

Quadrant E-C 5x14 ortholinear keyboard



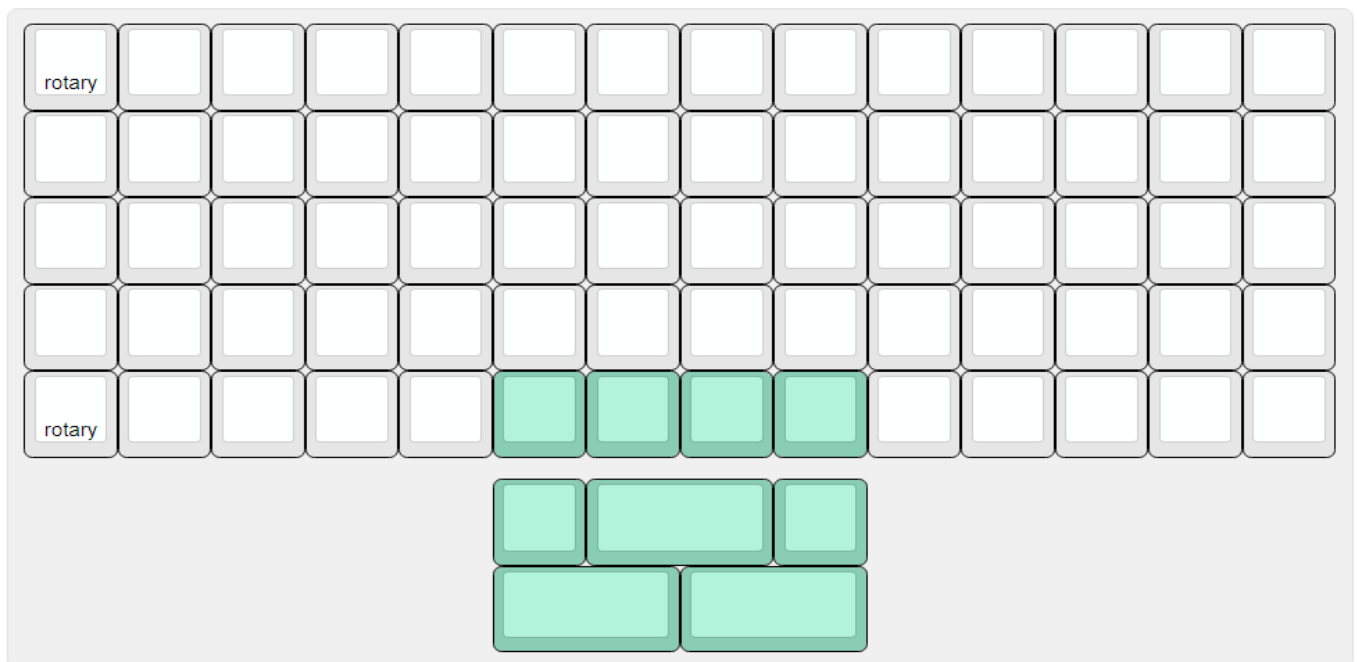
The kit

The kit includes the following:

- 1pc top PCB (it supports MX style, PCB mount switches, Kailh Choc low profile and Alps switches)
- 1pc bottom PCB
- 70pcs 1N4148 THT diodes
- 1pc 3mm LED (cool white)
- 1pc 1k Ohm resistor for LED
- 1pc 6x6mm tactile push button
- 10pcs M2x6mm standoffs
- 4pcs M2x10mm standoffs
- 12pcs M2x4mm screws
- 4pcs M2x6mm screws
- 4pcs M2x12mm screws
- 1pc transparent polycarbonate shield
- 4pcs rubber feet

Layout options

v1.3



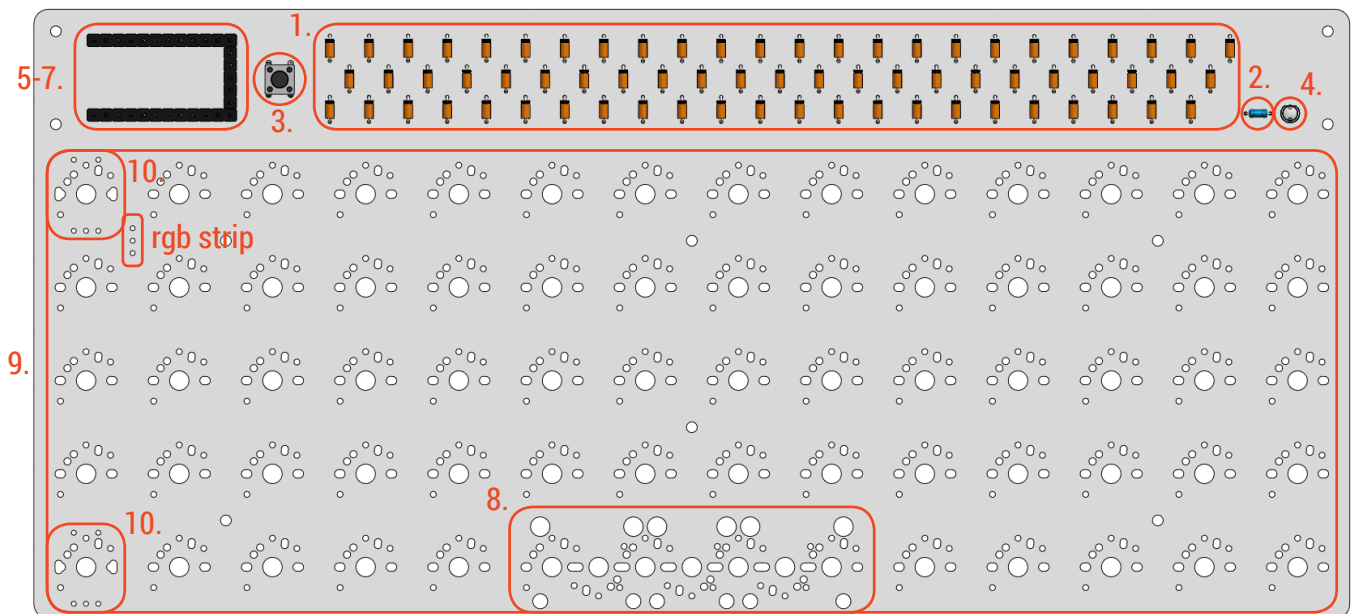
Build Guide

Parts needed to build the top PCB:

- 1pc Elite-C microcontroller
- 67-70pcs keyswitches (MX-compatible / Kailh Choc / Alps-style switches)
- 1 pc EC11 rotary encoder (optional)
- 1 or 2pcs 2u PCB mount MX stabilizers if using 2u keys
- 1pc USB-C - USB-A cable

Build steps:

1. Solder diodes on the top PCB (*D1-D70*). Caution: orientation of the diodes is matter, so position the diodes as the black half of them point to the upper edge of the PCB!
2. Solder resistor (*R1*; orientation doesn't matter)
3. Solder push button (*SW1*; orientation doesn't matter)
4. Solder LED (*LED1*; the longer leg of LED is the anode, so position the LED as the flat side points to the upper edge of the PCB)
5. Solder pin headers for Elite-C (*U1*)
6. Flash Elite-C via QMK
7. Solder Elite-C on the PCB (*U1*)
8. Add 2u stabilizers (optional)
9. Solder switches according to your preferred layout (*MX1-MX73*)
10. Solder rotary encoder (*ROT1* or *ROT2*; optional)



Using WS2818B LED strip:

You will need a WS2818B LED strip and 3 pcs of AWG18 or similar cable to connect the LED strip(s) to the three pin on back side of the top PCB (*rgb strip*). To make it work you have to enable the RGBLIGHT option in QMK files and configure the lighting setup in *config.h* according to your LED strip (number of LEDs, lighting steps, modes, etc.) More information about this feature on QMK's page: https://beta.docs.qmk.fm/features/feature_rgblight

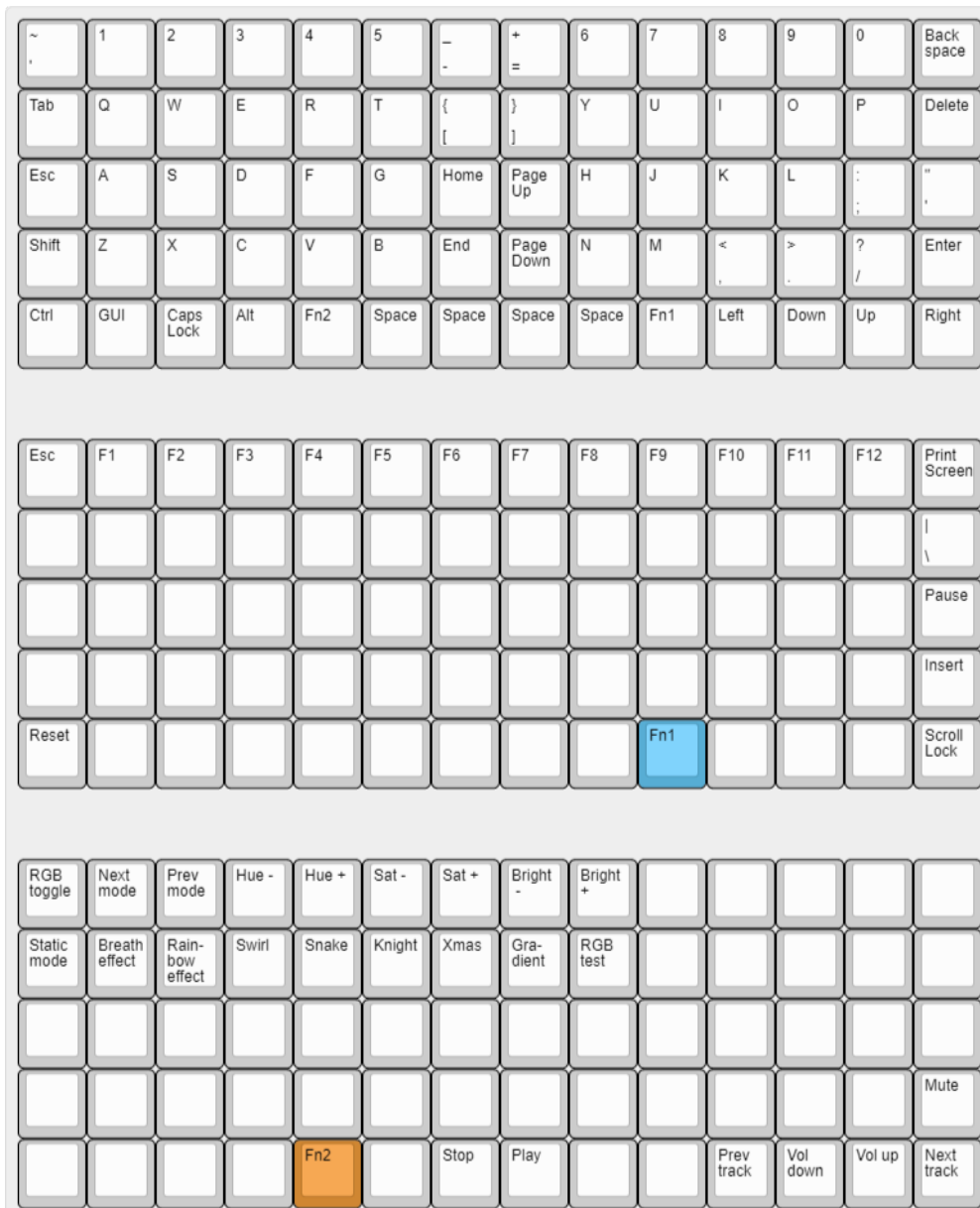
Firmware

First of all: if you don't have the appropriate QMK environment already, please follow the guidelines of this official tutorial: https://beta.docs.qmk.fm/newbs/newbs_getting_started

1. After setup, please download or add the **quadrant** folder (<https://github.com/Ealdin/keyboards/tree/master/Quadrant/firmware>) to your local QMK repository.
2. If you would like to change the layout or the keymap of your Quadrant, open the files in the folder with a code editor (i.e. Visual Studio Code):
 - 2.1. You can choose one of the 5 layout options, which is identical to the physical layout of your build:
 - 2.1.1. *Quadrant_5x14*: full grid layout;
 - 2.1.2. *Quadrant_1x2uC*: one 2u key in the middle of the bottom row (similar to Planck MIT layout);
 - 2.1.3. *Quadrant_1x2uL*: one 2u key arranged asymmetrically in the middle of the bottom row (left);
 - 2.1.4. *Quadrant_1x2uR*: one 2u key arranged asymmetrically in the middle of the bottom row (right);
 - 2.1.5. *Quadrant:2x2u*: two 2u keys in the middle of the bottom row.

The template matrices are found in the *quadrant.h* file.

- 2.2. You can modify the keys of your layout of choice in the *keymap.c* file. The default keymap with 2 function layers:

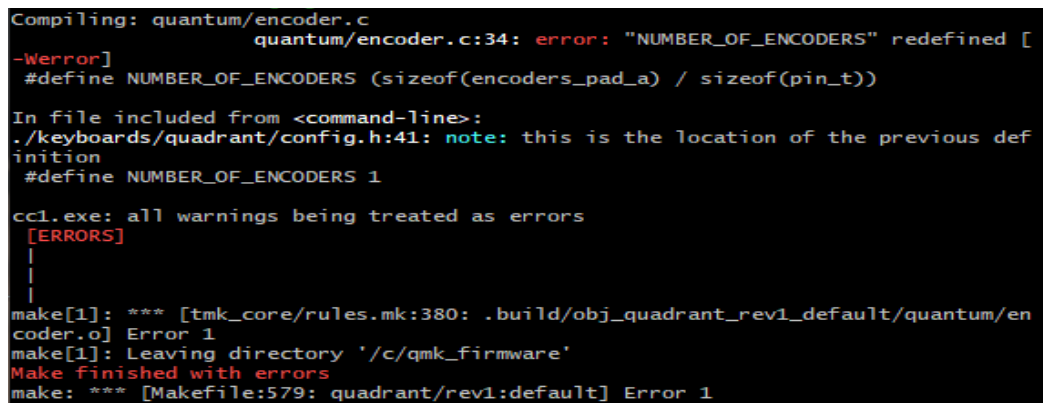


Please note that because of the schematic of the top PCB, the matrix is mirrored on the long side!

- 2.3. For RGB backlight, you can configure the RGB settings in the *config.c* file. More information on QMK's RGB tutorial: https://beta.docs.qmk.fm/features/feature_rgblight
- 2.4. The settings of the rotary encoder can be find in the *quadrant.c* file.
- 2.5. Save the files.
3. Compile the code*: *make quadrant:default*
4. Plug the Elite-C microcontroller into your PC.
5. Flash it via QMK Toolbox (<https://qmk.fm/toolbox/>), using the newly created *.hex* file.
6. Solder the Elite-C on your Quadrant.

For further informations, please check out QMK's Newbs Guide: <https://beta.docs.qmk.fm/newbs>

*There might be an issue when compiling the firmware: an error occurs in *quantum/encoder.c* file (see the picture below):



```
Compiling: quantum/encoder.c
quantum/encoder.c:34: error: "NUMBER_OF_ENCODERS" redefined [
-Werror]
#define NUMBER_OF_ENCODERS (sizeof(encoders_pad_a) / sizeof(pin_t))

In file included from <command-line>:
./keyboards/quadrant/config.h:41: note: this is the location of the previous def
inition
#define NUMBER_OF_ENCODERS 1

cc1.exe: all warnings being treated as errors
[ERRORS]
|
|
|
make[1]: *** [tmk_core/rules.mk:380: .build/obj_quadrant_rev1_default/quantum/en
coder.o] Error 1
make[1]: Leaving directory '/c/qmk_firmware'
Make finished with errors
make: *** [Makefile:579: quadrant/rev1:default] Error 1
```

In order to make it work, you should modify this file. The section of code which causes the issue is:

```
#define NUMBER_OF_ENCODERS (sizeof(encoders_pad_a) / sizeof(pin_t))
static pin_t encoders_pad_a[] = ENCODERS_PAD_A;
static pin_t encoders_pad_b[] = ENCODERS_PAD_B;0
```

Please make that section inactive (insert */** and **/* before and after it) and copy the following lines below that:

```
static pin_t encoders_pad_a[NUMBER_OF_ENCODERS] = ENCODERS_PAD_A;
static pin_t encoders_pad_b[NUMBER_OF_ENCODERS] = ENCODERS_PAD_B;
```

Then write *//* before this line (to make it inactive as well):

```
static int8_t encoder_pulses[NUMBER_OF_ENCODERS] = {0};
```

These modifications are highlighted on the picture on the next page.

If you have any problems with the firmware, please let me know so I can improve it!

```
14  * You should have received a copy of the GNU General Public License
15  * along with this program. If not, see <http://www.gnu.org/licenses/>.
16  */
17
18  #include "encoder.h"
19
20  // for memcpy
21  #include <string.h>
22
23
24  #ifndef ENCODER_RESOLUTION
25  | #define ENCODER_RESOLUTION 4
26  #endif
27
28  #ifndef NUMBER_OF_ENCODERS
29  | #error "Number of encoders not defined by NUMBER_OF_ENCODERS"
30  #endif
31
32  #if !defined(ENCODERS_PAD_A) || !defined(ENCODERS_PAD_B)
33  | #error "No encoder pads defined by ENCODERS_PAD_A and ENCODERS_PAD_B"
34  #endif
35
36  /*
37  #define NUMBER_OF_ENCODERS (sizeof(encoders_pad_a) / sizeof(pin_t))
38  static pin_t encoders_pad_a[] = ENCODERS_PAD_A;
39  static pin_t encoders_pad_b[] = ENCODERS_PAD_B;
40  */
41
42  static pin_t encoders_pad_a[NUMBER_OF_ENCODERS] = ENCODERS_PAD_A;
43  static pin_t encoders_pad_b[NUMBER_OF_ENCODERS] = ENCODERS_PAD_B;
44
45  static int8_t encoder_LUT[] = { 0, -1, 1, 0, 1, 0, 0, -1, -1, 0, 0, 1, 0, 1, -1, 0 };
46
47  static uint8_t encoder_state[NUMBER_OF_ENCODERS] = {0};
48  // static int8_t encoder_pulses[NUMBER_OF_ENCODERS] = {0};
49
50  #ifdef SPLIT_KEYBOARD
51  // slave half encoders come over as second set of encoders
52  static int8_t encoder_value[NUMBER_OF_ENCODERS * 2] = {0};
53  #else
54  static int8_t encoder_value[NUMBER_OF_ENCODERS] = {0};
55  #endif
56
57  __attribute__((weak))
58  void encoder_update_user(int8_t index, bool clockwise) { }
59
60  __attribute__((weak))
61  void encoder_update_kb(int8_t index, bool clockwise) {
62  | encoder_update_user(index, clockwise);
63  }
```

Assembling the keyboard

You will need a hex screwdriver with a 1.5mm bit for M2x4 and M2x12 screws and a T5 torx bit for M2x6 screws OR with PH1 Philips bit for all screws (depending on your kit).

1. Screw together 6 M2x4 screws, all 4 pcs of M2x12 screws and all of the M2x6mm standoffs, while the bottom PCB is sandwiched between them.
2. Place the built top PCB on the top of the standoffs and the M2x12 screws. Align the screw holes of the standoffs with the holes of the top PCB.
3. Use the remained 6 pcs of M2x4 screws to fasten the top PCB to the standoffs.
4. Screw on the 4 M2x10mm standoffs to the M2x12 screws.
5. Place the polycarbonate shield on the top of the M2x10mm standoffs and align the holes with the stand-offs.
6. Fasten the shield with the 4 M2x6 screws to the standoffs.
7. Put the rubber feet on the bottom of the keyboard.
8. Put on some fancy keycaps.
9. Plug in a USB-C - USB-A cable.
10. Enjoy your Quadrant!

