MoCoS

Monitoring & Control System

Manual

October 27, 2014

Florian M. Brunbauer

under supervision of
Dr. Leszek Ropelewski
Dr. Eraldo Oliveri
Dr. Filippo Resnati

The development of the Monitoring & Control System would not have been possible without the much appreciated help of all involved persons.

Special thanks go to Filippo Resnati, who provided the original motivation for the monitoring system and steered the project throughout all phases from conceptualisation to development and application. Mr. Resnati designed most of the signal conditioning circuits and his contribution was essential for the success of the project.

The Monitoring & Control System was equally influenced and supported by Eraldo Oliveri. He provided many crucial ideas and recommendations to arrive at a reliable device and was incredibly helpful throughout the course of the project. Thank you for all the support.

Many thanks to Leszek Ropelewski for providing a great work environment and atmosphere and for allowing me to work in his group long enough to finish the project.

I would also like to thank Patrik Thuiner and Silvia Franchino for many helpful discussions and all the help in the lab, Miranda Van Stenis for machining the front panels, Christophe Bault for 3D printing both the prototype and the final case for the portable version of the Monitoring & Control System, Hans Müller for providing sensors and help, Alexandru Rusu for his help in designing the signal conditioning PCB and everyone else who was involved in the project.

Thank you all!

Contents

List of Figures

1 Introduction

Aiming at making common monitoring and control tasks more intuitive and efficient, the Monitoring & Control System ("MoCoS") acts as an interface between various types of sensors and a data logging service. All acquired quantities are accessible online to make it easy to keep track of the status and the evolution of a system remotely. Additionally, MoCoS acts as a control system, which takes actions depending on the values obtained from connected sensors.

MoCoS consists of two main components as shown in figure 1: a MoCoS device which is physically connected to sensors and control hardware and a web service that keeps track of measurements and allows online configuration of monitoring and control channels. Additionally, a desktop application is available, which can be used to setup the MoCoS device on a local network and to access the latest measured values directly from the device.

In line with its goal of making the monitoring of systems as intuitive as possible, MoCoS features 16 universal inputs, which can be configured to work with different sensors and standard signals. While directly interfacing with some types of sensors such as resistive or capacitive sensors, standard voltage or current signals supplied by most existing measurement hardware can be related to physical quantities making MoCoS highly versatile and adaptable to existing setups.

Triggering actions based on the measured values can be done in a binary way using thresholds or in a more sophisticated manner using built in PID-logic controllers. The status of all control channels is accessible online in the same way as the monitored values.

The following pages will provide a detailed description of the capabilities of MoCoS, instructions for setting up and using the Monitoring & Control System and a technical overview of the devices. A quick-start guide to setup a new MoCoS device and start monitoring sensors within minutes is also supplied.

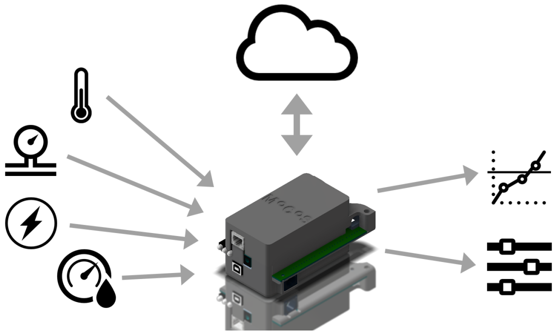


Figure 1: Overview of MoCoS: The device acquires data from different sensors and controls external devices. The web service provides the configuration and logs the measurements.

2 Quick-Start Guide

To setup your MoCoS device and start acquiring data within minutes, please follow the steps of this quick-start guide. The appropriate sections containing more detailed information about the features, usage and configuration of the Monitoring & Control System are referenced and should be consulted to learn more about the capabilities of MoCoS or when experiencing any difficulties.

2.1 Setting up a web service account

- Go to mocos.web.cern.ch and click on "Create new account".
- On the first page, fill in general information about your new account. The account is not your user account which you will use to login to the web service, but is uniquely associated with a single MoCoS device. Click "Next" when done.
- The next page pertains to your user account. Please enter your personal information and select a username and password you will later use to login to the web service. Click "Next" when done.
- Confirm the information on the next page. Once you have confirmed the shown information, an activation e-mail will be sent to the email address you entered.
- Click on the link in the activation email to activate your account. You have now created a MoCoS account and a user account associated with it.

For more information about web service accounts, please refer to section 6.

2.2 Connecting your device

- Connect an Ethernet cable to the RJ45 port at the front of your device to allow your MoCoS device to connect to your local network.
- Connect sensors that you want to monitor. For example, you could connect PT1000 temperature sensors between the positive and negative terminals of any of the universal inputs.
- Connect the device to a 7.5V DC power source (2.1mm plug, center pin positive) that can provide at least 500mA.
- Your device will now turn on.
- Check that the SD card can be read (left/top LED should turn green).
- Your device will not be able to connect to your network since its MAC and IP addresses are not configured yet (center LED will turn red).

For detailed information about the required connections and the status LEDs, please refer to section 10.

2.3 Specifying connection parameters

In order for your MoCoS device to work properly and start acquiring data, it must be connected to your local network and reach the configuration and logging servers (usually via the internet).

- Login to the web service with your newly created user account, download the MoCoS desktop application to your computer, which can be found in the "Help" section, and open it.
- Connect the MoCoS device to your computer via a USB cable.
- Select the port your device is connected to on the first screen of the desktop application and click "Connect".
- Wait until the right/bottom LED is yellow. This means that your device is not connected to the network but can receive data from the desktop application.
- Once connected, please enter the MAC address of your device (in the format "123, 123, 123, 123, 123, 123" - you can find it on a sticker at the bottom of the Ethernet Shield) and click "Apply".
- Enter the IP address the device should use on your local network (in the format "123, 123, 123, 123") and click "Apply".
- Enter the account username (can be found in the "Account" section of the web service) and click "Apply".
- Enter the account password (can be found in the "Account" section of the web service) and click "Apply".
- The configuration and logging server addresses should be left at the default values unless you have been instructed otherwise.
- Once you have entered and uploaded (by clicking "Apply") all necessary connected parameters, disconnect your device from your computer, close the desktop app and power-cycle the device.
- Your device should now be ready to connect to the network. If the connection is successful, the center LED will turn green.

For more information about using the desktop app to set the network connection parameters, please refer to section 9.

2.4 Configuring universal inputs

Once your device can connect to your network you can configure it using the web service.

- Login to your account on the MoCoS web service with the username and password chosen during account creation and go to the "Monitoring" section.

- If you have already connected a sensor to one of the universal inputs, move the cursor over the input it is connected to and click on the configure button (gear icon).

- You will now be presented with the configuration form used to correctly identify the connected sensor and convert the raw value it provides (e.g. voltage) to a meaningful physical quantity (e.g. temperature).

- Select the dependence of the physical quantity on the raw value (e.g. linear, inverse, exponential) and enter the slope, offset and, if applicable, logarithmic conversion parameters.

- Click on "Save" to apply the new configuration. If you do not know the conversion parameters, please refer to section 4.3 to perform a sensor calibration.

Your device will download the new configuration the next time it starts up.

2.5 Viewing data

Once you have connected a sensor to one of the universal inputs, specified its configuration and the device has downloaded these settings from the web service, it will start acquiring data.

- Go to the "Monitoring" section of the web service and select the input your sensor is connected to.

- You will be presented with a graph of the physical quantity measured by your sensor.

To learn more about the MoCoS web service, please refer to section 8.

Your MoCoS device is now setup and ready to start acquiring data from up to 16 sensors simultaneously. Please consult the appropriate sections to learn more about the monitoring and control capabilities of the system.

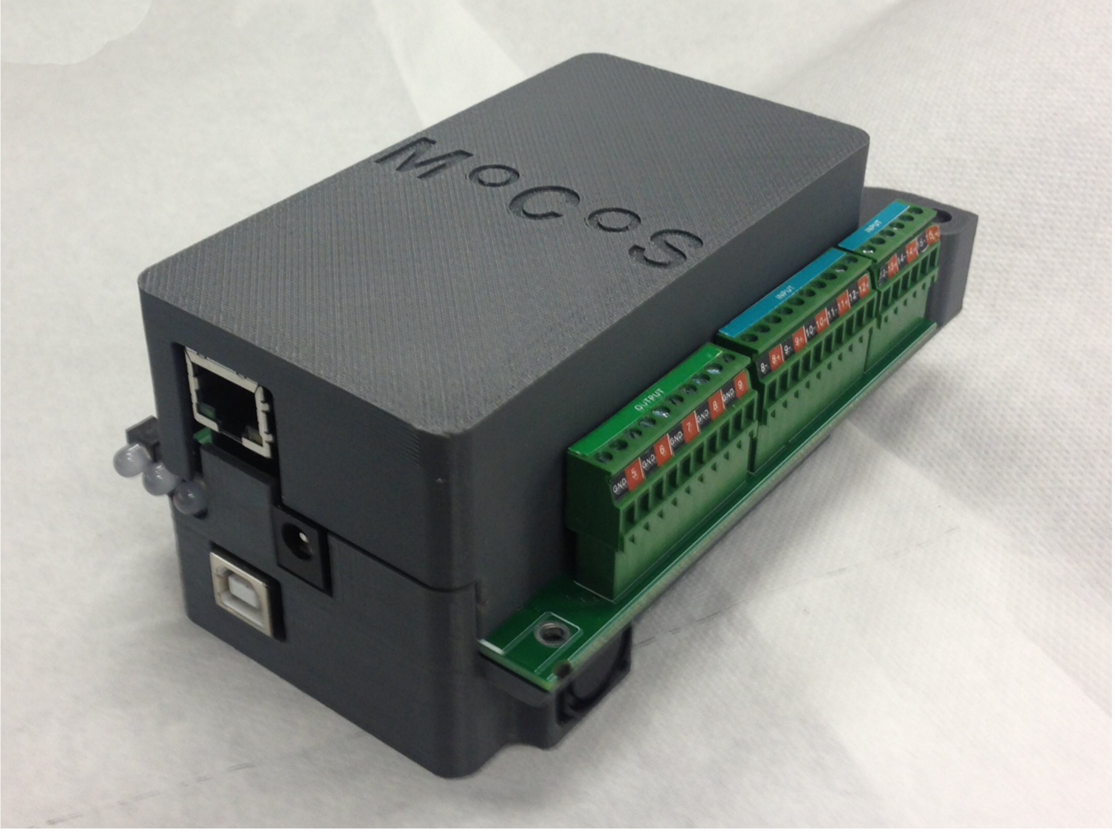


Figure 2: Standalone device: The portable, standalone version of the MoCoS device is housed in a 3D printed case and has its connectors arranged all around the device.

3 Device Models

There are two different models of MoCoS devices: a portable, standalone version ("portable device") and a Eurocrate panel version ("crate version"). Both versions have the same monitoring and control capabilities but differ in the way the connectors are arranged. This section will describe the main features and differences of the two versions.

3.1 Standalone device

To make the standalone device as small and versatile as possible, its connectors are arranged all around the device. While the power and communication ports (USB and Ethernet) are on the front of the device, there are universal inputs and outputs on the left and right sides of the device and two RS232 ports on the back. The input and output connectors are pluggable terminal blocks with a pitch of 3.5mm and are separated into two 10-way and two 6-way connectors for the universal inputs and two 10-way connectors for the outputs. While it is possible to mount any 3.5mm pitch terminal blocks, it is recommended to use the pluggable solution consisting of headers and plugs mentioned in the parts list in appendix B. For the portable device, the 3.5mm pitched pads along the sides of the MoCoS PCB which are closest to the pins connecting the circuit board to the Arduino should be used. An Arduino Mega 2560 is the basis of the electronics of the MoCoS device. The PCB containing all circuits for signal conditioning and other features is connected to the Arduino board as a shield, eliminating the need for cables. On top of the MoCoS PCB, an Arduino Ethernet Shield is mounted to make network connections possible. The MoCoS PCB, which by default is sized for the crate version, can be adapted for the portable device by cutting away two parts on both sides of the circuit board. Detailed information about trimming the MoCoS PCB to fit the portable device can be found in section 10.4. Please read the entire instructions on cutting the PCB carefully as improper handling can damage or destroy the circuit board. The housing for the portable version is a 3D printed box that consists of a bottom piece, to which the Arduino board is directly fixed and a top cover than can be slid on and fixed with two screws at the back of the device. The 3D files for the top and bottom parts of the box can be found in the "Help" section of the MoCoS web service.

3.2 Crate version

A crate version of the MoCoS device, which can be mounted in a Eurocrate chassis, is available. The main difference to the standalone version is the placement of the connectors: to make all inputs and outputs easily accessible when the device is mounted in a crate, all communication, supply, input and output ports are located on the front of the device (with the exception of one of the two RS232 ports, which is accessible from the back). There are 5mm pitched pads on the same side of the PCB on which the status LEDs and power socket are located. These are used to mount pluggable terminal blocks with a pitch of 5mm for the universal inputs and outputs for the crate version. The input and output connectors are separated into four 8-way connectors for the universal inputs and two 10-way connectors for the outputs. To fit all the connectors on a Eurocrate-compatible PCB, the terminal blocks are mounted on both sides of the circuit board. While it is possible to mount any 5mm pitch terminal blocks, it is recommended to use the pluggable solution consisting of headers and plugs mentioned in the parts list in appendix B. For the crate version, only the 5mm pitched pads along the front side of the MoCoS PCB should be used and the 3.5mm pitched pads used for the portable version should remain unused. As with the portable device, an Arduino Mega 2560 is the basis of the electronics of the MoCoS device. The PCB containing all circuits for signal conditioning and other features is connected to the Arduino board as a shield, eliminating the need for cables. On top of the MoCoS PCB, an Arduino Ethernet Shield is mounted to make network connections possible. Only the PCB is fixed to the front panel, while the Arduino and Ethernet Shield are kept in place by the headers connecting them to the circuit board between them.

The MoCoS PCB is by default the correct size to be mounted directly into a Eurocrate chassis (233mm high, 116mm wide). Therefore, no trimming of the PCB is necessary

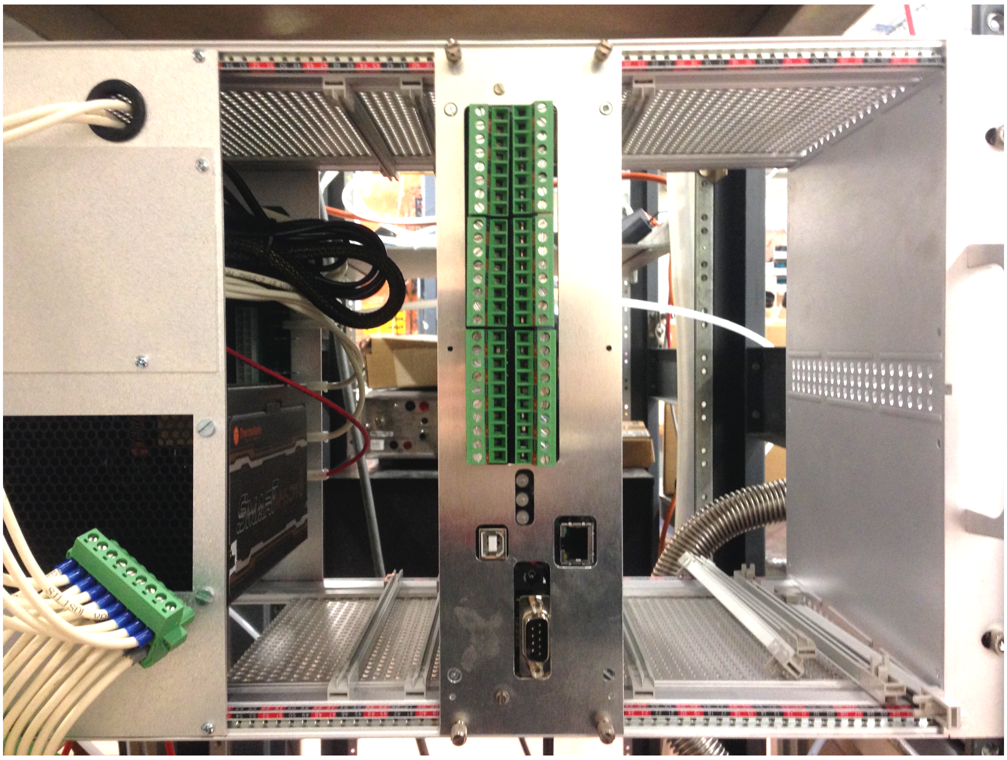


Figure 3: Crate version: The crate version of the MoCoS device can be mounted in a Eurocrate chassis and has its connectors on one side.

if you want to mount it in a crate. It can be mounted and fixed to a Eurocrate front panel by two two screws on the top and bottom of the front side of the circuit board.

4 Monitoring

The monitoring capabilities of the MoCoS device are based on 16 universal inputs and internal circuits for signal conditioning and interpretation that allow relating raw signals supplied by sensors or external devices to meaningful physical quantities with good precision and accuracy. The term "universal" inputs refers to the possibility to connect any supported sensor or signal to any input terminal of the MoCoS device. There is no need to worry about which input terminal a specific type of sensor has to be connected to since each input can be configured to work with any of the supported sensors and standard signals. This allows you to use the 16 inputs in the way best suited for your monitoring needs: You can connect 16 temperature sensors, 7 capacitive sensors and 9 standard voltage signals from external devices, 3 frequency signals and 13 TTL signals, ... As long as the inputs are properly configured, any combination of up to 16 sensors works.

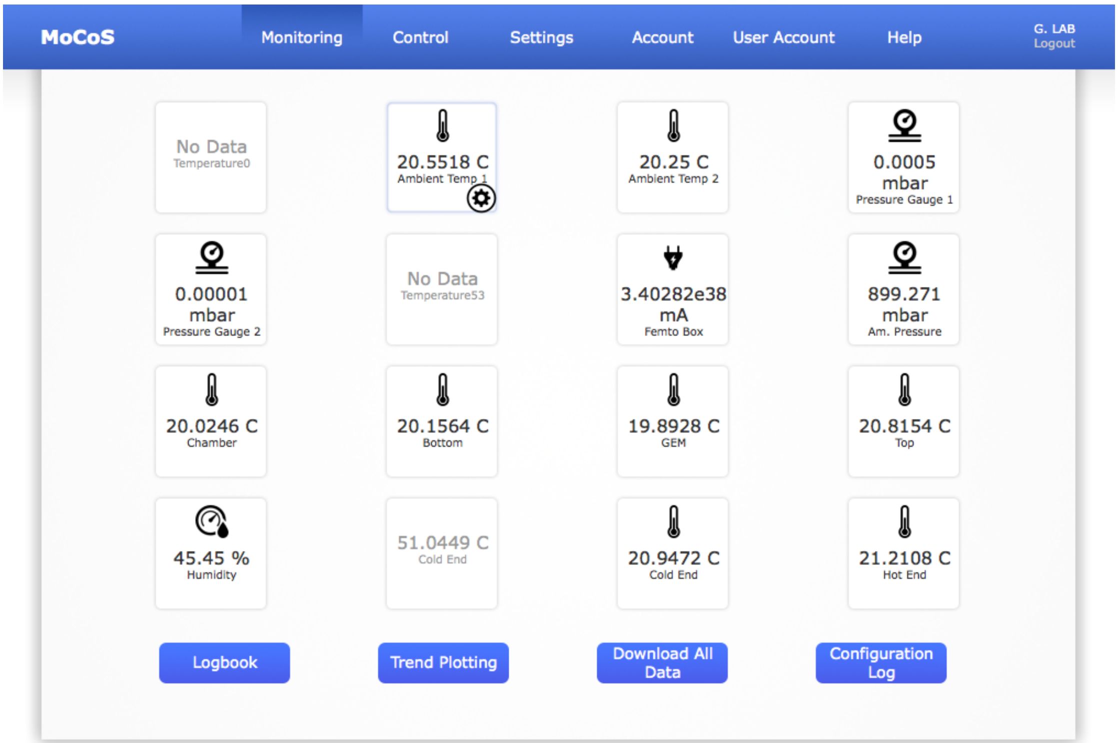
MoCoS supports the following types of sensors and standard voltage and current signals:

- Frequency signals (TTL logic)
- Voltage signals (0-5V and 0-10V)
- Current signals (4-20mA)
- Capacitive sensors
- Resistive temperature sensors (PT100 and PT1000)

For each sensor or input signal, the raw value measured by the microcontroller inside the MoCoS device has to be related to the physical quantity by a calibration procedure. For more details on calibrating a sensor, please refer to section 4.3. The configuration of universal inputs is performed exclusively through the MoCoS web service and downloaded and applied by the MoCoS device upon startup. Additionally to acquiring data from the supported types of sensors, the MoCoS device can also communicate with external devices through a RS232 serial interface. This allows you to query data from an external device, store it on the MoCoS web service and have it accessible online together with any other data acquired by MoCoS. For more information on setting up RS232 communication and querying values from an external device, please refer to section 11.

4.1 Configuring universal inputs

The process of configuring universal inputs will be explained using two examples: a resistive PT1000 temperature sensor and an external vacuum gauge controller that provides a voltage signal ranging from 0V to 10V which can be related to the measured pressure. Other types of sensors are setup in the same way as described here. Only



No Data
Temperature0

20.5518 C
Ambient Temp 1

20.25 C
Ambient Temp 2

0.0005 mbar
Pressure Gauge 1

0.00001 mbar
Pressure Gauge 2

No Data
Temperature53

3.40282e38 mA
Femto Box

899.271 mbar
Am. Pressure

20.0246 C
Chamber

20.1564 C
Bottom

19.8928 C
GEM

20.8154 C
Top

45.45 %
Humidity

51.0449 C
Cold End

20.9472 C
Cold End

21.2108 C
Hot End

Figure 4: Monitoring page: Selecting "Monitoring" in the navigation bar brings up an overview off the latest values measured by the input channels. Icons indicate the type of physical quantity monitored. When moving the cursor over a channel, a configuration button (gear icon) appears. Click this button to access the configuration page for a specific input channel.

the sensor type and calibration parameters have to be adjusted for other types of sensors. The configuration process described here has to be followed for each sensor or signal connected to a universal input that should be monitoring and used by MoCoS.

Before you start configuring a universal input to use a specific sensor or standard signal, please connect the sensor or signal to the input terminal you want to use and perform a calibration as explained in section 4.3.

When you have connected your sensor and performed a calibration to obtain the parameters necessary to relate the raw signal to a physical quantity, go to the MoCoS web service, login and select "Monitoring" from the menu bar. You will be presented with an overview of all 16 inputs. Enabled inputs will display the latest value acquired, while disabled ones will be greyed out. Move your cursor over the input your sensor is connected to (the numbers of the inputs are shown next to their names) and click on the configure button (gear icon). The "Monitoring" page with the configuration button for channel 1 is shown in figure 4. Clicking this button will bring up the configuration form for the selected input. First, you should specific which type of sensor or signal is connected to the input you are configuring. Please make sure that the type you select corresponds to the connected sensor or signal since selecting a wrong type can result in damages to the internal circuits of your MoCoS device. In the case of a resistive PT1000 temperature sensor, you should select "PT1000"" while the exemplary vacuum gauge controller should be configured as a "0-10V" signal.

If you want to start monitoring immediately, check the "Enabled" checkbox. To just setup the input and start monitoring later, leave the box unchecked.

The name and description of an input are not used by the MoCoS device but might make it easier to keep track of all your sensors on the MoCoS web service.

To limit the amount of data uploaded and saved, you can specify a "Significant Change" and enable "Limited Logging". This will only upload and save the measured values, when a value significantly different from the last logged one is measured. For example, if you enable limited logging and specify a significant change of 10%, any fluctuations of less than 10% around the last value that was saved will not be recorded. This allows you to limit the data sent and saved by MoCoS, while still registering all interesting events. If you prefer to log every measurement, simply disable limited logging.

The signal conversion parameters specify how a raw signal measured by the microcontroller is converted to a meaningful physical quantity. Please select the "Physical Unit" of the value that your sensor is monitoring. For the temperature sensor, this could be Celsius, while for the vacuum gauge controller you should use Millibar.

The conversion from the raw signal can be linear, inverse or exponential. For example, the temperature sensor may have a linear relationship between the raw value and the temperature, while the vacuum gauge controller will work over many orders of magnitude and will therefore have an exponential relationship between the raw signal and the pressure. The specific conversion parameters (slope and offset for linear and inverse relationships and exponential base, exponential period and offset for exponential relationships) can be obtained by performing a sensor calibration or directly from the manual of the connected sensor. In all cases, it is recommended to perform a calibration regardless of the supplied conversion parameters in order to achieve the highest possible accuracy and precision.

In the case of a linear relationship, the physical quantity y is obtained from the raw value x by the expression

$$y(x) = a * x + b \tag{1}$$

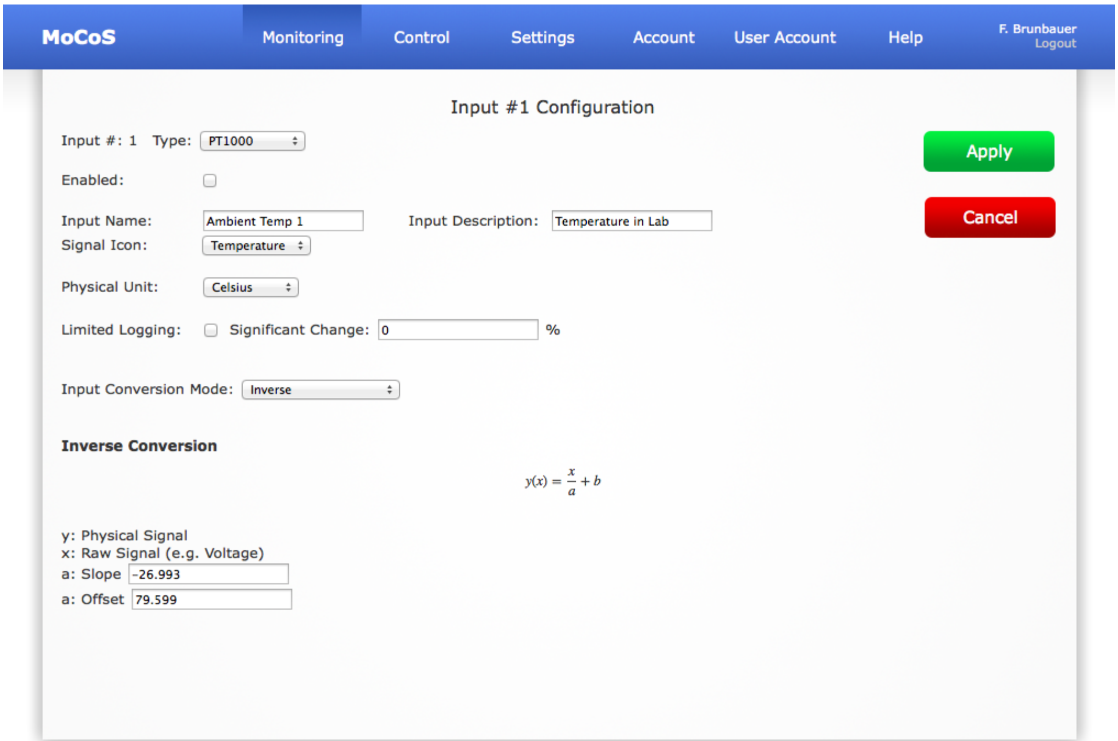
For example, a PT1000 temperature sensor might have an offset of 120 and a slope of -30, meaning that a raw signal of 2V would correspond to 60 °C, while a raw signal of 3V would mean 30 °C.

In the case of an exponential relationship, the physical quantity is obtained from the raw value either by the expression

$$y(x) = a^{\frac{x}{b}} \tag{2}$$

or by

$$y(x) = a^{x-b} \tag{3}$$



Input #1 Configuration

Input #: 1 Type: PT1000

Enabled: □

Input Name: Ambient Temp 1 Input Description: Temperature in Lab

Signal Icon: Temperature

Physical Unit: Celsius

Limited Logging: □ Significant Change: 0 %

Input Conversion Mode: Inverse

Inverse Conversion

$$y(x) = \frac{x}{a} + b$$

y: Physical Signal
x: Raw Signal (e.g. Voltage)
a: Slope -26.993
a: Offset 79.599

Figure 5: Input configuration: To configure a universal input to accept data from a connected sensor, first specify the type of sensor, enable the channel, enter a name and specify how the raw value should be converted to a physical quantity. Then, choose a physical unit and enter the conversion parameters (a and b) as shown in the mathematical expression for the chosen conversion type, enter a custom conversion expression or specify the query parameters for getting data via RS232 from an external device connected to one of the serial ports.

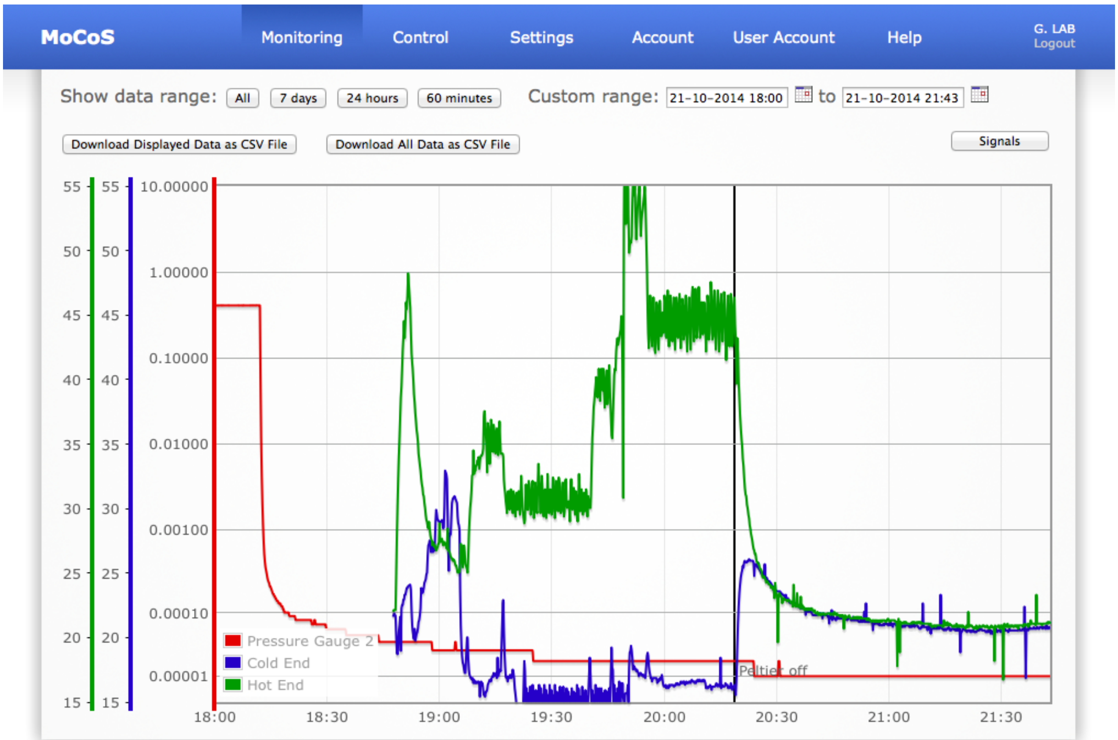
For example, the vacuum gauge controller might have an exponential base of 10, an exponential period of 1 and an offset of 10^{-10}, meaning that a raw signal of 2V would correspond to a pressure of 10^{-8} mbar, while a raw signal of 3.5V would mean $3.166 * 10^{-7}$ mbar.

For inverse conversions, the physical quantity is obtained from the raw value by the following expression:

$$y(x) = \frac{x}{a} + b \tag{4}$$

Once you select the conversion mode, the configuration page will display a conversion formula with parameters a and b that converts the raw value x to the physical quantity y, as shown in figure 5 for the case of an inverse conversion.

You can also enter a custom mathematical expression that calculates the physical



Custom range: 21-10-2014 18:00 to 21-10-2014 21:43

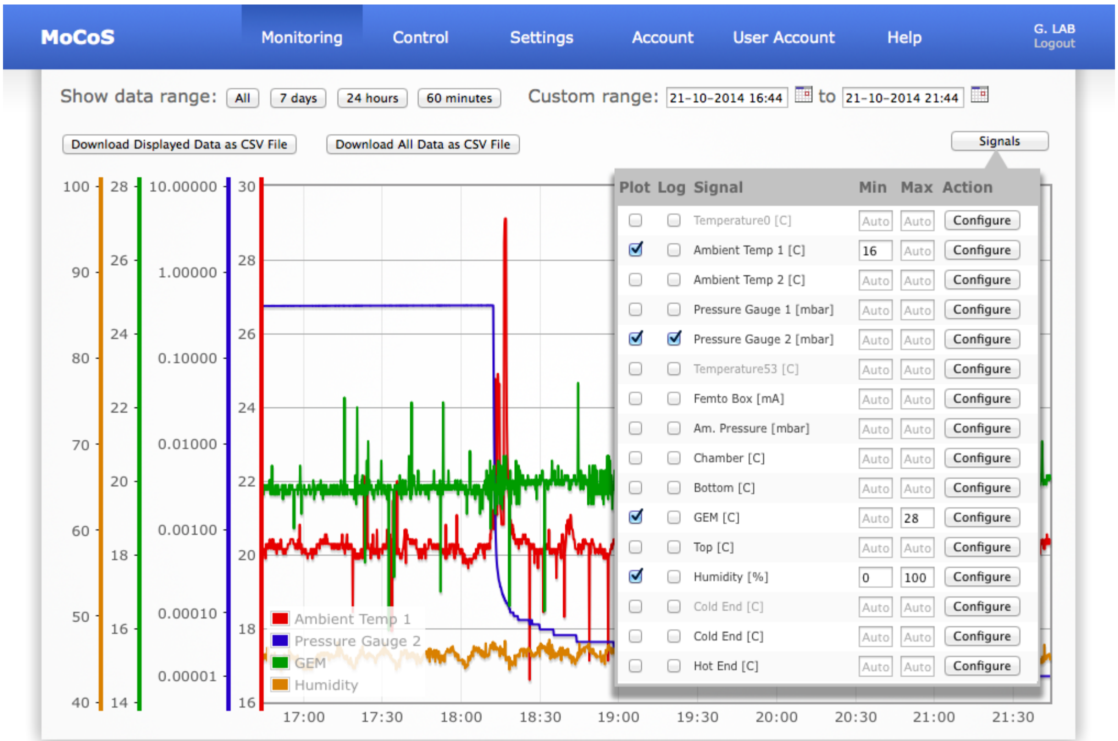
Figure 6: Trend graph: The acquired data can be displayed as a trend graph. The displayed time range can be selected using one of the options at the top of the page or by selecting start and end dates manually. Multiple signals can be plotted simultaneously and the displayed data can be downloaded as a CSV file.

quantity y from the raw voltage x. In contrast to all other conversion modes, the evaluation of custom expressions is done by the web service and not the MoCoS device. This means, that inputs using custom conversion expressions should not be used to control outputs.

When you have entered all configuration parameters, click "Apply". The new configuration will be saved and the MoCoS device will be notified that the configuration has changed. It will restart and download the new configuration upon startup.

4.2 Accessing acquired data

The MoCoS device will start acquiring and logging data as soon as the new configuration is downloaded and applied. To access the physical values obtained from sensors or standard signals, login to the web service, go to "Monitoring" and select the input channel you are interested (see figure 4). This will bring up a graph of the trend of values measured at this channel as shown in figure 6. The displayed graph can be customised to show only values recorded in a specific time range by selecting either one of the



Show data range: All | 7 days | 24 hours | 60 minutes

Custom range: 21-10-2014 16:44 to 21-10-2014 21:44

Plot	Log	Signal	Min	Max	Action
□	□	Temperature0 [C]	Auto	Auto	Configure
☑	□	Ambient Temp 1 [C]	16	Auto	Configure
□	□	Ambient Temp 2 [C]	Auto	Auto	Configure
□	□	Pressure Gauge 1 [mbar]	Auto	Auto	Configure
☑	☑	Pressure Gauge 2 [mbar]	Auto	Auto	Configure
□	□	Temperature53 [C]	Auto	Auto	Configure
□	□	Femto Box [mA]	Auto	Auto	Configure
□	□	Am. Pressure [mbar]	Auto	Auto	Configure
□	□	Chamber [C]	Auto	Auto	Configure
□	□	Bottom [C]	Auto	Auto	Configure
☑	□	GEM [C]	Auto	28	Configure
□	□	Top [C]	Auto	Auto	Configure
☑	□	Humidity [%]	0	100	Configure
□	□	Cold End [C]	Auto	Auto	Configure
□	□	Cold End [C]	Auto	Auto	Configure
□	□	Hot End [C]	Auto	Auto	Configure

Figure 7: Trend graph configuration: Clicking on the "Signals" button above the graph opens the configuration popover. There you can select which channels should be plotted and if they should be plotted logarithmically and specify minimum and maximum y-values for each axis.

predefined ranges or specifying one yourself. By clicking on "Signals", you can open the configuration popover shown in figure 7. There you can also specify minimum and maximum y-values to be used by the graph. This allows you to limit the region you are interested in both to a specific time range and a range of values. When entering minimum and maximum y-values, the changes will be applied as soon as you hit the return key. To compare the data acquire by multiple sensors, click on "Signals" and check all signals you would like to plot. In the same popover you can also specify which channels should be plotted logarithmically.

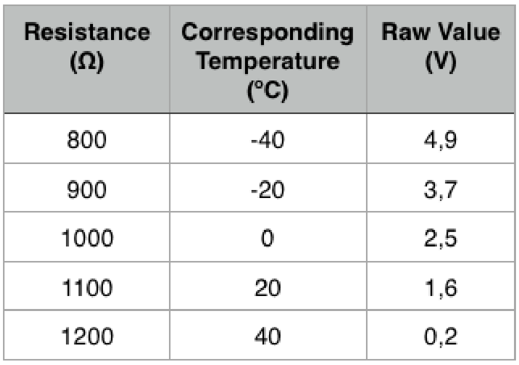
If you want to use the recorded data in another program or export it for other uses, you can click on "Download Displayed Data". This will download a comma-separated values (.csv) file to your computer containing the values of the selected sensors in the displayed time range. The data entries are listed in the file in the following format: UNIX timestamp, human readable time, recorded value, physical unit ID [value and unit ID for each of the 16 inputs]. You can also choose to "Download All Data" from the "Monitoring" page.

4.3 Calibration

Each sensor should be calibrated when it is first connected. This ensures that the values acquired are as accurate as possible and gives you the conversion parameters used to convert a raw signal (e.g. voltage) to a physical quantity. Please note that calibration is only necessary for sensors directly connected to the MoCoS device, whose raw values have to be related to a physical quantity by some kind of conversion expression. In case you are querying data from an external device through an RS232 serial interface, no calibration is required. Before starting the calibration process, please make sure your MoCoS device can connect to your local network and that you know its IP address. For more information on connecting the device to your network, please refer to section 9.2.

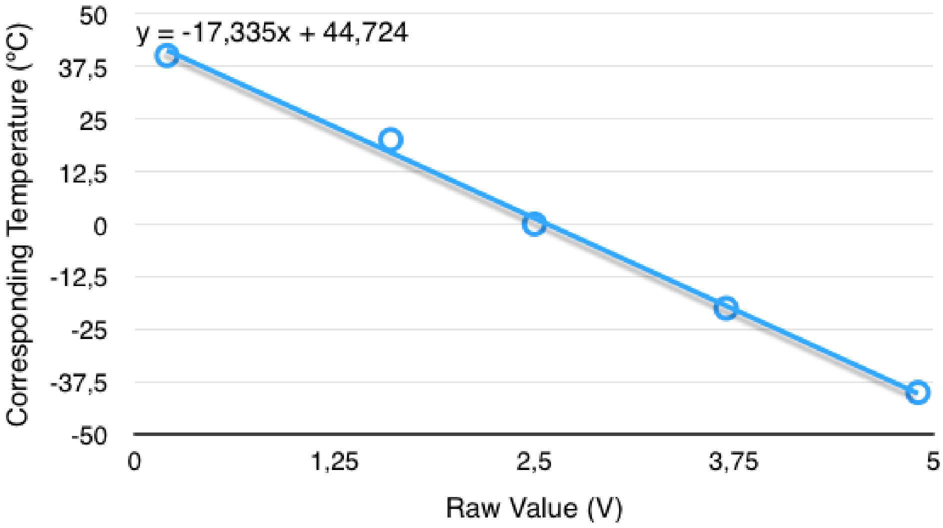
To calibrate a sensor, please connect it to one of the universal inputs of the MoCoS device and configure it by specifying the type of sensor on the web service. To do so, login to the web service, go to "Monitoring" and click on the configure button (gear icon) of the input your sensor is connected to. You can ignore all settings related to the signal conversion during the calibration process. Simply specify the type of sensor connected and enable it. Restart the MoCoS device and wait until it has finished downloading the updated configuration (right/bottom LED should turn green). Open a web browser and type the IP address of the MoCoS device into the address bar. The IP address will be the one specified during the setup process. If your device is connected to your local network and you entered the correct IP address, an HTML page will be displayed in your browser. This page lists configuration parameters, acquired data and the status of the control outputs. Scroll to the bottom of the page. The section "Raw Values - Calibration" displays the raw values acquired at each input channel. In most cases, these values refer to voltages measured by the microcontroller in your device. Make sure you can see the raw value of the channel your sensor is connected to. For the actual calibration you need reference samples. In the case of RTD temperature sensors these could be simple resistors. Connect a reference sample to the universal input instead of your sensor, refresh the page displayed in your browser and write down the raw value shown for the specific channel you are interested in. Also write down the physical quantity this reference sample corresponds to. If, for example, you are configuring a PT1000 temperature sensor and you connect a 950 Ohm resistor as a reference sample, your raw value may be 4.5 and it would correspond to a temperature of $-10\,^{\circ}$C. Repeat this process with multiple reference samples until you have enough data points to put a fit through the collected points. Use any spreadsheet or data analysis application to put a fit through the data points and obtain the conversion coefficients that relate the raw value to the physical quantity your are interested in.

As an example, the data points shown in figure 8a may be obtained when calibrating a PT1000 sensor. Plotting the corresponding temperature as a function of the raw value results in the graph shown in figure 8b. A linear fit through the data points yields the expression y(x) = -17.335x + 44.724. Therefore, the slope of the fit is -17.335 and the offset is +44.724. These are the values that you need to enter in the signal conversion section of the input configuration page. Once you have taken data points, put a fit through them and obtained the conversion parameters, go back to the "Monitoring"



Resistance (Ω)	Corresponding Temperature (°C)	Raw Value (V)
800	-40	4,9
900	-20	3,7
1000	0	2,5
1100	20	1,6
1200	40	0,2

(a) Calibration data points



(b) Calibration plot

Figure 8: Calibration example: The calibration of a PT1000 sensor with resistors corresponding to specific temperatures may yield the shown data points. The conversion parameters (slope and offset) can be obtained from a linear fit through these points.

page, click on the configure button of the input your sensor is connected to and enter the conversion parameters in the appropriate fields. You may specify different conversion modes such as linear, inverse or exponential depending on the response of your sensor and the fit through your data points. When you have completed the configuration, entered the conversion parameters and reconnected your sensor, you can restart your device and start monitoring the signal of the calibrated sensor.

Even if the conversion parameters could also be obtained from the data sheet of a sensor, it is recommended to follow the calibration procedure whenever possible to achieve the highest possible accuracy.

5 Control

MoCoS can also be used to control external devices such as heating or cooling controllers or switches. There are ten output terminals that can be used either as binary controls based on thresholds or PID controller outputs with continuous ranges of analogue output values. All outputs are voltage outputs and can provide voltage levels from 0V to 5V. Therefore, the outputs are compatible with TTL logic devices. Control outputs are assigned to one of the universal inputs. The physical quantity measured at the assigned input will be used to decide whether the output should be switched on or off for threshold-based outputs or to determine the output level for PID-based outputs. This section will describe how threshold-based and PID-based outputs work and how to set them up.

5.1 Thresholds

The simplest way to control an external device is through a TTL-based binary output. The voltage difference between the positive and the negative terminal of an output that is controlled via a threshold will be either 0V or 5V. This can be used directly to control logic-level-compliant MOSFETs or relays which in turn control other devices. The state of the output is determined by the physical quantity measured at the assigned universal input and the specified threshold. You can also specify two thresholds to have the control output in a specific state for a range of measured values. The state of the output is updated each time a measurement is taken. There are several types of thresholds that will switch the output to its ON state in different cases:

- On if below: The output will be in its ON (5V) state if the latest measured physical quantity is below the specified threshold value. Otherwise, the output will be in its OFF state (0V).
- On if above: The output will be in its ON (5V) state if the latest measured physical quantity is above the specified threshold value. Otherwise, the output will be in its OFF state (0V).
- On if in range: The output will be in its ON (5V) state if the latest measured physical quantity is above the first specified threshold and below the second threshold. Otherwise, the output will be in its OFF state (0V). Please note that the second threshold has to be larger than the first one.
- On if out of range: The output will be in its ON (5V) state if the latest measured physical quantity is below the first specified threshold or above the second threshold. Otherwise, the output will be in its OFF state (0V). Please note that the second threshold has to be larger than the first one.

You can assign multiple thresholds to the same universal input to control multiple external devices based on the same monitored value. Thresholds are useful for switching external devices on or off. To control devices capable of interpreting a continuous range of control voltages between 0V and 5V (e.g. MOSFETs, heating controllers), consider using a PID-based output.

5.2 PIDs

When using a binary state regulation device such as a heating controller that is either switched completely on or completely off, it is hard to reach and maintain an exact temperature. There will always be an overshoot above the threshold when heating up and an undershoot when cooling down and the temperature will at best oscillate around the desired setpoint. For this reason, using a control mechanism based on three errors between the current signal and the setpoint should be considered. Such control mechanism are named "PID controllers" since they determine the output level based on an error proportional to the difference between the current input value and the setpoint, the integral of past errors and an estimation of future errors based on the derivative of the error. MoCoS allows you have outputs which provide a voltage between 0V and 5V that is determined by a PID-logic controller. As for threshold-based outputs, the output value is determined by the latest measured value from an assigned universal input, but is then calculated from the current error, past errors and an estimation of future errors. To weigh the three types of errors and adapt the behaviour of the output's response to changes of the input value, you can specify three constants: a weighing factor for the error proportional to the current error between the measured physical quantity and the setpoint Kp, a weighing factor for the integrated past errors Ki and a weighing factor for the estimation of future errors obtained from the derivative of the error Kd. These three weighing factors will determine if the output is only slowly adapted to follow changes in the measured input value or if the output is changed rather "aggressively" when the input changes. There are two types of PID-based controllers that can be used to control the outputs of your MoCoS device:

- Analogue Output: PID-based outputs, which are configured as analogue outputs will provide a voltage signal between 0V and 5V in 255 steps, which is calculated from the physical quantities measured at the assigned input and the weighing factors. This corresponds to a resolution of the output signal of approximately 0.02V. The analog signal is generated by pulse width modulation (PWM). If you need a smooth DC output, you might want to consider a low pass filter between a PID-based MoCoS output and the controlled external device.

- Digital Output: PID-based outputs, which are configured as digital outputs will work in a way very similar to threshold-based outputs. They will also provide TTL-logic binary signals - either OFF (0V) or ON (5V). In contrast to threshold-based outputs, the output signal is not only changed when new measurements are taken. Between two measurements, the output will be a specific time in its ON state and the rest of the time in its OFF state, where the ratio between the time in ON state and the time in OFF state is determined by the PID-value calculated based on the

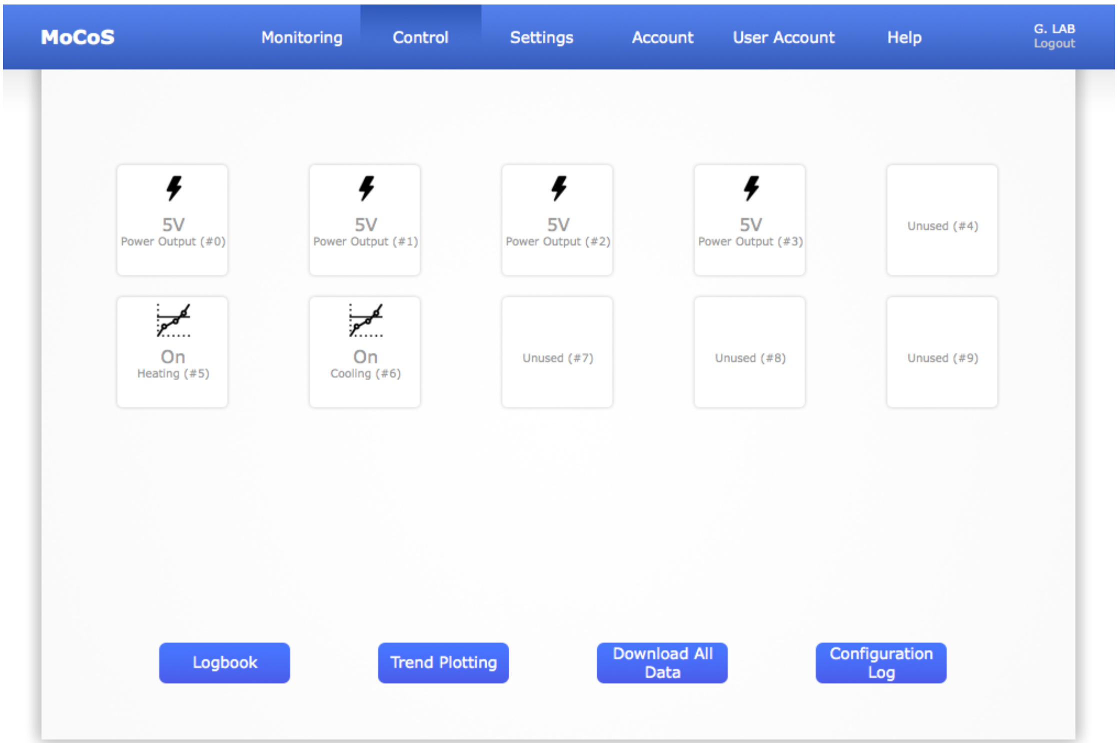


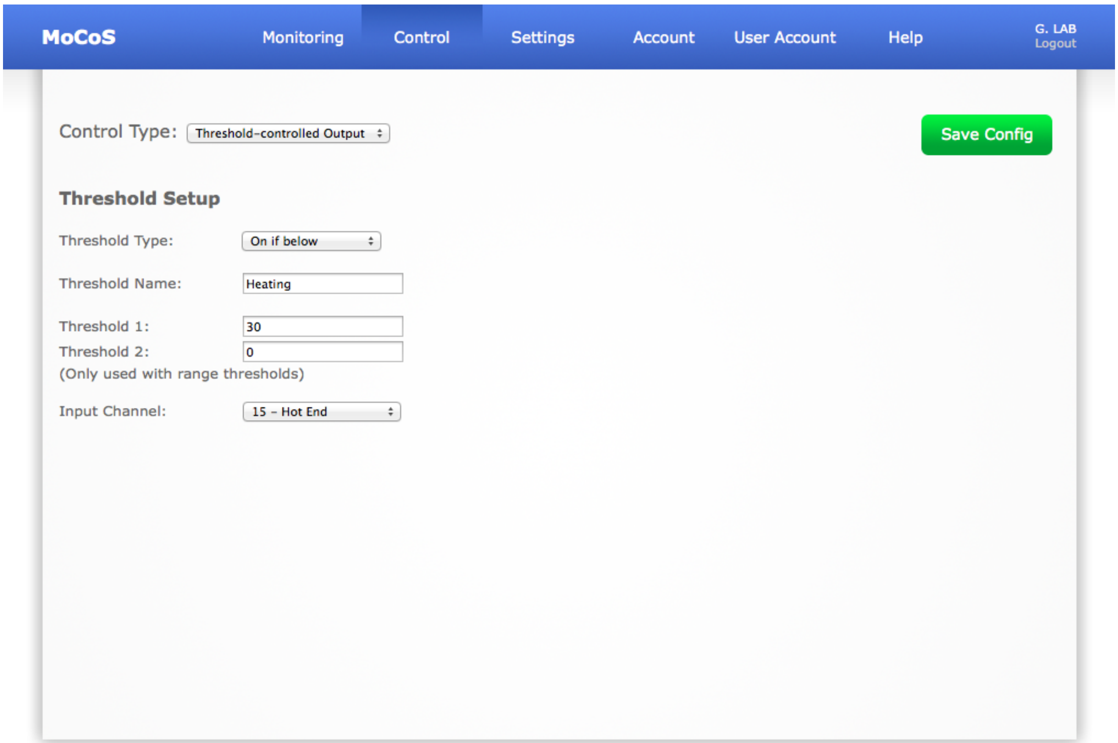
Figure 9: Web service control page: The "Control" page allows you to configure outputs and view their latest status.

latest measurement and the three weighing factors. For example, if the calculation of the output value would yield a value of 128, an analog output PID-based output would provide a voltage of 2.5V, while a digital output PID-based output would provide 5V for half the time of the measurement interval and 0V for the other half. If the calculated value would be, for example, 32, an analog output PID-based output would provide 0.625V, while an digital output PID-based output would switch the output to 5V for one eight of the measurement interval and to 0V for the rest of the time.

Even when using external devices that can only be switched fully on or fully off, you may want to consider to use a digital output PID-based output, since it may result in a much more stable behaviour of the controlled system.

5.3 Setting up threshold-based and PID-based outputs

You can configure both threshold-based and PID-based outputs through the MoCoS web service. To setup an output, go to the MoCoS website, login and select "Control". You will be presented with an overview of the ten control outputs and their current states



Control Type: Threshold-controlled Output

Threshold Setup

Threshold Type: On if below

Threshold Name: Heating

Threshold 1: 30
Threshold 2: 0
(Only used with range thresholds)

Input Channel: 15 – Hot End

Figure 10: Control configuration page: The configuration page for controls will be updated to match the output type selected. This screenshot shows the configuration form for threshold-based outputs.

as shown in figure 9. MoCoS features ten outputs which can be configured as either threshold-based or PID-based outputs or as power outputs to supply sensors with power. While each output can be configured to be a threshold-based output or a power output, only the first five output channels can be configured as PID-based outputs.

Move the cursor over one of the output channels and click the configure button (gear icon) to configure a specific output. You will be presented with the output configuration form, which allows you to specify the detailed behaviour of the output. First, select if you want this output to be used as a threshold-based output, a PID-based output or simply a power output.

5.3.1 Configuring a threshold-based output

When you select "Threshold-controlled Output" as the control type, you will be presented with the configuration form for threshold-based outputs as shown in figure 10. First, select the threshold type:

- On if below

- On if above
- On if in range
- On if out of range

The details of these four threshold types are described above. To enable a threshold, select one of the four types. You can also assign a name to a threshold. While the name is not used by the MoCoS device, it may make it easier to keep track of multiple thresholds on the MoCoS web service.

The thresholds themselves are specified using the fields "Threshold 1", "Threshold 2" and "Input Channel". First, select the input channel that will provide the measured physical quantity based on which you want to determine the state of the output you are configuring. Then, specify one threshold value if you are using "On if below" or "On if above" thresholds or both threshold values when using "On if in range" or "On if out of range" thresholds. The values of thresholds should be entered in the same unit as the one of the input channel's physical quantity. When you are using "On if below" or "On if above" thresholds, the value of "Threshold 2" will be ignored. Please note, that when using "Threshold 1" and "Threshold 2", the second threshold must always be larger than the first one. When you are done configuring the threshold-based output, click on "Apply" to save and upload the configuration to your MoCoS device. Please make sure that the external device you want to control does not draw more than 40mA as this may damage the microcontroller inside the MoCoS device. Your device will restart and the threshold-based output will be active after your device has restarted and downloaded the new configuration. The output pin of your MoCoS device marked with the number of the output you have configured will now change its voltage based on the value measured at the input channel and the thresholds specified.

5.3.2 Configuring a PID-based output

When you select "PID-controlled Output" as the control type, you will be presented with the configuration form for PID-based outputs. First, select the PID type:

- Analogue Output
- Digital Output

The details of these two PID types are described above. To enable a PID-based output, select one of the two types. To configure the behaviour of the selected PID-based output, first select the universal input channel that will provide the measured physical quantity based on which you want to determine the output level. Next, specify the level of the physical quantity measured by the selected input that you want to reach and maintain as the PID-controller's setpoint. The three weighing factors Kp, Ki and Kd determine how much the output level is affected when the input changes. You can find more details on the three weighing factors above. Specify these three constants to determine the dynamics of the PID-controller. You can also change the high and low voltage limits of the output. Please note, that these limits must always stay in the range of 0-5V. Please make sure that the external device you want to control does not draw more than 40mA as this may damage the microcontroller inside the MoCoS device.

When you are done configuring the PID-based output, click on "Apply" to save and upload the configuration to your MoCoS device. Your device will restart and the PID-based output will be active after your device has restarted and downloaded the new configuration. The output pin of your MoCoS device marked with the number of the output you have configured will now change its voltage based on the value measured at the input channel and the PID-control parameters specified.

5.3.3 Configuring a power output

When you select "Power Output" as the control type, the only value you have to specify is the voltage that should be provided at the output pin. You can specify any value from 0V to 5V. Please keep in mind, that the output channels should only be used to power sensors, which do not need a lot of power. The maximum current that can be supplied by a power output is 40mA. Please make sure any external device connected does not draw more than 40mA as this may damage the microcontroller inside the MoCoS device. Click on "Apply" to save and upload the configuration to your MoCoS device. Your device will restart and the selected power output will supply the specified voltage.

6 Accounts

MoCoS distinguished between accounts and user accounts. Each account is associated with a single MoCoS device and keeps track of its settings and the data it acquires. User accounts are associated with an account and can be used to login to the web service to view and edit all data pertaining to a specific device. There can be multiple user accounts associated with each account to allow several people to edit settings and view data collected by a single device. The following section will provide detailed instructions on how to create an account, link an account to a MoCoS device and how to create, modify and manage user accounts.

6.1 Creating a new account

Before you can start collecting data and controlling external devices, you have to create a new account with at least one associated user account and link it to your device. To create a new account, go to mocos.web.cern.ch and click on "Create new account". On the following page, you will be asked for an account name and an email address associated with the account. Please note that this is not your personal user account but the account associated with your MoCoS device. Fill in the required data and click on "Next". The next step is the creation of a new user account. Please fill in your first and last name and your personal email address and choose a username and password for your user account. Clicking on "Next" will display a summary of the data you provided. Please check all fields carefully and click on "Done" to confirm the data. Your account and user account will now be created. Please note that they are not activated yet. Before you can login to the MoCoS web service, you need to activate your new account. After confirming the entered data, you will be sent an email with an activation link. Open this link in a web browser or simply click on it to activate your account. Once your account has been activated, you will see the message "Account has been activated". You are now able to login to the MoCoS web service and start setting up your device.

6.2 Linking a device to an account

To make sure your device can only be configured from your account and only you have access to the data it collects, you have to link your MoCoS device to your newly created account. To link a device to your account, you will need the MoCoS desktop app and the account username and password from the web service.

First, locate the account username and password for your account. Please note, that this is not the same as your user account's login information. The account's username and password are automatically generated upon account creation and are used by your MoCoS device to authenticate with the web service. You can find your account's username and password in the "Account" section of the web service. Login to the web service, select the "Account" section and the information you need will be listed at the top of the page.

To link your device to your account, connect your MoCoS device to your computer with a USB cable and open the MoCoS desktop app. If you have not yet downloaded the MoCoS desktop app, please refer to section 9 for more information. Once you have connected your device to the computer and opened the desktop app, select the port the MoCoS device is connected to and click "Connect". If you have problems identifying the correct port, please refer to section 9.1. When you are successfully connected to the device, a configuration screen will be displayed. Please enter the account's username and password obtained in the previous step in the appropriate fields and click "Apply". The authentication information will be sent to and stored by your device. Your device is now linked to your account. You can close the desktop app, disconnect your device and restart it and login to the web service to configure the operation of your device or view the data it has acquired.

7 Software

The monitoring and control features of MoCoS are built on interconnected applications on the MoCoS device and an associated web server. The MoCoS device is based on the microcontroller platform Arduino and uses the Arduino software environment to control the various data acquisition, signal conditioning and control features. The web service is programmed using PHP and keeps track of the configuration and acquired data using a MySQL database. Additionally, a Processing desktop application running as a Java application is used The following section will briefly explain the main structure of the applications, their usage and how they interact with each other. This is a rough overview rather than a detailed technical discussion - code-level documentation is provided as comments in the respective source-code files.

7.1 MoCoS device software

The data acquisition and logging and all control mechanisms such as thresholds and PID controller-based outputs are handled by an ATmega2560 microcontroller on an Arduino Mega 2560 board. It is programmed through the Arduino IDE and uploaded to the chip as a compiled sketch. The source-code of the sketch can be found in the "Help" section of the web service. The basic structure follows that of all Arduino applications: It starts with the inclusion of the required libraries followed by the declaration and definition of global variables. Values related to inputs or outputs are stored as arrays and are assigned placeholder values during definition that will be later replaced by values stored on the SD card or downloaded from the web service. The program starts in the setup() loop, where input and output pins are assigned, communication with serial devices, Ethernet devices and the SD card is initialised, the last configuration is restored from the SD card and the most current configuration is downloaded from the web service. As soon as all steps in the setup() loop are completed, the program enters the loop() loop. This loop is repeated over and over again as long as the device is powered on. Inside this loop, the device constantly check whether there is an incoming Ethernet connection and displays a web page showing the latest acquired values and the configuration if there is one. Most importantly, however, the loop() loop contains the actual data acquisition, data logging and control function calls. Measurements are performed in the "Measurement interval" specified on the web service and control outputs are adjusted whenever measurements are taken. In intervals given by "Logging interval", the data is sent to the web service via the Ethernet connection.

There are several files containing specialised functions, which make up the MoCoS device software:

- "Parsing" contains code used to handle incoming, structured data. Such data can be read from the SD card or downloaded from the web service. The functions in "Parsing" will assign the data to the respective channels and store them in the appropriate variables or arrays.

- "Serial Connection" handles the communication of the MoCoS device software with the desktop application.
- "WebServer" checks for incoming HTTP requests and returns HTML code displaying the current status of the device and its configuration.
- "auxiliaryFunction" contains mostly functions used to convert data into the needed formats and code to display the status of the device using the 3 LEDs.
- "control" takes care of adjusting threshold-based and PID-controller based outputs according to the latest measured values.
- "dataLogging" contains code to upload the acquired data to the web service or send it to other destinations.
- "downloadConfiguration" features a collection of functions to download all required data from the web service and passes them to the appropriate parsing functions.
- "measuring" looks through all enabled universal inputs, routes the signals to the appropriate signal conditioning circuits and takes the data from the sensors. Additionally, it also contains code to query data from external devices via serial RS232 communication.
- "sdFunctions" contains all functions pertaining to reading from and writing data to the SD card.
- "signalConversion" is mainly used when measurements are taken to convert the raw voltage or other incoming signal from the sensor to a physical quantity that will be logged.

The Arduino sketch running on the MoCoS device can be downloaded from the "Help" section of the web service.

7.2 Web service software

The web service is based on HTML pages styled with CSS and made interactive by Javascript. The pages are prepared by PHP on the server before being sent to the client's browser. PHP also takes care of querying data from and writing data to the MySQL database used by MoCoS. A SQL dump file of the database can be downloaded from the "Help" section. The pages and scripts making up the web service are saved in the root directory of the web space of the MoCoS web service. There are a number of folders containing files for different kinds of actions:

- "data" contains image files used for the visual presentation of the web service
- "includes" houses files that are included by other files. These include scripts for connecting to the MySQL database, for checking the authentication status of the current session and a functions file containing auxiliary functions used by several other scripts.

- "pages" contains the web pages of the web service, that can be accessed after logging in.

- "scripts" contains pure PHP scripts that check, save or update data based on actions triggered by the web service. The "scripts" folder has two important subfolders: "getConfiguration" contains scripts that return the configuration for MoCoS' inputs and outputs and all other parameters used by the system in a formatted manner. These scripts are queried by the device and used to transfer configuration information from the web service to the MoCoS device. The other subfolder, "logging", contains code to receive data sent by the device and write it to the database.

- "style" contains the CSS style sheets, which define the visual appearance of the pages of the web service.

- In the root directory there are a number of additional HTML files, which make up the public area of the MoCoS web site consisting of the login page and pages for creating new accounts.

The web service software can be downloaded from the server hosting the service at mocos.web.cern.ch.

7.3 Web service database

A MySQL database hosted by the CERN Database on Demand service is used to store all acquired data and keep track of the configurations and settings specified. The database consist of a number of tables, which will be explained here:

- "accounts" stores all accounts that are used to link all configuration data and acquired values coming from one device together

- "channelAssignment" stores the types of sensors connected to the inputs and information about the logging of data from the enabled channels

- "channelTypes" contains all supported types of sensors that can be connected to the universal inputs

- "configurationChangeLog" stores records of any changes to the configuration of input or output channels

- "controlStatus" contains entries of the statuses of all control outputs reported by the MoCoS device

- "data" contains entries of all acquired measurement reported by the MoCoS device

- "logbook" stores messages about system changes that are displayed in trend graphs

- "pidControl" contains the configuration of PID-based outputs

- "powerOutputs" contains the configuration of power outputs
- "sensorConfigurations" can store input configurations so they can be reused
- "serialCommunication" stores settings pertaining to the RS232 communication protocol used when querying data from external devices
- "settings" contains general settings such as measurement and logging intervals
- "signalIcons" contains a list of the different signal icons that are displayed on the landing page of the "Monitoring" section
- "thresholds" stores the configuration of threshold-based outputs
- "unitConversion" can be used to convert signals from one unit to another
- "units" contains all available physical units
- "users" stores all user accounts, which are used to login to the web service

A SQL-dump of the database containing SQL statements to create the structure and essential content of the database can be downloaded from the "Help" section of the web service.

7.4 Processing desktop application

The MoCoS desktop application is used to set important communication parameters, that enable the MoCoS device to connect to the local network and the web service. These include the MAC and IP addresses, the URLs of the configuration and logging servers and the login information used to authenticate with the web service. Detailed information about the usage and features of the desktop application can be found in section 9. The desktop application was developed as a Processing application and exported as Java applications for Windows, Mac OS X and Linux. It can be downloaded from the "Help" section of the web service.

Upon startup, the USB port that the Arduino is connected to, has to be selected. Once connected, one can specify the communication parameters mentioned above. Each value is transferred to the device individually when "Apply" is clicked and stored on its SD card. Additionally, the desktop application receives the most recently acquired data from the device and displays it at the bottom of the window.

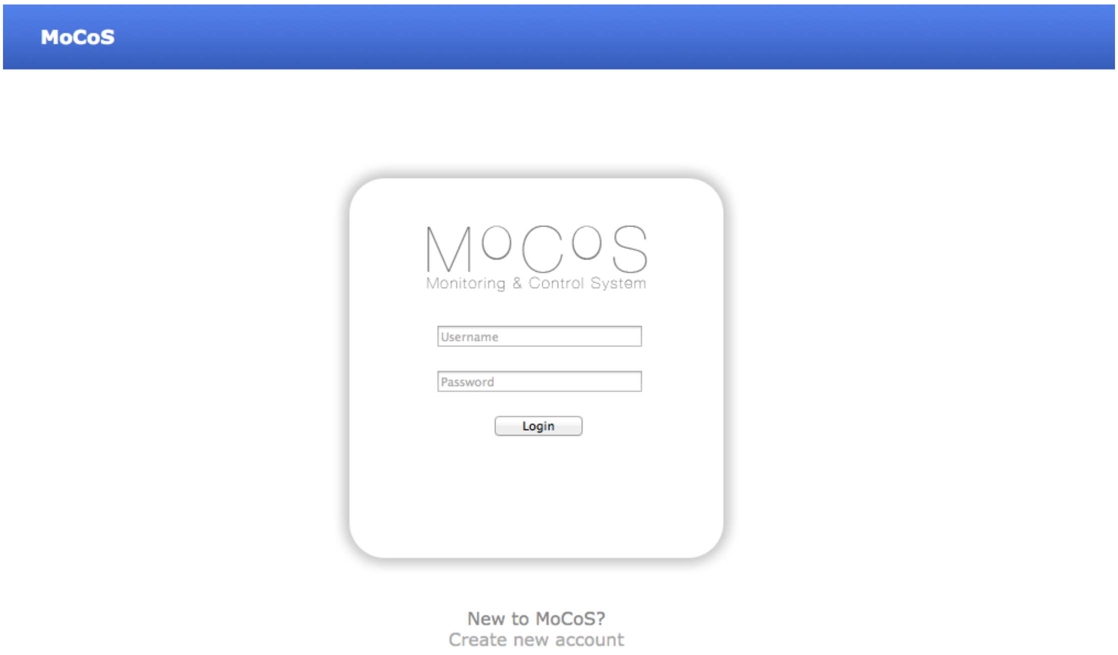


Figure 11: Web service login page: The MoCoS web service can be reached at mocos.web.cern.ch

8 Web Service

The MoCoS web service stores all acquired data and allows you to configure the MoCoS device. It also makes the data collected from the sensors available online, as trend graphs and as downloadable files. This section will briefly describe the main sections of the web service and focus on features not described elsewhere.

8.1 Accessing the web service

The MoCoS web service can be accessed at mocos.web.cern.ch. The login page of the web service is shown in figure 11. You need to create a MoCoS account to login to the web service. Please refer to section 6 for more information on creating an account and linking it to your device.

The different section of the web service can be accessed by the navigation bar at the top of the page.

8.2 "Monitoring" section

The landing page of the "Monitoring" section of the web service provides you with an overview of the latest values acquired from the connected sensors. When moving the cursor over one of the inputs, a configuration button (gear icon) is shown, that allows you to configure a specific universal input. Clicking on any of the inputs will display a trend graph of the values measured by this sensor over time. You can also plot multiple signals simultaneously to observer their dependence on each other. You can access other features through the first page of the "Monitoring" section. At the bottom of the page, there are four buttons labeled "Logbook", "Trend Plotting", "Download All Data" and "Configuration Log". For more information about the "Logbook", please refer to section 8.8. "Trend Plotting" takes you to the trend graph mentioned above. "Download All Data" allows you to download a CSV file containing all data acquired by your MoCoS device so far. Finally, "Configuration Log" lists any changes in the configuration of the universal inputs or outputs. This allows you to look up changes that may affect measurements at any time and possibly link strange signal behaviours to configuration changes. For more information about the monitoring capabilities of MoCoS, please refer to section 4.

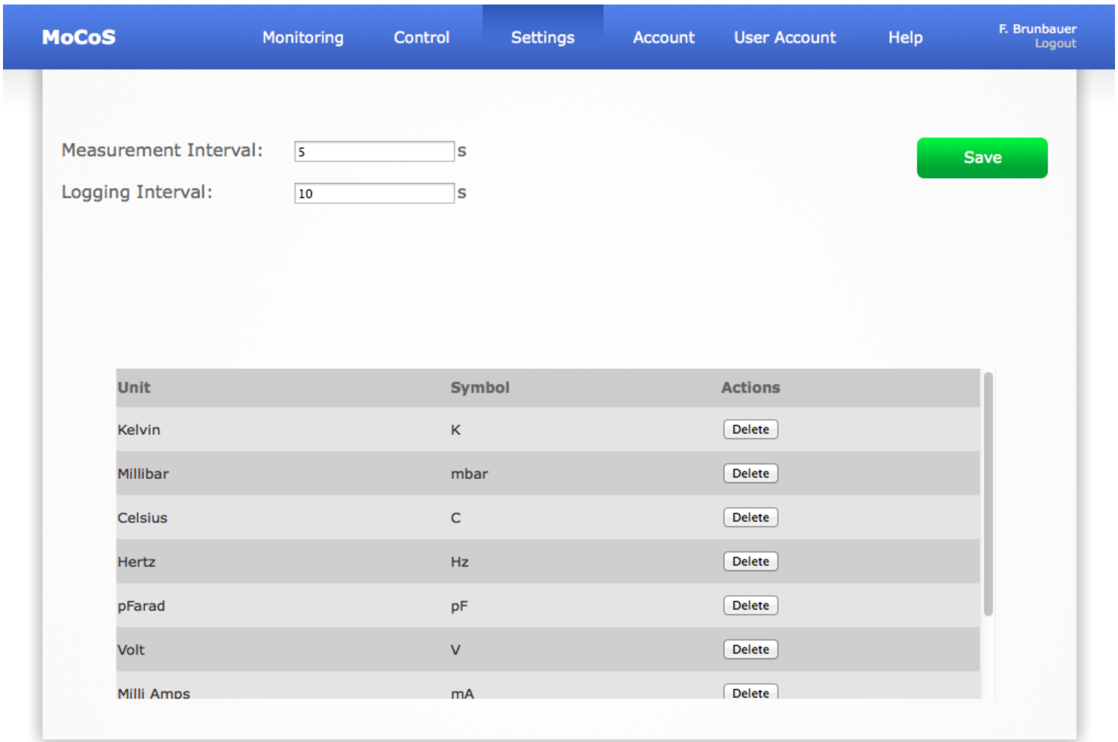
8.3 "Control" section

The landing page of the "Control" section provides you with an overview of the statuses of the control and power outputs of your device as shown in figure 9. Moving the cursor over an output displays a configuration button (gear) that takes to you the setup page for a specific output. The four buttons on the bottom of the page are the same as for the "Monitoring" section. For more information about the control capabilities of MoCoS, please refer to section 5.

8.4 "Settings" section

The "Settings" section contains general settings not pertaining to any specific universal input or output of the MoCoS device as shown in figure 12. Specifically, you can set the measurement and logging intervals. The "Measurement Interval" determines how often measurements are taken and how often the control outputs based on thresholds or PID-based controllers are adjusted. The "Logging Interval" determines the time between separate calls to the web service to upload the acquired data. It is recommended to specify a logging interval that is longer than the measurement interval. Additionally, the logging interval should be an integer multiple of the measurement interval. The defaults are 5s measurement and 10s logging intervals.

In the "Settings" section you can also manage the physical units used by the web service. Physical units can be assigned to input channels during configuration and will be displayed on the overview of acquired signals and logged with the measurements. To add new physical units or delete existing ones, please use the table at the bottom

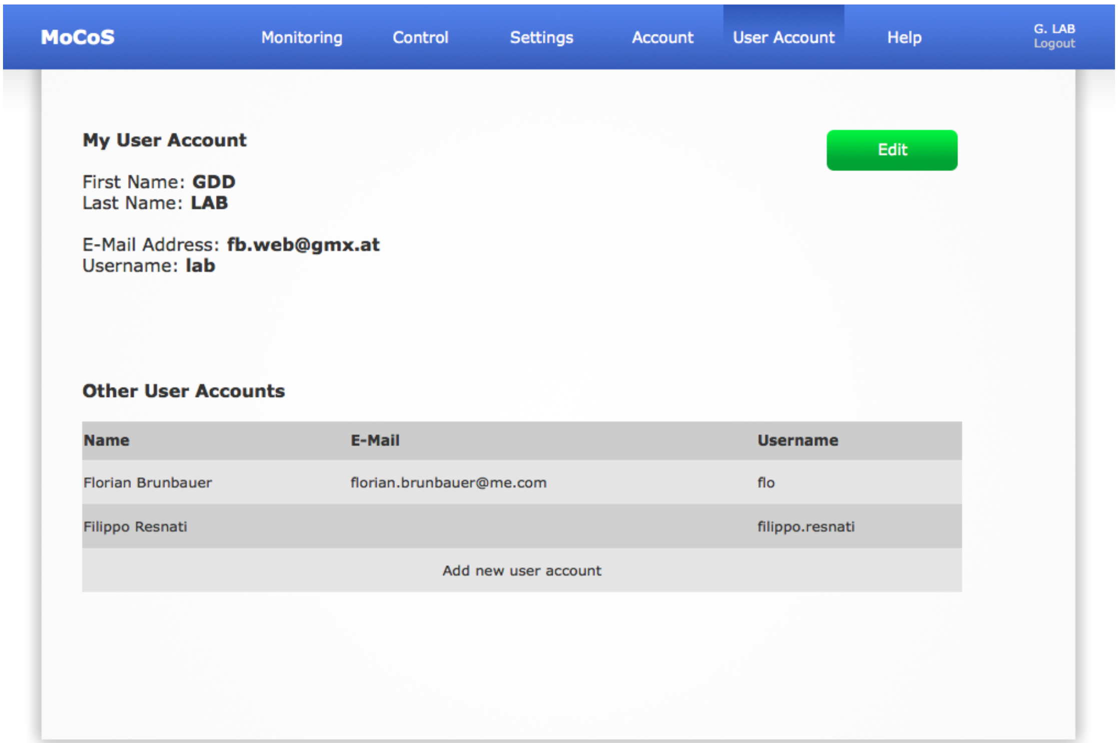


Measurement Interval: 5 s

Logging Interval: 10 s

Unit	Symbol	Actions
Kelvin	K	Delete
Millibar	mbar	Delete
Celsius	C	Delete
Hertz	Hz	Delete
pFarad	pF	Delete
Volt	V	Delete
Milli Amps	mA	Delete

Figure 12: Web service settings page: The "Settings" page allows you to specify measurement and logging intervals and manage physical units



My User Account

First Name: **GDD**
Last Name: **LAB**

E-Mail Address: **fb.web@gmx.at**
Username: **lab**

Other User Accounts

Name	E-Mail	Username
Florian Brunbauer	florian.brunbauer@me.com	flo
Filippo Resnati		filippo.resnati
Add new user account		

Figure 13: Web service user account page: The "User Account" page displays information about your user account and about other user accounts linked to the same account and device.

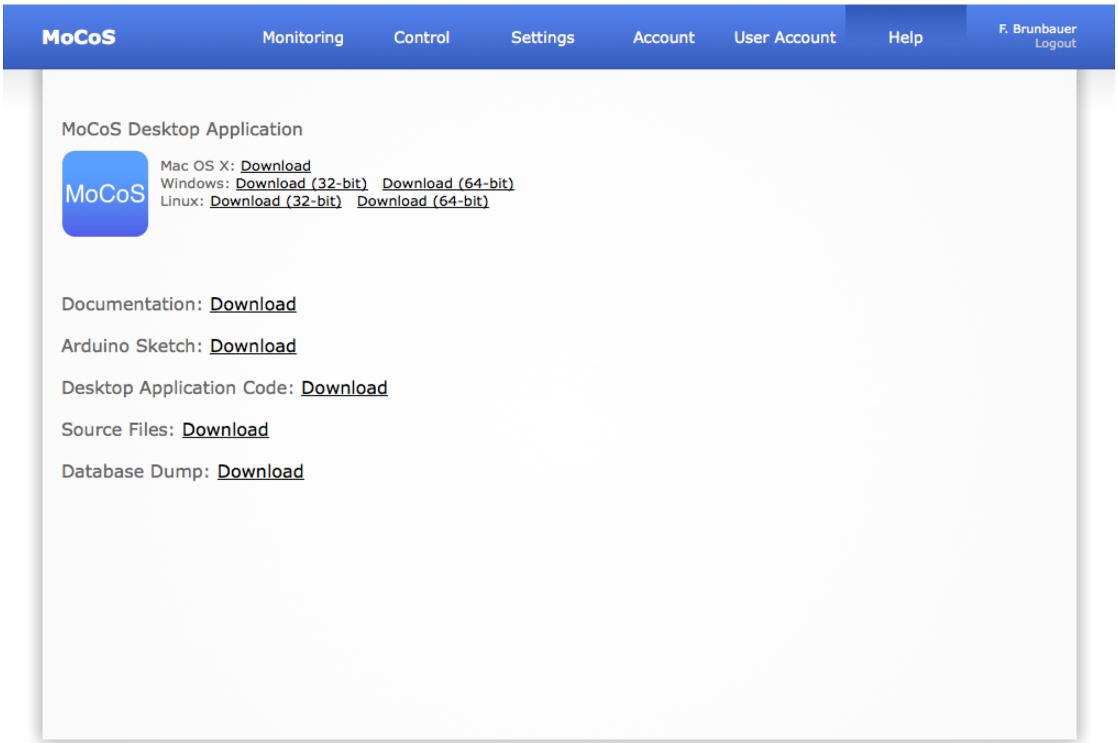
of the "Settings" page. Each physical unit has a name and a symbol. For example, a temperature unit may be called "Celsius" and have the symbol °C.

8.5 "Account" section

The "Account" section of the web service displays the information that should be used by the MoCoS device to authenticate with the web service when downloading its configuration or when uploading acquired data. To upload the account username and password, please refer to section 9.2.

8.6 "User Account" section

The "User Account" section of the web service displays your account information and allows you add additional user accounts as shown in figure 13. Multiple user accounts can be used to give other people access to the data acquired by your MoCoS device. This section also allows you to change your user account information and change your



MoCoS Desktop Application

Mac OS X: Download
Windows: Download (32-bit) Download (64-bit)
Linux: Download (32-bit) Download (64-bit)

Documentation: Download

Arduino Sketch: Download

Desktop Application Code: Download

Source Files: Download

Database Dump: Download

Figure 14: Web service help page: The "Help" section contains download links to files and documentation.

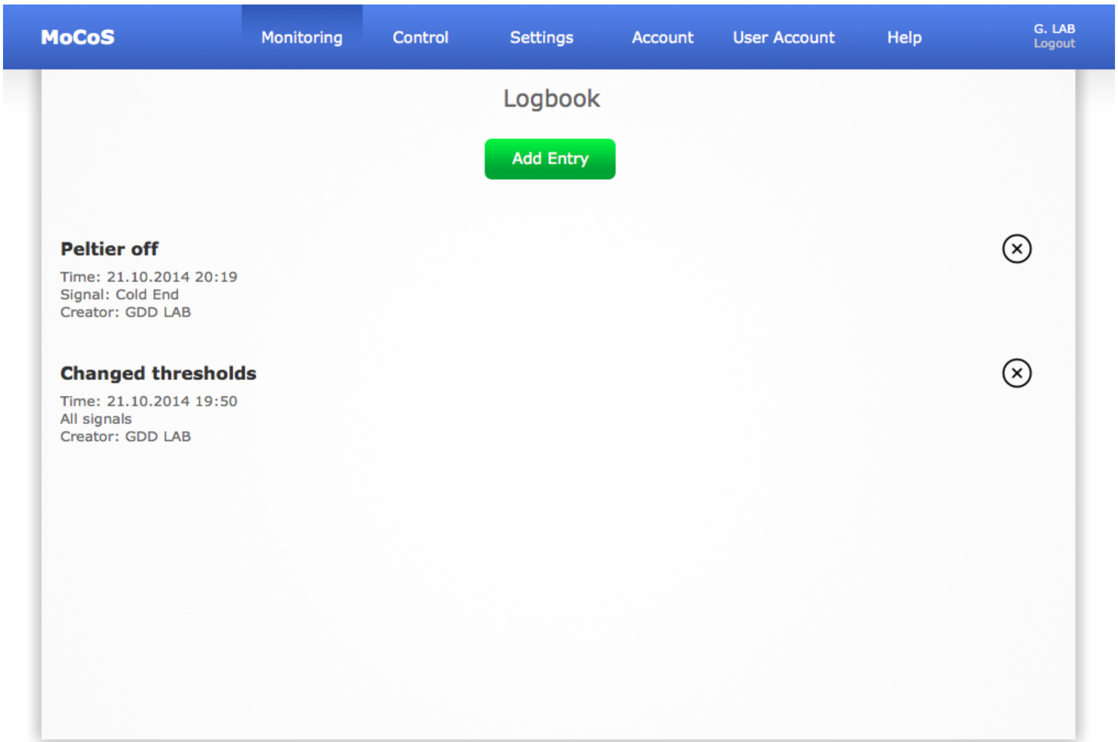
password. For more information about the system of accounts and user accounts used by MoCoS, please refer to section 6.

8.7 "Help" section

The "Help" section (shown in figure 14) provides links to download the MoCoS desktop application and the source files of the software and hardware that makes up the Monitoring & Control System. Additionally, this documentation is available as a downloadable PDF file.

8.8 Logbook

The web service also features a logbook, that allows you to keep track of any changes made to the system you are monitoring. Any changes that may affect measurements should be entered into the logbook, which can be accessed from both the "Monitoring" and "Control" sections of the web service. For example, a gas circulation system that works by convection would require different parts of the system to be at different temperatures. Changes to the system, such as changing the position of the cooling fan or



Logbook

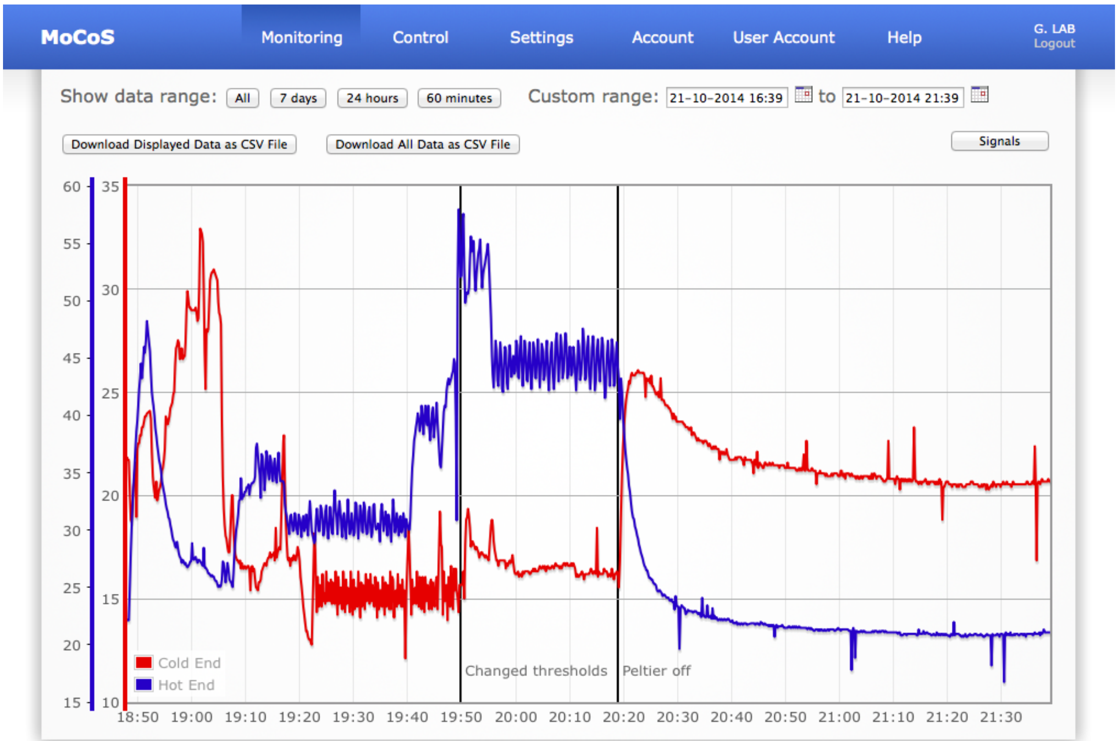
Peltier off

Time: 21.10.2014 20:19
Signal: Cold End
Creator: GDD LAB

Changed thresholds

Time: 21.10.2014 19:50
All signals
Creator: GDD LAB

Figure 15: Logbook: You can keep track of changes to the monitored system by saving them in the logbook, which is part of the web service.

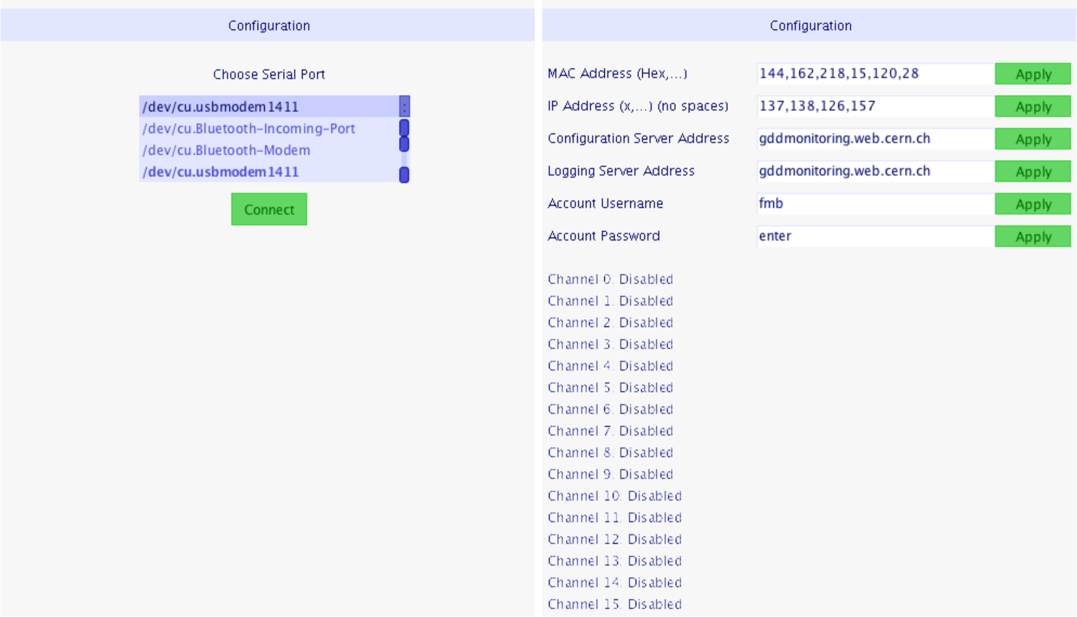


Custom range: 21-10-2014 16:39 to 21-10-2014 21:39

Figure 16: Logbook entries in trend graph: Entries in the logbook will be displayed on trend graphs.

switching off the Peltier elements that control the temperature can be entered into the logbook, as shown in figure 15.

Entries saved to the logbook can pertain to a specific input channel or to all of the channels. When viewing trend graphs, logbook entries will be displayed as vertical, black lines with labels (shown in figure 16), to make it easier to identify the effects of the logged events on the monitored quantities.



/dev/cu.usbmodem1411
/dev/cu.Bluetooth-Incoming-Port
/dev/cu.Bluetooth-Modem
/dev/cu.usbmodem1411

MAC Address (Hex,...)	144,162,218,15,120,28
IP Address (x,...) (no spaces)	137,138,126,157
Configuration Server Address	gddmonitoring.web.cern.ch
Logging Server Address	gddmonitoring.web.cern.ch
Account Username	fmb
Account Password	enter

Channel 0: Disabled
Channel 1: Disabled
Channel 2: Disabled
Channel 3: Disabled
Channel 4: Disabled
Channel 5: Disabled
Channel 6: Disabled
Channel 7: Disabled
Channel 8: Disabled
Channel 9: Disabled
Channel 10: Disabled
Channel 11: Disabled
Channel 12: Disabled
Channel 13: Disabled
Channel 14: Disabled
Channel 15: Disabled

(a) Port selection

(b) Specifying parameters

Figure 17: Desktop application: The MoCoS desktop application allows you to connect to your device by selecting the port of your computer it is connected to and specify connection parameters used to connect to your local network and authenticate with the web service.

9 Desktop Application

The MoCoS desktop application is necessary to configure basic connection settings used by your device. Before your device is connected to your local network and associated with your account, you will not be able to communicate with it through the web service. Therefore, the MoCoS desktop app is used to configure how your device connects to your local network and to associate it with your account.

If you have not already done so, you can download the desktop application from the "Help" section of the MoCoS web service. Login to your account, go to the "Help" section and information about the MoCoS desktop app will be displayed at the top of the page. The application is available as native binaries for Mac OS X, Windows and Linux. Please choose the appropriate version for your operating system and click on "Download". To run the MoCoS desktop app, you will need a Java runtime environment.

9.1 Communicating with your device

Once you have downloaded the application, you can connect your device to your computer with a USB cable and open the MoCoS desktop app. To connect to your device, you will need to select the port of your computer, which your device is connected to as shown in figure 17a. Select the port from the drop-down list and click "Connect". If you are not sure which port your MoCoS device is connected to, open the drop-down list and look the the available options - then close the desktop app, disconnect your device and open the application again. When you open the drop-down list now, one or two ports will have disappeared. These are the ports your device was using. Close the application, reconnect your device and open the application again, select the port identified in the previous step and click "Connect".

9.2 Configuring network parameters

When the desktop application has successfully connected to your MoCoS device, the configuration screen shown in figure 17b will be displayed. The configuration and logging server addresses should remain at the default values, unless you have been instructed otherwise. The fields "MAC Address" and "IP Address" have to be adjusted to fit your device and the network you are trying to connect to. You can find the unique MAC address of your device on the bottom of the Ethernet module of your device. Please enter the MAC address as six integers separated by commas in the field "MAC address". The IP address you should use depends on the local network the device should connect to. If you are using DHCP to assign IP addresses on your local network, choose any valid address and enter it in the field "IP Address" as four integers separated by commas. If your device should use a specifically assigned IP address, please enter it in the field "IP Address". The username and password entered in the corresponding fields are used by your MoCoS device to identify itself to the web service and link your device to your web service account. Enter the account username and password, which you can find in the "Account" section of the web service. Please note, that this is not the login information you use to login to your user account but the automatically generated authorisation information linking your device to your account. After entering each one of the required parameters, please click on "Apply" to upload the new value to your device. The network and authorisation settings will be transferred to and stored on your device. When your device is restarted the next time, the new settings will be used to connect to your local network and authenticate with the web service. You can now close the desktop app, disconnect your device from your computer and restart it by disconnecting and reconnecting the power supply of your device.

10 Hardware

The MoCoS device consists of an Arduino Mega 2560 microcontroller board, an Arduino Ethernet Shield, a custom signal conditioning PCB and a front panel or box depending on whether you are using a portable device or the crate version of the device. This section will describe the assembly of a MoCoS device and the hardware used.

10.1 Signal conditioning PCB

The signal conditioning PCB is the main part of the MoCoS device besides the Arduino Mega 2560 microcontroller board and the Ethernet Shield. It takes care of preparing different signals coming from sensors connected to the universal inputs for the ADCs of the microcontroller. Specifically, it contains several circuits that translate resistances, higher voltages or currents to voltages in the range of 0-5V, which can be read by the ADCs. The detailed layout of the signal conditioning circuits is shown in figure 24.

To allow any type of supported sensor to be connected to any universal input, analogue multiplexers select the signal that should be conditioned and acquired during the measurement process and route it to the appropriate conditioning circuit. This is achieved by two layers of multiplexers, whose common output/input terminals are connected. The first layer acts as a 16:1 multiplexer and selects the signal that should be measured, while the second layer acts as a 1:8 demultiplexer that routes the signal to the circuit used to condition signals of the specific type of sensor connected together. The conditioning circuits themselves consist of resistors, capacitors and operational amplifiers. The conditioned signals are read out either by the ADCs of the microcontroller or connected to interrupt pins (for frequency and capacitance measurements), where the rate of pulses is measured.

The signal conditioning PCB can be produced from the board design files described in the next section and assembled as described in section 10.3.

10.2 Ordering and producing parts

The Arduino Mega 2560 board and the Arduino Ethernet Shield can be readily acquired commercially. The custom signal conditioning PCB can be manufactured by a PCB fabrication house. The necessary Altium Designer and Gerber files can be found in the "Help" section of the web service.

If you want to assemble a portable version of the MoCoS device, a box housing the electronics should be 3D printed. The 3D model can be found in the "Help" section of the web service. If you want to build a Eurocrate version of the MoCoS device, a front panel has to be manufactured. The 3D drawing of the front panel can also be downloaded from the "Help" section of the web service.

The electronic components required for the signal conditioning PCB should be ordered from Farnell, Radiospares and Adafruit. All required parts and order numbers are listed in appendix B.

10.3 Assembling the signal conditioning PCB

If you are assembling a portable device, please cut the PCB at the lines shown in figure 18 before mounting any components. Use only cutters made specifically for PCB cutting, since improper handling may damage the signal conditioning circuits.

Once you have bought all parts and produced the signal condition PCB, you can mount the components. While all footprints are large enough to be solder manually, please pay attention to the mounting direction of the parts. Specifically, the operational amplifiers and the MAX232 RS232 transceiver are mounted upside-down compared to the multiplexers. Depending on whether you are assembling a portable device or a crate version, you will have to mount different terminal block headers. For the portable version, mount 3.5mm pitch headers at the top side of the PCB on the left and right sides of the board after cutting it. For the crate version, mount 5mm pitch headers on the top and bottom sides of the board at the front of the PCB.

After mounting all components, perform an electrical check using a multimeter and make sure all connections are as expected. Most importantly, check the resistance between any two points on the 5V and GND nets and make sure there are no unintentional shorts. Typically, this resistance is approximately 2 kΩ.

The assembled signal conditioning PCB for the portable device is shown in figure 19.

10.4 Assembling the portable device

When assembling a portable device, first glue M3 nuts into the gaps at the bottom and at the back of the 3D printed case. Then, mount the small fan in the hole at the left side of the bottom part of the case. Next, put the Arduino Mega 2560 board into the bottom part of the case and fix it with M3 screws. Once it is mounted, put the assembled signal conditioning PCB onto it and push it down carefully to connect the pins of the circuit board to the headers of the Arduino board. Check the connections and plug the fan into the DC power terminals labeled "5V" and "GND" on the custom PCB. Finally, plug the Ethernet Shield into the signal conditioning PCB, put the top part of the case on and connect the two parts of the case with two screws at the back of the device.

10.5 Assembling the crate version

When assembling the crate version, first plug the Arduino Mega 2560 board into the bottom of the signal conditioning PCB and plug the Ethernet Shield into the top of the PCB. Then, fix the PCB to the front panel using the two mounting holes at the top and bottom of the front side of the circuit board and the PCB fixtures supplied with the Eurocrate front panel.

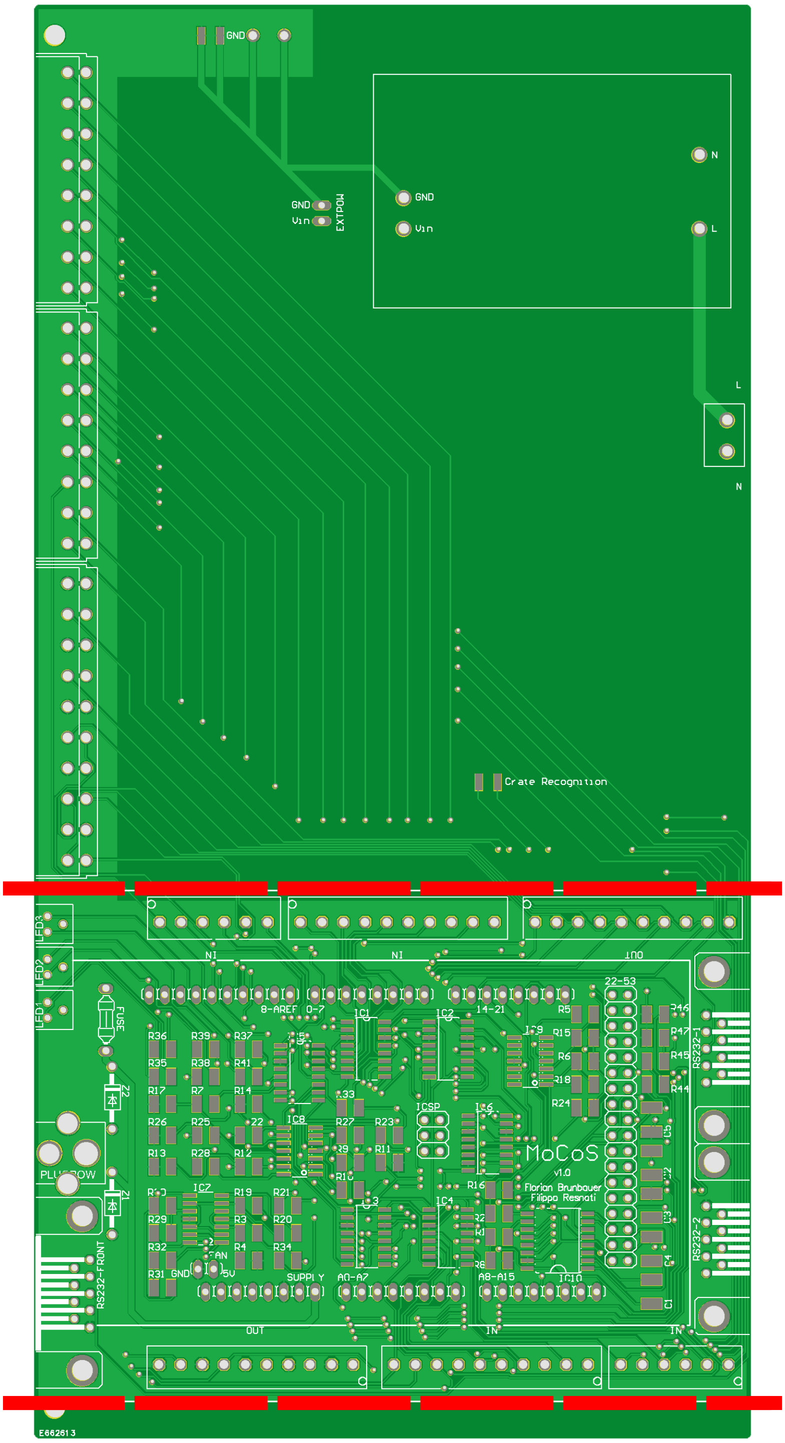


Figure 18: Cut PCB for portable version: When assembling a portable MoCoS device, please cut the signal conditioning PCB at the red lines.

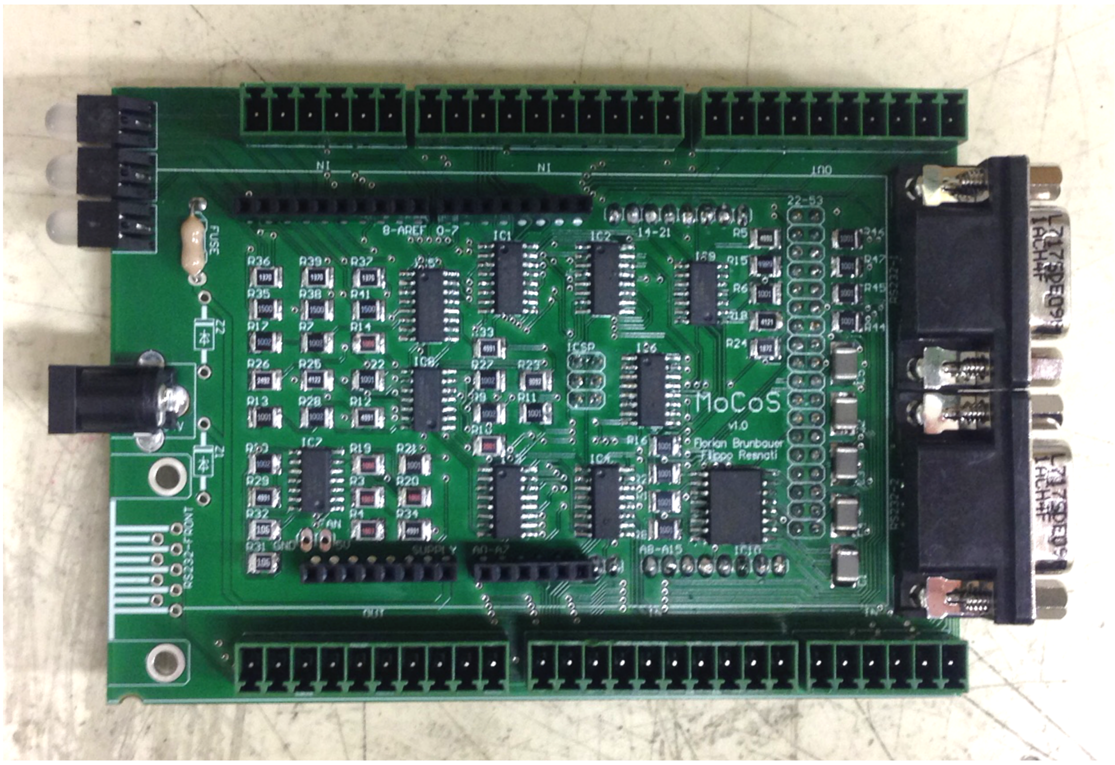


Figure 19: Assembled PCB: Signal conditioning board of the portable device with components and connector headers mounted.

10.6 Powering the device

The MoCoS device should be powered with a 7.5V DC power source that is capable of supplying at least 500mA. When using the portable device, connect an external DC power supply with a 2.1mm plug (center pin positive) to the socket at the front of the device. When using the crate version of the device, you can either power it in the same way as the portable version or mount a 7.5V AC-to-DC converter directly onto the PCB and connect a cable supplying 220V/50Hz to it.

10.7 SD card

The MoCoS device requires a micro SD card for storing its configuration and logging data locally. The SD card specified in the parts list (appendix B) is recommended, but any micro SD card can be used. The card has to be formatted to use a FAT16 or FAT32 file system. This can be done by inserting the card into a micro SD card reader connected to your computer.

After formatting the card, it should be inserted into the SD card slot of the Arduino Ethernet Shield. If the card is formatted and inserted properly, the left/top status LED will turn green when powering the device. Please make sure the SD card is accessible before starting any other configuration or monitoring tasks. If the SD card cannot be initialised or read by the MoCoS device (left/top LED turns and stays red), try to power-cycle the device, make sure the card is formatted and inserted correctly and check if you can use it with another computer.

10.8 Status LEDs

Three tri-color status LEDs are located at the front of the MoCoS device showing the current status of the device, the status of its network connectivity and whether data acquisition works correctly or not. On the portable version of the device, the LEDs are located on the front side to the left of the USB and RJ45 sockets. On the crate version of the device, the LEDs are mounted on the front panel above the USB and RJ45 connectors and are rotated by 90 degrees clockwise. Therefore, the leftmost LED of the portable device corresponds to the uppermost LED on the crate version. The center LEDs belong together and the rightmost LED of the portable device corresponds to the bottom LED of the crate version. This section will explain the colours displayed by each individual LED and provide a visual representation of common statuses.

10.8.1 Device Status and SD LED

The leftmost/uppermost LED indicates the overall status of the device. Specifically, it will turn yellow as soon as the MoCoS device is connected to a power supply and will turn green if the SD card can be initialised. Since the SD card has to be present for correct operation of the device, please make sure that this status LED turns green signalling the card is present and can be accessed. If this LED turns and stays red, the SD card could not be accessed. Try to power-cycle the MoCoS device. If the error persists, please check that the SD card is properly inserted into the Arduino Ethernet Shield and make sure it is formatted as FAT16 or FAT32. For more information about the SD card used in the MoCoS device, please refer to section 10.7.

10.8.2 Network Connectivity LED

The center LED indicates the network connectivity status of the device. After the SD card has been initialised, this LED will turn green if the MoCoS device can access your local network and red if it cannot obtain an IP address. If this LED turn red and your device cannot connect to your local network, please make sure that the connection parameters (MAC address and IP address) are correct. Look up the MAC address of your device (on a sticker at the bottom of the Ethernet Shield) and the IP address your device should use on your local network and follow the instructions in section 9.2 for uploading these network parameters to the device using the MoCoS desktop application.

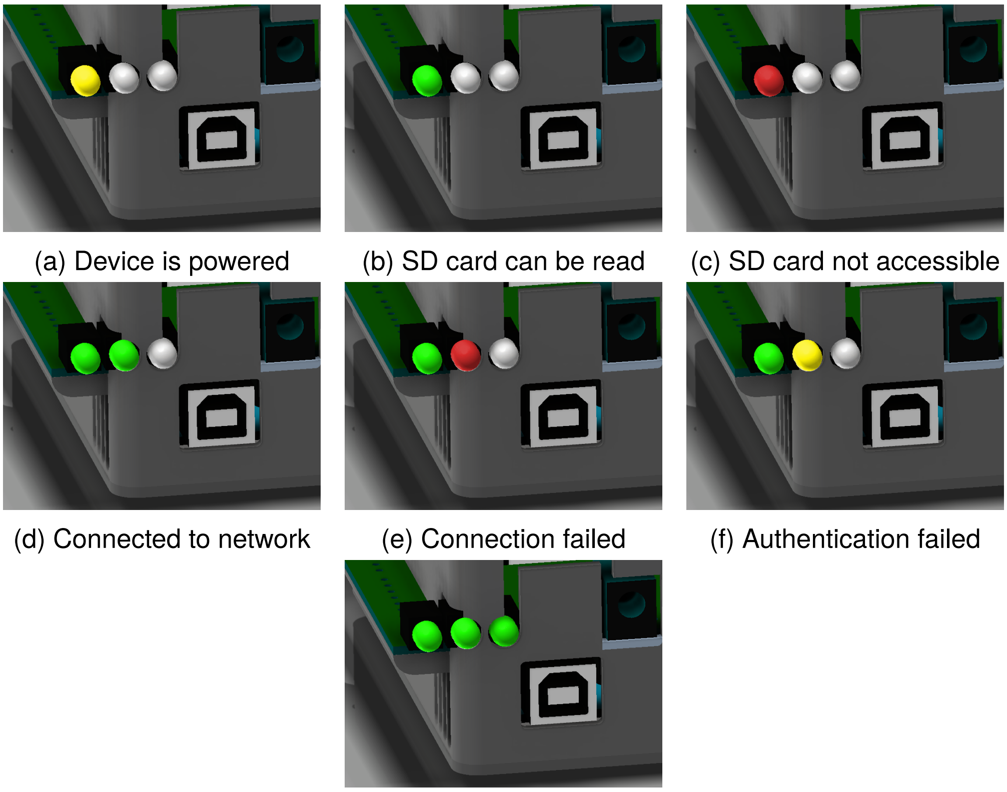
If your device can connect to the network and communicate with the configuration server, it will start downloading the latest configuration. To do so, it will authenticate with the web service using the account username and password specified during the setup procedure described in section 6.2. If the authentication works and the device can identity itself to the server, the network connectivity LED will stay green. If the server refuses the authentication information sent by the device, this LED will turn yellow. In that case, please check that the account username and password used by your device match the values shown in the "Account" section of the web service. The current username used by the device can be displayed by entering the IP address of your device on your local network in a web browser (provided your device can connect to your network).

10.8.3 Monitoring LED

The rightmost/bottom LED indicates the status of the data acquisition of the Monitoring & Control System. This LED will turn green if the data acquired can be successfully uploaded to and logged by the web service. If this LED does not turn and stay green after your device has finished its start up process and downloaded the configuration from the logging server, please check the account username and password used by your device to identify itself to the web service and check that the logging server is reachable.

10.8.4 Common Device Statuses

The device statuses shown in figure 20 represent common situations and issues during startup and operation of the device. Please refer sections 10.8.1, 10.8.2 and 10.8.3 for more details.



(a) Device is powered

(b) SD card can be read

(c) SD card not accessible

(d) Connected to network

(e) Connection failed

(f) Authentication failed

(g) All ok, monitoring

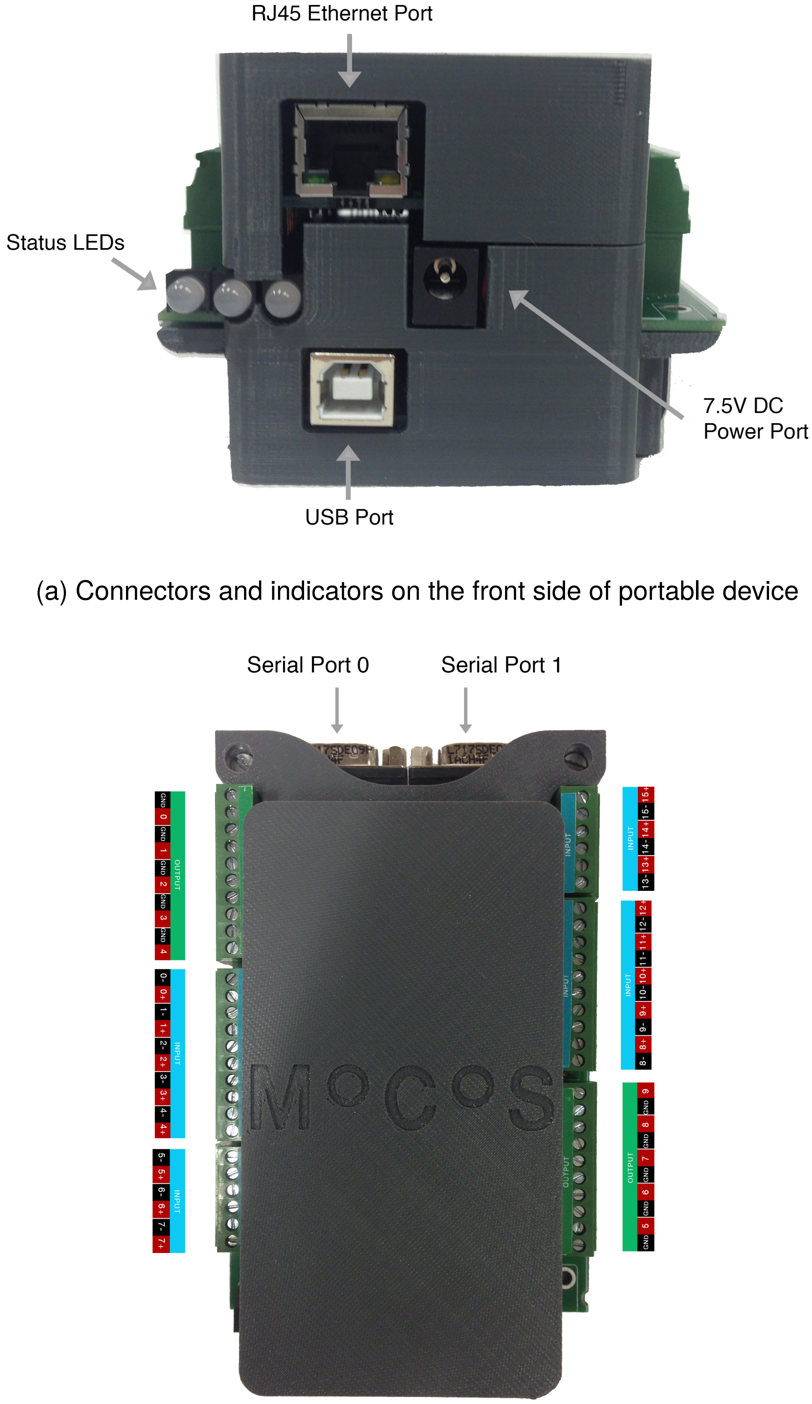
Figure 20: Common device statuses: The three status LEDs indicate the current status of the device. Common statuses and issues are shown, which can occur during startup or operation of the device.

10.9 Connectors

The MoCoS device features the following connectors and indicators:

- 16 universal inputs, which sensors of the supported types can be connected to
- 10 outputs, which can be controlled by threshold-based or PID-based logic or used as power outputs
- two D-Sub 9 RS232 serial ports
- RJ45 Ethernet port for connecting the device to your network
- USB-B port for communicating with the desktop application on your computer
- 2.1mm power plug (center pin positive), which should be used to supply the device with 7.5V DC
- three status LEDs to show the current status of the device

The connectors of the portable version of the device are shown in figure 21. The connectors or the crate version of the device are shown in figure 22.



(a) Connectors and indicators on the front side of portable device

(b) Connectors on the sides and back of portable device

Figure 21: Portable device connectors and indicators: Communication and power ports as well as status indicators are located at the front of the portable device. 16 universal inputs and 10 outputs are located along the sides, while two RS232 ports are located at the back.

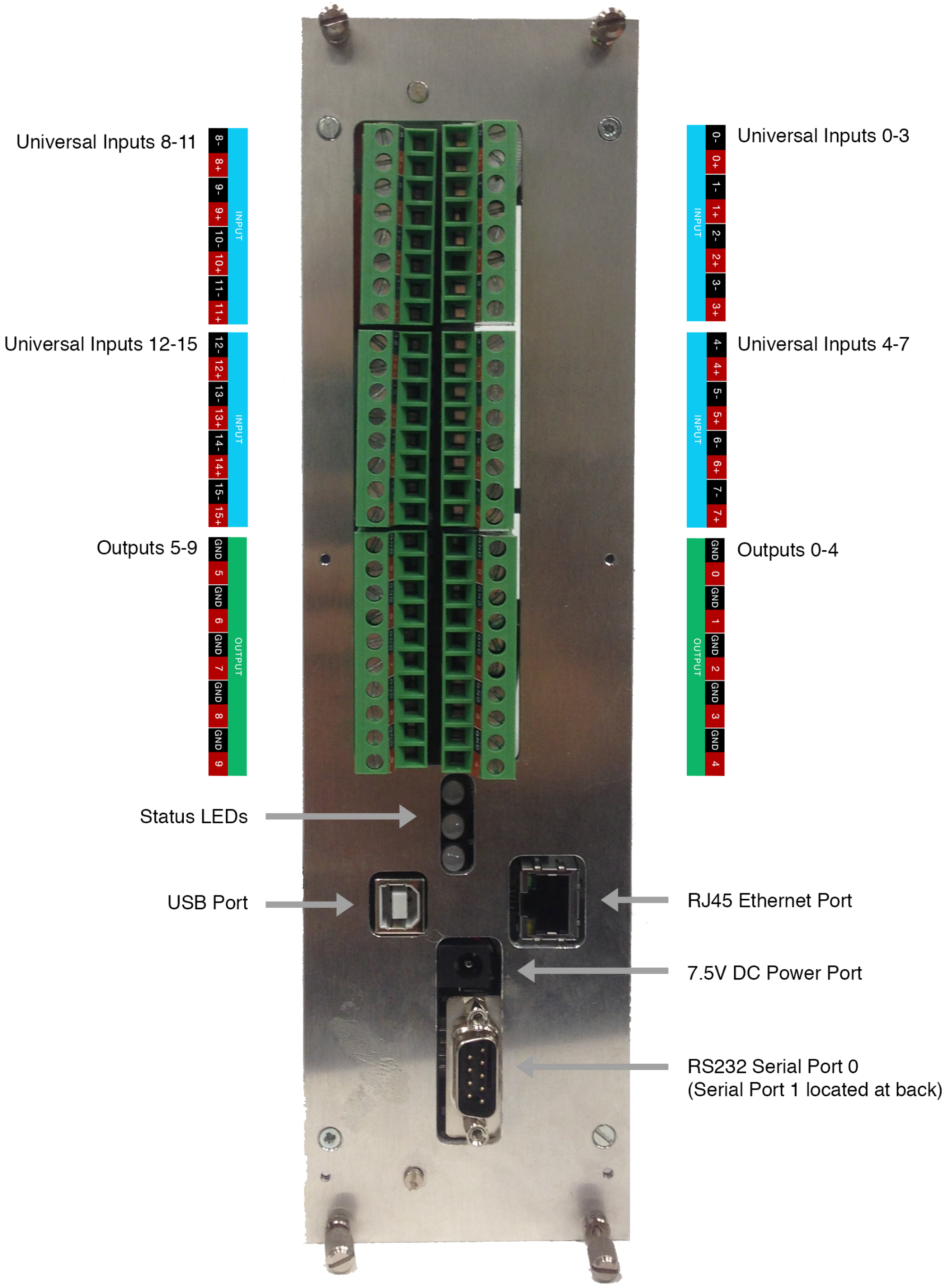


Figure 22: Crate version connectors and indicators: All communication and power ports as well as status indicators, universal inputs and outputs and one RS232 port are located on the front panel.

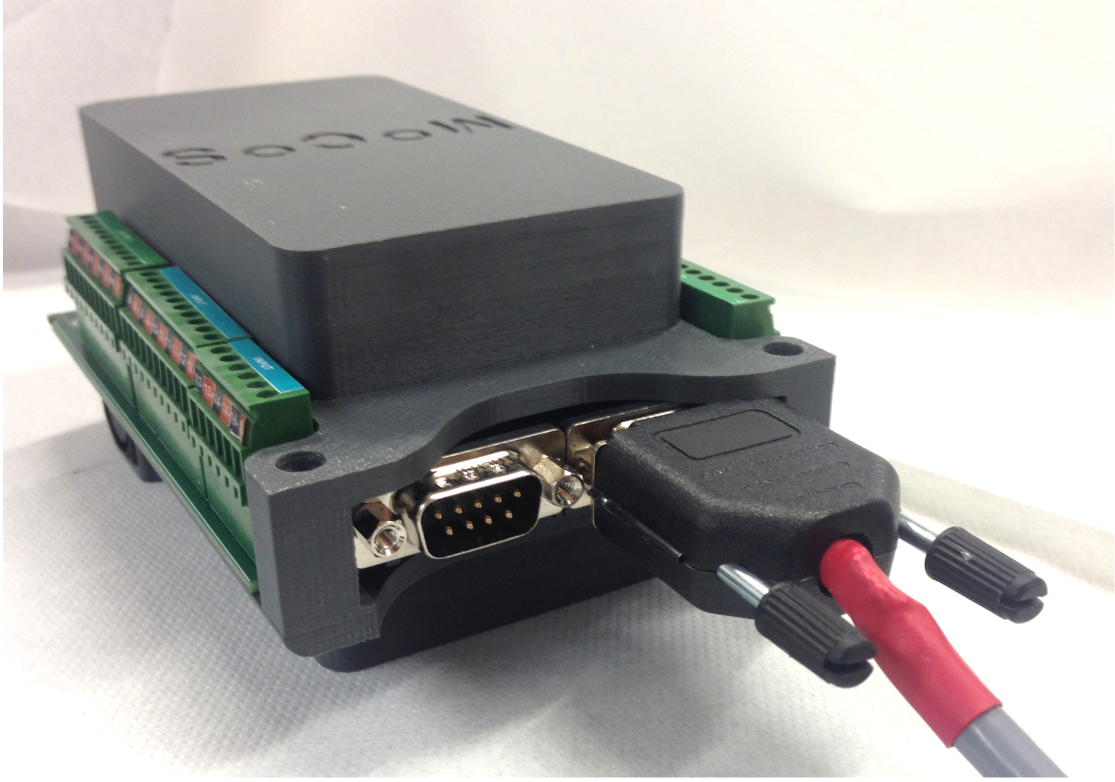


Figure 23: D-Sub 9 RS232 ports: Two D-Sub 9 ports are located at the back of the device (one at the front in the crate version), which allow communication with external devices over a serial RS232 interface. The picture shows the ports at the back of the portable device with an external device connected to one of them.

11 RS232 Communication

External sensors or other devices that provide a RS232 serial interface can be directly addressed by MoCoS. This allows you to query and collect data collected by another device without losing any information.

11.1 Connecting devices via RS232

There are two serial ports available on the MoCoS device as shown in figure 23. On the portable device, they are both located at the back of the device. The right one (when viewing from the back) is "Serial Port 0", the left one "Serial Port 1". On the crate version of MoCoS "Serial Port 0" is accessible from the front panel, while "Serial Port 1" is facing backwards. This allows you to connect up to two separate devices. For each device, you can specify multiple queries and pieces of data to be collected. This may be useful when querying data from an external device that has multiple channels. Both RS232 ports of the MoCoS device are male D-Sub 9 connectors. Therefore, you need a cable with a female D-Sub 9 plug to connect an external device to one of the serial ports.

11.2 Specifying RS232 communication protocol

Each device uses its own communication protocol consisting of end-of-line characters, query strings and a specific order or commands from both communicating parties that has to be followed. To accommodate a large number of protocols, MoCoS allows you to specify these details of the serial communication protocol to be used when querying a specific device. Please consult the manual of the device you want to communicate with for the necessary serial communication parameters. Make sure, that you set the device you want to communicate with to use a baud rate of 19200.

Once you have looked up the communication protocol used by the external device you have connected to your MoCoS device, login to the MoCoS web service, move the cursor over one of the unused inputs and click on the configure button (gear icon). Although you are not using any universal input, you need to choose one of the 16 universal input channels for data collection. The data collected via RS232 will be assigned to the selected input channel and this channel can not be used by any other sensor.

Choose "Serial Port 0" or "Serial Port 1" as the type of sensor depending on which serial port your external device is connected to. Next, enter the query string that will be sent by the MoCoS device to the external device to ask for a specific piece of information. Also, please specify the end of line character your device expects (using the ASCII code of the character). In most cases this will be either 13 (carriage return) or 10 (line feed).

Some devices will send you an acknowledgement response after the query. If this is the case for your external device, please check "Wait for acknowledgement from device" and enter the ASCII code of the acknowledgement character MoCoS should wait for. In most cases this will be 6 (acknowledge). To get the actual data after the acknowledgement has been registered by your MoCoS device, an enquiry character has to be sent. Please enter the ASCII code that will be used to query for the actual data after receiving an acknowledgement response in the field "Enquire data character" as an ASCII code. Commonly, this will be 5 (enquire). Finally, there may be multiple values returned by an external device for a single query. Use the "Response Delimiter" and "Value Index" field to specify which value of a data response should be logged. If, for example, your device returns a string like "1, 23.1, 9" and you are interested in the number "23.1", enter "," as the response delimiter and "1" as the index (without quotation marks). Select the physical unit the response is in and click on "Apply" to save and upload the new configuration.

Your MoCoS device will start querying the external device after downloading the updated configuration upon the next time it starts up and log the acquired data. Please note, that the end of line characters, the acknowledgement and enquire characters, the response delimiter and the value index are set for each serial port, while the query string can be set for each individual input channel. Therefore, you can query multiple pieces of information from one external device, but the same communication protocol will be used for all queries to the device connected to a specific serial port.

A PCB Schematic and Board

The signal conditioning PCB inside the MoCoS device was designed using Altium Designer. The board design files can be downloaded from the "Help" section of the web service. Additionally, CadSoft EAGLE schematic and board design files are also provided. While the Altium files were used for board manufacturing, the EAGLE schematic features the same functionality.

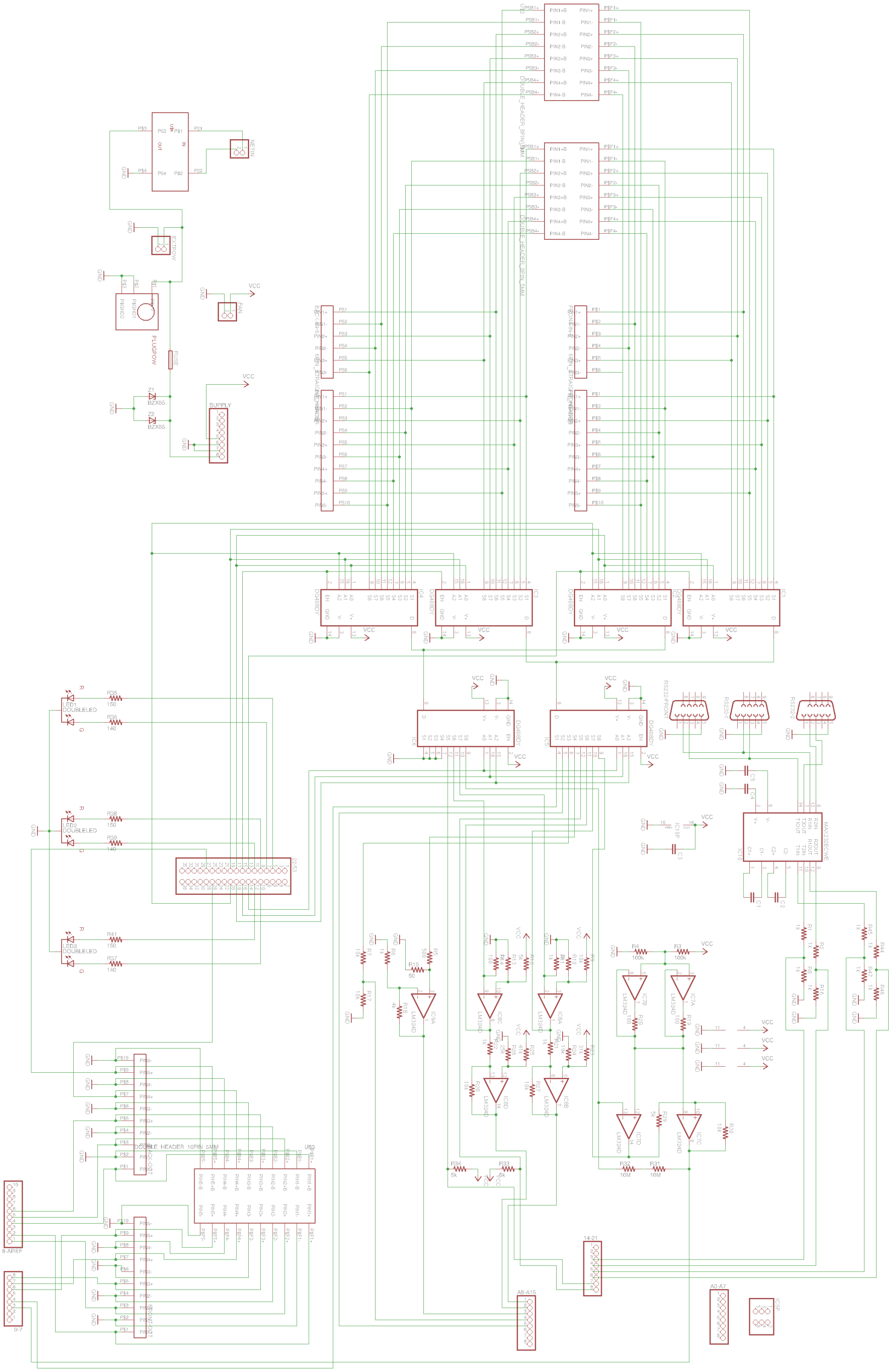


Figure 24: PCB schematic

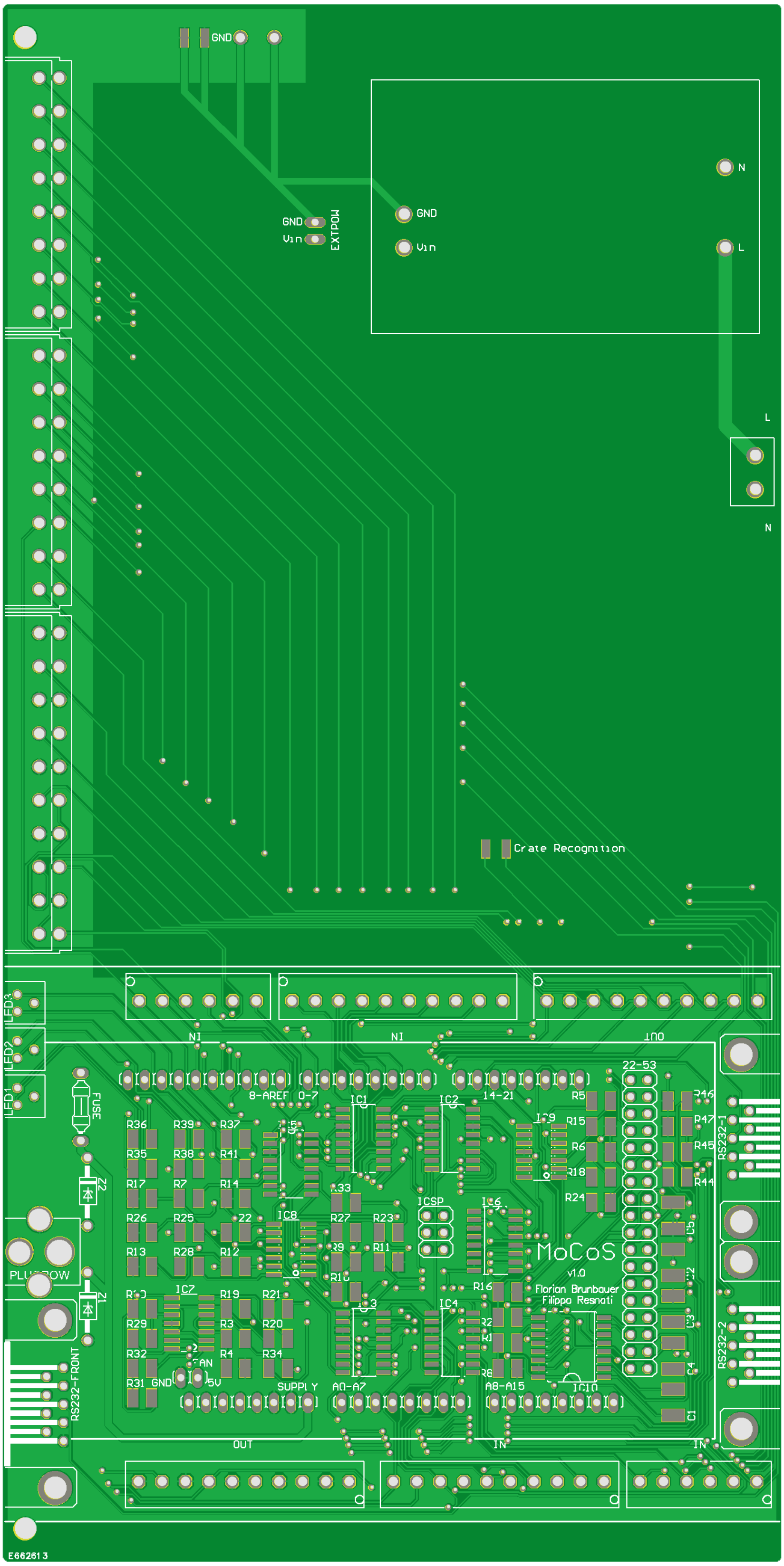


Figure 25: PCB layout front

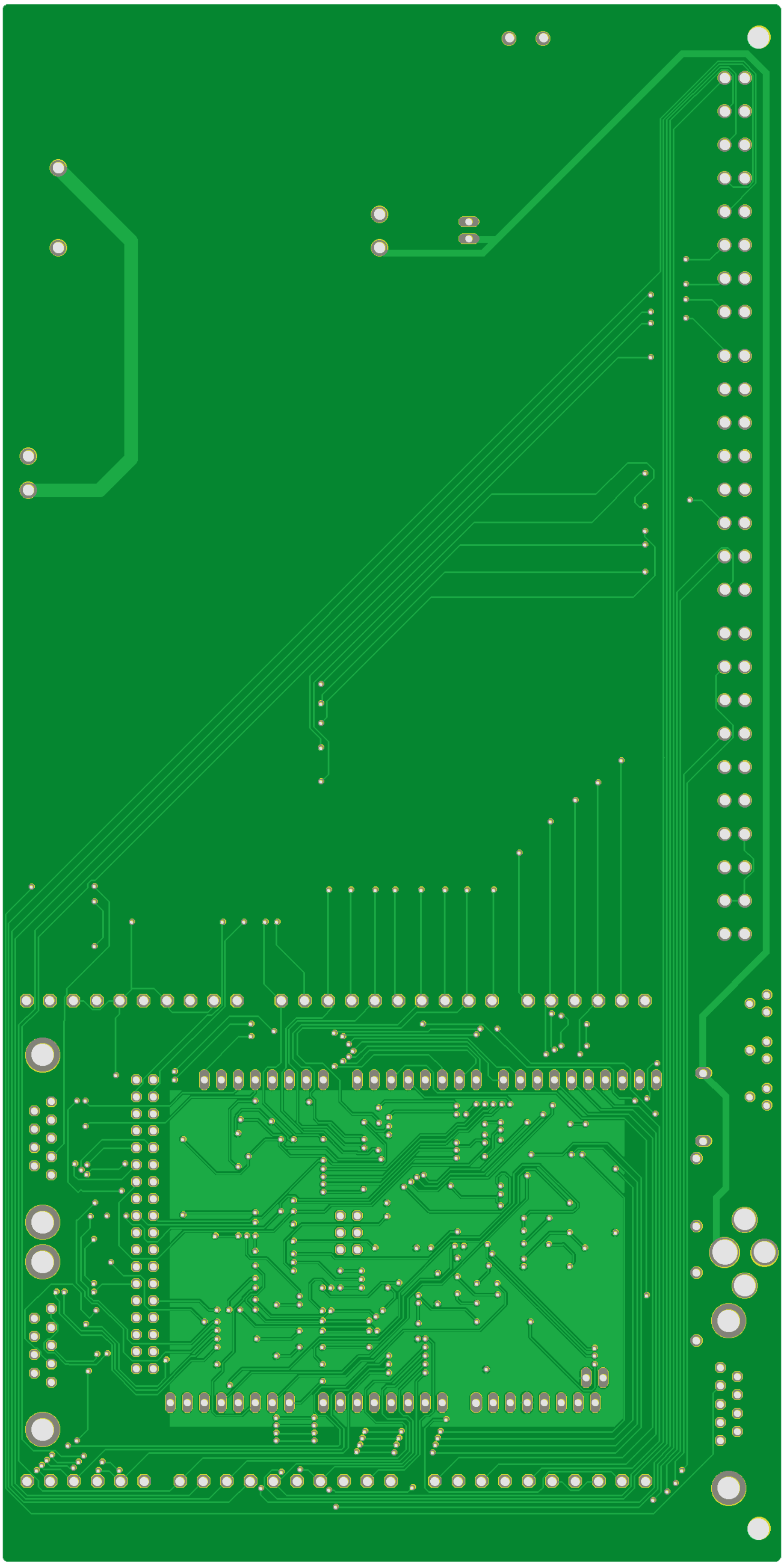


Figure 26: PCB layout back

B Parts List

Part	Value	Device	Package	Description	Model	Function / Part of circuit	Farnell Order #
0-7	8 way	1X08	Through Hole	Shield Header	Both	Connection to Arduino	Adafruit (ID: 85)
8-AREF	10 way	1X10	Through Hole	Shield Header	Both	Connection to Arduino	Adafruit (ID: 85)
A0-A7	8 way	1X08	Through Hole	Shield Header	Both	Connection to Arduino	Adafruit (ID: 85)
ICSP	2x3 way	2X03	Through Hole	Shield Header	Both	Connection to Arduino	Adafruit (ID: 85)
SUPPLY	8 way	1X08	Through Hole	Shield Header	Both	Connection to Arduino	Adafruit (ID: 85)
EXTPOW	2 way	1X02	Through Hole	Pin Receptable	Both	DC Power Input	1667535
FAN	2 way	1X02	Through Hole	Pin Receptable	Both	Connection to Fan	1667535
NETIN	2 way	1X02	Through Hole	Pin Receptable	Crate	AC Power Input	1667535
14-21	8 way	1X08	Through Hole	Pin Header	Both	Connection to Arduino	2311674
22-53	2x18 way	2X18	Through Hole	Pin Header	Both	Connection to Arduino	2025721
A8-A15	8 way	1X08	Through Hole	Pin Header	Both	Connection to Arduino	2311674
FUSE	1.25A	Fuse, Axial	Through Hole	Fuse	Both	Overcurrent Protection	1566018
Z1	8.2V	1N5344BG	Through Hole	Zener-Diode	Both	Overcurrent Protection	9558020
Z2	8.2V	1N5344BG	Through Hole	Zener-Diode	Both	Overcurrent Protection	9558020
IC1		DG408LDY	SOI16	8 Channel Multiplexer	Both	Positive Universal Input Routing	1077099
IC2		DG408LDY	SOI16	8 Channel Multiplexer	Both	Positive Universal Input Routing	1077099
IC3		DG408LDY	SOI16	8 Channel Multiplexer	Both	Negative Universal Input Routing	1077099
IC4		DG408LDY	SOI16	8 Channel Multiplexer	Both	Negative Universal Input Routing	1077099
IC5		DG408LDY	SOI16	8 Channel Multiplexer	Both	Measurement Circuit Routing	1077099
IC6		DG408LDY	SOI16	8 Channel Multiplexer	Both	Measurement Circuit Routing	1077099
IC7		LMC6484AIM	SOI16	Operational Amplifier	Both	Capacitance Measurement	1468891
IC8		LMC6484AIM	SOI16	Operational Amplifier	Both	PT100/PT1000 Measurement	1468891
IC9		LMC6484AIM	SOI16	Operational Amplifier	Both	4-20mA Measurement	1468891
IC10		MAX232DW	SOI16L (Wide)	RS232 Transceiver	Both	RS232 to TTL conversion	9589783
LED1	Red/Green	L-150A9VS	Through Hole	PCB-LED	Both	Monitoring Status Indicator	2373392
LED2	Red/Green	L-150A9VS	Through Hole	PCB-LED	Both	Ethernet Status Indicator	2373392
LED3	Red/Green	L-150A9VS	Through Hole	PCB-LED	Both	SD Card Status Indicator	2373392
PLUGPOW	2.1mm	NEB 21 R	Through Hole	2.1mm Socket	Both	DC Power Input	1217037
C1	1 µF	C-EUC1812	C1812	Capacitor	Both	RS232 to TTL conversion	1759472
C2	1 µF	C-EUC1812	C1812	Capacitor	Both	RS232 to TTL conversion	1759472
C3	1 µF	C-EUC1812	C1812	Capacitor	Both	RS232 to TTL conversion	1759472
C4	1 µF	C-EUC1812	C1812	Capacitor	Both	RS232 to TTL conversion	1759472
C5	1 µF	C-EUC1812	C1812	Capacitor	Both	RS232 to TTL conversion	1759472
R1	1 kΩ		R1210	Resistor	Both	RS232 to TTL conversion	1470030
R2	1 kΩ		R1210	Resistor	Both	RS232 to TTL conversion	1470030
R3	100 kΩ		R1210	Resistor	Both	Capacitance Measurement	2380674
R4	100 kΩ		R1210	Resistor	Both	Capacitance Measurement	2380674
R5	500 Ω		R1210	Resistor	Both	4-20mA Measurement	2312594
R6	1 kΩ		R1210	Resistor	Both	4-20mA Measurement	1470030
R7	10 kΩ		R1210	Resistor	Both	4-20mA Measurement	1470031
R8			R1210	Resistor	Both	Do not mount	
R9	10 kΩ		R1210	Resistor	Both	PT1000 Measurement	1470031
R10	2 kΩ		R1210	Resistor	Both	PT1000 Measurement	2380712
R11	1 kΩ		R1210	Resistor	Both	PT1000 Measurement	1470030
R12	5 kΩ		R1210	Resistor	Both	PT100 Measurement	2327563
R13	1 kΩ		R1210	Resistor	Both	PT100 Measurement	1470030
R14	100 Ω		R1210	Resistor	Both	PT100 Measurement	2380671
R15	50 Ω		R1210	Resistor	Both	4-20mA Measurement	1470034
R16			R1210	Resistor	Both	Do not mount	
R17	10 kΩ		R1210	Resistor	Both	4-20mA Measurement	1470031
R18	4 kΩ		R1210	Resistor	Both	4-20mA Measurement	2312581
R19	100 Ω		R1210	Resistor	Both	Capacitance Measurement	2380671
R20	100 Ω		R1210	Resistor	Both	Capacitance Measurement	1470030
R21	1 kΩ		R1210	Resistor	Both	PT1000 Measurement	1470030
R22	1 kΩ		R1210	Resistor	Both	PT100 Measurement	1470030
R23	31 kΩ		R1210	Resistor	Both	PT1000 Measurement	2312571
R24	19 kΩ		R1210	Resistor	Both	PT1000 Measurement	2312504
R25	41 kΩ		R1210	Resistor	Both	PT100 Measurement	2327534
R26	25 kΩ		R1210	Resistor	Both	PT100 Measurement	2312550
R27	10 kΩ		R1210	Resistor	Both	PT1000 Measurement	1470031
R28	10 kΩ		R1210	Resistor	Both	PT100 Measurement	1470031
R29	5 kΩ		R1210	Resistor	Both	Capacitance Measurement	2327563
R30	10 kΩ		R1210	Resistor	Both	Capacitance Measurement	1470031
R31	10 MΩ		R1210	Resistor	Both	Capacitance Measurement	2324010
R32	10 MΩ		R1210	Resistor	Both	Capacitance Measurement	2324010
R33	5 kΩ		R1210	Resistor	Both	I2C Communication	2327563
R34	5 kΩ		R1210	Resistor	Both	I2C Communication	2327563
R35	150 Ω		R1210	Resistor	Both	Monitoring LED	1153467
R36	140 Ω		R1210	Resistor	Both	Monitoring LED	2312454
R37	140 Ω		R1210	Resistor	Both	SD LED	2312454
R38	150 Ω		R1210	Resistor	Both	Ethernet LED	1153467
R39	140 Ω		R1210	Resistor	Both	Ethernet LED	2312454
R41	150 Ω		R1210	Resistor	Both	SD LED	1153467
R44	1 kΩ		R1210	Resistor	Both	RS232 to TTL conversion	1470030
R45	1 kΩ		R1210	Resistor	Both	RS232 to TTL conversion	1470030
R46			R1210	Resistor	Both	Do not mount	
R47			R1210	Resistor	Both	Do not mount	
Crate Recognition	1k		R1210	Resistor	Crate	Model Recognition	1470030
RS232-1	Male	F09HP	F09HP	RS232 Socket	Portable	RS232 Input	2401240
RS232-2	Male	F09HP	F09HP	RS232 Socket	Both	RS232 Input	2401240
RS232-FRONT	Male	F09HP	F09HP	RS232 Socket	Crate	RS232 Input	2401240
Front In	6 way	Terminal Block	3.5mm / Through Hole	Pluggable Terminal Block	Portable	Universal Input	2067457 + 2067402
Front In	10 way	Terminal Block	3.5mm / Through Hole	Pluggable Terminal Block	Portable	Universal Input	2067461 + 2067406
Front Out	10 way	Terminal Block	3.5mm / Through Hole	Pluggable Terminal Block	Portable	Output	2067461 + 2067406
Back In	6 way	Terminal Block	3.5mm / Through Hole	Pluggable Terminal Block	Portable	Universal Input	2067457 + 2067402
Back In	10 way	Terminal Block	3.5mm / Through Hole	Pluggable Terminal Block	Portable	Universal Input	2067461 + 2067406
Back Out	10 way	Terminal Block	3.5mm / Through Hole	Pluggable Terminal Block	Portable	Output	2067461 + 2067406
Top In	8 way	Terminal Block	5mm / Through Hole	Pluggable Terminal Block	Crate	Universal Input	1708294 + 1860117
Top In	8 way	Terminal Block	5mm / Through Hole	Pluggable Terminal Block	Crate	Universal Input	1708294 + 1860117
Top Out	10 way	Terminal Block	5mm / Through Hole	Pluggable Terminal Block	Crate	Output	1860171 + 1860118
Bottom In	8 way	Terminal Block	5mm / Through Hole	Pluggable Terminal Block	Crate	Universal Input	1708294 + 1860117
Bottom In	8 way	Terminal Block	5mm / Through Hole	Pluggable Terminal Block	Crate	Universal Input	1708294 + 1860117
Bottom Out	10 way	Terminal Block	5mm / Through Hole	Pluggable Terminal Block	Crate	Output	1860171 + 1860118
Power Suppply Unit	7.5V / 10W	VTX-214-010-107	Through Hole	DC Power Supply Unit	Optional	DC Power Supply	2401052
Power Adapter	5V / 15W	3A-161DB-07	2.1mm Plug	DC Power Supply Adapter	Optional	DC Power Supply	1279527
Cooling Fan	5V	Axial Fan	17x17mm	Cooling Fan	Portable	Cooling of Arduino	RS: 758-8358
Memory Card	2GB	TS2GUSDC	Micro SD	Memory Card	Both	Storing Configuration	2290242
Arduino	Mega 2560	Arduino Mega 2560			Both	Microcontroller Board	2212779
Ethernet Shield	R3	Ethernet Shield		Arduino Ethernet Shield	Both	Ethernet Connection	2212783

C Technical Specifications

Since the MoCoS device is based on the Arduino Mega 2560 microcontroller board, most technical specifications pertain to the Arduino Mega 2560 board. Please be careful about the power supply used to power the MoCoS device. Use only a power supply that provides 7.5V DC with a minimum supply current of 500mA. Higher voltages may result in excessive heating and can damage sensitive components. A fuse on the custom PCB protects the board against excessive current flow. If you connect a power source that supplies more than 8.2V, the fuse will break and cut off the supply line to the microcontroller and all other electronics. If this happens, please replace the fuse (labeled "FUSE" on the PCB).

- Supply voltage: 7.5V DC
- Supply current: min. 500mA

- Input impedance: 100 MΩ

- Output voltage range: 0-5V
- Output current (single output): max. 40mA
- Output current (total): max. 200mA

- Number of universal inputs: 16
- Number of outputs: 10
- Number of RS232 ports: 2

- Connections: USB-B, RJ45, D-Sub 9 RS232, 2.1mm power socket, terminal blocks for universal input and outputs