

DAS GIT REPOSITORY TI_BRD_BASIS.GIT

STRUKTUR: Dieses GIT Repository *TI_BRD_BASIS* speichert Musterlösungen für Praktikumsaufgaben und praktische Programmierprüfungen, die auf dem TI Board (Cortex M4 basiert) durchgeführt wurden.

Der *master* Branch speichert die Grundeinstellungen für die Keil µVision IDE. Jede Musterlösung ist ein separater Branch, der vom *master* Branch ausgeht. Die Namen der Branches sollten sprechend sein, z.B. *GS_Probeklausur_WS2016* oder *Taschenrechner_RPN*.

Darüber hinaus speichert der *master* Branch die Einstellungen des GIT *Submodule* *TI_BRD_LIB*. Diese Bibliothek enthält alle TI Board spezifischen Anpassungen wie zum Beispiel der Startup Code oder die Anpassungen für die Simulation. Weiterhin speichert die Bibliothek ergänzende C Module, wie zum Beispiel die TI Lib, das keypad Module und ein Module mit exemplarischen Einstellungen des Timer TIM2.

REMOTE REPOSITORY: Der gitlab Server des AI Labors (<https://gitlab.informatik.haw-hamburg.de>) speichert die remote Repositories *TI_BRD_BASIS.git* und *TI_BRD_LIB.git*. Diese Repositories gehören der gitlab Gruppe *tiboard_dev*. Über den Informatik Account oder die *abc123* Kennung der HAW kann man auf den gitlab Server zugreifen.

Typische GIT Befehle

Neben den üblichen *git* Befehlen *add*, *commit*, *push* und *pull* fallen folgende Aufgaben im GIT an:

Clonen des Remote Repository:

```
git clone --recursive https://gitlab.informatik.haw-hamburg.de/tiboard_dev/TI_BRD_BASIS.git Musterloesung
```

Anlegen eines Branches für eine neue Musterlösung:

```
git checkout master
git branch TemSensorsOneWireBusPolling
git checkout TemSensorsOneWireBusPolling
```

Damit der Branch “getracked” wird, muss ein *push* mit der der *-u* Option ausgeführt werden, also:

```
git push -u origin TemSensorsOneWireBusPolling
```

Auschecken des Branch einer Musterlösung:

Auch hier muss darauf geachtet werden, dass der Branch “getracked” wird. Somit muss ein *checkout* mit der *--track* Option durchgeführt werden, also:

```
git fetch  
git checkout --track origin/TemSensorsOneWireBusPolling
```

Merge mit neuen Standardeinstellungen aus dem master Branch

Wenn sich die Standardeinstellungen im *master* Branch geändert haben und diese in den Branch der Musterlösung übernommen werden sollen, dann muss der *master* Branch in den Branch der Musterlösung wie folgt “gemerged” werden.

```
git fetch  
git checkout master  
git pull  
git checkout TemSensorsOneWireBusPolling  
git merge master
```

push der Änderungen - nach erfolgreichem Test :-).

Achtung: Wenn das git Submodule *TI_BRD_LIB* aktualisiert wurde, dann muss

```
git submodule update
```

ausgeführt werden.

Änderungen im Submodule *TI_BRD_LIB* durchführen

Änderungen im git Submodule *TI_BRD_LIB* kann man mit den üblichen Befehlen im Baum des Submodules durchführen. Damit die Änderungen nicht verloren gehen, müssen sie über

```
git checkout -b derCoolerFixBranch
```

einen eigenen Branch bekommen. Man kann die Änderungen natürlich auch in einen schon existierenden Branch aus *TI_BRD_LIB* “mergen”.