

0 Hilfsmittel

Student Group

First Name	Surname	Matrikel Nr.

Table of Contents

0 Hilfsmittel	2
0.1 SimulIDE	2
Tipps für die Verwendung von Simulide - Konfiguration	2
0.2 Microchip Studio	3
Installation	3
Installationsanleitung	3
für Mac-User	3
Erster Test der Verbindung zum AVR-USB-Progi	4
wichtige Einstellungen	5
Tipps	5
Häufige Fehler	5
I2C	6
0.3 MEXLE GitLab	6
Installation der Software auf Ihrem Rechner	6
Anmeldung beim MEXLE GitLab	6
Installation der Software auf Ihrem Rechner	7
Upload / Commit von Dateien	8
Hinweise und Mögliche Fehler	9

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 rechtecklick » 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.

für Mac-User

Speziell für Mac User bietet sich folgendes Vorgehen an:

1. Öffnen Sie einen Browser.
2. Stellen Sie eine VPN Verbindung über eduVPN her.
3. Gehen Sie auf fernzugriff-pools.hhn.hs-heilbronn.de
4. Loggen Sie sich ein und wählen Sie einen Rechner aus.
5. Wählen Sie das Image von Herrn Ziegler mit "uC_PCB" (Suche nach Ziegler)
6. Dann dauert es etwas, da der Pool Rechner nun das Windows-Image startet.
7. Nun können Sie über Ihren Browser auf einen PC mit Microchip Studio zugreifen.
8. Beachten Sie, dass alle Daten nach dem Beenden der Session gelöscht sind. D.h. es bietet sich an das "U:" Laufwerk ("Home-Verzeichnis") einzurichten. Kommen Sie dazu auf Herrn Ziegler zu.

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 Progi (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._menueführung.hex](#) herunter.
4. Verbinden Sie nach dem Öffnen das Progi per USB-Kabel mit dem PC / Laptop. Dieser Schritt muss vor den folgenden geschehen!
5. Es sollte am Progi 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-Menü 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._menueführung.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 (-00) 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>)

Häufige Fehler

- **F_CPU not defined for** (z.B. <util/delay.h>) Das beste ist die Frequenz F_CPU im AVR Studio direkt anzugeben:
 - Gehe zu Menu: Projekt » (ProjektName) Eigenschaften » Toolchain » AVR/GNU C Compiler » Symbols
 - Füge F_CPU=8000000 (bzw. Passende Frequenz) ein
- **Das Programm kompiliert nicht TWSR not found** : Falls Sie einen modernen AVR Chip nutzen (z.B. 328PB) so kann dieser mehrere SPI und I2C Schnittstellen haben. Damit haben sich bei diesem Target auch die Register- und Interruptvektornamen geändert. Statt TWSR ist dann TWSR0 oder TSWR1 zu verwenden - je nach gewünschtem Pin. Dies ist am einfachsten über defines der fehlerhaften Namen, also #define TWSR TWSR0 usw.
- **Beim Flashen der realen Hardware über Tools » Device Programming finde ich im Tool nur "Simulation", aber kein STK500.** Versuchen Sie zunächst über Tools » Add tagret... STK500 und den entsprechenden Serial Port zu wählen. Falls Ihr Rechner mehrere USB Ausgänge hat, müssen Sie diese (COM1...COMx) beim Programmieren ausprobieren.
- **Beim Flashen der realen Hardware erhalte ich "Erasing device failed", "Error status**

received: Got 0xc9, expected 0x00 (An unknown command was sent)“.

- Steht bei Device Programming das Interface auf ISP? Falls nicht kann dies die Ursache sein. Das Programming geschieht immer mittels ISP.
- Hat das USB-Kabel/Progi/Adapterplatine/Kabel ein Problem? Probieren Sie eine andere Variante der Komponenten durch
- **Mein Chip hat keinen Speicherplatz mehr** bzw **Ich erhalte ein 'Memory Overflow' Fehler** Falls Sie Daten statt im SRAM im EEPROM speichern wollen, so können Sie das Befehlswort “PROGMEM” nutzen. Details dazu finden Sie z.B. auf der Seite von [Microchip](#)
- **Mein Programm scheint irgendwo nicht weiter zu kommen.** Dies kann verschiedene Gründe haben:
 - Endlosschleife
 - Speicherüberlauf im RAM: sobald die Speicherauslastung des RAM über ca 75% steigt, sind Probleme wie spontane Resets bei Bearbeiten von Pointern, Arrays, Strings oder Structs wahrscheinlich. Die kann über Debugging herausgefunden werden (entweder mit Steppen mit Debugger oder Ausgabe von Werten nach jeder Zeile).

I2C

- **Auf den I2C Leitungen ändert sich nichts, obwohl der IC etwas ausgeben sollte:**
 1. Überprüfen Sie die Pullup-Widerstände: Sind welche verbaut? Welche Größe haben diese? (typisch: 10kOhm). Wenn keine Verbaut sind, so wechselt das Signal nur zwischen 0V niederohmig und 0V hochohmig. Dies ist am Oszilloskop nicht zu unterscheiden.
 2. Ist ein hochohmiger Widerstand \$R_L\$ entlang der Leitungen verbaut? Falls ja erzeugt dieser einen Spannungsteiler mit dem Pullup-Widerstand. Wenn \$R_L\$ groß ist, so liegt zwischen \$R_L\$ und Pull-up fast die Versorgungsspannung an.
- **Der Master soll Daten vom Slave empfangen, aber hängt sich manchmal auf** Im “Master Receiver Mode” muss der Master das Ende der Kommunikation dem Slave mitteilen. Dazu muss beim Lesen der Daten TWEA = 0 gesetzt werden. Ansonsten kann es sein, dass der Slave meint er müsse noch Daten senden. Das kann unter Umständen dazu führen, dass der Slave die Datenleitung SDA am Ende der Kommunikation auf Low legt und damit den I2C stört.

0.3 MEXLE GitLab

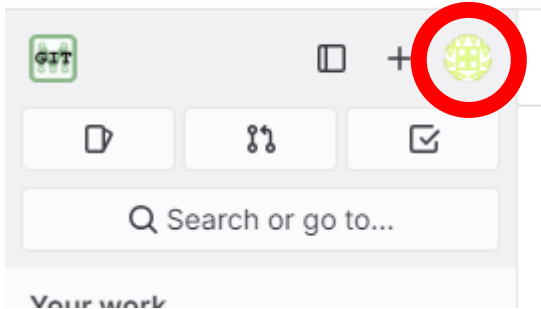
Installation der Software auf Ihrem Rechner

1. Laden Sie Git von folgender Seiter 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/>
totroiseGit 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)

Anmeldung beim MEXLE GitLab

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



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!

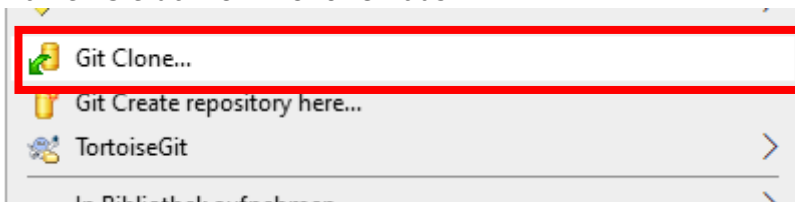
Installation der Software auf Ihrem Rechner

Die Abfolge ist wie folgt:

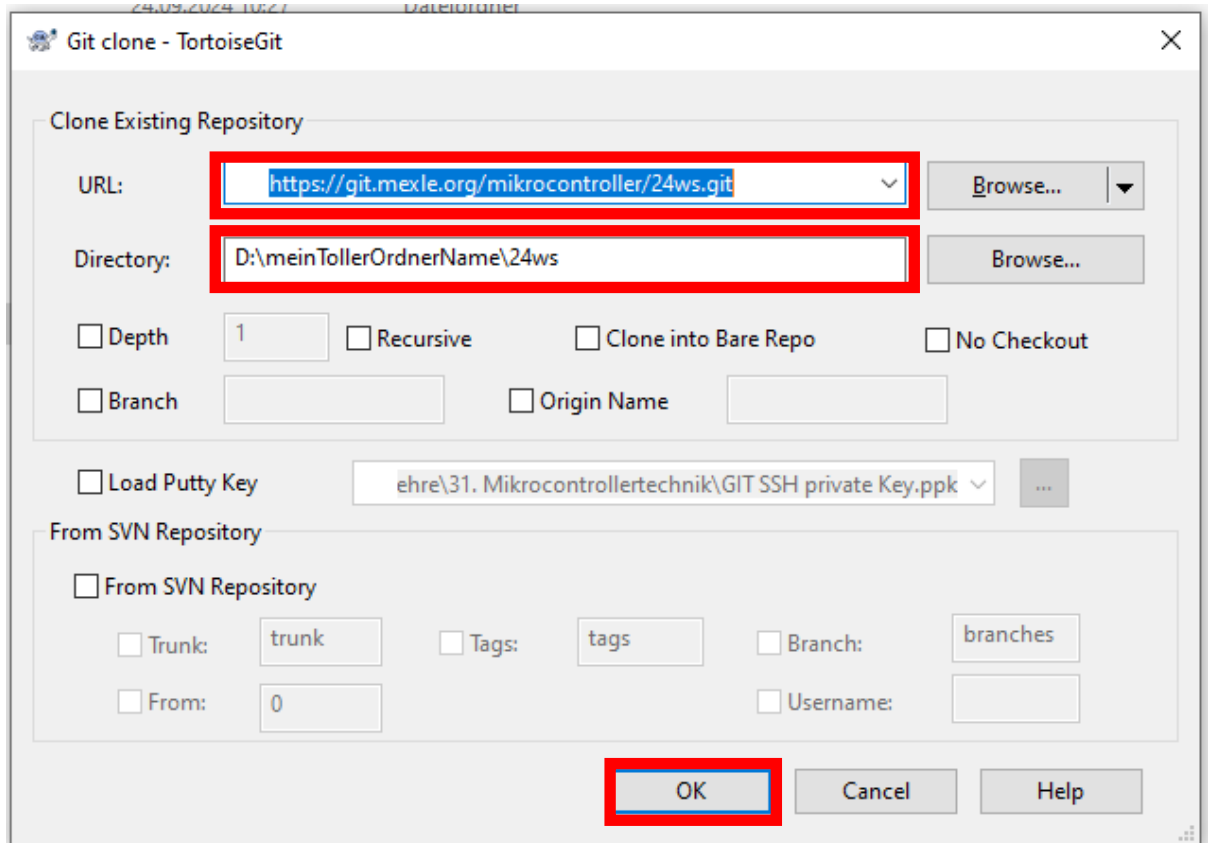


1. Erst müssen Sie sich in **GitLab anmelden**
2. Dann müssen Sie einem **Betreuer per Mail bescheid geben** (Prof. Fischer, Ralf Ziegler) , damit dieser Sie zum Projekt zuordnet
3. **Erst dann sind einem Projekt zugewiesen** und können die folgenden Punkte durchführen.

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 Kontextmenü 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/26ss.git` oder `https://git.mexle.org/elektronik/26ss.git` für das Sommersemester 2026
 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

Upload / Commit von Dateien

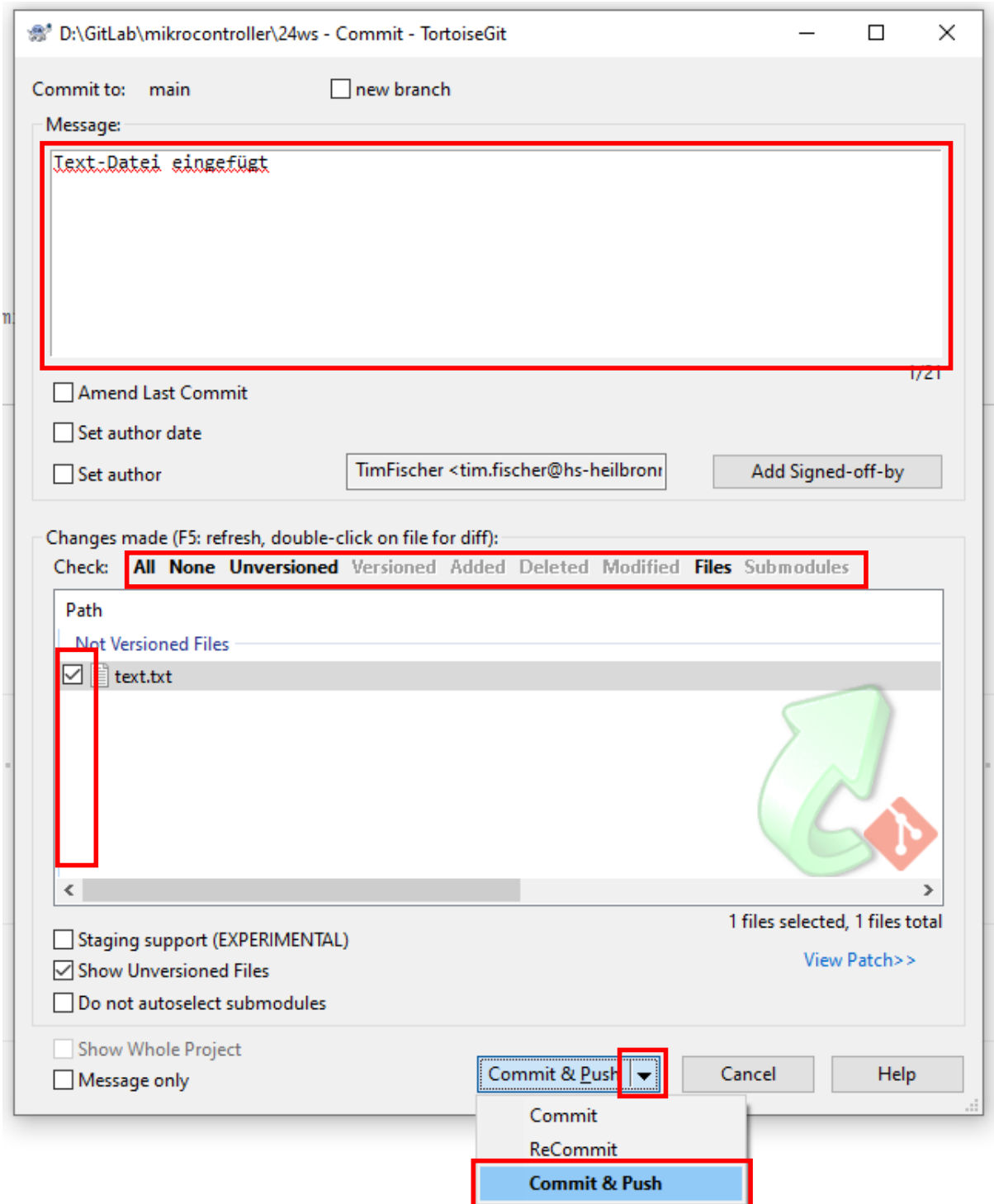
Das Hochladen und Ändern von Dateien bei Git wird als "Commit" bezeichnet. Diese Nomenklatur wird auch im Folgenden genutzt.

Im Folgenden wird ein Upload beschrieben; über einen Commit können aber auch Dateien gelöscht werden.

1. Rechtsklick auf den übergeordneten Ordner » Git Commit -> "main" ...



2. Falls eine Fehlermeldung erscheint, siehe nächstes Kapitel
3. Im folgenden Fenster ist nun einiges einzutragen:
 1. Tragen Sie unbedingt einen Text unter Message ein, ansonsten ist kein Upload möglich! Schreiben sie einen Text, welcher die Änderungen beschreibt.
 2. Im unteren Teil sind die zu ändernden Dateien zu markieren. Entweder Sie wählen die Dateien einzeln aus, oder Sie wählen z.B. All .
 3. Zum Commit an den Server wählen Sie am Button unten ▼ aus und dort Commit & Push



4. Bestätigen Sie nun den Commit über Druck auf den Button Commit & Push
4. Überprüfen Sie, ob der Commit erfolgreich war durch einen Blick auf die Homepage des Projekts.

Hinweise und Mögliche Fehler

1. Für Abschlussarbeiten und studentischen Projekte:
 1. Da diese häufig mehr als 100 MB hochladen, sollten Sie **nicht** [https://git.mexle.org/...](https://git.mexle.org/) nutzen, sondern: [http://git.mexle.te.hs-heilbronn.de/...](http://git.mexle.te.hs-heilbronn.de/)
 2. In diesem Fall müssen Sie im Hochschulnetz befinden (z.B. per eduVPN).
 3. Fragen Sie bei mir (Tim Fischer) nach, welches Git Repository für Sie das passende ist.

2. fatal: detected dubious ownership in repository at [...] is on a filesystem that does not record ownership

Das Problem ist, das 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/25WS` (ändern Sie ggf. elektronik in mikrocontroller und das Semester) und bestätigen Sie mit Return

From:

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

Permanent link:

https://mexle.te.hs-heilbronn.de/microcontrollertechnik/0_hilfsmittel

Last update: **2025/06/18 09:15**

