

# Dokumentace k IMP projektu

## Název zadání: Světelná tabule

*Vedoucí projektu: Ing. Václav Šimek*

### **Shrnutí projektu**

Zadáním bylo implementovat výpis různých nápisů na součástku maticový displej připojenou na FITkit 3. Rozsvicování diod na maticovém displeji by mělo být uděláno pomocí multiplexování řádků a sloupců. Mezi nápisy by se mělo dát přepínat pomocí tlačítek na FITkit 3.

### **Přehled použitých technologií.**

Projekt je implementován v jazyce C a byl vyvíjen a testován v prostředí Kinetis Design Studio. Program byl napsán pro desku FITkit 3 a maticový displej typu KWM-30881AGB. Jako vstup uživatele slouží 3 tlačítka na desce FITkit 3 – SW2, SW3, SW4.

Maticový displej se nastavuje nastavením kombinace řádkových vodičů a sloupcových vodičů. Pro rozsvícení určité diody je potřeba nastavit řádkový vodič, na kterém se dioda nachází, na log. 1 a nastavit konkrétní sloupcový vodič na log. 0. K řízení sloupců je použito dekodéru 4-na-16. Sloupec/ce, které tedy mají být zapnuté se nastavují kombinací 1 a 0 na řídicích pinech A0 – A3.

Pro řešení projektu jsem využil časovače PIT a tlačítka SW2, SW3, SW4 a GPIO funkcionality.

### **Řešení**

Při řešení jsem začal stáhnutím a zprovozněním vzorového projektu k otestování vybavení – IMP\_projekt\_had\_tabule\_test.zip. Tento projekt jsem použil jako základ mého projektu.

### **Inicializace**

Nejvíce jsem využil funkce SystemConfig – hlavně nastavení PORTA pinů na zapnutí GPIO funkcionality a nastavení výstupních pinů. Dále jsem přidal zapnutí časovače PIT a nastavení hodnoty tohoto časovače. Také jsem pro časovač povolil přerušení.

### **Obsluha displeje**

Obsluhu displeje jsem vyřešil pomocí multiplexingu. Pro sloupce jsem použil (upravenou) funkci z vzorového projektu – column\_select. Tato funkce přijme číslo mezi 0 a 15, zakóduje ho do 4 bitů a nastaví PDOR registr na PORTA podle této hodnoty.

Pro nastavení řádků jsem použil multiplexing 8 na 32 bitů, implementovaný ve funkci row\_select. Funguje stejně jako column\_select.

Pro tisknutí zpráv jsem si naimplementoval makro pro každé písmeno v (anglické) abecedě. Makro se skládá z více volání funkce `row_select`, `column_select` a `delay`. Makro má argument, který určuje od jakého sloupce se písmeno vykresluje (písmeno se vykresluje zleva doprava).

```
#define A(x)    row_select(255, x); \
               column_select(x); \
               delay(tdelay3, tdelay2); \
               row_select(17, x + 1); \
               column_select(x + 1); \
               delay(tdelay3, tdelay2); \
               row_select(17, x + 2); \
               column_select(x + 2); \
               delay(tdelay3, tdelay2); \
               row_select(255, x + 3); \
               column_select(x + 3); \
               delay(tdelay3, tdelay2);
```

Obrázek 1: Příklad makra pro písmeno A

Pro tisknutí zpráv existují 3 různé funkce. Tyto funkce tisknout “FIT” po stisknutí tlačítka SW2, “VUT” po stisknutí tlačítka SW3 a “IMP” po stisknutí SW4. Pro každou zprávu existuje oddělená funkce. Posouvání zprávy po displeji je řešeno pomocí přerušné generovaného PIT časovačem. Ve while cyklu probíhá vykreslování současné konfigurace displeje dokud nedojde k přerušení, poté se zpráva posune o jeden sloupec a dále se vykresluje pomocí while cyklu. Při vygenerování přerušení se také zkontroluje zda nebylo stisknuté nějaké tlačítko. Pokud ano tak se přepne přehrávání zpráva.

## ***Demonstrace funkčnosti***

[Odkaz na video](#)

## ***Autoevaluace***

F – funkčnost 5/5, řešení splňuje zadání.

Q – kvalita 3/3, pro spuštění/ověření stačí zmáčknout tlačítko. Zdrojový kód je přehledný, jelikož jsou použité makra pro tisknutí jednotlivých písmen tak na úrovni skládání slov není potřeba řešit logiku nastavování sloupců a řádků.

E – přístup 1/1

P – Presentace 1/1

D – dokumentace 3/4