

Computing

Programmierung des TXT mit Python

Torsten Stuehn

Dass man Programme für den TXT – nicht zuletzt dank des enthaltenen Linux-Betriebssystems – prinzipiell mit praktisch jeder Programmiersprache entwickeln kann, ist offensichtlich. Aber wie startet man ganz konkret? Der Beitrag zeigt, wie leicht das mit Python und dem vom Autor für die Community entwickelten TXT-Treiber `ftrobopy` gelingt.

Die Programmiersprache Python hat sich mittlerweile in vielen fischertechnik-Projekten mit dem TXT als Quasi-Standard neben der grafischen Programmierung mit ROBO Pro etabliert. Dies gilt sowohl für die originale fischertechnik-Firmware als auch ganz besonders für die community-Firmware [1, 2], die sogar selbst zum Teil in Python programmiert ist.

Insbesondere den Benutzern von Linux- und MacOSX-Betriebssystemen, die ROBO Pro nur mit Hilfe eines Windows-Emulators (z. B. Wine [3]) oder einer virtuellen Windows-Maschine verwenden können, bietet Python eine einfache und komfortable Möglichkeit, alle Fähigkeiten des TXT auszureizen. Durch seine weite Verbreitung in Forschung und Wissenschaft bietet Python unzählige Module, die dem ambitionierten fischertechniker kaum noch Grenzen bei der Umsetzung auch aufwendigster Projekte setzen.

Als ausgewachsene Programmiersprache bietet Python auch fortschrittliche Konzepte für den Zugriff auf Dateien und Datenbanken. Selbst ein einfacher Webserver auf dem TXT ist in nur einer Zeile umsetzbar, und Zugriffe auf die microSD-Karte des TXT sind (nach dem *mounten* der Karte) natürlich auch möglich.

Da die meisten Linux- und MacOSX-Benutzer ihren TXT wahrscheinlich längst in Python programmieren, wende ich mich im folgenden Text hauptsächlich an Windows-User, die einen einfachen und schnellen Einstieg in diese Programmiersprache suchen oder die sich für eine nicht-grafische Alternative zu ROBO Pro interessieren.

Ähnlich wie mit ROBO Pro kann man den TXT auch mit Python sowohl im *Online*- als auch im *Offline*-Modus programmieren:

- *Online*-Modus: Das Python-Programm läuft vollständig auf einem PC, der per WLAN, USB-Kabel oder Bluetooth mit dem TXT verbunden ist. Um den TXT im *Online*-Modus mit Python anzusteuern ist kein Eingriff auf dem TXT notwendig. Auf dem PC muss dafür ein Python-Interpreter installiert werden. Linux und MacOSX bringen diesen bereits mit.
- *Offline*-Modus: Das Python-Programm läuft vollständig auf dem TXT ab. Ein PC wird nur zum Editieren des Python-Programmes und zur Übertragung des Programmes auf den TXT und (normalerweise) zum Starten benötigt. Für diesen Modus muss ein Python-Interpreter auf den TXT kopiert werden. Die community-Firmware hat diesen bereits standardmäßig installiert.

Online-Modus

Python-Installation auf dem PC

Der einfachste Weg für den Einstieg in die Python-Programmierung des TXT führt über den *Online-Modus*. Darüber kann selbst ein neuer, frisch ausgepackter TXT ohne weitere Modifikationen sofort in Python angesteuert werden. Dafür benötigt man zunächst einen Python-Interpreter auf dem PC. Wer diesen noch nicht hat, der kann ihn sich z. B. von der Seite python.org [4] kostenfrei herunterladen. Ich empfehle die Installation der Python-Version 3.x (aktuell 3.6.1). Bei der Installation sollte man die Option „Add Python 3.6 to Path“ (siehe Abb. 1) setzen, damit man Python auch von der Kommandozeile aus aufrufen kann.

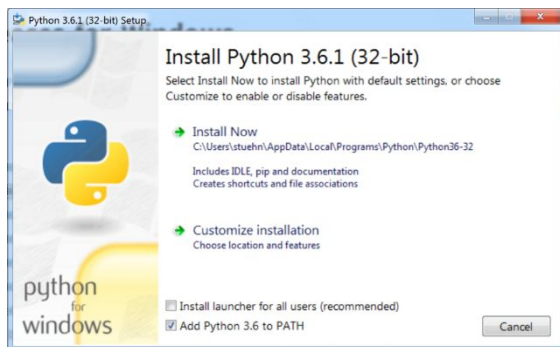


Abb. 1: Installationsdialog von Python

Nach einem Klick auf „Install Python 3.6.1“ steht ein einfacher Python-Interpreter auf der Windows-Kommandozeile zur Verfügung. Außerdem erreicht man über das Windows Start-Menü die Integrierte Python Entwicklungsumgebung (IDLE).

Um zu testen, ob Python richtig installiert wurde, kann man in der Windows-Kommandozeile („cmd.exe“) mit dem folgenden Befehl die Python-Version testen:

```
C:\Users\ts> python -version
```

Als Ausgabe sollte man dann das Folgende erhalten:

```
Python 3.6.1
```

Python ist damit erfolgreich auf dem PC installiert.

Installation von *ftrobopy* auf dem PC

Bevor der TXT in Python angesteuert werden kann, wird noch das Python-Modul *ftrobopy* benötigt. Dieses Modul kann mit dem folgenden Befehl über die Windows-Kommandozeile installiert werden:

```
C:\Users\ts> pip install ftrobopy  
(Unter Linux mit „sudo pip install ftrobopy“)
```

Falls es hierbei zu Problemen kommt, kann *ftrobopy* auch über github [5] heruntergeladen oder geklont werden. Die Datei *ftrobopy.py* kann dann von Hand in das Verzeichnis kopiert werden, in dem die eigenen Python-Programme für den TXT abgespeichert werden sollen. Dieser Weg sollte auch grundsätzlich unter MacOSX eingeschlagen werden (für die Mac-Experten: „pip install ftrobopy“ kann z. B. mit Anaconda-Python auch auf dem Mac verwendet werden).

Um in Windows den TXT über das USB-Kabel ansteuern zu können, wird der USB-Netzwerktreiber benötigt. Dieser wird automatisch bei der Erstinstallation von ROBO Pro installiert. Linux und MacOSX bringen den Treiber bereits mit. Wer also den TXT unter Windows über USB-Kabel ansteuern möchte, sollte, falls er dies noch nicht getan hat, [ROBO Pro](#) auf dem Rechner installieren.

Das Modul *ftrobopy* funktioniert grundsätzlich bereits ab der Firmware-Version 4.1.5. Es ist allerdings zu empfehlen, den TXT auf die neueste Version (aktuell 4.2.4) upzudaten.

Erste Schritte mit IDLE

Nun kann über das Windows Start-Menü die Python-Entwicklungsumgebung IDLE aufgerufen werden. Es handelt sich hier um eine interaktive Python-Shell, die jeden eingegebenen Python-Befehl unmittelbar ausführt (Abb. 2).

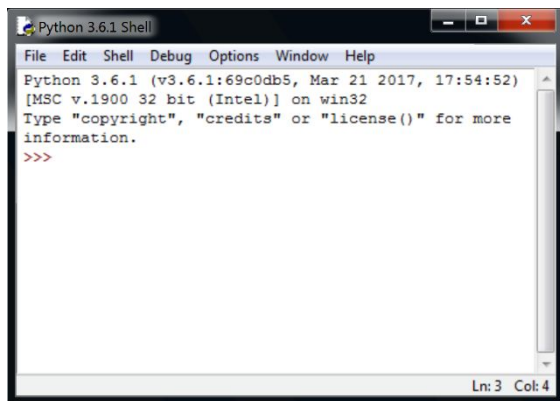


Abb. 2: Python-Entwicklungsumgebung IDLE

Die Zeichen „>>>“ in der Shell stellen das Command-Prompt von Python dar. Eine vereinfachte interaktive Python-Shell kann auch jederzeit in der Windows-Kommandozeile mit dem Befehl „python“ aufgerufen werden. In einem Linux- oder MacOSX-Terminal wird die interaktive Python-Shell auch einfach durch die Eingabe von „python“ gestartet.

Über das Menü Datei → Neue Datei (bzw. File → New File) kann auch ein Python-Editor aufgerufen werden, um komplexere Python-Programme zu schreiben und abzuspeichern. Diese werden nicht unmittelbar, sondern erst nach der Auswahl von „Run“ im Menü bzw. Drücken der Taste F5 ausgeführt.

Für den Einstieg und ein paar erste Tests, genügt die interaktive IDLE-Shell jedoch völlig. Zunächst muss dafür das *ftrobopy*-Modul importiert werden:

```
>>> import ftrobopy
```

Wenn dieses Modul nicht gefunden werden sollte, dann ist *ftrobopy* evtl. im falschen Verzeichnis installiert worden. Grundsätzlich muss die Datei *ftrobopy.py* entweder im aktuellen Verzeichnis oder im Python-System-Pfad vorhanden sein (z. B. unter „C:\Users\ts\AppData\Local\Programs\Python\Python36-32\Lib\site-packages“), damit das Modul importiert werden kann. Wenn keine Fehler aufgetreten sind, kann man mit

```
>>> txt=ftrobopy.ftrobopy('auto')
```

die Verbindung zum TXT herstellen. Der TXT muss dafür mit einer der Standard-Methoden (USB/WLAN/Bluetooth) angeschlossen sein. Falls der TXT als Client an einem WLAN-Accesspoint angeschlossen ist, muss anstelle von 'auto' die Client IP-Adresse (z. B. '192.168.2.114') angegeben werden. Ab diesem Zeitpunkt steht der TXT dann mit allen seinen Ein- und Ausgabe-Ports in Python zur Verfügung.

Wer mit der community-Firmware arbeitet, darf nicht vergessen, vorher die FT-GUI zu starten, um den „Online“-Modus zu ermöglichen.

Konfiguration und Ansteuerung der Ein- und Ausgänge des TXT

Zunächst können mit den folgenden Kommandos die Ein- und Ausgänge des TXT konfiguriert werden. Im Beispiel sind zwei Encoder-Motoren an den Ausgängen M1/C1 bzw. M2/C2 und ein Taster am Eingang I1 angeschlossen. Außerdem ist noch eine Lampe/LED an O5 angeschlossen:

```
>>> m1 = txt.motor(1)
>>> m2 = txt.motor(2)
>>> lampe = txt.output(5)
>>> taster = txt.input(1)
>>> lampe.setLevel(512)
>>> m2.setSpeed(0)
>>> m2.setSpeed(0)
>>> m1.setDistance(0, syncto=m2)
>>> m2.setDistance(0, syncto=m1)
>>> def go(v):
...     m1.setSpeed(v)
...     m2.setSpeed(v)
...     while taster.state() != 1:
...         txt.updateWait()
...     m1.setSpeed(0)
...     m2.setSpeed(0)
>>> go(500)
```

Hier ist *go()* eine Funktion, die (auch interaktiv) aufgerufen werden kann, um zwei Motoren synchron mit einer Geschwindigkeit *v* laufen zu lassen, bis der Taster gedrückt wird. Falls zur Eingabe des kleinen Programmes nicht der interaktive Modus, sondern der Datei-Editor (Strg+N)

verwendet wird, sieht das komplette Programm folgendermaßen aus:

```
import ftrobopy
txt=ftrobopy.ftrobopy('auto')
m1 = txt.motor(1)
m2 = txt.motor(2)
lampe = txt.output(5)
taster = txt.input(1)

txt.updateWait()

lampe.setLevel(512)
m1.setSpeed(0)
m2.setSpeed(0)
m1.setDistance(0, syncto=m2)
m2.setDistance(0, syncto=m1)
def go(v):
    m1.setSpeed(v)
    m2.setSpeed(v)
    while taster.state() != 1:
        txt.updateWait()
    m1.setSpeed(0)
    m2.setSpeed(0)

go(500)

txt.updateWait()
```

Es gibt hier zwei zusätzliche `txt.updateWait()`-Kommandos. Der Grund dafür ist der folgende: der TXT nimmt nur alle 0,01 Sekunden Kontakt zum PC auf, um die jeweils neuesten Befehle zu empfangen, die von einem *ftrobopy*-Hintergrundprozess bereitgehalten werden.

Das Kommando `txt.updateWait()` hält das Python-Programm so lange an, bis der nächste komplette Datenaustausch zwischen PC und TXT stattgefunden hat. Es könnte sonst passieren, dass das Python-Programm bis zum Ende einfach durchläuft, ohne dass auch nur ein einziger Befehl zum TXT gesendet wurde.

Der `txt.updateWait()`-Befehl am Ende des Python-Programms sorgt dafür, dass die Kommandos zum Stoppen der Motoren noch vollständig vom TXT empfangen werden, bevor das Python-Programm und damit auch der *ftrobopy*-Hintergrundprozess beendet werden.

Dies ist nur ein kleiner Ausschnitt der zur Verfügung stehenden Befehle. Die komplette Referenz mit allen Befehlen findet ihr im Benutzerhandbuch von *ftrobopy* [6].

Tipp: Grundsätzlich sollte man beim Erstellen von eigenen Python-Programmen möglichst darauf achten, Python-2 und -3 kompatiblen Code zu schreiben. Dies ist meistens recht einfach möglich. Im Web gibt es viele hilfreiche Seiten dazu (siehe z. B. [7, 8]). Das Modul *ftrobopy* ist selbst so geschrieben, dass es sowohl mit Python-2 (>2.7) als auch mit Python-3 verwendet werden kann.

Offline-Modus

Python-Installation auf dem TXT

Wer die Python-Programme nicht auf dem PC, sondern direkt auf dem TXT im *Offline-Modus* ausführen möchte, der benötigt einen Python-Interpreter auf dem TXT. Diesen kann man von den *ftcommunity*-Seiten herunterladen [9, 10]. Es handelt sich hier um einen Python-2.7-Interpreter mit einem Speicherplatzbedarf von ca. 19 MB. Dadurch ist es möglich den Interpreter auch auf einem TXT ohne zusätzliche microSD-Karte zu installieren. Ein Python-3-Interpreter ist deutlich grösser und lässt sich auf dem TXT nur mit zusätzlicher SD-Karte installieren.

Eine ausführliche Anleitung zur Installation des Python-Interpreters findet man bereits in einem früheren *ft:pedia*-Artikel [11].

Die *community-Firmware* (cfw) [2] für den TXT ist für den Python *Offline-Modus* deutlich besser geeignet als die original fischertechnik-Firmware. Überdies bringt die *community-Firmware* bereits standardmäßig einen Python-3-Interpreter mit. Auf den Seiten der cfw [2] gibt es weitergehende Hinweise zur Python-Programmierung des TXT. Insbesondere erhält man hier auch eine Anleitung für die Programmierung von Python-Programmen mit eigener grafischer Oberfläche auf dem Display des TXT.

Referenzen

- [1] Till Harbaum: *Auf zu neuen Ufern: Die Geschichte der „Community-Firmware“ für den TXT*. [ft:pedia 4/2016](#), S. 59-67.
- [2] [fischertechnik TXT Community firmware](#).
- [3] [Wine HQ](#).
- [4] [Python](#).
- [5] [ftrobopy](#), github.
- [6] Torsten Stuehn: [ftrobopy - Ansteuerung des fischertechnik TXT Controllers in Python](#). Manual, 2017.
- [7] Ed Schofield: [Cheat Sheet: Writing Python 2-3 compatible code](#). python-future.org.
- [8] [Bilingual Quick Reference](#). Python.org.
- [9] [Python 2.7 für den TXT](#). Downloadbereich der ftcommunity.
- [10] [Digitalkamera ftDigiCam v0.83](#). Downloadbereich der ftcommunity.
- [11] Torsten Stuehn: *Digitalkamera mit Autofokus und Live-Video-Vorschau*. [ft:pedia 1/2016](#), S. 74-76.