

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



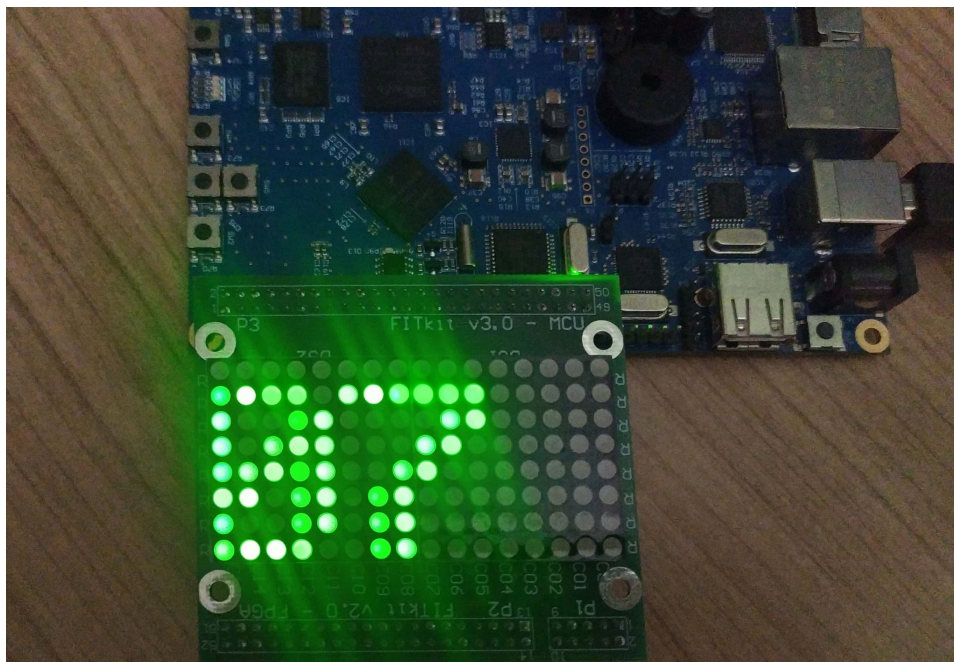
Mikroprocesorové a vestavěné systémy
Projekt - Světelná tabule

19. prosince 2020

Ladislav Ondris (xondri07)

1 Úvod

Úkolem bylo vytvořit vestavěnou aplikaci, která bude na maticovém displeji zobrazovat text, který se posouvá zprava doleva. Tento maticový displej je složený ze dvou LED displejů KWM-30881AGB. Je ovládán mikrokontrolérem **Kinetis K60**^[1] s jádrem ARM Cortex-M4. K tomuto účelu je využit výukový kit Minerva^[3]. Na obrázku 1 je vidět funkční řešení.



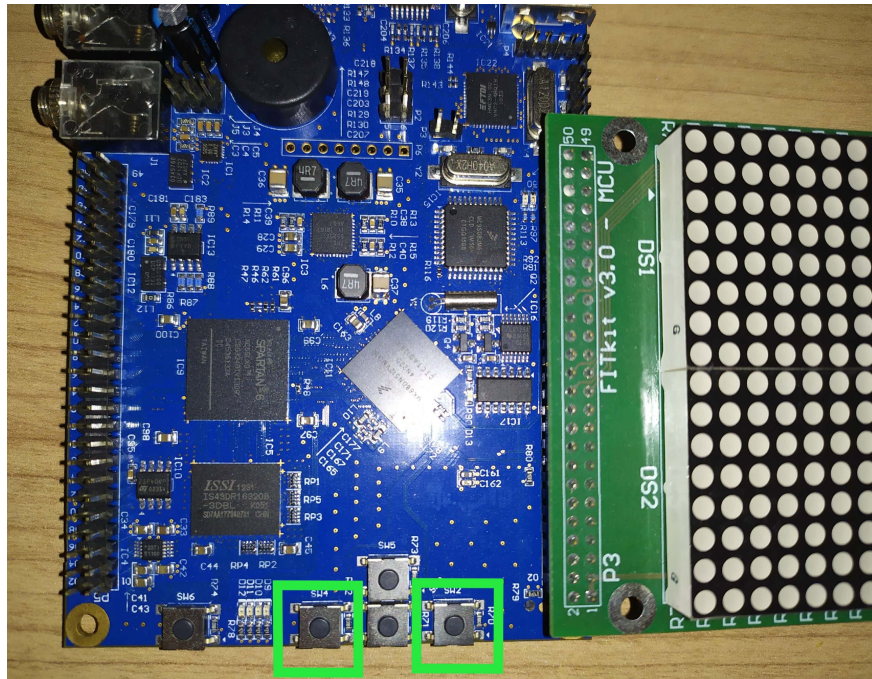
Obrázek 1: Na maticovém displeji se vyobrazuje text, který se posouvá vlevo. Fotoaparát zapříčiňuje zdvojení písmen na fotce, což je způsobeno posunem textu vlevo během pořízení fotky.

2 Ovládání

Aplikaci začne zobrazovat první předvolený text ihned po zapojení ke zdroji napájení. Zobrazovaný text lze měnit tlačítky a komunikací přes sériové asynchronní rozhraní.

2.1 Ovládání tlačítka

Aplikace podporuje tlačítka **SW4** a **SW2** jak je vidět na obrázku 2. Tlačítko SW4 přepíná předdefinované zprávy jedním směrem (předchozí zpráva) a tlačítko SW2 směrem opačným (následující zpráva). Jsou předdefinované 3 zprávy: 'xondri07', '42' a '69'.



Obrázek 2: Na obrázku jsou zeleně vyznačeny tlačítka SW4 (vlevo) a SW2 (vpravo), kterými lze přepínat mezi předdefinovanými zprávami. Na pravé straně obrázku je maticový displej, na který se vypisuje text.

2.2 Ovládání sériovou komunikací

Aplikace po spuštění vypíše nápovědu a vyzve uživatele k napsání textu, který se má zobrazit, viz obrázek 3. Po potvrzení se text začne zobrazovat na displeji. Aplikace opět vyzve uživatele k zadání textu. Uživatel má tímto způsobem možnost kdykoliv měnit zobrazovaný text.

K sériové lince se dá připojit například programem Putty, kde je třeba nastavit příslušný port, baud rate 115 200, 8 datových bitů, 1 stop bit a bez parity.

```

Application has started...
-----
Text on the display can be changed by pressing either the SW2, or the SW4 button
.
There are three predefined messages: 'xondri07', '42', and '69'.
You can set a custom message by sending it via this interface.
Don't worry, it can be changed any time while the application is running.
-----
Type text to display: 123456-*/*;.
Type text to display: ahojAhoj
Type text to display: 

```

Obrázek 3: Displej programu Putty připojeného k sériovému rozhraní, pomocí něhož lze nastavovat zobrazované zprávy. Po spuštění aplikace se vypíší informace a dále pak vyzývá uživatele k zadání textu k zobrazení.

Podporovanými znaky jsou všechny tisknutelné znaky z ASCII tabulky. Tedy znaky s ordinální hodnotou 32 až 127. Hodnoty menší než 32 se zobrazí jako mezera (32), hodnoty větší jako DEL (127). Z toho vyplývá, že není podporované mazání znaku například klávesou backspace. UTF znaky nejsou podporovány, protože jejich velikost je větší než jeden bajt.

3 Popis řešení

K vývoji bylo využito prostředí Kinetis Design Studio^[2].

3.1 Konfigurace MCU

Frekvence hodinového signálu pro periferie je nastavena na 48 MHz nastavením hodnoty DMX32 a DRST_DRS v registru MCG_C4.

3.2 Periferie

K dosažení požadované funkčnosti byly využity periferie PIT, UART, Tlačítka SW2 a SW4, a maticový displej.

Pro každou byly perifii byly zapnuty hodnoty zápisem do některého z registrů SIM->SCGCx. Pro každou periferii je potřeba nastavit alternativy pro příslušné piny. V případě GPIO se nastaví na vstupní, či výstupní a určí se další konfigurace.

3.2.1 PIT

PIT je časovač, který je využíván pro vygenerování přerušení k vykreslení sloupce na displeji a dále k posunování textu doleva.

Obnovací frekvence displeje

Pro tento účel je nastavena hodnota LDVAL na hodnotu 15 999, tedy každých 16 tisíc cyklů se vygeneruje přerušení od tohoto časovače. Aby časovač začal znovu pracovat, musí se v obslužné metodě vyčištěním příznaku o přerušení. Na nastavené frekvenci 48 MHz potrvá uvedený počet cyklů 0.33 ms (frekvence 3 KHz) do následujícího přerušení. Při každém přerušení se spustí obslužná funkce, která rozvíjí následující sloupec displeje. Displej má 16 sloupců, takže každých 5.33 ms (187.5 Hz) se obnoví celý displej.

Posunutí textu vlevo

Hodnota LDVAL je nastavena na hodnotu 9 599 999, což udává frekvenci posunutí displeje 5 Hz, tedy každých 200 ms. Při každém přerušení se posune celý text o jeden sloupec doleva. Za sekundu se tedy posune text o 5 sloupců.

3.2.2 UART

UART je modul, kterým je zajišťovaná komunikace přes sériové asynchronní komunikační rozhraní. V tomto projektu je využíván UART5 na portu E, piny s indexem 8 a 9. Piny je třeba propojit s UARTem nastavením multiplexoru pinu do varianty 3.

V registru UART5->C2 se nastaví TE a RE, čímž se aktivuje vysílač i přijímač. V UART5->BDL se nastaví baud rate na 115200 s 1 stop bitem. Registr UART5->C4 se nastaví na hodnotu 15, což vyjadřuje hodnotu oversamplingu, a to o jednu vyšší, tedy 16. Nakonec se pro UART5 aktivuje přerušení.

3.2.3 Tlačítka

Tlačítka **SW2** a **SW4** zajišťují přepínání mezi předdefinovanými texty. Odpovídající piny, na která jsou tlačítka připojena, je třeba nastavit jako výstupní, s přerušením na padající hranu a definovat pull-up. Na portu, ke kterému jsou připojena tlačítka, bylo třeba aktivovat přerušení. Při zachycení přerušení na daném portu se zjistí, které tlačítko bylo stisknuto a podle toho změnit zobrazovaný text na displeji.

3.2.4 Maticový displej

Na maticový displej se vyobrazuje text. Text se vyobrazuje vykreslováním jednotlivých sloupců. Sloupce jsou vykreslovány postupně určitou frekvencí, což

způsobí dojem, že svítí všechny sloupce současně.

3.3 Vnitřní reprezentace displeje

Obsah displeje je vnitřně uložen ve struktuře, viz 3.3, uchovávající aktuální obsah displeje, text, který se má zobrazovat na displeji a číselné indexy určující následující sloupec k zobrazení v aktuálním nebo následujícím znaku.

```
typedef struct display {  
    uint8_t matrix[16];  
    char *text;  
    int char_index;  
    int col_index;  
} display_t;
```

Obrázek 4: Struktura uchovávající informace o displeji a právě na něm zobrazeném textu.

3.4 Reprezentace znaků

Všechny podporované znaky jsou uloženy v dvojrozměrném poli uint8_t hodnot o velikosti 96×5 . Tedy každý znak je složen z 5 sloupců, kde každý sloupec je reprezentován datovým typem uint8_t. Každý bit značí, zda je konkrétní LED dioda zapnutá, či vypnutá.

4 Autoevaluace

Zadání doporučuje, aby si autor sám ohodnotil vypracování projektu dle hodnotícího klíče¹, což může pomoci při jeho vyhodnocování.

4.1 E - Přístup

1/1

Na projektu jsem z nadšení pro danou oblast začal pracovat včas a dokončil implementační část téměř dva měsíce před odevzdáním. Zadání nabízí dvě volby zobrazení textu - změnou tlačítka, nebo vstupem přes sériové rozhraní. Vypracoval jsem obě možnosti, abych si práci s MCU vyzkoušel co nejvíce.

¹http://www.fit.vutbr.cz/~strnadel/public/imp/-prj_eval_style.pdf

4.2 F - Funkčnost

5/5

Řešení funguje dle očekávání a splňuje všechny body v zadání. Text se vykresluje zprava doleva bez chyb, které by mohly vzniknout například při rozsvicování políček špatným selektováním signálů, nebo blikající displej, kterou by způsobila nízká frekvence vykreslování sloupců.

4.3 Q - Kvalita

Kvalita je rozdělena do třech podkategorií.

4.3.1 Uživatelská přívětivost

1/1

Ovládání aplikace je popsáno v dokumentaci. Taktéž při spuštění je přes UART uživateli vytištěna nápověda, která vysvětluje jakým způsobem může měnit text na obrazovce. Aplikace podporuje znaky ASCII tabulky 32-127, což výrazně zlepšuje uživatelskou přívětivost s porovnáním řešení, které podporuje například pouze velká písmena.

4.3.2 Přehlednost

1/1

Dekompozice, jak bude možné vidět v další sekci, z velké části napomáhá přehlednosti. Důležité části jsou komentovány a související bloky kódu extrahovány do funkcí.

4.3.3 Dekompozice

1/1

Zdrojové kódy rozdělují vnitřní logiku aplikace od přístupu k prvkům na úrovni hardwaru. Vnitřní reprezentace obsahu displeje je oddělen v samostatném souboru *display.h* a *display.c*. Taktéž struktura a kód pro změnu textu na displeji je oddělen v samostatném souboru *messages.h* a *messages.c*. Funkce jsou ve všech souborech smysluplné a srozumitelné.

4.4 P - Presentace

1/1

Bylo vytvořeno video prezentující funkčnost řešení dle požadavků zadání, které je dostupné na následující odkazu: <https://nextcloud.fit.vutbr.cz/s/xxni9NYNpsx2C2w>.

4.5 D - Dokumentace

Hodnotí se tři podsložky.

4.5.1 Úvod do problému

1/1

Úvod do problému je vysvětlen v úvodu a v rámci kapitoly popisu řešení.

4.5.2 Popis řešení

2/2

Všechny části řešení jsou v dokumentaci popsány.

4.5.3 Zhodnocení

1/1

Vše potřebné bylo uvedeno.

5 Závěr

Byla vytvořena vestavěná aplikace pro mikrokontrolér Kinetis K60. Svou funkčnost splňuje nad rámec zadání. Nicméně jako další rozšíření by se daly podporovat UTF znaky nebo alespoň podporovat speciální ASCII znaky s hodnotou menší než 32 (např. backspace). Řešení demonstruje práci s displejem, tlačítky, časovačem a sériovým komunikačním rozhraním.

Literatura

- [1] K60 Sub-Family Reference Manual. [online], rev. 2. červen 2012, [vid. 2020-10-26]. Dostupné z: <https://www.nxp.com/docs/en/reference-manual/K60P144M100SF2V2RM.pdf>
- [2] Kinetis Design Studio v2.0.0 User Guide. [online], rev. listopad 2014, [vid. 2020-10-26]. Dostupné z: <https://www.nxp.com/docs/en/user-guide/KDS200UG.pdf>
- [3] Šimek, V.: Schéma obvodového zapojení výukového kitu Minerva. [online], 16. března 2013, [vid. 2020-10-26]. Dostupné z: <https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIMP-IT%2Fexcs%2FFITkit3-schema.pdf&cid=13997>