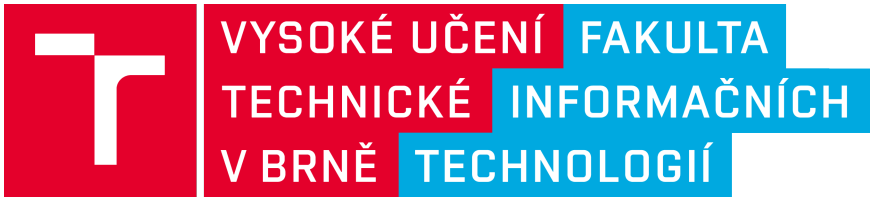


IMP Projekt
Projektová dokumentace
Světelná tabule

Veronika Nevařilová (xnevar00)

13. prosince 2023

Obsah

1 Úvod **2**

2 Úvod do problému **2**
2.1 Využité HW a SW prostředky 2
2.2 Konfigurační a stavové registry 2
2.3 Použité principy a metody 3

3 Řešení **3**

4 Shrnutí **4**
4.1 Shrnutí výsledků 5

1 Úvod

V této dokumentaci jsou popsány návrh a implementace světelné tabule pomocí platformy FITkit 3 a maticového displeje.

2 Úvod do problému

Vytvořený program umožňuje na displeji zobrazovat zprávy, které se posouvají zprava doleva a jejichž obsah může uživatel nadefinovat.

2.1 Využité HW a SW prostředky

V projektu byly využity následující HW prostředky:

- **Mikrokontroler MK60DZ10**, který je základním prvkem pro obsluhu desky.
- **Maticový displej**: Vizualizuje zprávy na základě nastavení.
- **Tlačítka (SW4 a SW5)**: Slouží ke změně zobrazované zprávy.
- **UART komunikace (UART5)**: Přijímá zprávy z počítače.
- **Časovač (PIT)**: Periodicky mění zobrazovanou zprávu.
- **USB B konektor** pro komunikaci mezi počítačem a FITkitem

Dále bylo využito některých SW prostředků, zejména:

- **Kinetis Design Studio**, ve kterém celá implementace probíhala
- **Aplikace PuTTY** pro zasílání zpráv FITkitu přes UART počítačem

2.2 Konfigurační a stavové registry

- **GPIO registry pro řízení displeje (PORT A)**: nastaveny pro řízení aktivace sloupců a řádků maticového displeje.
- **GPIO registry pro řízení tlačítek (PORT E)**: Konfigurace pro přerušení od tlačítek.
- **UART registry (PORT E)**: Nastavení pro přijímání zpráv přes UART.
- **PIT registry**: Nastavení pro vyvolání přerušení pravidelně po určitém čase

V programu je využito čtyř knihoven: knihovna k mikrokontroleru `MK60DZ10.h`, pomocí které bylo možné implementovat veškerou komunikaci s mikrokontrolerem a provést nastavení jednotlivých registrů a přerušení, 2 standardní knihovny jazyka C – `string.h` a `stdbool.h` a knihovna `main.h`, ve které jsou zadefinována především písmena ve tvaru pro zobrazení na displeji. .

2.3 Použité principy a metody

Jelikož architektura maticového displeje umožňuje mít zapnutý v jednu chvíli pouze jeden sloupec LED diod, bylo využito tzv. multiplexingu – sloupce jsou postupně za sebou aktivovány a během svícení jednoho sloupce vždy svítí jen zvolené diody na řádcích tohoto sloupce. Jelikož změny vybraných sloupců a tedy i rozsvicování a zhasínání LED diod probíhá velkou rychlostí, lidské oko nezaregistruje, že nesvítí najednou celý dísplej, a displej se mu jeví, jako by svítily sloupce všechny.

Pro změnu zprávy na základě stisknutí tlačítka byla použita přerušení. V momentě, kdy uživatel stiskne tlačítko `SW4` nebo `SW5`, je vyvoláno přerušení, jehož obsluha změní zobrazovanou zprávu. Přerušení je také implementováno u časovače.

Část zprávy, která se má zobrazit na displeji, je nastavena pomocí okna o šířce 16 bodů, což je šířka displeje.

3 Řešení

Program byl vytvořen z kostry vzorového projektu dostupného v zadání.

Na začátku vykonávání programu se provedou funkce `MCUInit` a `SystemConfig`, které nastaví hodiny, používané porty, UART5 pro umožnění příjmu znaků, povolí přerušení a inicializuje časovač.

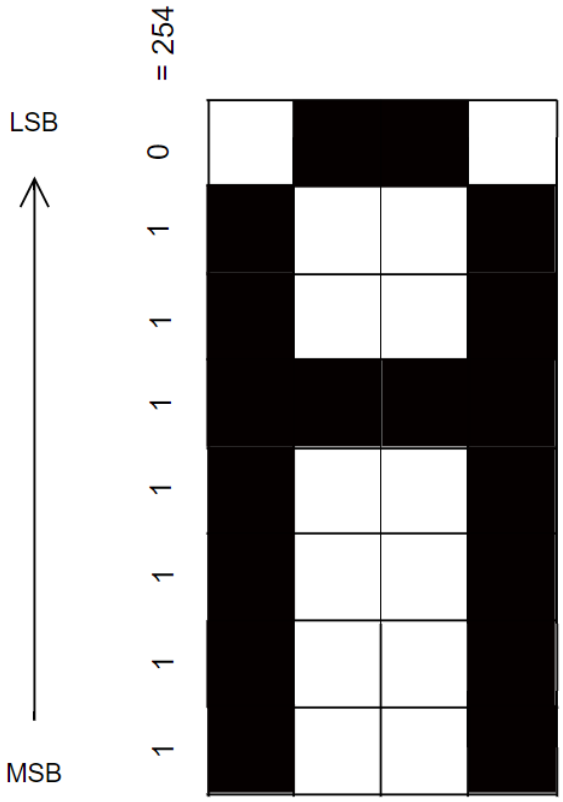
Dále je nutno zvolit, v jakém formátu ukládat jednotlivá písmena. Výška každého písmena je stejná, ale šířka každého písmena se mění. Například písmeno I má šířku 1, zatímco písmeno X například 5. Proto byla vytvořeno konstantní pole znaků `characters`, kde každý znak je reprezentován strukturou `FontCharacter`. O každém znaku je ukládána informace o šířce a samotná bodová reprezentace znaku je v poli osmibitových hodnot (každá hodnota pro jeden sloupec, jelikož výška maticového displeje je 8 bodů). Pole reprezentující znak A je tedy například `{254,9,9,254}` a na maticovém displeji se bude zobrazovat takto:

Pro převod zprávy, jakožto řetězce znaků, do výsledného pole `to_display` se volá funkce `process_message`. Ta prochází zprávu po znaku a do výsledného pole za sebe ukládá tyto osmibitové hodnoty polí, které jsou uloženy ve struktuře daného písmena. Mezi každými dvěma písmeny je pro odlišení písmen mezera s šířkou 1 bod a nejširší písmena mají šířku 5 bodů. Maximální délka zprávy proto byla stanovena na 30 znaků a maximální velikost pole je tedy $30 * (5 + 1)$ hodnot.

Pro samotné multiplexování sloupců při výpisu zprávy na maticovém displeji byla převzata funkce `column_select` ze vzorového projektu dodaného k zadání tohoto projektu, která převede číslo v dekadickém tvaru do pole binárních hodnot a dle toho nastaví hodnoty ve výstupním registru portu A. Podobný způsob byl také využit k implementaci funkce `row_select`, která vybírá, které body daného sloupce budou svítit právě dle hodnot v poli výsledného řetězce.

Aby měl výpis efekt posunu zprávy zprava doleva, bylo vytvořeno okno, jehož počáteční index znamená index v poli výsledné zprávy, od kterého má být výpis prováděn. Od tohoto indexu je zobrazováno 16 hodnot (16 bodů je šířka maticového displeje) výsledného pole – ovšem jen pokud je samozřejmě daný index platným indexem pole. Na začátku je nastaven na hodnotu -16 a postupně se zvyšuje. Toto zajistí, že se na začátku zpráva vynoří zprava a končí odjetím doleva. Po zmizení zprávy vlevo se index okna nastaví znovu na počáteční hodnotu a výpis zprávy začíná znovu.

Pokud je stisknuto tlačítko `SW4` nebo `SW5` nebo je zaslán přes UART znak z počítače, nastaví se proměnná `change` na hodnotu `True` spolu s proměnnou `activeMessage`, do které se uloží hodnota z výčtového typu, která říká, jaká zpráva se má zobrazit. Jakmile smyčka programu, ve které se zobrazuje zpráva, zjistí, že se zpráva má změnit, provede potřebné kroky a zobrazování se nastaví na výchozí hodnoty, aby se opět zpráva vynořila zleva.



Obrázek 1: Zobrazení písmena A na displeji na základě hodnot v jeho poli.

V programu je také nastaven časovač, který po 15 sekundách nečinnosti (žádné zmáčknutí tlačítka, nebyl přijat znak z počítače přes UART) změní zobrazovanou zprávu. Měnění zprávy je nastaveno tak, aby se provádělo pouze pokud je zrovna zobrazována nějaká ze dvou přednastavených zpráv, mezi kterými lze také manuálně měnit pomocí výše zmíněných tlačítek. Při zobrazování zprávy od uživatele přijaté prostřednictvím UART se přerušení časovače ignorují.

4 Shrnutí

Program se podařilo implementovat do finálního stavu. Ověřování funkčnosti bylo prováděno přímo na FITkitu. Bylo natočeno video pro demonstraci funkčního řešení[1].

Bylo dbáno na to, aby se eliminoval počet způsobů, jakými by mohl uživatel funkčnost řešení rozbít. Maximální délka zprávy je tedy 30 znaků a pokud se uživatel snaží přes aplikaci PuTTy zaslat více znaků, jsou tyto znaky na straně FITkitu jednoduše ignorovány a na displeji je zobrazováno pouze prvních 30 znaků.

Pokud uživatel zašle FITkitu pro zobrazení znak, který není implementován v poli struktur písmen `FontCharacter`, je tento znak přeskočen a zobrazují se pouze znaky, které program zná. Tyto znaky zahrnují písmena a-z a A-Z (malá písmena jsou na displeji zobrazována jako velká), čísla 0-9, mezeru a vykřičník.

[1]Video na Google Drive: `https://drive.google.com/file/d/1X6t2p_RM-tsJPUbGOkGLWJPeHuTy1Plk/view?usp=sharing`

4.1 Shrnutí výsledků

Efektivní řízení displeje

Projekt implementuje efektivní řízení maticového displeje pomocí multiplexingu, což umožňuje plynulé a dynamické zobrazování obsahu.

Změna obsahu přes UART

Komunikace přes UART umožňuje uživateli dynamicky měnit obsah zobrazované zprávy. Projekt reaguje na příchozí zprávy a aktualizuje zobrazený obsah.

Ovládání pomocí tlačítek

Tlačítka na desce jsou úspěšně integrována do projektu a lze pomocí nich provést rychlou změnu zobrazované zprávy.

Časovač

Projekt využívá časovač (PIT) pro periodickou změnu zprávy na displeji. Tato funkce byla úspěšně implementována a přispívá k dynamice zobrazení.

Správa chyb

Projekt obsahuje mechanismy pro vyhýbání se chybám, které jsou zmíněny výše. To pomáhá zajistit stabilitu a spolehlivost aplikace. Pro minimalizaci chyb bylo při implementaci využíváno především referenčního manuálu mikrokontroleru[2], vzorového projektu dostupného v zadání projektu a kódů ze cvičení předmětu IMP.

[2]K60 Sub-Family Reference Manual: `https://www.fit.vutbr.cz/~simekv/K60P144M100SF2RM.pdf`