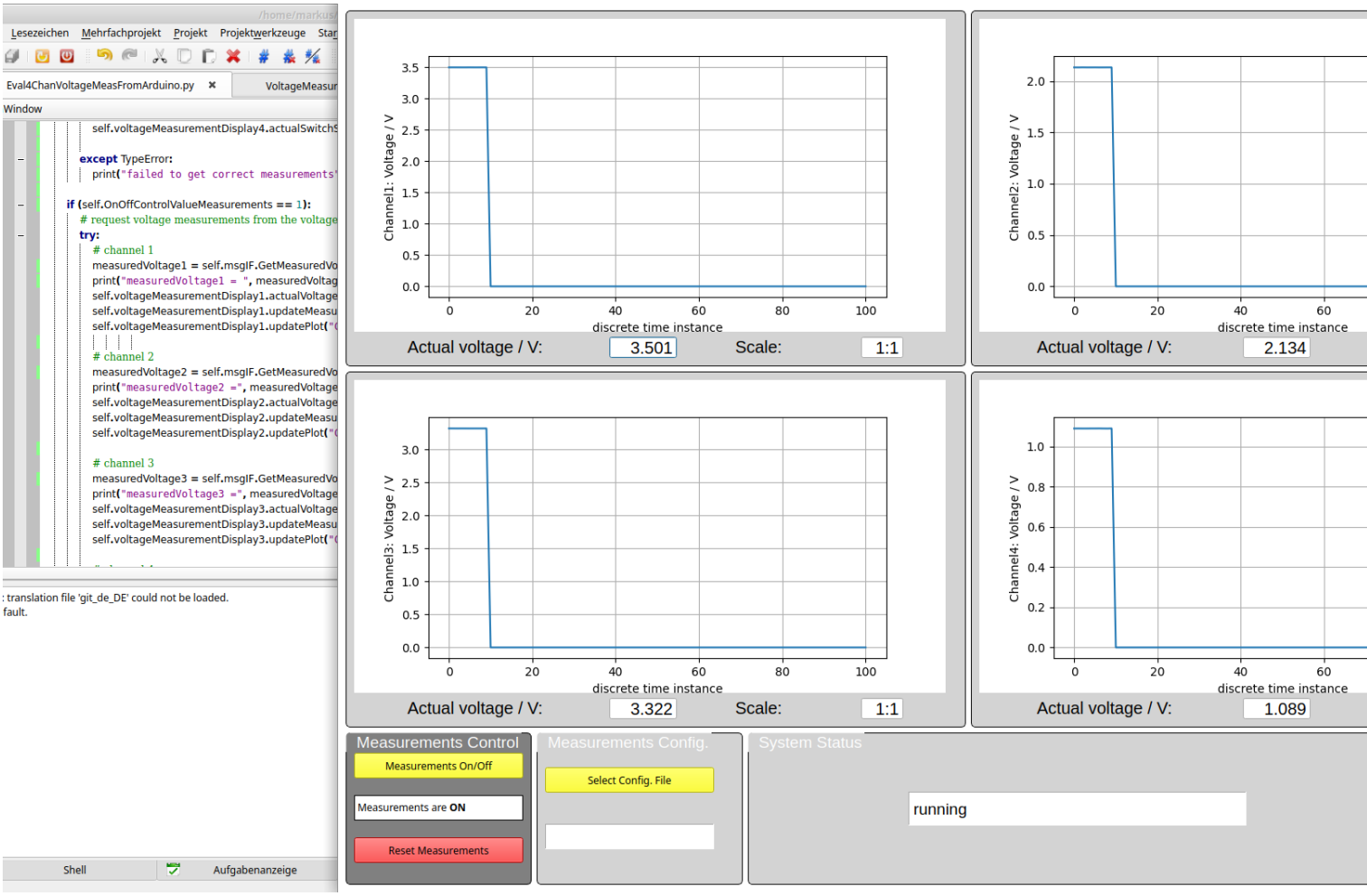
My Simple 4-Channel DAS

Description and Evaluation

by Dr. Markus Reinhardt

July 16, 2021



Contents

List of Figures

List of Tables

1 Concept description

A simple 4-channel low sample rate, high accuracy (16-bit ADC) Arduino based Data Acquisition System (DAS) to support general PC controlled universal measurement campaigns has been built.

The signals to be measured are fed to a 1:10 divider or via a 1:1 connection to the buffer amplifier of each channel. The signals are then ADC converted by the 4 channel ADS1115 ADC on the ADC module. The Arduino Uno is controlling the ADS1115 and receiving the digitized signals via the I2C interface. The data is then sent to the PC via the SoftwareSerial port, a USB/Serial module using the CmdMessenger library to the PC. On the PC a Python (PyQt) or C++ (Qt) program using also the (Py)CmdMessenger library is receiving, evaluating and displaying the data.
The projects consists of the following main HW components:

- An Arduino Uno to control the measurements and to report the measurements to the PC.
- An USB/Serial module to connect the PC with the Uno via the SoftwareSerial interface and an USB connector.
- A 4-Channel ADS1115 16-bit ADC module.
- A 4-Channel 1x buffer amplifier board, LM324 based.
- 4-Channel signal paths with 1:10 or 1:1 signal scaling selected by switches and related resistor dividers.
- A cheap plastic case.

The projects consists of the following main SW components:

- An Arduino sketch to control the ADC module and to send the data and the scaling switch states via the serial interface to the PC.
- A C++ / Qt based software library to sample, store and evaluate data from up to 4 channels.
- A PyQt based Python program running on the PC to receive and store the data from up to 4 channels and to scale the received signals according to the selected and reported scaling states.

2 Schematics

The schematics are shown in figure 1. The input section for each channel consists of

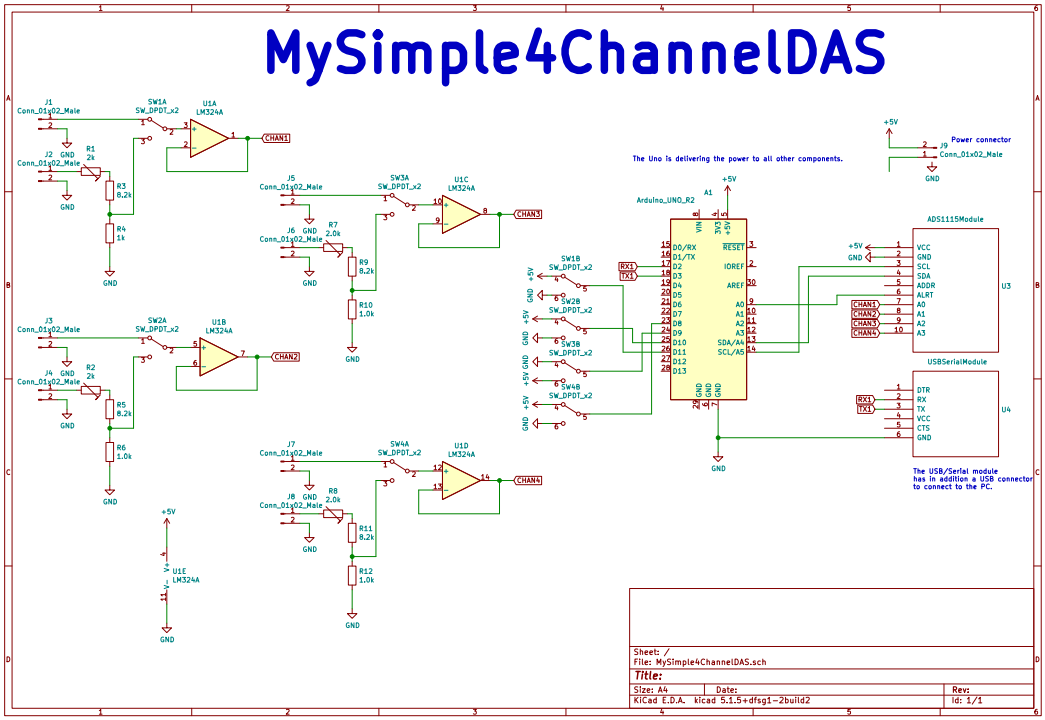


Figure 1: Schematics

the connectors for the 1:1 and the 1:10 scaling path. In the 1:10 path a resistor divider divides the input signal by 10 and feeds the attenuated signal to the switch that selects the input path. After the switch the signal is fed to the buffer amplifier with has a gain factor of 1.0 and from there to one of the four channels of the ADC module. Each switch is a DPDT type switch. One path is used for switching of the scaling path, the other for the reporting of the switch state to the micro-controller. The ADC module is controlled from an Arduino Uno that provides also the interface to the PC via a USB/Serial module.

3 Hardware setup

The four channel DAS front view of the case with the sockets and the switches four each of the four channels is shown in figure 2. The upper row of the sockets are the inputs of the paths with 1:10 scaling of the signals, the middle row of sockets are the inputs of

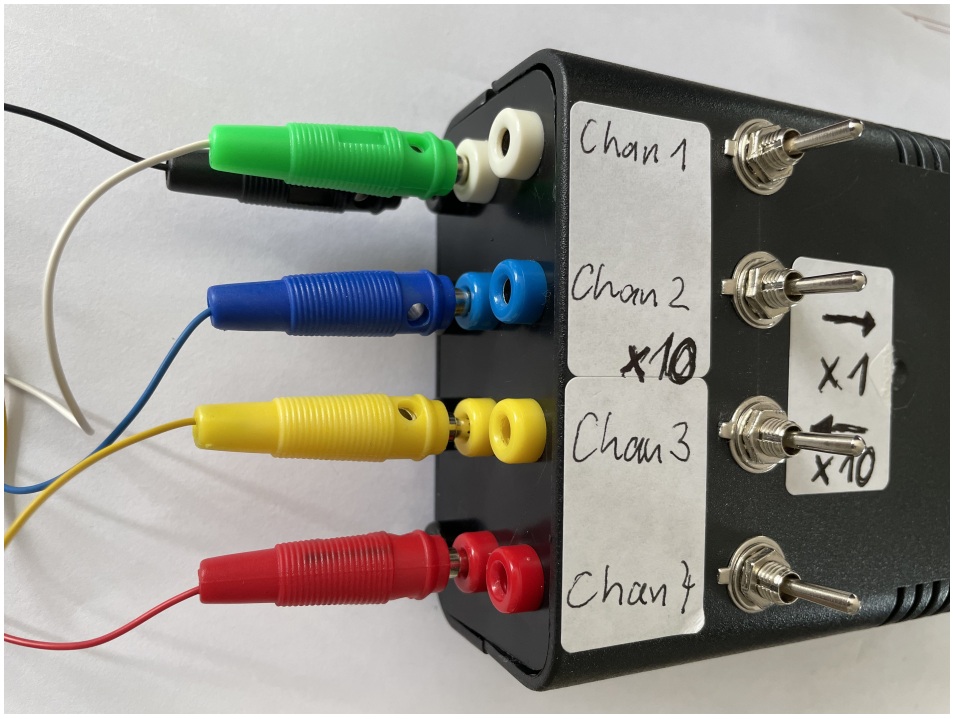


Figure 2: Four channel DAS front view of case)

the paths with 1:1 scaling of the signals. The lower row of the sockets are the ground connections.
The internal HW is shown in figure 3. It shows the Arduino Uno board, the ADC board, the USB/Serial converter module, the buffer amplifier (LM324) all mounted on a bread board. There are twelve 4mm connectors assembled on the front panel, four for ground connections, four for buffered signals and four for (by a factor of 1:10) attenuated and also buffered signals.
The used ADC module is shown in figure 4.

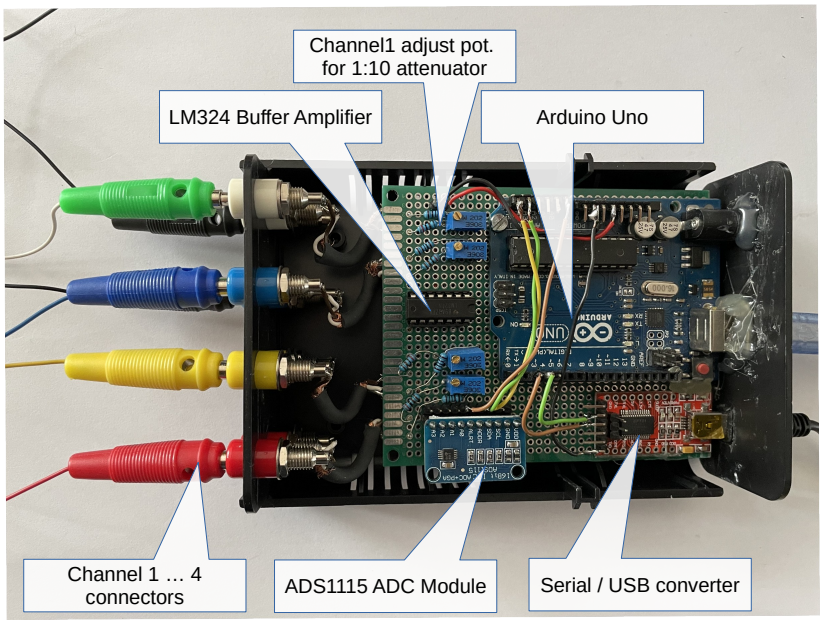


Figure 3: HW inside the case

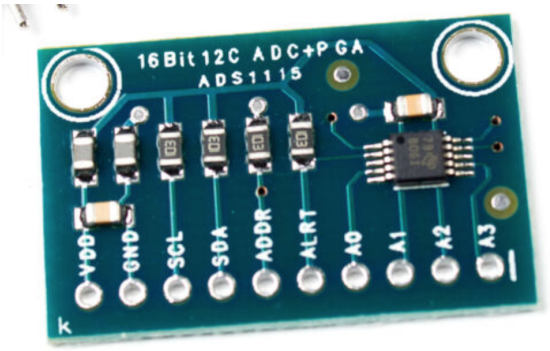


Figure 4: ADC module with ADS1115

4 Software

4.1 Arduino Sketch

The Arduino sketch is running on an Arduino Uno. It consists of 3 tasks.
The first task is the blinker task that blinks the on-board LED, i. e. it is working as an alive-signal.
The second task is the communication task that receives request/commands from the PC via the CmdMessenger Library and sends on request the voltage measurements back to the PC program. The CmdMessenger interface uses the SoftwareSerial library for the communication via a USB/Serial module to the PC. Arduino digital pins 2 (RX) and 3 (TX) are used by the SoftwareSerial library.
The third task is the measurement task that handles the I2C interface to the ADC module to get the measurement values. The standard I2C pins of the Arduino Uno are used here, i. e. Pins A4/SDA and A5/SCL). Pin A0 is used for the ADC module's ALERT pin interfacing.
The measured voltages and the status of the scale switches are also printed to the standard Arduino Serial port (/dev/ttyACM0) and can be watched with the Arduino Serial Monitor of the IDE. The trace of the measured voltages in the Arduino IDE Serial Monitor after the program setup is done is shown in figure 5.

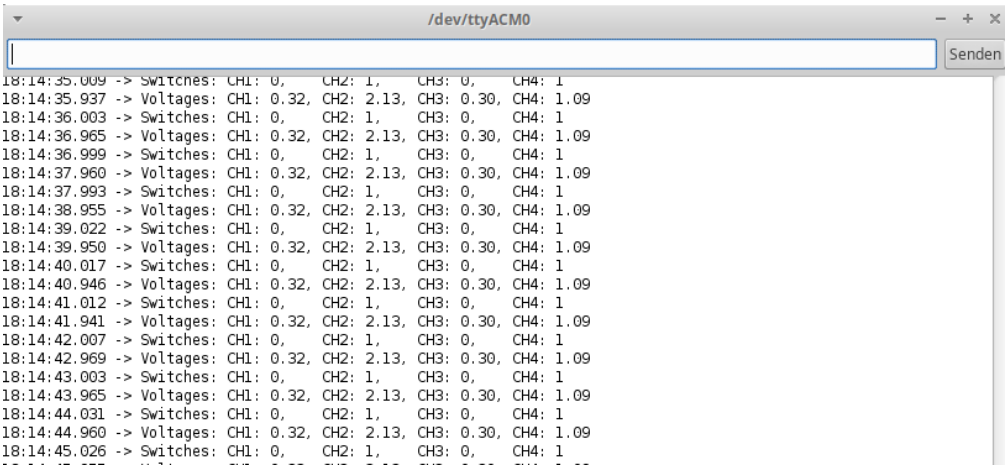


Figure 5: Arduino Serial Monitor: Trace of the four measured voltages and scale switches

4.2 PC Application

The GUI of the Python PC application together with the Eric6 IDE is shown in figure 6 after around 10 voltage samples have been processed for all four channels. The program is called by the shell script Eval4ChannelVoltageMeasurements which calls the python interpreter to execute the python program Eval4ChanVoltageMeasFromArduino.py.

The program reads via the PyCmdMessenger library the four measured voltages and the status of the scaling switches from the Arduino board via the USB/Serial module on port /dev/ttyUSB0 and displays the voltages vs. the discrete sample time index in four separate plots. The scaling of the ordinate of the plots is done automatically with the reception of each new sample. In addition the actual measured voltage values and the voltage scaling values (1:1 or 1:10) chosen by the switches for all channels are displayed below the plots.
The program also allows to store the measured data and to control the measurements via start/stop/reset commands in the GUI.

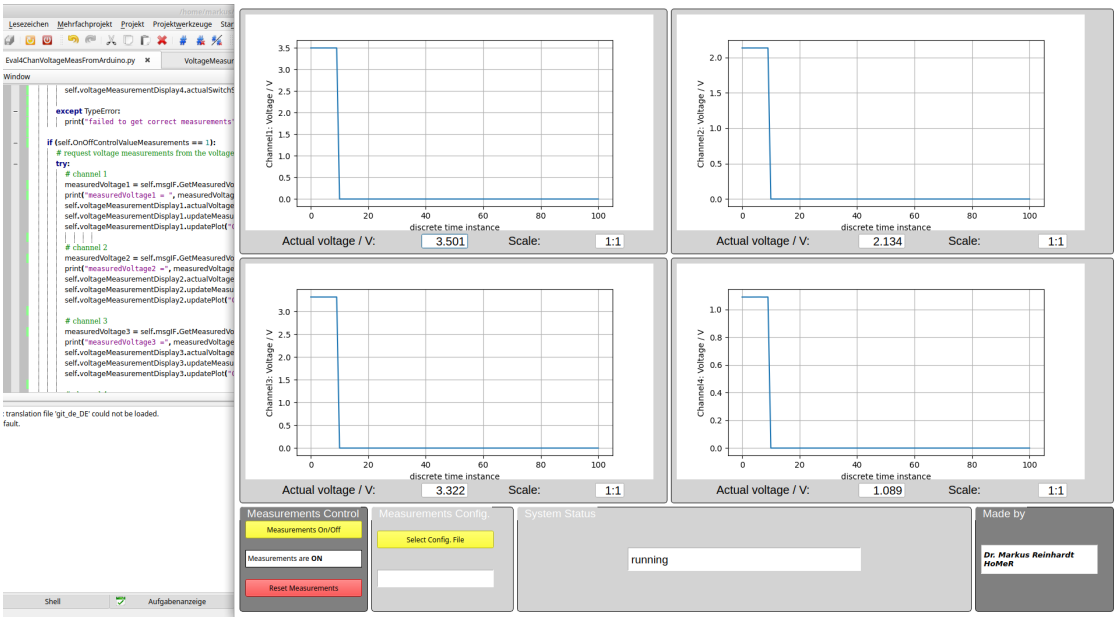


Figure 6: Python PC Application

The program allows to (re-)start, stop and reset the measurements.
The GUI of the PC program when the scaling of 1:10 is selected for the channels 1 and 3 and the scaling of 1:1 is selected for channels 2 and 4 is shown in figure 7.

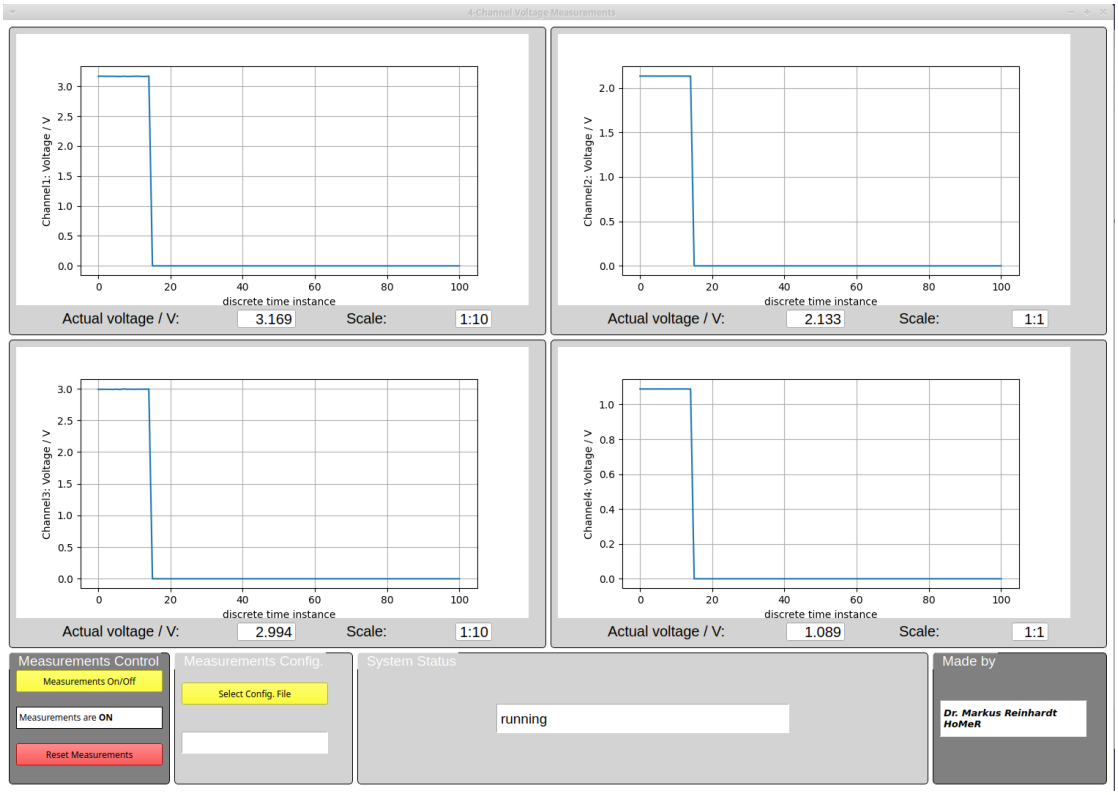


Figure 7: Python PC Application