0 Tools

Student Group

First Name	Surname	Matrikel Nr.

Table of Contents

0 Tools

0.1 SimulIDE

You can find the free electronics simulation program SimulIDE in ILIAS under `(DE/EN) Software`. Alternatively, the program can also be downloaded from the manufacturer's website. There, a “small contribution” must be entered, which can also be 0€.

In contrast to TINA TI, SimulIDE has a different focus and therefore the following advantages and disadvantages.

Advantages:

- Microcontrollers are simulated well (including Atmel chips). This is not possible in TINA TI.
- The microcontrollers can also be programmed. This means that microcontroller-suitable source files (hex files) can be used.
- Interaction between software and hardware is possible.

Disadvantages:

- Simulation of various electronic components is only implemented in a simplified way (e.g. operational amplifiers or FETs)
- The software is still being heavily developed. This means it is a good idea to use the version specified above in order to avoid compatibility problems.

To get started with SimulIDE, it helps to watch the developer's playlist.

Tips for Using SimulIDE - Configuration

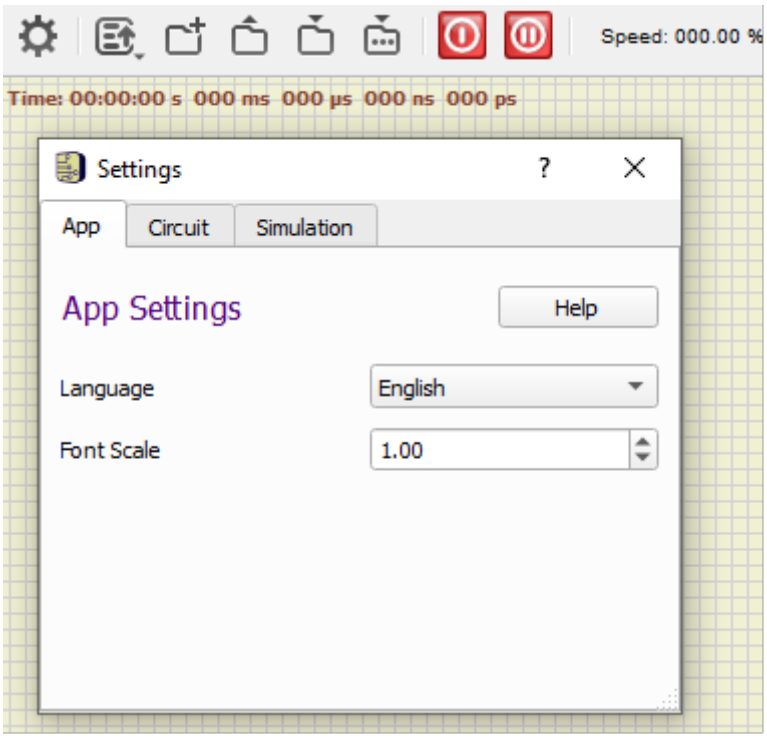


Fig. 1: Image for configuration

- If the text is displayed too small after opening the program, one of the following approaches may help:
  - In SimulIDE, after opening the program, click on the gear icon. Select the App tab. There, enter for example 2.0 for `Font Scale` and restart the program.
  - In Explorer: `right-click » Properties » Compatibility » Change high DPI settings » Override high DPI scaling behavior » check the box » select “System”`. (if necessary, choose “System (Enhanced)”)

0.2 Microchip Studio

Microchip Studio is a programming environment for creating a microcontroller-suitable source file (hex file) from C or C++.

Installation

Installation Guide

The current version of the program can be found on the Microchip homepage.

1. If nothing happens after pressing the `Download Microchip Studio` button: simply scroll down to `Downloads and Documents`
2. With fast internet connections, the web installer can be chosen; with slower connections, the offline installer allows the entire package to be downloaded before installation.
3. During installation, only the “AVR” architecture is required.
4. “Advanced Software Framework and Example Projects” are not required.
5. Depending on the speed of the computer and the internet connection, installation takes about 5 minutes.
6. If it is not already installed, Visual Studio will also be installed during installation.
7. The question about licenses for the C compiler should be skipped with `Next >`
8. In addition, various device drivers may also be installed. These make it possible to write to the chips using a programming device.
9. Afterwards, open Microchip Studio directly so that you can make the first important settings.

for Mac Users

The following procedure is especially recommended for Mac users:

1. Open a browser.
2. Establish a VPN connection via eduVPN.
3. Go to fernzugriff-pools.hhn.hs-heilbronn.de
4. Log in and select a computer.
5. Select Mr. Ziegler's image with “uC_PCB” (search for Ziegler)
6. Then it will take a little while, because the pool computer now starts the Windows image.
7. You can now access a PC with Microchip Studio through your browser.
8. Please note that all data is deleted after ending the session.
   This means it is a good idea to set up the “U:” drive (“home directory”). Please contact Mr. Ziegler for this.

First Test of the Connection to the AVR-USB-Progi

Carry out the following steps to perform an initial test with the hardware.

1. Preparations
    1. Connect the MiniMEXLE
        1. to the Progi (black box) via ribbon cable
        2. to the power outlet via charging cable with the barrel connector
    2. Open Microchip Studio (for example by pressing the <WIN> key, entering Microchip Studio, and pressing <Return>).
    3. Download the file 5._menufuehrung.hex.
    4. After opening the program, connect the Progi to the PC / laptop via USB cable. This step must be done before the following ones!
    5. The green LED for USB communication should light up on the Progi.
2. Enter USB connection
    1. Without creating a project, go directly to `Tools » Add Target...`
    2. In the displayed window, you should select `STK500` as the tool in the dropdown menu, which was previously empty.
    3. For the selection of the “Serial Ports”, choose the first available one, e.g. `COM3`, and confirm the selection with `Apply`.
    If no port is shown here, check whether the connection to the USB-Progi was successful. If the green LED is lit and the USB cable is connected correctly, refer to the instructions under 4.d.
3. First connection attempt
    1. Select `Tools » Device Programming` in the menu.
    2. Under Tools, select the corresponding USB connection, e.g. `STK500 COM3`.
    3. Under Device, you must select `ATmega88`; you can enter `88` and then click the correct value in the dropdown menu.
    4. Under Interface, ISP should now be shown.
    5. Now press `Apply`.
4. In case of error:
    1. If it takes longer after pressing `Apply`, the connection did not work. An error message `Unable to connect tool STK500 (COMx)` will then be displayed.
    2. You can acknowledge the error message, but after that it takes a few more seconds until the program responds correctly again.
    3. Then close the `Device Programming` window. Go back to point 2 in this guide and try the next serial port.
    Remember the number of the port, as this is needed under 3.b.
    4. If all available COM ports have been tested, or none were displayed from the beginning, installing a Virtual COM Port Driver may help. One can be found at FTDI.
5. If it works:
    1. You should now see a bit more in the `Device Programming` window.
    2. Before the next steps, connect the MiniMEXLE (= board with display and buttons) to the Progi with the ribbon cable and to the power supply with the barrel connector. Also plug the power supply into an outlet.
    3. Now press `Read` at the top next to `Device Signature`.
    4. A hexadecimal number `0x….` should appear. Your computer can now establish a connection to the MiniMEXLE for flashing.
    5. In the `Device Programming` window, go to `Memories`. Under `Flash`, insert the path to the downloaded file `5._menufuehrung.hex`. You can also select the path using the **…** button to the right.

6. Press `Program`
7. The MiniMEXLE should now display
   `- Experiment 5 -`
   `Program Menu`
   and then
   `Main Level`
   `P1 P2 P3 P4`
8. This means you have successfully tested the connection and the flashing process

Important Settings

- Use the **display of line numbers**: `Tools » Options » Text Editor » All languages » General » Line numbers`
- As soon as you work on the first project: Make sure to disable compiler optimization. This can be done with the following steps:
  - Menu `Project » <ProjectName> Properties… » AVR/GNU Compiler » Optimization`
  - The optimization level should be set to `None (-O0)`
- I recommend using ATMEL Studio in English. This makes the instructions in this course easier to follow correctly. If you accidentally selected German (e.g. during installation), you can correct it under `Tools » Options » environment » international settings » Language`.

Tips

- If the **Solution Explorer** (display of the files in the project) is not present on the right-hand side, you can find it under
  `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.

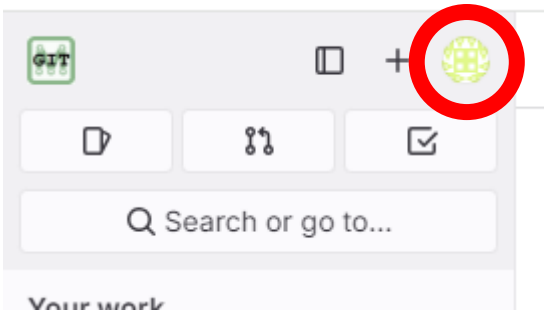
0.3 MEXLE GitLab

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



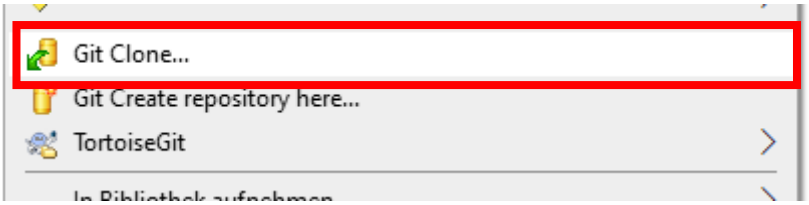
3. Geben Sie ein Passwort mit mindestens 16 Zeichen ein und merken Sie sich dieses Passwort. Dieses Passwort ermöglicht die Authentifizierung auf GitLab.
   Bitte nutzen Sie ein neues Passwort!

Installation der Software auf Ihrem Rechner

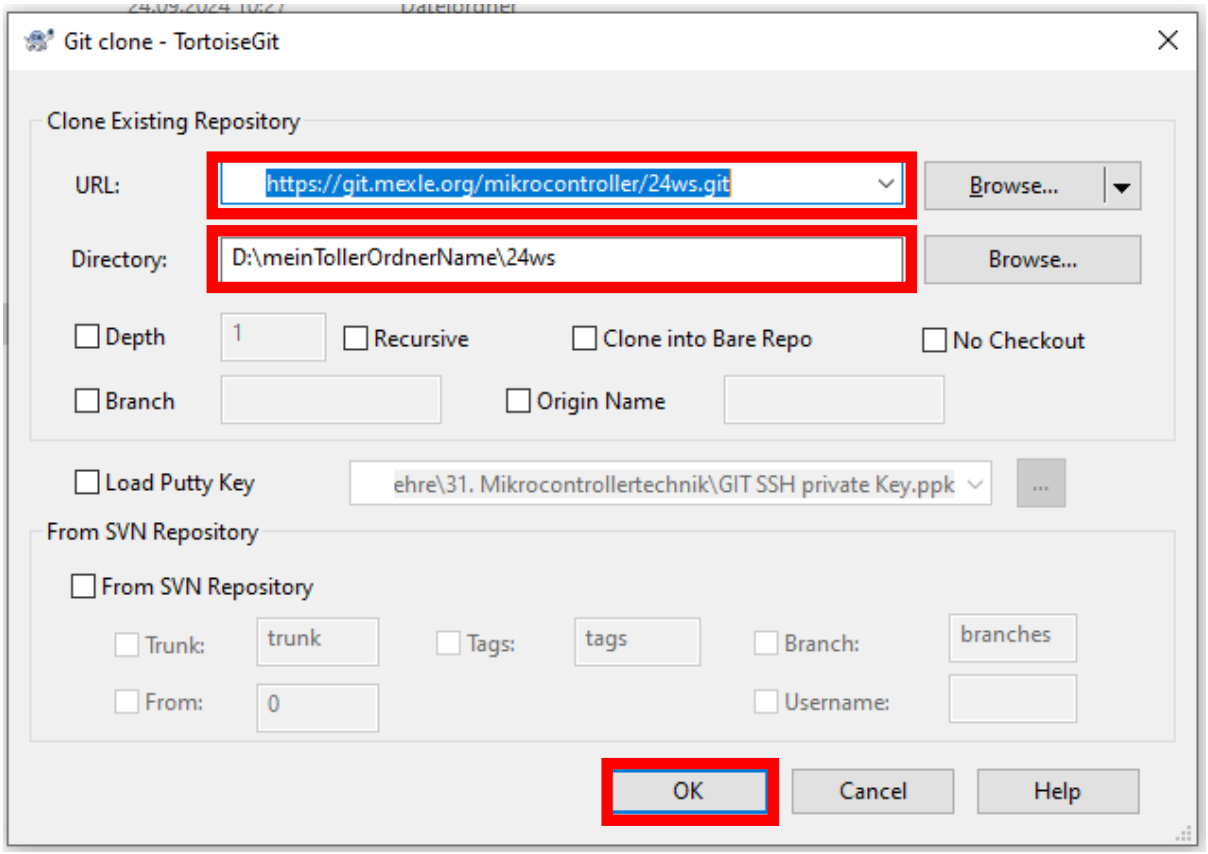
Die Abfolge ist wie folgt:

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`



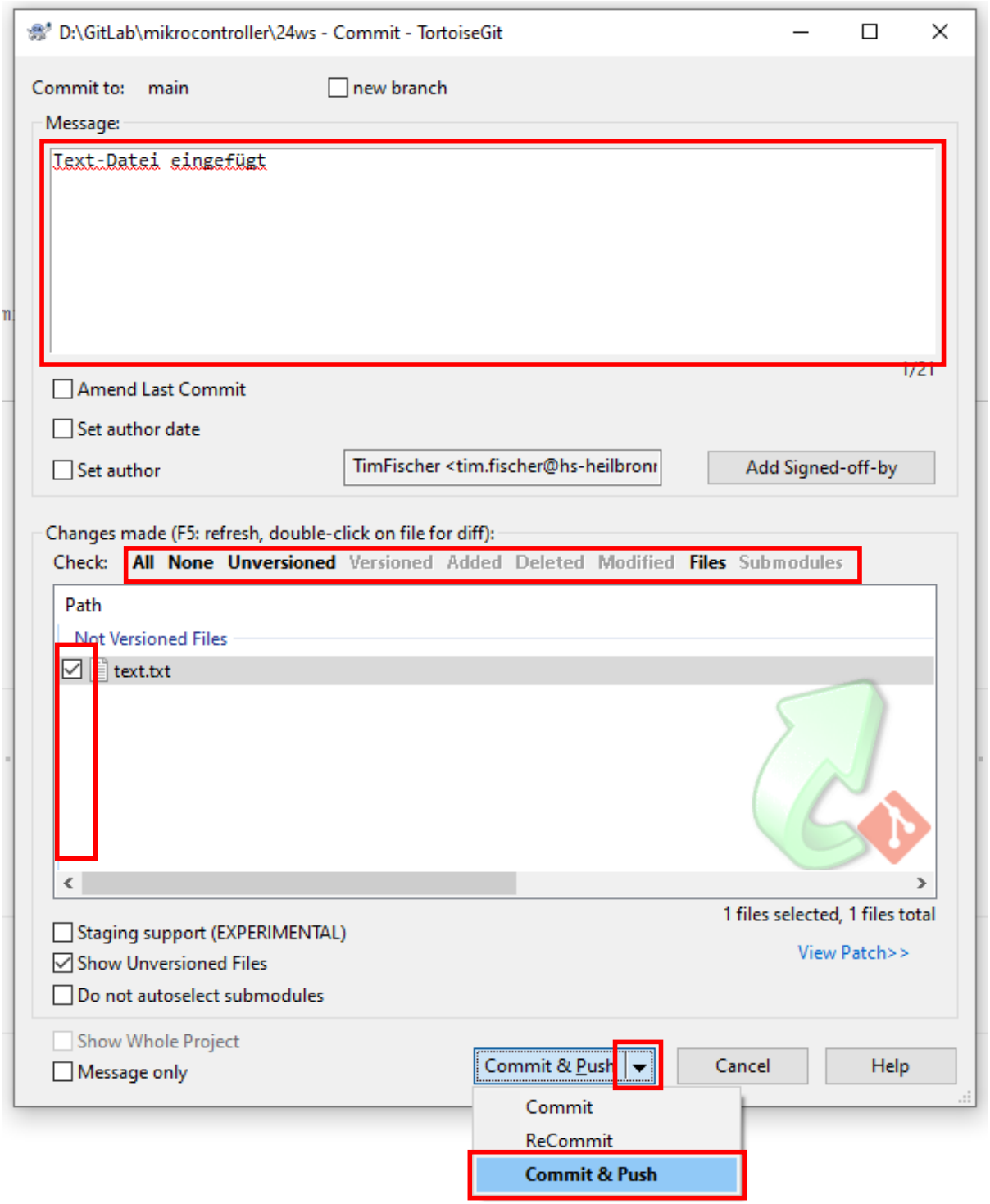
4. Im Anschluss sollten Sie eine Fehlermeldung erhalten, da noch das Passwort fehlt.
   Hier ist nun das vorher gewählte Passwort einzugeben.
   Dies müssen Sie auch nur einmalig machen.
5. Der Download sollte nun klappen und es sollten alle Ordner heruntergeladen werden

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/...` nutzen, sondern: **`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
3. `git did not exit cleanly (exit code 1)`
   Das Problem ist, dass ihr lokale Datenbank nicht mehr aktuell ist und sie zunächst die Datenbank vom Server herunterladen müssen (“Pull”).
   1. Generell hilft hier erst zu Pull'en dann zu Commit&Push'en