

# Mikrocontroller in der Leistungselektronik

## Dimmen einer Lampe

Schritt-für-Schritt Anleitung zur Installation der erforderlichen Programme, wenn der eigene Laptop genutzt werden soll.

### Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Installation von Git</b>	<b>2</b>
<b>3</b>	<b>Installation der Arduino IDE</b>	<b>3</b>
<b>4</b>	<b>Installation von Visual Studio Code</b>	<b>5</b>
<b>5</b>	<b>Erste Schritte mit Visual Studio Code und Git Graph</b>	<b>8</b>

## 1 Einleitung

Ihnen steht im Praktikumsraum ein Rechner zur Verfügung, mit dem Sie den Versuch durchführen können. Es besteht aber auch die Möglichkeit, den Versuchsstand mithilfe eines selbst mitgebrachten Rechners betreiben zu können. Dies bietet den Vorteil, das auf einem eigenen Rechner das Git Repositorium bereits angelegt werden kann und es möglich ist, die Branches und Programme für die einzelnen Aufgabenstellungen schon vorzubereiten. Dann müssen diese im Versuch nur noch debuggt werden. Dazu müssen Sie jedoch einige Programme im Vorfeld installieren. Diese umfassen:

- Das Versionsverwaltungsprogramm Git.
- Die Arduino IDE mit der dazugehörigen Erweiterung für den XMC1100 XMC2Go, um auch das Board von Infineon betreiben zu können.

Diese Programme reichen aus, um den Mikrocontroller mithilfe der Software von Arduino zu programmieren und die Versionsverwaltung beispielsweise

über die Kommandozeile von Git zu realisieren. Ein nützliches drittes Programm ist

- Visual Studio Code.

Mit Visual Studio Code kann über eine Erweiterung auch die Arduino Software genutzt werden und es wird gleichzeitig eine grafische Oberfläche für Git geboten, die mit einer weiteren Erweiterung auch den Graphen zu der Versionshistorie zeichnet. Voraussetzung sind dazu jedoch Git und die Arduino IDE. Alle hier beschriebenen Programme sind auf jedem gängigen Betriebssystem installierbar.

Im Folgenden wird beschrieben, wie diese Programme installiert werden können und welche Einstellungen getroffen werden müssen, um einen reibungslosen Ablauf im Praktikum zu gewährleisten. Danach werden kurz erste Schritte gezeigt, wie mit Visual Studio Code gearbeitet werden kann.

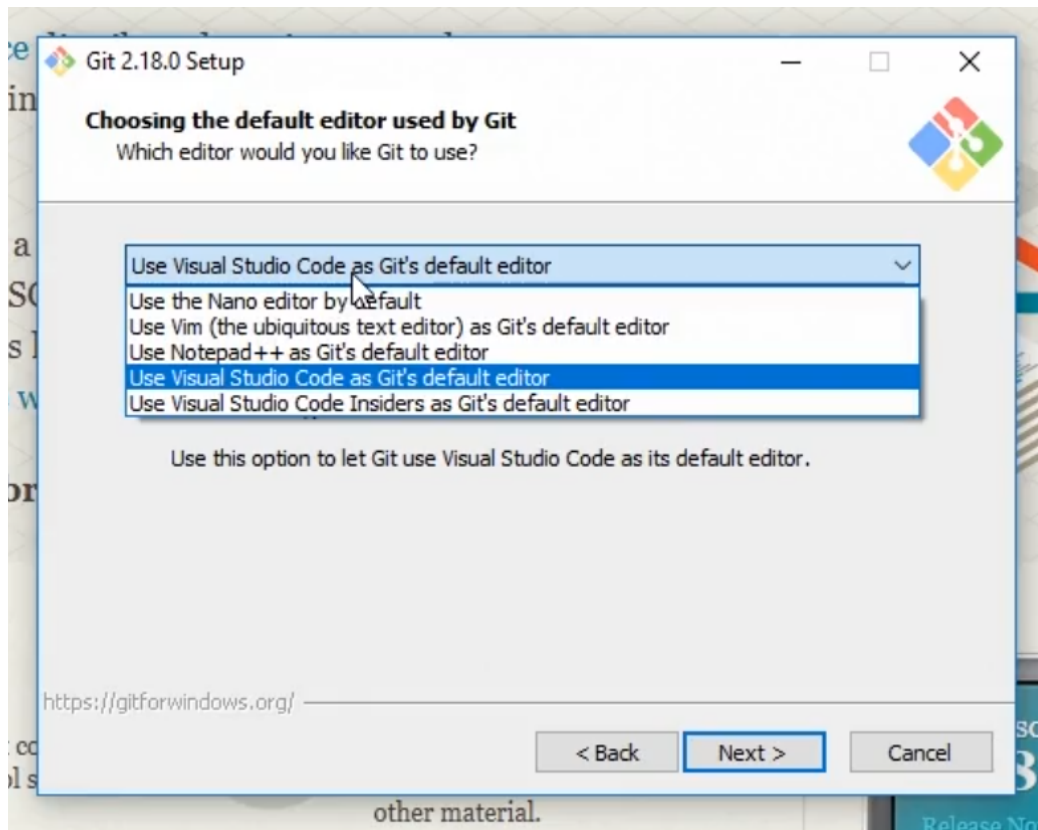
## 2 Installation von Git

Zuerst wird die Versionsverwaltung Git installiert. Dazu wird die Seite <https://git-scm.com/> aufgerufen und die Version für das jeweilige Betriebssystem heruntergeladen.



In Windows wird die `.exe` ausgeführt und im darauffolgenden Installationsprozess können die gegebenen Einstellungen übernommen werden. Als

Standardeditor sollte *Visual Studio Code* ausgewählt werden, wenn es später noch installiert werden soll.



In Linux erfolgt die Installation im Terminal über

```
sudo begin apt-get install git
```

und dem üblichen Installationsprozess im Terminal.

### 3 Installation der Arduino IDE

Die Arduino IDE ist notwendig, um die selbst geschriebenen Programme auf den Mikrocontroller zu übertragen. Da hier nicht ein Arduino genutzt wird, sondern ein Mikrocontroller von Infineon, muss noch eine Erweiterung sowie die Software *J-Link* installiert werden.

**Installation der IDE:** Die Arduino IDE kann unter <https://www.arduino.cc/en/main/software> für das jeweilige Betriebssystem heruntergeladen werden.

## Download the Arduino IDE



Klickt man auf *Getting Started* (<https://www.arduino.cc/en/Guide/HomePage>) links neben den Downloads, so findet man die Installationsanleitung für das jeweilige Betriebssystem. Unter Linux kann es sein, dass die Datei *install.sh* nicht ausführbar ist. Dann führt nach Wechseln in das entpackte Verzeichnis

```
chmod +x install.sh
sudo install.sh
```

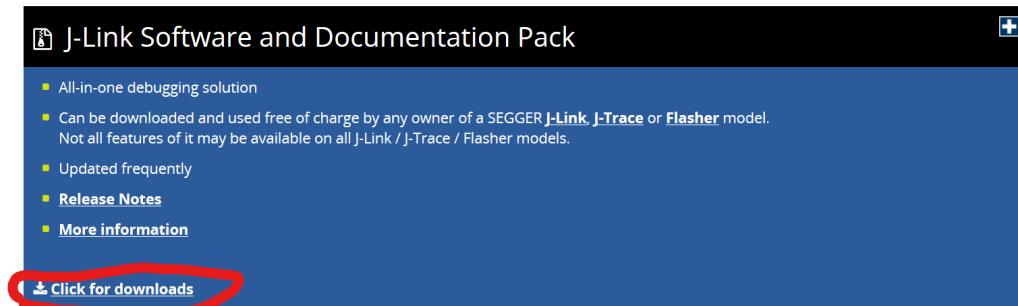
zur gewünschten Installation, falls

```
./install.sh
```

nicht funktioniert.

**Installation der Erweiterung für das Board XMC1100 XMC2Go:** Unter <https://github.com/Infineon/XMC-for-Arduino> findet man eine gut dokumentierte Anleitung, welche Schritte unternommen werden müssen, um die Kommunikation des XMC1100 XMC2Go mit der Arduino IDE zu gewährleisten. Es umfasst grob zwei Schritte.

1. Installation der Software SEGGER J-Link. Hier ist zu beachten, dass die Installationsdateien unter *J-Link Software and Documentation Pack* zu finden sind.

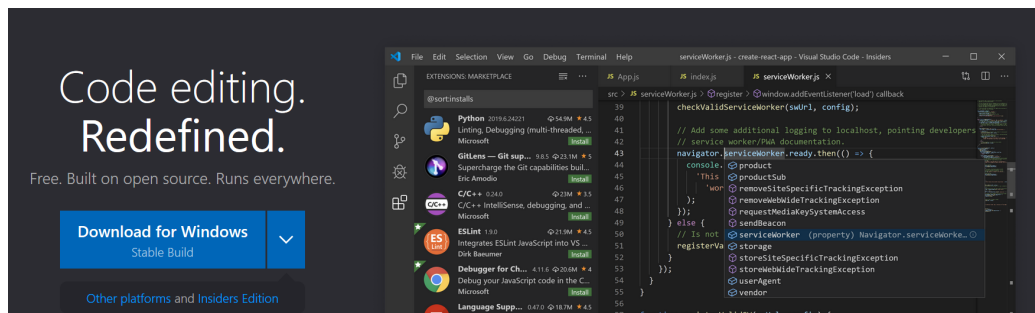


2. Installation der erforderlichen Erweiterung in der Arduino IDE.

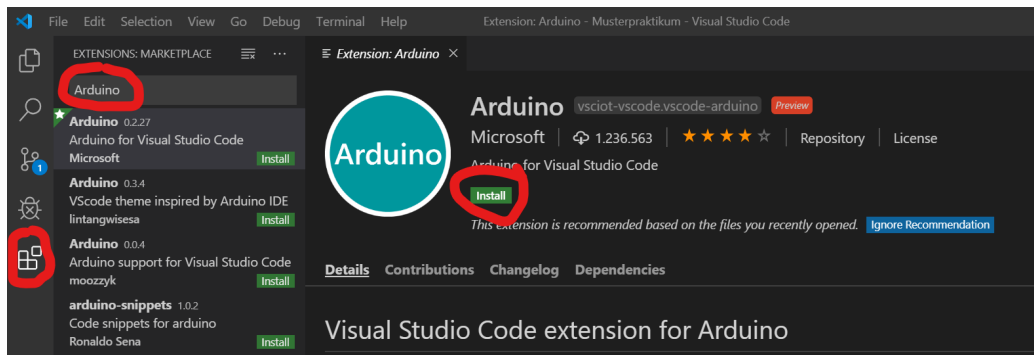
## 4 Installation von Visual Studio Code

Visual Studio Code ist ein sehr nützliches Programm, mit dem gleichzeitig der Code für den Mikrocontroller geschrieben und überspielt werden kann und eine grafische Oberfläche für Git zur Verfügung steht.

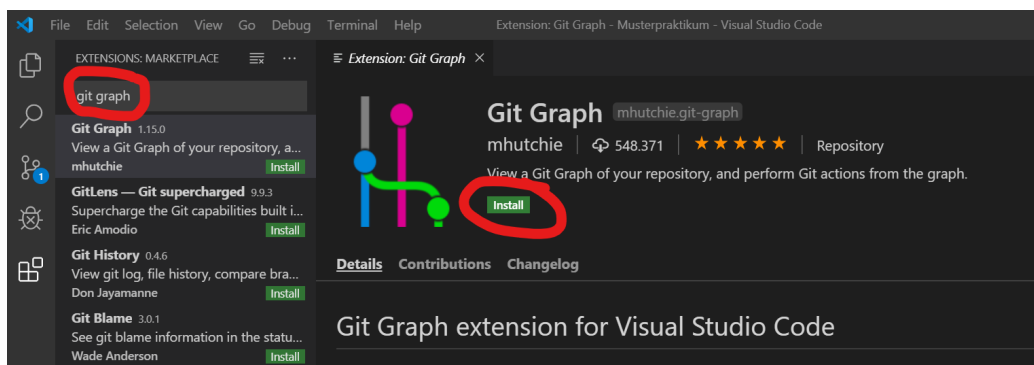
**Installation:** Visual Studio Code kann unter <https://code.visualstudio.com/> heruntergeladen und anschließend installiert werden.



**Installation der Erweiterungen für Arduino sowie Git Graph:** Nach Öffnen von Visual Studio Code sind die Erweiterungen links in der Seitenleiste zu finden (Vier Quadrate, von den eins abgekoppelt ist). Öffnet man dies, so kann man in der Suchleiste nach Erweiterungen suchen. Hier wird zuerst nach der Erweiterung für Arduino gesucht und diese installiert. Nun kann auch aus Visual Studio Code die Funktionen der Arduino IDE genutzt werden.



Als nächstes wird noch die Erweiterung *Git Graph* installiert, mit der das Repository grafisch dargestellt werden kann und auch gleich Git-Befehle ausgeführt werden können.



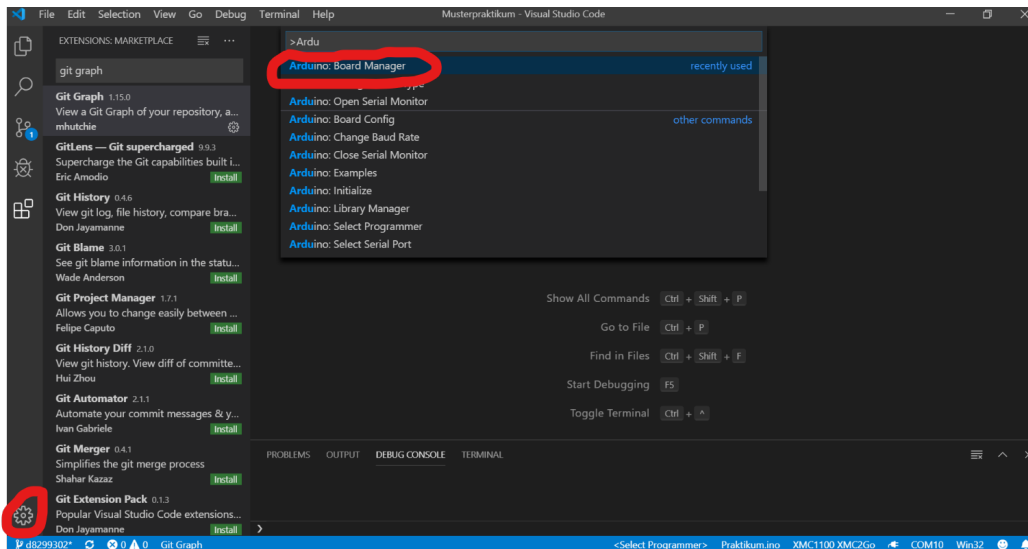
**Einstellen des Boards für den XMC1100 XMC2Go:** Als erstes sollte kontrolliert werden, ob die Erweiterung für den XMC2Go, die auf der Arduino IDE installiert wurde, auch von Visual Studio Code übernommen wurde. Dazu geht man entweder links unten auf der Seitenleiste auf das Zahnrad auf die *Command Palette...* oder nutzt den Shortcut

STRG + Umschalt + P

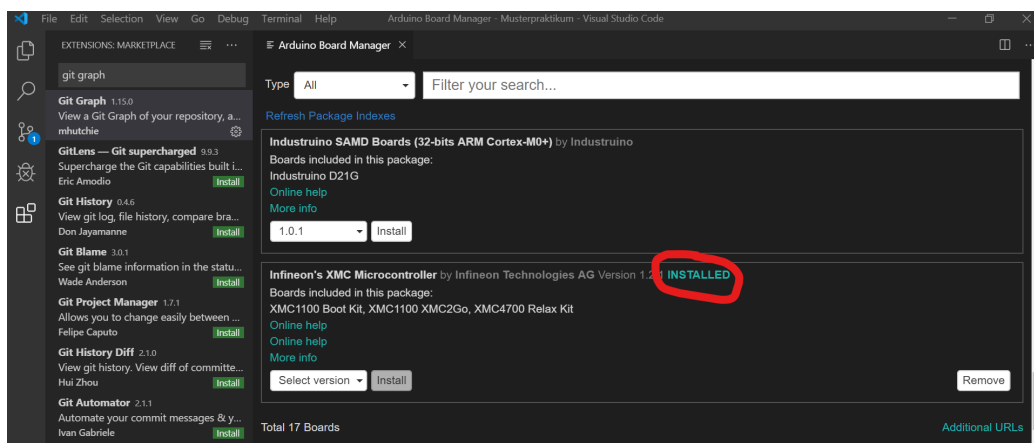
um zur Befehlsleiste zu kommen. Dort sucht man den Befehl

Arduino: Board Manager

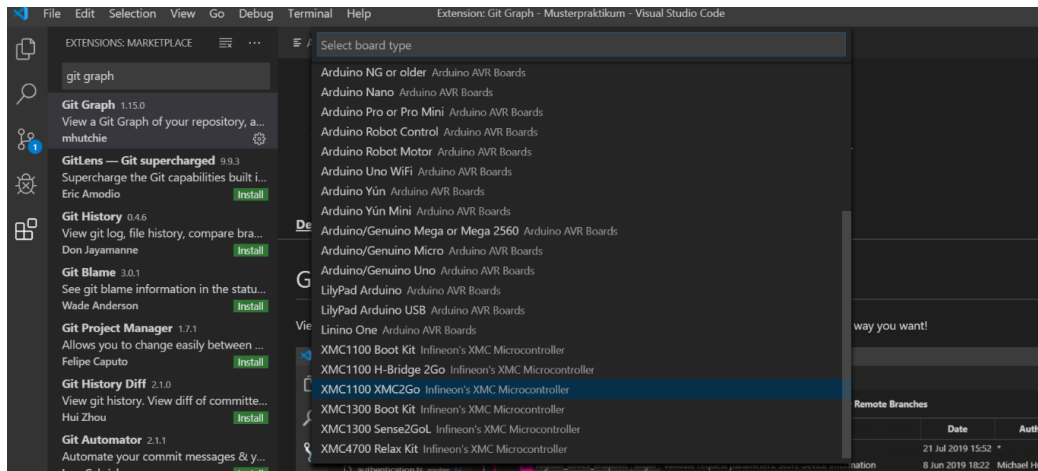
und öffnet diesen.



Hier überprüft man nun, ob die Erweiterung *Infineon's XMC Microcontroller* installiert ist. Sollte die Erweiterung noch nicht angezeigt werden, kann ein Neustart des Systems helfen.



Über die Befehlsleiste kann nun mit  
**Arduino: Change Board Type**  
 der XMC1100 XMC2Go als gewünschter Mikrocontroller eingestellt werden.



## 5 Erste Schritte mit Visual Studio Code und Git Graph

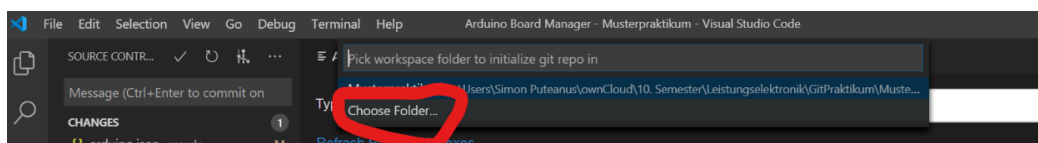
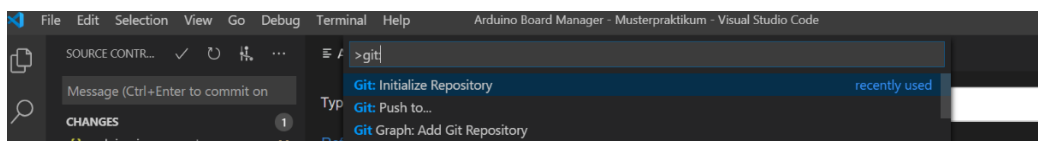
**Anlegen eines Repository:** Um ein neues Repository anzulegen, sollte zuerst ein leerer Ordner erstellt werden. Dann kann wieder die Befehlsleiste mithilfe von

STRG + Umschalt + P

geöffnet werden und mit dem Befehl.

Git: Initialize Repo

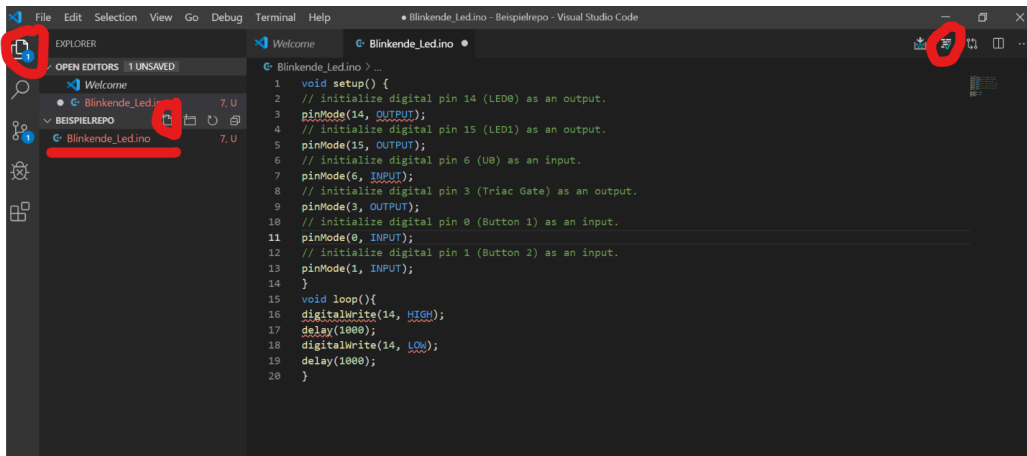
ein neues Repository angelegt werden. Dazu wird der neu angelegte Ordner ausgewählt, in dem dann das Repository angelegt wird.



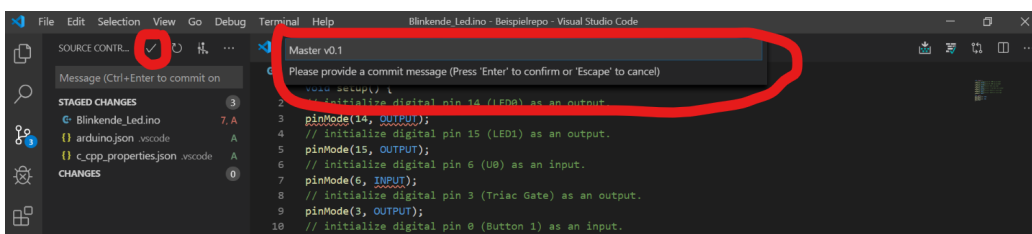
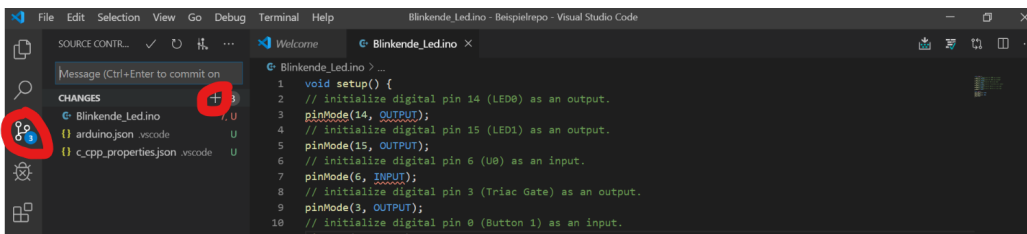
In diesem Repository kann dann im Explorer (linke Leiste, „Datei“ Zeichen) eine neue Datei erstellt werden (hier *Blinkende.Led.ino*), die dann mit



dem Beispielcode aus der Anleitung gefüllt wird. Durch die Dateierweiterung `.ino` erkennt Visual Studio, dass es sich hier um ein Programm für Arduino handelt, sodass man nun bereits das Programm kompilieren kann, um Fehler zu finden. Gleichzeitig wird dadurch die Datei auch gespeichert.

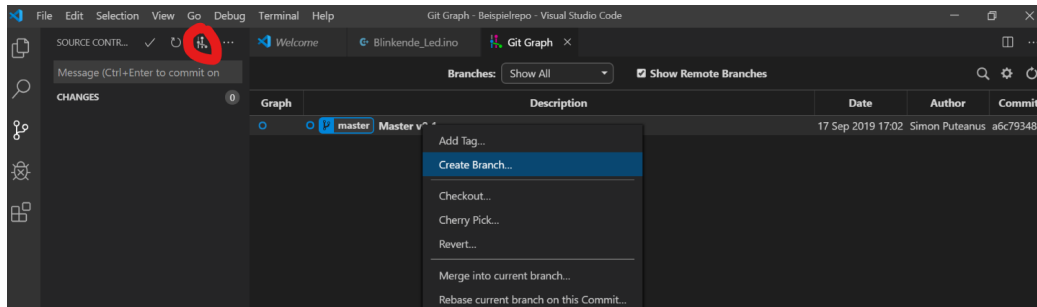


**Erstellen des ersten Commits:** Um nun einen ersten Eintrag im Repository anzulegen wechselt man in die Versionskontrolle (linke Leiste, „Graph“ Zeichen) und klickt dann erst auf *Stage all Changes* und dann auf *Commit* worauf man einen Kommentar für das Commit eingeben muss.

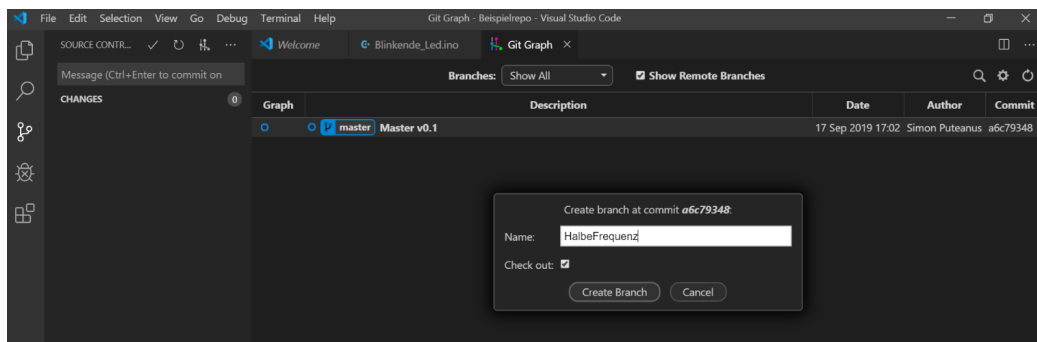


**Erstellen eines neuen Branches:** Wechselt man nun in die grafische Oberfläche von *Git Graph*, so kann man hier einfach durch Rechtsklick auf

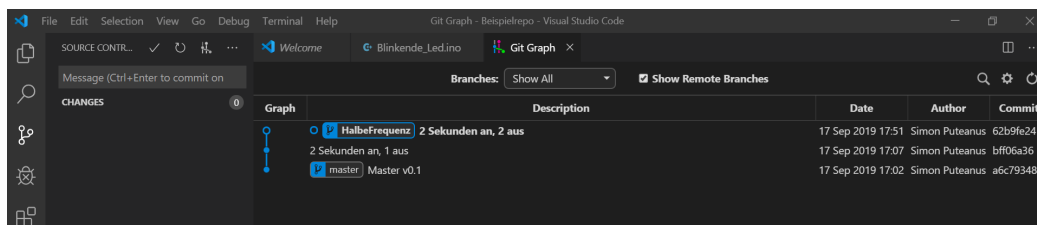
den aktuellen Commit einen neuen Branch erstellen, um beispielsweise ein neues Feature zu entwickeln.



Bei der Namensgebung des neuen Branches ist zu beachten, dass keine Leerzeichen erlaubt sind. Ein Häkchen bei *Check out* sorgt dafür, dass in den neuen Branch gewechselt wird.

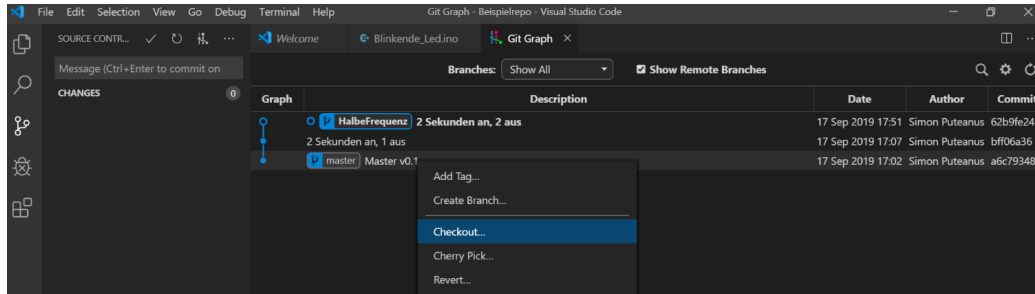


Nun kann das Programm beliebig angepasst werden und Zwischenergebnisse durch *Compile* oder *Save*, *Stage all changes* und *Commit* analog zum ersten Commit in der Versionsverwaltung übernommen werden, das Ergebnis sieht dann beispielsweise folgendermaßen aus.



**Mergen mit dem Master Branch:** Das erstellte und entwickelte Feature kann nun mit dem *master*-Branch gemerged werden. Dazu klickt man erst

auf den *master*-Branch und führt ein *Check out* aus. Nun befindet man sich wieder im *master*-Branch.



Als Nächstes macht man einen Rechtsklick auf den aktuellen Stand des Feature Branch und klickt *Merge into current Branch...* um beide Graphen zu mergen. In der dann sich öffnenden Box sollte ein Häkchen bei *Create a new commit, even a fast-forward is possible* gesetzt sein. Es ergibt sich folgendes Bild, das zeigt, dass der Branch erfolgreich in den *master*-Branch übernommen wurde.

