Improving Quantum Gate Fidelity Using Laser Intensity Based Post-Selection

Master Thesis of
Alexander Hungenberg

Mr Vlad Negnevitsky
supervisor

Prof. Jonathan Home
Trapped Ion Quantum Information Group
Institute for Quantum Electronics
ETH Zürich

30th June 2015

Contents

1 Acknowledgements

First of all I want to thank Prof. Jonathan Home for the many opportunities I got to participate in quantum computing research as part of this very successful group. I would also like to thank my supervisor, Vlad Negnevitsky a lot for his helpful hints and experience, interesting discussions and general support throughout the whole project. Additionally, many other people were involved with different parts. Especially Matteo Marinelli helped me a lot with the integration to Ionizer, while David Nadlinger was always available for helpful comments regarding the general system architecture and communication design. Dr. Daniel Kienzler and Robin Oswald kindly invested their time to proof read this thesis.

Finally, I don't want to skip on those who used the prototype and helped with bug hunting, like Frieder Lindenfelser and Lukas Gerster; or everybody else, who just showed an incredible amount of patience when changes to the code caused crashing of the lab control software.

I had a great time at the TIQI group.

2 Introduction

The idea of using quantum mechanics to build a new type of computer has now been around for a few decades and was originally proposed by Richard Feynman in 1982 [Fey82]. A few years later David Deutsch picked up this proposal and published a complete theoretical description of a quantum turing machine [Deu85]. Back in this time, the main reason for thinking about such machines was the complexity of simulating quantum physics on classical computers as written in the cited paper from Feynman. However, many physicists remained very sceptical if it would ever be possible to build such computers, as decoherence effects (section 3.5) and scaling issues are incredibliy difficult properties to deal with.

Later, in the beginning 90's Peter Shor, a mathematician and computer scientist, proposed a new kind of algorithm for quantum computers that allows to solve the prime-factoring problem with just polynomially growing complexity [Sho94]. This quickly resulted in growing interest for this topic and attracted more money for research, especially from intelligence agencies and defence industry. Just a year later Peter Shor also published a possible method to compensate for decoherence effects by forming a single, non-decohering logical qubit out of a few physical ones and applying his proposed error correcting algorithm [Sho95]. Afterwards, Andrew Steane then managed to reduce the required physical qubit count [Ste96]. These major breakthroughs finally marked the beginning of increasing research efforts and available money for quantum computing technologies all over the world. Many different possibilities for implementation have been and are still actively being investigated.

But also with the new error correcting framework it is still necessary to find very good physical qubits. They don't have to be perfect anymore, but still highly controlled. It is generally accepted that fidelities of more than 99.99 % (see section 3.6) are necessary to implement the error correction code reasonably well. In a strict sense this means that not more than 1 out of 5000 conducted experiments may return a wrong result due to failing laboratory equipment.

Achieving such a good stability is very hard, even in highly controlled environments. This is where the proposed post-selection scheme comes into play. The device which has been built as part of this thesis allows to observe and monitor laboratory equipment using high-speed analog-to-digital converters (ADCs) while conducting a single experiment and store the acquired data. In combination with the individual experimental results this can then be used to remove faulty results from the statistics which improves overall accuracy as further discussed in the result section of this thesis.

Additionally to this, the used hardware allows the long-term monitoring of DC analog and digital channels. This functionality is currently being used to check stability and correlations of slowly drifting systems in the laboratory (like cavities), as well as for displaying a quick status overview of the devices which are in operation and switched on.

3 Quantum Computing

Classical computers store information and state as an ordered system of multiple binary states ('bits'). They are usually represented as strings of zeros and ones. On top of these, gates are basically seen as black boxes, which act on one or multiple input strings and create a new output state in a predictable way, usually depending on the input. Choosing a finite amount of different gates wisely then allows to do any computation up to arbitrary precision.

However, this all comes at a cost, and the gate and/or operations count may increase a lot depending on the specific task. This can quickly grow to a level where building the required circuit or waiting for a classical processor to complete the requested calculation is not practical anymore.. This topic is extensively covered in computer science and in fact there are wide-spread applications which rely on the fact that there are problems which cannot be solved in a practical amount of time by any human or currently existing computer. Famous examples are modern encryption algorithms. On the other hand, problems like simulation of quantum systems remain mostly inaccessible so far, since the required computing power and amount of storage is huge, as soon as more than a few (10 - 20) quantum objects are involved.

At this point the field of quantum computation steps into place. It is a completely new and revolutionary approach to computing, as the the way information will be stored and accessed changes from classical to quantum systems. The biggest physical change will be at the level of a single bit. Where a classical two-level system (e.g. a transistor, vacuum tube, ...) can be described using a single number with two possible digits, for quantum objects this is no longer possible. The theoretical description has to switch from describing bits as scalars to qubits, which live in a two-dimensional Hilbert space. But there is more to quantum computing than just an increased amount of storage per two-level system. In fact, the most important resource will be the scaling of these building blocks when combining them to a bigger, global system. See section 3.2 for further details on this topic giving rise to an effect called 'entanglement'.

A few of the main challenges in building a quantum computer that have been identified so far are:

- Theoretically understand quantum systems. This part is already way ahead of experimental realizations as the mathematics is fairly simple and has been developed decades ago.
- To find a reliable system to implement qubits. Many different approaches have been investigated and are still part of active research. A clear winner is yet to be determined and the final architecture will probably be a hybrid of multiple technologies. Additionally, single- and two-qubit gate operations have to be implemented with high fidelities. The success with this varies depending on the chosen qubit technology.

Additional criteria have been elaborated by DiVincenzo in [DiV00].

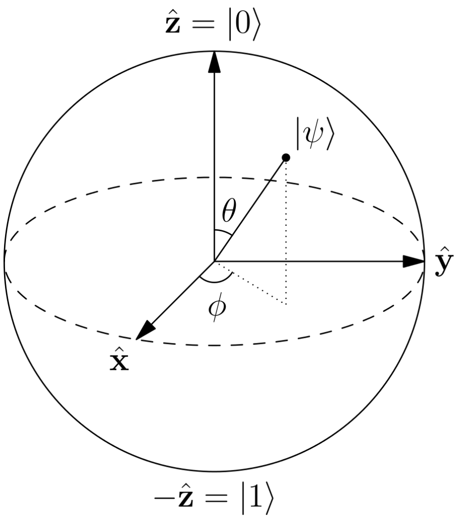


Figure 1: Bloch sphere representation of a two-level state [Glo]

3.1 Qubits and the Bloch Sphere Representation

The smallest logical unit in a quantum computer is called qubit and it is the quantum analog to a classical bit. As known from any introductory quantum mechanics course, states of quantum systems live in a Hilbert space. In case of a two-level system there exist two energy eigenstates ($|0\rangle$ and $|1\rangle$), such that an arbitrary pure state can be written in the following form

$$\alpha |0\rangle + \beta |1\rangle$$

where α and β are complex numbers that have to fulfil $|\alpha|^2 + |\beta|^2 = 1$ due to normalization constraints. Typical two-level systems found in nature are the spin configurations of spin-1/2 particles in a magnetic field. Further details of consequences and dynamics are extensively covered in many textbooks (e.g. [NC00]), but let us focus on a helpful way to visualize these pure states easily.

Due to the normalization constraint and the arbitrary choice of a global phase, the four parameters (real and imaginary part of both complex coefficients) of the arbitrary two level state written above can be reduced to two of them and represented on the so called Bloch sphere (fig. 1). This is a simple unit-sphere in 3-D, where each point on the surface corresponds to a specific qubit state. Using the standard definition of spherical coordinates, the inverse mapping from the sphere can be written as following.

$$|\Psi\rangle = \cos(\Theta/2) |0\rangle + e^{i\phi} \sin(\Theta/2) |1\rangle$$

This picture is very helpful to have in mind when thinking about the effect of certain gate operations.

3.2 Entanglement

Although this topic is not very relevant for the experiments covered in this thesis, it still captures some very important aspects of quantum computation, which is the reason a short introduction and example will follow in this section.

When looking at classical compositions of multiple subsystems, the global state is always 'just a collection' of the individual subsystem states. To put it the other way around, when there is precise knowledge of the global state, it is perfectly possible to write down the state of each classical subsystem. With quantum systems this is no longer the case. As an example, imagine a system composed of two individual qubits A and B. They are both in an arbitrary state which can be written as usual:

$$|\Psi\rangle_A = a\,|0\rangle_A + b\,|1\rangle_A \tag{1}$$
$$|\Psi\rangle_B = c\,|0\rangle_B + d\,|1\rangle_B \tag{2}$$

The composite system can be created by completing the tensor product

$$|\phi\rangle_{AB} = |\Psi\rangle_A \otimes |\Psi\rangle_B \tag{3}$$
$$= ac\,|0_A 0_B\rangle + ad\,|01\rangle + bc\,|10\rangle + bd\,|11\rangle \tag{4}$$

Now, we want to know the state of each qubit, given that the composite system is in the perfectly valid state (not normalized)

$$|\phi\rangle_{AB} = |00\rangle + |11\rangle$$

This should be an easy task, as we can use equation 4 and orthonormality of the basis states to obtain the following set of equations to determine a, b, c and d:

$$\begin{aligned} ac &= 1 & \qquad ad &= 0 \\ bd &= 1 & \qquad bc &= 0 \end{aligned}$$

However, it is easy to see that there is no set of solutions which satisfies all equations simultaneously. More formally speaking, it is impossible to write the global state as a tensor product of the individual systems. A possible interpretation of this result is, that composite quantum systems can enter so called 'entangled' states where subsystems lose their locality. Thus, when multiple qubits are involved, each basis state (in this case $|00\rangle$, $|01\rangle$, $|10\rangle$ and $|11\rangle$) must be parameterized independently for complete generality.

This fact is one of the important advantages and caveats of quantum computing at the same time, since the amount of information stored in a n-qubit system scales exponentially, instead of linearly as in the classical case. This is then of course also coupled to the previously mentioned, enormous amount of storage needed, when trying to simulate multi-qubit quantum systems on classical computers.

The given example (due to its small size) may not show this issue very obviously. However, imagine a 20-qubit system. To define the states of each individual qubit requires $20 \cdot 2 = 40$ stored complex coefficients in total, as in equation 1. But as shown before this won't be enough to classify the global state completely. Instead all coefficients of the global state have to be stored (eq. 4), which will amount to $2^{20} \approx 1000000$ complex numbers.

3.3 Trapped Ions

After the theoretical framework has been worked out, the next step is to find actual physical systems which behave sufficiently close to the two-level description. Many completely different approaches have been tried and quite a few are being actively developed in research projects nowadays, all with their own strengths and weaknesses.

Using trapped ions to create qubits has shown to be a promising approach over the last two decades. They are easily controllable and have reasonably long coherence times, such that good experimental control can be achieved using relatively cheap electronics. To date, ions have seen the highest-fidelity single- and two qubit gates and ran the most proof-of-concept quantum algorithm experiments. Additionally, since ions are charged particles, it is easy to use a combination of static (DC) and radiofrequency (RF) electric fields to confine their motion and trap them.

All the experiments in this thesis were conducted on $^{40}\mathrm{Ca}^{+}$ ions. They provide a convenient internal level structure, where two levels can be picked out as a computational basis. See figure 2 for an overview of the calcium energy splitting. While there are two possibilities to choose states for a two-level qubit, this experiment uses the optical quadrupole transition ($\lambda = 729\,\mathrm{nm}$) connecting the $S_{1/2}$ ($|0\rangle$) to the $D_{5/2}$ ($|1\rangle$) state. Complementing to this transition a huge framework of cooling, readout and state preparation mechanisms has been developed, allowing very good control of this specific system. Further details can be found in [Kie15], but for the purpose of this thesis it is sufficient to know that all these operations can be performed with a sufficient level of precision.

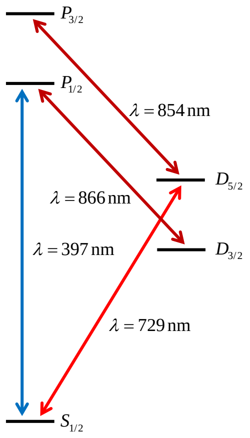
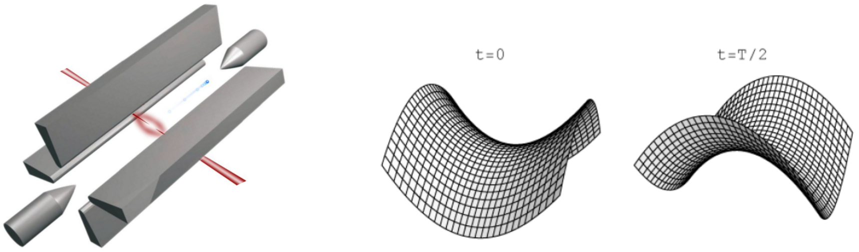

$P_{3/2}$
$P_{1/2}$
$\lambda = 854\,\mathrm{nm}$
$D_{5/2}$
$\lambda = 866\,\mathrm{nm}$
$D_{3/2}$
$\lambda = 397\,\mathrm{nm}$
$\lambda = 729\,\mathrm{nm}$
$S_{1/2}$

Figure 2: Simplified level structure of a $^{40}\mathrm{Ca}^{+}$ ion without any motional degree of freedom and no magnetic field applied. [Kie15]

However, as this simple level structure only holds as long as the ions are isolated and in vacuum, it is necessary to build a trap architecture. For this, the simple concept of a so called linear Paul-trap (fig. 3a) provides a good starting point. Here a confining potential is created by four long rod electrodes and two end-cap electrodes. Since physics does not allow to create a 3-D potential minimum using only DC fields (see Earnshaw's theorem), two of the rod electrodes have to be connected to a RF function generator, which



(a) A linear Paul trap setup [Ame+]

(b) The rotating radial potential of a Paul trap plotted at two different times [Mos]

Figure 3: Linear Paul trap potential and geometry

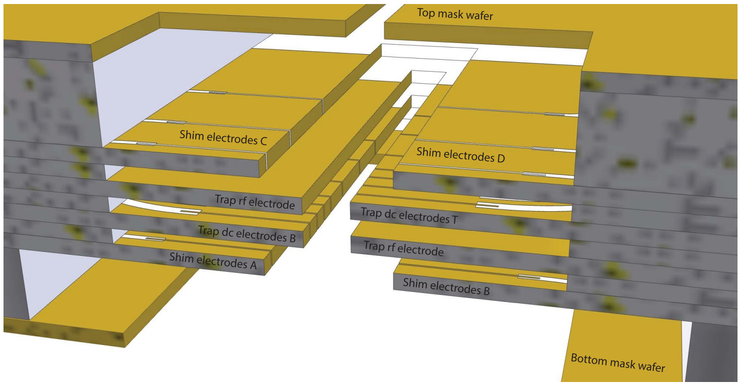


Figure 4: View of the segmented trap built by Daniel Kienzler as part of his PhD thesis [Kie15]. End cap electrodes are no longer needed as axial confinement can be achieved by applying correct voltages to the individual segments

will create a quickly rotating saddle potential (fig. 3b) which will then result in a quasistatic, 3-D confinement.

The trap used in this thesis is built according to a variation of this simple concept. The modified setup replaces the end cap electrodes with segmented DC electrodes on the side which will be used for the axial confinement (fig. 4). This way different voltages can be applied to the segments, allowing more fine-grained control of the axial potential which opens opportunities to transport ions (see upcoming publications of L. de Clercq).

3.4 Single Qubit Gates

Input dependent operations on qubits are called gates, in analogy to classical computing. Theoretical studies of quantum circuits show that a small, finite set of single- and two-qubit gates is sufficient to apply arbitrary quantum operations up to any needed precision by chaining them. To keep the theory easy, we want to stick with single-qubit gates in this thesis and pick out the example of a NOT operation. Let us have a look at the Hamiltonian for a trapped ion in the resonant case [HRB08], describing the so called carrier transition

$$\hat{H} = \hbar\Omega(\hat{\sigma}_+ + \hat{\sigma}_-)$$

Here resonant refers to no detuning of the 729 nm laser with respect to the ion qubit transition energy difference. The operators $\hat{\sigma}_+ = |1\rangle\langle 0|$ and $\hat{\sigma}_- = |0\rangle\langle 1|$ are excitation and deexcitation operators and Ω is the Rabi frequency, a quantity proportional to the 729 nm laser intensity.

In the following paragraph we will see how this Hamiltonian acts on a qubit initially prepared in the ground state and calculate the unitary time evolution. Since $\hat{H}$ is constant in time, we can easily use standard formulas to calculate the final state at a time t:

$$\begin{aligned} |\Psi(t)\rangle &= \hat{U}(t)\,|0\rangle && (5)\\ &= \exp(i\Omega(\hat{\sigma}_+ + \hat{\sigma}_-)t)\,|0\rangle && (6) \end{aligned}$$

Switching to a matrix representation simplifies calculating the exponential such that we obtain the resulting final state

$$\begin{aligned} &= \exp\left[i\Omega\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}t\right]\begin{pmatrix}1\\ 0\end{pmatrix} && (7)\\ &= \begin{pmatrix}\cos(\Omega t) & i\sin(\Omega t)\\ i\sin(\Omega t) & \cos(\Omega t)\end{pmatrix}\begin{pmatrix}1\\ 0\end{pmatrix} && (8) \end{aligned}$$

$$\Leftrightarrow \quad |\Psi(t)\rangle = \cos(\Omega t)\,|0\rangle + i\sin(\Omega t)\,|1\rangle \qquad (9)$$

From there we can calculate the probability to find our qubit in the $|0\rangle$ state after time t as

$$\begin{aligned} P_0(t) &= \frac{1}{2}(\langle\hat{\sigma}_z\rangle_t + 1) && (10)\\ &= \frac{1}{2}\left[\begin{pmatrix}\cos(\Omega t) & -i\sin(\Omega t)\end{pmatrix}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}\cos(\Omega t)\\ i\sin(\Omega t)\end{pmatrix} + 1\right] && (11)\\ &= \frac{1}{2}[1 + \cos^2(\Omega t) - \sin^2(\Omega t)] && (12)\\ &= \cos^2(\Omega t) && (13) \end{aligned}$$

This used the $\hat{\sigma}_z$ operator as a measurement for the projected value on the $\hat{z}$ axis. Another (less general) possibility to obtain this result in the given case would have been to just calculate the squared overlap amplitude $|\langle 0|\Psi(t)\rangle|^2$. To finally implement the NOT gate, now just Ω (laser intensity) and t (laser pulse duration) have to be chosen such, that $\Omega t = \pi/2$ as this will make the cosine in equation 13 go to zero. All in all, this is just a special case of a general rotation gate, which can all be implemented using Rabi flopping, a name for the state oscillation effect we just saw.

3.5 Decoherence

In a perfect world without disturbance to our system, quantum mechanics is actually completely deterministic. Time evolution of all states is described as unitary evolution satisfying Schrödinger's equation. Randomness only comes into play when a quantum system will be measured and collapses due to the interaction with a classical device. But nature is not that nice and there is no such thing as a closed, completely isolated quantum system. As it is impossible to write down a Hamiltonian for every interaction with the environment, frameworks like the density matrix formalism exist to ease the calculation of system evolution.

Nevertheless, such disturbance from the environment (a big, classical reservoir) has an irreversible effect on our (hopefully weakly) interacting quantum system. Without going into too much detail, the result will be a loss of information, making the quantum states less 'pure' and leading the system to behave more and more classically, which means it loses the ability to exhibit quantum behaviour. This process is called decoherence, as the state phase information will go away.

While environmental factors are one source of decoherence, noise can have a similar effect. Due to the statistical nature of quantum measurements it is necessary to repeat the *same* experiment multiple times to obtain good estimations for the underlying binomial distributions. However, if the experimental setup varies for each run, a similar contrast-reducing effect can be observed. The aim of the device created as part of this thesis, is to reduce decoherence due to a certain noise type, which will be described in detail in the next section.

3.6 Laser Amplitude Noise

As stated before, in this thesis we will particularly focus on the carrier Rabi-flopping experiment performed on a trapped calcium ion. In case of a constant laser drive intensity Ω_0, the state evolution was already calculated in equation 13 and a plot of the probability to collapse to state $|0\rangle$ during a measurement is shown in figure 5.

This is the behaviour which shall be replicated as well as possible in a real quantum computer. The overlap between the desired outcome and the actual system behavior can easily be quantified using the so called fidelity. For pure quantum states it is defined as

$$F = \sqrt{\langle\Psi|\phi\rangle\,\langle\phi|\Psi\rangle}$$

In our case we want to check it after a π-pulse, meaning that after a time $\tau = \pi/\Omega_0$ the overlap between the realized state $|\Psi(\tau)\rangle$ and the starting state $|0\rangle$ shall be calculated as in

$$\begin{aligned} F &= \sqrt{\langle\Psi(\tau)|0\rangle\,\langle 0|\Psi(\tau)\rangle} && (14)\\ &= |\,\langle\Psi(\tau)|0\rangle\,| && (15)\\ &= \sqrt{P_0(\tau)} && (16) \end{aligned}$$

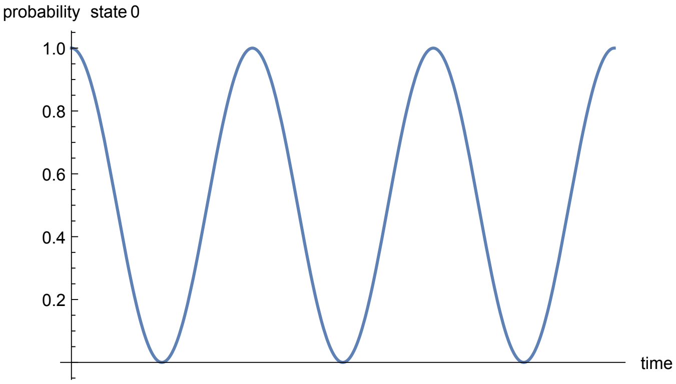


Figure 5: Carrier flopping in an ideal, closed quantum system. The probability to find the ion in state $|0\rangle$ is always 100 % for all $\tau = n\pi/\Omega_0$ with $n \in \mathbb{N}$.

If $F < 1$, there is some form of decoherence present in the experimental setup.

We now want to obtain the expected value of $P_0(\tau)$ in the case that our laser intensity Ω is different in each experiment while the pulse duration τ remains constant. We also assume that the Ω fluctuations are much slower compared to the duration of a pulse (in our setup this will be on the order of $\tau \approx 10\,\mu\text{s}$), such that it can be assumed to be constant while conducting a single shot. Each of these shots completes the following experimental sequence:

- Prepare ion in state $|0\rangle$
- Apply a 729 nm laser pulse for τ seconds and constant intensity Ω
- Read-out the final state (which will always collapse to $|0\rangle$ or $|1\rangle$)

Our final goal will be to calculate the fidelity. Therefore another assumption has to be made on the actual probability distribution function of Ω, which we will assume to be normally distributed around some Ω_0 with standard deviation σ.

$$
\begin{aligned}
P_0(\Omega,\tau) &= \cos^2(\Omega\tau) && (17)\\
P_0(\tau) &= E_\Omega(P_0(\Omega,\tau)) && (18)\\
&= E_\Omega(\cos^2(\Omega\tau)) && (19)\\
&= \int_{-\infty}^{+\infty} \cos^2(\Omega\tau) p(\Omega,\Omega_0,\sigma) d\Omega && (20)\\
&= \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} \cos^2(\Omega\tau) \exp\left[-\frac{1}{2}\left(\frac{\Omega-\Omega_0}{\sigma}\right)^2\right] d\Omega && (21)
\end{aligned}
$$

This relation is valid for any τ. E_Ω represents the expected value given the distribution of Ω. For the fidelity however, we need to consider the correct overlap amplitude for any given time. As we only calculated P_0 so far, this can

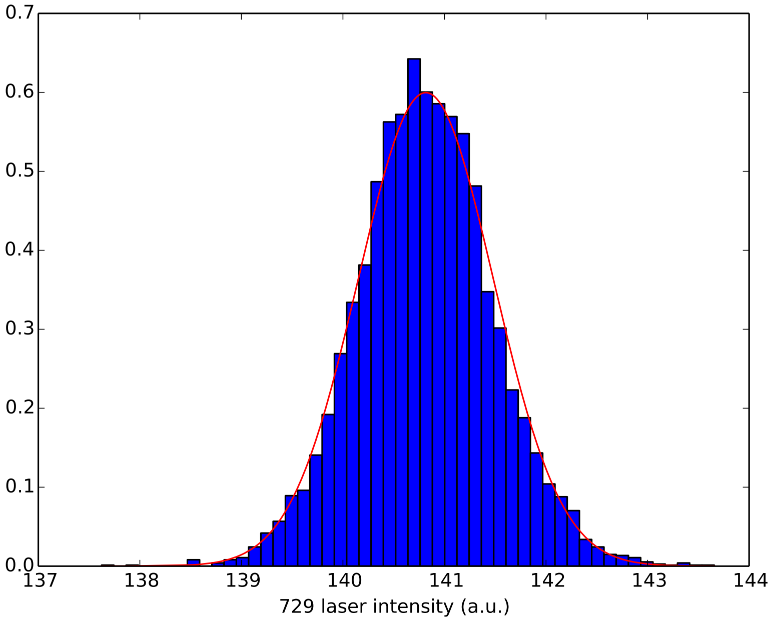


Figure 6: Histogram of the 729 nm laser intensity measured over one hour in 500 ms timesteps with intensity stabilization on. The red line corresponds to a fitted gaussian distribution. The relative standard deviation is $\hat{\sigma} \approx 0.47\,\%$

be used to write down the fidelity for any $n\pi$-gate ($n \in \mathbb{N}$)

$$F_\tau(\Omega_0, \sigma) = \sqrt{P_0(\tau)} = \sqrt{\frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{+\infty} \cos^2(\Omega\tau) \exp\left[-\frac{1}{2}\left(\frac{\Omega - \Omega_0}{\sigma}\right)^2\right] d\Omega}$$

Evaluating this integral analytically is actually pretty hard. So here is the required intensity stability for a 99.99 % π-gate fidelity was calculated numerically. For $\tau = \pi$ and $\Omega_0 = 1$ the intensity may not vary more than $\sigma = 0.0045$, which corresponds to a relative stability of 0.45 % if *only* intensity noise is causing decoherence. This required relative stability should be constant even with changing Ω_0/τ relations.

Another method of calculating the infidelity using a density matrix approach can be found in Thomas Harty's PhD thesis [Har13].

4 Results

For all experiments run in this group so far, there was no realtime measurement of the laser intensity as the timescales are fairly short. On the other hand some good effort has been put into actively stabilizing the 729 nm laser intensity. Figure 6 displays a histogram of the measured 729 intensity. As calculated before, this stability should be sufficient to achieve high fidelities with a π-gate. But for longer pulse times the required intensity stability increases and with the measured standard deviation the fidelity of a 3π-gate would already go down to only 99.90 %.

4.1 Data Acquisition

For the sake of proving the effectiveness of experimental data postselection on laser intensity we deliberately increased intensity fluctuations to a level at which they clearly dominate other sources of decoherence (like frequency noise) in the setup. Normally a tapered amplifier is used to increase the 729 nm laser intensity. It also has a feedback channel which is used to stabilize the intensity output using a PID controller attached to a photodiode behind the TA. To increase fluctuations this feedback port was fed with a ramp signal.

The specific algorithm used to acquire individual data points introduces another effect of intensity fluctuations, which does not cause decoherence, but distorts the shape of Rabi oscillations. Therefore it is necessary to have a look at the sequence in which our control software (Ionizer) acquires experimental data. If we look at a carrier flopping experiment, we want to obtain estimated probabilities at the end of different pulse lengths. For each pulse length it is again necessary to run the experiment multiple times with a constant pulse duration to estimate the underlying probability.

With our previous calculations we were mainly focusing on the single data point acquisition and saw a decay effect, if the laser intensity varies while obtaining a single gate. Per gate, all the individual experiment shots are conducted right after each other with a cooling period in between. As soon as one gate has been measured completely, Ionizer will randomly select a new data point to acquire and run the previous sequence with the new pulse duration.

Let us now for a moment assume, that the laser intensity stays constant while acquiring a single gate. As a result there will be no more loss of contrast. If instead now the laser mean intensity varies for each acquired pulse duration, this will introduce some random disorder if the state population is plotted against the desired pulse length. Such behaviour arises from the fact that the population is determined by the *pulse area* Ωt. Say, a resulting gate population shall be determined with a desired intensity Ω_0 at time τ. But now, due to some external effect the laser intensity has a constant offset of Δ. This is resulting in measuring the population of a pulse with the area

$$(\Omega_0 + \Delta)\tau = \Omega_0\tau + \Delta\tau\frac{\Omega_0}{\Omega_0} = \Omega_0\left(\tau + \frac{\Delta}{\Omega_0}\tau\right) \tag{22}$$

which is equal to a gate with a modified pulse duration of $\tau + \Delta\tau/\Omega_0$ and therefore results in a distortion, if the population is plotted at the original intended pulse duration of τ.

4.2 Carrier Flop Simulation

In this section simulated results of the various described effects are shown. The excellent Python quantum simulation package 'qutip' [JNN13] was used to calculate the dynamics and time evolution.

First we want to concentrate on the average gate-intensity offset. In fig. 7a this effect is plotted with a simulated acquisition of 500 data points in a timerange of 0 to 6π with a normally distributed laser intensity around $\Omega_0 = 1$ and 1 % relative standard deviation. For each data point the experiment was run 100 times with a data point constant, but randomly chosen laser intensity. Obviously, for later data points such intensity instabilities have a much bigger effect, as the noise will be 'amplified' by the pulse duration. A more drastic example has been computed in figure 7b with a noisier laser. It is obvious that in such a case the data becomes useless pretty quickly.

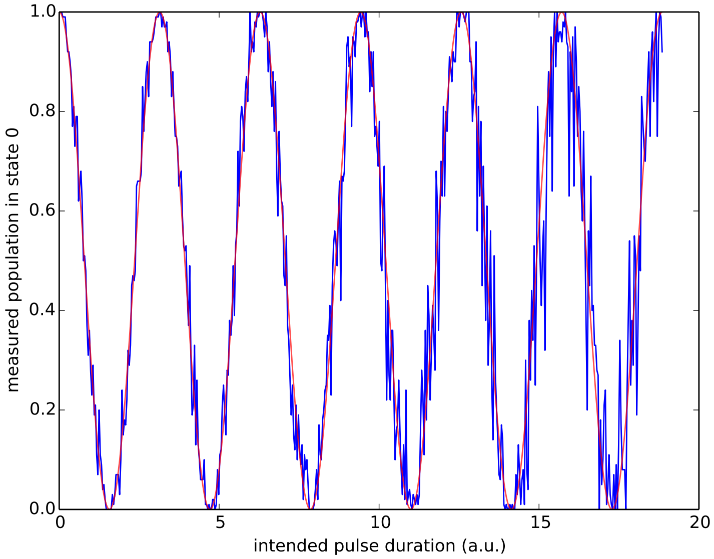
The second case we discussed so far was considering faster intensity noise which happens in between the acquisition of shots for a single data point. Such noise leads to decoherence and a simulation can be found in figure 8 with different noise levels. Both graphs were calculated using 5000 shots per data point to remove effects on the Rabi frequency when only considering a discrete spectrum. If only a few shots are acquired this will lead to a small frequency shift. The results show the predicted behaviour, as more noise leads to a faster decay.

4.3 Post-Selection

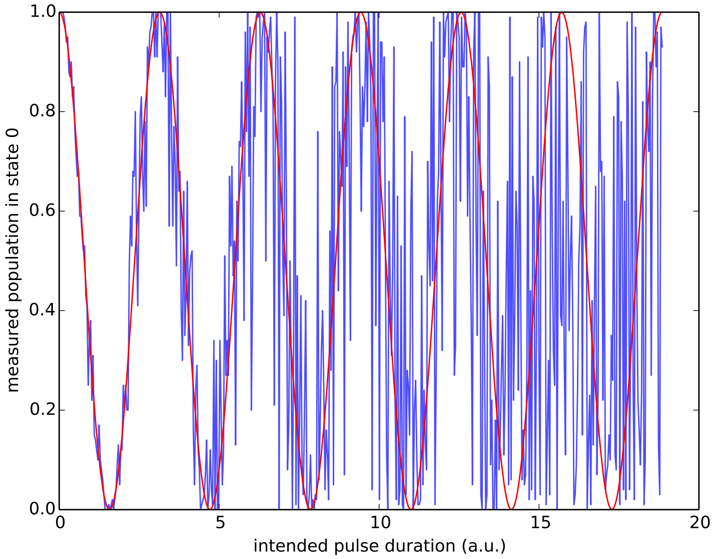
To show contrast improvements arising from intensity based data post-selection, carrier flops were recorded in an unstable lab condition with additional intensity noise introduced on the tapered amplifier as described earlier. The original, complete dataset is plotted in figure 9, where a fast decoherence is obvious. In the next step, a histogram of all measured laser intensities is created, such that a small intensity range which includes many data points can be selected. The result is shown in figure 10. This particular distribution is vastly different from a gaussian distribution due to its non-natural source and it has been introduced using a triangular ramp on the tapered amplifier (TA). In theory this should result in an almost rectangular intensity spectrum which then has been distorted due to non-linearity and hysteresis effects on the TA.

From the histogram it can be seen that a fair amount of data can be extracted by restricting the allowed intensity to values between 57 and 62. This will reduce the relative deviation to approximately 2.5 %. Of course such selection will inevitably make statistics worse, as the number of experiments reduces (drastically, under such extreme conditions). But as the confidence only scales as $\approx 1/\sqrt{n}$ (n is the number of experiments per data point), this degradation is tolerable.

Post-selection results based on the parameters written above are displayed in figure 11. With such a broad original intensity distribution the contrast improvements are quite remarkable and significant. In the second graph (fig. 11b) the post selected data has been reordered using the measured pulse area instead

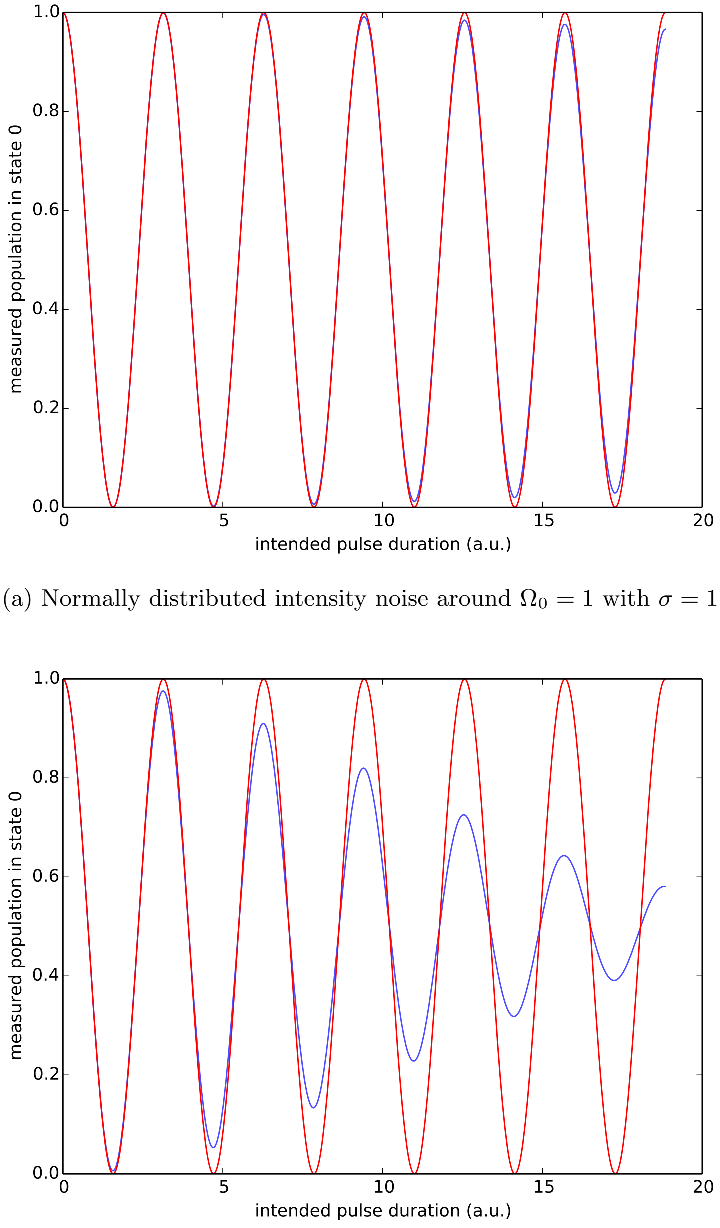


(a) Normally distributed intensity noise around $\Omega_0 = 1$ with $\sigma = 1\,\%$



(b) Normally distributed intensity noise around $\Omega_0 = 1$ with $\sigma = 5\,\%$

Figure 7: Simulated carrier flopping distortion with normally distributed intensity noise that stays constant during a single data point acquisition cycle. Each plot was simulated with 500 data points and 100 shots per data point. The red line displays an ideal behaviour.



(a) Normally distributed intensity noise around $\Omega_0 = 1$ with $\sigma = 1\,\%$

(b) Normally distributed intensity noise around $\Omega_0 = 1$ with $\sigma = 5\,\%$

Figure 8: Simulated carrier flopping decoherence with gaussian distributed intensity noise without any correlation to the current data point taken. Each plot was simulated with 500 data points and 5000 shots per data point to minimize numerical effects. The red line displays an ideal behaviour.

of the intended pulse duration. With this specific set of data results are not particularly compelling, but it can clearly be seen that it reduces the uncertainty on the x-axis. However, it should be noted that these results can vary a lot when analyzing different sets of data or even with the chosen data point downsampling. With some test runs pulse area reordering showed remarkable improvements, but it will probably be of more use with much bigger data sets where the resulting graph contains lots of closely spaced data points and also more shots per data point.

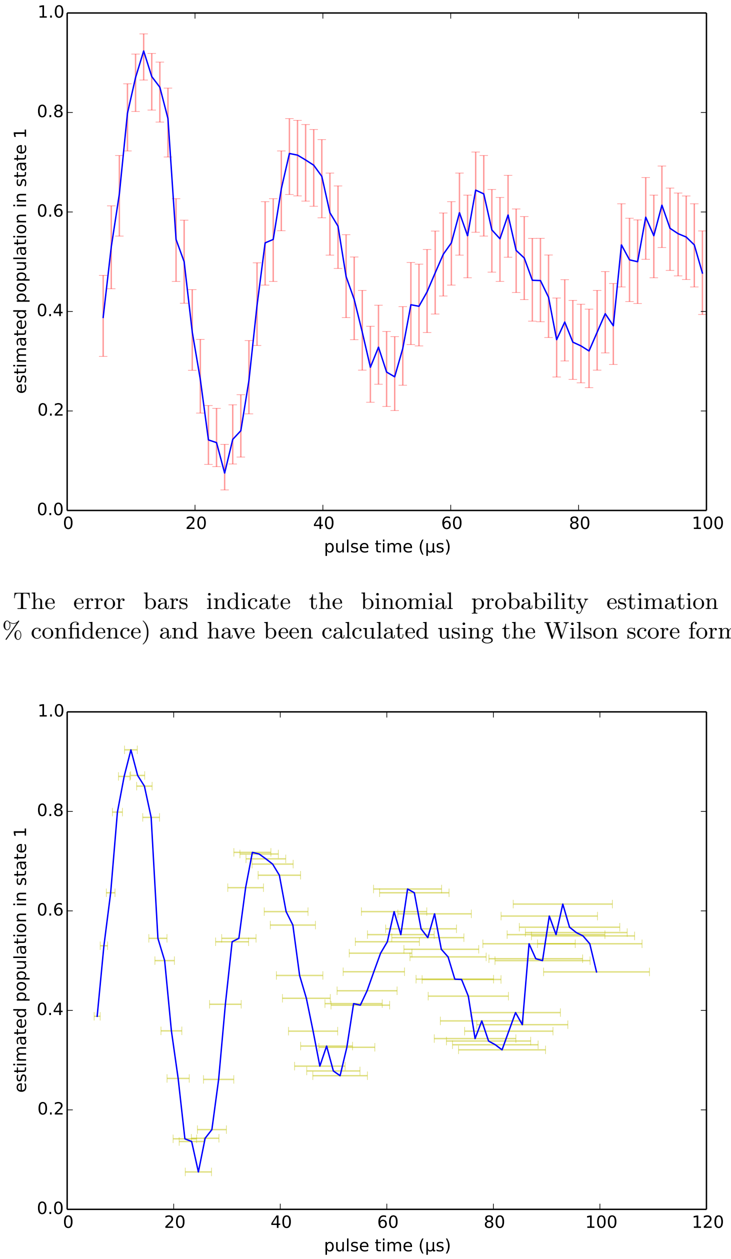
4.4 Conclusion

The experimental results have shown that postselection and data reordering can be used to significantly increase the carrier flopping contrast. Although only a single dataset is shown, similar results were obtained consistently during software development test runs. But unfortunately, due to time constraints and general lab equipment state it was not possible to take more sensible measurements, e.g. on different gate types or to better understand the electronically induced noise. Comparisons with the measured background intensity show, that all these results are above of the electronic noise level, but the signal-to-noise ratio could have been better. Many intensity measurements returned values in the lower 2 % of the analog conversion range, where electronical pickup noise immunity would certainly improve if the signal on the photodiodes could be amplified.

Also the preparations and building of the device took a major part of the available time for this project. As a result the data acquisition and analysis, as well as statistical evalution are carried out on a very basic level of accuracy and a more detailed study would be helpful. Unfortunately, by the time the device was ready to use the general lab setup was in a very unstable state due to several major changes to the equipment. While this was partly helpful to check the validity of this post-selection approach, it also introduced many other noise sources which are resulting in faster decoherence, even with a very well defined laser intensity.

5 Experimental Setup

Controlling trapped ions is a delicate task, as good control of many involved lasers and other devices is needed. For all the processes like readout, cooling, repumping, state preparation, gate operations or ionization, wavelengths of 375 nm, 397 nm, 423 nm, 729 nm, 854 nm and 866 nm are needed in varying polarizations and at different times. Moreover, the intensities and especially frequencies have to be very well defined to match the resonance of the corresponding transition. Due to all of these requirements, like switching beams on and off electronically or to introduce small detunings to the frequency, many AOMs (acousto optic modulators) have been put into the beam paths to allow control. Of course they again require driving amplifiers and (digitally controlled)



(a) The error bars indicate the binomial probability estimation error (95 % confidence) and have been calculated using the Wilson score formula.

(b) The horizontal error bars correspond to assumed 10 % laser intensity noise and are calculated using equation 22.

Figure 9: Complete acquired carrier flopping data consisting of a total of 5000 equally spaced data points of pulse lengths between 5 and 100 µs. Each data point was taken twice and then recombined to show 75 data points as in the given plots.

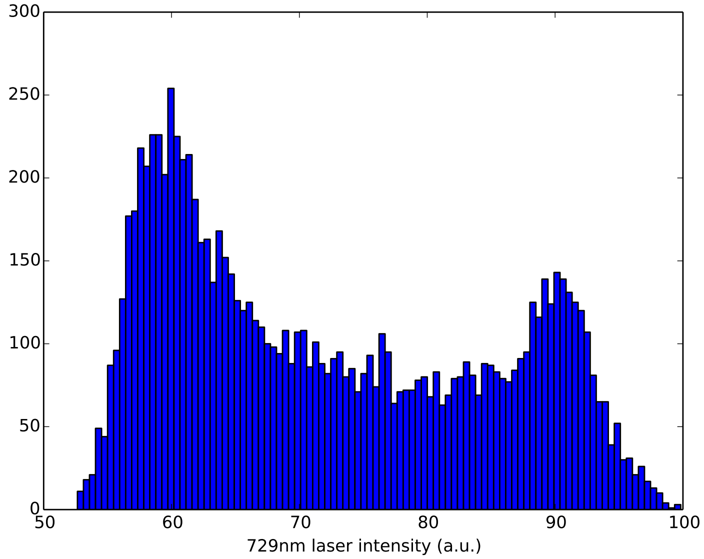
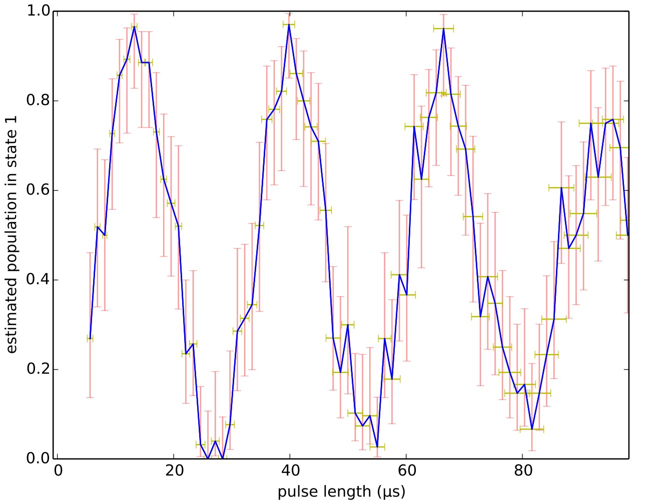
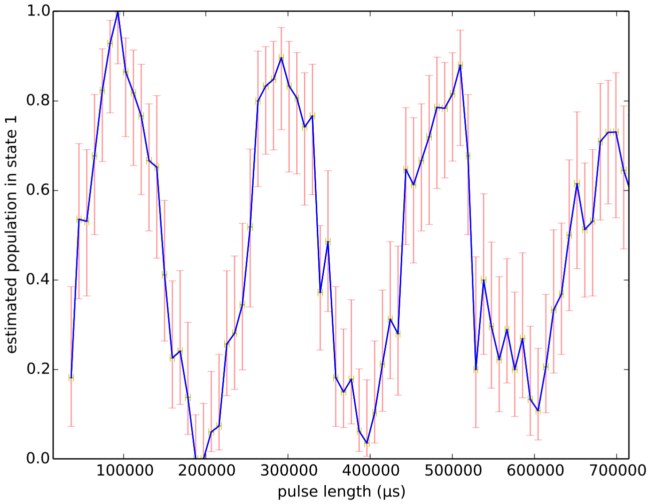


Figure 10: Histogram of the 729 nm laser intensity measured over 10 minutes with a triangular ramp on the tapered amplifier feedback amplifier resulting in large variations for demonstration purposes. Only shots with laser intensities between 57 and 62 were used for post-selection.



(a) Postselected carrier flopping of the same data as in figure 9



(b) Postselected and pulse-area sorted Rabiflopping of the same data as in figure 9

Figure 11: Post-selected carrier flopping with individual experiments sorted according to different parameters. The vertical error bars indicate the binomial probability estimation error (95 % confidence) and have been calculated using the Wilson score formula. Horizontal errors show standard deviations of the actual measured pulse area of all shots which have been assigned to the individual data point.

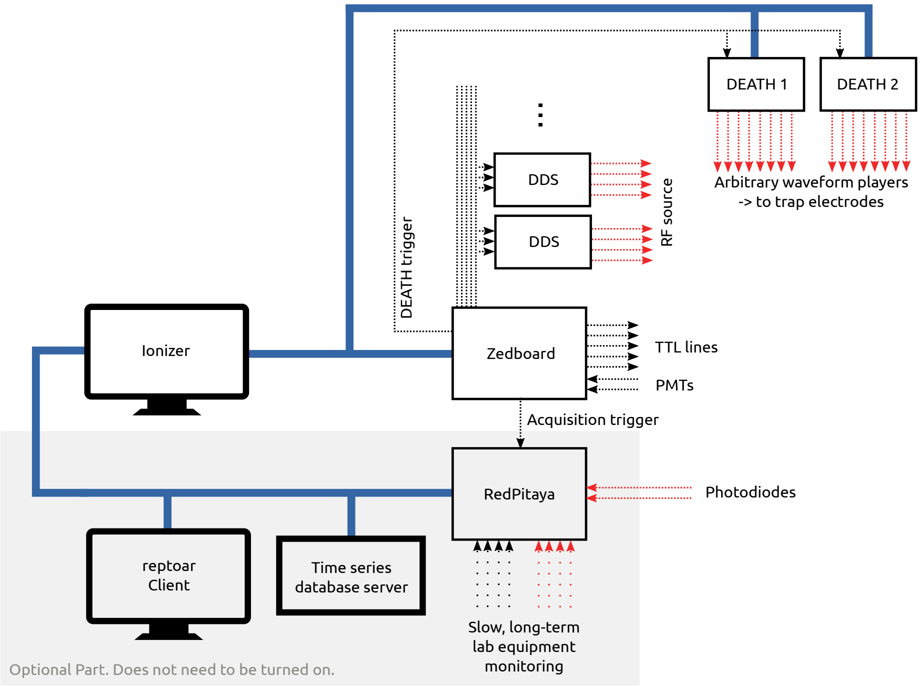


Figure 12: Schematic overview of the most important electronics involved in experimental control and data acquisition. Thick blue lines represent ethernet connections and dotted ones correspond to single line analog (red) or digital (black) inputs/outputs. A high dot frequency will indicate RF capable lines.

RF sources with good timing control.

A typical experimental sequence consists of 5 different sections. Starting with a precooling stage, some near resonant doppler cooling will be applied before preparing the ion in its ground state. Then, the actual gate can be performed (usually including some 729 nm pulses) and is followed by a readout phase which determines the state using fluorescence detection. Further details on this can be found in section 4.3 of [Kie15].

5.1 Lab Computer Control Overview

Such a complex sequence as the one given in the previous section requires good and reliable triggering of the correct devices at well defined times with small jitter. Additionally all this has to be synced with other readout devices which rely on fast acquisition (like photo-multiplier-tubes) and displayed or saved in a user friendly way.

To accomplish all of this, a control architecture has been developed which contains several different layers of software and hardware, completing different triggering and programming tasks at the required realtime constraints. An overview of everything control related is given in figure 12, which will now be described.

DEATH These devices are not of great importance for the experiments covered in this thesis. They contain multi channel, high speed digital-to-analog converters (DACs), which are connected to the individual segmented trap electrodes and thus provide axial confinement and transport capabilities. Waveforms are programmed using Ionizer and other software tools and then sent to the internal memory over an Ethernet connection. Afterwards, these waveforms can be triggered for playback via a single edge-triggered digital line (usually called TTL in this thesis).

DDS A short term for 'direct digital synthesis'. They contain sensitive circuitry to synthesize RF signals with a given phase, amplitude and frequency, and are mainly used as digitally controlled signal sources for AOMs. In the control rack these boards with 4-channels each, sit next to the Zedboard and are connected using a backplane board-to-board interconnect.

Zedboard The most important real-time component of the architecture. It contains a Xilinx Zynq chip, which is built of a dual-core ARM CPU and an FPGA. The programmable logic contains cores for communicating with and triggering the DDSes, as well as modules to trigger TTLs at specified times. Both elements are being controlled using FIFO execution queues which will be preprogrammed using some experiment-specific code that runs on the processor. Code is actually executed without an intermediate operating system layer, but Xilinx library functions are used to handle for example network communication.

In general, the Zedboard receives settings (like AOM frequencies, pulse lengths, ...) from Ionizer and is then responsible for executing a complete experimental sequence with a single set of parameters n times, and sending back the PMT readout results afterwards.

Ionizer This software is the visible user interface to run trap experiments. It is connected to the Zedboard via Ethernet to send the selected experiment including parameters. After completion it receives, stores and plots returned results. The GUI is able to schedule the execution of multiple, simultaneously selected experiments using a priority system. Since the Zedboard only executes a single set of parameters multiple times, Ionizer handles 1-D and 2-D scans by sending new sets of parameters as soon as needed, if for example the 729 nm laser pulse duration shall be swept over a certain range. All returned PMT counts from the experimental readout phase are then individually saved for later data analysis.

Redpitaya New addition to the control system which has been developed in this thesis and adds post-selection functionality. Its capabilities, functionality and integration will be extensively covered in the following sections.

5.2 RedPitaya Capabilities

The builtin components of a RedPitaya board are quickly explained. It consists basically of an almost identical Xilinx Zynq chip as the Zedboard. However, while the hard CPU cores and clock speeds are the same, the programmable logic part is a bit smaller. This SoC is then complemented with 512MB of DDR3 memory, a Gigabit Ethernet port and, the most important part for postselection: Two high speed (125 Msps) analog input channels, as well as two high speed analog outputs (not needed for this project). Additionally, certain functionality of the Zynq chip is exposed via pin headers. This includes the ability to slowly read up to 16 digital inputs, as well as four 12-bit analog inputs up to speeds of a few Hz.

The Hardware package is provided with an open-source ecosystem, including some preprogrammed FPGA bitstream to expose an oscilloscope like functionality to the CPU, as well as an easily runnable operating system with web applications to access the high-speed ADC data using a nice graphical user interface.

As the device is still fairly new and the manufacturer is pushing updates to a development branch almost on a daily basis, one big aim during the development in our group was not to touch the hardware part, so that we can easily profit from new developments on this side. This actually paid off a few times, as some critical bugs were fixed in the meantime, and the required DMA feature (see section 5.4) was added after a while. On the other hand it also complicated some things and causes significant slowdown, as missing features have to be compensated with software, which might add delays. Anyhow, especially for prototyping new features the software solution still turned out to be superior, as synthesizing the FPGA bitstream took approximately 15 minutes on a fairly well equipped workstation.

5.3 Software / Communication Architecture

The Redpitaya's fast analog input channels are used to capture laser intensities during a single experimental sequence, by measuring voltages of photodiodes which are placed in the beam after it passed through the trap. The resulting waveforms can then be sent to a computer running the 'reptoar' Python-based client software for further processing and saving. As the two 125 Msps 14-bit ADC channels could in theory create up to 3.5 GBit/s of data, sending all this over a single GBit network adapter seems to be impossible and a bad choice. And indeed, this architecture reduces the rate at which experiments may be run.

On the other hand, there are good reasons to move the data processing to higher abstraction levels in early stages of development or in a research environment. In both situations requirements and algorithms will change frequently and may also be written by persons who don't want to invest lots of time in learning hardware description languages (HDL) or embedded programming. Additionally, more complex signal processing algorithms might be impossible

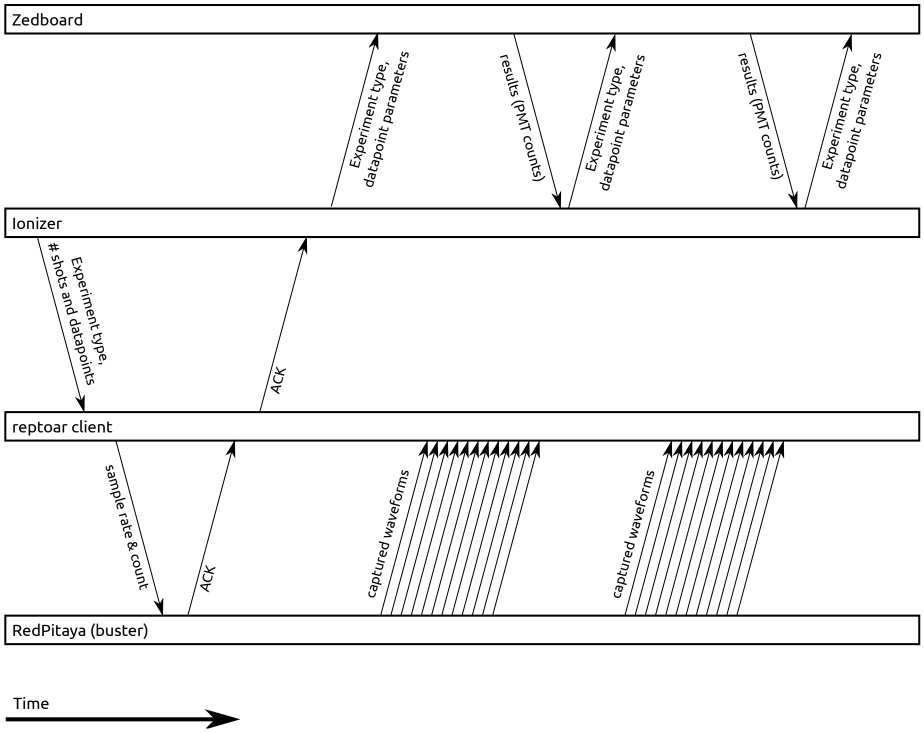


Figure 13: Communication timing diagram when the RedPitaya Post-selection functionality is activated and used. Arrows indicate exchanged messages.

to convert into HDL or will take much computation time on a processor. The latter thing should not happen on the RP's application processor as it needs to stay responsive for ADC acquisition reprogramming.

To transmit data over the network without adding lots of programming overhead, the ZeroMQ library is being used in combination with msgpack for data serialization. Especially ZeroMQ adds some network and processing overhead and therefore reduced the achievable throughput to ≈ 300 MBit/s. But using this widely adopted and well maintained library reduces our custom code complexity by a fair amount and therefore has a good impact on maintainability and stability.

A timing diagram of the RP communication is shown in fig. 13. Between 'reptoar-buster' and the reptoar client multiple message channels are open, all with a distinct purpose:

RPC This remote procedure call interface follows a strict request-response pattern, which is used to send commands from the reptoar-client to the RP. Such commands may lead to reconfiguring of ADC sample rates and durations, or a start/stop of streaming operation.

Streaming Used to send waveforms to interested clients in realtime. Uses the ZeroMQ publisher-subscriber pattern and is configured to quickly drop packages such that latencies are short.

Capture Sends the same type of packages as the streaming channel, but in push-pull configuration. This way it can be ensured that no message (waveform data) will be lost if the receiver can't keep up with the data rate for a short amount of time.

While the RPC channel is always open to receive commands, the streaming and capturing channel will be used depending on the currently selected experiment and mode. For post-selection, with a known amount of data points the capture channel is active, while for continuously running experiments the streaming channel will be used to give an oscilloscope like, realtime waveform display. The purpose of the reptoar client is to read Ionizer's synchronization message which contains the next experiment type and total expected amount of shots. This information will be connected with a configuration file to determine the required sampling settings. These are then sent to the Redpitaya. As soon as this is completed and the RP is ready for acquisition, Ionizer will be notified such that it can contact the Zedboard to start the experimental sequence.

For every incoming trigger, the reptoar-buster tool running on the RP will send captured waveforms out over the selected streaming or capturing connection. Usually these are received by the client, which then selectes the post-processing methods based on the configuration file and currently running experiments. In case of a capturing operation, as soon as the expected amount of triggers has been received the resulting, post-processed data will be saved to a .csv file for later analysis in combination with Ionizer's shot log.

Finally, a few words on the choice of Arch Linux as operating system. After we used the prepackaged 'Redpitaya OS' distribution for a while, it turned out that it is not very suitable for native application development. It does not come with a compiler installed, and cross-compiling applications on the development computer (as well as the required libraries) is possible at best, but usually *very* complicated. Trying to remotely debug our custom data acquisition tools then finally motivated a change to a new distribution as version mismatches between the host and target made it a nightmare to hunt down bugs.

5.4 Datapath / High-Speed Acquisition

As written in the previous section, both ADCs are able to produce big amounts of data very quickly. In order to make sure that no sample is being lost, usually the data will be pushed to a 16384 sample sized Block-RAM ring buffer within the FPGA. In the original configuration this buffer is then memory mapped and accessed using the general purpose 32-bit wide AXI (Advanced eXtensible Interface) bus interface between the processing system and programmable logic. However, as it turned out this particular bus interface bandwidth is way too small, as data rates cut at approximately 2 Mbit/s.

Figure 14 shows the new, direct-memory-access (DMA) based solution. Here the operating system reserves contiguous memory regions in the DDR3 system memory, which are then accessed directly from the FPGA using a high-performance 64-bit AXI bus interface. In the current configuration the CPU will then poll status registers on a regular basis for acquisition completion and make a memory copy as soon the scope module is done writing. After this memory copy operation the reserved DMA memory can again be used to store the next waveform. This process is a lot quicker than the old one, as optimized MDSI (multiple data, single instruction) commands provide a lot more bandwidth on the system memory than on the general purpose AXI bus.

Nevertheless, DMA based solutions are tricky to program, and the used approach gives some improvement, while still leaving room for many optimizations which will allow even smaller trigger dead times. More on this in section 6. As a proof of concept, the current implementation contains a small and simple kernel module which allocates the required contiguous physical memory regions on load time and writes the corresponding lower- and upper boundary addresses to the correct configuration registers of the AXI DMA controller module. This module also offers status registers (current and trigger write addresses) which can be used with a polling algorithm to check if the acquisition has been triggered and is completed.

So all in all, the job of the ‘reptoar-buster’ tool, which runs on the RedPitaya itself, is to read scope configurations received from the client, write them to their corresponding registers, wait for triggers, copy the data from the DMA region to a ZeroMQ message and send them back over the correct streaming or capturing channel.

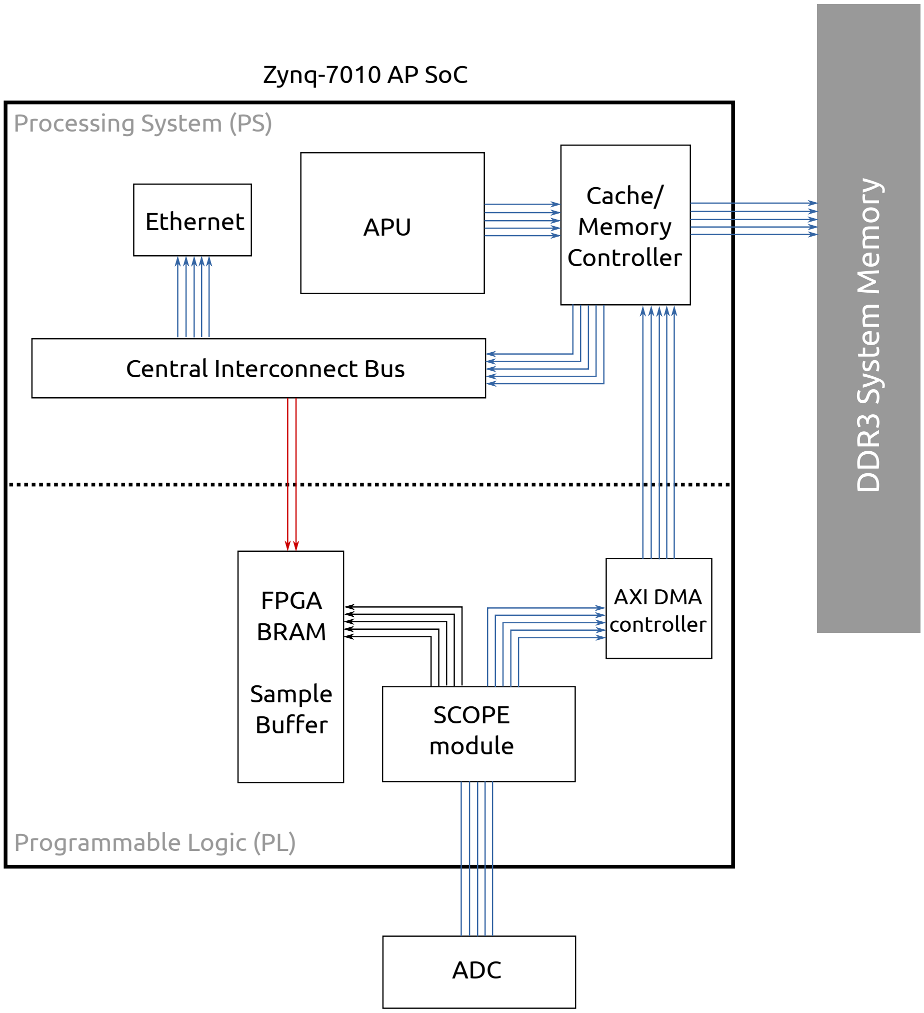


Figure 14: Connection diagram of the different soft- and hard-core modules involved in data acquisition on the RedPitaya. Arrows always point from the master device to the connected slave. Red indicates the slow general purpose AXI bus, while blue arrows are part of the new, DMA-based data path.

5.5 Postprocessing

Due to various reasons which were described in section 5.3, post-processing of the acquired waveforms is done on a full-blown workstation running the reptoar-client Python program. This keeps track of the currently running experiment and then uses a configuration file to pass received waveforms to the correct and helpful processing units (PU). In the client two different base types of PUs are available. The first one, a WaveformProcessor may add a completely new waveform to the raw ones received from the RedPitaya. Typical use cases are for example calculations of a moving average or background subtraction. The second type, a DatapointProcessor will calculate usually a small, finite amount of new data points which have nothing to do with time anymore. As an example one can think of an experiment which pulses different lasers on seperately for short times. Then some kind of algorithm can calculate the mean peak height from each of these pulses to compare individual laser intensities between several experiments and over a longer amount of time. Also, in case of the post-selection experiment a similar technique was used, where a DatapointProcessor measures the mean height of the 729 nm pulse, which is then used to calculate the area.

The client program also includes a simple GUI to display current waveforms and their processed data points as a nice, shot-to-shot timeseries, such that it can be used as a simple oscilloscope on the attached channels and to monitor time variations of important quantities. While the complete waveforms would be too much data to store over a longer time period, processed data points are also stored in a time-series database.

5.6 Long-term Logging

The Zynq SoC chip with the shipped RedPitaya hardware configuration allows for additional data acquisition using the built-in XADC module, which can be used for slow analog voltage readout of 4 external pins. On top of this, 16 digital (15 usable, as one is used as external trigger input) input pins are accessible via readout status registers, to query their state. Reading these values is a slow process, so it is only suitable for DC voltages (2 Hz readout rate), but such functionality is very helpful to monitor laboratory equipment over longer periods of time. The currently installed configuration for example monitors voltages applied to cavity piezos or thermocouples to get a better overview of correlations which affect laser system stabilities.

The digital channels, on the other hand, can be used to log or display status of systems like the Calcium ovens, PID lock states, and other events which can be thresholded or converted into a digital signal. But one has to be aware that these inputs are also only polled on a large time interval basis, so short events might be missed. The solution for this problem will be discussed in the next section.

To actually store and display all this logging data, a ready-to-use open source time series database called InfluxDB is being used. For this also nice and easily configurable web interfaces for realtime display and historic browsing are

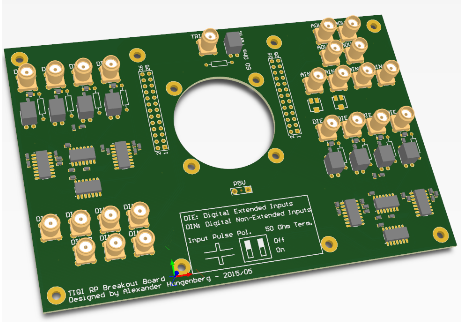

P5V
DIE: Digital Extended Inputs
DIN: Digital Non-Extended Inputs
Input Pulse Pol.
50 Ohm Term.
Off
On
50 Ohm Term.
TIQI RP Breakout Board
Designed by Alexander Hungenberg - 2015/05

Figure 15: 3D breakout board assembly visualization.

available. Our choice fell to Grafana, which provides excellent support for this specific database. Both these services run on an always-on database server that also has been indicated in figure 12. Additionally, it hosts a small script which receives logging broadcasts and converts them into the correct database API format. These broadcasts are emitted via small daemons ('reptoar-logger') running on the RedPitayas and sending out the logging messages in a publisher-subscriber ZeroMQ pattern.

5.7 Breakout Board

As all the slow, long-term logging inputs are exposed via pin headers, it is very hard to make reliable and easy-to-use connections in a crowded lab environment. On top of this, the described event-miss possibility for short pulses on the digital channels renders them useless for many applications that only emit brief pulses. An example for such application is a mains glitch/spike detection box built by Frieder Lindenfelser, which generates short pulses of unknown height if there are disturbances on the electric supply network. The unknown and possible high output voltage height also imposes a threat when connected directly to the FPGA, as high voltage can destroy circuitry without protection.

To eliminate all these shortcomings a breakout board has been designed which will connect to the RP's pin headers and make them easily accessible via SMA connectors, a cable type very commonly used in our lab. Furthermore, some circuitry was designed to improve the useability of the digital inputs with respect to the following design goals:

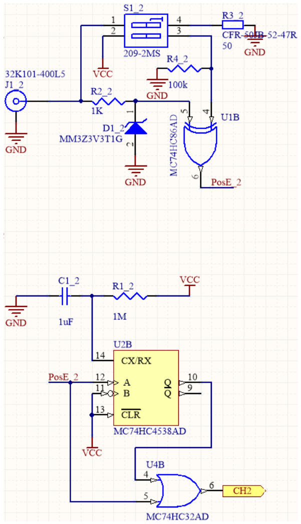


Figure 16: Digital pulse extension and overvoltage protection circuitry

- Selectable $50\,\Omega$ termination
- Overvoltage protection
- Pulse extension circuitry

The latter should make sure that short, digital pulses are extended to a duration which cannot be missed by the 2 Hz polling interval. Therefore a so called monostable multivibrator IC is used, which triggers a configurable length output pulse on a positive edge at the input pin. The circuit is shown in figure 16. At a first stage, the dip switch allows to configure the connection to a 500 mW termination resistor which can be used for input voltages up to 5 V. A second dip switch can be used to select a possible input signal inversion using a XOR gate, as the multivibrator expects a positive edge signal. This output is then sent to the pulse extender and recombined using an OR gate with the multivibrator output. This results in a final signal being sent to the FPGA which is high as long as the input signal is being pulsed, but at least 1 second long.

6 Conclusion and Outlook

This project has been developped as a proof of concept for the usefulness of data post-selection in quantum information experiments. Additionally it has proven to be very useful as a lab equipment status monitoring device of both slow moving analog and digital channels, as well as realtime experimental conditions. The resulting post-selected data sets showed significant improvements with respect to longer coherence times. The visibility of this effect was even more surprising as all of this data was taken in an unstable laboratory environment and post-selection reduced an already small dataset to a really tiny one. On the other hand, results of individual shot reordering based on the measured pulse area were promising in theory, but the acquired data could not keep up with high expectations. The reasons for this are unknown, but probably our assumptions were too strict compared to the real-world experimental setup.

In addition to this experimental part a theoretical calculation of decoherence effects due to intensity noise has been carried out, resulting in useful formulas to estimate required stability and possible effects in future setups. While currently, in stable lab condition timeframes, the intensity noise does not seem to be a major source of decoherence, the proposed scheme of post-selection can be of good use as soon as more complex single- and multi-qubit gates are considered which will include longer pulse durations.

As with any prototyping project, further possibilities for improvement were developed over the meantime. First, the fast acquisition DMA part would benefit a lot of increased sample buffer sizes and DMA pooling, a technique which assigns a new and empty memory region immediately after the previous write is done. This way the old data can be copied and transmitted while the next shot is already being acquired. This would help to get rid of the wait-time completely, which currently can't be used as the RP has to first send and empty the buffer before new data may be written into it. Also algorithms may be completely moved into the hardware as soon as they have become stable and proven to be effective. At least for thresholding or the calculation of means this could drastically improve post-processing computation times. Another very helpful addition would include a digitally implemented, configurable delay of the analog input channels. This would allow to record gate operations and the following readout phase at the same time which could be used to detect a dark ion due to a briefly unlocked 397 nm laser.

Regarding the long-term logging abilities of the RedPitaya also various improvements are possible. A major thing which did not make it onto the first prototype breakout board is the missing voltage clamping at the trigger line. Also the amount of slow analog input channels is small, as they quickly became very successful in the lab. Multiple solutions are possible to overcome this problem. Multiplexing would be one, but it might also be interesting to get more accurate and precise measurements. An additional ADC for this could easily be added as an SPI or I2C bus interface is also exposed. Such steps towards a global lab monitoring system seem to be very welcome and might become an interesting project for a future student.

References

[Fey82] RichardP. Feynman. "Simulating physics with computers". English. In: *International Journal of Theoretical Physics* 21.6-7 (1982), pp. 467–488. ISSN: 0020-7748. DOI: 10.1007/BF02650179. URL: http://dx.doi.org/10.1007/BF02650179.

[Deu85] D. Deutsch. "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer". In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 400.1818 (1985), pp. 97–117. ISSN: 0080-4630. DOI: 10.1098/rspa.1985.0070.

[Sho94] P. W. Shor. "Algorithms for Quantum Computation: Discrete Logarithms and Factoring". In: *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*. SFCS '94. Washington, DC, USA: IEEE Computer Society, 1994, pp. 124–134. ISBN: 0-8186-6580-7. DOI: 10.1109/SFCS.1994.365700. URL: http://dx.doi.org/10.1109/SFCS.1994.365700.

[Sho95] Peter W. Shor. "Scheme for reducing decoherence in quantum computer memory". In: *Phys. Rev. A* 52 (4 Oct. 1995), R2493–R2496. DOI: 10.1103/PhysRevA.52.R2493. URL: http://link.aps.org/doi/10.1103/PhysRevA.52.R2493.

[Ste96] Andrew Steane. "Multiple-Particle Interference and Quantum Error Correction". In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 452.1954 (1996), pp. 2551–2577. ISSN: 1364-5021. DOI: 10.1098/rspa.1996.0136.

[DiV00] David P. DiVincenzo. "The Physical Implementation of Quantum Computation". In: *Fortschritte der Physik* 48.9-11 (2000), pp. 771–783. ISSN: 1521-3978. DOI: 10.1002/1521-3978(200009)48:9/11<771::AID-PROP771>3.0.CO;2-E. URL: http://dx.doi.org/10.1002/1521-3978(200009)48:9/11%3C771::AID-PROP771%3E3.0.CO;2-E.

[NC00] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[HRB08] H. Häffner, C.F. Roos, and R. Blatt. "Quantum computing with trapped ions". In: *Physics Reports* 469.4 (2008), pp. 155–203. ISSN: 0370-1573. DOI: http://dx.doi.org/10.1016/j.physrep.2008.09.003. URL: http://www.sciencedirect.com/science/article/pii/S0370157308003463.

[Har13] Thomas P. Harty. "High-Fidelity Microwave-Driven Quantum Logic in Intermediate-Field 43Ca+". PhD thesis. 2013.

[JNN13] J.R. Johansson, P.D. Nation, and Franco Nori. "QuTiP 2: A Python framework for the dynamics of open quantum systems". In: *Computer Physics Communications* 184.4 (2013), pp. 1234–1240. ISSN: 0010-4655. DOI: http://dx.doi.org/10.1016/j.cpc.2012.11.019. URL: http://www.sciencedirect.com/science/article/pii/S0010465512003955.

[Kie15] Daniel Kienzler. "Quantum Harmonic Oscillator State Synthesis by Reservoir Engineering". PhD thesis. 2015.

[Ame+] Ben Ames et al. *Nanofiber*. URL: http://www.quantumoptics.at/index.php/en/research/nanofiber (visited on 06/29/2015).

[Glo] Glosser.ca. *Bloch Sphere*. URL: https://commons.wikimedia.org/wiki/File:Bloch_Sphere.svg (visited on 06/29/2015).

[Mos] Mostafa. *Quadrupole Potential Generation in Paul traps*. URL: http://physics.stackexchange.com/questions/82291/quadrupole-potential-generation-in-paul-traps (visited on 06/29/2015).