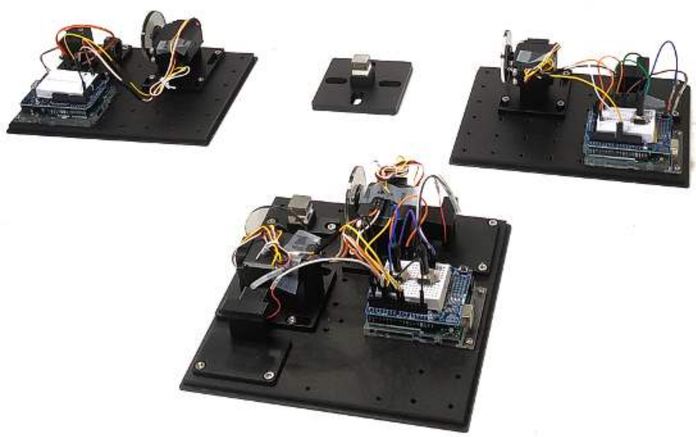
S-Fifteen Instruments Pte. Ltd. (Singapore)

EKPQC Ver. 1.2

Educational Kit: Programmable Quantum Cryptography

Assembly and Installation Manual

July 10, 2023



Contents

1 Safety

Please note the following safety considerations.

1.1 Laser Classification and Safety

The $< 1\,\mathrm{mW}$ $650\,\mathrm{nm}$ laser diode used in the EKPQC , is a Class 2 laser in accordance to Radiation Protection (Non-Ionising) Regulations 1991 [1]:

> Class 2 lasers emit visible light and are limited to a maximum output power of 1 milliwatt (mW). A person receiving an eye exposure from a Class 2 laser will be protected from injury by the person's natural aversion response – an involuntary response which causes the person to blink and turn their head, thereby avoiding further eye exposure.

Although there is no requirement for additional protective equipment, nonetheless, we would like to highlight the following safety precautions:

1. Ensure that the experimental setup is constructed below eye level.
2. Do not look directly at the laser, or point it at someone's eye.
3. Alert everyone on the beam path, so that they can avoid it.
4. Do not wear reflective accessories that may accidentally reflect the beam into somebody's eye e.g. watches, bracelets and rings.
5. Turn off the laser after use.

1.2 Electrical safety

Mains

As with any electronic devices, the EKPQC is susceptible to high voltage discharges. Ensure that the mains powering the power adaptor and also any power supply to the attached computer is protected from surges.

Electronics

The EKPQC also contains individual electronic components that are used to build the kit. Although it has been carefully designed to minimise any possible burnout of electronics, care should still be taken during installation. This will be labelled by the important box, as shown below.

> Any important point that should receive additional care will be noted in a box such as this.

The power adaptor used to power the Arduino UNO and the motors in the EKPQC is a 9 V 0.5A. Although the Arduino UNO accepts up to 12 V, **do not do this** since the motor can only accept a maximum of 8.4 V.[1]

Warning!

> ☞ This programmable QKD system is designed for pedagogical purposes only: it implements the BB84 QKD protocol albeit with a slight modification, which introduces a security loophole. We do not recommend using the system for establishing QKD in a non-educational context.

[1] There are 2 additional diodes, one on the UNO and one on the breadboard, in forward bias that reduces the 9 V input to less than 8.4 V in compliance with the maximum ratings.

2 Description

The Educational Kit: Progammable Quantum Cryptography (EKPQC) is a set of experimental tools and programs that can be used to educate students on Quantum Key Distribution following[2]. This manual only details the hardware assembly and software installation for the EKPQC . The Student Guide is the complementary document to this manual that is used in a classroom setting.

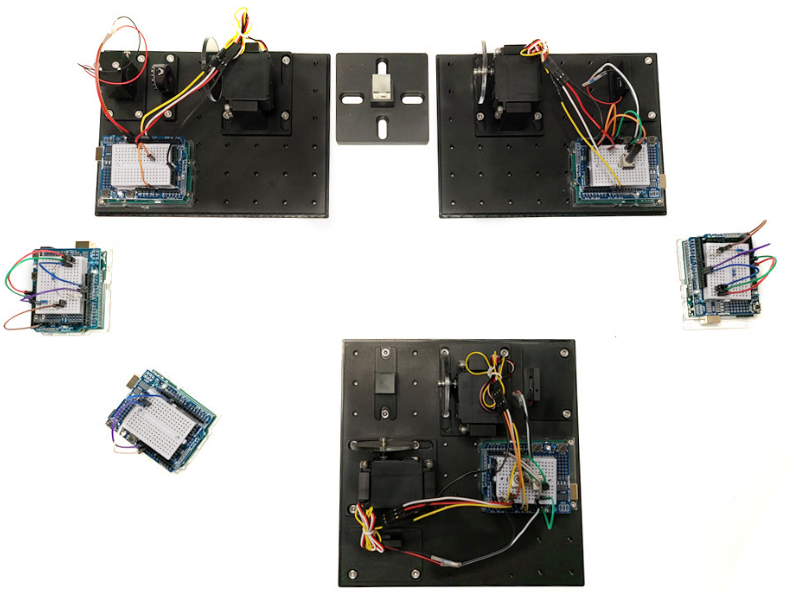


Figure 2.1: Top view of the layout of the EKPQC without the USB and power adaptor cables attached. Clockwise from top left: Alice's Quantum breadboard, beamsplitter for Eve to tap the quantum channel, Bob's Quantum breadboard, Bob's classical Arduino, Eve's Quantum breadboard, Eve's classical Arduino, Alice's classical Arduino.

2.1 Introduction

The main setup of the EKPQC after the complete assembly is shown in Figure 2.1. The attached computer and power adaptors to the Adruinos are not shown in the figure. The cover for when the kit is not in use is also not shown.

The EKPQC implements a BB84 equivalent protocol, with macroscopic laser light instead of single photons. The simplified schematic is shown in Fig. 2.2.

For ease of implementing the quantum channel, Bob will be measuring the polarization state of the photons using a motorized, rotatable linear polarizer, and a single photodetector.

Alice and Bob will also establish a public classical channel using infrared (IR) pulses by assembling IR transmitting and receiving circuits. Eve will attempt to eavesdrop on the quantum and classical channels using a beamsplitter and additional IR devices, respectively.

Apart from the implementation differences in the quantum channel, the steps involved in the BB84 protocol remains the same as the original protocol i.e. distributing and measuring polarization states in conjugate bases, and key sifting. However, we will not be performing any error correction or privacy amplification as the focus of the EKPQC is on the key distribution aspect of BB84. Nonetheless, the EKPQC is programmable and the end-user can in fact, write associated programs to perform these functions on the generated keys. Finally, you will also use the final key to encrypt a secret message.

2.2 Shipping List

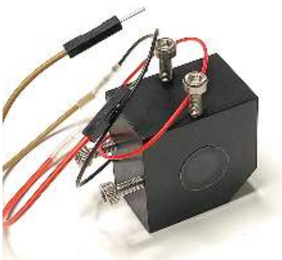
The EKPQC is shipped with the following items:

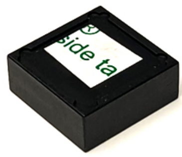
 Alice's Breadboard Qty: 1	 Bob's Breadboard Qty: 1	 Eve's Breadboard Qty: 1

 Tap Beamsplitter base Qty: 1	 High Speed Servo Qty: 4	 This manual Qty: 1
 2 m USB A to B cable Qty: 6	 9V DC supply Qty: 3	
 Adruino UNO Qty: 6	 Arduino UNO shield with solderless bread-board Qty: 6	

 Bag of electronics components	 Bag of screws	 Box of small parts
 Acrylic cover for Eve Qty: 1	 Acrylic cover for Alice and Bob Qty: 2	 Acrylic cover for NPBS Island Qty: 1

In the Box of small parts there are

 Laser mounted in tip tilt holder Qty: 1	 Laser mount base Qty: 1	 Quarter waveplate mounted in rotation mount Qty: 1
 Rotation mount base Qty: 1	 Motor stand Qty: 8	 Motor base Qty: 4
 Detector mounted in holder Qty: 3	 Detector mount base Qty: 3	 Detector hat Qty: 3

 Beam Splitter Qty: 2	 BS platform short Qty: 1	 BS platform short baseplate Qty: 1
 BS platform tall Qty: 1	 50 mm diameter linear polarizer Qty: 4	

In the Bag of electronic components there are:

 1 kΩ wired resistor Qty: 2	 47 Ω wired resistor Qty: 2	 1N4148 Diode Qty: 3
 TSAL7400 IR LED Qty: 3 (1 spare)	 TSOP38238 IR receiver Qty: 2	 BC547 NPN Transistor Qty: 2
 200 kΩ Potentiometer Qty: 3	 Jumper wires 10 cm male-male Qty: 45	

In the Bag of screws there are:

 M3 $\times$ 6 Qty: 12	 M3 self tapping for plastic Qty: 9	 M3 $\times$ 12, washer and spring washer Qty: 4
 M4 $\times$ 6 Qty: 41	 M4 $\times$ 8 Qty: 46	

2.3 Assembly tools

The following assembly tools are required but not provided by S-Fifteen:

- Size 3 mm hex key
- Size 2.5 mm hex key
- Phillips size #2 driver
- Tweezers
- Small pliers (useful to have but not essential)

2.4 Software tools

For installation of the EKPQC software programmes, a computer with a relatively recent operating system is required. The following base software are required:

- Python 3.6+ or higher[1]
- pip package installer for python
- Git 2+
- USB connection capable of enumerating a Communications Device Class[2]
- Arduino IDE

Please refer to the instruction manual, guide or help forum of the specific software on how to install these before continuing with the EKPQC software installation.

2.5 Operating Conditions

The EKPQC is relatively robust and can be operated in a temperature range of 5–50 °C. The relative humidity should be non-condensating to avoid moisture onto the optical elements.

If unused for long period of time, it is recommended to keep it covered with the provided cover with the power and USB cable disconnected.

[1]The EKPQC has only been tested extensively with python 3.9, mileage for lower versions of Python may vary.

[2]Any modern Windows, Linux or Mac OS will have this by default.

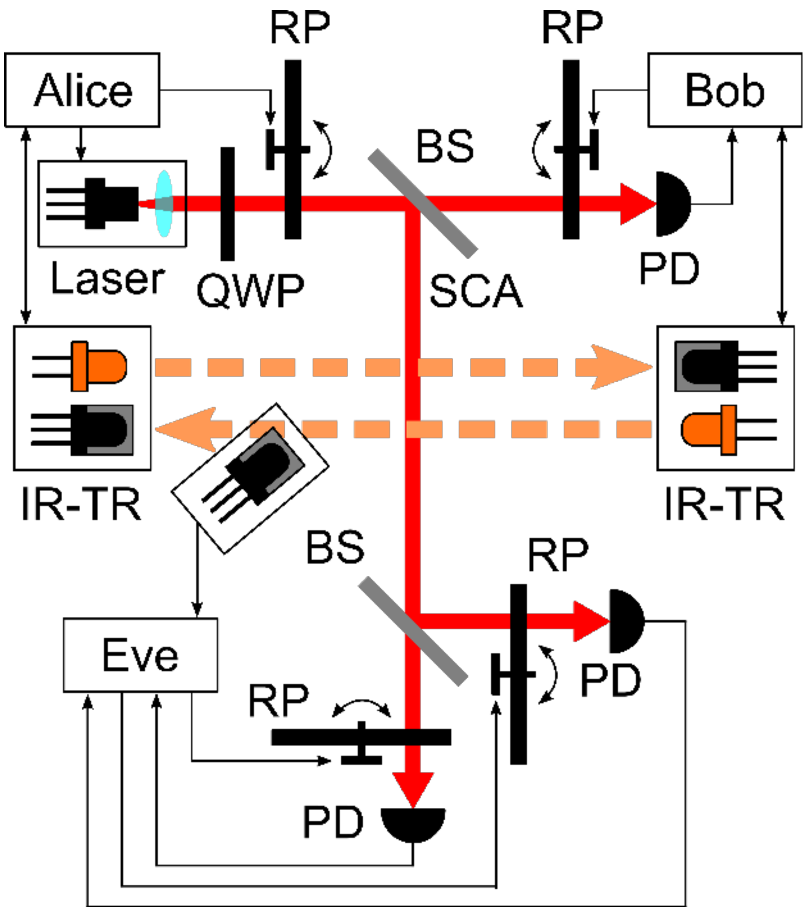


Figure 2.2: BB84 setup – quantum channel: Alice encodes a string of bits using different polarization choices set by her rotatable polarizer (RP). The quarter-wave plate (QWP) makes the laser light circularly polarized such that the output from Alice's polarizer has equal intensity for all linear polarization choices. Bob projects Alice's photons into different polarization bases, and measures the corresponding intensity with a photodetector (PD).
Classical channel: Using infrared transceivers (IR-TR), Alice and Bob communicate the matched bases and the encrypted message.
Side channel attack (SCA): Using a beam splitter (BS), Eve intercepts and measures some of Alice's photons in two different bases simultaneously. As Eve's basis choice is *a priori* not aligned to Alice and Bob's, she may not be able to distinguish between polarization states optimally. However, by measuring in two basis choice simultaneously, she improves her ability to identify distinct polarization states. She also intercepts messages from the classical channel.

3 Software installation

The software for operating the EKPQC is provided online at S-Fifteen's Github[1] repository. This repository contains the firmware for the Arduino's, python scripts for interacting with the EKPQC to demonstrate QKD via BB84 protocol. Installation of the repository can be easily done via git.

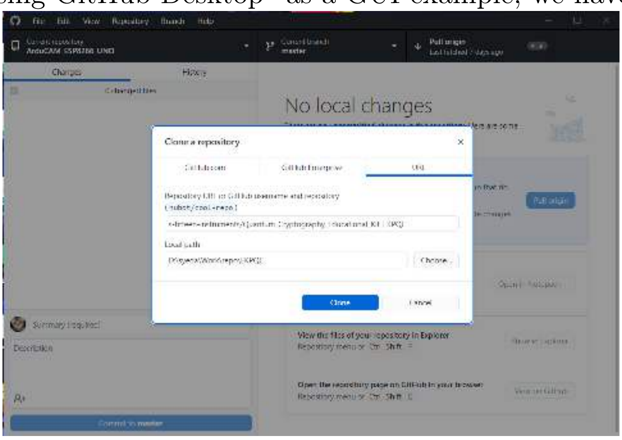
> ☞ It is recommended, but not absolutely necessary to install the software first before assembling the hardware because in the initial assembly/tuning, we need to switch on the laser and rotate the linear polariser to tune the Quarter waveplate.

Software download

The EKPQC software needs to be downloaded to the local computer. The simplest way of doing this is to clone the git repository locally.

3.1 Windows OS

Using GitHub Desktop[2] as a GUI example, we have



[1]https://github.com/s-fifteen-instruments/Quantum_Cryptography_Educational_Kit_EKPQC
[2]https://desktop.github.com/

or on the command line,

```
C:\> git clone https://github.com/s-fifteen-instruments/
Quantum_Cryptography_Educational_Kit_EKPQC path/to/directory
```

In this repository, there contains the `programs` directory which is divided into the different pedagogical topics of the EKPQC during the instruction. Additionally, in the first two topics(directories), `1_Classical` and `2_QuantumKey`, there are the Arduino C++ codes that needs to be uploaded once to the Arduinos.

3.1.1 Upload Arduino code[3]

The Arduino's are supplied without any onboard fuctional programs. They need to be uploaded using Arduino IDE. Separate the 6 Arduinos into

- 2 for Alice and Bob's Quantum channel
- 1 for Eve's Quantum channel
- 3 for everyone's classical channel.

To avoid confusion with too many Arduino's plugged in, it is recommend to upload the Arduino codes one at a time. Start Arduino IDE software.

Alice and Bob Quantum

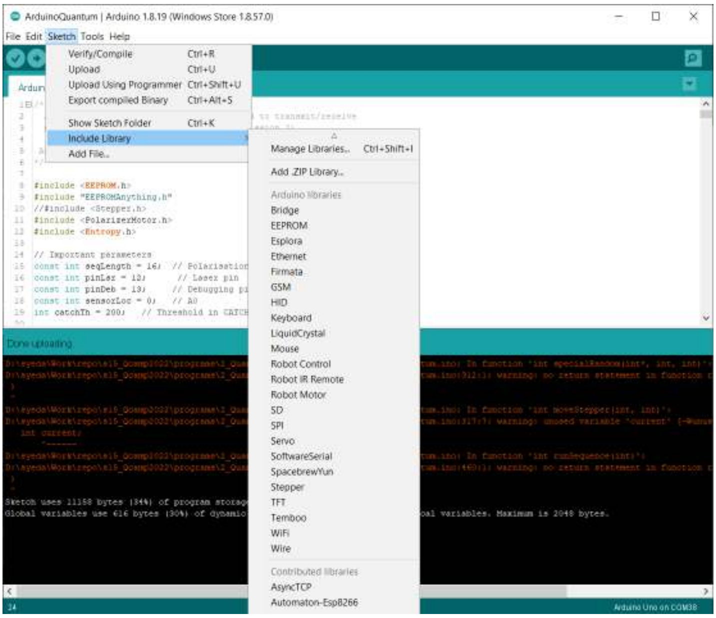
The C++ code for Alice's and Bob's Arduinos are identical. Before compiling and uploading the programs on the Arduinos, the supplied libraries need to be installed once first. They are the `Entropy, IRremote`, and `PolarizerMotor` libraries in the `Arduino_Libraries` directory.

> STOP It is recommended to install these directories to the path instead of to the individual sketches.

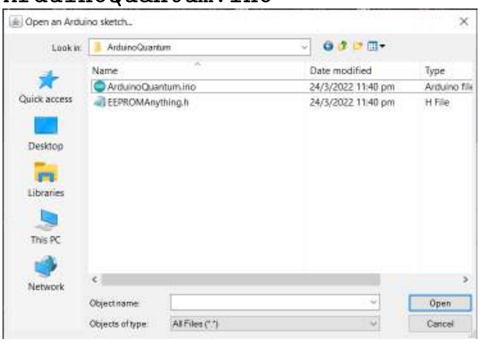
To do this, use the `Sketch->Include Library-> Add .ZIP Library...` to add it to the path that all future sketches will have access to them. Unfortunately, this has to be done one library at a time[4].

[3] This can also be done after the Arduino's are assembled to their respective breadboards to avoid any unintentional switching. Refer to section 4.1.1

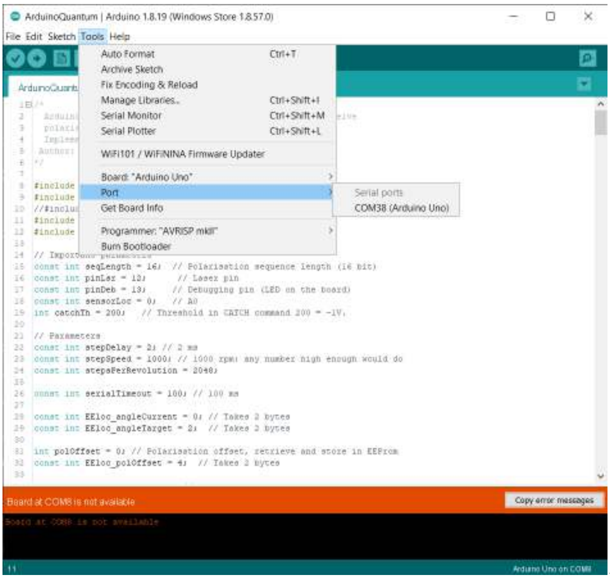
[4] It is possible to copy all three directories to the Arduino library path if the user is able to identify it on his/her machine. Refer to this forum. (Link in PDF)



Next, open a sketch and navigate to the directory where the software was cloned and into `programs/2_QuantumKey/ArduinoQuantum/` directory. Open `ArduinoQuantum.ino`



Connect one Arduino to the computer with the provided USB cable. Once identified by the OS, the Arduino IDE will identify the device. Select the device under `Tools->Ports`



.

Also choose the Arduino UNO as the board type under `Tools->Board->Arduino AVR Boards->Arduino UNO`.

Next upload the code via `Sketch->Upload` or clicking the second icon (right arrow). After some messages, it will say "Done uploading". To check open the serial monitor via `Tools->Serial Monitor`. Set the baud rate to 38400. Then type `HELP␣` followed by enter. If the upload was successful, then the Arduino will reply with the list of implemented commands.

> ☞ If the baud rate is wrong, garbled messages will be received instead. Set to 38400

> Two Arduinos needs to have this `ArduinoQuantum.ino` uploaded, one for Alice and one for Bob.

Plug out the current Arduino and repeat with a new Arduino for Bob.

Eve Quantum

For Eve's quantum Arduino, everything is the same, except for the sketch, which is located at `programs/4_HackTools/ArduinoEve/` directory. Open `ArduinoEve.ino` instead. Upload only *one* Arduino.

Classical

For everyone's classical Arduino, load the sketch located at `programs/1_Classical/ArduinoClassical/` directory. Open `ArduinoClassical.ino`. Make sure that the `IRremote` library has been installed as earlier in section 3.1.1 Upload *three* Arduinos.

Python

It is highly recommended to install the python requirements into a separate virtual environment to avoid conflicts with pre-existing packages. In Windows, this can be done by running `start.bat` by clicking the file. This will end with a command prompt with the following prompt.

```
(env) path/to/directory>
```

The software is now installed, ready to be used. To exit the virtual environment, type

```
(env) path/to/directory> deactivate
```

If not using any virtual environments, simply installl the required packages using `pip3`

```
path/to/directory> pip3 install -r ./requirements.txt --no-warn-script-location
```

☞ There will be some conflicts if `serial` is installed because `serial` and `pyserial` is imported the same way in python. To resolve this uninstall `serial` via `pip`

3.2 Linux

In Linux, it is easier to do everything from the terminal. Start your favourite terminal. Navigate to directory where you want to clone the repository to. In this example we clone to the default named directory in the `home` directory.

```
:~/code> git clone https://github.com/s-fifteen-instruments/
Quantum_Cryptography_Educational_Kit_EKPQC EKPQC
```

Upload Arduino code

Uploading the Arduino code using the Arduino IDE is exactly the same as that for windows. Refer to section 3.1.1.

Python

If several different versions of python3 are installed, it is recommended to run in a separate virtual environment. Refer to the OS manual or help in order to install the `python3-venv` package.

```
:~/EKPQC> source start.sh
```

Otherwise, if only a single python version is installed, this is not necessary. Simply install required packages using `pip`. Other additional requirements for python3-tk and/or pyQT5 may also be required depending on how python3 was installed.

```
:~/EKPQC> pip3 install -r ./requirements.txt --no-warn-script-location
```

4 Hardware assembly

The complete hardware assembly can take up to 1.5 hours. They can be divided to mechanical assembly and assembly of the electronic wires and components. It is also recommended to have a clean working space of about 0.5 m by 1.0 m to facilitate assembly.

4.1 Mechanics

The items needed for this assembly are the 3 breadboards, tap beamsplitter base, 4 high speed servos, 3 Arduino UNOs, box of small parts and bag of screws.

4.1.1 Arduino

First, take the three Arduino's that have been flashed and attach them to their respective breadboards. The plastic bases are secured to the breadboard with 4 M3×6 as shown below. The breadboard for Eve has 2 options to put the Arduino, choose ony one.

Once the plastic base are fastened, put the Arduino UNO, ensuring that the Eve Quantum Arduino goes to the Eve breadboard. Each Arduino is secured by 3 M3 self tapping screws for plastic.

☞ There are 4 screw holes on the Arduino UNO, but only 3 will fit with the proper clearance.

Next, take three Arduino UNO shields with solderless breadboard and connect them to the three Arduinos. Ensure that the pins line up to the correct pin assignments.

The next items we will install individually first before assembling all of them to the breadboard at the end.

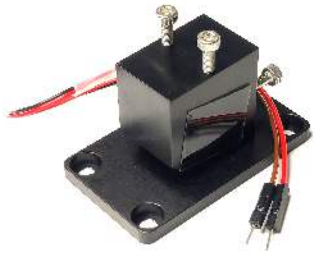
4.1.2 Light source

☞ All screws that are used from under the baseplates to the items above them are secured with M4×6 screws. For securing the baseplates to the breadboards and the Servo to the Servo stand, use M4×8 screws.

The laser that is used as the light source comes mounted in a tip-tilt holder for fine adjustment of the beam via 4 M3×12 screws. Also on the holder, are 2 plunger screws in the diagonal direction that provides the restoring force. If these screws needs adjustment (too deep or too shallow), a 0.9 mm hex driver is needed. There is a linear polarizer aligned and pasted on the holder that cleans up the laser polarization with maximum transmission[1].

Secure the tip-tilt holder onto the laser mount base with 2 M4×6 screws.

[1] If the laser gets accidentally rotated in the holder, the output laser power will dip slightly.



4.1.3 Quarter waveplate

The quarter waveplate(QWP) converts the linearly polarized laser light into circularly polarized light. It is mounted in a rotation mount with its axis aligned at ±45° to the laser polarization. If the QWP needs to be rotated, the rotation mount needs to be unlocked with a 1.5 mm hex driver.

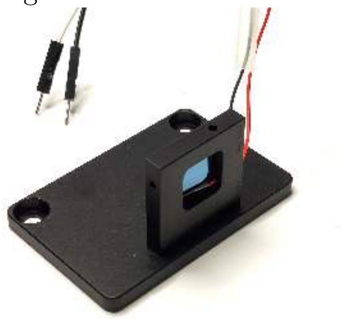
Secure the rotation mount onto the rotation mount base with one M4×6 screw. Ensure that the long edge of the base is in the same line as the plane of the QWP.[2]

[2]The recess for the base (in version 1) is slightly smaller than the rotation mount, thus making it slightly tilted. This is not an issue for operation.

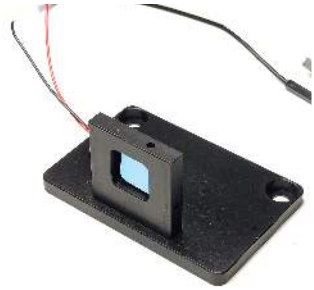
4.1.4 Detectors

The detector in used is 9.6 mm by 9.6 mm and mounted in a holder with leads attached. Attach the detector to the detector mount base with 2 M4×6 screws. The detectors can face either way, but try to make then as parallel to the edge as possible.

Repeat for the other 2 detectors. One of the detectors will have it's wire exiting on the other side for the other two.



Qty: 2



Qty: 1

4.1.5 Beamsplitter holders

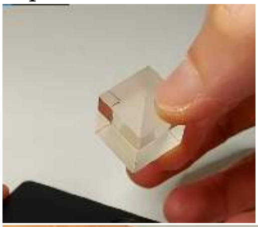
☞ Watch the Tap BS Assembly Video on YouTube

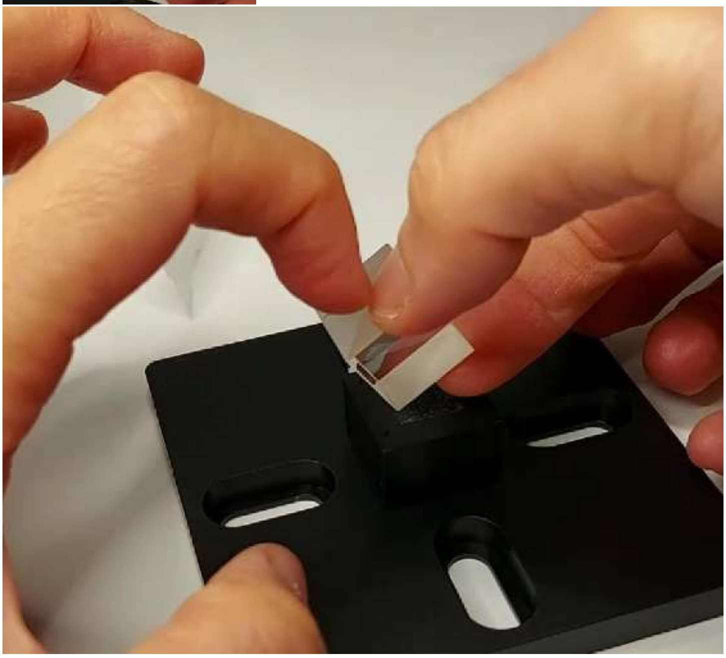
There are 2 beamsplitter (BS) holders, one for Eve to tap the quantum channel, and the other for her to measure in both polarization bases simultaneously.

> STOP The short BS platform goes with the BS platform short baseplate (rectangular) and the tall BS platform goes with the tap BS base (square).

Secure the platforms on their baseplates with one M4×6 screw each. Ensure that the edges are parallel to the sides.

Next, remove the double sided tape from the platforms with a tweezer. Then take the Beamsplitter cube from the wrapping carefully, handling on the top and bottom non-reflective sides and secure on the platform with the double sided tape. Once adhered to the BS, the tape should be visible through the BS sides.





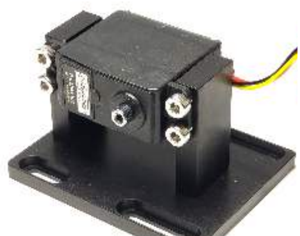
Repeat for the other platform.

4.1.6 Servos

☞ Watch the Servo Assembly Videos here on YouTube

Assembly of the Servos is the trickiest. There are two polarities needed, 3 of one kind and one of the other. This is to ensure that the Servo axis can be aligned to the set of screw holes. The simplest way to get the correct alignment is to get the wires of the Servo at the same side as the long slot of the base.

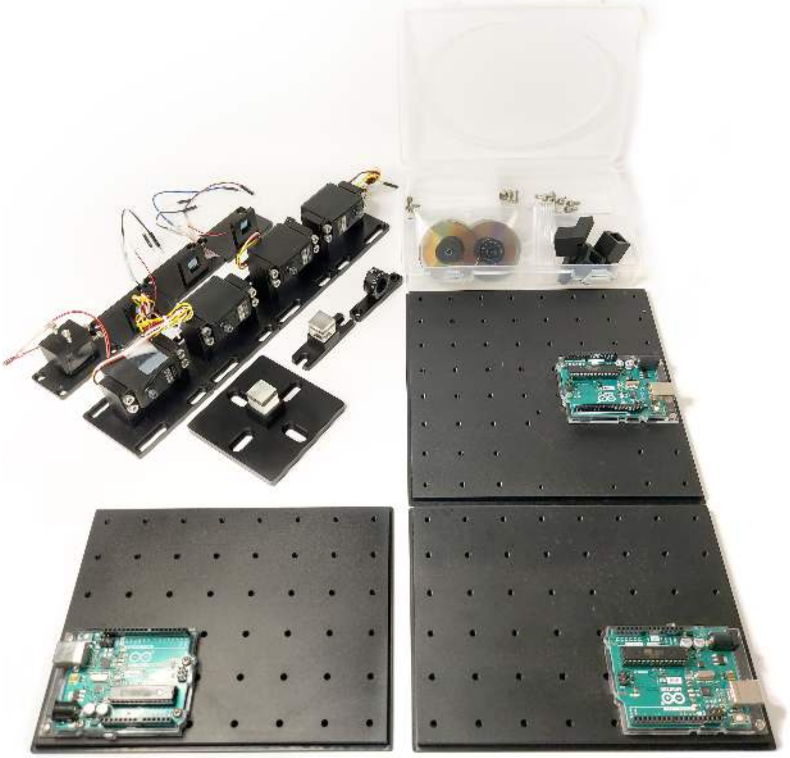
Qty: 3



Qty: 1

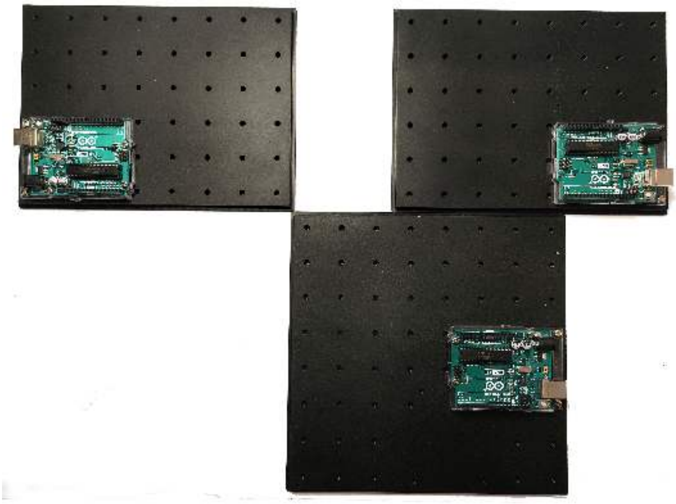
First screw in 4 M4×8 screws from the Servo to the Servo stand. Next, take a servo base and place the long slot in the same side as where the wires exit the Servo. Screw from underneath using 4 M4×6 screws. Check the polarity of the assembled Servo.

Assemble 3 more with the appropriate polarities. *Do not* attach the 50mm linear polarizers yet. Now, only the linear polarizers, and Detectors hats are left in the box for small parts.

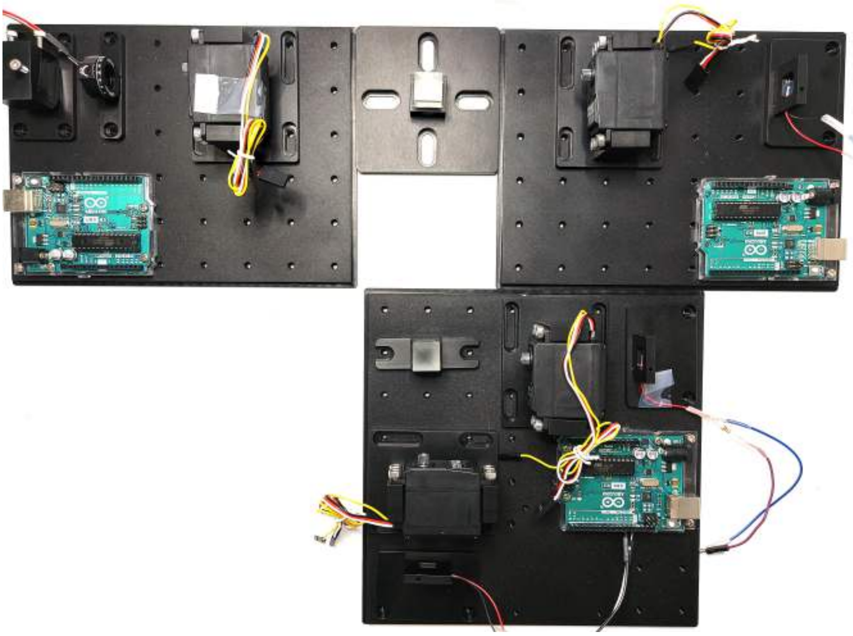


4.2 Putting together

Although not necessary, we recommend laying out the breadboards as shown below to easier follow the guides in the manual



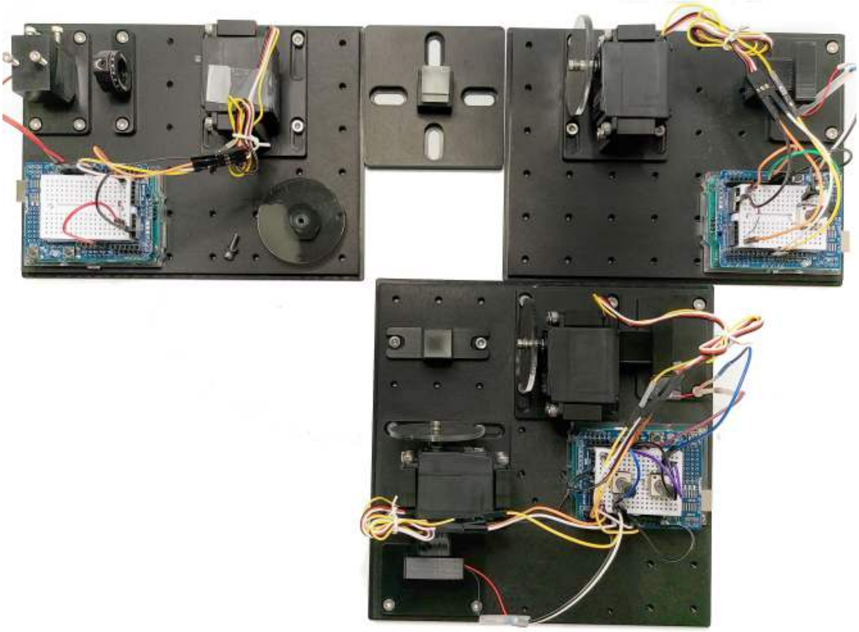
Next arrange the items as shown below, taking note of the two opposite polarities of the servo and detector on Eve's breadboard. Also align the BS correctly so that it will reflect the beam to the correct paths.



Secure all with M4×8 screws. Attach 3 linear polarizers to Bob and Eve with the M3×12 screw, washer and spring washer. Leave out the one for Alice for now.

☞ Remember to use the washers and spring washers provided when tightening the screws or you may risk damaging the polarizer acrylic.

Once everything is secured, it will look something like the image below. Refer to the next section on connecting the electronics. Once the electronics are connected, perform the visibility test as described in section 5.4



4.3 Electronics

The electronic components needed can be found in the bag of electronic components. For the quantum channels, only the 1N4148 diodes, 200kΩ potentiometers and jumper wires are needed.

For the quantum channels, they all have a few similar connections for the servo. The Servo has 4 female connections, which are Ground(Black), Vin(Red), Signal(White) and Feedback(Yellow). It is simpler to use male-male jumper wires to connect them to the Arduino. The Signal and Feedback pins goes to pins 8 and 9 of the Arduino UNO. For Eve who has two Servos, the second one

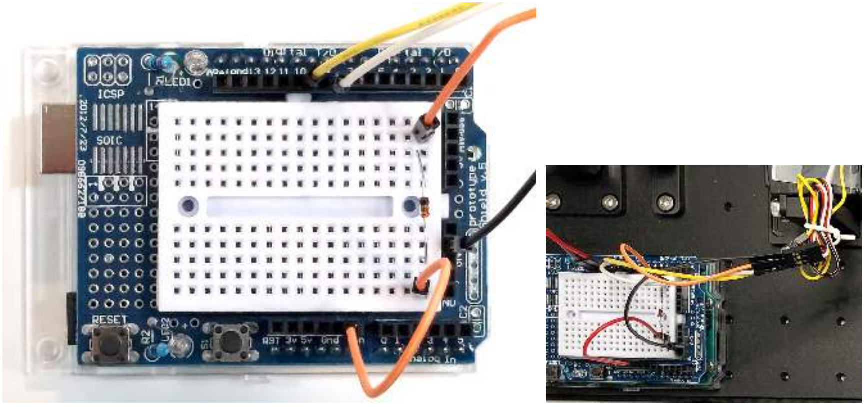


Figure 4.1: Left: Connections for the 1N4148 diode to the Arduino Uno and the Servo. Right: Connecting the Servo to the breadboard with jumper wires.

goes to pins 4 and 5. The Vin goes to the cathode of the 1N4148 diode, with it's anode pin connected to the Arduino Vin. The Ground goes to the GND pin. An elegant simple connection layout is shown in the image in figure 4.1.

Additional connections for each of the specific Quantum channels are explained in the following sections, but always beginning with this minimum layout.

4.3.1 Alice

Alice has only one the laser to connect as seen in figure 4.2. These goes to GND pin for the negative terminal and pin 12 for the positive terminal.

4.3.2 Bob

Bob has the detector and the potentiometer to connect. The detector has a 1 kΩ resistor soldered to prevent excessive current flow when reversed biased with the potentiometer turned to minimum resistance. Connect the potentiometer securely into the breadboard. The positive terminal of the detector goes to the first pin of the potentiometer, while the negative terminal of the detector goes to 5V. The first pin of the potentiometer is also connected to the Analog In pin A0. Finally, the second pin of the potentiometer is connected to GND. The last pin is unconnected. These connections are shown in figure 4.3

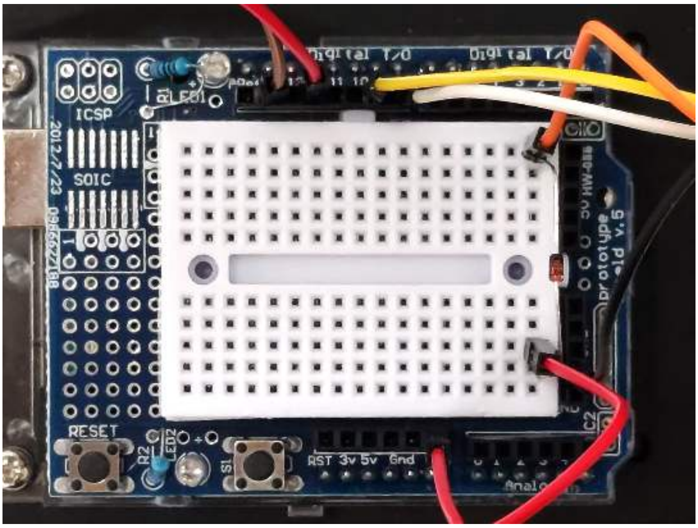


Figure 4.2: Alice's connections for quantum channel. Negative terminal of the laser is the brown wire and the positive terminal is the red wire to pin 12.

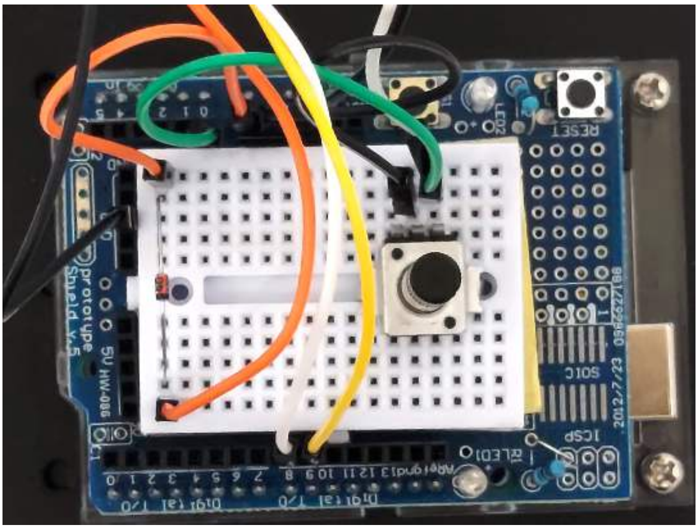


Figure 4.3: Bob's connection for the quantum channel.

4.3.3 Eve

Eve has the same connections as Bob for one set of his Servo and detector. This first set of the Servo signal and feedback goes to the pins 8 and 9 with the detector output going to pin A0.

The other set of the Servo signal and feedback goes to pins 4 and 5 with the detector output of the beam going through this Servo going to A1. The 5V, Vin and GND pins are the same. The recommended layout of the connection is shown in figure 4.4

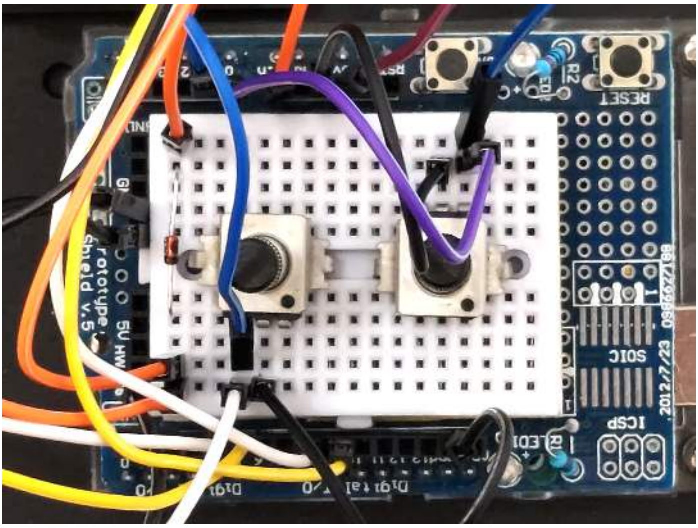


Figure 4.4: Eve's connections for the quantum channel, with two potentiometers. The one on the right is identical to Bob's and connected to analog pin A0. The one on the left is the extra one connected to analog pin A1. The extra Servo that goes with this is connected to pins 4 and 5.

4.3.4 Classical Arduino

The circuit diagram for the Classical channel is shown in Figure 4.5. The classical channels of the Alice, Bob and Eve has a receiver each. For the sender, only Alice and Bob has have them.

Sender

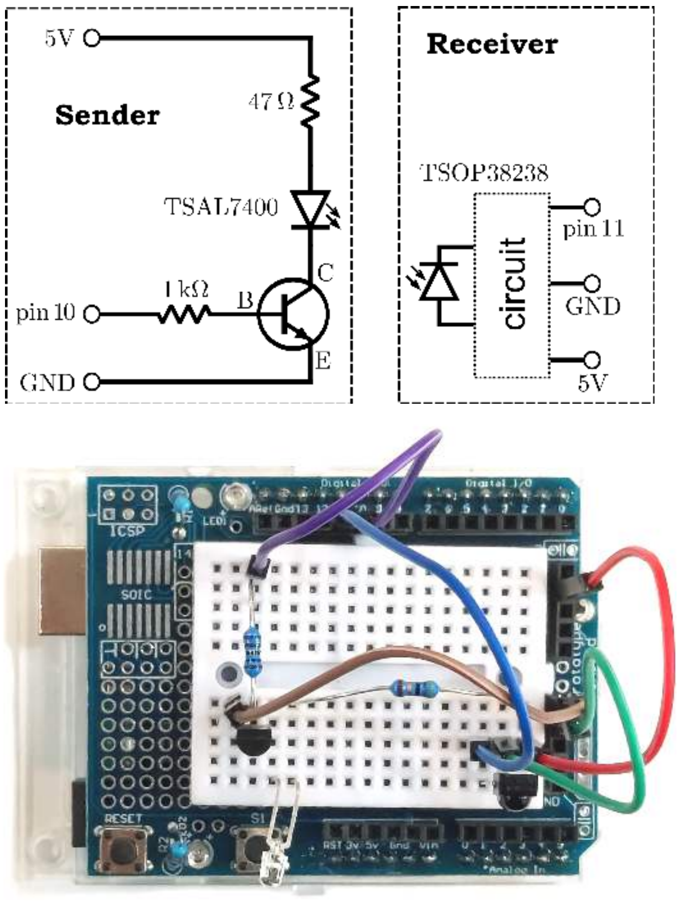


Figure 4.5: Top: Circuit diagram for the classical channel. Bottom: Electronic layout for the Classical Arduino on the breadboard

☞ Note that the transistor needs to be connected with the correct polarity. An incorrect connection may result in damage to the component.

The sender has an additional NPN transistor to drive the IR LED such that it can draw a larger (100 mA) current from the 5V pin since the digital pins of the Arduino UNO is only limited to 40 mA.

Receiver

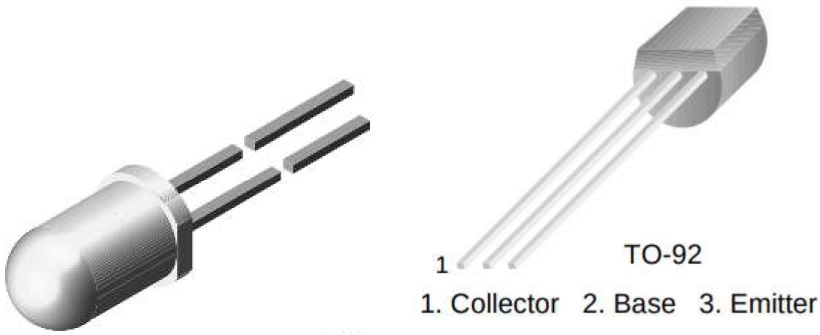


Figure 4.6: (Left) IR LED: Longer leg (anode) is to be connected to a higher potential. (Right) BC597 transistor, in a common TO-92 package, used to supply current to the IR LED.

☞ The IR receiver module may be damaged if connected in the wrong polarity.

Figure 4.7 shows the IR receiver module (TSOP38238). The module consists of an IR photodiode with a built-in demodulation circuitry. The output (OUT) of integrated circuit / photodiode module is directly connected to Arduino pin 11. The module is biased across the V_s and GND pins by connecting them to the 5 V and GND pins of the Arduino board, respectively.

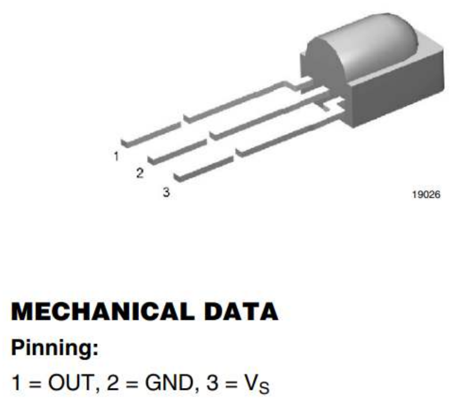


Figure 4.7: IR receiving unit. V_s connects to the voltage supply of 5 V,. GND can be assigned to the Arduino ground pin. The IR signal OUT pin is measured on using Arduino pin 11.

5 Setup and Operation

In this chapter the setup and operation of the EKPQC will be described. It will be concise and unexplanatory. For further explanations, please refer to the Student guide.

5.1 Before Starting

Place Alice's and Bob's breadboard on the same level, up to 5m apart. Ensure that they have a clear line of sight between them. If Eve is to be used, then put in the Tap beamsplitter in between the path and Eve's breadboad also on the same level to the other two.

Connect the USB cables and the 9V DC supply to the Arduinos and switch them on. It is possible to connect all the Arduinos to one computer since they will all identify as a unique serial port. The Arduinos can be be communicated to via the serial connection directly(Refer to section 5.7 for the commands). However, we will use high-level python commands to do the setup.

5.2 Serial port identification

First, we need to know the serial port which the Arduinos identify themselves with. to do this, run the following command

```
:~/EKPQC> python3 programs/list_arduinos.py
.
.
Ports identified!
['/dev/ttyACM3 (Classical)', '/dev/ttyACM2 (Classical)', '/dev/ttyACM1 (Quantum)',
'/dev/ttyACM0 (Quantum)', '/dev/ttyS1', '/dev/ttyS0']
```

5.3 Alignment

5.3.1 Alice and Bob

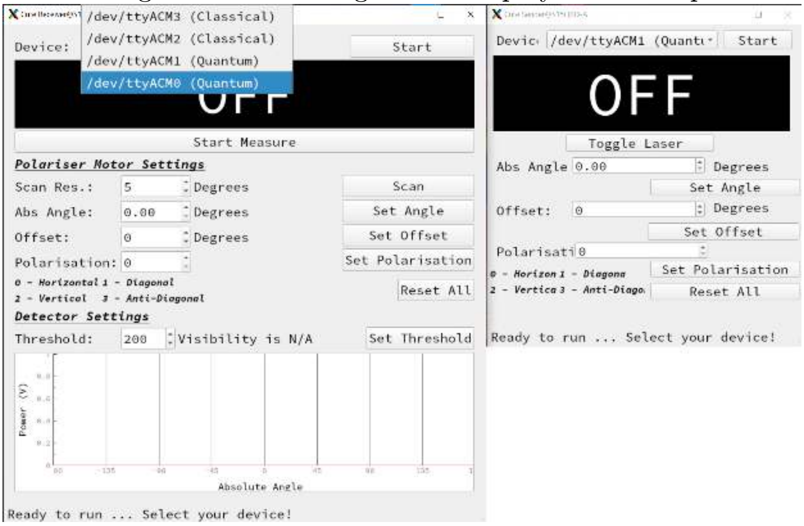
Next, we run the `runSender.py` and `runReceiver.py` GUIs with

```
:~/EKPQC> python3 programs/2_QuantumKey/AlignmentGUI/runReceiver.py
```

on one side and

```
:~/EKPQC> python3 programs/2_QuantumKey/AlignmentGUI/runSender.py
```

Choose the correct serial ports, click start. Once connected, toggle the laser on at the Sender and start measurement at the Receiver. This is done by clicking the button right below the large black display on the respective GUIs.



Align the laser beam such that it is going to Bob. This can be done coarsely by tilting the breadboard and finely with the screws on the tip-tilt mount. If the laser is hitting the detector, there should be a voltage reading, otherwise, tune the potentiometer to get approximately 4 V.

5.3.2 Eve

Eve performs the polarization state measurement simultaneously using two bases: first, by splitting the beam equally with a 50:50 beam-splitter, then by measuring the trans- mitted intensities through two linear polarizers, each oriented at different angles, using two photodiodes.

Here's how to ensure that Eve's linear polarizers are measuring at different angles. Set up Alice and Eve so that Alice's laser goes through the beamsplitter in the correct direction. Alice's polarizer must be attached. Turn on the laser[1].

The markings on the beamsplitter's top face indicate which direction the beam will split. Section 4.2 shows one possible way to orientate Alice and Eve

[1] See section 5.3 for instructions to turn on laser

(Bob's terminal is not necessary for this part). If this setup is used, the laser beam will pass through two beamsplitters, and the resultant intensity incident on Eve's detectors will be similar to an actual run of the experiment.

To readout Eve's detector values, you will need to use a serial terminal such as Arduino IDE's serial monitor. Issue the command `"volts?  "` with a whitespace at the end[2], the arduino will reply with two numbers indicating the strength of incident light. These numbers run from 0-1023, and roughly correspond to a 0-5 V reading on the detector. If you find that the readings are consistently near 0 or 1023, adjust the potentiometers accordingly.

```
>volts?
89 101
```

Rotate Eve's polarizers individually through 180°while repeatedly issuing the "volts? " command. You may rotate the polarizers by hand (gently) or via the "setang1 " and "setang2 " serial commands[3]. For example, "setang1 150 " rotates one of Eve's polarizers to 150°. If you choose to use the serial commands for rotation, please ensure that the 9V power supply is plugged into Eve's arudino and turned on.

```
>setang1 150
OK
```

You will now have a rough range of values for the two detectors. The two ranges may not be identical - that is fine. Next, rotate one of Eve's polarizers so that the voltage reading is at the maximum of its own range, and the other so that the reading is in the middle of its own range. This will ensure that Eve's polarizers are roughly 45°apart[4].

5.4 First visibility measurement

The quarter wave plate axis and that of the laser beam are shipped aligned to give a circularly polarized output. Nevertheless, it is good to check it first after assembly as a slight tilt in the direction may affect the performance.

With no linear polarizer in Alice's setup, the laser is circularly polarized. If you are self-assembling the kit, you may leave Alice's polarizer unattached for the beginning of this section. If your kit is already assembled, you may remove Alice's polarizer. This can be done with a 2.5 mm hex key.

[2]The whitespace is important for indicating the end-of-line (EOL).
[3]View section 5.7 for a full list of arduino serial commands.
[4]There is no need to be too precise with the exact angles

☞ When (re)attaching Alice's polarizer, remember to use the washers provided. The polarizer may crack if you tighten the screws without the washers.

Bob should measure a visibility of < 0.15 when he does a scan. If the visibility is > 0.2 unlock the rotation mount of the quarter waveplate with a 1.5 mm hex key and rotate slightly ($<1°$) to make the measurement in between the maximum and minimum values and do the scan again. Repeat until visibility is < 0.15. (Check subsection 4.1.3 for details on the assembly of the quarter waveplate mount.) Once done, attach Alice's linear polarizer[5] and set the offset and angle

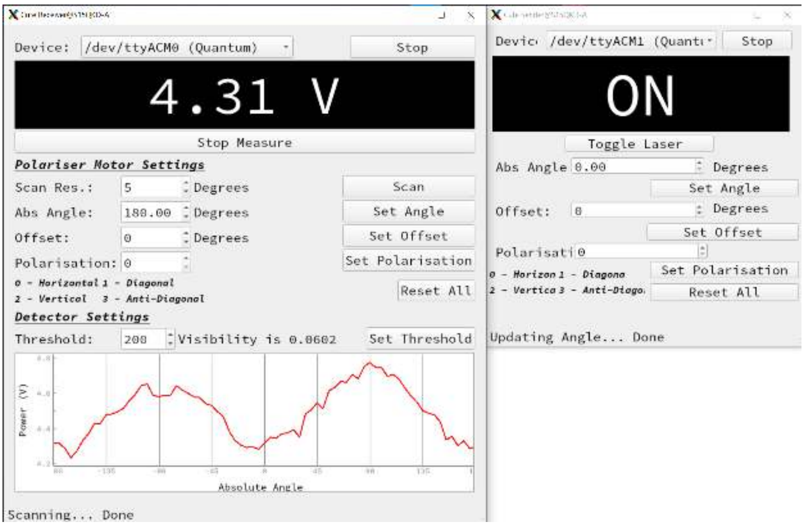


Figure 5.1: Visibility of the circularly polarized light is 0.06 here.

to 0°. Also set her polarization to be 0 (Horizontal). Now when Bob does a scan, he should get a large visibility of > 0.95 with a peak at some angle. Set his offset to be at the peak of the scan. This value is saved on the EEPROM of the Arduino and survives any power cycles. Now Alice's and Bob's definition of basis coincide[6].

Close the GUIs and start the calibration codes, starting with the receiver

[5]The beam might point away if the tip-tilt stage screws is accidentally pushed. Ensure that beam still falls on the detector after.

[6]For Diagonal and Anti-Diagonal, rotation direction is important. This is why both motors have their axes pointing in the same direction, which in the incoming beam.

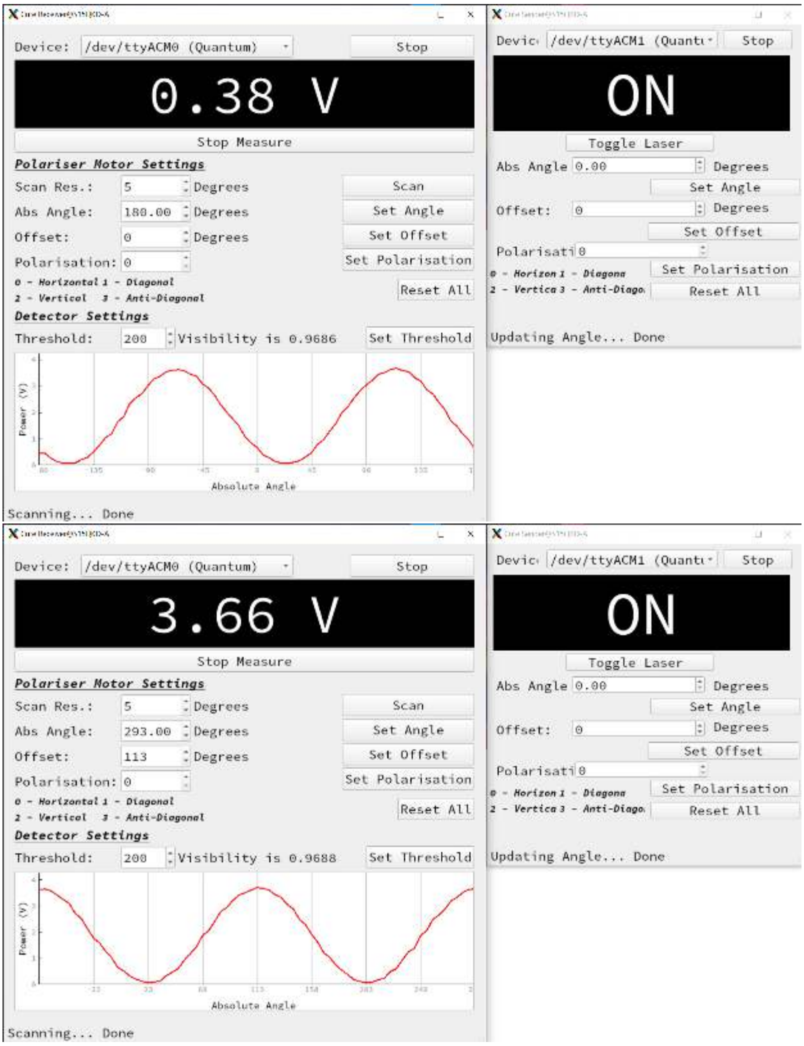


Figure 5.2: Visibility measurements before and after adding the offset for Bob.

```
:~/EKPQC> python3 programs/2_QuantumKey/recv_calibrate.py --serial /dev/ttyACM0
```

```
:~/EKPQC> python3 programs/2_QuantumKey/send_calibrate.py --serial /dev/ttyACM1
```

The receiver then will report the measurements done,

```
Listen for incoming signal...
Measurement done
```

```
                    Receiver
           |  H  |  D  |  V  |  A  |
       | H | 783 | 368 |  12 | 439 |
Sender | D | 347 | 708 | 342 |  13 |
       | V |  13 | 315 | 717 | 405 |
       | A | 468 |  16 | 294 | 761 |

  The mean is 375.062
  Signal degradation is 0.095
```

The signal degradation should be < 0.2. Note too the mean value (375 in this example) as that is the threshold that will be used in the next part.

5.5 Classical

Next we try out the classical channels on both sides to confirm that they are working. This can be done via the `chatting.py scripts` with the appropriate serial ports.

```
~:/EKPQC> python3 programs/1_Classical/chatting.py --serial /dev/ttyACMX
Opening the serial port...
Done

Qcumber ChatBox v1.00
To exit the program, use Ctrl+C

You are now in sending mode. To change to listening mode, press ENTER.
Write the message you want to send below:
```

The side that is listening can receive the message that is transmitted.

```
You are now in listening mode.
Waiting to receive the message...

--- START OF TEXT ---
This is just a test
--- END OF TEXT ---

You are now in sending mode. To change to listening mode, press ENTER.
Write the message you want to send below:
```

Eve should also test her classical channel by running the `listener.py` script with her Classical Arduino on the serial port to listen to the chatting of Alice and Bob.

```
~:/EKPQC> python3 programs/4_HackTools/Classical_Listener/listener.py \
  --serial /dev/ttyACMX
```

5.6 Operation

Finally, we run the automated key generation script `recv_32bitQKD.py` and `send_32bitQKD.py`. This script will need the serial ports of both classical and quantum channels and also the threshold value for Bob. An partial example output from both parties is as follows. Note that, in the steps below, we describe Bob on the left column since he first initiates the process by being ready to receive.

<table>
<tr><th>Bob (receiver)</th><th>Alice (sender)</th></tr>
<tr><td><pre>programs\3_QKDComm>
python recv_32bitQKD.py
--cserial /dev/ttyACM2
--qserial /dev/ttyACM0
--threshold 375

Opening the serial port...
Done

Hi Bob, are you ready?
Let's make the key!

Testing the public channel...
hex_string: 7070741
state: 0
hex_string: 54657374
state: 1
Test
Alice sends Test
You reply --OK!!--
OK!!

Public channel seems okay...
Sending the quantum keys...

Attempt 1
Generating random basis choices...
OK</pre></td><td><pre>programs\3_QKDComm>
python send_32bitQKD.py
--cserial /dev/ttyACM3
--qserial /dev/ttyACM1

Opening the serial port...
Done

Hi Alice, are you ready?
Let's make the key!

Testing the public channel...
You send --Test--
Test
hex_string: 7070742
state: 0
hex_string: 4F4B2121
state: 1
OK!!
Bob replies OK!!

Public channel seems okay...
Sending the quantum keys...

Attempt 1
Generating random polarisation
sequence...
OK</pre></td></tr>
</table>

For each round of communication, 16 pulses are sent and only about half ends up as a final key. As such, several rounds (attempts) are needed to accumulate 32-bit of final key. Once the procedure is completed, both parties should be presented with a final key:

```
DONE. The task is completed.
The 32 bit secret key is (in hex): 2237797c
```

The keys can then be used to encrypta and decrypt messages using the `encrypt.py` and `decrypt.py` scripts before sending though the `chatting.py` scripts, all in the same `programs/3_QKDComm/` directory.

Operation of Eve is more involved and can be done following the student guide. The scripts necessary for her can be found in the `programs/4_HackTools/` directory.

5.7 Low level serial commands

The serial communication settings are listed in Table 5.1 for reference..

Serial Communication Settings	
Baud rate	38400

Table 5.1: Caption

The Arduinos have a set of commands that can be communicated with through the serial connection. We list them in table 5.2 for reference. In normal operation, the python wrappers will use these commands.

Table 5.2: List of available commands

Alice and Bob Quantum Commands	
HELP	Print help statement
SETANG X	Set angle to X (in degrees)
ANG?	Ask for current angle
SETPOL X	Set polarisation to X -> 0(H), 1(D), 2(V), 3(A)
POL?	Ask for current polarisation
SETHOF X	Set angle offset for H polarisation
HOF?	Ask for the angle offset for H polarisation
POLSEQ X..	Set polarisation sequence as X..
RNDSEQ	Set random polarisation sequence
RNDBAS	Set random polarisation basis
SEQ?	Ask for sequence
LASON	Turn on laser
LASOFF	Turn off laser
VOLT?	Ask for sensor voltage
RUNSEQ	Run the sequence (generic)
TXSEQ	Run the sequence (as a sender)
RXSEQ	Run the sequence (as a receiver)
CATCH	Wait for laser light and display time

Table 5.2: List of available commands

TH?	Gets detector threshold for CATCH - 200=1V
SETTH X	Set laser detection threshold (0-1023) - 200=1V
*IDN?	Gets device name (Quantum/Classical)
Eve Quantum Commands	
HELP	Print help statement
SETANG1 X	Set angle to X (in degrees)
ANG1?	Ask for current angle
SETPOL1 X	Set polarisation to X -> 0(H), 1(D), 2(V), 3(A)
POL1?	Ask for current polarisation
VOLTS?	Ask for sensor voltage
SETANG2 X	Set angle to X (in degrees)
ANG2?	Ask for current angle
SETPOL2 X	Set polarisation to X -> 0(H), 1(D), 2(V), 3(A)
POL2?	Ask for current polarisation
*IDN?	Gets device name (Quantum/Classical)
Classical Commands	
HELP	Print help statement
LEDON	Turn on IR LED
LEDOFF	Turn off IR LED
SBLINK	Send blinking feature
RBLINK	Recv blinking feature
SEND X	Send a short message X (4 bytes)
RECV	Receive a short message (4 bytes)
*IDN?	Get device identifier (Classical/Quantum)

6 Maintenance

The EKPQC does not require regular maintenance if kept in a cool dry environment away from direct sunlight. If unused for extended periods of time, it is recommended to cover it with the provided covers.

For any other maintenance issues, please contact us at `info@s-fifteen.com` for assistance.

7 Troubleshooting

Table 7.1: Common problems and troubleshooting suggestions

Problem	**Possible cause**	**Suggestion**
No serial communication	Wrong device name / permissions	Verify that user has appropriate permissions to access the device[1].
	Ports opened by another program	Check that there are no other software/programs blocking the serial port.
Alignment GUI hangs	Problem with Bluetooth `COM` ports blocking the serial ports probe	If possible, disable Bluetooth COM ports temporarily for the GUI alignment probe.
Visibility > 0.15 for only Bob's polarizer in place	Quarter waveplate is not set at the correct angle	With the Bob's polarizer set to the minimum of the visibility curve, rotate the waveplate slightly ($< 1°$) to increase slightly the detector voltage reading to halfway between the maximum and minimum. Measure the visibility again and repeat until minimized.
	Quarter waveplate tilted with respect to the incoming beam.	Check that the beam is horizontal with respect to the base and Alice and Bob are placed on the same height. Also make the beam travel along the set of holes and tilt the quarter waveplate until the reflected beam is very close to the laser.

[1] In Linux, add the user to the `dialout` group. In Windows, check the COM number through device manager–>Ports.

	Linear polarizer on the laser mount fell out.	Ensure that the small circular piece of linear polarizer stuck to the laser mount is still attached.
Visibility < 0.95 for two polarizers in place	Scatter in between polarizers	A scatterer in the beam's path may change the linearity of the beam polarization and hence reduce visibility.
Basis measurements don't match	Beam deviates from detector when polarizer rotates	On a piece of paper placed far away, check the beam position as Alice's polarizer is rotated. The beam should be displaced a maximum of $< 5\,\mathrm{mm}$. If beam displaces more, check that the polarizer is mounted straight with the back reflection to of the laser going close to perpendicular.
	Beam might deviate if the Polarizer plate is warped	Swap polarizers with Bob of Eve who have shorter paths from their polarizers to their detectors.
	Offset set wrongly when aligning bases	Perform the alignment procedures is section 5.4 again and check Bob measures $> 3.5\,\mathrm{V}$ when they set the same polarizations.
Detector not responsive	Beam not in detector	Ensure that the beam falls on the detector. Remove detector hat to check.
	Potentiometer turned too much one way	Rotate the potentiometer until the voltage range is acceptable
	Laser power too small	Check that the laser power is about $0.9\,\mathrm{mW}$ with a power meter if available. Otherwise, try to rotate the laser in the holder until transmission through the polarizer on the tip-tilt mount maximises.

For further assistance, contact us at `info@s-fifteen.com`

8 Specifications

Table 8.1: Device Specifications

Parameters	Values
Laser wavelength	660 nm
Laser power	<1 mW
Software Control	
Physical port	USB 2.0, Type B
Communication	Serial communication via virtual COM port / USB CDC ACM class
Electrical Specifications	
Power Supply	9 V DC, 2.5 mm barrel socket. Europlug
Maximum power consumption	4 W
Physical dimensions	
Shipping Weight	6.0 kg

Bibliography

[1] N. E. Agency. Frequently Asked Questions on Lasers. `https://www.nea.gov.sg/our-services/radiation-safety/lasers/frequently-asked-questions-on-lasers`, 2022. [Online; accessed 5-Mar-2022].

[2] A. N. Utama, J. Lee, and M. A. Seidler. A hands-on quantum cryptography workshop for pre-university students. *American Journal of Physics*, 88(12):1094–1102, 2020.