

Micro Keypad Software

Inhalt

Dieses Video ist der zweite Part zu meinem Micro Keypad Projekt. Hier geht es um die Firmware kmk, welche Möglichkeiten sie bietet und wie man damit am Ende eine eigene Tastatur erstellen kann. Als Entwicklungsumgebung für Python kommt Thonny zum Einsatz. Und nein, man muss dafür kein Entwicklerprofi sein 😊

Youtube Video #12



Spenden

Wer meinen Kanal und meine Arbeit unterstützen möchte, kann das über folgenden Paypal Link:
https://www.paypal.com/donate/?hosted_button_id=68UZ68DMENVP6

Github

<https://github.com/DrKlipper/MikroKeypad/>

Anleitung

Entwicklung / Debugging

- Thonny
- Notepad++
- Serial Monitor (z.B. YAT → <https://sourceforge.net/projects/y-a-terminal/>)
Port Einstellungen: 115200 8n1 mit DTR

boot.py konfigurieren

boot.py

```
import supervisor
import board
import storage
import usb_cdc

from digitalio import DigitalInOut, Direction, Pull

supervisor.set_next_stack_limit(4096 + 4096)

row = DigitalInOut(board.GP3)
col = DigitalInOut(board.GP4)

row.direction = Direction.INPUT
col.direction = Direction.OUTPUT

row.pull = Pull.DOWN
col.value = True

# Wenn die Taste NICHT gedrückt ist wird alles abgeschaltet
# Nur wenn die Taste (#) gedrückt wird erscheint der serielle Port &
# das Laufwerk
if not row.value:
    # USB Laufwerk einblenden auf Knopfdruck (#)
    #storage.disable_usb_drive()
    # Seriellen Port einblenden auf Knopfdruck (#)
    #usb_cdc.disable()
```

Programmierung

Single Key

<http://kmkfw.io/docs/keycodes>

Beispiele

- KC.A
- KC.N1
- KC.ENTER
- KC.F13
- KC.LEFT

- `KC.LCTRL`

Sondertasten + Key

Beispiele

- `KC.LCTRL(KC.LALT(KC.DOT))`,
- `KC.LCTRL(KC.C)`

Sondertasten + Key mit Verzögerung

```
simple_key_sequence([
    KC.LALT(no_release=True),
    KC.MACRO_SLEEP_MS(30),
    KC.LEFT,
    KC.MACRO_SLEEP_MS(30),
    KC.LALT(no_press=True),])
```

String Kette

Beispiele

- `send_string('open https://github.com') → Achtung Sonderzeichen ! Siehe Fix folgend`
- `send_string('Dies ist nur ein Text !')`

Problemfix send_string

- für Umlaute und Sonderzeichen in deutschem Format ...
 - US Tastatur installieren
 - Text in ENG Layout schreiben
 - Wenn Windows auf Deu steht hat man die Umlaute und Sonderzeichen in Deutsch
 - Bsp: `send_string("http>&&www.heise.de':['\"{")`, ergibt → <http://www.heise.de>öäüÖÄÜ

Beispiel simple Keypad

Dieses Beispiel nutzt nur die Tasten mit einem Layer.

[code.py](#)

```
import board
from kmk.kmk_keyboard import KMKKeyboard
from kmk.keys import KC
from kmk.scanners import DiodeOrientation
from kmk.extensions.media_keys import MediaKeys
```

```
from kmk.handlers.sequences import send_string, simple_key_sequence

# Keyboard Instanz erzeugen
keyboard = KMKKeyboard()

# Debugging
keyboard.debug_enabled = True

# Setup
media_keys = MediaKeys()
keyboard.extensions.append(media_keys)

# Tastatur Matrix
keyboard.col_pins = (board.GP2, board.GP0, board.GP4)
keyboard.row_pins = (board.GP1, board.GP6, board.GP5, board.GP3)
keyboard.diode_orientation = DiodeOrientation.COL2ROW

# Keymaps
keyboard.keymap = [
    # Windows
    [
        #KC.MEDIA_PREV_TRACK,
        send_string("<http>&www.heise.de;'\[:\\"",
        KC.MEDIA_PLAY_PAUSE,
        KC.MEDIA_NEXT_TRACK,
        KC.LGUI(KC.LSHIFT(KC.C)),
        KC.LALT(KC.SPACE),
        KC.LCTRL(KC.LSHIFT(KC.ESCAPE)),
        KC.LGUI(KC.LCTRL(KC.LEFT)),
        KC.LGUI(KC.LCTRL(KC.D)),
        KC.LGUI(KC.LCTRL(KC.RIGHT)),
        KC.LGUI(KC.L),
        KC.LCTRL(KC.LALT(KC.LSHIFT(KC.C))),
        KC.NO
    ]
]

if __name__ == '__main__':
    keyboard.go()
```

Rotary Beispiel

[code.py](#)

```
import board
from kmk.kmk_keyboard import KMKKeyboard
```

```
from kmk.keys import KC
from kmk.scanners import DiodeOrientation
from kmk.extensions.media_keys import MediaKeys
from kmk.handlers.sequences import send_string, simple_key_sequence
from kmk.modules.encoder import EncoderHandler

# Keyboard Instanz erzeugen
keyboard = KMKKeyboard()

# Debugging
keyboard.debug_enabled = True

# Setup
media_keys = MediaKeys()
encoders = EncoderHandler()
keyboard.modules = [encoders]
keyboard.extensions.append(media_keys)

# Tastatur Matrix
keyboard.col_pins = (board.GP2, board.GP0, board.GP4)
keyboard.row_pins = (board.GP1, board.GP6, board.GP5, board.GP3)
keyboard.diode_orientation = DiodeOrientation.COL2ROW

# Encoder (Pin A, Pin B, Pin Button)
encoders.pins = (
    (board.GP9, board.GP10, board.GP11, False),
    # Second Encoder
)

# Keymaps
keyboard.keymap = [
    # Windows
    [
        #KC.MEDIA_PREV_TRACK,
        send_string("<http>&&www.heise.de;'\[:\\\"{")",
        KC.MEDIA_PLAY_PAUSE,
        KC.MEDIA_NEXT_TRACK,
        KC.LGUI(KC.LSHIFT(KC.C)),
        KC.LALT(KC.SPACE),
        KC.LCTRL(KC.LSHIFT(KC.ESCAPE)),
        KC.LGUI(KC.LCTRL(KC.LEFT)),
        KC.LGUI(KC.LCTRL(KC.D)),
        KC.LGUI(KC.LCTRL(KC.RIGHT)),
        KC.LGUI(KC.L),
        KC.LCTRL(KC.LALT(KC.LSHIFT(KC.C))),
        KC.NO
    ]
]

# Encoder
encoders.map = (
```

```
        # Layer 1 -> nach dem , käme Encoder 2 wenn vorhanden
        ((KC.VOLD, KC.VOLU, KC.MUTE),),
    )

if __name__ == '__main__':
    keyboard.go()
```

Layer

LEDs

Die LEDs zeigen an, welcher Layer gerade aktiv ist.

LED Code umbauen

Layer 0 wird leider nicht angezeigt sondern erst ab Layer 1. Deswegen wird der entsprechende Code leicht modifiziert 😎

Die Datei liegt unter `kmk\extensions\statusled.py`.

```
def _layer_indicator(self, layer_active, *args, **kwargs):
    """
    Indicates layer with leds

    For the time being just a simple consecutive single led
    indicator. And when there are more layers than leds it
    wraps around to the first led again.
    (Also works for a single led, which just lights when any
    layer is active)
    """

    if self._layer_last != layer_active:
        self.set_brightness(((self.brightness, 0)[layer_active!=0]), 1)
        self.set_brightness(((self.brightness, 0)[layer_active!=1]), 2)
        self.set_brightness(((self.brightness, 0)[layer_active!=2]), 3)
        print(layer_active)
        self._layer_last = layer_active
    # if self._layer_last != layer_active:
    #     # led_last = 0 if self._layer_last == 0 else 1 +
    (self._layer_last - 1) % 3
    #     # print(layer_active)
    # if layer_active > 0:
    #     # led_active = 0 if layer_active == 0 else 1 + (layer_active
    - 1) % 3
    #     # self.set_brightness(self.brightness, led_active)
```

```
        # self.set_brightness(0, led_last)
    # else:
        # self.set_brightness(0, led_last)
    # self._layer_last = layer_active
```

Layer umschalten

Das Layer Umschalten erfolgt über den Befehl `KC.DF(1)`. Die 1 ist der einzuschaltende Layer. Die Layer zählen von 0 an und nicht (!) von 1.

komplettes Beispiel mit Layern

[code.py](#)

```
import board
from kmk.kmk_keyboard import KMKKeyboard
from kmk.keys import KC
from kmk.extensions.media_keys import MediaKeys
from kmk.scanners import DiodeOrientation
from kmk.handlers.sequences import send_string, simple_key_sequence
from kmk.modules.layers import Layers
from kmk.modules.encoder import EncoderHandler
from kmk.extensions.statusled import statusLED
from kmk.extensions.international import International

# Keyboard Instanz erzeugen
keyboard = KMKKeyboard()

# Debugging
keyboard.debug_enabled = True

# LED Konfig für Layer 1..3
statusLED = statusLED(led_pins=[board.GP14, board.GP26, board.GP15])

# Setup
media_keys = MediaKeys()
layers = Layers()
encoders = EncoderHandler()
keyboard.modules = [layers, encoders]
keyboard.extensions.append(media_keys)
keyboard.extensions.append(statusLED)
keyboard.extensions.append(International())

# Tastatur Matrix
keyboard.col_pins = (board.GP2, board.GP0, board.GP4)
keyboard.row_pins = (board.GP1, board.GP6, board.GP5, board.GP3)
keyboard.diode_orientation = DiodeOrientation.COL2ROW
```

```
# Encoder (Pin A, Pin B, Pin Button)
encoders.pins = (
    (board.GP9, board.GP10, board.GP11, False),
    # Second Encoder
)

# Keymaps
keyboard.keymap = [
    # Windows
    [
        KC.MEDIA_PREV_TRACK,
        KC.MEDIA_PLAY_PAUSE,
        KC.MEDIA_NEXT_TRACK,
        KC.LGUI(KC.LSHIFT(KC.C)),
        KC.LALT(KC.SPACE),
        KC.LCTRL(KC.LSHIFT(KC.ESCAPE)),
        KC.LGUI(KC.LCTRL(KC.LEFT)),
        KC.LGUI(KC.LCTRL(KC.D)),
        KC.LGUI(KC.LCTRL(KC.RIGHT)),
        KC.LGUI(KC.L),
        KC.LCTRL(KC.LALT(KC.LSHIFT(KC.C))),
        KC.DF(1)
    ],
    # OBS
    [
        KC.F13,
        KC.F14,
        KC.F15,
        KC.F16,
        KC.F17,
        KC.F18,
        KC.F19,
        KC.F20,
        KC.F21,
        KC.F22,
        KC.F23,
        KC.DF(2)
    ],
    # Schnitt
    [
        KC.LCTRL(KC.C),
        KC.LCTRL(KC.V),
        KC.LCTRL(KC.X),
        KC.B,
        KC.R,
        KC.F,
        KC.LCTRL(KC.BSPACE),
        KC.LSHIFT(KC.S),
    ]
]
```

```

        KC.LCTRL(KC.N2),
        KC.COMMA,
        KC.DOT,
        KC.DF(0)
    ]
]

# Encoder
encoders.map = (
    # Layer 1 -> nach dem , käme Encoder 2 wenn vorhanden
    ((KC.VOLD, KC.VOLU, KC.MUTE),),
    # Layer 2
    ((KC.VOLD, KC.VOLU, KC.MUTE),),
    # Layer 3
    ((KC.LCTRL(KC.LALT(KC.COMMA)),
    KC.LCTRL(KC.LALT(KC.DOT)), KC.LCTRL(KC.LSHIFT(KC.N7))),),
)

if __name__ == '__main__':
    keyboard.go()

```

Layer Beschreibung

Layer 1 (Windows)

Key	Funktion	Key	Funktion	Key	Funktion
1	Kopieren STRG+C	3	Einfügen STRG+V	3	Ausschneiden STRG+X
4	Coloir Picker WIN+Shift+C	5	Power Run ALT + Space	6	Taskmanager Strg + Shift + Esc
7	Previous Desktop WIN + STRG + Left	8	New virtual Desktop WIN + CTRL + D	9	Next Desktop WIN + STRG + RIGHT
*	Lock WIN + L	0	Taschenrechner STRG * ALT * SHIFT + C (Shortcut)	#	Layer 2

Rotary

Press → Mute

Left → Leiser

Right → Lauter

Layer 2 (OBS)

Key	Funktion	Key	Funktion	Key	Funktion
1	F13	3	F14	3	F15
4	F16	5	F17	6	F18

Key	Funktion	Key	Funktion	Key	Funktion
7	F19	8	F20	9	F21
*	F22	0	F23	#	Layer 3

Rotary

Press → Mute
Left → Leiser
Right → Lauter

Layer 3 (Schnitt)

Key	Funktion	Key	Funktion	Key	Funktion
1	Kopieren STRG+C	3	Einfügen STRG+V	3	Ausschneiden STRG+X
4	Medien B	5	Bibliothek R	6	Favoriten F
7	Löschen und Lücke schließen STRG + Back	8	Audio stummschalten SHIFT + S	9	Eigenwschaften An / Aus STRG + 2
*	Frame zurück , (Komma)	0	Frame vor . (Punkt)	#	Layer 1

Rotary

Press → Zoom All
Left → Zeitleiste Links
Right → Zeitleiste rechts

Links

- <http://kmkfw.io/>
- <https://kmkfw.io/docs/>
- https://github.com/KMKfw/kmk_firmware

From:
<https://drklipper.de/> - Dr. Klipper Wiki

Permanent link:
https://drklipper.de/doku.php?id=hardware:12_-_micro_keypad_software&rev=1695868218

Last update: 2023/10/20 08:59

