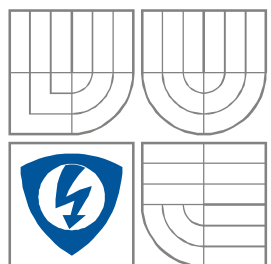


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**

ÚSTAV RADIOELEKTRONIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF RADIO ELECTRONICS

HERNÍ KONZOLE S 8BITOVÝM PROCESOREM

GAMEPAD WITH 8-BIT PROCESSOR

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

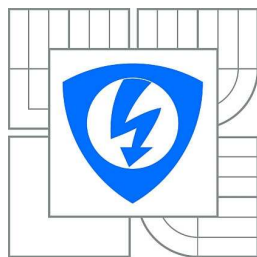
PETR VITKO

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. TOMÁŠ FRÝZA, Ph.D.

BRNO 2011



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav radioelektroniky

Bakalářská práce

bakalářský studijní obor
Elektronika a sdělovací technika

Student: Petr Vitko

ID: 112062

Ročník: 3

Akademický rok: 2010/2011

NÁZEV TÉMATU:

Herní konzole s 8bitovým procesorem

POKYNY PRO VYPRACOVÁNÍ:

Prostudujte možnosti vytvoření jednoduché herní konzole řízené 8bitovým procesorem. Uvažujte vhodné ovládací prvky a navrhnete grafický výstup celého zařízení. Vytvořte a realizujte obvodové zapojení a návrh desek plošného spoje. Provedte oživení hardwarové části.

Naprogramujte 2 až 3 aplikace pro herní konzoli v jazyce symbolických adres a v jazyce C. Provedte testování celého zařízení z hlediska funkce a početní náročnosti. Provedte detailní analýzu spotřeby celého zařízení a zajistěte možnost napájení z baterií.

DOPORUČENÁ LITERATURA:

[1] Atmel Corporation. AVR 8-Bit RISC. [online]. 2008 – [cit. 2. února 2009]. Dostupné na WWW: <http://www.atmel.com/products/avr/>

[2] BARNETT, R.H., COX, S. O'CULL, L. Embedded C Programming and the Atmel AVR. New York: Thomson Delmar Learning, 2007. 532 stran. ISBN 1-4180-3959-4

[3] Freescale Semiconductor. 8-bit Microcontrollers. [online]. 2008 – [cit. 16. května 2008]. Dostupné na WWW: <http://www.freescale.com/webapp/sps/site/homepage.jsp?nodeId=0162468449>

Termín zadání: 7.2.2011

Termín odevzdání: 27.5.2011

Vedoucí práce: doc. Ing. Tomáš Frýza, Ph.D.

prof. Dr. Ing. Zbyněk Raida
Předseda oborové rady

ABSTRAKT

Tato bakalářská práce je zaměřena na vývoj aplikací pro herní konzolu. Aplikace jsou programovány v jazyce symbolických adres a v jazyce C. Konzolu tvoří 8-bitový mikroprocesor AVR ATmega32, monochromatický LCD displej s rozlišením 128x64 bodů a pro snadné ovládání je připojen joystick.

KLÍČOVÁ SLOVA

MCU, mikroprocesor, grafický displej, joystick, video paměť, herní konzola

ABSTRACT

This bachelor thesis is focused on development of applications for gamepad. Applications are programmed in assembler language and in C. Gamepad are created by 8-bit microprocessor AVR ATmega32, monochromatic LCD display with screen resolution 128x64 dots and the joystick is connected in order to manage the easy control.

KEYWORDS

MCU, microprocessor, graphic display, joystick, video memory, gamepad

VITKO, P. *Herní konzole s 8bitovým procesorem*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2011. 35 s., 4 s. příloh. Semestrální práce. Vedoucí práce: doc. Ing. Tomáš Frýza, Ph.D.

PROHLÁŠENÍ

Jako autor bakalářské práce na téma Herní konzole s 8bitovým procesorem dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Děkuji vedoucímu bakalářské práce doc. Ing. Tomáši Frýzovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne

.....

(podpis autora)

OBSAH

Seznam obrázků	vii
Seznam tabulek	viii
Úvod	1
1 Konzola	2
1.1 Displej	3
1.2 Joystick	4
1.3 Reproduktor	6
2 Hledání min	7
2.1 Cíl hry	7
2.2 Podstata hry.....	7
2.3 Start hry.....	9
2.3.1 Generování pole min.....	9
2.3.2 Samotná hra	9
2.4 Zobrazování	10
3 Tetris	13
3.1 Cíl hry	13
3.2 Podstata hry.....	13
3.3 Vlastní hra.....	14
3.4 Ovládání.....	15
3.5 Akce jednotlivých tlačítek	18
3.6 Generování náhodných kostek.....	20
3.7 Vykreslování	20
3.8 Zvuky	22
4 Spaceman	23
4.1 Cíl hry	23
4.2 Podstata hry.....	23
4.3 Hlavní program	24
4.4 Objekty a jejich vykreslování	25
4.4.1 Pozadí.....	26

4.4.2	Nepřítel	26
4.4.3	Stíhačka.....	27
4.4.4	Kulky	28
5	Závěr	29
	Literatura	30
	Seznam příloh	31

SEZNAM OBRÁZKŮ

Obr. 1.1 - Blokové schéma konzoly	2
Obr. 1.2 - Konzola	2
Obr. 1.3 - Popis zápisu na displej	3
Obr. 1.4 - Uspořádání bytů ve video paměti.....	3
Obr. 1.5 - Rozvržení tlačítek na joysticku	5
Obr. 1.6 - Popis signálů na sběrnici joysticku	5
Obr. 1.7 - Záznam z osciloskopu při nestisknutém žádném tlačítku	6
Obr. 1.8 - Záznam z osciloskopu při stisknutém tlačítku T_UP1	6
Obr. 2.1 - Vývojový diagram hry Hledání min.....	8
Obr. 2.2 - Vývojový diagram odhalování okolních min.....	10
Obr. 2.3 - Ukázka jednotlivých dlaždic	11
Obr. 2.4 - Ukázka vykreslení levé poloviny displeje.....	11
Obr. 2.5 - Ukázka vykreslení pravé poloviny displeje bez textu.....	11
Obr. 2.6 - Ukázka hry Hledání min	12
Obr. 2.7 - Ukázka konce hry Hledání min	12
Obr. 3.1 - Přehled kostek a jejich hodnot	14
Obr. 3.2 - Ukázka padající kostky a její změny	14
Obr. 3.3 - Vývojový diagram algoritmu testující tlačítka.....	17
Obr. 3.4 - Ukázka rotace padající kostky.....	19
Obr. 3.5 - Přehled všech možných rotací.....	19
Obr. 3.6 - Ukázka gravitace hry Tetris	20
Obr. 3.7 - Zobrazené buňky	21
Obr. 3.8 - Ukázka hry Tetris	21
Obr. 3.9 - Ukázka hry Tetris	21
Obr. 4.1 - Vývojový diagram hry Spaceman	25
Obr. 4.2 - Princip zobrazování bloků ve hře Spaceman	26
Obr. 4.3 - Typy nepřátel ve hře Spaceman	26
Obr. 4.4 - Předlohy letadel.....	27
Obr. 4.5 - Předlohy letadel pro zápis "bílých" bodů.....	27
Obr. 4.6 - Ukázka hry Spaceman - úroveň 1	28
Obr. 4.7 - Ukázka hry Spaceman - úroveň 2	28

SEZNAM TABULEK

Tab. 2.1 - Význam hodnoty v buňce.....	7
Tab. 2.2 - Přehled důležitých registrů a míst v paměti	8
Tab. 2.3 - Popis akce vyvolané po stisku tlačítka	9
Tab. 3.1 - Velikost přírůstku v závislosti na rychlosti a počtu řádků	13
Tab. 3.2 - Význam hodnot buňky matice.....	13
Tab. 3.3 - Přehled nejdůležitějších proměnných a registrů.....	15
Tab. 3.4 - Funkce jednotlivých tlačítek	16
Tab. 3.5 - Popis jednotlivých bitů proměnné GAME	19
Tab. 4.1 - Přehled počítadel	23
Tab. 4.2 - Popis nejdůležitějších globálních proměnných	24

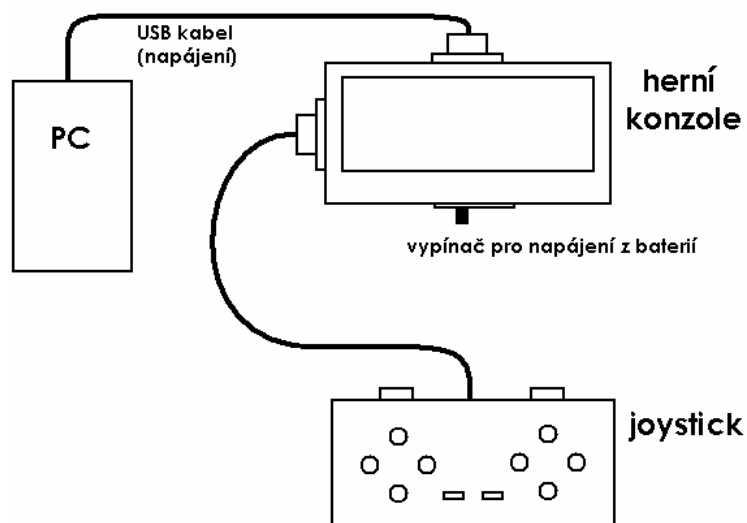
ÚVOD

Tato práce se zabývá vytvořením několika herních aplikací pro 8-bitovou konzolu tvořenou mikrokontrolérem AVR ATmega32. Podobnou tematikou (programováním her pro procesory AVR) se zabývá i projekt Uzebox [6], [7]. Jedná se také o herní konzolu, emulátor na konzoly NES/SNES (Nintendo Entertainment System / Super Nintendo Entertainment System), s 8-bitovým procesorem AVR ATmega644. Uzebox k ovládání her využívá dvou joysticků NES/SNES a k zobrazování využívá televizi (S-video, SCART) nebo monitor. Z důvodu absence funkčních joysticků NES/SNES jsem se rozhodl pro vlastní zapojení joysticku. Jelikož je herní konzola konstruována jako přenosné zařízení malých rozměrů a vykreslování na televizi ztrácí význam, je pro zobrazování určen displej.

V mládí mě bavilo hraní podobných her a už delší dobu mě zajímá programování, tak jsem chtěl tyto dvě záliby skloubit. V následujících kapitolách je popsána samotná herní konzola, způsob vykreslování na displeji, popis joysticku a vytváření zvuků reproduktorem. Dále jsou popisovány jednotlivé aplikace, jejich základní podstata, ovládání, vykreslování. Jedná se o aplikace Hledání min, dobře známá ze stolních počítačů, dále je to nejúspěšnější hra všech dob Tetris a třetí aplikací je Spaceman, obdoba hry z mobilních telefonů.

1 KONZOLA

Konzola je tvořena procesorem typu AVR ATmega32 firmy Atmel, obsahuje 32kB Flash programové paměti, 1kB EEPROM paměti a 2kB SRAM paměti. Dále je konzola tvořena LCD displejem a reproduktorem. Ke konzole se pro účely snadné ovladatelnosti her připojuje joystick. Na obr. 1.1 je zobrazeno blokové schéma konzoly a na obr. 1.2 její výsledné provedení. V příloze je obvodové zapojení celé konzoly.



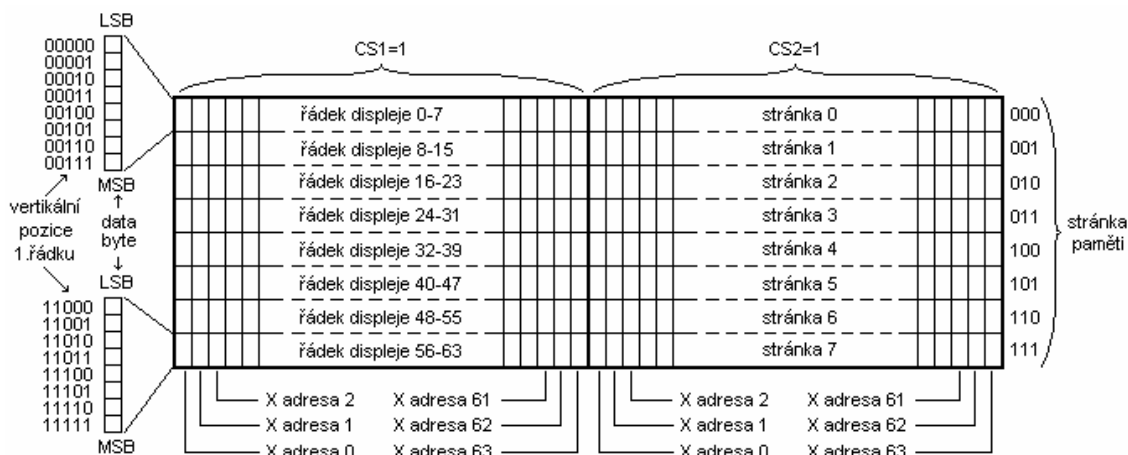
Obr. 1.1 - Blokové schéma konzoly



Obr. 1.2 - Konzola

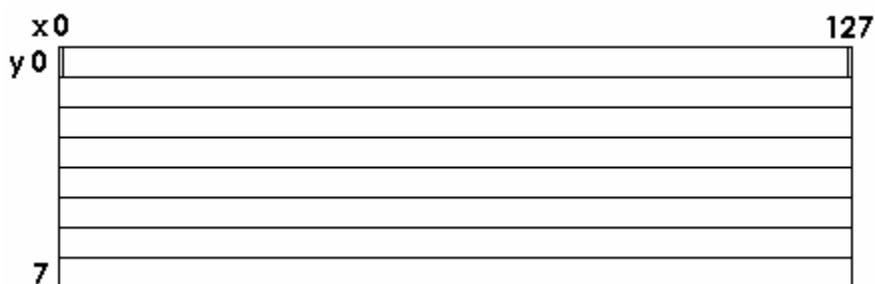
1.1 Displej

Jedná se o LCD s rozlišením 64x128 bodů. K dostání je pod různým názvem (ATM12864, PG12864...) a v různém barevném podsvětlení (modrá, bílá, zeleno-žlutá) Toto LCD obsahuje dva samostatné řadiče, z nichž každý ovládá jednu polovinu displeje, tudíž plochu 64x64 bodů. Každá tato polovina obsahuje ještě 8 stránek. Při zápisu na displej se tedy nezapisuje každý bod samostatně, ale vykreslují se byty v řádcích (stráncích). Tento způsob zápisu je zobrazen na obr. 1.3 (tento obrázek byl převzat a upraven [3]).



Obr. 1.3 - Popis zápisu na displej

Tento zápis na LCD je ovšem velmi nevhodný při vykreslování objektů u složitějších her. Proto bylo nutné vytvořit tzv. video paměť, framebuffer (ve zdrojových kódech označovaná jako VIDEO_RAM). Rozvržení bytů ve video paměti je zobrazeno na obr. 1.4.



Obr. 1.4 - Uspořádání bytů ve video paměti

V RAM paměti procesoru je tedy alokován prostor o velikosti 1kB. Tato velikost je dána počtem bodů na šířku (128 bodů) vynásobena počtem stránek (8 stránek, řádků).

$$128 * 8 = 1024 \text{ B} = 1 \text{ kB}$$

Vykreslují se sice pořád samostatné byty (8 bitů - 8 zobrazovaných bodů), ovšem při zapisování odpadá povinnost přepínat mezi jednotlivými polovinami displeje a zapisujeme tedy celý řádek o 128 bodů. O vykreslování se stará samotný podprogram přerušeni vyvolávaný čítačem/časovačem 2, který v daném časovém intervalu vypisuje

celou video paměť (obnovovací frekvence je 30 Hz). Uživatel tedy pouze zapisuje potřebná data do RAM. Mezi další přednosti tohoto vykreslování je možnost rychlého zpracovávání a využívání těchto dat, jako je připsování nebo umazání jednotlivých bitů (bodů) v zobrazovaném bytu, jednoduchým přičtením pomocí logického součtu OR (připsání bodu) nebo násobením logického součinu AND (umazání bodu). Pohybování se v tomto druhu paměti není nijak složité. V jazyce C jde o matici s 128 sloupci a 8 řádky, v JSA (jazyce symbolických adres, assembler) je tato paměť definována jako posloupnost bytů. Způsob jak v JSA přistupovat k jednotlivým bytům v paměti je pomocí jednoduchého výpočtu, kde musíme znát souřadnice x a y:

offset ... počáteční adresa video paměti

x ... prezentuje sloupce (0 - 127)

y ... prezentuje řádky (0 - 7)

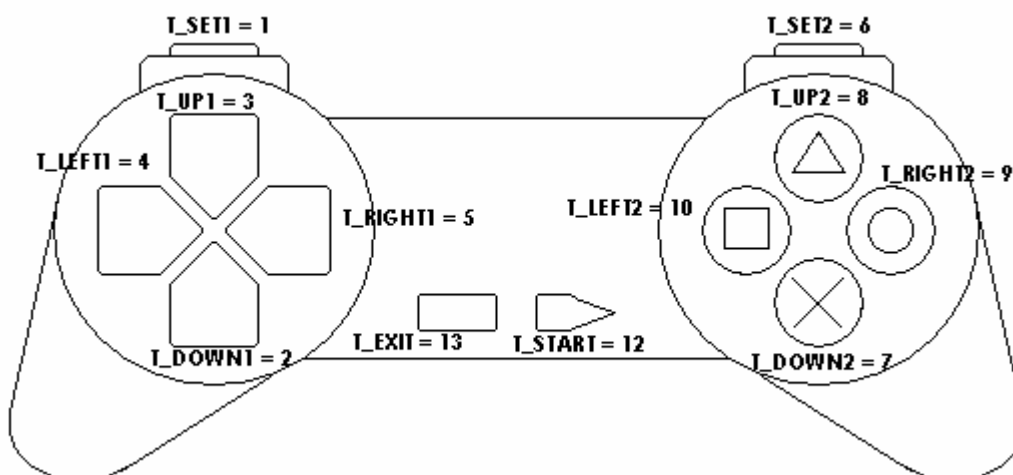
adr ... prezentuje adresu bytu

$$\text{adr} = \text{offset} + (y * 128) + x$$

Cílem konzoly Uzebox byla podpora grafických režimů s 256 barvami, 8 úrovní červené, 8 úrovní zelené a 4 úrovně modré. Pro zobrazení obrázku nebo hrací plochy o velikosti např. 240x224 pixelů a použití 8 bitů reprezentující barvy na pixel by framebuffer musel mít velikost 52kB. Toto není realizovatelné z nedostatku SRAM u procesoru ATmega644 i ATmega32. Proto využívají k vykreslování tzv. dlaždic a tyto dlaždice pak tvoří výsledný obraz. V paměti jsou uloženy pouze 16-ti bitové (2B) ukazatele na oblast v paměti Flash, odkud se daná dlaždice vykreslovala. Takže pro obraz o rozlišení 240x224 a velikosti dlaždice 6x8 pixelů je potřeba 40x28 dlaždic. Ve výsledku se v paměti vyhradila paměť $40 \times 28 \times 2 = 2240$ bajtů. Tento režim se nazývá Mode1. Další režimy se liší rozlišením obrazovky, velikostí dlaždic a dalšími vlastnostmi ovlivňující využití RAM paměti, rychlosti CPU. Pouze Mode 8 je bitmapový režim se skutečným framebufferem. Ovšem rozlišení je pouze 120x96 pixelů a počet barev snížen na 4, spotřeba paměti je tedy rovna $120 \times 96 \times 2 = 2880$ bajtů = 2kB. Další zajímavý režim je Mode 9, využívající taktéž dlaždic, avšak data jednotlivé dlaždice nejsou čteny z Flash, ale každá dlaždice realizována samostatným podprogramem, což urychluje vykreslování, protože instrukce typu LDI trvá pouze jeden strojový takt.

1.2 Joystick

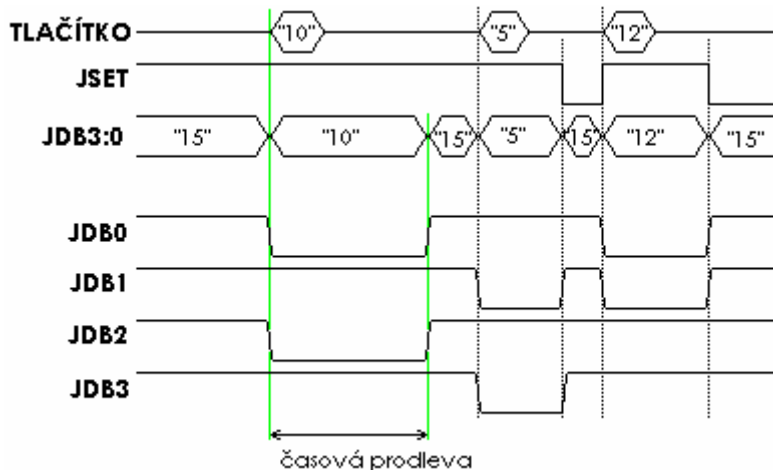
Pro snadné a intuitivní ovládání her se používá joystick. Ten se připojuje pomocí 15-ti pinového CAN konektoru k boční straně konzoly. Obsahuje 12 tlačítek, z nichž dvě čtveřice jsou uzpůsobeny jako směrová tlačítka, další dvě tlačítka na horní straně pro střelbu či potvrzování a tlačítka Start a Exit. Rozvržení tlačítek na joysticku a jejich kód je zobrazeno na obr. 1.5.



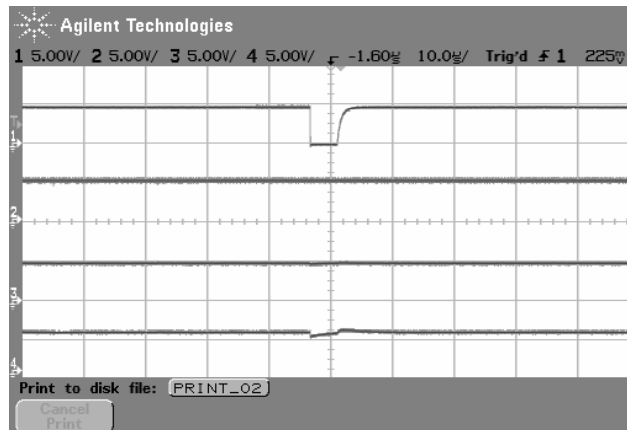
Obr. 1.5 - Rozvržení tlačítek na joysticku

Při potřebě zjištění jakéhokoliv tlačítka se volá podprogram (funkce) `cti_joy`, který vrací hodnotu stisknutého tlačítka (v případě nestisknutého tlačítka hodnota `0x0F`), v JSA je tato hodnota uložena v registru `r_joy` (registr `r22`) nebo `acc` (registr `r16`).

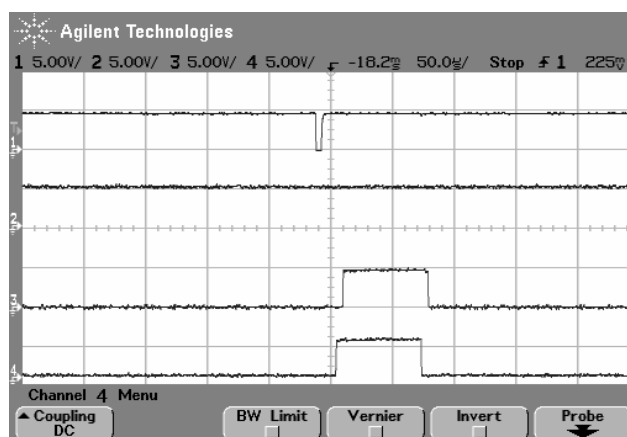
Joystick používá ke komunikaci 7 vodičů, 2 napájecí (+5V, zem), 4 datové (JD3:0) a jeden potvrzovací (JSET). Způsob komunikace a časová prodleva jsou naznačeny na obr. 1.6. Na obr. 1.7 a obr. 1.8 jsou časové průběhy z osciloskopu připojeného ke konektoru joysticku a konzoly. Kanál 1 osciloskopu je připojen k vývodu JSET, kanály 2, 3 a 4 jsou připojeny k datové sběrnici joysticku. Jelikož řadič joysticku pracuje na nižších kmitočtech než ATmega32 (v řádu kHz), je na obr. 1.8 po vyslání potvrzovacího signálu JSET vidět časové prodlevy mezi jednotlivými instrukcemi, nastavení vývodů do pozice T_NIC a následné nastavení do pozice dle stisknutého nebo drženého tlačítka.



Obr. 1.6 - Popis signálů na sběrnici joysticku



Obr. 1.7 - Záznam z osciloskopu při nestisknutém žádném tlačítku



Obr. 1.8 - Záznam z osciloskopu při stisknutém tlačítku T_UP1

1.3 Reprodaktor

Pro ještě větší zážitky při hraní her patří hudba na jejich pozadí k neodmyslitelné součásti hry. U této herní konzoly si musíme vystačit pouze s obdélníkovým signálem, který je generován čítačem/časovačem 0 o frekvencích od velké oktávy (od 65,406 Hz) po dvojjárkovou oktávu včetně (do 1 975,533 Hz), viz [9]. Procesor je sice schopen generovat tóny vyšších frekvencí, je to ovšem zbytečné. Každá nota má vlastní frekvenci a délku, frekvence je potom prezentována unikátním nastavením registrů časovače 0 a délka prodlevou, v níž je tato frekvence generována. Generování zvukového signálu je na vývodu PB3, ke kterému je připojen trimr pro řízení hlasitosti zvuku a k němu je připojen miniaturní reproduktor.

Při správném zadání frekvence (oktávy a výšky noty) a délky noty (pro pomlku je frekvence rovna 0) se vytvoří sekvence, která se dá bez problému přehrávat. Na stejném principu funguje hudební formát .MID [8]. Takováto sekvence lze vytvořit velmi jednoduše za pomoci jakéhokoliv zpěvníku nebo zápisu not. Jediná nevýhoda přehrávání na této konzole je, že daná sekvence se musí načítat během právě vykonávané instrukce, tudíž nelze přehrávání spustit na pozadí - absence čtvrtého čítače/časovače, který by zajišťoval na pozadí načítání dané sekvence.

2 HLEDÁNÍ MIN

Patrně nejznámější logická hra, která existuje. Je totiž přikládána ke všem verzím OS Windows, takže se s ní setkal skoro každý člověk používající tento operační systém.

2.1 Cíl hry

Vaším úkolem je označit všechna políčka obsahující miny nebo odkrýt políčka neobsahující miny. Klepnutím na políčko se hráč dozví, zda obsahuje minu, pokud ji neobsahuje, zobrazí se prázdné místo nebo číslo. Prázdné místo znamená, že v okolí není žádná mina a číslo reprezentuje počet min dotýkajících se tohoto políčka (stěn políčka i rohů). To pomáhá hráči zjistit kde se miny zrovna nachází. Pokud odkryjete všechna políčka neobsahující minu hra končí vítězstvím, pokud odkryjete minu prohráli jste. Tato verze hry nabízí pouze jedinou obtížnost a to pole o velikosti 8x8 min. Větší obtížnost nelze na tomto displeji z důvodu nedostačujícího rozlišení zobrazit. Během hry je možné aktuální políčko (zobrazeno negovaně) odkrýt nebo na něj zapíchnout vlaječku pro snazší orientaci kde jsou miny.

2.2 Podstata hry

Celá hra je postavena na matici 8x8 představující minové pole, jež se snažíme bezchybně odkrýt. Každá buňka (políčko) matice může nabývat hodnot od 0 po 15, jedna buňka zabírá v paměti polovinu bajtu. Význam jednotlivých hodnot je uveden v tab. 2.1.

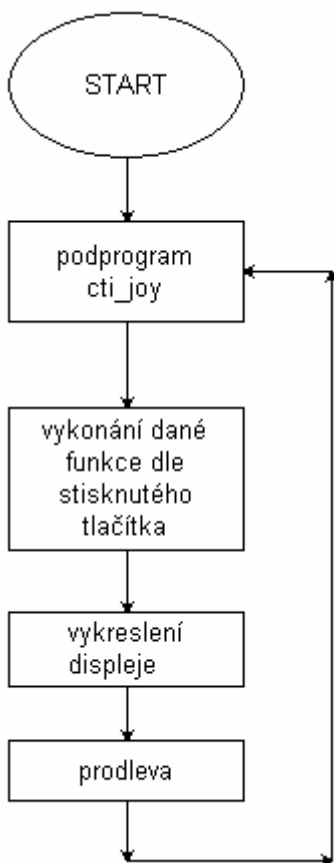
Tab. 2.1 - Význam hodnoty v buňce

hodnota	její význam
0	nic - před otevřením
1	1 mina v okolí
2	2 miny v okolí
3	3 miny v okolí
4	4 miny v okolí
5	5 min v okolí
6	6 min v okolí
7	7 min v okolí
8	8 min v okolí
9	nic - po otevření
10	hodnota pro zjišťování min v okolí
11	vlajka (prázdné pole)
13	vlajka (s minou)
14	mina - po otevření
15	mina - před otevřením

Tab. 2.2 - Přehled důležitých registrů a míst v paměti

název	registr / adresa v SRAM	popis
acc	r16	pro lokální výpočty, hodnota stisknutého tlačítka
XXX	r24	registr sloužící pro orientaci se v matici v horizontální ose
YYY	r25	registr sloužící pro orientaci se v matici ve vertikální ose
JJOY	0x0080	pro porovnání předchozího stisknutého tlačítka
ZOBMIN	0x0081	pro uložení aktuální hodnoty miny
GAME	0x0083	byte v němž jsou uloženy informace o pauze, konci hry
SEKUN	0x0085	zde je uložena délka hraní hry
MINUT	0x0086	
HODIN	0x0087	
POC_MIN	0x0089	celkový počet min ve hře

V tab. 2.2 je přehled nejpoužívanějších a nejdůležitějších registrů. Jelikož u tohoto typu hry (logické) není potřeba věnovat zvýšenou pozornost odezvě na stav tlačítek, je tělo programu tvořeno podprogramem `cti_joy` zjišťující stav tlačítek, prodlevou a vyhodnocováním, jaké tlačítko bylo stisknuto, jak je vidět na obr. 2.1.



Obr. 2.1 - Vývojový diagram hry Hledání min

2.3 Start hry

2.3.1 Generování pole min

Po zapnutí konzoly, nastavení důležitých registrů a provedení inicializace displeje, se na LCD objeví úvodní obrazovka oznamující, že hra je připravena a zahájíme ji stisknutím tlačítka T_START. Po jeho stisku se spustí smyčka čítající posloupnost čísel od 0 do 7 představující pozici miny v jednom řádku matice. Jsme potom ještě dvakrát dotázáni na další stisk jakéhokoliv tlačítka, tyto hodnoty slouží k naplnění matice min. Tímto způsobem je zajištěna po každém startu hry jiná mapa s minami. Po prvním stisknutí se zapíše miny do předem vymazané matice do řádku 0 - 5 na pozici danou generováním posloupnosti a zapisují se hodnoty „15“ (mina před otevřením). Po druhém stisku se zapíše do 7. řádku, 2x do 6. řádku a pak do řádku 0, 2, 4. Jelikož je možné, že se hodnoty na těchto řádcích přepíše, mění se tedy počet min mezi jednotlivými mapami (hrami). Po tomto naplnění se vykreslí mapa s polem (podprogram vyk_mapu), vynuluje se časomíra a spustí se časovač 0.

2.3.2 Samotná hra

Po spuštění hry se volá podprogram cti_joy pro čtení stavu tlačítek. A podle daného stisknutého tlačítka se vybere příslušná akce, jak je uvedeno v tab. 2.3.

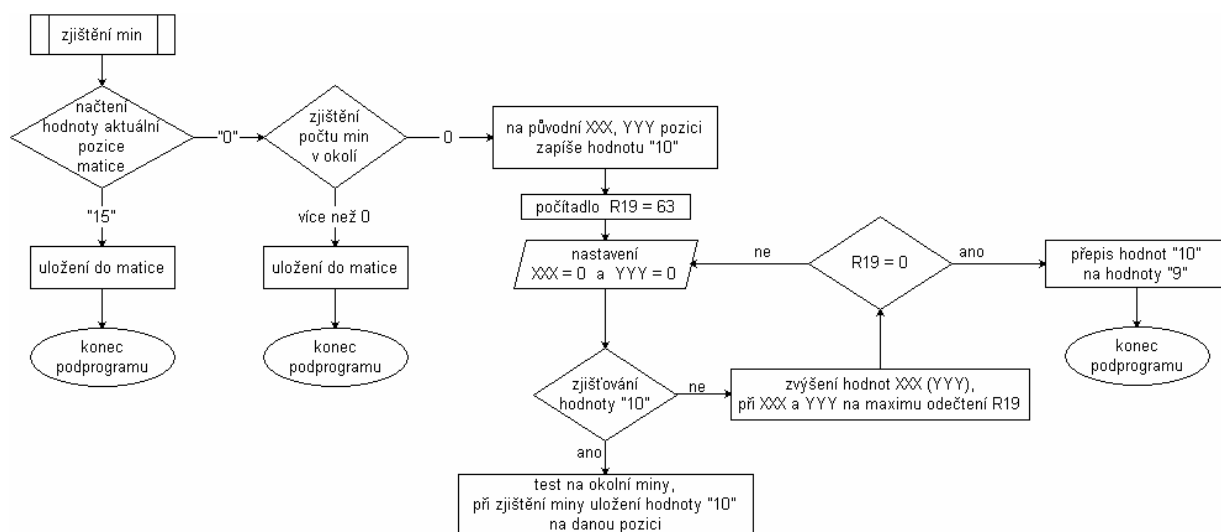
Tab. 2.3 - Popis akce vyvolané po stisku tlačítka

tlačítko	akce
T_EXIT	ukončení hry
T_START	vyvolání pauzy
T_UP1	pohyb v minovém poli
T_DOWN1	
T_LEFT1	
T_RIGHT1	
T_DOWN2	odkrytí políčka
T_RIGHT2	
T_UP2	zapíchnutí vlaječky
T_LEFT2	

Po stisku tlačítka T_EXIT se provede skok na začátek celého programu a vynulují se potřebné registry. Po stisku T_START během hry se na displeji vykreslí nápis, zastaví se běh čítače/časovače 0 (počítání hodin) a začne se testovat joystick na stisk tlačítka T_START, jiným tlačítkem se není možné z této smyčky pauzy dostat. Během této smyčky je zastaveno vykreslování mapy min a nápis označující pauzu je tedy na displeji zobrazen do jejího ukončení.

Stiskem směrových tlačítek s příponou 1 se mění hodnota v registrech XXX a YYY vždy o 1 (jsou inkrementovány nebo dekrementovány) a tím se mění naše pozice v minovém poli. Pokud jsme na okraji minového pole (na okraji matice) a stiskneme tlačítko pro pohyb ještě dál z minového pole dostaneme se na jeho opačnou stranu.

Po zmáčknutí tlačítek T_DOWN2 nebo T_RIGHT2 se odkrývá aktuální políčko. Přečte se hodnota buňky na souřadnicích XXX a YYY a začne vyhodnocování. Pokud je to mina („15“), zapíše se do SRAM do proměnné GAME hodnota „1“ představující konec hry. Pokud tam ovšem mina není („0“) začne se zjišťovat počet min v okolí tohoto políčka (dotyk stěn i rohy, vzdálenost 1). Pokud v okolí toho políčka je několik min, zapíše jejich počet zpátky do matice (počtu min odpovídá daná hodnota). Když v okolí není žádná mina, zapíše se do matici hodnota „9“ a začnou se prohledávat další políčka. Všechny políčka ve vzdálenosti 1 se začnou testovat na počet min v jejich okolí. Pokud u některého políčka nebude zjištěna v okolí žádná mina, budou se její okolní buňky znovu testovat na počet min v jejich okolí, atd. Na obr. 2.2 je vývojový diagram odhalování okolních min.

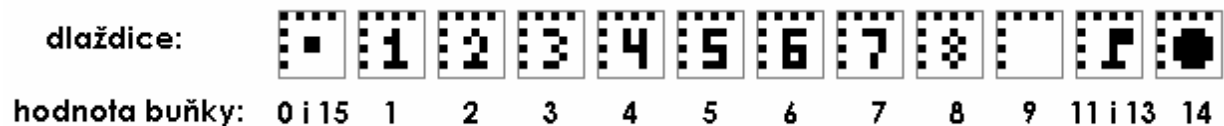


Obr. 2.2 - Vývojový diagram odhalování okolních min

Zmáčknutím tlačítka T_UP2 nebo T_LEFT2 se načte aktuální hodnota buňky. Při hodnotě „0“ (neotevřená bez miny) se do matice zpátky uloží hodnota „11“ (vlajka bez miny), tato hodnota představuje námi špatně předpokládanou pozici miny a tudíž špatně položenou vlajku. Při této změně se sníží zobrazovaný počet neodhalených min. Naopak při stisku příslušného tlačítka a načtené hodnotě „11“ (vlajky bez miny) se do matice vrátí hodnota „0“ (neodkrytá pozice bez miny) a zvyšuje se počet neodhalených min. To stejné platí i pro hodnoty „15“ (neodkrytou minou) a hodnotu „13“ (vlajku s minou).

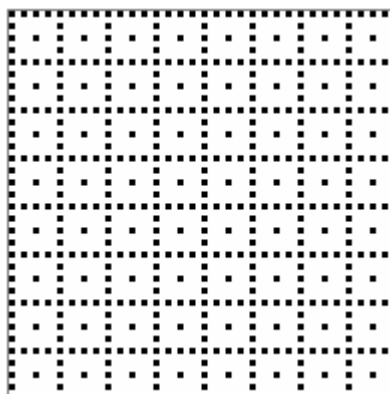
2.4 Zobrazování

Vždy po vykonání akce vyvolané stiskem tlačítka se smaže displej a vykreslí se mapa s minami. Vykreslování funguje na stejném principu jako Model u konzoly Uzebox. Čtou se postupně jednotlivé buňky matice a podle hodnoty této buňky se vykreslí daná dlaždice. Na obr. 2.3 jsou ukázány jednotlivé dlaždice a hodnoty podle ní jsou dlaždice vykreslovány.

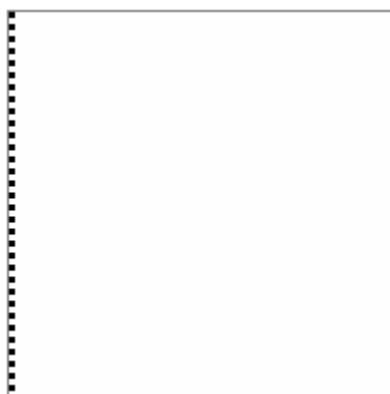


Obr. 2.3 - Ukázka jednotlivých dlaždic

Stejně jako u Mode1 jsou tyto dlaždice zapsány ve Flash (programové) paměti a jednotným podprogramem vykreslovány do levé poloviny displeje. Každá dlaždice má rozměry 8x8 pixelů a vyobrazuje nejen rast minového pole, ale i informace samotného políčka (počet min, vlajku, atd.). Na obr. 2.4 je ukázka vykreslených neodkrytých dlaždic.



Obr. 2.4 - Ukázka vykreslení levé poloviny displeje



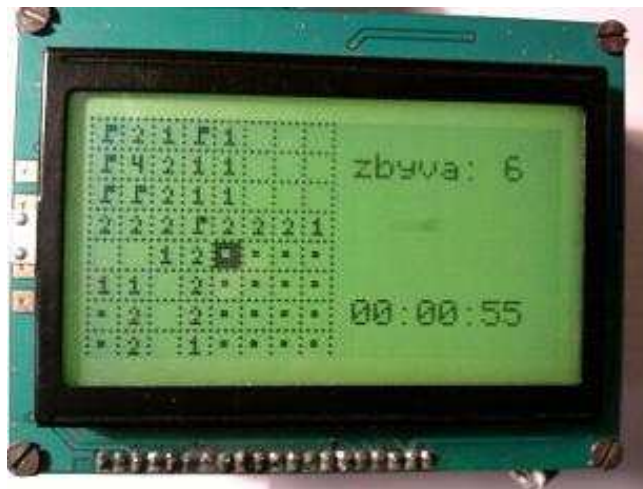
Obr. 2.5 - Ukázka vykreslení pravé poloviny displeje bez textu

Po vykreslení minového pole se vykresluje pravá polovina displeje. Tato polovina se vykresluje nezávisle na tom jestli bylo nějaké tlačítko stisknuto, protože jsou zde zobrazována časomíra a musí být zobrazována alespoň jednou za vteřinu (prakticky je pravá polovina vykreslována vícekrát za vteřinu).

Nejprve se vykreslí svislá čára, která je na obr. 2.5. Provede se test celé matice na počet vlajek, tento počet se odečte od celkového počtu min zjištěného na začátku hry při

jejich generování a představuje počet ještě neodhalených min. Vypíše se text: „zbyva:“ a číslice s tímto počtem. Pak se zobrazují číslice časoměry. Vypisování textu je založeno takéž na zobrazování jednotlivých dlaždic o rozměrech 6x8 pixelů. Každá dlaždice představuje jedno písmeno, znaky (dlaždice) jsou uloženy v EEPROM paměti a jdou poskládány v pořadí jako v ASCII tabulce.

Po vykreslení mapy min a textu se provede testování celé matice na hodnoty „0“ (prázdné neodkryté políčko) a „11“ (vlajka bez miny), jestliže v matici tyto hodnoty nejsou, znamená to, jsme odkryli správně všechna políčka a do paměťového prostoru na adrese GAME se zapíše hodnota 2. Po tomto testu se čte zmíněný prostor GAME. Hodnota „0“ znamená pokračování hry, hodnota „1“ prohru jelikož byla odkryta mina a hodnota „2“ vítězství. Pokud hra končí je vypsán text informující nás o výhře nebo prohře. Příklad jak vypadá hra je na obr. 2.6 a obr. 2.7.



Obr. 2.6 - Ukázka hry Hledání min



Obr. 2.7 - Ukázka konce hry Hledání min

3 TETRIS

Tetris je počítačová hra rozšířená po celém světě. Tuto hru vyvinul Rus Alexendrej Pažitnov v roce 1984. Je nejslavnější hrou všech dob, částečně díky svému úspěchu jako videohra. Tetris a jemu podobné klony se totiž objevili na téměř na každém přístroji kde se dali hrát hry [4]. Jedná se základní verzi hry, kdy je možno manipulovat s 19 typy kostek. Každá kostka se skládá ze čtyř čtverečků, buněk.

3.1 Cíl hry

Vaším úkolem je skládat tyto padající kostky do zdi. Při vyplnění řádku bez děr, je řádek smazán. Manipulace s kostkou představuje její posun doprava, posun doleva nebo rotace vpravo nebo rotaci vlevo. Pro urychlení hry je možno padání kostky zrychlit. Hra končí když zeď dostoupá k hornímu okraji hrací plochy. Se zvyšujícím se počtem smazaných řádků se zvyšuje rychlost padání kostek a hra nabývá na náročnosti.

Zdatní hráči si mohou před zahájením hry si můžou nastavit počáteční rychlost padání kostek a úroveň od které chce hráč začít hrát.

Najednou mohou zmizet nejvýše 4 řádky, za pomoci „I“ (kostka č. 1), to se nazývá tetris. Čím více řádků zmizelých najednou tím větší přírůstky na skóre, v tab. 3.1 jsou vypsány přírůstky na skóre v závislosti na počtu smazaných řádků a rychlosti padání kostky.

Tab. 3.1 - Velikost přírůstku v závislosti na rychlosti a počtu řádků

Počet řádků	Rychlost padání kostek										
	0	1	2	3	4	5	6	7	8	9	10
jeden	40	80	120	160	200	240	280	320	360	400	440
dva	100	200	300	400	500	600	700	800	900	1000	1100
tři	300	600	900	1200	1500	1800	2100	2400	2700	3000	3300
čtyři (tetris)	1200	2400	3600	4800	6000	7200	8400	9600	10800	12000	13200

3.2 Podstata hry

Hra je založena na matici 10x20, 10 sloupců a 20 řádků. Každá buňka matice může nabývat několika hodnot, které jsou uvedeny v tab. 3.2.

Tab. 3.2 - Význam hodnot buňky matice

hodnota	název	popis
2	PADA	padající buňka
1	SPAD	dopadnutá (ležící) buňka
0	NIC	žádná (prázdná) buňka

Jelikož tyto informace zabírají jen 2 bity (1/4 B) jsou pro úsporu paměti SRAM

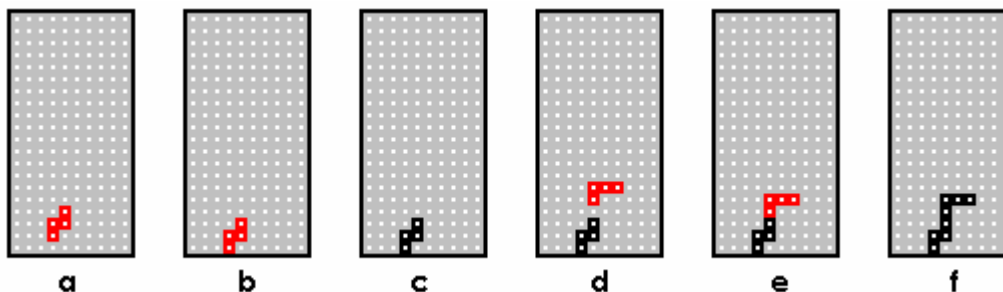
v jednom bajtu uloženy informace rovnou 4 buněk. Tímto způsobem zabírá matice v SRAM pouhých 50B místo 200B. Po začátku hry nebo dopadu kostky je generována nová kostka, každá kostka má své identifikační číslo, a podle něj jsou do matice na souřadnice XXX a YYY určené touto kostkou zapsány do buněk hodnoty PADA (padající kostky). Na obr. 3.1 je přehled všech generovaných kostek.



Obr. 3.1 - Přehled kostek a jejich hodnot

3.3 Vlastní hra

Základ hry samotné tvoří posun kostek po určitém časovém intervalu. K tomuto účelu je využit čítač/časovač 1. V jehož podprogramu přerušení se samotný posun kostek provádí. Na začátku podprogramu je počítadlo vyvolání přerušení CC1, je zde z důvodu změny rychlosti padání jednotlivých kostek. Při větší hodnotě počítadla je samotný posun vyvolán méně často. Na obr. 3.2 je naznačen způsob padání a přeměny kostek, prázdné buňky (hodnota NIC) představují šedé čtverečky, padající buňky (PADA) červené čtverečky a dopadnuté buňky (SPAD) černé čtverečky.



Obr. 3.2 - Ukázka padající kostky a její změny

Na začátku se zjistí zda se spodní buňky kostky dotýkají dna hrací plochy (nastaví se souřadnice matice pro $YYY=19$ (20. řádek matice) a provede se cyklus s testem na buňky obsahující hodnotu PADA), pokud jsou spodní buňky na dně hrací plochy změní se padající kostka (všechny buňky s hodnotou PADA v matici) na dopadnutou nebo-li ležící, změnou z PADA na SPAD, tuto situaci představuje obr. 3.2b a obr. 3.2c.

Dále se v celé matici zjišťuje od předposledního řádku směrem nahoru přítomnost buňky s obsahem PADA, jeli tato buňka nalezena, zjišťuje se zda se pod ní nachází buňka s hodnotou SPAD, pokud se nachází znovu se změní všechny buňky v matici z PADA na SPAD, viz obr. 4.2e a obr. 4.2f. Toto testování je důležité pro samotnou stavbu zdi a zjištění zda se má kostka kam posunout.

Pokud se kostka nenachází na dně hrací plochy (matice) nebo se pod jejími jednotlivými buňkami nenachází už dopadnuté kostky, posune se kostka o jednu pozici směrem dolů. Posun se provádí zjišťování matice od předposledního řádku postupně nahoru na hodnotu PADA, ta se zapíše do stejného sloupce o řádek níže a původní místo se nahradí hodnotou NIC (prázdné buňky).

Při změně buněk se provede zápis do proměnné GAME. Po těchto akcích se provede přepis matice do video paměti.

3.4 Ovládání

Před samotným spuštěním hry je možnost nastavit si výchozí parametry. Jedná se nastavení rychlosti, které se provádí tlačítka T_UP1 (větší rychlost) a T_DOWN1 (menší rychlost). Tato hodnota rychlosti je ukládá do proměnné SPEED uložené v SRAM a může rozsah hodnot 0-10. Nastavení počáteční úrovně se zvyšuje nebo snižuje pomocí tlačítek T_UP2 nebo T_DOWN2, tato hodnota se ukládá do proměnné LEVEL a může nabývat hodnot 0-17. Před startem je u této aplikace možnost zapnutí zvukových efektů stiskem tlačítka T_EXIT. V tab. 3.3 je přehled nejdůležitějších proměnných v SRAM a registrů.

Tab. 3.3 - Přehled nejdůležitějších proměnných a registrů

název	popis
acc	pro lokální výpočty
XXX	registr r24, orientace v matici v horizontální ose
YYY	registr r25, orientaci se v matici ve vertikální ose
GAME	informace o proběhnutých událostech
SPEED	počáteční / aktuální rychlost padání kostek
LEVEL	počáteční úroveň zaplnění hrací plochy
RAND	náhodná kostka (rozsah hodnot 0-18)
NEXT	2B; 1. bajt - příští kostka; 2. bajt - právě padající kostka
SCORE	6B, každý bajt obsahuje jednu číslici
CC1	počítadlo vyvolání přerušení - pro změnu rychlosti padání kostek

Po stisku T_START se vymaže celá matice, vymaže se skóre, do 1. bajtu proměnné NEXT se zapíše (zkopíruje) náhodně vygenerována kostka z proměnné RAND, do matice se zapíše vybraná kostka, do 2. bajtu NEXT se přesune 1. bajt NEXT a zapíše se nová hodnota RAND náhodné následující kostky, která bude zobrazena vedle hrací plochy, nastaví se rychlost padání kostek a nastaví se počáteční úroveň, příklad uživatelem navolené počáteční úrovně je na obr. 4.8 a obr. 4.9. Celá matice se přepíše do video paměti (podprogram ZAPIS_RAM) a spustí se hra. Ve hře mají tlačítka funkce odpovídající tab. 3.4.

Tab. 3.4 - Funkce jednotlivých tlačítek

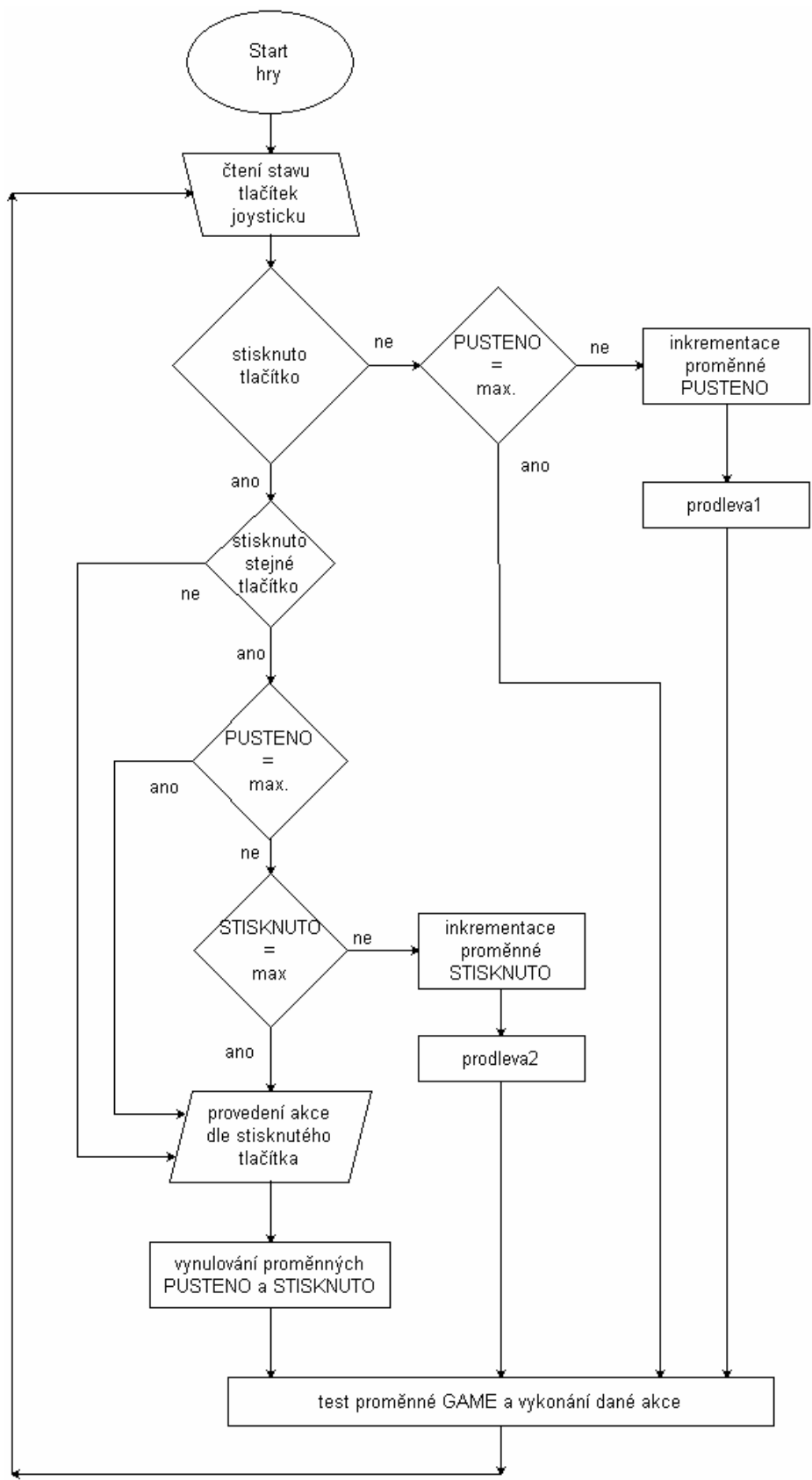
tlačítko	akce
T_LEFT1	posun kostky vlevo
T_RIGHT1	posun kostky vpravo
T_DOWN1	zrychlení padající kostky
T_UP1	zpomalení rychle padající kostky
T_LEFT2 a T_DOWN2	rotace kostky vlevo
T_RIGHT2 a T_UP2	rotace kostky vpravo
T_EXIT	ukončení hry
T_START	vyvolání pauzy

U tohoto typu hry je velmi důležitá odezva stisknutých tlačítek, závisí na tom rychlost posunu kostky na námi vybranou pozici a rychlost otočení. Jelikož se po zjištění stavu tlačítek odesílá informace o jeho přijetí, posílá nám joystick další stav tlačítek. Čtení stavu se provádí ve smyčce, může nám program i krátké stisknutí vyhodnotit jako několik stisknutí po sobě. Tomuto se dá vyhnout použitím správně dlouhé prodlevy.

Jelikož tělo hlavního programu tvoří test stavu tlačítek, vykonání dané akce po jejich stlačení a test na proměnou GAME, je možné vytvořit algoritmus testovací stav tlačítek s mnohem citlivější odezvou. Na obr. 3.3 je vývojový diagram algoritmu zajišťující mnohem lepší odezvu než použití pouze prodlevy.

Nejprve se testuje zda bylo nějaké tlačítko stisknuto. Pokud bylo právě puštěno některé tlačítko, je prakticky měřena doba, jak dlouho toto tlačítko nebylo stisknuto. Stiskneme-li to stejné tlačítko až po delší době (větší než je počet proměnné PUSTENO) provede se jeho příslušná akce. Stiskneme-li to jej dříve, bude generována vícekrát krátká prodleva2 (počet odpovídá proměnné STISKNUTO), tato prodleva může dosahovat řádu sekund. Stejný případ nastává při držení tlačítka, kdy potřebujeme, aby se nám akce vykonala jednou a pak se čeká velmi dlouho než se provede akce znovu nebo než se tlačítko pustí a testuje se jak je dlouho puštěno.

Při stisku jiného tlačítka je automaticky provedena jeho akce a negenerují se žádné prodlevy.



Obr. 3.3 - Vývojový diagram algoritmu testující tlačítka

3.5 Akce jednotlivých tlačítek

Při stisku T_START během hry se zastaví běh časovače 1 představující pohyb kostek směrem dolů. Do video paměti a tedy i LCD se zapíše text informující nás o zapnuté pauze. Během této pausy se pouze testuje stav tlačítka T_START na ukončení pausy a T_EXIT na ukončení hry.

Po stisku během hry tlačítka T_EXIT se hra ukončí. Program se vrátí na začátek, vynulují se registry a proměnná GAME. Objeví se úvodní obrazovka a čeká se na spuštění nové hry.

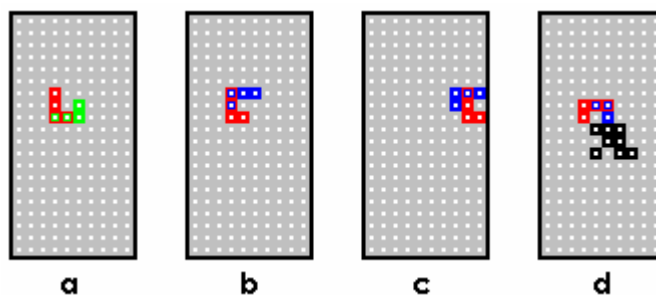
Funkce tlačítek T_RIGHT1 a T_LEFT1 je skoro stejná. Obě posunují celou kostku doprava nebo doleva. Nejprve se testuje krajní sloupec podle směru pohybu, při směru doprava se testuje pravý krajní sloupec a naopak. Obsahuje-li krajní sloupec buňku s hodnotou PADA není možné kostku posunout, protože není kam. Není-li v krajním sloupci žádná padající kostka (buňka), testují se ostatní sloupce na přítomnost buněk s obsahem hodnoty PADA, je-li taková buňka nalezena, je ve směru pohybu (vodorovně) testována její sousední buňka na přítomnost hodnoty SPAD, nemůžeme totiž posunout kostku na místo kde už je postavená jiná. Pokud tedy kostka není u krajní stěny nebo vedle ní nejsou spadnuté kostky, jsou buňky padající kostky posunuty.

Tlačítkem T_DOWN1 se zvýší rychlost padání kostky. Jelikož je u časovače1 (řízení posunu kostky směrem dolů) použito počítadlo vyvolání přerušení CC1, je po stisku T_DOWN1 nastavena hodnota počítadla CC1 na nejnižší hodnotu, tudíž se vyvolání posuvu zrychlí. Po dopadu kostky, její změny na dopadnutou, se načte z proměnné SPEED hodnota rychlosti a počítadlo vyvolání přerušení CC1 se nastaví na původní hodnotu.

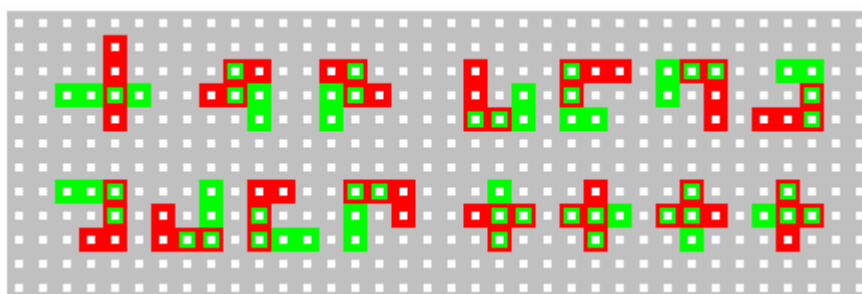
Máme-li kostku urychlenou, lze ji u této verze Tetrisu zpomalit tlačítkem T_UP1. Po jeho stisknutí dojde ke změně počítadla CC1 opět na původní hodnotu.

Stiskem druhých směrových tlačítek se padající kostka otočí doprava (T_RIGHT2, T_UP2) nebo doleva (T_LEFT2, T_DOWN2). K rotaci je snazší využívat informace o aktuálně padající kostce a podle druhu směru otáčení zapsat do matice jinou kostku. Příklad rotace je uveden na obr. 3.4, kde aktuálně padající kostka je kostka č. 11 (červeně), při rotaci vlevo (obr. 3.4a) se místo původní kostky zapíše kostka č.14 (zeleně) a při otáčení doprava (obr. 3.4b) se zapíše kostka č.12 (modře).

Při rotaci se musí testovat pozice kostky vůči stěnám hrací plochy, na obr. 3.4c je vidět, je-li „stará“ kostka u stěny, musí „nová“ kostka posunout, dále se musí testovat ještě okolní buňky (zelené pro rotaci doleva, modré pro rotaci doprava) na přítomnost spadnutých kostek (viz obr. 3.4d), v tomto případě otočení není možné, spadnuté kostky jednoduše zavazí. Kromě kostky „0“ (která se neotáčí) se pro každý směr a každou kostku testují okolní buňky zvlášť. Na obr. 3.5 jsou naznačeny veškeré možné rotace prováděné v prostoru, jsou-li kostky u stěny hrací plochy jsou některé posunovány, rotace vpravo je zelená->červená a rotace vlevo je červená->zelenou.



Obr. 3.4 - Ukázka rotace padající kostky



Obr. 3.5 - Přehled všech možných rotací

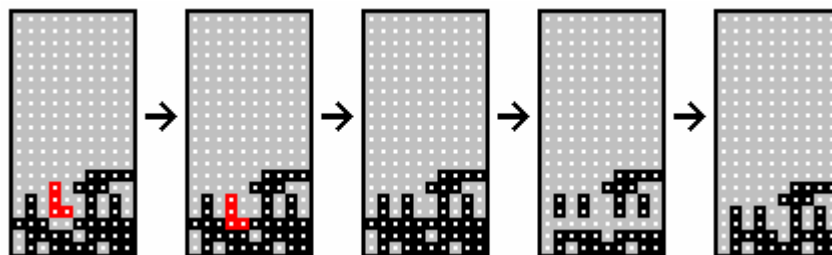
Testování proměnné GAME se provádí ihned po testování stavu tlačítek joysticku nebo po provedení akce daného tlačítka. V tab. 3.5 je uveden popis jednotlivých bitů proměnné GAME.

Tab. 3.5 - Popis jednotlivých bitů proměnné GAME

bit	popis
0	stisknuto tlačítko
1	spadnutá kostka
2	pauza
3	nová kostka překrývá spadnuté (GAME OVER)

Po stisku jakéhokoli tlačítka se zapíše log. 1 na první pozici (bit 0). Tohoto je využíváno při generování náhodných čísel pro novou kostku. Při generování čísel se tato hodnota vynuluje.

Pokud je padající kostka přeměněna spadnutou, je zapsána log. 1 na druhou pozici (bit 1). Je-li při testování této proměnné zjištěna log.1, otestují se nejdříve všechny řádky matice zda všech deset buněk obsahuje hodnotu SPAD (zda je celý řádek zaplněn). Pokud ano, je celý řádek smazán (viz obr. 3.6) a všechny kostky nad tímto řádkem posunuty směrem dolů. Počet řádků se vynásobí hodnotou SPEED (rychlosti padání) zvýšenou o 1 a toto číslo se přičte ke skóre. Poté je na hrací plochu „poslána“ nová kostka, hodnota NEXT (další kostka) se zkopíruje do NEXT+1 (druhý bajt proměnné NEXT, aktuálně padající kostka) a do proměnné NEXT se zkopíruje hodnota RAND (náhodné číslo kostky). Potom je tato pozice v GAME vynulována.



Obr. 3.6 - Ukázka gravitace hry Tetris

Stiskem během hry tlačítka T_START se do GAME na 3. pozici zapíše log.1, která při testování proměnné GAME vyvolá pauzu.

Je-li do matice zapisována „nová“ kostka testují se jednotlivé buňky na přítomnost už spadnutých buněk (SPAD), je-li alespoň na jedné pozici takováto hodnota, je „nová“ buňka zapsána do matice, přepíše stávající a je i do proměnné GAME zapsána log.1 na pozici 4, značící konec hry.

3.6 Generování náhodných kostek

Po inicializaci CPU, LCD a registrů se do SRAM paměti se na adresu RAND uloží hodnota „0“. Tato proměnná může nabývat hodnot od 0 po 18, to odpovídá počtu kostek. Při vyvolání přerušování časovače 2, je tato proměnná pokaždé několikrát zvýšena o 1 (inkrementována). Kolikrát je zvýšena závisí na třech událostech. Pokud zrovna dopadla kostka je proměnná RAND inkrementována jednou, pokud bylo stisknuto jakékoliv tlačítko je inkrementována dvakrát a jestliže nenastala ani jedna z těchto možností je inkrementována třikrát.

Časovač 2 slouží pro vykreslování displeje, běží tedy na pozadí pořád a je zapnut po inicializaci. Přerušování je vyvoláváno s frekvencí 30Hz, padání kostky je v řádech sekund, což vytváří dostatečně náhodnou posloupnost.

3.7 Vykreslování

Pro zobrazování celistvého obrazu a potřeby pracování s již vykreslovanými daty bylo potřeba vytvořit v SRAM paměti prostor, kam se tyto data uloží a poté vykreslí. O vykreslování se stará čítač/časovač 2. Tento způsob vykreslování je podobný Mode8 u konzoly Uzebox, kdy jednotlivé bity ve video paměti odpovídají jednotlivým pixelům na displeji.

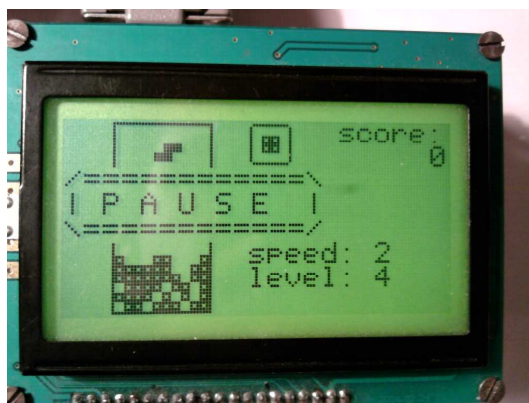
Během hry jsou do video paměti zapisovány data představující matici (hrací plochu), ohraničení matice, následující kostku, počáteční úroveň, aktuální rychlost a skóre. Toto zajišťuje podprogram ZAPIS_RAM. Nejprve se celá video paměť smaže, poté se do ní vypíše matice. Jednotlivá buňka představuje čtvereček o velikosti 3x3 pixely, jelikož řádek má šířku 8 bitů (pixelů), zobrazí se v něm tedy 2 celé čtverečky a může obsahovat část třetího nebo i čtvrtého čtverečku. Na obr. 3.7 je přehled jednotlivých čtverečků představujících buňky matice a jejich hodnota, podle níž jsou zapisovány do řádků video paměti. Pro zobrazení celé matice se každý řádek zapisuje

individuálně. Například pro přepsání prvního řádku matice (souřadnice $YYY=0$) se načte první buňka (souřadnice $XXX=0$) a zjišťuje se její obsah, pokud je prázdná (hodnota NIC), nevykresluje se, pokud obsahuje SPAD zapíše se do prvního řádku čtvereček F ze sady č.1, pokud obsahuje PADA zapíše se čtvereček F ze sady č.2. Při hře se snadno rozezná, které buňky (kostka) padají (sada č.2) a které jen leží (sada č.1). Tímto způsobem zapíše celý řádek matice do video paměti. Pro zobrazení druhého řádku matice ($YYY=1$) se zapisuje čtvereček C znovu do prvního řádku video paměti, musí se ale data přičítat pomocí logického součtu OR, aby se neztratila původní data (už zapsaný první řádek matice). Takto se přepíše celá matice.

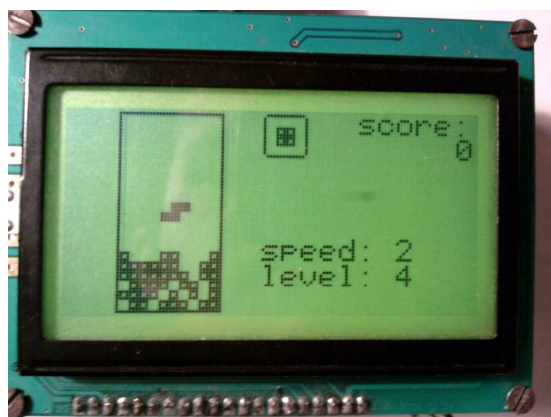


Obr. 3.7 - Zobrazené buňky

Po zapsání matice se zapíše okraje hrací plochy, zapíše se obrázek kostky, která má následovat, zapíše se informace o skóre, informace o počáteční úrovni a o rychlosti padání kostek. Na obr. 3.8 a obr. 3.9 je možno vidět vykreslený celý displej.



Obr. 3.8 - Ukázka hry Tetris



Obr. 3.9 - Ukázka hry Tetris

3.8 Zvuky

Jako jediná aplikace nabízí hra možnost zvukového doprovodu, tento doprovod se nastavuje před startem hry a po skončení hry se vypne. Zvukový doprovod představuje „pípnutí“ nebo „cvaknutí“ po stisku nějakého tlačítka a melodii po ukončení hry.

4 SPACEMAN

Jste pilotem vesmírné stíhačky a musíte se vyhnout všem nástrahám vesmíru a na vzdálené planetě zničit všechny nepřátele.

4.1 Cíl hry

Vaším úkolem je zůstat na živu po dobu pěti úrovní (levelů), lze toho docílit vyhýbáním se nebo ničením objektů a nepřátel. Váš palebný arsenál představuje kulomet, laser (pouze 3 střely) a nukleární střela (pouze 3 střely). Kulomet vystřeluje velmi průrazné náboje ničící cokoli se jim postaví do cesty, na displeji může být zobrazeno až 20 kulek. Laser zničí nepřátele a objekty letící ve stejné výšce jako vy (ve výšce hlavně kulometu/paprskometu). Laser má ovšem vysoké nároky na energii, takže máte pouze možnost vystřelit pouze 3 paprsky. Nejničivější zbraní je nukleární raketa, schopná zničit vše co vidíte. Pod křídly máte 3 takovéto rakety.

První úroveň představuje cesta vesmírem (tmavé pozadí a pomalu pohybující se hvězdy), na níž se musíte vyhýbat nebo sestřelovat asteroidy. Na druhé úrovni vám cestu atmosférou nepřátelské planety doprovází mraky a znepríjemňují nepřátelské letouny a další roje meteoritů. Třetí level představuje let nad zabydlenou oblastí, nesmíte však letět příliš nízko, aby se vám nic nestalo. V cestě vám stojí vysoké mrakodrapy, jež lze sestřelit, a další letka nepřátelských letounů. Na čtvrté a páté úrovni prolétáte jeskyní, nesmíte letět příliš nízko ani příliš vysoko, aby vaši stíhačku nepoškodily krápníky jimiž je celá jeskyně poseta, v páté úrovni proti vám stojí navíc nepřátelská letadla.

4.2 Podstata hry

Hru tvoří pohyb, posunutí, vyvolaný čítačem/časovačem 1. Po vyvolání přerušení je čten stav tlačítek (změna směru, střelba...), dále jsou inkrementována (zvýšena o 1) počítadla vyvolání, viz tab. 4.1.

Tab. 4.1 - Přehled počítadel

počítadlo	nejvyšší hodnota	popis
COUNT_LEVEL	2300	počítadlo délky úrovně
BULLET	2	počítadlo pro rychlost střely
ENEMY	2	počítadlo pro rychlost nepřátele
POZADI	2	počítadlo pro rychlost pozadí

Počítadlo COUNT_LEVEL se zvyšuje při každém vyvolání přerušení. Udává nám délku celé jedné úrovně. Po načítání do hodnoty MAX_COUNT_LEVEL = 2300 se toto počítadlo vynuluje, zastaví čítání čítače/časovače 1 a proměnná SMYCKA je nastavena na 0, tato proměna slouží k ukončení smyčky levelu v těle hlavního programu.

Počítadlo BULLET je zvyšováno také ihned po vyvolání přerušení. Určuje rychlost vystřelené střely a jelikož maximum (MAX_BULLET) je rovno 2, vystřelená kulka má tedy poloviční rychlost než vaše letadlo. Po načítání do maxima se provede posun kulek (kulky putují zleva doprava) zvýší se jejich x-ová hodnota a zvýší hodnota v počítadle ENEMY.

Toto počítadlo udává rychlost udává rychlost pohybu nepřítele. Po dosažení maxima (MAX_ENEMY = 2) se počítadlo vynuluje, provede pohyb nepřítele (asteroidy, meteority, letadla, budovy, krápníky) - odečtení 1 z jeho x-ové souřadnice (pohyb zprava doleva) a zvýší se hodnota v počítadle POZADI. Rychlost pohybu nepřítele je poloviční než rychlost kulek.

To slouží k posunutí pozadí, dokresluje to představu pohybu. Pozadí se pohybuje poloviční rychlostí než je rychlost nepřítele (MAX_POZADI = 2).

Po zvyšování hodnot počítadel se provede test na úroveň levelu a zapíše se podle toho do video paměti pozadí a typ nepřítele. Případně se provede test na letovou výšku stíhačky (viz úroveň 3, 4 a 5).

Nakonec se zapíše (vykreslí) do video paměti stíhačka, vystřelené kulky, případný efekt laseru a do rohů video paměti (displeje) počet zbývajících nukleárních raket a počet laserových střel.

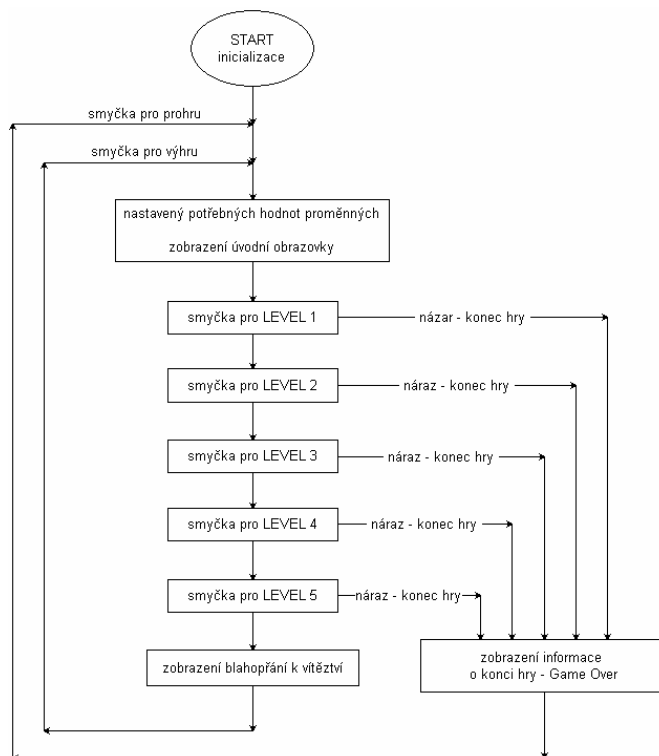
4.3 Hlavní program

Na obr. 4.1 je vývojový diagram těla hlavního programu a v tab. 4.2 je přehled nedůležitějších globálních proměnných. Po zapnutí konzoly, provedení inicializace displeje a nastavení základních parametrů se skočí do dvou smyček (vnější - smyčka pro prohru a vnitřní - smyčka pro výhru), ty slouží pro restart hry v případě výhry nebo prohry.

Následně se vypíše výzva na stisk T_START, při jeho stisku se zahájí hra. Nyní se vypíše informace „LEVEL 1“, zobrazí se animace stíhačky (ze středu displeje se posune k levému okraji) a skočí se do „nekonečné“smyčky v jejíž nitru se vkládají na hrací plochu nepřítele, provede se test na sestřelení nepřítele a test na náraz do nepřítele. Pokud do nepřítele narazíme je z této a následné smyčky vyskočeno a zobrazí se konec hry (výpis „Game Over“ a po chvíli úvodní obrazovka). Tato „nekonečná“ smyčka je však ukončena při napočítání do hodnoty MAX_COUNT_LEVEL, ze smyčky se vyskočí, zobrazí se „LEVEL 2“ a skočí se do druhé smyčky, kde se znovu provádí nahrání nepřítele na hrací plochu a jeho sestřelení. Takto je realizováno všech 5 levelů.

Tab. 4.2 - Popis nejdůležitějších globálních proměnných

proměnná	popis
LEVEL	uložené číslo právě hrané úrovně
POSUN	vzdálenost o kterou je pozadí posunuto
RAKETA	pozice stíhačky(2B pro osu X a Y)
NUCLEAR, LASER	počet zbývajících střel



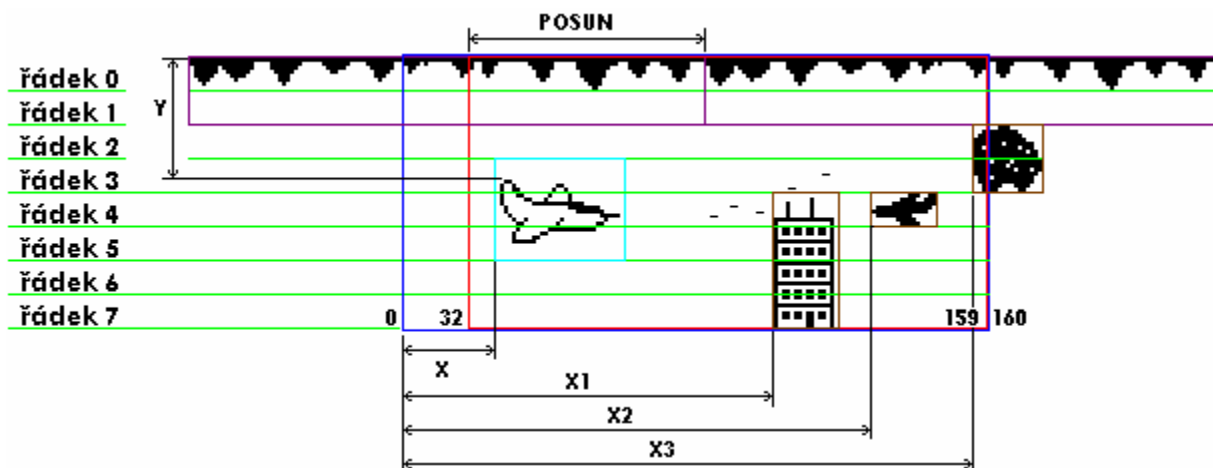
Obr. 4.1 - Vývojový diagram hry Spaceman

Proměnná LEVEL využívá k určení aktuální úrovně, v případě ukončení hry (náraz, T_EXIT) je tedy nepotřebná, uloží se zde hodnota „6“ reprezentující právě ukončení hry. Na tuto hodnotu je LEVEL testován v nejvnitřnější smyčce každé úrovně z níž se vyskočí po zjištění hodnoty „6“ a ve vnitřní smyčce pro výhru z níž se také vyskočí po zjištění této hodnoty. Zobrazí se text „Game Over“, provede se skok do vnitřní smyčky pro výhru, vymažou se potřebné proměnné a čeká se na nové spuštění hry.

4.4 Objekty a jejich vykreslování

K vykreslování se zde používá stejný princip jako u Tetrisu. Data, která se mají zobrazit, se zapisují do video paměti, kde se s nimi dále pracuje a nezávisle na jejich obsahu jsou vykreslována na displej při vyvolání čítače/časovače 2, jehož nastavení je stejné jako u Tetrisu. Do video paměti jsou zapisovány 4 druhy objektů, které se zapisují v pořadí: pozadí, nepřítel, stíhačka a kulky.

Kromě kulek mají všechny zapisované objekty rozměry v násobcích 8 bitů (8 pixelů, např.: 8x8, 16x16, 32x24) pro usnadnění zapisování do video paměti při jejich změně pozice. U vykreslování těchto objektů se vychází z jejich zadaných souřadnic (x-ová osa, řádek displeje-video paměti) a pohyb po x-ové ose se provádí po bodech (od 0 po 160), a změna výšky změnou řádku (0-7). Hrací plocha má tedy rozměry 161x64 bodů, na šířku se vykresluje až od 32. bodu 128 bodů (šířka displeje/video paměti).



Obr. 4.2 - Princip zobrazování bloků ve hře Spaceman

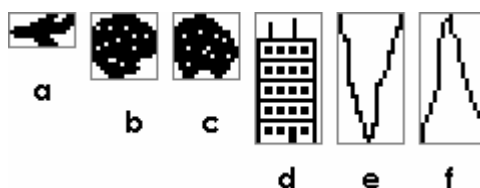
Na obr. 4.2 je naznačen způsob zápisu jednotlivých bloků do video paměti. Video paměť představuje červený rámeček, celá hrací plocha modrý. Hrací plocha je větší z důvodu plynulého zobrazení nepřátel na obrazovce, jejich tzv. „vplutí“ zpoza displeje, např.: meteorit na obr. 4.2.

4.4.1 Pozadí

Pozadí představuje zapisovaný obrázek o rozměrech 128x16 pixelů (bitů). Zapisují se tedy celé 2 řádky. Podprogram zajišťující zápis do video paměti vyžaduje: název (které pozadí se má vlastně zapsat), řádek (horní číslo řádku) a POSUN. Tato proměnná představuje o kolik pixelů je pozadí posunuto vůči začátku (hrany) video paměti. Je to hodnota, od které se ve video paměti začne zapisovat obrázek pozadí.

Na obr. 4.2 je zobrazen příklad vykreslení pozadí představující strop jeskyně. Pozadí je orámováno fialovou barvou.

4.4.2 Nepřítel



Obr. 4.3 - Typy nepřátel ve hře Spaceman

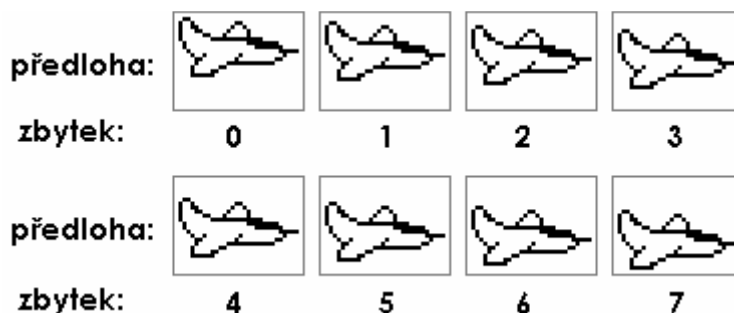
Mezi nepřátele se řadí malé černé letadlo (16x8 bodů, viz obr. 4.3a), asteroidy a meteority (16x16 bodů, viz obr. 4.3b, c) a budovy s krápníky (16x32, viz obr. 4.3d, e, f). Vzhledem k rozdílné velikosti jednotlivých nepřátel se používá pro každého jiný podprogram na zápis do video paměti. Rozdíl mezi nimi je pouze ve velikosti vykreslovaného objektu. Na obr. 4.2 jsou nepřátelé zakresleni v hnědém rámečku. V každém levelu je 20 nepřátelských objektů jednoho typu.

Po zapnutí konzoly je v SRAM paměti vytvořeno 80 (20 meteority, 40 budovy a krápníky, 20 letadla) struktur „parametry“. Před začátkem samotné úrovně jsou

z programové paměti zkopírovány informace o nepřítelích do těchto struktur, jsou zapisovány položky lev_count a Yradek, Xpoz se nastaví na hodnotu „0“. Jedna samotná struktura tedy představuje jednoho nepřítele. Struktura obsahuje 3 položky: lev_count, Yradek, Xpoz. Lev_count je informace o tom kdy má být nepřítel „vyvolán“, jedná se o porovnávací hodnotu k počítadlu COUNT_LEVEL, pokud se tedy hodnota COUNT_LEVEL rovná hodnotě lev_count je v samotné struktuře změněna položka Xpoz na hodnotu „160“. Je to z toho důvodu, že podprogram provádějící posun nepřátelských objektů snižuje ve všech strukturách hodnoty v položce Xpoz (na obr. 4.2 to jsou X1, X2, X3). Obsahuje-li struktura v položce Xpoz hodnotu „0“ není tento objekt vykreslován.

4.4.3 Stíhačka

Je to jediný objekt u kterého není výška definována v řádcích, ale v bodech, aby bylo zajištěno plynulé posunování nahoru a dolů. Tyto souřadnice jsou uloženy v proměnné RAKETA. Toho je docíleno výběrem 1 z 8 předloh. Hodnota y-ové osy se podělí celočíselně 8, výsledek představuje pozici nejvyššího řádku pro objekt letadla, poté se provede dělení y-ové osy MOD 8 (výsledkem je zbytek po dělení 8) a podle tohoto čísla se vybere předloha. Na obr. 4.4 jsou ukázány předlohy.



Obr. 4.4 - Předlohy letadel

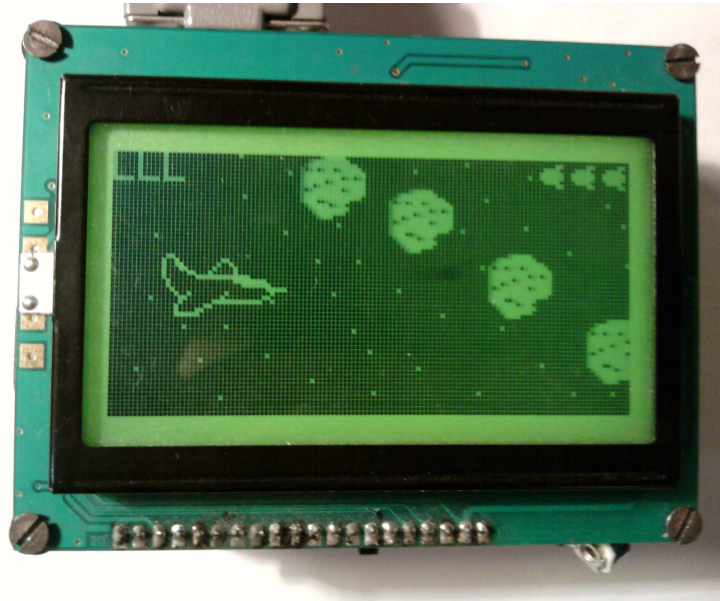
Po výběru předlohy se tento obrázek do video paměti připiše pomocí logického součtu OR, tzn. že se zapíše všechny černé body. Potom se do video paměti připiše předloha stejné hodnoty, ale obrázek se připiše pomocí logického součinu AND, kdy se zapíše bílé body. Tyto „negativy“ jsou na obr. 4.5.



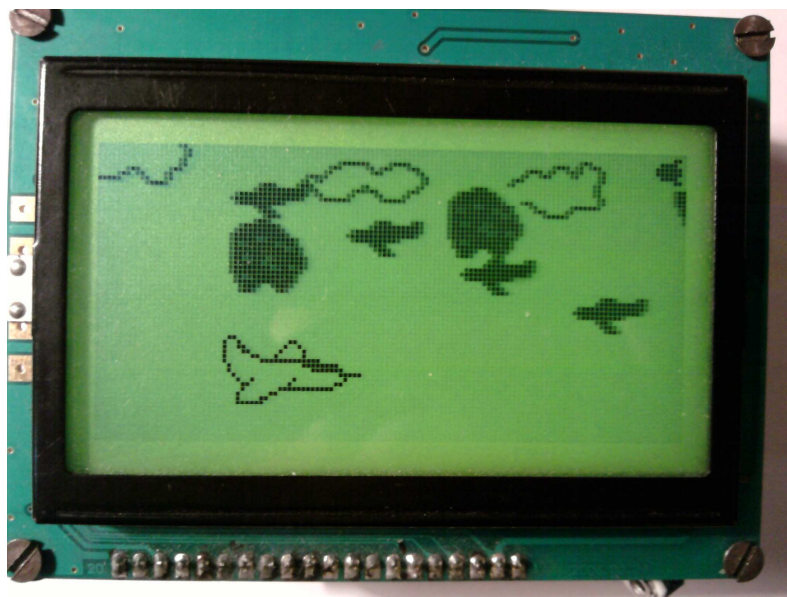
Obr. 4.5 - Předlohy letadel pro zápis "bílých" bodů

4.4.4 Kulky

Po startu je vytvořena matice o velikosti 20x2, představující maximálně 20 vystřelených kulek, nichž každá nese informaci o X ose a Y ose. Při vystřelení se zkopírují souřadnice stíhačky zvýšené o délku stíhačky a polohu hlavně kulometu do matice na první pozici jejíž x-ová osa obsahuje hodnotu „0“. Posun kulky se provádí přičtením 1 k x-ové ose, dosáhne-li hodnoty „156“ (konec obrazovky pro kulky) je u kulky vynulována x-ová osa a může se na tuto pozici uložit nová vystřelená kulka.



Obr. 4.6 - Ukázka hry Space Invaders - úroveň 1



Obr. 4.7 - Ukázka hry Space Invaders - úroveň 2

5 ZÁVĚR

Cílem práce bylo vytvořit herní konzolu. Na tuto konzolu byly vytvořeny aplikace Hledání min a Tetris, psané v JSA, a aplikace Spaceman v C. Všechny tři aplikace byly dlouho testovány a jsou plně funkční. Veškeré zdrojové kódy jsem psal samostatně. Přehled a srovnání velikosti programu a využití paměti najdete v tab. 5.1, EEPROM je u všech programů zaplněna stejným množstvím dat - obsahuje totiž znakovou sadu pro displej.

Tab. 5.1 - Porovnání velikosti programu

	hra		
	Hledání min	Tetris	Spaceman (bez optimalizace)
počet řádků v JSA	3500	8000	12000
paměť programu	6760 B	13176 B	26536 B
	20,60%	40,20%	81,00%
SRAM paměť	41 B	1104 B	1406 B
	2%	53,90%	68,70%

Samotná konzola je doplněna o napájení z baterií. Jsou zde použity nabíjecí baterie o kapacitě 2700mA/hodin. V tab. 5.2 je uvedena spotřeba proudu při běhu dané aplikace. Měření proudu proběhlo ve dvou cyklech (nahrání aplikace, změření, nahrání druhé aplikace, změření, atd.). Při použití zmíněných baterií je tedy možné hrát až 10hodin. Spotřeba proudu se při stisku tlačítek zvyšovala. Nejmenší spotřeba je během pauzy kdy je vše vypnuté, testuje se zda je stisknuta tlačítka. Největší spotřeba je při zapnutém zvukovém doprovodu u aplikace Tetris.

Tab. 5.2 - Přehled spotřeby jednotlivých aplikací

Událost	Hra						
	Spaceman		Tetris			Hledání min	
				bez zvuku	se zvukem		
	U [V]	I [mA]	U [V]	I [mA]	I [mA]	U [V]	I [mA]
Úvodní obrazovka	4,85	220	4,85	240	240	4,85	280
Během hry bez stisku tlačítek		220		245	250		280
Během hry se stiskem tlačítek		240		250	320		300
Pauza		220		220	220		280
Znělka		-		-	360		-

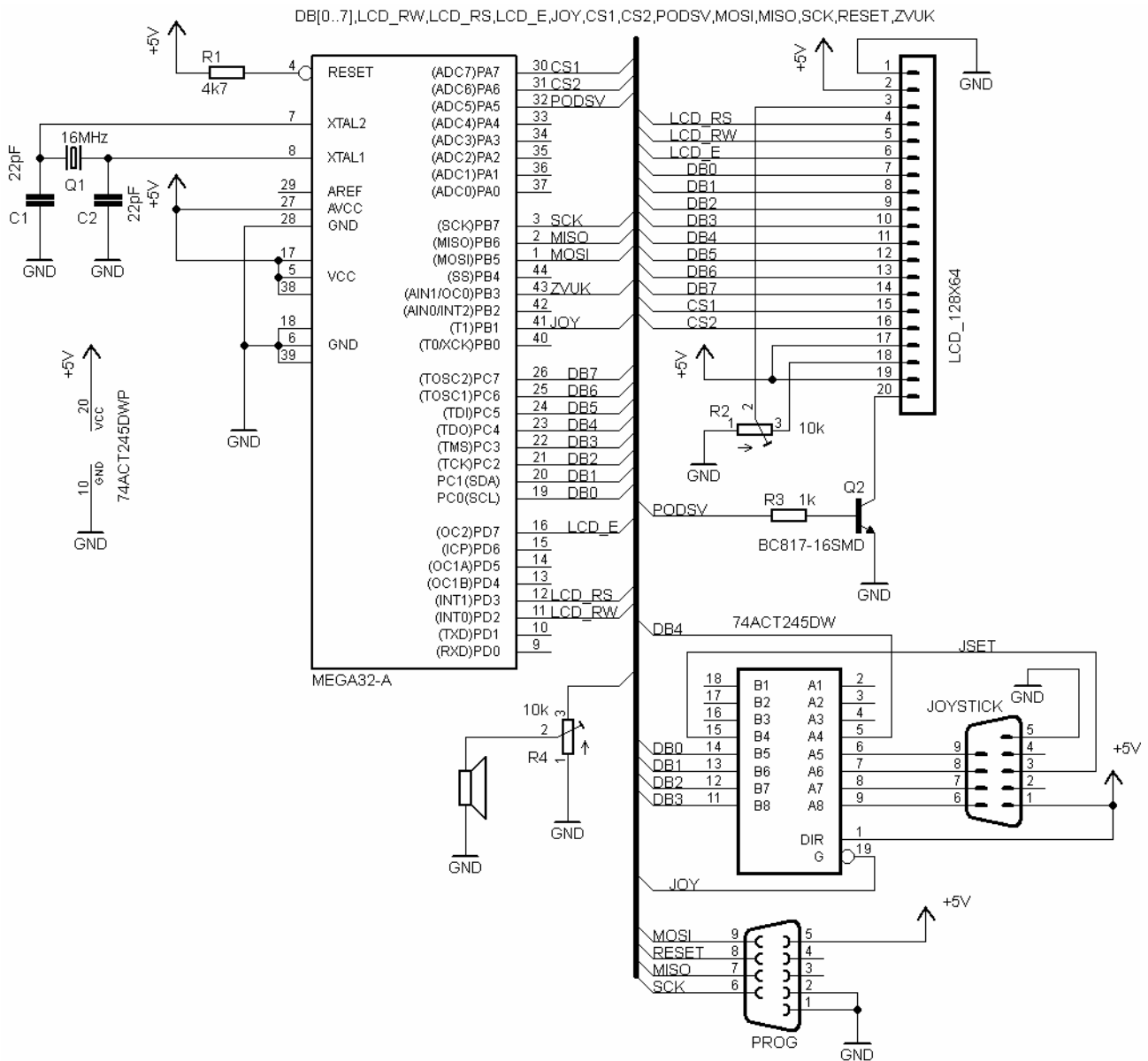
LITERATURA

- [1] Claudio Lanconelli. PonyProg2000. [online]. 1997 - [cit. 15. dubna 2011]. Dostupné na WWW: <http://www.lancos.com/prog.html>
- [2] Atmel Corporation. AVR 8-Bit RISC. [online]. 2011 - [cit. 15. dubna 2011]. Dostupné na WWW: <http://www.atmel.com/products/avr/>
- [3] Stanislav Mašláň. Ovládání grafických LCD modulů s řadičem KS0108 (S6B0108). [online]. 2011 - [cit. 15. dubna 2011]. Dostupné na WWW: <http://elektronika.kvalitne.cz/ATMEL/necoteorie/LCDmatKS0108.html>
- [4] Wikipedia. Tetris. [online]. 2011 - [cit. 15. dubna 2011]. Dostupné na WWW: <http://cs.wikipedia.org/wiki/Tetris>
- [5] Wikipedia. Hledání min. [online]. 2011 - [cit. 15. dubna 2011]. Dostupné na WWW: http://cs.wikipedia.org/wiki/Hledání_min
- [6] Pavel Tišnovský. Herní konzole Uzebox s osmibitovým mikrořadičem ATmega644. [online]. 2011 - [cit. 15. dubna 2011]. Dostupné na WWW: <http://www.root.cz/clanky/herni-konzole-uzebox-s-osmibitovym-mikroradicem-atmega644/>
- [7] Pavel Tišnovský. Herní konzole Uzebox (2) a nástroje pro programování mikrořadičů Atmel AVR. [online]. 2011 - [cit. 15. dubna 2011]. Dostupné na WWW: <http://www.root.cz/clanky/herni-konzole-uzebox-2-a-nastroje-pro-programovani-mikroradicu-atmel-avr/>
- [8] Pavel Tišnovský. Rozhraní MIDI na osobních počítačích. [online]. 2011 - [cit. 15. dubna 2011]. Dostupné na WWW: <http://www.root.cz/clanky/rozhrani-midi-na-osobnich-pocitacich/>
- [9] pianoFORTE.cz. Sedm a půl oktávy. [online]. 2011 - [cit. 15. dubna 2011]. Dostupné na WWW: <http://pianoforte.cz/cs/pevne-zaklady/sedm-a-pul-oktavy>

SEZNAM PŘÍLOH

A	Obvodové zapojení herní konzole	32
B	Návrh desky plošného spoje	33
	B.1 Deska plošného spoje - top (strana součástek)	33
	B.2 Deska plošného spoje - bottom (strana spojů)	34
C	Seznam součástek	35

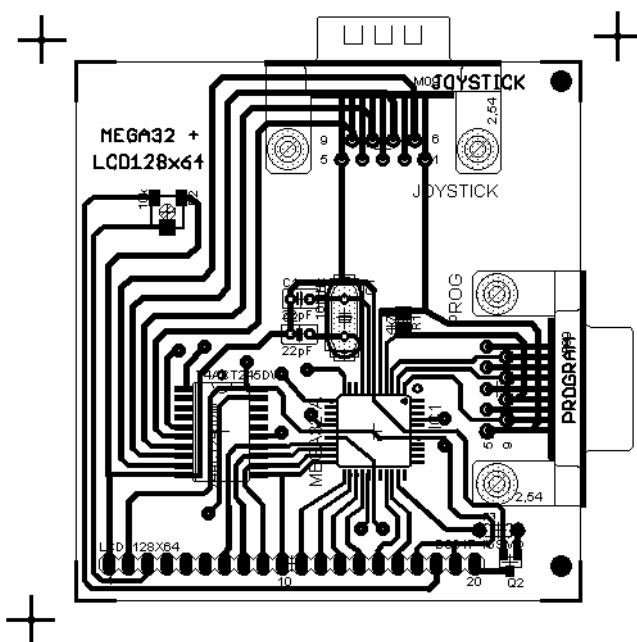
A OBVODOVÉ ZAPOJENÍ HERNÍ KONZOLY



Obvodové zapojení celé herní konzoly

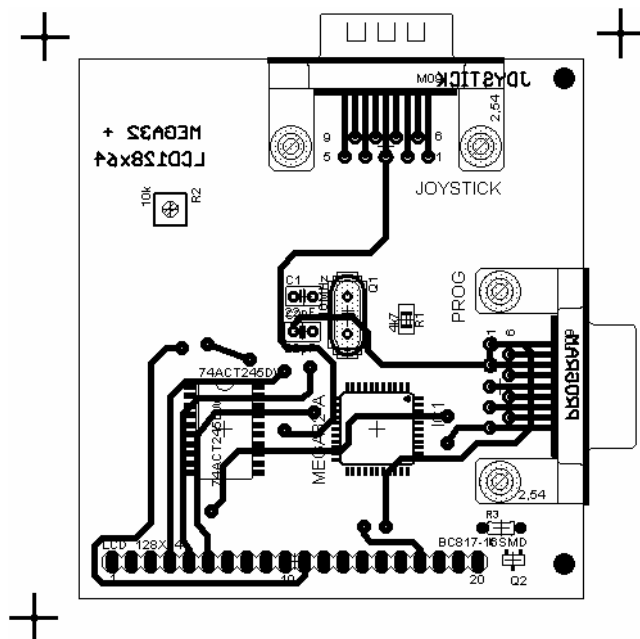
B NÁVRH DESKY PLOŠNÉHO SPOJE

B.1 Deska plošného spoje - top (strana součástek)



Rozměry desky 67 x 72 [mm], měřítko M1:1

B.2 Deska plošného spoje - bottom (strana spojů)



Měřítko M1:1

C SEZNAM SOUČÁSTEK

Označení	Hodnota	Pouzdro	Popis
C1	22p	2,5x5	Keramický kondenzátor
C2	22p	2,5x5	Keramický kondenzátor
R1	4,7k	M0805	Drátový rezistor
R2	10k	TRIM3314	Trimr
R3	1k	0204/5	Drátový rezistor
R4	10k	TRIMR3314	Trimr
Q1	16MHz	HC49/C	Krystal
Q2	BC817C-SMD	SOT23	Univerzální tranzistor NPN
IC1	MEGA32-A	TQFP	Mikrokontrolér
IC2	74ACT245-SMD	SMD	Budič sběrnice