

Computing

ROBO Pro, ftrobopy und ftScratch auf dem TXT 4.0

Torsten Stuehn

Das Programm *ftrobopy_server* emuliert auf dem TXT 4.0 das ROBO Pro-Kommunikationsprotokoll des TXT. Damit kann der TXT 4.0 von allen Anwendungen, die für den Online-Modus des TXT entwickelt wurden angesteuert werden. Mit Python und *ftrobopy* können auch Offline-Programme, unabhängig von PC und Netzwerk, auf dem TXT 4.0 gestartet werden.

Hintergrund

Mit ROBO Pro Coding hat fischertechnik sich von seiner seit Jahrzehnten favorisierten graphischen Programmiersprache ROBO Pro (und deren Vorläufern wie Lucky Logic) verabschiedet und baut nun mit ROBO Pro Coding, ähnlich der Programmierplattform eines bekannten dänischen Spielzeugherstellers, auf dem Blockly-System von Google auf [1, b, c].

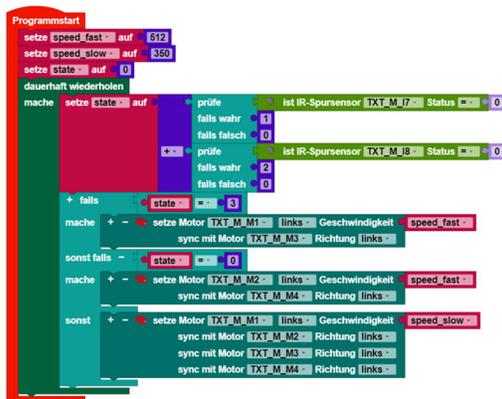


Abb. 1: Spurfolger in ROBO Pro Coding (Programm-Beispiel von fischertechnik)

Die neue Blockly basierte Programmiersprache ROBO Pro Coding für den TXT 4.0 wird „unter der Haube“ in Form von Python-Programmen auf dem TXT 4.0 ausgeführt. Deshalb liegt es nahe, dem Anwender diese Python-Programme bereits in der

ROBO Pro Coding GUI als editierbare, textbasierte Alternative zur Verfügung zu stellen. Einsteigern soll damit der Schritt zur textuellen Programmierung in Python erleichtert werden.

```

1 from fischertechnik.controller.Motor import Motor
2 from lib.controller import *
3
4
5 speed_fast = None
6 speed_slow = None
7 state = None
8
9
10 speed_fast = 512
11 speed_slow = 350
12 state = 0
13 while True:
14     state = (1 if (TXT_M_I7_trail_follower.get_state() == 0) else 0) + (2 if (TX
15     if state == 3:
16         TXT_M_M1_encodermotor.set_speed(int(speed_fast), Motor.CCW)
17         TXT_M_M3_encodermotor.set_speed(int(speed_fast), Motor.CCW)
18         TXT_M_M1_encodermotor.start_sync(TXT_M_M3_encodermotor)
19     elif state == 0:
20         TXT_M_M2_encodermotor.set_speed(int(speed_fast), Motor.CCW)
21         TXT_M_M4_encodermotor.set_speed(int(speed_fast), Motor.CCW)
22         TXT_M_M2_encodermotor.start_sync(TXT_M_M4_encodermotor)
23     else:
24         TXT_M_M1_encodermotor.set_speed(int(speed_slow), Motor.CCW)
25         TXT_M_M2_encodermotor.set_speed(int(speed_slow), Motor.CCW)
26         TXT_M_M3_encodermotor.set_speed(int(speed_slow), Motor.CCW)
27         TXT_M_M4_encodermotor.set_speed(int(speed_slow), Motor.CCW)
28         TXT_M_M1_encodermotor.start_sync(TXT_M_M2_encodermotor, TX
29

```

Abb. 2: Spurfolger in Python (automatisch von ROBO Pro Coding generiert)

Aufgrund der automatischen Generierung durch Blockly unterliegen diese Python-Programme jedoch einigen Einschränkungen (z. B. sind nur globale Variablen

erlaubt) und machen dadurch den Code teilweise unübersichtlich und nur schwer lesbar. Als Einführung in eine textbasierte Programmiersprache sind die Blockly-generierten Python-Programme von ROBO Pro Coding dadurch für Programmieranfänger eher weniger geeignet. Erfahrene Programmierer können aber durchaus davon profitieren und bestehende Limitierungen von Blockly (bzw. ROBO Pro Coding) durch eigene Erweiterungen in Python umgehen.

Vorher hat fischertechnik auf dem älteren TXT parallel zum originalen ROBO Pro auch die Programmiersprachen C/C++ [4] und, aufgrund von Initiativen aus der unabhängigen fischertechnik Community heraus, auch Python [5, f] unterstützt.

Auch die speziell für Programmierneinsteiger vom MIT entwickelte graphische Programmierumgebung Scratch [7] wurde von fischertechnik zunächst über ftScratchX [8], später als ein „Fork“ von Scratch3 (ftScratch3) für den TXT angeboten [9] und hat sich in vielen Schulprojekten als Einführung in die Programmierung etabliert, insbesondere für jüngere Schülerinnen und Schüler. In der ft:pedia 1/2022 hat Dirk Fox einen sehr schönen Artikel zum Einsatz von Scratch mit fischertechnik geschrieben [10].

Offiziell wird auf dem neuen TXT 4.0 bisher keine dieser Programmiermöglichkeiten mehr angeboten. Auch der automatisch vom neuen ROBO Pro Coding erzeugte Python-Code ist nicht kompatibel mit dem ftrobopy-basierten Python-Code des älteren TXT – obwohl der C++/Python-Unterbau in der Firmware des TXT 4.0 große Ähnlichkeiten zur ftrobopy-Struktur aufweist.

Leider gibt es damit keine einheitliche plattformübergreifende Programmiersprache zwischen dem alten TXT und dem neuen TXT 4.0 mehr. Der Parallelbetrieb

beider Systeme erfordert deshalb die Wartung einer doppelten Codebasis und stellt insbesondere im Schulbetrieb eine Einschränkung dar.

Der ftrobopy-Server

Aus der Motivation heraus, ganz unabhängig vom Controller (TXT 4.0 oder TXT), einheitliche Python-Programme ausführen zu können, wurde die Idee für den ftrobopy_server geboren. Dieser emuliert das ROBO Pro-Kommunikationsprotokoll des TXT über die Ports 65000 (I/Os), 65001 (Camera) und 65002 (I2C). Der TXT 4.0 kann damit von allen Anwendungen angesteuert werden, die für den Online-Modus des TXT entwickelt wurden. Dazu gehören derzeit ftrobopy, ROBO Pro und ftScratch.

Der ftrobopy_server ist in seiner aktuellen Version kompatibel zu ROBO Pro 4.7.0. Die ROBO Pro Versionsnummer ist jedoch nur ein Parameter, der einfach auch für andere ROBO Pro Versionen (z. B. 4.2.4 oder 4.6.6) angepasst werden kann, falls dies notwendig sein sollte (z. B. wenn im Schulbetrieb noch ältere ROBO Pro-Versionen verwendet werden müssen).

Möglicherweise kann der ftrobopy_server auch eingesetzt werden um die C/C++-Programme der älteren fischertechnik Lernfabrik (ohne den Zusatz „Industrie 4.0“) mit dem TXT 4.0 im Online-Modus einzusetzen. Dies ist jedoch nur eine Vermutung und bisher nicht getestet.

Die vom TXT bekannte Android- bzw. iOS-Smartphone-App TXTCamControl zur Fernsteuerung von Robotern auf TXT-Basis funktioniert derzeit noch nicht mit dem ftrobopy_server und dem TXT 4.0. Eine zukünftige Unterstützung dieser Smartphone-App ist jedoch geplant.

Die Neuerungen des TXT 4.0, wie z. B. die drei Servo-Ausgänge und die höhere Rechenleistung durch zwei ARM-Kerne, bleiben mit ftrobopy_server erhalten und können z. B. auch unter ROBO Pro genutzt

werden. Zusätzlich soll die Möglichkeit der Kontrolle des TXT 4.0 mit einer fischer-technik-Fernbedienung ermöglicht werden, sowohl mit `ftrobopy` als auch mit ROBO Pro. Da der TXT 4.0 im Unterschied zum TXT keinen eingebauten IR-Empfänger hat, wird vorerst nur die BT Remote unterstützt.

Die `ftrobopy`-Python-Programme können über den `ftrobopy_server` sowohl im Online- als auch im Offline-Modus (über die localhost-Adresse 127.0.0.1) auf dem TXT 4.0 ausgeführt werden.

ROBO Pro-Programme können dagegen nur im Online-Modus ausgeführt werden, da das ROBO Pro-Offline-Binärformat gänzlich undokumentiert ist und dessen Reverse-Engineering derzeit aufgrund des zu hohen Zeitaufwandes unrealistisch ist. ROBO Pro-Programme benötigen also eine dauerhafte Netzwerkverbindung über USB-Kabel, WLAN oder Bluetooth zwischen TXT 4.0 und dem Windows PC, auf dem die ROBO Pro-Anwendung läuft.

Der `ftrobopy_server` ist eine in C/C++ entwickelte Anwendung und wird als (compiliertes) Binary auf dem TXT 4.0 direkt ausgeführt. Das Programm kann dabei bequem per Touch über das TXT 4.0 GUI gestartet und gestoppt werden. Außer dem Kopieren des Programmes auf den TXT 4.0 sind keine Modifikationen notwendig. Das Programm kann auch auf eine SD-Karte kopiert und von dort per GUI gestartet werden.

Aufgrund des compilierten C/C++-Codes läuft das Programm deutlich schneller ab als die mit ROBO Pro Coding erzeugten Python-Programme. Auch die Online-Kommunikation zu ROBO Pro kann wegen der schnelleren ARM-CPU's des TXT 4.0 schneller durchgeführt werden als mit dem (vom alten TXT bekannten) 100-ms-Zeitintervall. Die genauen Maximalwerte dafür wurden bisher noch nicht ausgelotet.

Die im Rahmen der Entwicklung des `ftrobopy_server` aufgedeckten Möglichkeiten zur direkten C/C++-Programmierung des TXT 4.0 im Offline-Modus eröffnen ein interessantes Feld für die Entwicklung von Steuerungsaufgaben mit kurzen Reaktionszeiten für den TXT 4.0. Denkbar sind damit z. B. balancierende Roboter à la Segway oder ähnliche Modelle. Ein Beitrag zur C/C++-Programmierung des TXT 4.0 ist für eine der nächsten ft:pedia-Ausgaben geplant.

Der `ftrobopy_server` befindet sich derzeit noch im Beta-Stadium und wird als Open-Source unter der MIT-Lizenz auf der Github-Plattform entwickelt. Der C++-Quellcode ist bisher noch recht unstrukturiert. In zukünftigen Überarbeitungen wird sich dieser Zustand sicher noch ändern. Seit der Version 0.9.8 ist der Server jedoch recht stabil und kann deshalb bereits sinnvoll in eigenen ROBOPro, `ftrobopy` oder `ftScratch`-Projekten eingesetzt werden.

Es kann sich lohnen hin-und-wieder auf der Github-Plattform nach neuen Versionen Ausschau zu halten.

Download und Installation des `ftrobopy_servers`

Das aktuelle `ftrobopy_server`-Paket kann von Github als zip-Datei heruntergeladen oder über git geklont werden. Die Download-URL des zip-Paketes lautet:

https://github.com/ftrobopy/ftrobopy_server/archive/refs/heads/main.zip

Das Paket sollte anschließend in einen beliebigen Ordner entpackt werden. Zur Installation des `ftrobopy_servers` auf dem TXT 4.0 werden nur die beiden folgenden Dateien benötigt:

```
ftrobopy_server # das Executable
ftrobopy_server.cpp # der
Quellcode
```

Die Datei `ftrobopy_server.cpp` befindet sich im Unterordner `src`. Diese beiden Dateien müssen in einen eigenen Ordner auf den

TXT 4.0 kopiert werden. Dies kann entweder mit ssh/scp-Befehlen erfolgen oder, wie im Folgenden beschrieben, über den Datei-Upload des auf dem TXT 4.0 laufenden Webservers.

Es wird hier davon ausgegangen, dass der TXT 4.0 per USB-Kabel mit dem PC verbunden ist und damit über die IP-Adresse 192.168.7.2 erreichbar ist. Sollte der TXT 4.0 als Client im WLAN angemeldet sein, muss die IP-Adresse entsprechend angepasst werden.

Die URL für die Anmeldung am TXT 4.0 Webserver lautet: <http://192.168.7.2>

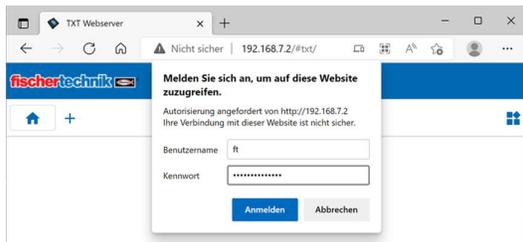


Abb. 3: Anmeldefenster des TXT 4.0-Webservers

Als Benutzername (Abb. 3) muss der User ft verwendet werden, das Kennwort lautet: fischertechnik.

Zuerst sollte nun über das „+“ (oben links im Fenster) im angezeigten Web-Homeverzeichnis ein neuer Ordner, z. B. mit dem Namen ftrobopy_server, angelegt werden (Abb. 4).

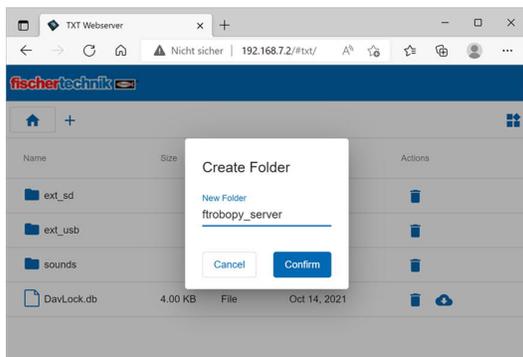


Abb. 4: Anlegen eines Ordners

In diesen Ordner sollten dann die beiden Dateien ftrobopy_server und

ftrobopy_server.cpp hochgeladen werden, wieder durch klicken des „+“ und Add Files (Abb. 5).

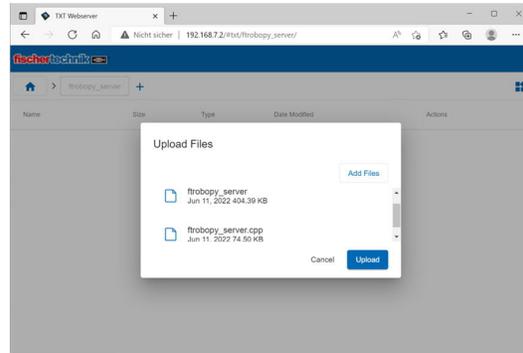


Abb. 5: Hochladen auf den TXT 4.0

Um nach dem Upload das Upload-Fenster wieder zu verlassen, muss man die Esc-Taste drücken. (Das Klicken des Finish-Buttons hilft nicht weiter.)

Abschließend muss noch das Execute-Flag der beiden Dateien gesetzt werden, damit man den ftrobopy_server aus der TouchGUI des TXT 4.0 heraus starten kann. Dafür loggt man sich per ssh, z. B. aus der Windows-Kommandozeile heraus (Windows-Taste+R, dann „cmd“ eingeben), auf dem TXT 4.0 ein:

```
ssh ft@192.168.7.2 (Passwort:
fischertechnik)
```

Das Setzen des Execute-Flags erfolgt dann auf dem TXT 4.0 mit (siehe Abb. 6):

```
cd workspaces
cd ftrobopy_server
chmod a+x *
```

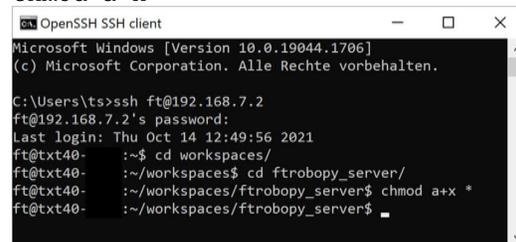


Abb. 6: Ändern des Execute-Flags

Die ftrobopy_server-Anwendung kann anschließend über die TXT 4.0 GUI nach Bedarf oder automatisch beim Hochfahren des TXT 4.0 gestartet werden und steht für

den Online-Betrieb mit ROBO Pro, ftrobopy (auch Offline) oder ftScratch zur Verfügung (Abb. 7).

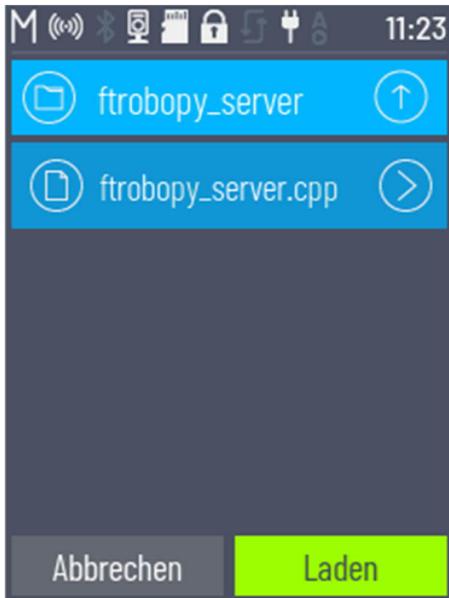


Abb. 7: Laden des ftrobopy_server

Anmerkung:

In der GUI des TXT 4.0 werden ausführbare binäre Dateien nicht angezeigt. Nur Dateien mit Dateierendungen (.py, .cpp, ...) können per Touch gestartet werden. Bei einer .cpp-Datei (C++-Quellcode) wird dann automatisch die ausführbare Binärdatei gleichen Namens (ohne .cpp-Endung) gestartet. Der tatsächliche Inhalt der .cpp-Datei ist nicht relevant. Sie kann auch komplett leer sein.

Nach dem Starten des ftrobopy_servers über die Touch-GUI erscheint auf dem Bildschirm des TXT 4.0 die Meldung „ftrobopy_server.cpp Program initializing“ (Abb. 8). Davon sollte man sich nicht täuschen lassen: tatsächlich läuft das ftrobopy_server-Binary bereits.

Ansteuerung der drei Servos des TXT 4.0

Da ROBO Pro die Servo-Ausgänge S1-S3 nicht direkt unterstützt, wird deren Ansteuerung über den Umweg einer Extension ermöglicht: Die drei Ausgänge O1-O3 der ersten Extension (in ROBO Pro: EM1, Abb.

9) werden von ftrobopy_server auf die drei Servo-Ausgänge des TXT 4.0 gemappt. Der Wert 4 (oder 256) steht für die Mittelstellung eines Servos. Die Werte 1 und 8 (oder 512) stehen jeweils für den Vollausschlag eines Servos nach Links oder Rechts. Zwischenwerte sind natürlich auch erlaubt.



Abb. 8: ftrobopy_server läuft bereits

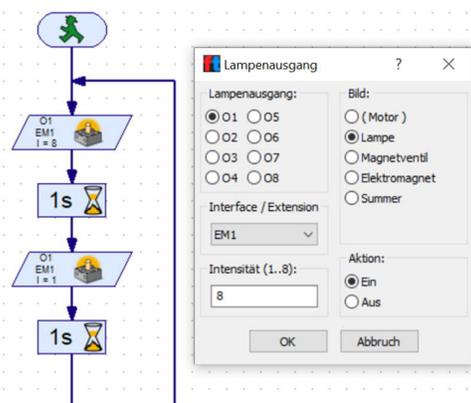


Abb. 9: Ansteuerung des Servos S1

In Python (mit ftrobopy) können die drei Servos äquivalent über die Extension Nr. 1 angesteuert werden. Bei der Initialisierung von ftrobopy müssen dafür die Extensions mit dem Parameter `use_extensions=True` erlaubt werden:

```
>>> import ftrobopy
>>> txt = ftrobopy.ftrobopy('192.
168.7.2', use_extension=True)
>>> servol = txt.output(1, ext=1)
>>> servol.setLevel(256) #
Mittelstellung
```

Unter ftScratch ist dieser Trick leider nicht möglich, da ftScratch keine Extensions unterstützt. Deshalb können die Servos des TXT 4.0 dort vorerst noch nicht angesteuert werden. Als Idee zur Ansteuerung der Servos in ftScratch könnte evtl. die Sound-Funktion erhalten, falls diese nicht unbedingt benötigt wird.

Synchronisation von vier Motoren

Auch die Aktivierung der Synchronisation aller vier Motoren, die nicht direkt von ROBO Pro unterstützt wird, erfolgt über den Extension-Umweg: Wird der Ausgang O4 der Extension EM1 auf 8 (bzw. 512) gesetzt, schaltet ftrobopy_server die Synchronisierung aller vier Motoren auf dem TXT 4.0 ein (Abb. 10)..

Unter Python mit ftrobopy funktioniert der Trick wieder genauso:

```
>>> import ftrobopy
>>> txt = ftrobopy.ftrobopy('192.
168.7.2', use_extension=True)
>>> motor1 = txt.motor(1)
>>> motor2 = txt.motor(2)
>>> motor3 = txt.motor(3)
>>> motor4 = txt.motor(4)
>>> motorsync = txt.output(4,
ext=1)
>>> motorsync.setLevel(512)
```

Unter ftScratch gibt es bisher keine Möglichkeit zur Synchronisation aller vier Motoren. Allerdings könnte auch hier die Sound-Funktion evtl. weiterhelfen.

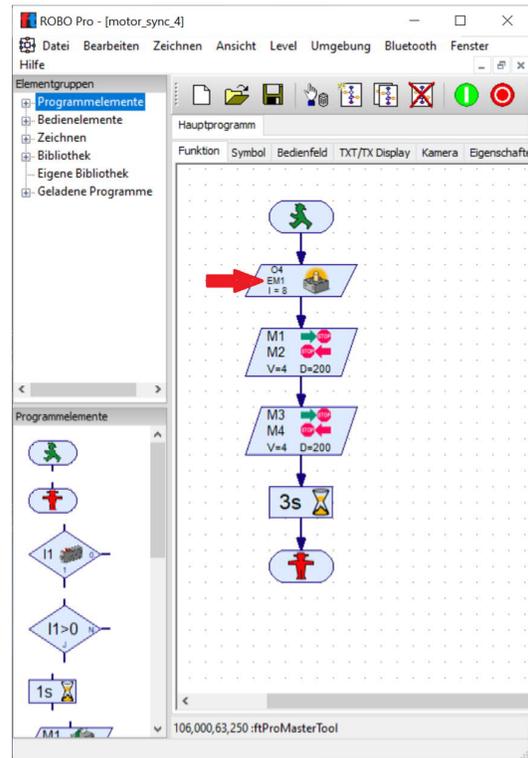


Abb. 10: Synchronisation aller vier Motoren

Aktueller Stand der Entwicklung

Wie bereits erwähnt befindet sich ftrobopy_server derzeit noch in der Entwicklung. Nicht alle Funktionen sind absolut stabil und es gibt (selten) auch noch Hänger. Im Zusammenspiel mit ROBO Pro kann es passieren, dass die Verbindung von ROBO Pro wegen CRC-Fehlern unterbrochen wird; ftrobopy reagiert bei Checksummen-Fehlern etwas unempfindlicher.

Die aktuelle Version 0.9.8 des ftrobopy_server ist aber definitiv schon für einfache Projekte brauchbar. Auch Langzeittests liefen bisher erfolgreich; Memory-Leaks sollten keine mehr vorhanden sein.

Im Einzelnen werden die folgenden TXT 4.0 Funktionen in der Version 0.9.8 von ftrobopy_server bereits unterstützt:

- vier Motoren (inklusive deren Synchronisierung) oder acht einfache Ausgänge

- acht Eingänge (Taster und Ultraschallsensor funktionieren, analoge spannungsbasierte Sensoren wie z. B. der Farbsensor funktionieren noch nicht zuverlässig)
- fischertechnik Kamera (bisher nur über USB-Netzwerk stabil)
- drei Servo-Ausgänge

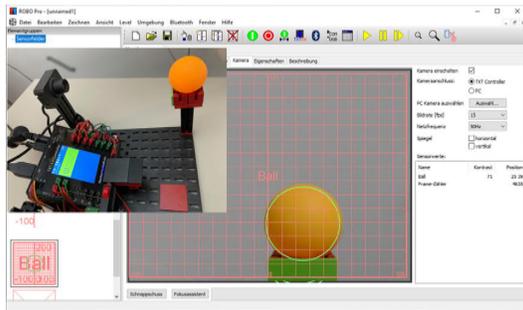


Abb. 11: Der ROBO Pro-Ballerkenner-Sensor mit dem TXT 4.0 (im Inset ist der Aufbau des Testmodells mit dem TXT 4.0 zu sehen)

Noch ausstehend sind:

- I2C-Support (damit werden dann alle ROBO Pro-Treiber für I2C-Geräte auch für den TXT 4.0 benutzbar)
- BT-Remote Anbindung (ist bereits weitgehend implementiert, allerdings ist dafür eine geringfügige Modifikation auf dem TXT 4.0 mit root-Berechtigung notwendig)
- Soundunterstützung
- Unterstützung von bis zu 8 Extensions (auch unter ROBO Pro). Wenn alle 8 Extensions verwendet werden, bleiben keine freien Extension-Ausgänge für die Servos mehr übrig.
- Unterstützung der TXTCamControl-App

Die BT Smart und Early Coding Controller von fischertechnik und alte TXTs sollen in einer zukünftigen Version auch als Extensions (über Bluetooth bzw. WLAN mit dem

TXT 4.0 verbunden) am TXT 4.0 unterstützt und mit fTrobopy und ROBO Pro angesteuert werden können. ftScratch wird aufgrund der fehlenden Extension-Unterstützung damit wahrscheinlich nicht funktionieren.

Quellen

- [1] Till Harbaum: *Von Lucky Logic zu RoboPro Coding*. [ft:pedia 1/2021](#), S. 103–109.
- [2] Till Harbaum: *Brickly auf dem TXT: Grafische Programmierung à la Google-Blockly*. [ft:pedia 1/2017](#), S. 92–98.
- [3] Google Developers: [Blockly – A JavaScript library for building visual programming editors](#).
- [4] fischertechnik: [TXT-C-Programming-Kit](#). Github.com (leider ist die ursprüngliche Version 4.1.6 nicht mehr verfügbar)
- [5] Torsten Stuehn: *Programmierung des TXT mit Python*. [ft:pedia 2/2017](#), S. 58–62.
- [6] Torsten Stuehn: [fTrobopy – a python module for the fischertechnik TXT controller](#). Github.com
- [7] <https://scratch.mit.edu>
- [8] Dirk Fox: *Scratch mit fischertechnik*. [ft:pedia 1/2018](#), S. 69–78.
- [9] fischertechnik: [FTScratchTXT](#). Github.io
- [10] Dirk Fox: *Scratch mit fischertechnik – Update 2022*. [ft:pedia 1/2022](#), S. 86–92.
- [11] Torsten Stuehn: [fTrobopy server - fTrobopy / ROBOPRO / ftScratchTXT server for the fischertechnik TXT 4.0 controller](#). Github.com