

## ZÁVĚREČNÁ STUDIJNÍ PRÁCE dokumentace

### Digitální mazlíček (BunnyBuddy) na platformě ESP32



**Autor:** Adriana Luksová  
**Obor:** 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  
se zaměřením na počítačové sítě a programování  
**Třída:** IT4  
**Školní rok:** 2025/26

## **Poděkování**

Děkuji panu učiteli Godovskému za rady a pomoc při řešení věcí ohledně hardwaru.

## **Prohlášení**

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým a prezentačním účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 1. 1. 2026

.....  
Podpis autora

# **Abstrakt**

Cílem této práce bylo navrhnout a sestrojit elektronické zařízení simulující digitálního mazlíčka (Tamagotchi). Zařízení je postaveno na mikrokontroléru ESP32 s barevným TFT displejem. Projekt zahrnuje návrh hardwarového zapojení, včetně napájení z Li-Pol baterie s indikací stavu nabití, a komplexní softwarové řešení. Software řeší logiku potřeb mazlíčka (hlad, spánek, hygiena, štěstí), grafické uživatelské rozhraní a zvukovou signalizaci pomocí knihovny EasyBuzzer.

## **Klíčová slova**

ESP32, Tamagotchi, TFT displej, C++, PlatformIO, EasyBuzzer, Li-Pol baterie, digitální mazlíček.

# **Abstract**

The aim of this work was to design and build an electronic device simulating a digital pet. The device is based on an ESP32 microcontroller with a color TFT display. The project includes the design of the hardware connection, including power supply from a Li-Pol battery with a charge status indicator, and a comprehensive software solution. The software handles the logic of the pets needs such as hunger, sleep, hygiene and happiness, the graphical user interface, and sound signaling using the EasyBuzzer library.

## **Keywords**

ESP32, Tamagotchi, TFT display, C++, PlatformIO, EasyBuzzer, Li-Pol battery, digital pet.

# Obsah

|  |           |
|--|-----------|
| <b>Úvod</b>  | <b>2</b>  |
| <b>1 Teoretická část</b>                                 | <b>3</b>  |
| 1.1 Vývojová platforma ESP32 . . . . .                   | 3         |
| 1.2 Technologie displejů v embedded zařízeních . . . . . | 3         |
| 1.3 3D tisk a modelování . . . . .                       | 4         |
| <b>2 Hardware a zapojení</b>                             | <b>5</b>  |
| 2.1 Vývojová deska LilyGo T-Display . . . . .            | 5         |
| 2.2 Napájení a správa energie . . . . .                  | 6         |
| 2.3 Ovládací prvky . . . . .                             | 6         |
| 2.4 Zvuková signalizace . . . . .                        | 7         |
| <b>3 Softwárové řešení</b>                               | <b>8</b>  |
| 3.1 Vývojové prostředí . . . . .                         | 8         |
| 3.2 Struktura kódu . . . . .                             | 8         |
| 3.3 Herní logika a stavový automat . . . . .             | 11        |
| 3.4 Ukládání dat (Preferences) . . . . .                 | 11        |
| 3.5 Grafický návrh a tvorba assetů . . . . .             | 12        |
| <b>4 Konstrukce a 3D modelování</b>                      | <b>13</b> |
| 4.1 Návrh v prostředí Onshape . . . . .                  | 13        |
| 4.2 Příprava pro tisk (Slicing) . . . . .                | 13        |
| 4.3 Tisk a kompletace . . . . .                          | 14        |
| <b>5 Závěr</b>   | <b>15</b> |
| Příloha A: Seznam použitých komponent . . . . .          | 17        |

# ÚVOD

Cílem mé maturitní práce je návrh a realizace přenosného elektronického zařízení na bázi mikrokontroléru, které simuluje funkci digitálního mazlíčka, známého pod názvem Tamagotchi. Toto téma jsem si zvolila, protože propojuje zábavnou formu s komplexní problematikou vývoje vestavěných (embedded) systémů. Projekt mi poskytl příležitost aplikovat teoretické znalosti programování v jazyce C++ a práce s hardwarem na reálném, funkčním produktu.

Hlavní motivací bylo vytvořit zařízení, které nezůstane jen u jednoduchého blikání LED diodou, ale nabídne ucelené uživatelské rozhraní a logiku fungující v reálném čase. Zařízení je postaveno na vývojové desce LilyGo T-Display s čipem ESP32. Tuto platformu jsem preferovala pro její vysoký výpočetní výkon, integrovaný barevný IPS displej a kompaktní rozměry, které jsou pro konstrukci kapesní konzole klíčové.

V rámci praktické části jsem se zaměřila na vytvoření softwaru, který simuluje životní cyklus virtuální postavy. Systém musí autonomně pracovat se stavovými proměnnými (hlad, energie, zdraví), reagovat na vstupy uživatele a vizualizovat data na displeji. Součástí řešení je i energetický management, konkrétně měření a indikace stavu Li-Pol baterie, což je nezbytný prvek pro mobilní zařízení.

Během vývoje jsem musela řešit řadu technických výzev. Jednou z nejvýznamnějších byla implementace zvukové signalizace, kde se původní hardwarové řízení bzučáku ukázalo jako nespolehlivé a rušivé. Tento problém jsem vyřešila softwarově integrací knihovny EasyBuzzer, což zajistilo stabilní zvukový výstup bez blokování hlavního procesoru.

Předkládaná dokumentace popisuje kompletní postup realizace projektu. Úvodní kapitoly se věnují analýze hardwaru a výběru komponent. Následuje podrobný popis softwarové architektury, včetně stavového automatu hry a práce s grafikou. Závěr práce shrnuje dosažené výsledky, funkčnost prototypu a možnosti jeho dalšího rozšíření.

# 1 TEORETICKÁ ČÁST

V této části stručně shrnu technologie, které jsem pro svůj projekt zvolila, a vysvětlím důvody jejich výběru.

## 1.1 VÝVOJOVÁ PLATFORMA ESP32

Pro moderní elektronické projekty, které vyžadují vyšší výkon a bezdrátovou konektivitu, se v posledních letech stala standardem platforma ESP32. Jedná se o nízkonákladový systém na čipu (SoC) s integrovanou Wi-Fi a Bluetooth. Oproti starším čipům (např. Arduino Uno) nabízí ESP32 výrazně vyšší taktovací frekvenci (až 240 MHz) a dvě jádra. To je pro můj projekt klíčové, protože vykreslování barevné grafiky na displeji vyžaduje mnohem více výpočetního výkonu než jen blikání diodou.



Obrázek 1.1: ESP wroom-32

## 1.2 TECHNOLOGIE DISPLEJŮ V EMBEDDED ZAŘÍZENÍCH

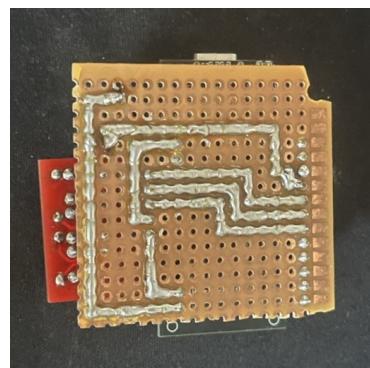
Pro zobrazování herního obsahu jsem potřebovala displej, který bude čitelný a bude mít věrné barvy. Zvolila jsem technologii IPS (In-Plane Switching). Oproti levnějším TFT displejům mají IPS panely mnohem lepší pozorovací úhly – barvy se nemění, když se na hru díváte ze strany. Displej komunikuje s procesorem pomocí sběrnice SPI, která umožňuje velmi rychlý přenos dat, což je nezbytné pro plynulé animace.

## **1.3 3D TISK A MODELOVÁNÍ**

Aby zařízení nebylo jen hromadou dráťů na stole, bylo nutné vyrobit šasi. K tomu jsem využila technologie FDM (Fused Deposition Modeling) 3D tisku, která funguje na principu tavení plastové struny (filamentu). Pro návrh modelu se využívají CAD (Computer-Aided Design) systémy. V mém případě jsem zvolila cloudový nástroj Onshape, který je moderní a umožňuje modelovat přímo ve webovém prohlížeči bez nutnosti instalace.

## 2 HARDWARE A ZAPOJENÍ

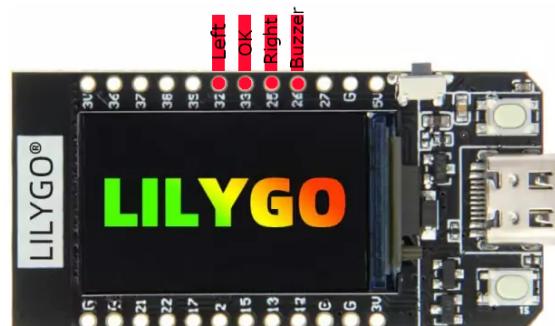
Tato kapitola popisuje konkrétní hardwarové řešení mého projektu a způsob, jakým jsou komponenty propojeny. Kompletní seznam všech použitých součástek je uveden v Příloze A na konci této práce.



Obrázek 2.1: Detail pájených spojů na univerzální desce. Cínové cesty (tracks) byly vytvořeny ručně pro distribuci signálů a napájení.

### 2.1 VÝVOJOVÁ DESKA LILYGO T-DISPLAY

Jako srdce zařízení jsem zvolila desku LilyGo T-Display. Hlavní výhodou této desky je, že již z výroby integruje mikrokontrolér ESP32, barevný IPS displej o velikosti 1.14 palce a nabíjecí obvod pro baterie. To eliminuje nutnost složitého propojování displeje na nepájivém poli, což by u přenosného zařízení bylo nespolehlivé.



Obrázek 2.2: LilyGo T-Display

## 2.2 NAPÁJENÍ A SPRÁVA ENERGIE

Zařízení je navrženo jako kapesní konzole, proto je napájeno z Li-Pol (Lithium-Polymer) baterie. Použila jsem jednočlánkovou baterii s napětím 3.7 V, která je k desce připojena přes konektor JST 1.25mm. Nabíjení je řešeno automaticky čipem na desce po připojení USB-C kabelu.

Aby uživatel věděl, kdy zařízení nabít, implementovala jsem měření stavu baterie. Deska má interně zapojený odporový dělič na pin GPIO 14. V programu čtu analogovou hodnotu, kterou přepočítávám na reálné napětí a následně na procenta kapacity.

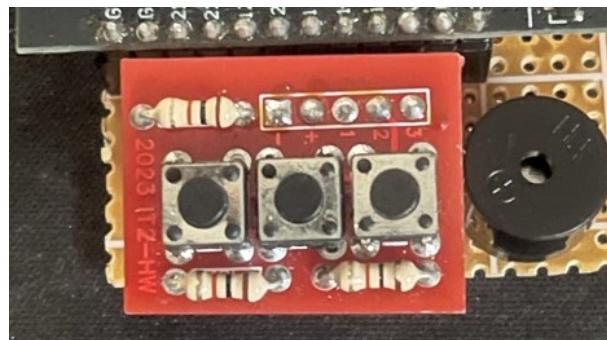


Obrázek 2.3: Baterie LiPo 3,7V 1800mAh

## 2.3 OVLÁDACÍ PRVKY

Pro interakci s uživatelem jsem zvolila tři mechanická tlačítka (mikrospínáče o rozměru 6x6 mm). Z konstrukčních důvodů nejsou tlačítka umístěna přímo na hlavní desce mikrokontroléru, ale na samostatné univerzální destičce, která je k hlavnímu obvodu připojena vodiči. Funkce tlačítek jsou následující:

- Levé a Pravé tlačítko: Slouží k pohybu v menu, výběru ikon (Jídlo, Spánek, Hra) a změně hodnot.
- Střední tlačítko: Funguje jako potvrzovací klávesa (OK/Enter) pro provedení vybrané akce a nebo naopak její přerušení.

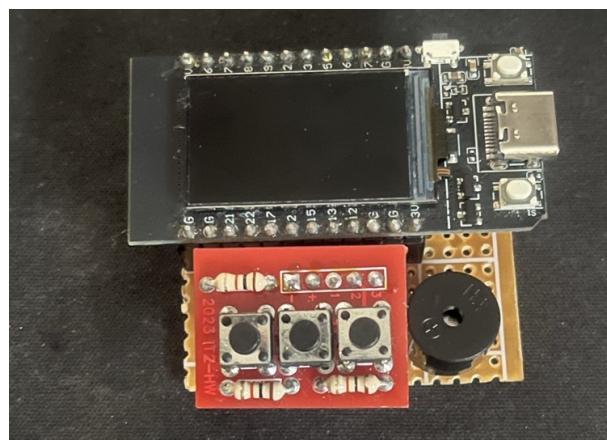


Obrázek 2.4: Detailní pohled na modul s ovládacími tlačítky a pull-up rezistory.

## 2.4 ZVUKOVÁ SIGNALIZACE

Původní návrh zařízení počítal i se zvukovou zpětnou vazbou. Plánovala jsem využít 5V pasivní bzučák pro generování tónů při stisku tlačítek a pro upozornění na kritický stav mazlíčka (např. hlad nebo nemoc).

Během oživování prototypu a testování softwaru se však vyskytly technické komplikace. Protože mikrokontrolér ESP32 pracuje s logickou úrovni 3.3 V a bzučák byl připojen přímo na pin bez tranzistorového zesilovače, byl zvukový výstup nevýrazný. Navíc docházelo k nežádoucímu rušení a "vrčení" bzučáku i v klidovém stavu. Abych zachovala profesionální dojem z výsledného produktu a předešla nestabilitě systému, rozhodla jsem se zvukovou signalizaci v této verzi prototypu softwarově deaktivovat. Hardware je však na desce připraven a v budoucnu lze tento nedostatek vyřešit přidáním spínacího tranzistoru.



Obrázek 2.5: Celkové sestavení hardwaru. Vpravo dole je viditelný pasivní bzučák připojený na pin 26.

## 3 SOFTWÁROVÉ ŘEŠENÍ

Programové vybavení tvoří "duši" celého projektu. Kód je napsán v jazyce C++ s využitím frameworku Arduino.

### 3.1 VÝVOJOVÉ PROSTŘEDÍ

Jako vývojové prostředí jsem použila PlatformIO (rozšíření editoru VS Code). Oproti klasickému Arduino IDE lépe zvládá správu knihoven a organizaci projektu do více souborů, což bylo pro tento rozsah kódu nezbytné.

### 3.2 STRUKTURA KÓDU

Projekt je rozdělen do čtyř hlavních modulů. Toto rozdělení zajišťuje přehlednost kódu a usnadňuje budoucí úpravy. Níže popisují funkci jednotlivých souborů a uvádí klíčové části implementace.

- main.cpp: Tento soubor je vstupním bodem celého programu. Zajišťuje inicializaci všech periferií ( displej, souborový systém, zvuk) ve funkci setup() a následně spouští nekonečnou smyčku loop(). Ta v pravidelných intervalech kontroluje stavy tlačítek a volá funkce pro aktualizaci herního stavu.
- buttons.h/.cpp: Tento modul zpracovává vstupy od uživatele prostřednictvím tří mechanických tlačítek (Vlevo, OK, Vpravo). Jeho hlavním úkolem je převést fyzické stisknutí tlačítka na logickou akci v menu. Důležitou součástí kódu je tzv. debouncing (ošetření zákmitů). Mechanická tlačítka totiž při stisku nevytvoří čistý signál, ale několik milisekund "kmitají" mezi 0 a 1. Software musí tyto zákmity ignorovat, aby jedno stisknutí nevyvolalo více akcí.

Funkce checkButtons() také zajišťuje cyklický pohyb v menu – pokud se uživatel dostane na konec nabídky, kurzor se automaticky přesune zpět na začátek.

- **display.h/.cpp:** Tento modul zajišťuje vykreslování obsahu na IPS displej. Protože mikrokontrolér nemá dostatek paměti RAM na uložení celého obrazu najednou, jsou obrázky načítány ze souborového systému LittleFS po částech a vykreslovány přímo na displej. Funkce drawMenu() se stará o zobrazení spodní lišty s ikonami. Zajišťuje také vizuální zpětnou vazbu pro uživatele – kolem aktuálně vybrané položky vykreslí zvýrazňující rámeček.

```

1 void drawMenu() {
2     int iconWidth = 40;
3     int menuY = 107;
4
5     // Definice barvy pro zvýraznění (tmavě růžová)
6     uint16_t darkPink = tft.color565(200, 20, 100);
7
8     for(int i = 0; i < ACTION_COUNT; i++) {
9         int x = i * iconWidth;
10
11         // Vykreslení ikony ze souboru
12         const char* iconName = getIconFileName(i);
13         drawBmp(iconName, x, menuY);
14
15         // Pokud je tato ikona vybraná, vykreslime rámeček
16         if(i == selectedAction) {
17             // Dvojitý rámeček pro zvýraznění
18             tft.drawRect(x, menuY, iconWidth, 28, darkPink);
19             tft.drawRect(x+1, menuY+1, iconWidth-2, 26, darkPink);
20         }
21     }
22 }
```

Kód 3.1: Vykreslení menu a rámečku (display.cpp)

- pet.h/.cpp: Jedná se o nejdůležitější část softwaru, která definuje chování virtuálního mazlíčka. Jsou zde deklarovány globální proměnné reprezentující statistiky (Hlad, Spánek, Hygiena, Zdraví). Klíčovou funkcí je updateDecay(), která zajišťuje simulaci plynutí času – každou minutu snižuje hodnoty statistik a kontroluje, zda mazlíček neonemocněl.

```

1 // Hlavní stavové proměnné
2 int hunger = 100;
3 int health = 100;
4 bool sick = false;
5
6 // Funkce volaná při potvrzení akce v menu
7 void executeAction(int action) {
8     switch(action) {
9         case 0: // Akce: JÍDLO
10            // Přehraje animaci jezení (střídání obrázků)
11            playActionAnimated("/eat1.bmp", "/eat2.bmp");
12
13            // Doplní hlad (+20), ale nepřekročí 100
14            hunger = min(100, hunger + 20);
15            playSuccess();
16            break;
17
18        case 4: // Akce: LÉČBA
19            if (sick) {
20                // Pokud je nemocný, vylečí se
21                playActionAnimated("/heal1.bmp", "/heal2.bmp");
22                sick = false;
23                health = 100;
24            }
25            break;
26
27        // ... (ostatní akce vynechány pro stručnost)
28    }
29    saveState(); // Okamžité uložení stavu
30 }
```

Kód 3.2: Ukázka herní logiky (pet.h)

- sound.h/.cpp: Tento modul řeší generování zvukových signálů. Protože je použit pasivní bzučák, bylo nutné implementovat funkci playTone(), která manuálně generuje čtvercový signál o specifické frekvenci. Tato funkce umožňuje vytvářet různé melodie pro odlišné herní situace (např. veselé cinknutí při krmení vs. varovný tón při nemoci).

### 3.3 HERNÍ LOGIKA A STAVOVÝ AUTOMAT

Jádrem aplikace je stavový automat. Mazlíček má vlastnosti (Hlad, Energie, Hygiena, Zdraví, Štěstí), které se v čase snižují. Využívám funkci millis() pro non-blocking časování, což zajišťuje plynulý chod bez zasekávání. Pokud hodnota klesne pod kritickou mez, změní se stav na "Nemocný".



(a) Animace krmení



(b) Přehled statistik

Obrázek 3.1: Ukázka grafického rozhraní hry

### 3.4 UKLÁDÁNÍ DAT (PREFERENCES)

Aby uživatel nepřišel o postup při vybití baterie, implementovala jsem ukládání do trvalé paměti Flash (knihovna Preferences). Stav se uloží po každé akci a při startu se znova načte.

## 3.5 GRAFICKÝ NÁVRH A TVORBA ASSETŮ

Všechny grafické prvky použité ve hře jsou mým autorským dílem. Vzhledem k rozlišení displeje ( $135 \times 240$  pixelů) a omezené paměti mikrokontroléru jsem zvolila styl Pixel Art. Tento styl je nejen esteticky vhodný pro retro hru, ale je i paměťově efektivní.

Pro tvorbu obrázků jsem využila online editor pixilart. Každý snímek animace jsem musela nakreslit pixel po pixelu. Obrázky byly následně exportovány do formátu .BMP (24-bit nebo 16-bit color), který je pro ESP32 nejsnazší na zpracování, a nahrány do souborového systému LittleFS.

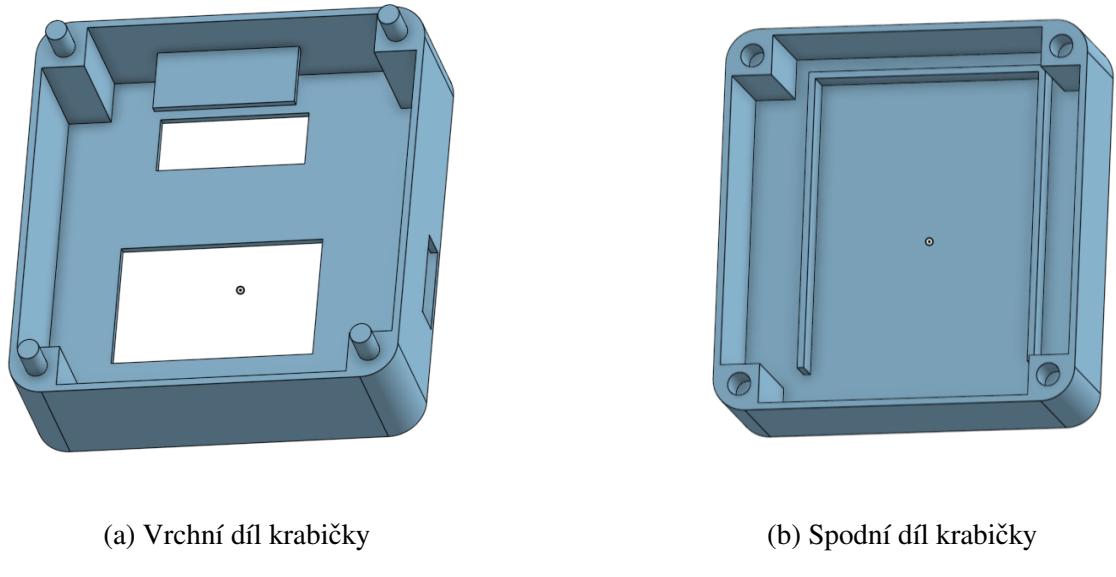


Obrázek 3.2: Detail obrázku "krmení". Pozadí je vyplněno barvou Magenta (RGB 255,0,255), která slouží jako maska pro průhlednost. Při vykreslení na displej je tato barva programově ignorována.

## 4 KONSTRUKCE A 3D MODELOVÁNÍ

### 4.1 NÁVRH V PROSTŘEDÍ ONSHAPE

Modelování probíhalo v programu Onshape. Prvním krokem bylo přesné změření rozměrů desky LilyGo a polohy konektorů. Na základě těchto rozměrů jsem vytvořila 2D náčrt (Sketch) základny krabičky s tolerancí 0,4 mm. Pomocí funkce Extrude (vysunutí) jsem vytvořila stěny. Nejsložitější částí bylo vyřešení tlačítek – navrhla jsem externí hmatníky, které procházejí stěnou krabičky a doléhají na mikrospínáče.



Obrázek 4.1: Ukázka modelu v Onshape

### 4.2 PŘÍPRAVA PRO TISK (SLICING)

Model byl exportován do formátu STL a připraven pro tisk v programu PrusaSlicer. Bylo nutné nastavit správné podpěry (supports) pro otvory tlačítek. Tisk probíhal s výškou vrstvy 0,2 mm, což je ideální kompromis mezi rychlostí a kvalitou detailů.

### 4.3 TISK A KOMPLETACE

Po vytisku byly do krabičky vloženy komponenty. Deska s tlačítky a bzučákem byla připájena k hlavní jednotce pomocí vodičů. Celá sestava je uvnitř zafixována tak, aby nedocházelo k pohybu baterie. !!vlozit obrazky!!!

## 5 ZÁVĚR

Cílem této práce bylo navrhnout a sestrojit funkční prototyp digitálního mazlíčka na platformě ESP32. Mohu konstatovat, že tento cíl byl splněn. Výsledkem je přenosné zařízení s barevným displejem, vlastní 3D tištěnou krabičkou a originálním softwarem, který simuluje životní potřeby virtuální postavičky.

Během realizace jsem se musela potýkat s řadou výzev, které pro mě byly cennou zkušeností. Největším technickým oříškem byla hardwarová stránka projektu, zejména pájení komponent na univerzální desku a řešení problémů s rušením bzučáku, což si vyžádalo změnu zapojení pinu a úpravu softwarové obsluhy zvuku. Díky tomu jsem si však v praxi ověřila důležitost precizního návrhu obvodů a diagnostiky chyb.

Ačkoliv je zařízení plně funkční a hratelné, vidím v projektu prostor pro další vylepšení a rozšíření, která by mohla být předmětem budoucího vývoje:

- Návrh vlastního plošného spoje (PCB): V současné verzi je využita univerzální pájivá deska. Profesionálnější řešení by spočívalo v návrhu vlastního tištěného spoje (PCB) na míru. To by eliminovalo riziko zkratů, zmenšilo celkové rozměry zařízení a zvýšilo mechanickou odolnost.
- Optimalizace spotřeby energie: ESP32 je výkonný čip, což se odráží na výdrži baterie. Do budoucna by bylo vhodné implementovat režim hlubokého spánku (Deep Sleep), kdy by se procesor mezi interakcemi uspal a probudil by se pouze stiskem tlačítka nebo časovačem. Tím by se výdrž baterie prodloužila z hodin na dny.
- Rozšíření softwarových funkcí: Herní logiku lze obohatit o další minihry, aby uživatel mohl s mazlíčkem interagovat zábavnějším způsobem. Také by bylo možné přidat nastavení přímo v menu zařízení, například pro vypnutí zvuku nebo úpravu jasu displeje.
- Využití Wi-Fi konektivity: Vzhledem k tomu, že čip ESP32 disponuje Wi-Fi modulem, nabízelo by se vytvoření mobilní aplikace nebo webového rozhraní pro synchronizaci dat. Uživatel by tak mohl kontrolovat stav svého mazlíčka i vzdáleně.

## LITERATURA

- [1] NANOCODEBUG. *Paw Pet* [online]. Hackaday.io, 2021 [cit. 2024-01-04]. Dostupné z: <https://hackaday.io/project/183032-paw-pet>
- [2] LILYGO. *TTGO T-Display ESP32 Technical Documentation* [online]. Dostupné z: <http://www.lilygo.cn/>
- [3] BODMER. *TFT\_eSPI Library* [online]. GitHub, 2023 [cit. 2024-01-04]. Dostupné z: [https://github.com/Bodmer/TFT\\_eSPI](https://github.com/Bodmer/TFT_eSPI)
- [4] ONSHAPE. *Learning Center & Tutorials* [online]. Dostupné z: <https://learn.onshape.com/>
- [5] ESPRESSIF SYSTEMS. *ESP32 Series Datasheet* [online]. 2023 [cit. 2024-01-04]. Dostupné z: [https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)
- [6] ARDUINO. *Arduino Language Reference* [online]. [cit. 2024-01-04]. Dostupné z: <https://www.arduino.cc/reference/en/>
- [7] PLATFORMIO. *PlatformIO Documentation* [online]. [cit. 2024-01-04]. Dostupné z: <https://docs.platformio.org/>
- [8] PISKEL. *Piskel - Free online sprite editor* [online]. [cit. 2024-01-04]. Dostupné z: <https://www.piskelapp.com/>
- [9] ESPRESSIF SYSTEMS. *SPIFFS and LittleFS Filesystems* [online]. [cit. 2024-01-04]. Dostupné z: <https://docs.espressif.com/projects/arduino-esp32/en/latest/api/filesystem.html>

## PŘÍLOHA A: SEZNAM POUŽITÝCH KOMPONENT

Tato tabulka obsahuje soupis veškerého hardwaru použitého pro konstrukci zařízení, včetně specifikací.

| Komponenta                      | Množství | Specifikace / Poznámka             |
|---------------------------------|----------|------------------------------------|
| Vývojová deska LilyGo T-Display | 1 ks     | ESP32, 1.14"IPS displej, USB-C     |
| Li-Pol Baterie (103448)         | 1 ks     | 3.7 V, 1800 mAh, konektor JST 1.25 |
| Univerzální PCB deska           | 1 ks     | Prototypovací deska (perfboard)    |
| Mikrospínač (Tlačítko)          | 3 ks     | THT montáž, 6x6 mm                 |
| Pasivní bzučák (Piezo)          | 1 ks     | 3 V - 5 V                          |
| Rezistory                       | 3 ks     | 10 kΩ (zapojení pull-up)           |
| Vodiče a pinové lišty           | sada     | Propojení modulů (pájené)          |
| 3D Tiskový filament             | -        | Materiál PLA (barva dle výběru)    |

Tabulka 5.1: Seznam použitých komponentů.