

8Bit Producer

Dokumentace projektu pro předmět Mikroprocesorové a vestavěné systémy

Martin Knapovský
VUT FIT Brno
xknapo02@stud.fit.vutbr.cz

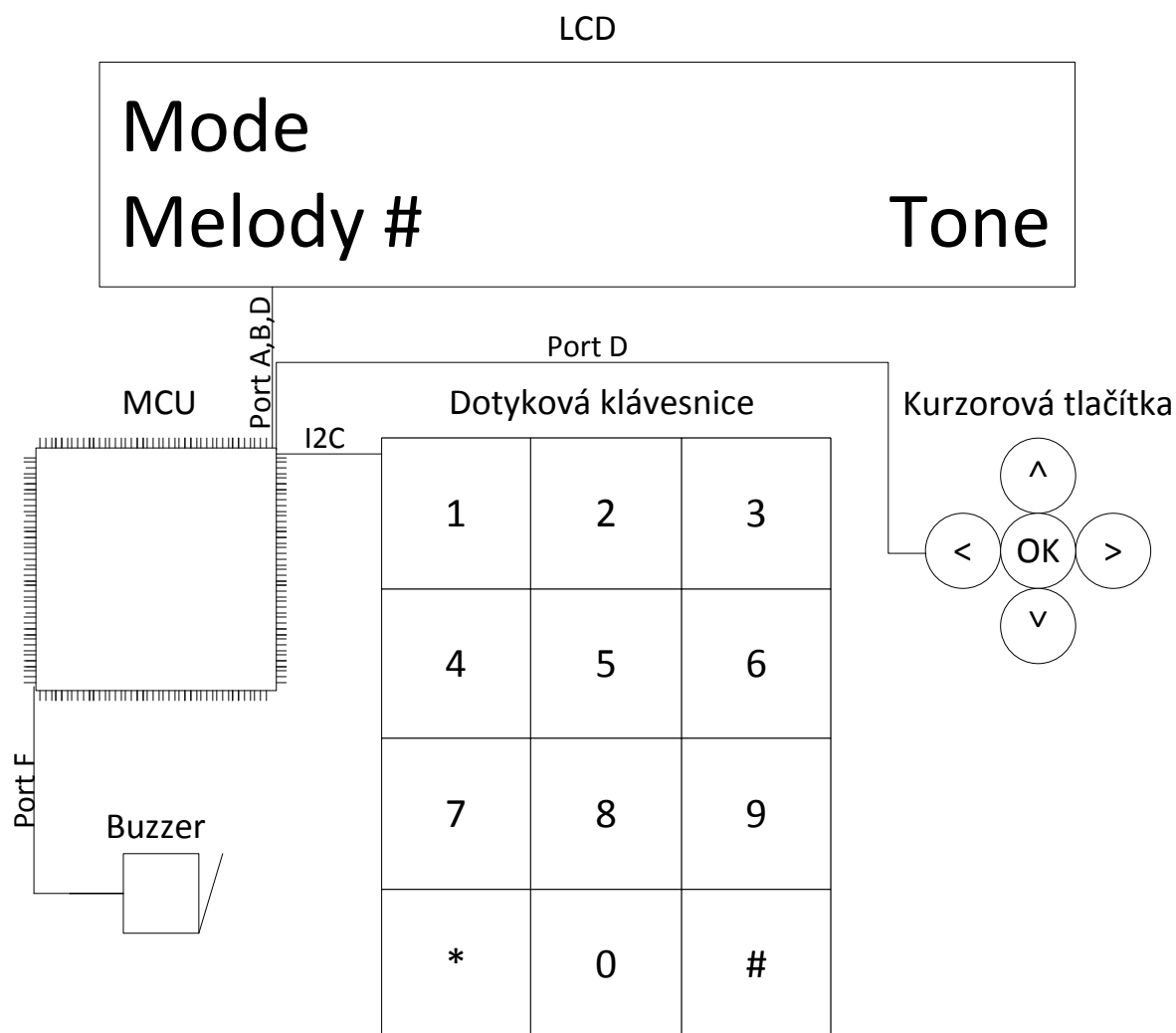
Obsah

Úvod	3
Schéma prvků využitých programem	4
Popis funkčnosti programu	5
Režimy programu	6
Piano	6
Nahrávání	6
Přehrávání	7
Zadávání tónů	7
Popis implementace	8
Inicializace	8
Inicializace paměťového úložiště (seznamu melodií)	8
Programová smyčka	8
Obsluha přerušení kurzorových tlačítek	9
Obsluha přerušení od modulu <i>MPR121</i> (Dotykových tlačítek)	9
Obsluha přerušení od modulu <i>RTC</i>	10
Generování tónu a jeho vizualizace	10
Analýza paměťových nároků	11
Nastavení programu	11
Použité zdroje	11
Metriky Projektu	12

Úvod

Cílem projektu bylo implementovat jednoduchový záznamník a přehrávač melodií s vnitřní pamětí a vizualizací přehrávaného tónu. Projekt byl realizován na vývojovém přípravku s *MCU MC9S08JM60* od firmy *Freescale*. Osobním cílem bylo využít většinu dostupných periférií MCU a vývojového kitu tak, aby byl program co možná nejméně náročný na výpočetní výkon a paměť.

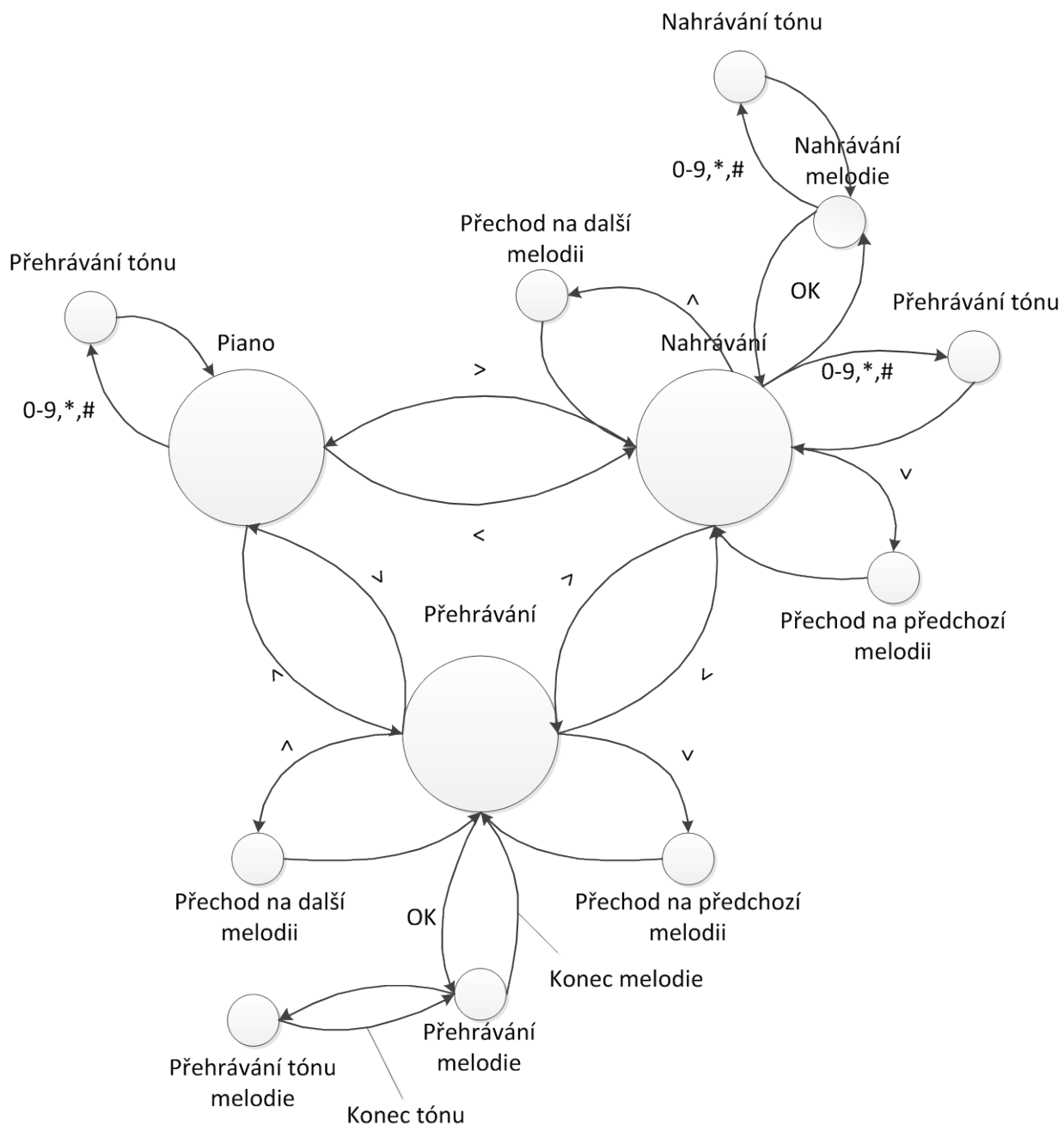
Schéma prvků využitých programem



Obr. 1 – Ilustrativní schéma periférií využitých programem a způsob jejich připojení na MCU

- LCD - *EA DOGM162*
- Řadič dotykové klávesnice - *MPR121QR2*
- Buzzer - *PS1740P02*
- MCU - *MC9S08JM60*

Popis funkčnosti programu



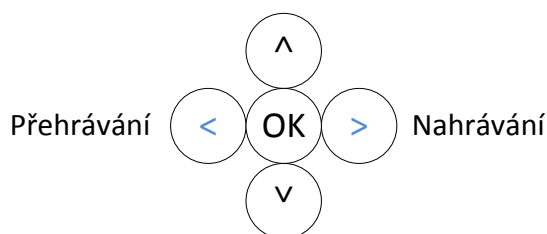
Obr 2. Diagram funkčnosti programu

Režimy programu

Podle zadání projektu má program pracovat ve 3 různých režimech. Mezi režimy je možné přepínat pomocí kurzorových tlačítek. Funkční tlačítka daného módu jsou indikována podsvícením ve formě modrých LED diod. Následuje popis jednotlivých režimů.

Piano

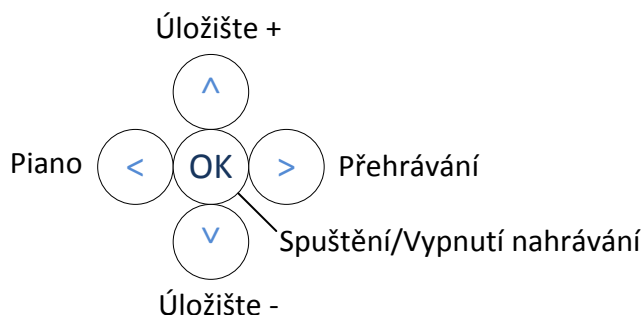
V prvním režimu, který je označen jako „*Piano mode*“, program přijímá informace od dotykové klávesnice a na základě stlačeného tlačítka přehrává tón a mění podsvícení LCD displaye. Pro přechod do následujícího, nebo předchozího režimu je možné využít kurzorových tlačítek.



Obr 3. Piano mode

Nahrávání

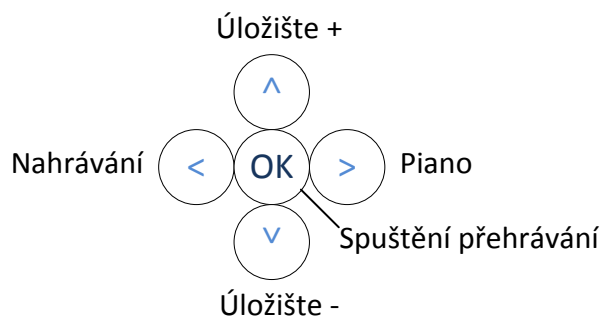
„*Recording mode*“ pracuje podobně jako „*Piano mode*“, avšak umožňuje uživateli vybrat kurzorovými tlačítky číslo úložiště, do kterého může zaznamenat posloupnost tónů (melodii) pomocí dotykové klávesnice. Tlačítko OK spouští nahrávání, které je indikováno změnou podsvícení OK tlačítka z blikání na svícení. Nahrávání je ukončeno dalším stiskem tlačítka OK, jehož podsvícení se opět vrátí do stavu blikání. Tlačítka na změnu melodie a módu jsou v průběhu nahrávání blokována. Pokud chce uživatel nahrávat do úložiště, kde je již dříve nahraná melodie, pak je původní melodie odstaněna a přepsána novou. Při nahrávání nejsou ukládány délky tónů a jejich odstupy. Pro přechod do jiného režimu je opět využito kurzorových tlačítek



Obr 4. Mód nahrávání

Přehrávání

„*Playing mode*“ přehrává předem zvolenou melodii z paměťového úložiště. Tóny jsou odděleny konstantním intervalem. Pro výběr melodie a přechod do jiného režimu slouží opět kurzorová tlačítka. Pro spuštění přehrávání je použito tlačítko OK a jako indikace přehrávání je využito změny podsvícení OK tlačítka z blikání na jeho zhasnutí (pro odlišení s módem nahrávání). Při přehrávání melodie je dotyková klávesnice blokována.



Obr 5. Mód přehrávání

Zadávání tónů

Jak již bylo několikrát uvedeno, tóny jsou zadávány pomocí dotykové klávesnice. Přehrávané tóny jsou ve frekvenčním rozsahu 1047 – 1976 Hz, což odpovídá třetí oktávě (C3 – H3).

Dotyková klávesnice

C	C#	D
D#	E	F
F#	G	G#
A	A#	H

Obr 6. Rozložení stupnice na dotykové klávesnici

Po zadání tónu je jeho název vypsán na display jak je znázorněno na Obr. 1. Tón je vizualizován dočasnou změnou podsvícení displaye, které je nastaveno tak, aby na sebe navazovalo – červená pro tón C až po světle modrou pro tón H, kde tóny mezi nimi jsou vizualizovány lineární interpolací zabarvení těchto dvou tónů. Tóny je možné zadávat bez jakéhokoliv omezení hned za sebou, avšak jejich délka je programově omezena a není možné bez opětovného stisknutí hrát tón delší jak 0,5s.

Popis implementace

K implementaci bylo využito prostředí *CodeWarrior 6.3* a programovací jazyk C, který byl upřednostněn před Assemblerem díky tomu, že je více abstraktní a také kvůli mým dlouholetým zkušenostem s tímto jazykem.

Program není rozdělen do modulů, avšak obsahuje logické části, které je vhodné od sebe v popisu odlišit. Předem upozorňuji na stručnost popisu vzhledem k velmi kvalitním komentářům samotného zdrojového kódu, kde je možno o implementaci zjistit podrobnější informace, které by dokumentaci zbytečně znepřehledňovaly.

Inicializace

Základní inicializace *MCU* vypíná modul Watchdog, nastavuje *SPI* a základní podsvícení *LCD* displaye na modrou barvu (0x0C). Dále je inicializován modul *RTC*, který slouží k změně podsvícení tlačítka OK v režimu nahrávání a přehrávání. *LCD* je inicializováno sekvencí příkazů (lze je nalézt v dokumentaci k použitému displayi) a následným vypsáním informací o programu. Po ukončení animace na kurzorových tlačítkách je započato nastavení *I2C* sběrnice a modulu *MPR121*, který je na sběrnici adresován hodnotou 0x5A. Po odpovědi od modulu *MPR121* jsou adresovány jeho registry, kde jsou vhodně nastaveny parametry dotykové klávesnice, které byli získány z dokumentace k tomuto modulu. Především je nutné nastavit správnou citlivost tlačítek, jelikož ty mohou být nastaveny i jako proximity senzor. Přerušeni od modulu *MPR121* je generováno jako *IRQ EXT*, které je nutné pro jeho přijetí povolit zápisem do registru *IRQSC*. Na konci inicializačního bloku je nastaven 4. bit registru *PTFDD*, čímž je 4. pin portu D nastaven jako výstupní a je tak možné generovat tón na *buzzeru*.

Inicializace paměťového úložiště (seznamu melodií)

Vzhledem k tomu, že předem nevíme, kolik tónu uživatel nahraje, je potřeba vyhradit jednu pozici pro ukončovací znak (popř. ke každé melodii ukládat její velikost, což je stejně paměťově náročné jako ukládání ukončovacího znaku). Při inicializaci se tedy prochází seznamem melodií a na začátek každé melodie je zápsán tento ukončovací znak.

Programová smyčka

Po inicializaci se provádění programu nachází v nekonečné programové smyčce, kde se pomocí obslužných rutin reaguje na přerušení, která mohou přijít od kurzorových tlačítek, dotykové klávesnice, nebo modulu *RTC*, který byl nastaven tak, aby generoval přerušení každé 0,5 sekundy.

Obsluha přerušení kurzorových tlačítek

U kurzorových tlačítek může docházet k zámkům, což je řešeno tím, že se po přijetí přerušení uloží současný stav tlačítek (bitové pole, kde hodnota 1 vyjadřuje stav „stisknuto“) a čeká se po určitý počet cyklů *MCU* (v programu 10000). Po uplynutí tohoto intervalu se opět zkontroluje stav tlačítek a následná obsluha je provedena pouze tehdy, když jsou oba stavy stejné.

Funkce tlačítek se liší dle aktuálního režimu programu (piano, nahrávání, přehrávání). Popis funkcí tlačítek v daných režimech byl již uveden, proto ho není potřeba zde znovu uvádět. Pro přechod na předchozí, nebo následující režim je volána funkce *change_state_to(STAV)*, která mění globální proměnou *state* uchovávající informaci o aktuálním režimu programu, přepíná podsvětlení pomocí funkce *led_change()* a vypíše aktuální informace na *LCD* display (režim na první řádek a číslo zvolené melodie na řádek druhý). Podobně jako pro změnu režimu, tak i pro změnu melodie je volána funkce *shift_melody(direction)*, která podle zadaného parametru mění melodie směrem nahoru, nebo dolů (číslo právě zvolené melodie je uloženo v globální proměnné *melody*) a opět vypíše aktuální informace na *LCD*. Pro ošetření přechodu na melodii, která není v paměti alokována je použita operace *modulo MAX_MELODY_COUNT* (maximální počet melodií).

Pokud je program v režimu nahrávání a je stisknuto tlačítko *OK*, pak jsou vypnuta ostatní kurzorová tlačítka, aby uživatel nemohl v průběhu nahrávání změnit režim na jiný, nebo změnit melodii, do které se právě nahrává. Po stisknutí tohoto tlačítka je nastavena indikace nahrávání (globální proměnná *recording*), která rozhoduje o blikání/neblikání *OK* tlačítka a o tom, zda má být tón přijatý z klávesnice zaznamenán, či nikoliv. Melodie je smazána (na začátek melodie je zapsána ukončovací hodnota *MELODY_END*) a je očekáváno zadávání tónů uživatelem. Ty jsou následně zpracovány v obsluze přerušení od *MPR121 (IRQ EXT)*.

Podobným způsobem probíhá i zpracování přerušení od *OK* tlačítka v režimu přehrávání. Zde je volána funkce, která nastaví příznak přehrávání (*playing* - opět pro změnu způsobu podsvětlení *OK* tlačítka), vypne tlačítka pro přechod na jiný režim či melodii a následně pomocí cyklu prochází aktuálně zvolenou melodii, dokud nenarazí na hodnotu reprezentující konec melodie. Interval mezi přehrávanými tóny je pevně nastaven na 60000 cyklů *MCU*. Po ukončení přehrávání jsou opět povolena tlačítka, která uživateli umožní změnit melodii, nebo režim. Také je odstraněn příznak *playing*, který způsobí rozblikání tlačítka *OK*. Způsob přehrávání tónu a jeho generování bude popsán dále.

Obsluha přerušení od modulu *MPR121* (Dotykových tlačítek)

Vzhledem k tomu, že na 1 stisk tlačítka jsou generována 2 přerušení (stisk a puštění tlačítka), je nutné zjišťovat, o které přerušení se jedná. To je poměrně snadné, jelikož při puštění tlačítka již nejsou na sběrnici *I2C* data, která by reprezentovala některé ze stlačených tlačítek. Opět se zde rozlišuje mezi režimy nahrávání, přehrávání a piano a jsou prováděny jim odpovídající činnosti. V režimu piano je volána funkce *play_tone()*, která zprostředkovává přehrávání tonu. Stejná funkce je volána i režimu nahrávání s tím rozdílem, že přímo za ní následuje volání funkce *rec_tone()*, která podle toho, zda je nahrávání zapnuto (*recording* = 1) zapíše aktuální tón na konec zvoleného paměťového uložení (na

konec právě zvolené melodie). Zápis se provádí vyhledáním pozice ukončovací hodnoty v melodii a následným zapsáním tónu na tutu pozici a ukončení melodie posunem ukončovací hodnoty na pozici následující. Pokud by chtěl uživatel nahrát melodii delší než je její maximální délka, pak funkce aktuální tón nezapiše a ponechá ukončovací hodnotu na pozici *MAX_MELODY_SIZE + 1* (za posledním tónem melodie).

Obsluha přerušení od modulu *RTC*

Jak již bylo dříve uvedeno, modul *RTC* slouží v programu k indikaci stavu nahrání/přehrávání, kde pokud jsou dané příznaky nastaveny, pak na ně obsluha přerušení od modulu *RTC* (přerušení nastaveno na 0,5s) reaguje dle následující tabulky:

Režim	playing = 1	recording = 1	Playing = 0	Recording = 0
Přehrávání	Nesvíí	Nedefinováno	Blikání	Nedefinováno
Nahrávání	Nedefinováno	Svíí	Nedefinováno	Blikání

Dále je modul *RTC* použit ke generování tónu, které je popsáno v následující podkapitole.

Generování tónu a jeho vizualizace

Generování tónu zprostředkovává funkce *play_tone(tone)*, která jako parameter přijímá typ přehrávaného tónu, podle kterého nastaví do globální proměnné *tone_freq* jeho frekvenci, vypíše jeho název do pravého rohu spodního řádku displaye a zavolá funkci *timer2_init()*. Tato funkce se stará o syntézu tónu pomocí časovače *TPM2*, jehož výstup 0 se zapisuje přímo do *data registru portu F* a určuje tak zda má *buzzer* produkovat tón (1), či nikoliv (0). Časovač je nastaven do režimu *Edge-aligned PWM: high-true-pulses*, což umožňuje generovat tón pomocí pulsně šířkové modulace. Jako zdroj hodin časovače slouží *BUS Rate Clock MCU*, který je dělen hodnotou 128. Nastavení Modulo registru *TPM2MOD* a hodnoty kanálu *TPM2C0V* pak probíhá podle následujících vztahů.

$$TPM2MOD = MCU_FREQ/PRESCALER/tone_freq * 256$$

$$TPM2C0V = TPM2MOD / 2,$$

kde *MCU_FREQ* je frekvence na které běží mikrokontrolér (nastaveno na 8000000), *PRESCALER* je předdělič této frekvence (128), *tone_freq* je frekvence tónu, který chceme generovat. Hodnota kanálu *TPM2C0V* je polovina hodnoty *TPM2MOD* – generovaný signál tónu totiž vždy v polovině svého intervalu mění hodnotu z 1 na 0. Funkce *timer2_init()* po inicializaci časovače resetuje čítač modulu *RTC* na hodnotu 0 a ve chvíli, kdy od časovače přijde přerušení (0,5s) je přehrávání tónu ukončeno. *MCU* tak nečeká v žádné smyčce a program není tak náročný.

Pro vizualizaci volá funkce *timer2_init()* funkci sobě velmi podobnou. Jedná se o funkci *timer_init()*, která však oproti *timer2_init()* zprostředkovává pulsně šířkovou modulaci na portu F, na který je na pinech 0, 1, 2, 3 a 5 připojen *LCD display* (resp. jeho podvětlení). Každý z výstupů časovače *TPM1*, který tato funkce ovládá je nastaven do stejného režimu jako časovač *TPM2*, do jeho modulo registru *TPM1MOD* je

vložena hodnota `LCD_COL_MAX` (250) a hodnoty registrů `TPM1C3V`, `TPM1C4V` a `TPM1C5V` jsou nastaveny dle odpovídajících hodnot požadované barvy. Změna podsvícení na původní hodnotu je opět provedena v obsluze přerušení od modulu `RTC`.

Analýza paměťových nároků

Přestože má mikrokontrolér `MC9S08JM60` velké množství paměti (60KB), je program implementován tak, aby v paměti nezabíral zbytečné místo. Kde to bylo možné, byly voleny datové typy `unsigned char`, avšak bylo nutné deklarovat i několik proměnných typu `unsigned int` (pro uložení frekvence tónu a pro pomocné proměnné které byly použity ve for cyklech). Nejvíce paměti zabraly řetězce pro výpis informací na display – zde je stále prostor pro optimalizaci.

Program zabere v paměti `ROM` 6072 bajtů a v paměti `RAM` 206 bajtů při nastavení velikosti úložiště na 5 melodií o deseti tónech.

Nastavení programu

Ve zdrojovém souboru `main.c` jsou vypsány použitelné frekvence, které je možné nahradit za aktuálně definované, aby mohl program generovat jinou oktávu, či jakékoliv jiné kombinace tónu. Je však potřeba dávat si pozor na příliš nízké frekvence, jelikož v závislosti na kvalitě a typu `buzzeru` se může generovaný tón stát neslyšitelným.

Je také možné přizpůsobit syntézu tónu jiné frekvenci `MCU`, avšak není zaručeno, že při změně frekvence `MCU` bude funkční komunikace na sběrnici `I2C`.

Z uživatelského hlediska je nejpodstatnějším nastavením velikost seznamu melodií a maximální délka melodie, kterou lze uložit. Toto nastavení se provádí změnou hodnot v definicích `MAX_MELODY_SIZE` a `MAX_MELODY_COUNT`. Je nutno brát ohled na paměťové nároky programu při nastavení vysokých hodnot těchto definic.

Použité zdroje

- Přednášky IMP
- Cvičení IMP
- [Datasheet MC9S08JM60](#)
- [Datasheet DOG Series 3.3V](#)
- [Datasheet PS1740P02-TDK](#)
- [Datasheet MPR121](#)
- Rady pana Ing. Josefa Stranadela, Ph.D. a pana Ing. Václava Šimka

Metriky Projektu

1860 řádků kódu