LUDWIG-MAXIMILLIANS-UNIVERSITÄT
MÜNCHEN

MASTER'S THESIS

Building and Testing a miniaturized QKD sender

Thomas Schwarzwälder

supervised by
Prof. Dr. Harald Weinfurter

21.06.2018

LUDWIG-MAXIMILLIANS-UNIVERSITÄT
MÜNCHEN

MASTERARBEIT

Bau und Test eines miniaturisierten QKD Senders

Thomas Schwarzwälder

betreut von
Prof. Dr. Harald Weinfurter

21.06.2018

Contents

Chapter 1

Introduction

More and more aspects of our professional and private lives move into the digital domain. Smartphones can be seen as personal electronic assistants, which take care of our social lives, dating, health, navigation, entertainment as well as financial issues. All of those tasks are typically achieved via a combination of some local computation on the smartphone, together with some remote data processing on a sever. The exchanged information is often first encrypted and then transmitted over the Internet. The security of modern cryptographic techniques is usually based on unproven, but widely believed assumptions. These techniques are practically secure, but theoretically already broken. There exist algorithms for a quantum computer, which can be used to break modern encryption methods, like RSA. However, today there exists no quantum computer that can actually do this, but a lot of research is currently conducted in this field. Companies like Google, IBM, Microsoft and Intel are currently trying to build quantum computers, each based on different approaches. Most remarkably, Google recently announced that they are working on a 72 qubit universal quantum computer with low error rates, which they want to complete building this year [1]. IBM already offers a cloud-computing service where a 16 qubit quantum computer can be programed [2]. Note however that this current absence of a powerful enough quantum computer does not mean that todays communication is still safe. In principle, an eavesdropper can store all of todays encrypted classical communication and break the encryption once a suitable quantum computer is built. This means that safe cryptographic methods are not an issue of the future but of the present. Fortunately there already exists a provably secure cryptographic method, the One-Time-Pad (OTP). This technique requires the distribution of a symmetric key between the two parties. This is done via a key exchange protocol, which is typically the weakest link in the chain and similarly vulnerable like RSA encryption method. Here Quantum Key Distribution (QKD) can be used to provide a higher security standard, than current classical key exchange protocols do.

This Master's thesis concerns itself mostly with the characterization of a new VCSEL (vertical-cavity surface-emitting laser) array, which acts as the light source for a QKD sender module. To ensure that the array is suitable for QKD and does not weaken its security promise several tests need to be performed. Additionally, the design of a completely new VCSEL carrier board is also presented, which features new connectors and is only half as big as the original board. This new board completes the redesign of the complete sender module. From these redesigned modules the mainboard is then tested, together with the Field-programmable gate array (FPGA) configuration.

The exact organization of this thesis is as follows. Chapter 2 will provide a overview of classical cryptography, which is followed up by a review of the key concepts of quantum mechanics. Then an introduction to QKD is provided, with a focus on the BB84 protocol, where theoretical, as well as experimental issues are addressed. In chapter 3 a description of the hand-held QKD experiment of the experimental quantum physics group of the LMU can be found. The three following chapters focus on my experimental work. Chapter 4 describes the assembly of an old VCSEL carrier board with a new VCSEL array, as well as the design of a completely new VCSEL carrier board. In chapter 5 this new VCSEL array is characterized regarding its practical use for QKD. Here several measurements are described, together with their results and consequences. Chapter 6 deals with the newly designed electronics mainboard and with the FPGA configuration, which is the main processor of the electronics module of the sender unit. Here the FPGA signal to the electronics components is measured and compared with the needed patterns. All of the gathered results are summarized in the final chapter, which ends with an outlook.

Chapter 2

Theoretical Foundations

In this chapter I will first introduce classical cryptographic methods that allow two parties to communicate securely. A general problem common to these methods is the distribution of an initial key or secret. This problem will be explored and two solutions will be presented, one of which is QKD. The uniqueness and strength of QKD will be explored, after a short review of the key concepts of quantum mechanics on which QKD rests.

2.1 Classical Cryptography

The field of cryptography deals, among other things, with different techniques and their security analysis, which allow two parties, usually called Alice and Bob, to communicate securely (privately) via an insecure (public) classical channel. This channel might be monitored or controlled by an adversary or eavesdropper, typically called Eve[1] [37].
For the following we assume that Alice wants to send a message m_A to Bob. Alice and Bob both posses a secret key, denoted k_A for Alice and k_B for Bob. The message and the key(s) consist of symbols, taken from some previously agreed upon alphabet[2]. Since the keys are secret, only Alice knows k_A and only Bob knows k_B. Alice transforms the message m_A into a ciphertext c via a cipher or encryption-function[3] $Enc(m_A, k_A) = c$. This ciphertext c is then transmitted to Bob, via the insecure classical channel. Bob receives the ciphertext c and reconstructs the orig-

[1] The details of Eve's exact capabilities become important during a security analysis of a given protocol. A careful analysis may then show that this protocol can only be considered secure for some attacks, but it might be insecure for others.

[2] An alphabet is a list of symbols.

[3] This encryption function is in general an algorithm which depends on the message m_A and the key k_A.

inal message via a cipher or decryption function[4] $Dec(c,k_B) = m_B$.
If the protocol is correct, we have $m_A = m_B$ and the message has been successful transmitted to Bob. If the protocol is secure, an eavesdropper has gained no information about m_A, m_B, given just c and the knowledge of the encryption function Enc and the decryption function Dec (but not of k_A, k_B, since any cryptography scheme is pointless if the keys are known to the adversary). In modern security proofs one always assumes that the adversary knows the encryption and decryption technique, this is known as Kerckhoffs's principle. In the past the encryption and decryption methods were held secret, in order to make the complete cryptographic scheme more secure. In modern times the encryption and decryption techniques are made publicly available so that everyone may check the scheme for errors and weaknesses that the original inventor might have missed. All of this can be made more mathematically rigorous. Modern cryptography provides mathematical proofs of security, given precise mathematical assumptions and definitions of all involved concepts (like the otherwise vague notion of security, secrecy and ignorance for example)[5].
If Alice and Bob both use related keys, so $k_A = f(k_B)$, or in the simplest case $k_A = k_B$, they can use a symmetric scheme, also often called private-key encryption, since both keys must be held secret from other unauthorized parties (like Eve). If Alice and Bob hold unrelated keys, so $k_A \neq f(k_B)$, they use an asymmetric scheme, also often called public-key encryption. Here it is assumed that $f(k_B)$ is a function which can be easily evaluated by the two parties[6]. Both schemes will be further explored in the following chapters. These schemes assume that Alice and Bob both already have these keys. Therefore a way to distribute these keys is needed, before any cryptographic scheme can be implemented. We will see that the key exchange for asymmetric schemes is rather easy, whereas the key distribution for the symmetric method is much more difficult to achieve. After we explored both cryptographic methods we will take a look at the key exchange problem and possible solutions to it.

[4] This decryption function is in general an algorithm which depends on the ciphertext c and the key k_B.

[5] The field of modern cryptography started in 1949 when Claude Shannon published his paper *Communication Theory of Secrecy Systems* [3]. There Shannon derives theorems based on his axioms and definitions.

[6] The asymmetric RSA scheme (section 2.1.3), also uses strictly speaking two keys which are related by a function, but this function can not be easily evaluated (see section 2.1.2 for more on hard and easy functions).

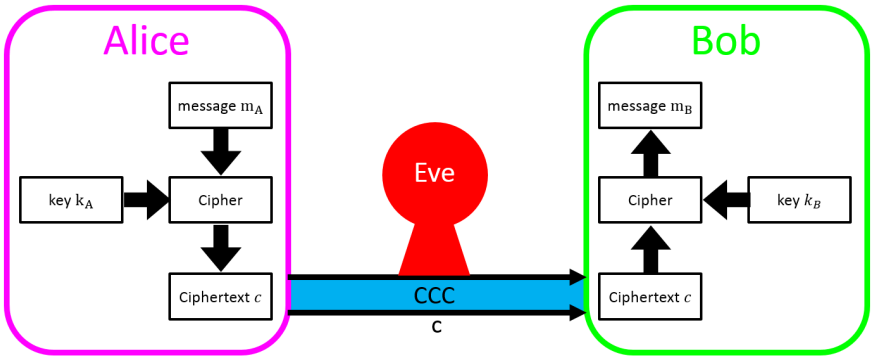


Figure 2.1: Alice and Bob want to share a secret message over a classical communication channel (CCC). Alice encrypts the message m_A with her cipher and her key k_B and then transmits the ciphertext c via the CCC. Bob decrypts the ciphertext c with his key k_B and his cipher to reconstruct the message m_B. Eve is monitoring the public channel and tries to reconstruct the message from the ciphertext. If the cipher is secure Eve can not learn anything about the message, so Alice and Bob have successfully exchanged a secret via the insecure/public channel that connects their safe environments, which Eve can not penetrate.

2.1.1 Codes and ciphers

Before the field of cryptography is further explored it is important to stress the difference between codes and ciphers.

A code operates on semantics, it transforms the meaning of (words or) symbols. Therefore to keep the meaning invariant the symbols to represent a certain idea or concept are exchanged. The code is written down in a codebook and this codebook is needed to encode and decode a message correctly.
Code example: We will often use a code where we encode each letter in the (English) alphabet with a number corresponding to its position in the alphabet. So *A* will be encoded as 0, *B* as 1 and so on[7]. If we want to decode a certain number we map it to the letter with the position of that number in the alphabet. In the following chapters this code will be referred to as the *AN-code* (alphabet-number-code).

A cipher operates on syntax, it transforms the structure of (words or) symbols. A cipher is an algorithm which needs a key to transform a message into a ciphertext.
Cipher example: In the following chapters we will take a deeper look at the Caesar cipher, the Vigenère cipher and the one-time pad in particular.

[7] We may also use binary numbers if it is more convenient.

2.1.2 Cryptanalysis

The field of cryptanalysis deals with breaking cryptographic algorithms or ciphers, used for encryption and decryption, but also attacking their implementations. Thus one can attack the theoretical side, by looking for a mathematical or logical weakness in the algorithm, or one can attack the practical side, by looking for a weakness in the physical implementation. The physical implementation may have certain elements which were not present in the theoretical model or which only come close to the ideal theoretical elements needed. The theoretical weaknesses of an algorithm can in general be analyzed for all implementations which use that algorithm. The practical realization of a given cryptographic protocol on the other hand may vary strongly from implementation to implementation, which makes it much harder to ensure that there is no weakness in it. A weakness in the implementation is typically called side-channel.
A cryptographic protocol is considered to be (theoretically) secure if one can prove mathematically the security of a protocol, based only on the assumptions of this protocol. Most cryptographic schemes are (only) practically secure. This means that additionally to the assumptions of the specific protocol further assumptions are made, e. g.,

1. **The mathematical assumption**
   The adversary does not have any algorithms at his or her disposal which can solve mathematical problems that are considered to be hard[8] (by mathematicians and computer scientists).

2. **The computational assumption**
   The adversary has limited computational power at his or her disposal.

The first assumption states that the adversary has not made any mathematical breakthrough that is unknown to other experts in the field. Or in other words, if no expert has yet found a way to solve a certain hard mathematical problem, then the assumption states that the adversary has also not found an easy (or fast) way to do so. Most prominent examples for such problems are (among others) the integer factorization problem and the discrete logarithm problem, both of which will be discussed later.
The second assumption basically states that the adversary does not have access to unlimited computational power. The limit that one assumes can vary from security proof to security proof, but often it is assumed that the adversary has access to a (classical) computer with very large computational power, like the computational

[8] The time for an algorithm to solve a "hard" problem grows exponentially with the size of the input. An "easy" problem grows polynomially in time with the input (loosely speaking).

power of all computers on earth, and sometimes even more power is granted, trying to anticipate future advancements. However, it is also often assumed that the adversary only has a classical computer, but not a quantum computer[9], at his or her disposal.

Examples for hard problems:

1. **The integer factorization problem**
   It is assumed to be hard to factorize an arbitrary integer into its prime factors.

2. **The discrete logarithm problem**
   It is assumed to be hard to solve the equation[10] $y = g^x mod(P)$ for x given only g, P, y.

These two problems are considered to be hard, which means that nobody has yet found an easy way to solve them efficiently[11] on a classical computer. At the same time there is no proof that such a method does not exist. Thus it is widely believed that there exists no such method, but it is not proven and thus we can not be certain. In fact, algorithms have been found to solve these two problems, but a quantum computer is necessary to run them. The field of "post-quantum"cryptography deals with ciphers which are secure even against attacks employing a quantum computer.

So far only one cipher is proven to be mathematically secure, without using any mathematical or computational assumptions, this cipher is the OTP[12]. The OTP will be explained in detail in section 2.1.4.

2.1.2.1 Shor's algorithm

In 1994 Peter Shor published a paper [4] in which he presents two algorithms for a hypothetical quantum computer, that can easily (or in polynomial time) solve the integer factorization problem and the discrete logarithm problem. Both distinct algorithms are typically referred to as *Shor's algorithm*. Therefore any cryptographic method whose security is based on these two problems would become

[9] This will be further explored in section 2.1.2.1.

[10] In this equation g denotes the generator and P the prime modulus.

[11] The computational time to solve a problem efficiently does not grow exponentially but polynomially with the size of the input.

[12] Which means that the OTP is unbreakable.

obsolete once a quantum computer, that can run Shor's algorithm, has been constructed. Today no quantum computer with a large enough memory and sufficiently low error rate does exist, which is why these schemes are still considered to be practically secure. However, the existence of Shor's algorithm questions the life-time of schemes whose security is based on these two problems, as well, and even more importantly, the security of data already encrypted using those schemes.

2.1.3 Asymmetric Encryption

Asymmetric encryption is also typically known as public-key cryptography. The reason is that in such a scheme Alice (or Bob) possesses two keys, a private key and a public key. The public key is, as the name suggests, publicly available. Anyone who wants to send Alice a secret message uses her public key to encrypt a message intended for Alice. She then decrypts the ciphertext with her private key and thus recovers the message. Additionally, Alice can also use her private key to sign a message and the recipient can check the validity of her signature via her public key. Since Alice can just openly publish her public key there is no key distribution problem, Alice only has to make sure that her private key stays private.
Asymmetric encryption schemes are typically slower than symmetric ones, which grant the same level of security, because they need bigger keys to reach that same level, which in turn increases the runtime. Due to this asymmetric encryption is sometimes used to distribute a key for a symmetric encryption scheme. Such a scheme would be in total a hybrid scheme, but parts of it can be associated to symmetric and asymmetric methods.
The theoretical idea for public-key cryptography was first sketched in a paper by W. Diffie and M. Hellman [5]. In this paper they did not present a definite protocol for public-key cryptography, that was done later by R. Rivest, A. Shamir and L. Adleman (RSA) [6]. The RSA algorithm is one of the first and most widely used public-key encryption system. Its security is related to the difficulty of evaluating Euler's totient function, on the hardness of the integer factorization, and of the discrete logarithm problem. RSA is considered practically secure if a key with length of 1024 bits or more is used. RSA with a key of length of 768 bits (232 digits) has already been broken and it can also be broken for arbitrary key lengths with a quantum computer and Shor's algorithm. Strictly speaking a classical computer can also break RSA for arbitrary key lengths, but on a classical computer the runtime grows exponentially with the size of the key, whereas on a quantum computer it grows linearly. This much shorter runtime makes an attack with a quantum computer more practical.

2.1.4 Symmetric Encryption

In symmetric encryption schemes the same key is used for encryption and decryption. Symmetric ciphers are typically faster than asymmetric ones, because the same level of security can be guaranteed with shorter keys (than an asymmetric protocol would need). Drawbacks of symmetric encryption schemes are the need for a secure key distribution protocol and the resulting complex key management that is required if symmetric keys are to be used with a large group of people. In the following a couple of symmetric ciphers will be explored in more detail.

2.1.4.1 The Caesar cipher

The Caesar cipher is a monoalphabetic cipher that shifts the whole alphabet by a fixed amount. The shift is determined by the key k, which is a number, whose maximal value depends on the size of the alphabet in the used language. If the alphabet has N letters the key should be at least one and at most $N-1$. Each letter is encoded into a numerical value (typically its position in the used alphabet, which we call here the *AN-code*). Then the key-value k is added to the letter value l, using modular arithmetic $l+k = s \ mod(N)$. After that the shifted number s is decoded back into a letter, again via the *AN-code*. An example for this technique is given in picture 2.2 were the message $ERWIN$ gets encrypted with a key $k = 7$. The decryption is done similarly, the only difference being that the key is subtracted instead of being added.

There are two effective ways to crack the Caesar cipher. The short message we used can be easily cracked by using a brute-force attack. Here a list is created for every possible key value $k \in [1, N-1]$. If such an attack is performed on the example cipher $LYDPU$ a list is retrieved where 25 words correspond to nonsensical letter combinations and only one key value gives a normal word or name - $ERWIN$.
For longer texts that are encrypted with the Caesar cipher a frequency-analysis can also be used to determine the original message. Here one uses the fact that in languages, for example the English language, not all letters are used equally often. This gives each language a distinct letter distribution or fingerprint. The Caesar cipher leaves the shape of this fingerprint intact and only shifts it by an amount determined by the key. This enables an attack to deduce the shift by looking at the letter frequency of the ciphertext and comparing it with the letter frequency of the used language.

E	R	W	I	N	Encode
4	17	22	8	13	Encrypt
11	24	3	15	20	Decode
L	Y	D	P	U	

Figure 2.2: An example for the encryption with the Caesar cipher: First we encode the message (via the *AN-code*) in terms of numbers, according to their position in the alphabet. Then we perform the shift, addition modulo 26, determined by the key (here the key $k = 7$ and the number of letters in the used alphabet. Then we decode the shifted numbers back to letters using the same code as before and thus arrive at the cipher $LYDPU$. The receiver performs the same steps for the en- and decoding, but for the decryption he subtracts the key to transform the ciphertext $LYDPU$ back to the original message $ERWIN$.

2.1.4.2 The Vigenère cipher

The Vigenère cipher is a polyalphabetic cipher. Here several distinct shifts of the alphabet are performed. These shifts are determined by the key k, which is no longer just one number but a sequence of numbers (or a word which is then encoded into a sequence of numbers via the *AN-code*).

Example: we want to encrypt the message *hide in the cellar* with the key or keyword *apple*. First we encode each letter via the *AN-code*, which turns our message m_{Text} into the sequence m_{Nr} and the keyword K into $(0,15,15,11,4)$. Now we again add[13] these and extend the key-length to the message-length by repeating the key K over and over (this repeated key is denoted by the letter k). The modular addition gives us the number sequence c_{Nr}, which we again decode into letters via the *AN-code* to finally end up with the ciphertext c_{Text}.
An example for the encryption with the Vigenère cipher is given in table 2.1.
This scheme is more difficult to break than the Caesar cipher, since it basically uses a bunch of different Caesar ciphers with different keys, which all have to be broken at the same time. However, this cipher can still be broken, but the attack is now more complex. Due to this the cipher was unbroken for about 300 years, but today there are many techniques known to break this cipher. For example, this cipher can still be attacked via frequency analysis because the letter distribution that the cipher creates is not completely flat, but still maintains some fingerprint or trace which can be used to break the cipher.

[13] We use again modular addition with mod(26).

m_{Text}		h	i	d	e	i	n	t	h	e	c	e	l	l	a	r
m_{Nr}		7	8	3	4	8	13	19	7	4	2	4	11	11	0	17
k	$\oplus$	0	15	15	11	4	0	15	15	11	4	0	15	15	11	4
c_{Nr}		7	23	18	15	12	13	7	22	15	6	4	20	20	11	21
c_{Text}		h	x	s	p	m	n	i	w	p	g	e	a	a	l	v

Table 2.1: The message *hide in the cellar* is encrypted via the Vigenère cipher with the keyword *apple*. First each letter is encoded via the *AN-code*, which turns our message m_{Text} into the number sequence m_{Nr} and the keyword K into $(0,15,15,11,4)$. We repeat K until it is as long as m_{Nr}. Then we compute $c_{Nr} = m_{Nr} \oplus kmod(26)$, which we again decode into letters via the *AN-code* to finally end up with the ciphertext c_{Text}.

2.1.4.3 The one-time pad

The OTP is a provably secure cryptographic method. It can be seen as a special case of the Vigenère cipher where the (binary) keyword $k = k_1k_2...k_N$ with $k_i \in \{0,1\}$ and $i \in \{1,2,...,N\}$ is as long as the message $m = m_1m_2...m_N$ with $m_j \in \{0,1\}$ and $j \in \{1,2,...,N\}$. The key must be a completely random string of (binary) numbers. The message m is then bitwise XORed with the key k, which in turn creates a random cipher $c = m \oplus k$ with $c = c_1c_2...c_N$ and $c_p \in \{0,1\}$ where $p \in \{1,2,...,N\}$. The ciphertext is perfectly secret, in the sense of Shannon secrecy, which means that given the ciphertext c nothing about the message m can be deduced from it. The OTP completely erases any language fingerprint within the message via the random key, it produces a completely flat letter distribution. The OTP is the only known mathematically secure cipher and is thus unbreakable. For the decryption the key k is again added to the ciphertext c, which results in the reconstruction of the original message m, since $c \oplus k = m \oplus k \oplus k = m$.

It is instructive to compare the OTP with the Caesar cipher. Earlier the Caesar cipher was used to encrypt the message *ERWIN*. The Caesar cipher offers for a key $k \in [1,25]$ exactly 25 different encryptions or possible ciphertext. If the OTP is used to encrypt *ERWIN* we get $26^5 \approx 11.8 * 10^6$ possible ciphertexts, which corresponds to a list of all possible five letter words. This illustrates why even a brute-force attack is fruitless against the OTP. The randomness of the key directly causes the randomness of the ciphertext. Note that a brute-force attack is equivalent to guessing the message without having any hints.

Despite its strength in practice this method is rarely used, because it can be very cumbersome and difficult to exchange keys which are as long as the message. The security would be also reduced if a key is used twice for different messages,

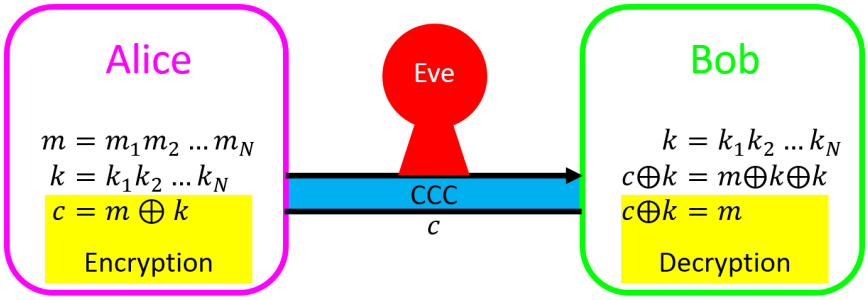


Figure 2.3: One-time pad: Alice encrypts her binary message $m = m_1 m_2 ... m_N \in \{0,1\}^N$ by performing an addition mod(2), denoted by $\oplus$, of m and the key $k = k_1 k_2 ... k_N \in \{0,1\}$ to create the ciphertext $c = m \oplus k \in \{0,1\}^N$. Alice sends the ciphertext c to Bob via a classical communication channel. Eve can read and copy the ciphertext c. Bob decodes the ciphertext by repeating the addition mod(2) with the same key. Since the OTP is unbreakable Eve can not deduce anything about m given c. Alice and Bob have thus successfully exchanged a secret message m.

which is why for every message a new key has to be used and thus established beforehand. This makes the OTP unpractical enough that it is only used where absolute secrecy is essential.

2.1.4.4 AES

The Advanced Encryption Standard (AES) is the new symmetric encryption standard by the U.S. National Institute of Standards and Technology (NIST)[14], which is also approved by the National Security Agency (NSA). AES is the successor of DES, which is no longer considered to be secure[15]. During and after its selection by NIST this protocol has been analyzed by many experts and nobody has yet found a flaw that would compromise its practical security. Many modern computer applications of symmetric encryption use AES or are built upon it, like the SSH protocol for secure network communication and the WPA2 encryption for wireless networks.
There is no known practical attack[16] that can break AES, not even a quantum algorithm. At the same time, there is no proof that no such attack or algorithm does exist.

[14] AES was established as the new standard for symmetric encryption in 2001.
[15] Nevertheless people still use an improved version of DES, which is called 3DES.
[16] Brute-Force is here considered to be an impractical attack.

2.1.5 The Key Distribution Problem

To establish a secure communication channel, say via the OTP, a symmetric key is required. To distribute such a symmetric key a secure communication channel is required. Thus we have arrived at a chicken-egg problem or a catch-22. A way out of this problem is given by key exchange protocols that do not require a secure communication channel. However, key exchange protocols do require a Classical Authenticated Channel (CAC) between Alice and Bob. Alternatively, it can also be required that Alice and Bob share a small initial secret, which they then use to authenticate themselves to establish the CAC. A good key exchange protocol then allows them to grow that initial secret such that they can establish a secure communication channel with the OTP cipher. A secure key should also be generated even in the case if an adversary listens to all of the information that Alice and Bob exchange during the protocol. The overall security is only as good as its weakest part, so we would like a key distribution scheme which does not reduce the security of the OTP. This means we would like a protocol that is not just practically secure, but instead one that is unconditionally secure (without any requirements on the capabilities of the adversary). We will see in the following chapter that the Diffie-Hellman Key Exchange (DHKE) protocol provides a practically secure method. In chapter 2.3 we will learn that QKD provides a secure method. Note that both protocols require a CAC or short initial secret.

2.1.5.1 Diffie-Hellman key exchange

W. Diffie and M. Hellman introduced a protocol for key exchange, which is known as DHKE, in their paper [5] from 1976. The security of this protocol is based on the computational hardness of the discrete logarithm problem (see chapter 2.1.2 for details). How this protocol works is demonstrated in figure 2.4.

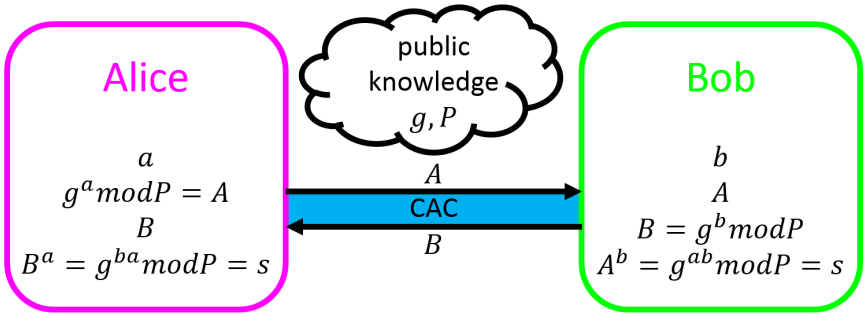


Figure 2.4: Diffie-Hellman key exchange: Alice and Bob both publicly announce their choice for the generator g, which is some integer, and for the prime modulus P, which is a prime number strictly larger than the generator g. Alice has a secret integer a and Bob has a secret integer b, both secrets are ideally a large random number and strictly smaller than P. Alice calculates A and Bob calculates B, which both then transmit to the other via a CAC. Then both parties take the received number and raise it to the power of their secret number, which leads to the secret s for both of them. Alice and Bob have now successfully generated a secret (key) s, which only they know.

The security of the protocol is based on the assumption that an attacker can not easily calculate the secrets of Alice and Bob, denoted by a, b, given A, B.
Today the DHKE protocol is often used to distribute keys for the symmetric AES cipher. This key exchange protocol is considered to be safe as long as the prime modulus P is at least 2048 bits long, as it has been broken for prime modulus lengths up to 1024 bits [47].

2.2 Quantum Mechanics

In the previous chapter we saw that the key exchange problem can be solved with the Diffie-Hellman key exchange protocol, assuming that the discrete logarithm problem is in practice unsolvable for an adversary (see section 2.1.2). QKD offers a way to distribute keys in a way were no such assumption is necessary. In fact, to prove security of QKD it is not necessary to use the mathematical and/or the computational assumption. Before QKD can be explained it is however necessary to briefly review the foundations of Quantum Mechanics on which QKD rests.

2.2.1 States and Observables

The most fundamental building blocks of any physical theory are a state description and a dynamical law for that state. The state describes all relevant degrees of freedom of the system under consideration. The dynamical law describes how the state changes with time [39] [40] [41].

In quantum mechanics we use vectors to model states. If the degree of freedom under consideration is continuous an infinite dimensional vector is used. If the degree of freedom under consideration is discrete a finite dimensional vector suffices. In QKD we are concerned with quantum bits, short qubits, to describe states. These qubits are two-dimensional vectors which live in an complex vector space that is equipped with an inner product. This vector space is called Hilbert space and is denoted by $\mathbb{C}^2$. In general any qubit is a state that can be written as $|\Psi\rangle = a|0\rangle + b|1\rangle$ where $|0\rangle, |1\rangle \in \mathbb{C}^2$ form an orthonormal basis, $a, b \in \mathbb{C}$ and $|a|^2 + |b|^2 = 1$.

The dynamical law in quantum mechanics is given by the Schrödinger equation. In QKD the qubits usually do not change (much) with time, which is why in this special case no dynamical law is needed. For the complete theory of quantum mechanics another object needs to be discussed - the observable.

An observable is a book keeping device which helps you to keep track of the measurement results and the post measurement state. In quantum mechanics a measurement does in general change the state which is being measured[17]. This book keeping is done in terms of eigenvalues which represent measurement results and eigenvectors which represent post measurement states. Thus the complete

[17] This effect will be further explored and quantified in section 2.2.2

book keeping is encapsulated in the eigenequation:

$$\hat{O}\,|e_n\rangle = e_n\,|e_n\rangle\,.$$

Here the operator $\hat{O}$ has the eigenstates $|e_n\rangle \in \mathbb{C}^2$ with their corresponding eigenvalues $e_n \in \mathbb{R}$ for $n \in \{1,2\}$. This operator $\hat{O}$ represents some corresponding observable. To ensure that all eigenvalues e_n (or measurement results) are real-valued the operator has to be hermitian. We consider only operators that can act on qubits, which is why this operator has only two eigenvalues. Such operators can be represented by $2x2$ matrices. The set of all such operators that act on vectors or states of $\mathbb{C}^2$ is denoted as $\mathfrak{L}(\mathbb{C}^2)$.

Before any kind of calculation can be performed, a representation for the vectors has to be chosen. Typically one chooses the Cartesian basis to represent the standard basis $|0\rangle, |1\rangle$. This means we make the following identification:

$$|0\rangle = \begin{pmatrix}1\\0\end{pmatrix}$$
$$|1\rangle = \begin{pmatrix}0\\1\end{pmatrix}.$$

This fixes the representation of all other vectors and matrices.

For the calculation of the inner product $\langle\Psi\,|\,\Psi\rangle$ of two vectors a dual vector $\langle\Psi|$ needs to be constructed. Given a vector $|\Psi\rangle = a\,|0\rangle + b\,|1\rangle$ the dual vector is given by $\langle\Psi| = a^*\,\langle 0| + b^*\,\langle 1|$ where the $*$ denotes the complex conjugate and $\{\langle 0|, \langle 1|\}$ denotes two basis vectors of the dual space. A dual vector can be represented as a row vector, while a *normal* vector is represented as a column vector, so

$$|\Psi\rangle = \begin{pmatrix}a\\b\end{pmatrix} \leftrightarrow \langle\Psi| = \begin{pmatrix}a^* & b^*\end{pmatrix} \text{ and } \langle\Psi\,|\,\Psi\rangle = |a|^2 + |b|^2.$$

It is also possible to describe a state with a special operator, the density operator. This allows you to describe pure quantum mechanical states as well as probabilistic mixtures of pure states. The general definition for the density operator $\hat{\rho}$ with a proper choice of the basis $\{|n\rangle\}$ is

$$\hat{\rho} = \sum_n^N p_n\,|n\rangle\,\langle n|\,.$$

Here p_n denotes the probability that the system is in the state $|n\rangle$. If $N = 1$ the system is in a pure state, else the system is a probabilistic mixture of pure states. Every density operator is positive semidefinite, hermitian and its trace equals one.

Using the Bloch representation[18] it is possible to represent a qubit $|\Psi\rangle \in \mathbb{C}^2$ as a Bloch(vector) $\vec{v}_B \in \mathbb{R}^3$. This Bloch vector can be constructed given $|\Psi\rangle$ or $\hat{\rho}$. In the later case[19] it is necessary to represent $\hat{\rho}$ in terms of the identity and the Pauli matrices

$$\hat{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \hat{\sigma}_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \hat{\sigma}_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \hat{\sigma}_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

These representations are fixed due to our choice of representation for the standard basis $|0\rangle, |1\rangle$ which we want to consist of eigenstates of $\hat{\sigma}_z$.
Using these we can express any density operator as

$$\hat{\rho} = \frac{1}{2}\left(\hat{1} + v_x\hat{\sigma}_x + v_y\hat{\sigma}_y + v_z\hat{\sigma}_z\right) \text{ with } \vec{v}_B = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}.$$

A pure state satisfies $|v_x|^2 + |v_y|^2 + |v_z|^2 = 1$, while a mixture satisfies to $0 \leq |v_x|^2 + |v_y|^2 + |v_z|^2 < 1$. This means that pure states lie on a sphere of radius one and mixtures lie within this sphere.

2.2.2 The Uncertainty Principle

The uncertainty Δ of an observable $\hat{A}$ for a state $|\Psi\rangle$ is defined as

$$\Delta\hat{A}(|\Psi\rangle) = \sqrt{\langle\Psi|\hat{A}^2|\Psi\rangle - \langle\Psi|\hat{A}|\Psi\rangle^2}.$$

Using this definition of the uncertainty and the Cauchy–Schwarz inequality it is possible to derive[20] the uncertainty principle[21] which states:

$$\Delta\hat{A}(|\Psi\rangle) * \Delta\hat{B}(|\Psi\rangle) \geq \frac{1}{2}|\langle\Psi|\hat{A}\hat{B} - \hat{B}\hat{A}|\Psi\rangle|.$$

If two operators commute, meaning $\hat{A}\hat{B} = \hat{B}\hat{A}$, it is possible to find a find a set of common eigenstates spanning the whole space (that sets both uncertainties to zero). Any arbitrary state for two commuting observables can still give rise to non-zero uncertainties. This is due to the fact that the commutator only gives rise to the lower bound for the uncertainties.
Intuitively this principle demonstrates that if two operators do not commute, one

[18] More details regarding the Bloch sphere can be found in the appendix chapter 8.1.
[19] $\hat{\rho}$ can always be constructed given $|\Psi\rangle$ since then the density operator simplifies to $\hat{\rho} = |\Psi\rangle\langle\Psi|$
[20] This derivation is omitted here but it can be found in the standard literature [38].
[21] In order to understand the uncertainty principle it is useful to have an intuitive understanding of the uncertainty of an observable (for a given state) which is provided in the appendix section 8.2.

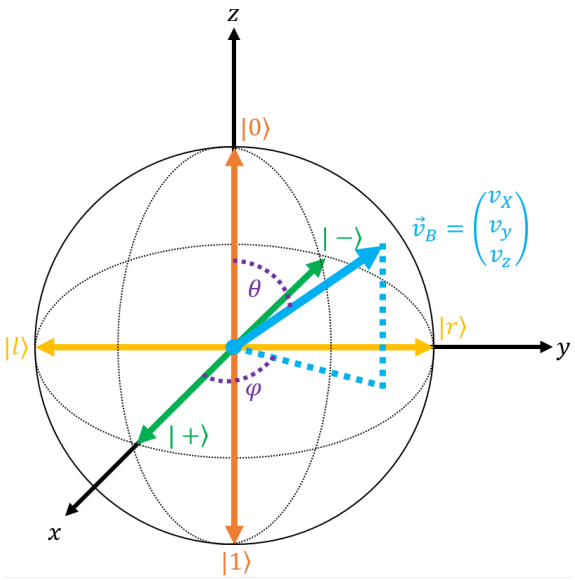


Figure 2.5: The Bloch sphere can be used to represent one qubit as a Bloch vector in three-dimensional space. The orange states $|0\rangle, |1\rangle$ are the eigenstates of $\hat{\sigma}_z$. The green states $|+\rangle, |-\rangle$ are the eigenstates of $\hat{\sigma}_x$. The yellow states $|l\rangle, |r\rangle$ are the eigenstates of $\hat{\sigma}_y$. The blue state is an arbitrary Bloch vector $|B\rangle = cos(\frac{1}{2}\theta)|0\rangle + e^{i\varphi} sin(\frac{1}{2}\theta)|1\rangle$ or equivalently in terms of the density operator $\hat{\rho} = |B\rangle\langle B| = \frac{1}{2}\left(\hat{1} + v_x\hat{\sigma}_x + v_y\hat{\sigma}_y + v_z\hat{\sigma}_z\right)$.

of them (or both) will alter[22] the state $|\Psi\rangle$. Thus it is impossible to measure both observables for the same state $|\Psi\rangle$, if these observables do not commute and if the state is not in an eigenstate of both observables.

2.2.3 The no-cloning-theorem

The no-cloning-theorem states that there can not exist a cloning machine for arbitrary quantum states. To prove this theorem we assume that there exists some cloning machine, whose action we denote by the unitary (and thus linear) operator $\hat{C}$. We expect that this operator $\hat{C}$ will copy or clone any state. To achieve this we will *feed* the copying machine with an input state $|\Psi\rangle$ we want to copy and a blank state $|e\rangle$ which is supposed to be transformed into the input state via $\hat{C}$.
Let us begin by copying the standard basis states:

$$\begin{aligned}\hat{C}\,|0\rangle\,|e\rangle &= |0\rangle\,|0\rangle\\ \hat{C}\,|1\rangle\,|e\rangle &= |1\rangle\,|1\rangle\end{aligned}$$

Now we want to copy an arbitrary input state $|\Psi\rangle = a\,|0\rangle + b\,|1\rangle$. We would expect to get:

$$\begin{aligned}\hat{C}\,|\Psi\rangle\,|e\rangle &= |\Psi\rangle\,|\Psi\rangle\\ &= \big(a\,|0\rangle + b\,|1\rangle\big)\big(a\,|0\rangle + b\,|1\rangle\big)\\ &= a^2\,|0\rangle\,|0\rangle + ab\,|0\rangle\,|1\rangle + ba\,|1\rangle\,|0\rangle + b^2\,|1\rangle\,|1\rangle\end{aligned}$$

but if we use the linearity of $\hat{C}$ we find instead:

$$\begin{aligned}\hat{C}\,|\Psi\rangle\,|e\rangle &= \hat{C}\big(a\,|0\rangle + b\,|1\rangle\big)\,|e\rangle\\ &= \hat{C}\big(a\,|0\rangle\,|e\rangle + b\,|1\rangle\,|e\rangle\big)\\ &= a\hat{C}\,|0\rangle\,|e\rangle + b\hat{C}\,|1\rangle\,|e\rangle\\ &= a\,|0\rangle\,|0\rangle + b\,|1\rangle\,|1\rangle\\ &\neq |\Psi\rangle\,|\Psi\rangle\,.\end{aligned}$$

Thus we have arrived at an contradiction. We conclude that no machine can exist, which can create copies of arbitrary states. This contradiction arises from the necessary linearity of $\hat{C}$, which comes from the fact that the dynamics of quantum mechanics is governed by the Schrödinger equation, which is a differential equation that is linear in the state $|\Psi\rangle$. This linearity forbids the existence of a cloning machine for arbitrary quantum states.

[22] In this context *alter* means that the state is not just rescaled by some number, while maintaining the same orientation in the Hilbert space.

2.3 Quantum Key Distribution

Compared to classical key distribution schemes QKD offers several advantages. QKD provides in principle unconditional security, which in particular means that no technological assumptions about the adversaries capabilities have to be made. Additionally it is possible to calculate how much information about the key an adversary might have gained during the key distribution, this is completely impossible in classical schemes.
QKD offers a variety of protocols to address the key distribution problem (explained in more detail in chapter 2.1.5). These protocols can be grouped in various ways, for example according to the origin for their security. For some protocols their security comes from the fact that they use entangled states. The security of such protocols is given from the property of monogamy of entangled states and from a test of the entanglement via a violation of Bell's inequality[42]. The first entanglement based protocol was introduced in 1991 by Artur E. Ekert [10]. Other protocols use the uncertainty principle (see chapter 2.2.2) to their advantage, in so called prepare-and-measure protocols. We want to focus on the latter and look at one particular protocol in more detail, the BB84 protocol.
The BB84 protocol was developed in 1984 by Charles Bennett and Gilles Brassard in their famous paper [9] and it is one of the most widely used QKD protocols.

2.3.1 Requirements

The following list describes the starting situation and highlights important requirements, which are sometimes only stated implicitly. It is necessary to stress which conditions go into the security that QKD provides, so that it is clear what benefits can be gained at what costs.

- **Eve has no access to the laboratories of Alice and Bob.**
  Alice and Bob protect their laboratories against probing and manipulations from attackers.Their laboratories also do not leak any unwanted information to the outside world. This may sound easy but is very hard to achieve, since Alice and Bob can not forbid all interactions with the outside world, because they still want to interact with each other. How well this assumption is approximated depends heavily on the specific implementation. Since this is no trivial matter it should be carefully checked. The importance of this verification can not be overestimated, since a violation of this assumption can compromise the whole security of the key distribution scheme.

- **Alice and Bob share a classical authenticated channel.**
  Alice and Bob share a classical authenticated channel, which means that

an adversary can read their messages, but can not alter or forge them. This specifically does not allow man-in-the-middle attacks and prevents Eve from posing as Alice and/or Bob to one or both parties. This assumption can be exchanged with the assumption that Alice and Bob share a small amount of secret prior to their attempt of QKD, which they then use to authenticate themselves. Due to this QKD can also be seen as a secret growing protocol.

- **Alice and Bob share a quantum channel.**
  Alice can send qubits to Bob via the quantum channel. Ideally the channel leaves the original qubit state intact. If this is not possible than a characterization of the qubit rotation should be possible, so that Bob may counteract these effects. Strictly speaking, the channel is not *quantum*, but the information carriers are [23]. Eve can perform any operation allowed by the laws of physics on the quantum channel and on the qubits transmitted over it.

- **Alice and Bob each have a random number generator.**
  A random number generator is needed to add some unpredictability to the protocol, otherwise Eve might be able to guess some deliberate choices by Alice and Bob.

- **Alice and Bob agreed to use the BB84 protocol and agreed on some conventions.**
  When Alice and Bob agree on using the BB84 protocol they also need to agree to certain conventions that they will use. Within this protocol they agree on the use of two bases, the eigenstates of $\hat{\sigma}_x$, which is called the Hadamard basis and the eigenstates of $\hat{\sigma}_z$, which is called the standard basis. These eigenstates are used encode the key bits. Alice and Bob need to map the measured qubits to the same bits. In particular this means that they agree on some qubit-bit correspondence, like

$$\{|0\rangle, |+\rangle\} \Leftrightarrow 0,$$
$$\{|1\rangle, |-\rangle\} \Leftrightarrow 1.$$

  They also need to agree on a bit value that denotes their basis choice. Their N basis choices will be recored in a basis bit string $B = B_1 B_2 ... B_N \in \{0,1\}^N$ for Alice and $\widetilde{B} = \widetilde{B}_1 \widetilde{B}_2 ... \widetilde{B}_N \in \{0,1\}^N$ for Bob. Alice and Bob agree that,

$$B_j = 0 \Leftrightarrow \{|0\rangle, |1\rangle\} \text{ Standard basis},$$
$$B_j = 1 \Leftrightarrow \{|+\rangle, |-\rangle\} \text{ Hadamard basis}.$$

[23] Contrary to the classical channel, the information is coded without redundancy in the quantum channel or its information carriers.

Additionally they also agree on a scheme for error correction and for privacy amplification. The scheme they chose will alter the information they need to exchange on the CAC to perform each task.

- **Alice and Bob trust their devices.**
  In the BB84 protocol it is necessary that Alice and Bob trust their devices. Due to this they need to check those devices before they use them for the protocol to ensure that they function properly and to check that these devices do not leak any unwanted information. This means in particular that Alice has to verify her state preparation and Bob his state measurements. Both parties also need to check their random number generators they use for the basis selection. Alice also has to check her random number generator for the bit selection.

- **Eve can do anything allowed by the laws of physics.**
  Eve can do anything that is not forbidden by the laws of physics. This means in particular that Eve can not influence events instantaneously (or faster than the speed of light) and that Eve can not posses a quantum copying machine. It is assumed that these statements will always be true, even if Eve would have knowledge of a higher theory. In its respective realm it is assumed that Special Relativity and Quantum Mechanics will always be true. Similarly to classical mechanics, which is still a correct theory for macroscopic objects that move with a velocity much slower than the speed of light.
  This is however the only limitation imposed on Eve. This means that Eve can do practically anything that is physically possible. Eve may have practically unlimited computational power, access to algorithms that solve hard problems fast (if such algorithms can exist), like a quantum computer on which she may run Shor's algorithm. Eve may also posses a quantum repeater and a quantum memory. Contrary to classical cryptography, where the mathematical and computational assumptions are used, these assumptions are not used in typical QKD scenarios. Basically, there are no conditions on the capabilities of Eve, which is why it is often said that, QKD offers unconditional security. Note that this does not mean that there no conditions or assumptions are made in general, but there are no conditions with respect to the technological capabilities of Eve.

These assumptions are often only implicitly stated or are taken for granted. We see that the unconditional security of QKD is not solely based on the laws of nature.

2.3.2 The BB84 protocol

This section describes the BB84 protocol including the steps for error correction and privacy amplification.

1) Alice chooses a random N-bit string $x_A = x_1 x_2 ... x_N \in \{0,1\}^N$ and a random N-bit basis string $B = B_1 B_2 ... B_N \in \{0,1\}^N$.

2) Bob chooses a random N-bit basis string $\widetilde{B} = \widetilde{B}_1 \widetilde{B}_2 ... \widetilde{B}_N \in \{0,1\}^N$.

3) Alice sends to Bob bits $x_j \in \{0,1\}$ encoded in the basis $B_j \in \{0,1\}$ via the qubit $\left|x_j\right\rangle_{B_j} \in \mathbb{C}^2$ over the quantum channel.

4) Bob measures the qubits in the basis $\widetilde{B}_j \in \{0,1\}$ and obtains the measurement result $\widetilde{x}_j \in \{0,1\}$.

5) Bob announces receipt of the qubits.

6) Alice and Bob publicly exchange bases strings $B, \widetilde{B}$. They discard all cases where $B_j \neq \widetilde{B}_j$.

7) Alice randomly chooses n of the remaining cases to test. She tells Bob which rounds are tested and they exchange the bits $x_j, \widetilde{x}_j$ of those cases. Alice and Bob compute the error ratio δ. If δ exceeds a threshold they abort the protocol.

8) Alice and Bob exchange error-correcting information. This is called information reconciliation.

9) For the so-called privacy amplification Alice chooses an extractor seed y and sends it to Bob.
Alice computes $k_A = Ext(x_{A,remain}, y)$.
Bob computes $k_B = Ext(x_{B,remain}, y)$.

A run of this protocol may look like the one in figure 2.7.

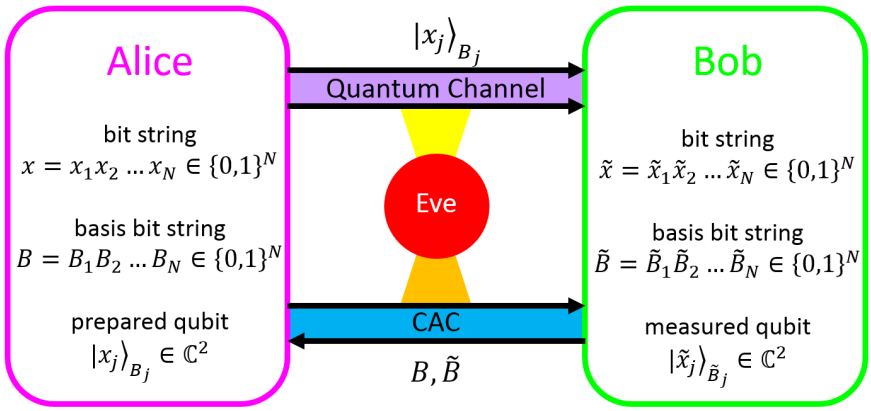


Figure 2.6: The laboratories of Alice and Bob are connected via a one-way quantum channel and a two-way classical authenticated channel. Eve can perform any kind of physically possible action on the quantum channel, while she can only listen to the messages exchanged on the classical channel. The quantum channel is used to exchange qubits. The CAC is used to exchange the basis bit strings, error correction information and extractor seed y for privacy amplification.

j	1	2	3	4	5	6	7	8	9	...	N
x_j	1	0	1	1	1	0	0	1	0	...	0
B_j	0	1	0	0	0	1	1	0	1	...	1
$\tilde{B}_j$	1	0	0	1	0	1	0	0	0	...	1
$\tilde{x}_j$	0	0	1	1	1	0	0	1	0	...	0
$B_j = \tilde{B}_j$	×	×	✓	×	✓	✓	×	✓	×		✓
X_j	-	-	1	-	1	0	-	1	-	...	0

Figure 2.7: Alice encodes the bit x_j in the basis B_j and encodes these two bits in the qubit $|x_j\rangle_{B_j}$ which she sends via the quantum channel to Bob. Bob picks a measurement basis $\widetilde{B}_j$ and obtains the measurement result $\widetilde{x}_j$. They perform a basis reconciliation which turns the raw key x into the sifted key X.

2.3.3 Why it works

It is important that Alice chooses her bits x_j and her basis at random, since any information about her choice could allow an attacker to guess some information about the resulting key. If an attacker would know which basis Alice chooses to encode a bit the whole protocol would become completely insecure.
Bob also needs to pick his measurement basis $\widetilde{B}_j$ at random, otherwise an intercept-and-resend attack becomes much harder to detect and the whole security would be again undermined.

Alice encodes the bit in a qubit, which corresponds to some quantum mechanical two-level system. The information (the bit) is just encoded once, in this precise system, or more specifically, in a degree of freedom of that system. A classical bit, transmitted via an electrical signal is encoded many times via a comparatively huge macroscopic signal. This enables an eavesdropper to copy a classical bit without detection. The use of a single qubit for the information coding is what makes QKD so secure. Tampering with the qubit will be detected due to the uncertainty principle. If Eve is unaware of the state Alice prepared she can not perform with certainty a non-destructive measurement. Her tampering or changing of the state can be detected by Alice and Bob.

Alice and Bob need to make sure that they identify the qubits in the same way. If a qubit gets lost along the way, for example due to scattering, it should not appear in their lists, otherwise Alice's $j-th$ qubit might be Bob's $l-th$ qubit and so on. This process is called sifting. The prepared qubit from Alice will only be correlated with the measured qubit from Bob if they have chosen the same basis. If their basis choice differ the result will be perfectly random. This can be easily seen by calculating the probabilities, say for the case that Alice prepared a bit 0 in the standard basis ($|0\rangle_0 = |0\rangle$), then there are four possibilities for what Bob will measure and thus:

$$\begin{aligned}
\mathbb{P}(|0\rangle) &= |\langle 0|0\rangle|^2 = 1 \\
\mathbb{P}(|1\rangle) &= |\langle 0|1\rangle|^2 = 0 \\
\mathbb{P}(|+\rangle) &= |\langle 0|+\rangle|^2 = 0.5 \\
\mathbb{P}(|-\rangle) &= |\langle 0|-\rangle|^2 = 0.5
\end{aligned}$$

Thus we see that if Bob picks the same basis as Alice his result will coincide with certainty with the bit value that Alice prepared. If he uses the other basis there is a 50% chance that he will get the right or wrong bit value, so he has no way of knowing which bit value Alice prepared. Thus all cases where their bases differ

have to be discarded.

Alice and Bob can calculate the error δ, which allows them to calculate an upper bound for the information an adversary might have collected. If the error is too high[24] the protocol has to be aborted and restarted. If the error is small enough Alice and Bob can improve the security of their key via classical schemes [11] [12] [45].
Since any real channel can introduce errors (like bit-flip errors) Alice and Bob need to perform error correction. An adversary could also introduce errors, which is why δ is used to estimate the information an adversary might have collected.
The quantum bit error rate (QBER) is defined as the ratio of the number of wrong bits and the number of total bits, which where exchanged: $QBER = \frac{N_{wrong}}{N_{total}}$.
If an eavesdropper has gained any information about the key, which can be bound by a function of δ, Alice and Bob can reduce this information by performing privacy amplification. Here Alice and Bob use a randomness-extractor, which is a two-universal hash-function, that needs a seed y and their remaining bits $x_{A,remain}$. After that Eve will have no information about k_A, k_B.

Note that the resulting key is neither decided by Alice nor by Bob. The key is a result of their random choices. Alice could not enforce the generation of some specific key and neither could Bob.

[24] Which is the case if Eve has more information about the key than Bob.

2.3.4 Photons as information carriers

In general photons are used as the physical information carrier, more specifically, the polarization degree of freedom of photons[25]. We need to identify four suitable polarization states for the earlier introduced BB84 states $|0\rangle$, $|1\rangle$, $|+\rangle$, $|-\rangle$, which can be related to $|H\rangle$, $|V\rangle$, $|P\rangle$, $|M\rangle$ via the identification described in table 2.2.

Basis	B_j	x_j	BB84 state	light states	Polarization
$\hat{\sigma}_z$	0	0	$\|0\rangle_0 = \|0\rangle$	$\|H\rangle$	Horizontal
$\hat{\sigma}_z$	0	1	$\|1\rangle_0 = \|1\rangle$	$\|V\rangle$	Vertical
$\hat{\sigma}_x$	1	0	$\|0\rangle_1 = \|+\rangle$	$\|P\rangle = \frac{1}{\sqrt{2}}(\|H\rangle + \|V\rangle)$	Diagonal
$\hat{\sigma}_x$	1	1	$\|1\rangle_1 = \|-\rangle$	$\|M\rangle = \frac{1}{\sqrt{2}}(\|H\rangle - \|V\rangle)$	Anti-Diagonal
$\hat{\sigma}_y$	-	-	-	$\|R\rangle = \frac{1}{\sqrt{2}}(\|H\rangle + i\|V\rangle)$	Right-Circular
$\hat{\sigma}_y$	-	-	-	$\|L\rangle = \frac{1}{\sqrt{2}}(\|H\rangle - i\|V\rangle)$	Left-Circular

Table 2.2: The standard basis will be identified with the horizontal and vertical linear polarization states. The Hadamard basis will be identified as the diagonal and anti-diagonal linear polarization. The right-circular and left-circular polarization states are not needed for the BB84 protocol, since fours state suffice.

It is possible to transform the standard basis into the Hadamard basis via the Hadamard transformation

$$\hat{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

This transformation can be used to switch the basis.

$$\begin{aligned} \hat{H}\,|H\rangle &= |+\rangle \\ \hat{H}\,|V\rangle &= |-\rangle \\ \hat{H}\,|+\rangle &= |H\rangle \\ \hat{H}\,|-\rangle &= |V\rangle \end{aligned}$$

Note that $\hat{H}$ is unitary, meaning $\hat{H}^{-1} = \hat{H}^\dagger$ and hermitian, meaning $\hat{H}^\dagger = \hat{H}$. Thus in total we can conclude that $\hat{H}^{-1} = \hat{H}$ from which follows that $\hat{H}\hat{H} = \hat{1}$.

2.3.4.1 Wire grid polarizers

A wire grid polarizer (see figure 2.8) can be used to polarize or filter light. If the grid is for example horizontally oriented, only vertically polarized light will pass

[25] Other protocols may use different degrees of freedom.

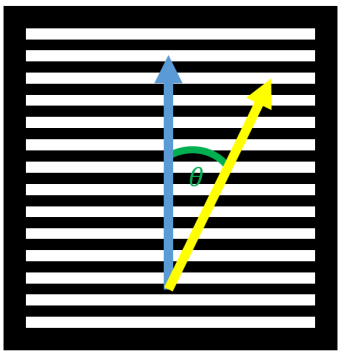


Figure 2.8: A wire grid polarizer. The wire grid is oriented along the horizontal axes, which means that only light that is linearly polarized along the vertical axes (blue arrow) passes through. For linearly polarized light which makes an angle θ with the axis of the polarizer (yellow,) the probability that the photon will be transmitted is given by $cos(\theta)^2$.

through this polarizer, whereas horizontally polarized light will be (almost) completely blocked. In general we can write a polarization state $|a\rangle$ as a superposition of $|H\rangle, |V\rangle$ via some appropriate coefficients

$$|a\rangle = \alpha |H\rangle + \beta |V\rangle ,$$

where again $|\alpha|^2 + |\beta|^2 = 1$, in general $\alpha, \beta \in \mathbb{C}$ may also depend on time and $|\alpha|^2$ and $|\beta|^2$ can be interpreted as the probabilities that a photon will pass through the polarizer ($\mathbb{P}_{pass} = |\alpha|^2$) or be blocked ($\mathbb{P}_{block} = |\beta|^2$). In this case the action of the polarizer on the polarization state $|a\rangle$ can be described via a projector $\hat{P} = |V\rangle \langle V|$. We use a wire grid polarizer to prepare the needed polarization states, since every photon that passes through it will be linearly polarized along the polarization axes of the wire grid polarizer.

2.3.4.2 Weak coherent pulses

The BB84 protocol assumes that Alice has access to a single photon source. For practical QKD with prepare-and-measure schemes, like BB84, there is no practical single photon source. What makes existing single photon sources unpractical is on the one hand their modulation speed and on the other hand their size, because they require big cryostats [23]. Strongly attenuated laser pulses are no true single photon sources but can instead be modeled as a source for coherent states. A coherent state is a ground state of a quantum harmonic oscillator, denoted by

$|0\rangle$, which gets translated by $\widetilde{x}$ in position with a translation operator $\hat{T}_{\widetilde{x}}$

$$\hat{T}_{\widetilde{x}}|0\rangle = |\widetilde{x}\rangle$$
$$\hat{T}_{\widetilde{x}} = e^{-\frac{i\hat{p}\widetilde{x}}{\hbar}}.$$

It is however more convenient to relabel such a state as $|\mu\rangle$ instead of $|\widetilde{x}\rangle$ due to its expectation value of the number operator $\hat{N}$. If we then express $|\widetilde{x}\rangle$ in terms of the energy eigenstates we rewrite this state as:

$$|\mu\rangle = e^{-\frac{\mu}{2}} \sum_{n=0}^{\infty} \frac{\mu^{\frac{n}{2}}}{\sqrt{n!}} |n\rangle .$$

Here $|n\rangle$ denotes the *photon number eigenstates*[26]. This means that for the state $|\psi\rangle = |n\rangle$ the probability to measure $m = n$ photons is 1 and the probability to measure $m \neq n$ photons is zero. For the expectation value of the photon number we find

$$\langle \mu | \hat{N} | \mu \rangle = \mu.$$

Which is why it is convenient to label coherent states with μ.

The probability distribution for a coherent state $|\mu\rangle$ is given by the Poissonian statistics:

$$\mathbb{P}(n) = e^{-\mu} \frac{\mu^n}{n!}.$$

The Poisson distribution of the photon number is a problem with regard to the security of QKD. The protocol assumes that a single photon source is used, but now there is a chance that two or more identical photons are emitted.
For $\mu = 0.1$, the probability for a multiphoton pulse is about 5%, which can be exploited by performing a so called photon-number splitting (PNS) attack [14]. An attacker who performs a PNS attack blocks all single-photon pulses. The multiphoton pulses are transmitted to Bob, but the attacker removes one or more photons from the pulses. He then stores them safely[27] and waits until Alice and Bob (publicly) perform their basis reconciliation. Then the attacker mimics the basis choices of Bob for the measurements on the stored photons. This allows the attacker to fully construct the secret key without leaving any trace.
The most recent method to address this issue the decoy protocol has been developed by X.-B. Wang [16]. Here two different laser intensities are used and monitored, which allow Alice and Bob to detect any tampering by Eve.

[26] Definition of the (photon) number operator $\hat{N}$: $\hat{N}|n\rangle = n|n\rangle$.

[27] In a way that protects them from decoherence.

2.3.5 The quantum channel

The quantum channel can be realized either by optical fibers or free space, which will be discussed in the following.

2.3.5.1 Fiber

Using a fiber can be attractive, since the preexisting fiber network (within buildings and cities) could be used, which would allow the construction of *plug-and-play* QKD components that can just be added to the existing infrastructure [25], [26], [27]. Every fiber (with length l) comes with a fixed attenuation α, depending on the used material, which determines the transmission:

$$t = 10^{-\alpha l/10}.$$

For fibers in the existing network, we find a minimum in the attenuation at 1330 *nm* ($\alpha \approx 0.34 dB/km$) and 1550 *nm* ($\alpha \approx 0.2 dB/km$). Since the fiber fixes the wavelength and the loss we can not change anything to decrease the loss, which leads to a practical length limitation of about 100 *km* for a such a fiber based link, unless a quantum repeater could be used [28].

Due to birefringence fibers are not practical for polarization based QKD, but when other degrees of freedom are used to encode the information, like time-phase coding, fibers can be utilized [11].

2.3.5.2 Free space

While for a fiber only two wavelength ranges can be effectively used free space links offer a much broader range of wavelengths that can be used due to the several transmission windows of the atmosphere (see figure 2.9).
Additionally the atmosphere is only weakly dispersive, at the typical transmission window wavelengths. Together with the fact that the atmosphere is also practically non-birefringent it leads to a preservation of the polarization state during the transmission through the atmosphere.
However, there are also disadvantages. Since the photons are sent through the air the quality of this channel depends drastically on the atmospheric conditions, where turbulences and clouds can drastically change the quality of the channel in a more or less unpredictable manner [22]. To overcome this, the current conditions can be probed with a reference beam.
Another issue that affects the performance of the QKD system is background radiation (from terrestrial and extraterrestrial sources). By employing spectral, spatial

[28] The current record for fiber-based QKD is 404 km [24].

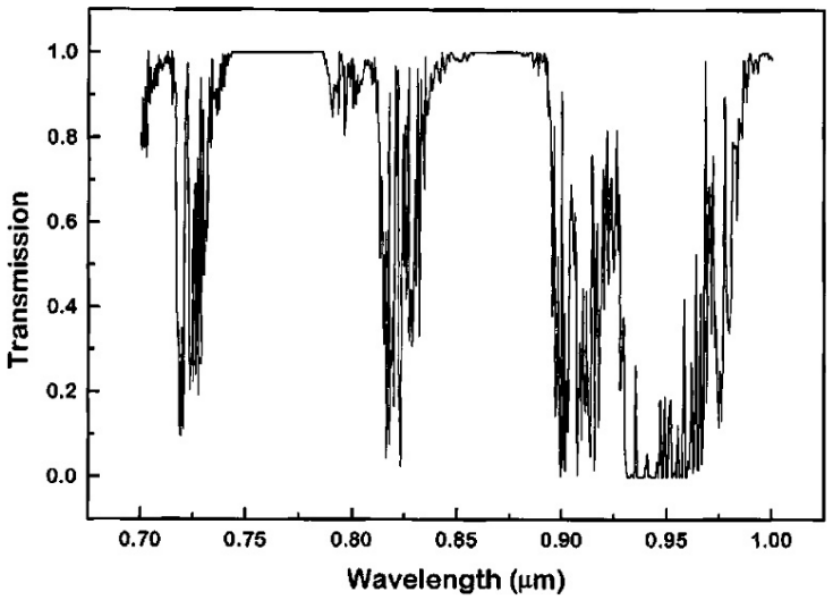


Figure 2.9: The Transmission of the atmosphere for optical and near infra-red light. This curve is the result of a simulation using the LOWTRAN code for earth-to-space transmission at the elevation and location of Los Alamos, USA. Taken from [11].

and temporal filtering, it is possible to do QKD even in daylight conditions, which has been demonstrated in the past [21].
Besides the atmospheric losses, there also exists geometrical losses due to beam divergence: $tan(\theta) = \frac{\lambda}{\pi w_0}$, where θ denotes the divergence angle, λ the wavelength and w_0 the waist. This effect can also be reduced with appropriate optics for the sender and receiver. However, especially in a satellite this becomes challenging, as one may not have the freedom for arbitrary optical corrections due to size, weight and budget restrictions.
Finally it should be noted that most losses in free space links are not due to the attenuation, which is much lower compared to the attenuation in fibers, but due to scattering. This is why QKD with satellites is so attractive, since the scattering decreases as the density of the atmosphere becomes smaller and smaller. This makes free space links possible for distance that could never be covered with a conventional fiber.

Chapter 3

The hand-held QKD experiment

In this chapter we explore the hand-held QKD experiment designed by several scientists of the group of Prof. Weinfurter, most notably by Dr. Gwenaelle Mélen [31] and Tobias Vogl [35]. The inner workings of the sender unit *Alice* will be described. The crucial components of electronics and optics will be explained such that it will be clear which role they serve for the complete hand-held module and its operation. The final section deals with the receiver *Bob*. The receiver not only consists of a BB84 state analysis unit but also incorporates elements to correct for the rotation and the movements of the users hand.

3.1 Alice

The Alice hand-held QKD module is designed for short range free space key distribution using the polarization degree of freedom of weak coherent light pulses. This module mainly consists of two distinct parts: the Field-programmable gate array (FPGA) driven electronics module, which generates fast electrical pulses and the optics module, which creates the corresponding optical pulses.

3.1.1 The Alice electronics module

Figure 3.1 depicts a schematic of the electronics module. The module can be controlled by a computer via a USB 2.0 connection. On the computer a program called *alice-control* can be used via the bash to pass pulse parameters onto the FPGA. The FPGA then distributes the corresponding signals to the delay chip and the VCSEL driver. The logic gate has a fixed configuration and is not altered for different sets of pulse parameters. After passing through these elements the signal generated by the VCSEL driver is transmitted via a connector to the optics board, where the VCSELs receive the signal. In the following section the action

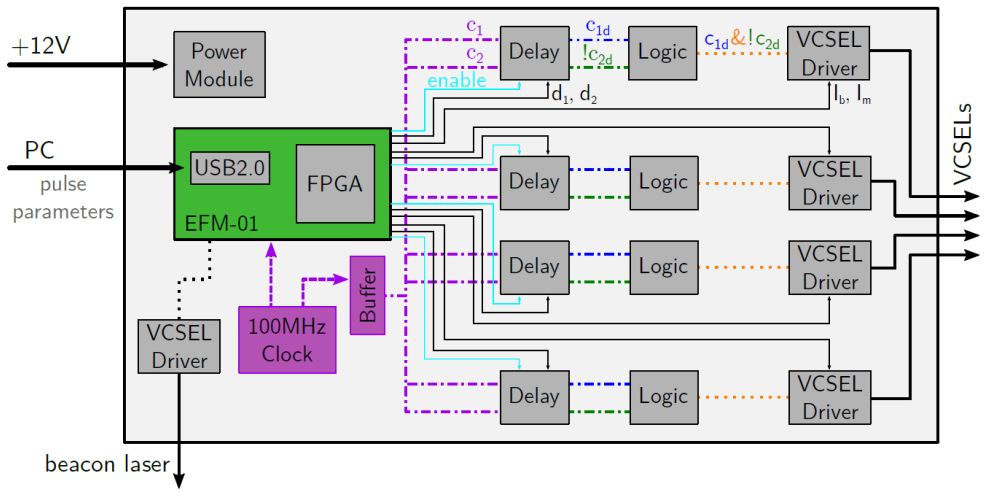


Figure 3.1: A schematic picture of the Alice electronics module. They key components are displayed together with some important connections. The module can be controlled via a computer, which connects via USB to the FPGA *Spartan 3E*. The FPGA sits on the development board *EFM-01*. The FPGA controls the actions of the delay chips (*Micrel SY89297U*) and of the VCSEL driver (*ONET4291VA*). The driver signal is then transmitted onto the optics board via a connector. The logic gates (*Micrel SY55851*) programming is fixed to act as an AND gate. A 100 MHz clock is distributed via a buffer to the FPGA and the delay lines. In particular we see two clock signals, denoted c_1, c_2 being distributed to a Delay chip. Both signals get delayed by d_1, d_2 respective, denoted by the additional d in the subscript. The second signal gets additionally inverted. The AND gate then combines both signals. The VCSEL Driver receives this signal together with the values for the bias current I_{bias} and the modulation current I_{mod}. The signal from the VCSEL Driver is then forwarded to the Alice optics module via some connector. Adapted from [33].

of each component will be explained in more detail, but still simplified, such that an operational understanding can be achieved.

The delay chips (*Micrel SY89297U*) receive two identical clock signals from the buffer, which just copies and distributes the signal from the 100 MHz clock. The delay chip receives the delay parameters $da, db \in [0, 1023]$ from the FPGA and delays one or both clock signals by an amount $t_a = 5ps * da$ and $t_b = 5ps * db$. Additionally the output wiring is such that one clock signal gets inverted. This is more convenient for the next component. The two (delayed) clock signals are then passed on to the logic gate.

The logic gate (*Micrel SY55851*) is fixed to work as an *AND* gate. Here a short pulse is created via an *AND* gate of two (delayed) clock signals, where one clock signal is inverted[1]. This allows the logic gate to effectively create short pulses with a pulse width of about $\Delta t \approx |da - db| * 5ps$. The combination of the delay chips and the *AND* gate enables this module to create very short pulses, which can additionally be shifted in time. This short electrical pulse is then sent to the VCSEL driver.

The VCSEL driver (*ONET4291VA*) receives control signals from the FPGA which define the bias current I_{bias} and the modulation current I_{mod} in our configuration (open loop configuration) in the following way:

$$I_{bias} = 100\,\mu A + 47\,\mu A * b$$
$$I_{mod} = 100\,\mu A + 68\,\mu A * m$$

with the bias parameter $b \in [0, 255]$ and the modulation parameter $m \in [0, 255]$. The modulation current can be tuned, by changing the MODR setting (called MODR in the Driver manual) from $68\mu A$ to $51\mu A$, but this needs to be done via a reprogramming of the FPGA (this can not be done with alice-control).
When the VCSEL driver receives the signal from the logic gate its modulation is triggered (with the saved parameters for I_{bias}, I_{mod}).

[1] Note that without an inversion of one of the two clock signals the minimum pulse length would be about 5 *ns* (figure 3.2).

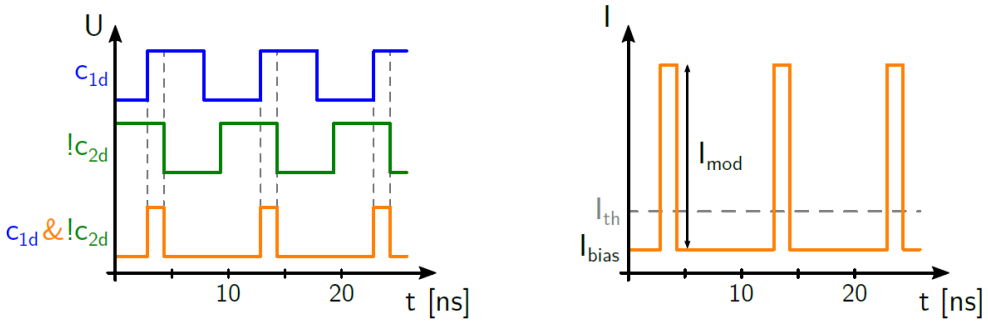


Figure 3.2: The left picture illustrates the effect of the delay chips and of the logic gate. The right pictures depicts how the net current (orange) results from the bias current I_{bias} and the modulation current I_{mod}. I_{th} marks the threshold current for the VCSELs under which they do not emit any light. Note that the bias current is below the threshold current. The VCSELs are switched on only for a short amount of time, determined by the width of the modulation peak. Taken from [33].

The parameters are entered via the before mentioned *alice-control* program, which has the following control parameters and values:

parameter	description	values
-c	channel	$0, 1, 2, 3$ or -1 for all
-b	bias	[0,255]
-m	modulation	[0,255]
-da	delay line a	[0,1023]
-db	delay line b	[0,1023]
-bb	beacon bias	[0,255]
-bm	beacon modulation	[0,255]
-h	help message	-

Table 3.1: The Alice module is controlled via the bash program *alice-control*. This table contains all parameters that can be easily set, without needing to reprogram the FPGA. The parameters are forwarded to the FPGA, who then passes them on to the delay chips and the driver in the appropriate format.

Example: *alice-control -c 3 -b 5 -m 2 -da 100 -db 330* delays the clock signal a by 500 ps and the clock signal b by 1650 ps. The resulting electrical pulse has a width of about 1150 ps. The VCSEL driver creates a bias current of 335 μA and a modulation current of 236 μA.

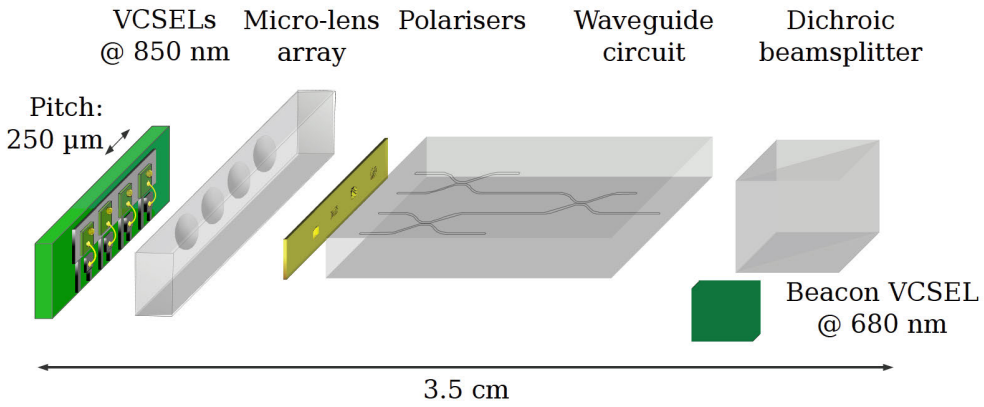


Figure 3.3: The VCSELs (*VI Systems V25A-850C12SM*) emit light roughly around the wavelength 850 *nm*. The distance between the four VCSELs is 250 μm. A Micro-lens array is used to focus the VCSEL beams onto four different waveguide inputs, before which the polarizers are used to prepare the four BB84 states. The waveguide has four inputs but only one output and enforces a spatial overlap of the four states. A beacon at the wavelength of 680 *nm* is used for clock synchronization, beam tracking and besides that, also aiming becomes easier as a red spot can be seen by the user at the entrance pinhole of the receiver. Taken from [30]

Note that the delay chip and the driver will be explored in more detail in section 6.3 and 6.2.

3.1.2 The Alice optics module

The optics module consists of a PCB VCSEL carrier board and several optical components. The PCB contains only a connector, to enable a cable connection with the Alice electronics board and a special gold surface where the VCSEL array is glued on. A schematic of the optics module with an explanation of the purpose of each component can be seen at figure 3.3, but note that the PCB is very simplified in this figure. A more detailed figure of the PCB VCSEL carrier board can be seen at figure 3.4, also with some additional remarks. Finally it should be noted that this PCB VCSEL carrier board was redesigned, but this will explored later in section 4.2.

VCSELs are solid-state lasers, which emit their light perpendicular to their surface. Their cavity consists of distributed Bragg reflectors, which are made up of alternating layers of AlAs and GaAs. Each layer has a thickness corresponding to one quarter of the laser wavelength. The laser wavelength is determined by the

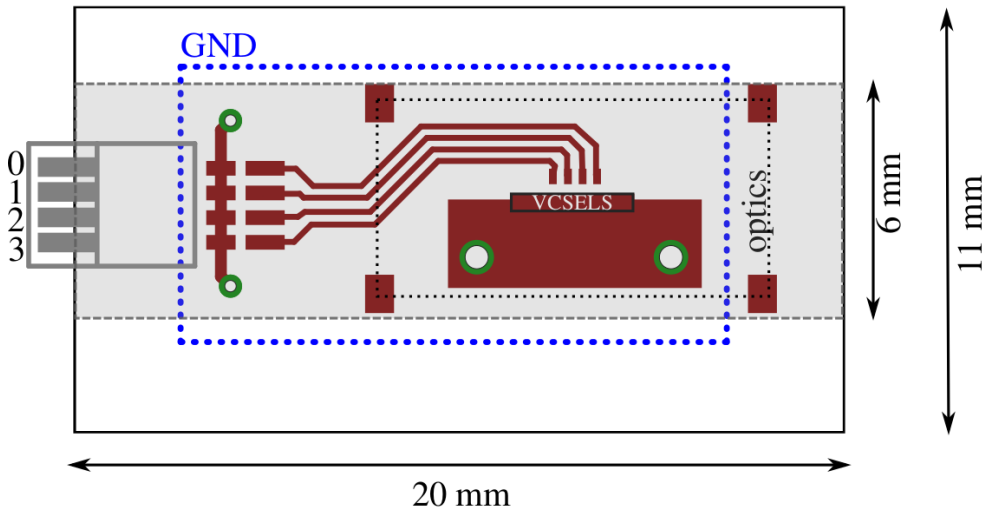


Figure 3.4: The Eagle design layout of the PCB VCSEL carrier board. On the left side the connector can be attached, which routes the signals to the four Vertical-cavity surface-emitting laser diodes (VCSEL). The VCSEL array is glued onto a gold plane (the red rectangle with the black border). The wires running from the connector to the gold plane also have a gold end (the red rectangles in front of the rectangle labeld *VCSELS*). The VCSELs are connected to the wires via bonding. The wire contacts are gold, as well as the VCSELs contacts, which is why a gold wire is used for the bonding process. The bonding process will explained in more detail in section 4.1.3. On the bottom side (in blue) of the PCB is a big groundplane (abbreviated as GND). The optical components (micro-lens array, polarizers, waveguide circuit) will be mounted above the *optics* region (marked by the black doted rectangle). The green rings are viases which connect the VCSELs groundplane to the PCB ground plane, which is in turn connected to the connector ground. Taken from [31].

length of the cavity. The active area inside the cavity is made up of InGaAs-GaAs quantum wells. The VCSEL is typically electrically connected via top and bottom contacts to the PCB.
VCSELs have several advantages compared to edge-emitting laser (EEL) and light-emitting diodes (LED), like the lower requirement of power or current, a more effective conversion of energy and the very narrow and directed beam.
VCSELs are widely used for example for data transmission in optical fibers, laser printers and in optical mice.
More details about VCSEL can be found in the book by Rainer Michalzik [36].

The VCSELs (*VI Systems V25A-850C12SM*) receive the electronic pulse generated by the VCSEL drivers via the connector from the electronics board. The VCSELs then emit a corresponding optical pulse. First an array consisting of twelve VCSELs was used, but only four of those were chosen to be used as a starting point for the BB84 states. The light coming out of the VCSELs maybe polarized along some defined direction or it can also be completely unpolarized[2]. These pulses have been further characterized for a different VCSEL array and the results will be explained in detail section 5.4 and section 5.5.

The micro-lens array focuses the light coming from the VCSEL array into the waveguide to ensure a good coupling with minimal losses.

The wire-grid polarizers (see figure 3.5) are used to prepare the BB84 states $|V\rangle, |M\rangle, |P\rangle, |H\rangle$. These polarizers act as filters (as explained in section 2.3.4.1). This set of polarizers consists of a thin gold grating, created via focused ion beam milling. It features an extinction ratio between 1150 and 1800 and has a transmission coefficient of 0.09 for light with 850 *nm*.
Note that the polarizers are oriented in a way such that after the waveguide the four BB84 states appear, and not right after the polarizer. The birefringence of the waveguide changes the polarization and this effect is counteracted by preparing these adapted states instead.

The waveguide circuit ensures that the four VCSEL outputs are spatially overlapped, which prohibits to distinguish the four BB84 states by just looking which VCSEL lights up. This would basically couple the spatial degree of freedom to the polarization degree of freedom, which is used to encode the information. Thus an observation of the spatial position would allow immediately to determine the

[2]Unpolarized light is not not polarized, but is instead randomly polarized, making it impossible to predict which polarization a photon will have. The polarization quality can be quantified via the degree of polarization, which is 1 for perfectly polarized light and 0 for unpolarized light.

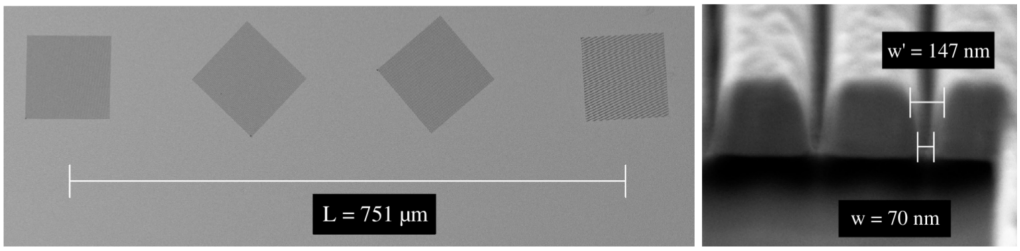


Figure 3.5: The left picture shows the orientations of the four wire-grid polarizers. The gold grid is 265 *nm* thick on top of a glass substrate. Each polarizers covers an area of 120 $\mu m * 120\ \mu m$. The distance between the center of each polarizer is 250 μm. The right picture shows a closeup of the grid with a spatial period of 500 *nm* (from [31]).

polarization, without affecting it. So this action by an attacker would not affect the state and thus would not be detectable by Alice and Bob. This weakness could compromise the complete security of the BB84 protocol. In the description of the BB84 protocol no such element is mentioned, because it is assumed the the four BB84 states are completely uncorrelated with other degrees of freedom. In the experiment or implementation this has to be considered and enforced, since otherwise the security is compromised by this side-channel.
This single-mode waveguide was fabricated by the group of Prof. Dr. R. Osellame at the Politecnico di Milano in Italy. The group used tightly focused ultrashort laser pulses to write the different paths into the glass substrate (*Corning*® $EAGLE^{2000TM}$). The laser effectively increases the index of refraction locally, creating a waveguide structure similar to a singlemode fiber. At so called interaction zones, two of these waveguides are only separated by 7 μm, with the result of evanescent coupling between the modes. The interaction length is 450 μm,which results in a 50 : 50 splitting ration [30]. Unfortunately the polarization states are changed while traveling through the waveguide, due to its small birefringence $\Delta n = 7 * 10^{-5}$. This rotation needs to be compensated, which can be precompensated by the orientation of the polarizers.

When the light emerges from the waveguide it is overlapped with a beacon signal via a dichroic beamsplitter. The beacon VCSEL emits visible light with 680 *nm* (red). Since this Alice module is supposed to be a prototype for a hand-held unit this beacon facilitates aiming, since the wavelengths of the BB84 light is not visible (850 *nm*) for the human eye. Additionally the beacon is used for clock synchronization between the sender and the receiver. A detailed analysis of the hand-held performance of this module was done by Jannik Luhn in his master's thesis [34].

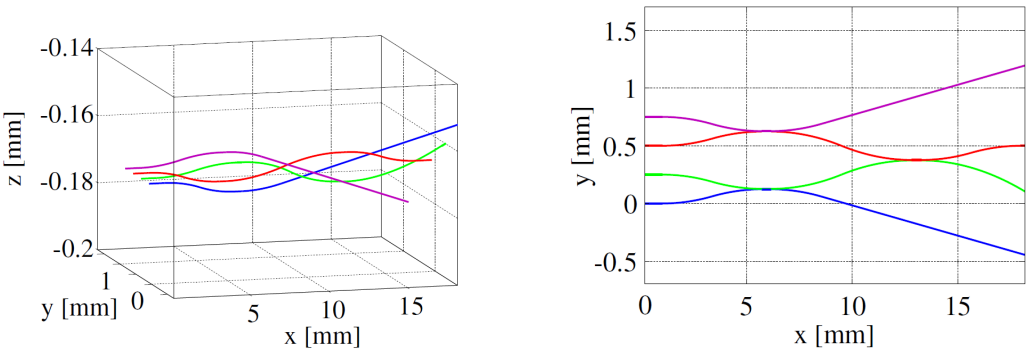


Figure 3.6: On the left side a picture of the waveguide illustrate the arrangement of the four paths in three-dimensional space. On the right side the top view onto the waveguide to see the four paths in a two-dimensional plane. The light enters on the left side, where the four paths are separated by a distance of 250μm. On each intersection of two paths the coupling ratio is 50%, which leads to a unbiased coupling into each of the four paths by all four inputs. The output of the blue, green and purple path is blocked. Only the red path has an open output, which is used as our output of the prepared BB84 state. Taken from [31].

3.2 Bob

Figure 3.7 depicts the receiver setup, consisting of a standard BB84 polarization analyzer unit (PAU) and additional components required for the hand-held operation of the sender module.

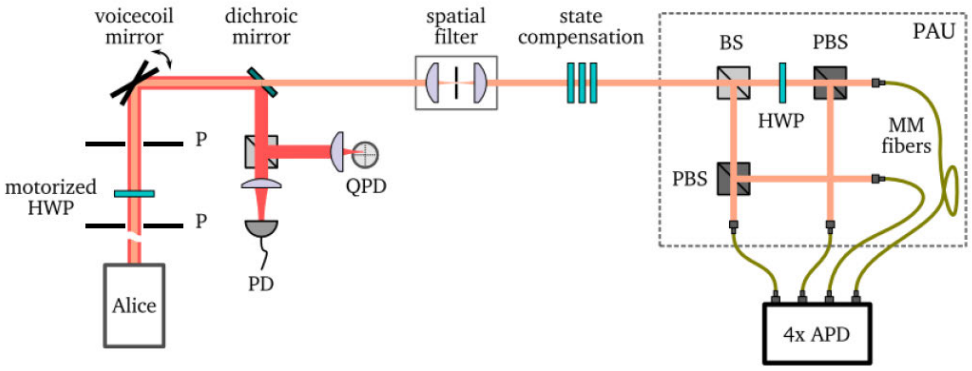


Figure 3.7: A schematic of the receiver setup. The motorized half-wave plate (HWP) is used for the reference frame alignment. Two pinholes (P) enclose the HWP. The voicecoil mirror and the quadrant photo diode (QPD) enable beam tracking. The fast photodiode (PD) serves for clock synchronization. The beam splitter distributes the beacon onto the QPD and the PD. The spatial filter is used to minimize stray light entering the receiver and to close the spatial mode side-channel [29]. The state compensation consists of three wave plates that are used for phase compensation of the polarization states. Then the light enters the polarization analyzer unit (PAU) to measure the four BB84 states. Adapted from [33].

A motorized half-wave plate is used to compensate a rotation that the hand-held user may introduce by holding the device tilted. For the correction a smartphone is placed on the Alice module that reads out its orientation via a gyroscope. The data are than transmitted via Wi-Fi to the receiver computer, which takes the data to orient the motorized half-wave plate mounted in a step motor serving for the alignment of Alice's and Bob's reference frame.

It can be practically impossible for a person to aim through the spatial filter, as the acceptance angle is only $\pm 0.08°$. Due to this additional beam tracking is required. Here, the beacon gets separated by the dichroic mirror from the BB84 beam and is diverted onto a quadrant photo diode (QPD). This creates the reference signal for an electronically controllable voicecoil mirror. This mirror can correct for angular

misalignments up to 3° for both axes.

Alice and Bob need to synchronize their clocks so that they label each pulse or qubit with the same integer number j. Two independent clocks would run out of synchronization after a short amount of time. Due to this the beacon laser is modulated with the clock of the sender. Bob decouples the beacon from the BB84 signal via the dichroic mirror and forwards it into an amplified photodiode (PD), which feeds a clock recovery chip electronics. The clock recovery signal is then transmitted to the timestamp unit that also records the times of the detection events.
This scheme is not ideal, as a loss of the beacon signal requires complex post processing, described in more detail in [34]. This clock scheme does not correspond to an absolute clock synchronization, which is why in the current state of this experiment a known key of definite length was repeatedly sent.

The spatial filter is designed to close the spatial mode side-channel, which would otherwise compromise the security of the setup [29].

In any real setup Alice can not produce each BB84 state with a fidelity of one. Additionally components at the receiver side can be polarization dependent, which means that they additionally alter the states in a non-uniform way. Due to this, a state compensation is employed via three wave plates. Since every incoming state is rotated the same it is unlikely that all states will be perfect afterwards, but the rotation will be performed such that the QBER is minimized (see section 2.3.3).

To further aid a hand-held user to aim through both pinholes (P) (separated by about 15 *cm*) a audio feedback was installed. Here a deep pitch indicates a good coupling and a high pitch indicates bad coupling. Due to this even an untrained user could achieve key rates of several kbits per second [34].

The polarization analyzer unit (PAU) measures the four BB84 states. The beam splitter (BS) reflects and transmits the incoming photon with a probability of 50%. If the photon is reflected it travels along the $\hat{\sigma}_z$ path where it encounters a polarizing beam splitter (PBS) that transmits $|H\rangle$ and reflects $|V\rangle$. If the photon is transmitted by the BS it travels along the $\hat{\sigma}_x$ path. Here the photon encounters a half-wave plate, which performs a Hadamard transformation. After that follows another polarizing beam splitter (PBS) that transmits $|H\rangle$ and reflects $|V\rangle$, which results in total in a measurement of $|+\rangle$ and $|-\rangle$.

Chapter 4

Assembly and design of the PCB VCSEL carrier board

In this chapter we will first see how a PCB VCSEL carrier board is assembled. We see in particular how a new VCSEL array (*RayCan RC12xxx1-A4*) is connected to a PCB VCSEL carrier board. This process consists of three major steps, all of which are explained in the following section. The second section then explores the design of a new VCSEL carrier board. This new board will feature a new connector to allow connections to a new Alice electronics module (designed by C. Sonnleitner [33]). Additionally the new board was made as small as possible, which allowed a decrease of its size (or its area) by a factor of two.

4.1 Assembly

Here we will explore the assembly of a PCB VCSEL carrier board. First we need to clean the PCB. After that the new VCSEL array (*RayCan RC12xxx1-A4*) can be glued to the PCB. This step requires a special electrically conductive glue, because the glue fixes the position of the VCSEL array on the board and also connects the bottom contact of the VCSEL array with the electronic wiring of the PCB. The final step is concerned with the bonding procedure. Here the VCSEL top contacts get connected to the PCB wiring. To perform this step a special bonding machine is required.

This new VCSEL array got assembled on an old PCB VCSEL carrier board (figure 4.1), old compared to the new design of the next section 4.2. This old carrier board works well with the old Alice electronics module, which is explained in detail in chapter 3. The new PCB VCSEL carrier board of the next section only works with the new modular electronics module, unless a connection-converter

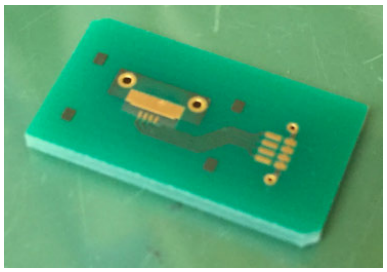


Figure 4.1: The old PCB VCSEL carrier board. The VCSEL will be placed on the left big golden rectangular region. The VCSEL will be glued to that golden rectangle via a specific electrically conductive glue. On the right side a connector will be added, to allow an electrical connection to the Alice electronics module. The VCSEL will then be directly controlled by the laser driver of the Alice electronics module.

module is used. This new electronics module consisits of several smaller modules, which allows faster testing of new electronic components, like new delay lines, logic gates or VCSEL drivers. The old electronics module is one module, where all electronic components are on the same PCB, because the goal was to create a module for hand-held operations for short-distances. Due to the different goals the designs vary drastically.

4.1.1 Preparation

Before any component (Glue, VCSEL array, gold wire or connector) can be added to the PCB it needs to be cleaned. The first cleaning process involves Acetone. The whole board should be engulfed in a small Acetone bath to remove of any fats and oils. The second cleaning process involves Isopropyl. Isopropyl should be applied similarly to Acetone. Both substances remove fats, but the order is important, because Acetone vaporizes rather quickly and leaves a thin Acetone film on the PCB surface. This small surface is then removed with Isopropyl, which vaporizes much more slowly and does not leave such a film. After the cleaning processes the PCB should be briefly dried.
It is also possible to clean a PCB using Oxygen plasma. However, this can not be recommended for PCBs with gold surfaces. The plasma cleaning process made bonding to the gold surfaces practically impossible. The reason may be that the Oxygen plasma made the gold perceptible to create Gold-oxid. This Gold-oxid then seemed to forbid any bonding attemps with a pure gold wire.

4.1.2 Gluing

The VCSEL get glued onto the PCB with a special electrically conductive glue (*EPO-TEK® H20S* from Epoxy Technology). This glue consists of two components, A and B, which have to be mixed together with a mass ratio of 1:1. Note that both components have a different density, which means the mass ratio does not translate to a volume ratio. A precision scale is necessary to arrive at the desired mass ration for these two components. Once the components are combined the glue can be put on the gold rectangle, where the VCSEL array is supposed to be placed. The glue can be applied with the edge of a razor, or alternatively via a very thin wire. The glue should only be applied very thin, such that the VCSEL array can not be completely submerged in it. If too much glue is used one can easily clean the PCB board for a second try. Proper glue distribution is rather tricky and requires some training. Once the glue is appropriately distributed on the gold surface the VCSEL array can be placed using a tweezer on top of the glue.
Before the VCSEL array can be placed on the glue one should check its orientation. The individual VCSEL, which make up the VCSEL array, are not rotationally symmetric, which can be seen on figure 4.3. Therefore the placement of the VCSEL arrays in the package (the package in which they are delivered) has to be checked via a microscope. The correct way to place the VCSEL array is with all electrical top contacts facing the four golden wires, coming from the connector. If the VCSEL array is oriented differently (180° degrees rotated) the bonding wire can block a part of the VCSEL beam.
After the placement of the VCSEL array inside the glue, with the right orientation and parallel to the side of the gold rectangle, the glue needs to be hardened. This is done by putting the VCSEL carrier board inside a oven for 90 minutes at a temperature of 80° Celsius.

4.1.3 Bonding

A wedge-bonding machine is used to establish an electrical connection between the wiring of the PCB VCSEL carrier board and the VCSEL. A wedge-bonder uses ultrasonic energy and pressure to connect the gold wire via a (bonding) needle to the gold surface. The bonding machine has six parameters which can be tuned to change the bonding behavior. It is possible to change the bonding height, the temperature of the bonding wire, the power of the first bond P_1, the bonding-time of the first bond T_1, the power of the second bond P_2 and the bonding-time of the second bond T_2. But the sample defines the nature of the wire (in this case we use gold) which in turn defines the wire temperature. The sample also defines the bonding height so effectively one can only tune the four remaining parameters for our PCB VCSEL carrier board. The appropriate choice of parameters depends on

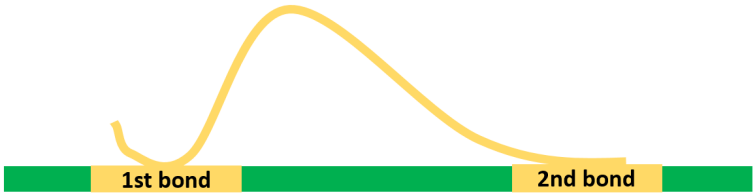


Figure 4.2: This picture illustrates the difference between the two bonds. The first bonding process or step leaves a bit of an extra wire dangling around. The second bonding process or step leaves no extra wire and creates a flat end. This needs to be taken into account, when the first and second bonding surface are chosen. When the VCSEL array is bonded the first bond should be on the gold contact at the end of the PCB wiring, while the second bond should be on the VCSEL top contact. This way no dangling wire can hang into the light emission area of the VCSEL. It should also be stressed that the first bond can be done more accurately than the second bond. During the first bonding process the bonding needle can be seen directly, while during the second bonding process the gold wire is partially in front of the needle which makes the proper placement more difficult.

the sample and has to be found empirically.
In order to find the correct bonding parameters the bonding behavior should be tested with a dummy. Here ideally an identical PCB is used to find a set of bonding parameters for this particular PCB. At the same time the opportunity should be used to learn a skillful use of the bonding machine. It should also be noted that the bonding process is not completely symmetric, which means that there is a difference between the first and the second bond. This difference is displayed in figure 4.2 and explained in the caption. A failed bonding attempt always leaves behind some gold-dirt. Ideally each of the four bonding attempts works at the first time, since the number of tries is limited, because at some point too much gold-dirt will be on the bonding pad. The quality of the bonding can be superficially checked via a microscope. Ultimately the electronic connections have to checked by connecting a laser driver to the VCSEL and confirming via a camera that they emit light.

4.1.4 α 2017

The PCB VCSEL carrier board I assembled was named α 2017, because it was the first new VCSEL board that was created in 2017. The preparation went well, but unfortunately I encountered difficulties during the other steps.
During the gluing process I distributed too much glue on the VCSEL carrier sur-

face, but noticed it only after the hardening of the glue, when the VCSEL array was already submerged in it (figure 4.3). The placement of the VCSEL array inside the glue is also not optimal, because the array is slightly tilted, which can be seen once photos of the array are taken. There not all four VCSEL can be focused at once, hinting at the fact that the array is tilted and the different VCSEL are at different focal lengths from the objective.
If another module is assembled it should be checked that the glue is distributed extremely thin, which requires a lot of practice. Additionally the VCSEL placement needs to be done with great care. It should be ensured that the array is placed parallel to the side of the gold rectangle and that the array is parallel to the surface of the carrier board. This could be done by pushing the VCSEL array carefully down inside the glue. During this it has to be ensure that no extra glue spills on top of the VCSEL, thereby creating a short circuit between top and bottom contacts or spilling onto the light emission area.
For the bonding of α 2017 (figure 4.3) I found the bonding parameters $\left(P_1 = 5, T_1 = 5, P_2 = 5, T_2 = 5\right)$ effective. Due to the differences of the first and the second bond, the first bond was placed on the ends of the wires and the second bond was placed on the VCSEL top contact, to ensure that no dangling wire hangs above the light emission area of the VCSEL.

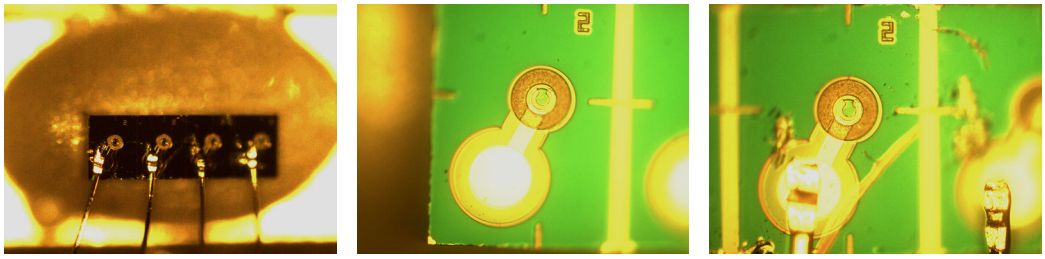


Figure 4.3: The left picture show the whole array after the bonding procedure. The picture in the middle shows first VCSEL of the array (counting from left to right) before the bonding procedure. The right picture shows the second VCSEL of the array after the bonding procedure.

4.2 Design of a new VCSEL carrier board

In the previous chapter a new VCSEL array was assembled onto an old PCB VCSEL carrier board. In this chapter I describe the design of a new PCB VCSEL carrier board. This new board features new micro-coaxial connectors (*CONN UMC RCPT STR 50 OHM SMD* from Molex). These new connectors are neces-

sary to use the new modular Alice electronics module designed by C. Sonnleitner [33]. Additionally the goal was to make this new board as small as possible, in order to improve the capability to be integrated. Furthermore it was important to have matching impedances of the PCB wires and the connectors to minimize reflections of the signal at the connections.

The old carrier board used one big connector, containing wires for all four VCSELs. The new connectors can only carry signals for one VCSEL each, which is why now four connectors are needed. This allows to use different delay chips, logic gates and drivers for each of the four VCSELs and improves the individual VCSEL testing. These new connectors are also more firmly placed on the PCB, while the old ones could often easily brake from the Alice electronics module. The total size of these four connectors basically determined the total size of the module. The different connector types can be seen on figure 4.4.

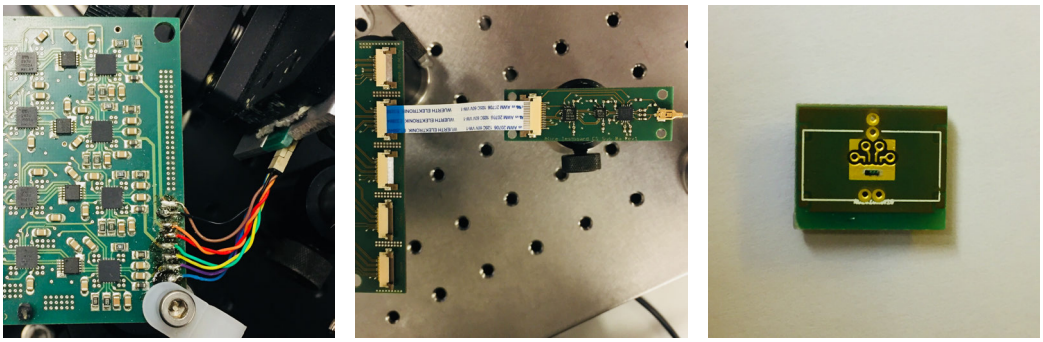


Figure 4.4: The left picture shows the old connector between the Alice electronics module and the PCB VCSEL carrier board. On the electronics side the cables are simply soldered onto the PCB. The picture in the middle shows the new connectors. The mainboard, containing the FPGA and the clock, connects via flexible flat cables to the module containing the delay chip, the logic gate and the driver. This module then has the micro-coaxial connectors (*CONN UMC RCPT STR 50 OHM SMD* from Molex) which connects it to the PCB VCSEL carrier board. The picture on the right shows the new PCB VCSEL carrier board. The array is glued on the PCB and bonded to it, but the bonding wires are too small to be seen.

Besides the addition of the new connectors the size should also be reduced. Since the minimum size of the board is determined by the total size of the four connectors their size was reduced. In particular the size of their soldering pads was reduced by a factor of two. This increases the soldering difficulty of these connectors. Soldering by hand turned out to be rather difficult and not stable as connectors broke off the PCB when a cable was supposed to be taken out of the

connector. Using reflow soldering for this task turned out to be a good choice. The soldering paste could be applied by hand via a syringe, without the need to use a soldering-mask. This way reliable connections could be achieved, without creating short-circuits between the signal and the ground. This allowed the construction of a compact carrier board 11.2 *mm* wide and 9.3 *mm* high, whose total size is reduced by a factor of two, compared to the previous carrier board. Both of these size related issues can be seen in figure 4.6, as well as a fully soldered board with the connectors.

The third issue that needed to be addressed was the impedance matching. The cables and the connectors have an impedance of 50 Ω, which had to be matched by the PCB wires, to minimize the reflections of the signal at the intersection of the wires and the connectors. For this carrier board groundplanes were chosen instead of ground wires, where the impedance Z calculates as 64.09 Ω. This calculation was performed via an online-calculator for coplanar waveguides with ground characteristic impedance[1]. The evaluated formula is taken from the book *Transmission Line Design Handbook* by Brian C. Wadell [46]. Unfortunately this does not precisely match the desired impedance of 50 Ω, but this was the best possible value that could be achieved. In principle there are five parameters that affect the impedance, but three of them are fixed by the PCB manufacturer, like the thickness $h = 1.5\ mm$ of the PCB, the relative dielectric constant $\varepsilon_{eff} = 4.3$ and the wire height $w_h = 0.07\ mm$. The wire width and the gap width are also not arbitrary and for this design the wire width $w = 0.5\ mm$ and the gap width $g = 0.16\ mm$ was chosen.

Note that when the four wires come together (in front of the VCSEL placement surface) they decrease in size from 0.5 *mm* to 0.2 *mm*. This is necessary to ensure that the four wire ends are close enough together to allow a bonding with the VCSEL, who are separated by merely 250 μm. Figure 4.5 shows the eagle layout of the board.

[1] Link: http://chemandy.com/calculators/coplanar-waveguide-with-ground-calculator.htm

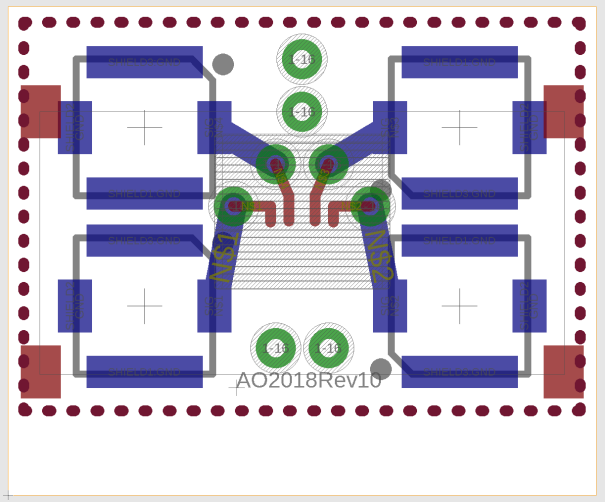


Figure 4.5: The Eagle design layout. Elements which are placed on the top side are red. Elements which are placed on the bottom side are blue. Both sides have two groundplanes, indicated by the doted rectangle around the four connectors. Only one rectangle is visible, because they are placed on top of each other. The orange border marks the physical size of the board. The green circles are viases which connect the two groundplanes and allow a faster heath distribution. The new carrier board is 11.2 *mm* wide and 9.3 *mm* high.

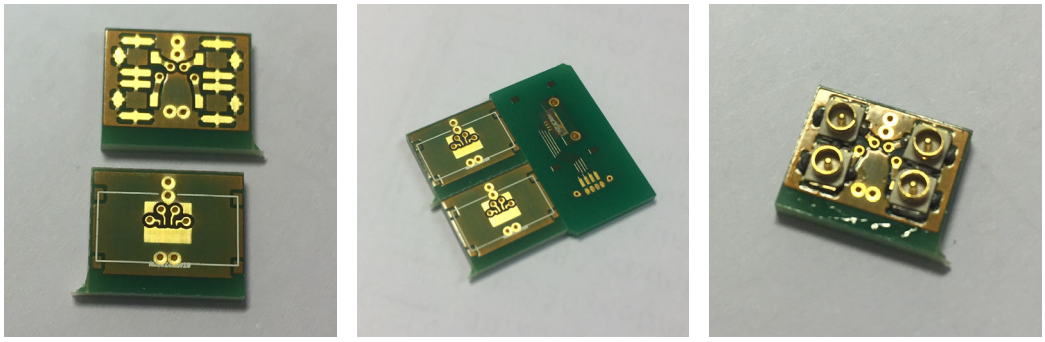


Figure 4.6: The left picture shows the top and bottom side of the new VCSEL carrier board (11.2 *mm* x 9.3 *mm*). Both sides have big ground planes. All the wiring is done with gold, because the VCSEL need to be bonded to gold contacts. For further processing (addition of the other optics module) a 1.5 *mm* wide (naked) border was added. The white rectangle provides guidance and orientation during the placement of the additional optical components (spacers, polarizers etc.). The picture in the middle shows that the new module is roughly only half as big as the original module. The picture on the right shows the bottom side of the module with the connectors attached.

Chapter 5

Characterization of the VCSEL

In the previous chapter the assembly of a PCB VCSEL carrier board (α 2017) was explained. There a new VCSEL array (*RayCan RC12xxx1-A4*) was placed on an old PCB VCSEL carrier board. In this chapter this newly assembled board, or more precisely the VCSEL array, is characterized to test how suitable it is for QKD. A comprehensive characterization is crucial, since a detailed understanding of the photon source is necessary to ensure optimal operation parameters, as well as being certain that the source does not offer any side-channels that can be exploited.

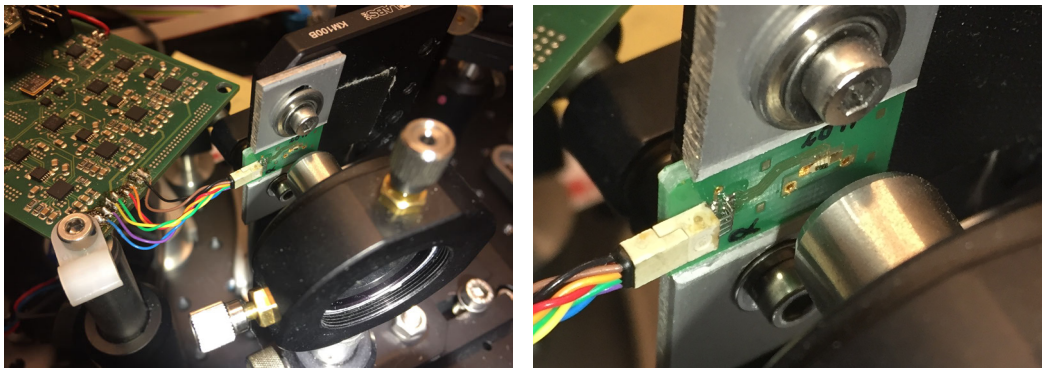


Figure 5.1: The left picture shows the Alice electronics module, the PCB VCSEL carrier board α 2017 and a lens. This basic setup is used for all characterization measurements of the VCSEL. The right pictures shows a zoomed in view on α 2017. The grey blob on the PCB is the glue that connects the VCSEL to the PCB ground plane. The small black rectangle inside the glue is the VCSEL array.

5.1 Optical output power measurement

The first questions with regard to this new VCSEL array is how the optical output power depends on the driving current, which is provided by the laser driver. This can be measured with a power meter, that picks up the light emitted by the VCSELs for different currents. This measurement will also reveal the threshold current at which the VCSELs start lasing.

5.1.1 Measurement setup

The measurement of the power emitted by the VCSELs can be easily done via a power meter, on which the light of the VCSELs is collimated on. The power meter is connected via USB to a computer, which also connects to the Alice module. This allows the control of both devices with the computer. For this purpose a measurement script was written in Python, which can change the parameters for the electronics module and record data from the power meter.
For this measurement it is sufficient to operate the VCSELs in the continuous wave mode, which is why all pulse parameters are set to zero. The only parameter that will be varied is the bias b, which determines the bias current

$$I_{bias} = 100\,\mu A + 47\,\mu A * b.$$

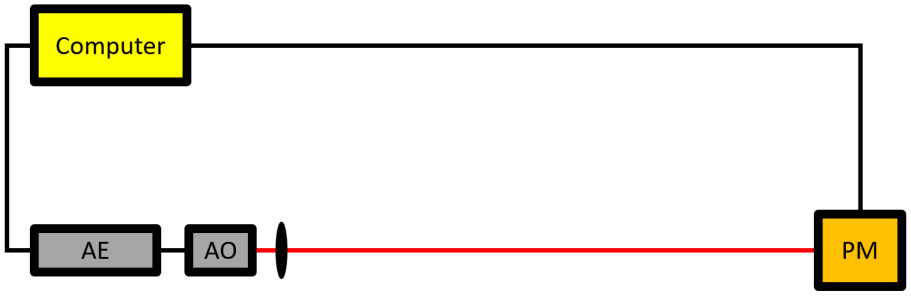


Figure 5.2: The measurement setup for the current-power measurement. A computer is connected via USB to the Alice module. The Alice module consists of the electronics module (AE) and the PCB VCSEL carrier board (AO). The VCSELs emit light which is collimated via a lens onto the power meter (PM). The power meter is connected via USB to a computer. The laser light is depicted as a red beam. The cable connections are depicted as black lines. This holds for all future measurement sketches as well.

In general we have $b \in [0,255]$, but the VCSELs can not take arbitrary high currents. The maximum current for this new VCSEL array is $I_{MAX} = 5\,mA$, according

to its data sheet. This means the maximal bias $b_{MAX} = 104$. Due to this limit a maximal bias value $b_{max} = 100$ was chosen, which results in maximum output current $I_{max} = 4.8\ mA$. This bias value will be approached in steps of size five, for the first measurement run. This first measurement run will give an overview over the overall behavior of the power with respect to the current. It is expected that there is some peak power for some current, after which the power will drop again for increasing currents, this so called thermal roll-over is caused due to the internal heating for high currents [36].
A second measurement run will take a closer look at the region for small output currents to determine the threshold current. Here the current-power relation is expected to be linear. Due to this the second measurement run will approach a maximal bias $b_{max} = 42$ in steps of size one.
The following two tables give an overview over all used parameters and their respective values:

parameter	description	values
-c	channel	$0, 1, 2, 3$
-b	bias	0, 5, 10, 15, ..., 100
-m	modulation	0
-da	delay line a	0
-db	delay line b	0
-bb	beacon bias	0
-bm	beacon modulation	0

Table 5.1: The alice-control parameter set for the first current-power measurement run.

parameter	description	values
-c	channel	$0, 1, 2, 3$
-b	bias	0, 1, 2, 3, 4, 5, ..., 42
-m	modulation	0
-da	delay line a	0
-db	delay line b	0
-bb	beacon bias	0
-bm	beacon modulation	0

Table 5.2: The alice-control parameter set for the second current-power measurement run.

5.1.2 Measurement results and discussion

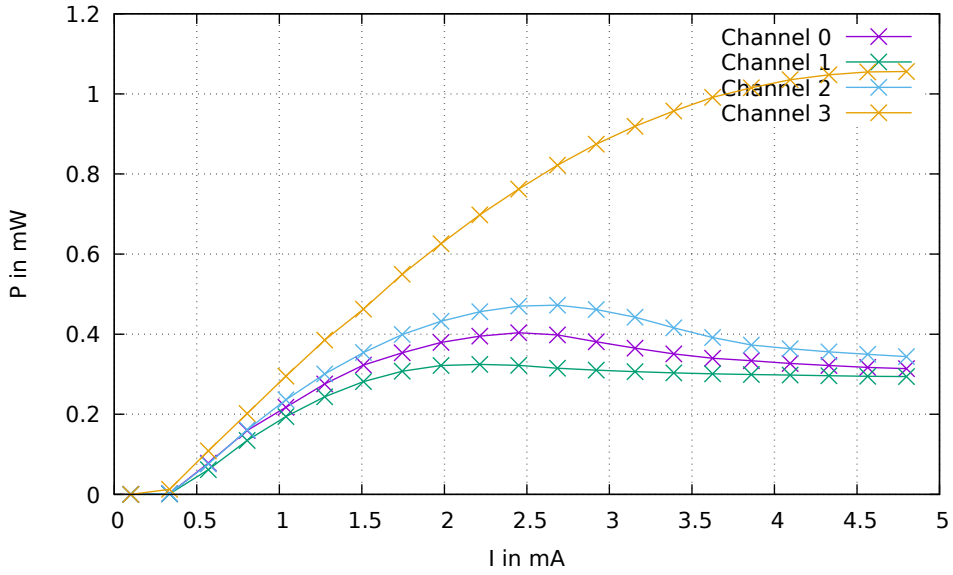


Figure 5.3: The measured current-power curves for the first measurement run.

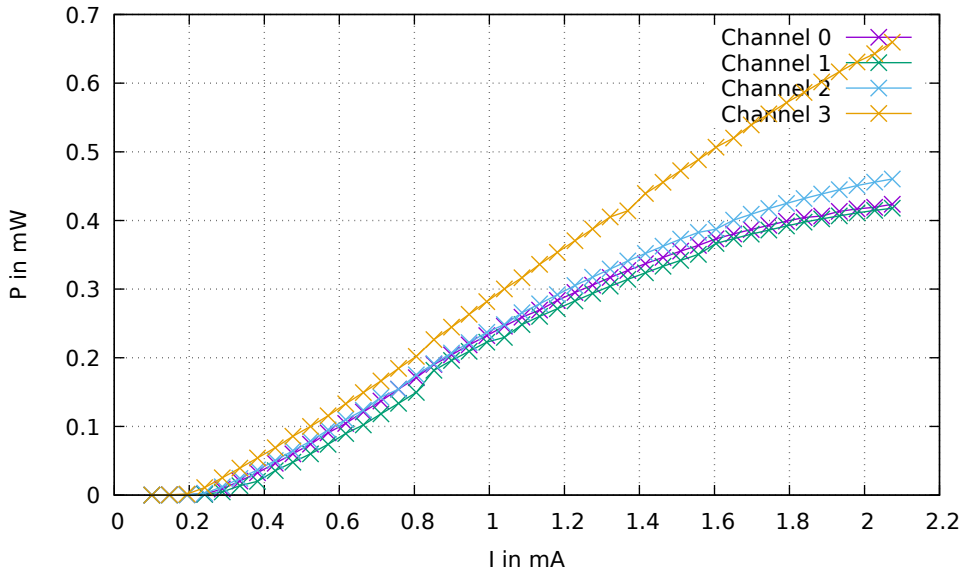


Figure 5.4: The measured current-power curves for the second measurement run.

The overall shape of both measurement runs is as expected. The first run shows a peak power, after which the power drops again for an increasing current. This run also shows a rather different behavior of channel 3 with respect to the other channels. This extraordinary behavior was repeatedly observed. It turned out that it is reproducible and not a fluke. The other three channels on the other hand behave rather similar.
The second run shows a linear behavior, for small current values, starting at threshold current until roughly 1 mA. The run also shows that for small currents the four channels behave more similarly. However, channel 3 still sticks out as the most *powerful* channel. The other three channels have very similar power outputs for a fixed current. The second run also shows that the VCSELs lase for a bias $b = 3$. Thus the lasing threshold current lies somewhere between $I_{b=2} = 194\ \mu A$ and $I_{b=3} = 241\ \mu A$.
The extraordinary behavior of channel 3 can not be explained in a satisfying way. It is possible that the bonding quality is different for the four VCSELs and that the connection for channel 3 is much better than the others, which could result in a higher efficiency. However, under the microscope no difference can be seen. There is unfortunately no known other way to test the bonding quality, except via the microscope or by checking the light output. It could also be possible that this difference is introduced during the manufacturing process, possibly during the cutting of the array, since this special VCSEL corresponding to channel 3 is at one of the two end points (see Appendix section 8.3 for a figure illustrating the VCSEL channel correspondence). Similar behavior has been measured for other arrays by Dr. G. Mélen during her characterizations of VCSEL arrays [31]. There also extraordinary behviour was observed for a VCSEL at the edge of the array.
These results show that if all four VCSELs need to be operated such that the intensity for all four channels is equal, then different parameters should be be used. At least Channel 3 may need a different set of parameters then the other channels. If this is not checked during the operation this may introduce a side-channel, since the different intensities could be used to distinguish the four VCSELs and thus the four BB84 states.

5.2 Stability of the output power

After the measurement of the power it is important to check the stability of the emitted power, for a fixed bias or bias current. If the power fluctuates fast and drastically future measurements can be more difficult, because fluctuating power can influence the measurement results. Future measurements, discussed in this thesis, will take only a couple of minutes, so this timescale will be of interest to us. For this measurement the same setup as before can be used.

5.2.1 Measurement setup

The measurement setup for the time-power measurement is identical to the setup for current-power measurement, depicted in figure 5.2.

The computer controls again the Alice module and receives data from the power meter. Now we are interested in the stability of the power for a fixed current or bias value. In principle any current value from the allowed ones, which is below the maximum current of the VCSELs, can be taken. The stability can also be checked for any amount of time. For future experiments a stability for a couple of minutes should be sufficient to ensure that the VCSELs behave the same during one measurement run.
For this measurement a bias $b = 85$ was chosen, which corresponds to a bias current $I_{bias} = 4.095\ mA$. The power for this fixed current was measured continuously for ten minutes and for each of the four VCSELs.

parameter	description	values
-c	channel	$0, 1, 2, 3$
-b	bias	85
-m	modulation	0
-da	delay line a	0
-db	delay line b	0
-bb	beacon bias	0
-bm	beacon modulation	0

Table 5.3: The alice-control parameters for the time-power measurement.

5.2.2 Measurement results and discussion

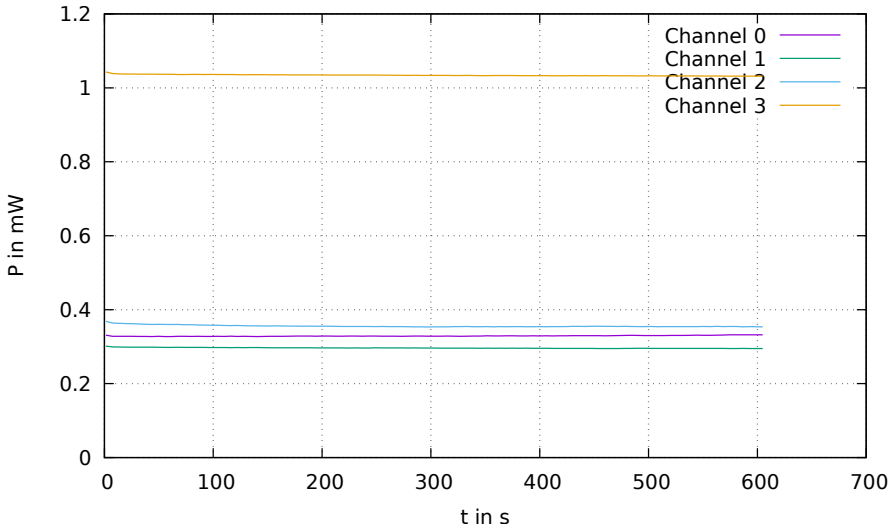


Figure 5.5: The measured time-power curve.

Several measurements of this kind were performed, all of which are accurately represented by the observed behavior depicted on the right plot of figure 5.5. All four channels show a stable behavior on the order of minutes.
After that similar measurements were performed with longer measurement times. Additionally runs were performed with breaks of several minutes between different measurement runs. Short pulse parameters, like the one from the spectrum measurement (see section 5.3), were also tested. All the gathered data showed a stable behavior for all four VCSELs and for each measurement run.
The stability of the VCSELs is pretty good (on the order of μW), since future measurements rely on a constant behavior of the VCSELs.

5.3 The Spectrum

A measurement of the spectrum serves many purposes. The security of the BB84 protocol rests on the fact that the four BB84 states that we prepare are identical in every degree of freedom, with the exception of the the degree of freedom, which is used for encoding. Due to this it is mandatory that the four VCSELs have identical wavelengths, or that their slightly different spectra have a big enough overlap so that a filter can be used to cut off any unique feature of the spectrum from one particular VCSEL.

5.3.1 Measurement setup

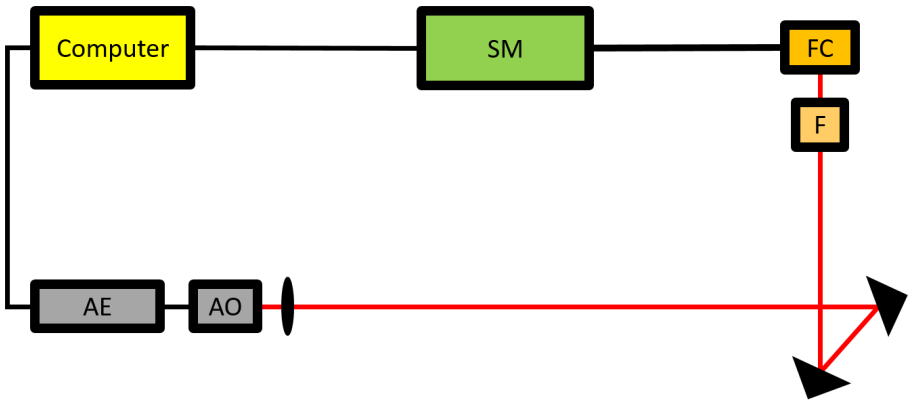


Figure 5.6: The spectrometer setup for the measurement of the spectrum. A computer is connected via USB to the Alice module. The Alice module consists of the electronics module (AE) and the PCB VCSEL carrier board (AO). The VCSELs emit light which is collimated via lens and directed into a fiber coupler (FC). Filters (F) are necessary to keep the power at the spectrometer (SM) under a certain level to avoid damaging the spectrometer. The spectrometer consists of a grating and a camera. The deviation of the light beam caused by the grating is used to deduce its wavelength.

The measurement setup for the spectrum is depicted in figure 5.6. The spectrometer consists of a diffraction grating and a camera. The grating deflects the light according to its wavelength and distributes it onto different positions on the camera, from which we obtain a wavelength dependent intensity.

Since we use short pulses for QKD we measure the spectrum with pulse parameters.

parameter	description	values
-c	channel	0, 1, 2, 3
-b	bias	1
-m	modulation	255
-da	delay line a	100
-db	delay line b	403
-bb	beacon bias	0
-bm	beacon modulation	0

Table 5.4: The alice-control parameters for the measurement of the spectrum.

5.3.2 Measurement results and discussion

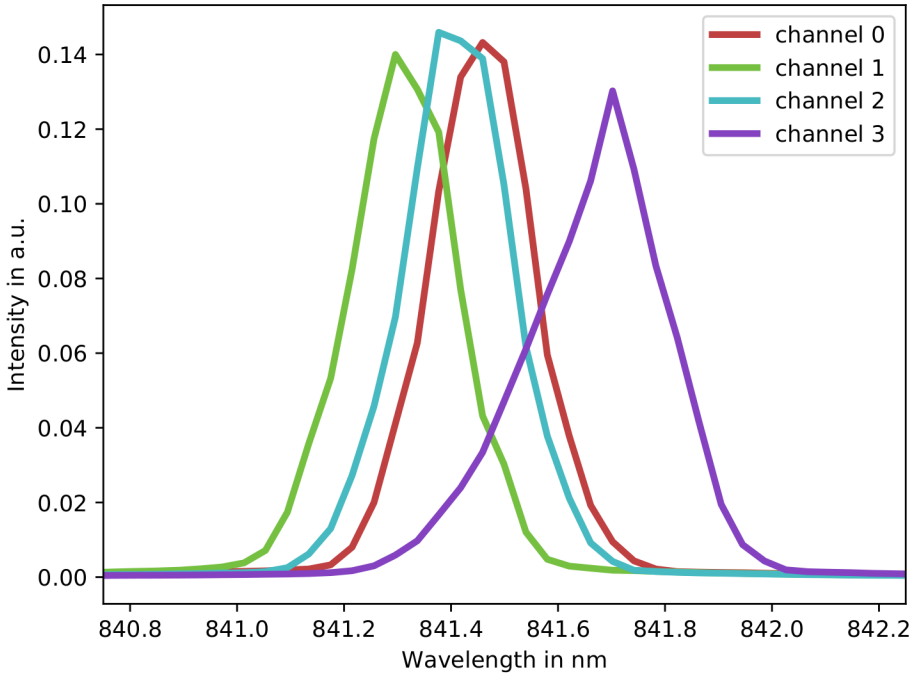


Figure 5.7: The measured spectrum of the four VCSELs.

The four VCSELs have a sharp spectrum. The VCSELs corresponding to the channel 0,1 and 2 also exhibit a somewhat large overlap. Unfortunately channel 3 is very far off and has practically no overlap with channel 0. Due to this no wavelength-filter can be used to cut off unique parts of the spectrum. This VCSEL array thus opens a side-channel, since at least some of the four BB84 states can be distinguished by measuring the wavelength.

To close this side-channel for future arrays one could pre select single diodes, by characterizing each one and then placing the most suitable ones on the PCB VCSEL carrier board. Another option would be to use MEMS-tunable VCSELs, where the wavelength can be manually tuned to enforce the overlap. A different solution might also be to use local heating of the VCSEL to tune its wavelength.

5.4 Temporal pulse shape

Since we want to generate short pulses with the four VCSELs we need to explore the pulse parameters space, in order to find which parameters work best for each VCSEL. It is important to ensure that each pulse is as short as possible. It is also vital that all four VCSELs can create similar shapes, so that they can not be distinguished via their temporal pulse shape, since this would otherwise create a side-channel. For this measurement an avalanche photodiode (APD) and an oscilloscope is needed.

5.4.1 Measurement setup

The oscilloscope (*LeCroy Waverunner 640 Zi*) is configured to record the time-difference between the signal from the APD, which corresponds to the detection time of a photon, and one from the clock (Alice's clock). The oscilloscope then just counts how often which time difference occurs. This creates a histogram that depicts the temporal pulse shape.
The APD can measure maximally four million events per second. A higher value should not be reached, since this can permanently damage the APD. To prevent this several filters are mounted in front of the APD, together with a wavelength-filter. The wavelength-filter reduces stray light from the laboratory and ensures that practically only light from the VCSELs triggers the APD.
The computer can read out the latest APD counts via the time-to-digital-converter (TDC) and the program *counter*. Additionally the data can also be recovered from the oscilloscope. Several manipulations on the received data can be done with the oscilloscope, which is why it is used to directly construct the histogram, which corresponds to the temporal pulse shape. For the histogram the time-difference of Alice's clock and an APD detection event is measured.

In the next section the measurement results for the parameters given in the table below can be seen. These parameters correspond to very short pulses, which are typically of interest for QKD.

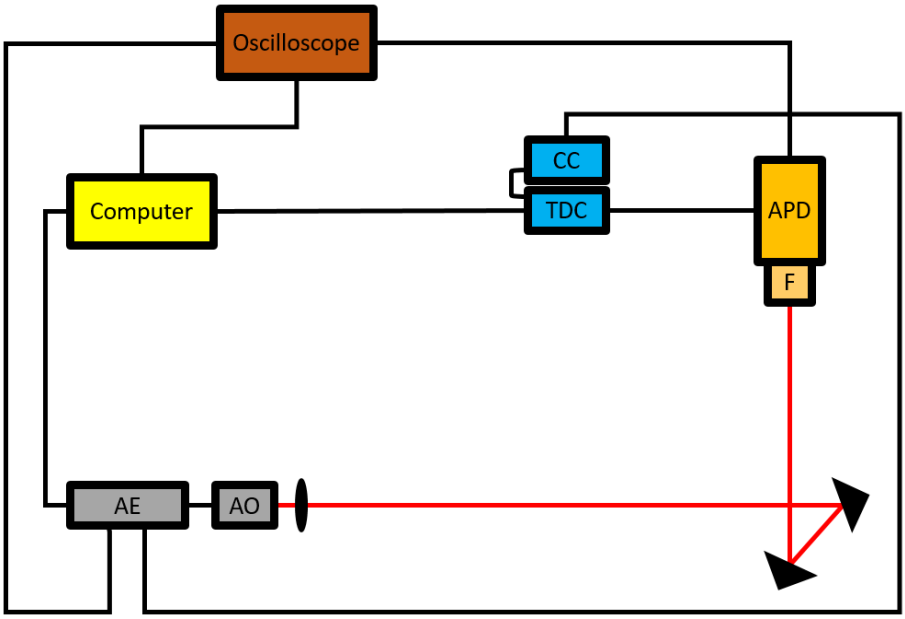


Figure 5.8: The measurement setup for the temporal pulse shape measurement. A computer is connected via USB to the Alice module and to the oscilloscope. The Alice module consists of the electronics module (AE) and the PCB VCSEL carrier board (AO). The VCSELs emit light, which is here depicted as a red beam, which is collimated with a lens. The light is then coupled via two mirrors into an avalanche photodiode (APD). Before the light hits the APD several filters (F) decrease its intensity, to ensure that the APD is not oversaturated. The clock signal from the Alice electronics module (AE) is forwarded to a clock conversion unit (CC). There the clock is converted from a 100 MHz clock to a 40 MHz clock. The 40 MHz clock is then coupled into a time-to-digital-converter (TDC) which is also connected to the computer and the APD. Each trigger of the APD is here coupled with a time-stamp. This combined data is then forwarded to the computer. Additionally the signal from the APD and from the Alice module clock is transmitted to an oscilloscope. The oscilloscope is configured to record the temporal pulse shape. The pulse shape data can then be accessed via the computer, which is connected via USB to the oscilloscope. The TDC can be used to read out data directly from the APD, via the *counter* program. The oscilloscope can also do that, but it can additionally perform more complex data manipulation, which is necessary for the measurement of the temporal pulse shape.

parameter	description	values
-c	channel	$0, 1, 2, 3$
-b	bias	2
-m	modulation	255
-da	delay line a	150
-db	delay line b	360
-bb	beacon bias	0
-bm	beacon modulation	0

Table 5.5: The alice-control parameters for the temporal pulse shape measurement.

Note that a wider set of parameters has also been explored whose plots can be found in the appendix section 8.4.

5.4.2 Measurement results and discussion

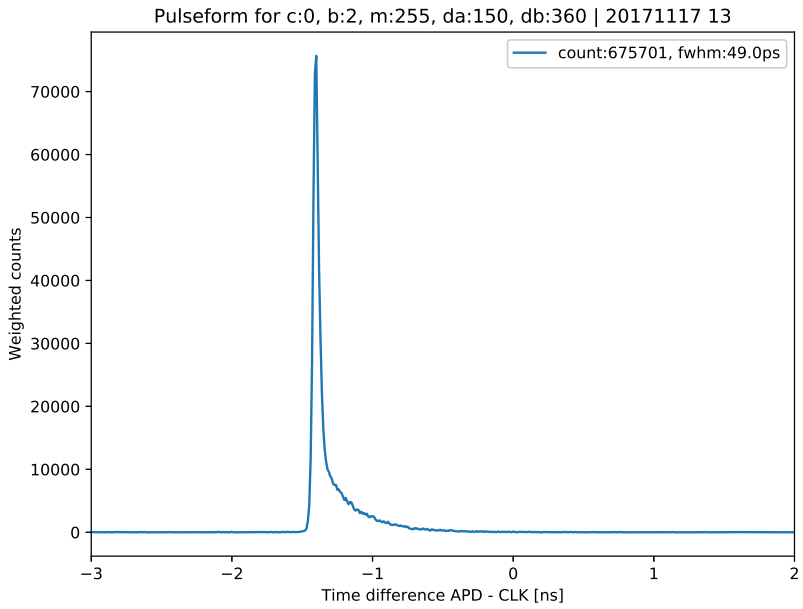


Figure 5.9: The measured temporal pulse shape for channel 0.

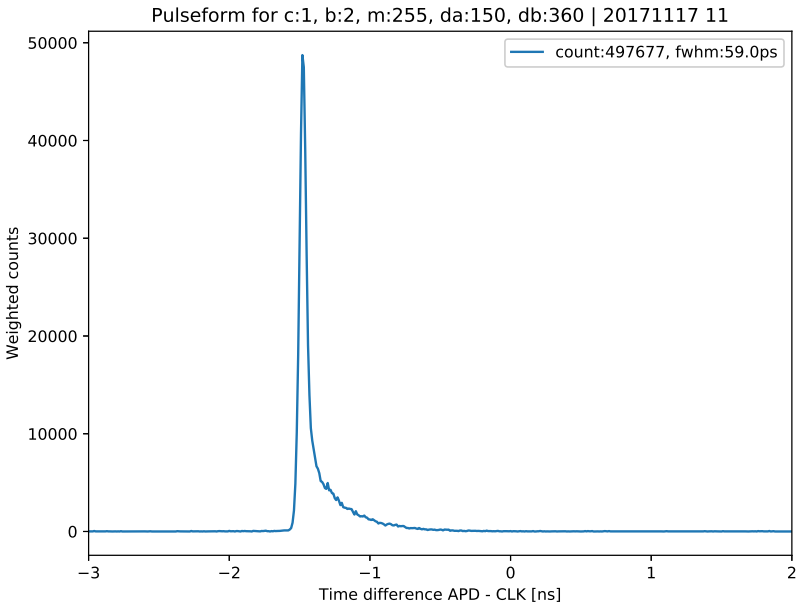


Figure 5.10: The measured temporal pulse shape for channel 1.

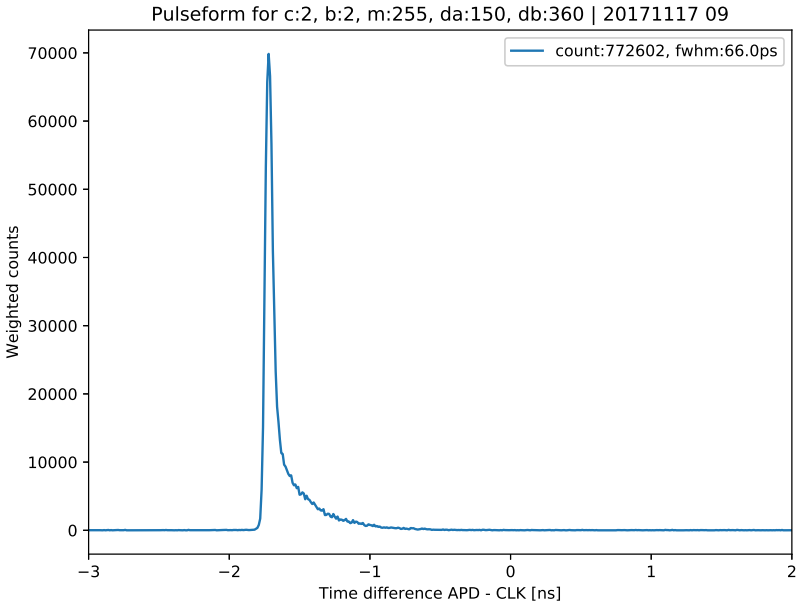


Figure 5.11: The measured temporal pulse shape for channel 2.

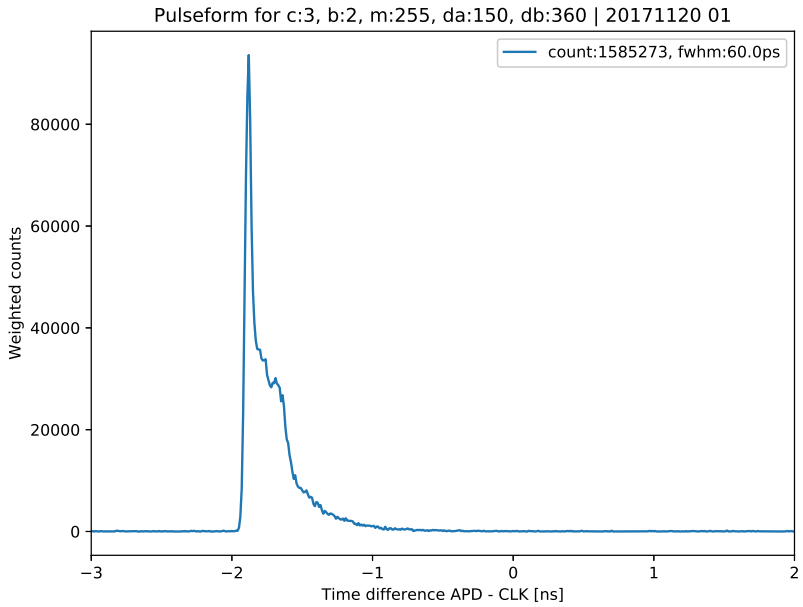


Figure 5.12: The measured temporal pulse shape for channel 3.

The weighted counts N_w are calculated as follows from the measured count:

$$N_w = N_{\Delta t} * \frac{c_{sec}}{\sum_{\Delta t} N_{\Delta t}}.$$

Here $N_{\Delta t}$ denotes the number of photons that were detected with the time difference Δt. The counts of photons per second are denoted by c_{sec}. The measured number for c_{sec} is in the top right corner of each plot denoted as *count*, next to the Full width at half maximum (FWHM), which characterizes the temporal width of the temporal pulse shape. $\sum_{\Delta t} N_{\Delta t}$ denotes the sum of $N_{\Delta t}$ over all measured time difference Δt, which is the the total number of detected photons.

Further measurements of the temporal pulse shapes for different parameters can be found in section 8.4.

It is possible to create short pulses with a FWHM of about 60 *ps* with all four VCSELs. Channel 0,1 and 2 have a similar temporal pulse shape. The shape of channel 3 is a bit odd, but in later experiments a similar shape could be changed to shape which comes closer to the shape of the other three channels simply by improving the position and orientation of the lens in front of the VCSELs (see section 8.4 figure 8.4).

Identical arrivals times of the VCSELs can also be achieved by using the same delta delay $|da - db|$ to create the same temporal pulse shape, while increasing both delays by the same amount. For the presented pusles (figures 5.9, 5.10, 5.11 and 5.12) the biggest time difference is 500 *ps*, which can be eliminated by adding or subtracting 100 from both delay parameters.
Despite some differences, it seems possible to find suitable parameters, which may be distinct, for all four VCSEL to create four identical temporal pulse shapes. It is important that this can be done, since a different temporal pulse shape could otherwise be exploited to distinguish the four BB84 states. Thus such a difference would constitute another side-channel.
Since identical temporal pulse shapes can be created with this VCSEL array it can be used for QKD.
The additional plots in the appendix section 8.4 show further interesting behavior for different sets of parameters. More differences between the four VCSELs become apparent for longer pulse lengths and for an increased bias.

5.5 Tomography

After the measurement of the temporal pulse shape it is of importance to know which polarization state are emitted by the VCSELs, as well as their respective degree of polarization. For this purpose a tomography has to be performed. The tomography is supposed to be done for interesting pulse parameters. In order to also allow a measurement of the temporal pulse shape the oscilloscope is added to the setup.

5.5.1 Measurement setup

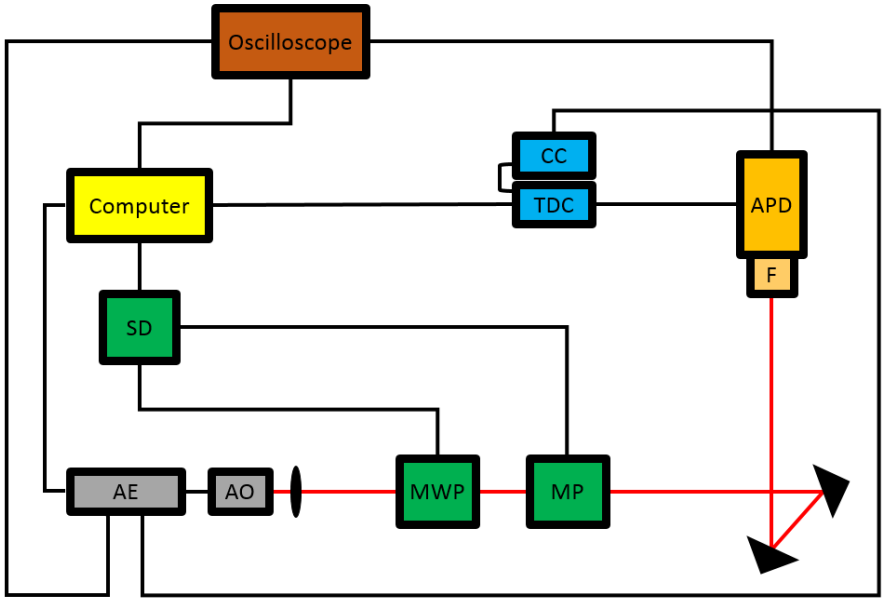


Figure 5.13: The measurement setup for the tomography and the temporal pulse shape measurement. A large part of this setup is identical to the temporal pulse shape measurement setup (figure 5.8), only three components have been added. A step drive motor controller (SD) which is connected to a computer and to two motorized components. These components are a motorized quarter-wave plate (MWP) and a motorized polarizer (MP). Both components can be directly controlled via the computer, via the *cutecom* program.

This measurement setup allows a tomography, as well as a temporal pulse shape measurement. To achieve the later only the oscilloscope needs to be added. All other components are required for the tomography. A tomography is the measurement of the polarization vector, which can be expressed as a Bloch vector $\vec{B} \in \mathbb{R}^3$,

instead of some complex vector $|b\rangle \in \mathbb{C}^2$. A measurement of the Bloch vector allows the reconstruction of the polarization state and the determination of the degree of polarization.

The key components for the tomography are the motorized quarter-wave plate and the motorized polarizer. The polarizer is a nanoparticle linear film polarizer and works as explained in section 2.3.4.1. A quarter-wave plate turns linearly polarized light into left/right circular polarized light, if the angle between the axes of the polarizer and the axes of the quarter-wave plate is 45°. If the relative angle is 0° nothing happens and for all other angles elliptically polarized light is created.

For the measurement of the Bloch vector (or polarization state) we need to measure the projections onto the six polarization states $|H\rangle, |V\rangle, |P\rangle, |M\rangle, |L\rangle$ and $|R\rangle$ (table 2.2). These projections can be achieved with appropriate orientations of the polarizer and the quarter-wave plate:

projection	MWP orientation	MP orientation
$\|H\rangle$	0°	0°
$\|V\rangle$	0°	90°
$\|R\rangle$	0°	45°
$\|L\rangle$	0°	−45°
$\|P\rangle$	−45°	−45°
$\|M\rangle$	−45°	45°

Table 5.6: The different orientations of the motorized quarter-wave plate (MWP) and the motorized polarizer (MP) to ensure a projection onto these six polarization states.

Before a tomography is performed the complete setup is tested. This can be done by preparing a specific polarization state, like $|H\rangle$, and performing a tomography on this known state. To ensure a proper working of the tomography it is necessary to find the correct step-motor position of the motorized components to realize the orientations of table 5.6. Due to this all used components were calibrated before the test of the setup. The step-motor positions were experimentally determined, in a measurement of the polarization axes of the MP, as well as a measurement of the the fast or slow axes of the MWP, by using a (vertical) reference polarizer.

The correct orientations of the motorized components are programmed into a python script, which adjusts the positions of the MWP and the MP, while it records also the measured events via the APD. The APD basically measure the intensity for the six polarization projections, which in turn allows a reconstruction of the

polarization state or Bloch vector $\vec{B} \in \mathbb{R}^3$, whose components are calculated as

$$\vec{B} = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = \begin{pmatrix} \frac{I_H - I_V}{I_H + I_V} \\ \frac{I_P - I_M}{I_P + I_M} \\ \frac{I_R - I_L}{I_R + I_L} \end{pmatrix}$$

The six polarization projection states correspond to the following Bloch vectors:

$$|H\rangle \leftrightarrow \vec{B}_H = \begin{pmatrix} +1 \\ 0 \\ 0 \end{pmatrix} \qquad |V\rangle \leftrightarrow \vec{B}_V = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} \qquad |P\rangle \leftrightarrow \vec{B}_H = \begin{pmatrix} 0 \\ +1 \\ 0 \end{pmatrix}$$
$$|M\rangle \leftrightarrow \vec{B}_M = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} \qquad |R\rangle \leftrightarrow \vec{B}_R = \begin{pmatrix} 0 \\ 0 \\ +1 \end{pmatrix} \qquad |L\rangle \leftrightarrow \vec{B}_L = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$$

If the intensity during the measurement is constant one should find $I_H + I_V = I_P + I_M = I_R + I_L$. Given $\vec{B}$ the degree of polarization (DOP) corresponds to the length of the Bloch vector and is thus calculated as

$$DOP = \sqrt{{v_x}^2 + {v_y}^2 + {v_z}^2}.$$

A $DOP = 1$ corresponds to completely polarized light, meaning that every photon has the same polarization. If the $DOP = 0$ the light is unpolarized, meaning that the polarization of each photon is random and there is no pattern or preference with respect to some polarization. A DOP between 0 and 1 corresponds to partially polarized light, which means that some amount of the photons exhibits some preferred polarization, but not all photons share that preference.

At first a wide parameter sweep was done, during which many temporal pulse shape measurements were performed. Then it was decided to look at three pulse parameter sets, which give rise to short pulses (with bias b=1) and to medium and long pulses (with bias=2). These three types of pulses are generated with the following parameters:

parameter	description	values
-c	channel	0, 3
-b	bias	1
-m	modulation	255
-da	delay line a	100
-db	delay line b	403
-bb	beacon bias	0
-bm	beacon modulation	0

Table 5.7: The alice-control parameters for the tomography for short pulses.

parameter	description	values
-c	channel	0, 3
-b	bias	2
-m	modulation	255
-da	delay line a	100
-db	delay line b	600, 900
-bb	beacon bias	0
-bm	beacon modulation	0

Table 5.8: The alice-control parameters for the tomography for medium and long pulses.

Additionally a tomography was also performed for the continuous wave emission of channel 0 and channel 3 with a bias $b = 17$.

Note that the measurement of the polarization state, as well as the measurement of the temporal pulse shape, are only performed for channel 0 and channel 3. This should be sufficient, since in the past channel 0,1 and 2 always behaved quite similarly. Channel 3 often stood out, which is why these two channels of the four available once have been chosen for this measurement. Note that the temporal pulse shape of all four channels have been measured previously (see section 5.4).

It was necessary to increase the bias for the medium and long pulses, because otherwise too few photons were measured for channel 0. The increased bias improved the counts per second (denoted as *count* in the plots) by two orders of magnitude.

5.5.2 Measurement results and discussion

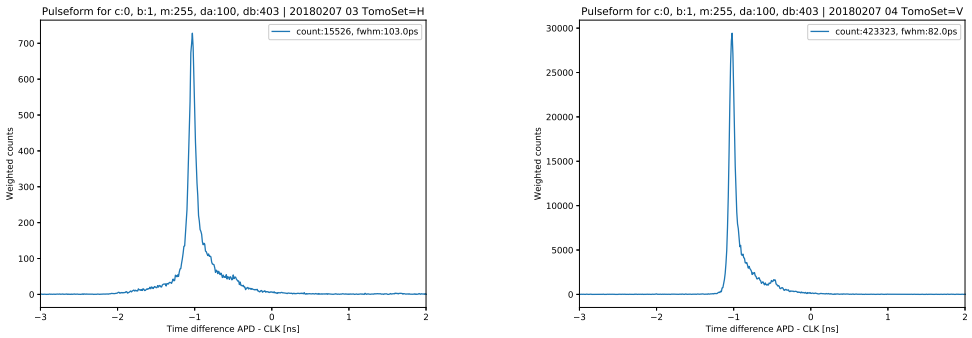


Figure 5.14: The measured temporal pulse shape for channel 0 of the short pulse with bias $b = 1$. The left picture shows the temporal pulse shape with the projection onto the horizontal polarization state. The right picture shows the temporal pulse shape with the projection onto the vertical polarization state.

For channel 0 with the short pulse parameters the following Bloch vector and DOP was measured:

$$\vec{B}_{0,s} = \begin{pmatrix} -0.93210 \pm 0.00034 \\ -0.0476 \pm 0.0009 \\ -0.0032 \pm 0.0009 \end{pmatrix} \quad , \quad DOP_{0,s} = 0.93243 \pm 0.00034$$

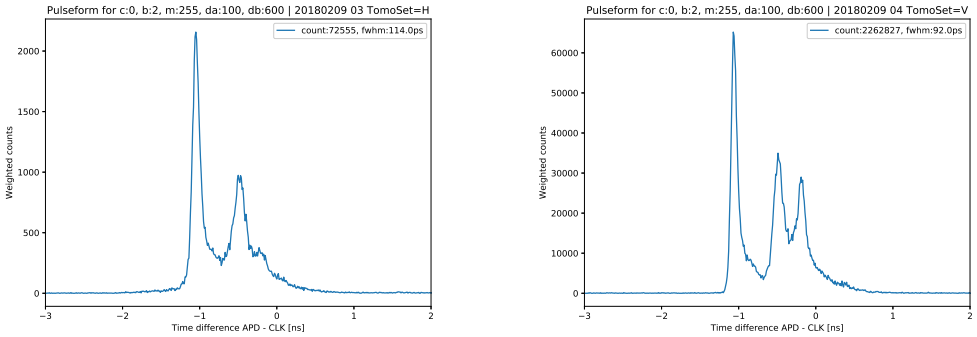


Figure 5.15: The measured temporal pulse shape for channel 0 with the medium pulse length and with bias $b = 2$. The left picture shows the temporal pulse shape with the projection onto the horizontal polarization state. The right picture shows the temporal pulse shape with the projection onto the vertical polarization state.

For channel 0 with the medium pulse parameters the following Bloch vector and DOP was measured:

$$\vec{B}_{0,m} = \begin{pmatrix} -0.93642 \pm 0.00015 \\ -0.0499 \pm 0.0004 \\ -0.0065 \pm 0.0004 \end{pmatrix} \quad , \quad DOP_{0,m} = 0.93777 \pm 0.00015$$

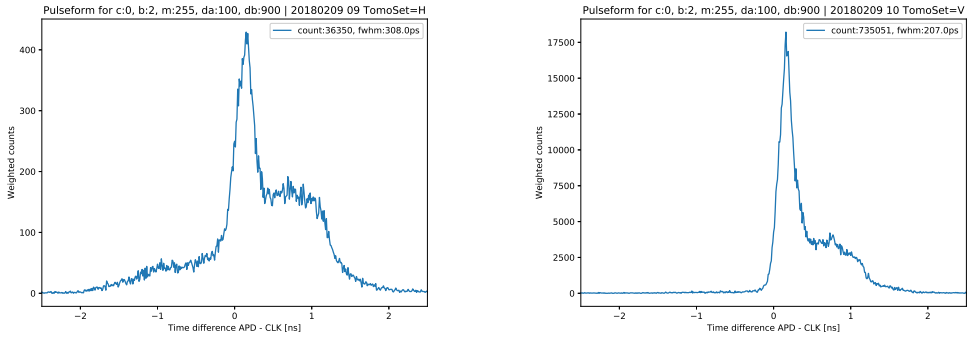


Figure 5.16: The measured temporal pulse shape for channel 0 with the long pulse length and with bias $b = 2$. The left picture shows the temporal pulse shape with the projection onto the horizontal polarization state. The right picture shows the temporal pulse shape with the projection onto the vertical polarization state.

For channel 0 with the long pulse parameters the following Bloch vector and DOP was measured:

$$\vec{B}_{0,l} = \begin{pmatrix} -0.90409 \pm 0.00031 \\ -0.0474 \pm 0.0007 \\ -0.0128 \pm 0.0007 \end{pmatrix} \quad , \quad DOP_{0,l} = 0.90542 \pm 0.00032$$

For channel 0 with the continuous wave parameters the following Bloch vector and DOP was measured:

$$\vec{B}_{0,cw} = \begin{pmatrix} -0.98833 \pm 0.00022 \\ -0.0700 \pm 0.0015 \\ 0.0270 \pm 0.0015 \end{pmatrix} \quad , \quad DOP_{0,cw} = 0.99117 \pm 0.00025$$

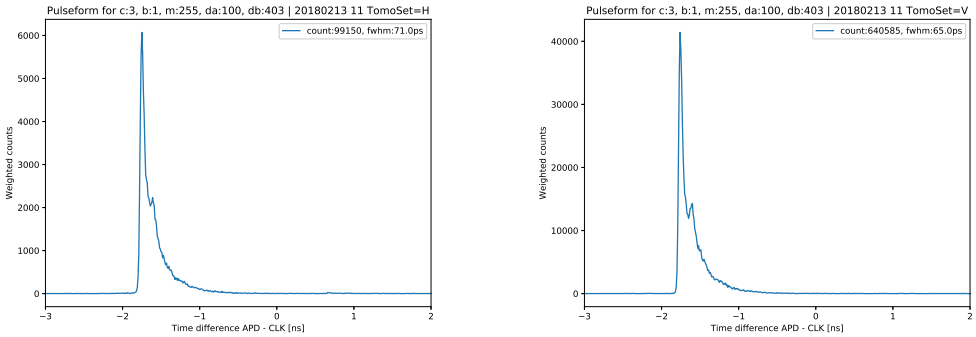


Figure 5.17: The measured temporal pulse shape for channel 3 with the short pulse with bias $b = 1$. The left picture shows the temporal pulse shape with the projection onto the horizontal polarization state. The right picture shows the temporal pulse shape with the projection onto the vertical polarization state.

For channel 3 with the short pulse parameters the following Bloch vector and DOP was measured:

$$\vec{B}_{3,s} = \begin{pmatrix} -0.7263 \pm 0.00005 \\ -0.0754 \pm 0.0008 \\ -0.0304 \pm 0.0008 \end{pmatrix} \quad , \quad DOP_{3,s} = 0.7309 \pm 0.0005$$

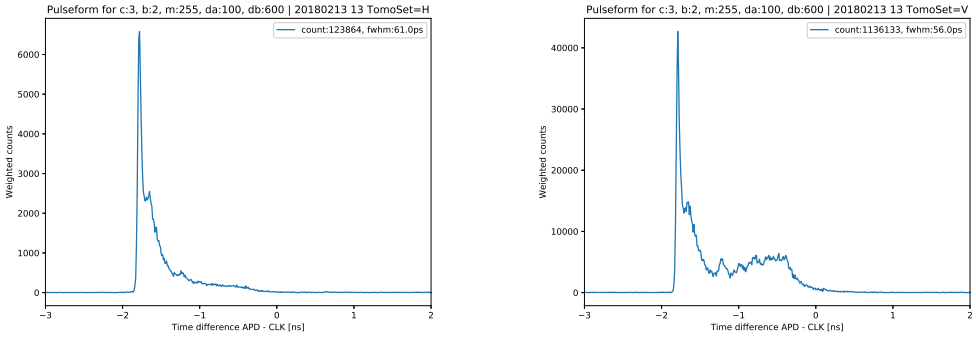


Figure 5.18: The measured temporal pulse shape for channel 3 with the medium pulse with bias $b = 2$. The left picture shows the temporal pulse shape with the projection onto the horizontal polarization state. The right picture shows the temporal pulse shape with the projection onto the vertical polarization state.

For channel 3 with the medium pulse parameters the following Bloch vector and

DOP was measured:

$$\vec{B}_{3,m} = \begin{pmatrix} -0.80716 \pm 0.00033 \\ -0.0768 \pm 0.0006 \\ -0.0342 \pm 0.0006 \end{pmatrix} \quad , \quad DOP_{3,m} = 0.81154 \pm 0.00033$$

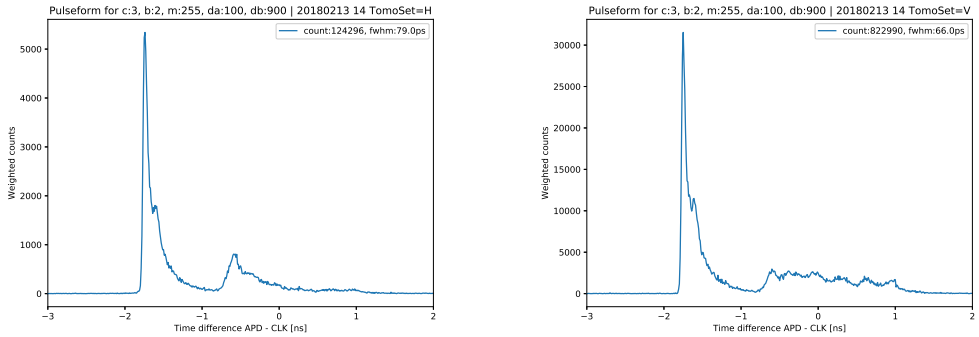


Figure 5.19: The measured temporal pulse shape for channel 3 with the long pulse with bias $b = 2$. The left picture shows the temporal pulse shape with the projection onto the horizontal polarization state. The right picture shows the temporal pulse shape with the projection onto the vertical polarization state.

For channel 3 with the long pulse parameters the following Bloch vector and DOP was measured:

$$\vec{B}_{3,l} = \begin{pmatrix} -0.7336 \pm 0.0004 \\ -0.0630 \pm 0.0007 \\ -0.0364 \pm 0.0007 \end{pmatrix} \quad , \quad DOP_{3,l} = 0.7372 \pm 0.0004$$

For channel 3 with the continuous wave parameters the following Bloch vector was measured:

$$\vec{B}_{3,cw} = \begin{pmatrix} 0.98145 \pm 0.00008 \\ 0.1253 \pm 0.0004 \\ 0.0353 \pm 0.0004 \end{pmatrix} \quad , \quad DOP_{3,cw} = 0.99005 \pm 0.00009$$

Additional plots can be found in the appendix section 8.5. Here temporal pulse shape plots for projection onto the polarization $|P\rangle, |M\rangle, |R\rangle$, and $|L\rangle$ are shown. The shapes for these state are all fairly similar, so there is no big difference like for the $|H\rangle$ and $|V\rangle$ polarization states. This due to the fact the the emitted polarization states are close to $|V\rangle$.

The following tables allow a quick comparison of the results for channel 0 and channel 3. Here $FHWM_{av}$ denotes the average for all six projections onto $|H\rangle$, $|V\rangle$, $|P\rangle$, $|M\rangle$, $|R\rangle$, and $|L\rangle$ (the additional plots can be found in the appendix section 8.5). The counts per second are summed for the projection onto $|H\rangle$ and $|V\rangle$ and are denoted as c_{H+V}.

short pulse	channel 0	channel 3
Bloch vector	$\begin{pmatrix} -0.93210 \\ -0.0476 \\ -0.0032 \end{pmatrix}$	$\begin{pmatrix} -0.72635 \\ -0.0754 \\ -0.0304 \end{pmatrix}$
DOP	0.93243	0.7309
$FWHM_{av}$	92.5 ps	67.2 ps
c_{H+V}	438849	739735

Table 5.9: Comparison of the two channels for short pulse parameters.

medium pulse	channel 0	channel 3
Bloch vector	$\begin{pmatrix} -0.93642 \\ -0.0499 \\ -0.0065 \end{pmatrix}$	$\begin{pmatrix} -0.80716 \\ -0.0768 \\ -0.0342 \end{pmatrix}$
DOP	0.93777	0.81154
$FWHM_{av}$	93.8 ps	58.3 ps
c_{H+V}	2335382	1259997

Table 5.10: Comparison of the two channels for medium pulse parameters.

long pulse	channel 0	channel 3
Bloch vector	$\begin{pmatrix} -0.90409 \\ -0.0474 \\ -0.0128 \end{pmatrix}$	$\begin{pmatrix} -0.7336 \\ -0.0630 \\ -0.0364 \end{pmatrix}$
DOP	0.90542	0.7372
$FWHM_{av}$	236.8 ps	70.0 ps
c_{H+V}	771401	947286

Table 5.11: Comparison of the two channels for long pulse parameters.

continuous wave	channel 0	channel 3
Bloch vector	$\begin{pmatrix} -0.98833 \\ 0.0700 \\ 0.0270 \end{pmatrix}$	$\begin{pmatrix} 0.98145 \\ 0.1253 \\ 0.0353 \end{pmatrix}$
DOP	0.99117	0.99005

Table 5.12: Comparison of the two channels for continuous wave parameters.

The average DOP for channel 0 is 0.92, if all measurements with vastly different pulse parameters are considered (all but the continuous wave parameters). For channel 3 the average DOP is 0.76, if all measurements with vastly different pulse parameters are considered (all but the continuous wave parameters). Both channels show thus a high degree of polarization, for short, medium and long pulses. This is a vastly different behavior with respect the old VCSEL array. The old array had a low DOP for short pulses (around 0.30), while only long pulses resulted in a higher DOP [31]. The emitted state comes close to the vertical polarization state, except for the continuous wave operation of channel 3, which comes close to the horizontal polarization state, but this changes with different biases. However, since short pulses are of interest the inclination towards the vertical polarization state needs to be considered. Additionally with this high[1] degree of polarization the preparation of the BB84 states needs to take these two facts into account.

For these sets of parameters the pulses from channel 3 are shorter or have a lower FWHM than the pulses from channel 0. However, for other sets of pulse parame-

[1]The degree of polarization of these VCSELs (RayCan) is high compared to the previously used VCSELs from VI Systems [31].

ters it is possible to get similar pulse widths.

Since channel 3 already turned out to be brighter, for the same bias current in section 5.1, it is not surprising to see that channel 3 also typically gets higher counts per second (denoted by c_{H+V}). The brightness should only be adjusted after the decision which VCSEL will be used to prepare which BB84 state, since this will alter the brightness. If the VCSEL of channel 0 or 3 are used to prepare the $|P\rangle$ or $|M\rangle$ via a wire-grid polarizer the intensity will be reduced by a factor of 2, for example.

Before the complete Alice optics module gets assembled a full tomography on all four VCSELs should be performed. Then it is possible to chose the optimal distribution of BB84 states onto the four VCSELs.

5.6 Electrical Pulses

Previous measurements have shown that channel 3 acts very differently than the other channels. The origin of this difference may lie in a different electronic response for the same pulse parameters. This may come from some minor defect in a component, the electrical wiring of the PCB or in one of the connections of a component with the wiring. To analyze this the electrical pulses need to be observed. If the electrical pulses are identical for channel 3 and 2 then the optical response should be the same, unless the origin for their difference lies within the VCSELs themselves.

5.6.1 Measurement setup

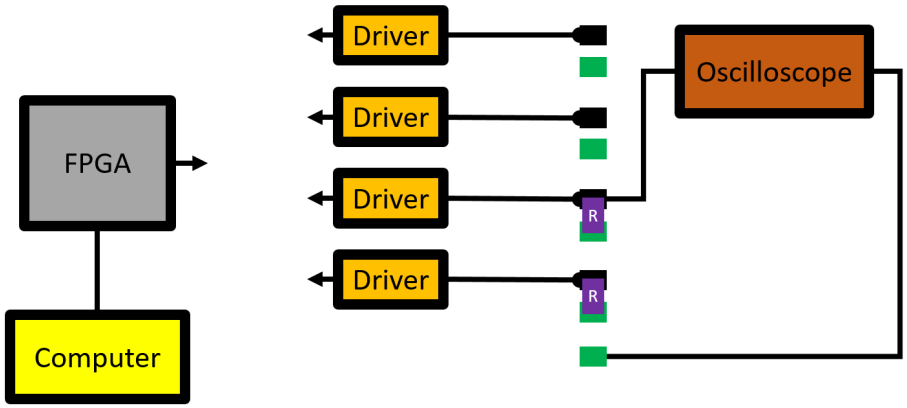


Figure 5.20: The measurement setup for the electronic pulse measurement. A computer is connected via USB to the Alice module. From the Alice module only the FPGA and the driver are displayed, the other components are omitted. The common ground is displayed in green. All those ground pads are connected via some layer in the PCB of the electronics module. The connection to the VCSEL carrier board is replaced by a resistor. This resistor mimics the VCSEL and allows a measurement of the electrical pulse shape via the oscilloscope. The oscilloscope is connected to one of the common grounds of the Alice module. The other part of the probe then measures the electrical pulse at the resistor. For this measurement the channels two and three were chosen as a representative sample.

The electrical pulses can not be measured while the VCSELs are still connected, because the current or voltage of the probe may severely damage them. Due to this the VCSELs are disconnected from the electronics module (or their connector

is) and instead a resistor is used to replace them. The resistance is chosen similar to the resistance of the VCSEL. The old VCSELs from VI Systems have a resistance of 50 Ω, while the new RayCan VCSELs have a resistance of 300 Ω. These values are taken from the VCSEL array data sheets, but there is no way to experimentally verify them. The RayCan VCSEL typically have a resistance of 300 Ω, but the maximum value they may have is 500 Ω, which means there is a big uncertainty in the actual resistance of the array. However, due to this two different resistances are used so that the behavior of pulses for these two different resistance can be used to deduce from them the behavior for even higher or lower resistances.

It is expected that the electrical pulse length is determined by the two delay values da, db. Given these parameters we should find that the electrical pulse length

$$t_{DD} = |da - db| * 5\ ps,$$

if the component behaves accordingly to its data sheet. For fixed $da = 100$ different values for db are picked such that the pulse of every successive run is twice as long as his predecessor.

parameter	description	values
-c	channel	2, 3
-b	bias	2
-m	modulation	255
-da	delay line a	100
-db	delay line b	200, 300, 500, 900
-bb	beacon bias	0
-bm	beacon modulation	0

Table 5.13: The alice-control parameters for the electronic pulse shape measurement.

The delta-delay is defined as

$$DD = |da - db|.$$

Due to this we find for the fixed $da = 100$ and for the db from table 5.13 $DD \in \{100, 200, 400, 800\}$. Which in turn results in the following predicted electrical pulse lengths

$$t_{100} = 0.50\ ns,$$
$$t_{200} = 1.00\ ns,$$
$$t_{400} = 2.00\ ns,$$
$$t_{800} = 4.00\ ns.$$

5.6.2 Measurement results and discussion

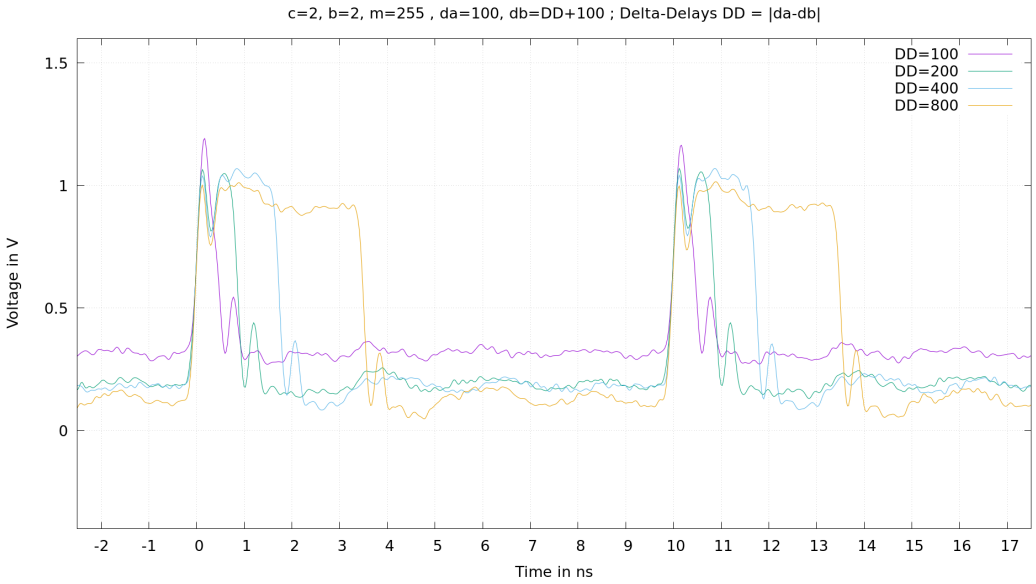


Figure 5.21: Electronic channel 2 with a 50 Ω resistor

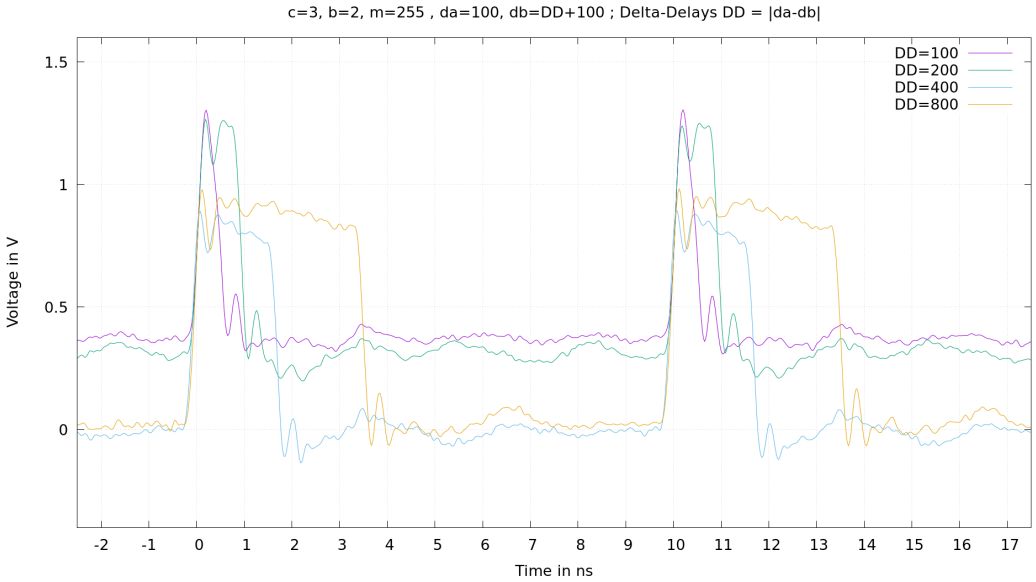


Figure 5.22: Electronic channel 3 with a 50 Ω resistor

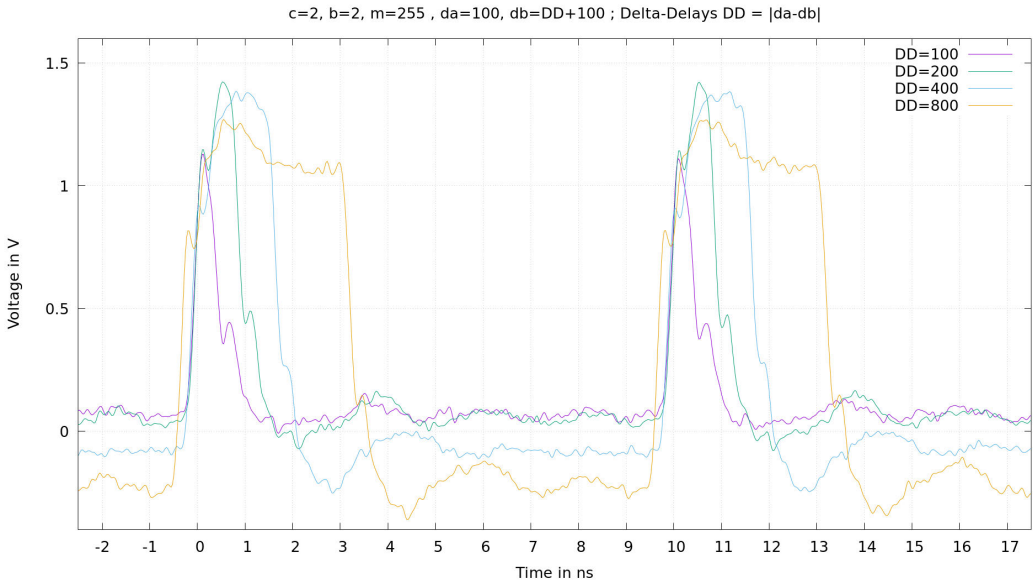


Figure 5.23: Electronic channel 2 with a 300 Ω resistor

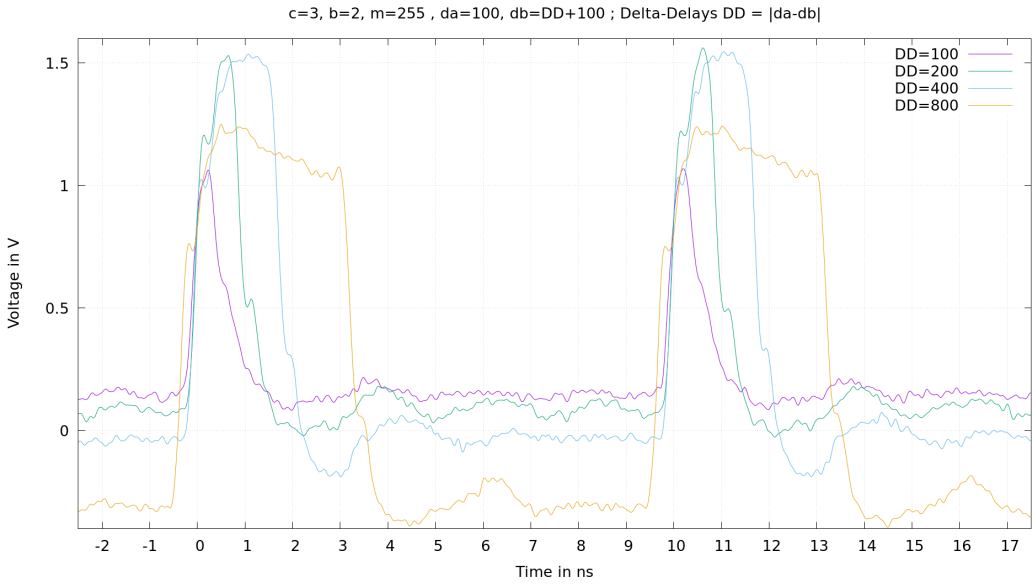


Figure 5.24: Electronic channel 3 with a 300 Ω resistor

Additional plots can be found in the appendix section 8.6. Here the pulse parameters from the tomography section were used, together with one additional short parameter set for bias $b = 2$.

Channel	2	2	2	2	2	2	2	2
R in Ω	50	50	50	50	300	300	300	300
DD	100	200	400	800	100	200	400	800
$FWHM_p$ in ns	0.5	1.0	2.0	4.0	0.5	1.0	2.0	4.0
$FWHM_m$ in ns	0.5	0.9	1.8	3.5	0.6	1.0	1.9	3.5

Table 5.14: Comparison of the predicted pulse width $FWHM_p$ and the measured pulse width $FWHM_m$ for channel 2.

Channel	3	3	3	3	3	3	3	3
R in Ω	50	50	50	50	300	300	300	300
DD	100	200	400	800	100	200	400	800
$FWHM_p$ in ns	0.5	1.0	2.0	4.0	0.5	1.0	2.0	4.0
$FWHM_m$ in ns	0.6	1.0	1.8	3.5	0.5	1.0	2.0	3.5

Table 5.15: Comparison of the predicted pulse width $FWHM_p$ and the measured pulse width $FWHM_m$ for channel 3.

The measured pulse widths $FWHM_m$ are read off figures 5.21, 5.22, 5.23 and 5.24.

The electrical pulses behave approximately as expected, at least for delay-delays $DD \in [100, 400]$, meaning that their electrical pulse width roughly obey the formula

$$t_{DD} = |da - db| * 5\ ps.$$

Longer pulses (around $DD = 800$) seem to deviate more from it.

While comparing the two responses for the two different resistors it becomes apparent that a higher resistance results in pulses which are closer to the ideal rectangular shape. The smaller resistance yields pulses with additional features besides the basic rectangular structure. These features are much smaller for the higher resistance and they become more notable for shorter pulses, especially for the lower resistance.

The electrical pulses of the electronic channel 2 and 3 look (qualitatively) very similar. There seems to be no discernible difference between those two electronic channels, or striking features that are unique to a specific channel.

Due to this the difference in their temporal pulse shapes for identical parameters can not be attributed to different electronic channels.

The VCSELs themselves may behave differently than the resistors, since the temporal pulse shape of the optical pulses looks very different than the pulse shape of the electrical pulses. It is also possible that the different VCSELs within the array have different resistances, which in turn changes the electrical pulses they receive. The origin of this may lie in manufacturing differences or a different number of bonding attempts for each VCSEL. Each bonding attempt affects the VCSEL and it is possible that the introduced change depends on the number of attempts until a successful bond is created. A difference in their respective bonding quality may also be the reason, but this can not be easily determined.

Chapter 6

Electronics and FPGA control

This chapter deals with characterization of the electronics designed by Clemens Sonnleitner [33]. The purpose of this new electronics is to enable faster testing of new components, as well es new techniques for the short pulse generation. Specifically the mainboard (*Alice_test_mainboard_rev2*) is tested together with the configuration of the FPGA (*Spartan 3E*). The role of the FPGA is to send pulse control parameters it receives from the computer via alice-control to the delay chip and the VCSEL driver. Those signals are measured and compared to the required patterns.

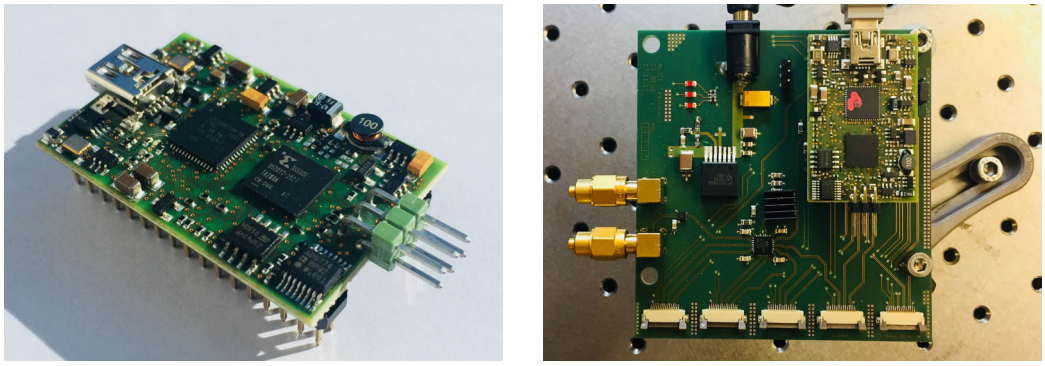


Figure 6.1: The left picture shows the FPGA on the EFM 01 evaluation board. The right picture shows the mainboard, with the FPGA mounted on it.

6.1 The FPGA

The FPGA used here is a *Spartan 3E* from *Xilinix* embedded in an *EFM 01* evaluation board from *Cesys*. This board (and thus the FPGA) can be connected via an

USB platform cable from Xilinx to a computer. Then the FPGA can be configured using the *ISE Design Suite* from *Xilinx*.
For this measurement a new FPGA was configured using a template from the existing program specifically written for the design of the hand-held Alice unit described in section 3.1. The program had to be adapted to fit the newly designed electronics mainboard. In particular, the old board used shared (SCLK) connections for some components, whereas here every component has an individual connection for each signal[1]. Due to this difference the original configuration can only be used to control two of the four channels (or VCSEL drivers) at the same time, which is why the it was adapted to use channel 0 and channel 3[2]. To use all four channels (at the same time) the configuration of the FPGA needs to be rewritten.

6.2 The delay chip

The delay chip (*Micrel SY89297U*) receives data serially via the four input signals SLOAD, SDATA, SCLK and EN (see figure 6.2). SDATA (20 bits) transmits the delay data *da*, *db*, if SCLK (clock signal) is high. This data is written into a register on the chip. SLOAD then commands to transfer the data to the two delay lines. A short high signal at SLOAD transfers the data, whereas a constantly low SLOAD signal leads to no such transfer and the transmitted delay parameters are not used. If the data is transfered then EN enables the delay with the parameters that were transfered from the first register to the specific delay line (a or b).
Note that the delay parameters for SDATA are written in reverse order, meaning that the least important bit is the left-most bit and the most important bit is the right-most bit.
All of this will be shown in detail in section 6.5.

[1] In the previous design a SCLK connection is shared between the two delay chips of channel 0 and 1, as well as another shared SCLK connection between the two VCSEL drivers. Channel 2 and 3 are similarly connected.

[2] Note that it is possible to use either channel 0 or channel 1 together with channel 2 or channel 3.

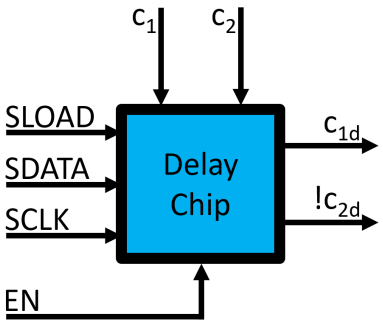


Figure 6.2: A schematic of the delay chip together with important input and output signals from the FPGA.

6.3 The VCSEL driver

The VCSEL driver (*ONET4291VA*) receives serially data via the two input signals SDA and SCLK. SCLK is a clock signal, while the parameters are transfered via the SDA signal.
The driver has an 11 Bit shift register, consisting of 8 bits of data and 3 bits to address one of three possible registers.

3 Bit address	register
000	control functions
001	modulation current
010	bias current

Table 6.1: The three VCSEL driver register addresses and their purpose.

The control function register only uses 6 of the available 8 bits.

Bit	register	purpose
0	-	unused
1	-	unused
2	MODR	Sets the modulation current range
3	FLTEN	Enables and disables the fault detection
4	OLE	Enables and disables the open loop
5	PDR	Sets the photodiode current range
6	PDP	Sets the photodiode polarity
7	ENA	Enables and disables the laser driver chip

Table 6.2: The control function bits of the control function register.

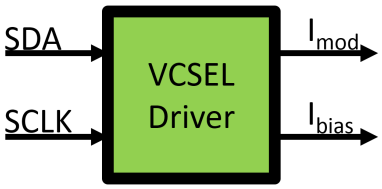


Figure 6.3: A schematic of the VCSEL driver together with important input and output signals from the FPGA. The signals correspond to parameters set via alice-control.

The modulation current and bias current register are fairly similar. Both receive 8 bit numbers, which directly correspond to a set bias b and modulation m. Since 8 bits are used both parameters range between 0 and 255.

All of the values for the control, modulation and bias register are transmitted via the SDATA signal, if the SCLK signal is high as well. This will be shown in detail in section 6.5.

6.4 Measurement of the FPGA signal

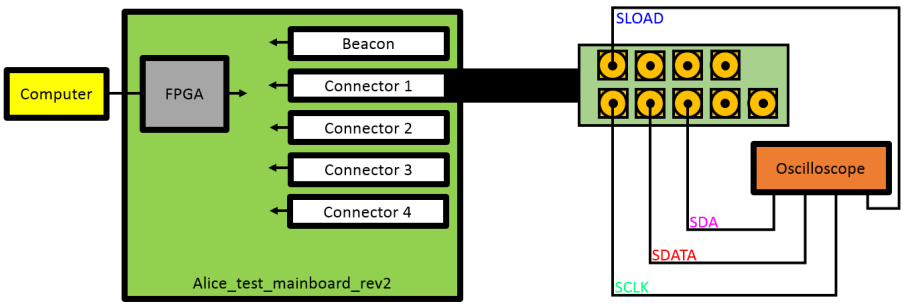


Figure 6.4: Measurement setup for the analysis of FPGA signals. The FPGA is controlled via a computer, which is connected to the FPGA board via USB. At the mainboard, the FPGA is connected to a clock and a clock buffer. All other components, like delay chips, logic gates and the VCSEL drivers are placed on other subboards, which get connected to the mainboard via the four connectors and flexible flat ribbon cables. All signals from the FPGA reach a connector line, for the beacon or for one of the four VCSELs. The *Alice_test_mainboard_rev2* here is connected to the *empty* board containing only SMA connectors for the the oscilloscope.

The measurement setup is depicted in figure 6.4. Here different parameters are sent to the FPGA, via the computer program alice-control, and the signal from the FPGA is then transmitted to an additional board via the flat ribbon cable. There an oscilloscope is connected to display the signal of channel 0, which corresponds to connector 1.
For this measurement, several pulse parameters were sent via the computer to the FPGA. The FPGA then sends corresponding signals to the delay chip and the VCSEL driver. These signals are then captured via the oscilloscope and evaluated.

parameter	description	values
-c	channel	0
-b	bias	42
-m	modulation	67
-da	delay line a	222
-db	delay line b	555
-bb	beacon bias	0
-bm	beacon modulation	0

Table 6.3: The alice-control parameters for the FPGA signal measurement.

6.5 Measurement results

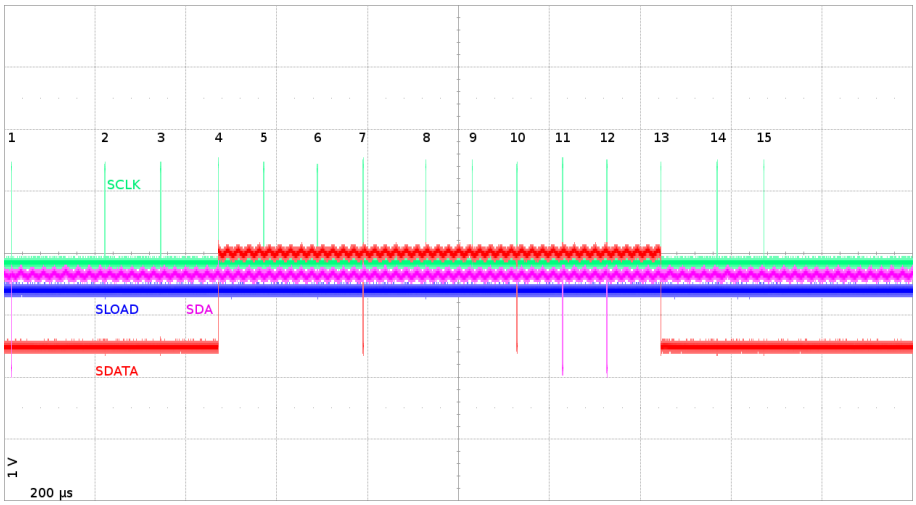


Figure 6.5: The four signals from the FPGA for programming the delay chip and the VCSEL driver measured with an oscilloscope. The four signal have 15 time slots were parameters are set, via a high SCLK signal.

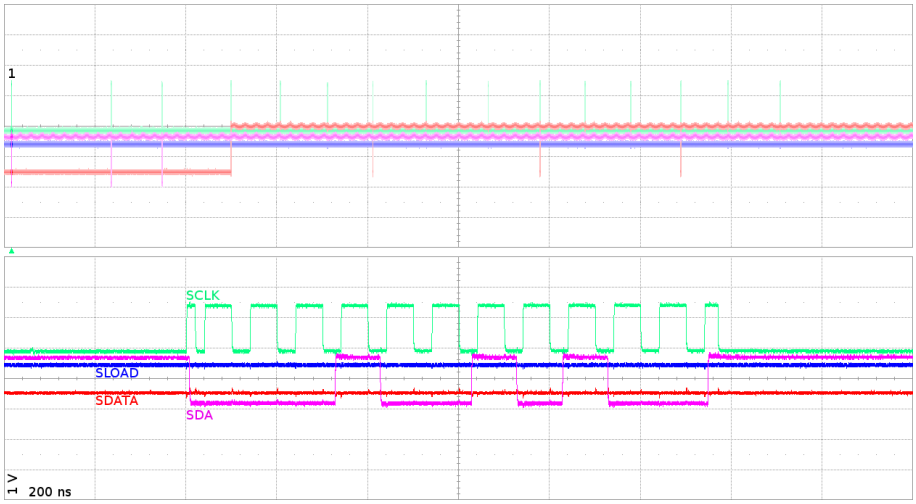


Figure 6.6: Time slot 1 determines the VCSEL driver control functions for all channels.

Reconstructing the binary code for the first time slot we find

000 10010100

Since this is an 11 bit signal we see that this is a valid signal for the VCSEL driver.

The first three bits (counting from left to right) determine that the following values are for the control functions. Using table 6.2 we can deduce the meaning of the remaining 8 bits, of which only 6 bits are used. The forth bit enables the chip. The fifth bit sets the photodiode polarity bit as 0, which corresponds to the operation as a common cathode. The sixth bit determines the photodiode current range to be between 0 μA and 250 μA with a resolution of 1 μA. The seventh bit enables the open loop configuration, which sets the bias to be

$$I_{bias} = 100\,\mu A + 47\,\mu A * b.$$

The eighth bit disables the fault detection. The ninth bit sets the modulation current range to be between 100 μA and 15400 μA with a step size of 68 μA (instead of the default 12000 μA with a step size of 51 μA). The last two bits are not used.

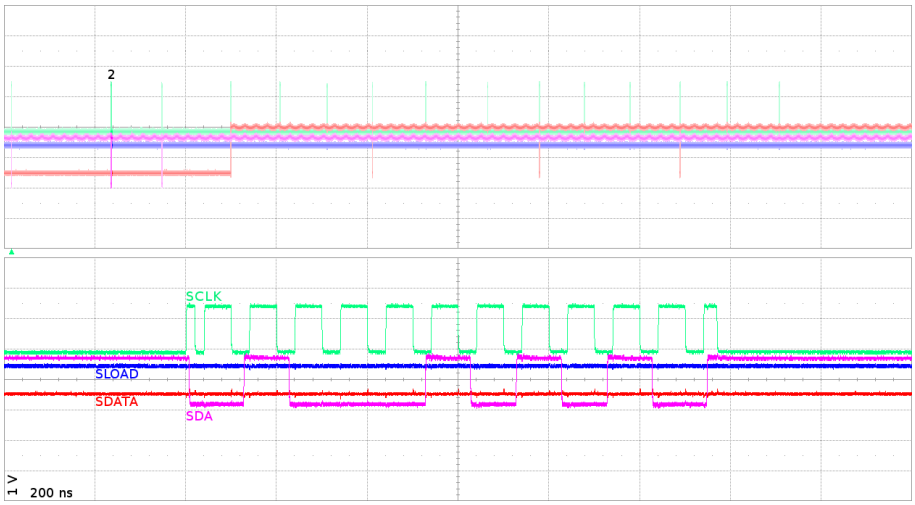


Figure 6.7: Time slot 2 determines the VCSEL driver bias current for channel 0.

For the second time slot we find

$$010\,00101010$$

The first three bits (counting from left to right) determine that the following values fix the bias current. The following eight bits determine the bias current value

$$00101010_2 = 42_{10}.$$

Thus we have found the bias value $b = 42$, which we set via alice-control for channel 0. Thus we conclude that position 2 determines the bias current for the VCSEL driver 0.

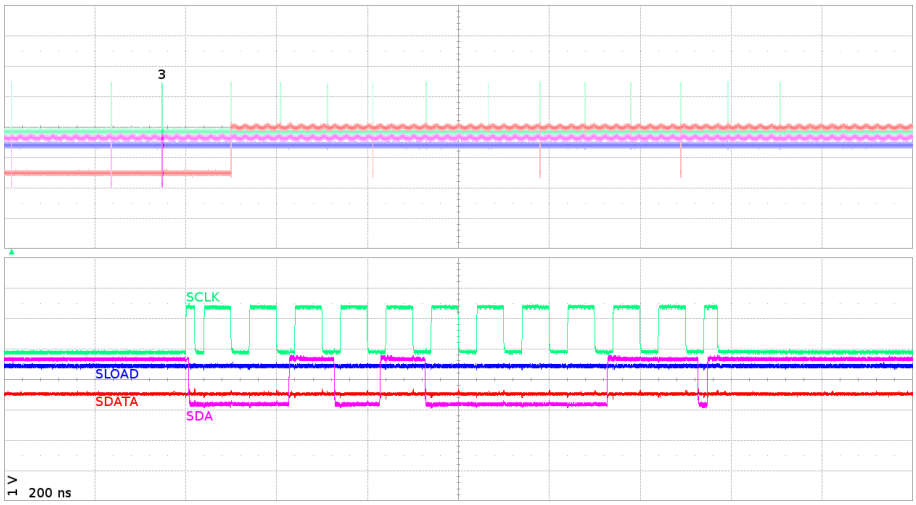


Figure 6.8: Time slot 3 determines the VCSEL driver modulation current for channel 0.

For the third time slot we find

$$001\ 0100011$$

The first three bits (counting from left to right) determine that the following values are for the modulation current. The following eight bits determine the bias current value

$$01000011_2 = 67_{10}.$$

Thus have found the modulation value $m = 67$, which we set via alice-control.

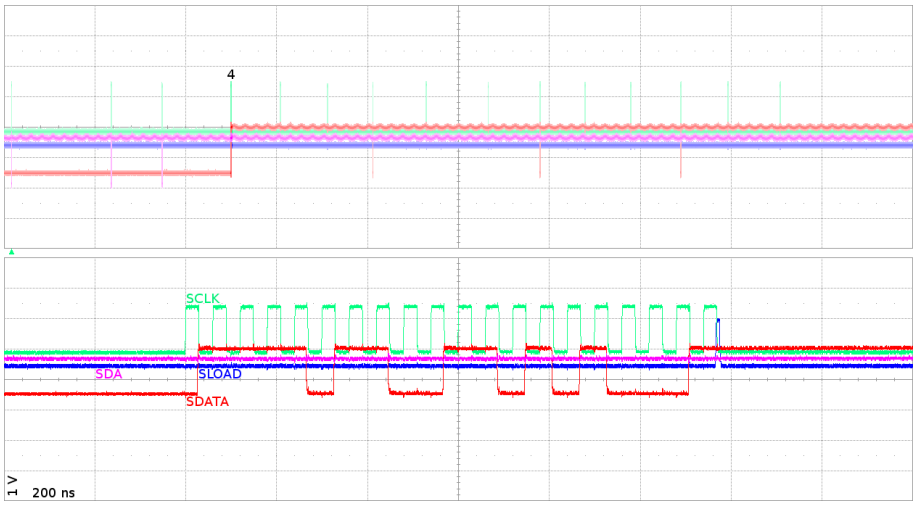


Figure 6.9: Time slot 4 determines the delay chip parameters for channel 0

Reconstructing the binary code for the forth time slot we find

$$0111101100\ 1101010001$$

Since this is a 20 bit signal we see that this is a valid signal for the delay chip.

These 20 bit should be the da, db values, but before they can be decoded we need to reverse their order

$$0111101100\ 1101010001 \rightarrow 1000101011\ 0011011110.$$

Now we can decode these two 10 bit values

$$1000101011_2\ 0011011110_2 = 222_{10}\ 555_{10}.$$

Thus we have found the delay values $da = 222$ and $db = 555$, which we set via alice-control. These values were written into the register and than forwarded to the two delay lines, because at the end of the two signals (SCLK and SDATA) a peak of SLOAD can be seen, which triggers the transfers of the delay values from the register to the two respective lines. Without this peak of SLOAD the delay data would not be used.

Similar data was collected for channel 3 with similar results. Here time slot 11 encodes the bias current and time slot 12 sets the modulation current for VCSEL driver 3. Time slot 13 programs the delay parameters for delay chip 3.
Additionally many other alice-control parameter sets were used and measured, all of which agree with the presented data and explanation.

6.6 Conclusion

The new mainboard (*Alice_test_mainboard_rev2*) functions properly, as well as the existing FPGA configuration. The FPGA sends the correct signals to the delay chip and the VCSEL driver. As a result of the measured and decoded data the structure summarized in table 6.4 behind the time slots has been found.

Time slot	Purpose
01	VCSEL driver control functions
02	VCSEL driver 0 bias current
03	VCSEL driver 0 modulation current
04	Delay chip 0 delay parameters
11	VCSEL driver 3 bias current
12	VCSEL driver 3 modulation current
13	Delay chip 3 delay parameters

Table 6.4: The FPGA signal time slot identification.

With the existing configuration two of the four channels can be used simultaneously, but to use all four channels at the same time the FPGA configurations needs to be adapted to the new electronics design.
The remaining time slots are used to program the delay chips and VCSEL driver belonging to channel one and two, as well as the VCSEL driver for the beacon. Due to the measured data it seems likely that time slots five and six set the bias and modulation current of channel one, while time slot seven sets the parameters for the delay chip. Time slot eight and nine contain the bias and modulation parameters of channel two, while time slot ten programs the delay values. This leaves the final two time slots, fourteen and fifteen for the beacon bias and modulation.

Chapter 7

Summary and outlook

This thesis describes the assembly of a VCSEL carrier board and the characterization of a new VCSEL array. Also, first steps were taken towards the verification of the new sender mainboard designed by Clemens Sonnleitner [33], as well as a test of the FPGA configuration.

The first major part of this master's thesis dealt with the assembly of a VCSEL carrier board. Here a new VCSEL array was placed on an old carrier board. Additionally a new carrier board was designed, with new connectors. The new board is also only half as big as the old one. A further decrease in size is only possible if other connectors are used. This small module was designed to allow a connection with the newly designed modular electronics, as well as a possible implementation in a satellite.

In the second part of this thesis the new VCSEL array was characterized with respect to its usability for QKD. Several tests were performed, all of which allowed for an employment in a QKD unit. The 4th VCSEL (or channel 3) of this array showed in several measurements extraordinary behavior, which was confirmed to be due to the VCSEL itself and not due to differences in its electronics channel. This has to be considered during the assembly of the complete optics module. The temporal pulse shape measurements showed that rather short pulses ($FWHM \approx 60\ ps$) can be created with these VCSELs and similar pulse shapes can also be found. The tomography showed that these VCSEL show a high degree of polarization ($DOP_{channel0} \approx 0.92$, $DOP_{channel3} \approx 0.76$) and their respective polarization state for short pulses comes close to the vertical polarization state. The only issue compromising the security is presented by the four different wavelengths, which makes this sender unit vulnerable to attacks, where this difference is exploited to distinguish the four BB84 states.
The extraordinary behavior of channel 3 may be due to the cutting of the VCSEL

array. This could be tested in the future by purchasing an array with six VCSELs, where only the inner four VCSELs are used for the preparation of the BB84 states. Alternatively an even bigger array could be bought in the hope that some of the four VCSELs, who lie next to each other, have a spectrum with big enough overlaps to allow a wavelength filtering to close the wavelength side-channel.

Finally the newly designed electronics mainboard and the FPGA configuration was tested. Both tests came out positive, meaning that the mainboard and the FPGA configuration both work well. For a complete utilization of the mainboard the configuration needs to be adapted further, to allow the use of the full potential of the new modular design. The reaming electronics module containing the delay chip, the logic gate and the VCSEL driver also need to be tested. Adding new electronic components will also require additional changes of the FPGA configuration.

Chapter 8

Appendix

8.1 The Bloch sphere

Given $\hat{\sigma}_x, \hat{\sigma}_y$ and $\hat{\sigma}_z$ we can construct an operator for some arbitrary direction (of spin). This new operator is defined via a sort of dot-product with a unit vector $\vec{n}$ which points along some direction in 3D space. Our new operator is defined as $\hat{\sigma}_{\theta,\varphi} = \vec{n} \cdot \hat{\vec{\sigma}}$ with:

$$\vec{n} = \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} = \begin{pmatrix} sin(\theta)cos(\varphi) \\ sin(\theta)sin(\varphi) \\ cos(\theta) \end{pmatrix}$$

This unit vector $\vec{n}$ determines a direction in space in terms of θ and φ.

$$\vec{n} \cdot \hat{\vec{\sigma}} = \begin{pmatrix} sin(\theta)cos(\varphi) \\ sin(\theta)sin(\varphi) \\ cos(\theta) \end{pmatrix} \cdot \begin{pmatrix} \hat{\sigma}_x \\ \hat{\sigma}_y \\ \hat{\sigma}_z \end{pmatrix} = sin(\theta)cos(\varphi)\hat{\sigma}_x + sin(\theta)sin(\varphi)\hat{\sigma}_y + cos(\theta)\hat{\sigma}_z$$

Thus we find our new operator:

$$\hat{\sigma}_{\theta,\varphi} = \begin{pmatrix} cos(\theta) & sin(\theta)e^{-i\varphi} \\ sin(\theta)e^{+i\varphi} & -cos(\theta) \end{pmatrix}$$

The eigenvaules are $+1$ and -1.
The eigenstates are (in the $\hat{\sigma}_z$ eigenstate-basis):

$$|\vec{n};+\rangle = cos\left(\frac{\theta}{2}\right)|0\rangle + sin\left(\frac{\theta}{2}\right)e^{+i\varphi}|1\rangle \text{ and}$$
$$|\vec{n};-\rangle = sin\left(\frac{\theta}{2}\right)|0\rangle + cos\left(\frac{\theta}{2}\right)e^{+i\varphi}|1\rangle$$

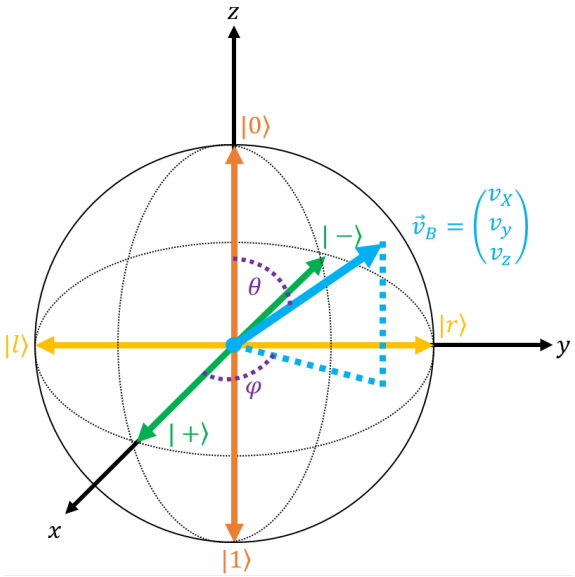


Figure 8.1: The bloch sphere depicts a complex qubit state in real 3D space.

We can picture these states on the Bloch sphere.
If we imagine we start out with a state $|\vec{n};+\rangle$ where $\theta = 0$ and $\varphi = 0$ we get $|u\rangle$. If we rotate θ of this state by 360° or 2π we get $|d\rangle$. We see that a regular full rotation does not equal a full rotation in Hilbert space (where the state vector lives)[1]. In order to rotate fully in Hilbert space, so that we end up with the original state, we have to rotate by 720° or 4π.

[1]Mathematically this additional factor of two that we need to rate fully comes from the fact that $SU(2)$ is a double cover of $O(3)$

8.2 The uncertainty of an observable

In order to understand the uncertainty principle it is useful to have an intuitive understanding of the uncertainty of an observable (for a given state).

The uncertainty Δ of an observable $\hat{A}$ for a state $|\Psi\rangle$ is defined as

$$\Delta\hat{A}(|\Psi\rangle) = |(\hat{A} - \langle\Psi|\hat{A}|\Psi\rangle\,\hat{1})\,|\Psi\rangle\,|, \text{ or equivalently}$$
$$\Delta\hat{A}(|\Psi\rangle) = \sqrt{\langle\Psi|\hat{A}^2|\Psi\rangle - \langle\Psi|\hat{A}|\Psi\rangle^2}.$$

Here the dependence of the uncertainty on the state $|\Psi\rangle$ is explicitly stressed, but for the rest of this thesis this dependence will be omitted and the uncertainty $\Delta\hat{A}(|\Psi\rangle)$ will be denoted as $\Delta\hat{A}$.
There is an intuitive picture that helps visualizing this otherwise abstract notion of the uncertainty of an observable for some state. First we construct the projector $\hat{P}$ onto the state $|\Psi\rangle$:

$$\hat{P} = |\Psi\rangle\langle\Psi|.$$

Then we act with this projector $\hat{P}$ onto the state $\hat{A}|\Psi\rangle$ and obtain the new state[2] $|\Psi_{||}\rangle$:

$$\begin{aligned}\hat{P}\hat{A}|\Psi\rangle &= |\Psi\rangle\langle\Psi|\hat{A}|\Psi\rangle \\ &= \langle\Psi|\hat{A}|\Psi\rangle\,|\Psi\rangle \\ &= |\Psi_{||}\rangle.\end{aligned}$$

Now we look at the difference of $\hat{A}|\Psi\rangle$ and $|\Psi_{||}\rangle$ which we denote as $|\Psi_\perp\rangle$:

$$\begin{aligned}\hat{A}|\Psi\rangle - |\Psi_{||}\rangle &= \hat{A}|\Psi\rangle - \langle\Psi|\hat{A}|\Psi\rangle\,|\Psi\rangle \\ &= (\hat{A} - \langle\Psi|\hat{A}|\Psi\rangle\,\hat{1})\,|\Psi\rangle \\ &= |\Psi_\perp\rangle\end{aligned}$$

Thus we see that $|\,|\Psi_\perp\rangle\,| = \Delta\hat{A}(|\Psi\rangle)$. There we see that the uncertainty $\Delta\hat{A}$ measures how much a state $|\Psi\rangle$ is changed if the observable $\hat{A}$ acts on it. This change is the addition of $|\Psi_\perp\rangle$ to the state $|\Psi\rangle$, whose absolute value or length corresponds to the uncertainty $\hat{A}$. In general $\Delta\hat{A}$ will be non-zero, unless $|\Psi\rangle$ is in an eigenstate of $\hat{A}$. Eigenstates only get rescaled by their respective eigenvalue, but

[2] This new state $|\Psi_{||}\rangle$ is not normalized, but this is not that important for the following argument, because here basically only the direction is of concern.

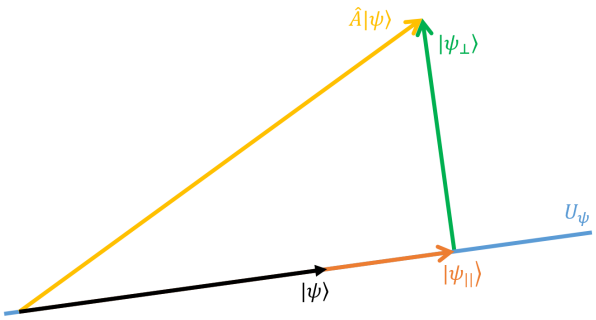


Figure 8.2: The operator $\hat{A}$ acts on the state $|\Psi\rangle$, which is then decomposed into a component parallel to its original state and into a component perpendicular to its original state. The length of the perpendicular component corresponds to the uncertainty. U_Ψ denotes the one-dimensional subspace spanned by the state $|\Psi\rangle$.

they do not change their orientation in the Hilbert space. Thus eigenstates are the only states that are not altered by a measurement. All other states change upon measurement due to the interaction of the measurement device with the system that is being measured. All of this is visualized in figure 8.2.

8.3 VCSEL Channel correspondence

The following figure shows which VCSEL on the array is addressed as which channel via alice-control.

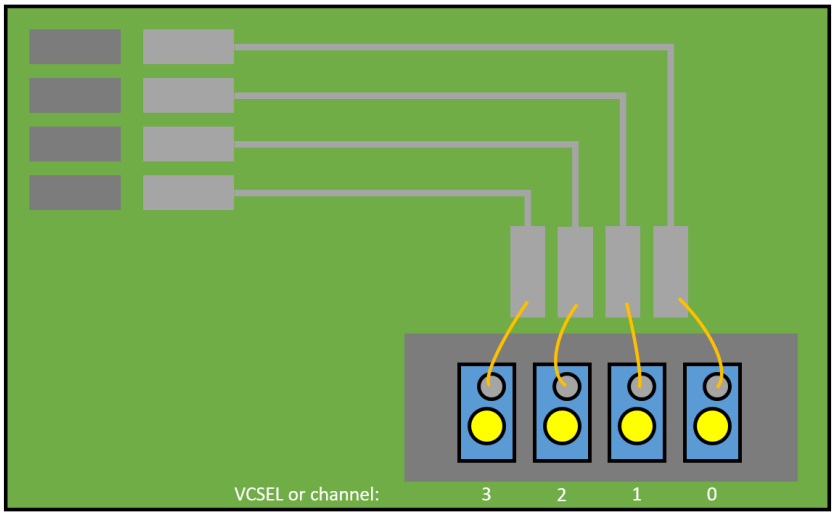


Figure 8.3: A schematic of the old PCB VCSEL carrier board that shows which VCSELs correspond to which channels.

8.4 Additional temporal pulse shape plots

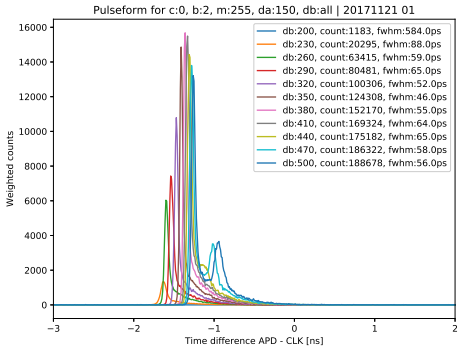
The alice-control parameters for each measurement are contained in the heading of each plot.

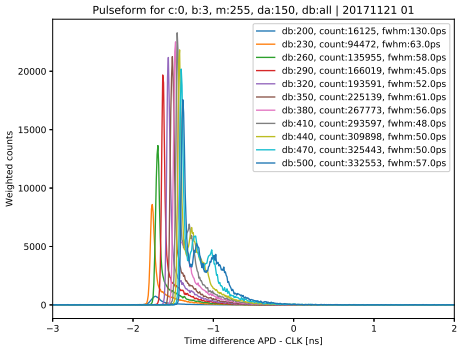
Channel 0 develops with increasing pulse length a second peak. If the bias is also increased from two to three an additional third peak appears for long pulse lengths.

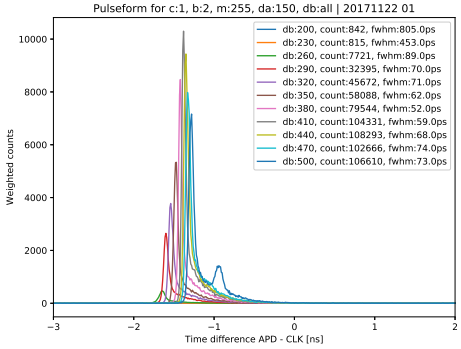
Channel 1 develops with increasing pulse lengths a second peak, but only for the last delay value. If the bias is also increased from two to three an additional third peak appears for long pulse lengths.

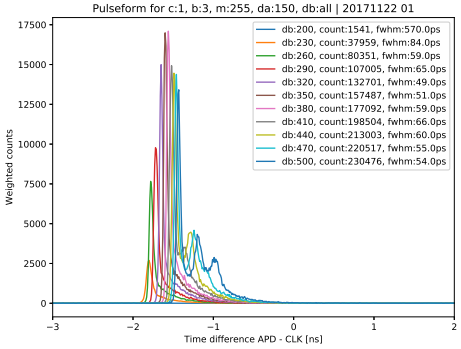
Channel 2 develops with increasing pulse length a second and a third peak. If the bias is also increased from two to three the additional peaks almost form a plateau.

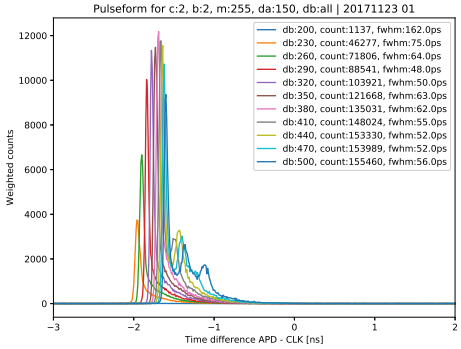
Channel 3 develops with increasing pulse length no additional peak, but gets broader. If the bias is also increased from two to three a plateau emerges similar to channel 2.

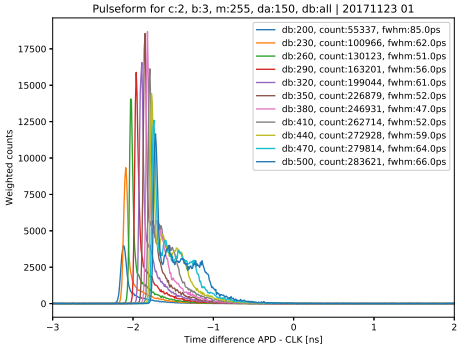

Pulseform for c:0, b:2, m:255, da:150, db:all | 20171121 01
db:200, count:1183, fwhm:584.0ps
db:230, count:20295, fwhm:88.0ps
db:260, count:63415, fwhm:59.0ps
db:290, count:80481, fwhm:65.0ps
db:320, count:100306, fwhm:52.0ps
db:350, count:124308, fwhm:46.0ps
db:380, count:152170, fwhm:55.0ps
db:410, count:169324, fwhm:64.0ps
db:440, count:175182, fwhm:65.0ps
db:470, count:186322, fwhm:58.0ps
db:500, count:188678, fwhm:56.0ps
Weighted counts
Time difference APD - CLK [ns]

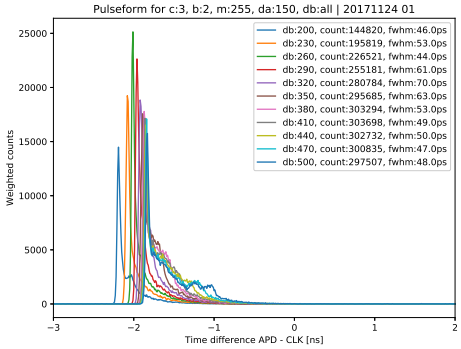

Pulseform for c:0, b:3, m:255, da:150, db:all | 20171121 01
db:200, count:16125, fwhm:130.0ps
db:230, count:94472, fwhm:63.0ps
db:260, count:135955, fwhm:58.0ps
db:290, count:166019, fwhm:45.0ps
db:320, count:193591, fwhm:52.0ps
db:350, count:225139, fwhm:61.0ps
db:380, count:267773, fwhm:56.0ps
db:410, count:293597, fwhm:48.0ps
db:440, count:309898, fwhm:50.0ps
db:470, count:325443, fwhm:50.0ps
db:500, count:332553, fwhm:57.0ps
Weighted counts
Time difference APD - CLK [ns]

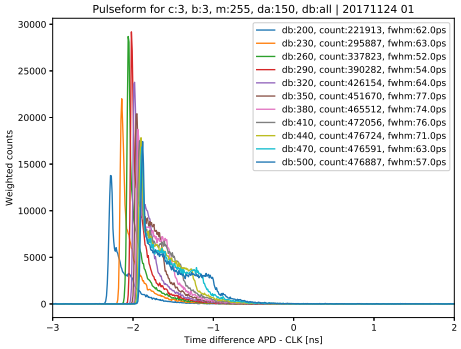

Pulseform for c:1, b:2, m:255, da:150, db:all | 20171122 01
db:200, count:842, fwhm:805.0ps
db:230, count:815, fwhm:453.0ps
db:260, count:7721, fwhm:89.0ps
db:290, count:32395, fwhm:70.0ps
db:320, count:45672, fwhm:71.0ps
db:350, count:58088, fwhm:62.0ps
db:380, count:79544, fwhm:52.0ps
db:410, count:104331, fwhm:59.0ps
db:440, count:108293, fwhm:68.0ps
db:470, count:102666, fwhm:74.0ps
db:500, count:106610, fwhm:73.0ps
Weighted counts
Time difference APD - CLK [ns]


Pulseform for c:1, b:3, m:255, da:150, db:all | 20171122 01
db:200, count:1541, fwhm:570.0ps
db:230, count:37959, fwhm:84.0ps
db:260, count:80351, fwhm:59.0ps
db:290, count:107005, fwhm:65.0ps
db:320, count:132701, fwhm:49.0ps
db:350, count:157487, fwhm:51.0ps
db:380, count:177092, fwhm:59.0ps
db:410, count:198504, fwhm:66.0ps
db:440, count:213003, fwhm:60.0ps
db:470, count:220517, fwhm:55.0ps
db:500, count:230476, fwhm:54.0ps
Weighted counts
Time difference APD - CLK [ns]


Pulseform for c:2, b:2, m:255, da:150, db:all | 20171123 01
db:200, count:1137, fwhm:162.0ps
db:230, count:46277, fwhm:75.0ps
db:260, count:71806, fwhm:64.0ps
db:290, count:88541, fwhm:48.0ps
db:320, count:103921, fwhm:50.0ps
db:350, count:121668, fwhm:63.0ps
db:380, count:135031, fwhm:62.0ps
db:410, count:148024, fwhm:55.0ps
db:440, count:153330, fwhm:52.0ps
db:470, count:153989, fwhm:52.0ps
db:500, count:155460, fwhm:56.0ps
Weighted counts
Time difference APD - CLK [ns]


Pulseform for c:2, b:3, m:255, da:150, db:all | 20171123 01
db:200, count:55337, fwhm:85.0ps
db:230, count:100966, fwhm:62.0ps
db:260, count:130123, fwhm:51.0ps
db:290, count:163201, fwhm:56.0ps
db:320, count:199044, fwhm:61.0ps
db:350, count:226879, fwhm:52.0ps
db:380, count:246931, fwhm:47.0ps
db:410, count:262714, fwhm:52.0ps
db:440, count:272928, fwhm:59.0ps
db:470, count:279814, fwhm:64.0ps
db:500, count:283621, fwhm:66.0ps
Weighted counts
Time difference APD - CLK [ns]


Pulseform for c:3, b:2, m:255, da:150, db:all | 20171124 01
db:200, count:144820, fwhm:46.0ps
db:230, count:195819, fwhm:53.0ps
db:260, count:226521, fwhm:44.0ps
db:290, count:255181, fwhm:61.0ps
db:320, count:280784, fwhm:70.0ps
db:350, count:295685, fwhm:63.0ps
db:380, count:303294, fwhm:53.0ps
db:410, count:303698, fwhm:49.0ps
db:440, count:302732, fwhm:50.0ps
db:470, count:300835, fwhm:47.0ps
db:500, count:297507, fwhm:48.0ps
Weighted counts
Time difference APD - CLK [ns]


Pulseform for c:3, b:3, m:255, da:150, db:all | 20171124 01
db:200, count:221913, fwhm:62.0ps
db:230, count:295887, fwhm:63.0ps
db:260, count:337823, fwhm:52.0ps
db:290, count:390282, fwhm:54.0ps
db:320, count:426154, fwhm:64.0ps
db:350, count:451670, fwhm:77.0ps
db:380, count:465512, fwhm:74.0ps
db:410, count:472056, fwhm:76.0ps
db:440, count:476724, fwhm:71.0ps
db:470, count:476591, fwhm:63.0ps
db:500, count:476887, fwhm:57.0ps
Weighted counts
Time difference APD - CLK [ns]

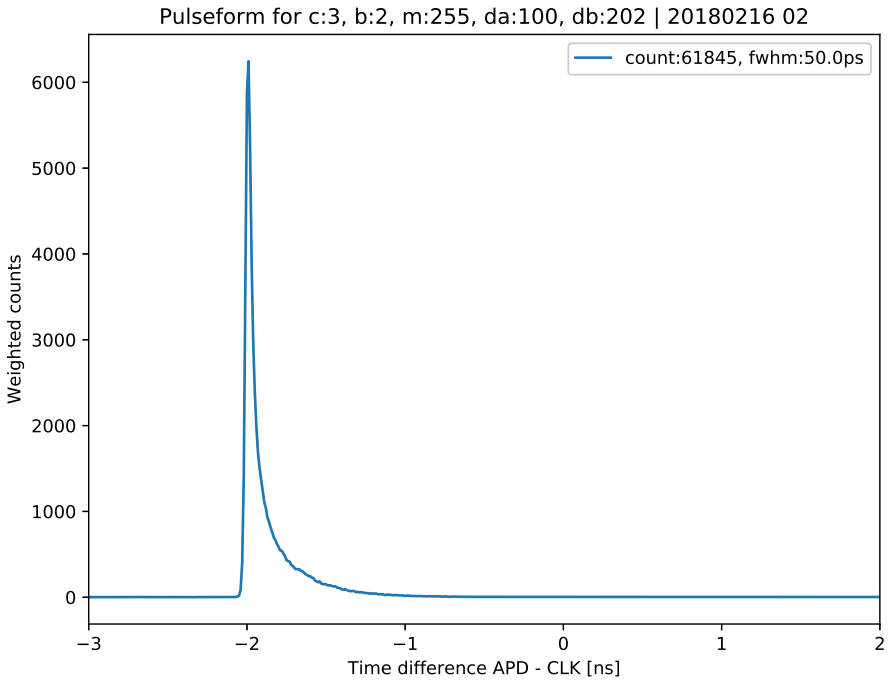


Figure 8.4: After improving the position and orientation of the lens in front of the VCSEL array this pulse shape could be obtained for channel 3. This shape is very similar to the pulse shape of the other three channels.

8.5 Further temporal pulse shape tomography plots

The following plots display the temporal pulse shape for channel 0 and channel 3 with medium and short pulse parameters with projections onto the polarization states $|P\rangle, |M\rangle, |R\rangle$ and $|L\rangle$. These temporal pulse shape are fairly similar for one channel and one set of parameters, because the polarization state comes close to the state $|V\rangle$, which is can be expressed as equal superpositions of $|P\rangle$ and $|M\rangle$ or $|R\rangle$ and $|L\rangle$ (see section 2.3.4).

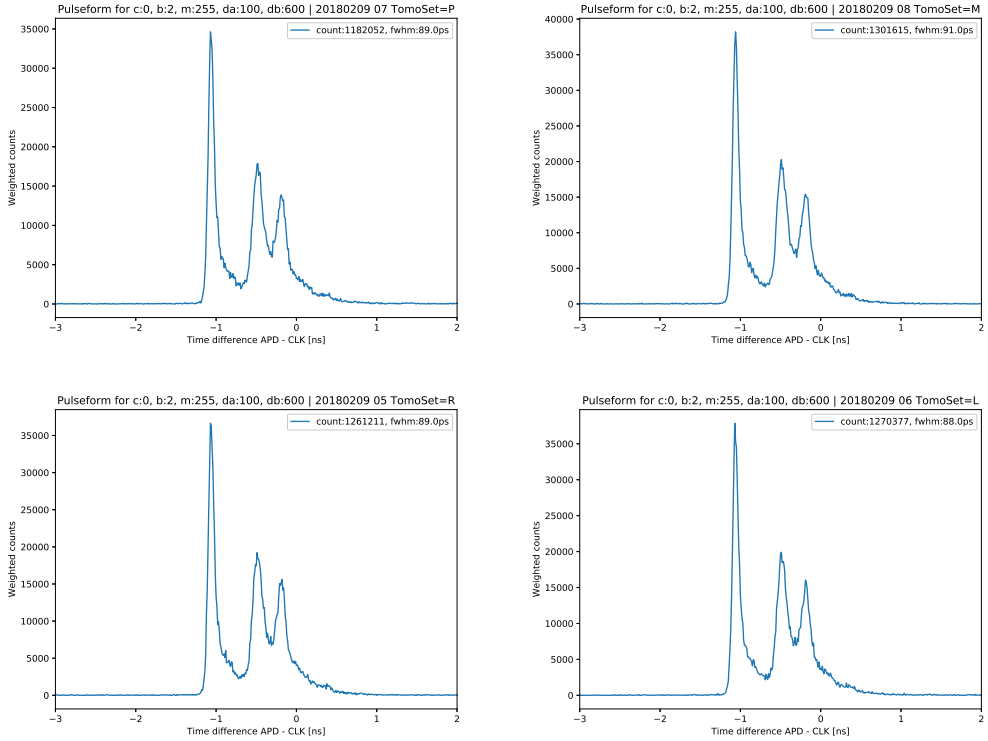


Figure 8.5: The measured temporal pulse shape for channel 0 with the medium pulse parameters. The top left picture shows the temporal pulse shape with the projection onto the diagonal (or plus) polarization state. The top right picture shows the temporal pulse shape with the projection onto the antidiagonal (or minus) polarization state. The bottom left picture show the temporal pulse shape with the projection onto the right-circular polarization state. The bottom right picture show the temporal pulse shape with the projection onto the left-circular polarization state.

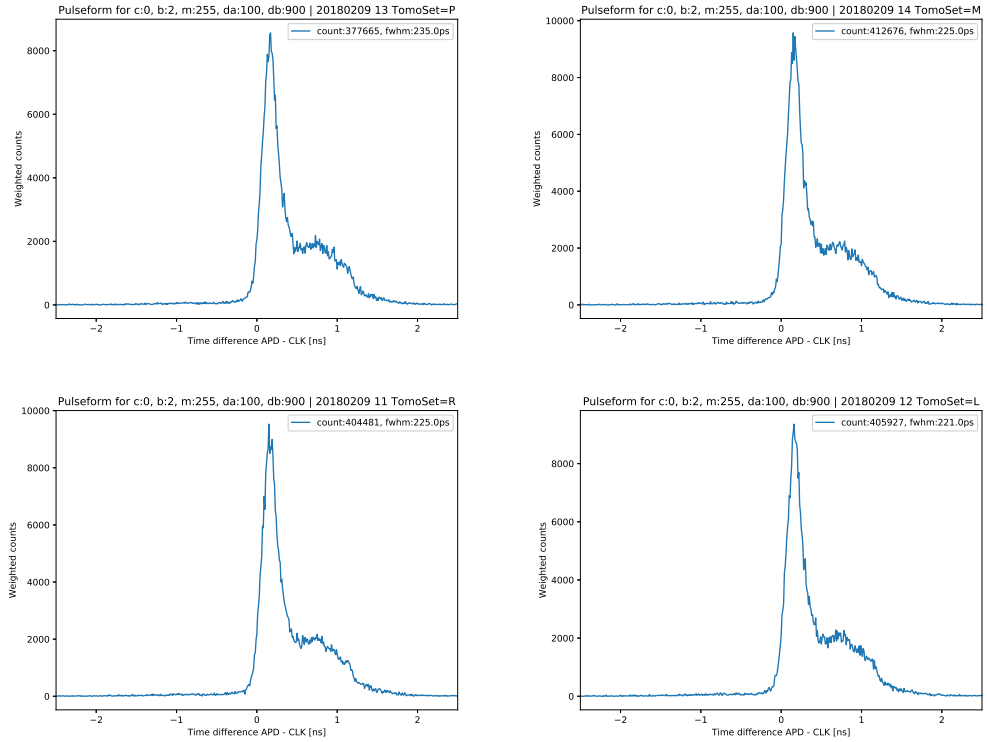


Figure 8.6: The measured temporal pulse shape for channel 0 with the long pulse parameters. The top left picture shows the temporal pulse shape with the projection onto the diagonal (or plus) polarization state. The top right picture shows the temporal pulse shape with the projection onto the antidiagonal (or minus) polarization state. The bottom left picture show the temporal pulse shape with the projection onto the right-circular polarization state. The bottom right picture show the temporal pulse shape with the projection onto the left-circular polarization state.

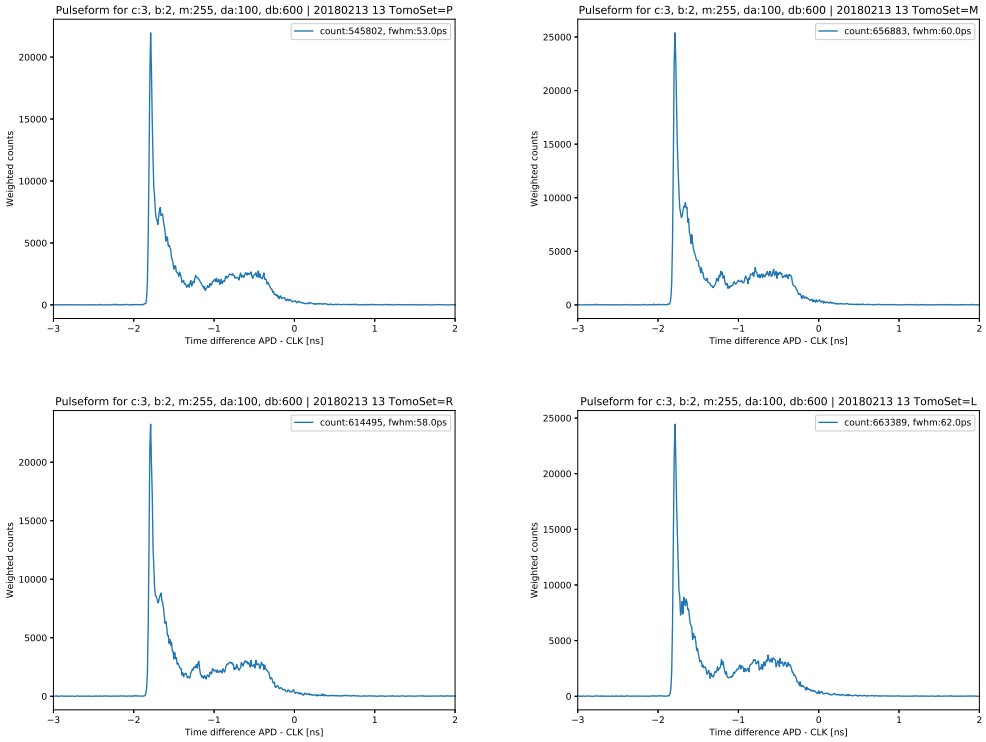


Figure 8.7: The measured temporal pulse shape for channel 3 with the medium pulse parameters. The top left picture shows the temporal pulse shape with the projection onto the diagonal (or plus) polarization state. The top right picture shows the temporal pulse shape with the projection onto the antidiagonal (or minus) polarization state. The bottom left picture show the temporal pulse shape with the projection onto the right-circular polarization state. The bottom right picture show the temporal pulse shape with the projection onto the left-circular polarization state.

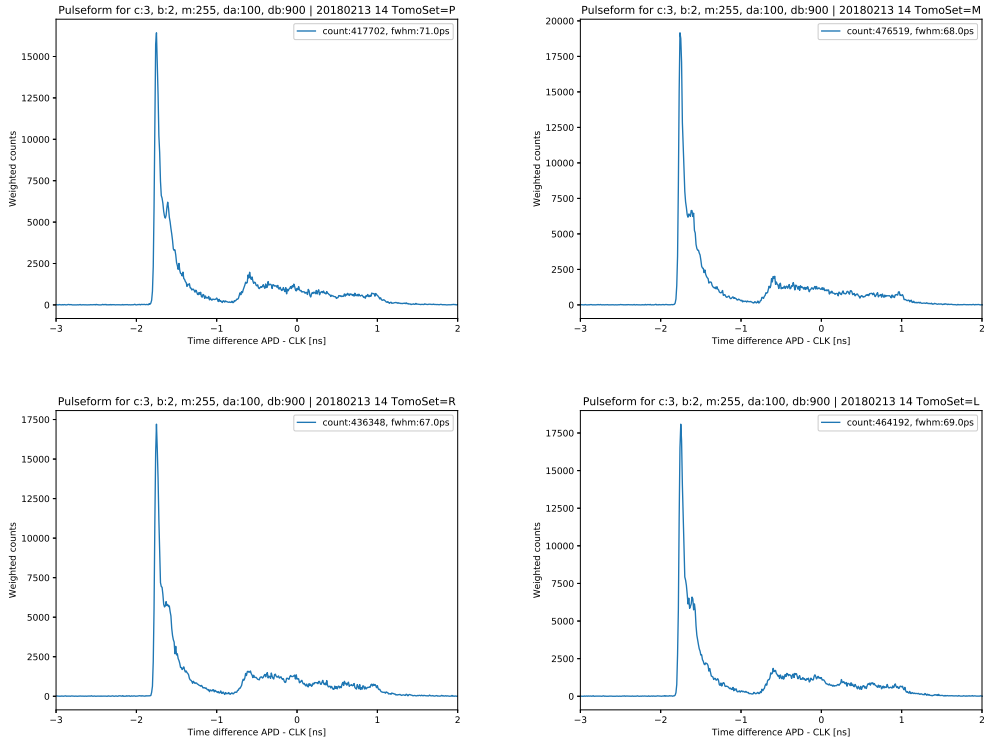


Figure 8.8: The measured temporal pulse shape for channel 3 with the long pulse parameters. The top left picture shows the temporal pulse shape with the projection onto the diagonal (or plus) polarization state. The top right picture shows the temporal pulse shape with the projection onto the antidiagonal (or minus) polarization state. The bottom left picture show the temporal pulse shape with the projection onto the right-circular polarization state. The bottom right picture show the temporal pulse shape with the projection onto the left-circular polarization state.

8.6 Additional electrical pulse shape plots

These plots show the electrical pulse shape for the parameter sets from the tomography measurement.

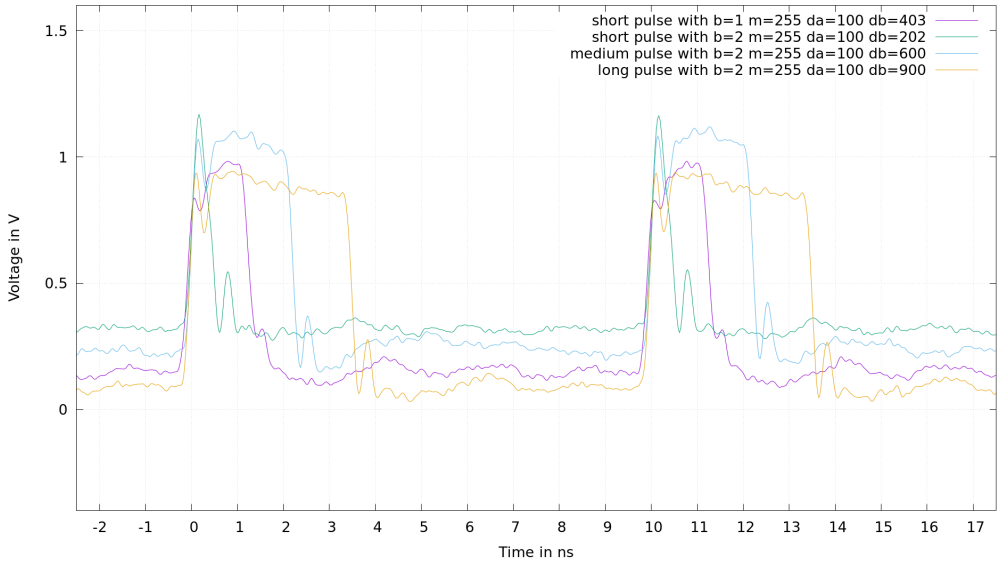


Figure 8.9: Electronic channel 2 with a 50 Ω resistor

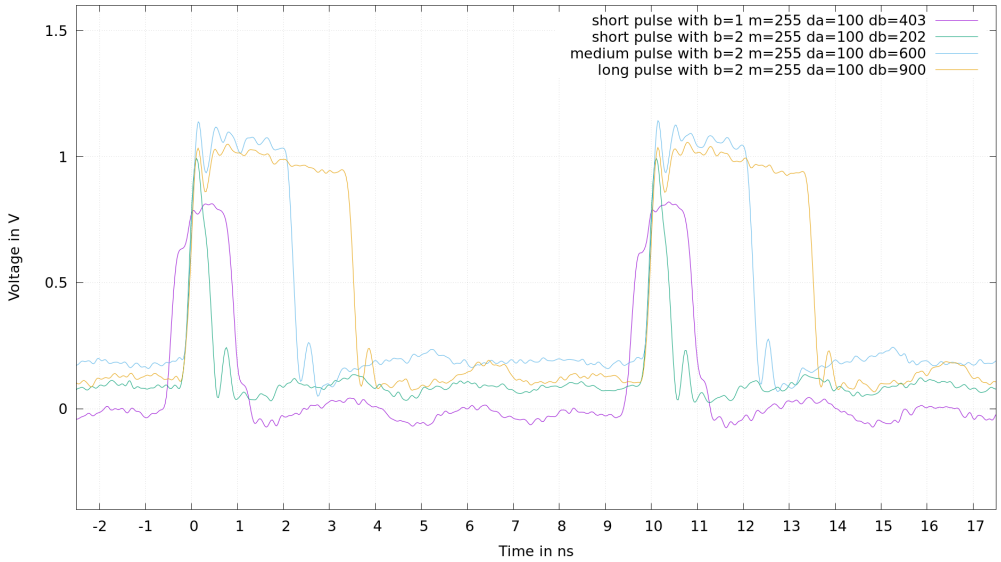


Figure 8.10: Electronic channel 3 with a 50 Ω resistor

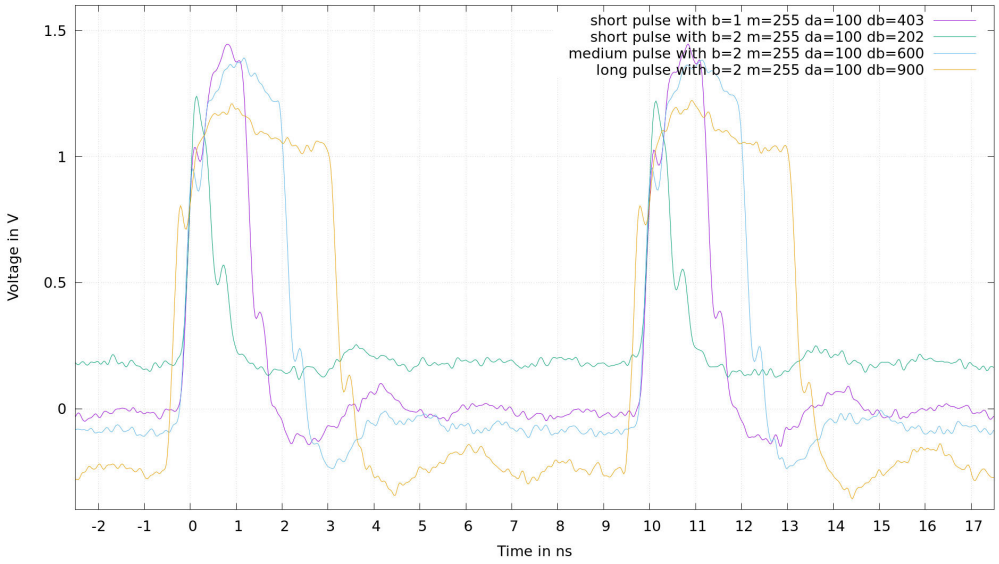


Figure 8.11: Electronic channel 2 with a 300 Ω resistor

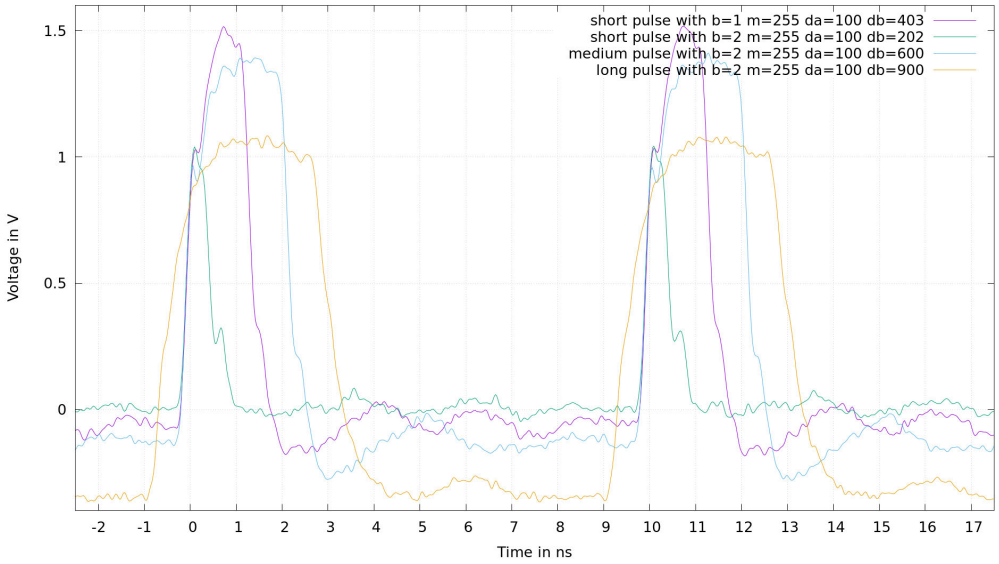


Figure 8.12: Electronic channel 3 with a 300 Ω resistor

Acronyms

QKD Quantum Key Distribution 1
OTP One-Time-Pad 1
FPGA Field-programmable gate array 2
AES Advanced Encryption Standard 12
CAC Classical Authenticated Channel 13
DHKE Diffie-Hellman Key Exchange 13
PNS photon-number splitting 29

Bibliography

[1] A Preview of Bristlecone, Google's New Quantum Processor. URL https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html (visited on 01/06/2018)

[2] IBM Q Experience. URL https://quantumexperience.ng.bluemix.net/qx/experience (visited on 01/06/2018)

[3] C. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, **28**:656–715, 1949.

[4] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, **41**(2):303–332, 1999.

[5] W. Diffie and M. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, Nov 1976.

[6] R. L. Rivest, A. Shamir and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, Feb 1978.

[7] NIST. Announcing the advanced encryption standard. *Federal Information Processing Standards Publication*, **197**, 2001.

[8] S. Vernam. Secret signaling system. *UNITED STATES PATENT*, **US 1310719 A**, 1919.

[9] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. *International Conference on Computer System and Signal Processing, IEEE, 1984*, pages 175–179, 1984. URL http://ci.nii.ac.jp/naid/20001457561/en/.

[10] A. K. Ekert. Quantum cryptography based on bell's theorem. *Phys. Rev. Lett.*, **67**:661–663, Aug 1991. doi: 10.1103/PhysRevLett.67.661. URL http://link.aps.org/doi/10.1103/PhysRevLett.67.661.

[11] N. Gisin, G. Ribordy, W. Tittel and H. Zbinden. Quantum cryptography. *Rev. Mod. Phys.*, 74:145–195, Mar 2002. doi: 10.1103/RevModPhys.74.145.

[12] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus and M. Peev. The security of practical quantum key distribution. *Rev. Mod. Phys.*, 81:1301–1350, Sep 2009. doi: 10.1103/RevModPhys.81.1301.

[13] H.-K. Lo, M. Curty and K. Tamaki. Secure quantum key distribution. *Nature Photonics*, 8:595–604, 2014. doi: 10.1038/nphoton.2014.149.

[14] M. Dušek, O. Haderka and M. Hendrych. Generalized beam-splitting attack in quantum cryptography with dim coherent states. *Optics Communications*, **169**:103–108, 1999. doi: http://dx.doi.org/10.1016/S0030-4018(99) 00419-8. URL http://www.sciencedirect.com/science/article/pii/S0030401899004198.

[15] H.-K. Lo, X. Ma and K. Chen. Decoy state quantum key distribution. *Phys. Rev. Lett.*, **94**:230504, Jun 2005. doi: 10.1103/PhysRevLett.94.230504. URL http://link.aps.org/doi/10.1103/PhysRevLett.94.230504.

[16] X.-B. Wang. Beating the photon-number-splitting attack in practical quantum cryptography. *Phys. Rev. Lett.*, **94**:230503, Jun 2005. doi: 10.1103/PhysRevLett 94.230503. URL http://link.aps.org/doi/10.1103/PhysRevLett.94.230503.

[17] X. Ma, B. Qi, Y. Zhao and H.-K. Lo. Practical decoy state for quantum key distribution. *Phys. Rev. A*, **72**:012326, Jul 2005. doi: 10.1103/PhysRevA.72.012326. URL http://link.aps.org/doi/10.1103/PhysRevA.72.012326.

[18] B. Huttner, N. Imoto, N. Gisin and T. Mor. Quantum cryptography with coherent states. *Phys. Rev. A*, **51**:1863–1869, Mar 1995. doi: 10.1103/PhysRevA.51.1863. URL http://link.aps.org/doi/10.1103/PhysRevA.51.1863.

[19] C. H. Bennett, G. Brassard, C. Crepeau and U. M. Maurer. Generalized privacy amplification. *IEEE Transactions on Information Theory*, **41**(6):1915–1923, Nov 1995. ISSN 0018-9448. doi: 10.1109/18.476316.

[20] C. H. Bennett and G. Brassard and J.-M. Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, **17**, Apr 1988.

[21] M. P. Peloso, I. Gerhardt, C. Ho, A. Lamas-Linares and C. Kurtsiefer. Daylight operation of a free space, entanglement-based quantum key distribution system. *New J. Phys.*, Apr 2009. doi: 10.1088/1367-2630/11/4/045007. URL http://iopscience.iop.org/article/10.1088/1367-2630/11/4/045007/meta.

[22] D. Vasylyev, A. A. Semenov, W. Vogel, K. Günthner, A. Thurn, Ö. Bayraktar and Ch. Marquardt. Free-space quantum links under diverse weather conditions. *Phys. Rev. A*, **96**:043856, Jul 2017. doi: 10.1103/PhysRevA.96.043856. URL https://journals.aps.org/pra/abstract/10.1103/PhysRevA.96.043856.

[23] T. Heindel, C. A. Kessler, M. Rau, C. Schneider, M. Fürst, F. Hargart, W.-M. Schulz, M. Eichfelder, R. Roßbach, S. Nauerth, M. Lermer, H. Weier, M. Jetter, M. Kamp, S. Reitzenstein, S. Höfling, P. Michler, H. Weinfurter and A. Forchel. Quantum key distribution using quantum dot single-photon emitting diodes in the red and near infrared spectral range. *New J. Phys.*, Aug 2012. doi: 10.1088/1367-2630/14/8/083001. URL http://iopscience.iop.org/article/10.1088/1367-2630/14/8/083001.

[24] H.-L. Yin, T.-Y. Chen, Z.-W. Yu, H. Liu, L.-X. You, Y.-H. Zhou, S.-J. Chen, Y. Mao, M.-Q. Huang, W.-J. Zhang, H. Chen, M. J. Li, D. Nolan, F. Zhou, X. Jiang, Z. Wang, Q. Zhang, X.-B. Wang and J.-W. Pan. Measurement device independent quantum key distribution over 404 km optical fibre. *Phys. Rev. Lett.*, **117**:190501, Jun 2016. doi: 10.1103/PhysRevLett.117.190501. URL https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.117.190501.

[25] M. Peev, C. Pacher, R. Alléaume, C. Barreiro, J. Bouda, W. Boxleitner, T. Debuisschert, E. Diamanti, M. Dianati, J. F. Dynes, S. Fasel, S. Fossier, M. Fürst, J.-D. Gautier, O. Gay, N. Gisin, P. Grangier, A. Happe, Y. Hasani, M. Hentschel, H. Hübel, G. Humer, T. Länger, M. Legré, R. Lieger, J. Lodewyck, T. Lorünser, N. Lütkenhaus, A. Marhold, T. Matyus, O. Maurhart, L. Monat, S. Nauerth, J.-B. Page, A. Poppe, E. Querasser, G. Ribordy, S. Robyr, L. Salvail, A. W. Sharpe, A. J. Shields, D. Stucki, M. Suda, C. Tamas, T. Themel, R. T. Thew, Y. Thoma, A. Treiber, P. Trinkler, R. Tualle-Brouri, F. Vannel, N. Walenta, H. Weier, H. Weinfurter, I. Wimberger, Z. L. Yuan, H. Zbinden and A. Zeilinger. The secoqc quantum key distribution network in vienna. *New J. Phys.*, **11**(7):075001, 2009. URL http://stacks.iop.org/1367-2630/11/i=7/a=075001.

[26] D. Stucki, M. Legré, F. Buntschu, B. Clausen, N. Felber, N. Gisin, L. Henzen, P. Junod, G. Litzistorf, P. Monbaron, L. Monat, J.-B. Page, D. Perroud, G. Ribordy, A. Rochas, S. Robyr, J. Tavares, R. Thew, P. Trinkler, S. Ventura, R. Voirol, N. Walenta and H. Zbinden. Long-term performance of the swissquantum quantum key distribution network in a field environment. *New J. Phys.*, **13**(12):123001, 2011. URL http://stacks.iop.org/1367-2630/13/i=12/a=123001.

[27] M. Sasaki, M. Fujiwara, H. Ishizuka, W. Klaus, K. Wakui, M. Takeoka, S. Miki, T. Yamashita, Z. Wang, A. Tanaka, K. Yoshino, Y. Nambu, S.

Takahashi, A. Tajima, A. Tomita, T. Domeki, T. Hasegawa, Y. Sakai, H. Kobayashi, T. Asai, K. Shimizu, T. Tokura, T. Tsurumaru, M. Matsui, T. Honjo, K. Tamaki, H. Takesue, Y. Tokura, J. F. Dynes, A. R. Dixon, A. W. Sharpe, Z. L. Yuan, A. J. Shields, S. Uchikoga, M. Legré, S. Robyr, P. Trinkler, L. Monat, J.-B. Page, G. Ribordy, A. Poppe, A. Allacher, O. Maurhart, T. Länger, M. Peev and A. Zeilinger. Field test of quantum key distribution in the tokyo qkd network. *Opt. Express*, **19**(11):10387–10409, May 2011. doi: 10.1364/OE.19.010387. URL http://www.opticsexpress.org/abstract.cfm?URI=oe-19-11-10387.

[28] J. Yin, Y. Cao, Y.-H. Li, S.-K. Liao, L. Zhang, J.-G. Ren, W.-Q. Cai, W.-Y. Liu, B. Li, H. Dai, G.-B. Li, Q.-M. Lu, Y.-H. Gong, Y. Xu, S.-L. Li, F.-Z. Li, Y.-Y. Yin, Z.-Q. Jiang, M. Li, J.-J. Jia, G. Ren, D. He, Y.-L. Zhou, X.-X. Zhang, N. Wang, X. Chang, Z.-C. Zhu, N.-L. Liu, Y.-A. Chen, C.-Y. Lu, R. Shu, C.-Z. Peng, J.-Y. Wang and J.-W. Pan. Satellite-based entanglement distribution over 1200 kilometers. *Science*, Vol. 356, Issue 6343, pages 1140-1144, Jun 2017. doi: 10.1126/science.aan3211.

[29] M. Rau, T. Vogl, G. Corrielli, G. Vest, L. Fuchs, S. Nauerth and H. Weinfurter. Spatial mode side channels in free-space qkd implementations. *IEEE Journal of Selected Topics in Quantum Electronics*, **21**(3):187–191, May 2015. ISSN 1077-260X. doi: 10.1109/JSTQE.2014.2372008.

[30] G. Vest, M. Rau, L. Fuchs, G. Corrielli, H. Weier, S. Nauerth, A. Crespi, R. Osellame and H. Weinfurter. Design and Evaluation of a Handheld Quantum Key Distribution Sender module. *IEEE Journal of Selected Topics in Quantum Electronics*, **21**(3):131–137, May 2015. ISSN 1077-260X. doi:10.1109/JSTQE.2014.2364131.

[31] G. Mélen. Integrated quantum key distribution sender unit for hand-held platforms. Dissertation, Ludwig-Maximilians-University Munich, 2016.

[32] P. Freiwang. Towards Hand-held Quantum Key Distribution. Master's Thesis, Ludwig Maximilian University of Munich, 2017.

[33] C. Sonnleitner. Towards a practical integrated QKD sender. Master's Thesis, Ludwig Maximilian University of Munich, 2018.

[34] J. Luhn. Handheld Quantum Key Distribution. Master's Thesis, Ludwig Maximilian University of Munich, 2017.

[35] T. Vogl. Mobile Free Space Quantum Key Distribution for short distance secure communication. Master's Thesis, Ludwig Maximilian University of Munich, 2016.

[36] R. Michalzik. *VCSELs: fundamentals, technology and applications of vertical-cavity surface-emitting lasers*, volume 166. Springer, 2012.

[37] H. Delfs and H. Knebl. *Introduction to Cryptography*, Sprinter, 3rd edition, 2015.

[38] D. J. Griffiths. *Introduction to Quantum Mechanics*, Pearson, 2nd edition, 2013.

[39] L. Susskind and G. Hrabovsky. *Classical Mechanics: The Theoretical Minimum*, Penguin, 1st edition, 2014.

[40] L. Susskind and A. Friedman. *Quantum Mechanics: The Theoretical Minimum*, Penguin, 1st edition, 2015.

[41] D. Dürr and S. Teufel. *Bohmian Mechanics: The Physics and Mathematics of Quantum Theory*, Springer, 2009.

[42] J. S. Bell. *Speakable and Unspeakable in Quantum Mechanics*, Cambridge University Press, 2nd edition, 2004.

[43] S. Axler. *Linear Algebra Done Right*, Springer, 3rd edition, 2014.

[44] E. Hecht. *Optics*, Pearson Education Limited, 5th Edition, 2016.

[45] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*, Cambridge University Press, 10th Edition, 2010.

[46] B. C. Wadell, *Transmission Line Design Handbook*, Artech House, 1991.

[47] Weak Diffie-Hellman and the Logjam Attack. URL: https://weakdh.org/.

Danksagung

An dieser Stelle möchte ich mich bei allen Menschen bedanken die mir den erfolgreichen Abschluss dieses letzten Studienjahres ermöglicht haben:

- Prof. Harald Weinfurter für die Möglichkeit, die experimentelle Seite der Physik kennenzulernen und an interessanten Projekten der QKD-Gruppe mitarbeiten zu dürfen.
- Dr. Wenjamin Rosenfeld für seine Hilfe bei allerlei Problemen von theoretischer und experimenteller Natur, die sich mir in den Weg gestellt haben.
- Peter Freiwang für die Chance meiner persönlichen und beruflichen Weiterbildung in der QKD-Gruppe.
- Martin Zeitlmair für seine Hilfe im Labor und der Lösung bei meinen zahlreichen IT-Problemen.
- Kai Redeker, Robert Garthoff und Daniel Burchardt für ihre stetige Hilfsbereitschaft.
- Clemens Sonnleitner und Lukas Grimmeißen für die spaßige Zusammenarbeit im Labor und am Schreibtisch.
- Und schlussendlich bei meinen Eltern für ihre Unterstützung während meines gesamten Studiums.

Erklärung

Hiermit erkläre ich, die vorliegende Arbeit selbständig verfasst zu haben und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel benutzt zu haben.

München, den 21. Juni 2018

_____

Thomas Schwarzwälder