

Fischertechnik und Python

Inzwischen in aller Munde - Python als Programmiersprache auch für Einsteiger. Da darf die Steuerung vom TXT-Controller nicht fehlen. Es gibt im Internet zahlreiche Dokumente zu diesem Thema. Ich beziehe mich vor allem auf die Anregung von Torsten Stuehn aus der ft:pedia Heft 2/2017. [1]

Entweder sind die Dokumentationen sehr Komplex oder nur Rudimentär. Es braucht also eine Menge Zeit, um den Konsens herauszuarbeiten und nicht jeder hat dazu Zeit oder Lust. Auch ist nicht jeder fachlich so versiert, sondern möchte nur einmal etwas ausprobieren. Im Folgenden habe ich die wichtigsten Python-Befehle zu Steuerung von Fischertechnik-Elementen zusammengetragen. Um sinnvolle Dinge zu programmieren, sind allerdings Grundkenntnisse der Python-Programmierung erforderlich. Diese habe ich unter : http://chobe.info/dokus/_Grundlagen.pdf bereitgestellt.

```
*test1.py - C:/Users/Axel/AppData/Local/Programs/Python/Python39/test1.py (3.9.6)*
File Edit Format Run Options Window Help
import ftrobopy
txt = ftrobopy.ftrobopy('192.168.178.54')
Taster = txt.input(1) # Objekt Taster vom Eingang 1
Lampe = txt.output(8) # Objekt Lampe von Ausgang 08
while 1: # while Schleife
    if Taster.state() == 0: # Vergleich auf Zustand 0
        Lampe.setLevel(512) # Lampe einschalten
    if Taster.state() == 1: # Vergleich auf Zustand 1
        Lampe.setLevel(0) # Lampe ausschalten
```

Alle Beispiele können mit copy and paste ausprobiert werden. Dabei ist folgendes zu beachten:

Die IP-Adresse ist dem eigenen TXT anzupassen oder auf „auto“ umzustellen.

Die Hochkommata sind eventuell zu ersetzen.

Kontrolle, ob die Einrückungen bei Schleifen gesetzt sind.

Zeilenenden werden durch Line-Feed-Zeichen gesetzt. Beim Editor muss das beachtet werden.

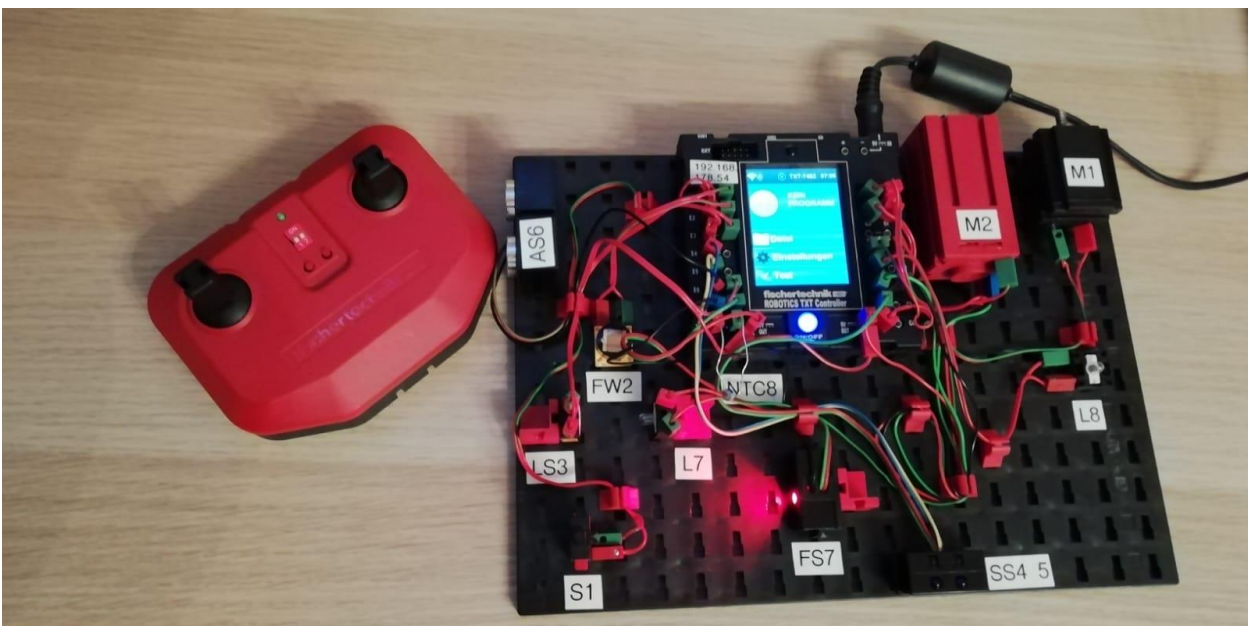

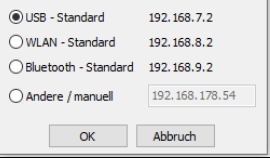
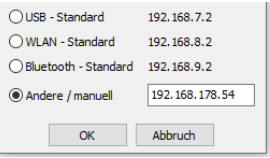


Abb. Meine Testumgebung für die einzelnen Komponenten.

Die folgenden Seiten können bei Bedarf auch als Nachschlagewerk ausgedruckt werden

Fischertechnik	
Entwicklungsumgebung  About IDLE 3.9.6 (64 bit)	Die Abkürzung IDLE steht für „Integrated Development and Learning Environment“ Download: https://www.python.org/
Bibliothek für TXT laden	<pre>import ftrobopy #Download: https://github.com/ftrobopy #Kopieren nach: Benutzer/xxx/AppData/Local/programs/Python/Python39/Lib</pre>
Verbinden USB 	# die Verbindungsart muss vorher unter RoboPro festgelegt werden: <pre>import ftrobopy txt = ftrobopy.ftrobopy('192.168.7.2', 65000) # Verbindung über USB # der Port 65000 ist Standard und kann daher auch bei Aufruf entfallen txt = ftrobopy.ftrobopy('auto')</pre>
Verbinden WLAN 	<pre>import ftrobopy txt = ftrobopy.ftrobopy('192.168.178.54', 65000) # Verbindung über WLAN # IP-Adresse dem TXT entnehmen print('Name des Controllers: ', txt.getDevicename()) # Ausgabe Gerätename print('Version: ', hex(txt.getVersionNumber())) # Ausgabe Versionsnummer</pre>
Lampe (Funktion output)	<pre>import ftrobopy # Bibliothek laden import time # Bibliothek laden (für Pausefunktion) txt = ftrobopy.ftrobopy('192.168.178.54') # Verbindung zum TXT Lampe = txt.output(8) # Erzeugung eines allgemeinen Output-Objekt, Ausgang 8 Lampe.setLevel(512) # Methode des Objektes (Helligkeit zw. 0 und 512) time.sleep(2) # 2 sek. Pause , Modul time erforderlich Lampe.setLevel(0) # Lampe aus</pre>
Motor (Funktion motor)	<pre>import ftrobopy # Bibliothek laden import time # Bibliothek laden (für Pausefunktion) txt = ftrobopy.ftrobopy('192.168.178.54') Motor = txt.motor(2) # Erzeugung eines Motor-Objekt, Motor 2 Motor.setSpeed(512) # Methodes des Objektes (Geschwindigkeit zw. 0 u. 512) time.sleep(2) # 2 sek. Pause, Modul time erforderlich Motor.stop() # Motor aus // (andere Richtung 0 bis -512)</pre>
Motor mit Zähleingang	<pre>import ftrobopy txt = ftrobopy.ftrobopy('192.168.178.54') Motor1 = txt.motor(2) # Erzeugung eines allgemeinen Motor-Objekt, Motor 2 Motor1.setDistance(144) # Distance auf 144 Impulse (72 Impulse pro Volldrehung) Motor1.setSpeed(512) while not Motor1.finished(): #solange der Motor läuft cd = Motor1.getCurrentDistance() #Methode um Distance auszulesen print("Current distance motor 1=", cd,) txt.updateWait()</pre>
2 Motoren synchronisieren	<pre>import ftrobopy txt = ftrobopy.ftrobopy('192.168.178.54') MotorL = txt.motor(3) # Erzeugung eines Motor-Objektes, MotorL MotorR = txt.motor(2) # Erzeugung eines Motor-Objektes, MotorR MotorL.setDistance(500, syncto=MotorR) # Synchronisation MotorL mit MotorR MotorR.setDistance(500, syncto=MotorL) # Synchronisation MotorR mit MotorL MotorL.setSpeed(512) # Motor starten mit Geschwindigkeit 512 MotorR.setSpeed(512) # Motor starten mit Geschwindigkeit 512</pre>
Eingang Digital Taster (Funktion input)	<pre>import ftrobopy txt = ftrobopy.ftrobopy('192.168.178.54') Taster = txt.input(1) # Objekt Taster vom Eingang 1 Lampe = txt.output(8) # Objekt Lampe von Ausgang 08 while 1: # while Schleife if Taster.state() == 1: # Vergleich auf Zustand 1 Lampe.setLevel(512) # Lampe einschalten if Taster.state() == 0: # Vergleich auf Zustand 0 Lampe.setLevel(0) # Lampe ausschalten</pre>

Eingang Digital Lichtschranke mit Fototransistor (Funktion input)	<pre>import ftrobopy import time txt = ftrobopy.ftrobopy('192.168.178.54') Lampe7 = txt.output(7) # Lampe für Lichtschranke Lampe7.setLevel(512) # Lampe 7 einschalten time.sleep(1) # Verzögerung der Auswertung bis Lampe an FT = txt.input(3) # Fototransistor while 1: # Dauerschleife (true) print(FT.state()) # Ausgabe des Status (1 = frei, 0 = Hindernis)</pre>
Eingang Digital Spurensucher (Funktion trailfollower)	<pre>import ftrobopy txt = ftrobopy.ftrobopy('192.168.178.54') SL = txt.trailfollower(4) # Objekt SL vom Eingang 4 SR = txt.trailfollower(5) # Objekt SR vom Eingang 5 while 1: # Dauerschleife print("links=", SL.state(), "rechts=", SR.state()) # Methode state() für Ausgabe</pre>
Eingang Analog IR Joystick (Funktion joystick) Auswertung Steuerkreuz	<pre>import ftrobopy import time txt = ftrobopy.ftrobopy('192.168.178.54') JS0 = txt.joystick(0) # Objekt JS0 vom linken Joystick (1=rechts) while 1: print("Links-Rechts =", JS0.leftright()) # Ausgabe: -1=links,0=mitte,1=rechts print("Oben-Unten =", JS0.updown()) # Ausgabe: 1=oben,0=mitte,-1=unten time.sleep(1)</pre>
Eingang Digital IR Joystick (Funktion joybutton) Auswertung Button	<pre>import ftrobopy txt = ftrobopy.ftrobopy('192.168.178.54') BT0 = txt.joybutton(0) # Objekt BT0 vom linken Button (1=rechts) Motor = txt.motor(1) # Objekt Motor vom Ausgang 1 while 1: # while Schleife if BT0.pressed() == 1: # Vergleich auf Zustand 1 Motor.setSpeed(512) # Motor einschalten if BT0.pressed() == 0: # Vergleich auf Zustand 0 Motor.setSpeed(0) # Motor ausschalten</pre>
Eingang Digital IR Joystick (Funktion joydipswitch) Auswertung DIP-Schalter	<pre>import ftrobopy txt = ftrobopy.ftrobopy('192.168.178.54') DS = txt.joydipswitch() # Objekt DS erstellen print("DIP: ", DS.setting()) # Ausgabe mit Methode setting # OFF OFF = 0; ON OFF = 1; OFF ON = 2, ON ON = 3; Joystick muss betätigt sein!!!</pre>
Eingang Analog Foto- u. Widerstand (Funktion resistor, Methode value)	<pre>import ftrobopy txt = ftrobopy.ftrobopy('192.168.178.54') R = txt.resistor(2) # Objekt R vom Eingang 2 print("Der Widerstand beträgt ", R.value()) # Ausgabe mit Methode value von R</pre>
Eingang Analog Temperatur mit NTC (Funktion resistor, Meth. ntc.Temperature)	<pre>import ftrobopy txt = ftrobopy.ftrobopy('192.168.178.54') T = txt.resistor(8) # Objekt T vom Eingang 8 print("Temperatur: ", T.ntcTemperature()) # Methode ntc.... Für Ausgabe T</pre>
Eingang Analog Ultraschallsensor (Funktion ultrasonic)	<pre>import ftrobopy txt = ftrobopy.ftrobopy('192.168.178.54') US = txt.ultrasonic(6) # Objekt US vom Eingang 6 while 1: # Dauerschleife print("Abstand: ", US.distance(), " cm.") # Ausgabe mit Methode distance()</pre>
Eingang Analog Spannungsmessung (Funktion voltage)	<pre>import ftrobopy txt = ftrobopy.ftrobopy('192.168.178.54') B = txt.voltage(1) # Objekt B vom Eingang 1 print("Spannung ", B.voltage(), " mV") # Ausgabe Spannung mit Methode voltage</pre>
Eingang Analog Farbsensor (Funktion colorsensor)	<pre>import ftrobopy txt = ftrobopy.ftrobopy('192.168.178.54') FS = txt.colorsensor(7) # Objekt FS vom Eingang 7 print("Der Farbwert ist: ", FS.value()) # Ausgabe des Wertes mit Methode value</pre>

	<code>print("Die Farbe ist: ", FS.color())</code>	# Ausgabe als Wort mit Methode color
Soundausgabe auf internen Lautsprecher (Funktion <code>play_sound</code>)	<code>import ftrobopy</code> <code>txt = ftrobopy.ftrobopy('192.168.178.54')</code> <code>txt.play_sound(4,1,50)</code>	# 4 = Nummer des Sounds 0 bis 29 (Siehe entspr. Doku) # 1 = Anzahl der Wiederholungen, 50 = Lautstärke 0 bis 100

Bildaufnahme mit Kamera

Das Packet PIL (Python Imaging Library) muss installiert werden. Die neuere Version heißt dabei Pillow. Das kann alles in der Kommandozeile (CMD) durchgeführt werden. Zuerst sollte geprüft werden, ob der Paketmanager *PIP* installiert ist.

```
C:\Users\Axel\AppData\Local\Programs\Python\Python39\Scripts>pip
```

Mit unten gezeigten Aufruf, kann PIP evt. auf den neuesten Stand installiert werden.

```
\Python\Python39>python.exe -m pip install --upgrade pip
```

```
Successfully installed pip-23.1.2
```

Nun wird Pillow installiert.

```
C:\Users\Axel\AppData\Local\Programs\Python\Python39\Scripts>pip install pillow
```

Es wird automatisch aus dem Netz geladen und installiert.

```
Successfully installed pillow-9.5.0
```

```
import ftrobopy
from PIL import Image
import time
txt = ftrobopy.ftrobopy('192.168.178.53')
txt.startCameraOnline()      #startet den Prozess auf dem TXT
time.sleep(2.5)              # Das Starten des Prozess dauert 2 Sek, deshalb eine Pause
im = "TXTimage.jpg"
pic = txt.getCameraFrame()
with open(im,'wb') as f:
    f.write(bytearray(pic))
txt.stopCameraOnline()
image = Image.open(im)      # Anzeige des Bildes
image.show()
```



Das Bild wird nun angezeigt und kann als png gespeichert werden.

Status der Ein und Ausgänge auslesen

Man kann den aktuellen Status mit der Funktion getConfig() auslesen. Diese Funktion gibt zwei Listen zurück. Die erste Liste enthält den Status der Ausgänge (Motor M1..M4 oder einpoliger Ausgang O1..O8). Die zweite Liste enthält 8 Tupel mit zwei Elementen. Das erste Element ist der Typ des Eingangs und das zweite ist der Modus des Eingangs (analog oder digital). In den Listen sind nur Zahlen enthalten.

```
import ftrobopy
```

```
txt = ftrobopy.ftrobopy('192.168.178.53')
```

```
M, I = txt.getConfig()
```

```
Lampe = txt.output(1)
```

```
Motor = txt.motor(2)
```

```
SL = txt.trailfollower(1)
```

```
R = txt.resistor(2)
```

```
B = txt.voltage(3)
```

```
FS = txt.colorsensor(4)
```

```
US = txt.ultrasonic(5)
```

```
T = txt.resistor(6)
```

```
FT = txt.input(7)
```

```
B = txt.voltage(8)
```

```
print("Outputs configuration:", M)
```

```
print("Inputs configuration:", I)
```

Ausgabe:

```
Outputs configuration: [0, 1, 1, 1]
```

```
Inputs configuration: [(0, 1), (1, 0), (0, 0), (0, 0), (3, 0), (1, 0), (1, 1), (0, 0)]
```

Type		
C_OUTPUT	0	Lampe, Summer, Elektromagnet, Magnetventil
C_MOTOR	1	Klassischer Motor, Encodermotor

Mode		C_ANALOG	C_DIGITAL
Type		0	1
C_VOLTAGE	0	Farbsensor, Spannung	Spurensucher
C_SWITCH	1		Taster, Magnetsensor
C_RESISTOR	1	Poti, NTC, Widerstand	Phototransistor
C_ULTRASONIC	3	Ultraschall	