

**University of Stuttgart**  
Germany

A nanosecond-resolution computer  
control system for deterministic  
excitation of Rydberg superatoms

Bachelor Thesis

by

Udo Hermann

31.07.2012

Prof. Dr. Tilman Pfau

5th Institute of Physics

University of Stuttgart

## **Declaration**

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person, except where due acknowledgment has been made in the text.

Udo Hermann  
Stuttgart, July 31, 2012

## Contents

<b>1</b>	<b>German Introduction</b>	<b>4</b>
<b>2</b>	<b>Introduction</b>	<b>5</b>
<b>3</b>	<b>Motivation</b>	<b>6</b>
3.1	Theoretical background . . . . .	6
3.1.1	Rydberg atoms . . . . .	6
3.1.2	Deterministic excitation, Rabi-oscillation . . . . .	6
3.1.3	The three-level system . . . . .	7
3.2	Application . . . . .	8
3.2.1	Example: The experiment on the 4th floor . . . . .	8
3.2.2	Future: The experiment on the 5th floor . . . . .	8
<b>4</b>	<b>Concept</b>	<b>11</b>
4.1	Hardware . . . . .	11
4.1.1	Features . . . . .	11
4.1.2	Application . . . . .	11
4.1.3	Short pulse feature . . . . .	12
4.1.4	OP_CODE . . . . .	12
4.1.5	Bugs . . . . .	13
4.2	Software . . . . .	13
4.3	Advantages over the direct programming . . . . .	14
4.4	Pulseblaster controller . . . . .	15
4.4.1	Graphical user interface . . . . .	15
4.4.2	Wrapper class . . . . .	18
4.4.3	Data backend . . . . .	18
4.5	Variables . . . . .	19
4.6	Errors . . . . .	20
<b>5</b>	<b>Measurements</b>	<b>21</b>
5.1	Reflections at the transition from a ribbon cable to a <b>50 <math>\Omega</math></b> cable . . . . .	21
5.2	Pulse shape . . . . .	22
5.3	Jitter . . . . .	23
5.3.1	Measurements with an Oscilloscope . . . . .	23
5.3.2	Measurements with a SensL HRMTime TDC . . . . .	24
5.4	Crosstalk . . . . .	27
<b>6</b>	<b>Outlook</b>	<b>28</b>
6.1	Computer control system . . . . .	28
6.2	Future in the laboratory of the Pulseblaster board . . . . .	28
6.3	Jitter measurement with SensL HRMTime TDC . . . . .	28

## 1 German Introduction

1885 entdeckte Johann Balmer einen Zusammenhang der Spektrallinien von Wasserstoff [12]. Dieser ist ein Spezialfall der Rydberg-Ritz-Gleichung, welche die Wellenlängen der Übergänge in Wasserstoff beschreibt. Atome, welche ein einzelnes Elektron in einem angeregten Zustand mit einer Hauptquantenzahl, welche größer ist als die der restlichen Grundzustand-Elektronen, besitzen, nennt man Rydberg-Atome. Rydberg-Atome sind von großem Interesse für die Atomphysik, viele Laboratorien weltweit, einschließlich dem 5. Physikalischen Institut der Universität Stuttgart, führen auf dem Gebiet der Atomphysik eine Vielzahl an Experimenten durch.

Für diese Experimente ist die kontrollierte Anregung in bestimmte Rydberg-Zustände von großer Bedeutung. Die Wechselwirkung eines zwei-Niveau-Systems mit einem Laser wird durch die Theorie der Rabi-Oszillation (3.1.2) beschrieben. Um Atome in höhere Niveaus anzuregen sind daher kurze Laserpulse (im Bereich von Nanosekunden) mit hoher Präzision nötig. Die Lichtschalter hierfür werden meist mit akustooptischen oder elektrooptischen Modulatoren realisiert. Die Ansteuerung dieser Bauteile erfolgt elektronisch. Daher ist eine Soft- und Hardware nötig, welche derartige Pulse realisieren kann. Eine mögliche und sehr kompakte Variante hierfür ist die Verwendung einer Pulseblaster PCI-Karte von SpinCore, welche je nach Modell Pulslängen von 2 ns erzeugen kann.

Ziel und Aufgabe dieser Bachelorarbeit ist die Implementierung der PCI-Karte, das Prüfen auf Funktionalität, das Ausmessen technischer Daten und die Programmierung einer Steuerungssoftware welche alle Optionen beinhaltet, die die Karte kann und die sich Anwender wünschen könnten. Diese Steuerungssoftware soll zusammen mit einer weiteren Steuerungssoftware, welche Geräte von National Instruments ansteuert, zu einem großen Programm zusammengeführt werden, welches für die Rydberg-Experimente des 5. Physikalischen Instituts der Universität Stuttgart in dem 5. Stock verwendet wird.

## 2 Introduction

In 1885, Johann Balmer figured out a connection between the spectral lines of hydrogen [12]. This connection is a special case of the Rydberg-Ritz equation, which describes the transitions of a hydrogen atom. Atoms with a single electron excited to a state with a principal quantum number much larger than the ones of the remaining ground state electrons are called Rydberg atoms. Rydberg atoms are very interesting for atom physics, many laboratories worldwide, including the 5th Institute of Physics of the University of Stuttgart, execute a lot of experiments on the field of atomic physics.

For these experiments, the controlled excitation to specific Rydberg levels is very important. The interaction of a two-level system with a laser is described by the theory of Rabi oscillation (3.1.2). To excite atoms to a higher level, short laser pulses (nanosecond scale) with a high precision are needed. The light switches for this application mostly are acousto-optic or electro-optic modulators. These switches are controlled electronically. Therefore, a software and hardware is needed which can create suitable pulses. One possible hardware which is also very compact is the Pulseblaster PCI board from SpinCore, that can create pulses (depending on the board frequency) down to 2 ns.

The aim of this bachelor thesis is the implementation of the PCI board, perform checks on functionality, measure technical data and create a control software which includes all features of the PCI board which a user could expect. This control software will be merged to a control system, together with another control software for devices from National Instruments. This control system will be used for the Rydberg experiments of the 5th Institute of Physics of the University of Stuttgart on the 5th floor.

### 3 Motivation

Rydberg experiments use lasers to excite the atoms. The excitation to the Rydberg level is a quantum mechanical process and the probability of finding the atom in the excited state is a time-dependant  $\sin^2 \Omega t$ -function. Therefore, it is necessary to switch the excitation laser within some nanoseconds. The National Instruments experiment control is able to set output on a microsecond-scale. This is not suitable for the deterministic excitation which needs some nanoseconds. SpinCore Technologies provides PCI boards which are able to create digital pulses that are 2 ns short [1].

#### 3.1 Theoretical background

This chapter will introduce the theoretical background that is important for this thesis. First, there will be a short introduction to Rydberg atoms. After that, the deterministic excitation and Rabi-oscillation of a two-level system will be introduced, which is a central aspect of this thesis. Finally, the three-level system used in the experiment will be explained.

##### 3.1.1 Rydberg atoms

Highly excited atoms that have high principal quantum numbers ( $\gtrsim 30$ ) are called Rydberg atoms. The formula for the energy of a Rydberg level of hydrogen is [11]:

$$E_n = -\frac{R}{n^2} \quad (1)$$

$n$  is the principal quantum number and  $R = hcR_\infty$  the Rydberg energy, 13.6 eV. For rubidium, the Rydberg energy changes [5]:

$$R_{Rb} = \frac{R_\infty}{1 + m_e/M} \quad (2)$$

$M$  is the mass of the rubidium atom,  $m_e$  the mass of the electron. The scaling property for rubidium is the same as the scaling property for hydrogen, there is only a factor of  $\frac{1}{1+m_e/M}$ . The electron in the highly excited state, called Rydberg state, is bound weakly to the nucleus. To avoid collisions with the gas around the experiment, which would heat up the ultra-cold gas immediately, the experiment is performed within vacuum. The radius of a hydrogen Rydberg atom scales with  $n^2$  [13]:

$$r_n = n^2 a_0 \quad (3)$$

$a_0$  is the Bohr radius which is 0.0529 nm [13]. E.g. an atom with the principal quantum number of 100 has a radius of 0.529  $\mu\text{m}$ . Rydberg atoms with a high principal quantum number like this have shown very interesting behaviour, such as the ability to form a two-atomic molecule on this length scale for a short amount of time (about 18 ms at the 37S state [2]) with another ground state atom.

##### 3.1.2 Deterministic excitation, Rabi-oscillation

We assume a two-level system with the ground state  $E_0$  and the excited state  $E_1$ . A resonant light field has the frequency  $\omega$ :

$$\hbar\omega = E_1 - E_0 \quad (4)$$

The dipole transition moment of this transition is  $\mu$ . When the light field is applied to the atom, the electron of the ground state will start to oscillate between the two states  $E_0$  and  $E_1$ . The oscillation frequency is the Rabi frequency [7]:

$$\Omega = \frac{\mu E}{\hbar} \quad (5)$$

$E$  is the electric field component of the light field. The propability of finding the electron in the state  $E_i$  is [7]:

$$p_{E_0}(t) = \cos^2 \frac{\Omega t}{2} \quad (6)$$

$$p_{E_1}(t) = \sin^2 \frac{\Omega t}{2} \quad (7)$$

To successfully excite the electron from the ground state to the excited state, meaning a propability of finding the electron in the excited state of 100 %, the light field has to be applied for a time  $t_\pi$ , named a  $\pi$ -pulse:

$$\frac{\pi}{2} = \frac{\Omega t_\pi}{2} \rightarrow t_\pi = \frac{\pi}{\Omega} \quad (8)$$

A Rabi frequency of 1 GHz needs a pulse length in the range of 3 ns. 3.2.2 shows that this is the time scale which can be reached with the new experiment.

### 3.1.3 The three-level system

The experiment uses  $^{87}\text{Rb}$ . The start level is  $5S_{\frac{1}{2}}$ , the aim is about  $50S_{\frac{1}{2}}$ . The standard solution for this is a two-photon excitation via the  $5P_{\frac{3}{2}}$  level due to the selection rule for the azimuthal quantum number  $\Delta l = \pm 1$ .

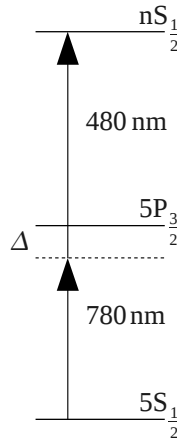


Figure 1: The three-level system for  $^{87}\text{Rb}$

Each transition has a Rabi frequency  $\Omega_i$  related to the intensity of the coupling light. The three-level system can be simplified to a two-level system with the effective Rabi frequency [3]:

$$\Omega_{Eff} = \frac{\Omega_1 \Omega_2}{\Delta} \quad (9)$$

with the detuning  $\Delta$ .

## 3.2 Application

The existing experiment of the same institute is located on the 4th floor whereas the second generation experiment which is still under construction is located on the 5th floor.

### 3.2.1 Example: The experiment on the 4th floor

The time scale of the experiment is given by the reachable Rabi frequency. In addition to creating short excitation pulses, various other components have to be controlled on the same time scale:

- Switch lasers for Rydberg excitation on and off.
- Switching the light frequency.
- Manipulate Rydberg states with an electrical field.
- Switch a great magnetic field for ionisation.
- Synchronized start of detection:
  - Device to read the ion detector.
  - Activate single-photon detectors.

The shortest pulses needed on the 4th floor are:

Usage	Time scale
Ramsey	100 ns
Spectroscopy	5 – 20 $\mu$ s

Table 1: Smallest time scales of the experiment on the 4th floor

All other processes will take more time, so they are not listed.

### 3.2.2 Future: The experiment on the 5th floor

This section will show the calculation of the Rabi frequencies with the parameters of the experiment and show why it is necessary to implement new hardware for this time scale. The reason for a higher Rabi frequency are the lasers. Using a higher laser power will help to focus the laser better, meaning a smaller waist. However, a higher laser power will increase the Rabi frequency. The other way around, being able to create shorter pulses will allow the usage of a higher laser power and thereby a better laser focus.

#### Necessary formulas and values

The power  $P$  of a laser is [9]:

$$P = A \cdot I \quad \text{resp. for non-rectangular beams} \quad P = \iint_A I dA \quad (10)$$

We assume a Gaussian beam [10] with the waist  $\omega_0$  and the beam diameter  $\omega$ :

$$E = E_0 \frac{\omega_0}{\omega} e^{-\frac{r^2}{\omega^2}} \quad (11)$$

The intensity is proportional to the square of the electric field:

$$I = I_0 \frac{\omega_0^2}{\omega^2} e^{-\frac{2r^2}{\omega^2}} \quad (12)$$



Integrating this over an area with the radius  $R$ :

$$\begin{aligned}
P &= \int_0^{2\pi} d\varphi \int_0^R r dr I_0 \frac{\omega_0^2}{\omega^2} e^{-\frac{2r^2}{\omega^2}} \\
&= I_0 2\pi \omega_0^2 \frac{1}{4} \int_0^R \frac{4r}{\omega^2} e^{-\frac{2r^2}{\omega^2}} \\
&= I_0 \frac{\pi \omega_0^2}{2} \left[ -e^{-\frac{2r^2}{\omega^2}} \right]_0^R \\
&\text{for } R \rightarrow \infty \\
&= I_0 \frac{\pi \omega_0^2}{2}
\end{aligned} \tag{13}$$

With the peak intensity  $I_0$  and the waist of the beam  $\omega_0$ . The lasers in the experiment are specified by their power, so the needed formula is:

$$I_0 = \frac{2P}{\pi \omega_0^2} \tag{14}$$

The peak intensity of a laser is [8]:

$$I_0 = \frac{1}{2} \epsilon_0 c E_0^2 \tag{15}$$

$\epsilon_0$  (vacuum permittivity) and  $c$  (speed of light in vacuum) are known natural constants,  $E$  is the electric field. To calculate the electric field for a given intensity, the formula is:

$$E_0 = \sqrt{\frac{2I_0}{\epsilon_0 c}} \tag{16}$$

The transition dipole moment from  $5S_{\frac{1}{2}}$  to  $5P_{\frac{3}{2}}$  is given [4]:

$$\mu(5S_{\frac{1}{2}} \rightarrow 5P_{\frac{3}{2}}) = 3.584(4) \cdot 10^{-29} \text{ C} \cdot \text{m} \tag{17}$$

The transition dipole moment from  $5P_{\frac{3}{2}}$  to  $nS_{\frac{1}{2}}$  scales with  $n^{-1.5}$  [6]:

$$\mu(5P_{\frac{3}{2}} \rightarrow nS_{\frac{1}{2}}) = 4.4154n^{-1.5} \text{ a.u.} \tag{18}$$

Whereas 1 a.u. is the product of the elementary charge and the Bohr radius:

$$1 \text{ a.u.} = e \cdot a_0 = 1.6022 \cdot 10^{-19} \text{ C} \cdot 5.2918 \cdot 10^{-11} \text{ m} = 8.4754 \cdot 10^{-30} \text{ Cm}$$

The Rabi frequency is (see (5)):

$$\Omega = \frac{\mu E}{\hbar} \tag{19}$$

### Rabi frequency of the $5S_{\frac{1}{2}} \rightarrow 5P_{\frac{3}{2}}$ excitation

The used values are:

$$\begin{aligned}
\mu &= 3.584 \cdot 10^{-29} \text{ C} \cdot \text{m} \\
P_{laser} &= 1 \mu\text{W} \\
\omega &= 5 \mu\text{m}
\end{aligned}$$

The Rabi frequency is:

$$\frac{\Omega}{2\pi} = 236.926 \text{ MHz} \tag{20}$$

**Rabi frequency of the  $5P_{\frac{3}{2}} \rightarrow nS_{\frac{1}{2}}$  excitation**

The used values are:

$$\begin{aligned} n &= 50 \\ \mu &= 1.0588 \cdot 10^{-31} \text{ C} \cdot \text{m} \\ P_{laser} &= 10 \text{ mW} \\ \omega &= 5 \mu\text{m} \end{aligned}$$

The Rabi frequency is:

$$\frac{\Omega}{2\pi} = 69.9956 \text{ MHz} \quad (21)$$

**Effective Rabi frequency of the  $5S_{\frac{1}{2}} \rightarrow nS_{\frac{1}{2}}$  excitation**

See (9). The detuning  $\Delta$  is assumed to 500 MHz. The effective Rabi frequency is:

$$\frac{\Omega_{Eff}}{2\pi} = 33.1675 \text{ MHz} \quad (22)$$

The time to excitate (see (8)) is:

$$t_{\pi} = 15.0750 \text{ ns} \quad (23)$$

The assumed values (e.g. laser power) are only estimated, they are just an example to show the time scale on which the light switches should be able to be turned on and off. The Rabi frequency was calculated without the usage of the Clebsch-Gordan coefficient, for an exact calculation, this has to be included into the calculations.

The Pulseblaster boards have minimal pulse lengths of 2 ns to 3.3 ns, so this hardware is suitable for the estimated pulse durations. The duration of pulses of the Pulseblaster board can only be adjusted with steps of 2 ns to 3.3 ns, so the exact tuning has to be done by adjusting the laser power. In the measurement part of this thesis, a very exact pulse of 20 ns was measured and analyzed. The power of the  $5P_{\frac{1}{2}} \rightarrow nS_{\frac{1}{2}}$  laser will be adjusted. The propability to find the atom in the excited state is given by (7). To get 100 % propability, the laser power must be 5.68137(6) mW. If the pulse length is not exactly 20 ns, the propability to find the atom in the excited state is:

Jitter [ps]	Pulse duration [ns]	Propability to find the atom in the excited state (ppm)
-10000	10	707107
-3000	17	972370
-1000	19	996917
-300	19.7	999722
-100	19.9	999969
-30	19.97	999997
-10	19.99	1000000
0	20	1000000
10	20.01	1000000
30	20.03	999997
100	20.1	999969
300	20.3	999722
1000	21	996917
3000	23	972370
10000	30	707107

Table 2: Propability to find the atom in the excited state versus the jitter of a 20 ns pulse

To have a propability of over 999999 ppm, the pulse length must not jitter more than 10 ps. The measured pulse seems to have an inaccuracy of 1 ps, this is good enough to fulfil the requirements.

## 4 Concept

This chapter will explain the different parts that were used (hardware) or created (software) for this bachelor thesis. At first, there will be a short summary about the Pulseblaster board and a small introduction how to program the board. After that, the software will be described. The concept behind the program will be explained and a short introduction to the usage of the software will be given.

### 4.1 Hardware

The used hardware is a SpinCore Pulseblaster ESR-PRO 400 MHz board [1]. This board has the following features and can be programmed easily as shown in 4.1.2.

#### 4.1.1 Features

- 21 output channels that can be set each 15 ns.
- Short pulse feature for pulse lengths down to 2.5 ns.
- Up to 4096 instructions can be written to the board.
- Simple Loop and Jump instructions.

For 300 MHz the values change to 20 ns and 3.3 ns, for 500 MHz to 12 ns and 2 ns

#### 4.1.2 Application

To directly program the Pulseblaster board, a list of instructions is necessary. Each instruction consists of:

- A channel bitmask that indicates which channels should be turned on (e.g. 1110 0000 0000 0000 0101 will turn on channel 0 and 2) and also sets the short pulse feature by the 3 highest bits:
  - 111 for ON, this means all desired channels stay at 3.3 V for the whole instruction length.
  - 001 to 101 for short pulses of 2.5 ns (001) to 12.5 ns (101), see 4.1.3 for details about the short pulse feature.
- An instruction that tells the board what to do after the execution of this instruction (called OP\_CODE), see detailed information on 4.1.4. Some examples are:
  - BRANCH: Continues at the desired instruction.
  - JSR / RTS: Jumps to a selected instruction (JSR) and is able to jump back with RTS.
  - LOOP / END\_LOOP: Loops all instructions from LOOP to END\_LOOP  $n$  times,  $n$  is defined in the additional value for the OP\_CODE.
  - WAIT: Stops program execution and waits for a trigger signal.
- An additional value for the OP\_CODE, like a jump address or a number of desired loops.
- A double value defining the length of this instruction.

Further information can be found on the SpinCore website [1].

### 4.1.3 Short pulse feature

The short pulse feature is a special invention of SpinCore. Turning channels on and off can be done each 6 clock cycles (400 MHz clock means 15 ns). For pulses smaller than this, there is the short pulse feature. It provides the possibility to turn on all channels for a short time period (at least 1 clock cycle). A short pulse feature will always affect all channels, it is not possible to keep one channel HIGH while applying a 1 cycle-pulse (400 MHz: 2.5 ns), all channels will be set to LOW after the pulse. A short pulse feature instruction has to be at least 6 cycles long and after the falling edge, three clock cycles on LOW state have to be awaited. So the minimum instruction length is:

Short pulse duration (clock cycles)	Minimum instruction length (clock cycles)
6 and greater	the same as the pulse duration
1	6
2	6
3	6
4	7
5	8

Table 3: Minimum instruction lengths

### 4.1.4 OP\_CODE

Each instruction that is written to the Pulseblaster board can be given a so-called OP\_CODE. This code tells the board what to do after having finished the actual instruction. Some codes need extra values e.g. jump addresses. The available OP\_CODES and values are:

OP_CODE	Value	Functionality
CONTINUE	Ignored	The program will execute the instruction and then continue with the next.
STOP	Ignored	The program will stop execution until the board is resetted oder re-programmed.
LOOP	Number of desired loops	This OP_CODE specifies the beginning of a loop.
END-LOOP	Address of the beginning of the loop	The program will re-execute the code between the beginning of the loop and this end instruction for the specified number of loops.
JSR	Address of instruction	The program will jump to this instruction.
RTS	Ignored	Jumps back to the instruction after the last JSR instruction.
BRANCH	Address of instruction	The program continues execution at the specified address.
LONG_DELAY	Number of desired loops	For long durations, a single instruction can be repeated to create longer pulses.
WAIT	Ignored	Waits for a hardware or software trigger and then continues program execution after the duration of this instruction plus 6 clock cycles

Table 4: Available OP\_CODES

### 4.1.5 Bugs

When implementing the Pulseblaster board, there occurred quite a lot of bugs that were already implemented on the board by SpinCore. These bugs were wiped out while working with the board by direct communication with some developers of SpinCore.

#### 64-bit

The Pulseblaster API (SpinAPI) for C++ is written for 32-bit windows. Using it on 64-bit windows required some changes which were made by SpinCore after I identified these bugs for them. The SpinAPI also had to be adapted for Matlab. It took a lot of time to fix this bug for all kind of application (64-bit C++, 64-bit Matlab Mex, 64-bit C#).

#### Hardware trigger

After applying a hardware trigger signal, the Pulseblaster ESR-Pro board with a 400 MHz clock frequency will wait about 11 seconds and then start the program execution. This only occurs on the 400 MHz version of the board, not on the 300 MHz version. This bug is not fixed yet.

#### Installation on 64-bit

The installation of the board is not as simple as it should be. To properly install the Pulseblaster board, there are some unusual things to be done:

- At first, the SpinAPI has to be installed. Therefore, the board must not be in the computer.
- After installing the SpinAPI, the board can be plugged into the computer.
- To install the drivers, an admin console has to be started.
  - Change directory to /SpinCore/SpinAPI/drivers/win/amd64.
  - Execute

```
wdreg.exe -inf windrvr6.inf install
```
- Device manager - Scan for hardware changes. The driver will now be installed.

## 4.2 Software

The first version of the program was written in Matlab, using the Matlab add-on Mex to compile C++ code. After adding some essential features, the performance with Matlab was too slow, some operations took about 20s. So the computer control system was completely re-written in C#. The current version splits up into two great parts, the National Instruments controller, written by Stephan Jennewein and the Pulseblaster controller, written by myself. There are some other components that are necessary for both controllers. The whole program contains the following parts:

- Main window
- National Instruments controller
- Pulseblaster controller
- Variables
- Errors

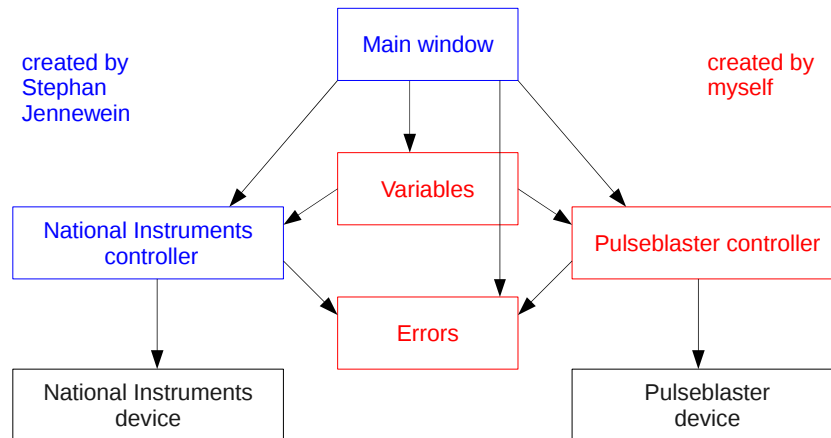


Figure 2: Structure of the whole control system

The focus of this bachelor thesis lies on the Pulseblaster controller, the variables and the errors will only be introduced shortly. The Pulseblaster controller allows the user to access all available functions of the Pulseblaster ESR-PRO board. This features pulse lengths down to 2.5 ns with the 400 MHz board. In the final application there will be a trigger signal for the Pulseblaster board. After receiving the trigger, the board will run some instructions and then wait for the next trigger or a new program.

### 4.3 Advantages over the direct programming

Using the direct programming of the board as shown in 4.1.2 would request a lot of practice because programming the board would be like writing a great list of numbers. For the user, there is a different surface to program the board and the program will convert the user input into the list of numbers. The essential changes of the surface to the direct programming are:

- The user does not need to set all channels at the same time, the channels can be set individually and the program merges this to instructions at the same time.
- The user can re-use sequences with the flowchart on the right of the pulseblaster control window, e.g. call a certain sequence 5 times in a row, the program will write the same code 5 times to the board and the user only has to write the sequence once.

### 4.4 Pulseblaster controller

The Pulseblaster controller is split up into three parts: Graphical user interface, wrapper class and data backend. The three parts are connected like this:

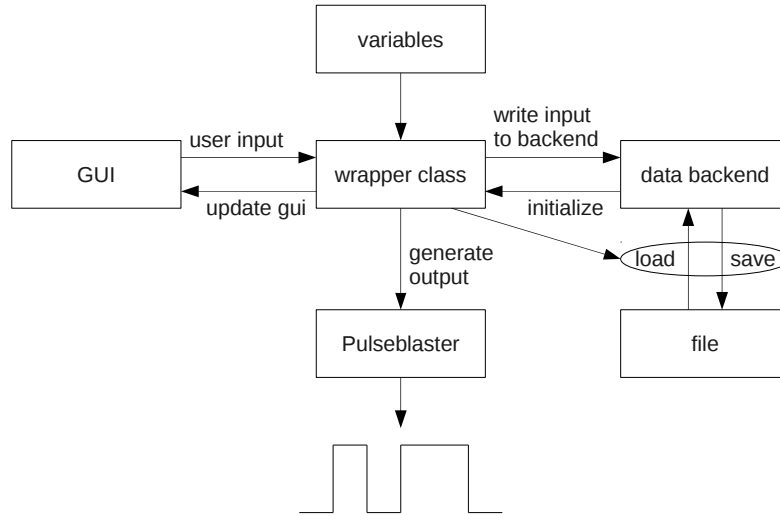


Figure 3: Structure of the Pulseblaster control system

#### 4.4.1 Graphical user interface

At first, we will focus on the graphical user interface. Instead of creating one great list, the programming is split up into sequences. Each sequence can switch each channel individually. At this point, each sequence contains all 21 channels and each channel contains 12 timesteps.

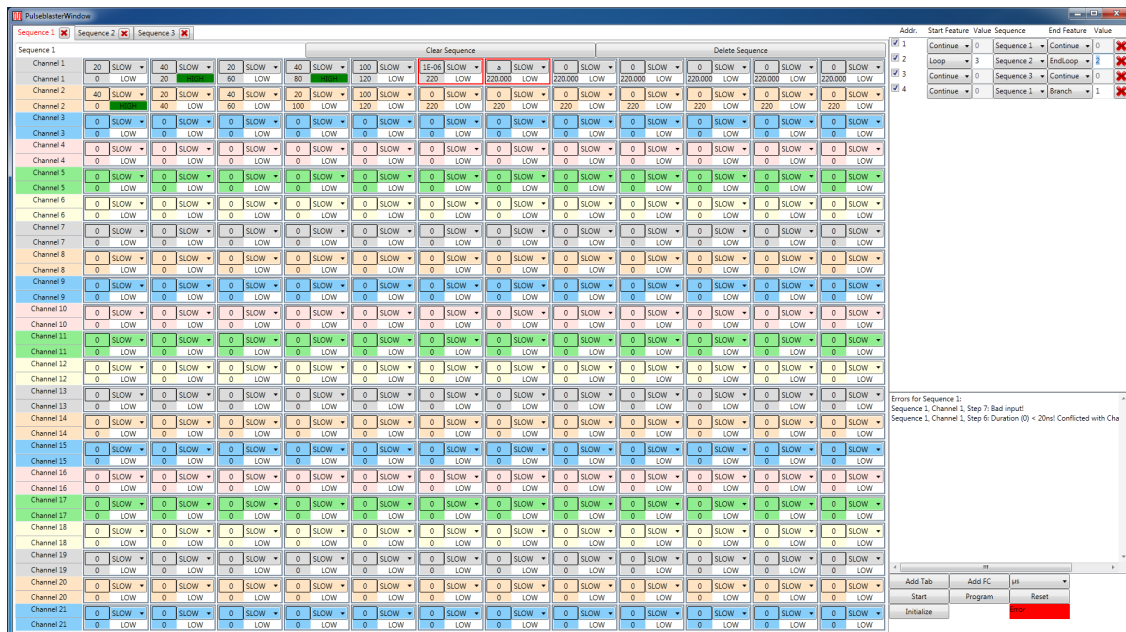


Figure 4: Graphical user interface

On the top, the sequences are listed as a tab control. Each sequence can be named individually by just entering the name in the TextBox right below the sequences tab control.

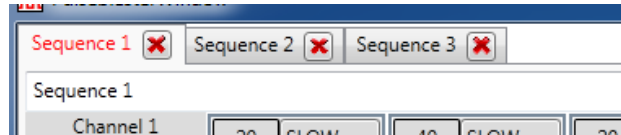


Figure 5: Sequences tab control

On the left, there are the channel headers, each channel can be given a name. In this picture, all channels are named “Channel i”:

Channel 1	0
Channel 2	40
Channel 2	0
Channel 3	0
Channel 3	0
Channel 4	0
Channel 4	0
Channel 5	

Figure 6: Channel headers

On the right side of the Pulseblaster window, there is the so-called flowchart. The user is able to apply the sequences in a desired order and repeat (loop) them or jump from one instruction to another.

	Addr.	Start Feature	Value	Sequence	End Feature	Value	
<input checked="" type="checkbox"/>	1	Continue	0	Sequence 1	Continue	0	✘
<input checked="" type="checkbox"/>	2	Loop	3	Sequence 2	EndLoop	2	✘
<input checked="" type="checkbox"/>	3	Continue	0	Sequence 3	Continue	0	✘
<input checked="" type="checkbox"/>	4	Continue	0	Sequence 1	Branch	1	✘

Figure 7: Flowchart

#### A short introduction to the usage of the flowchart

The flowchart consists of 8 columns:

**Check button:** This button will activate an element so it will be included into the program or deactivate one so the program will not use this element.

**Addr. (Address):** The address of this instruction, this value is needed on some of the value fields.

**Start Feature / End Feature:** The start or end feature will be applied to the first respectively last command. Possible selections: **Continue** and **Loop** for the start feature, **Continue**, **Stop**, **EndLoop**, **JSR**, **RTS**, **Branch** and **Wait**.

Using **Continue** will not affect the program execution, the sequence will be worked off one after another.



Using **Loop** will start to loop all instructions from the first instruction of the selected sequence to an sequence that has the **EndLoop** end feature. The loop will be repeated by the given value in the value field for the start feature and the value for the end feature must be the address that has the accordant **Loop** start feature.

**Stop** will stop all program execution and the board can only be reactivated by resetting or reprogramming.

**JSR** will jump to another instruction with the address being specified in the value field after the instruction has finished. **RTS** will jump back from an jump instruction to the instruction after the last **JSR** instruction.

**Branch** works like **JSR** but there is no possibility to jump back.

**Wait** is suitable for the trigger function, the program will stop execution and wait for the trigger signal and then continues the execution after the length of the last instruction plus 6 clock cycles.

**Value:** This field contains parameters for some of the features.

**Sequence:** With this field, the user can select which sequence should be used for this flowchart element.

**Delete button:** This button will delete a flowchart element.

On the right bottom, there are some control buttons, to initialize, reset, program and start the pulseblaster board, to add a new sequence (Add Tab) or a new element to the flowchart (Add FC). With a dropdown-menu, the time unit can be choosen, from nanoseconds to seconds. Every time input will be interpreted with this unit.

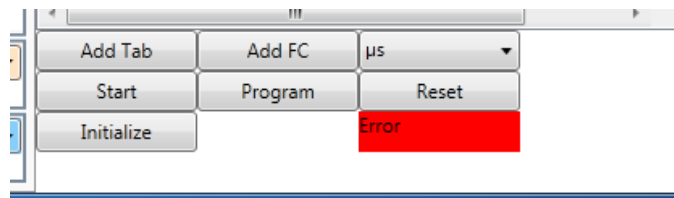


Figure 8: Control buttons

Between the buttons and the flowchart, there is a small error output which will show all the errors that occurred within the selected sequence:

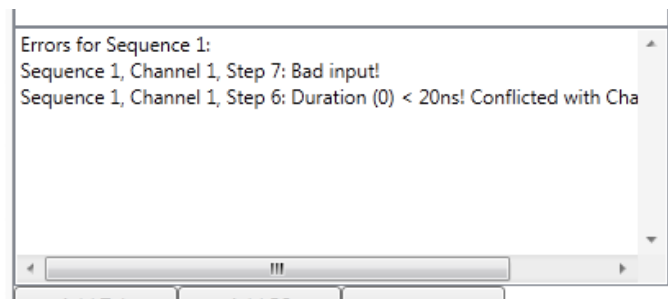


Figure 9: Error output

The main part of the window are the timesteps. A timestep looks like this:

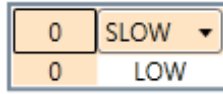


Figure 10: A single timestep

On the top left, there is a user input field where the user can type in the duration of the step. On the lower right, there is a LOW or HIGH button, which indicates whether the channel output should be switched to LOW (0 V) or HIGH (3.3 V). The lower left text output shows the start time of this step. The drop-down menu (SLOW) controls the short pulse feature. See 4.1.3 for details. The timestep has got a border which can indicate errors. A red border means that there is an error in this timestep, a purple border shows that this timestep has been modified and an error occurred. Figure 4 shows two errors at timesteps 6 and 7 in channel 1.

#### 4.4.2 Wrapper class

The wrapper class establishes the communication between the graphical user interface and the backend. When a user enters a new value, the wrapper class will save this to the backend. If a value in the backend changes (e.g. when loading or when changing a variable) the wrapper class will show this change on the graphical user interface. The wrapper class also implements the communication with the Pulseblaster PCI-board, reads the status of the board and programs the instructions to the board. When a user enters a value, this value has to be checked on validity. At first, there is a general check. No negative numbers or characters are allowed as values for the duration of a step. Afterwards, the program generates all sequences and checks for all kind of errors:

- Whether the distance of two instructions is smaller than the minimum length (depending on the applied short pulse feature, see 4.1.3).
- Whether applied features (applied on the flowchart) to a sequence contain errors, such as wrong jump addresses.

The wrapper class is also responsible for the save and load function, it will save the whole data backend to a binary file respectively load a binary file into the data backend. This is done by serialisation and deserialisation of a class (the data backend).

#### 4.4.3 Data backend

The data backend consists of a lot of variables containing all values. It is not possible to serialize classes that are connected (e.g. data binding or events) to another class (e.g. graphical user interface). So the wrapper class saves all values into the data backend but the graphical user interface will be updated by data binding to the wrapper class (which will read the data backend).

## 4.5 Variables

The variables are an essential part of the Pulseblaster controller and will be available in the National Instruments controller, too. Variables can be used to pre-set parameters or change several values at one point instead of changing these values at each point they occur. With the iterator variables (see below), it is possible to perform n-dimensional parameter scans. To have a solution for all kind of application, there are three different types of variables:

- Dynamics: Variables of this type are the result of python code which can access all static and iterator variables. The type of the variable can be boolean (e.g. to switch digital channels on or off) or double (e.g. to set a duration of an instruction).
- Iterators: These variables contain a start and goal, a step (size) and a value. After every run, the variable on the bottom will be increased by step (works also if step is negative and start is greater than goal). When a variable exceeds the goal, the variable will be set to start and the variable above will be increased by its step. The variables can be switched by the two black arrows.
- Statics: Static variables contain a user-input value (double or boolean) and will keep this value until the user manually enters a new value.

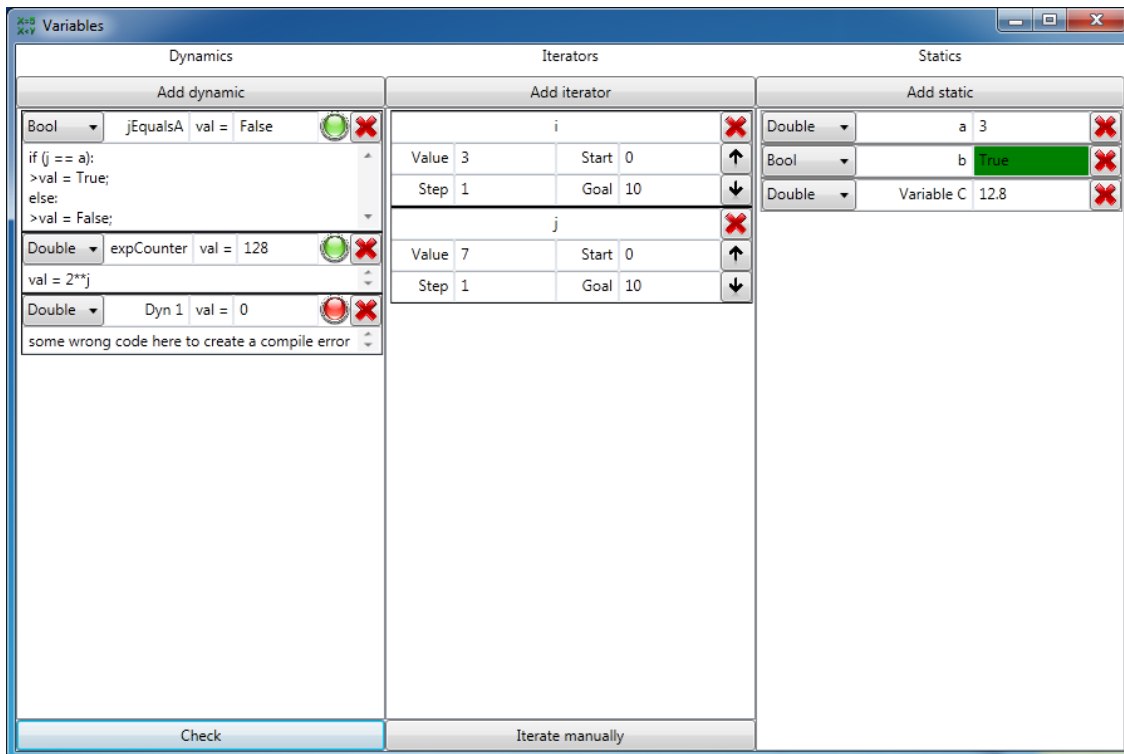


Figure 11: Variables window with some variables

## 4.6 Errors

The error window shows all errors collected at one point. It will be available in both the Pulseblaster and the National Instruments controller. The error window is split up into three categories:

- Errors for the National Instruments controller.
- Errors for the variables window.
- Errors for the Pulseblaster controller.

Each can be expanded to show all errors or be fold to hide all errors. There are two types of errors:

- Normal errors: If this error is corrected, it will disappear from the error window.
- Critical errors: If this error occurs, it will stay until the user deletes it.

Each category can be cleaned up (to delete all normal errors, broom symbol) and deleted (delete all errors, even the critical ones, X symbol).

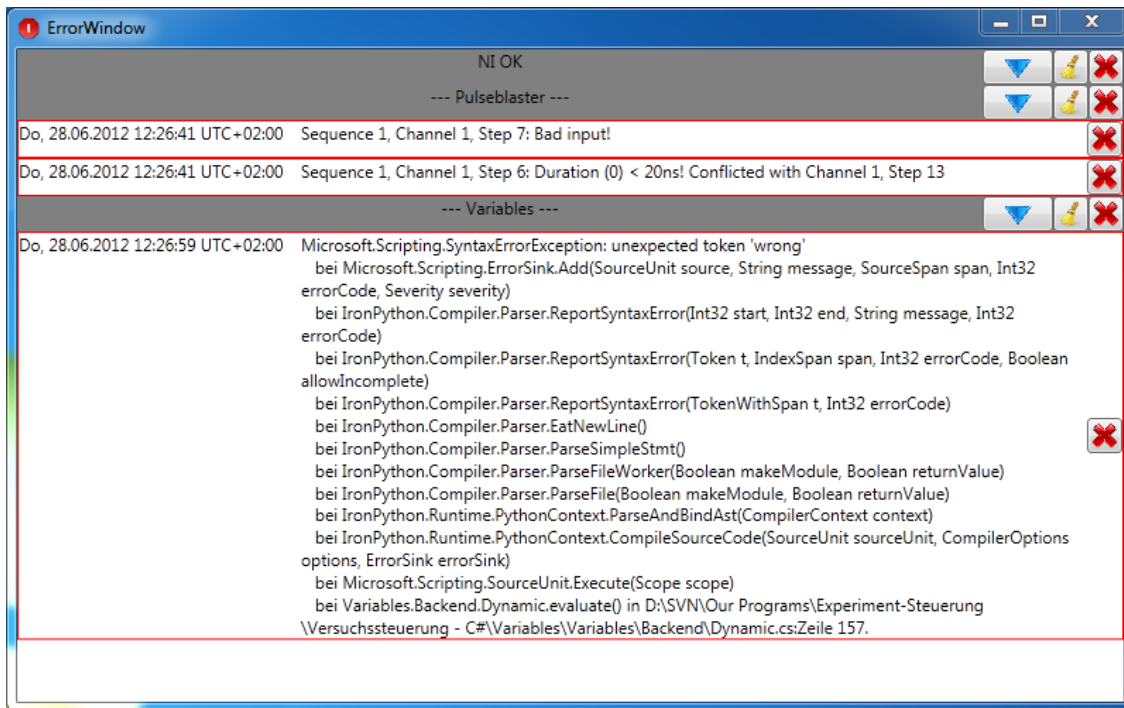


Figure 12: Error window with some errors

## 5 Measurements

This chapter will show the measurements that were made to check whether the board works as it is expected and fulfils the required specifications.

### 5.1 Reflections at the transition from a ribbon cable to a $50\ \Omega$ cable

When connecting all 21 channels of the Pulseblaster board, this must be done with two 26-pin IDC-headers. This connector and the connected ribbon cable do not have a wave resistance of  $50\ \Omega$ , so there will always be reflections between the ribbon cable and BNC connectors for connecting  $50\ \Omega$  wires.

#### Experimental setup

A ribbon cable is shortened to several lengths: 3.25 m, 1.7 m, 1.15 m and 0.15 m. On the one side, there is a IDC connector. On the other side, there are BNC connectors. All channels are connected to an oscilloscope with different lengths of  $50\ \Omega$  cables to keep the total length constant for all channels. Figure 13 shows the reflections at different lengths of the ribbon cable with a pulse length of 2.5 ns.

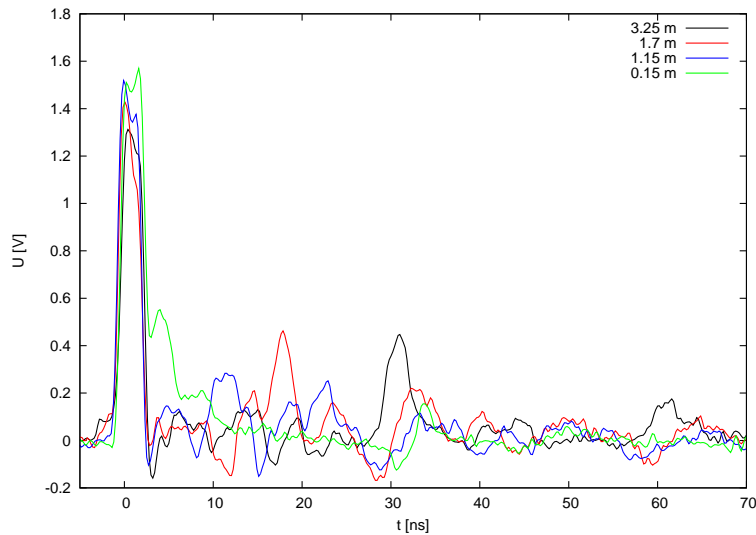


Figure 13: Reflections with different cable lengths

Even the 0.15 m cable causes a visible reflection which could cause problems. The best solution is using an adaptor board which fits directly onto the IDC connector. SpinCore is currently evolving an adaptor board which could be used (see figure 14). The distance to the  $50\ \Omega$  cable is about 1 cm, so this will reduce the reflections. There is also the possibility to create a printed circuit board in  $50\ \Omega$ -design, so there will be no reflections. Therefore, the board and the track must have a certain thickness depending on the board material.

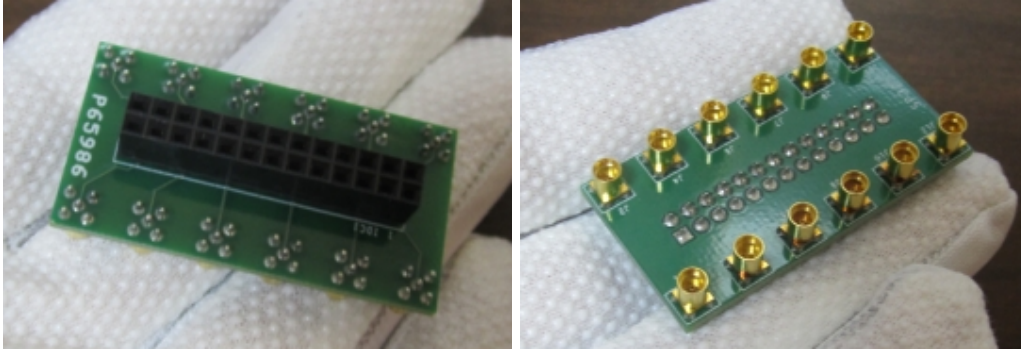


Figure 14: Adaptor board for the Pulseblaster board [1]

## 5.2 Pulse shape

To analyse the pulse shape, for example to check the pulse length, channels 1 to 3 send a 20 ns pulse which is measured with a 1 GHz LeCroy oscilloscope.

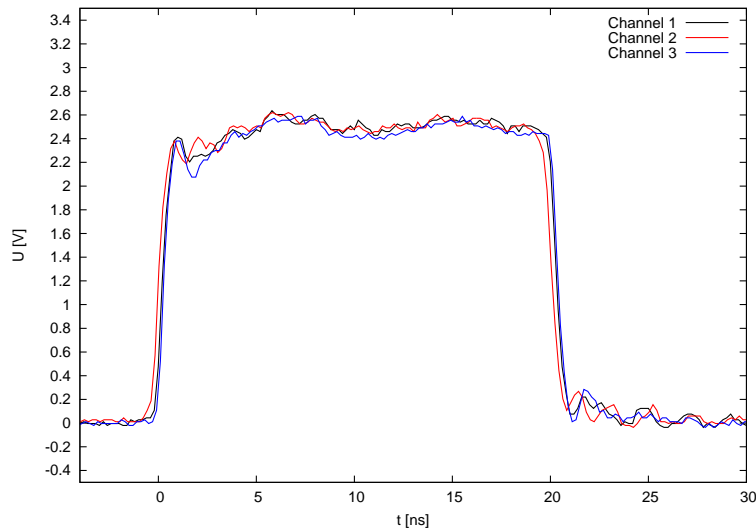


Figure 15: Pulse shape of the different channels

The HIGH voltage is  $(2.4 \pm 0.2)$  V. The pulse lengths at half maximum (1.2 V) are:

Channel	Rising edge	Falling edge	Width
1	0.2 ns	20.2 ns	20.0 ns
2	0.0 ns	20.0 ns	20.0 ns
3	0.3 ns	20.3 ns	20.0 ns

Table 5: Pulse lengths

The pulse lengths are 20.0 ns, the edges can only be determined with an inaccuracy of 200 ps. The measurements were taken at 1 GHz. This means a 1 ns resolution. Within the accuracy of measurement, the pulses are exactly 20 ns.

### 5.3 Jitter

#### 5.3.1 Measurements with an Oscilloscope

Channels 1 to 3 send a 20 ns pulse which is measured with a 1 GHz LeCroy oscilloscope.

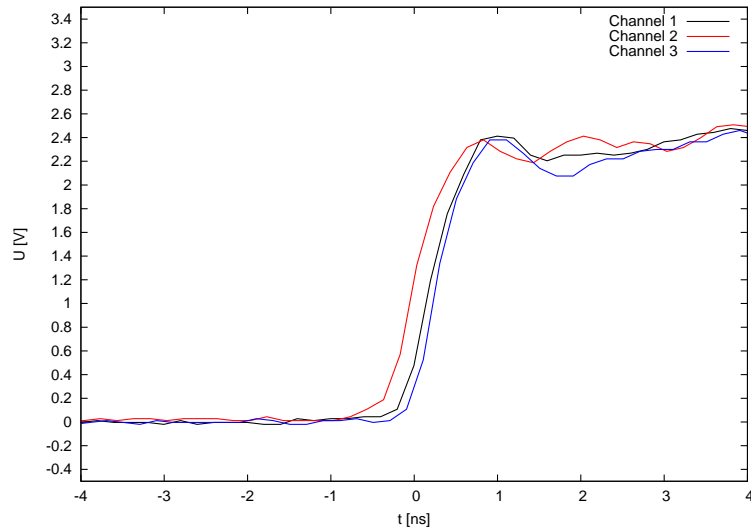


Figure 16: Jitter of the rising edge

Channel	Rising edge
1	0.196 ns
2	0.032 ns
3	0.308 ns

Table 6: Positions of rising edge

The time between channel 2 and 3 is 0.276 ns. Assuming a velocity of  $2 \cdot 10^8$  m/s, this is a distance of 5.5 cm. This effect could be traced back to different wire lengths on the Pulseblaster board.

### 5.3.2 Measurements with a SensL HRMTime TDC

The HRMTime TDC is a 4-channel module. Each channel has a start and a stop input. After receiving a start input signal, the device registers every stop input and lists all stop events in a list. There were made 4 measurements with the HRMTime TDC:

Measurement 1: Channel 1 is sending a start pulse and 2 is sending stop pulses (20 ns pulse each 1 ms).

Measurement 2: Channel 2 is sending a start pulse and 1 is sending stop pulses (20 ns pulse each 1 ms).

Measurement 3: A frequency generator is sending a start pulse and channels 1 to 4 are sending stop pulses (20 ns pulse each 1 ms).

Measurement 4: A frequency generator is sending a start pulse and channels 1, 2 and 4 are sending stop pulses (20 ns pulse each 1 ms) via the self-built IDC to BNC adaptor.

#### Measurement 1 and 2

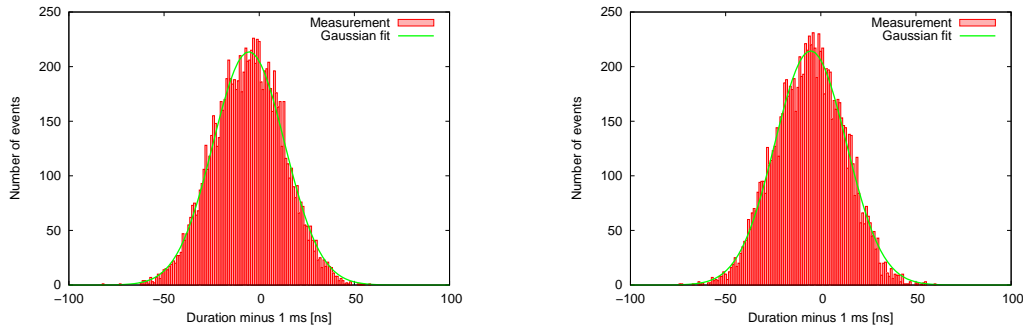


Figure 17: Histogram of the first (left) and second (right) measurement

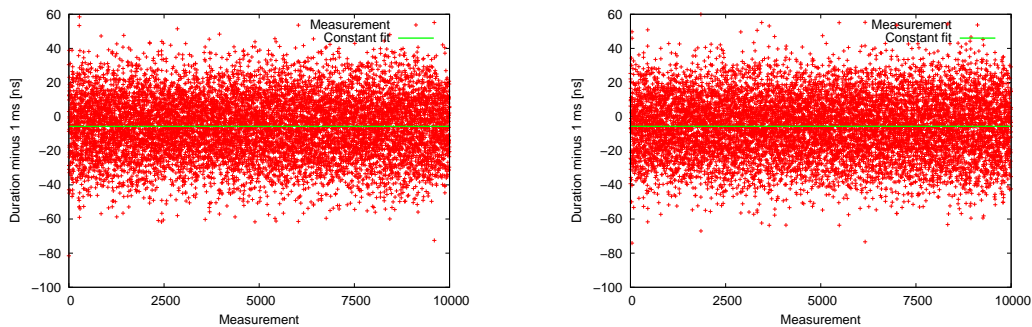


Figure 18: Time-dependency of the first (left) and second (right) measurement

Figure 17 shows the histogram for the pulse distances (minus 1 ms). The pulse distances are 5.4227 ns (first) and 5.3321 ns (second) shorter than they should be (gaussian fit). Figure 18



shows that this jitter is not time-dependant. The constant fit returns average pulse distances of 5.7044 ns (first) and 5.6782 ns (second) shorter than 1 ms, what the pulse distances should be. The two channels behave the same way, there is no major difference in the histograms. This means that this effect is not coming from a single channel, this jitter effect may come from either the intern clock of the Pulseblaster board or the intern clock of the HRMTime TDC.

### Measurement 3 and 4

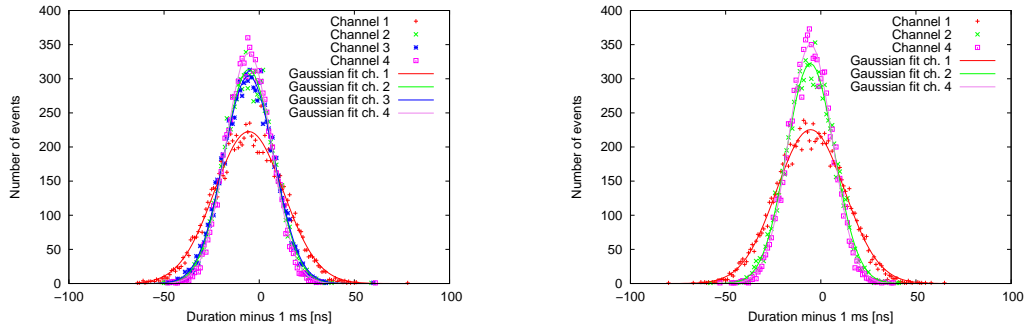


Figure 19: Time-dependency of the third (left) and fourth (right) measurement

Using the self-built IDC to BNC adaptor does not change the measurement to the worse. Channel 1 has the greatest spreading of measured values, channel 4 the smallest. The reason for this could be some differences in the wiring on the Pulseblaster board. The pulse distances are shorter as 1 ms by the following values:

Channel	Board output connector	Self-built adaptor
1	5.4374 ns	5.3645 ns
2	5.4719 ns	5.4372 ns
3	5.4677 ns	not measured
4	5.4836 ns	5.4509 ns

Table 7: Time differences to 1 ms

Figure 20 shows the difference of the pulse start time of channel 1 to the other channels.

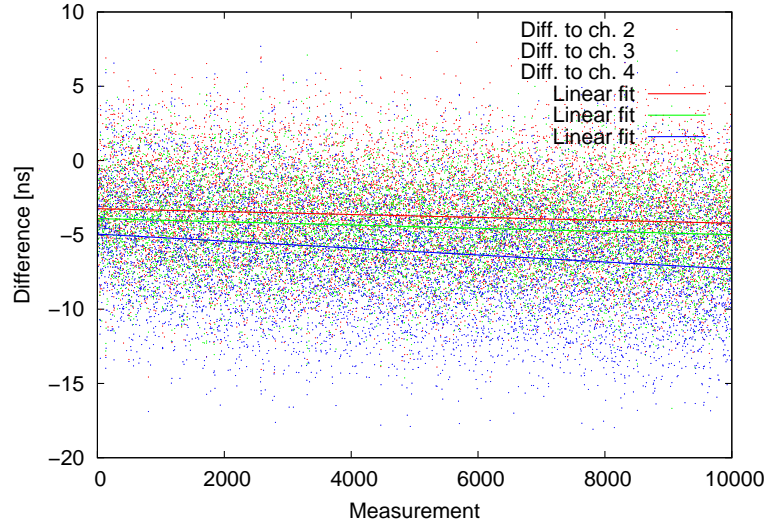


Figure 20: Time difference of the pulse start time of channel 1 to the other channels

The time values are much greater than shown in figure 16. Figure 16 has been watched with the oscilloscope over a short period of time and the image did only change negligibly and not as figure 20 shows it. This points out, that there must be some problems within the measurement that were not solved yet. The divergence of the pulse start times points on a synchronization problem inside the HRMTime Module. The values of table 7 could be an effect either of the Pulseblaster board, that the clock is running a bit too quickly, or of the HRMTime TDC, that the clock is running a bit too slowly.

## 5.4 Crosstalk

For this experiment, the same setup is used as in 5.1. A 2.5 ns pulse is applied to channel 2. Figure 21 shows the crosstalk depending on the length of parallel ribbon cable. At a length of 1.7 m, there is a crosstalk about 25 %. At 1.15 m, there is only 7 %. For lengths smaller than 1 m, this effect can be neglected comparing to the reflections and the noise. The SpinCore adaptor board has lengths about 1 cm, a self-build IDC to BNC cable has a length of 25 cm, so here will only be a negligible influence to the signal.

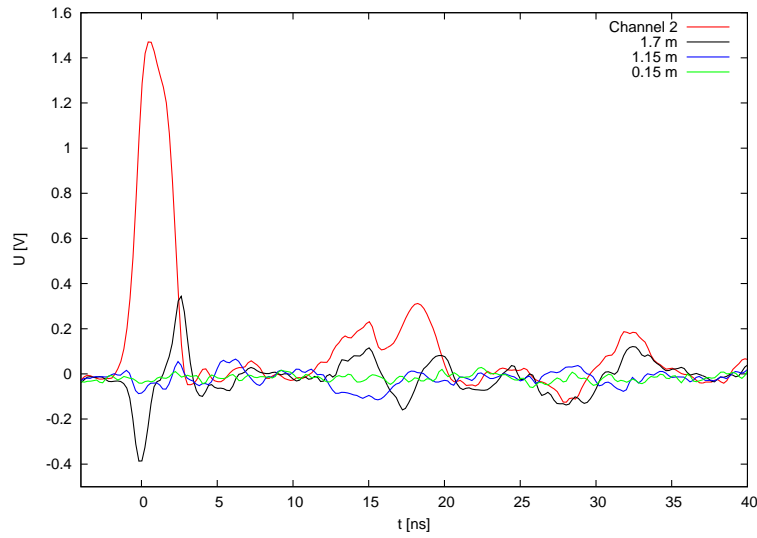


Figure 21: Cross-talk with different lengths of parallel cable to channel 2

## 6 Outlook

The Pulseblaster board has been installed and implemented and is ready to be used. The Pulseblaster controller, the variables and the errors are also working correctly.

### 6.1 Computer control system

The computer control system has to be extended. At first, the Pulseblaster controller and the National Instruments controller have to be put together into one big program. This was not performed yet because the National Instruments controller had to be used for some applications and testings on the experiment already and so it was more important to have a running National Instruments controller without the other parts than to have the whole program with the Pulseblaster controller. Afterwards, the error handling and the variables have to be included into the National Instruments controller. The Pulseblaster controller has been used for some small test applications in the laboratory already and there were no failures yet.

### 6.2 Future in the laboratory of the Pulseblaster board

The pulse shape measurement shows a very precise pulse. As calculated in 3.2.2, this pulse is suitable for the deterministic excitation. The Pulseblaster board is perfectly matching to the requirements.

### 6.3 Jitter measurement with SensL HRMTime TDC

The jitter measurements provide conflicting results. The qualitative measurement with the oscilloscope shows only negligible jitter whereas the quantitative measurement with the HRMTime TDC shows extreme large jitter. These measurements have to be repeated with a greater range to ensure the proper functionality of both the Pulseblaster board and the HRMTime TDC device. The time between the pulses has to be modified to determine the source for the bad data. The connections have to be monitored with an oscilloscope to be shure that there are no reflections. The measurement for the pulse distances should also be repeated with another device to determine if the Pulseblaster board is sending too short pulses or the HRMTime TDC is measuring not the correct time. Also, the measurement of the pulse start times has to be repeated with an oscilloscope to monitor whether the start times of the Pulseblaster channel outputs are diverging or whether this is a internal synchronisation problem of the HRMTime TDC.

## References

- [1] SpinCore Technologies, <http://www.spincore.com/>.
- [2] V. Bendkowsky, B. Butscher, J. Nipper, J. P. Shaffer, R. Löw, and T. Pfau (2009): **Observation of ultralong-range Rydberg molecules**, Nature 458, 1005-1008, page 1.
- [3] W. K. Lai, V. Buzek, P. L. Knight (01.11.1991): **Dynamics of a three-level atom in a two-mode squeezed vacuum**, Physical Review A, Volume 44, Number 9, page 6045, (3.5).
- [4] D. A. Steck (25.09.2001): **Rubidium 87 D Line Data**, <http://steck.us/alkalidata/rubidium87numbers.1.6.pdf> [31.7.2012], page 17.
- [5] B. Butscher (29.05.2011): **A Rydberg Interferometer**, page 23.
- [6] S. Hofferberth (Munich, 10.05.2012): **Hollow fibers and Rydberg atoms - Novel approaches to few-photon optical nonlinearities**.
- [7] M. Fox (2006): **Quantum Optics, An Introduction**, page 174, (9.32).
- [8] M. Fox (2006): **Quantum Optics, An Introduction**, page 12, (2.28).
- [9] M. Fox (2006): **Quantum Optics, An Introduction**, page 77, (5.1).
- [10] D. Meschede: **Gerthsen Physik**, 24. überarbeitete Auflage, page 799, (16.12).
- [11] D. Meschede: **Gerthsen Physik**, 24. überarbeitete Auflage, page 723, (15.16).
- [12] P. A. Tipler, G. Mosca: **Physik für Wissenschaftler und Ingenieure**, 2. Auflage, page 1166.
- [13] P. A. Tipler, G. Mosca: **Physik für Wissenschaftler und Ingenieure**, 2. Auflage, page 1169, (36.11) and (36.12).