



Vyšší odborná škola obalové techniky a Střední škola. Štětí, Kostelní 134, příspěvková organizace
18-20-M/01 Informační technologie – internet věcí



PROGRAMOVATELNÁ MAKRO-KLÁVESNICE

MATURITNÍ PRÁCE

Studijní program: 18-20-M/01 Informační technologie

Studijní obor: internet věcí

Jméno a příjmení: Jakub Malík

Třída: 4.I

Vedoucí práce: Bc. Jakub Škrabánek

1	Obsah	
2	Prohlášení.....	4
3	Poděkování.....	5
4	Anotace	6
5	Představení projektu	7
6	Teoretická část	8
6.1	Co je to programovatelná makro-klávesnice	8
6.2	Popis využitých technologií	8
6.2.1	3D Tisk	8
6.2.2	Makro (software)	8
6.2.3	MIDI	9
6.3	Seznam použité techniky.....	9
7	Praktická část.....	10
7.1	Popis techniky.....	10
7.2	Napájení.....	11
7.3	Popis postupu	11
8	Programovací jazyk a knihovny	12
8.1	Python.....	12
8.2	Knihovna CircuitPython.....	12
9	Popis kódu	13
10	Závěr	22
11	Přílohy	23
11.1	Raspberry Pi Pico	23
11.2	OLED displej 1.5" 128x128	23
11.3	AW9523 expandér a LED řadič	24
11.4	LED Tlačítko.....	24
11.5	5ti směrný spínač.....	25
11.6	První prototyp v den dodání tlačítek.....	25
12	Využití zdroje	26

2 Prohlášení

Prohlašuji, že tuto maturitní práci jsem vypracoval samostatně. Veškeré použité zdroje jsou řádně citovány a všechny části práce jsou mé vlastní tvorby.

Ve Štětí dne 12. dubna 2024

.....

Podpis studenta

3 Poděkování

Tímto chci srdečně poděkovat mému vedoucímu práce za všechnu pomoc, které mi bylo při vytváření tohoto projektu poskytnuto. Zároveň chci poděkovat celé škole za tuto zkušenost.

4 Anotace

Jako projekt v rámci maturitní práce z IVE jsem se rozhodl vybrat programovatelnou klávesnici, což bylo rozhodnutí motivované mojí vášní pro hudbu.

Očekávám, že tento projekt mi vylepší požitek z produkce hudby a zároveň bude sloužit jako ideální nástroj pro urychlení pracovních procesů v daném odvětví.

Kromě prohloubení mého hudebního hobby jsem hledal projekt, který bude praktický a snadno integrovatelný, a do budoucna aplikovatelný do každodenního života. Tímto způsobem si u mě klávesnice získala své místo a věřím, že se mi povede ji zpracovat kvalitně.

5 Představení projektu

Představení Projektu: Programovatelná makro-klávesnice

Můj vybraný projekt představuje vývoj programovatelné klávesnice, která je ztělesněním kombinace hardwaru a software pro specifické využití v různých aplikacích, zejména v oblasti hudby.

Tato klávesnice je navržena tak, aby poskytovala nadstandardní úroveň personalizace a interaktivity pro uživatele. Klávesnice má zakomponovanou vizuální zpětnou vazbu prostřednictvím OLED displeje a tlačítek s LED osvětlení, což zvyšuje úroveň jak po vizuální stránce, tak i uživatelský komfort. Díky použití Raspberry Pi Pico a rozšířených I/O možností budu moci klávesnici do budoucna snadno rozšiřovat o další sensory, což ji činí ideálním nástrojem pro kreativní hudebníky, a technologické nadšence jako jsem já.

6 Teoretická část

6.1 Co je to programovatelná makro-klávesnice

Programovatelná makro klávesnice je specializované zařízení, které umožňuje uživatelům přiřadit řadu příkazů nebo akcí k jakékoliv klávese nebo tlačítku. Tyto akce (neboli makra) mohou zahrnovat komplexní sekvence klávesových zkratk, ovládání softwaru, spouštění aplikací, nebo dokonce automatizaci opakujících se úkolů. Díky jejím možnostem jsme schopni ji používat jak při programování, hudební produkci, hraní her, nebo u jakékoliv práce, kde je efektivita a zrychlení pracovních procesů klíčová.

6.2 Popis využitých technologií

6.2.1 3D Tisk

3D tisk neboli **aditivní výroba** také inkrementální nebo přírůstková výrobní technologie (anglicky *3D printing* neboli *additive manufacturing (AM)*) je proces tvorby třídimenzionálních pevných objektů z digitálního souboru (Additive Manufacturing File – AMF). V [aditivních procesech](#) je objekt vytvořen pokládáním souvislých vrstev materiálu, dokud není celý projekt dokončen. Každá z těchto vrstev může být považována za úzce rozříznutou horizontální sekci daného objektu. Z mechanismů 3D tisku také vychází technologie 3D per.

Tuto technologii jsem použil pro tisknutí krabičky a držáku veškeré techniky, která se nachází uvnitř krabičky a je součástí mojí programovatelné makro klávesnice.

6.2.2 Makro (software)

Makro (od „macroinstruction“ z řeckého slova „μακρό“ – „velké“ nebo „daleko“) je v [informaticce](#) definice pravidla, jak bude vstupní posloupnost transformována na výstupní posloupnost (znaků, akcí, výpočtů a podobně). Tuto transformaci označujeme jako *substituci* nebo *expanzi* makra. Původní termín byl zaveden překladači [jazyka symbolických adres](#), kde [programátor](#) může pomocí makra zaměnit dlouhou sekvenci příkazů jediným krátkým výrazem (makrem).

Technologii makro jsem použil z důvodu přizpůsobování kláves. Díky makro technologii je totiž celý tento projekt možný zrealizovat.

6.2.3 MIDI

MIDI ([akronym](#) **Musical Instrument Digital Interface**, česky „digitální rozhraní hudebního nástroje“) je volně přístupný průmyslový standard, který spravuje organizace MIDI Manufacturers Association (MMA). Tento standard obsahuje specifikace [hardware](#) i [software](#) pro digitální komunikaci mezi hudebními nástroji navzájem a dalšími zařízeními jako jsou [sekvencery](#), počítače, mixery nebo i jevištní technika, třeba ovládání reflektorů.¹

Vzhledem k tomu že mojí programovatelnou makro-klávesnici budu používat zejména pro produkci hudby, použil jsem MIDI technologii která mi pomáhá poskytovat detailní kontrolu nad hudebními parametry jako třeba tón, intenzita nebo rytmus.

6.3 Seznam použité techniky

- Raspberry Pi Pico
- OLED displej 1.5 palců 128x128
- 5ti cestný spínač
- AW 9523 GPIO Expander a LED řadič
- Krabíčka vytisknutá na 3D tiskárně
- 16 LED tlačítek

7 Praktická část

7.1 Popis techniky

Raspberry Pi Pico: Je mikrokontroler založený na čipu RP2040 od Raspberry Pi Foundation, představený v roce 2021. Má dvoujádrový procesor ARM Cortex-M0+, 264Kb SRAM a 2MB Flash paměti pro ukládání kódu. Má mnoho GPIO pinů a podporuje komunikaci přes I2C, SPI a UART. Lze programovat v C/C++ nebo Pythonu

OLED displej 1.5" 128x128: OLED displej o velikosti 1.5 palce a rozlišením 128x128 pixelů je kompaktní obrazovka vhodná pro malé projekty jako mám já. Díky OLED technologii umožňuje, že každý pixel je samostatně osvětlený, což vede k živým barvám. Obrazovka poskytuje vysoký kontrast a široké pozorovací úhly.

AW9523 GPIO Expander a LED ovladač: Slouží jako GPIO expander a ovladač LEDm poskytující dodatečné GPIO piny a možnosti řízení LED osvětlení, kde je omezen počet dostupných pinů mikrokontroléru. Použil jsem ho z důvodu velkého množství LED tlačítek

5ti cestný spínač: Použil jsem ho pro ovládání menu možností tlačítek, komponenta umožňuje uživateli ovládat zařízení v pěti směry pohybu (doleva, doprava, nahoru, dolů a stiskem do středu). Je často používán jako navigační nástroj jako např. v mém projektu.

LED tlačítko: Komponenta, která integruje LED diodu s tlačítkem, umožňuje vizuální zpětnou vazbu a ovládání zároveň.

7.2 Napájení

Ve svém projektu jsem se rozhodl že pro napájení zvolím USB Micro, protože je to široce dostupný a univerzální konektor, který umožňuje snadné připojení k jakémukoliv zařízení. Zároveň USB Micro poskytuje jak napájení, tak přenos dat, což usnadňuje integraci klávesnice do různých pracovních prostředí a zajišťuje minimální nároky na prostor a příslušenství.

7.3 Popis postupu

Objednání součástek:

Jako první bylo potřeba objednat všechny potřebné součástky z internetu. Pro objednání všech komponent jsem zvolil laskakit.cz, temu.com a adafruti.com.

3D tisk krabičky: Během doručování všech mých součástek jsem mezitím zapracoval na 3D krabičce do které všechny komponenty patří.

Montáž součástek:

Po doručení součástek jsem přistoupil k montáži. Nejprve jsem umístil Raspberry Pi Pico do vytisklého držáku, následně jsem vložil veškerá tlačítka do vrchního krytu 3D krabičky společně s OLED displejem a pěti-směrným tlačítkem. Dále jsem propojil pomocí kabelů všechny LED tlačítka displej a spínač do Raspberry Pi Pica a zbylé LED tlačítka do GPIO řadiče, který jsem také umístil do vytisklého držáku. Nakonec jsem všechny kabely napojil a na závěr jsem Micro USB expandér který vede do Raspberry Pi Pica.

Nahrávání knihoven do Raspberry Pi Pica:

Po úspěšném sestavení jsem připojil Raspberry Pi do počítače, nahrál jsem potřebné knihovny pro ovládání GPIO pinů a LED tlačítek a začal jsem programovat.

Programování:

Celý projekt jsem programoval pomocí Pythonu, a jelikož jsem zvolil metodu pokus omyl byl to hrozný porod. Po několika víkendech a nocích stráveným nad dumáním zkoušením koukáním na návody testováním a čtením Githubu jsem celou klávesnici naprogramoval a otestoval.

Testování a ukončení postupu:

Po nekonečném trápení s programováním jsem konečně mohl svůj projekt vyzkoušet v praxi. Spodní část krabičky jsem zaklapnul, spustil jsem program na tvoření hudby a užíval jsem si funkčnost mého projektu až do pozdního večera.

8 Programovací jazyk a knihovny

8.1 Python

Python je vysokoúrovňový [programovací jazyk](#), který v roce [1991^{\[1\]}](#) navrhl [Guido van Rossum](#). Nabízí [dynamickou kontrolu datových typů](#) a podporuje [různá programovací paradigmatata](#), včetně [objektově orientovaného](#), [imperativního](#) nebo [funkcionálního](#). V roce 2018 vzrostla jeho popularita a zařadil se mezi nejoblíbenější jazyky. V řadě různých žebříčků dosahuje jedno z prvních třech míst, výjimkou nebývají první místa.

Python je vyvíjen jako [open source](#) projekt, který zdarma nabízí instalační balíky pro většinu běžných [platform](#) ([Unix](#), [MS Windows](#), [macOS](#), Android); ve většině [distribucí](#) systému [GNU/Linux](#) je Python součástí základní instalace.

8.2 Knihovna CircuitPython

Je verze Pythonu určená pro programování mikrokontrolerů a hardwarových projektů. Díky této knihovně se dá rychle a snadně vyvinout kód pro elektronické komponenty jako např. senzory, displeje a motory, s důrazem na jednoduchost a přístupnost pro širokou škálu lidí od začátečníků po profesionály.

9 Popis kódu

```
1
2
3 import time
4 import board
5 import displayio
6 import terminalio
7 import adafruit_aw9523
8 import busio
9 import adafruit_ssd1327
10 import digitalio
11 from adafruit_display_text import label
12 from adafruit_display_shapes.circle import Circle
13 from adafruit_display_shapes.rect import Rect
14 import usb_midi
15 import adafruit_midi
16 from adafruit_midi.note_on import NoteOn
17 from adafruit_midi.note_off import NoteOff
18
```

V této části kódu dochází k nejdůležitějšímu importu klíčových knihoven a modulů pro ovládání displeje, komunikaci s periferiemi a manipulaci s časem, které jsou základem pro funkčnost této programovatelné klávesnice.

```
19 displayio.release_displays()
20
21
22 i2c = busio.I2C(board.GP1, board.GP0, frequency=1000000)
23
24 display_bus = displayio.I2CDisplay(i2c, device_address=0x3D)
25
26 aw = adafruit_aw9523.AW9523(i2c)
27
28 midi = adafruit_midi.MIDI(midi_out=usb_midi.ports[1], out_channel=0)
29
30
```

Tento úsek inicializuje několik klíčových komponent projektu s využitím I2C komunikace a MIDI funkcionality. Začíná nastavením I2C sběrnice s vyšší frekvencí 1 MHz, což zlepšuje rychlost obnovy displeje, přičemž využívá piny GP1 a GP0 na desce Raspberry Pi Pico. Následuje konfigurace I2C displeje, kde je displej nastaven na konkrétní adrese 0x3D. Dále je inicializován GPIO expandér AW9523 přes stejnou I2C sběrnici, což umožňuje rozšíření počtu dostupných pinů pro další periferie. Nakonec je nastavena MIDI komunikace přes USB MIDI port, specifikován jako výstupní zařízení na portu 1 s výstupním kanálem 0, což umožňuje ovládání externích MIDI zařízení nebo softwaru. Tento kód je základem pro propojení hardwaru a softwaru, umožňující komplexní řízení a interakci s MIDI zařízeními.

```

31  WIDTH = 128
32  HEIGHT = 128
33
34  display = adafruit_ssd1327.SSD1327(display_bus, width=WIDTH, height=HEIGHT, brightness=0.01)
35
36
37  splash = displayio.Group()
38
39  circle_group = displayio.Group()
40
41  text_group = displayio.Group()
42
43
44  spots = (
45      (16, 16),
46      (48, 16),
47      (80, 16),
48      (112, 16),
49      (16, 48),
50      (48, 48),
51      (80, 48),
52      (112, 48),
53      (16, 80),
54      (48, 80),
55      (80, 80),
56      (112, 80),
57      (16, 112),
58      (48, 112),
59      (80, 112),
60      (112, 112),
61  )
62
63
64  for spot in spots:
65      circle = Circle(x0=spot[0], y0=spot[1], r=14, fill=0x888888)
66
67      circle_group.append(circle)
68
69  rect = Rect(0, 0, 33, 33, fill=None, outline=0x00FF00, stroke=3)
70
71  splash.append(circle_group)
72  splash.append(rect)

```

Tento úsek kódu se zabývá inicializací a konfigurací displeje, zde vytvářím uživatelské rozhraní a definuji grafické prvky pro mojí klávesnici. Nejdříve jsou definovány rozměry displeje a následně je displej nastaven s použitím OLED displeje, kde je specifikována šířka, výška a minimální jas. Celkově tato část kódu zajišťuje vytváření a organizaci grafických prvků na displeji, které slouží pro vizuální interakci uživatele s klávesnicí.

```

75 texts = [
76     {'num': "60", 'pos': (12, 16)},
77     {'num': "61", 'pos': (44, 16)},
78     {'num': "62", 'pos': (76, 16)},
79     {'num': "63", 'pos': (108, 16)},
80     {'num': "64", 'pos': (12, 48)},
81     {'num': "65", 'pos': (44, 48)},
82     {'num': "66", 'pos': (76, 48)},
83     {'num': "67", 'pos': (108, 48)},
84     {'num': "68", 'pos': (12, 80)},
85     {'num': "69", 'pos': (44, 80)},
86     {'num': "70", 'pos': (76, 80)},
87     {'num': "71", 'pos': (108, 80)},
88     {'num': "72", 'pos': (12, 112)},
89     {'num': "73", 'pos': (44, 112)},
90     {'num': "74", 'pos': (76, 112)},
91     {'num': "75", 'pos': (108, 112)},
92 ]
93 text_labels = []
94
95 for text in texts:
96     text_area = label.Label(terminalio.FONT, text=text['num'], color=0xFFFFFF)
97     text_area.x = text['pos'][0]
98     text_area.y = text['pos'][1]
99     text_labels.append(text_area)
100     text_group.append(text_area)
101 splash.append(text_group)

```

Tato část kódu zahrnuje inicializaci a umístění textových štítků na displeji, které odpovídají MIDI notám.

```

104 big_splash = displayio.Group()
105
106 big_circle = Circle(x0=64, y0=64, r=62, fill=0x888888)
107 big_splash.append(big_circle)
108
109 big_text = label.Label(terminalio.FONT, text=' ', color=0xFFFFFF)
110 big_text.x = 43
111 big_text.y = 62
112 big_text.scale = 4
113 big_splash.append(big_text)

```

Kódu v tomto úseku se věnuje nastavení sekundárního grafického rozhraní, které se zobrazí, když si vyberete nějaké tlačítko. Tato část má za úkol vytvořit sekundární vizuální prvek, který je použit pro zvýraznění aktuálně vybrané funkce nebo interakce uživatele s klávesnicí.

```

116     leds = []
117     led_pins = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
118
119     for led in led_pins:
120         led_pin = aw.get_pin(led)
121         led_pin.direction = digitalio.Direction.OUTPUT
122         leds.append(led_pin)

```

Zde se inicializují a nastavují LED diody řízený pomocí AW9523 GPIO expandéru. Pomocí této části umožňuje klávesnice se programově připravit a spravovat LED diody připojené k GPIO expandéru.

```

127     note_pins = [board.GP7, board.GP8, board.GP9, board.GP10, board.GP11,
128                  board.GP12, board.GP13, board.GP14, board.GP16, board.GP17,
129                  board.GP18, board.GP19, board.GP20, board.GP21, board.GP22, board.GP26]
130
131     note_buttons = []
132
133     for pin in note_pins:
134         note_pin = digitalio.DigitalInOut(pin)
135         note_pin.direction = digitalio.Direction.INPUT
136         note_pin.pull = digitalio.Pull.UP
137         note_buttons.append(note_pin)
138
139
140     note0_pressed = False
141     note1_pressed = False
142     note2_pressed = False
143     note3_pressed = False
144     note4_pressed = False
145     note5_pressed = False
146     note6_pressed = False
147     note7_pressed = False
148     note8_pressed = False
149     note9_pressed = False
150     note10_pressed = False
151     note11_pressed = False
152     note12_pressed = False
153     note13_pressed = False
154     note14_pressed = False
155     note15_pressed = False
156
157     note_states = [note0_pressed, note1_pressed, note2_pressed, note3_pressed,
158                   note4_pressed, note5_pressed, note6_pressed, note7_pressed,
159                   note8_pressed, note9_pressed, note10_pressed, note11_pressed,
160                   note12_pressed, note13_pressed, note14_pressed, note15_pressed]

```

Tato část kódu se zabývá inicializací tlačítek pro mojí klávesnici, nastavuje jejich piny jako vstupy a sleduje jejich stav. Každé tlačítko je tak spojeno s konkrétním pinem na mikrokontroléru. Tímto způsobem jsem vytvořil infrastrukturu pro detekci stisku tlačítek, umožňující jejich čtení a reakci v hlavním programovém smyčce.

```

162 select = digitalio.DigitalInOut(board.GP6)
163 up = digitalio.DigitalInOut(board.GP5)
164 down = digitalio.DigitalInOut(board.GP4)
165 left = digitalio.DigitalInOut(board.GP3)
166 right = digitalio.DigitalInOut(board.GP2)
167
168 joystick = [select, up, down, left, right]
169
170 for joy in joystick:
171     joy.direction = digitalio.Direction.INPUT
172     joy.pull = digitalio.Pull.UP

```

Tato část kódu inicializuje piny pro 5-směrný přepínač, nastavuje je jako vstupní a aktivuje pro ně jejich vnitřní rezistory. Zároveň přiřazuje směry přepínače GPIO pinům na mikrokontroleru.

```

182 select_x = [0, 32, 64, 96]
183 select_y = [0, 32, 64, 96]
184
185
186 y_pos = 0
187
188 x_pos = 0
189 sub_state = False
190
191 midi_num = 60
192
193 button_num = 0
194
195 button_pos = 0
196
197 led_check = None
198
199 clock = time.monotonic()

```

V této části kódu sem definoval proměnné pro sledování stavů 5-směrného přepínače pro navigaci v hlavním uživatelském rozhraní.

```

202 up_scroll = 0
203 down_scroll = 0
204 left_scroll = 0
205 right_scroll = 0
206 switch_coordinates = [(0, 0), (1, 0), (2, 0), (3, 0),
207                       (0, 1), (1, 1), (2, 1), (3, 1),
208                       (0, 2), (1, 2), (2, 2), (3, 2),
209                       (0, 3), (1, 3), (2, 3), (3, 3)]
210
211
212 midi_notes = [60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75]

```

V této části kódu jsou nastaveny proměnné pro sledování směrového listování. Tyto proměnné umožňují sledovat pohyb v rámci uživatelského rozhraní, například pro navigaci v menu nebo mezi různými grafickými prvky. Zároveň můžeme níže v poli midi-notes vidět výchozí midi noty, které jsou přiřazeny k tlačítkům.

```

217 while True:
218
219
220     if up.value and up_state == "pressed":
221         print("Button pressed.")
222         up_state = None
223     if down.value and down_state == "pressed":
224         print("Button pressed.")
225         down_state = None
226     if left.value and left_state == "pressed":
227         print("Button pressed.")
228         left_state = None
229     if right.value and right_state == "pressed":
230         print("Button pressed.")
231         right_state = None
232     if select.value and select_state == "pressed":
233         print("Button pressed.")
234         select_state = None
235

```

Zde můžeme vidět část kde kód neustále kontroluje stav spínače. Používá metodu debounce díky které zjišťuje zda tlačítko bylo stisknuto nebo ne.

```

237     for i in range(16):
238         buttons = note_buttons[i]
239
240         if not buttons.value and note_states[i] is False:
241
242             midi.send(NoteOn(midi_notes[i], 120))
243             note_states[i] = True
244             leds[i].value = True
245
246         if buttons.value and note_states[i] is True:
247
248             midi.send(NoteOff(midi_notes[i], 120))
249             note_states[i] = False
250             leds[i].value = False
251

```

Zde můžeme vidět část kódu která detekuje stisk a uvolnění tlačítek, odesílá MIDI noty a ovládá LED indikaci. Pokud je tlačítko stisknuto, odesílá se příslušná MIDI nota a rozsvítí se LED světlo v tlačítku. Při uvolnění tlačítka se MIDI nota zastaví a LED světlo zhasne.

```

253     if not sub_state:
254
255         if not up.value and up_state is None:
256             up_state = "pressed"
257
258             up_scroll -= 1
259             if up_scroll < 0:
260                 up_scroll = 3
261             y_pos = up_scroll
262             down_scroll = up_scroll
263
264         if not down.value and down_state is None:
265             down_state = "pressed"
266
267             down_scroll += 1
268             if down_scroll > 3:
269                 down_scroll = 0
270             y_pos = down_scroll
271             up_scroll = down_scroll
272
273         if not left.value and left_state is None:
274
275             left_state = "pressed"
276
277             left_scroll -= 1
278             if left_scroll < 0:
279                 left_scroll = 3
280             x_pos = left_scroll
281             right_scroll = left_scroll
282
283         if not right.value and right_state is None:
284
285             right_state = "pressed"
286
287             right_scroll += 1
288             if right_scroll > 3:
289                 right_scroll = 0
290             x_pos = right_scroll
291             left_scroll = right_scroll
292
293
294     rect.y = select_y[y_pos]
295     rect.x = select_x[x_pos]

```

Zde je část kódu, která má za úkol dělat jednu věc. Pokaždé, když stisknete spínač nahoru, dolů, doleva nebo doprava, aktualizují se hodnoty mezi 0 a 3. Tyto hodnoty se používají jako souřadnice pro sledování vaší pozice v uživatelském rozhraní.

```

298         for coords in switch_coordinates:
299             if x_pos == coords[0] and y_pos == coords[1]:
300                 button_pos = switch_coordinates.index(coords)
301
302                 button_num = text_labels[button_pos].text

```

Tato část kódu má na starosti sledovat pozici přepínače v rozhraní porovnáváním souřadnic.

```

305         if not select.value and select_state is None:
306             select_state = "pressed"
307
308             midi_num = int(button_num)
309
310             sub_state = True

```

Zde zas kód reaguje na to, že když se v rozhraní dostaneme na pozici libovolného tlačítka na klávesnici, můžeme stisknutím přepínače přijít do režimu úpravy vybraného tlačítka.

```

313         if sub_state:
314
315             display.root_group = big_splash
316
317             big_text.text = str(midi_num)
318
319
320             if (time.monotonic() > (clock + 1)) and led_check is None:
321                 leds[button_pos].value = True
322                 led_check = True
323                 clock = time.monotonic()
324             if (time.monotonic() > (clock + 1)) and led_check is True:
325                 leds[button_pos].value = False
326                 led_check = None
327                 clock = time.monotonic()

```

Zde má kód na práci to, že když vstoupíme to režimu úpravy tlačítka tak v rozhraní začne tlačítko blikat, dokud režim neopustíte

```

329         if not select.value and select_state is None:
330             select_state = "pressed"
331
332             sub_state = False
333
334             text_labels[button_pos].text = str(midi_num)
335
336             display.root_group = splash
337
338             leds[button_pos].value = False

```

A v této části kódu se pomocí opětovného stisknutí spínače dostanete pryč z možnosti úpravy tlačítka. Zároveň se se stisknutím úprava uloží a v rozhraní vám tlačítko přestane blikat.

10 Závěr

Maturitní práce zaměřená na vytvoření programovatelné makro-klávesnice s využitím Raspberry Pi Pico vypadá na první pohled jednoduše, ale opak je pravdou. Cílem mé práce bylo vytvořit funkční makro klávesnici, který by byla schopna mi pomáhat, urychlovat a obohacovat moji produkci hudby.

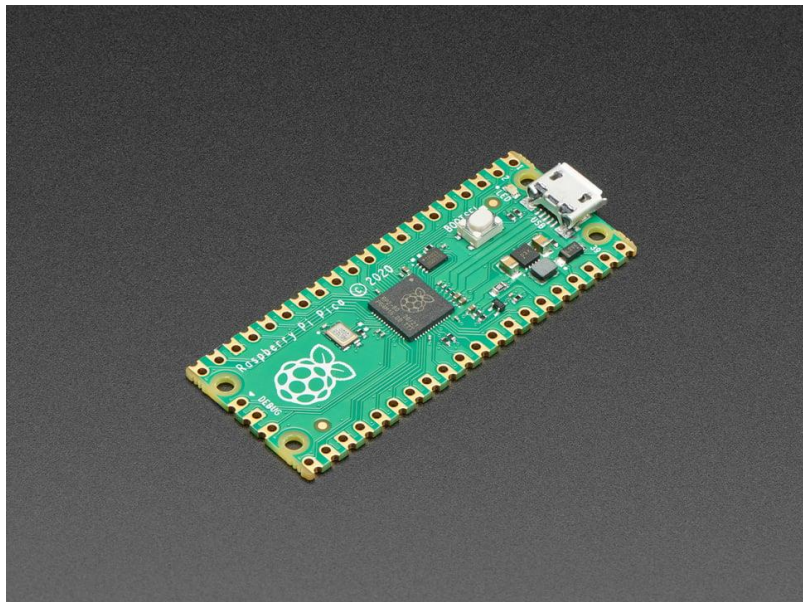
V průběhu práce, konkrétně při programování, jsem zjistil, jaký to je opravdu oříšek si něco podobného vytvořit, včetně správného zapojení a pájení.

Výsledná klávesnice splňuje nejen moje očekávání ale i stanovené požadavky a dělá to co jsem si představoval.

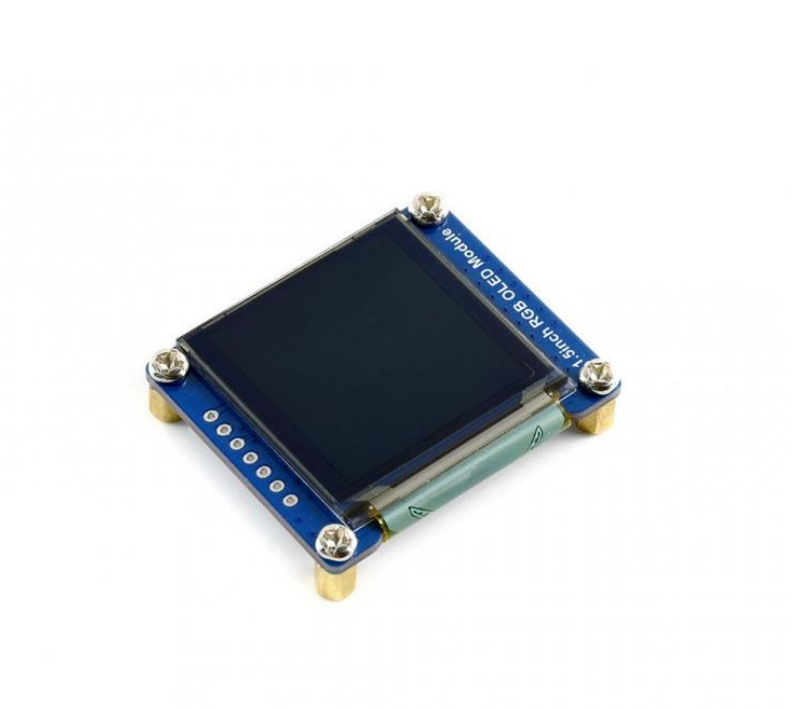
Po dokončení projektu jsem vděčný a plný nových zkušeností které jsou k nezaplacení.

11 Přílohy

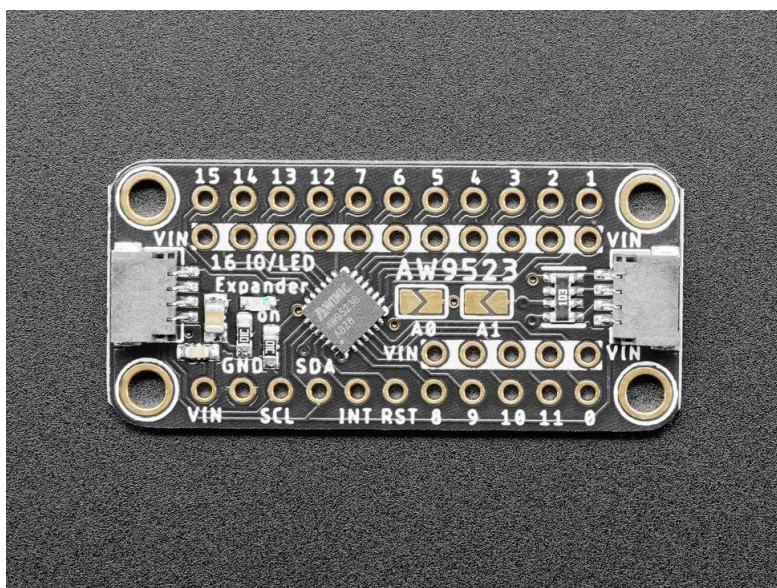
11.1 Raspberry Pi Pico



11.2 OLED displej 1.5" 128x128



11.3 AW9523 expandér a LED řadič



11.4 LED Tlačítko



11.5 5ti směrný spínač



11.6 První prototyp v den dodání tlačítek



12 Využité zdroje

<https://www.wikipedia.org/>

<https://www.arduino.cc/reference/en/libraries/>

<https://botland.cz/blog/deska-pcb-co-to-je/>

<https://www.youtube.com/>

<https://github.com/>

https://amos.odbornaskola.cz/pluginfile.php/38962/mod_resource/content/1/ArduinoPriruckaProgramatora.pdf

<https://learn.adafruit.com/welcome-to-circuitpython/creating-and-editing-code>

<https://en.wikipedia.org/wiki/CircuitPython>

<https://cs.wikipedia.org/wiki/Python>

[https://cs.wikipedia.org/wiki/Makro_\(software\)](https://cs.wikipedia.org/wiki/Makro_(software))

https://cs.wikipedia.org/wiki/Musical_Instrument_Digital_Interface

https://cs.wikipedia.org/wiki/3D_tisk