

# 0 Hilfsmittel

## Student Group

| First Name | Surname | Matrikel Nr. |
|------------|---------|--------------|
|            |         |              |
|            |         |              |
|            |         |              |

## Table of Contents

|                                                       |   |
|-------------------------------------------------------|---|
| <b>0 Hilfsmittel</b>                                  | 2 |
| <b>0.1 SimulIDE</b>                                   | 2 |
| Tipps für die Verwendung von Simulide - Konfiguration | 2 |
| <b>0.2 Microchip Studio</b>                           | 3 |
| <b>Installation</b>                                   | 3 |
| <b>Installationsanleitung</b>                         | 3 |
| <b>Erster Test der Verbindung zum AVR-USB-Progi</b>   | 3 |
| <b>wichtige Einstellungen</b>                         | 5 |
| <b>Tipps</b>                                          | 5 |
| <b>0.3 TortoiseGit</b>                                | 5 |
| <b>MEXLE Git</b>                                      | 5 |
| Installation und Verbinden mit dem MEXLE Git          | 5 |
| Installation der Software auf Ihrem Rechner           | 5 |
| Vorbereitung des Rechners: Installationen und Ordner  | 5 |
| Installation der Software auf Ihrem Rechner           | 6 |
| <b>Mögliche Fehler</b>                                | 7 |

# 0 Hilfsmittel

## 0.1 SimulIDE

Das kostenlose Elektronik-Simulationsprogramm SimulIDE finden Sie in [ILIAS](#) unter (DE/EN) Software.

Alternativ ist das Programm auch über die [Hersteller-Seite](#), herunterladbar. Dort muss dann ein "Obulus" angegeben werden, welcher auch 0€ sein kann.

Im Gegensatz zu TINA TI hat SimulIDE einen anderen Fokus und damit folgende Vor- und Nachteile.

Vorteile:

- Microcontroller wird gut simuliert (u.a. von Atmel Chips). Das ist in TINA TI nicht möglich.
- Die Microcontroller sind auch programmierbar. D.h. man kann Microcontroller-geeignete Source-File ([hex-File](#)) nutzen.
- Interaktion Software und Hardware ist möglich.

Nachteile:

- Simulation von verschiedenen Elektronikkomponenten nur vereinfacht implementiert (z.B. Operationsverstärker oder FETs)
- Die Software wird noch sehr stark weiterentwickelt. D.h. es ist gut die oben angegebene Version zu nutzen, um Kompatibilitätsprobleme zu vermeiden.

Zur Einführung in SimulIDE hilft es die [Playlist des Entwicklers](#) anzusehen.

### Tipps für die Verwendung von Simulide - Konfiguration

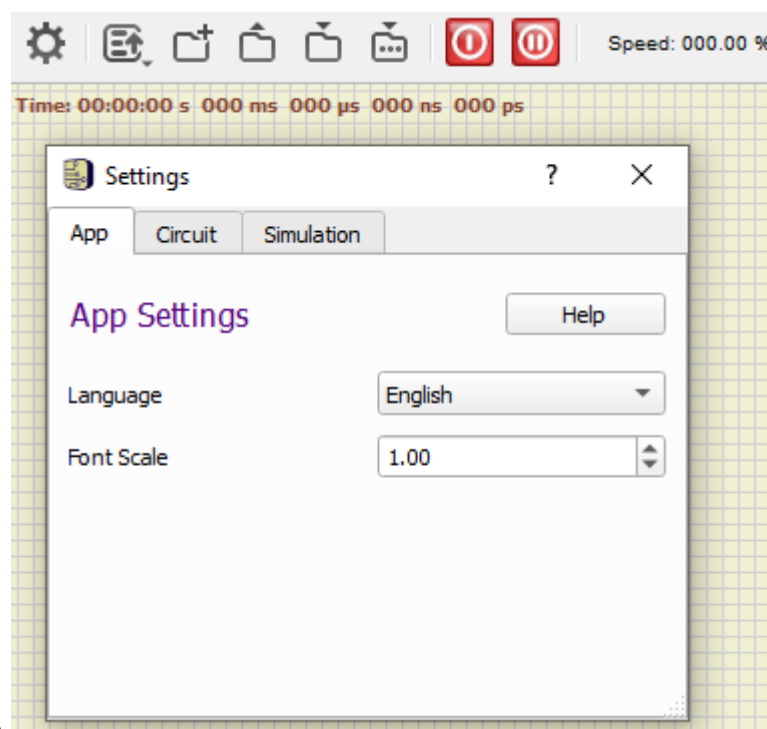


Fig. 1: Bild zur Konfiguration

- Falls der Text nach dem Öffnen des Programms zu klein dargestellt wird, kann eines der folgenden Vorgehen helfen:
  - Im Programm SimulIDE nach dem Öffnen auf das Zahnrad klicken. Den Reiter App auswählen. Dort bei Font Scale beispielsweise 2,0 eintragen und das Programm neu starten.
  - Im Explorer rechtsklick » Eigenschaften » Kompatibilität » Hohe DPI Einstellungen ändern » Hohe DPI-Skalierung überschreiben » Häkchen setzen » "System" auswählen. (ggf. "System (Erweitert)" wählen)

## 0.2 Microchip Studio

Microchip Studio ist eine Programmierumgebung, um aus C oder C++ ein Microcontroller-geeignetes Source-File ([hex-File](#)) zu erstellen.

### Installation

### Installationsanleitung

Die aktuelle Version des Programms finden Sie auf der [Microchip-Homepage](#).

1. Falls nach dem Drücken des Buttons Download Microchip Studio nichts passiert: einfach herunterscrollen zu Downloads and Documents
2. Bei schnellen Internetverbindungen kann der Web Installer gewählt werden, bei langsamen lässt sich mit dem Offline Installer das gesamte Paket vor der Installation herunterladen.
3. Bei der Installation ist als Architektur nur "AVR" notwendig.
4. "Advanced Software Framework and Example Projects" sind nicht notwendig
5. Die Installation dauert je nach Geschwindigkeit des Rechners und der Internetverbindung etwa 5 Minuten
6. Falls nicht schon installiert, wird bei der Installation auch Visual Studio mit installiert
7. Die Frage nach Lizenzen für den C compiler ist mit Next > zu überspringen
8. Weiterhin kann es sein, dass verschiedene Geräte-Treiber mit installiert werden. Diese ermöglichen das Beschreiben der Chips mit einem Programmiergerät.
9. Öffnen Sie im Anschluss direkt Microchip Studio, damit Sie die ersten, wichtigen Einstellungen vornehmen können.

### Erster Test der Verbindung zum AVR-USB-Progi

Führen Sie folgende Schritte durch, um einen ersten Test mit der Hardware durchzuführen.

1. Vorbereitungen
  1. Verbinden Sie das MiniMEXLE
    1. mit dem Proggi (schwarze Box) per Flachbandkabel
    2. mit der Steckdose per Ladekabel mit dem Hohlstecker
  2. Öffnen Sie Microchip Studio (z.B. durch Druck auf die <WIN> Taste und Eingabe von Microchip Studio, sowie Druck auf <Return>).
  3. Laden Sie die Datei [5.\\_menufuehrung.hex](#) herunter.
  4. Verbinden Sie nach dem Öffnen das Proggi per USB-Kabel mit dem PC / Laptop. Dieser Schritt muss vor den folgenden geschehen!
  5. Es sollte am Proggi die Grüne LED für die USB-Kommunikation leuchten.
2. USB-Verbindung eintragen

1. Gehen Sie ohne Erstellen eines Projekts, direkt auf Tools » Add Target...
2. Im dargestellten Fenster sollten Sie beim bisher leer angezeigten Dropdown-Menu als Tool STK500 auswählen.
3. Bei der Auswahl des "Serial Ports" sollten Sie den ersten verfügbaren auswählen z.B. COM3 und bestätigen Sie die Auswahl mit Apply.  
Sollte hier keine Port angezeigt werden, prüfen Sie, ob die Verbindung mit dem USB-Progi erfolgreich war.  
Falls die grüne LED leuchtet und das USB Kabel korrekt eingesteckt ist, gehen Sie zu den Hinweisen unter 4.d.
3. Erster Verbindungsversuch
  1. Wählen Sie im Menü Tools » Device Programming aus.
  2. Wählen Sie bei Tools die entsprechende USB-Verbindung aus, z.B. STK500 COM3.
  3. Bei Device müssen Sie ATmega88 auswählen, Sie können dazu 88 eingeben und im Dropdownmenü dann den korrekten Wert anklicken.
  4. Beim Interface sollte nun ISP stehen
  5. Drücken Sie nun auf Apply.
4. Im Fehlerfall:
  1. Falls es nach Druck auf Apply länger dauert, hat die Verbindung nicht geklappt. Es wird dann eine Fehlermeldung Unable to connect tool STK500 (COMx) angezeigt.
  2. Die Fehlermeldung können Sie quittieren, jedoch dauert es danach nochmals einige Sekunden, bis das Programm wieder korrekt reagiert.
  3. Schließen Sie dann das Fenster Device Programming. Gehen Sie in dieser Anleitung zurück zu Punkt 2. und versuchen Sie den nächsten Serial Port.  
Merken Sie sich dabei die Nummer des Ports, da diese unter 3.b. benötigt wird.
  4. Wenn alle vorhandenen COM-Ports durchgetestet wurden, oder zu Beginn keiner angezeigt wurde, kann die Installation eines Virtual COM Port Drivers helfen. Einen solchen finden Sie unter [FTDI](#).
5. Im Gutfall:
  1. Es sollte nun etwas mehr im Fenster Device Programming sichtbar worden sein.
  2. Schließen Sie vor den nächsten Schritten das MiniMEXLE (= Platine mit Display und Knöpfen) mit dem Flachbandkabel an das Progi und mit dem Hohlstecker an das Netzteil an. Stecken Sie das Netzteil auch in eine Steckdose
  3. Drücken Sie nun auf Read oben neben Device Signature.
  4. Es sollte eine hexadezimale Zahl 0x... erscheinen. Ihr Rechner kann nun für das Flashen eine Verbindung zum MiniMEXLE herstellen.
  5. Gehen Sie auf im Fenster Device Programming auf den Punkt Memories. Fügen Sie unter dem Punkt Flash den Pfad zur heruntergeladenen Datei 5.\_menufuehrung.hex ein. Sie können dazu auch über die ... rechts daneben den Pfad auswählen
  6. Drücken Sie auf Program
  7. Auf dem Minimexle sollte nun eine Anzeige  
- Experiment 5 -  
Programm Menu  
und dann  
Main Level  
P1 P2 P3 P4  
erscheinen
  8. Damit haben Sie die Verbindung und das Flashen erfolgreich getestet

## wichtige Einstellungen

- Nutzen Sie die **Anzeige von Zeilennummern**: Tools » Options » Text Editor » All languages » General » Line numbers
- Sobald Sie das erste Projekt bearbeiten: Deaktivieren Sie unbedingt die Compiler Optimierung. Dies geht mit folgenden Schritten:
  - Menu Project » <ProjectName> Properties... » AVR/GNU Compiler » Optimization
  - Das Optimization level sollte hier auf None ( -O0 ) stehen
- Ich empfehle ATMLE Studio in der Sprache "English" zu nutzen. Damit sind die Hinweise in diesem Kurs korrekt zu finden. Falls Sie versehentlich (z.B. bei der Installation) die deutsche Sprache eingestellt hatten, können Sie es unter Werkzeuge » Optionen » environment » international settings » Language korrigieren.

## Tipps

- Falls auf der rechten Seite der **Solution Explorer** (Anzeige der Dateien im Projekt) nicht vorhanden ist, so finden Sie diesen unter View » Solution Explorer (<CTL>+<ALT>+<L>)

## 0.3 TortoiseGit

### MEXLE Git

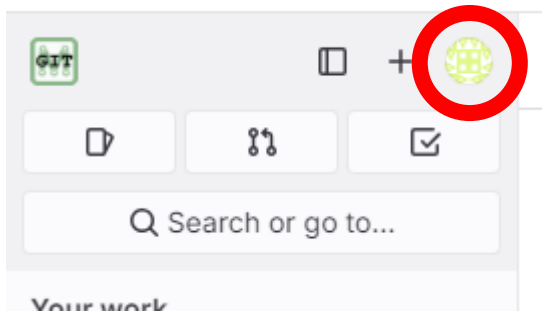
#### Installation und Verbinden mit dem MEXLE Git

#### Installation der Software auf Ihrem Rechner

1. Laden Sie Git von folgender Seite herunter: <https://git-scm.com/download/win> » "Standalone Installer"  
Git bietet die Möglichkeit mit dem GitLab Server der Hochschule oder mit GitHub in Kontakt zu treten
2. Installieren Sie das Git (alle mit "Ok" bzw "Weiter" bestätigen)
3. Laden Sie TortoiseGit von folgender Seite herunter: <https://tortoisegit.org/download/>  
tortoiseGit bindet den Explorer direkt an die Services des Git an. Damit ist Git direkt in den Explorer eingebunden.
4. Installieren Sie das TortoiseGit (alle Hinweise mit "Ok" bestätigen)

#### Vorbereitung des Rechners: Installationen und Ordner

1. Melden Sie sich bei GitLab mit den Hochschul-Credentials an: <https://git.mexle.org/>
2. Gehen Sie zu User settings » Preferences » Password:  
z.B. über folgenden Link: [https://git.mexle.org/-/user\\_settings/password/edit](https://git.mexle.org/-/user_settings/password/edit)



3. Geben Sie ein Passwort mit mindestens 16 Zeichen ein und merken Sie sich dieses Passwort. Dieses Passwort ermöglicht die Authentifizierung auf GitLab. Bitte nutzen Sie ein neues Passwort!

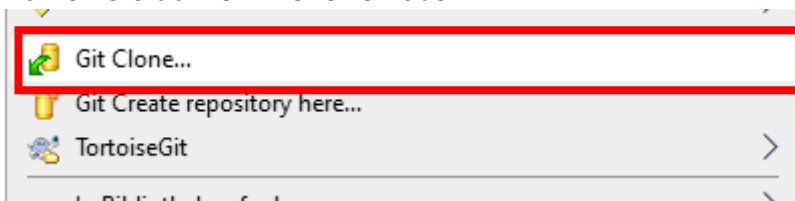
## Installation der Software auf Ihrem Rechner

Die Abfolge ist wie folgt:

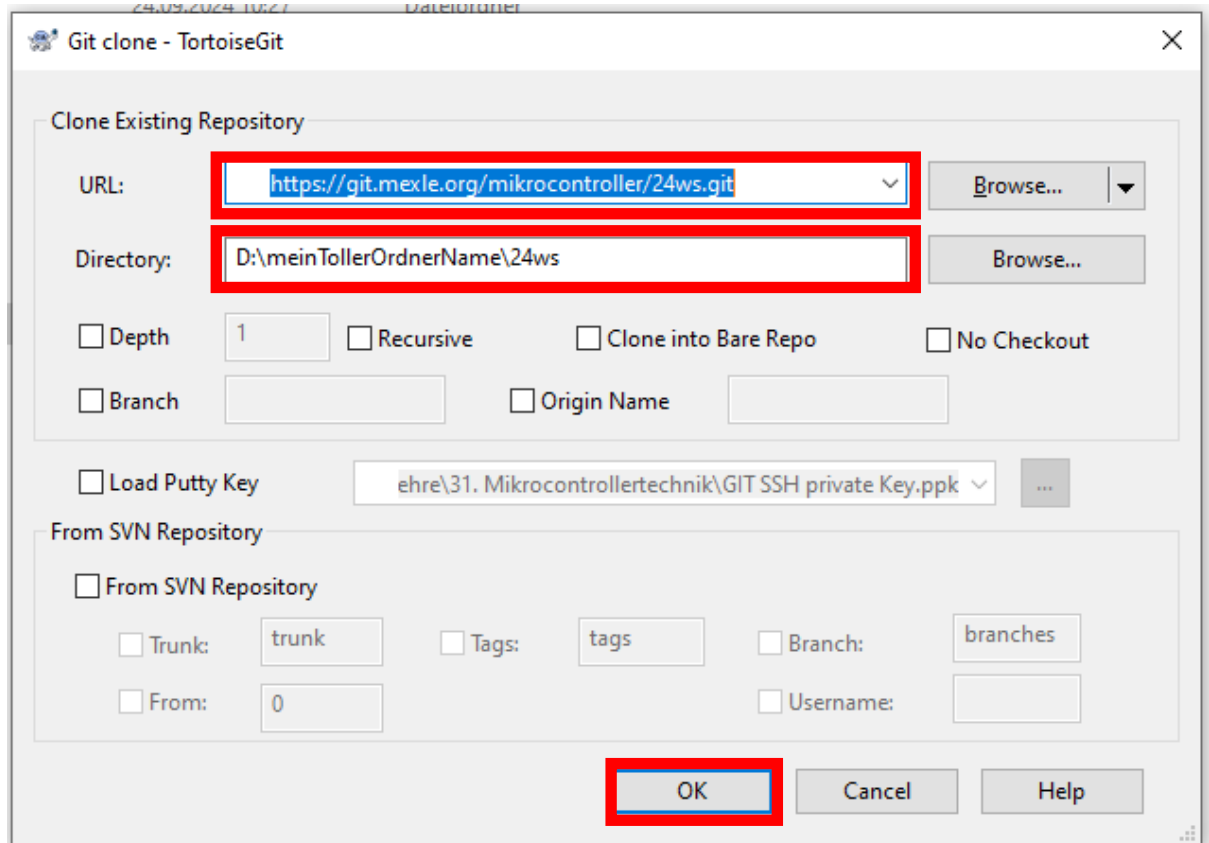


1. Erst müssen Sie sich in GitLab angemeldet haben
2. Dann müssen Sie einem Betreuer (Prof. Fischer, Ralf Ziegler) per Mail bescheid geben, damit dieser Sie zum Projekt zuordnet
3. Erst dann sind die folgenden Punkte durchführbar.

1. Gehen Sie in den (Windows) Explorer und legen Sie einen neuen Ordner für die Vorlesung an, z.B. Mikrocontroller oder Elektronik Labor
2. Klicken Sie mit der rechten Maustaste auf den Ordner, um in das Kontextmenu zu kommen. Wählen Sie dort Git Clone... aus.



3. Im erscheinenden GitClone Fenster sollten Sie Folgendes eingeben:
  1. als URL: `https://git.mexle.org/[Fach]/[Semester].git`  
also z.B. `https://git.mexle.org/mikrocontroller/24ws.git` oder `https://git.mexle.org/elektronik/24ws.git` für das Wintersemester 2024
  2. als Directory sollte der ausgewählte Ordner eingetragen sein
  3. Klicken Sie nun auf Ok



4. Im Anschluss sollten Sie eine Fehlermeldung erhalten, da noch das Passwort fehlt.  
Hier ist nun das vorher gewählte Passwort einzugeben.  
Dies müssen Sie auch nur einmalig machen.
5. Der Download sollte nun klappen und es sollten alle Ordner heruntergeladen werden

## Mögliche Fehler

1. fatal: detected dubious ownership in repository at [...] is on a filesystem that does not record ownership  
Das Problem ist, dass der Ordner auf einem Laufwerk liegt, welches keine Benutzerzuordnung erlaubt (z.B. ein USB-Stick) .  
Die Lösung wird gleich mitgeliefert:
  1. Rechtsklick im Explorer auf den entsprechenden Ordner (z.B. 24WS) » Open Git Bash here » Es öffnet sich eine Text-Konsole
  2. Fügen Sie Folgendes ein `git config --global --add safe.directory D:/GitLab/elektronik/24WS` (ändern Sie ggf. elektronik in mikrocontroller und das Semester) und bestätigen Sie mit Return

From:  
<https://wiki.mexle.te.hs-heilbronn.de/> - **MEXLE Wiki**

Permanent link:  
[https://wiki.mexle.te.hs-heilbronn.de/mikrocontrollertechnik/0\\_hilfsmittel?rev=1729453999](https://wiki.mexle.te.hs-heilbronn.de/mikrocontrollertechnik/0_hilfsmittel?rev=1729453999)

Last update: **2024/10/20 21:53**

