

Computing

Programming the TXT with Python

Torsten Stuehn

That you have programs for the TXT - not least thanks to the Linux operating system included - can develop in principle with practically any programming language is obvious. But how do you start in concrete terms? The post shows how easy this is with Python and the TXT driver developed by the author for the communityftrobopy succeed.

The Python programming language has now established itself in many fischertechnik projects with the TXT as a quasi-standard alongside graphic programming with ROBO Pro. This applies both to the original fischertechnik firmware and especially to the community firmware [1, 2], which is even partially programmed in Python.

In particular for users of Linux and MacOSX operating systems, who can only use ROBO Pro with the help of a Windows emulator (e.g. Wine [3]) or a virtual Windows machine, Python offers a simple and convenient way to use all the capabilities of the TXT to exhaust. Due to its widespread use in research and science, Python offers countless modules that hardly set any limits for the ambitious fischertechniker in the implementation of even the most complex projects.

As a full-blown programming language, Python also offers advanced concepts for accessing files and databases. Even a simple web server on the TXT can be implemented in just one line, and access to the TXT's microSD card is possible (according to the *mount* the card) of course also possible.

Since most Linux and MacOSX users have probably long been programming their TXT in Python, in the following text I am mainly addressing Windows users who are looking for a simple and quick introduction to this programming language or who are looking for a non-graphical alternative to ROBO Pro care.

Similar to ROBO Pro, the TXT can also be used with Python in *On-line* as well as in *Offline*-Program mode:

- ∂ *On-line*-Mode: The Python program runs entirely on a PC that is connected to the TXT via WLAN, USB cable or Bluetooth. To get the TXT in *On-line*-To control mode with Python, no intervention on the TXT is necessary. A Python interpreter must be installed on the PC for this. Linux and MacOSX already bring this with them.
- ∂ *Offline*-Mode: The Python program runs completely on the TXT. A PC is only required to edit the Python program and to transfer the program to the TXT and (usually) to start it. For this mode a Python interpreter has to be copied to the TXT. The community firmware has this already installed by default.

Online mode

Python installation on the PC

The easiest way to get started with Python programming of the TXT is via the *On-line-Mode*. Even a new, freshly unpacked TXT can be controlled immediately in Python without further modifications. To do this, you first need a Python interpreter on the PC. If you don't have this yet, you can get it for example. B. from the side python.org Download [4] free of charge. I recommend installing Python version 3.x (currently 3.6.1). During the installation you should set the option "Add Python 3.6 to Path" (see Fig. 1) so that Python can also be called from the command line.

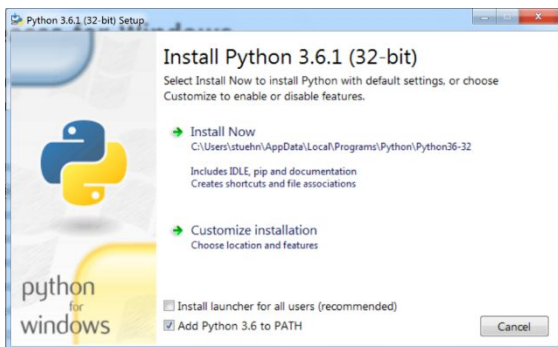


Fig. 1: Python installation dialog

After clicking on "Install Python 3.6.1", a simple Python interpreter is available on the Windows command line. The Integrated Python Development Environment (IDLE) can also be reached via the Windows Start menu.

To test whether Python has been installed correctly, you can test the Python version in the Windows command line ("cmd.exe") with the following command:

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

The output should then be the following:

```
Python 3.6.1
```

Python is now successfully installed on the PC.

Installation of ftrobopy on the PC

Before the TXT can be controlled in Python, the Python module *ftrobopy* needed. This module can be installed with the following command from the Windows command line:

```
C: \ Users \ ts> pip install ftrobopy
```

(Under Linux with "sudo pip install ftrobopy")

If there are problems with this, you can *ftrobopy* can also be downloaded or cloned via github [5]. The file *ftrobopy.py* can then be copied manually into the directory in which your own Python programs for the TXT are to be saved. This path should also be followed under MacOSX (for Mac experts)

ten: "pip install ftrobopy" Can z. B. can also be used with Anaconda-Python on the Mac).

In order to be able to control the TXT via the USB cable in Windows, the USB network driver is required. This is installed automatically when ROBO Pro is installed for the first time. Linux and MacOSX already bring the driver with them. So if you want to control the TXT under Windows via a USB cable, if you haven't already done so, [ROBO Pro](#) install on the computer.

The module *ftrobopy* basically works from firmware version 4.1.5. However, it is recommended to update the TXT to the latest version (currently 4.2.4).

First steps with IDLE

Now the Python development environment IDLE can be called via the Windows Start menu. This is an interactive Python shell that immediately executes every Python command entered (Fig. 2).



Fig. 2: Python development environment IDLE

The characters ">>>" in the shell represent the Python command prompt. A simplified, interactive Python shell can also be called at any time in the Windows command line with the "python" command. In a Linux or MacOSX terminal, the interactive Python shell is also started simply by entering "python".

A Python editor can also be called via the File → New File (or File → New File) menu in order to write and save more complex Python programs. These are not carried out immediately, but only after selecting "Run" in the menu or pressing the F5 key.

For the introduction and a few first tests, the interactive IDLE shell is completely sufficient. First of all, you have to *ftrobopy*-Module to be imported:

```
>>> import ftrobopy
```

If this module cannot be found, then is *ftrobopy* may have been installed in the wrong directory. Basically, the file *ftrobopy.py* either in the current directory or in the Python system path (e.g. under

"C:\Users\ts\AppData\Local\Programs\Python\Python36-32\Lib\site-packages") So that the module can be imported. If there are no errors, you can press

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

establish the connection to the TXT.

The TXT must be connected using one of the standard methods (USB / WLAN / Bluetooth). If the TXT is connected as a client to a WLAN access point, instead of 'automobile' the client IP address (e.g. '192.168.2.114') must be specified. From this point on, the TXT is available with all of its input and output ports in Python.

Whoever works with the community firmware must not forget to start the FT-GUI beforehand in order to enable the "online" mode.

Configuration and control of the inputs and outputs of the TXT

First of all, the inputs and outputs of the TXT can be configured with the following commands. In the example, two encoder motors are connected to outputs M1 / C1 or M2 / C2 and a button is connected to input I1. A lamp / LED is also connected to O5:

```
>>> m1 = txt.motor (1)
>>> m2 = txt.motor (2)
>>> lamp = txt.output (5)
>>> button = 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)
```

Here is *go ()* a function that can be called (also interactively) to make two motors synchronous at one speed *v* run until the button is pressed. If the interactive mode is not used to enter the small program, but the file editor (Ctrl + N)

is used, the complete program looks like this:

```
import ftrobopy
txt = ftrobopy.ftrobopy ('auto') m1 =
txt.motor (1)
m2 = txt.motor (2)
lamp = txt.output (5)
button = 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 ()
```

There are two additional ones here `txt.updateWait ()`-Commands. The reason for this is as follows: the TXT only contacts the PC every 0.01 seconds to receive the latest commands from a *ftrobopy*-Background process to be kept ready.

The command `txt.updateWait ()` holds the Python program until the next complete data exchange between PC and TXT has taken place. Otherwise it could happen that the Python program simply runs through to the end without a single command being sent to the TXT.

Of the `txt.updateWait ()`Command at the end of the Python program ensures that the commands to stop the motors are still completely received by the TXT before the Python program and thus also the *ftrobopy* background process are terminated.

This is only a small selection of the commands available. The complete reference with all commands can be found in the user manual of *ftrobopy* [6].

tip: When creating your own Python programs, you should always make sure that you write Python-2 and -3 compatible code. This is usually quite easy to do. There are many helpful pages on the web (see, for example, [7, 8]). The module *ftrobopy* is itself written in such a way that it can be used with both Python-2 (> 2.7) and Python-3.

Offline mode

Python installation on the TXT

If you don't run the Python programs on the PC, but directly on the TXT in *Offline-Want to run mode* that requires a python interpreter on the txt. This can be downloaded from the *ftcommunity* pages [9, 10]. This is a Python 2.7 interpreter with a memory requirement of approx. 19 MB. This makes it possible to install the interpreter on a TXT without an additional microSD card. A Python 3 interpreter is significantly larger and can only be installed on the TXT with an additional SD card.

Detailed instructions for installing the Python interpreter can be found in an earlier *ft: pedia* article [11].

The community firmware (cfw) [2] for the TXT is for the Python *Offline-Mode* much more suitable than the original *fischertechnik* firmware. In addition, the community firmware comes with a Python 3 interpreter as standard. On the pages of the cfw [2] there are further information on Python programming of the TXT. In particular, you will also find instructions for programming Python programs with their own graphical user interface on the display of the TXT.

credentials

- [1] Till Harbaum: *Off to new shores: The story of the "community firmware" for the TXT*. [ft: pedia 4/2016](#), Pp. 59-67.
- [2] [fischertechnik TXT Community firmware](#).
- [3] [Wine HQ](#).
- [4] [python](#).
- [5] [ftrobopy](#), github.
- [6] Torsten Stuehn: [ftrobopy - control of the fischertechnik TXT controller in Python](#). Manual, 2017.
- [7] Ed Schofield: [Cheat Sheet: Writing Python 2-3 compatible code](#). pythonfuture.org.
- [8th] [Bilingual Quick Reference](#). Python.org.
- [9] [Python 2.7 for the TXT](#). Download area of the ftcommunity.
- [10] [Digital camera ftDigiCam v0.83](#). Download area of the ftcommunity.
- [11] Torsten Stuehn: *Digital camera with auto focus and live video preview*. [ft: pedia 1/2016](#), Pp. 74-76.