

This repository | Search

[Pull requests](#) [Issues](#) [Marketplace](#) [Gist](#)[greenfox-academy](#) / [teaching-materials](#) Private[Watch](#) ▾

13

[★ Star](#) 4[Fork](#) 22[Code](#)[Issues](#) 11[Pull requests](#) 0[Projects](#) 0[Wiki](#)[Insights](#) ▾Branch: [master](#) ▾[teaching-materials](#) / [project](#) / [hardware](#) / [temperature-logger](#) /[Create new file](#)[Upload files](#)[Find file](#)[History](#)

farkaszolt committed on GitHub Update README.md ...

Latest commit c37c557 on Jun 12

..

img	feat(hw-temp-log-proj): add tasks	2 months ago
README.md	Update README.md	a month ago
SKILL-IO.md	feat(hw-temp-log-proj): add SKILL-IO	2 months ago

[README.md](#)

Temperature logger

Create a program on the MCU and on your PC which can communicate with each other

Objectives

- Create a bigger project
- Learn how to use the serial port in a PC software

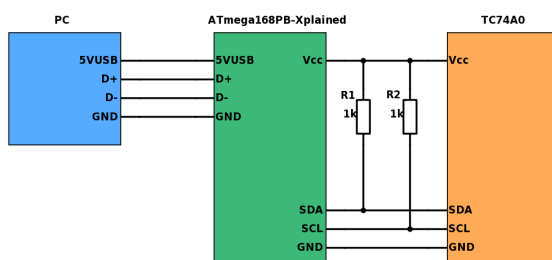
Materials & Resources

- [time.h](#)
- [RS232 library](#)

Workshop

This project is based on the [I2C workshop](#).

A program should be written which can log temperature data on a PC. The ATmega168PB Xplained board is used as a gateway between the PC and the temperature sensor.



The program can take few different commands. Please use the `getch()` function to implement them.

Basic tasks

- [Print usage](#)
- [Show the command list](#)
- [Exit from the program](#)
- [List available ports](#)
- [Set port name](#)
- [Set write filename](#)

- [Open port](#)
- [Start logging / Stop logging](#)
- [Close port](#)

Advanced tasks

- [Set read filename](#)
- [Calculate average in given period](#)

Print usage

- At application startup the following message should be shown:

```
Todo application
=====
Commands:
h      Show the command list
e      Exit from the program
l      List available ports
p      Set port name
w      Set write filename
o      Open port
s      Start logging / Stop logging
c      Close port
r      Set read filename
a      Calculate average in given period
```

Show the command list

- if the user presses the "h" button
- the program should print out the command list (like at startup)

Exit from the program

- if the user presses the "e" button
- the program exits

List available ports

- if the user presses the "l" button
- the program should list out the available serial ports

Set port name

- if the user presses the "p" button
- the program should ask the user to enter a port name
- the user input errors should be handled

Set write filename

- if the user presses the "w" button
- the program should ask the user to enter a file name
- this file will be used as the log file
- the user input errors should be handled

Open port

- if the user presses the "o" button
- the program should open the previously set COM port
- the errors should be handled

Start logging / Stop logging

- if the user presses the "s" button

- the program should start logging from the opened port into the set write file
- if the program is already logging, then the logging should be stopped
- the errors should be handled
- the logfile should look like this:

```
2017-05-31 10:46:22      23.6
2017-05-31 10:46:23      23.7
2017-05-31 10:46:24      23.8
2017-05-31 10:46:25      23.9
2017-05-31 10:46:26      23.2
2017-05-31 10:46:27      23.5
2017-05-31 10:46:28      23.4
```

Close port

- if the user presses the "c" button
- the program should close the opened port
- the errors should be handled

Set read filename

- if the user presses the "r" button
- the program should ask the user to enter a file name
- this file will be used as the input file for the average calculation
- the user input errors should be handled

Calculate average in given period

- if the user presses the "a" button
- the program should ask the user to enter a date period
- the program then reads data from the read file and calculates the average temperature in the given period
- the errors should be handled

Solution

[Solution](#)

