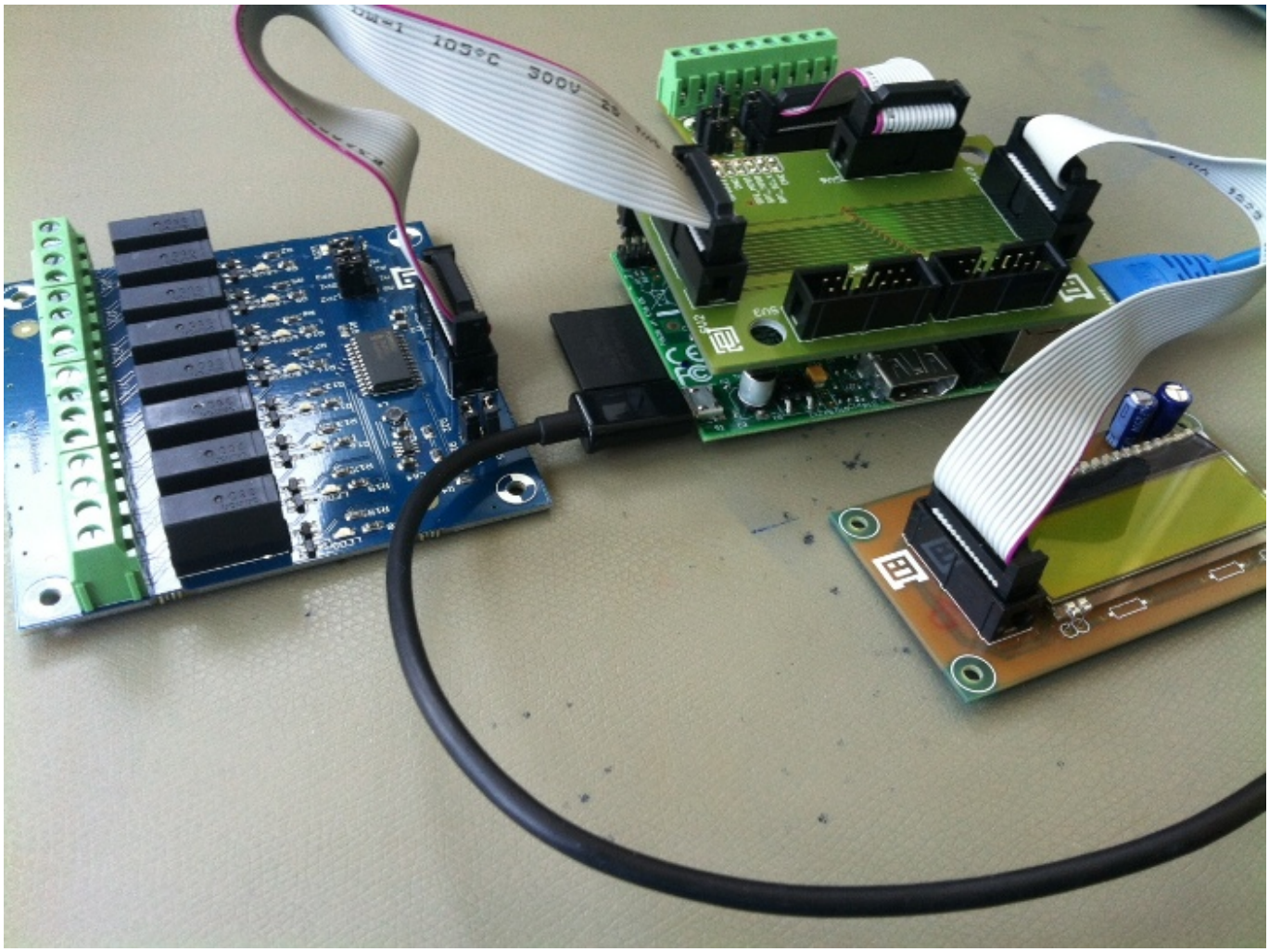
Der riesen Vorteil von GNUBLIN sind die vielen Module. Zu jedem Modul gibt es ein kleines Tool auf der Kommandozeile, mit dem man das Modul bereits ansteuern kann. Es passiert das, was man erwartet: Motor dreht sich, Temperaturwert wird ausgegeben, am Display erscheint Text, usw.

Passend zu jedem Modul gibt es in der API Aufrufe um eigene Programme in C und C++ schreiben zu können. Neben der Unterstützung für Gnublin wird und wurde ein Support für RaspberryPi umgesetzt.

Auf diesen Seiten werden alle Informationen gesammelt.

- API
- Beispielprogramme für die API
- YouTube Film über die GNUBLIN & RaspberryPi Module



# Quickinstall

Das RaspberryPi muss gestartet sein und eine Verbindung in das Internet haben. Mit git kann man jetzt das Repository klonen. Sollte git auf RaspberryPi nicht installiert sein macht man das wie folgt:

```
pi@raspberrypi ~ $ sudo apt-get install git
```

Jetzt kann man das Repository abholen:

```
pi@raspberrypi ~ $ git clone https://github.com/embeddedprojects/gnublin-api.git
```

Wechselt in das Verzeichnis:

```
pi@raspberrypi ~ $ cd gnublin-api
```

Bevor man die API übersetzen kann, muss man angeben, das die API auf einem RaspbberryPi laufen soll. Dazu ändert man in der Datei **API-config.mk** zwei Zeilen ab:

Folgenden Zeilen müssen auskommentiert werden: 2, 12, 17. Dafür muss bei folgenden zeilen das # am Anfang entfernt werden: 6, 14, 18. Die API-config.mk 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:
14. Architecture = armhf
15.
16. #Define which Board you want:
17. #BOARD := GNUBLIN
18. BOARD := RASPBERRY_PI
19.
20. #DO NOT EDIT BEYOND THIS LINE!
21. BOARDDEF := -DBOARD=$(BOARD)
```

Nun kann man die API übersetzen und die Programme und Beispiele installieren (kann ca. 2-3min dauern):

```
pi@raspberrypi ~ $ make && sudo make install
```

Die API braucht folgende Treiber:

```
pi@raspberrypi ~ $ sudo modprobe spi-bcm2708
pi@raspberrypi ~ $ sudo modprobe i2c-bcm2708 baudrate=21500
pi@raspberrypi ~ $ sudo modprobe i2c-dev
```

Diese befinden sich in der aktuellen RaspberryPi Version.

Optional kann man die Treiber fest in die Datei /etc/modules eintragen, diese werden dann beim booten automatisch geladen:

```
spi-bcm2708
i2c-bcm2708 baudrate=21500
i2c-dev
```

Pro Zeile ein Modul. Die Treiber werden allerdings erst bei einem neustart geladen!

Hat man jetzt bereits eine Erweiterungsplatine angeschlossen, kann man sofort die kleinen Tools testen.

## Die API

Das ganze geht wie bei Gnublin beschrieben:

- mit Codeblocks, Eclipse oder per Makefile
- Beispielprogramme

Das einzige, was anders ist, ist das man

```
#define BOARD RASPBERRY_PI
```

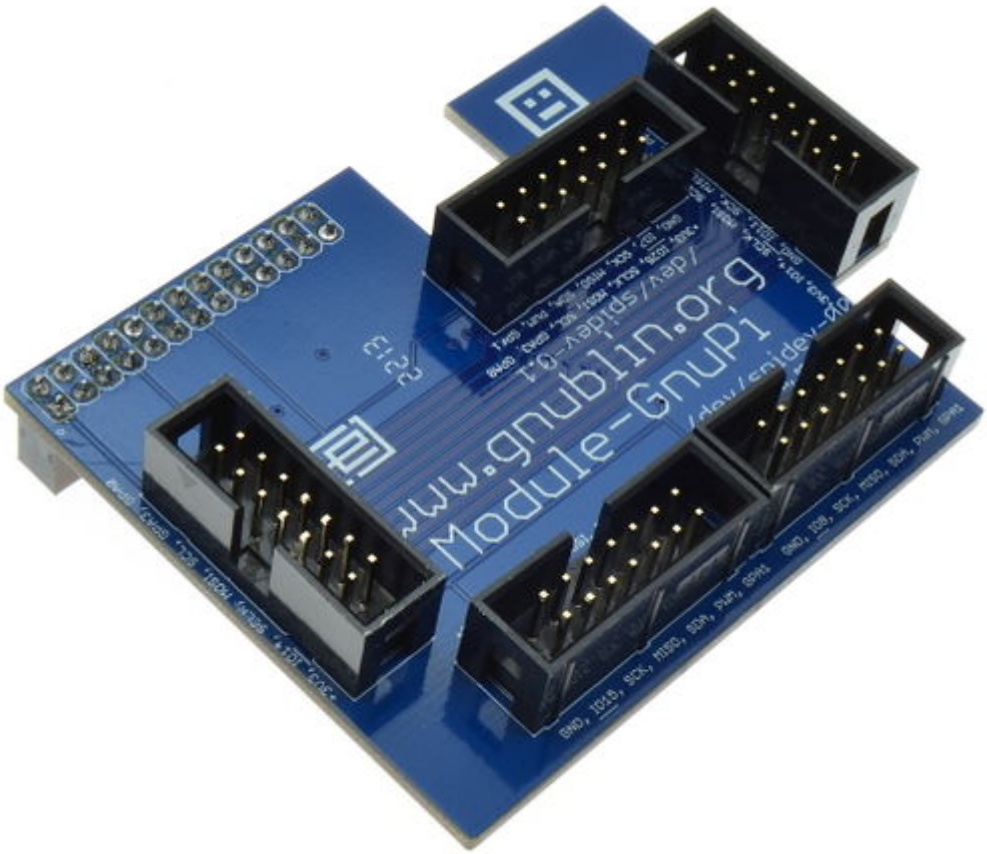
**vor** dem

```
#include "http://wiki.gnublin.orggnublin.h"http://wiki.gnublin.org
```

setzten muss.

# Die Adapter Platine

Die Adapterplatine wird direkt mit dem RaspberryPi verbunden. Auf der Platine befinden sich 5 weitere Anschlussmöglichkeiten für Module. Module werden einfach per Flachbandkabel angeschlossen.




Das Display kann nur an den mit der Beschriftung "http://wiki.gnublin.org/dev/spidev"http://wiki.gnublin.org versehen Connectorn betrieben werden.

## SPI

Der SPI Bus ist auf zwei Buchsen herausgeführt, jeweils mit unterschiedlicher Chipselect-Leitung. Diese sind mit /dev/spidev0.0 und /dev/spidev0.1 gekennzeichnet.

In der API und den GNUBLIN-Tools wird standardmäßig der CS 0 verwendet, also die Buchse mit /dev/spidev0.0. Mann kann aber auch jederzeit den anderen verwenden, wenn man den entsprechenden CS-Pin angibt.

Beim Tool "http://wiki.gnublin.orggnublin-dogm"http://wiki.gnublin.org kann der andere Connector so genutzt werden:

```
pi@raspberry:~# sudo gnublin-dogm -g 4 -c 1 -i
```

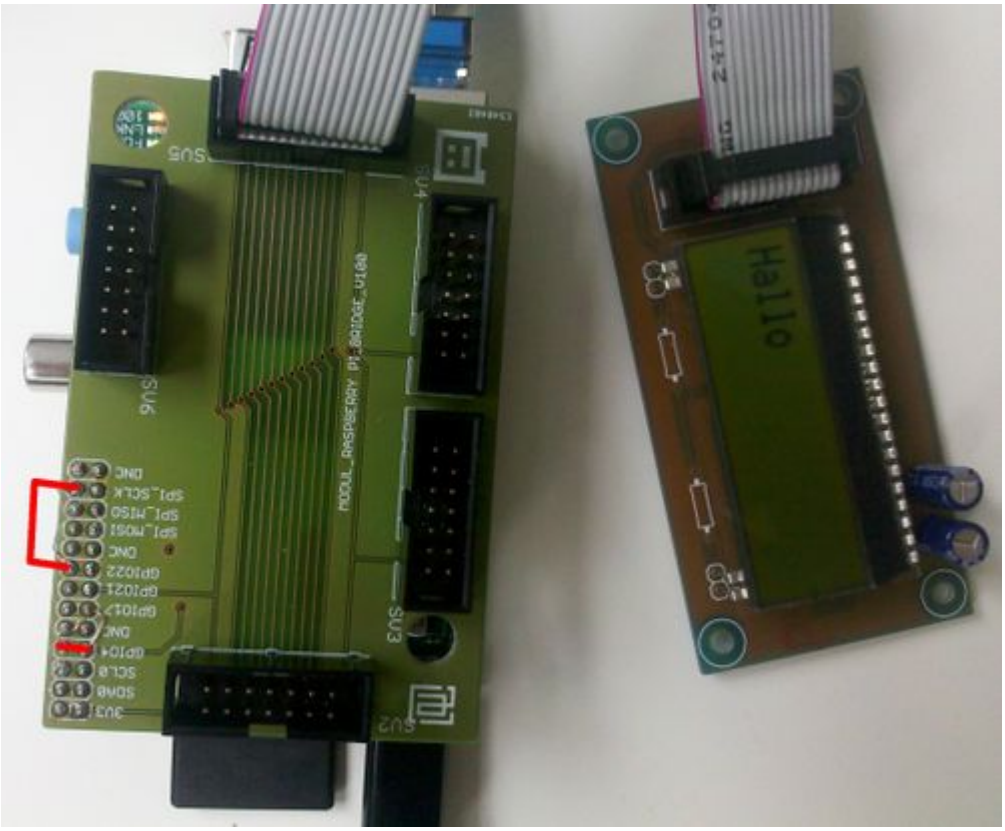
Der RS-Pin liegt auf dem GPIO 4 und als SPI kommt spidev0.1 zum Einsatz. Auf spidev0.0 ist es GPIO 25.

## alte Version

**Bei der ersten Version des GnuPi (grüne Platine) ist zudem zu beachten:** Das Display funktioniert nur an folgendem Stecker. Zusätzlich muss man die zwei rot markierten Brücken einlöten (GPIO8 -> GPIO23; GPIO4 -> GPIO14):

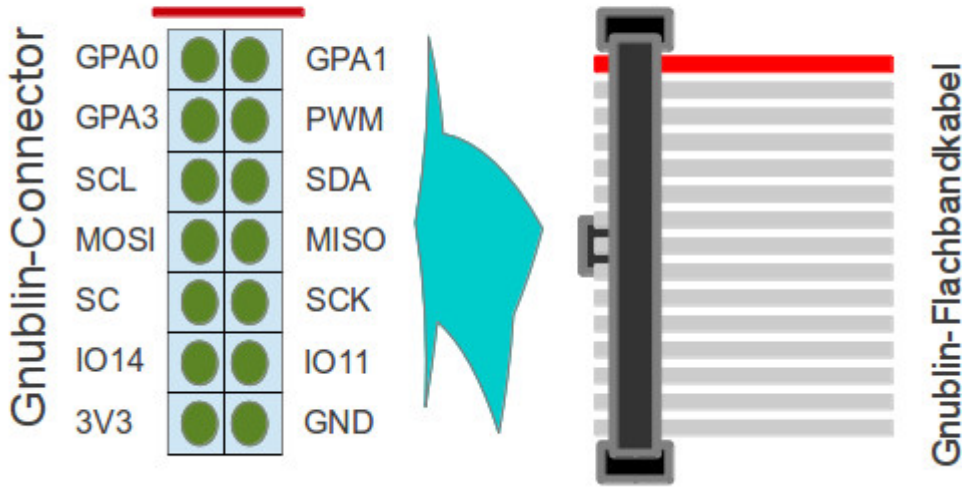
**Hinweis:** Der GPIO14 muss vor der Verwendung per Hand auf LOW gesetzt werden!

```
pi@raspberrypi ~ $ sudo su
root@raspberrypi:/home/pi# echo "http://wiki.gnublin.org14"http://wiki.gnublin.org >
/sys/class/gpio/export
root@raspberrypi:/home/pi# echo "http://wiki.gnublin.orgout"http://wiki.gnublin.org >
/sys/class/gpio/gpio14/direction
root@raspberrypi:/home/pi# exit
```




# Pin Belegung

Gnublin-Standard




## Signale die 1:1 an alle Buchsen gehen

- I2C (SDA, SCL)
- SPI (MISO, MOSI, SCK)
- GP (3 Stück)
- GND
- 3.3V

## Unterschiede an den Buchsen

Verwendet man I2C Module, kann man diese anstecken wo man möchte. Bei SPI Module muss man eine CS-Leitung und zum Teil noch eine IRQ-Leitung definieren.

**I2C**

- GNUBLIN Module-Temperature
- GNUBLIN Module-Step
- GNUBLIN Module-Relay
- GNUBLIN Module-ADC
- GNUBLIN Module-RTC
- GNUBLIN Module-Portexpander

**SPI**

- GNUBLIN Module-LCD_2x16
- GNUBLIN Module-LAN
- GNUBLIN Module-CAN
- Module-RS232
- Module-RS485

- Download Schaltplan

## Toolchain

TDB (passen wir gerade an)

```
wget https://raw.github.com/embeddedprojects/gnublin-api/master/gnublin.cpp
wget https://raw.github.com/embeddedprojects/gnublin-api/master/gnublin.h
```

Es wird als g++ auf RaspberryPi arm-linux-gnueabihf-g++ verwendet! Aktuell habe ich einen Softlink angelegt, damit wir die Makefiles nicht anpassen müssen.

```
sudo ln -s /usr/bin/g++ /usr/local/bin/arm-linux-gnueabi-g++
```

# Doku Portierung Gnublin Tools

In diesem Abschnitt findet man Infos zu den portieren Tools und API Schnittstellen.

## GNUBLIN Module-LCD 2x16

- Schnittstelle: SPI
- Tool: OK
- API: OK

Infos zum Tool:

```
pi@raspberrypi ~ $ sudo modprobe spi-bcm2708
pi@raspberrypi ~ $ sudo gnublin-dogm -i
pi@raspberrypi ~ $ sudo gnublin-dogm -n -w
"http://wiki.gnublin.orgHello"http://wiki.gnublin.org
```

- CS Pin (hängt auf Modul Pin 11 : RaspberryPi Pin 8 CE0)
- RS Pin (hängt auf Modul Pin 14 : RaspberryPi Pin 4)

**Hinweis:** Der GPIO14 muss vor der Verwendung per Hand auf LOW gesetzt werden!

```
pi@raspberrypi ~ $ sudo su
root@raspberrypi:/home/pi# echo "http://wiki.gnublin.org14"http://wiki.gnublin.org >
/sys/class/gpio/export
root@raspberrypi:/home/pi# echo "http://wiki.gnublin.orgout"http://wiki.gnublin.org >
/sys/class/gpio/gpio14/direction
root@raspberrypi:/home/pi# exit
```

## GNUBLIN Module-Temperature

- Schnittstelle: I2C
- Tool: OK

- API: OK

Infos zum Tool:

```
pi@raspberrypi ~ $ sudo modprobe i2c-bcm2708
pi@raspberrypi ~ $ sudo modprobe i2c-dev

pi@raspberrypi ~ $ sudo gnublin-lm75
```

## GNUBLIN Module-Step

- Schnittstelle: I2C
- Tool: OK
- API: OK

Infos zum Tool:

```
pi@raspberrypi ~ $ sudo modprobe i2c-bcm2708
pi@raspberrypi ~ $ sudo modprobe i2c-dev
```

## GNUBLIN Module-Relay

- Schnittstelle: I2C
- Tool: OK
- API: OK

Infos zum Tool:

```
pi@raspberrypi ~ $ sudo modprobe i2c-bcm2708
pi@raspberrypi ~ $ sudo modprobe i2c-dev
```