



# Mikroprocesorové a vestavěné systémy

## Světelná tabule

# 1 Úvod

Cílem projektu je vytvoření vestavné aplikace, pomocí které je možné zobrazit krátkou zprávu formou tzv. běžícího textu. Text je vykreslen postupně ve směru zprava doleva po jednotlivých sloupcích na přípravku zvaném *"Maticový displej"* (2ks maticového LED displeje s typovým označením **KWM-30881AGB**). Projekt je řešen na platformě **FITkit 3**<sup>[2]</sup> obsahující mikrokontrolér **Kinetis K60**<sup>[1]</sup> (s jádrem ARM Cortex-M4).

Aplikace je implementována v **jazyce C** v prostředí Kinetis Design Studio (KDS). Odevzdaný archiv obsahuje složku **src**, kterou je možné rovnou importovat do KDS 3.0, a poté ověřit správnou funkcionalitu aplikace.

## 2 Funkce a ovládání

Po spuštění aplikace se postupně zobrazí výchozí text: **"Hello World"**. Předdefinované zprávy je možné pomocí tlačítek **SW5** a **SW3** přepínat za běhu. Zprávy jsou postupně vykreslovány zprava doleva po příslušných znacích. Znaky jsou interně reprezentovány na displeji  $4 \times 8$  a odděleny mezerou o délce jednoho sloupce. Po vykreslení všech znaků zprávy jsou všechny diody displeje postupně deaktivovány. Jakmile dojde k deaktivaci všech diod, celý proces se cyklicky opakuje a zpráva je znova vykreslena.

### 2.1 Předdefinované zprávy

Aplikace obsahuje 6 předdefinovaných zpráv:

- "HELLO WORLD"
- "FIT VUT"
- "IMP"
- "Alexander Polok"
- "UNKNOWN #\$(&^^@"
- "SPACES   "

Je možné vykreslit **alfabetické znaky**, **mezeru**, **tečku**, **vykřičník** a **otazník**. V případě zadání znaku, který nenáleží do výše definované množiny, je vykreslená neznáma hodnota interpretována aktivací všech diod příslušné mřížky  $4 \times 8$  (Demonstrováno na předposlední zprávě). Alfabetické znaky jsou vždy zobrazeny jako **velká písmena**, tedy zpráva *"Alexander Polok"* je zobrazena jako *"ALEXANDER POLOK"*.

Textový řetězec pro vykreslení je možné vybrat pomocí tlačítek **SW5**, které slouží pro výběr předchozí zprávy v cyklickém bufferu a **SW3**, které vybere následující element bufferu. Po zmáčknutí jednoho z výše zmíněných tlačítek je aktuální zpráva na displeji smazána a vykreslení řetězce probíhá od začátku.

### 2.2 Video ukázka

Pro demonstraci funkcionality aplikace byla vytvořena video ukázka, na které je demonstrováno standardní použití aplikace. Video je dostupné na adrese <https://nextcloud.fit.vutbr.cz/s/HRNfPnWEEJcYgKW>.

### 3 Implementace

Jak již zadání zmiňuje zobrazení hodnot je provedeno pomocí multiplexingu. Jelikož je nutné hodnoty periodicky aktualizovat, byl využit modul **PIT** (Periodic Interrupt Timer). Frekvence hodinového signálu byla nastavena na hodnotu **48kHz**. Pro plynulé obnovování neviditelné holým okem sloupců maticového displeje byla zvolena perioda 12 000 taktů, tedy odpovídající frekvence **4kHz**. K posunu zprávy dochází z frekvencí **4Hz**, tedy co 12 000 000 taktů.

V funkci **SystemConfig** jsou konfigurovány použité moduly a rozhraní podle referenčního manuálu[1]. Nejprve dochází k nastavení hodinového generátoru **MCG** (Multipurpose Clock Generator), registru děliče systémových hodin **SIM\_CLKDIV1** (System Clock Divider) a vypnutí systémového *watchdogu*. Následně jsou povoleny systémové hodiny pro příslušné porty a aktivují se **vstupní/výstupní piny**. Obnova displeje je zaručena pomocí konfigurace **PIT** (Periodic Interrupt Timer) v periodách určených výše.

#### 3.1 Obnovování obsahu displeje

Modul PIT vyvolává periodická přerušení, které jsou obsloužena v příslušných funkcích a tímto způsobem dochází k periodickému obnovování obsahu displeje. Hodnoty displeje a zprávy jsou definovány jako globální struktury **display\_struct \*display** a **message\_struct \*message**, k jejichž inicializaci dochází mezi konfigurací periferií a spuštěním záchyty obsluhy přerušení. Takto je možná změna hodnot těchto struktur v rámci obsluhy přerušení.

Struktury obsahují hodnoty nezbytné pro aktualizaci obsahu displeje, které eliminují nutnost pro každý znak zprávy vypočítávat jeho hodnotu za běhu. Ze vstupní zprávy je vytvořené pole obsahující **indexy znakové sady** a poté pouze dochází k vyčtení hodnoty pomocí modulace počtu vykreslených sloupců zvolené zprávy.

#### 3.2 Reprezentace znaků na displeji

Znaková sada je definována v souboru **display.h** jako 2D pole **uint8\_t** o velikosti 31x4. Obsahuje **31 znaků**. Každý znak je definován pomocí **4 uint8\_t hodnot**, příslušné hodnoty tedy odpovídají tomu, které diody mají být aktivní v daném sloupci. Hodnoty jsou zapsány binárně a tedy po jejich řádkové transformaci, lze vidět jak daný znak bude zobrazen na displeji.

Aby bylo možné postupně aktivovat příslušné řádky daného sloupce dochází k **bitovému součinu** příslušné hodnoty sloupce znaku a masky daného řádku.

Pro představu znak A je reprezentován jako {0b11111110, 0b00011001, 0b00011001, 0b11111110} a po jeho řádkové transformaci na displeji vypadá takto:

```
. .  
. .  
. .  
....  
....  
. .  
. .  
. .
```

#### 3.3 Načtení vstupní zprávy

Vstupní zprávy typu řetězec jsou nejprve přeloženy do podoby, kterou je možné jednoduše zobrazit na displeji. Dochází k výpočtu jejich ascii hodnoty, která je transformována v funkci **get\_char\_set\_index** na odpovídající index výše popsané znakové sady. Tímto procesem se naplní pole **char\_set\_indexes** struktury **message\_struct \*message**, pomocí něhož jsou hodnoty zobrazeny na displeji.

Následně je vypočten počet kroků nutných pro zobrazení celého obsahu zprávy, pomocí kterého dochází k indexaci sloupce znaku.

Po stisku tlačítka **SW3** nebo **SW5** jsou původní hodnoty struktur **display\_struct \*display** a **message\_struct \*message** vynulovány a dochází k výpočtu nových hodnot příslušné zprávy. Aby tento proces nebyl vyvolán vícekrát v době jednoho stisku je nutné zaznamenat zmačknutí tlačítka, i jeho uvolnění.

## 4 Autoevaluace (13.7/14)

### 4.1 Dokumentace - D (4 / 4)

- Úvod do problému (1/1) - dokumentace obsahuje vše nutné, pro to aby čtenář byl schopný porozumět problému a poté aplikaci reimplementovat
- Popis řešení (2/2) - všechny části řešení jsou v dokumentaci popsány
- Zhodnocení (1/1) - video ukázka demonstruje funkcionalitu výše popsaných částí

### 4.2 Přístup - E (0.7/1)

Autor projekt implementoval z měsíčním předstihem, avšak mohl řešení zdokonalit nad rámec zadání (například využitím modulu UART pro načtení zpráv od uživatele).

### 4.3 Funkčnost - F (5/5)

Řešení dovoluje provést všechny úkony sepsané ve vstupních požadavcích, pro implementaci jsou využité dostupné periferie. Při testování nebyly odhaleny nedostatky.

### 4.4 Presentace - P (1/1)

Příložená video ukázka obsahuje krátkou prezentací řešení. Na ukázce je vše potřebné demonstrováno.

### 4.5 Kvalita - Q (3/3)

- Přehlednost (1/1) - zdrojové kódy jsou řádně komentovány, kód je čitelný
- Dekompozice (1/1) - řešení parciálních problému je logicky rozděleno do příslušných funkcí
- Uživatelská přívětivost (1/1) - aplikace je ovládána pomocí dvou tlačítek, vše je řádně zdokumentováno

## 5 Závěr

V rámci řešení projektu vznikla vestavná aplikace demonstrující použití světelné tabule. Aplikace splňuje zadání a je možné ji použít pro zobrazení textových řetězců na maticovém displeji. Znakovou sadu by bylo možné pro další použití dále rozšířit a zároveň implementovat možnost načtení zprávy pomocí sériové komunikace (např. aplikace PUTTY). Světelné tabule je možné pozorovat na každém kroku až už v MHD nebo na všude přítomných bilboardech. Je skvělé mít možnost vyzkoušet si vizualizovat vlastní zprávy.

## Literatura

- [1] K60 Sub-Family Reference Manual. [online], červen 2012, naposledy zobrazeno: 20. prosince 2020. Dostupné z: <https://www.nxp.com/docs/en/reference-manual/K60P144M100SF2V2RM.pdf>
- [2] Šimek, V.: FitKit 3 (Minerva) – schéma zapojení. [online], červenec 2013, naposledy zobrazeno: 20. prosince 2020. Dostupné z: <http://www.fit.vutbr.cz/~simekv/schematics%20-%20FITkit%20v3.0.pdf>