

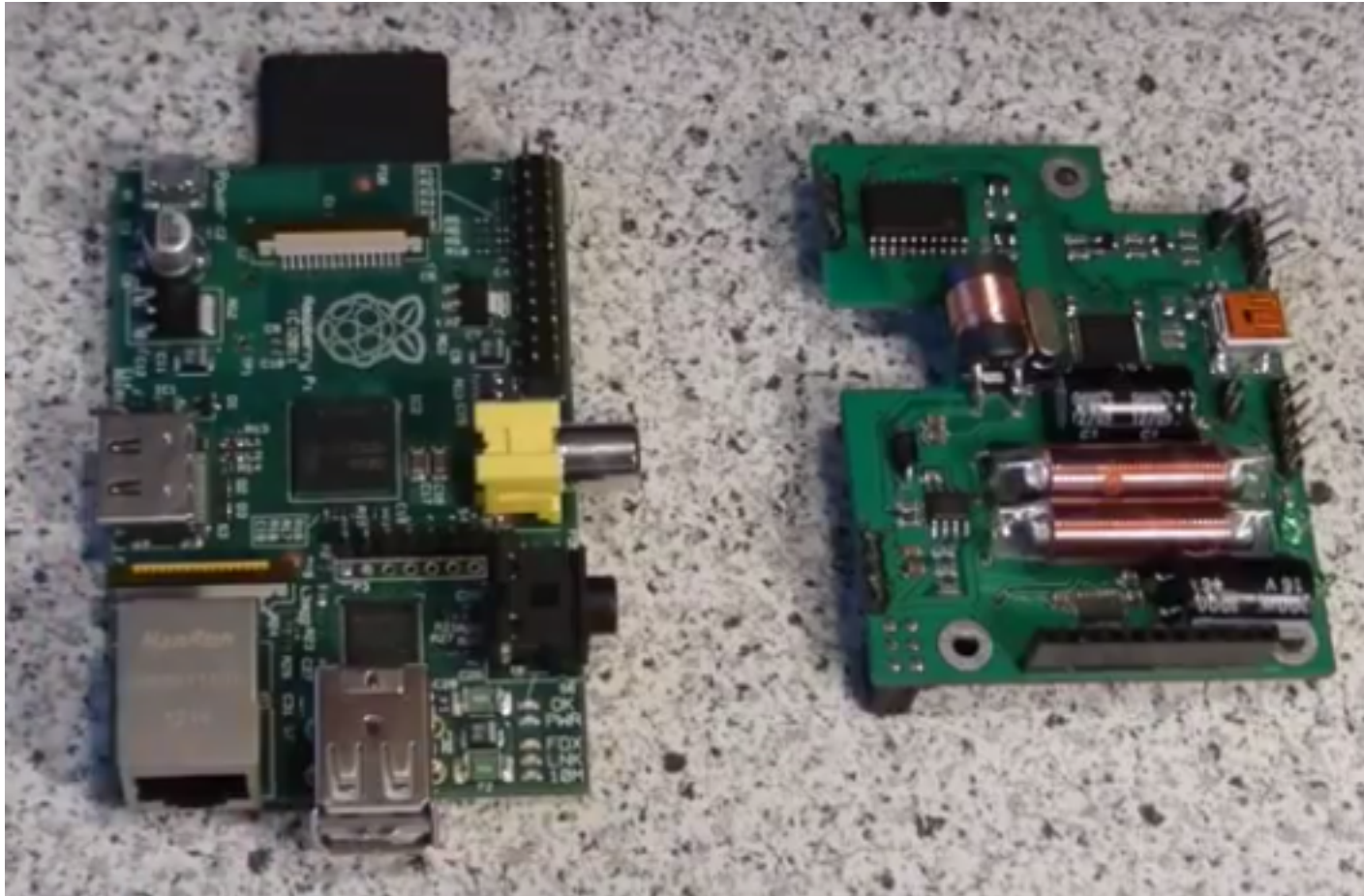
A few approaches to firmware
projects design

Boards set for Raspberry Pi extension.



- No compiler/toolchain needed for making the system work.
- ***Python*** based computer side SDK.
- ***Pawn*** embedded scripting machine.
- Boards in stack are driven by either main board, PC or Raspberry Pi.
- If coupled with Raspberry can provide it with power when needed.

Boards set for Raspberry Pi extension.



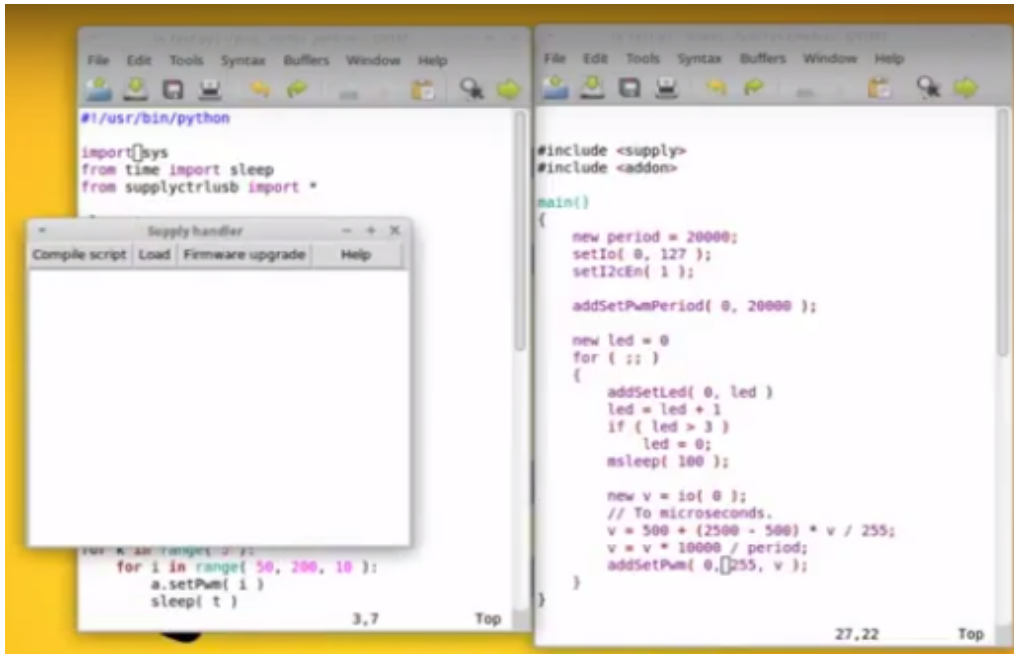
Boards set for Raspberry Pi extension.



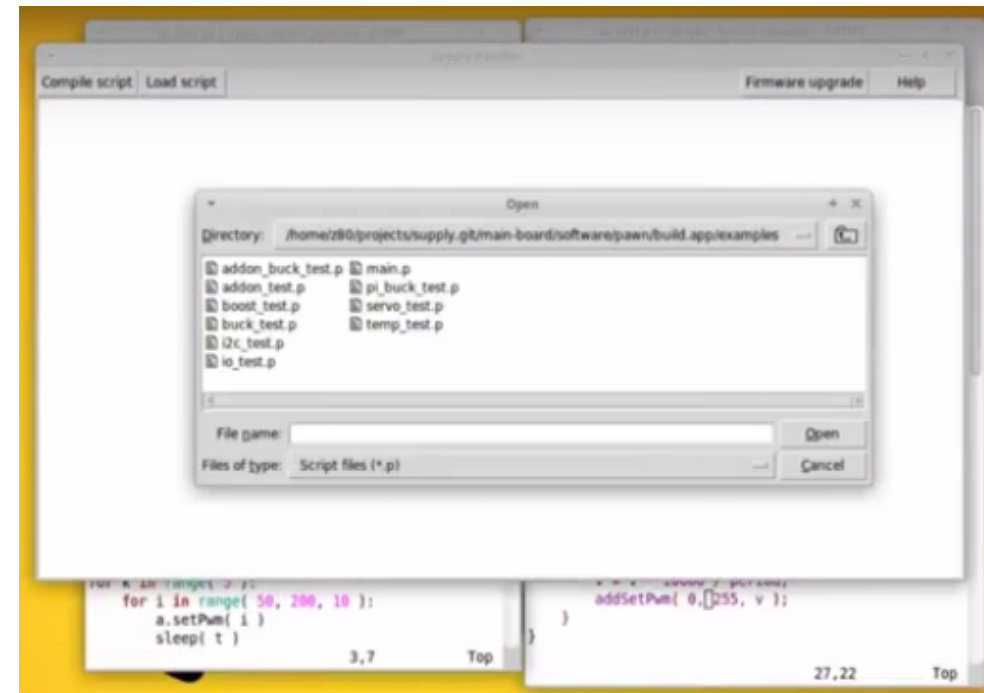
- Two extension boards with motor drivers, PWM, IO pin arrays.
- Main board with adjustable Buck, Boost converters, and standard periphs (USART, SPI, I2C, USB) exposed at edge connectors.

Boards set for Raspberry Pi extension.

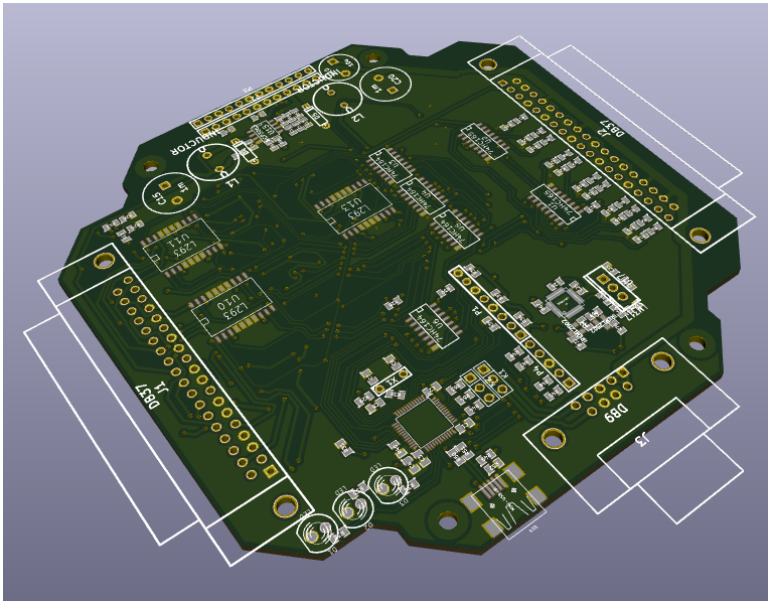
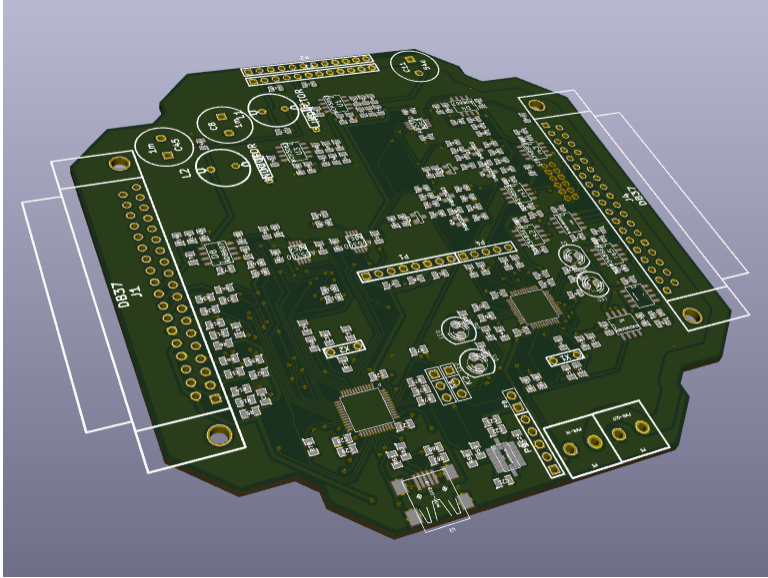
- ***Python*** and ***Pawn*** scripts examples side by side.



- New embedded script flashing dialog.

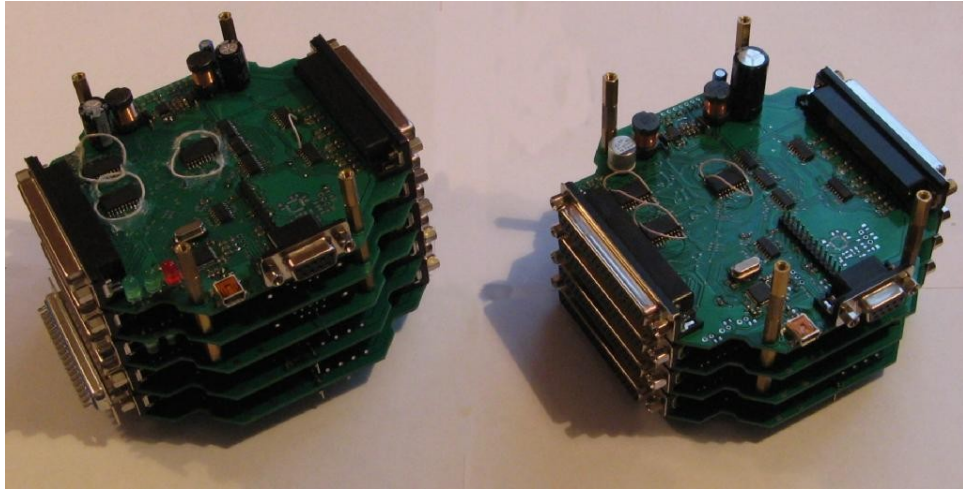


Scalable controller for pneumatic mechanism.



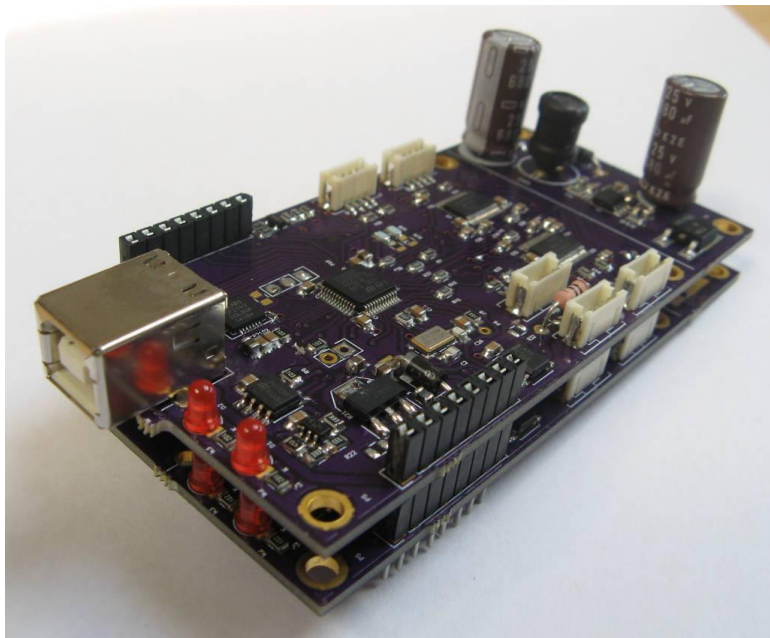
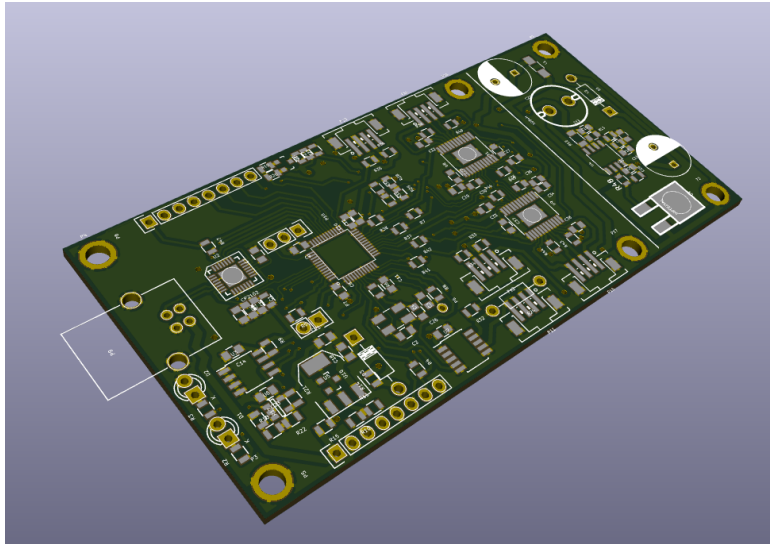
- The goal was to design cheap controller for a few mechanisms driven by pneumatic valves and electric motors.
- Minimize parts cost and PCB tooling cost.
- Number of controllers was small.
- Number of elements to control was not fixed and was relatively large ~150-200.

Scalable controller for pneumatic mechanism.



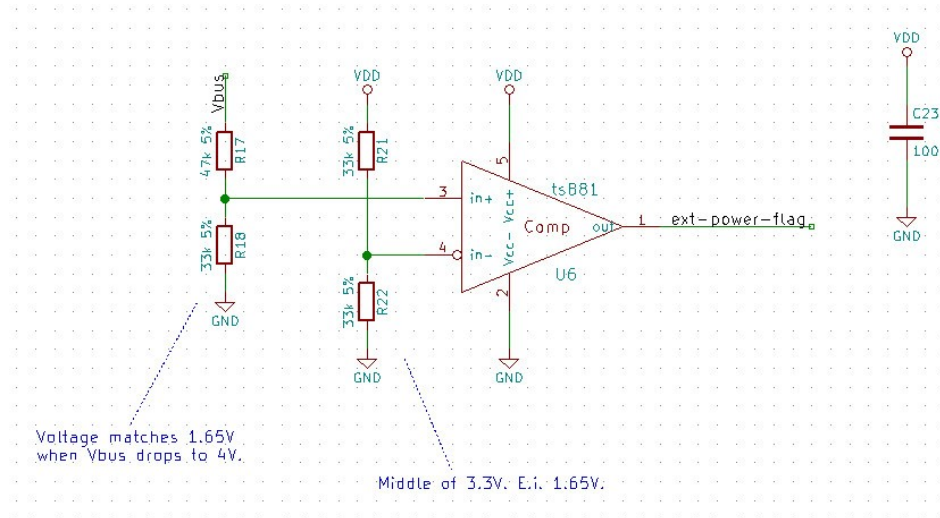
- Taking into account all possible combinations of pneumatic actuators, motors and sensors in all mechanism configurations
- it turned out that “**optimal**” numbers of different PCB types was 2.
- Boards are interconnected by **power** lines, common **I2C** bus and 2 **digital** lines for synchronous triggering.
- External communication was performed via either **I2C** or **USB** by single board computer running Linux OS.
- During debugging stage it also was equipped with embedded **Pawn** machine for easy movement algorithms trying/implementing.

USB powered expandable motor controller



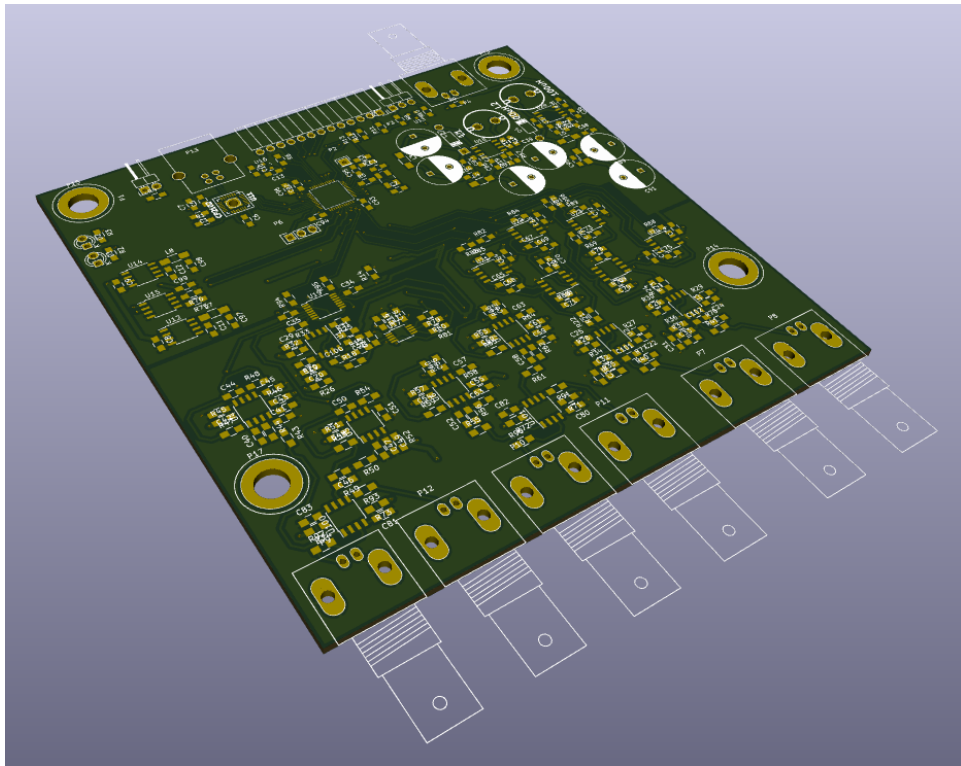
- The goal to minimize development resources, unify board design to maximum number of applications, provide power off immunity.
- The same scalable approach was applied.
- PCB with integrated USB driven power supply. Power supply part may be cut off for on top mounting or left blank for bottom mounting, see picture.
- One board can drive 2 stepper motors, 1 DC motor, 2 sensors.
- Scalable design by reusing the same board a few times and addressing individual boards using I2C bus and GPIO triggering for synchronization.

USB powered expandable motor controller



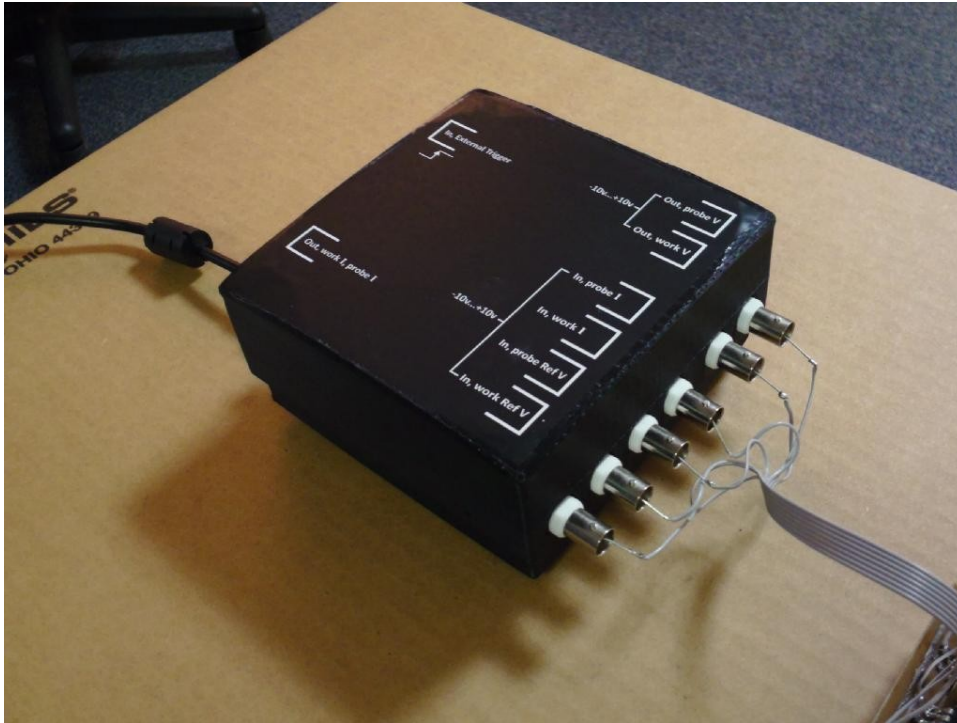
- Power off detection circuit.
- On power off event microcontroller gets power from capacitor precharged to relatively high voltage.
- Microcontroller has enough time to safely stop all movement processes, update state to EEPROM and shutdown peacefully.
- On next powerup state is recovered from EEPROM, validated using **crc8** checksum and applied.

3rd party bipotentiostat control module.



- Objective was to simultaneously spend minimum resources as client had no extra money to finance the development but achieve challenging requirements.
- Analog precision to achieve $\sim 1\text{mV}$
- 2 analog outputs $\pm 10\text{V}$
- 4 analog inputs $\pm 10\text{V}$
- External digital trigger
- One switchable analog output
- USB computer interface
- ~ 1000 points (4 readings per point) per second, 10V per second sweep rate.
- Parameters of hardware to be controlled were unknown.

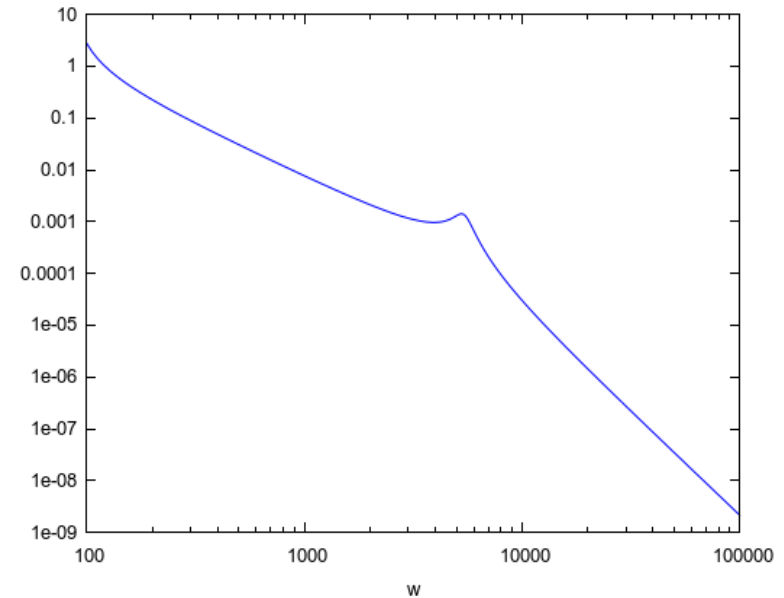
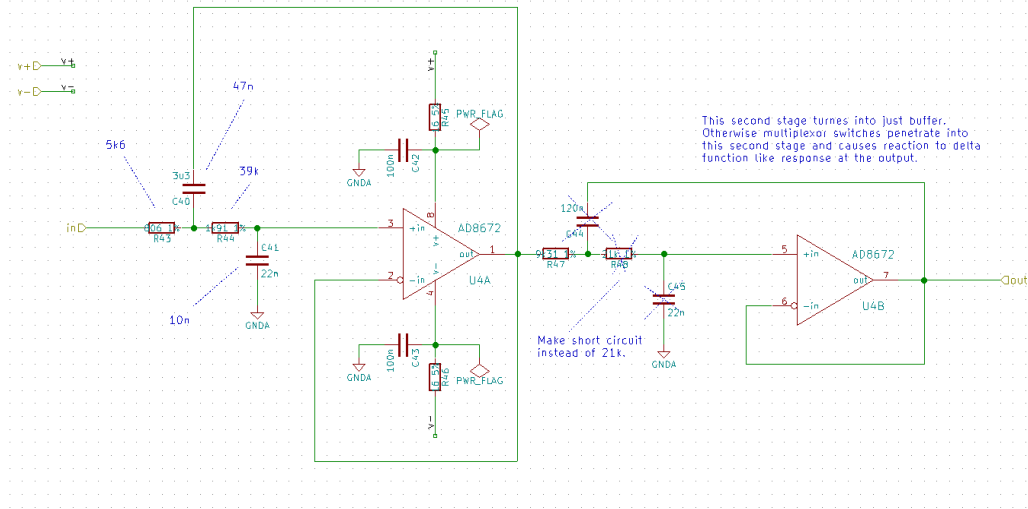
3rd party bipotentiostat control module.



- Mostly used **5%** or **1%** resistors
- Built in **mcp3063** based **boost 5V** to **12V** and **inverting 5V** to **-12V** switching sources.
- On board **temperature sensor** for calibration purposes.

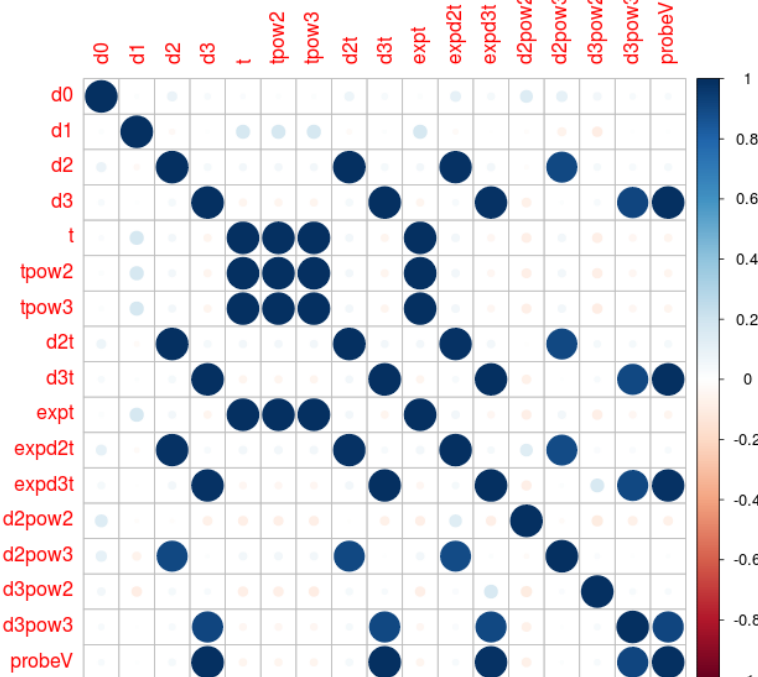
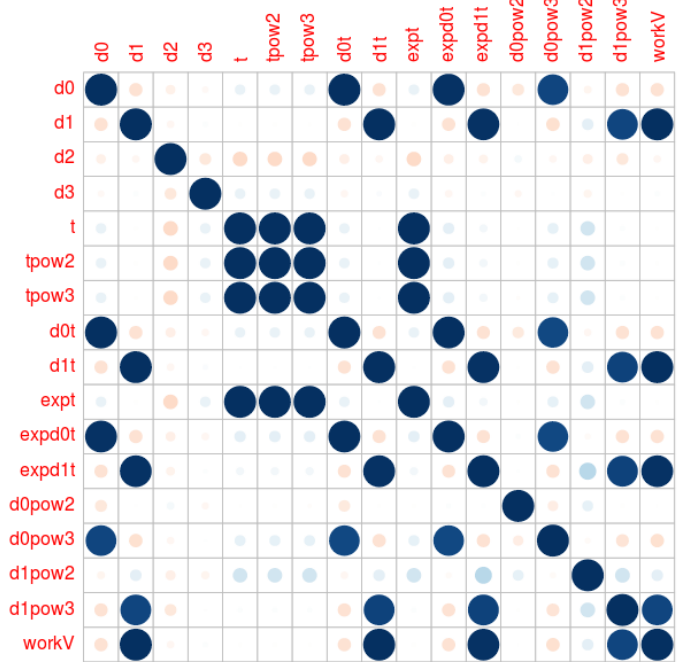
- Calibration setup was equipped with potentiometers and a heater under near the device for statistical data acquisition.

3rd party bipotentiostat control module.



- Input filter design and Bode plot.
- Based on Chebyshev filter but with lowered response near cutoff frequency to reduce oscillations caused by step response.

3rd party bipotentiostat control module.



- Examples of parameters and outputs cross correlations

- After gaining statistical data went to higher dimensional space by involving first few powers of each parameter, exponentiation and a few parameter products.
- Used this data set to find the best mean out of sample error by applying linear regression to all possible parameter combinations.
- Obtained formulas and dropped parameters which didn't improve match better then chosen margin (10^{-5})

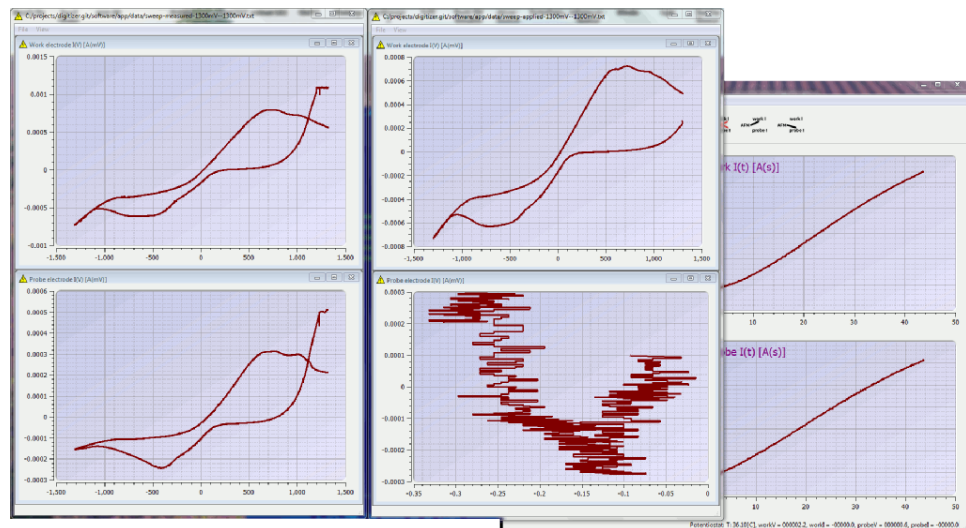
3rd party bipotentiostat control module.

- Formulas obtained for data readings conversion

$$V_{out} = a_1 * dac_1 + a_2 * dac_2 + a_3 * dac_1 * T + a_4 * dac_2 * T + a_5 * T^3 + a_6$$

$$V_{in} = a_1 * adc + a_2 * T + a_3 * T^2 + a_4 * T^3 + a_5$$

3rd party bipotentiostat control module.



- PC based software for driving control module.



- Just nice looking electrochemical cycling I(V) curve obtained during system testing.