

## What is QuanClick?

QuanClick is a user-configurable mouse and keyboard code generator.



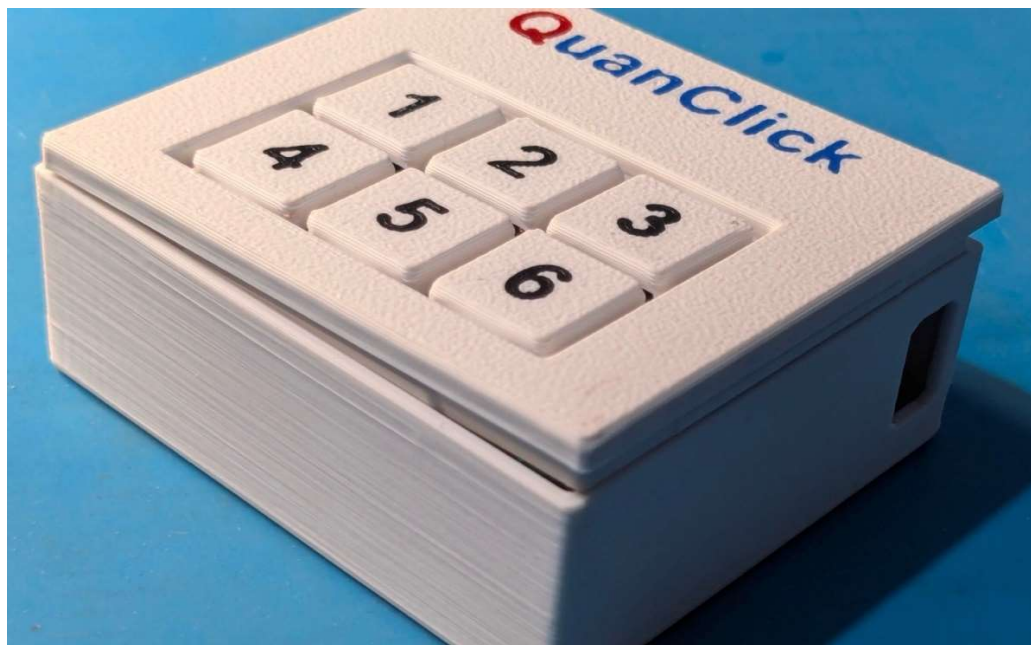
It reacts to presses from a 6-button keypad and sends sequences of mouse, or keyboard, control codes over USB just like a real mouse or keyboard would do.

Because you can send sequences of codes (or macros) you can replace many mouse or keyboard actions with just one button press.

For example, you can program one button to type out your email address with just one click.

QuanClick uses a Raspberry Pi Pico 2 microcontroller that we will refer to as a '**Pico**' throughout this document.

If you choose to use other variants of Pico (such as the original Pico 1) you will need to substitute the UF2 for the board you are using.



QuanClick was designed and built by members of the  
**Quantum Technology Club**

For further details about the Quantum Technology Club, please visit

**[www.quantumtech.club](http://www.quantumtech.club)**

## Assembly Instructions

### Connect the cables to the buttons

Attach the pins from one end of the multiway cable to the pin header on the keypad.

Your multiway cable may not have the identically coloured wires as in this photo - that's not important, just that they are all different colours so you can easily identify them either end.

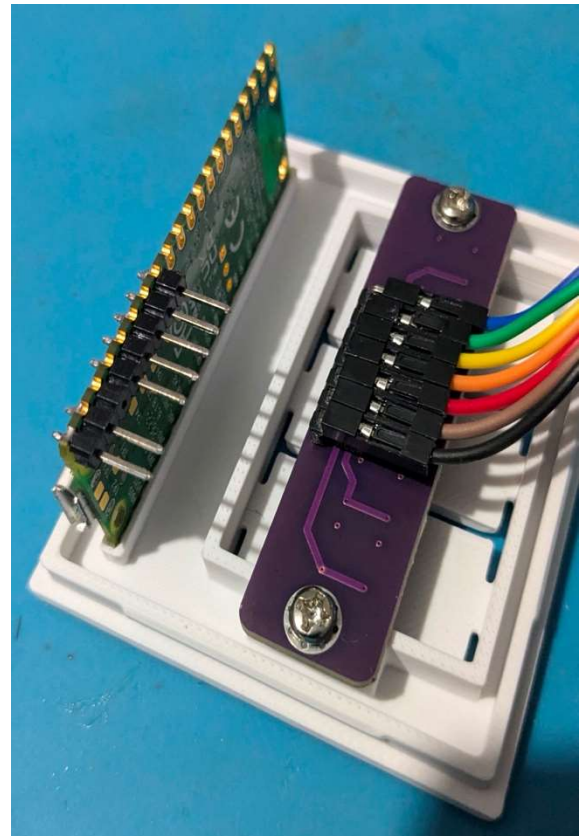
Just make sure all the connectors are pushed in fully.

The connection to the left (in this picture the pin with the black wire) is the common connection, the other 6 are for each of the buttons.



### Insert the Pico into the case

Carefully insert the Pico into the printed slot as shown in the photo below. Insert one corner first then work the other corner in by pushing down on the edge of the board. It is a firm fit but be careful not to damage the board.



### Connect the cables to the Pico

Connect the wires to the Pico as shown in the photo matching your appropriate colours between button board and Pico.

Note the gap between connection 2 and 3.

Check that the common wire on the button board is connected to the 7<sup>th</sup> end connection pin on the Pico.



### Close the lid

Fit the lid onto the box by aligning the left (non-USB socket) side of the lid first.



### Check board alignment

Look through the hole in the case and check that the bottom of the Pico board is matched into the slot in the base of the case as shown in the photograph.

If the board is not aligned, then lift the lid and gently move the Pico in its lid slot and try again.

Once the board aligns well then click the lid down.

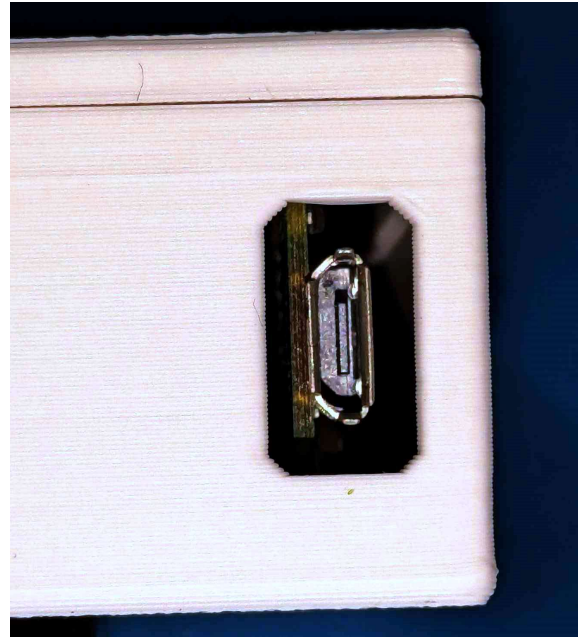


## Check Micro-USB alignment

Look through the hole in the case and check that the Micro-USB socket sits square in the aperture.

Insert the Micro-USB cable into the Pico.

Assembly is now complete.



## Configuring the Pico

---

### Plug in the Pico into your computer

Once you plug your Pico into your computer, Windows will install the device drivers.

Once installed, unplug the Pico.

### Put the Pico into UF2 download mode

Hold down the BOOT/SEL button on the Pico using a small screwdriver through the hole in the case and, whilst keeping the button pressed, plug the Pico into your computer.

Once plugged in, release the BOOT/SEL button.

Windows should open an Explorer (file manager) window (if not open an explorer window). Navigate to “My Computer” where you should see a new drive.

The Pico acts like a USB memory stick and this is how you add your programs and also the firmware to make the Pico operate – we need to install the correct version of firmware to make QuanClick work.

### Install Circuit Python

Download the latest stable CircuitPython .UF2 file (v9.2.1 at time of writing) from ...

[https://circuitpython.org/board/raspberry\\_pi\\_pico2/](https://circuitpython.org/board/raspberry_pi_pico2/)

Drag and drop the downloaded file into the Pico root folder in the Explorer window that opened earlier.

The Pico should accept the file, recognise it as a firmware update, install it automatically and reboot itself. Windows will detect this reboot and momentarily disconnect the drive then reconnect it. The Pico drive should now show with the updated firmware files.

Configuration is now complete.



## Setting up Thonny with Python

### Download from the Thonny web site

Download Thonny with Python from the Thonny web site at ...

<https://thonny.org>

When the page loads, you will see the links for the computer you are using in the top right corner (as highlighted in the image below).

**Thonny**  
Python IDE for beginners



Download version **4.1.6** for  
**Windows • Mac • Linux**

### Download from GitHub download for the portable version

The portable version of Thonny can be downloaded from the GitHub web site at ...

<https://github.com/thonny/thonny/releases/download/v4.1.6/thonny-py38-4.1.6-windows-portable.zip>

Extract the contents of the zip file to C:\Users\{user\_name}\Thonny

### Install Thonny

Run Thonny.exe using the following settings ...

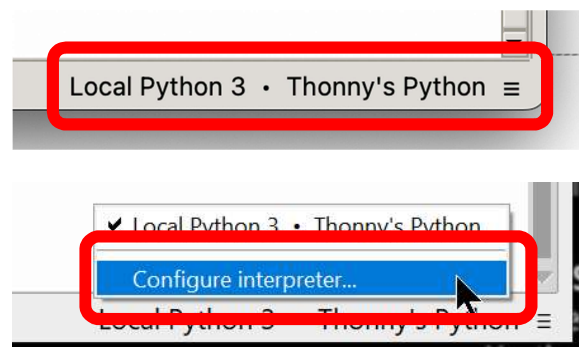
Language: English (UK)

Initial Settings: Raspberry Pi (simple)

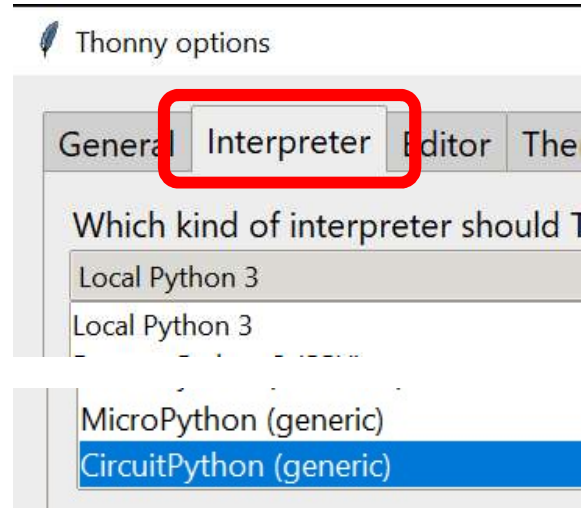
Once Thonny is up-and-running ...

Click in the Thonny menu located in the lower right corner of the window (it's highlighted in red).

When the menu pops up, select **Configure interpreter ...**



When the Thonny Options dialog box appears, click over the **Interpreter** tab (if it is not already selected) ...



... and then select the **Circuit Python (generic)** programming language interpreter.

Click over the **[OK]** button.

Thonny should detect your Pico and the 'com port' it is connected to and display it in the bottom right corner of the Thonny window.

Your computer is now set up with Thonny.

## Programming the Pico

---

### QuanClick project files

You can download the source code and all the support files from ...

<https://github.com/QuantumTechClub/QuanClick>

### Adding the libraries

In the `board_files/lib` directory there is a `adafruit_hid` directory that contains library files used in our project

That entire directory and its contents need to be copied to the `lib` directory on the Pico device.

Drag and drop the entire folder across.

### Open the code in Thonny

Open the `src/quanclick.py` python file in Thonny ...

**File > Open > This computer**

... and then navigate to where you downloaded the file to and click over the **[Open]** button.

Run the script ...

**Run > Run the script**

### Changing the macros

The code can be modified to set personal actions for any key, some examples pre-exist in the source code.

The definitions are listed in the `'button_actions'` array at line 27 in the code.

There are 3 types of actions – **keyboard**, **mouse\_move** and **mouse\_click**.

- The **keyboard** actions contain Keycode types which are encoded codes for each key and special keys like [SHIFT] and [WINDOWS].
- The **mouse\_move** actions contain an x,y co-ordinate move.
- The **mouse\_click** actions contain Mouse types like [LEFT\_BUTTON].

The Appendix contains the codes you can use.

### Installing onto the Pico

Once you are happy with the definition changes then you need to save the python file onto the Pico

- Select File > Save As
- Choose to save the file on the device



## Getting the Pico to autorun the file

To use the Pico in stand alone (ie without running the application via Thonny) we need to change the contents of the default file code.py on the Pico.

- Open the code.py file on the Pico
- Change the contents of that file to read
  - `import quanclick`
- Save the file back to the Pico

The quanclick code will now autorun

## Appendix

### Keycodes

Keycode	Value	Function
A	4	a and A
ALT	226	Alias for LEFT_ALT; Alt is also known as Option (Mac)
APPLICATION	101	also known as the Menu key (Windows)
B	5	b and B
BACKSLASH	49	\ and
BACKSPACE	42	Delete backward (Backspace)
C	6	c and C
CAPS_LOCK	57	Caps Lock
COMMA	54	, and <
COMMAND	227	Labelled as Command on Mac keyboards, with a clover glyph
CONTROL	224	Alias for LEFT_CONTROL
D	7	d and D
DELETE	76	Delete forward
DOWN_ARROW	81	Move the cursor down
E	8	e and E
EIGHT	37	8 and *
END	77	End (often moves to end of line)
ENTER	40	Enter (Return)
EQUALS	46	=` and ``+
ESCAPE	41	Escape
F	9	f and F
F1	58	Function key F1
F10	67	Function key F10
F11	68	Function key F11
F12	69	Function key F12
F13	104	Function key F13 (Mac)
F14	105	Function key F14 (Mac)
F15	106	Function key F15 (Mac)
F16	107	Function key F16 (Mac)

F17	108	Function key F17 (Mac)
F18	109	Function key F18 (Mac)
F19	110	Function key F19 (Mac)
F2	59	Function key F2
F20	111	Function key F20
F21	112	Function key F21
F22	113	Function key F22
F23	114	Function key F23
F24	115	Function key F24
F3	60	Function key F3
F4	61	Function key F4
F5	62	Function key F5
F6	63	Function key F6
F7	64	Function key F7
F8	65	Function key F8
F9	66	Function key F9
FIVE	34	5 and %
FORWARD_SLASH	56	/ and ?
FOUR	33	4 and \$
G	10	g and G
GRAVE_ACCENT	53	` and ~
GUI	227	Alias for LEFT_GUI; GUI is also known as the Windows key or Command (Mac)
H	11	h and H
HOME	74	Home (often moves to beginning of line)
I	12	i and I
INSERT	73	Insert
J	13	j and J
K	14	k and K
KEYPAD_ASTERISK	85	Keypad *
KEYPAD_BACKSLASH	100	Keypad \ and   (Non-US)
KEYPAD_EIGHT	96	Keypad 8 and Up Arrow
KEYPAD_ENTER	88	Keypad Enter
KEYPAD_EQUALS	103	Keypad = (Mac)
KEYPAD_FIVE	93	Keypad 5

KEYPAD_FORWARD_SLASH	84	Keypad /
KEYPAD_FOUR	92	Keypad 4 and Left Arrow
KEYPAD_MINUS	86	Keypad -
KEYPAD_NINE	97	Keypad 9 and PgUp
KEYPAD_NUMLOCK	83	Num Lock (Clear on Mac)
KEYPAD_ONE	89	Keypad 1 and End
KEYPAD_PERIOD	99	Keypad . and Del
KEYPAD_PLUS	87	Keypad +
KEYPAD_SEVEN	95	Keypad 7 and Home
KEYPAD_SIX	94	Keypad 6 and Right Arrow
KEYPAD_THREE	91	Keypad 3 and PgDn
KEYPAD_TWO	90	Keypad 2 and Down Arrow
KEYPAD_ZERO	98	Keypad 0 and Ins
L	15	l and L
LEFT_ALT	226	Alt modifier left of the spacebar
LEFT_ARROW	80	Move the cursor left
LEFT_BRACKET	47	[ and {
LEFT_CONTROL	224	Control modifier left of the spacebar
LEFT_GUI	227	GUI modifier left of the spacebar
LEFT_SHIFT	225	Shift modifier left of the spacebar
M	16	m and M
MINUS	45	- ` and `` _
N	17	n and N
NINE	38	9 and (
O	18	o and O
ONE	30	1 and !
OPTION	226	Labelled as Option on some Mac keyboards
P	19	p and P
PAGE_DOWN	78	Go forward one page
PAGE_UP	75	Go back one page
PAUSE	72	Pause (Break)
PERIOD	55	. and >
POUND	50	# and ~ (Non-US keyboard)
POWER	102	Power (Mac)

PRINT_SCREEN	70	Print Screen (SysRq)
Q	20	q and Q
QUOTE	52	' and "
R	21	r and R
RETURN	40	Alias for ENTER
RIGHT_ALT	230	Alt modifier right of the spacebar
RIGHT_ARROW	79	Move the cursor right
RIGHT_BRACKET	48	] and }
RIGHT_CONTROL	228	Control modifier right of the spacebar
RIGHT_GUI	231	GUI modifier right of the spacebar
RIGHT_SHIFT	229	Shift modifier right of the spacebar
S	22	s and S
SCROLL_LOCK	71	Scroll Lock
SEMICOLON	51	; and :
SEVEN	36	7 and &
SHIFT	225	Alias for LEFT_SHIFT
SIX	35	6 and ^
SPACE	44	Alias for SPACEBAR
SPACEBAR	44	Spacebar
T	23	t and T
TAB	43	Tab and Backtab
THREE	32	3 and #
TWO	31	2 and @
U	24	u and U
UP_ARROW	82	Move the cursor up
V	25	v and V
W	26	w and W
WINDOWS	227	Labelled with a Windows logo on Windows keyboards
X	27	x and X
Y	28	y and Y
Z	29	z and Z
ZERO	39	0 and )

## Mouse codes

Mouse Code	Value	Function
BACK_BUTTON	8	Back mouse button.
FORWARD_BUTTON	16	Forward mouse button.
LEFT_BUTTON	1	Left mouse button.
MIDDLE_BUTTON	4	Middle mouse button.
RIGHT_BUTTON	2	Right mouse button.