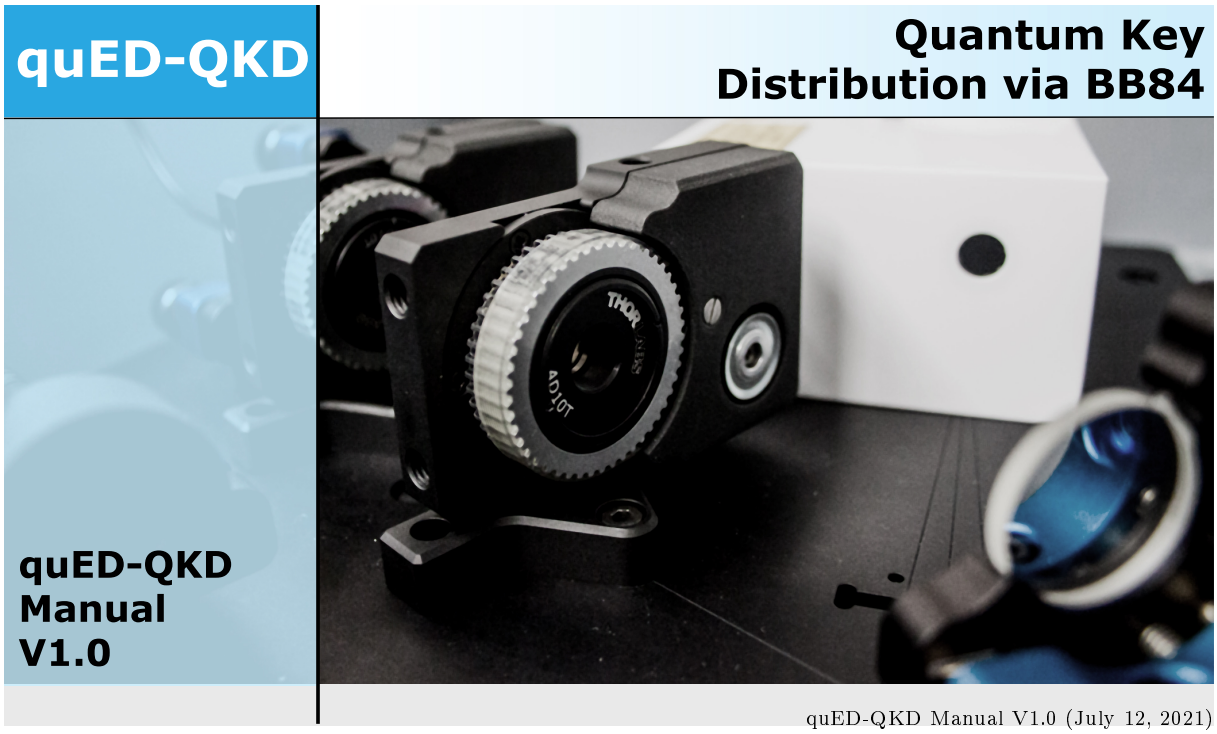


# Contents

# 1 quED-QKD: Quantum Key Distribution Manual

## 1.1 Quickstart Manual

With the quED-QKD Add-On, you gain the ability to pulse the pump laser diode, and with that, to generate weak coherent pulses (almost as good as single photons) on the push of a button. To do that, you can simply switch to pulsed mode in the laser tab of the quCR control unit interface, see Fig. 1.1.

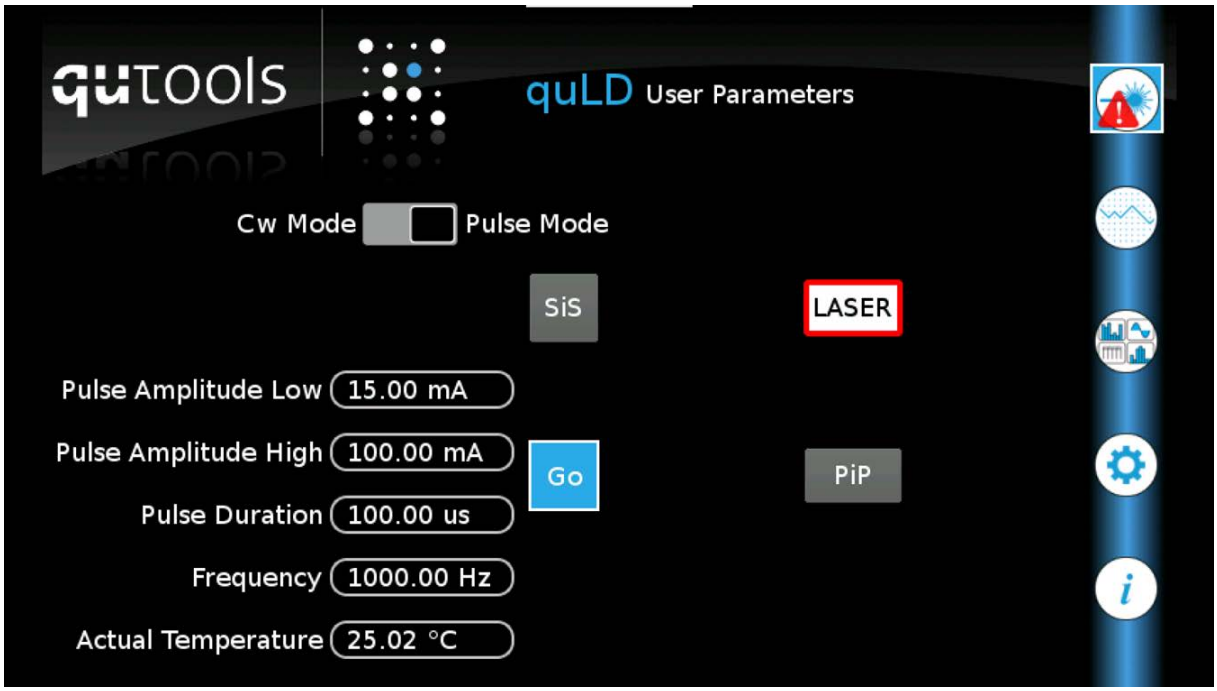


Figure 1.1: The pulsed laser menu of the quCR control interface.

You can tweak the following factors:

1. **Pulse Amplitude:** How much current will flow during the pulse
2. **Pulse Duration:** How long will a single pulse last (together with pulse amplitude, this defines the average photon number per pulse)
3. **Frequency:** How often a pulse will be generated (only applicable if the "'Go"' Button is pressed)

Table 1.1: The settings needed for pulses with approximately a single photon per pulse (on average).

| Factor | value |
|---|---|
| Pulse Amplitude | operating current, see quED datasheet |
| Pulse Duration | $1\mu$s |

To generate pulses with approximately a single photon per pulse (on average), use the settings in Tab. 1.1. Please note that you will be detecting much less than a photon per pulse, because of detector and coupling efficiencies.

Additionally, please notice new functionalities in the count rates tab as well as new tabs to generate and measure a single pulse. If you have the motorized version of the quED-QKD, the BB84 Experiment can be performed step by step in the BB84 tab of the quCR control interface.

# 2 Experiments with the quED-QKD

## 2.1 Weak coherent pulses

A single photon at the push of a button - a subject still being researched heavily. Weak coherent pulses are an easy approximation, sufficient for many applications, e.g. quantum key distribution.

---



---

### 2.1.1 Theoretical Background

Weak coherent pulses are just weak laser pulses. As such, the photon numbers in one pulse obey the Poisson distribution, where the probability of $k$ photons in one pulse is given by

$$P_k = e^{-\lambda} \frac{\lambda^k}{k!} , \tag{2.1}$$

with the average photon number $\lambda$. For most applications here, we take into account the probabilities for no photon in a pulse, exactly one photon in a pulse and more than one photon in a pulse. Some calculations of these probabilities for different average photon numbers can be found in Tab. 2.1.

You can see that by reducing the average photon number, the relation between pulses with one photon compared to pulses with more than one photon becomes greater, which is good. But, at the same time, more and more pulses don't have any photons in them, making it hard to reach acceptable count rates. This is a consideration one has to make for each individual task.

Table 2.1: Probabilities for the number of photons given by the Poisson distribution for different average photon numbers.

| $\lambda$ | $P_0$ | $P_1$ | $P_{>1}$ |
|---|---|---|---|
| 0.1 | 90.5 % | 9.05 % | 0.45 % |
| 0.5 | 60.7 % | 30.3 % | 9.0 % |
| 1.0 | 36.8 % | 36.8 % | 26.4 % |
| 2.0 | 13.5 % | 27.1 % | 59.9 % |
| 10.0 | 0.005 % | 0.05 % | 99.945 % |

### 2.1.2 Implementation with the quED

#### Necessary Components

- quED source
- quCR control rack upgraded with the pulsed laser option

#### Experimental description

To generate and gauge weak coherent pulses, first remove the polarizers from the quED and the half waveplate from the source, and realign the setup for maximal coincidences. After that, switch to the pulsed laser mode in the laser tab and activate the Picture-in-Picture (PiP) Overlay, such that you can modify the pulse settings from anywhere in the quCR software. A single pulse is specified by its amplitude and its duration, and you can modify the frequency with which the pulses are generated.

Switch to the count rate panel to see the signals from the APDs. Please note that (while the *sync* checkbox is active) only counts happening during a laser pulse are displayed, but they are still integrated over the integration interval. So, if you set the frequency to 1000 Hz and the Integration time to 100ms, you can see how many photons are detected during 100 pulses. You can also display the average number of photons per pulse if you activate the *per pulse* checkbox.

The average number of photons per pulse can then be adjusted by changing the pulse duration and the pulse amplitude. You can also try what happens when the laser is switched off.

#### Measurement example

Here, we set the pulse duration as short as possible such that we still observe pulses with 0.5 photons on average, see Fig. 2.1.

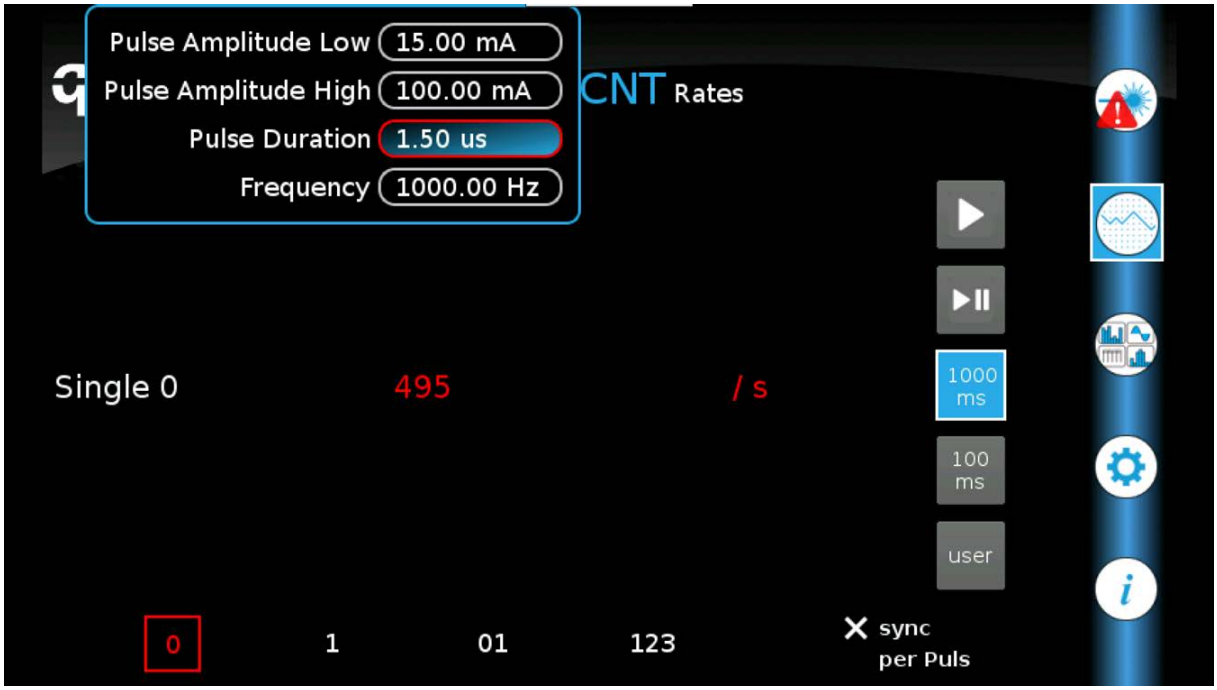


Figure 2.1: A screenshot of the quCR with the average number of photons per pulse for a given setting.

### 2.1.3 Didactic Material

1. Why is it of advantage to set the pulse duration as low as possible?
2. Calculate the ratio between pulses with no photon and pulses with one photon and the ratio between pulses with one photon and pulses with two photons for an average photon number of $\lambda$.
3. What problems does this measurement of the average photon number have, especially at very short pulse durations? (Hint: think about properties of the APDs.)
4. How could one try to measure this effect?

### 2.1.4 Sample Solution

For the sample solution please refer to the qutools quED-QKD page `http://qutools.com/quED-QKD`.

## 2.2 Quantum Key Distribution: The BB84 Protocol

Most modern cryptography schemes are based on the impossibility to decompose a large number into its prime factors during acceptable time intervals. With enough computing power though, it is theoretically possible to crack such an encryption. In contrast, two parties can communicate intrinsically secure by employing quantum effects.



### 2.2.1 Theoretical Background

#### Introduction

Conventionl cryptography schemes that are founded on a secret key are only then completely secure when each key is as long as the message itself and used only once. Thus, the encryption problem shifted to the exchange of a secret key. At the moment, mostly asymmetric schemes (e.g. RSA) using a public key (for encryption) and a private key (for decryption) are being employed. The secureness relies essentially on the impossibility to factorize a large number in its prime factors. With faster computers, the possibility of a quantum computer or newly found algorithms, the security of these conventional systems is diminishing.

With the BB84 protocol, named after its inventors and the year of publication (Bennett and Brassard, 1984 [1]), it is possible to use the quantum physical properties of photons to transfer a secret key between two parties, tap-proof. When that has happened, the message can be encrypted and sent via an open classical channel. Because of that, the term *quantum cryptography* is actually misleading, it should be called *quantum key distribution* (QKD) instead.

#### The BB84-Protocol

The two parties involved in the secure communication are called *Alice* and *Bob* by convention, see also Fig. 2.2. Alice operates a source of single photons that she can individually prepare in a linear polarization state known to her and sends them to Bob. Alice chooses between two bases, e.g. straight $\oplus$ and diagonal $\otimes$. Each basis consists of two states, namely $|H\rangle$ and $|V\rangle$ ($\oplus$) and $|P\rangle$ and $|M\rangle$ ($\otimes$), respectively. Each state represents a binary value 0 ($|H\rangle$ und $|P\rangle$) or 1 ($|V\rangle$ and $|M\rangle$). As such, Alice can prepare a random

Table 2.2: The BB84 protocol, step by step

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **quantum channel:** | | | | | | | | | | | | | | |
| Alices random bit sequence | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| Alices random base choice | ⊗ | ⊗ | ⊕ | ⊕ | ⊗ | ⊕ | ⊕ | ⊕ | ⊗ | ⊕ | ⊗ | ⊗ | ⊕ | ⊕ |
| sent photon polarization $\|\cdot\rangle$ | P | M | V | H | M | V | H | H | P | V | P | M | H | V |
| Bobs random base choice | ⊕ | ⊗ | ⊗ | ⊕ | ⊕ | ⊗ | ⊕ | ⊗ | ⊕ | ⊗ | ⊗ | ⊕ | ⊕ | ⊕ |
| received bits | 1 | 1 | 1 | | 0 | 1 | 0 | 1 | 0 | 0 | 0 | | 0 | 1 |
| **classical channel:** | | | | | | | | | | | | | | |
| Bob sends bases | ⊕ | ⊗ | ⊗ | | ⊕ | ⊗ | ⊕ | ⊗ | ⊕ | ⊗ | ⊗ | | ⊕ | ⊕ |
| Alice confirms | | ✓ | | | | | ✓ | | | | ✓ | | ✓ | ✓ |
| probably shared bits | | 1 | | | | | 0 | | | | 0 | | 0 | 1 |
| **eavesdropper detection** | | | | | | | | | | | | | | |
| Bob shares randomly | | 1 | | | | | | | | | 0 | | | |
| Alice confirms | | ✓ | | | | | | | | | ✓ | | | |
| **remaining key** | | | | | | | 0 | | | | | | 0 | 1 |

bit sequence in random bases and send it. Bob receives the photons and can choose one basis per photon for a polarization measurement. In doing so, a meaningful bit is only received when Bob and Alice chose the same basis. Thus, they have to communicate their basis choice. If Bob detects no photon because of losses in the quantum channel, the corresponding bit is discarded.

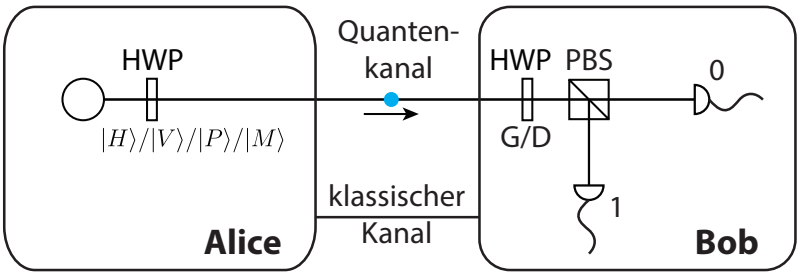


Figure 2.2: Schematic setup for implementation of the BB84 protocol.

The following steps are then done using a classical (public) channel. Bob sends a list with his measurement bases to Alice. She compares the list to her own and tells Bob which bases match. All bits with non-matching bases are discarded by both parties. The remaining bits make up the secret key.

### Detection of Eavesdroppers

In general, it is assumed that *Eve* (from *eavesdropper*) can intercept both the classical communication channel and the photons sent by Alice, see Fig. 2.3. So how is it still

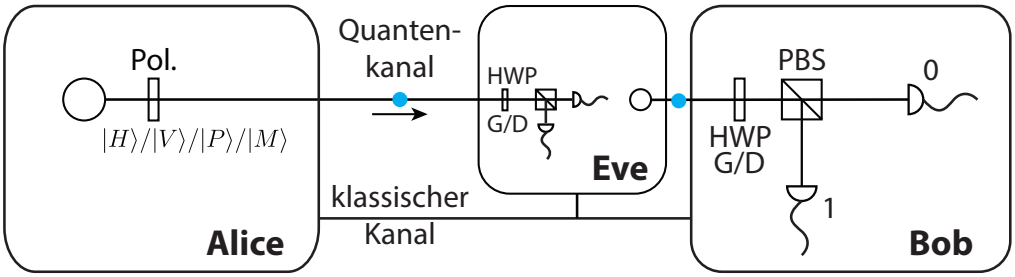


Figure 2.3: Eve can eavesdrop on the classical channel and interept the photons in the quantum channel.

possible to exchange a secret key, or rather, to detect the attempt? The BB84 protocol takes advantage of two important properties of single photons:

- By a single measurement, the polarization state of a single photon can not be determined fully.
- The polarization state of a single photon may not be copied. (No-Cloning-Theorem, [2])

Eve could, e.g., intercept part of the photons from Alice and perform a polarization measurement, just as Bob would. But, if Bob receives no photons at the proper time, the bits are just discarded and Eve gains no information about the key. Therefore, Eve sends one photon onward to Bob in each case, with the polarization measured by Eve. This kind of attack is called an intercept-resend attack.

Bob and Alice can detect such an attempt by comparing part of the key. That is to say, if Eve chooses a basis that does not conform with Alices and Bobs, the Bit received by Bob is random. Thus, it can happen that Bob's Bit does not match with Alice's. If some of those "'errors"' are found, they can assume that the photon channel is compromised and their key is not secure. There are of course more sophisticated ways of comparing the key than just publishing part of it, called error correction schemes. Some popular choices are the *cascade* protocol or the *low density parity check*.

**Remark on the quantum channel:** It is important that Alice sends just one photon for each bit. That is, experimentally, not as simple as it sounds, since Alice has to know exactly when a photon was generated and sent, she needs a real *single photon source*. There are still intensive research efforts going on in that direction. Therefore, sources for *weak coherent pulses* are often used at the moment. Here, the photon number per pulse obeys the Poisson distribution.

If Alice sends multiple Photons in a pulse, Eve could employ a so called *beam splitter attack*. She intercepts only part of the photons for her measurements. Bob still receives photons directly from Alice and can therefore not determine the presence of an attacker.

In the case of weak coherent pulses, the average number of photons per pulse has to be taken into account when calculating the security of the protocol. As a rule of thumb, the average photon number has to be smaller than the transmission of the quantum channel

(proof of GLLP). By using more sophisticated protocols like the *Decoy state* protocol, the photon number can be increased, though.

**Remark on the classical channel:** Alice and Bob can communicate publicly through the classical channel, but they have to be sure that they communicate directly with each other and Eve does not change the content of their messages. This can be done (cue authentication) by a previously exchanged secret key.

### 2.2.2 Implementation with the quED

#### Necessary Components

- quED source
- quCR control rack upgraded with the pulsed laser option
- polarizer
- half waveplate

#### Experimental description

**Set up the source** The source should be set up without the half waveplate inserted, such that only horizontally polarized photons will be produced. Perform a short point-degree of freedom alignment to maximize the count rate in the left arm of the quED. The right arm will not be used in this experiment.

**Weak coherent pulses** Then, set up the number of photons per pulse as described in 2.1 Weak coherent pulses. In practice, the more photons per pulse are sent, the less secure the protocol will be against an attack. But, with less photons per pulse, one needs more pulses to transmit a secret key, limiting the key rate. Since the coupling and detection of photons in the quED has an efficiency of approximately 10%, the actual number of photons per pulse is ten times that of the detected number. For a secure protocol, less than 1 actual photons per pulse should be sent, meaning less than 0.1 should be detected. Of course, especially in the manual version, you might want to turn this number up a bit (to about one detected per pulse) such that a reasonable key rate can be reached, sacrificing security.

**General setup** Since the procedure for the experiment differs greatly for the two versions (manual and motorized), they will be described separately below. Both versions have in common that a polarizer is used in Bob's Setup instead of a polarizing beam splitter, see Fig. 2.4. This is because the quED does not have the option to accomodate the beam splitter and the additional fiber coupler. Using a polarizer only leads to a greater number of necessary experiment runs, since less photons will be detected on Bobs side, making it necessary to discard the respective run.

**Motorized Version** With the motorized version, just set up the experiment as shown in Fig. 2.5. Then switch to the BB84 Tab in the quAPP menu, see Fig. 2.6.

In this tab, you can run through every step of the BB84 protocol:

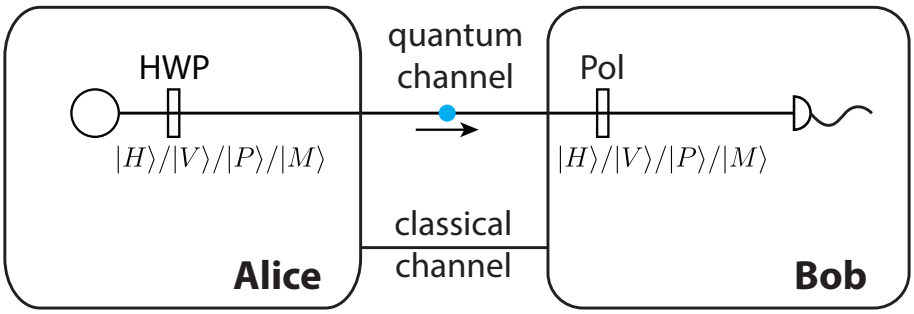


Figure 2.4: Schematic for the setup of the BB84 experiment with the quED. Instead of a polarizing beam splitter, a polarizer is used.

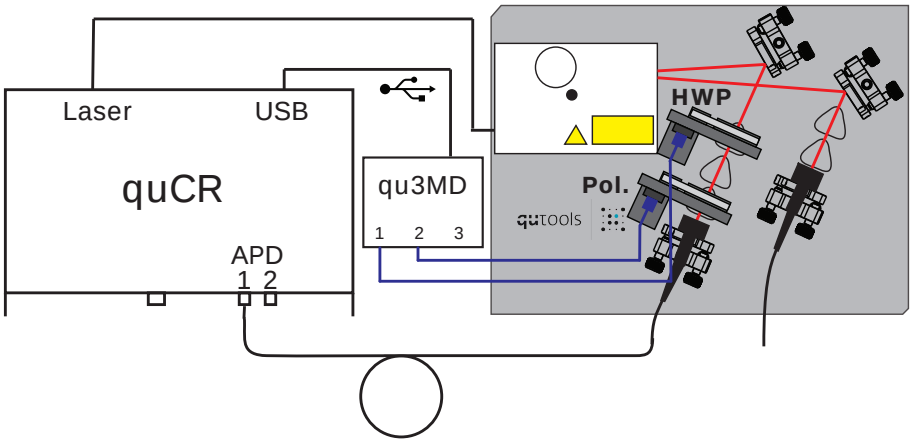


Figure 2.5: Setup of the BB84 experiment with the motorized quED.

1. Alice and Bob need randomly chosen measurement bases and bit values. The randomness can either be mathematically produced (a) are loaded from a file (b). These files can e.g. be produced by hand or with the random number generator tab if you have the quED-HBT Add-On.
2. The actual key exchange over the quantum channel can be done run by run (a), or continuously (b). The motors will be set to the angles specified by base and bit value automatically, after which a single pulse specified by the values in the laser tab will be released. On Bob's side, it will be noted if the respective pulse led to a detection event.
3. All operations over the classical channel can be done using the buttons in the middle of the two tables:
   a) The first button is for communicating which runs led to an detection event. On the first push, the detections are communicated to Alice and all experiment runs without a detection event are marked for deletion. The second push

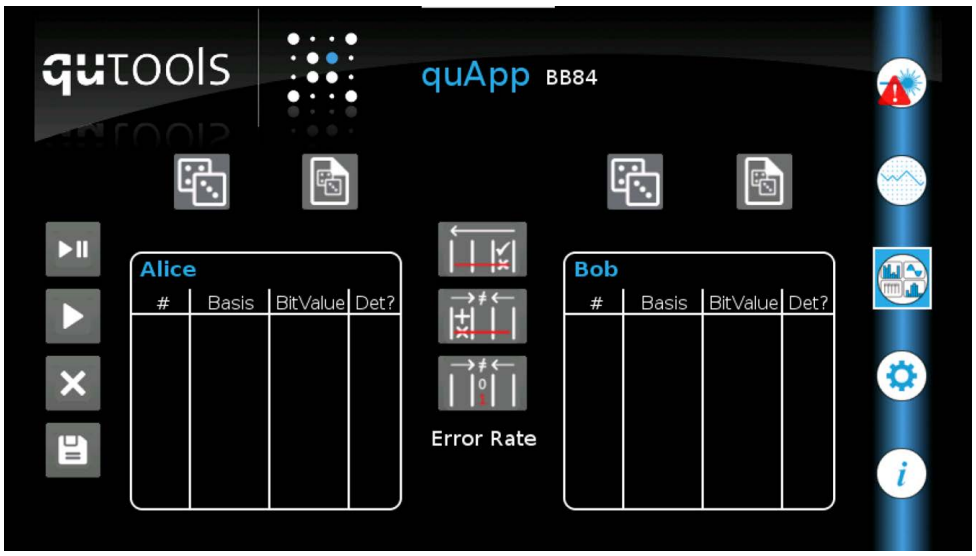


Figure 2.6: The BB84 tab is made for an semi-automated key exchange with the motorized version.

removes them from the tables, while the third push will remove the *detected* column. You can restore all that by pushing a fourth time.

b) The second button is for comparing the bases. Like the first button, it can be pressed multiple times, toggling the different states: 1) Bases are communicated, differing ones are marked for deletion. 2) These are removed. 3) The whole column is removed. 4) Reset.

c) The third button is simulating an error correction protocol, with which an attacker can also be identified. Again, the four states can be toggled, this time regarding errors in the bit values.

d) The error rate is shown beneath the buttons and can be used to determine how much information an attacker could have about the key.

4. The tables can also be cleared (a) and saved (b) as a csv file.

**Manual version** In the manual version, all steps of the BB84 protocol have to be done by hand. You can use the templates supplied below. For detection, use the *Single Pulse* Tab in the quAPP menu, see Fig. 2.7. Just set the half waveplate (Alice) and the polarizer (Bob) to their respective angles, push the button to release a single pulse and record if a detection occured. If the round LED glows with color, something was detected, if it is grey there was no detection during the pulse.

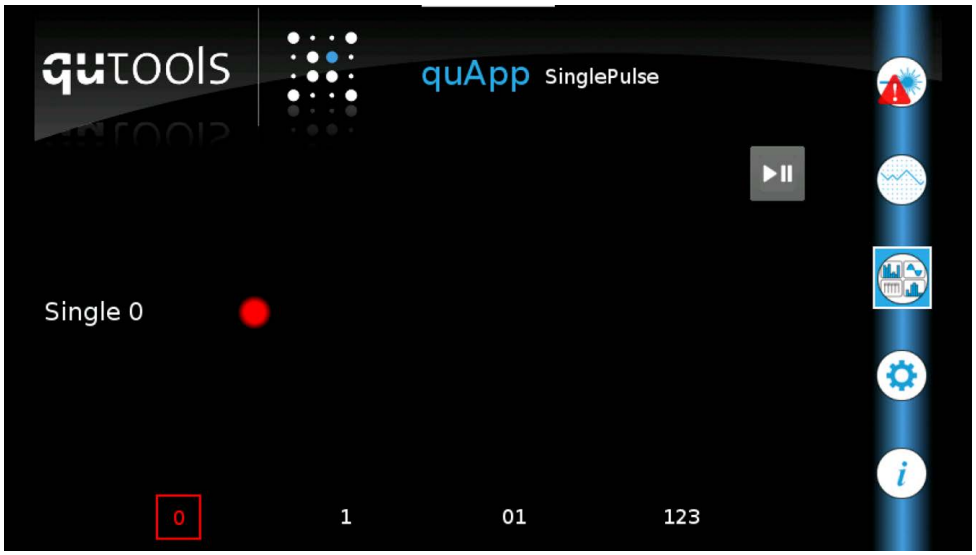


Figure 2.7: The single pulse tab can be used for manually going through the BB84 protocol.

### 2.2.3 Didactic Material

1. How can one improve the efficiency?
2. How can the basis choice at Bob's be made really (quantumly) random?
3. What is the simplest attack on such a system? What is the special problem of the system? (Hint: Think about authentication.)

# quED - Quantum cryptography

Quantum key distribution QKD

qutools

## Sender Alice

| Basis | + | | × | | | | |
|---|---|---|---|---|---|---|---|
| Bit value | H - 0 | V - 1 | P - 0 | M - 1 | | | |
| HWP | 0° | 45° | 22,5° | -22,5° | Received? | Equal basis? | Key |
| 1 | | | | | | | |
| 2 | | | | | | | |
| 3 | | | | | | | |
| 4 | | | | | | | |
| 5 | | | | | | | |
| 6 | | | | | | | |
| 7 | | | | | | | |
| 8 | | | | | | | |
| 9 | | | | | | | |
| 10 | | | | | | | |
| 11 | | | | | | | |
| 12 | | | | | | | |
| 13 | | | | | | | |
| 14 | | | | | | | |
| 15 | | | | | | | |
| 16 | | | | | | | |
| 17 | | | | | | | |
| 18 | | | | | | | |
| 19 | | | | | | | |
| 20 | | | | | | | |
| 21 | | | | | | | |
| 22 | | | | | | | |
| 23 | | | | | | | |
| 24 | | | | | | | |
| 25 | | | | | | | |
| 26 | | | | | | | |
| 27 | | | | | | | |
| 28 | | | | | | | |
| 29 | | | | | | | |
| 30 | | | | | | | |
| 31 | | | | | | | |
| 32 | | | | | | | |
| 33 | | | | | | | |

Key _____

Message _____

Cipher _____

# quED - Quantum cryptography

Quantum key distribution QKD

qutools

## Receiver Bob

| Basis | + | | × | | | | |
|---|---|---|---|---|---|---|---|
| Bit value | H - 0 | V - 1 | P - 0 | M - 1 | | | |
| Polarizer | 0° | 90° | 45° | -45° | Received? | Equal basis? | Key |
| 1 | | | | | | | |
| 2 | | | | | | | |
| 3 | | | | | | | |
| 4 | | | | | | | |
| 5 | | | | | | | |
| 6 | | | | | | | |
| 7 | | | | | | | |
| 8 | | | | | | | |
| 9 | | | | | | | |
| 10 | | | | | | | |
| 11 | | | | | | | |
| 12 | | | | | | | |
| 13 | | | | | | | |
| 14 | | | | | | | |
| 15 | | | | | | | |
| 16 | | | | | | | |
| 17 | | | | | | | |
| 18 | | | | | | | |
| 19 | | | | | | | |
| 20 | | | | | | | |
| 21 | | | | | | | |
| 22 | | | | | | | |
| 23 | | | | | | | |
| 24 | | | | | | | |
| 25 | | | | | | | |
| 26 | | | | | | | |
| 27 | | | | | | | |
| 28 | | | | | | | |
| 29 | | | | | | | |
| 30 | | | | | | | |
| 31 | | | | | | | |
| 32 | | | | | | | |
| 33 | | | | | | | |

Key _____

Cipher _____

Message _____

### 2.2.4 Sample Solution

For the sample solution please refer to the qutools quED-QKD page `http://qutools.com/quED-QKD`.

# Bibliography

[1] C.H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, 175:8, 1984.

[2] W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, October 1982.