

BiDiBOne aus dem Repository laden



Unser BiDiBOne-Repository - also die Sammlung der Quellen, die für den Bau einer BiDiBOne-AddOn-Anwendung notwendig ist - haben wir derzeit auf [GitLab](#) hinterlegt. Dort muss man einen entsprechenden Zugang (Account) eingerichtet haben.

Weiterhin ist zur Verwendung dieser Quellen eine Zugangsberechtigung notwendig, die von support@fichtelbahn.de zur Verfügung gestellt wird.

Das Dokument [Anleitung-GitLab](#) enthält wichtige Hinweise zur Einrichtung von GitLab und den notwendigen Werkzeugen.

Beachte: Bei Nutzung von Eclipse C/C++ als Entwicklungsumgebung ist das im Folgenden beschriebene Vorgehen nicht unbedingt notwendig, da Eclipse seine eigenen Werkzeuge mitbringt! Die Hintergrundinformationen sind allerdings die gleichen.

BiDiBOne Basis

Die BiDiBOne-Basis ist (Stand Dez. 2014) aufgeteilt in die Projekte: core (<https://gitlab.com/bidib/core.git>) und support (<https://gitlab.com/bidib/support.git>). Es werden keine Submodule (mehr) verwendet, d.h. alle benötigten Projekte müssen separat geladen werden.

Die aktuelle lauffähige Firmware für die BiDiBOne-Projekte steht auf dem Branch **master** zur Verfügung!

BiDiBOne AddOn laden

Zum Bau eines neuen AddOns empfiehlt es sich, neben den Basis-Repositorys das Modul AddOnStub (<https://gitlab.com/bidib/addonstub.git>) zu laden. Wie das funktionieren könnte, sei am Beispiel von SourceTree und dem Basis-Projekt erklärt.

(Server/Repository-Adressen: Stand März 2014. Das HTTPS-Protokoll muss u.U. bei bestimmten Einstellungen einer vorgeschalteten Firewall verwendet werden.)

Hauptmenü ⇒ Clone/New

Clone / Add / Create Repository

Clone Repository Add Working Copy Create New Repository

Source Path / URL: git@gitlab.com:bidib/basis.git

Repository Type: This is a Git repository

Destination Path: D:\Dev\OpenDCC\MeinErstesAddOn

Advanced Options

Checkout branch: OneControl_Anpassung Clone depth: 0

☒ Recurse submodules

☐ No hardlinks

Bookmarks

☒ Bookmark this repository

Name: MeinErstesAddOn

Folder: [Root]

Clone Cancel

- **SourcePath/URL:** git@gitlab.com:bidib/addonstub.git **ODER** <https://gitlab.com/bidib/addonstub.git>
- **Destination Path und Name:** <parallel zu den Projekten *core* und *support*>
- **(Checkout branch:** Nötig, wenn man nur auf dem angegebenen Branch arbeiten möchte.)

Cloning from git@gitlab.com:bidib/basis.git to D:\Dev\OpenDCC\MeinErstesAddOn

Cancel

☐ Show Full Output

Laden der Quellen ...

Cloning from git@gitlab.com:bidib/basis.git to D:\Dev\OpenDCC\MeinErstesAddOn

Cancel

☒ Show Full Output

git -c diff.mnemonicprefix=false -c core.quotePath=false clone --recursive git@gitlab.com:bidib/basis.git D:\Dev\OpenDCC\MeinErstesAddOn
Cloning into 'D:\Dev\OpenDCC\MeinErstesAddOn'...

Completed successfully.

Close

⇒ Close

MeinErstesAddOn D:\Dev\OpenDCC\MeinErstesAddOn

master

Working Copy Changes

Filename	Path
.gitignore	env
Basis.cproj	env
addon_version.h	src\addon
dmxout.c	src\addon
dmxout.h	src\addon
dmxout_hardware.h	src\addon
4_8_coding.c	src\core
4_8_coding.h	src\core
Flash_Defines.h	src\core
adc_driver.c	src\core
adc_driver.h	src\core
avr_compiler.h	src\core

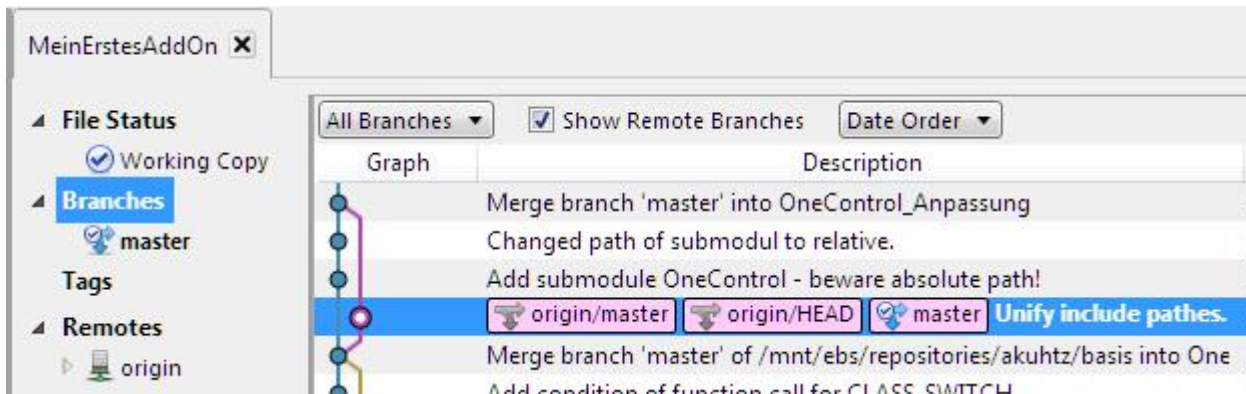
File Status Log / History Search

... das Ergebnis.

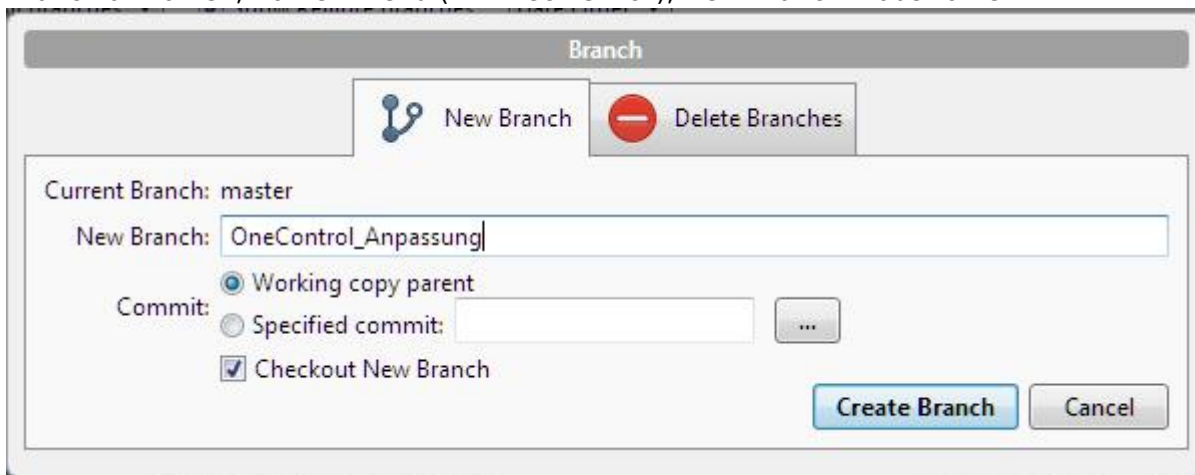
Achtung, hier wurde der master Branch geladen!

Aktuellen Branch einstellen

Jetzt muss das Projekt auf einen aktuellen Arbeitsbranch, z.B. OneControl_Anpassung eingestellt werden:

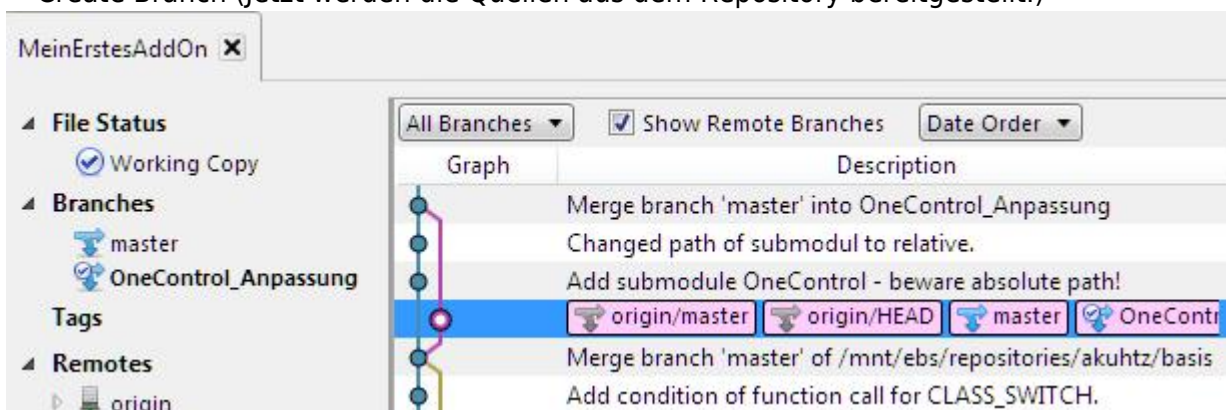


Branch anwählen, Kontextmenü (z.B. Rechtsklick), New Branch... auswählen:

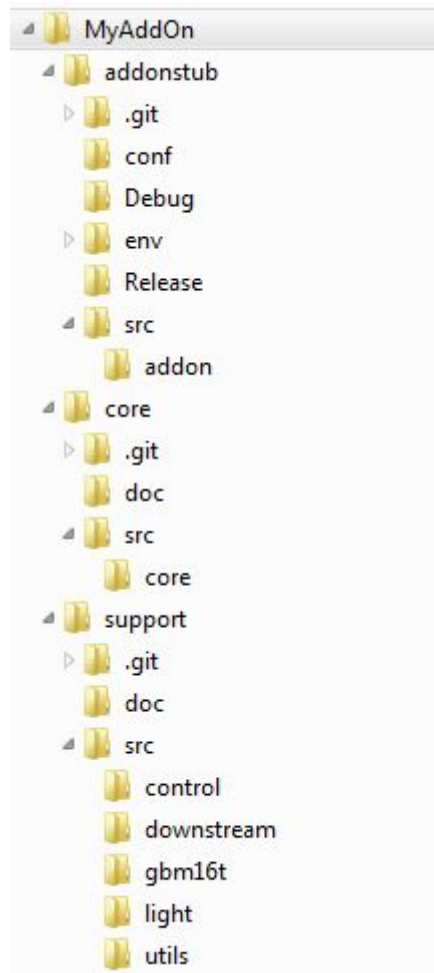


New Branch, z.B.: OneControl_Anpassung (Stand: Jan. 2014)

⇒ Create Branch (Jetzt werden die Quellen aus dem Repository bereitgestellt.)



Jetzt stehen die benötigten Quellen auf dem angegebenen Pfad zur Verfügung. Die Verzeichnisstruktur



müsste etwa wie folgt aussehen:

Jetzt ist der neueste Stand vom Haupt-Repository geladen.

Arbeiten mit AtmelStudio6

Zum Arbeiten mit dem AtmelStudio6 steht im Verzeichnis: **AddOn/env/AtmelStudio6** die Projektdatei AddOn.cproj zur Verfügung. Die muss ins AddOn-Grundverzeichnis (siehe oben am Beispiel der OneControl) kopiert werden, um die eingestellten Pfade nutzen zu können. Diese Trennung ist notwendig, da nicht jeder Entwickler mit dem AtmelStudio6 arbeitet. Somit verbleiben die „persönlichen“ Einstellung auf dem heimischen Rechner.

Mit z.B. einem Doppelklick auf die cproj-Datei startet man bestenfalls jetzt das AtmelStudio6. Das Studio legt (z.B. nach einem Speichervorgang) seine benötigten Dateien an.

Beachte: Sollen Änderungen in der Projektdatei im Repository gespeichert werden, müssen sie zuvor von hier ins Ursprungsverzeichnis zurück kopiert werden!

From:
<https://forum.opendcc.de/wiki/> - BiDiB Wiki

Permanent link:
<https://forum.opendcc.de/wiki/doku.php?id=bidiboneausrepository&rev=1430650403>

Last update: **2016/07/05 10:47**



