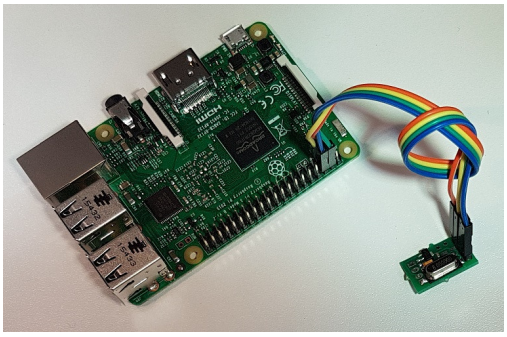
APPLICATION NOTE

How to read multiple (remote)Smartec temperature Sensors over your LAN

By means of a commercial Raspberry Pi board and some Smartec SMT172 sensors a very accurate data acquisition system can be built with an accuracy of better than 0.1 C. The temperatures can be readout via your LAN system which is available on the internet. The software example below is, as ever, can be downloaded for free and of course can be changed by the user.



Connection of one Smartec I²C board to the rPi

The Smartec SMT172 to I²C interface board allows a number of SMT172 temperature sensors to be connected to an I²C bus. In the SMT172 to I²C interface manual it is explained how this can be done using a Raspberry Pi (**rPi**) board and a few very simple Python programs for programming the I²C addresses and reading the data.

In this application note we will show you how you can use the same setup for accessing the temperatures from your PC over your LAN via ethernet cable or WIFI. This application note is based on an example I²C bus setup with two Smartec SMT172 temperature sensors connected by means of two interface boards, exactly as described in the interface manual. The two respective secondary data addresses that we use in this example are the same as before i.e. 0x70 and 0x73.

How to read multiple (remote)Smartec temperature Sensors over your LAN

On the rPi side

The pins Vin, GND, I²C_SCL and I²C_SDA have to be connected to rPi pins no 4, 6, 5 and 3 respectively of the GPIO header. If you run the following Python program on the rPi, then it will listen to the chosen port (in this case 13000) for a data request from your PC, while continuously reading the values from the temperature sensors. The data request in our example is no more than one single character 'X', but that is a free choice, as long as the chosen string is the same on the rPi and the PC. On reception of the request, the rPi will send the available temperature values as one string, for example '22.34 23.41' . Based on the number of sensors that you have connected to the I²C bus, the program can easily be extended to support more than 2 sensors.

Note

The program is an oversimplified implementation of IP socket communication and only given as an example of how a number of (remote) temperature sensors can be accessed using your network. Because of this simple setup, any other device connected to your LAN could also 'listen' to the temperature strings being transmitted.

```
import os
import smbus
import select
from socket import *
bus = smbus.SMBus(1)
pda  = 0x6f # 111d primary device address
sda1 = 0x70 # 112d secondary device address from the first
I2C interface
sda2 = 0x73 # 115d secondary device address from the second
I2C interface
host = ''
port = 13000
buf = 1024 #for receiving requests only
addr = (host,port)
UDPSock = socket(AF_INET,SOCK_DGRAM)
UDPSock.bind(addr)
UDPSock.setblocking(0)


try:
```

How to read multiple (remote)Smartec temperature Sensors over your LAN

```
        while 1:
            a = bus.read_byte_data(sda1, 0)
            b = bus.read_byte_data(sda1, 1)
            c = bus.read_byte_data(sda1, 2)
            temp1 = round((a*256*256+b*256+c-273150)/1000.,2)
            if temp1 == round (temp1,1):
                temp1Str = str(temp1) +'0'
            else:
                temp1Str = str(temp1)
            a = bus.read_byte_data(sda2, 0)
            b = bus.read_byte_data(sda2, 1)
            c = bus.read_byte_data(sda2, 2)
            temp2 = round((a*256*256+b*256+c-273150)/1000.,2)
            if temp2 == round (temp2,1):
                temp2Str = str(temp2) +'0'
            else:
                temp2Str = str(temp2)
            print (temp1Str,temp2Str)
            ready = select.select([UDPSock],[],[],2)
            if ready[0]:
                (data,addr) = UDPSock.recvfrom(buf)
                if data.decode() =="X":
                    UDPSock.sendto(temp1Str +' '+temp2Str, addr)

    except KeyboardInterrupt:
        UDPSock.close()
        print('Program ended')
```

On the PC side

The following Python program can be run on your PC. Every second it will send a data request over the LAN to host 192.168.1.15 port 13000. You will have to replace the IP number by the IP number of your own rPi. On reception of the data string, the temperatures will be displayed. Depending on the number of sensors connected, you will have to increase the delay between readings, to allow the rPi to read all of the used I^2C addresses, before being able to respond to a data request. That's all there is to get it work.

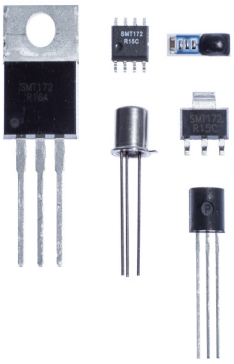
How to read multiple (remote)Smartec temperature Sensors over your LAN

```
import os
import select
from socket import *
from time import sleep
host = '192.168.1.15'
port = 13000
buf = 1024
addr = (host,port)
UDPSock = socket(AF_INET, SOCK_DGRAM)

def readSocketOnce():
    global addr
    UDPSock.sendto("X".encode(), addr)
    ready = select.select([UDPSock], [], [], 2)
    while not ready[0]:
            print('x', end = '', flush = True)
            UDPSock.sendto("X".encode(), addr) # retry
            ready = select.select([UDPSock], [], [], 2)
    (data, addr) = UDPSock.recvfrom(buf)
    print (data.decode())

try:
    while 1:
            readSocketOnce()
            sleep(1)

except(KeyboardInterrupt):
    UDPSock.close()
    print('Program ended')
```

How to read multiple (remote)Smartec temperature Sensors over your LAN

Related products:

SMT172-SOT223	SMT172 in SOT223 encapsulation
SMT172-TO18	SMT172 in TO-18 encapsulation
SMT172-TO92	SMT172 in TO-92 encapsulation
SMT172-TO220	SMT172 in TO-220 encapsulation
SMT172-SOIC	SMT172 in SOIC-8 encapsulation
SMT172-HEC	SMT172 in HEC encapsulation



SMT172 to IIC	converts SMT172 signal to I²C signal (or to SMT160 signal)
SMTRVS1802	4.5 mm ss. probe with SMT172 TO18 sensor and 5 meter shielded cable

sales@smartec-sensors.com