Calibration and Regulation of BrainScaleS' PowerIt Subsystems

Patrick Nisblé

August 18, 2018

Monitoring System status and regulation parameters within a complex System such as BrainScaleS, provides multiple critical points, which in case of misbehaviour can result in problems within the complete system. To reduce the eroneous data created within BarainScaleS, the PowerIt Board, one of its submodules, was upgraded and received a software ovehaul, containing calibration for the on board measurements and regultion capability for its most critical output terminal.

Monitoring System Status und Regulations Parameter innherhalb eines komplexen Systems, wie etwa BrainScaleS, entahlten kritische Punkte des Systems, welche im Flalle eines Fehlers in Promblemen mit dem kompletten System resultieren können. Um die innerhalb von BrainScales erzeugten fehlerhaften Daten zu reduzieren, hat das PowerIt Submodul ein Firmware Upgrade erhalten, diese enthält nun Kalibrationen für die Board eigenen Messungen, sowie die Fähigkeit die Systemkritischen Ausgänge zu regulieren.

Contents

1 Introduction

1.1 What is the BrainScale System?

The BrainScale Wafer System [1], developed and used in the electronic visions Group at Heidelberg University is a neuromorhic hardware implementation.

For this thesis the following core components are of importantance:

- the mixed-signal ASICs, named HICANNs, structured in packs of 8 into "Reticles"
- the Control Units for Reticles, short CURE Boards
- the analog breakout Boards, AnaB for short
- and the power supply, called PowerIt.

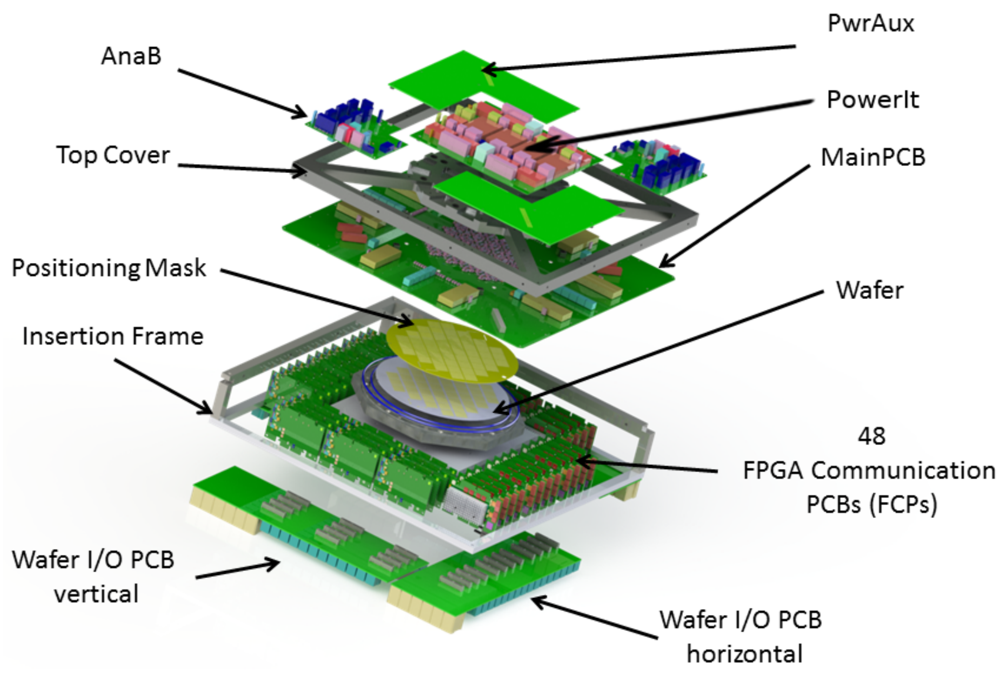


Figure 1.1: The BrainScaleS wafer-scale hardware system, marked are the main components comprising a single wafer system. [2]

1.2 About the PowerIt Subsystem

The main subject of this thesis is the PowerIt board (Figure 1.2). It functions as power supply inside of the WaferScale System (Figure 1.1). In which it is providing the Wafer with 1.8V and the FPGAs with 9.6V. Its maximum rated Powerdraw is 2kW. [3]

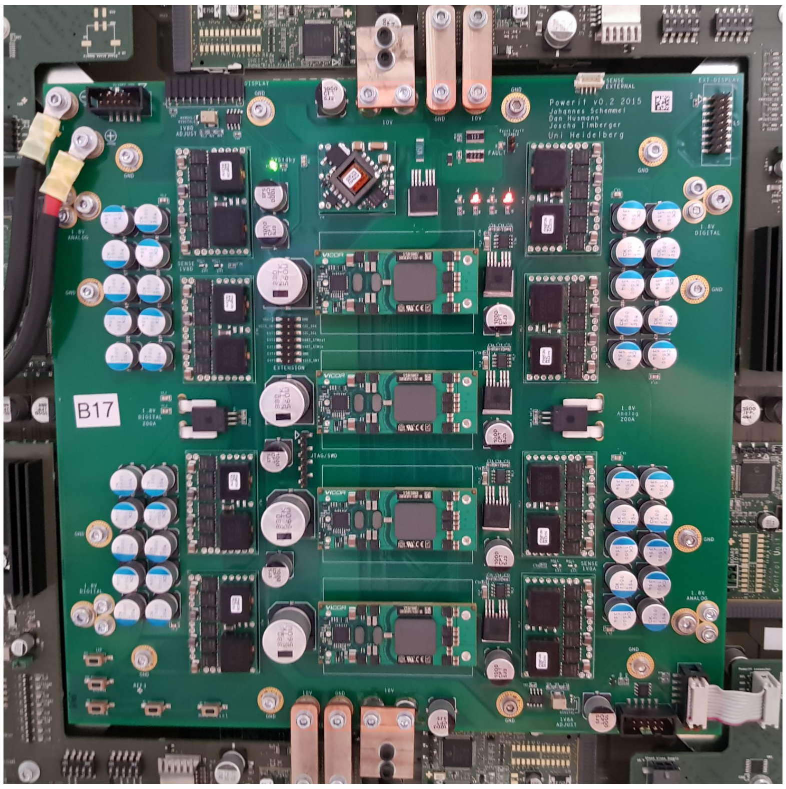


Figure 1.2: PowerIt Board, top view, receiving 48V as input (top left) and outputting 9.6V (top and bottom) as well as 1.8V (analog: top left, bottom right; digital: top right, bottom left)

The "Brain" of these PowerIt Boards is a STM32 Chip[1] which runs a custom Firmware based on ChibiOS [5].

[1]STM32F405RGT [4]

1.3 Intentions of this Work

- Upgrade the PowerIt Firmware to be able to calibrate all on board measurements, voltages and currents.

  This requires the Firmware to handle calibration changes on the fly.

- Provide a communication interface for changing those parameters at runtime.

  Assuming v2 of the communication Protocol (PItCOMM) we require write access to all coefficients, defining the degree of polynomial is done at compile time.

- Provide access to more parameters within the PowerIt while unifying the protocol used.

  Collect all writable, readable and static parameters in a single interface using PItCOMM v2, creating a mapping for reference to those values.

- Calibrate the onboard measurement circuits, using a database containing uniquely mappable values.

  Characterizing the circuits, providing a default for fallback and a Database file, readable by the BrainScaleS Monitoring System and the Person calibrating each Board.

- Provide a regulation mechanism for stable, modifiable output at its endpoints.

  The connection between PowerIt and HICANNs can be seen as a non trivial Resistor, which requires the 1.8V output to regulate based on experimental data.

2 Theory

This Chapter will be discussing the fundamental principles used in the experiments. These will contain the circuis and their respective Equations as well as component behaviour as specified in their respective Datasheets by their Manufacturer

2.1 Hardware Component Behavior

Before discussing the experimental results it needs to be clear what circuitry is used in these experiments and what behavior we expect. Keeping in mind, that these values are purely theoretical and will most likely not be exactly the same as those found in actual hardware.

Each of the three voltage regimes that will be observed on the PowerIt Board, 48V 9.6V and 1.8V, has a voltage- and in the cases of 48V and 1.8V also a current-measurement circuit. Additionaly we have a temperature sensor built into the STM32 Chip.

2.1.1 48V Input Voltage

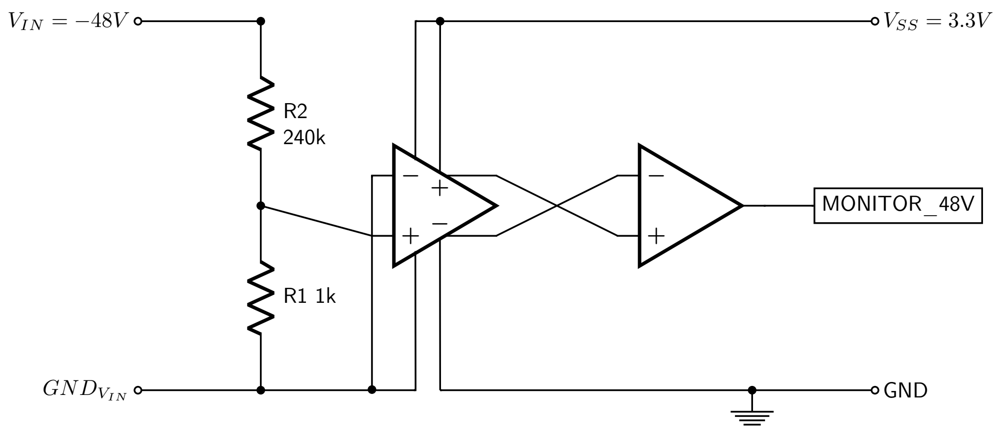


Figure 2.1: Circuit for measuring the 48V input Voltage, consisting of input potential (left), two resistors as voltage divider, one full differential isolation amplifier (full Diff Op Amp, left), one operational Amplifier (right), output voltage as well as the connection to the STM32-Chips input pin (right)

The circuits for measuring input Voltage and current are the most complex, because for Voltage measurement the circuit needs to

- divide our input voltage into a usable potential range
- decouple the input from our signal potential
- amplify the voltage, to be in the Chips Voltage range of 0–3.3V

The already implemented Cicuit can be seen in Figure 2.1. It consists of a 1:240 Voltage Divider, a full differential isolation amplifier taking in the $\approx$ 200mV (nominal voltage range), and amplifying it by a factor of 8 ($r_{\text{diffOpAmp}}$ [6]). It is also decoupling the input and output voltages, so our 48V and 3.3V circuit parts are electricly insulated. The remaining operational amplifier provides futher amplification by a factor of 1.1 (r_{OpAmp})

This circuit results in the following equation for calculating the input voltage from a pin voltage:

$$\begin{aligned} & V_{\text{48V in}} \cdot \frac{R_1}{R_1+R_2} \cdot r_{\text{diffOpAmp}} \cdot r_{\text{OpAmp}} = V_{\text{48V pin}} \\ \Leftrightarrow \quad & \frac{V_{\text{48V pin}}}{r_{\text{diffOpAmp}} \cdot r_{\text{OpAmp}}} \cdot \frac{R_1+R_2}{R_1} = V_{\text{48V in}} \end{aligned} \tag{2.1}$$

2.1.2 48V Input Current

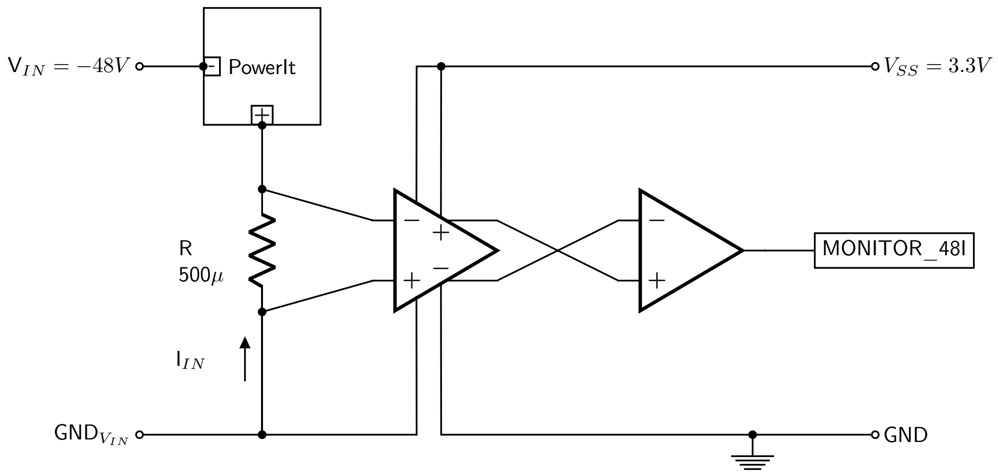


Figure 2.2: Circuit for measuring the 48V input Current, consisting of the powerit Input Circuit, one shunt-resistor, one full diff isolating Amplifier, one operational amplifier, output potential, as well as the connection to the STM32-Chip input pin

In case of the current measurement circuit we require the following:

1. use a shunt resistor, with minimal heat dissipation

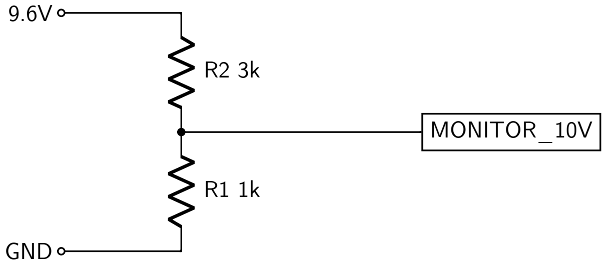
2. still providing a good resolution also within the Chips Specifications

Here we use the same Amplifiers and so we can use the following equation for our input current:

$$\begin{aligned} I_{\text{48V IN}} \cdot R_{\text{shunt}} \cdot r_{\text{diffOpAmp}} \cdot r_{\text{OpAmp}} &= V_{\text{48I pin}} \\ \Leftrightarrow \quad \frac{V_{\text{48I pin}}}{R_{\text{shunt}}} \cdot \frac{1}{r_{\text{diffOpAmp}} \cdot r_{\text{OpAmp}}} &= I_{\text{48V IN}} \end{aligned} \tag{2.2}$$

2.1.3 9.6V Output Voltage

This Circuit consists of a a simple 1:4 Voltage Divider.



And the following equation shows that:

$$V_{\text{10V PIN}} = \frac{V_{\text{10V IN}} \cdot R_1}{R_1 + R_2} \tag{2.3}$$

2.1.4 1.8V Output Voltage

until now the voltages and current could only be measured, now the mechanism for setting a resulting Voltage at the 1.8V Terminals is known. Figure 2.3 is the circuit for generating 1.8V, it consists of a power module and a resulting resistance at given pind, defined by R_{series}, $\text{R}_{\text{parallel}}$ and R_{pot}, whose job is to set the output to a given voltage of around 1.8V, but we can vary R_{pot}, because this resistance is set vie a digital potentiometer.

The in Figure 2.3 used 1.8V Converters have a characteristic formual [7], and the in this circuit used Potentiometer is a linear 10kΩ Rheostat resulting in the following equations:

$$R_{potentiometer} = P_{val} \frac{10k\Omega}{256} \tag{2.4}$$

$$\begin{aligned} R_{SET} &= 1/\left(\frac{1}{R_{potentiometer}} + \frac{1}{R_{parallel}}\right) + R_{series} \\ &= \frac{R_{\text{potentiometer}} \cdot R_{\text{parallel}}}{R_{\text{potentiometer}} + R_{\text{parallel}}} + R_{\text{series}} \end{aligned} \tag{2.5}$$

$$V_O = \frac{30.1k\Omega}{R_{SET} + 6.49k\Omega} \cdot 0.7V + 0.7V \tag{2.6}$$

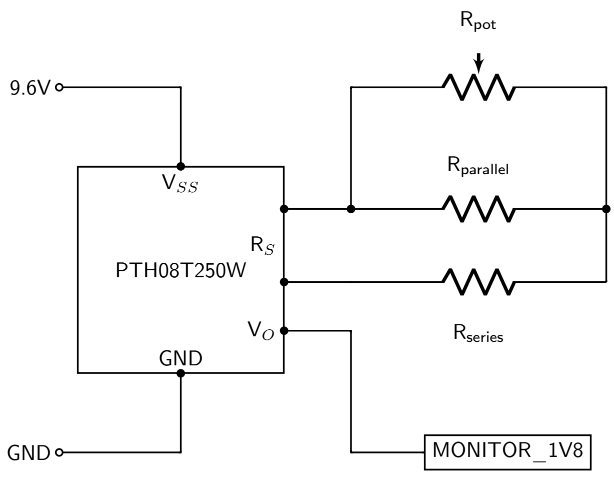


Figure 2.3: Circuit for generating a changable Output Voltage, consisting of the DC-DC Converter, a resistor chain, supply voltage (left) and resulting voltage (right)

This equation is in contrast to all previous behavior models not of a linear nature but proportional to $\frac{1}{x}$ as visualized in Figure 2.4

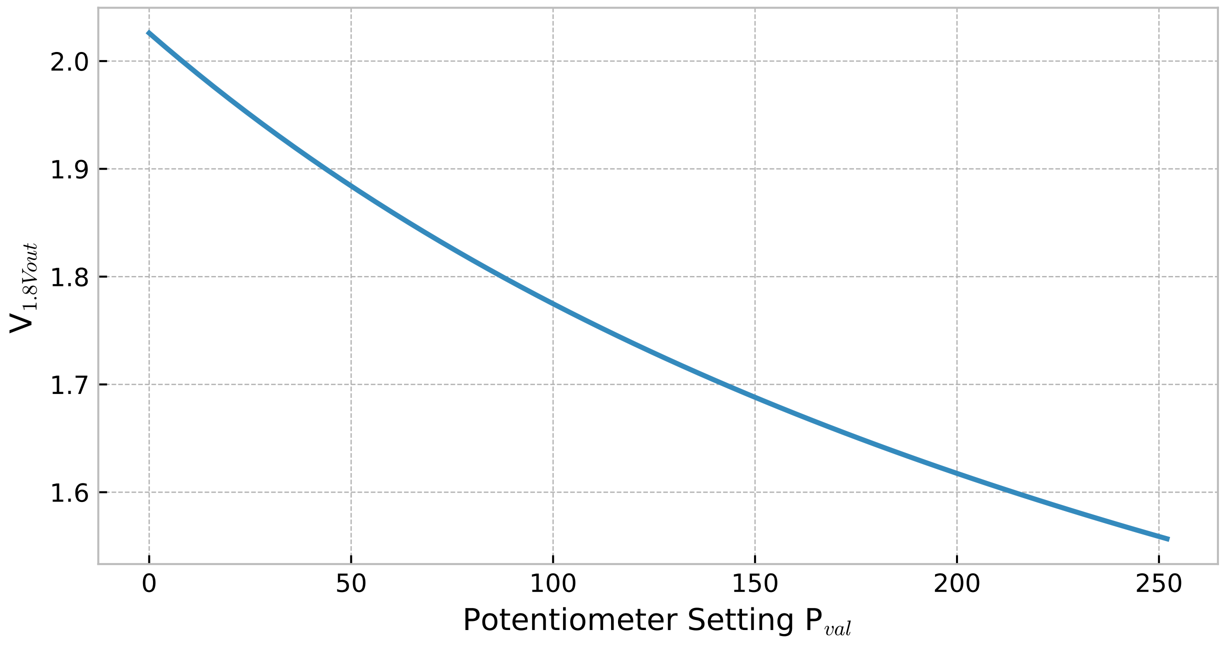
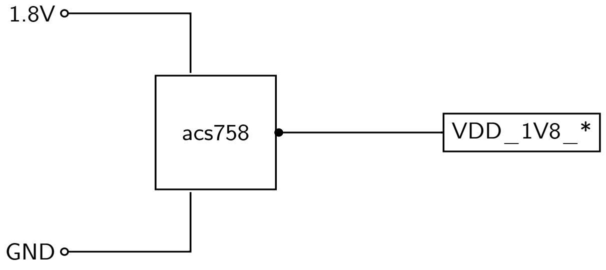


Figure 2.4: Expected behavior of our output voltage by setting the potentiometer

2.1.5 1.8V Output Current

The circuit for measuring current is also quite straight forward. It consists of a current sensing IC, which is Hall sensor based, and is in series with the wafer connection. One each for digital and analog.



2.2 ADC Calibration

As mentioned beforehand, the actual hardware will differ in behavior from its theoretical counterpart. These discrepancies will in fact differ by more than we can accept and use without countermeasures. Therefore we can say that all signals with a signoficant difference of behavior ($\approx 5\%$) will need to be corrected.

To calibrate these readouts we need to employ some simple actions.

2.2.1 serial ADC readout

While the measurements done by the STM32-Chip are using a 12bit ADC, there are not enough of these inside the chip to be able to completely parallelize the measurements, so only one ADC will be switching between all connected pins. This Behavior can be problematic in regards to measuring accurately. The timing used to measure a single line can be programmatically set from 3 up to 480 clock ticks[1]

2.3 1.8V Output Regulation

For Regulting the Output the method used is a numerical one, we calculate the voltage wanted at the putpu terminal and then we calculate a potentiometer setting, which changes the voltage produced (see Figure 2.3). The second part is already done beforehand and then available as lookup table to the firmware. On the other hand, to calculate the voltage to ouput, it is necessary to classify the connections between the PowerIts ouput terminals and the reticle.

[1]this clock is the internal adc clock, with a frequency of

2.3.1 Power Wafer

To test the 1.8V Regulation the so called Power Wafer is going to be used, it bahves similarly to a in BrainScales used “fuctional” Wafer module. But it is fundamentally different, as it cannot be used for computation, but only to test for voltages and currents. Its internals behave like switchable ohmic resistors, which provides us with a maximum powerdraw per Reticle of what is possible inside a usable wafer module.

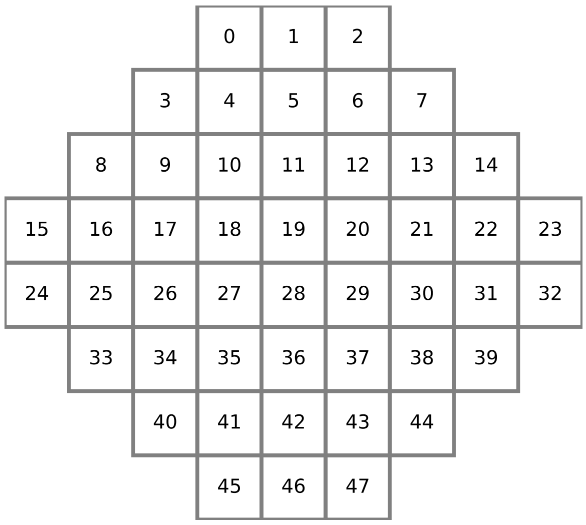


Figure 2.5: reticle diagram of a wafer in BrainScaleS, orientation as used in software and to better visualize connections on the wafer. All 48 Reticles are shown, all numbered from top left to bottom right

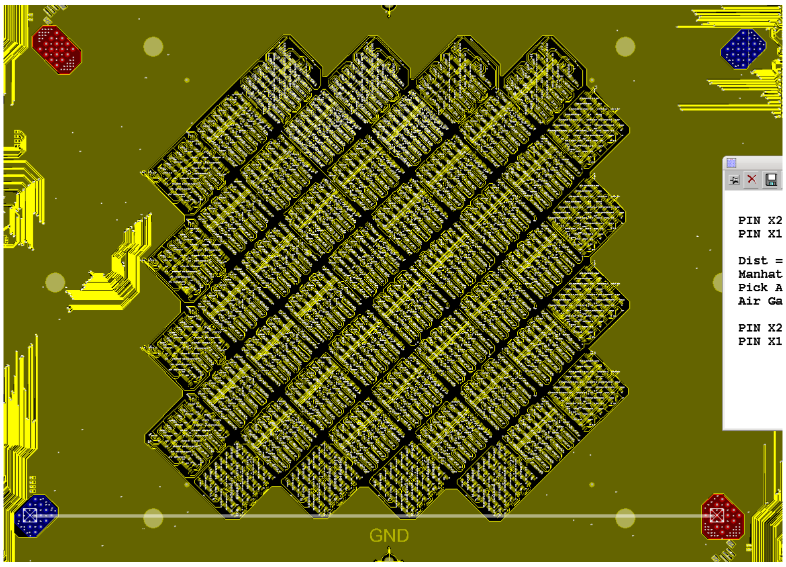


Figure 2.6: part of the mainpcb on which a wafer is placed, in its realworld orientation (rotated 45° from Figure 2.5), visible are the 48 Reticles and two terminals each for 1.8V Digital (blue) and Analog (red)

It has the same layout as its system counterparts and each of the 48 Reticles can be accessed, digitaly as well as electricaly. Each Reticle is connected to its corresponding CURE Board, which can read voltages of each reticle, right after the PowerFETs, reponsible for switching on power to a Reticle (switches in Figure 2.7)

Another specialization of the Power wafer is, that all reticles voltages are connected directly to pins on the Analog Readout Boards [8]. There it is possible to measure a voltage, which is the one after the load resistors in Figure 2.7

2.3.2 Simple Wafer Resistance Model (SWRM)

The circuit in Figure 2.7 can be used, as a first step, to describe the connections powering Reticles inside a wafer.

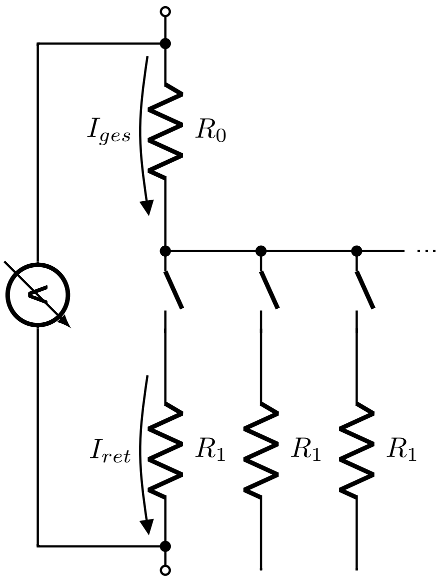


Figure 2.7: model of the to measure resistances and their currents, R_0 describes the resistance of a connection between the PowerIt Output and up to the FET, while R_1 is a Resistance between FET and Reticles. The measurement is done between Output Terminals on the PowerIt and pins on a Analog readout board

SWRM allowes for two fixed resistance values and their respective currents. The current flowing through R_1 will be either 0 or a constant current I_{ret}. The current through R_0 will change depending on the number of reticles that are powered n_{ret}

$$I_{ges} = n_{ret} \cdot I_{ret} \tag{2.7}$$

Therefore the voltage Differential as measured by a Voltmeter (see Figure 2.7) can be described with Equation 2.8

$$\begin{aligned} V_{dip} &= V_{R_1} + V_{R_0} \\ &= R_1 \cdot I_{ret} + R_0 \cdot I_{ges} \\ &= I_{ret} \cdot (R_1 + R_0 \cdot n_{ret}) \end{aligned} \tag{2.8}$$

Combining Equations 2.4, 2.5, and 2.6, we gather Equation 2.9. This equation is a reversed function of the one used in Figure 2.3

$$P_{val} = \frac{R_{par}\left[\left(\frac{0.7V \cdot 30.1k\Omega}{V_O - 0.7V} - 6.49k\Omega\right) - R_{ser}\right]}{R_{par} + \left(\frac{0.7V \cdot 30.1k\Omega}{V_O - 0.7V} - 6.49k\Omega\right) - R_{ser}} \cdot \frac{256}{10k\Omega} \tag{2.9}$$

inside the code used for Regulation, Equation 2.9 will be used to create a lookup table, while Equation 2.11 will be used at runtime, for which Equation 2.8 and 2.10 are needed.

$$V_{dip} = V_O - V_{off} \tag{2.10}$$
$$\Rightarrow V_O = I_{ret} \cdot (R_1 + R_0 \cdot n_{ret}) + Voff \tag{2.11}$$

2.3.3 Distance Wafer Resistance Model

Although the through SWRM gained functions are useful for determinig a theoretical Regulation procedure, it is still not near the realworld scenario. In a wafer, the distance between reticles and voltage connector (see Figure 2.6) are resulting in additional resistance, proportionally.

Therefore we adapt the DWRM after Circuit 2.8 in which each different Distance requires additional Resistors.

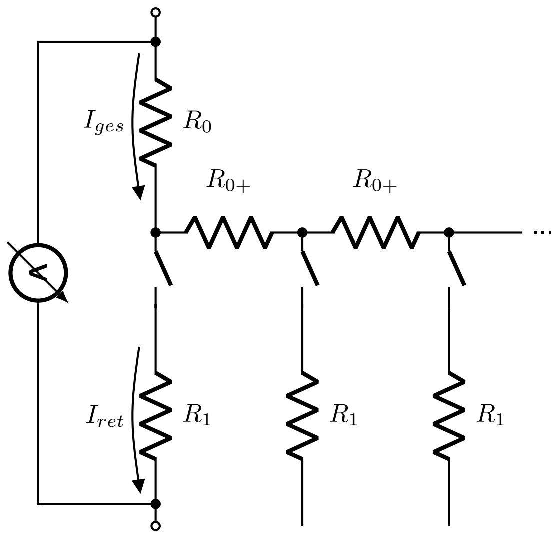


Figure 2.8: retpow2

With this model the voltage is now expected to change depending on the reticles distance instead of being the same. The distances inside a wafer are visualized in Figure 2.9

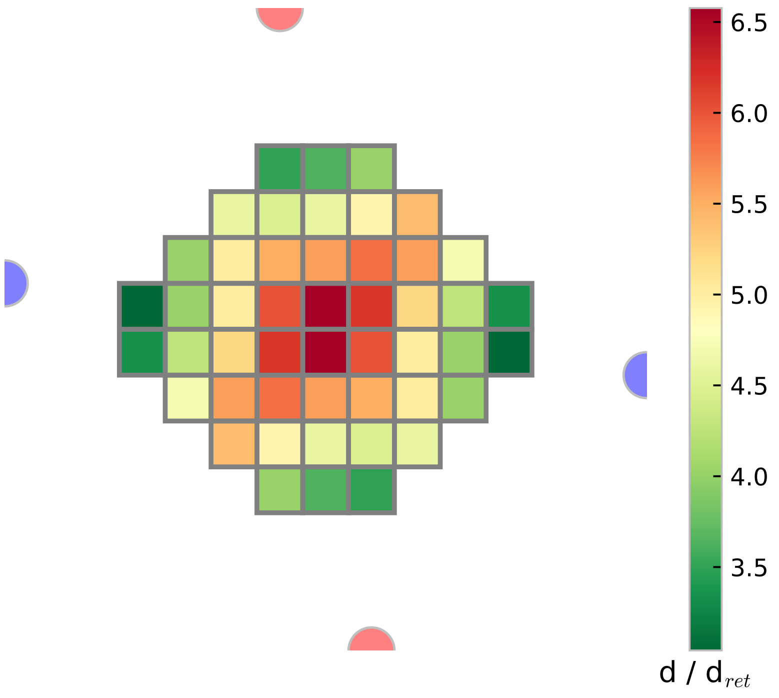


Figure 2.9: Distances of reticles to the nearest voltage supplying connection for DWRM, distance is in reticle-side length

3 Experiments

Now that the theoretical model sicomplete, experiments can be done to start checking that model and get results to use for in system components.

3.1 Characterization

The first experiments to run are the caracterization of hardware behavior. These will then result in a PowerIt Calibration, which later then can be used as basis for creating a regulation method.

3.1.1 sampling time

First up was selecting an optimal number of cycles for which the adc will probe a to it connected pin, like described in subsection 2.2.1.

In this case the uncalibrated measurement of input voltage was taken as example, and repeated with each of the possible 8 settings.

To be able to compare a reference voltage measurement was taken with an external Voltmeter. The resulting errors, from a set Voltage, can be seen in figures 3.1

Figure 3.1 contains the relative error of the measured voltage compared to the theoretical, set input voltages. therefore the reference measurements (yellow), taken with an external multimeter, are not at 0. Also shown are the calculated gain erors, in case of all 8 settings.

Important to note is the relative error in only the 0th case, here the `cycleTime`-Setting was set to 0 and therefore the smallest available sampletime of 3 Ticks. This excludes 0 a possible value to use. All other measurements are within errormargin of each other, and because a smaller timeframe is preferred, the best value to use is 1, resulting in a measuretime of 15 Ticks.

3.1.2 Voltages

Now that a sampleTime is chosen, it is possible to proceed with the voltage calibration measureemnts. Note, that Measuremts are expected to be less accurate, the more components are contained in their respective measurement circuit. Because small errors will accumulate and in e.g. the case of 48V's be amplified by a factor of 8.

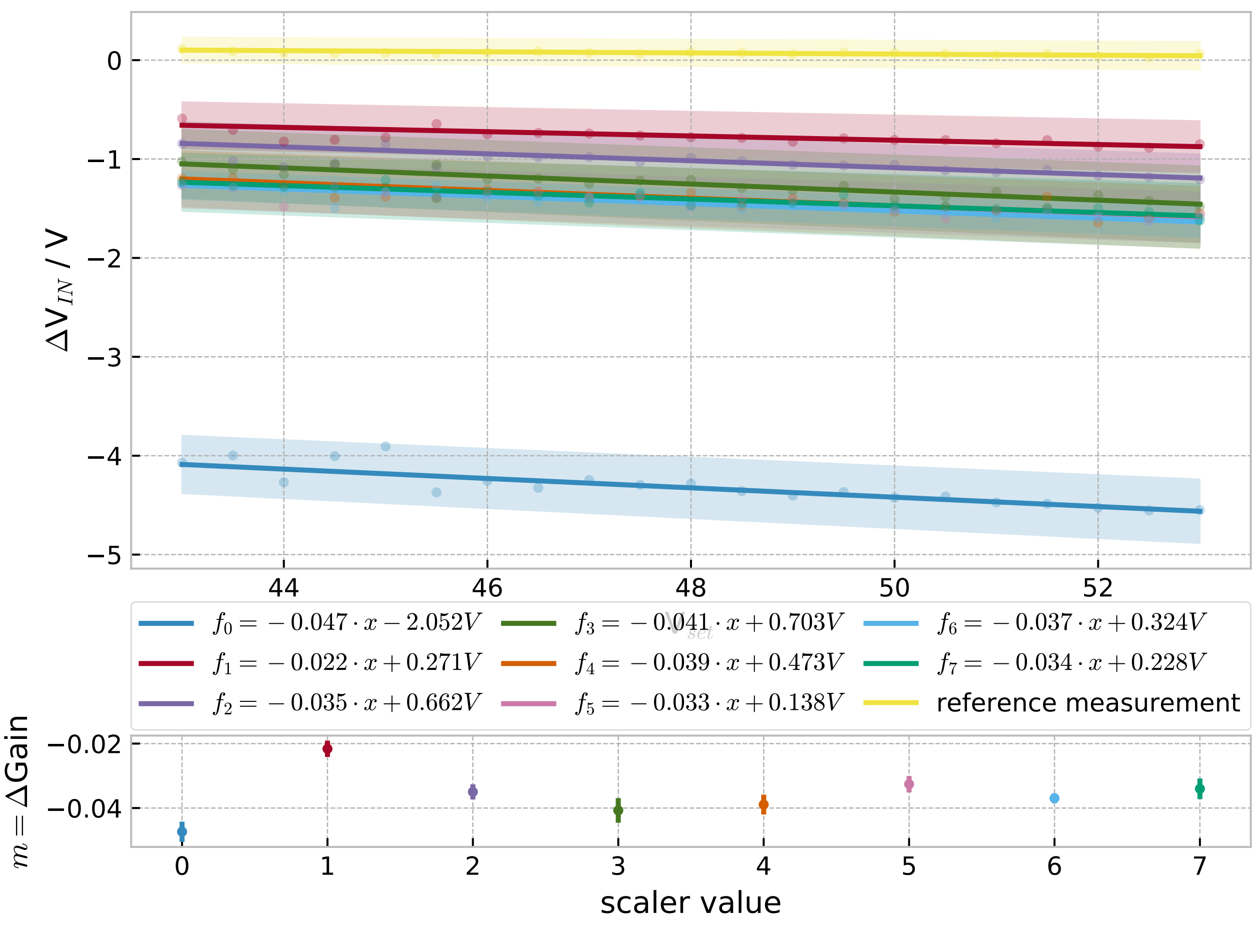


Figure 3.1: plotted difference from set input voltage, and fitted linearly, May 29th 2018, ≈32°C

48V Input

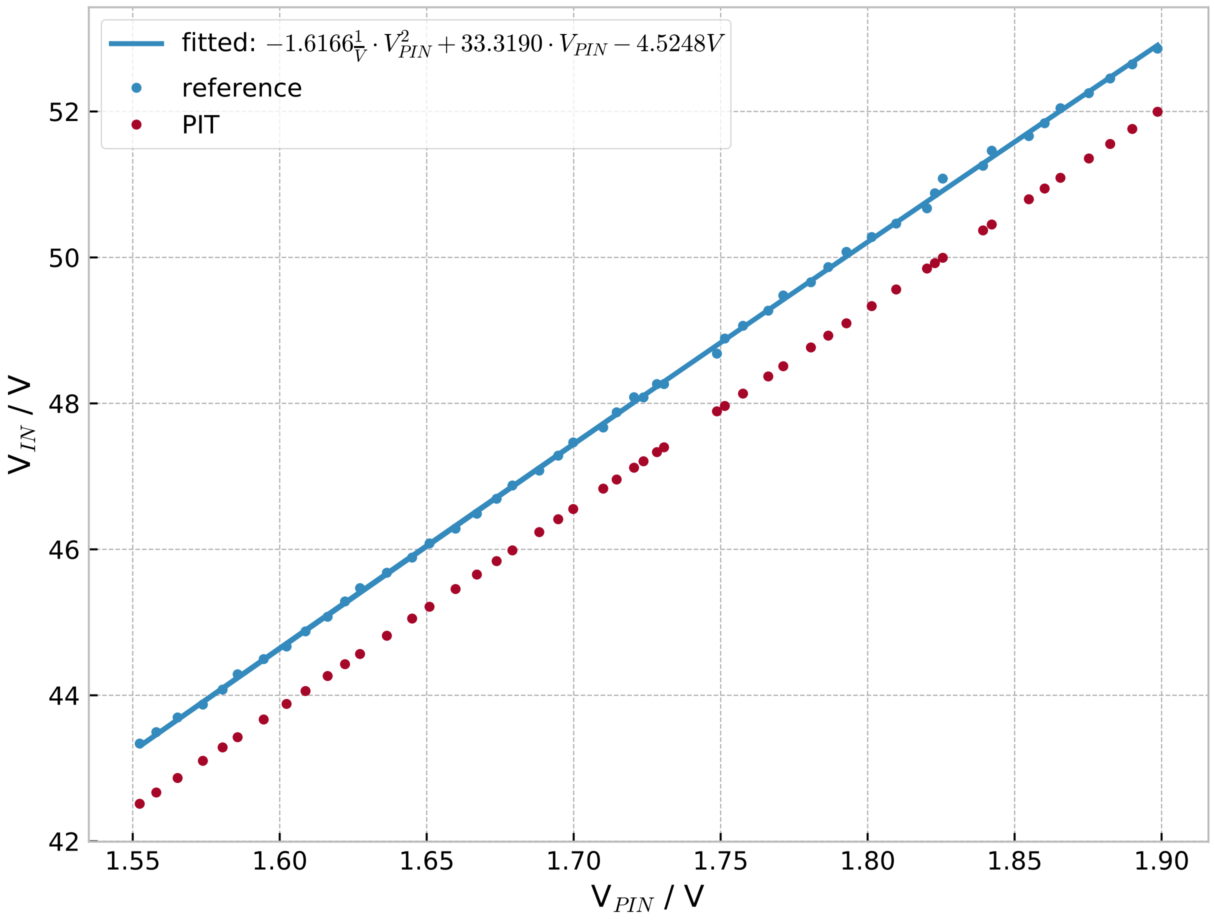


Figure 3.2: Calibration of input voltage, plotted are a external measurement and internal values, vs the recalculated pin voltage based on the internal value and used default function (default coefficients see Figure 4.2)

When looking at calibrating the input voltage (Figure 3.2), we can clearly see a relatively constant offset of ≈1V. In Figure 3.2 a polynomial fit of 2nd degree[1] is done and its coeffficients extracted (**??**, line 9). These coefficients not only show an offset, but also some deviation in the incline and curve from the default values.

[1]A Fit of second degree will be used in the complete calibration process

9.6V Output

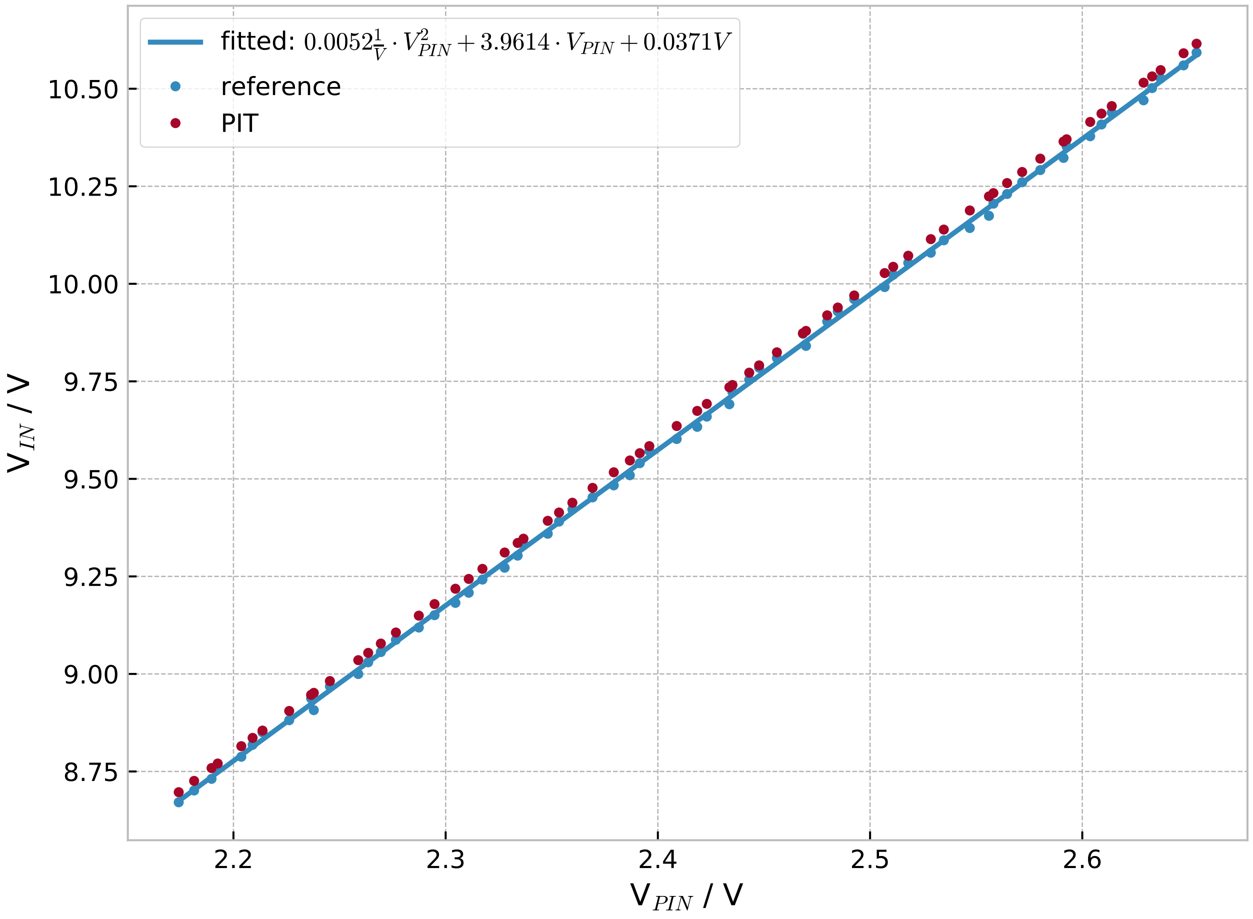


Figure 3.3: Calibration of 9.6V output Voltage, plotted are an external measurement and internal values vs the recalculated pin voltage based on the default coefficients (Figure 4.2)

The 9.6V Calibration, in contrast, shows only a slight deviation of the internal values and the reference measurement, which results in a list of coefficients (**??**, line 7), very similar to those set in the theoretical defaults.

This small difference is explained by the simple voltage division used as our circuitry, and no amplification, as for the input voltage circuit.

1.8V Output

The last Voltage to calibrate is divided into two domains, one for supplying the analog circuitry inside the wafer system, and one for the digital side. Each deliver between 1.5and 2.022V and each is settable by its own circuit (both as in Figure 2.3).

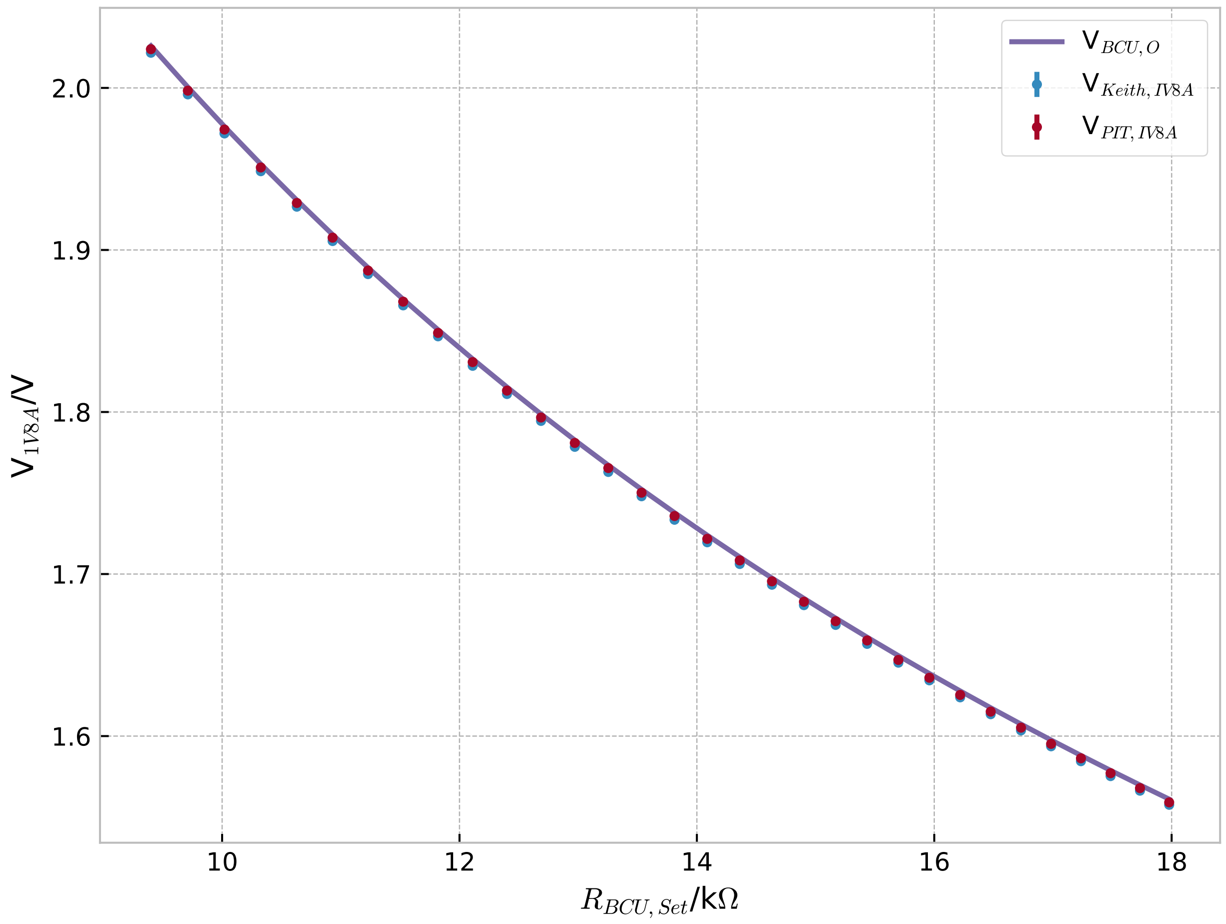


Figure 3.4: Calibration: analog 1.8V Output voltage, plotted are external measurement and internal values vs set resistance at the BCU Voltage Module.

Visualized in Figure 3.4 is the analog domains calibration, showing nearly no difference in board and reference measurements. Mostly due to direct connection between created voltage and the STM-Chips pin.

3.1.3 Currents

With now calibrated Voltages, the next step is to measure the behavior of the measuring circuits. Note that the 9.6V Output does in fact not have a include circuit for measuring its current draw, and that this number will be obtainable from all other (calibrated) measurements.

48V Input

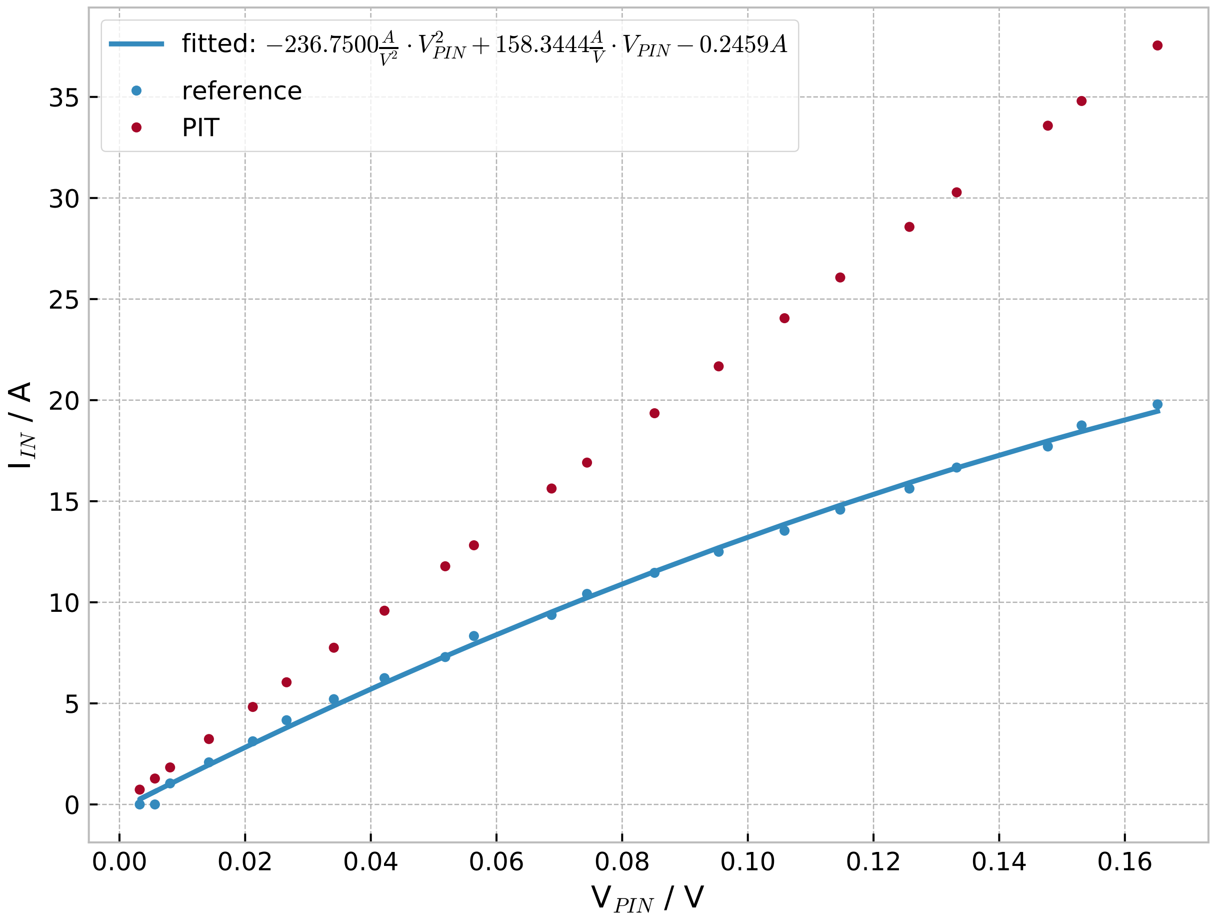


Figure 3.5: Calibration of input current adcs 21.06.2018

1.8V Output

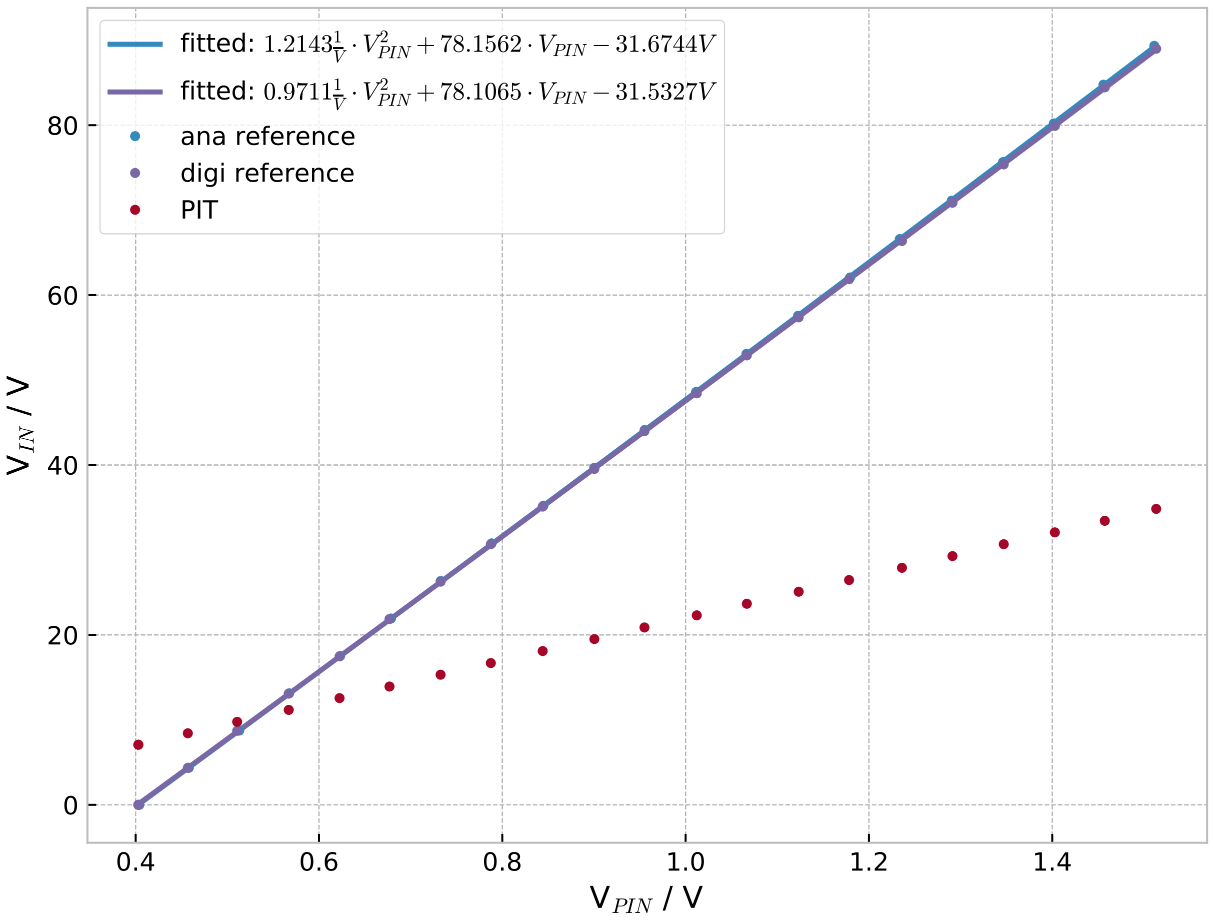


Figure 3.6: Pre Calibration Measurement of Output Current at the 1.8V Analog and Digital Terminal (2.7.2018)

3.2 1.8V Regulation

As Described beforehand the Output Voltages for both analog and digital can be adjusted to some degree and therefore we can compensate for the dropoff occuring between PowerIt Output Terminals and Reticles.

3.2.1 Characterization of Dropoff

Wanting to observe and characterize the voltage drop, first the connections between PowerIt and Reticles can be measured with the in Figure 2.7 described connections, which in actuallity are the PowerIT Terminal and corresponding analog readout pin on a Analog readout board.

IN Figure 3.7 a single reticles Voltage Dip for different Current Draws is visualized. A relatively linear trand and residuals of a trigonometric behavior can be observed, most likely the result of the inaccurately measureable currrent which is in this Figure done inside the PowerIt.

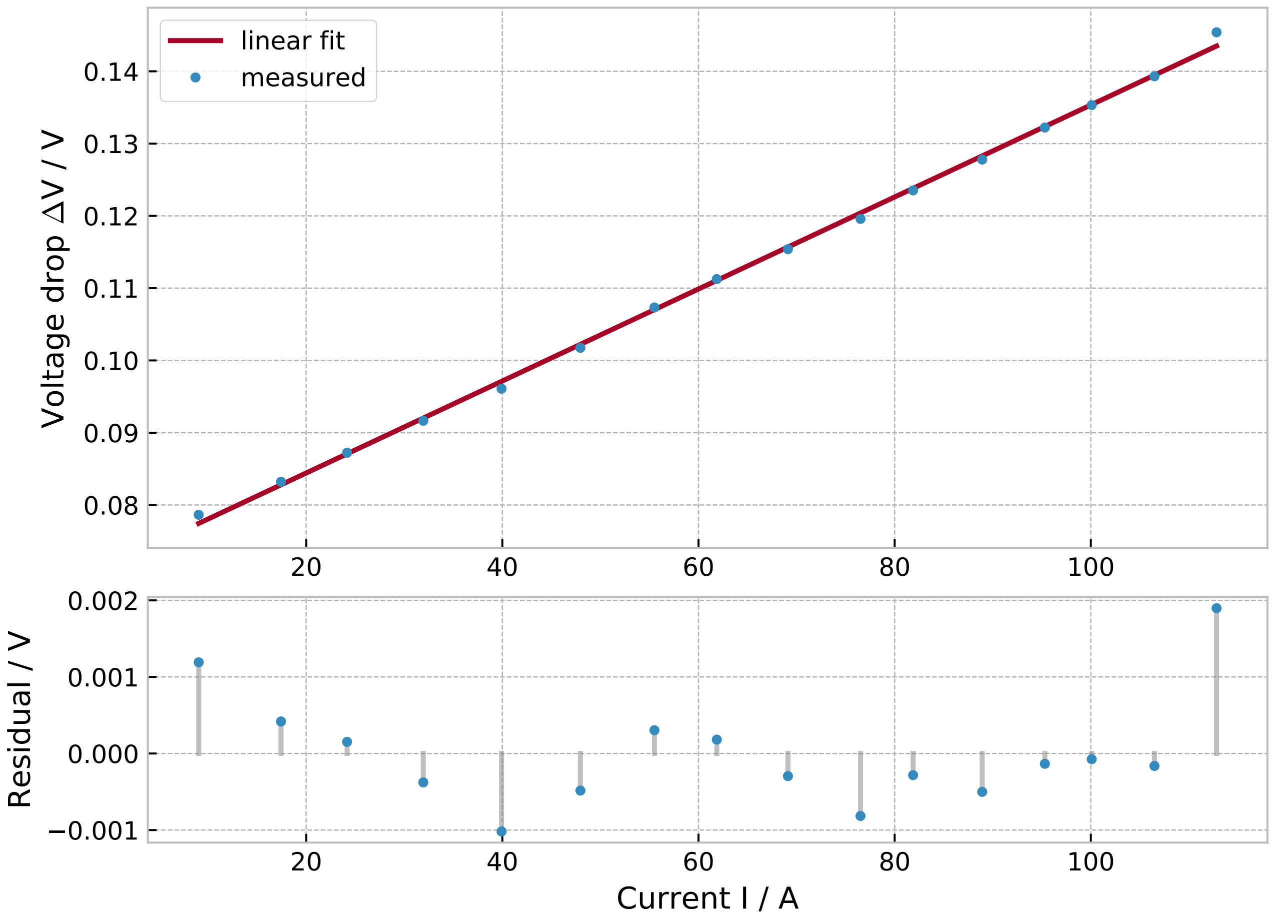


Figure 3.7: Voltage dip observed between PowerIt and HICANN, each point represents the state after enabling additional Reticles on the PowerWafer ()

This measuring circuit has some inconsitencies because of the 2nd degree polynimial fit, which is not the appropriate description fr the used compenents, but intead should be some function proportional to $\frac{1}{x}$, like the one described in Equation 2.9.

3.2.2 after Numerical-Correction

The initial approach is a numerical. Through derivation from Figures 3.7 and **??** we can plot a function which maps the measured output current to a corresponding potentiometer setting (Figure 3.8) for which the observed dropoff will be mitigated (or at least near that). Also important is that it is not possible to use non interger values for the potentiometer setting.

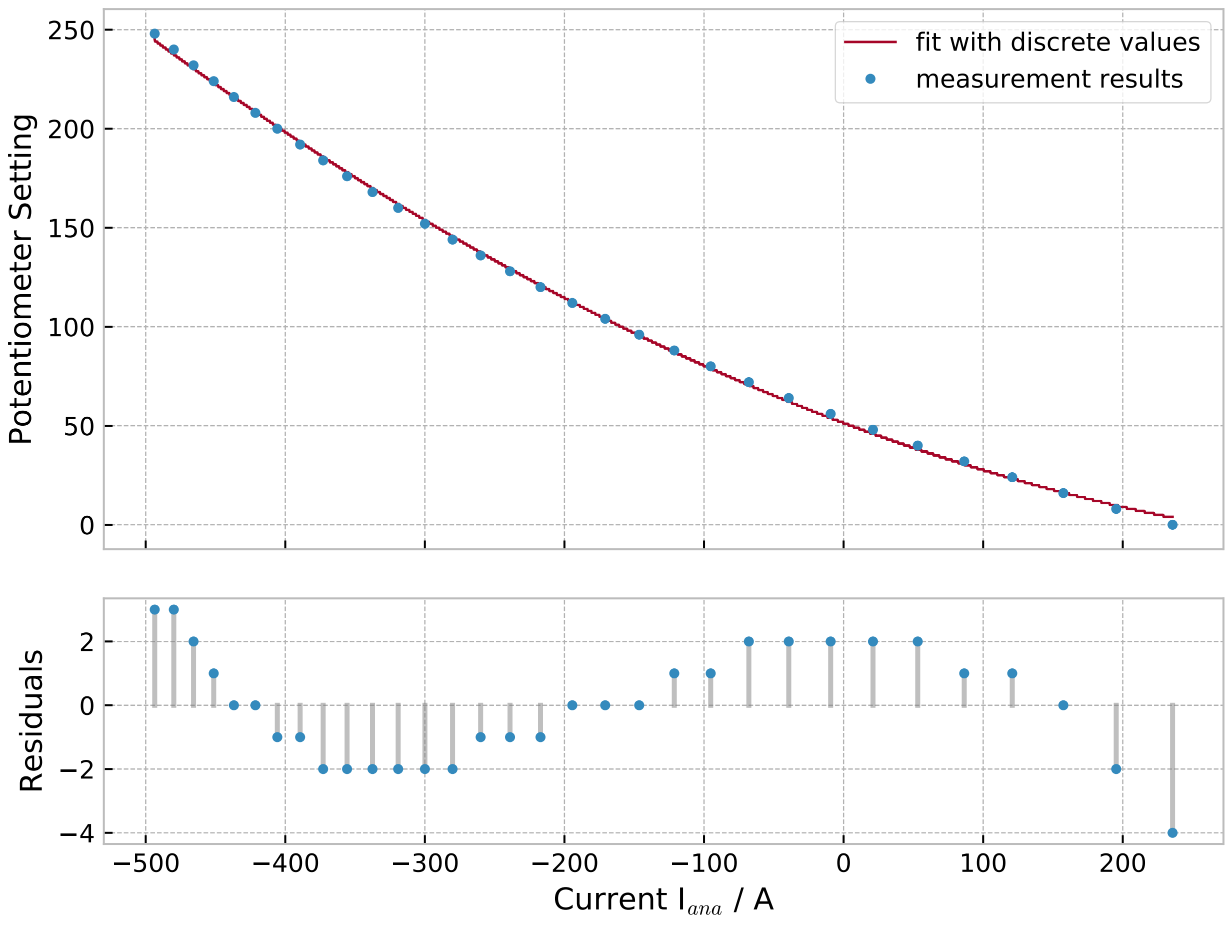


Figure 3.8: Potentiometer Setting (discrete integer), derived from ouput current (discrete floating point).

Fitting these values, with a polynomial of 2nd degree, we obtain:

$$P_{val} = \lfloor m_2 \cdot I_{ana}^2 + m_1 \cdot I_{ana} + m_0 \rceil \tag{3.1}$$

$$m_2 = 51.390262 \frac{1}{A} \tag{3.2}$$

$$m_1 = -0.263850 \frac{1}{A}$$

$$m_0 = 0.000258 \frac{1}{A}$$

Which is the numeraical solution if the only desired voltage on HICAN Chips is 1.8V. But if we want to change these, we need a more general solution.

Assuming the 2nd order Term to be small enough, we can assume a linear proportionality between the current and voltage:

$$I_{ana,eff} = I_{ana} - \frac{V_{out} - 1.8V}{c} \tag{3.3}$$

where c is obtained from the linear fit (incline) in figure 3.7

$$c = 71.6978 \cdot 10^{-3} \frac{V}{A} \tag{3.4}$$

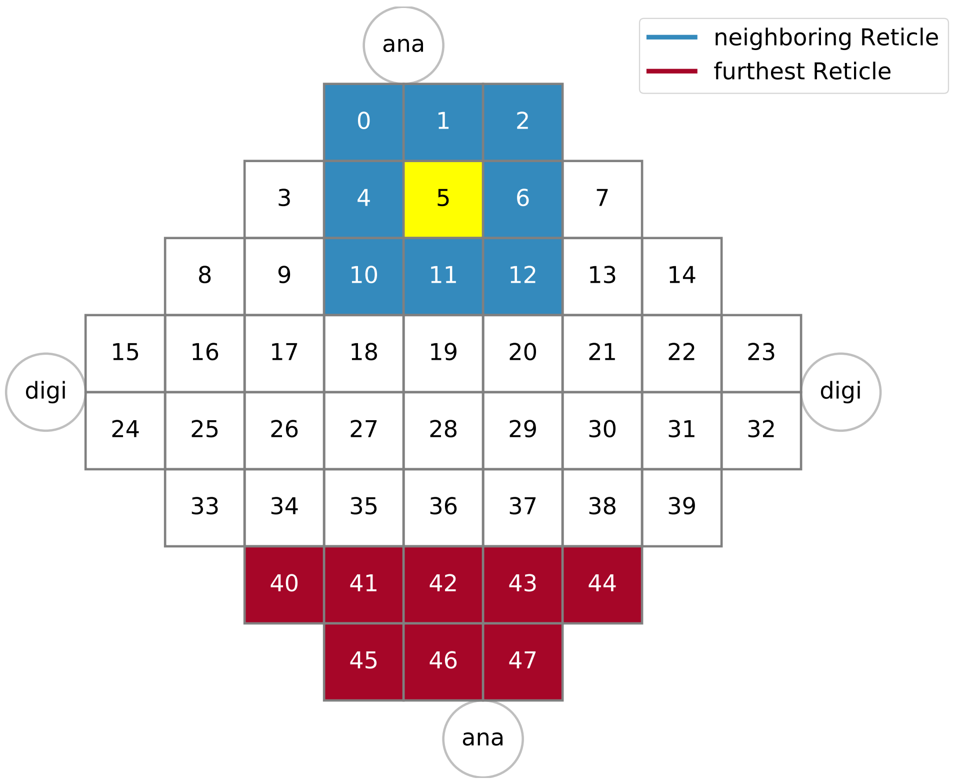


Figure 3.9: ret5wafer

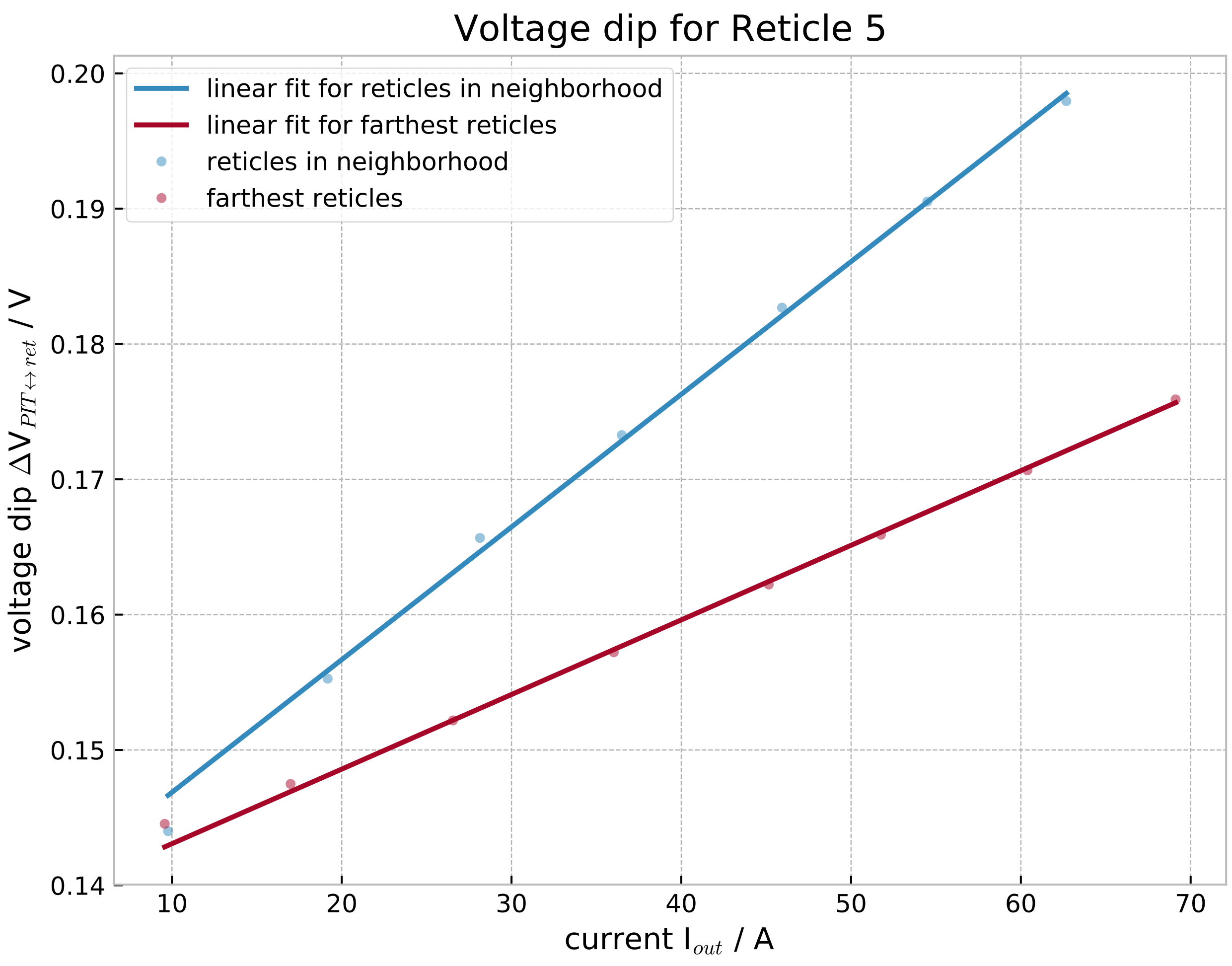


Figure 3.10: ret5

$$R_{0,\text{neighbor}} = (7.1278 \pm 0.1567)\,\text{m}\Omega \tag{3.5}$$

$$R_{0,\text{farthest}} = (4.0079 \pm 0.0537)\,\text{m}\Omega \tag{3.6}$$

$$R_{0,\text{mean}} = (1.4029 \pm 0.6233)\,\text{m}\Omega \tag{3.7}$$

$$R_{0,\text{mean,corrected}} = (1.4568 \pm 0.2644)\,\text{m}\Omega \tag{3.8}$$

$$R_{1,\mathsf{neighbor}} = (14.1708 \pm 0.1779)\,\mathrm{m}\Omega \tag{3.9}$$

$$R_{1,\mathsf{farthest}} = (14.2218 \pm 0.1503)\,\mathrm{m}\Omega \tag{3.10}$$

$$R_{1,\mathsf{mean}} = (17.4234 \pm 7.4256)\,\mathrm{m}\Omega \tag{3.11}$$

$$R_{1,\mathsf{mean,corr}} = (15.4141 \pm 2.4697)\,\mathrm{m}\Omega \tag{3.12}$$

3.3 Pitfalls

4 Results

This Chapter summaizes all of the resulting Workflow that has been developed during this Bachelor Thesis Work. Mainly the Firmware Changes compared to the state at wich it was left off after the previously taken Internship [?]

4.1 Firmware

The PowerIt Firmware was updated to allow for the Calibration Procedure (described in 4.2) and the Rgulation of its 1.8V output. That resulted in a new Version of the I2C Protocol between the Monitoring System and PowerIts.

4.1.1 PI2CProto v2

The new Commmunication protocoll is based on the old one, and while the from the Host send Commands are compatible with version 1 the returned Message is not. And just like in the old Version a Master can send a message to the PowerIt with the in the `command_t` struct described structure:

```
pi2c::command_t tosend { <crc>, 2, <CMD>, <optional_data0>,
 ↪  <optional_data1> };
```

Here the `<CMD>` can be `CMD_SET` for setting a value in the corresponding table (fig. 4.1), using `<optional_data0>` as address and `<optional_data1>` as value.

Reading a number of bytes from the table can be accomplished with `CMD_READ`, giving the address to start and number of bytes in that order.

While reading the complete Table is done with the `CMD_READALL` command.

THe `<crc>` byte is the CRC8 value of all following bytes inside `command_t`

4.1.2 I2C mapped Register-Table

When sending or reading values to or from the PowerIt, the address requireed is one of the 152 bytes documented in this table.

addr	name	type	size	perm
0x00	`onmask`	byte	1	rw
0x01	`offmask`	byte	1	rw
0x02	`anapot`	9bit	2	rw
0x04	`digipot`	9bit	2	rw
0x06	`polyFit.V48`	float arr	12	rw
0x12	`polyFit.I48`	float arr	12	rw
0x1e	`polyFit.V8`	float arr	12	rw
0x2a	`polyFit.V18`	float arr	12	rw
0x36	`polyFit.I18`	float arr	12	rw
0x42	`polyFit.T`	float arr	12	rw
0x4e	`sampleTicks`	byte	1	rw
0x4f	`V_out`	float	4	rw
0x53	`TEMP_SENSOR`	float	4	r
0x57	`EXT_AIN`	float	4	r
0x5b	`MONITOR_48V`	float	4	r
0x5f	`MONITOR_48I`	float	4	r
0x63	`MONITOR_8VBUS`	float	4	r
0x67	`MONITOR_8IBUS`	float	4	r
0x6b	`MONITOR_8V_0`	float	4	r
0x6f	`MONITOR_8V_1`	float	4	r
0x73	`MONITOR_8V_2`	float	4	r
0x77	`MONITOR_8V_3`	float	4	r
0x7b	`VDD_1V8_ANA`	float	4	r
0x7f	`VDD_1V8_IOUT_ANA`	float	4	r
0x83	`VDD_1V8_DIGI`	float	4	r
0x87	`VDD_1V8_IOUT_DIGI`	float	4	r
0x8b	`CommitHash`	float	4	s
0x8f	`CommitDirtyFlag`	byte	1	s
0x90	`STM32UUID`	96bit	12	s

Figure 4.1: memory mapping of the packed struct moved over i2c, `addr` is the address to use, `type` is the c++ type, `size` is in bytes and `perm` denotes read-writability. writability

4.2 Calibration

One of the goals of this Bachelor Thesis is to provide anyone required to calibrate a PowerIt Board with a comprehensive guide of, and inside into this process.

4.2.1 Calibration-Table

Of important note is that al calibration values of each PowerIt can be mapped via the naming and uuid scheme provided either by the corresponding stickers or the STM32-Chips internal uuid (accessible through address `0x8c`, see figure 4.1). There now also exists a global calibration Database, which will be loaded by the system on startup.

An example entry for each PowerIt entry looks like figure 4.2

Figure 4.2: example entry of pitdb.yaml

```yaml
---
uuid: 'default'
name: 'Bxx'
poly18i: [-3.0, 25.0, 0.0]
poly48i: [0.0, 227.27, 0.0]
poly10v: [0.0, 4.0, 0.0]
poly18v: [0.0, 1.0, 0.0]
poly48v: [0.0, 27.386, 0.0]
```

4.2.2 How to calibrate a PowerIt Board

The Calibration process is based on the PItSTOP Python scripts[1]. These are split into `server` and `aggregator`. While the Server is handling the translation between raw I^2C data, and the JSON formatted result, the Aggregator takes this JSON and calculates a calibration.

Using the script any one of the following Values can be tested and calibrated:

- Input Voltage (`pitstop.Aggregator.test_v_48()`)
- Input Current (`pitstop.Aggregator.test_i_48()`)
- 9.6V Output Voltage (`pitstop.Aggregator.test_v_10()`)
- 1.8V Output Voltage (`pitstop.Aggregator.test_v_18()`)
- 1.8V Output Current (`pitstop.Aggregator.test_i_18()`)

Setting up the Test Environment

The simplest way to setup your environment consists of cloning the PItSTOP Project onto your Client:

```
$ git clone https://url.to.pitstop
```

then substituting the `rsync` target:

[1] PItSTOP Repo

```
all:
    rsync --progress ./*.py /remote.url/
```

, to be your server (should be a RaspberyyPi connected to the PowerIt)

Running a Test

Runnig the test requires the following commands
Serverside:

```
$ python server.py
```

Clientside:

```
$ python aggregator.py
```

Now just following the instructions given, the selected test can be run:

```
Setting up calibration test for {}
Please be sure to:
 - connect the {} to the RaspberryPi running server.py.
 - connect the PowerIt to the RaspberyyPi as described in the
  ↪  documentation
 - and be sure to connect the {} to the {} Terminal.

Continue (y/N): y

What is the Name given to the connected PowerIt? [Bxx]: B05
```

4.3 Regulation

5 Outlook

Bibliography

[1] A. P. Davison, E. Müller, *et al.*, *HBP Neuromorphic Computing Platform Guidebook*. Human Brain Project, 2015-2018. EV Guide ↗.

[2] A. P. Davison, E. Müller, *et al.*, *HBP Neuromorphic Computing Platform Guidebook - BrainScaleS Hardware Configuration*. Human Brain Project, 2015-2018. `https://electronicvisions.github.io/hbp-sp9-guidebook/pm/pm_hardware_configuration.html`.

[3] P. Nisble, "Firmware upgrade des powerit (de)," 2018. ↗.

[4] STMicroelectronics, *STM32F405xx Datasheet (DM00037051)*. STMicroelectronics, 2016. STM32F405RG.

[5] G. di Sirio. ChibiOS RTOS [accessed 2018/01/11].

[6] TexasInstruments, *AMC1200/B Fully-Differential Isolation Amplifier*, 2015.

[7] Texas Instruments, *PTH08T250W Datasheet*, 2017.

[8] D. Husman, *Analog Pin Distributions*. electronic visions, Universtity of Heidelberg, 2016.