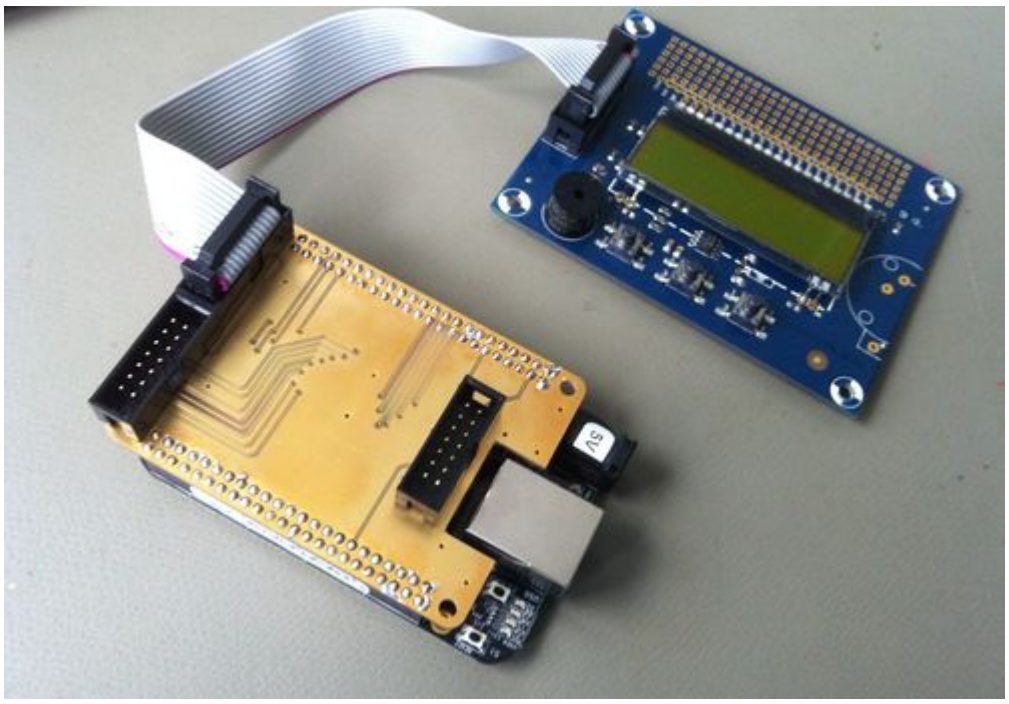

API installieren

Um die API auf dem BeagleBone Black zu instalieren, lädt man am besten das aktuelle Repository als zip-Datei herunter und entpackt es:

> root@beaglebone:~# curl -k -o api.zip
> https://codeload.github.com/embeddedprojects/gnublin-api/zip/master
> root@beaglebone:~# unzip api.zip

Bevor man nun die API für das BeaglBone Black übersetzen kann muss man noch ein paar Anpassungen in der *API-config.mk* vornehmen.

Die Zeilen 2, 12, 17 werden auskommentiert. Dafür wird bei den Zeilen 6, 14, 19 das # am Anfang entfernt. Die Datei sieht dann wie folgt aus:

```
 1. #Crosscompiler for Gnublin
 2. #CXX := arm-linux-gnueabi-g++
 3. #Crosscompiler for Raspberry Pi:
 4. #CXX := arm-linux-gnueabihf-g++
 5. #Compiler for onboard compilation:
 6. CXX := g++
 7.
 8. #Compilerflags:
 9. CXXFLAGS = -Wall
10.
11. #Architecture for gnublin:
12. #Architecture = armel
13. #Architecture for raspberryPi an beaglebone black:
14. Architecture = armhf
```

```
15.
16.  #Define which Board you want:
17.  #BOARD := GNUBLIN
18.  #BOARD := RASPBERRY_PI
19.  BOARD := BEAGLEBONE_BLACK
20.
21.  #DO NOT EDIT BEYOND THIS LINE!
22.  BOARDDEF := -DBOARD=$(BOARD)
```

Anschließend kann man die API übersetzen

```
root@beaglebone:~# make
```

Um die SPI-Schnittstelle zu nutzen zu können, muss man erst noch ein paar Einstellungen vornehmen.

GNUBLIN Tools installieren

```
root@beaglebone:~# mkdir -p /usr/local/bin
```

```
root@beaglebone:~# cd gnublin-api
```

```
root@beaglebone:~# make install
```

API anwenden

Um die API in eigenen Programmen anwenden zu können benötigt man lediglich die beiden Dateien 'gnublin.h' und 'gnublin.cpp' am bestem im selben Ordner wie das eigene Projekt.

Das ganze funktioniert wie mit dem GNUBLIN beschrieben.

- mit Codeblocks, Eclipse oder per Makefile
- Beispielprogramme

Das einzige was anders ist, ist dann man beim Kompilieren das Attribut **-DBOARD=BEAGLEBONE_BLACK** mit angeben muss, damit die API weiß, dass es sich um ein Beaglebone handelt.

Der Kompilierbefehl lautet dann:

```
root@beaglebone:~# g++ -o meinProgramm meinProgramm.cpp gnublin.cpp
-DBOARD=BEAGLEBONE_BLACK
```

SPI aktivieren

[Quelle: https://groups.google.com/d/msg/beagleboard/jYpYHZPl1ig/dkRGcI0wcNYJ]

Die SPI-Schnittstelle ist ist beim Beaglebone Black nicht standardmäßig aktiviert, dies muss man manuell vornehmen.

Die Schnittstellenzuweisung auf die einzelnen Pins funktioniert beim BeagleBone Black über den sogenannte Device Tree. Um nun SPI zu aktivieren muss ein Overlay in den Device Tree eingefügt werden.

Der Beaglebone Black stellt zwei SPI Schnittstellen zu Verfügung, wir werden hier **SPI0** aktiveren, da dies der Standard in der API ist und der SPI1 sich mit dem HDMI-Ausgang überschneidet und so wesentlich aufwendiger ist diesen einzurichten.

Overlay erstellen

Für das Overlay erstellen wir eine Datei BB_SPI0DEV-00A0.dts mit folgendem Inhalt:

```
/*
 * Copyright (C) 2013 CircuitCo
 *
 * Virtual cape for SPI0 on connector pins P9.22 P9.21 P9.18 P9.17
 * Modified to enable /dev/spidev1.0 by Carl Johnson
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation.
 */
/dts-v1/;
/plugin/;

/ {
    compatible = "ti,beaglebone", "ti,beaglebone-black";

    /* identification */
    part-number = "BB-SPI0DEV";
    version = "00A0";

    /* state the resources this cape uses */
    exclusive-use =
        /* the pin header uses */
        "P9.17",    /* spi0_cs0 */
        "P9.18",    /* spi0_d1 */
        "P9.21",    /* spi0_d0 */
        "P9.22",    /* spi0_sclk */
        /* the hardware ip uses */
        "spi0";

    fragment@0 {
        target = <&am33xx_pinmux>;
        __overlay__ {
            bb_spi0_pins: pinmux_bb_spi0_pins {
                pinctrl-single,pins = <
```

```
                    0x150 0x30  /* spi0_sclk.spi0_sclk, INPUT_PULLUP | MODE0
*/
                    0x154 0x30  /* spi0_d0.spi0_d0, INPUT_PULLUP | MODE0 */
                    0x158 0x10  /* spi0_d1.spi0_d1, OUTPUT_PULLUP | MODE0 */
                    0x15c 0x10  /* spi0_cs0.spi0_cs0, OUTPUT_PULLUP | MODE0
*/
                >;
            };
        };
    };

    fragment@1 {
        target = <&spi0>;   /* spi0 is numbered correctly */
        __overlay__ {
            status = "okay";
            pinctrl-names = "default";
            pinctrl-0 = <&bb_spi0_pins>;
            #address-cells = <1>;
            #size-cells = <0>;
            spidev@0 {
                #address-cells = <1>;
                #size-cells = <0>;
                spi-max-frequency = <24000000>;
                reg = <0>;
                compatible = "linux,spidev";
            };
        };
    };
};
```

Diese Datei muss nun compiliert werden:

```
root@beaglebone:~# opkg install dtc
root@beaglebone:~# dtc -O dtb -o /lib/firmware/BB-SPI0DEV-00A0.dtbo -b 0 -@
BB-SPI0DEV-00A0.dts
```

Nun kann SPI mit folgendem Befehl während der Laufzeit aktiviert werden:

```
root@beaglbone:~# echo BB-SPI0DEV > /sys/devices/bone_capemgr.*/slots
```

Es sollte nun das Device **/dev/spidev1.0** angelegt worden sein. Dies kann man mit *ls /dev* überprüfen.

SPI beim Systemstart aktivieren

Ab dem Kernel 3.8.13-r23a.17 kann das SPI-Overlay auch während des Systemstarts automatisch aktiviert werden.

Dazu ist ein Eintrag in der Datei *uEnv.txt* in der Bootpartition notwendig.

Diese Datei kann man auf zwei Wegen bearbeiten:

- am Desktop-PC, wenn der BeagleBone per USB angeschlossen ist. Dort findet man sie in dem Flashlaufwerk, das der Beaglbone anlegt
- auf dem Beaglebone selbst:

```
root@beaglebone:~# if [ ! -d /mnt/boot ]; then mkdir /mnt/boot; fi
root@beaglebone:~# mount /dev/mmcblk0p1 /mnt/boot
```

Die Datei findet man dann unter: */mnt/boot/uEnv.txt*.

Am Ende der "http://wiki.gnublin.orgoptargs"http://wiki.gnublin.org-Zeile fügt man folgendes ein:

```
capemgr.enable_partno=BB-SPI0DEV
```

Nun sollte beim Systemstart die SPI-Schnittstelle automatisch aktiviert werden.