

Einbau Onboard

Notlösung

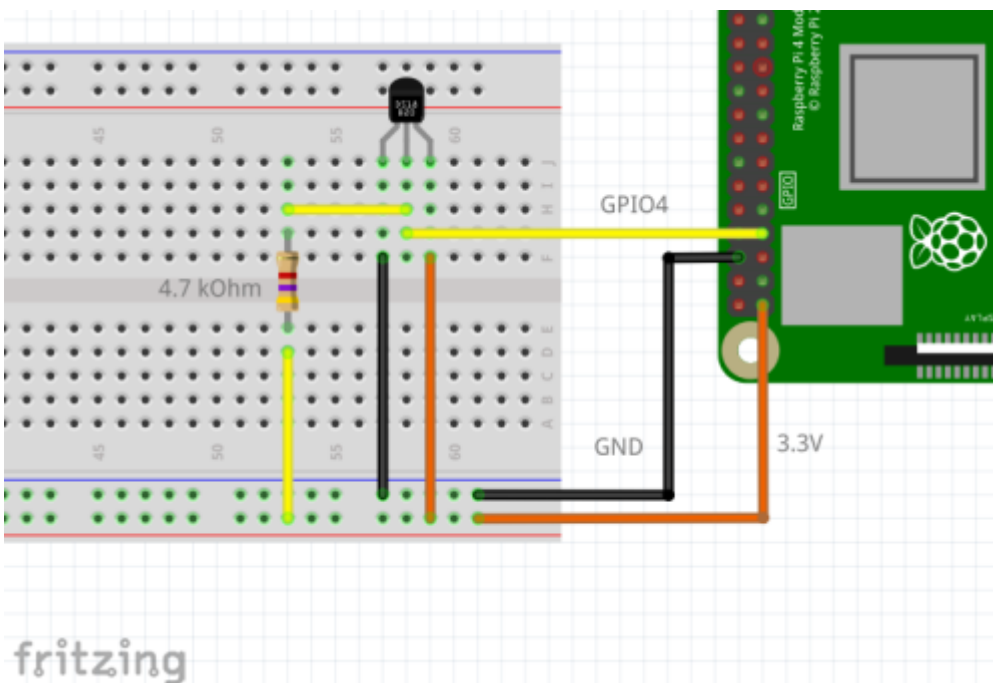


Bitte betrachtet die hier vorgestellte Nutzung eines GPIO-Busmasters nur als Notlösung. Auf Grund eines Kernel-Bugs (Linux) friert Euch das System bei häufigen Device-Wechseln am GPIO-Busmaster ein (z. B. bei Verwendung von iButtons). Auch läuft der GPIO nur mit 3,3V, 1-Wire läuft hingegen standardmäßig mit 5V. Für einige wenige Temperatursensoren funktioniert das gut, will man viele Devices an einem langen (Haus-)Bus betreiben, nutzt einen richtigen Busmaster.

Das Plugin unterstützt auch den internen 1-Wire-Busmaster. Dieser emuliert per Software das 1-Wire-Protokoll auf GPIO Pin 4. Um einige wenige Sensoren auszulesen funktioniert das ohne Probleme. Aktuell kann der GPIO Pin 4 vom Plugin selbst nicht umprogrammiert werden. Händisch geht das allerdings, siehe Kapitel unten. Zum Anschluss wird noch ein 4,7kOhm PullUp-Widerstand benötigt. Die Verschaltung ist unten dargestellt.

Hardware

Die Verschaltung erfolgt wie unten dargestellt: Ihr benötigt 3.3V (**nicht** 5V! Damit zerstört ihr euren Raspberry), GND und GPIO 4. Zwischen GPIO 4 und 3.3V kommt ein PullUp-Widerstand von 4.7 kOhm.



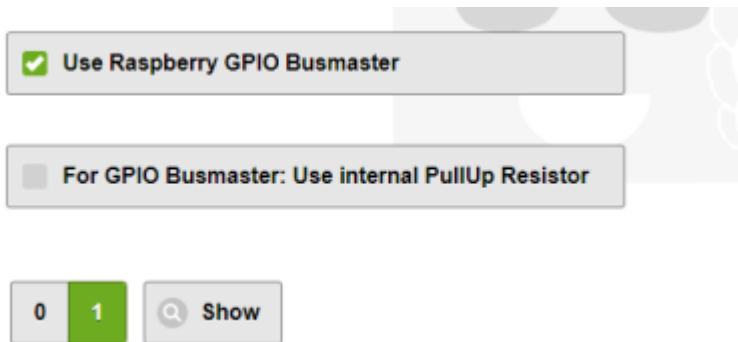
Laut mikrocontroller.net scheint es manchmal hilfreich zu sein, einerseits einen kleineren Pullup-Widerstand zu verwenden (bspw. 2,2 kOhm; damit wird die 3,3V Versorgung mit 1,5 mA belastet) und einen kleinen Widerstand nach dem GPIO *in Reihe* zu schalten.



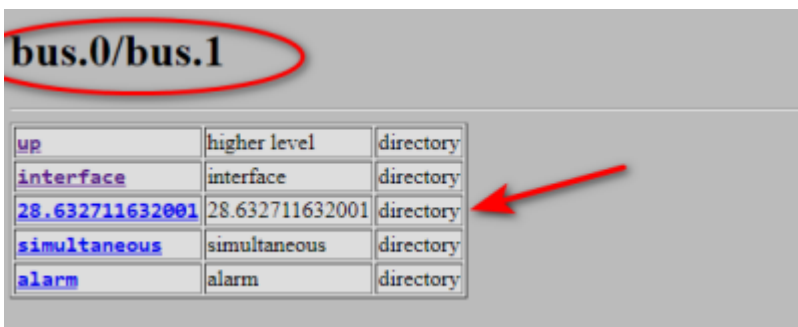
Hintergrund: Die Slew Rate mit dem großen Pullup-Widerstand ist durch die 1,5 mA Strom begrenzt, während der Raspberry Pi 3 oder Pi 4 mit 16 mA auf Masse zieht (entspräche einem Pulldown-Widerstand von ~ 200 Ohm). D.h. es wird sehr schnell auf Masse gezogen, was in großen Netzen zu Überschwingern und Reflektionen führen kann. Ein kleiner Widerstand in Reihe begrenzt die Slew Rate in Richtung Masse etwas. Beide Widerstände können und sollten der Netzwerkgröße angepasst werden.

Software

Aktiviert zunächst den GPIO Busmaster und **deaktiviert** den internen PullUp-Widerstand! Dieser soll theoretisch funktionieren, praktisch habe ich damit keine vernünftigen Resultate erzielt. Vermutlich ist der interne PullUp-Widerstand des Raspberry mit 10 kOhm zu groß. Nutzt wie oben geschrieben einen externen 4.7 kOhm PullUp-Widerstand! Nachdem ihr die Einstellungen gemacht habt, müsst ihr neu Booten!



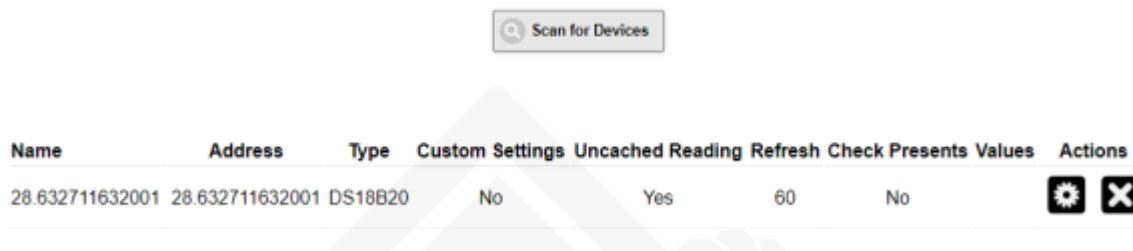
Nach dem Booten kommt zunächst der Dienst "OWFS2MQTT" noch nicht hoch. Das ist erst einmal normal. Geht jetzt in die OWFS WebUI und schaut, ob euer Sensor erkannt wurde. Es werden zwei Busse erkannt, die ihr im Verzeichnisbaum beide unter "Bus.0" findet. Der "Bus.0/Bus.0" heisst "W1 bus monitor" - die Funktion ist mir nicht ganz klar, auf jeden Fall könnt ihr diesen Bus ignorieren - er beinhaltet keinerlei Sensoren. Euch interessiert "Bus.0/Bus.1", wo ihr auch euren Sensor mit seiner Seriennummer finden solltet:





Wenn ihr euren Sensor nicht findet, überprüft eure Verkabelung. Wenn das geklappt hat, geht zurück in die Plugin-Oberfläche und aktiviert nun "Bus 1" für das Scannen nach Devices:



Die Services des Plugins nun neu starten. Dann in das "Devices" Tab des Plugins wechseln. Euer Bus sollte durchsucht werden und auch euer Sensor hier in der Pluginoberfläche angezeigt werden:



Name	Address	Type	Custom Settings	Uncached Reading	Refresh	Check Presents	Values	Actions
28.632711632001	28.632711632001	DS18B20	No	Yes	60	No		 

Wenn das funktioniert hat, startet die Services ein letztes Mal neu. Jetzt sollte auch der Dienst "OWFS2MQTT" mitstarten und ihr solltet die Werte des Sensors in eurem MQTT Broker bzw. in der Incoming Overview oder im Finder des MQTT Widgets sehen können.

Geschafft! 😊

Raspberry GPIO Busmaster: Anpassungen und Empfehlungen

Standardmäßig wird nach dem Neustart des Raspberry Pi's und bei aktivem "Use Raspberry GPIO Host Master" folgende Zeile in die `/boot/config.txt` am Ende eingefügt:

```
dtoverlay=w1-gpio,gpiopin=4,pullup=on
```

D.h. es wird der Standardpin GPIO 4 für das OneWire Signal genutzt, so wie bspw. auch hier oben im Anschlussdiagramm gezeigt. Falls aus irgendwelchen Gründen ein anderer Pin genutzt werden soll (bspw. im Argon ONE Gehäuse liegt der Power Button auf GPIO 4), kann der genutzte Pin wie folgt geändert werden:

[Eine SSH Verbindung als superuser \("su"\) aufbauen](#)

Folgenden Befehl ausführen:

```
sudo nano /boot/config.txt
```

Anschließend ans Ende des Files scrollen, bis Ihr den Eintrag `dtoverlay=w1-gpio,gpiopin=4,pullup=on` findet. Diesen entsprechend Eurer Wünsche anpassen. Es bieten sich GPIO 17 (Pin 11) und 27 (Pin 13) an. Eine Übersicht über alle zur Verfügung stehenden GPIO Pins findet Ihr bspw. [hier auf de.pinout.xyz](#).

Falls Ihr nicht den parasitären Modus nutzt, sondern die Betriebsspannung 5V jedem Sensor zur Verfügung stellt, würde ich empfehlen, `pullup=off` zu setzen. Der neue Eintrag schaut dann bspw. so aus:

```
[all]
#dtoverlay=vc4-fkms-v3d
dtoverlay=w1-gpio,gpiopin=17,pullup=off
```

Anschließend speichern und schließen: Strg+O, "Y"es wir wollen überschreiben, Strg+X und fertig:

```
exit
```

Zum Schluss noch einige Hinweise zur Verwendung des GPIO Busmasters:

- Laut [w1-gpio Kernel Doku](#) sind auch mehrere GPIO Busmaster parallel möglich. Dafür müsst Ihr in der /boot/config.txt mehrere Zeilen einfügen und jeweils den GPIO abändern. Falls Eure Netze instabil werden, könntet Ihr so die Last auf mehrere GPIO Busmaster aufteilen.
- Laut [mikrocontroller.net](#) scheint es hilfreich zu sein, einerseits einen recht kleinen Pullup-Widerstand zu verwenden (bspw. 2,2 kOhm; damit wird die 3,3V Versorgung mit 1,5 mA belastet) und einen kleinen Widerstand nach dem GPIO *in Reihe* zu schalten. Hintergrund: Die Slew Rate mit dem großen Pullup-Widerstand ist durch die 1,5 mA Strom begrenzt, während der Raspberry Pi 3 oder Pi 4 mit 16 mA auf Masse zieht (entspräche einem Pulldown-Widerstand von ~ 200 Ohm). D.h. es wird sehr schnell auf Masse gezogen, was in großen Netzen zu Überschwingern und Reflektionen führen kann. Ein kleiner Widerstand in Reihe begrenzt die Slew Rate in Richtung Masse etwas.

From:

<https://wiki.loxberry.de/> - **LoxBerry Wiki - BEYOND THE LIMITS**

Permanent link:

https://wiki.loxberry.de/plugins/1_wire_ng/einbau_onboard

Last update: **2024/02/19 06:11**