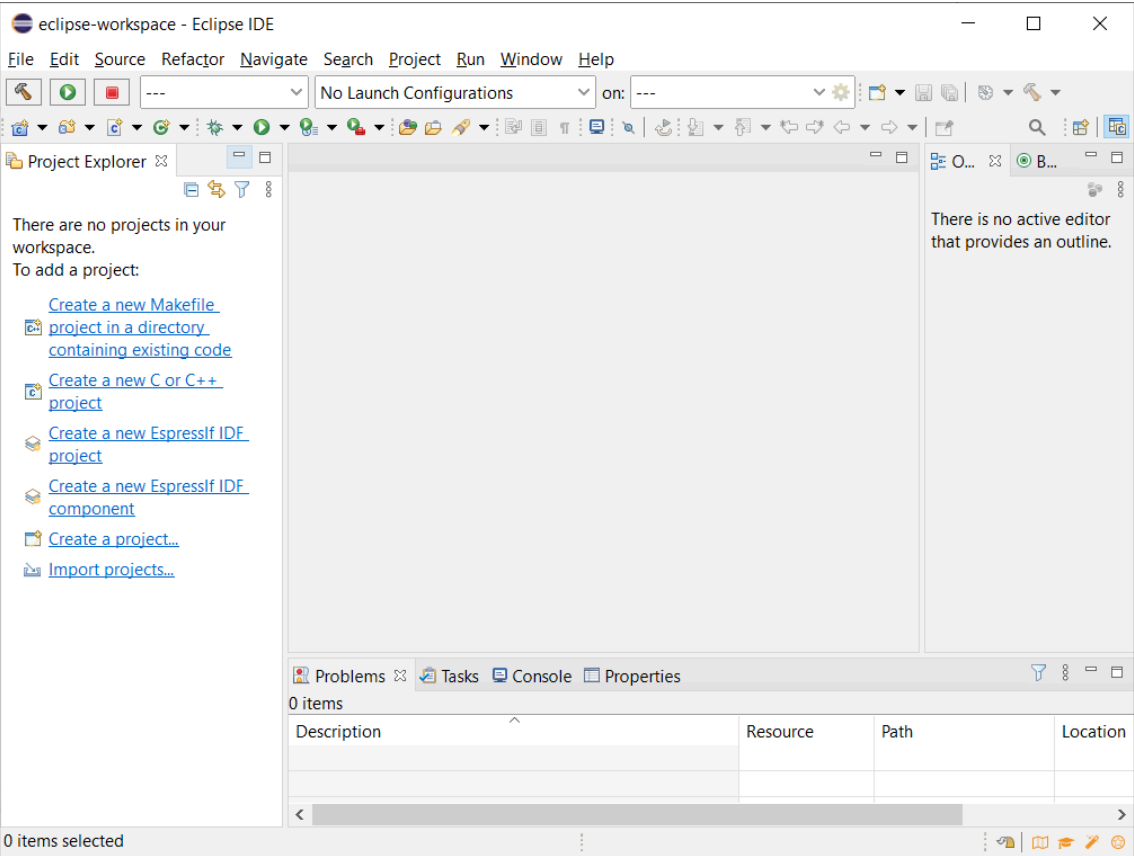


26.0.2 Schritt_15:

Workbench -> Hide (rechts oben>`_



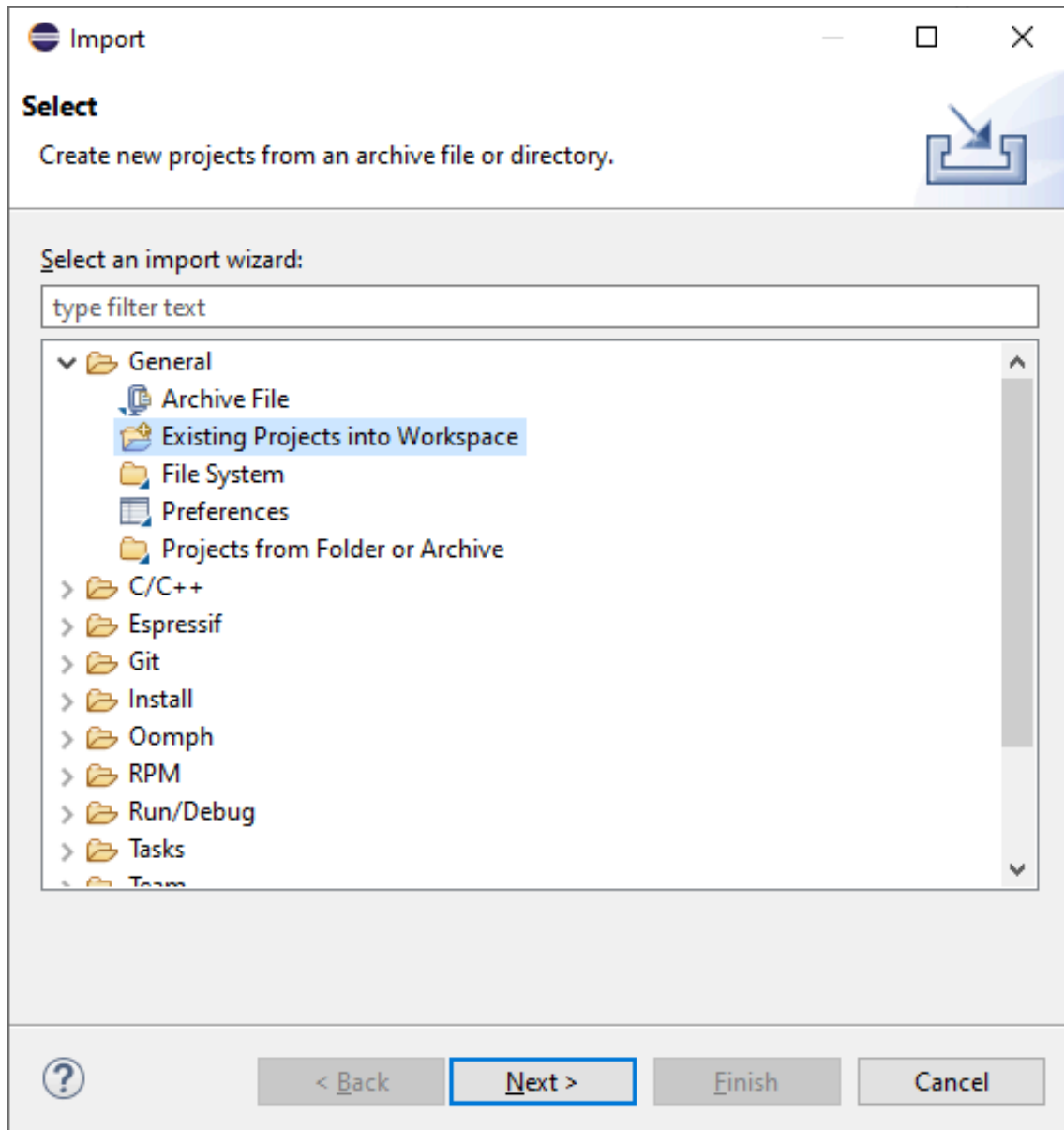
26.0.3 Schritt_16:

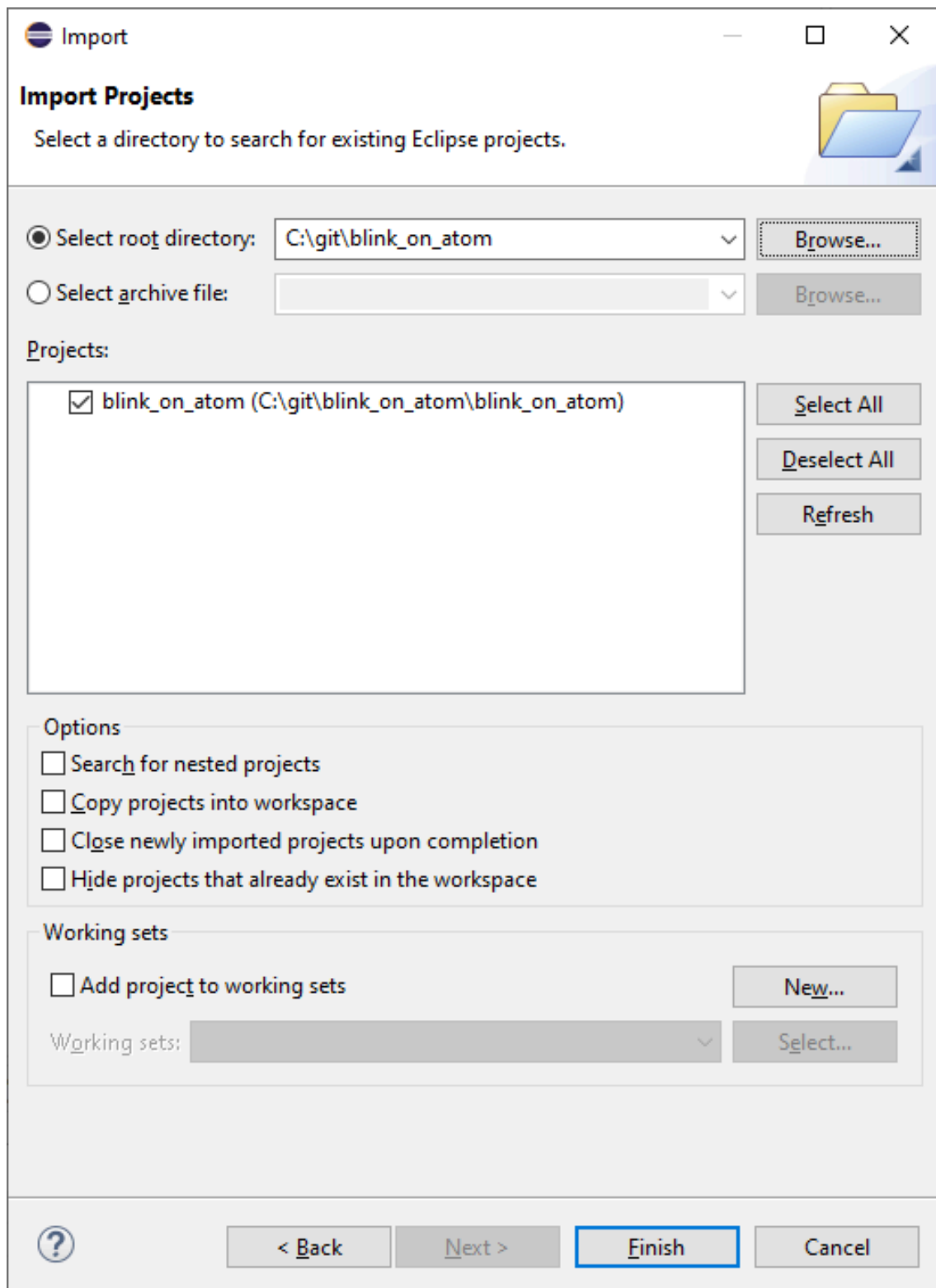


26.0.4 Schritt_17:

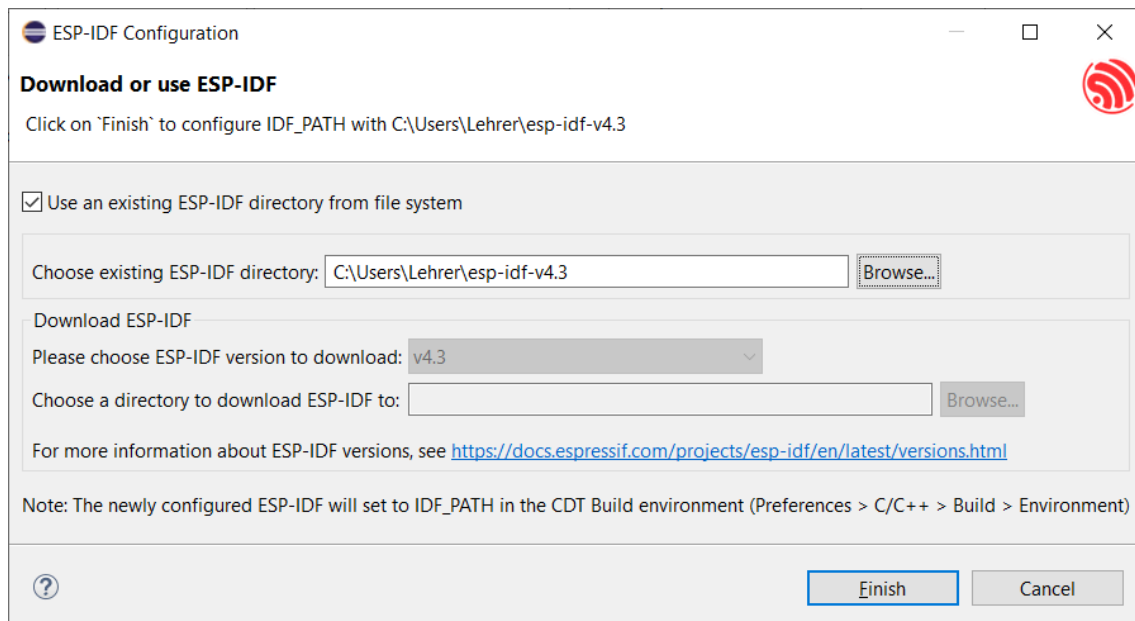
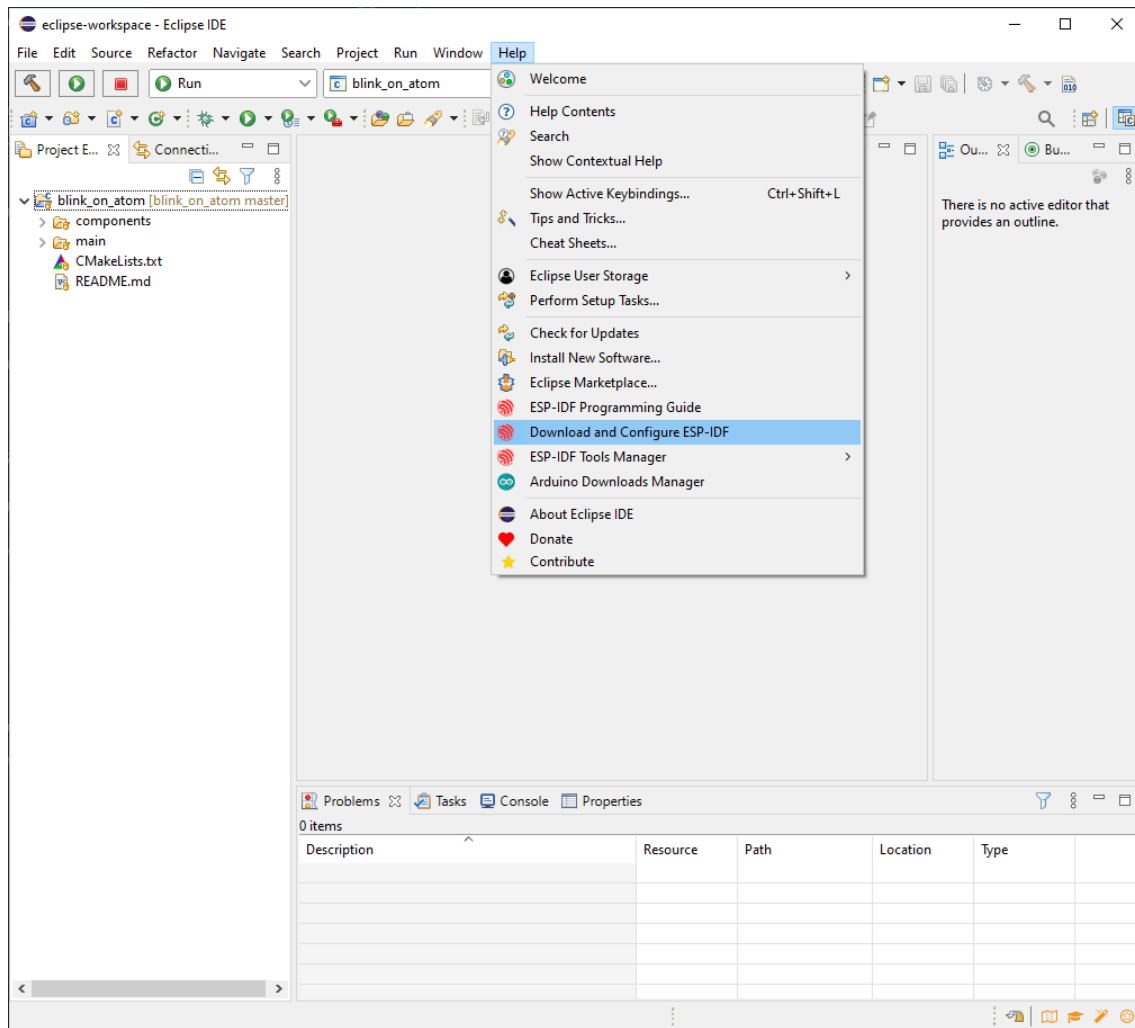
gehen Sie jetzt auf Import Projects

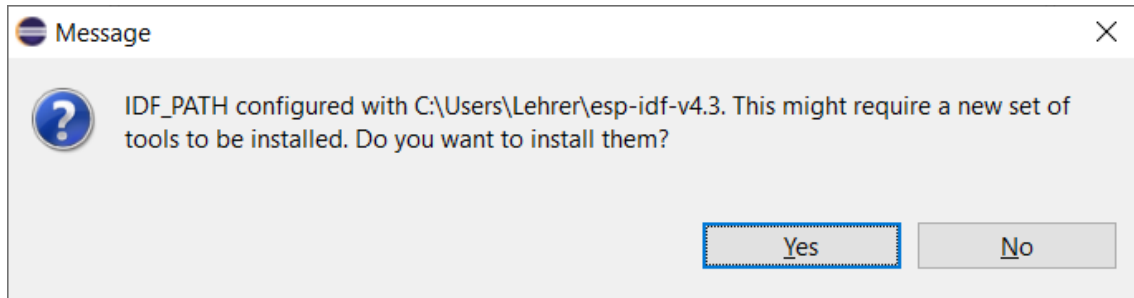
(Diesen Vorgang machen Sie später Analog für andere Projekte !!! Prägen Sie sich das ein !!!>`_





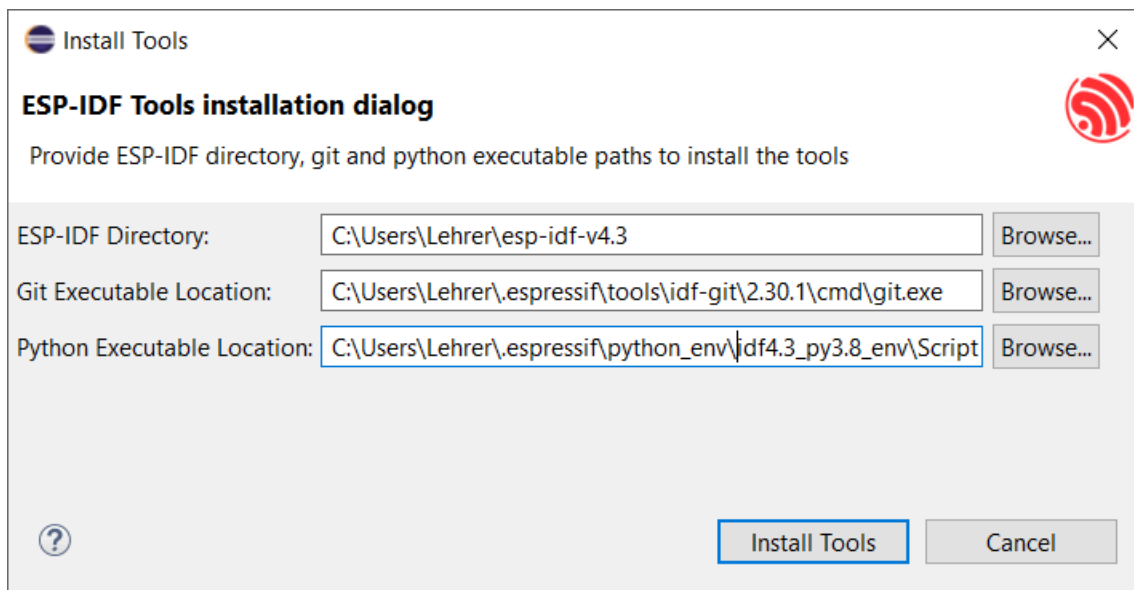
IMMER wenn man einen neuen Workspace aufmacht: (den man an diesem PC noch nie geöffnet hatte>`\_`
IMMER siehe: <<https://github.com/espressif/idf-eclipse-plugin/blob/master/README.md#installing-esp-idf-tools>





YES

26.0.5 Schritt_18:



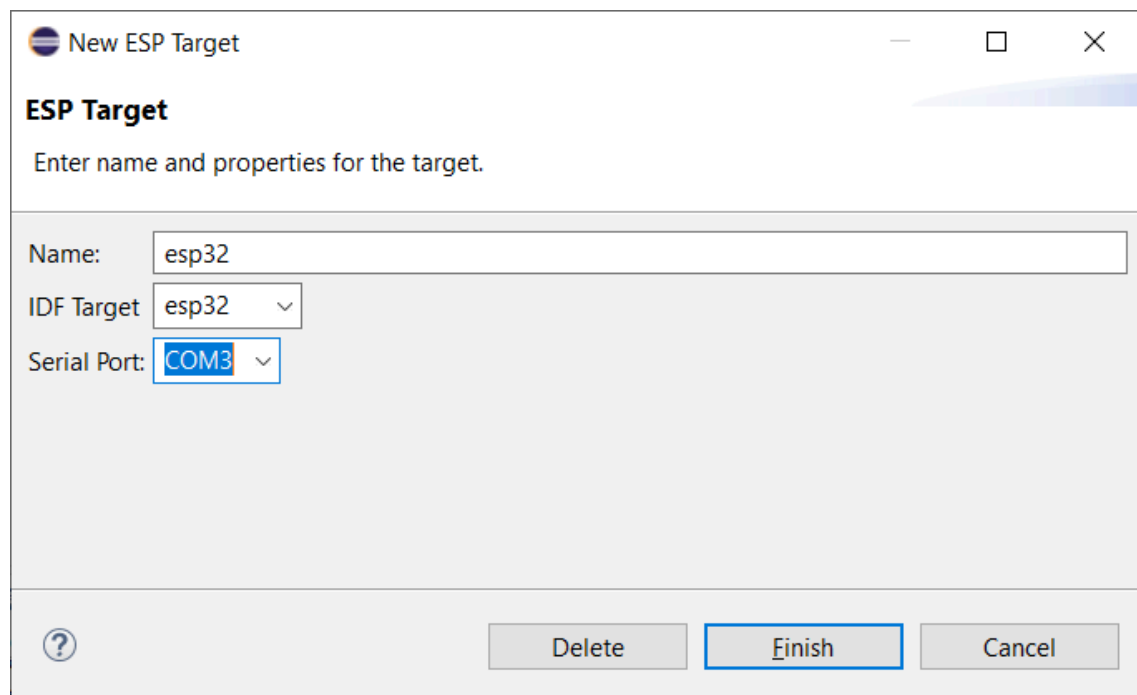
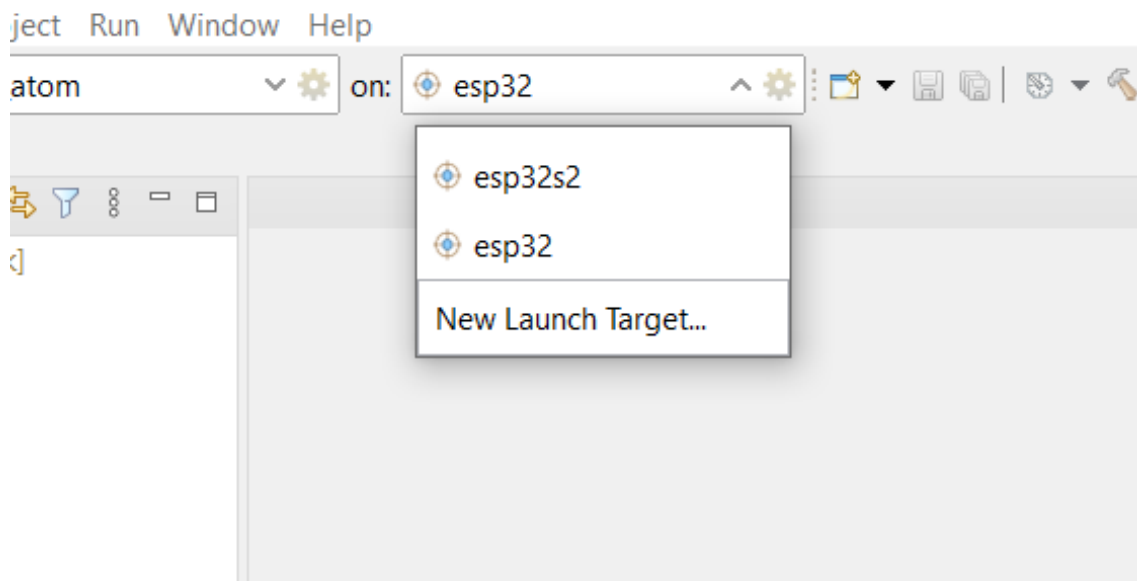
C:\Users\Lehrer\esp-idf-v4.3

C:\Users\Lehrer\espressif\tools\idf-git\2.30.1\cmd\git.exe

C:\Users\Lehrer\espressif\python_env\idf4.3_py3.8_env\Scripts\python.exe

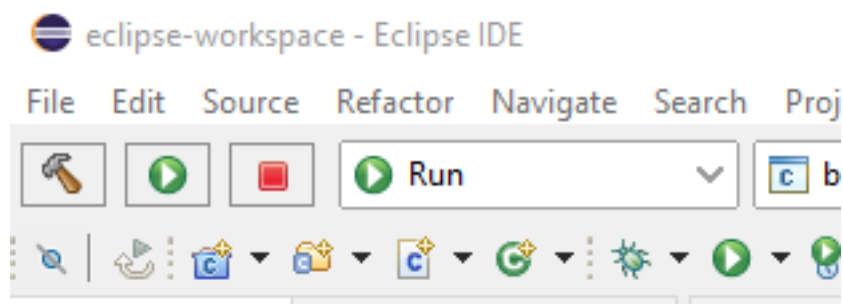
26.0.6 Schritt_20:

<<https://github.com/espressif/idf-eclipse-plugin/blob/master/README.md#configuring-launch-target>

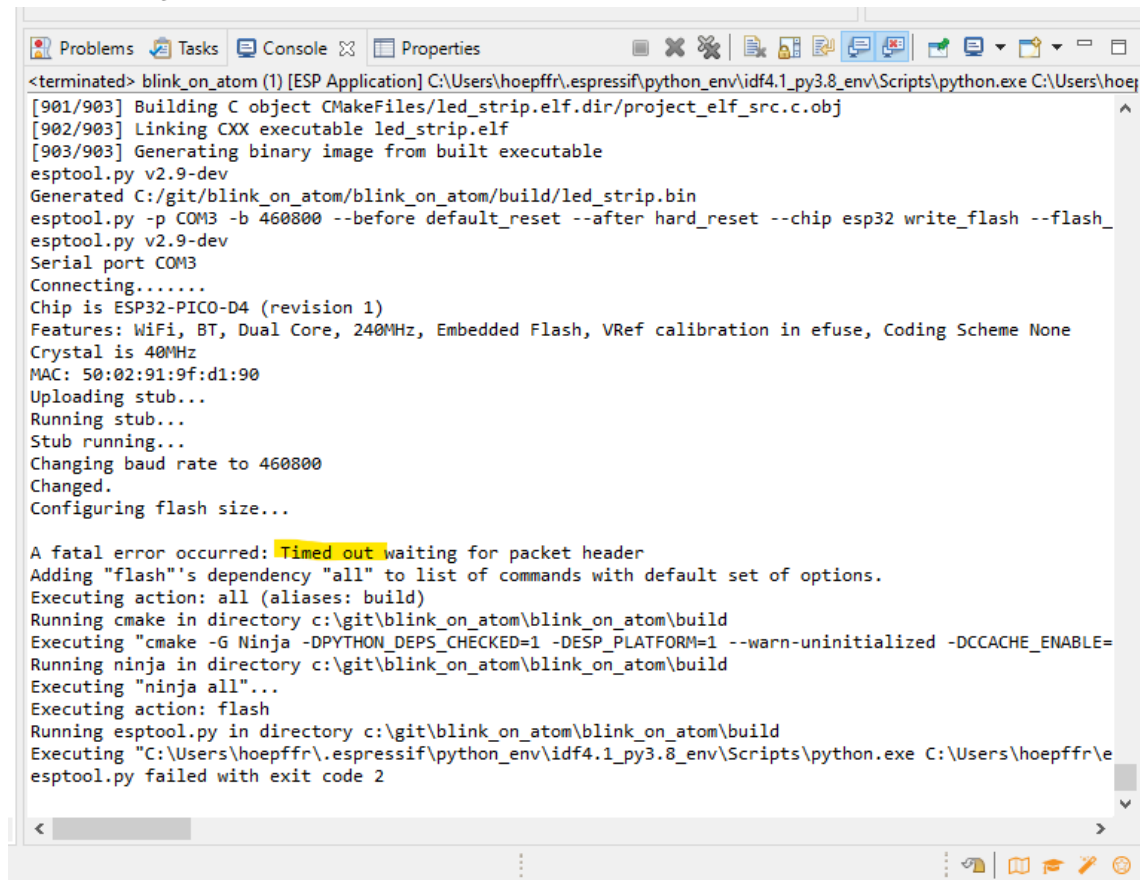


26.0.7 Schritt_21

Drücken Sie dann erst den Hammer,
dann den „Play“ Button.



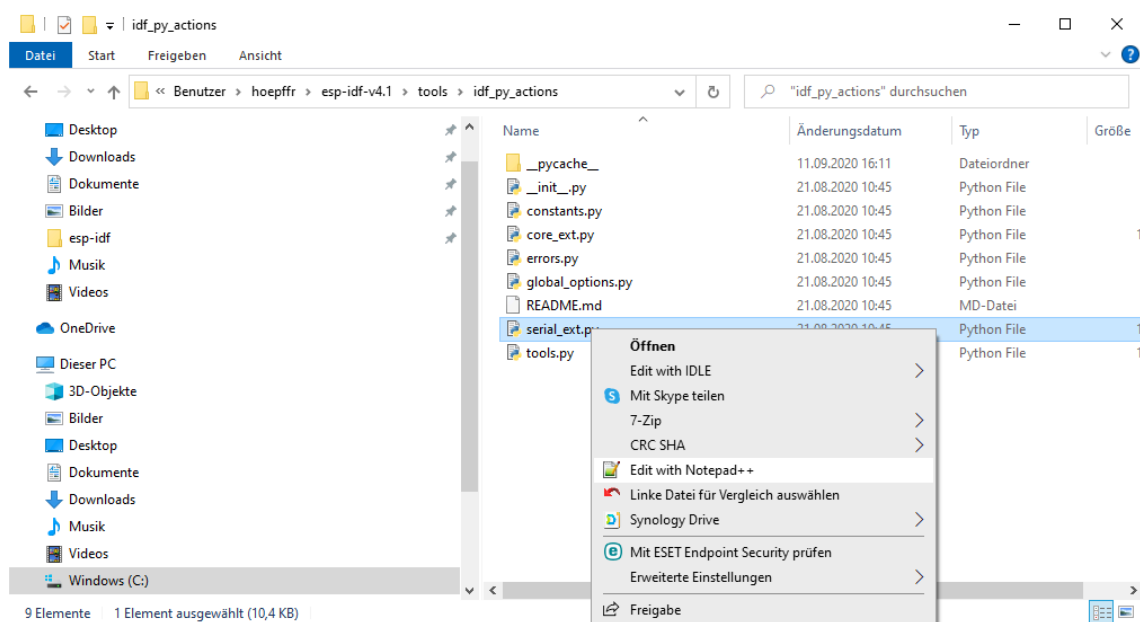
wenn jetzt folgendes erscheint:



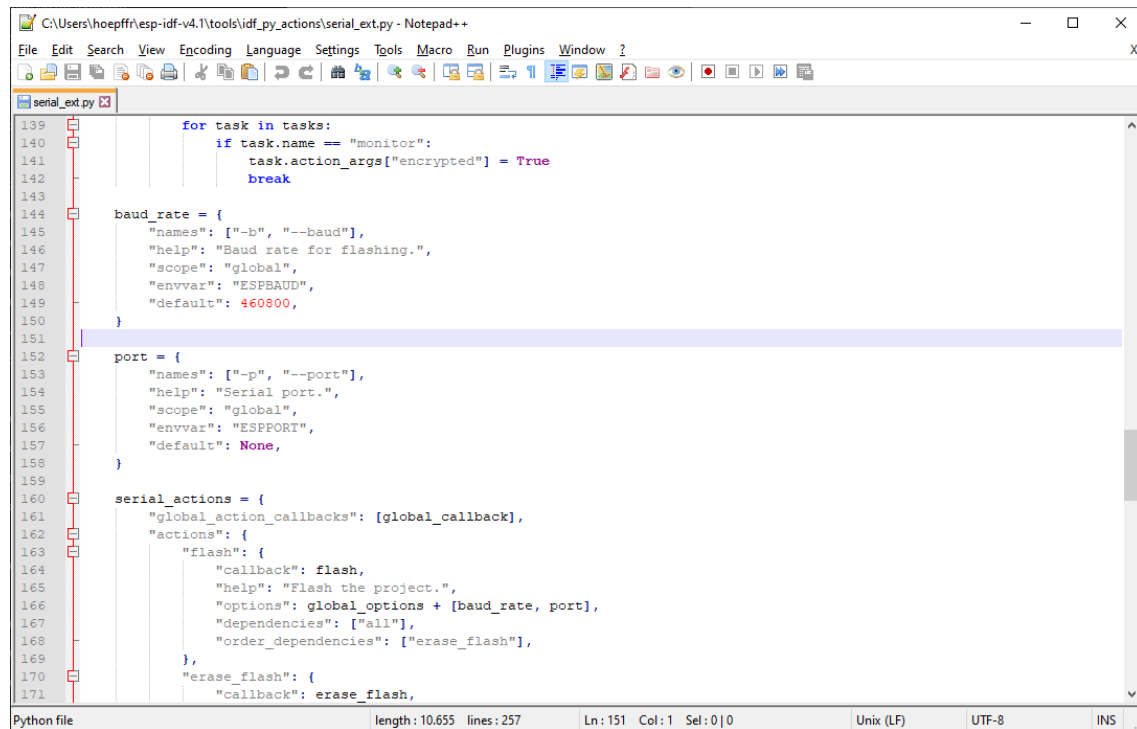
```
<terminated> blink_on_atom (1) [ESP Application] C:\Users\hoepffr\.espressif\python_env\idf4.1_py3.8_env\Scripts\python.exe C:\Users\hoepffr\
[901/903] Building C object CMakeFiles/led_strip.elf.dir/project_elf_src.c.obj
[902/903] Linking CXX executable led_strip.elf
[903/903] Generating binary image from built executable
esptool.py v2.9-dev
Generated C:/git/blink_on_atom/blink_on_atom/build/led_strip.bin
esptool.py -p COM3 -b 460800 --before default_reset --after hard_reset --chip esp32 write_flash --flash_
esptool.py v2.9-dev
Serial port COM3
Connecting.....
Chip is ESP32-PICO-D4 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, Embedded Flash, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: 50:02:91:9f:d1:90
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...

A fatal error occurred: Timed out waiting for packet header
Adding "flash"'s dependency "all" to list of commands with default set of options.
Executing action: all (aliases: build)
Running cmake in directory c:\git\blink_on_atom\blink_on_atom\build
Executing "cmake -G Ninja -DPYTHON_DEPS_CHECKED=1 -DESP_PLATFORM=1 --warn-uninitialized -DCCACHE_ENABLE=
Running ninja in directory c:\git\blink_on_atom\blink_on_atom\build
Executing "ninja all"...
Executing action: flash
Running esptool.py in directory c:\git\blink_on_atom\blink_on_atom\build
Executing "C:\Users\hoepffr\.espressif\python_env\idf4.1_py3.8_env\Scripts\python.exe C:\Users\hoepffr\
esptool.py failed with exit code 2
```

dann öffnen Sie die Datei



C:\Users\hoepffr\esp-idf-v4.1\tools\idf_py_actions\serial_ext.py



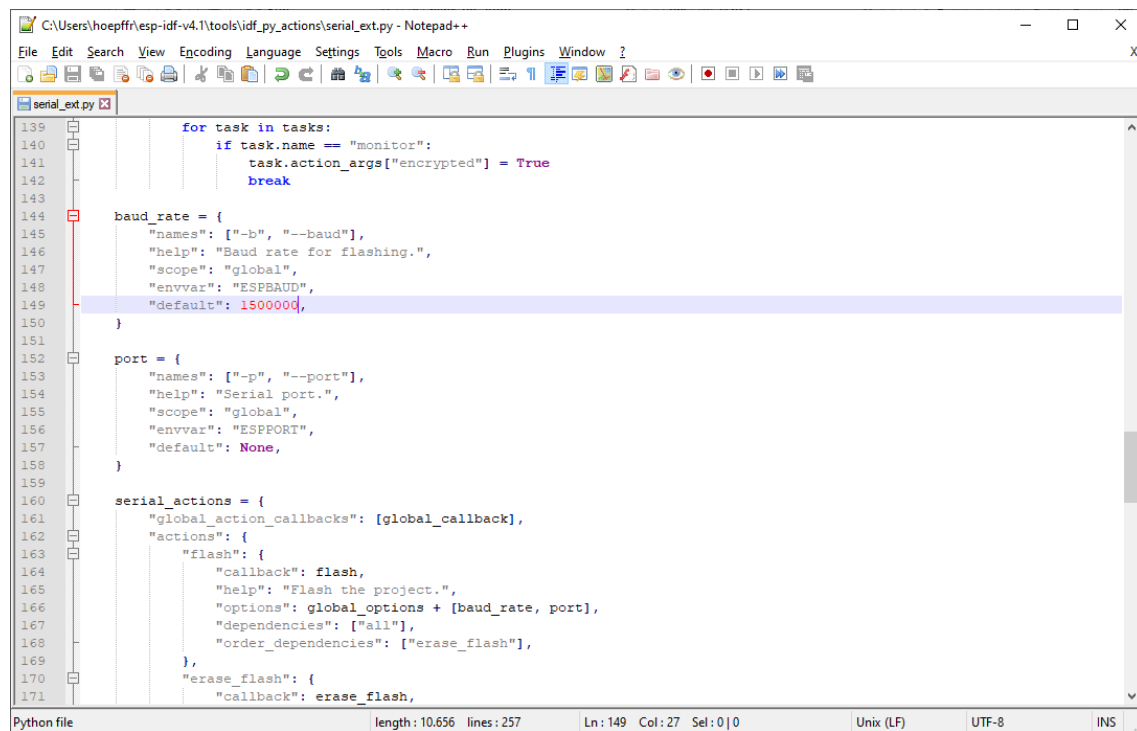
```

139
140     for task in tasks:
141         if task.name == "monitor":
142             task.action_args["encrypted"] = True
143             break
144
145     baud_rate = {
146         "names": ["-b", "--baud"],
147         "help": "Baud rate for flashing.",
148         "scope": "global",
149         "envvar": "ESPBAUD",
150         "default": 460800,
151     }
152
153     port = {
154         "names": ["-p", "--port"],
155         "help": "Serial port.",
156         "scope": "global",
157         "envvar": "ESPSPORT",
158         "default": None,
159     }
160
161     serial_actions = {
162         "global_action_callbacks": [global_callback],
163         "actions": {
164             "flash": {
165                 "callback": flash,
166                 "help": "Flash the project.",
167                 "options": global_options + [baud_rate, port],
168                 "dependencies": ["all"],
169                 "order_dependencies": ["erase_flash"],
170             },
171             "erase_flash": {
172                 "callback": erase_flash,

```

Python file length: 10.655 lines: 257 Ln: 151 Col: 1 Sel: 0|0 Unix (LF) UTF-8 INS

und ändern die Zahl 460800 ab in 1500000



```

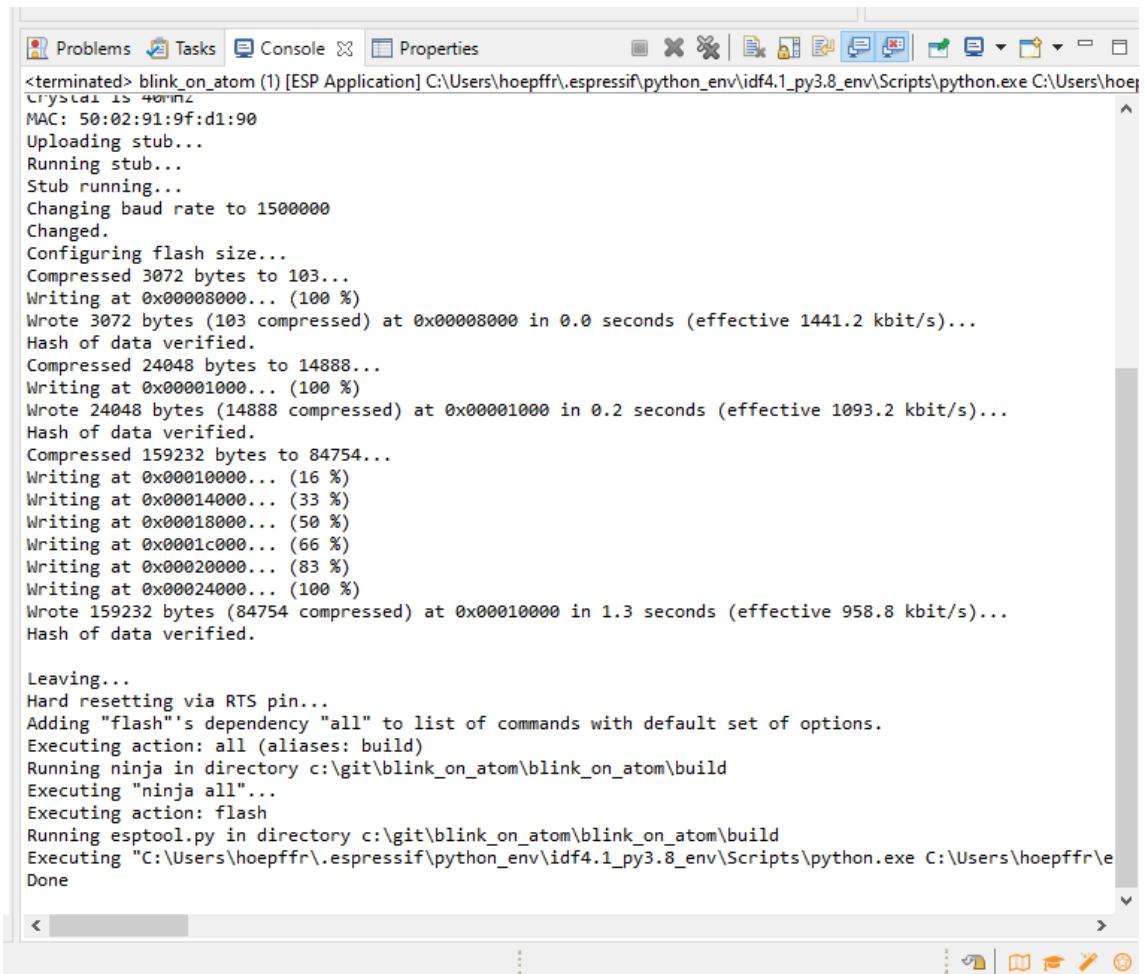
139
140     for task in tasks:
141         if task.name == "monitor":
142             task.action_args["encrypted"] = True
143             break
144
145     baud_rate = {
146         "names": ["-b", "--baud"],
147         "help": "Baud rate for flashing.",
148         "scope": "global",
149         "envvar": "ESPBAUD",
150         "default": 1500000,
151     }
152
153     port = {
154         "names": ["-p", "--port"],
155         "help": "Serial port.",
156         "scope": "global",
157         "envvar": "ESPSPORT",
158         "default": None,
159     }
160
161     serial_actions = {
162         "global_action_callbacks": [global_callback],
163         "actions": {
164             "flash": {
165                 "callback": flash,
166                 "help": "Flash the project.",
167                 "options": global_options + [baud_rate, port],
168                 "dependencies": ["all"],
169                 "order_dependencies": ["erase_flash"],
170             },
171             "erase_flash": {
172                 "callback": erase_flash,

```

Python file length: 10.656 lines: 257 Ln: 149 Col: 27 Sel: 0|0 Unix (LF) UTF-8 INS

SPEICHERN nicht vergessen !

jetzt nochmal den PLAY Button drücken:



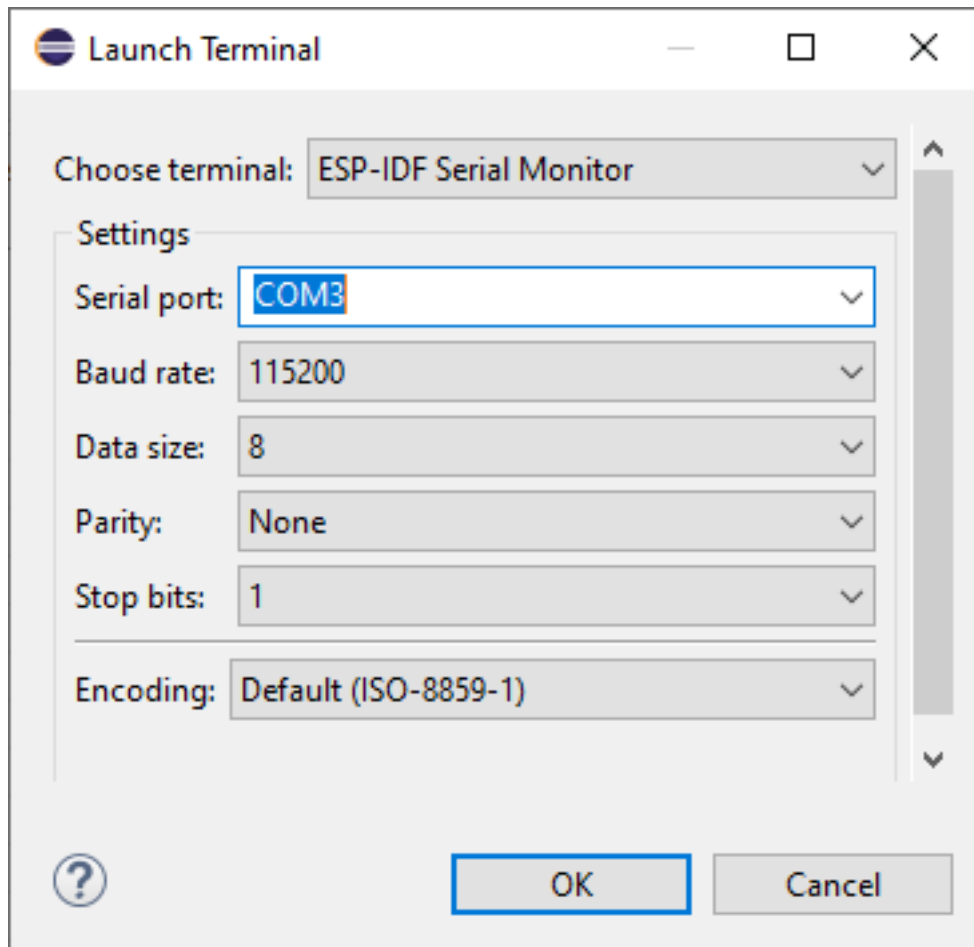
```
<terminated> blink_on_atom (1) [ESP Application] C:\Users\hoepffr\python_env\idf4.1_py3.8_env\Scripts\python.exe C:\Users\hoepffr\python_env\idf4.1_py3.8_env\Scripts\python.exe C:\Users\hoepffr\python_env\idf4.1_py3.8_env\Scripts\python.exe
Crystal is 40MHz
MAC: 50:02:91:9f:d1:90
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 1500000
Changed.
Configuring flash size...
Compressed 3072 bytes to 103...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (103 compressed) at 0x00008000 in 0.0 seconds (effective 1441.2 kbit/s)...
Hash of data verified.
Compressed 24048 bytes to 14888...
Writing at 0x00010000... (100 %)
Wrote 24048 bytes (14888 compressed) at 0x00010000 in 0.2 seconds (effective 1093.2 kbit/s)...
Hash of data verified.
Compressed 159232 bytes to 84754...
Writing at 0x00010000... (16 %)
Writing at 0x00014000... (33 %)
Writing at 0x00018000... (50 %)
Writing at 0x0001c000... (66 %)
Writing at 0x00020000... (83 %)
Writing at 0x00024000... (100 %)
Wrote 159232 bytes (84754 compressed) at 0x00010000 in 1.3 seconds (effective 958.8 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
Adding "flash"'s dependency "all" to list of commands with default set of options.
Executing action: all (aliases: build)
Running ninja in directory c:\git\blink_on_atom\blink_on_atom\build
Executing "ninja all"...
Executing action: flash
Running esptool.py in directory c:\git\blink_on_atom\blink_on_atom\build
Executing "C:\Users\hoepffr\python_env\idf4.1_py3.8_env\Scripts\python.exe C:\Users\hoepffr\python_env\idf4.1_py3.8_env\Scripts\python.exe C:\Users\hoepffr\python_env\idf4.1_py3.8_env\Scripts\python.exe"
Done
```

jetzt sollte es so aussehen,
und der ATOM sollte blinken.
machen Sie sich mit dem Programm vertraut.

26.0.8 Schritt_22

<https://github.com/espressif/idf-eclipse-plugin/blob/master/README.md#viewing-serial-output>



```

I (69) boot: 0 nvs                WiFi data      01 02 00009000 00006000
I (77) boot: 1 phy_init          RF data        01 01 0000f000 00001000
I (84) boot: 2 factory           factory app     00 00 00010000 00100000
I (92) boot: End of partition table
I (96) boot_comm: chip revision: 1, min. application chip revision: 0
I (103) esp_image: segment 0: paddr=0x00010020 vaddr=0x3f400020 size=0x05f78 ( 24440) map
I (122) esp_image: segment 1: paddr=0x00015fa0 vaddr=0x3ffb0000 size=0x02180 ( 8576) load
I (125) esp_image: segment 2: paddr=0x00018128 vaddr=0x40080000 size=0x00404 ( 1028) load
I (130) esp_image: segment 3: paddr=0x00018534 vaddr=0x40080404 size=0x07ae4 ( 31460) load
I (153) esp_image: segment 4: paddr=0x00020020 vaddr=0x400d0020 size=0x13f28 ( 81704) map
I (184) esp_image: segment 5: paddr=0x00033f50 vaddr=0x40087ee8 size=0x02e88 ( 11912) load
I (195) boot: Loaded app from partition at offset 0x10000
I (196) boot: Disabling RNG early entropy source...
I (196) cpu_start: Pro cpu up.
I (200) cpu_start: Application information:
I (204) cpu_start: Project name:      led_strip
I (210) cpu_start: App version:      555a728
I (215) cpu_start: Compile time:      Sep 11 2020 16:12:44
I (221) cpu_start: ELF file SHA256:  8260228b89c4e775...
I (227) cpu_start: ESP-IDF:          v4.1-dirty
I (232) cpu_start: Starting app cpu, entry point is 0x400810b4
I (0) cpu_start: App cpu up.
I (242) heap_init: Initializing. RAM available for dynamic allocation:
I (249) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (255) heap_init: At 3FFB29C0 len 0002D640 (181 KiB): DRAM
I (261) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (268) heap_init: At 3FFE4350 len 00018CB0 (111 KiB): D/IRAM
I (274) heap_init: At 4008AD70 len 00015290 (84 KiB): IRAM
I (280) cpu_start: Pro cpu start user code
I (298) spi_flash: detected chip: gd
I (299) spi_flash: flash io: dio
W (299) spi_flash: Detected size(4096k) larger than the size in the binary image header(2048k). Using th
e size in the binary image header.
I (309) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
I (321) example: LED Rainbow Chase Start

```

26.0.9 Programm verändern

verändern Sie zum Beispiel die CHASE_SPEED auf 10ms,
 und schauen Sie was passiert.
 machen Sie solche Änderungen noch ein paar mal, um mit dem Programm vertraut zu werden.

27. cci_EasyExample

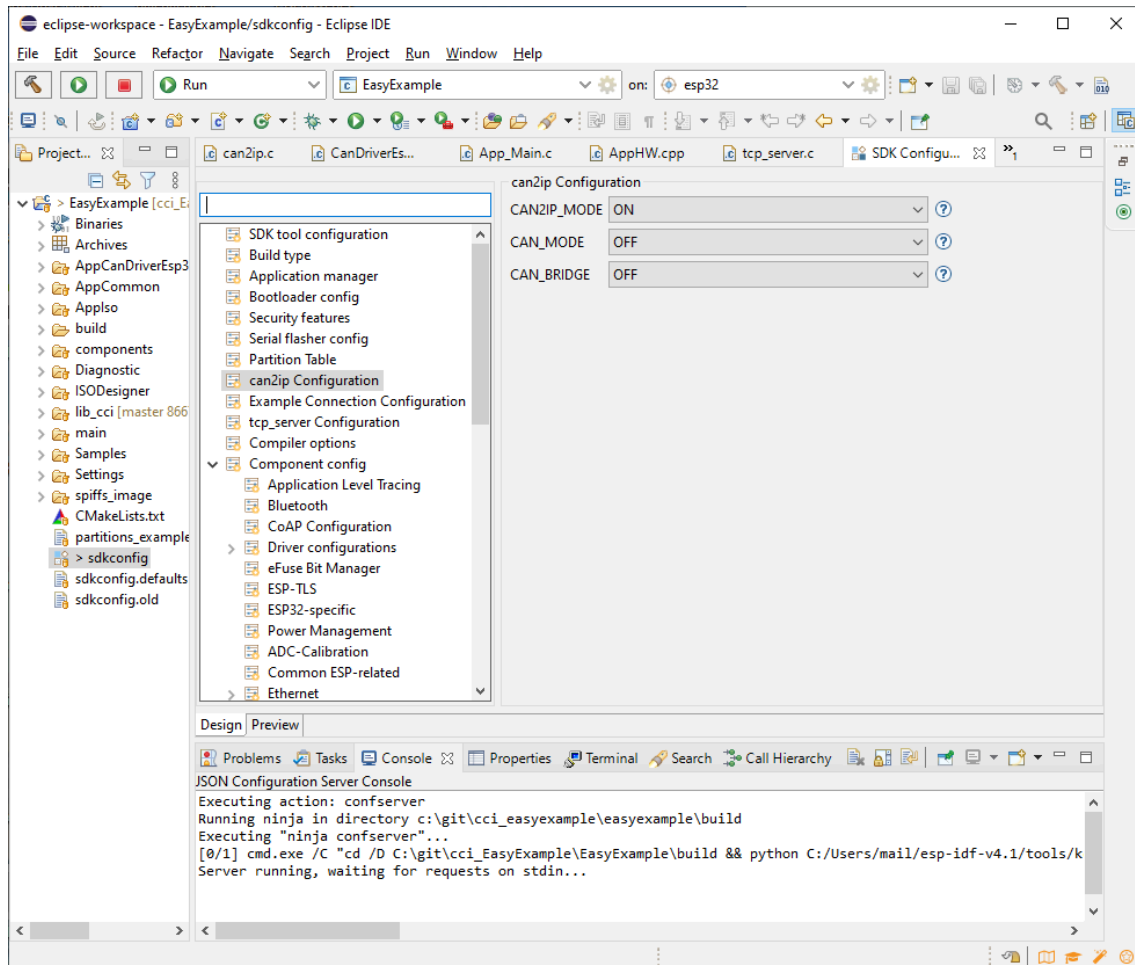
Sie können jetzt ja schon alles:

- sie holen sich mit Gitkraken das cci_EasyExample,
- öffnen den Workspace C:\git\cci_EasyExample\eclipse-workspace
- importieren das Projekt,
- setzen die ESP-IDF Version,
- setzen das Build Target,
- compilieren das ganze und testen es auf dem Traktor oder ISOBUS Teststation.
- die Pools für ISO-Designer finden Sie unter: C:\git\cci_EasyExample\EasyExample\ISODesigner\MyWorkspace1

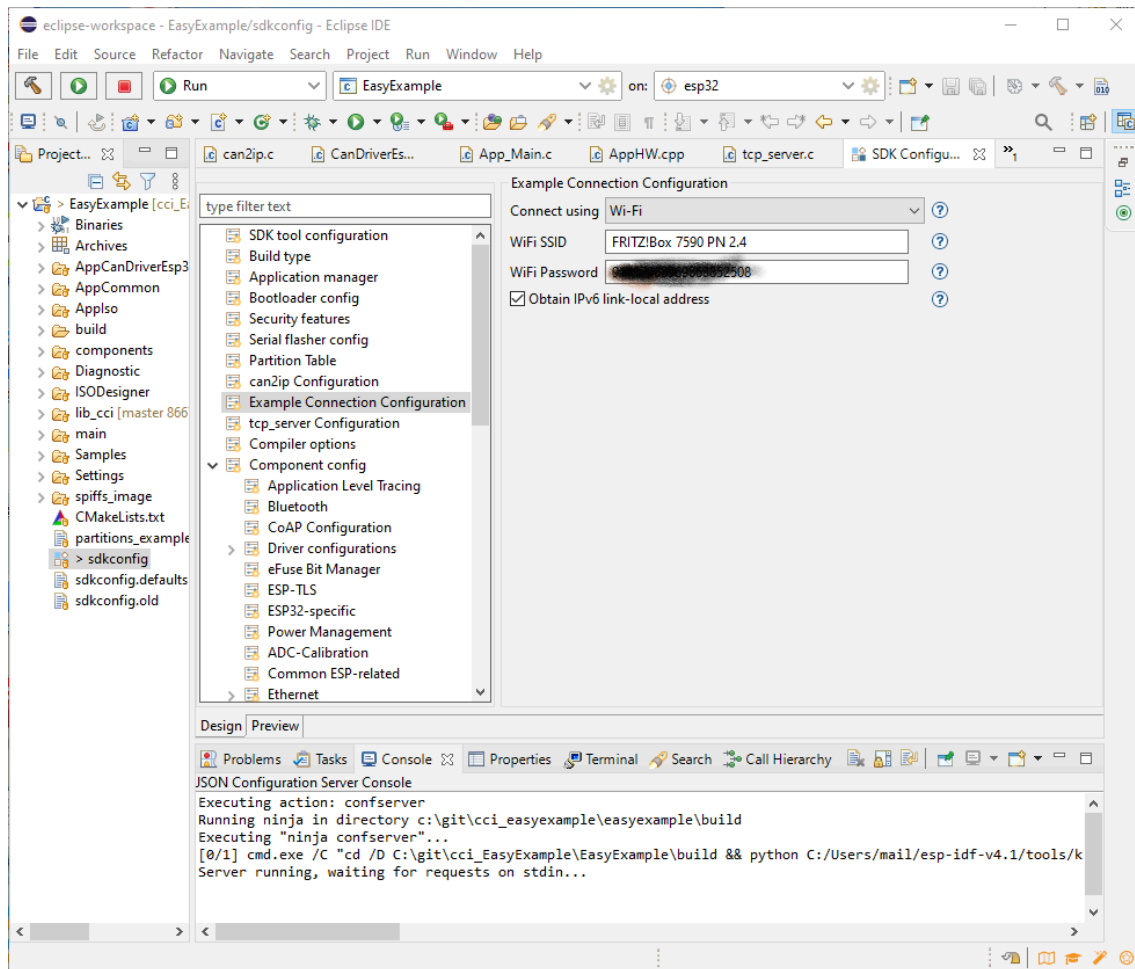
28. cci_EasyExample_CAN2IP

Besondere Umstände erfordern besondere Maßnahmen:

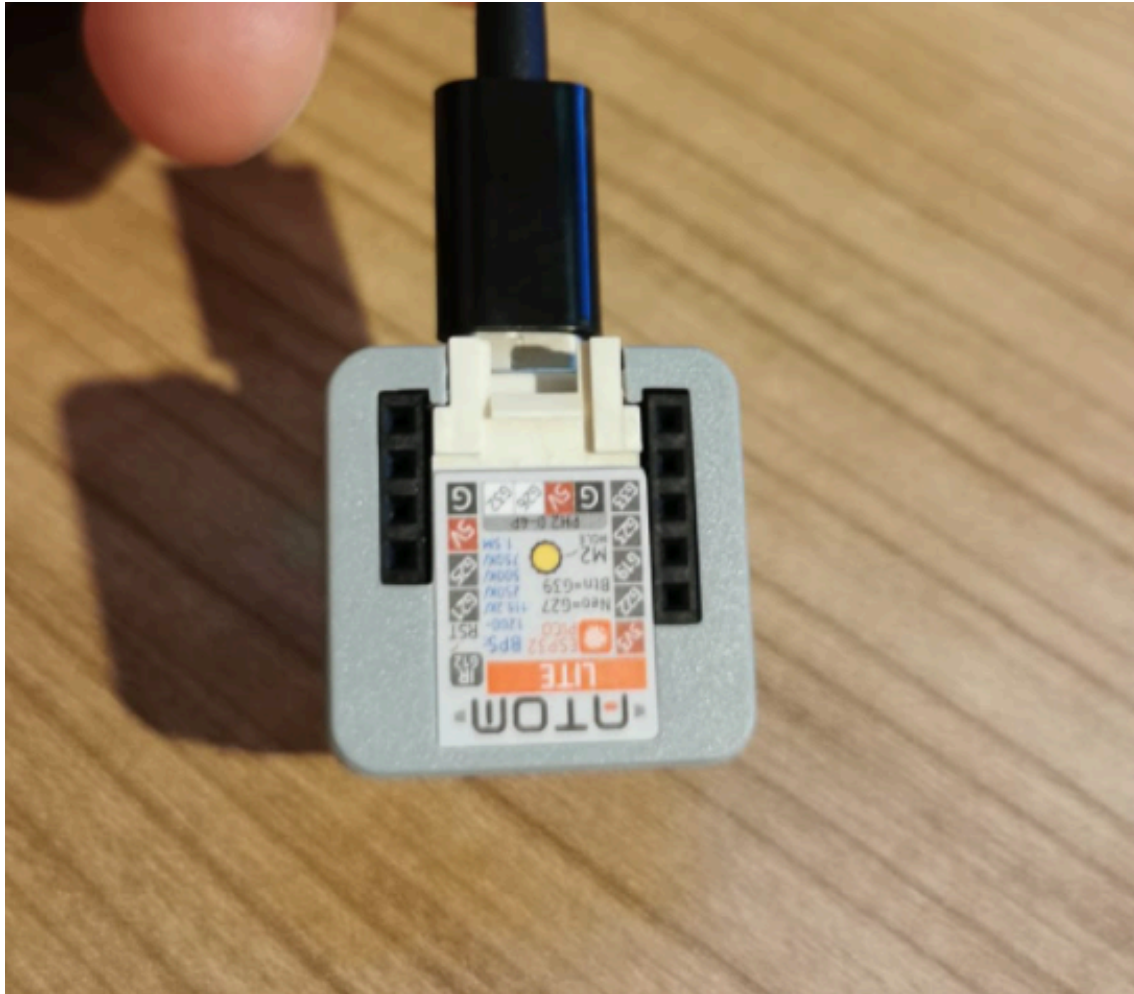
- führen Sie einen PULL auf cci_EasyExample aus
- im sdkconfig sollte „CAN2IP“ gewählt sein * Screenshot:



- Erklärung: wir haben nun 2 Wege um sich zur ISOBUS ECU zu verbinden: * den „virtuellen“ CAN, der eigentlich WLAN ist: CAN2IP * den „bekannten“ physischen CAN, der auch im Traktor vorhanden ist: CAN
- ebenfalls im sdkconfig müssen Sie ihre SSID und Kennwort eingeben * Screenshot:



- nun laden Sie das Programm auf den M5 Atom
- die CAN Hardware brauchen Sie nicht unbedingt hierzu.



- in der Console erscheint nun eine IP-Adresse

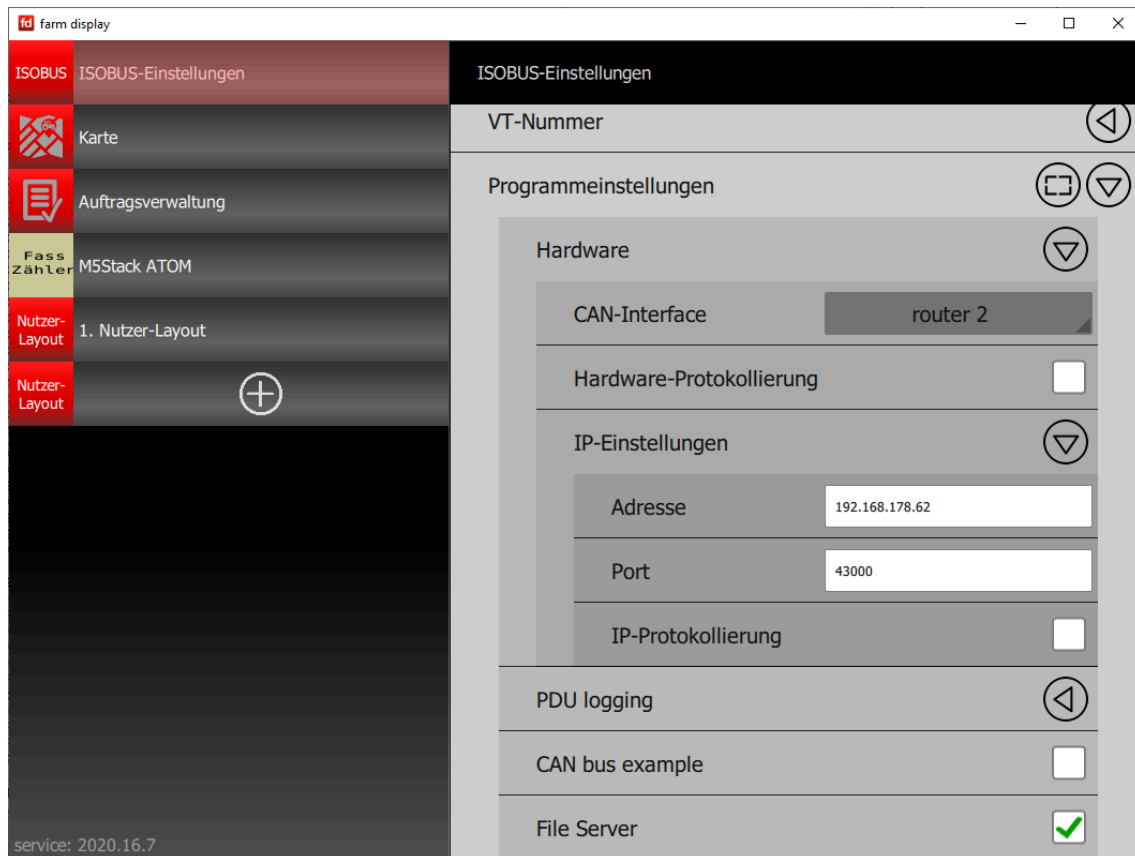
```

I (809) wifi:Init static rx buffer size: 1600
I (814) wifi:Init static rx buffer num: 10
I (819) wifi:Init dynamic rx buffer num: 32
I (824) example_connect: Connecting to FRITZ!Box 7590 PN 2.4...
I (919) phy: phy_version: 4180, cb3948e, Sep 12 2019, 16:39:13, 0, 0
I (924) wifi:mode : sta (50:02:91:9f:51:1c)
I (924) example_connect: Waiting for IP
I (1044) wifi:new:<1,0>, old:<1,0>, ap:<255,255>, sta:<1,0>, prof:1
I (1824) wifi:state: init -> auth (b0)
I (1829) wifi:state: auth -> assoc (0)
I (1834) wifi:state: assoc -> run (10)
I (1869) wifi:connected with FRITZ!Box 7590 PN 2.4, aid = 5, channel 1, BW20, bssid = dc:39:6f:65:9a:d7
I (1869) wifi:security: WPA2-PSK, phy: bgn, rssi: -63
I (1874) wifi:pm start, type: 1

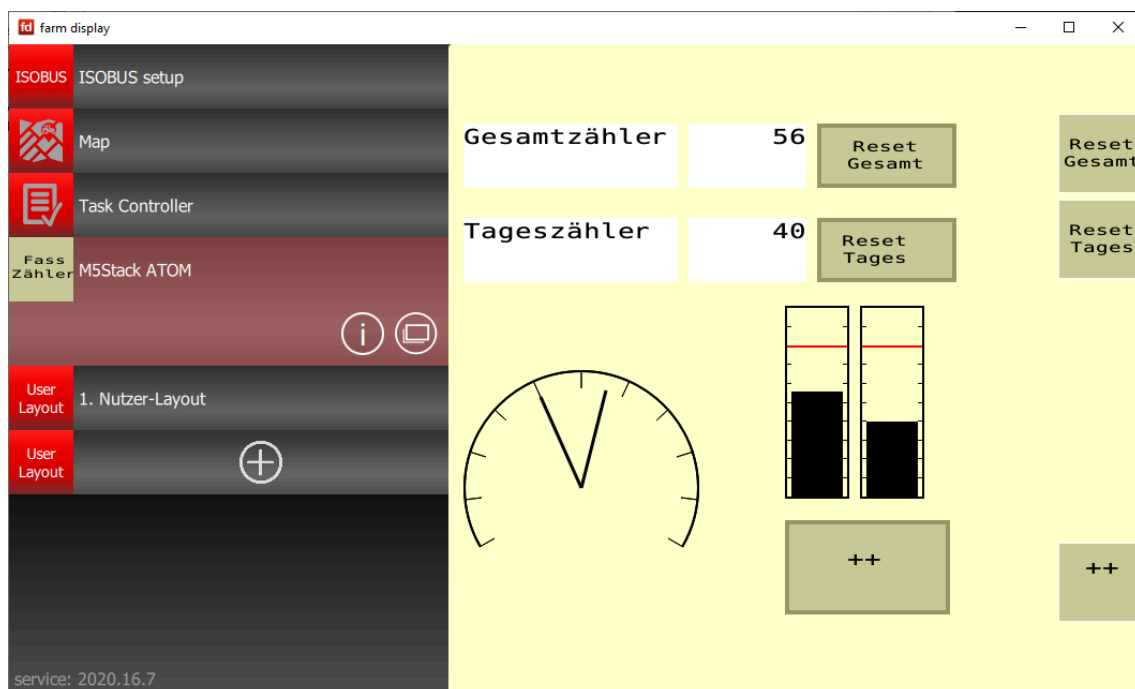
I (1919) wifi:AP's beacon interval = 102400 us, DTIM period = 1
I (2744) esp_netif_handlers: sta ip: 192.168.178.62, mask: 255.255.255.0, gw: 192.168.178.1
I (2749) example_connect: Got IP event!
I (3744) example_connect: Got IPv6 event!
I (3744) example_connect: Connected to FRITZ!Box 7590 PN 2.4
I (3744) example_connect: IPv4 address: 192.168.178.62
I (3749) example_connect: IPv6 address: fe80:0000:0000:0000:5202:91ff:fe9f:511c
app_main_tcp_server DONEapp_main_tcp DONEI (3759) tcp_server: Socket created
I (3764) tcp_server: Socket bound, port 43000
I (3769) tcp_server: Socket listening
I (4299) settings: getU8, section = CF-A, key = sourceAddress, value = 140
NE CH 0: This Active CF 8c (1400) Time: 3647

```

- diese IP-Adresse geben sie im Farm-Display ein * installation siehe: [nx_farm_display](#) * Als CAN-Interface wählen Sie „router 2“ * Screenshot:



- klicken Sie „Neustart“
- nun sollte der Object-Pool erscheinen im Farm Display.



Sie haben nun eine Komplette Virtuelle Umgebung rund um den ISOBUS.
 Vorteil: Sie müssen keine Hardware bauen !

29. Das steht im Lehrplan: C-Programmierung in ISOBUS

Quellennachweis & Rechtsgrundlage:

Quelle: Rahmenlehrplan für die Vorbereitung auf die Meisterprüfung im Land- und Baumaschinenmechatroniker-Handwerk

Herausgeber: LandBauTechnik-Bundesverband e. V., Alfredstraße 102, 45131 Essen (Stand: 25.02.2025)

Verordnung: Meisterprüfungsverordnung (LandBauMechMstrV) vom 09.09.2024 (BGBl. 2024 I Nr. 277, in Kraft ab 01.08.2025)

Zentrales Portal der Meisterschulen München:

Visit the main platform: www.ms-muc-docs.de |  Interaktiver Rahmenlehrplan 2025 |  Wiki-Übersicht |  PDF-Downloads

29.1 **Relevanz der C-Programmierung & ISO 11783 C-Libraries im Lehrplan 2025**

Die C/C++-Programmierung für eingebettete Systeme ist das Fundament für die Implementierung normgerechter Kommunikation an Land- und Baumaschinen nach ISO 11783.

29.2 **Lehrplan-Module & Schwerpunkte**

29.2.1 **1. Teil I – Lerneinheit 1.4: Automatisierte Funktionen (80 UE)**

- **Implementierung von ISOBUS-Protokollen:** Einbindung von C-Libraries für CAN-Kommunikation und IsoAgLib.

29.2.2 **2. Teil II – Lerneinheit 1.1: Systemtechnik (120 UStd.)**

- **Software-Architektur:** Speicherverwaltung, Bitoperationen, Zeiger und Systemtreiber in C/C++ für Mikrocontroller.

29.2.3 **Passende Themen-Unterseiten auf ms-muc-docs.de**

-  Eclipse 4diac IDE & Farb-Referenz auf ms-muc-docs.de
-  MCU vs. MPU Vergleichsguide auf ms-muc-docs.de

29.3 **Übersicht, Hauptportal & Direktlinks zu allen 6 Wikis**

 **Hauptportal:** www.ms-muc-docs.de (Zentrale Informationsplattform der Meisterschulen am Ostbahnhof München)

Wiki	Thema / Beschreibung	Link auf ReadTheDocs
Wiki 0	Wiki der Meisterschulen Allgemeine Informationen, Lehrplan 2025 & Haupt-Wiki.	 Öffnen
Wiki 1	C-Programmierung in ISOBUS Grundlagen der C-Programmierung und ISOBUS-Umgebung.	 Öffnen
Wiki 2	ISOBUS Virtual Terminal (VT) Benutzeroberflächen, ISO-Designer, VT-Objektpools & HMI.	 Öffnen
Wiki 3	ISOBUS Technik & Maschinensysteme Hydraulik, Pneumatik, Motoren, HV-Technik, Getriebe & Maschinensicherheit.	 Öffnen
Wiki 4	Visuelle Programmiersprachen Low-Code Entwicklung, IEC 61499 & 4diac IDE (Lehrplan-Hauptthema).	 Öffnen
Wiki 5	Elektronik-Werkzeuge & Messgeräte Handbuch für Crimptechnik, Lötstationen, Oszilloskope & Mikroskope.	 Öffnen