

Guide complet : Création d'un Macropad 4×4 avec QMK et Pro Micro

1. Introduction

Ce guide explique comment construire et programmer un macropad 4×4 utilisant un microcontrôleur Pro Micro et le firmware QMK.

Le macropad peut envoyer des raccourcis clavier, contrôler le volume, lancer des applications ou déclencher des macros.

Nous verrons :

- Le matériel nécessaire
- L'installation de QMK
- La création d'un nouveau clavier
- La configuration des fichiers
- La compilation et le flash du firmware
- L'utilisation de QMK Toolbox
- L'ajout d'un écran OLED

2. Matériel nécessaire

- 1 microcontrôleur Pro Micro (ATmega32U4)
- 16 switches Cherry MX ou compatibles
- 16 diodes 1N4148
- 16 keycaps
- 1 écran OLED I2C 128x64
- fils électriques
- fer à souder et étain
- câble USB C

Le Pro Micro est très populaire pour les projets de clavier DIY car il est compatible avec QMK et possède suffisamment d'entrées/sorties pour gérer la matrice du clavier.

3. Principe de fonctionnement

Les claviers utilisent une matrice de lignes et colonnes.

Chaque touche relie une ligne et une colonne.

Exemple matrice 4x4 :

4 lignes

4 colonnes

16 touches

Quand une touche est pressée :

la ligne et la colonne sont connectées et le microcontrôleur détecte la position.

Les diodes empêchent les conflits appelés "ghosting".

4. Installation de QMK

Installer les outils nécessaires :

QMK MSYS :

<https://msys.qmk.fm/>

QMK Toolbox :

<https://qmk.fm/toolbox/>

QMK MSYS permet de compiler le firmware.

QMK Toolbox permet de flasher le microcontrôleur.

Dans QMK MSYS :

```
qmk setup
```

Cette commande télécharge le firmware QMK et configure l'environnement.

Ensuite vérifier l'installation :

```
qmk doctor
```

5. Création d'un clavier

Créer un nouveau projet clavier :

```
qmk new-keyboard
```

Répondre aux questions :

Nom du clavier : exemple bobito

Nom github : votre pseudo ou n'importe quoi

Nom réel : ou n'importe quoi

Layout : ortho 4x4 (taper le numéro associé dans la liste)

Carte de développement : yes

Type : promicro (taper le numéro associé dans la liste)

QMK crée automatiquement un dossier dans :

C:\Users\" utilisateur" \qmk_firmware/keyboards/"nom_du_clavier"/

6. Structure du projet

La structure doit être :

qmk_firmware/keyboards/nom_du_clavier/

├── config.h

├── keyboard.json

├── rules.mk

└── keymaps/

 └── default/

 └── keymap.c

Chaque fichier a un rôle spécifique dans la configuration du clavier.

7. Configuration config.h

config.h définit la matrice et le matériel.

```
#define MATRIX_ROWS 4
```

```
#define MATRIX_COLS 4
```

```
#define MATRIX_ROW_PINS { D4, D7, C6, E6 }
```

```
#define MATRIX_COL_PINS { B2, B6, B5, B4 }
```

```
#define DIODE_DIRECTION COL2ROW
```

Configuration OLED :

```
#define OLED_DISPLAY_ADDRESS 0x3C
```

```
#define OLED_DISPLAY_128X64
```

```
#define OLED_TIMEOUT 0
```

8. rules.mk

rules.mk active les fonctionnalités du firmware.

```
MCU = atmega32u4
```

```
BOOTLOADER = caterina
```

```
OLED_ENABLE = yes
```

```
OLED_DRIVER = ssd1306
```

9. keyboard.json

Ce fichier décrit le clavier.

Il permet aussi d'activer certaines fonctionnalités.

```
{  
  
  "manufacturer": "custom",  
  
  "keyboard_name": "macro4x4_oled",  
  
  "maintainer": "custom",  
  
  "bootloader": "caterina",  
  
  "processor": "atmega32u4",  
  
  "matrix_size": {  
  
    "rows": 4,  
  
    "cols": 4  
  
  },  
  
  "features": {  
  
    "extrakey": true,  
  
    "nkro": true  
  
  },  
  
  "usb": {  
  
    "vid": "0xFEED",  
  
    "pid": "0x0001",
```

```
"device_version": "1.0.0"
```

```
},
```

```
"layouts": {
```

```
  "LAYOUT_ortho_4x4": {
```

```
    "layout": [
```

```
      { "matrix": [0, 0], "x": 0, "y": 0 },
```

```
      { "matrix": [0, 1], "x": 1, "y": 0 },
```

```
      { "matrix": [0, 2], "x": 2, "y": 0 },
```

```
      { "matrix": [0, 3], "x": 3, "y": 0 },
```

```
      { "matrix": [1, 0], "x": 0, "y": 1 },
```

```
      { "matrix": [1, 1], "x": 1, "y": 1 },
```

```
      { "matrix": [1, 2], "x": 2, "y": 1 },
```

```
      { "matrix": [1, 3], "x": 3, "y": 1 },
```

```
      { "matrix": [2, 0], "x": 0, "y": 2 },
```

```
      { "matrix": [2, 1], "x": 1, "y": 2 },
```

```
{ "matrix": [2, 2], "x": 2, "y": 2 },  
  
{ "matrix": [2, 3], "x": 3, "y": 2 },  
  
{ "matrix": [3, 0], "x": 0, "y": 3 },  
  
{ "matrix": [3, 1], "x": 1, "y": 3 },  
  
{ "matrix": [3, 2], "x": 2, "y": 3 },  
  
{ "matrix": [3, 3], "x": 3, "y": 3 }  
  
]  
  
}  
  
}  
  
}
```

10. Keymap.c

Ce fichier permet de configurer le comportement du clavier

```
#include QMK_KEYBOARD_H
```

```
enum layers {  
    _BASE,  
    _VIDEO,  
    _STREAM,  
}
```

```

    _CAD
};

const uint16_t PROGMEM keymaps[][MATRIX_ROWS][MATRIX_COLS] = {

    [_BASE] = LAYOUT_ortho_4x4(
        KC_MUTE, KC_VOLD, KC_VOLU, KC_MPLY,
        LCTL(KC_C), LCTL(KC_V), LCTL(KC_X), LCTL(KC_S),
        LCTL(KC_1), LCTL(KC_3), LCTL(KC_5), LCTL(KC_7),
        LCTL(KC_W), LCTL(KC_Y), KC_CALC, TO(_VIDEO)
    ),

    [_VIDEO] = LAYOUT_ortho_4x4(
        KC_F13, KC_F14, KC_F15, KC_F16,
        KC_F17, KC_F18, KC_F19, KC_F20,
        KC_F21, KC_F22, KC_F23, KC_F24,
        KC_TRNS, KC_TRNS, KC_TRNS, TO(_STREAM)
    ),

    [_STREAM] = LAYOUT_ortho_4x4(
        KC_F13, KC_F14, KC_F15, KC_F16,
        KC_F17, KC_F18, KC_F19, KC_F20,
        KC_F21, KC_F22, KC_F23, KC_F24,
        KC_TRNS, KC_TRNS, KC_TRNS, TO(_CAD)
    ),

    [_CAD] = LAYOUT_ortho_4x4(

```

```
KC_ESC, KC_ENT, KC_SPC, KC_BSPC,  
KC_LEFT, KC_DOWN, KC_UP, KC_RGHT,  
KC_Z, KC_Y, KC_S, KC_DEL,  
KC_TRNS, KC_TRNS, KC_TRNS, TO(_BASE)  
)  
};
```

```
#ifdef OLED_ENABLE
```

```
oled_rotation_t oled_init_user(oled_rotation_t rotation) {  
    return OLED_ROTATION_0;  
}
```

```
bool oled_task_user(void) {  
    oled_clear();  
  
    oled_set_cursor(0, 0);  
  
    if (layer_state_is(_BASE)) {  
        oled_write_ln_P(PSTR("BASE"), false);  
    } else if (layer_state_is(_VIDEO)) {  
        oled_write_ln_P(PSTR("VIDEO"), false);  
    } else if (layer_state_is(_STREAM)) {  
        oled_write_ln_P(PSTR("STREAM"), false);  
    } else if (layer_state_is(_CAD)) {  
        oled_write_ln_P(PSTR("CAD"), false);  
    }
```

```
} else {  
    oled_write_ln_P(PSTR("UNDEF"), false);  
}  
  
return false;  
}  
  
#endif
```

Vous pouvez trouver la liste complète des macros ici :
https://docs.qmk.fm/keycodes?utm_source=chatgpt.com

11. Compiler le firmware

Dans QMK MSYS :

```
qmk compile -kb nom_du_clavier -km default
```

Cela génère un fichier .hex utilisé pour programmer le microcontrôleur dans
C:\Users\"utilisateur"\qmk_firmware

13. Flash avec QMK Toolbox

1. Ouvrir QMK Toolbox
2. Charger le fichier .hex
3. Connecter le Pro Micro
4. Court-circuiter RST et GND un instant

Le bootloader reste actif environ 7 secondes.

Cliquer sur Flash.

Si tout fonctionne : Flash Complete

14. Dépannage

Touches multimédia ne fonctionnent pas :

vérifier extrakey dans keyboard.json

Touches inversées :

modifier MATRIX_COL_PINS

Ligne incorrecte :

modifier MATRIX_ROW_PINS

OLED bruit ou glitch :

vérifier SDA SCL VCC GND

15. Améliorations possibles

Ajouter :

- encodeur rotatif
- écran OLED personnalisé
- macros avancées
- RGB
- macros OBS
- macros AutoHotkey

Un macropad peut devenir un outil très puissant pour la productivité ou le streaming.

