Application Programming with FOTEMP Devices

Weidmann Technologies Deutschland GmbH

V1 2019-07-12

Tino Schurzmann

Content

1 General

Existing software for FOTEMP devices can be found here:

- http://www.optocon.de/support/downloads-fotemp/ or
- http://www.optocon.de/support/dokumentationen-publikationen/

Not all application scenarios will be covered with the existing material. Also Weidmann has not the ressources to develop each possible use case.
Therefore the decision was made years before today to open up the protocol description of the so called “Optocon Ascii Interface”.

This interface can be used with every terminal emulation program like RealTerm (https://sourceforge.net/projects/realterm/files/latest/download).

Setting up the program as shown in Figure 1 and 2 is enough to communicate with a FOTEMP device via USB or RS232.

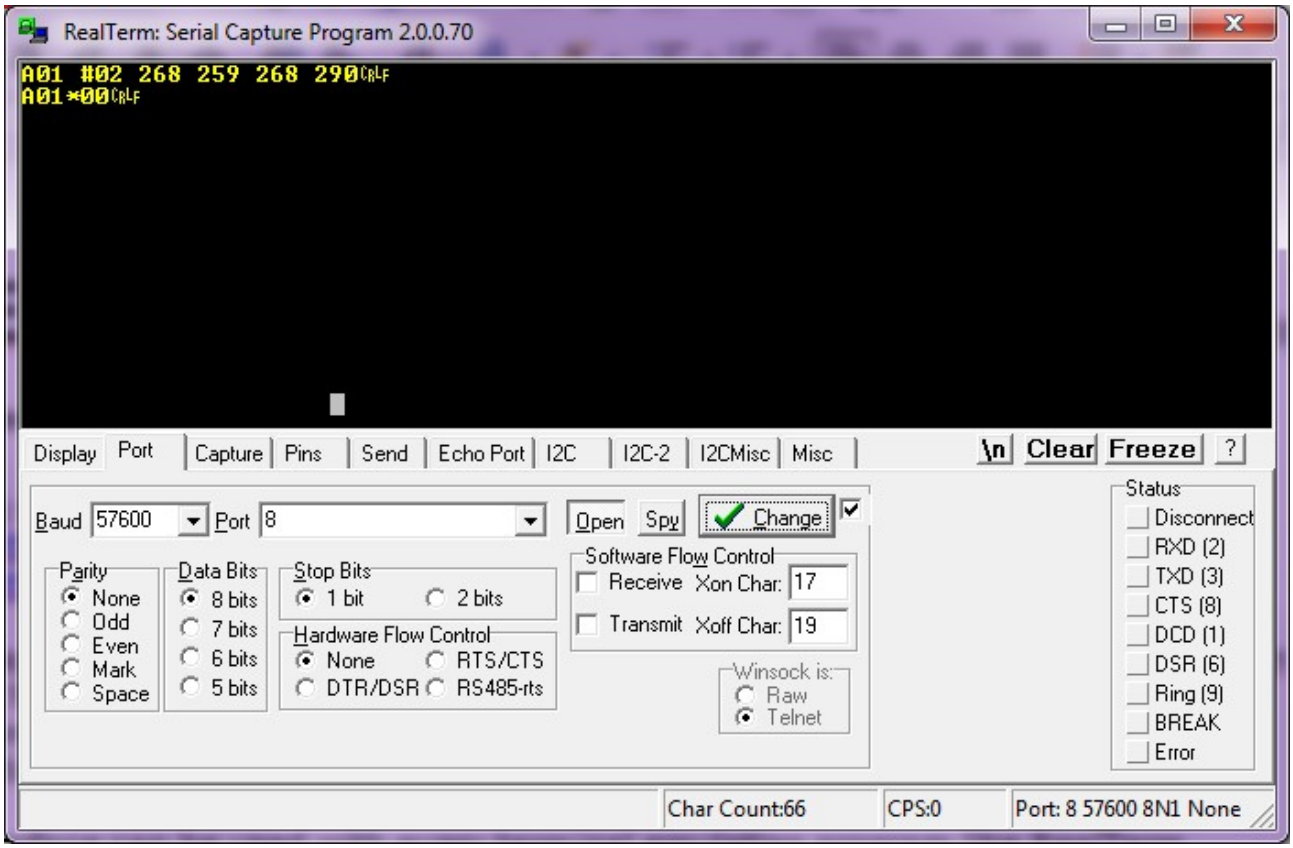

Figure 1 FOTEMP communication – Port Settings

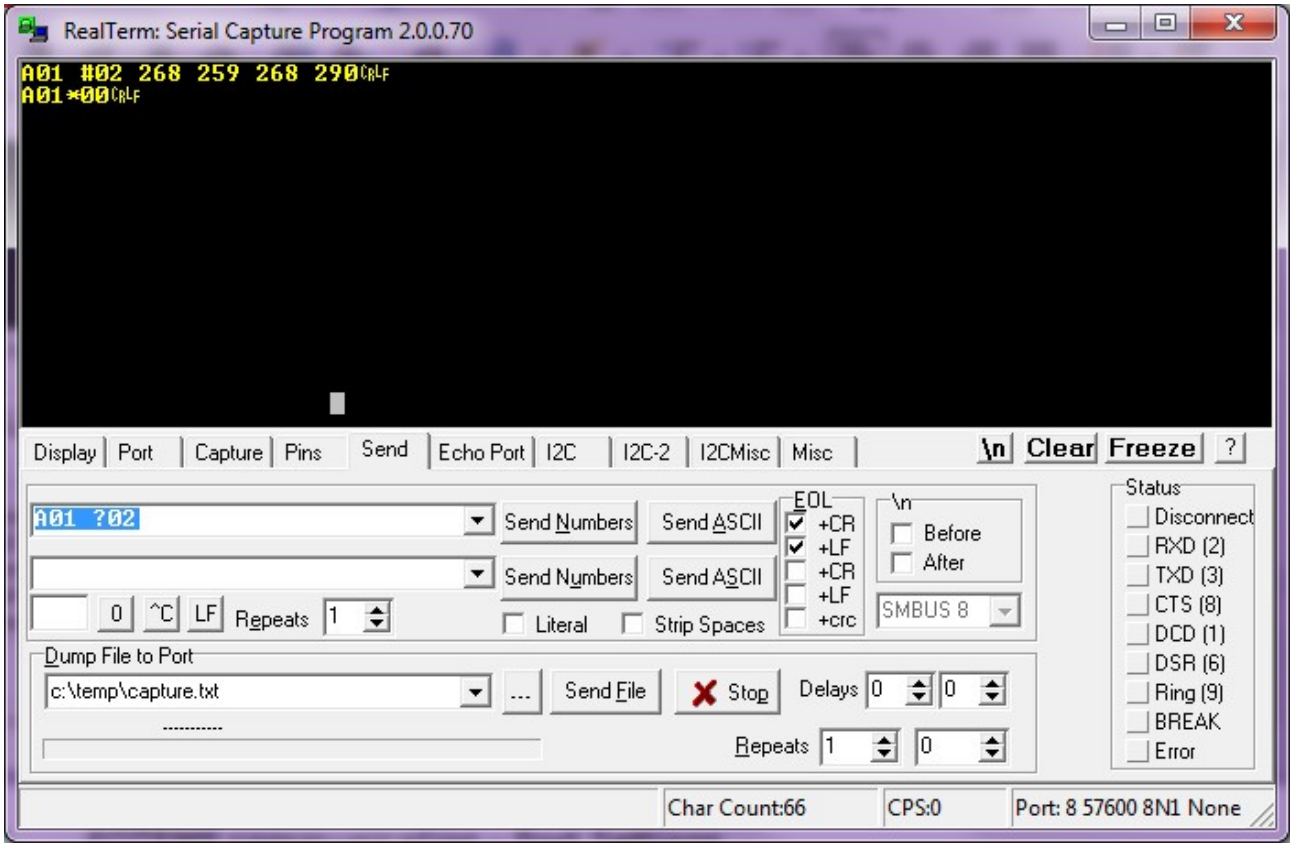

Figure 2 FOTEMP communication – data exchange

In Figure 1 RealTerm was set to 57600 baud, 8n1, without any handshake option.
Figure 2 shows the command and respons to read all temperatures. As it can be seen, the “CR” and “LF” options are enabled.

Using devices other than FOTEMP FTMS, the leading “A01” for talking to module 1 does not exists.

More to that in the several communication protocol documents at www.optocon.de.

2 Preinstallation of the PC

The later used scripts are Python scripts. Therefore the programming language interpreter “Python” has to be installed in its version 2.7.10 or higher. Additionally a communication via a COM port (RS232 or USB) is needed, so the package “pyserial-2.7” must be installed too. This document shows the usage on a Win10 PC. Linux works in the same way.

All scripts need a COM port, so we are starting here.

1. Click with your mouse on the lower left desktop corner and choose “Device manager” and find the COM port under “Connections”
2. download “python-2.7.16” from here for your system: https://www.python.org/downloads/release/python-2716/, Linux has Python preinstalled
3. download “pyserial-2.7.win32” from here for your system: https://sourceforge.net/projects/pyserial/files/pyserial/2.7/, Linux: enter “pip install pyserial” in bash
4. install Python (from step 1)
5. install “PySerial” from step 2
6. reboot your Windows PC
7. copy the script files (find them on www.optocon.de)
8. only for Linux: change the lines “*ser = serial.Serial("COM"+str(comport), COMPORT_SPEED, timeout=COMPORT_TIMEOUT)*” to “*ser = serial.Serial("<your_registered_device_path>"+str(comport), COMPORT_SPEED, timeout=COMPORT_TIMEOUT)*” in all scripts
9. open a command line (Win+R, enter “cmd”, press Enter key, Ctrl+Alt+T on Linux)
10. browse to the script folder

You are now ready to execute the scripts from the following pages.

2.1 Logging temperatures with FOTEMP OEM-MNT 1-Channel Device

This section shows how a temperature measurement can be done using a measurement script. Exemplarilly the used script is shown in Figure 3.

```
#!/usr/bin/env python
# coding: utf8
# --------------------------------------------------------------------------------
# python 2.7.10 and pyserial2.7
# HW: FOTEMP-OEM-MNT 1CH
# --------------------------------------------------------------------------------
# TEST SETTINGS ...
COMPORT = 'COM41'
COMPORT_TIMEOUT = 1                     # in seconds
COMPORT_SPEED = 57600                   # in baud
# --------------------------------------------------------------------------------
import serial,time
from datetime import datetime

def get_temp(ser):
        rc1=""
        ser.write("?02\r\n")
        rc1 =ser.readline()
        rc1+=ser.readline()
        rc2=rc1.split('\r')[0]
        rc3=rc2.split(' ')
        return rc3[1]

def get_id_addr(ser):
        rc1=""
        ser.write("?41\r\n")
        rc1 =ser.readline()
        rc1+=ser.readline()
        rc2=rc1.split('\n')
        rc3=rc2[0].replace('\r','')
        rc3=rc3.replace("#41",'')
        rc3=rc3.replace(' ','')
        rc3=rc3.replace('3','')
        return rc3

if __name__ == "__main__":
        print "*start at "+datetime.now().strftime('%Y-%m-%d %H:%M:%S.%f')
        ser = serial.Serial(COMPORT, COMPORT_SPEED, timeout=COMPORT_TIMEOUT)
        print "*used com port: ",ser.name
        print "*dev_id="+ get_id_addr(ser)
        while 1:
                s = datetime.now().strftime('%Y-%m-%d %H:%M:%S.%f')
                s=s+" "+str(int(get_temp(ser))/10.0)+"C"
                print s
                time.sleep(1) #sec
        #print "*done"
        #ser.close()                    ... not needed, break with Ctrl+C
```

Figure 3 reading temperatures from a single channel FOTEMP OEM-MNT device

2.2 Logging temperatures with FOTEMP OEM-PLUS 4-Channel Device

This section shows how a temperature measurement can be done using a measurement script. Exemplarilly the used script is shown in Figure 4.

```
#!/usr/bin/env python
# coding: utf8
# --------------------------------------------------------------------------------
# python 2.7.10 and pyserial2.7
# HW: FOTEMP-OEM-PLUS 4CH
# --------------------------------------------------------------------------------
# TEST SETTINGS ...
COMPORT = 'COM7'
COMPORT_TIMEOUT = 1                     # in seconds
COMPORT_SPEED = 57600                   # in baud
# --------------------------------------------------------------------------------
import serial,time
from datetime import datetime

def get_temp(ser):
        rc1=""
        ser.write("?02\r\n")
        rc1 =ser.readline()
        rc1+=ser.readline()
        rc2=rc1.split('\r')[0]
        rc3=rc2.split(' ')
        return rc3[1:]

def get_id_addr(ser):
        rc1=""
        ser.write("?41\r\n")
        rc1 =ser.readline()
        rc1+=ser.readline()
        rc2=rc1.split('\n')
        rc3=rc2[0].replace('\r','')
        rc3=rc3.replace("#41",'')
        rc3=rc3.replace(' ','')
        rc3=rc3.replace('3','')
        return rc3

if __name__ == "__main__":
        print "*start at "+datetime.now().strftime('%Y-%m-%d %H:%M:%S.%f')
        ser = serial.Serial(COMPORT, COMPORT_SPEED, timeout=COMPORT_TIMEOUT)
        print "*used com port: ",ser.name
        print "*dev_id="+ get_id_addr(ser)
        while 1:
                s=datetime.now().strftime('%Y-%m-%d %H:%M:%S.%f')
                temperatures=get_temp(ser)
                #print temperatures
                s=s+" C1: "+str(int(temperatures[0])/10.0)
                s=s+" C2: "+str(int(temperatures[1])/10.0)
                s=s+" C3: "+str(int(temperatures[2])/10.0)
                s=s+" C4: "+str(int(temperatures[3])/10.0)
                print s
                time.sleep(1) #sec
                # break with Ctrl+C
```

Figure 4 reading temperatures from a 4-channel FOTEMP OEM-PLUS device

2.3 Logging temperatures with FOTEMP ModuleSystem (FTMS)

This section shows how a temperature measurement can be done using a measurement script. Exemplarilly the used script is shown in Figure 5. Figure 6 shows the resulting output.

```
#!/usr/bin/env python
# coding: utf8
# --------------------------------------------------------------------------------
# python 2.7.10 and pyserial2.7
# HW:  FOTEMP-FTMS with four modules (RS232)
# Modules:
# slot 1: 4ch+display
# slot 2: none
# slot 3: 2ch+display
# slot 4: 1ch+display
# slot 5: 4ch (no display)
# --------------------------------------------------------------------------------
# TEST SETTINGS ...
COMPORT = 'COM8'
COMPORT_TIMEOUT = 1                     # in seconds
COMPORT_SPEED = 57600                   # in baud
# --------------------------------------------------------------------------------
import serial,time
from datetime import datetime
DEBUG=0
#DEBUG=1

def get_temp(ser,address):
        rc1=""
        ser.write("A0"+str(address)+" ?02\r\n")
        rc1 =ser.readline()
        rc1+=ser.readline()
        rc2=rc1.split('\r')[0]
        rc3=rc2.split(' ')
        return rc3[1:]

def get_id_addr(ser,address):
        rc1=""
        ser.write("A0"+str(address)+" ?41\r\n")
        rc1 =ser.readline()
        rc1+=ser.readline()
        rc2=rc1.split('\n')
        rc3=rc2[0].replace('\r','')
        rc3=rc3.replace("#41",'')
        rc3=rc3.replace("A0"+str(address),'')
        rc3=rc3.replace(' ','')
        rc3=rc3.replace('3','')
        return rc3

if __name__ == "__main__":
        print "*start at "+datetime.now().strftime('%Y-%m-%d %H:%M:%S.%f')
        ser = serial.Serial(COMPORT, COMPORT_SPEED, timeout=COMPORT_TIMEOUT)
        print "*used com port: ",ser.name
        slots=[1,3,4,5]
        for i in slots:
                print "*dev_id"+str(i)+"="+get_id_addr(ser,i)
                time.sleep(0.5) #sec
        while 1:
                for i in slots:
                        s=datetime.now().strftime('%Y-%m-%d %H:%M:%S.%f')
                        temperatures=get_temp(ser,i)
                        temperatures=temperatures[1:]
                        for c in temperatures:
                                s=s+" "+str(int(c)/10.0)
                        print s
                        #print s
                time.sleep(1) #sec
        #print "*done"
        #ser.close()                    ... not needed, break with Ctrl+C
```

Figure 5 reading temperatures from a multi-channel FOTEMP Modulesystem

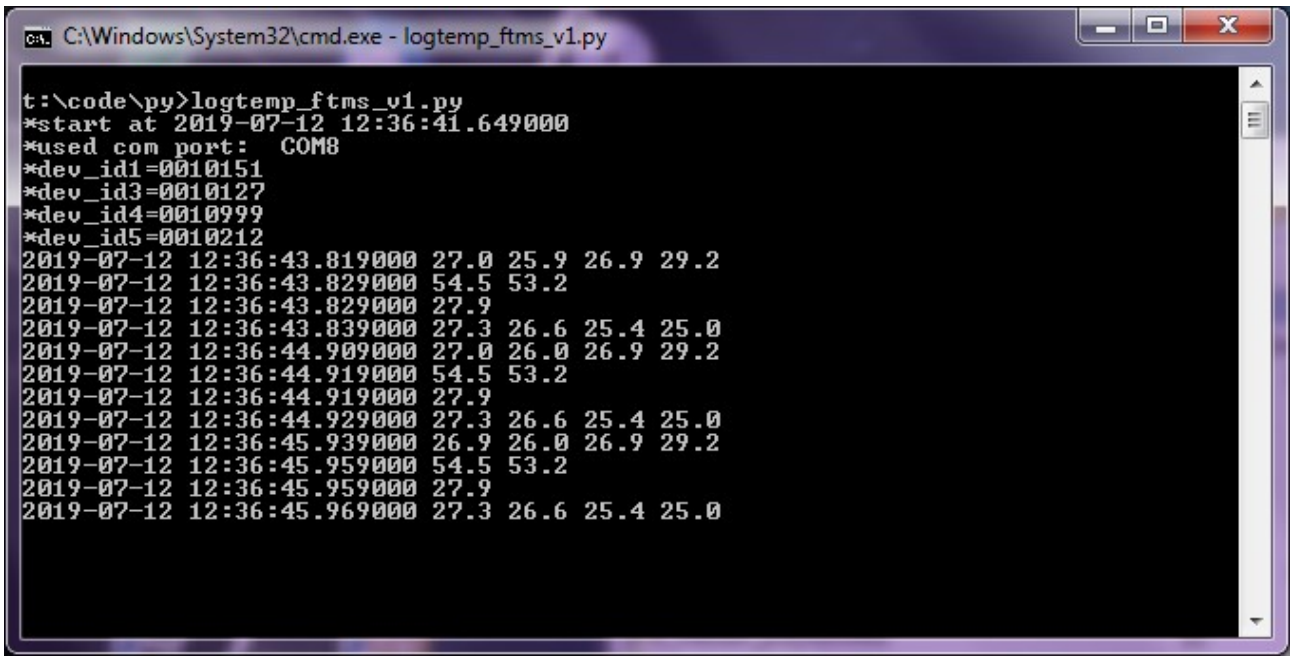

Figure 6 FOTEMP FTMS script results

- EoD -