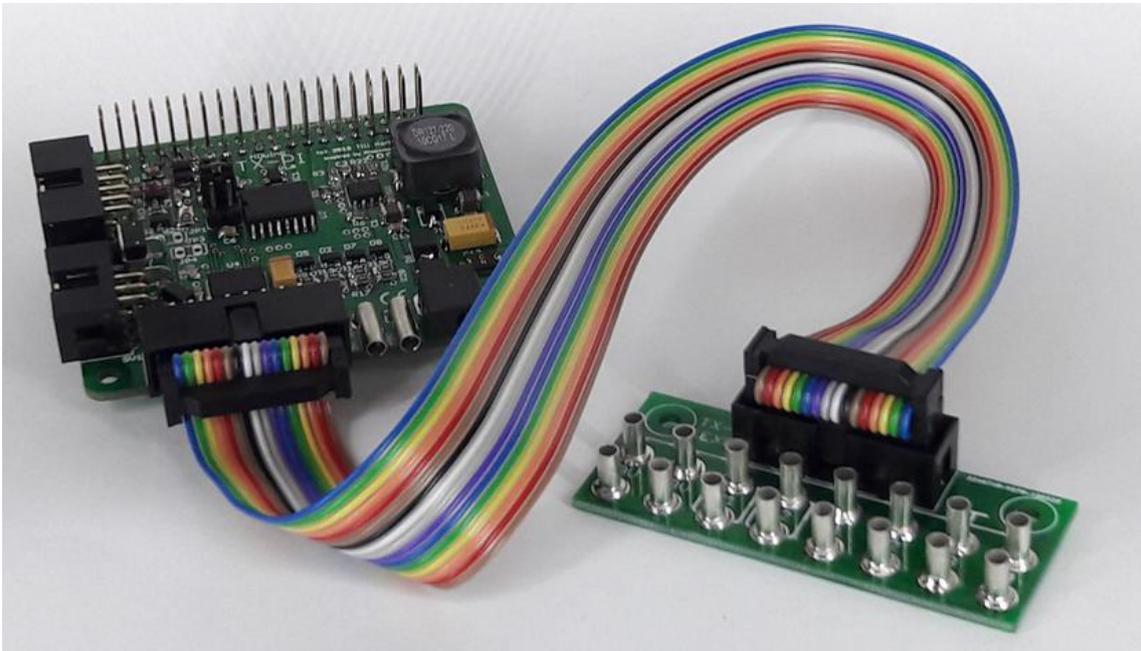


ft-HAT

ein **fischertechnik**-kompatibler HAT für den Raspberry-Pi



Kurzanleitung

Dr.-Ing. Till Harbaum

25. Juni 2020

© 2019 Dr.-Ing. Till Harbaum <till@harbaum.org>

Project homepage: <http://tx-pi.de>

Forum: <https://forum.ftcommunity.de/>

Inhaltsverzeichnis

1	Einleitung	4
1.1	Lieferumfang	4
2	Inbetriebnahme	6
2.1	Spannungsversorgung	6
2.2	Kontrolle der Spannungsversorgung durch den Raspberry-Pi ("Auto-Aus")	7
3	I²C-Busse und -Komponenten	9
3.1	Aktivierung der I ² C-Busse am Raspberry-Pi	9
3.2	Nutzung der internen I ² C-Komponenten des ft-HAT	11
3.2.1	EEPROM-Speicher AT24C32	11
3.2.2	RTC-Uhrenbaustein DS3231	11
3.3	Nutzung von externen I ² C-Komponenten am ft-HAT	12
3.3.1	fischertechnik-Umweltsensor 167358	12
3.3.2	fischertechnik-Kombisensor 158402	13
3.3.3	TX- und ftDuino-Sensoren	13
4	Anschluss an fischertechnik-Modelle	14
4.1	Eingänge für Schalter, Taster u.ä.	15
4.2	Ausgänge für Motoren, Lampen u.ä.	15
5	Verwendung mit einem Aufsteckbildschirm	17

Kapitel 1

Einleitung

Dies ist die deutsche Kurzanleitung. Mehr Informationen können der detaillierteren englischen Anleitung entnommen werden.

Der ft-HAT ist ein sogenannter HAT, eine Aufsteckplatine für den Raspberry-Pi¹. Der ft-HAT stellt die Verbindung zwischen dem Raspberry-Pi und der fischertechnik-Welt her. Der ft-HAT bietet:

fischertechnik-Spannungsversorgung: Der ft-HAT erlaubt es, den Raspberry-Pi aus fischertechnik-üblichen 9-Volt Spannungsversorgungen zu betreiben.

fischertechnik-Eingänge: Vier fischertechnik-kompatible digitale Eingänge erlauben das Abfragen von fischertechnik-Tastern, -Schaltern u.ä.

fischertechnik-Ausgänge: Zwei fischertechnik-kompatible analoge Ausgänge erlauben das Steuern von fischertechnik-Motoren, -Lampen u.ä.

fischertechnik-I²C: Zwei fischertechnik-kompatible Anschlüsse erlauben den Betrieb sämtlicher I²C-Sensoren, die für den fischertechnik-TX- und den fischertechnik-TXT-Controller entwickelt wurden am Raspberry-Pi.

Display-Anschluss: Der ft-HAT führt einigen zum Betrieb eines Aufsteckbildschirms am Raspberry-Pi benötigten Signale weiter, so dass sich ein Bildschirm auf den ft-HAT aufstecken lässt.

Echtzeituhr: Die im ft-HAT verbaute Echtzeituhr (RTC) erlaubt u.a. ein zeitgesteuertes Einschalten des ft-HAT und des Raspberry-Pi.

1.1 Lieferumfang

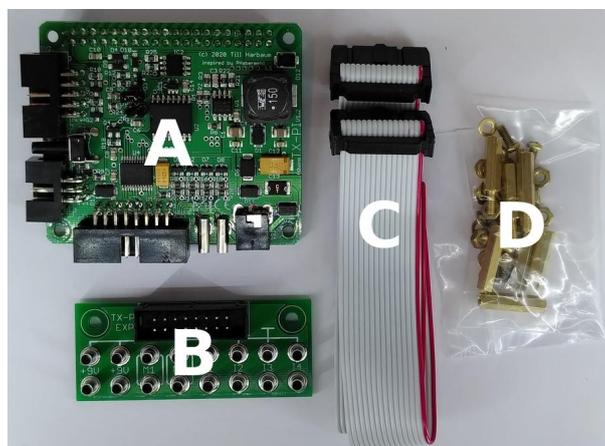


Abbildung 1.1: Lieferumfang des ft-HAT

¹Raspberry-Pi, <https://www.raspberrypi.org/>

Im Lieferumfang des ft-HAT befinden sich:

- A:** die ft-HAT-Aufsteckplatine
- B:** die ft-HAT-E/A-Erweiterungsplatine
- C:** das ft-HAT-Erweiterungskabel
- D:** der ft-HAT-Montagesatz

Das Erweiterungskabel verbindet den ft-HAT mit der Erweiterungsplatine. Der Montagesatz dient der stabilen Montage des ft-HAT auf dem Raspberry-Pi und ggf. der Befestigung eines Bildschirms.

Kapitel 2

Inbetriebnahme

Der ft-HAT kann sofort auf dem Raspberry-Pi installiert werden, indem er auf die 40-polige Erweiterungsleiste des Raspberry-Pi aufgesteckt wird. Der ft-HAT kann so z.B. bereits während der Betriebssystem-Installation und Inbetriebnahme des Raspberry-Pi zur Spannungsversorgung genutzt werden.



Abbildung 2.1: Raspberry-Pi4 mit aufgestecktem ft-HAT

Mit Hilfe des Montagesatzes kann der ft-HAT stabil mit dem Raspberry-Pi verschraubt werden.

2.1 Spannungsversorgung

Der ft-HAT kann direkt an ein fischertechnik-übliches Netzteil angeschlossen werden, wie es z.B. mit dem fischertechnik-TXT-Controller ausgeliefert wird. Alternativ kann die Versorgung über die beiden neben der Versorgungsbuchse liegenden Steckhülsen z.B. aus einem 9-Volt-Batteriefach oder dem fischertechnik-Akku versorgt werden.

Über zwei Jumper kann das Verhalten der Spannungsversorgung beeinflusst werden:

Auto-An Ist dieser Jumper aufgesteckt, so startet der ft-HAT den Raspberry-Pi direkt nach dem Anschluss einer Spannungsversorgung. Ist dieser Jumper nicht gesteckt, so muss der ft-HAT manuell durch Drücken des Tasters eingeschaltet werden.

Permanent-An Ist dieser Jumper gesteckt, so ist die Spannungsversorgung des ft-HAT permanent aktiv und sie kann weder durch durch Auto-An-Jumper noch durch z.B. den Raspberry-Pi selbst oder die Echtzeituhr des ft-HAT beeinflusst werden.

In der Regel ist der Auto-On-Jumper installiert und der Permanent-An-Jumper ist nicht installiert.

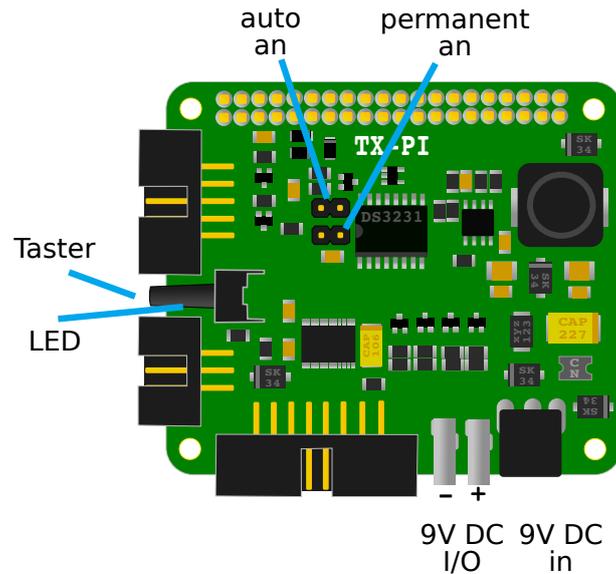


Abbildung 2.2: Spannungsversorgungskomponenten des ft-HAT

Eine Leuchtdiode (LED) unterhalb des Tasters zeigt an, wenn die Spannungsversorgung aktiv ist und der Raspberry-Pi mit Spannung versorgt wird.

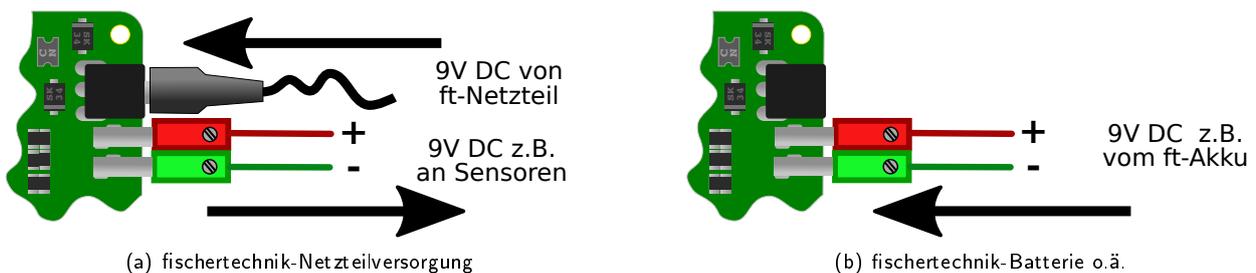


Abbildung 2.3: Spannungsversorgungsoptionen des ft-HAT

Empfohlen wird die Verwendung einer der folgenden Spannungsquellen:

- fischertechnik-Netzteil 505287 wie von fischertechnik für den TXT-Controller vertrieben.
- Die meisten modernen stabilisierten 9V-Netzteile mit passenden Steckern. Das Netzteil sollte mindestens einen Strom von 2,5 Ampere liefern können.
- Das fischertechnik-9V-Batteriegehäuse mit einer handelsüblichen 9V-Einweg-Blockbatterie.
- Der fischertechnik-Akkupack

Bei Versorgung mit dem Netzteil können die Steckhülsen als Ausgang zur Versorgung von Sensoren o.ä. verwendet werden.

2.2 Kontrolle der Spannungsversorgung durch den Raspberry-Pi ("Auto-Aus")

Einer der bekannten Nachteile des Raspberry-Pi ist, dass er keine Kontrolle über seine eigene Spannungsversorgung hat. Er kann sich also nicht wie ein PC selbst ausschalten. Der ft-HAT hebt diese Beschränkung auf und gibt dem Raspberry-Pi die Möglichkeit, sich selbst bei Bedarf abzuschalten. Diese Funktion ist besonders interessant in Zusammenhang mit der Echtzeituhr des ft-HAT, der das zeitgesteuerte (Wieder-)Einschalten ermöglicht.

Der sogenannte GPI004-Pin an der Erweiterungsleiste des Raspberry-Pi kann vom Raspberry-Pi kontrolliert werden, um die Spannung bei Bedarf zu unterbrechen. Diese Funktion des Raspberry-Pi ist in der Standard-Installation des Raspberry-Pi nicht aktiv. Um sie zu aktivieren ist die folgende Zeile in der Datei `/boot/config.txt` auf dem Raspberry-Pi zu hinterlegen:

```
dtoverlay=gpio-poweroff,gpiopin=4,active_low=1
```

Nach Einfügen dieser Zeile sollte der Raspberry-Pi die Spannungsversorgung auf dem ft-HAT komplett abschalten, sobald er heruntergefahren ist. Alle Leuchtdioden, sowohl am Raspberry-Pi als auch am ft-HAT sollten dann erlöschen. Einschalten lässt sich der Raspberry-Pi dann z.B. über die Taste zwischen den beiden I²C-Anschlüssen am ft-HAT.

Kapitel 3

I²C-Busse und -Komponenten

Die Verwendung des I²C am ft-HAT ist optional. Will man vor allem Modelle mit Hilfe von Motoren und Tastern steuern, so kann dieses Kapitel problemlos übersprungen werden.

Der Raspberry-Pi verfügt über zwei sogenannte I²C-Busse, über die diverse passende Erweiterungen angeschlossen werden können. Der ft-HAT nutzt beide I²C-Busse des Raspberry-Pi, einen um ft-HAT-interne Komponenten mit vom Raspberry-Pi aus abfragen zu können und den zweiten, im extern den den ft-HAT angeschlossene I²C-Komponenten vom Raspberry-Pi aus anzusprechen.

Auch fischertechniksetzt I²C-basierte Sensoren ein und sowohl der TX- als auch der TXT-Controller haben jeweils eine 6- bzw. 10-polige Anschlussleiste, über die sich entsprechende Sensoren anschließen lassen. Die Spannungen am TX betragen 5 Volt und am TXT 3,3-Volt, so dass sich die Sensoren auch elektrisch nur am entsprechenden Controller betrieben werden durften.



Abbildung 3.1: Original fischertechnik-Kombi- und Umweltsensor

Der ft-HAT stellt daher an seiner schmalen Seite zwei hierzu kompatible Buchsen zur Verfügung. Die linke sechspolige Buchse führt 5V-Signale und ist mechanisch und elektrisch zum TX-Controller kompatibel und kann zum Anschluss von TX-Sensoren verwendet werden. Die rechte zehnpolige Buchse führt dagegen 3,3-Volt-Signale und ist sowohl mechanisch als auch elektrisch zum TXT kompatibel und ermöglicht es so, TXT-Sensoren am Raspberry-Pi zu betreiben.

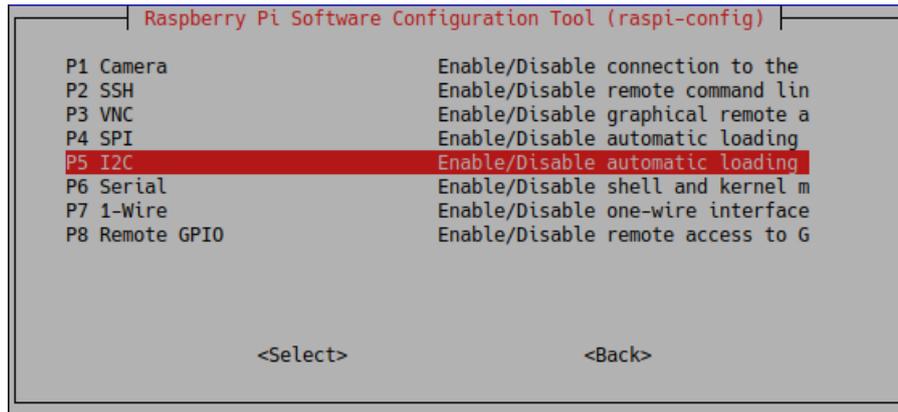
3.1 Aktivierung der I²C-Busse am Raspberry-Pi

Die I²C-Anschlüsse des Raspberry-Pi sind in der Regel in den Standardinstallationen nicht aktiviert. Mit dem Kommando `sudo raspi-config` wird dazu das Konfigurationsprogramm des Raspberry-Pi geöffnet. Unter dem Menüpunkt 6 - Interfacing Options lässt sich im Menüpunkt I2C einer der beiden I²C-Busse aktivieren.

Der zweite I²C-Bus wird durch einen zusätzlichen Eintrag in der Datei `/boot/config.txt` aktiviert. Dazu öffnet man die Datei z.B. durch das Kommando

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

und fügt die Zeile

Abbildung 3.2: Aktivierung des I²C-Bus in raspi-config

```
dtparam=i2c_vc=on
```

ein. Nach einem Reboot des Raspberry-Pi sollten beide I²C-Busse durch das `i2cdetect`-Programm des Raspberry-Pi nutzbar sein. Bei Bedarf muss dieses Programm z.B. mit dem Kommando

```
sudo apt-get install -y i2c-tools
```

installiert werden. Das Vorhandensein beider I²C-Busse kann damit überprüft werden:

```
$ i2cdetect -l
i2c-0  i2c          bcm2835 I2C adapter          I2C adapter
i2c-1  i2c          bcm2835 I2C adapter          I2C adapter
```

Der Bus `i2c-0` wird auf dem ft-HAT für die internen Komponenten (RTC und EEPROM) verwendet, der Bus `i2c-1` stellt die externe Verbindung über die 6- und 10-poligen Anschlüsse am ft-HAT her.

Auf beiden Bussen kann mit dem gleichen Programm nach Chips gesucht werden. Sind an den I²C-Anschlüssen des ft-HAT keine Geräte angeschlossen, so sind nur auf dem Bus `i2c-0` der RTC und das EEPROM unter den Adressen 50 und 60 zu finden:

```
$ i2cdetect -y 0
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  50  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  68  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
```

Sind extern z.B. der Umweltsensor und der Kombisensor von fischertechnik angeschlossen, so werden insgesamt vier Geräte erkannt:

```
$ i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  10  --  --  --  --  --  --  --  18  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
60:  --  --  --  --  --  --  --  --  68  --  --  --  --  --  --
70:  --  --  --  --  --  --  76  --  --  --  --  --  --  --  --
```

Das liegt daran, dass der Kombisensor intern aus drei unterschiedlichen Bausteinen zur Lage- und Bewegungserfassung besteht, die jeweils separat erkannt werden.

3.2 Nutzung der internen I²C-Komponenten des ft-HAT

Auf dem ft-HAT selbst befinden sich zwei I²C-Komponenten, die über den I²C-Bus i2c-1 angesprochen werden können:

- Ein EEPROM-Speicherbaustein unter Adresse 50, der eine Beschreibung des ft-HAT enthält und dem Raspberry-Pi hilft, den ft-HAT zu erkennen.
- Ein RTC-Uhrenbaustein unter Adresse 68, der die aktuelle Uhrzeit speichern kann und zum automatischen Einschalten des ft-HAT und damit des Raspberry-Pi genutzt werden kann.

3.2.1 EEPROM-Speicher AT24C32

Das EEPROM des ft-HAT ist vom Typo AT24C32 und ist bei Auslieferung bereits fertig programmiert und als Anwender hat man in der Regel keinen Anlass, diese Programmierung zu ändern.

Bei Bedarf kann das EEPROM mit den dafür vorgesehenen Programmen¹ ausgelesen und umprogrammiert werden. Details dazu findet man im Internet².

Beim Booten erkennt der Raspberry-Pi die Programmierung des EEPROMs und stellt seinen Inhalt unter `/proc/device-tree/hat/` zur Verfügung. Der Inhalt der dort liegenden Informationen ist die bevorzugte Art, das Vorhandensein des ft-HAT zu erkennen.

3.2.2 RTC-Uhrenbaustein DS3231

Hauptaufgabe eines Echtzeituhrenbaustein (engl. Real-Time-Clock, RTC) wie dem DS3231 des ft-HAT ist es, die Uhrzeit zu speichern und fortzuschreiben, auch wenn das Hauptsystem ausgeschaltet ist. Anders als viele PCs verfügt der Raspberry-Pi selbst nicht über solch einen Baustein. Da der Raspberry-Pi aber in der Regel mit dem Internet verbunden ist kann er von dort nach dem Einschalten die aktuelle Uhrzeit beziehen.

Der im ft-HAT verbaute RTC verfügt zusätzlich über eine Alarmfunktion, durch die die Spannungsversorgung des ft-HAT eingeschaltet werden kann. Der RTC des ft-HAT ist immer mit Spannung versorgt, wenn der ft-HAT mit einer Spannungsquelle verbunden ist. Wenn der ft-HAT ausgeschaltet ist wird der RTC weiterversorgt und kann die Uhrzeit weiterführen und den Raspberry-Pi über die Spannungsversorgung des ft-HAT einschalten.

Für die Uhrzeitfunktionen des verwendeten DS3231 hat der Linux-Kernel des Raspberry-Pi passende Treiber. Leider unterstützen diese die Alarmfunktion nicht, so dass der RTC zur Benutzung der Alarmfunktion direkt programmiert werden muss. Eine Bibliothek, um den RTC unter Python zu programmieren und z.B. ein Zeitgesteuertes Starten des Systems zu ermöglichen findet sich auf Github³.

Wird der ft-HAT durch den RTC eingeschaltet, so muss der Alarm des RTC quittiert werden, bevor die Spannungsversorgung des ft-HAT während des Herunterfahrens des Raspberry-Pi wieder automatisch abgeschaltet werden kann. Wird der Alarm nicht quittiert, so hält der noch anstehende Alarm die Spannungsversorgung permanent eingeschaltet. Das folgende Kommando quittiert den Alarm:

```
i2cset -y 0 0x68 0x0f 0x00
```

Es kann z.B. in der Datei `/etc/rc.local` eingefügt werden und wird dort bei jedem Systemstart automatisch ausgeführt:

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
```

¹HAT-Tools für den Raspberry-Pi: <https://github.com/raspberrypi/hats>

²HAT EEPROM anschließen und konfigurieren: <https://blog.gc2.at/post/hat-eprom/>

³DS3231-Bibliothek für Python: <https://github.com/harbaum/cfw-apps/blob/master/packages/tx-pi-hat-test/DS3231.py>

```

# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# Print the IP address
_IP=$(hostname -I) || true

if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi

# ack pending RTC wakeup
/usr/sbin/i2cset -y 0 0x68 0x0f 0x00

exit 0

```

3.3 Nutzung von externen I²C-Komponenten am ft-HAT

Sowohl der fischertechnik-TX-Controller, als auch sein Nachfolger der TXT-Controller verfügen über I²C-Erweiterungsanschlüsse.

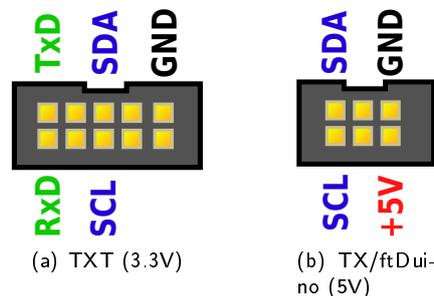


Abbildung 3.3: Die I²C-Anschlüsse des ft-HAT

Während es für den TX von fischertechnik selbst keine entsprechenden Sensoren gab werden für den TXT die sogenannten Umwelt- und Kombisensoren. Von Dritten gab und gibt es auch für den TX-Controller sowie für den zu dessen I²C-Anschluss kompatiblen ftDuino-Controller zahlreiche I²C-Sensoren.

Der Anschluss eines TXT-Sensors sowie aller anderen auf 3,3V-Versorgung ausgelegten Sensoren erfolgt über den 10-poligen 3,3V-Anschluss des ft-HAT. Sensoren aus, die für den TX-Controller, den ftDuino oder andere 5V-Geräte entwickelt wurden gehören an den 6-poligen-5V-Anschluss des ft-HAT.

Zur Ansteuerung dieser Sensoren reicht es in der Regel, den darin enthaltenen Sensorchip zu kennen. Treiber für das Linux-Betriebssystem des Raspberry-Pi lassen sich dann in der Regel leicht finden.

3.3.1 fischertechnik-Umweltsensor 167358

Der Umweltsensor 167358 basiert auf dem Bosch BME680-Sensor. Für diesen Chip gibt es u.a. eine passende Python-Bibliotheken, die für den Raspberry-Pi verfügbar sind.

Die zur Versorgung der fischertechnik-Sensoren nötigen 9-Volt können direkt dem Anschluss am ft-HAT entnommen werden.

Die Installation beschränkt sich daher auf das übliche:



Abbildung 3.4: fischertechnik-Umweltsensor am ft-HAT

```
sudo apt-get install python3-bme680
```

Beispiele zur Programmierung finden sich u.a. auf Github⁴.

3.3.2 fischertechnik-Kombisensor 158402

Der fischertechnik-Kombisensor 158402 basiert auf dem Bosch Sensortec BMX055. Er beinhaltet einen Gyrosensor, einen Beschleunigungssensor und einen Kompasssensor.

Python-Beispiele zur Ansteuerung dieses Sensors finden sich auf Github⁵.

3.3.3 TX- und ftDuino-Sensoren

Von Dritthistellern sind auch Sensoren mit passendem Anschluss für den TX-Controller bzw. den ftDuino erhältlich. Diese lassen sich direkt an 6-poligen Anschluss anschließen.



Abbildung 3.5: ftDuino-Mini-Servo-Adapter und Original-fischertechnik-Servo am ft-HAT

Passende Treiber finden sich in der Regel bei den dazugehörigen Projekten. Dass der Raspberry-Pi in der Maker-Szene weit verbreitet ist trägt dazu bei, dass es für praktisch jeden I²C-Sensor auch passende Software für diese Plattform gibt.

⁴ BME680-Python: <https://github.com/pimoroni/bme680-python>

⁵ BMX055-Python: <https://github.com/ControlEverythingCommunity/BMX055/blob/master/Python/BMX055.py>

Kapitel 4

Anschluss an fischertechnik-Modelle

Die Hauptfunktion des ft-HAT ist die Möglichkeit, fischertechnik-Modelle direkt an den Raspberry-Pi anzuschließen. Da die GPIO-Anschlüsse des Raspberry-Pi aber mit maximal 3,3 Volt betrieben werden dürfen sind sie nicht direkt für den Betrieb der fischertechnik-üblichen 9-Volt geeignet.

Der ft-HAT enthält der spezielle Treiber- und Schutzschaltungen, um einige der GPIO-Pins fischertechnik-kompatibel zu machen. Insgesamt verfügt der ft-HAT über vier digitale fischertechnik-kompatible Eingänge und zwei fischertechnik-kompatible Motorausgänge.

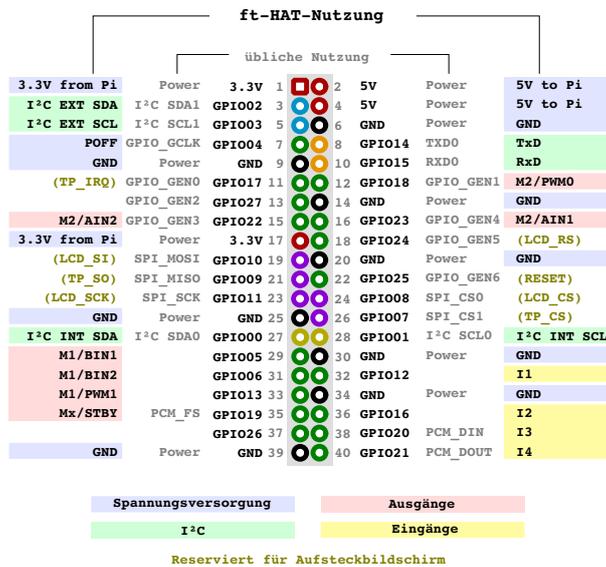


Abbildung 4.1: Belegung der GPIO-Leiste des Raspberry-Pi

Die fischertechnik-kompatiblen Anschlüsse sind auf einem 16-poligen Stecker seitlich am ft-HAT herausgeführt. Zur leichteren Verwendung in fischertechnik-Modellen liegt dem ft-HAT eine Erweiterungsplatine bei, die mit dem ebenfalls mitgelieferten Kabel am ft-HAT angeschlossen werden kann. Die Erweiterungsplatine bietet dann die fischertechnik-üblichen Anschlüsse.

Die folgenden Beispiele nutzen die RPi.GPIO-Bibliothek¹ unter der Programmiersprache Python. Die Nutzung anderer auf dem Raspberry-Pi verfügbarer Programmiersprachen und Bibliotheken erfolgt auf vergleichbare Weise und entspricht den diversen Tutorials im Internet zur Abfrage und Ansteuerung der GPIO-Anschlüsse des Raspberry-Pi.

¹RPi.GPIO: <https://pypi.org/project/RPi.GPIO/>

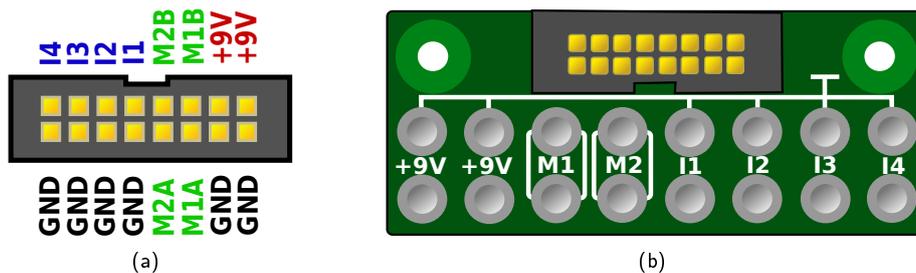


Abbildung 4.2: fischertechnik-Anschlüsse am ft-HAT

4.1 Eingänge für Schalter, Taster u.ä.

Die vier digitalen Eingänge I1 bis I4 des ft-HAT sind auf die GPIO-Pins GPIO12, GPIO16, GPIO20 und GPIO21 abgebildet. Im Ruhezustand liefert diese den Wert 1 (high). Wird ein Eingang durch einen Taster, Schalter oder Fototransistor betätigt, so wechselt sein Zustand auf 0 (low).

In Python lässt sich z.B. der Eingang I1 an GPIO12 wie folgt abfragen.

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(12, GPIO.IN)
print("I1:", GPIO.input(12))
```

Die Abfrage in anderen Programmiersprachen erfolgt auf vergleichbare Weise und kann den üblichen Programmerhinweisen des Raspberry-Pi entnommen werden.

4.2 Ausgänge für Motoren, Lampen u.ä.

Die Ausgänge werden über einen sogenannte Leistungstreiber auf dem ft-HAT realisiert. Der verwendete Treiber vom Typ TB6612 kann zwei Motoren ansteuern. Die Anschlüsse des TB6612 sind wie folgt auf die GPIO-Pins des Raspberry-Pi abgebildet:

	M1	M2
IN1	BIN1/GPIO05	AIN1/GPIO23
IN2	BIN2/GPIO06	AIN2/GPIO22
PWM	PWM1/GPIO13	PWM0/GPIO18
STBY	GPIO19	

Um einen Ausgang anzusteuern muss zunächst der Treiberchip über den Anschluss STBY via GPIO19 aktiviert werden. Die Signale AIN1 und AIN2 sowie PWM0 für den Motorausgang M1 bzw. BIN1 und BIN2 sowie PWM1 legen den Betriebsmodus des Ausgangs fest:

STBY	IN1	IN2	PWM	state
0	x	x	x	Treiber aus (hohe Impedanz)
1	0	0	1	Motor Stopp (hohe impedanz)
1	1	1	x	Kurzschlussbremse (beide Motoranschlüsse auf Masse)
1	0	1	1	Drehen gegen Uhrzeigersinn (ein Ausgang an 9V, einer auf Masse)
1	0	1	0	Kurzschlussbremse (beide Motoranschlüsse auf Masse)
1	1	0	1	Drehen im Uhrzeigersinn (ein Ausgang an 9V, einer auf Masse)
1	1	0	0	Kurzschlussbremse (beide Motoranschlüsse auf Masse)

Folgendes Python-Beispiel schaltet den Ausgangstreiber ein:

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(19, GPIO.OUT)
GPIO.output(19, GPIO.HIGH)
```

Die vier zum Betrieb des Motors M1 nötigen Signale für GPIO5,GPIO6,GPIO13 und GPIO19 lassen sich in Python z.B. folgendermaßen setzen:

```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)

pins = { 19:GPIO.HIGH, 5:GPIO.HIGH, 6:GPIO.LOW, 13:GPIO.HIGH }
for p in list(pins.keys()):
    GPIO.setup(p, GPIO.OUT)
    GPIO.output(p, pins[p])
```

Eine Geschwindigkeitsregelung des angeschlossenen Motors kann über die PWM-Signale erfolgen. Beispiele dafür finden sich z.B. auf Github².

Die Ansteuerung aus anderen Programmiersprachen erfolgt auf vergleichbare Weise und kann den üblichen Programmerhinweisen des Raspberry-Pi entnommen werden.

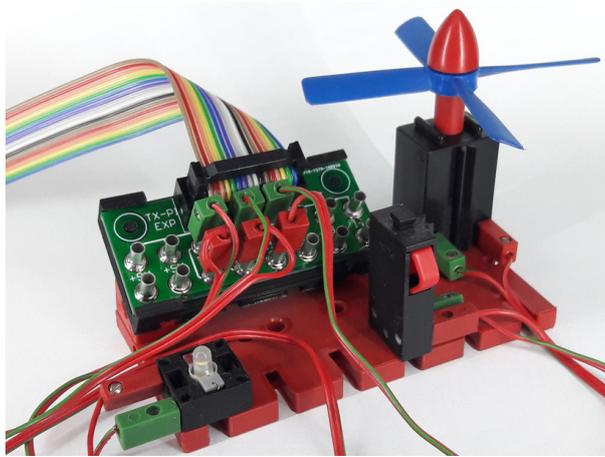


Abbildung 4.3: Benutzung des Breakout-Boards

²ft-HAT-Demo: <https://github.com/harbaum/cfw-apps/blob/master/packages/tx-pi-hat-test/tx-pi-hat-test.py>

Kapitel 5

Verwendung mit einem Aufsteckbildschirm

Der ft-HAT lässt einige Signale des GPIO-Ports unberührt, die häufig von kleinen Aufsteckbildschirmen für den Raspberry-Pi verwendet werden. Speziell die 3,5 Zoll großen Touchbildschirme der Firma Waveshare und dazu kompatible Bildschirme lassen sich auf den ft-HAT aufstecken und gleichzeitig nutzen.



Abbildung 5.1: Raspberry-Pi mit ft-HAT und aufgestecktem Bildschirm

Speziell bei Einsatz der Software und Gehäuse aus dem TX-Pi-Projekt¹ erhält man mit dem ft-HAT ein komplett auf den Einsatz in fischertechnik-Modellen abgestimmtes Setup.

¹TX-Pi: <http://tx-pi.de>