


Inhaltsverzeichnis

Beschreibung der vom RONPAS-Compiler unterstützten Programmer	2
1. mySmartUSB light	2
1.1. mySmartUSB light - Link	2
1.2. mySmartUSB light - Treiber	2
1.3. mySmartUSB light im RONPAS-Compiler	5
2. Diamex	8
2.1. Diamex - Link	8
2.2. Diamex - Treiber	8
2.3. Diamex im RONPAS-Compiler	9
3. Pololu AVR Programmer v2.1	11
3.1. Pololu AVR Programmer - Link	12
3.3. Pololu AVR Programmer im RONPAS-Compiler	15
4. USBasp AVR Programmer (6-polig und 10-polig ISP-Adapter)	18
4.1. USBasp AVR Programmer - Link (Beispiele)	18
4.2. USBasp - Treiber	19
4.3. USBasp AVR Programmer im RONPAS-Compiler	21
5. AVR ISP mkII Programmer (Atmel kompatibel)	24
5.1. AVR ISP mkII Programmer - Link (Beispiele)	24
5.2. AVR ISP mkII Programmer - Treiber	24
5.3. AVR ISP mkII im RONPAS-Compiler	28

	<p>Programmier-Projekte Lazarus / Free Pascal Ronald Daleske</p>	<pre>Form1 = class(TForm) private public private declare public declare end; var Form</pre>
---	--	---

Startseite	Programmier-Projekte	RONPAS-Compiler
----------------------------	--------------------------------------	------------------------

Beschreibung der vom RONPAS-Compiler unterstützten Programmer

1. mySmartUSB light



Der "mySmartUSB light" Programmer wird von mir seit vielen Jahren genutzt und ist mein Favorit unter den Programmern. Er hat in der Anwendung noch nie Probleme bereitet.

Nachteilig ist nur, dass der Programmer unter Windows 10 ein Treiber benötigt.

Hinweis: Für den "mySmartUSB light" gibt es ein eigenes PROMMER-Programm (myAVR_ProgTool.exe). Es unterstützt aber nur wenige Prommer-Varianten anderer Hersteller. Daher habe ich dieses Programm kaum benutzt.

1.1. mySmartUSB light - Link

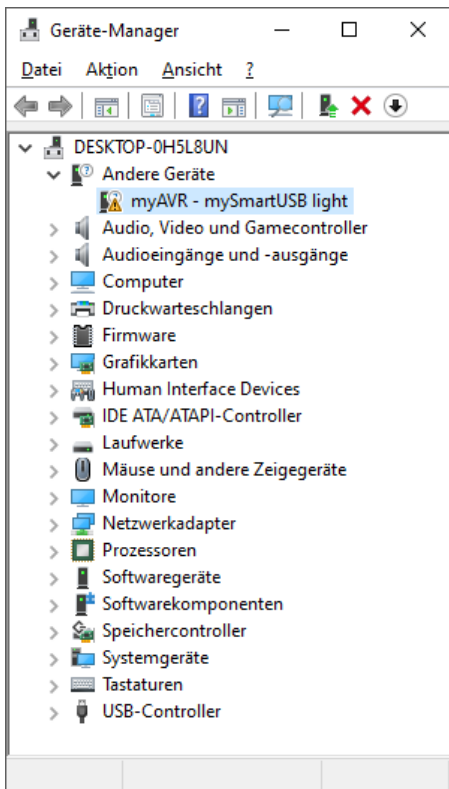
<https://shop.myavr.de>

Preis etwa 15 EUR.

Geschwindigkeit: 42 Sekunden (für die Hex-Datei des Referenzprojektes z-meic, etwa 31KB)

1.2. mySmartUSB light - Treiber

1.2.1. mySmartUSB light ohne Treiber



Nach dem Einstecken des "mySmartUSB light" an den USB-A-Port des PC erscheint im Geräte-Manager (c:\Windows\System32\devmgmt.msc) unter "Andere Geräte" der Eintrag "myAVR - mySmartUSB light". Der Programmierer wurde also grundsätzlich erkannt. Das gelbe Dreieck mit dem Ausrufezeichen kennzeichnet, dass für dieses Gerät kein passender Treiber installiert ist.

1.2.2. mySmartUSB light - Treiber downloaden

Der Treiber kann von der Internetseite des Herstellers heruntergeladen werden:

<https://shop.myavr.de>

Im Shop unter Download nach Treiber suchen und auf den PC laden.

In meinem Fall hatte der Treiber den Namen:

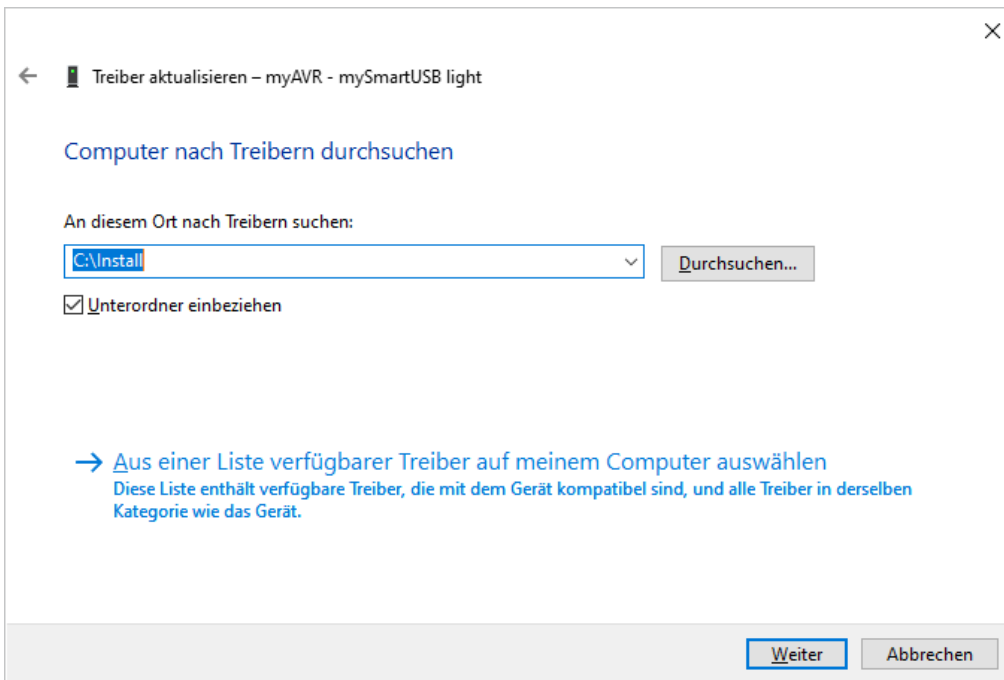
tool_usb-treiber-myavr-board-v6.7.zip

ZIP-Datei entpacken. Hier unter:

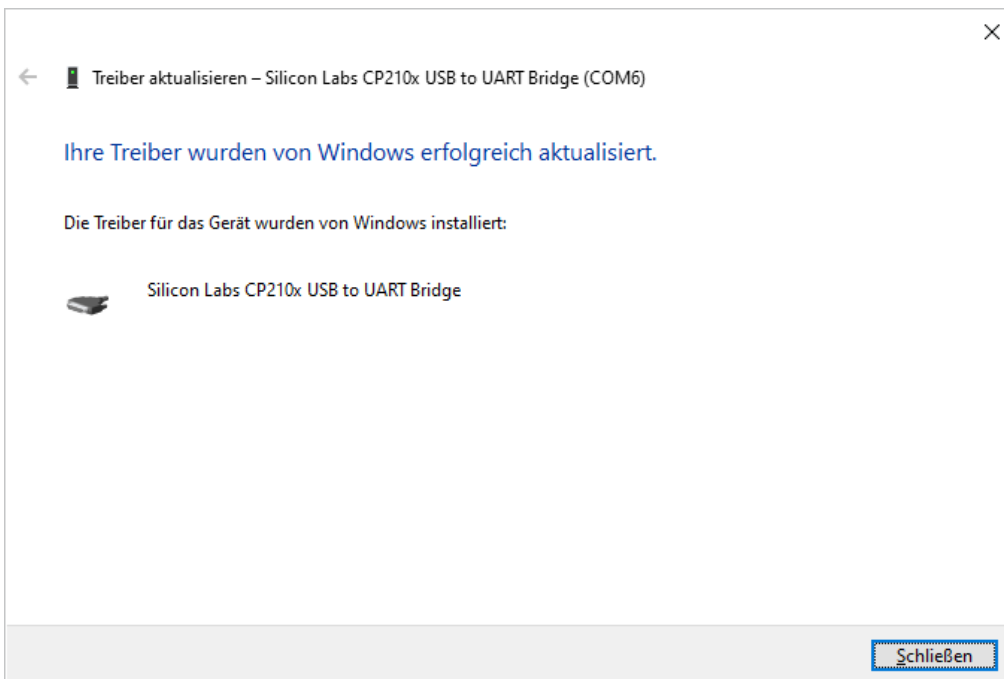
c:\Install\CP210x_VCP_Windows

1.2.3. mySmartUSB light - Treiber installieren

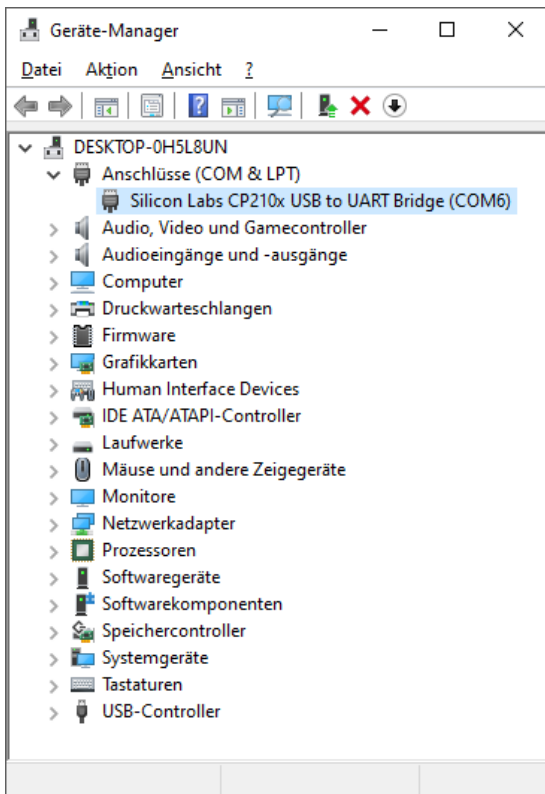
Im Geräte-Manager den Eintrag "myAVR - mySmartUSB light" mit der linken Maustaste aktivieren und dann mit der rechten Maustaste das Kontextmenü öffnen.



"Computer nach Treibern durchsuchen" und dort das Kästchen "Unterordner einbeziehen" aktivieren.



Nach dieser Meldung wurde der Treiber ordnungsgemäß installiert.



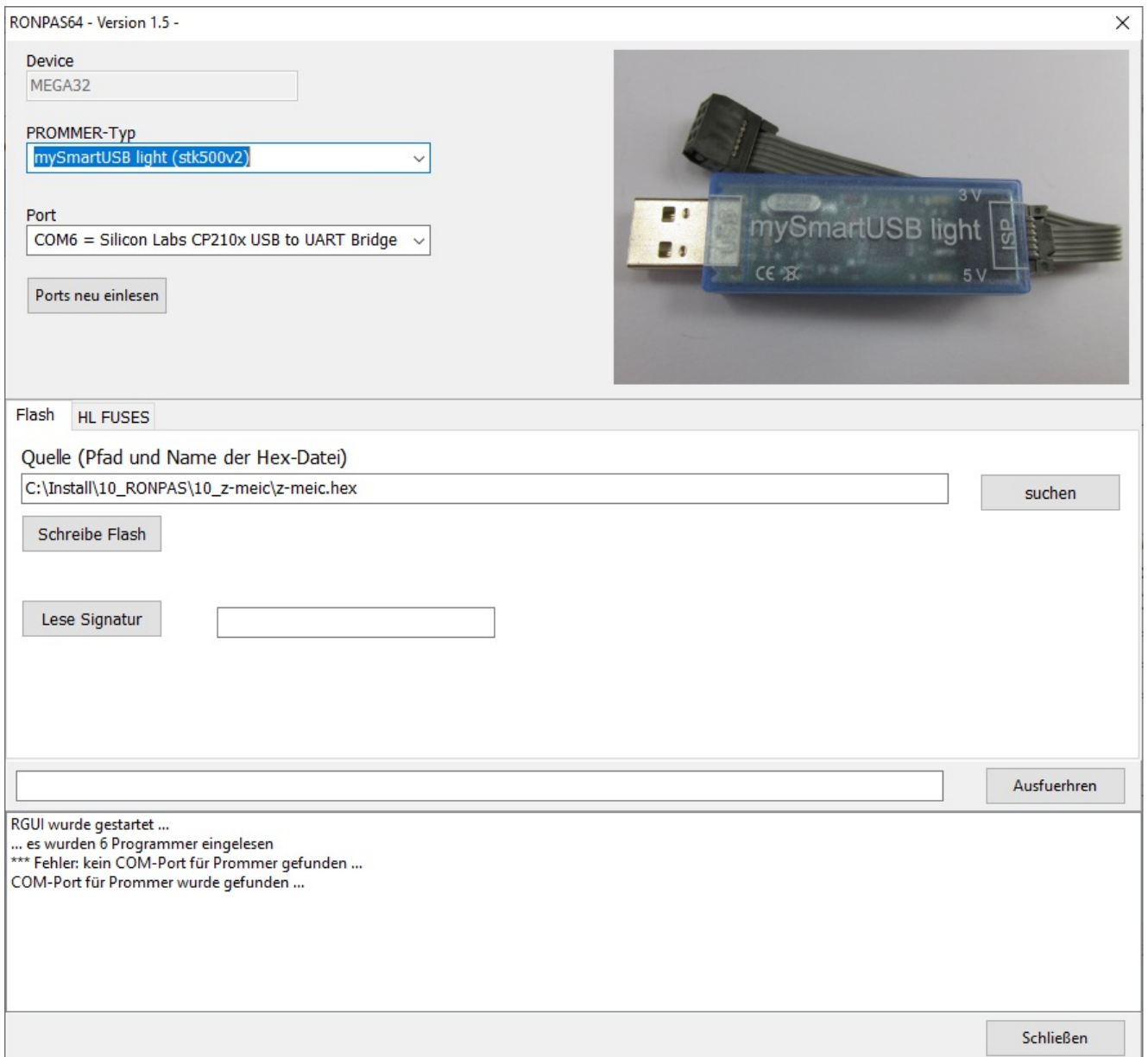
1.3. mySmartUSB light im RONPAS-Compiler

1.3.1. Neu Auswählen des mySmartUSB light-PROMMER im RONPAS-Compiler

Ist der mySmartUSB light-PROMMER noch nicht ausgewählt, muss dieser erst einmal ausgewählt werden.



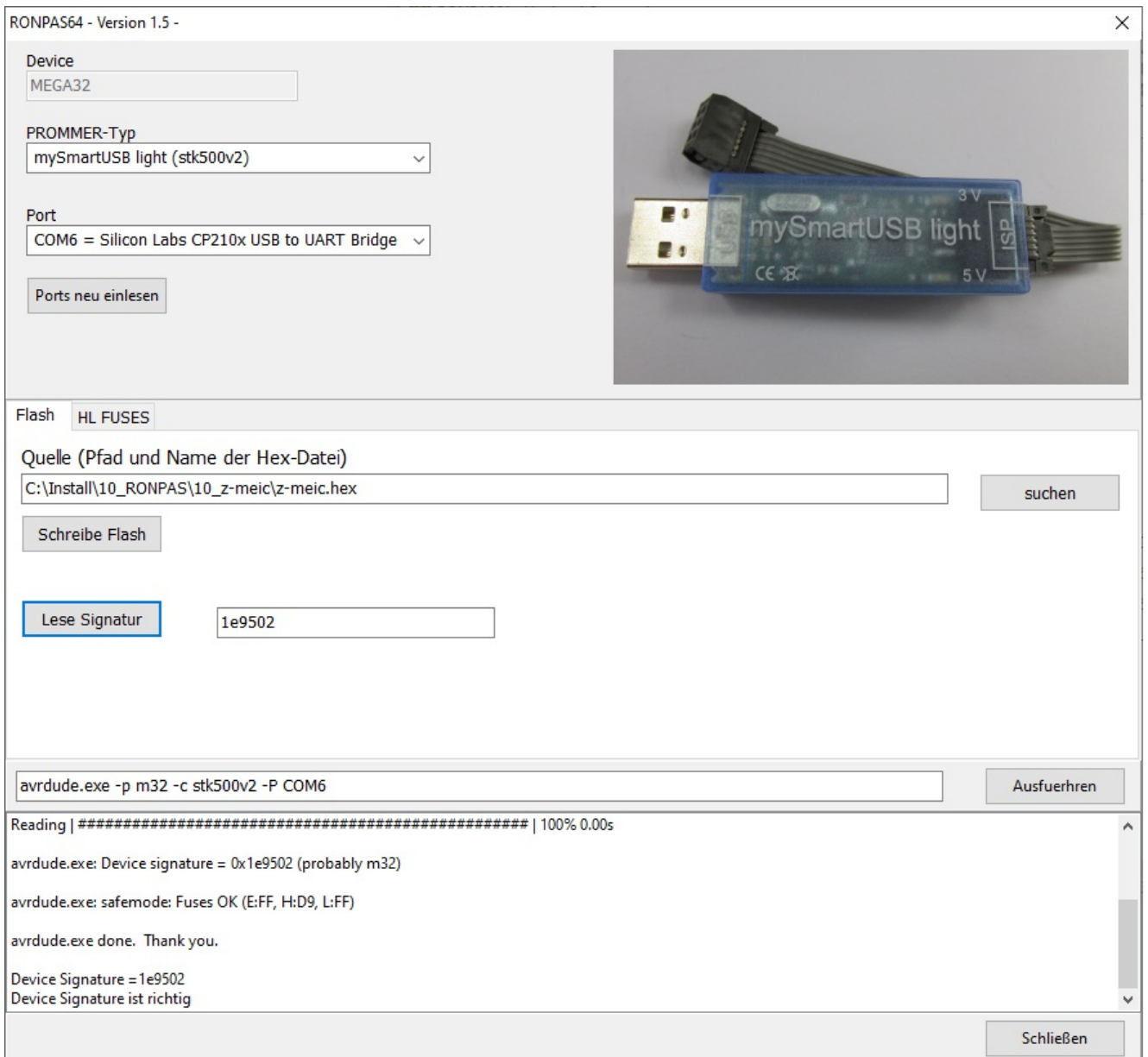
Nach dem Drücken des PROMMER-Button gelangt man in das PROMMER-Formular.



Dort wird unter "PROMMER-Typ" der "mySmartUSB light (stk500v2)" ausgewählt. Zum sicheren Abspeichern der neuen Einstellungen das Formular "PROMMER" mit dem Button "Schliessen" beenden. Danach auch den RONPAS-Compiler beenden.

1.3.2. mySmartUSB light-PROMMER am Board testen

Das zu programmierende Board mit dem ATMEGA muss eingeschaltet sein (+5V Betriebsspannung ist angeschlossen). Der mySmartUSB light-PROMMER muss über die USB-A-Buchse an den PC angeschlossen und er sollte über den 6-poligen Stecker mit dem Board verbunden sein. Erst jetzt wird der RONPAS-Compiler gestartet, da er zum Beginn des Starts die COM-Ports abfragt. Nach dem Start des RONPAS-Compilers noch einmal über den Button PROMMER in das PROMMER-Formular gehen.



Unter "Port" sollte ein COM-Port und die Bezeichnung "Silicon Labs CP210x USB to UART Bridge" angezeigt werden (z.B. "COM6 = Silicon Labs CP210x USB to UART Bridge").

Um die einwandfreie Funktion des mySmartUSB light-PROMMERs zu testen wird einfach der Button "Lese Signatur" mit der Maus angeklickt werden. Über das Programm AVRDUDE wird nun die Signatur des angeschlossenen ATMEGAs (z.B. ATMEGA32) gelesen und mit der abgespeicherten Signatur des erkannten Controllers verglichen.

Stimmt die gelesene Signatur mit der abgespeicherten Signatur überein, wird im unteren Anzeigefenster die Meldung:

"Device Signature ist richtig."

angezeigt. Damit ist der mySmartUSB light-PROMMER einsatzbereit und das PROMMER-Formular kann wieder geschlossen werden.

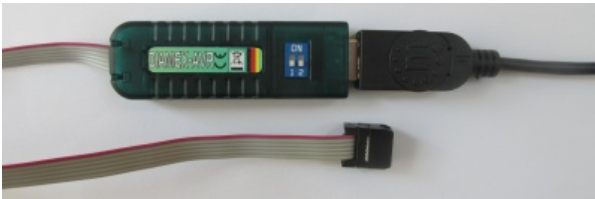
1.3.3. AVR programmieren mit dem mySmartUSB light-PROMMER



Der aktuelle Programmcode kann nun einfach mit dem Drücken des "Kompilieren und Programmieren-Button" übersetzt

(PAS -> ASM -> HEX) und anschliessend in den angeschlossenen ATMEGA gebrannt werden.

2. Diamex



Der Diamex ist für mich nach dem "mySmartUSB light" auf Rang 2. Er benötigt unter Windows 10 keinen eigenen Treiber und arbeitet reibungslos. Preislich liegt er etwas über dem "mySmartUSB light".

Preis etwa 20 EUR.

Geschwindigkeit: 50 Sekunden (für die Hex-Datei des Referenzprojektes z-meic, etwa 31KB)

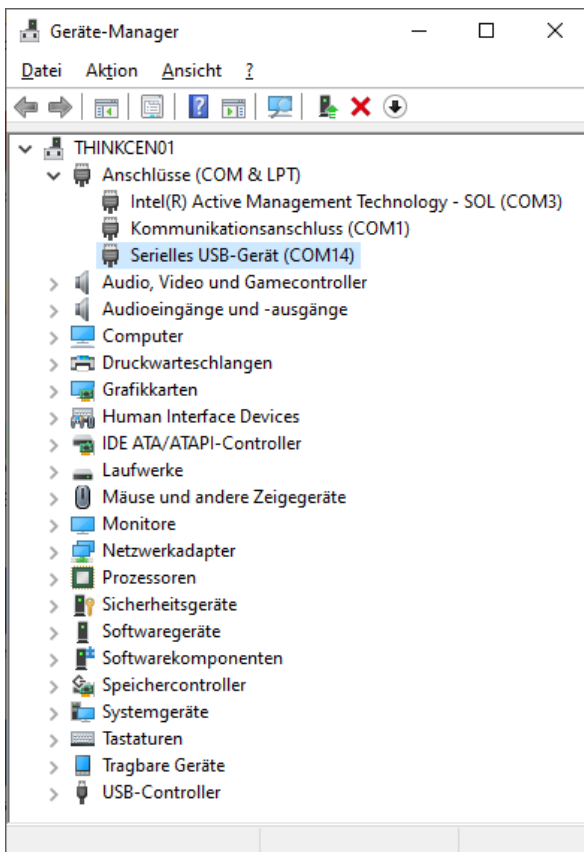
2.1. Diamex - Link

<https://www.diamex.de>

2.2. Diamex - Treiber

Der Diamex benötigt für Windows 10 keinen Treiber. Nach dem Einstecken des Programmers in den USB-Port weist Windows den Treiber automatisch zu.

Im Geräte manager (c:\Windows\System32\devmgmt.msc) unter "Anschlüsse (COM&LPT)" erscheint der Diamex als "Serielles USB-Gerät" (hier mit dem COM-Port "COM14" verbunden).



Unter den hier vorgestellten Programmen ist der Diamex der einzige Programmer, bei dem kein Treiber installiert werden muss um im RONPAS-Compiler genutzt werden zu können.

Das ist schon etwas Besonderes.

2.3. Diamex im RONPAS-Compiler

2.3.1. Neu Auswählen des Diamex-PROMMER im RONPAS-Compiler

Ist der Diamex-PROMMER noch nicht ausgewählt, muss dieser erst einmal ausgewählt werden.



Nach dem Drücken des PROMMER-Button gelangt man in das PROMMER-Formular.

RONPAS64 - Version 1.5

Device
MEGA32

PROMMER-Typ
Diamex (stk500v2)

Port
COM14 = Serielles USB-Gerät

Ports neu einlesen

Flash | HL FUSES

Quelle (Pfad und Name der Hex-Datei)
P:\Projekte\10_RONPAS\54_z-meic_V4\20_z-meic\17_z-meic_IORQ-Tabelle_ok_REF\z-meic.hex

suchen

Schreibe Flash

Lese Signatur

Ausfuehren

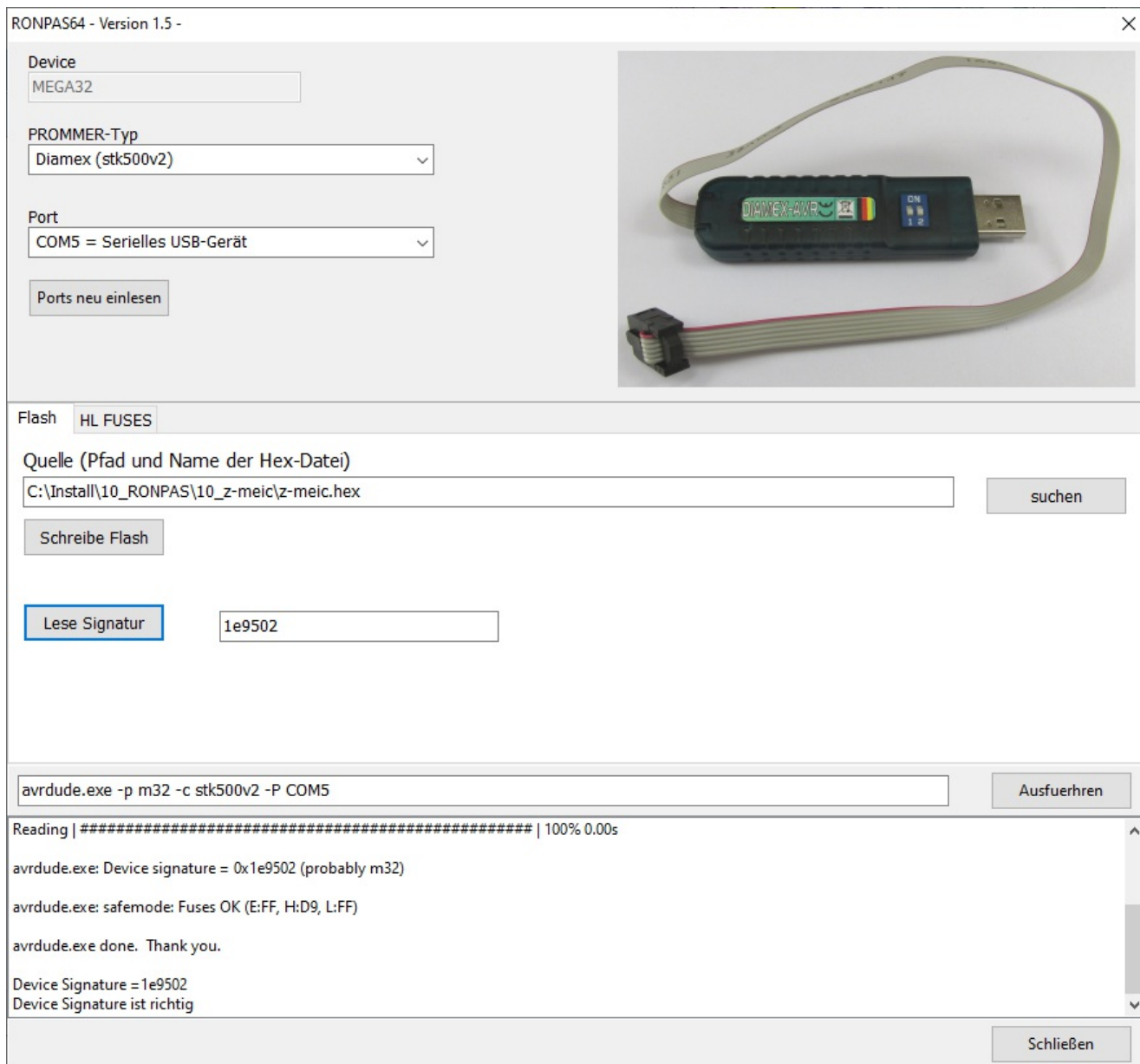
RGUI wurde gestartet ...
... es wurden 6 Programmer eingelesen
COM-Port für Prommer wurde gefunden ...

Schließen

Dort wird unter "PROMMER-Typ" der "Diamex (stk500v2)" ausgewählt. Zum sicheren Abspeichern der neuen Einstellungen das Formular "PROMMER" mit dem Button "Schliessen" beenden. Danach auch den RONPAS-Compiler beenden.

2.3.2. Diamex-PROMMER am Board testen

Das zu programmierende Board mit dem ATMEGA muss eingeschaltet sein (+5V Betriebsspannung ist angeschlossen). Der Diamex-PROMMER muss über die USB-A-Buchse an den PC angeschlossen und er sollte über den 6-poligen Stecker mit dem Board verbunden sein. Erst jetzt wird der RONPAS-Compiler gestartet, da er zum Beginn des Starts die COM-Ports abfragt. Nach dem Start des RONPAS-Compilers noch einmal über den Button PROMMER in das PROMMER-Formular gehen.



The screenshot shows the RONPAS64 - Version 1.5 software window. The 'Device' field is set to 'MEGA32', the 'PROMMER-Typ' is 'Diamex (stk500v2)', and the 'Port' is 'COM5 = Serielles USB-Gerät'. A button 'Ports neu einlesen' is visible. An image of the Diamex PROMMER device is shown on the right. Below the configuration, the 'Flash' tab is active, showing 'HL FUSES'. The 'Quelle (Pfad und Name der Hex-Datei)' field contains 'C:\Install\10_RONPAS\10_z-meic\z-meic.hex' with a 'suchen' button. A 'Schreibe Flash' button is also present. The 'Lese Signatur' button is highlighted, and the '1e9502' field contains the read signature. Below this, the command 'avrdude.exe -p m32 -c stk500v2 -P COM5' is entered in a text box with an 'Ausfuehren' button. The output window shows the following text: 'Reading | ##### | 100% 0.00s', 'avrdude.exe: Device signature = 0x1e9502 (probably m32)', 'avrdude.exe: safemode: Fuses OK (E:FF, H:D9, L:FF)', 'avrdude.exe done. Thank you.', 'Device Signature = 1e9502', and 'Device Signature ist richtig'. A 'Schließen' button is at the bottom right.

Unter "Port" sollte ein COM-Port und die Bezeichnung "Serielles USB-Gerät" angezeigt werden.

Um die einwandfreie Funktion des Diamex-PROMMERs zu testen wird einfach der Button "Lese Signatur" mit der Maus angeklickt werden. Über das Programm AVRDUDE wird nun die Signatur des angeschlossenen ATMEGAs (z.B. ATMEGA32) gelesen und mit der abgespeicherten Signatur des erkannten Controllers verglichen.

Stimmt die gelesene Signatur mit der abgespeicherten Signatur überein, wird im unteren Anzeigefenster die Meldung:

"Device Signature ist richtig."

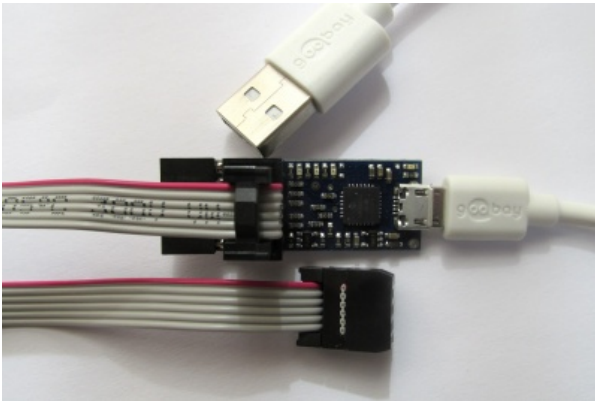
angezeigt. Damit ist der Diamex-PROMMER einsatzbereit und das PROMMER-Formular kann wieder geschlossen werden.

2.3.3. AVR programmieren mit dem Diamex-PROMMER

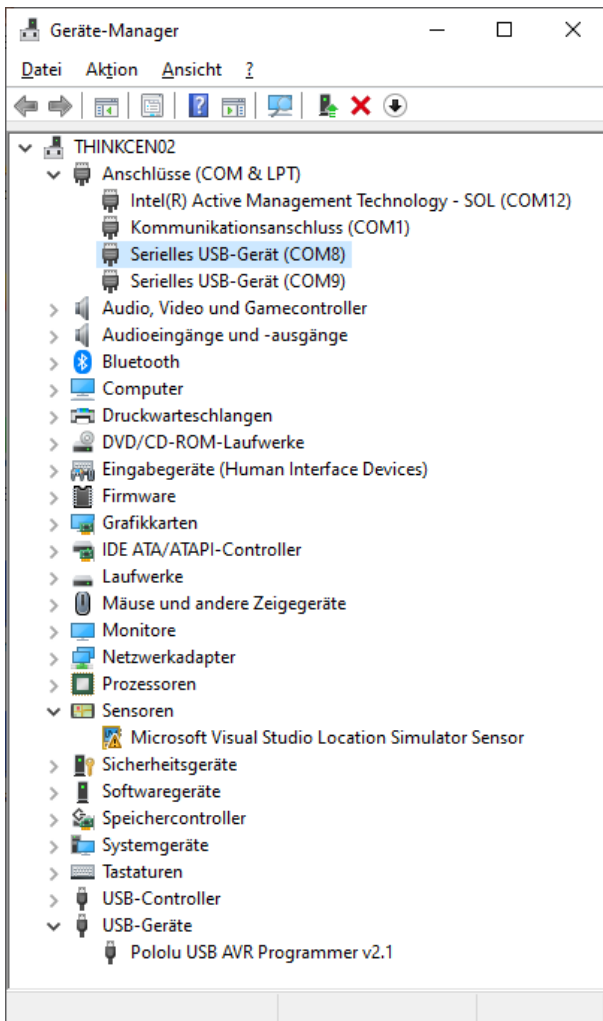


Der aktuelle Programmcode kann nun einfach mit dem Drücken des "Kompilieren und Programmieren-Button" übersetzt (PAS -> ASM -> HEX) und anschliessend in den angeschlossenen ATMEGA gebrannt werden.

3. Pololu AVR Programmer v2.1



Der Pololu AVR Programmer benötigt theoretisch (wie der Diamex) keinen Treiber unter Windows 10. Nach dem Einstecken in die USB-Buchse erscheint im Gerätemanager (c:\Windows\System32\devmgmt.msc) unter "Anschlüsse (COM&LPT)" der Pololu als "Serielles USB-Gerät" (hier mit dem COM-Port "COM3" verbunden).



Soweit der positive Teil des Programmers. Leider haben sich die Entwickler offensichtlich gedacht, dass es eine gute Sache wäre, wenn man in den Programmer zusätzlich eine serielle Schnittstelle integrieren würde. In der Praxis entstehen damit aber sofort ein Probleme, denn auch die serielle Schnittstelle meldet sich im Gerätemanager (c:\Windows\System32\devmgmt.msc) unter "Anschlüsse (COM&LPT)" als "Serielles USB-Gerät" (hier mit dem COM-Port "COM4" verbunden). Will man den Programmer ansprechen muss man die dazugehörige serielle Schnittstelle auswählen. Da es aber 2 Schnittstellen mit der gleichen Bezeichnung gibt, muss man sich für eine Schnittstelle entscheiden (meist ist die erste Schnittstelle dem Programmer zugeordnet, es kann aber auch anders herum sein). Wählt man die richtige Schnittstelle, ist alles ok. Wählt man die falsche Schnittstelle, so hängt sich das dazugehörige Programm (hier der RONPAS-Compiler) auf und man muss den aktuellen User am PC abmelden und wieder anmelden. Eine sehr aufwendige und unschöne Prozedur.

Um diese Fehlermöglichkeit zu umgehen und wieder eine automatische Zuordnung erreichen zu können, kommt man nicht umher, doch den Treiber des Herstellers unter Windows 10 zu installieren.

Preis ab etwa 20 EUR (da dieser Programmer in den USA hergestellt wird gibt es mehrere Anbieter in Deutschland und der Preis schwankt dort recht stark (20-47 EUR).

Geschwindigkeit: 62 Sekunden (für die Hex-Datei des Referenzprojektes z-meic, etwa 31KB)

3.1. Pololu AVR Programmer - Link

[Pololu USB AVR Programmer v2.1 - https://www.pololu.com](https://www.pololu.com)

3.2.2. Pololu - Treiber downloaden

Der Treiber kann von der Internetseite des Herstellers (in den USA) heruntergeladen werden:

<https://www.pololu.com>

In meinem Fall hatte der Treiber den Namen:

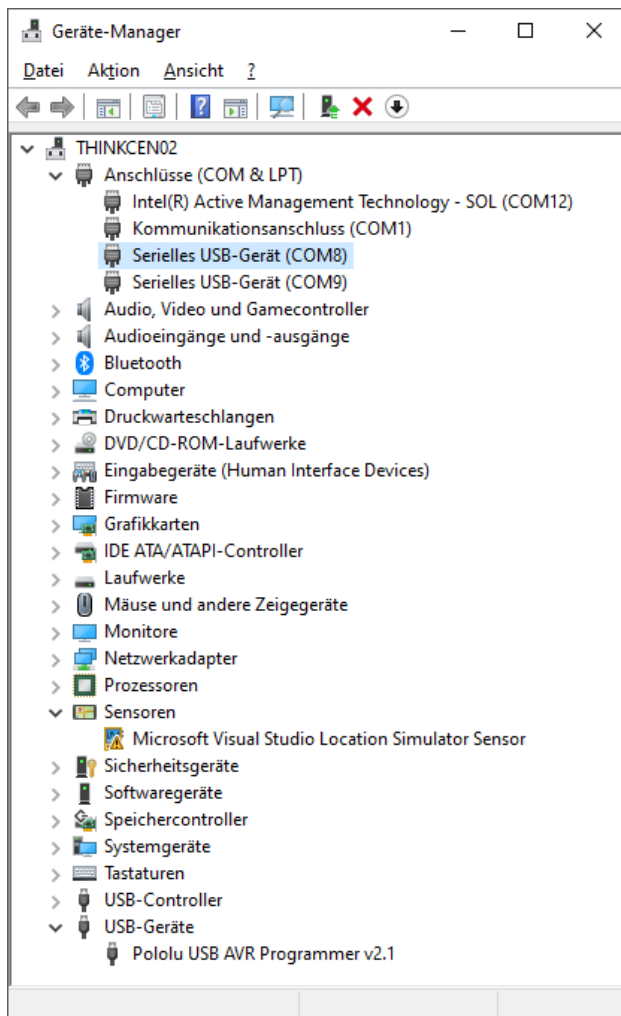
pololu-usb-avr-programmer-v2-1.1.0-win.msi

Programme und Treiber installieren unter:

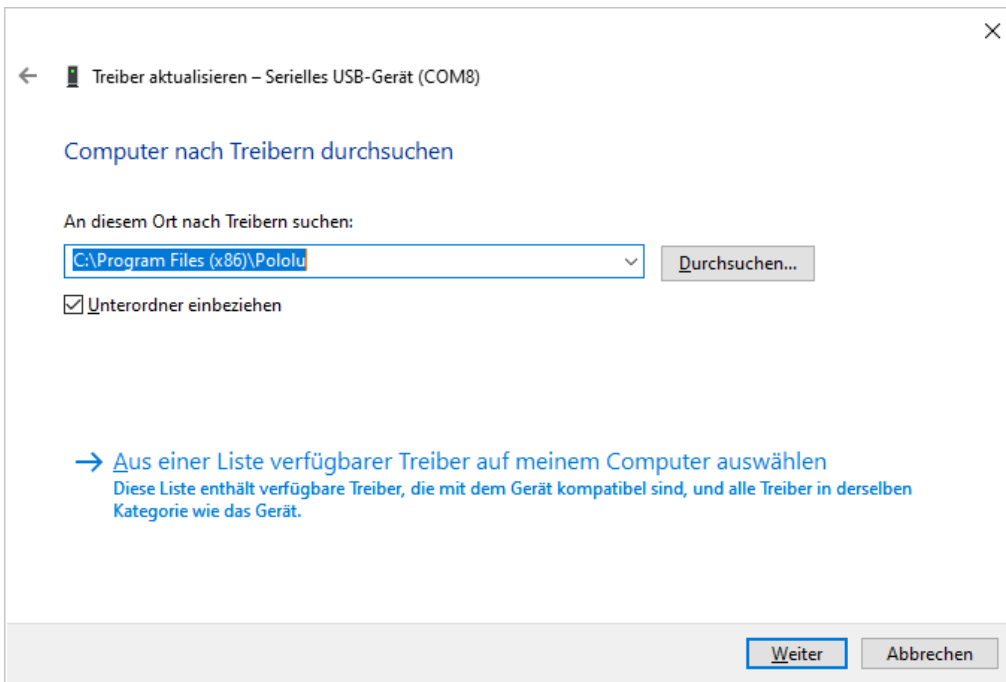
"C:\Program Files (x86)\Pololu\USB AVR Programmer v2"

3.2.3. Pololu - Treiber installieren

Nachdem die "Programme und Treiber" über die msi-Datei installiert wurden, sieht es im Geräte-Manager immer noch so wie vorher aus.



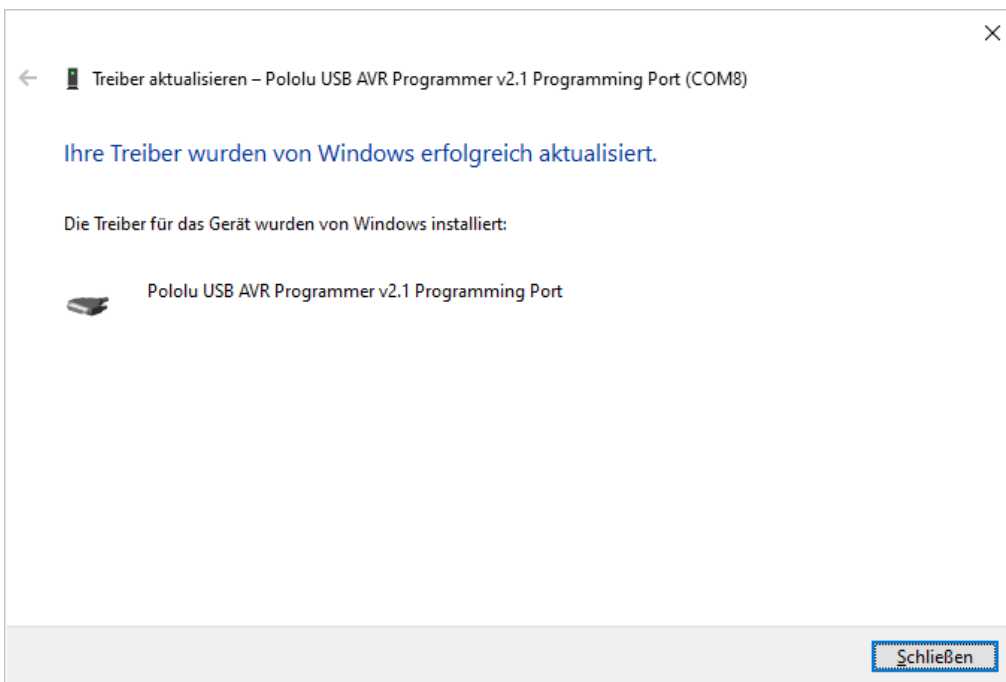
Im Geräte-Manager einen der beiden Einträge "Serielles USB-Gerät (COMXX)" mit der linken Maustaste aktivieren und dann mit der rechten Maustaste das Kontextmenü öffnen.



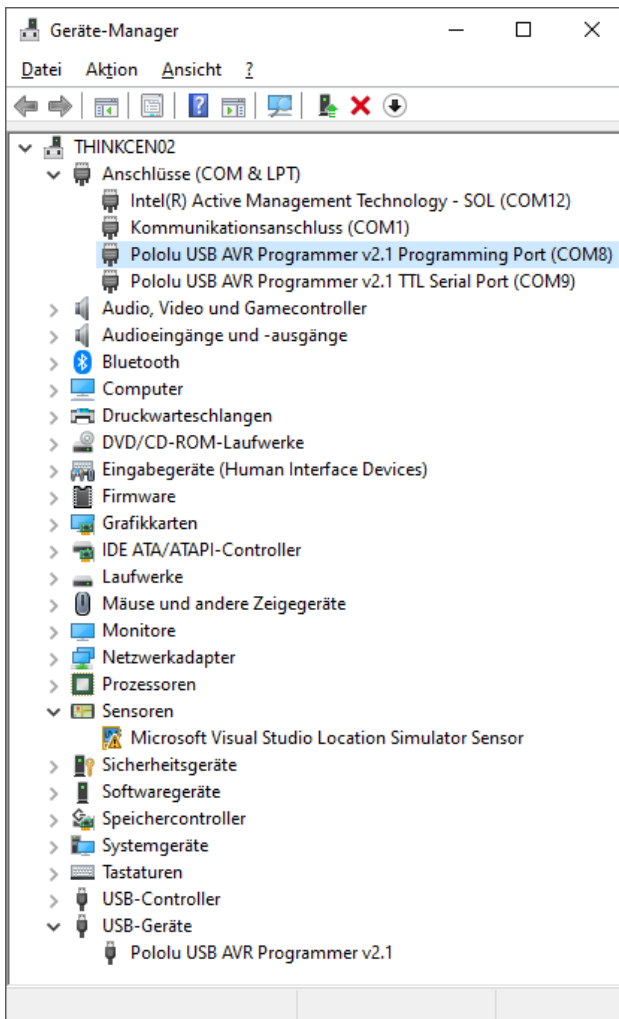
Den Pfad angeben, in dem die Treiber installiert wurden:

"C:\Program Files (x86)\Pololu"

"Computer nach Treibern durchsuchen" und dort das Kästchen "Unterordner einbeziehen" aktivieren.



Nach dieser Meldung wurde der Treiber ordnungsgemäß installiert.



3.3. Pololu AVR Programmer im RONPAS-Compiler

3.3.1. Neu Auswählen des Pololu AVR Programmer im RONPAS-Compiler

Ist der Pololu AVR Programmer noch nicht ausgewählt, muss dieser erst einmal ausgewählt werden.



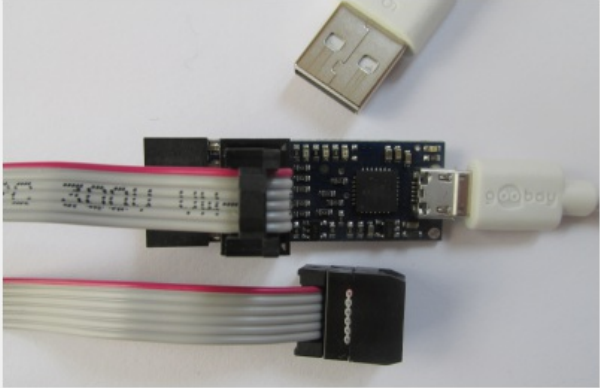
Nach dem Drücken des PROMMER-Button gelangt man in das PROMMER-Formular.

RONPAS64 - Version 1.5 - X

Device

PROMMER-Typ

Port



Flash | HL FUSES

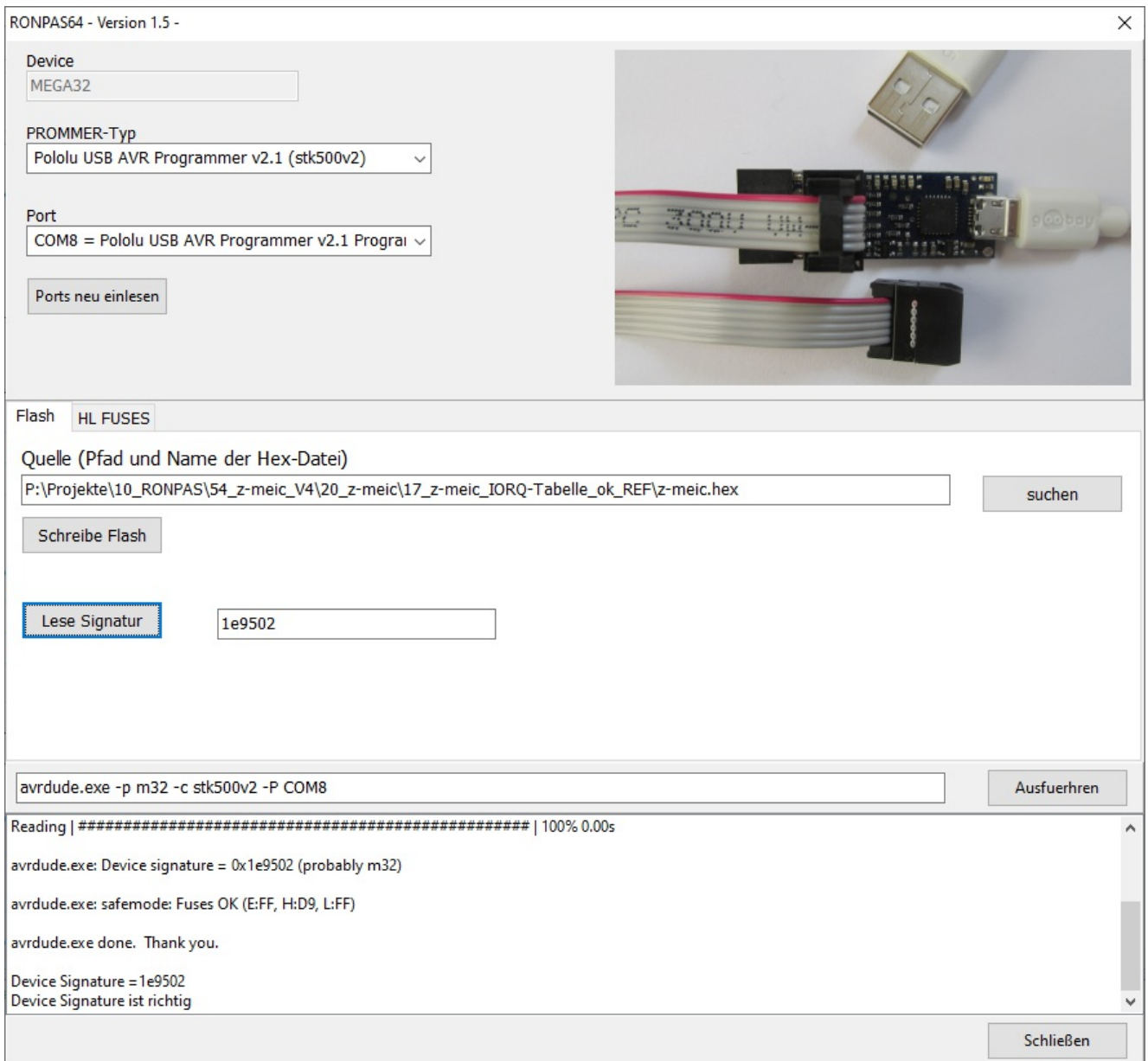
Quelle (Pfad und Name der Hex-Datei)

RGUI wurde gestartet ...
 ... es wurden 6 Programmer eingelesen
 COM-Port für Prommer wurde gefunden ...

Dort wird unter "PROMMER-Typ" der "Pololu USB AVR Programmer v2.1 (stk500v2)" ausgewählt. Zum sicheren Abspeichern der neuen Einstellungen das Formular "PROMMER" mit dem Button "Schliessen" beenden. Danach auch den RONPAS-Compiler beenden.

3.3.2. Pololu AVR Programmer am Board testen

Das zu programmierende Board mit dem ATMEGA muss eingeschaltet sein (+5V Betriebsspannung ist angeschlossen). Der Pololu AVR Programmer muss über die USB-A-Buchse an den PC angeschlossen und er sollte über den 6-pinigen Stecker mit dem Board verbunden sein. Erst jetzt wird der RONPAS-Compiler gestartet, da er zum Beginn des Starts die COM-Ports abfragt. Nach dem Start des RONPAS-Compilers noch einmal über den Button PROMMER in das PROMMER-Formular gehen.



Unter "Port" sollte ein COM-Port und die Bezeichnung "Pololu USB AVR Programmer v2.1 Programming Port" angezeigt werden (z.B. "COM8 = Pololu USB AVR Programmer v2.1 Programming Port").

Um die einwandfreie Funktion des Pololu-PROMMERS zu testen wird einfach der Button "Lese Signatur" mit der Maus angeklickt werden. Über das Programm AVRDUDE wird nun die Signatur des angeschlossenen ATMEGAs (z.B. ATMEGA32) gelesen und mit der abgespeicherten Signatur des erkannten Controllers verglichen.

Stimmt die gelesene Signatur mit der abgespeicherten Signatur überein, wird im unteren Anzeigefenster die Meldung:

"Device Signature ist richtig."

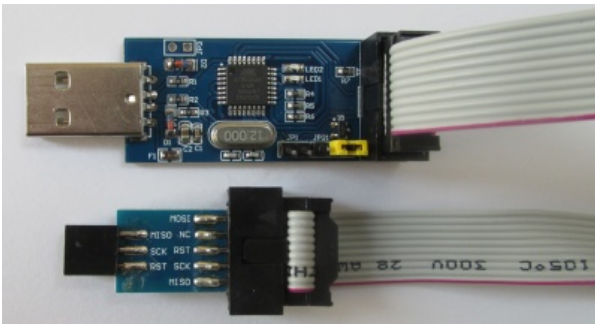
angezeigt. Damit ist der Pololu-Programmer einsatzbereit und das PROMMER-Formular kann wieder geschlossen werden.

3.3.3. AVR programmieren mit dem Pololu-Programmer



Der aktuelle Programmcode kann nun einfach mit dem Drücken des "Kompilieren und Programmieren-Button" übersetzt (PAS -> ASM -> HEX) und anschliessend in den angeschlossenen ATMEGA gebrannt werden.

4. USBasp AVR Programmer (6-polig und 10-polig ISP-Adapter)



Alle USBasp AVR Programmer basieren auf dem Projekt von Thomas Fischl:

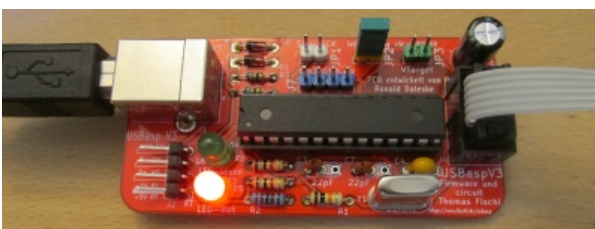
[USBasp AVR Programmer - https://www.fischl.de/usbasp/](https://www.fischl.de/usbasp/)

Es gibt den USBasp AVR Programmer bei vielen Lieferanten zu kaufen. Da der Preis (inklusive Versand) bei etwa 4 EUR beginnt (Lieferung aus China), wird der USBasp oft von Anfängern gekauft.

Leider ist für diesen Programmer die Installation eines (unter Windows) nicht signierten Treibers erforderlich. Das ist zwar möglich unterscheidet sich aber doch von der normalen Installation eines Treibers.

Preis: ab 4 EUR

Geschwindigkeit: 45 Sekunden (für die Hex-Datei des Referenzprojektes z-meic, etwa 31KB)



Eigenbau des USBasp AVR Programmers von Thomas Fischl. Siehe auch:

[USBasp Version 3 - https://www.daleske.de/projekte/](https://www.daleske.de/projekte/)

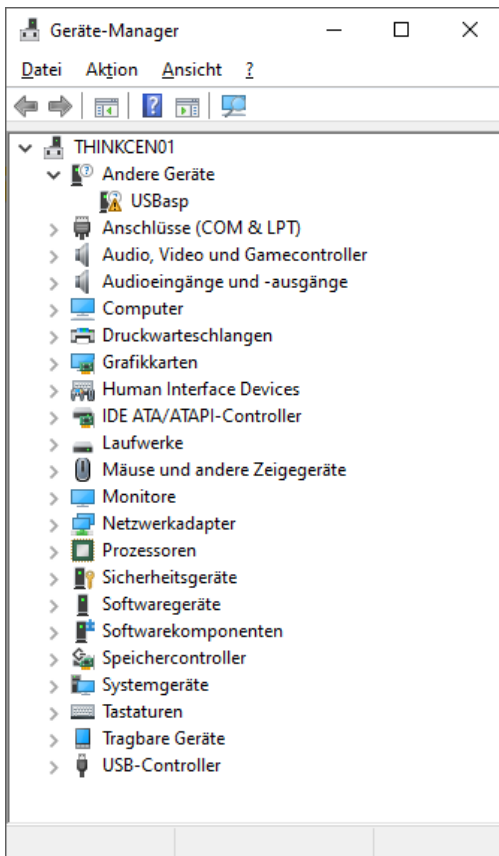
4.1. USBasp AVR Programmer - Link (Beispiele)

[USBasp AVR Programmer - DollaTek](#)

[USBasp AVR Programmer - WINGONEER](#)

4.2. USBasp - Treiber

4.2.1. USBasp ohne Treiber



Nach dem Einstecken des "USBasp" an den USB-A-Port des PC erscheint im Geräte manager (c:\Windows\System32\devmgmt.msc) unter "Andere Geräte" der Eintrag "USBasp". Der Programmierer wurde also grundsätzlich erkannt. Das gelbe Dreieck mit dem Ausrufezeichen kennzeichnet, dass für dieses Gerät kein passender Treiber installiert ist.

4.2.2. USBasp - Treiber und Hilfsprogramm downloaden

Für den USBasp AVR Programmierer wird der Windows USB drivers (libusb-win32 oder libusbK) benötigt. Dieser kann von verschiedenen Quellen heruntergeladen werden. In diesem Beispiel von:

<https://github.com/mcuae/libusb-win32>

mit dem Namen:

libusb-win32-master.zip

Da der Treiber für Windows nicht signiert ist, wird zur Installation des Treibers das Programm von Zadig benötigt. Es kann von:

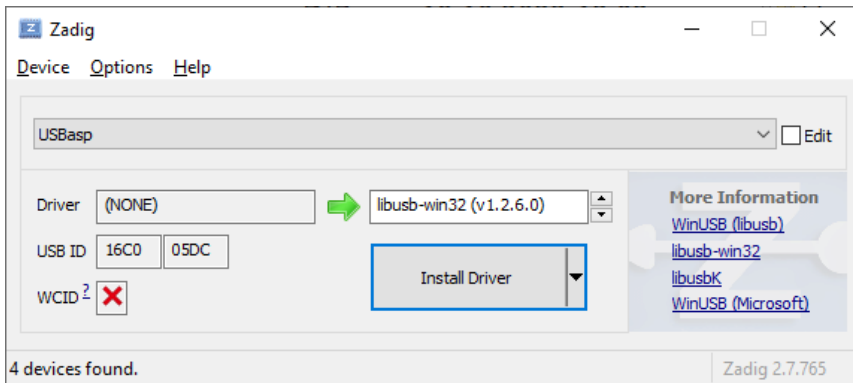
<https://zadig.akeo.ie/>

herunter geladen werden. Hier die Version vom 01.11.2021 mit dem Namen:

zadig-2.7.exe

4.2.3. USBasp - Treiber mit Hilfsprogramm installieren

Starten des Programms **zadig-2.7.exe**.



Unter **"Options"** -> **"list alle devices"** auswählen.

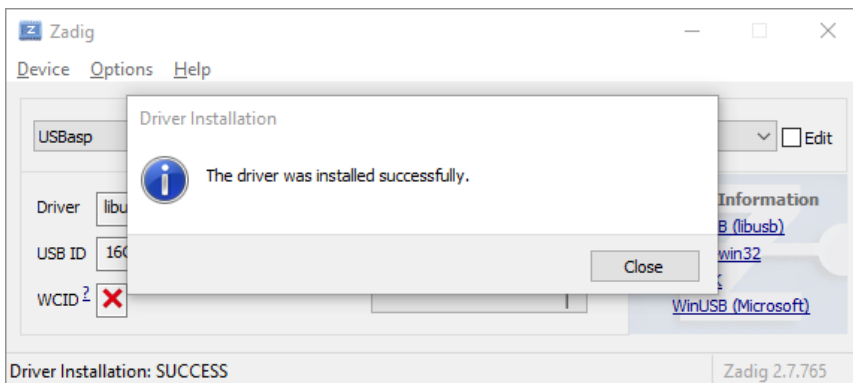
"USBasp" auswählen.

Unter **"Driver"** -> rechte Seite **"libusb-win32 (xxx)"** auswählen.

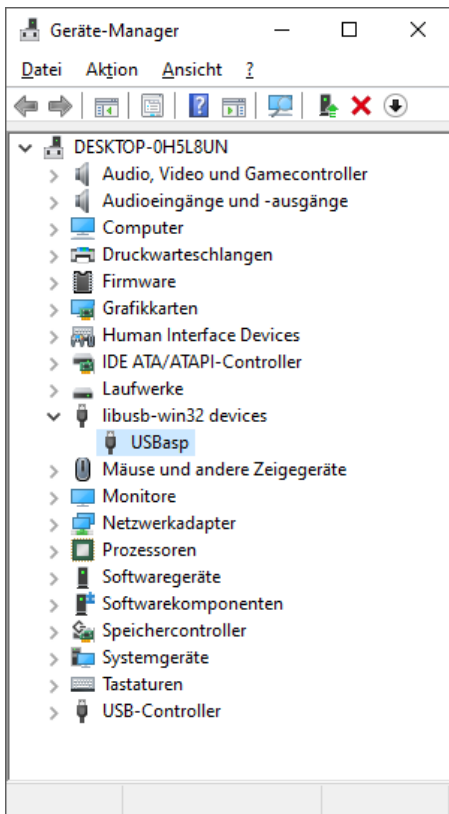
"Install Driver" drücken.

Die Installation des Treibers dauert einige Sekunden.

Wird dieses Fenster angezeigt:



dann wurde der Treiber ordnungsgemäß installiert.



4.3. USBasp AVR Programmer im RONPAS-Compiler

4.3.1. Neu Auswählen des USBasp AVR Programmer im RONPAS-Compiler

Ist der USBasp AVR Programmer noch nicht ausgewählt, muss dieser erst einmal ausgewählt werden.



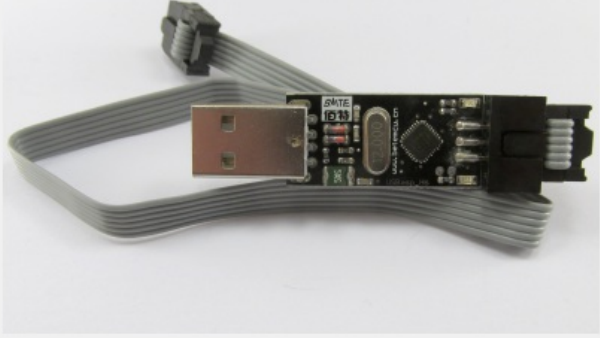
Nach dem Drücken des PROMMER-Button gelangt man in das PROMMER-Formular.

RONPAS64 - Version 1.5 - X

Device

PROMMER-Typ

Port



Flash HL FUSES

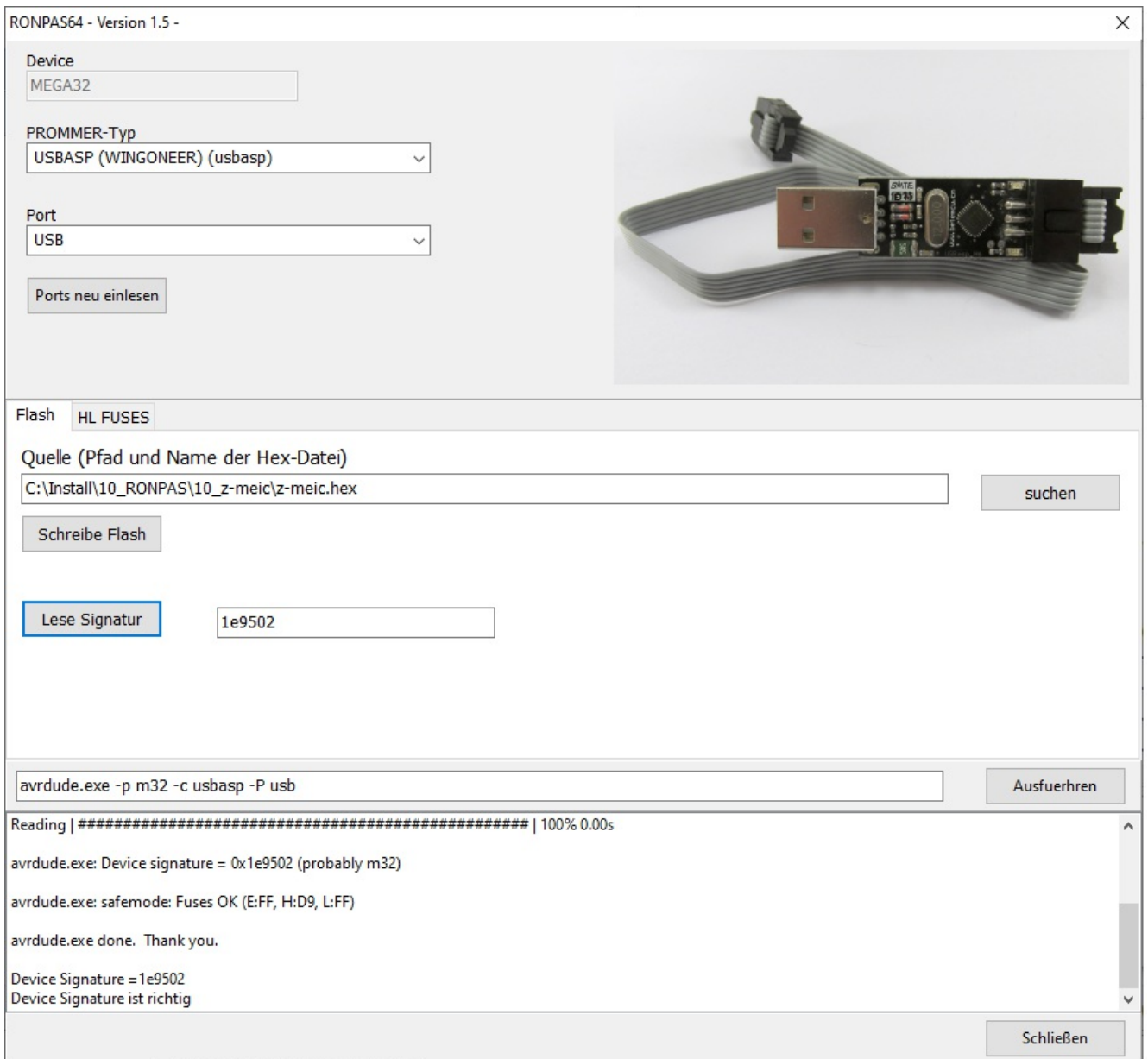
Quelle (Pfad und Name der Hex-Datei)

RGUI wurde gestartet ...
 ... es wurden 6 Prommer eingelesen

Dort wird unter "PROMMER-Typ" der "USBASP (WINGONEER) (usbasp)" ausgewählt. Zum sicheren Abspeichern der neuen Einstellungen das Formular "PROMMER" mit dem Button "Schliessen" beenden. Danach auch den RONPAS-Compiler beenden.

4.3.2. USBasp AVR Programmer am Board testen

Das zu programmierende Board mit dem ATMEGA muss eingeschaltet sein (+5V Betriebsspannung ist angeschlossen). Der USBasp AVR Programmer muss über die USB-A-Buchse an den PC angeschlossen und er sollte über den 6-poligen Stecker mit dem Board verbunden sein. Erst jetzt wird der RONPAS-Compiler gestartet, da er zum Beginn des Starts die USB-Ports abfragt. Nach dem Start des RONPAS-Compilers noch einmal über den Button PROMMER in das PROMMER-Formular gehen.



Unter "Port" sollte "USB" angezeigt werden.

Um die einwandfreie Funktion des USBasp AVR Programmiers zu testen wird einfach der Button "Lese Signatur" mit der Maus angeklickt werden. Über das Programm AVRDUDE wird nun die Signatur des angeschlossenen ATMEGAs (z.B. ATMEGA32) gelesen und mit der abgespeicherten Signatur des erkannten Controllers verglichen.

Stimmt die gelesene Signatur mit der abgespeicherten Signatur überein, wird im unteren Anzeigefenster die Meldung:

"Device Signature ist richtig."

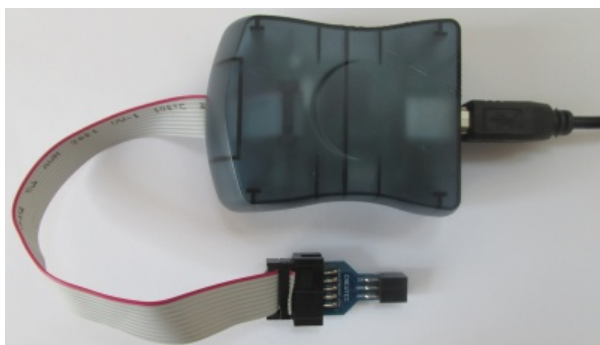
angezeigt. Damit ist der USBasp AVR Programmer einsatzbereit und das PROMMER-Formular kann wieder geschlossen werden.

4.3.3. AVR programmieren mit dem USBasp AVR Programmer



Der aktuelle Programmcode kann nun einfach mit dem Drücken des "Kompilieren und Programmieren-Button" übersetzt (PAS -> ASM -> HEX) und anschliessend in den angeschlossenen ATMEGA gebrannt werden.

5. AVR ISP mkII Programmier (Atmel kompatibel)



Der AVR ISP mkII Programmier ist ein Clone des Original ATMEL AVR ISP mk2 Programmiers (den es nicht mehr zu kaufen gibt). Während die oben vorgestellten Programmier schon seit Jahren bei mir im Einsatz sind, habe ich den AVR ISP mkII Programmier erst vor kurzem gekauft. Das Original war mir immer zu teuer und nun wollte ich diesen Programmier als Clone einmal testen. Die ersten Tests liefen sehr schlecht und ich konnte den AVR ISP mkII Programmier nicht einfach und schnell zum Laufen bekommen. Da ich immer noch die bewährten Programmier hatte, wurde die Einrichtung immer wieder verschoben.

Hier aber nun die ausführliche Anleitung zur Nutzung des AVR ISP mk2 USB Programmiers.

Auch für diesen Programmier wird ein Treiber benötigt, den man erst einmal suchen muss, da die chinesischen Hersteller des Clones in der Regel keine Angaben zur Installation des Treibers machen.

Preis etwa 20 EUR.

Geschwindigkeit: 26 Sekunden (für die Hex-Datei des Referenzprojektes z-meic, etwa 31KB) Dies ist der kleinste Wert und somit der schnellste Programmier in Test.

5.1. AVR ISP mkII Programmier - Link (Beispiele)

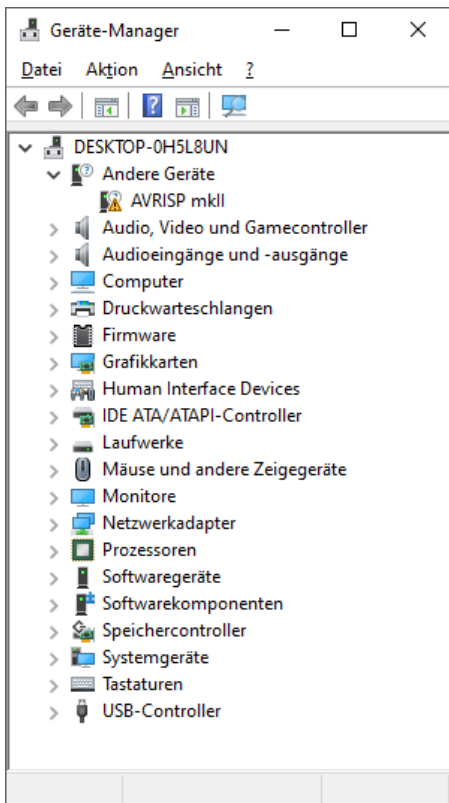
[AVR ISP mkII - ARCELI](#)

[AVR ISP mkII - DollaTek](#)

[AVR ISP mkII - Waveshare](#)

5.2. AVR ISP mkII Programmier - Treiber

5.2.1. AVR ISP mkII Programmier ohne Treiber



Nach dem Einstecken des "AVR ISP mkII" an den USB-A-Port des PC erscheint im Gerätemanager (c:\Windows\System32\devmgmt.msc) unter "Andere Geräte" der Eintrag "AVRISP mkII". Der Programmierer wurde also grundsätzlich erkannt. Das gelbe Dreieck mit dem Ausrufezeichen kennzeichnet, dass für dieses Gerät kein passender Treiber installiert ist.

5.2.2. AVR ISP mkII - Treiber downloaden

Der Treiber kann von der Internetseite "Elektronik Hannes Jochriem" heruntergeladen werden:

<https://www.ehajo.de/dokuwiki/artikel:libusb-mkiiclone>

Auf dieser Internetseite kann der Treiber unter der Bezeichnung des Links:

Signierter Treiber für avr isp mkII Klon

herunter geladen werden.

In meinem Fall hatte der Treiber den Namen:

ehajo_avrisp_mkii.zip

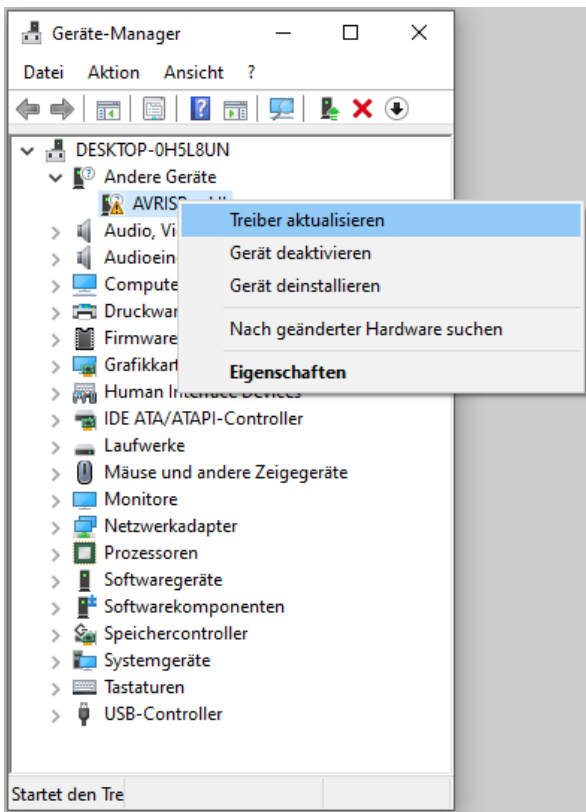
ZIP-Datei entpacken. Hier unter:

"c:\InstalleHaJo avrisp_mkII"

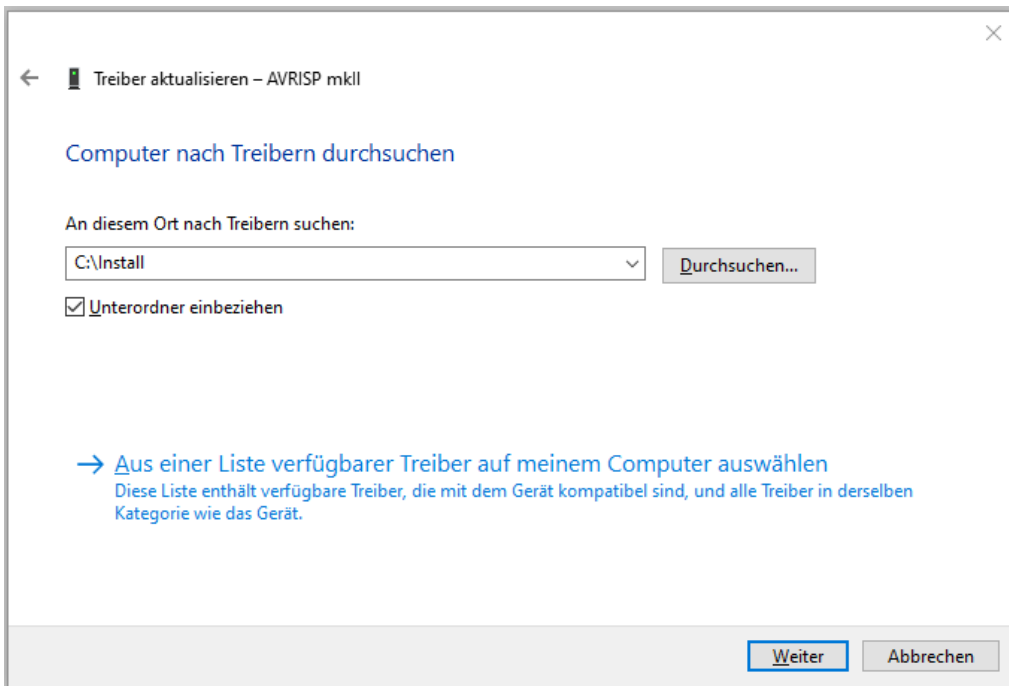
Die Installation des Treibers ist auf dieser Internetseite ausführlich beschrieben.

5.2.3. AVR ISP mkII - Treiber installieren

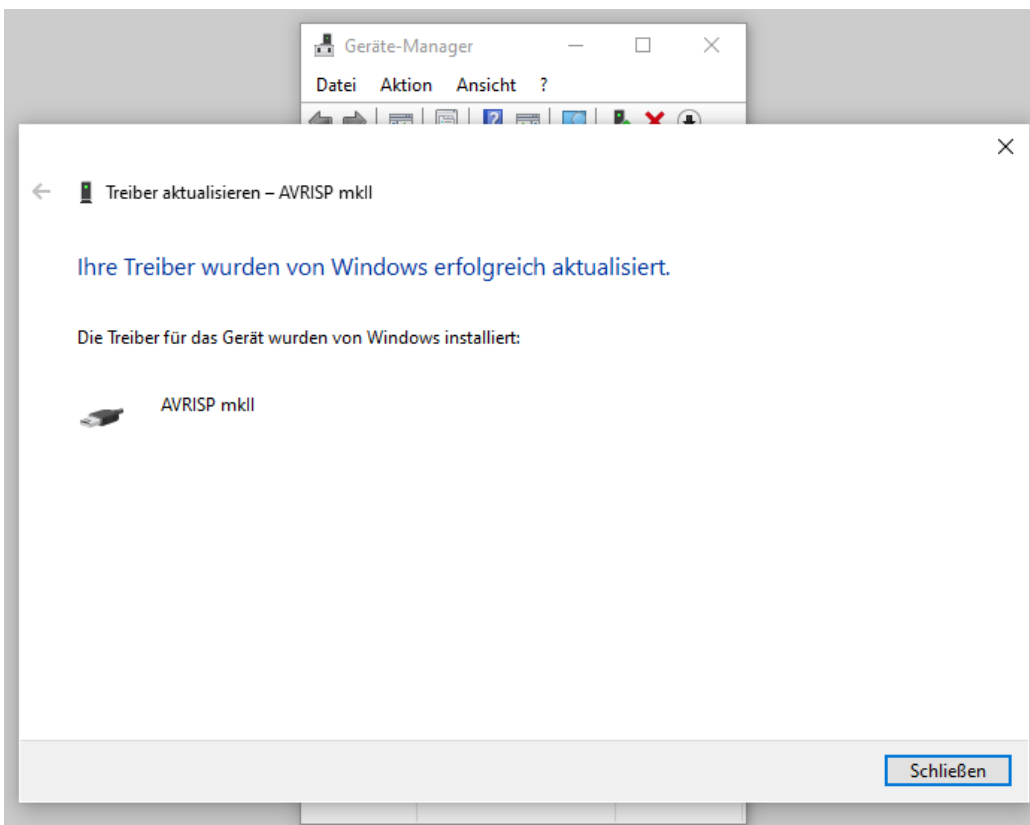
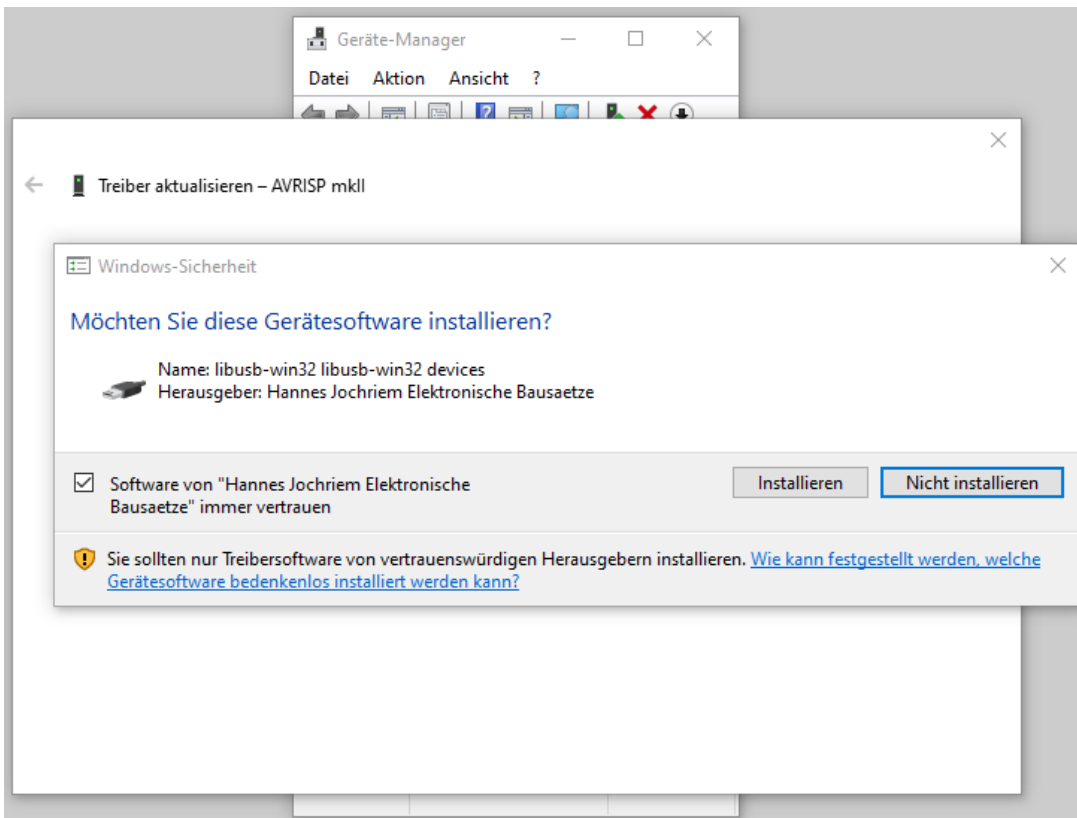
Im Geräte-Manager den Eintrag "AVRISP mkII" mit der linken Maustaste aktivieren und dann mit der rechten Maustaste das Kontextmenü öffnen und "Treiber aktualisieren".



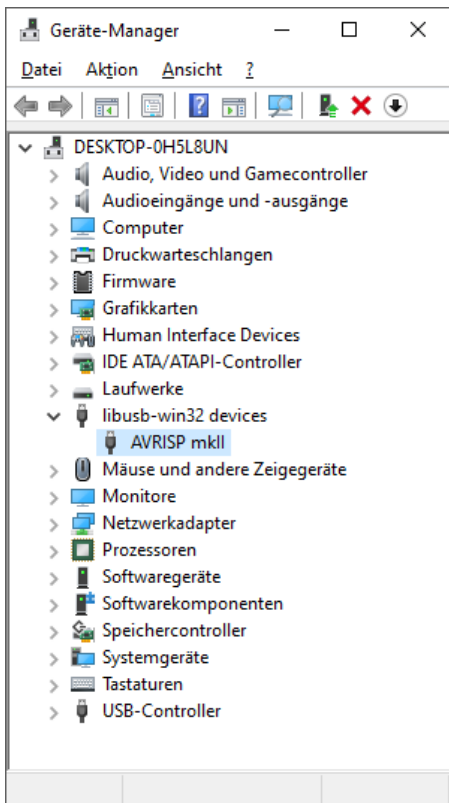
"Computer nach Treibern durchsuchen" und dort das Kästchen "Unterordner einbeziehen" aktivieren.



"Möchten Sie diese Gerätesoftware installieren?" und auf "Installieren" klicken.



Nach dieser Meldung wurde der Treiber ordnungsgemäß installiert.



5.3. AVR ISP mkII im RONPAS-Compiler

5.3.1. Neu Auswählen des AVR ISP mkII Programmers im RONPAS-Compiler

Ist der AVR ISP mkII Programmierer noch nicht ausgewählt, muss dieser erst einmal ausgewählt werden.



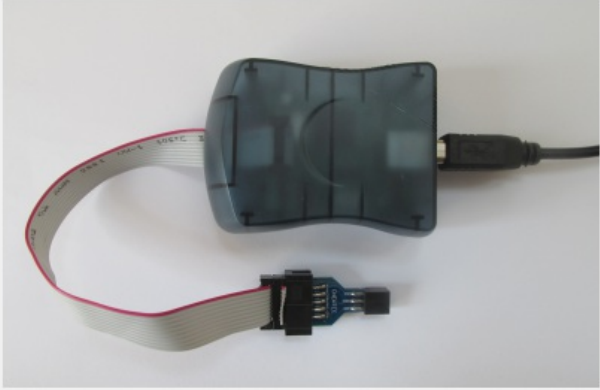
Nach dem Drücken des PROMMER-Buttons gelangt man in das PROMMER-Formular.

RONPAS64 - Version 1.5 - X

Device

PROMMER-Typ

Port



Flash

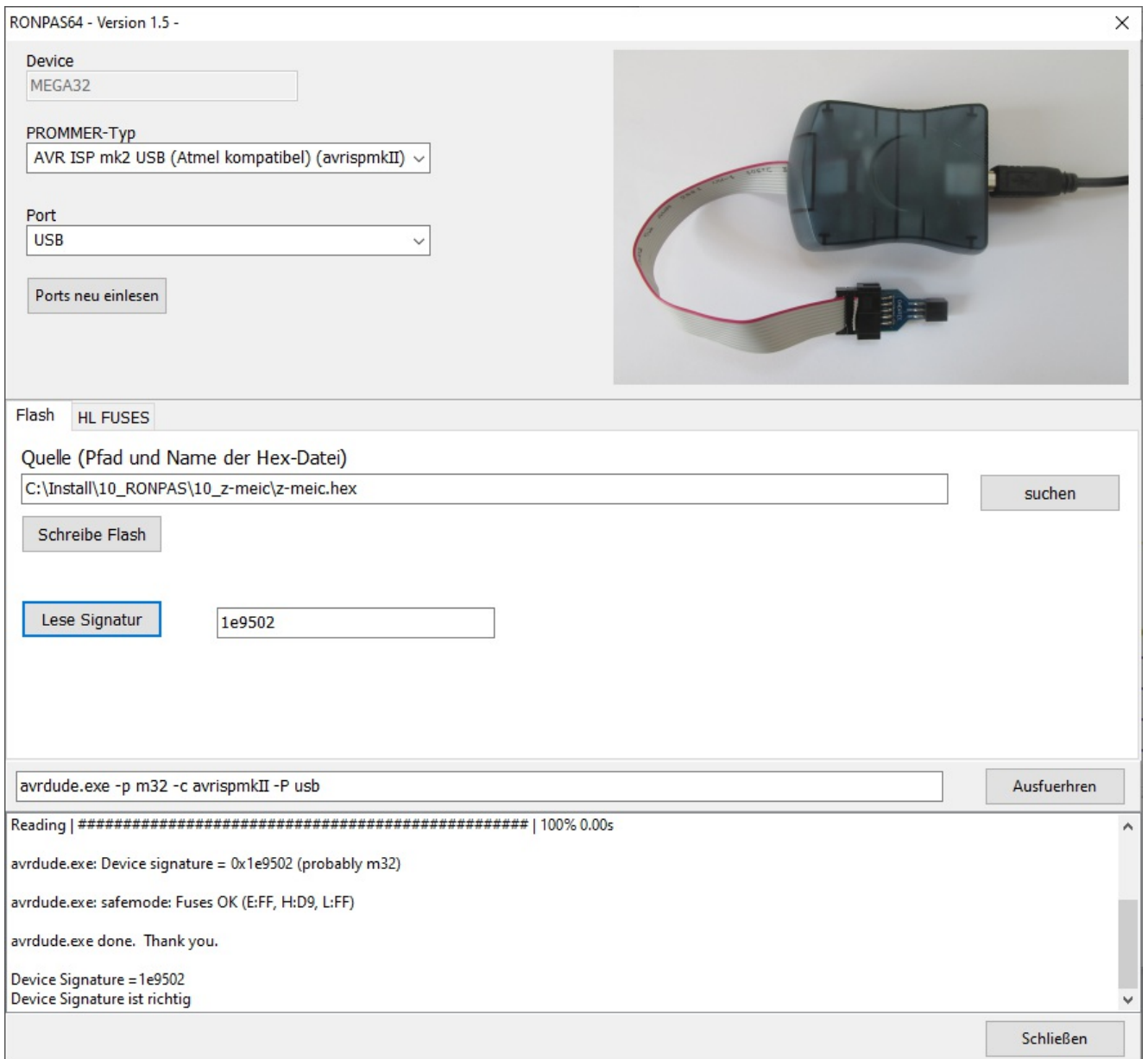
Quelle (Pfad und Name der Hex-Datei)

RGUI wurde gestartet ...
 ... es wurden 6 Programmer eingelesen

Dort wird unter "PROMMER-Typ" der "AVR ISP mk2 USB (Atmel kompatibel) (avrismkII)" ausgewählt. Zum sicheren Abspeichern der neuen Einstellungen das Formular "PROMMER" mit dem Button "Schliessen" beenden. Danach auch den RONPAS-Compiler beenden.

5.3.2. AVR ISP mkII Programmer am Board testen

Das zu programmierende Board mit dem ATMEGA muss eingeschaltet sein (+5V Betriebsspannung ist angeschlossen). Der AVR ISP mkII Programmer muss über die USB-A-Buchse an den PC angeschlossen und er sollte über den 6-poligen Stecker mit dem Board verbunden sein. Erst jetzt wird der RONPAS-Compiler gestartet, da er zum Beginn des Starts die COM-Ports abfragt. Nach dem Start des RONPAS-Compilers noch einmal über den Button PROMMER in das PROMMER-Formular gehen.



Unter "Port" sollte "USB" angezeigt werden.

Um die einwandfreie Funktion des AVR ISP mkII Programmers zu testen wird einfach der Button "Lese Signatur" mit der Maus angeklickt werden. Über das Programm AVRDUDE wird nun die Signatur des angeschlossenen ATMEGAs (z.B. ATMEGA32) gelesen und mit der abgespeicherten Signatur des erkannten Controllers verglichen.

Stimmt die gelesene Signatur mit der abgespeicherten Signatur überein, wird im unteren Anzeigefenster die Meldung:

"Device Signature ist richtig."

angezeigt. Damit ist der AVR ISP mkII Programmer einsatzbereit und das PROMMER-Formular kann wieder geschlossen werden.

5.3.3. AVR programmieren mit dem AVR ISP mkII Programmer



Der aktuelle Programmcode kann nun einfach mit dem Drücken des "Kompilieren und Programmieren-Button" übersetzt (PAS -> ASM -> HEX) und anschliessend in den angeschlossenen ATMEGA gebrannt werden.

Startseite	Programmier- Projekte	RONPAS-Compiler
----------------------------	---	------------------------

Copyright © 2023 Ronald Daleske