

# Mikroprocesorové a vestavěné systémy – Projekt

Dokumentace  
**M – Hra na displeji**

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Konfigurace zařízení a prostředí</b>	<b>2</b>
2.1	Technologie . . . . .	2
2.1.1	ESP32 . . . . .	2
2.1.2	I2C OLED displej . . . . .	2
2.1.3	Proporcionální joystick . . . . .	2
2.2	IDF . . . . .	2
2.2.1	Platformio . . . . .	3
2.3	Zapojení a nastavení . . . . .	3
<b>3</b>	<b>Implementace</b>	<b>4</b>
3.1	Knihovna ssd1306.h . . . . .	4
3.1.1	Vlastní úprava knihovny . . . . .	4
3.2	Herní logika . . . . .	4
3.2.1	Pohyblivé objekty . . . . .	4
3.2.2	Hráč . . . . .	4
3.2.3	Míč . . . . .	4
3.2.4	Kolize . . . . .	5
3.2.5	Vstup joysticku . . . . .	5
<b>4</b>	<b>Vlastnosti projektu</b>	<b>6</b>
4.1	Obrazovky . . . . .	6
4.2	Průběh hry . . . . .	6
4.3	Ovládání . . . . .	6

# 1 Úvod

Cílem tohoto projektu bylo realizovat jednoduchou hru pro platformu *ESP32* za použití proporcionálního *joysticku* a *I2C displeje* jako vstupního a výstupního zařízení. Hra, která byla implementovaná, je celosvětově známá pod názvem *Ping pong*. Důvodem zvolení této hry, je právě její nenáročnost na hardware, kdy je třeba pracovat pouze se třemi v čase měnícími polohu objekt a to pouze v rámci dvojrozměrného pole.

## 2 Konfigurace zařízení a prostředí

V této sekci budou přiblíženy technologie využity při vypracování projektu a popsáno připojení periférií k zařízení *ESP32*.

### 2.1 Technologie

Počet využitých technologií při vypracování této práce byl opravdu velký, proto budou níže zmíněny se základním popisem pouze ty nejvýznamější.

#### 2.1.1 ESP32

*ESP32* je série energeticky nenáročných technologií soustředěná na jeden čip. Disponuje 4 MB Flash a 320 KB RAM pamětí a 240 Mhz procesorem Xtensa® dual-core 32-bit LX6 microprocessor a další spoustou funkcí (např.: WiFi, Bluetooth, I2C, UART, 34 programovatelných obecných registrů, atd.)(více na [2]).

#### 2.1.2 I2C OLED displej

Display využívá pro komunikaci s deskou rozhraní *I2C*. *I2C* je synchronní seriové rozhraní, které využívá využívá pro komunikaci model *Master/Slave* a dva vodiče (viz [1]):

- vodič *SDA* (Serial clock) – datový vodič,
- vodič *SLC* (Data clock) – vodič hodinového signálu.

Displej využívá technologii OLED, která se vyznačuje faktem, že diody reprezentující černou barvu opravdu nevyzařují žádné světlo a v podání černobílého displeje tak vytváří kontrast, který produkuje ostrý obraz i pro malá rozlišení, které je tomto případě *128x64* pixelů.

#### 2.1.3 Proporcionální joystick

Joystick má v sobě zabudované 2 na sobě nezávislé *potenciometry* pro každou osu (*X* a *Y*), pomocí kterých jsme schopni za pomoci *Analogově-digitálního převodníku* určit přesně v jaké poloze se páčka nachází. Joystick slouží zároveň také jako tlačítko. Pro komunikaci využívá vodiče:

- vodič *VRx* (Voltage Proportional x) – napětí úměrné ose *X*,
- vodič *VRy* (Voltage Proportional y) – napětí úměrné ose *Y*,
- vodič *SW* (Switch) – stlačovací tlačítko.

### 2.2 IDF

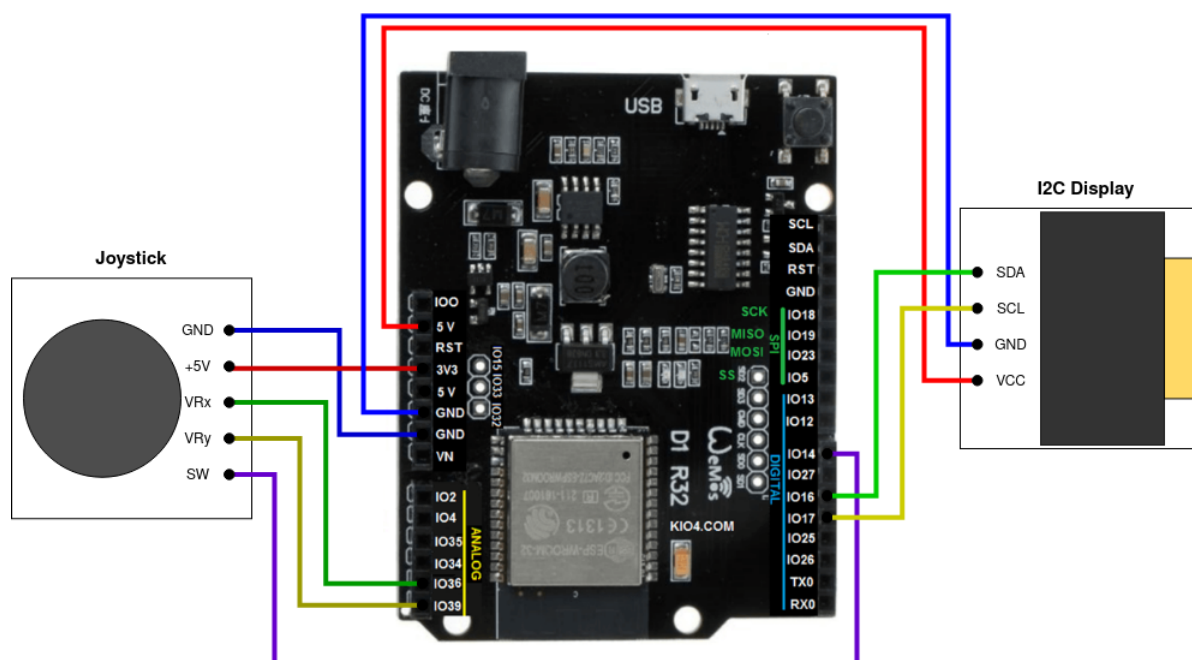
*IDF* neboli *Espressif IoT Development Framework* je prostředí pro vývoj mikrokontrolérů *ESP32*. Jakožto framework poskytuje celou řadu nástrojů, knihoven a dokumentaci pro vývoj. *IDF* nativně podporuje jazyk *C* ve kterém byl taky napsán tento projekt.

### 2.2.1 Platformio

*Platformio* je vývojové prostředí, které je implementováno jako rozšíření pro aplikaci *Visual Studio Code*. Jako pro jeden z mnoha poskytuje jednotnou vývojovou platformu pro framework IDF. Umožňuje jednoduše spravovat projekty, připojená zařízení, ale také pomocí konzolového uživatelského rozhraní nastavovat obecné registry pro periferie a tak dále.

### 2.3 Zapojení a nastavení

Zapojení bylo provedeno podle účelů jednotlivých vodičů. Na analogové registry *GPIO36* a *GPIO39* byly připojeny jako vstup vodiče joysticku *VRx* a *VRy*. Jejich vstupní hodnoty jsou zachycovány s šířkou 12 bitů (převáděno na hodnoty 0 až 4095). Pro tlačítko na joysticku *SW* byl použit digitální registr *GPIO14* a byl nastaven jako vstupní. Pro datový vodič *SDA* a vodič hodinového signálu *SLC* byly využity registry *GPIO16* a *GPIO17* a nastaveny v *Menuconfig* dle zadání. Joystick byl připojen na 3.3 Voltový zdroj. Tento zdroj byl zvolen kvůli tomu, že AD převodník je pouze 12 bitový a nedokáže tedy reprezentovat hodnotu 5000 mV, kterou by joystick při 5 V zdroji produkoval. Špatně by se tedy pracovalo s hodnotami ořízlými na 4095 mV, při kterých by joystick přišel o téměř polovinu rozsahu v polohách od střední výchozí polohy po krajní polohu páčky.



Obrázek 1: Schéma zapojení periferií k desce ESP32

## 3 Implementace

Pro implementaci bylo využito dříve zmiňované prostředí *IDF*. Pro práci s displejem byla poté využita knihovna *ssd1306* a hlavičkový soubor *font8x8\_basic*. Pro čtení analogových hodnot joysticku byly využity funkce z knihovny *driver/adc.h*.

### 3.1 Knihovna *ssd1306.h*

Knihovna *ssd1306* poskytuje funkce pro jednoduché vykreslování bitmapových obrázků, textu nebo třeba linek na I2C displej. Nejvíce využívanou funkcí byla funkce *ssd1306\_bitmaps*, která umožňuje vykreslit předem definovaný bitmapový obrázek na konkrétní bod displeje definovaný souřadnicemi *X* a *Y*. Dalšími hojně využívanými funkcemi byly funkce *ssd1306\_clear\_screen* pro nastavení všech pixelů obrazu na černou barvu nebo funkce *ssd1306\_display\_text* pro vykreslení textu.

Všechny znaky ASCII tabulky jsou v podobě osmi Bajtových bitmapových obrázků definovány v hlavičkovém souboru *font8x8\_basic*.

#### 3.1.1 Vlastní úprava knihovny

Fyzické překreslení displeje, tedy zobrazení vyrovnávací paměti na displeji, je časově velice náročné. Proto pro větší plynulost programu (plynulost pohybu objektů po hrací ploše), bylo potřeba z definice funkce *ssd1306\_bitmaps* odstranit zpoždění a fyzické překreslení displeje při každém využití funkce. Díky tomu bylo možné, v rámci jedné iterace herního stavu, překreslovat objekty na displeji pouze ve vyrovnávací paměti. K pokynu k fyzickému překreslení displeje dojde vždy jednou po aktualizaci celé herní plochy.

### 3.2 Herní logika

Aby hra působila jako plynulá animace reagující na uživatelský vstup z joysticku, je třeba neustále v cyklu aktualizovat herní stav, myšleno aktuální pozice všech pohyblivých objektů, na základě konečného počtu událostí, ke kterým ve hře může dojít. Těmito událostmi jsou kolize míče s hráčem či horní nebo dolní hranicí, vstup od uživatele a konec hry.

#### 3.2.1 Pohyblivé objekty

Pohyb herních objektů je realizován překreslováním objektů na jiné místo v rámci herního pole v čase, tedy v jednotlivých herních stavech. Při překreslení prvku je potřeba prvek v původní pozici nejprve smazat a až poté je možné vykreslit objekt na pozici nové. Kdyby k mazání před přesunem nedošlo, objekt by za sebou zanechával čáru a to není žádoucí. Vzhledem k malému rozlišení displeje dochází k posunům v rámci jednotek pixelů. Větší skoky by mohly nabourat plynulost hry.

#### 3.2.2 Hráč

Hráč je herní pohyblivý objekt reprezentován bitmapovým obrázkem oddělníku a souřadnicemi *X* a *Y*, které definují, kde se nachází levý horní roh objektu. Hráč reaguje na polohu páčky joysticku v ose *Y* pohybem nahoru a dolů. Vzhledem k tomu, že je joystick proporcionální, určuje poloha páčky, o kolik pixelů se hráč v ose posune v daném herním stavu. Velikost tohoto skoku reprezentuje rychlost jakou se hráč pohybuje. Tímto způsobem může hráč dosáhnout třech různých rychlostí v obou směrech.

#### 3.2.3 Míč

Míč má na rozdíl od hráče, který je reprezentován bitmapovým obrázkem oddělníku a souřadnicemi polohy, definovaný navíc směr, kterým se neustále pohybuje, jak v ose *X*, tak v ose *Y*. Směr je polohový vektor a určuje tedy o kolik pixelů se posune v každém herním stavu.

Míč změní směr pokaždé při kolizi s:

- hráčem – směru *X* je přiřazeno opačné číslo aktuální hodnoty směru *X*,

- spodní nebo horní hranicí – směru Y je přiřazeno opačné číslo aktuální hodnoty směru Y.

Kolize poté působí jako odraz míčku od stěny, kdy na něj nepůsobí žádná gravitace. Úhel dopadu se tedy rovná úhlu odrazu.

### **3.2.4 Kolize**

K rozpoznání kolize míče s jiným objektem dochází pomocí porovnávání poloh jednotlivých objektů. Jedná-li se o dvojrozměrné objekty, je třeba při kontrolách kolizí s pravou a spodní stranou objektu přičítat k aktuální poloze taky velikost objektu. Pro rozpoznání kolize míče s hráčem je třeba porovnat polohu míče vůči hráči v obou osách.

### **3.2.5 Vstup joysticku**

Hodnota napětí, kterou na vstup joysticku produkuje ve výchozí poloze, se pohybuje kolem 1790 mV (hodnota byla vypořovávána). Tuto hodnotu považujeme za výchozí. Dojde-li k významnému poklesu této hodnoty, víme, že páčka byla posunuta vzhůru a že se hráč bude posunovat v ose Y vzhůru, dokud se zase neustálí kolem výchozí hodnoty. To samé, ale opačně, platí pro pohyb směrem dolů. Dále jsou definovány jisté prahy, které jsou porovnávány s rozdílem mezi aktuální a výchozí hodnotou. Jejich postupné překračování určuje rychlost pohybu.

## 4 Vlastnosti projektu

Celý program je ovládán jen za pomoci připojeného joysticku, a to pohybem nebo stisknutím páčky. Po skončení kola, kdy jedním kolem je myšlena doba od začátku samotné hry po prohru jednoho z hráčů, je možné spustit hru znovu od začátku.

### 4.1 Obrazovky

Obsah, který je na displej vykreslován, je logicky členěn na tzv. *Obrazovky*. V rámci programu je možné se setkat s následujícími třemi druhy obrazovek:

#### 1. Úvodní obrazovka –

- zobrazí se ihned po připojení mikrontroléru ke zdroji a její zobrazení trvá 5 vteřin
- po uplynutí 5 vteřin se přepne na obrazovku číslo 2
- zobrazuje informace o čísle projektu, autorovi a roku vytvoření programu

#### 2. Obrazovka s aktivním čekáním –

- zobrazuje textovou výzvu, aby bylo stisknuto tlačítko pro zahájení hry
- program je ve stavu, kdy aktivně čeká na stisk tlačítka
- po stisknutí tlačítka se přepne na obrazovku číslo 3

#### 3. Obrazovka hry –

- vykresluje samotný průběh hry
- po skončení hry se přepne zpět na obrazovku číslo 2

### 4.2 Průběh hry

Po zahájení hry tlačítkem dochází k odpočtu zahájení hry, který je zobrazován ve středu obrazovky. Po odpočtu hry se dá míč do pohybu a cílem hráčů je odrážet míč svojí postavou na soupeřovu stranu. Pokud kterýkoliv z hráčů míč v čas neodrazí (míč se dostane za hráče), tak to znamená prohru pro daného hráče.

### 4.3 Ovládání

Hráč je ovládán pohybem páčky joysticku v ose Y. Nahnutím páčky lze dosáhnout až tří různých rychlostí pohybu hráče. Pokud je páčka ve výchozím stavu, hráč zůstává stát na místě. Vzhledem k tomu, že hra je určena pro 2 hráče a k dispozici je joystick pouze 1, tak dohází během hry k přepínání, který hráč je aktuálně ovládán. Na tahu je vždy ten hráč, proti kterému se v dané době pohybuje míč.

## Literatura

- [1] BIDLO, M.: Principy sériové komunikace, sériová komunikační rozhraní. [online], rev. 2023, [vid. 2023-12-10]. Dostupné z: [https://moodle.vut.cz/pluginfile.php/707394/mod\\_folder/content/0/2022IMP04.pdf?forcedownload=1](https://moodle.vut.cz/pluginfile.php/707394/mod_folder/content/0/2022IMP04.pdf?forcedownload=1)
- [2] contributors, W.: ESP32 — Wikipedia, The Free Encyclopedia. rev. prosinec 2023, [vid. 2023-12-10]. Dostupné z: <https://en.wikipedia.org/w/index.php?title=ESP32&oldid=1188583650>