

Stredná priemyselná škola informačných technológií
Nábřežná 1325, 024 01 Kysucké Nové Mesto

ARDUINO HERNÁ KONZOLA

Vlastný projekt na praktickú časť odbornej zložky maturitnej skúšky

Vedúci projektu: Ing. Peter Remiš

Autor projektu: Norbu Borbély IV.CI

Šk. rok: 2024/2025

Stredná priemyselná škola informačných technológií
Nábřežná 1325, 024 01 Kysucké Nové Mesto

ARDUINO HERNÁ KONZOLA

Vlastný projekt na praktickú časť odbornej zložky maturitnej skúšky

Študijný odbor: 2561 M - informačné a sieťové technológie

Vedúci projektu: Ing. Peter Remiš

Autor projektu: Norbu Borbély IV.CI

Šk. rok: 2024/2025

Čestné prehlásenie

Čestne prehlasujem, že som predkladanú prácu vypracoval samostatne, za použitia uvedenej literatúry.

V Kysuckom Novom Meste, dňa

.....

podpis

ABSTRAKT

Táto práca sa zaoberá vytvorením projektu na platforme Arduino Nano, na ktorej je možné hrať hru Tetris. Celý projekt je navrhnutý a implementovaný s využitím vlastných knižníc, okrem knižnice Arduino.h. Výsledkom je plne funkčná hra bežiaca na mikrokontroléry Arduino Nano, ktorá efektívne využíva hardvérové aj softvérové zdroje.

Kľúčové slová: Arduino Nano, softvér, hardvér, vlastné knižnice, Tetris, herná konzola, Visual Studio Code, PlatformIO

Rozsah: 20 s. vrátane príloh, z toho 19 s. textovej časti

ABSTRACT

This work focuses on the creation of a project on the Arduino Nano platform, where it is possible to play the game Tetris. The entire project is designed and implemented using custom libraries, except for the Arduino.h library. The result is a fully functional game running on a microcontroller Arduino Nano, that efficiently utilizes both hardware and software resources.

Keywords: Arduino Nano, software, hardware, custom libraries, Tetris, game console, Visual Studio Code, PlatformIO

Size: 20 p. including appendix, 19 p. of main part

OBSAH

0	Úvod	7
1	Problematika a prehľad literatúry	8
1.1	Mikrokontrolér Arduino Nano	8
1.2	LcdTft Display	9
1.3	PCB	10
1.4	Visual Studio Code	11
	Zdroj: https://seekvectors.com/post/visual-studio-code-logo	11
1.5	PlatformIO	11
2	Ciele práce	13
3	Materiál a metodika	14
3.1	Navrhovanie hardvéru	14
3.1.1	Výber komponentov	14
3.1.2	Schéma zapojenia	15
3.1.3	Návrh PCB	16
3.1.4	3D modelovanie puzdra	17
3.2	Príprava hardvéru	17
3.3	Príprava vývojového prostredia	18
3.4	Vytvorenie zdrojového kódu	19
3.4.1	Vytváranie vlastných knižníc	19
3.4.2	Integrácia knižníc a hernej logiky	22
4	VÝSLEDKY PRÁCE A DISKUSIA	23
5	ZÁVERY PRÁCE	24
6	ZHRNUTIE	25
	Zoznam použitej literatúry	26

ZOZNAM TABULIEK, GRAFOV A ILUSTRÁCIÍ

ZOZNAM TABULIEK

Tab. 1: Zoznam komponentov	15
----------------------------------	----

ZOZNAM ILUSTRÁCIÍ

Obr. 1: Mikrokontrolér Arduino Nano	9
Obr. 2: LCDTFT Display	10
Obr. 3: PCB Doska	10
Obr. 4: Logo Visual Studio Code	11
Obr. 5: Logo PlatformIO	12
Obr. 6: Schéma zapojenia	15
Obr. 7: PCB návrh	16
Obr. 8: 3D model	17
Obr. 9: Metóda LCD_write()	19
Obr. 10: Metóda init()	20

ZOZNAM SKRATIEK

MCU	Microcontroller Unit
PCB	Printed Circuit Board
PIO	PlatformIO
TFT LCD	Thin-Film Transistor Liquid Crystal Display
UART	Universal Asynchronous Receiver-Transmitter
VSCode	Visual Studio Code

0 ÚVOD

V našej práci sa budeme venovať vytvoreniu hernej konzoly založenej na platforme Arduino Nano, ktorá bude schopná prevádzkovať hru Tetris. Cieľom je navrhnuť a zrealizovať funkčný systém, ktorý kombinuje softvér a hardvér na vytvorenie plne funkčnej hernej konzoly. Jedným z kľúčových prvkov projektu je vývoj vlastných knižníc, ktoré sú optimalizované na nízku spotrebu pamäte a výpočtových zdrojov. Tieto knižnice sú nevyhnutné pre efektívne ovládanie TFTLCD displeja a tlačidiel, a bez ich optimalizácie by nebolo možné dosiahnuť požadovanú funkcionality a výkon. Projekt je spravovaný a vyvíjaný pomocou PlatformIO v prostredí Visual Studio Code, čo umožňuje efektívny vývoj a testovanie všetkých komponentov.

Prvá kapitola sa zameriava na teoretickú časť, ktorá zahŕňa analýzu technológií, ako je mikrokontrolér Arduino Nano, TFTLCD displej a hardvérové prvky, ako tlačidlá a batéria, ktoré sú potrebné na zabezpečenie hernej funkčnosti konzoly. Zároveň bude podrobne popísaný význam vlastných knižníc, ktoré tvoria základ softvérového riešenia. Ich optimalizácia a ľahkosť umožňuje mikrokontroléru efektívne spracovávať grafické aj herné operácie, čím zaručuje plynulý chod hry Tetris na limitovanom hardvéri.

V druhej časti práce sú definované hlavné a vedľajšie ciele projektu.

Tretia kapitola obsahuje popis metodiky, kde je vysvetlené, ako sa vytvára a testuje softvérové a hardvérové riešenie. Budeme sa zaoberať zapojením komponentov, návrhom PCB a implementáciou vlastných knižníc pre ovládanie jednotlivých častí konzoly.

Našou prácou sa snažíme vytvoriť hernú konzolu, ktorá poskytuje klasický herný zážitok v kombinácii s modernými technológiami a vlastným vývojom softvéru.

1 PROBLEMATIKA A PREHĽAD LITERATÚRY

Táto práca sa zaoberá vývojom hernej konzoly, ktorá využíva mikrokontrolér Arduino Nano, TFTLCD displej a ďalšie hardvérové prvky, ako sú tlačidlá a batéria, na dosiahnutie hernej funkčnosti. V tejto kapitole sa budeme sústrediť na teoretickú analýzu technológií, ktoré sú základom tohto projektu.

1.1 Mikrokontrolér Arduino Nano

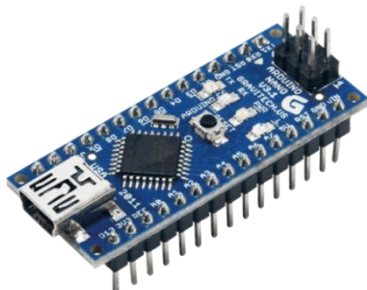
Arduino Nano je malá a kompaktná vývojová doska založená na mikrokontroléri ATmega328P, ktorá je vhodná pre embedded systémy, IoT projekty a prototypovanie. Svojou funkcionalitou je veľmi podobná Arduino UNO, ale má menšie rozmery a neobsahuje konektor pre napájanie DC jack.

Doska obsahuje 14 digitálnych I/O pinov, z ktorých 6 podporuje PWM, a 8 analógových vstupov s 10-bitovým ADC prevodníkom. Je možné ju programovať cez Mini-USB konektor, pričom využíva CH340 alebo FT232RL ako USB-to-serial prevodník. Napájanie je možné cez USB (5V) alebo externý zdroj 6-12V cez pin VIN, pričom prevádzkové napätie logických pinov je 5V.

Nano má 32 KB Flash pamäte, z čoho 0,5 KB je vyhradené pre bootloader, 2 KB SRAM a 1 KB EEPROM. Maximálna taktovacia frekvencia je 16 MHz. Doska podporuje komunikáciu cez SPI, I2C (TWI) a UART, čím umožňuje jednoduchú integráciu s rôznymi senzormi, displejmi a inými perifériami.

Vďaka svojim malým rozmerom (45×18 mm), nízkej spotrebe a širokým možnostiam rozšírenia je Arduino Nano obľúbené v projektoch ako automatizácia, robotika, nositeľná elektronika, senzory a riadiace systémy.

Na rozdiel od Arduino UNO nemá odnímateľný mikrokontrolér, čo znamená, že ak sa ATmega328P poškodí, nie je ho možné jednoducho vymeniť.[1]



Obr. 1: Mikrokontrolér Arduino Nano

Zdroj: <http://denontek.com.pk/Arduino-Nano-V3>

1.2 LcdTft Display

Displej s tenkovrstvovým tranzistorom a kvapalnými kryštálmi (TFT LCD) je typ LCD displeja, ktorý využíva technológiu tenkovrstvových tranzistorov na zlepšenie kvality obrazu, ako sú adresovateľnosť a kontrast. TFT je aktívna matricová LCD technológia, na rozdiel od pasívnych matricových LCD alebo jednoduchých priamo riadených LCD (t. j. so segmentmi priamo pripojenými k elektronike mimo LCD).

TFT displeje sa používajú v televízoroch, počítačových monitoroch, mobilných telefónoch, herných konzolách, osobných digitálnych asistentoch, navigačných systémoch, projektoroch a palubných doskách niektorých automobilov a stredne až vysoko výkonných motocyklov.

Vďaka svojej ostrosti a nízkej spotrebe energie sa často používa aj v prenosných zariadeniach, kde je dôležitá efektívnosť batérie. Moderné TFT displeje môžu mať rôzne typy podsvietenia, pričom LED podsvietenie je najčastejšie v súčasných modeloch. Technológia sa neustále vyvíja, čo umožňuje vyššie rozlíšenia a lepšiu reprodukciu farieb.

TFT displeje sú široko dostupné a ich výroba je cenovo efektívna, čo prispieva k ich rozšíreniu v spotrebnej elektronike.[2]



Obr. 2: LCDTFT Display

Zdroj: <https://techfun.sk/en/product/tft-touch-screen-2-8--240-x-320-a/>

1.3 PCB

Doska s plošnými spojmi (PCB) je elektronická zostava, ktorá používa medené vodiče na vytvorenie elektrických spojení medzi komponentmi. Okrem toho poskytuje mechanickú oporu pre elektronické súčiastky, aby bolo možné zariadenie upevniť v kryte.

Každá PCB je vytvorená z vrstiev vodivej medi, ktoré sa striedajú s elektricky izolačným materiálom. Vodivé prvky na PCB zahŕňajú medené cesty (traces), pájkovacie plošky (pads) a vodivé plochy (planes). Mechanická štruktúra je tvorená izolačným materiálom laminovaným medzi vrstvami vodičov. Celková konštrukcia je následne povrchovo pokovaná a pokrytá nevodivou spájkovacou maskou (solder mask).[3]



Obr. 3: PCB Doska

Zdroj: <https://www.rocket-pcb.com/rocket-pcb-high-tech-what-is-printed-circuit-board->

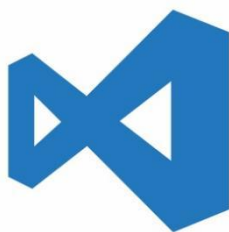
1.4 Visual Studio Code

Visual Studio Code (VS Code) je moderný a ľahký editor zdrojového kódu, ktorý ponúka vývojárom výkonné funkcie na všetkých hlavných platformách – Windows, macOS a Linux. Poskytuje vstavanú podporu pre viacero programovacích jazykov, ako sú JavaScript, Python, C++, Java a mnoho ďalších.

VS Code vyniká svojou prispôbitel'nosťou vďaka rozsiahlej knižnici rozšírení, ktoré umožňujú rozšíriť jeho funkcionality podľa potrieb používateľa. Obsahuje integrovaný terminál, nástroje na ladenie a správu verzií, ako aj podporu pre Git.

Medzi jeho kľúčové výhody patrí aj integrácia s GitHub Copilot, ktorý využíva AI na asistované písanie kódu. VS Code je vhodný pre profesionálnych vývojárov aj začiatočníkov, ktorí hľadajú výkonné a efektívne vývojové prostredie.

VS Code je vhodný pre profesionálnych vývojárov aj začiatočníkov, ktorí hľadajú výkonné a efektívne vývojové prostredie.[4]



Obr. 4: Logo Visual Studio Code

Zdroj: <https://seekvectors.com/post/visual-studio-code-logo>

1.5 PlatformIO

PlatformIO je profesionálne integrované vývojové prostredie (IDE) určené na vývoj softvéru pre embedded systémy a internet vecí (IoT). Poskytuje multiplatformový ekosystém, ktorý podporuje rôzne hardvérové architektúry, frameworky a vývojové dosky.

Jednou z hlavných výhod PlatformIO je jeho automatizovaný systém správy nástrojových reťazcov, závislostí a knižníc, vďaka čomu odstraňuje zložitú typickú

pre tradičné vývojové prostredia. Vývojári môžu pracovať v rôznych editoroch, ako je Visual Studio Code, Atom alebo iné príkazové rozhrania.

PlatformIO ponúka široké spektrum funkcií, vrátane pokročilého ladenia, jednotkového testovania a integrácie s CI/CD systémami.[5]



Obr. 5: Logo PlatformIO

Zdroj: https://commons.wikimedia.org/wiki/File:PlatformIO_logo.svg

2 CIELE PRÁCE

Hlavné ciele:

- Navrhnuť a implementovať funkčnú verziu hry Tetris na platforme Arduino Nano
- Vytvoriť vlastné softvérové knižnice s výnimkou knižnice Arduino.h
- Navrhnuť PCB pre daný projekt
- Navrhnuť 3D model pre obal konzoly

Vedľajšie ciele:

- Prispôsobiť vizuálny vzhľad hier a používateľského rozhrania
- Rozšíriť projekt o ďalšie hry
- Pridať spôsob nabíjania konzole

3 MATERIÁL A METODIKA

V tejto kapitole sa podrobne zameriame na proces realizácie projektu, od návrhu koncepcie až po finálne testovanie.

3.1 Navrhovanie hardvéru

V tejto časti sa budeme venovať výberu hardvérových komponentov a ich zapojeniu pre náš projekt.

3.1.1 Výber komponentov

Prvým krokom bolo starostlivé vybratie vhodných elektronických komponentov. Pre riadiacu jednotku sme zvolili mikrokontrolér Arduino Nano pre jeho dostupnosť, výkon a bohatú komunitu. Pre zobrazovanie informácií sme zvolili TFT LCD displej s rozlíšením 240x320 pixelov, ktorý využíva 8-bitové paralelné pripojenie, čo umožňuje rýchlejšie vykresľovanie viacerých pixelov súčasne, a ponúka dostatočný priestor pre zobrazenie herného poľa, menu a ďalších dôležitých prvkov.

Na ovládanie hry sme použili osem tlačidiel:

1. Pohyb doľava v hre a v menu.
2. Pohyb dole v hre a v menu.
3. Okamžitý pohyb dole (drop) v hre a pohyb hore v menu.
4. Pohyb doprava v hre a v menu.
5. Rotácia proti smeru hodinových ručičiek v hre.
6. Udržanie bloku v hre.
7. Otvorenie pauzovacieho menu v hre a potvrdenie v menu.
8. Rotácia v smere hodinových ručičiek v hre.

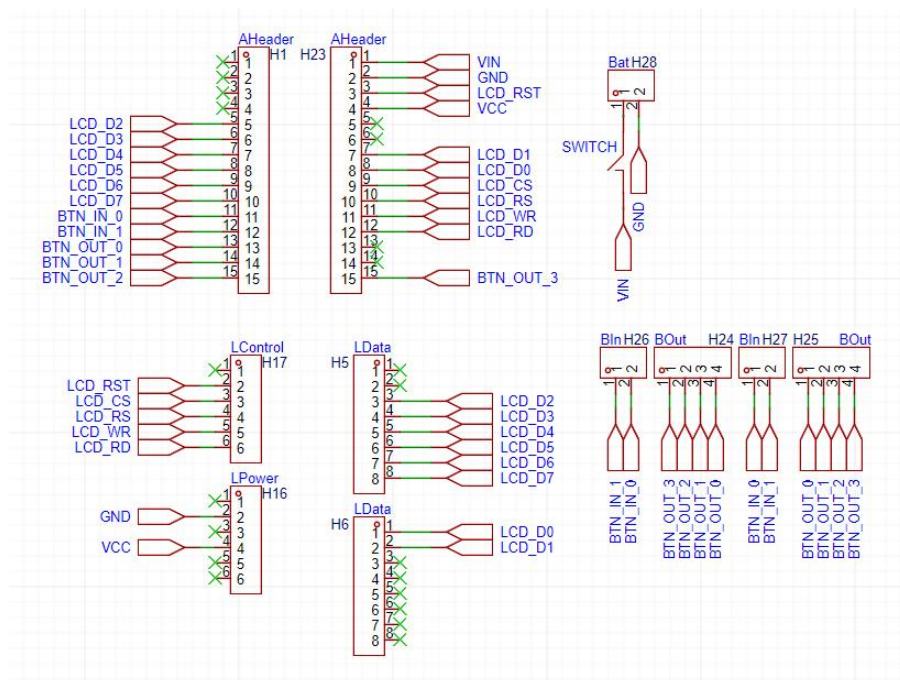
Ako zdroj napájania sme zvolili 7,4V Li-Po akumulátorový batériový balík. Tento výber sme urobili z dôvodu jeho vysokej kapacity, ktorá umožňuje dlhšiu prevádzku zariadenia na jedno nabitie. Ďalšou výhodou je jednoduchosť nabíjania, čo zvyšuje pohodlie používania.

Tab. 1: Zoznam komponentov

Názov	Popis	Počet Kusov
Arduino Nano	Riadiaca jednotka	1
TftLcd Display	Používateľské rozhranie	1
PCB	Plošný spoj	1
Battery Pack	Zdroj napájania	1
3D Model	Krit	1
Tlačidlá	Ovládanie	8
Prepínač	Ovládanie napájania	1

3.1.2 Schéma zapojenia

Na základe vybraných komponentov sme v programe EasyEDA vytvorili schému zapojenia. Schéma detailne zobrazuje pripojenie jednotlivých komponentov k mikrokontroléru Arduino Nano, vrátane pripojenia displeja, tlačidiel a napájacích



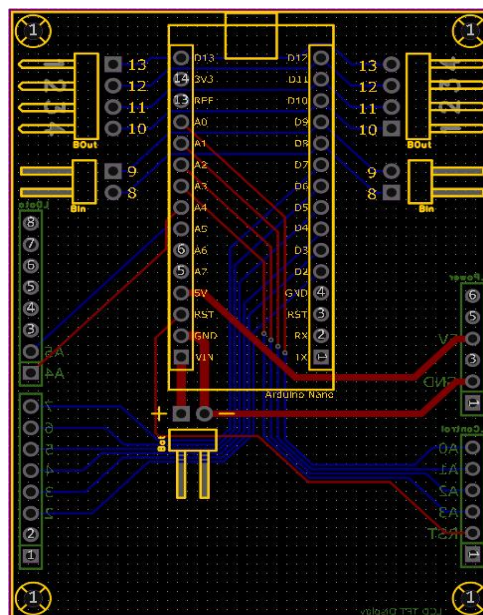
Obr. 6: Schéma zapojenia

zdrojov. Pri tvorbe schémy sme sa zamerali na prehľadnosť a čitateľnosť.

3.1.3 Návrh PCB

Na základe schémy zapojenia sme v programe EasyEDA navrhli vlastnú dvojvrstvovú PCB. Pri návrhu sme sa zamerali na optimálne umiestnenie komponentov, čo minimalizovalo dĺžku spojov a znížilo riziko rušenia. Spoje boli navrhnuté s dostatočnou šírkou a odstupom, aby sa zabezpečila stabilita signálu a zabránilo sa skratom. Využili sme viazacie otvory pre prepojenie jednotlivých vrstiev. Na zabezpečenie pevného uchytenia v 3D tlačnom puzdre sme do PCB integrovali štyri montážne otvory, po jednom v každom rohu.

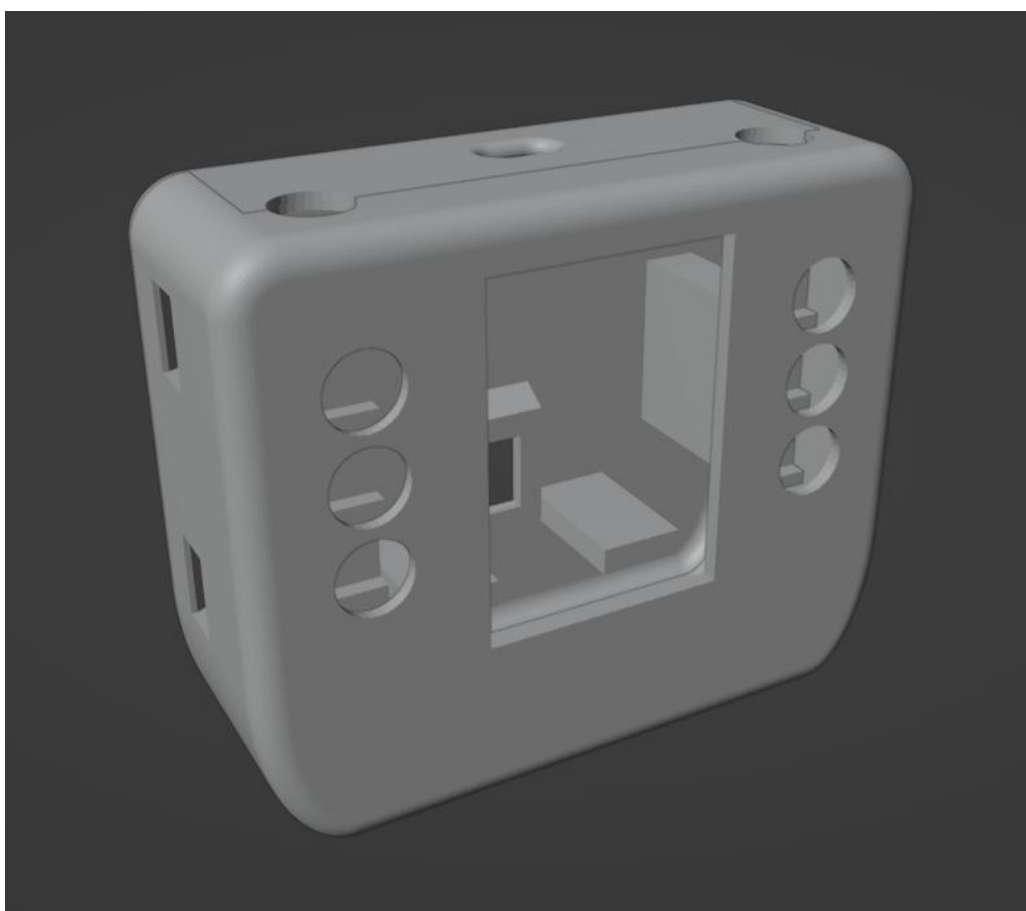
Po dôkladnej kontrole návrhu sme PCB poslali do výroby v spoločnosti JLCPCB. Po obdržaní vyrobených dosiek sme vykonali dôkladnú kontrolu kvality výroby pomocou voltmetra. Následne sme osadili všetky komponenty na PCB s použitím spájkovacej výbavy. Po osadení sme opätovne skontrolovali kvalitu spájkovania a vykonali funkčné testy jednotlivých komponentov na PCB.



Obr. 7: PCB návrh

3.1.4 3D modelovanie puzdra

Na navrhovanie puzdra sme použili program Blender. Vytvorili sme 3D model puzdra, ktorý zohľadňuje rozmery PCB, displeja, tlačidiel a ostatných komponentov. Pre maximálnu presnosť sme vymodelovali aj všetky jednotlivé komponenty a potom okolo nich vymodelovali puzdro. Model bol navrhnutý tak, aby zabezpečil pohodlné používanie konzoly a zároveň chránil elektroniku pred mechanickým poškodením. Po dokončení 3D modelu sme vytvorili 3D tlačiteľný model a následne puzdro vyrobili pomocou 3D



Obr. 8: 3D model

tlače.

3.2 Príprava hardvéru

Po obdržaní vyrobených PCB sme sa postarali o osadenie komponentov na ne. Tento proces bol dôležitý, pretože správne osadené komponenty sú kľúčové pre správne

fungovanie elektronického zariadenia. Pri osadzovaní sme dbali na správne umiestnenie každého komponentu a jeho správne pripojenie.

Po dokončení osadzovania sme vykonali dôkladnú vizuálnu kontrolu a aj kontrolu pomocou voltmetra, aby sme sa uistili, že všetky komponenty sú správne umiestnené a pripojené. Táto kontrola je nevyhnutná pre odhalenie možných chýb alebo nedostatkov v procese osadzovania.

Po úspešnej vizuálnej kontrole sme prešli k ďalšiemu kroku, ktorým bolo otestovanie PCB na jeho funkčnosť. Tento test nám umožnil overiť, či všetky komponenty pracujú správne a či zariadenie funguje podľa očakávania.

Po úspešnom otestovaní sme sa venovali zostaveniu herného zariadenia. To zahŕňalo vloženie osadenej PCB do 3D tlačenej puzdra a pripojenie všetkých potrebných káblov. Tento proces bol dôležitý pre dokončenie výroby herného zariadenia a jeho pripravenosť na použitie.

3.3 Príprava vývojového prostredia

Na vývoj softvérovej časti projektu sme zvolili integrované vývojové prostredie Visual Studio Code. Stiahli sme a nainštalovali Visual Studio Code z oficiálnej webovej stránky. Následne sme nainštalovali rozšírenie PlatformIO IDE z ponuky rozšírení.

Pri výbere integrovaného vývojového prostredia sme sa rozhodli pre Visual Studio Code, pretože poskytuje bohaté možnosti konfigurácie a podporuje širokú škálu programovacích jazykov a nástrojov. Po stiahnutí a inštalácii sme sa presunuli k rozšíreniu PlatformIO IDE, ktoré nám umožňuje efektívne pracovať s mikrokontrolérmi a vytvárať projekty pre rôzne platformy.

V rámci PlatformIO sme vytvorili nový projekt. Vybrali sme framework "Arduino" a dosku "Arduino Nano ATmega328".

Nastavili sme knižnice, kompilátor a ďalšie parametre projektu, čo zabezpečuje správne fungovanie nášho softvéru. Dôležité je mať všetky potrebné nástroje a zdroje

pripravené a nastavené správne od začiatku, aby sme mohli efektívne pracovať a minimalizovať možné problémy počas vývoja softvéru.

3.4 Vytvorenie zdrojového kódu

V tejto časti sa budeme zaoberať procesom návrhu a vývoja zdrojového kódu pre náš projekt. Pri tvorbe zdrojového kódu je dôležité mať predstavu o funkcionalite a štruktúre programu, ako aj o limitoch použitých zariadení. Preto sme navrhli väčšinu funkcionality ako vlastné knižnice pre lepšiu čitateľnosť kódu a ľahšie narábanie s vyvinutými funkcionalitami.

3.4.1 Vytváranie vlastných knižníc

Pre náš projekt sme vyvinuli knižnice na mieru s dvoma hlavnými cieľmi. Prvým cieľom bola minimalizácia pamäťovej náročnosti. Zamerali sme sa na vytvorenie kompaktných a optimalizovaných knižníc, aby sme efektívne využili obmedzené zdroje Arduino Nano. Druhým cieľom bolo maximalizovať výkon a rýchlosť, najmä pri práci s displejom.

Najťažšie bolo jednoznačne vyvinúť knižnicu ST7789V pre displej. Knižnica sa môže zdať krátka a jednoduchá, no nájsť správne príkazy, ktoré sa posielajú displeju cez 8-bitové paralelné rozhranie spolu s pinmi RD, WR, RS a CS, bolo dosť náročné, pretože si to vyžadovalo preštudovanie oficiálnej dokumentácie pre náš displej. Hlavnými časťami tejto knižnice sú metódy LCD_write() a init().

```
void ST7789V::LCD_write(uint8_t d) {  
    digitalWrite(LCD_WR, LOW);  
  
    PORTD = (PORTD & 0b00000011) | ((d) & 0b11111100);  
    PORTC = (PORTC & 0b11001111) | ((d << 4) & 0b00110000);  
  
    digitalWrite(LCD_WR, HIGH);  
}
```

Obr. 9: Metóda LCD_write()

Metóda `LCD_write()` je najdôležitejšou metódou zo všetkých, pretože sa používa na absolútne všetko, od zapisovania dát až po zapisovanie a vykonávanie príkazov. Metóda `LCD_write()` sa začína nastaviť pinu `LCD_WR` na `LOW`, čím dáme displeju signál, že má očakávať dáta alebo príkaz, ktorý mu chystáme poslať. Pomocou registrov nastavíme 8-bitové paralelné rozhranie na dáta alebo príkaz, ktorý chceme zapísať do displeja. Je dôležité si uvedomiť, že hodnota môže mať len 8 bitov a pre väčšie hodnoty, ako napríklad farba, ktorá má 16 bitov, musíme farbu poslať v dvoch príkazoch zápisu, najprv prvú polovicu farby a potom druhú polovicu. Následne nastavíme pin `LCD_WR` na `HIGH`, čím dáme displeju signál, aby zapísal dáta alebo príkaz a vykonal ho. To, či posielame dáta

```
void ST7789V::Init() {
    DDRD = 0b11111100;
    DDRC = DDRC | 0b00111111;
    PORTC = PORTC | 0b00111111;

    digitalWrite(LCD_CS, HIGH);
    digitalWrite(LCD_WR, HIGH);
    digitalWrite(LCD_RD, HIGH);
    digitalWrite(LCD_CS, LOW);

    LCD_command_write(0xC5);
    LCD_data_write(0x54);
    LCD_data_write(0x00);

    LCD_command_write(0x36);
    LCD_data_write(0b00000000);

    LCD_command_write(0x3A);
    LCD_data_write(0x55);

    LCD_command_write(0x11);
    delay(10);

    LCD_command_write(0x29);
    LCD_command_write(0x2c);
}
```

Obr. 10: Metóda `init()`

alebo príkaz, môžeme nastaviť pomocou pinu `LCD_RS`. Keď nastavíme tento pin na `HIGH`, povieme displeju, že posielame dáta, a keď ho nastavíme na `LOW`, povieme mu, že posielame príkaz..

V metóde `init()` nastavujeme piny pre 8-bitové paralelné rozhranie aj pre riadiace piny buď ako vstupné, alebo ako výstupné. Následne použijeme sekvenciu príkazov, ktoré nastavujú potrebné nastavenia displeja ako jeho inicializáciu, nastavenie správneho farebného priestoru a nastavenie rotácie.

Medzi ďalšie vytvorené metódy patria kreslenie obdĺžnikov, rámov a kreslenie textu, ktorá využívajú náš vlastný bitmapový font.

3.4.2 Integrácia knižníc a hernej logiky

S využitím našich vlastných knižníc sme implementovali menu pre úvodnú obrazovku, záznamy (skóre hráčov), hru Tetris a menu pozastavenia. Všetky tieto časti sú prepojené, čo umožňuje našej konzole pamätať si, kde sa hráč nachádzal v menu pri prechode do iného menu, alebo dokonca kde bola hra pozastavená s možnosťou pokračovať a hrať ďalej. V menu záznamov sme hráčovi dali možnosť prezerať si svoje predchádzajúce skóre alebo ich dokonca aj vymazať.

4 VÝSLEDKY PRÁCE A DISKUSIA

V rámci projektu sme úspešne vyvinuli funkčnú hernú konzolu založenú na platforme Arduino Nano, ktorá umožňuje hrať hru Tetris. Realizácia projektu však nebola bez výziev. Obmedzený výkon a pamäťový priestor mikrokontroléra Arduino Nano predstavovali významné výzvy pri implementácii hry.

Aby sme prekonalí tieto obmedzenia, pristúpili sme k niekoľkým dôležitým krokom. V prvom rade sme sa zamerali na optimalizáciu hardvérovej konfigurácie. Pri návrhu hardvéru sme dôkladne zvážili umiestnenie tlačidiel a displeja na 3D tlačennom puzdre. Cieľom bolo zabezpečiť pohodlné ovládanie a zároveň minimalizovať rušenie signálu medzi jednotlivými komponentmi.

Ďalším kľúčovým krokom bolo vytvorenie vlastných optimalizovaných knižníc pre ovládanie hardvéru. Knižnica pre ovládanie TFT LCD displeja bola navrhnutá s dôrazom na minimalizáciu spotreby pamäte a optimalizáciu vykresľovacích algoritmov. Podobne aj knižnica pre ovládanie tlačidiel bola navrhnutá s ohľadom na efektívnu detekciu stlačení a minimalizáciu rušenia.

Implementácia hernej logiky tiež vyžadovala dôkladnú optimalizáciu. Pri tvorbe algoritmov pre generovanie tetromín, kontrolu pohybu, rotáciu a detekciu kolízií sme sa snažili minimalizovať počet výpočtov a znížiť nároky na procesor.

Výsledkom našej práce je funkčná herná konzola s možnosťou hrania hry Tetris, ktorá demonštruje efektívne využitie dostupných hardvérových zdrojov. Aj keď súčasná verzia podporuje iba jednu hru, použitý prístup s dôrazom na optimalizáciu a využitie vlastných knižníc by umožnil relatívne jednoduchú implementáciu ďalších jednoduchých hier v budúcnosti.

5 ZÁVERY PRÁCE

V rámci projektu sme úspešne vyvinuli funkčnú hernú konzolu založenú na platforme Arduino Nano. Táto konzola umožňuje hrať hru Tetris a predstavuje dôkaz toho, že je možné vytvoriť jednoduchý, no funkčný herný systém na veľmi obmedzenom hardvéri.

Splnili sme všetky hlavné stanovené ciele projektu. Podarilo sa nám vytvoriť vlastné knižnice pre ovládanie hardvéru, implementovať logiku hry Tetris a zabezpečiť plynulý chod hry na Arduino Nano.

Z hľadiska využitia pamäte sme zaznamenali nasledujúce výsledky:

- **RAM** - Využitie statickej RAM predstavuje približne 54,2% (1 110 bajtov) z celkovej kapacity 2 048 bajtov. Využitie dynamickej RAM sme odhadli na základe postupného pridávania nových funkcií a sledovania stability programu. Pri pokuse o implementáciu ďalších funkcií, ako napríklad zvýšenie zložitosti hry alebo pridanie nových grafických prvkov, dochádzalo k nepredvídateľnému správaniu programu, ako napríklad zamrznutiu alebo nezmyselné vykresľovanie. Po odstránení týchto ďalších funkcií sa program vrátil k stabilnému chodu, čo potvrdzuje, že dostupná RAM je plne využitá.

- **FLASH** - Využitie FLASH pamäte je približne 56,3% (17296 bajtov). Tento pomer využitia FLASH pamäte bol dosiahnutý vďaka optimalizácii kódu, ako je napríklad minimalizácia dĺžky premenných, používanie efektívnych algoritmov a vyhýbanie sa zbytočnej redundancii v kóde.

V budúcnosti by sme mohli rozšíriť funkcionality konzoly o ďalšie hry, napríklad Minesweeper, Snake alebo Pong. Ďalším možným vylepšením by bolo pridanie zvukového výstupu alebo použitie výkonnejšieho mikrokontroléra pre podporu komplexnejšej grafiky a hier.

6 ZHRNUTIE

Cieľom našej práce bolo vytvoriť funkčnú hernú konzolu založenú na platforme Arduino Nano. Podarilo sa nám úspešne navrhnuť a implementovať systém, ktorý umožňuje hrať hru Tetris na tomto obmedzenom hardvéri. Využitím vlastných optimalizovaných knižníc a dôkladným návrhom hernej logiky sme dosiahli plynulý a zábavný herný zážitok. Tento projekt demonštruje možnosti vývoja jednoduchých, no funkčných elektronických zariadení s využitím moderných nástrojov a technológií.

ZOZNAM POUŽITEJ LITERATÚRY

1. Arduino Nano. 2025. [online]. 2025, [cit. 2025-2-12]. Dostupné na internete: <<https://docs.arduino.cc/resources/datasheets/A000005-datasheet.pdf?> >
2. TFT LCD. [online]. 2025, [cit. 2025-2-12]. Dostupné na internete: <https://en.wikipedia.org/wiki/TFT_LCD>
3. What is a PCB and Intro to PCB Design. [online]. 2025, [cit. 2025-2-12]. Dostupné na internete: <<https://resources.altium.com/p/what-is-a-pcb>>
4. Our code editor. [online]. 2025, [cit. 2025-2-12]. Dostupné na internete: <<https://code.visualstudio.com>>
5. What is PlatformIO? [online]. 2025, [cit. 2025-2-12]. Dostupné na internete: <<https://docs.platformio.org/en/latest/what-is-platformio.html>>

PRÍLOHA

USB príloha

Priložené USB obsahuje:

- Dokumentáciu v elektronickej podobe
- Fotografie a video záznam finálneho stavu projektu
- Schému zapojenia
- Zdrojový kód pre Arduino Nano
- Dokumentácie použité pri vytváraní zdrojového kódu
- Zdrojové súbory 3D modelu (blend, stl)