

VYSOKÉ UČENIE TECHNICKÉ V BRNE  
FAKULTA INFORMAČNÝCH TECHNÓGIÍ

Dokumentácia k projektu z predmetu IMP  
**Písanie na displeji maticovou klávesnicou**

# **Obsah**

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Technická realizácia</b>	<b>3</b>
2.1	Klávesnica . . . . .	3
2.2	Displej . . . . .	3
<b>3</b>	<b>Programová implementácia</b>	<b>4</b>
3.1	Algoritmus vycítania klávesnice . . . . .	4
3.2	Logika spracovávania vstupu . . . . .	4
<b>4</b>	<b>Užívateľská príručka</b>	<b>5</b>
4.1	Popis užívateľského rozhrania . . . . .	5
4.2	Ovládanie terminálu . . . . .	5
4.3	Písanie správ . . . . .	5
4.4	Nápoveda . . . . .	6
<b>5</b>	<b>Záver</b>	<b>7</b>
<b>A</b>	<b>Video</b>	<b>9</b>

# 1 Úvod

Cieľom projektu bol návrh a implementácia textového terminálu na platforme ESP32, ktorý simuluje písanie textových správ na starých tlačítkových telefónoch. Zariadene slúži na písanie krátkych textových správ (SMS) s využitím maticovej klávesnice a grafického displeja.

Projekt bol implementovaný v prostredí **Arduino IDE**, avšak s dôrazom na kvalitu riešenia. Aplikácia okrem základného snímania kláves a cyklického zadávania znakov, obsahuje ďalšiu funkcionality ako napríklad:

- **Vstupné režimy** – Prepínanie medzi malými písmenami, veľkými písmenami a inteligentným režimom, ktorý automaticky kapitalizuje začiatky viet.
- **Správa displeja** – Implementácia stránkovania pre text správy dlhší ako kapacita obrazovky, vizualizácia kurzora s rešpektovaním okrajov displeja a zobrazenie panelu s aktuálnymi informáciami.
- **Navigácia a editácia** – Logika posunu kurzora v štyroch smeroch v rámci správy s možnosťou spätnej upravy správy.
- **Interaktívna nápoveda** – Zobrazenie menu ovládania terminálu podržaním funkčnej klávesy \*.

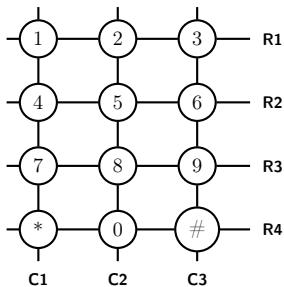
Kľúčovým technickým aspektom bola požiadavka na **manuálne** vyčítanie maticovej klávesnice bez použitia knižnice. Tento proces využívajúci GPIO vstupy a výstupy mikrokontroléra je popísaný v algoritme 1. Výsledkom je plne funkčný prototyp terminálu, ktorý umožňuje užívateľovi interaktívne písať textové SMS správy.

## 2 Technická realizácia

Na realizáciu terminálu bol použitý mikrokontrolér ESP32, konkrétny model WeMos D1 R32 [3], ktorý zabezpečuje spracovanie vstupov a komunikáciu s displejom. Jednotlivé komponenty boli zapojené na nepájivom poli (angl. *breadboard*), čo umožnilo rýchlu prototypovú realizáciu zapojenia. Používateľský vstup je realizovaný pomocou maticovej klávesnice, zatiaľ čo výstupné informácie sú zobrazované na displeji.

### 2.1 Klávesnica

Na používateľský vstup bola použitá maticová klávesnica s troma stĺpcami a štyrmi riadkami [5]. Na obrázku 1 je zobrazené rozloženie jednotlivých kláves a v tabuľke 1 je zobrazené mapovanie výstupných pinov klávesnice na jednotlivé riadky a stĺpce.



Obr. 1: Schéma rozloženia klávesnice

Pin	Riadok/Stĺpec
1	COL 2
2	ROW 1
3	COL 1
4	ROW 4
5	COL 3
6	ROW 3
7	ROW 2

Tabuľka 1: Mapovanie výstupných pinov

Jednotlivé piny klávesnice pre riadky a stĺpce boli následne pripojené na GPIO (General Purpose Input/Output) piny mikrokontroléra, konkrétnie čísla pinov je možné vidieť v tabuľkách 2 a 3.

Stĺpec	GPIO pin
COL 1	25
COL 2	26
COL 3	13

Tabuľka 2: GPIO piny pre stĺpce klávesnice

Riadok	GPIO pin
ROW 1	14
ROW 2	12
ROW 3	19
ROW 4	16

Tabuľka 3: GPIO piny pre riadky klávesnice

### 2.2 Displej

Na zobrazovanie správy bol použitý OLED displej s uhlopriečkou 0,96 palca [4]. **Synchrónna sériová** komunikácia medzi mikrokontrolérom a displejom prebieha pomocou protokolu SPI. V nasledujúcej tabuľke 4 je zobrazené mapovanie SPI pinov displeja na GPIO piny mikrokontroléra.

SPI signál	GPIO pin
CS	5
DC	27
RES	17
SCLK (D0)	18
MOSI (D1)	23

Tabuľka 4: Kofigurácia pinov displeja

### 3 Programová implementácia

Na prácu s displejom bola využité knižnice **Adafruit-GFX** [1] a **Adafruit\_SSD1306** [2]. Vlastná implementácia programu bola rozdelená do nasledovných modulov:

- **Keypad** – zabezpečuje obsluhu klávesnice a spracovanie vstupov používateľa
- **Display** – realizuje vykreslovanie informácií na displej ako aj pohyb v v správe
- **Buffer** – spravuje pole znakov pre aktuálnu správu a poskytuje základné operácie

#### 3.1 Algoritmus vyčítania klávesnice

Čítanie maticovej klávesnice je realizované metódou aktívneho skenovania stĺpcov. Každý stĺpec je postupne nastavovaný do aktívneho logického stavu (logická nula). Následne sa vyhodnocujú úrovne na riadkových vstupoch mikrokontroléra. V prípade, že je na niektorom z riadkov detegovaná aktívna logická úroveň (nízka úroveň pri použitých PULLUP rezistoroch), je možné jednoznačne určiť pozíciu stlačenej klávesy na základe kombinácie aktuálne skenovaného stĺpca a príslušného riadku. Po detekcii stlačenia klávesy je stĺpec uvedený späť do neaktívneho stavu a algoritmus vracia identifikátor zodpovedajúcej klávesy. Ak počas celého cyklu skenovania nie je detegované žiadne stlačenie, výsledkom je informácia o absencii vstupu. Tento spôsob snímania umožňuje efektívne čítanie klávesnice s minimálnym počtom vstupno-výstupných pinov a je bežne používaný pri maticových klávesničach.

---

**Algoritmus 1:** Čítanie maticovej klávesnice

---

```
for každý stĺpec c klávesnice do
    nastaviť stĺpec c na logickú 0;
    for každý riadok r klávesnice do
        if riadok r je v logickej 0 then
            detegovaná klávesa na pozícii [r, c];
            nastaviť stĺpec c na logickú 1;
            return identifikátor klávesy [r, c];
    nastaviť stĺpec c na logickú 1;
return žiadna klávesa;
```

---

#### 3.2 Logika spracovávania vstupu

Logika písania znakov je založená na spracovávaní vstupov z maticovej klávesnice a ich mapovaní na konkrétné znaky zobrazované na displeji. Po detekcii stlačenia klávesy je jej identifikátor vyhodnotený riadiacim algoritmom, ktorý rozhoduje o výslednom znaku na základe kontextu aktuálneho vstupu. Implementovaná je podpora viacnásobného stlačenia tej istej klávesy v definovanom časovom intervale, čo umožňuje cyklické prepínanie medzi viacerými znakmi priradenými k jednej klávese. Ak čas medzi jednotlivými stlačeniami prekročí stanovenú hranicu, aktuálny znak je potvrdený a kurzor sa posunie na nasledujúcu pozíciu.

Súčasťou logiky písania je aj rozlišovanie medzi krátkym a dlhým stlačením klávesy. Krátke stlačenie slúži na zadávanie znakov, zatiaľ čo dlhé stlačenie je využité na vyvolanie špeciálnych funkcií, ako je napríklad pohyb po displeji alebo zobrazenie pomocných informácií. Tento prístup umožňuje efektívnu prácu s terminálom aj pri obmedzenom počte vstupných prvkov.

## 4 Užívateľská príručka

V tejto kapitole je stručne popísane užívateľské rozhranie, ovládanie terminálu pomocou kláves a písanie správ na termináli, ako aj návod na zobrazenie nápovedy.

### 4.1 Popis užívateľského rozhrania

Displej je rozdelený na dve hlavné časti a to **Stavový riadok** a **Textovú oblasť**. V tabuľke 5 je popísané rozloženie a vysvetlenie jednotlivých prvkov stavového riadku.

Pozícia	Formát	Význam
<b>Vľavo</b>	abc, ABC, Abc	<b>Vstupný režim:</b> Aktuálne zvolený režim veľkosti písmen.
<b>Stred</b>	Line X	<b>Riadok:</b> Číslo riadku, na ktorom sa nachádza kurzor.
<b>Vpravo</b>	Znaky/Strana	<b>Štatistika:</b> Prvé číslo udáva počet zostávajúcich znakov z maximálnej dĺžky správy 160 znakov [7]. Druhé číslo označuje aktuálnu stránku zobrazenú na displeji.

Tabuľka 5: Rozloženie a význam prvkov stavového riadku

V textovej oblasti je zobrazená aktuálna správa s aktuálnou pozíciovou pomocou kurzoru. V rámci správy je možné sa pohybovať a následne editovať alebo mazať jednotlivé znaky.

### 4.2 Ovládanie terminálu

Ovládanie terminálu je realizované pomocou maticovej klávesnice, pričom funkcia jednotlivých kláves sa mení v závislosti od dĺžky stlačenia a aktuálneho kontextu. Hlavné navigačné prvky sú namapované na číselné klávesy:

- **Klávesa 0:** Dlhé podržanie klávesy vymaže celú správu a nastaví kurzor na začiatok správy.
- **Klávesy 2, 4, 6, 8:** Slúžia na pohyb kurzora v štyroch smeroch. Pri dosiahnutí okraja obrazovky dochádza k automatickému zalamovaniu alebo prepnutiu stránky.
- **Klávesa 5:** Dlhé podržanie klávesy „odošle“ správu a užívateľ môže následne pokračovať v písaní novej správy.
- **Klávesa #:** Vykonáva funkciu *Backspace* – zmaže znak pred alebo pod kurzorom.
- **Klávesa \*:** Krátkym stlačením cyklicky mení režimy zadávania textu (**abc** → **ABC** → **Abc**).

### 4.3 Písanie správ

Zadávanie textu prebieha metódou **Multi-tap** [6], ktorá je typická pre telefóny s numerickou klávesnicou. Opakoványm stláčaním tej istej klávesy v krátkom časovom intervale užívateľ prepína medzi dostupnými znakmi danej klávesy. Znaky pre jednotlivé klávesy sú zobrazené v tabuľke 6.

Klávesa	Priradené znaky
0	„0
1	. , ? ! 1
2	a b c 2
3	d e f 3
4	g h i 4
5	j k l 5
6	m n o 6
7	p q r s 7
8	t u v 8
9	w x y z 9

Tabuľka 6: Mapovanie kláves na znaky

Špeciálnou funkciou je inteligentný režim **Smart Case** (indikovaný ako **Abc**). V tomto režime terminál analyzuje predchádzajúci text, ak deteguje koniec vety, automaticky nastaví nasledujúci znak ako veľké písmeno. Táto funkcionalita výrazne zrýchľuje písanie súvislých správ bez nutnosti manuálneho prepínania režimov.

#### 4.4 Nápoveda

Pre užívateľa je k dispozícii v prípade potreby nápoveda s popisom ovládacičích prvkov terminálu. Pre zobrazenie nápovedy je potrebné podržať na klávesnici tlačidlo \*, pre zavretie nápovedy a návrat do aktuálnej pozície v správe stačí dané tlačidlo pustiť.

0: Clear	2: UP
5: Send	4: LEFT
*: Mode	6: RIGHT
#: Del	8: DOWN

Tabuľka 7: Obsah zobrazenej nápovedy

## 5 Záver

V rámci projektu sa podarilo úspešne navrhnúť a implementovať textový terminál s využitím mikrokontroléra ESP32. Hlavným prínosom riešenia je manuálne vyčítanie maticovej klávesnice, ktorá bola realizovaná výhradne prostredníctvom priamej manipulácie s GPIO vstupmi a výstupmi mikrokontroléra, čím bola v plnom rozsahu splnená kľúčová požiadavka zadania na nepoužívanie externých kničníč na prácu s klávesnicou. Na implementáciu aplikácie bolo využité prostredie **Arduino**, pričom bol kladený dôraz na výslednú kvalitu riešenia.

## Referencie

- [1] Adafruit Industries. *Adafruit GFX Graphics Library*. Arduino library providing common graphics functions for displays. Adafruit Industries. 2025. URL: <https://github.com/adafruit/Adafruit-GFX-Library>.
- [2] Adafruit Industries. *Adafruit SSD1306 OLED Display Library*. Arduino library for SSD1306-based monochrome OLED displays. Adafruit Industries. 2025. URL: [https://github.com/adafruit/Adafruit\\_SSD1306](https://github.com/adafruit/Adafruit_SSD1306).
- [3] AZ-Delivery / Halloweenfreak.de. *D1 R32 ESP32 Board Documentation (English)*. D1\_R32\_ENG.pdf — English user manual and specifications. AZ-Delivery / Halloweenfreak.de. 2025. URL: [https://www.halloweenfreak.de/arduino/pdfs/D1\\_R32\\_ENG.pdf](https://www.halloweenfreak.de/arduino/pdfs/D1_R32_ENG.pdf).
- [4] Dragonwake / LCDWIKI. *0.96 inch OLED SPI Module MSP096X User Manual (English)*. User manual and technical specifications for the MSP096X OLED SPI module (Rev. 1.0). Dragonwake Technology / LCDWIKI. 2019. URL: [https://www.dragonwake.com/download/LCD/0.96inch-spi/0.96inch\\_OLED\\_SPI\\_Module\\_MSP096X\\_User\\_Manual\\_EN.pdf](https://www.dragonwake.com/download/LCD/0.96inch-spi/0.96inch_OLED_SPI_Module_MSP096X_User_Manual_EN.pdf).
- [5] Premier Farnell / element14 / Newark. *Keypad Datasheet*. Version 1.1. Premier Farnell plc. Jún 2012. URL: <https://www.farnell.com/datasheets/1662617.pdf>.
- [6] Wikipedia contributors. *Multi-tap*. Accessed: 2025-12-21. Wikimedia Foundation. 2025. URL: <https://en.wikipedia.org/wiki/Multi-tap>.
- [7] Wikipedia contributors. *SMS*. Accessed: 2025-12-21. Wikimedia Foundation. 2025. URL: <https://en.wikipedia.org/wiki/SMS>.

## A Video

<https://youtube.com/shorts/qLNc-jT4QBc?feature=share>