Temp Monitor Readme

By Guy Koren,

23.09.2017

This file contains documentation on how to install and run the GUI for monitoring temperature measurements from MAX6755 k-thermocouple sensors via Arduino MEGA interface.

The code files can be downloaded here: https://github.com/gkoren/TempMonitor/tree/master, download the folder to your computer and follow these instructions.

There are 2 steps to control the monitoring:

Step 1: Arduino code

The first step is to install the Arduino IDE – which allows to upload code onto the Arduino and controlling the sensors. The Arduino code should run in the background regardless of the GUI.

Arduino has great documentation, and if there are any parts I left out you can look for info in their web page: https://www.arduino.cc/ or simply google the issue – there are usually answers there.

In the Arduino folder there are 3 subfolders: 2 containing the code files (one for serial communication and one for wifi communication via UDP), and another folder with necessary libraries (don't need to do nothing with these files, just have them saved).

- **Read_Temperature_Serial** contains the Arduino code (.ino) for reading the data via serial. This file needs to be uploaded on the board through the IDE. The file can pretty much be ran "out-of-the-box", Simply open the Arduino IDE and open one the file, then click upload, and wait to receive the message "Done Uploading". At the moment it is set to read only the first three sensors, if you want to change the number of sensors you can do it from the code by changing the value for "n_sensors".
- **Read_Temperature_WiFi** is very similar to the serial code, only that it requires some small editing in the code before using with a new computer. The code is set to use in the lab, and if you want to run it on another network I'll add another explanation soon.

The code needs to know 2 parameters: IP address ('HOST_NAME'), and port ('HOST_PORT'). You can retrieve the IP address by running 'ipconfig' from the command line in Windows (look for IPv4), or just google how to do it with mac/unix. For the port – it's just a name (number) so shouldn't matter much. I had no issues with any number on my Lenovo, but some computers may have different constraints on the port number. Right now it is set to 4444 (Which works for me and Taya) and I wouldn't change it unless you see some problems.

Note that there are two extra files in this folder, they are needed for the wifi card libraries.

After uploading these files you can check to see that everything is working by clicking the "Serial Monitor" in the Arduino IDE.

Note: When using the serial monitor: The baud-rate is different for serial (9600) and wifi (115200) so you need to change that in the serial monitor window depending on what type of communication you're using.

After you're able to successfully run (one of) these files move on to install the GUI for easy monitoring and saving the data.

Step 2: Temp Monitor GUI

The GUI will only be of use when the Arduino code runs in the background. Once the Arduino code is uploaded, nothing you do in this screen affects it, and you can run as many tests as you want from here without having to change anything in the Arduino code.

There are two ways to run the GUI.

- The easier way - Temp_monitor.exe (Windows only):
  I bundled the python code to an .exe file which runs the GUI on Windows (no need to have python installed). The files are in the 'dist' folder. It's pretty large (~120 MB) and there are quite a lot of files in the folder, but all you need to do is run Temp_monitor.exe
- Using the .py scripts:
  You need to have python 2.7 or higher to use this method, and you need to download two more python libraries: 1. *Matplotlib* and 2. *PyQt4*. Just look for them online and find how to get them. In the home folder there are .py and .ui files. (These are the same files that are in the python folder). The main script is Temp_monitor.py, and just run it with: *'python Temp_monitor.py'*

Step 3: Starting the measurements

The main window is the same no matter what type of communication (serial/wifi) you're using. Just chose from the top panel.

- For serial: There are two parameters to set in the panel. The baud-rate is set to 9600 and no need to change that. For the Port, you need to enter the USB port name that the Arduino is connected to. If you can't find it you can look for it in the IDE: Tools->Port. When you set these parameters click Start and the monitoring begins.
- For Wifi: You need to enter the IP address and UDP port (the same one that are in the Arduino code that you've uploaded). Again click Start to begin monitoring.

After clicking Start you'll see (within a few sec) that the data is showing and being updated on the main window. You can click on 'sensors statistic' for a visual display of 'temperature vs time' for each sensors. Only the last 20 measurements are saved in the GUI window, but when you export the data to .csv it contains the full set of measurements. You don't need to click 'export to .csv', you'll be asked automatically when you click 'Stop' to terminate the test.

This pretty much it. Email me if you have any additional questions.

Guy Koren

guy.koren@cern.ch