

Vysoké učení technické v Brně  
Fakulta informačních technologií  
IMP - Dokumentace projektu  
Varianta č. 6: Stopky na 7-segmentovém displeji

Jan Demel (xdemel01)

10. Prosince 2019

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
1.1	Zadání . . . . .	2
<b>2</b>	<b>Implementace</b>	<b>3</b>
2.1	Chování 7-segmentového displeje . . . . .	3
2.2	Zapojení displeje . . . . .	3
2.3	Programování aplikace stopky . . . . .	4
2.4	Využití HW prostředky . . . . .	5
<b>3</b>	<b>Ovládání</b>	<b>5</b>
<b>4</b>	<b>Závěr</b>	<b>5</b>

# 1 Úvod

Projekt vznikl jako součást studia na VUT FIT v rámci předmětu IMP - Mikroprocesorové a vestavěné systémy. Je implementován v jazyce C a zabývá se prací se 7-segmentovým displejem pomocí FitKitu3.

## 1.1 Zadání

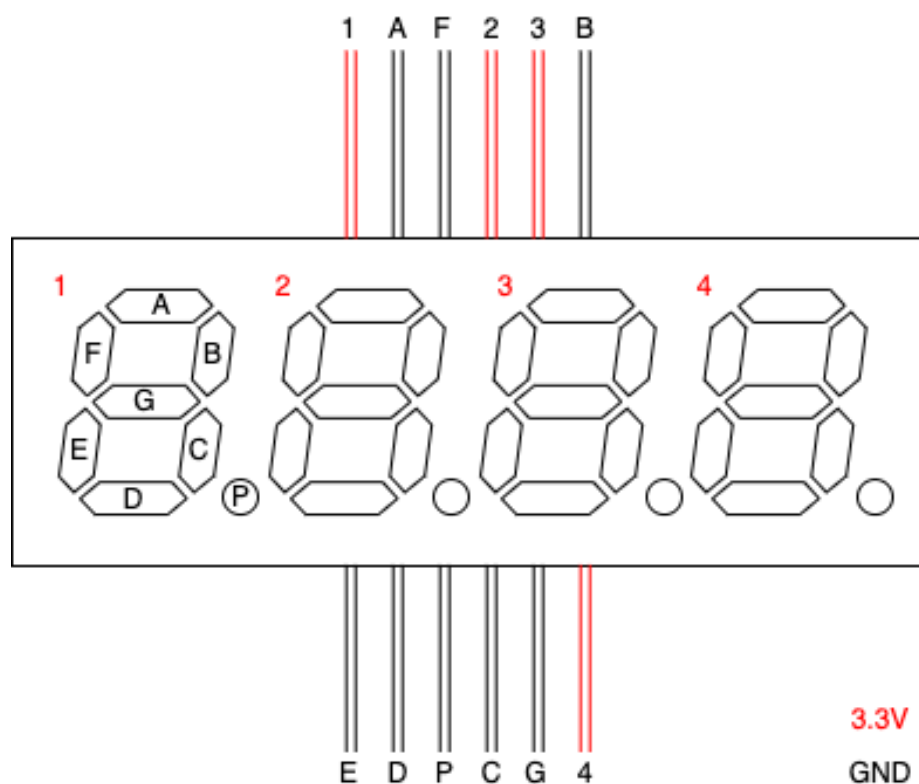
Zadáním bylo navrhnout vestavný SW, který realizuje digitální stopky se zobrazením na 7-segmentovém displeji. Měli jsme použít tlačítka na kitu k vhodnému ovládání a implementovat následující funkcionalitu:

- Rozsah čítání přizpůsobte možnostem displeje (tj. 0-99 sekund, přesnost 0.01 s, překročení max. doby realizujte prostým přetečením)
- Po aktivaci aplikace zobrazte 0.00 s, po spuštění čítání zobrazujte aktuální čas
- Možnost uložení mezičasu, jehož zobrazení bude možno přepínat po zastavení čítání
- Vynulování systému

## 2 Implementace

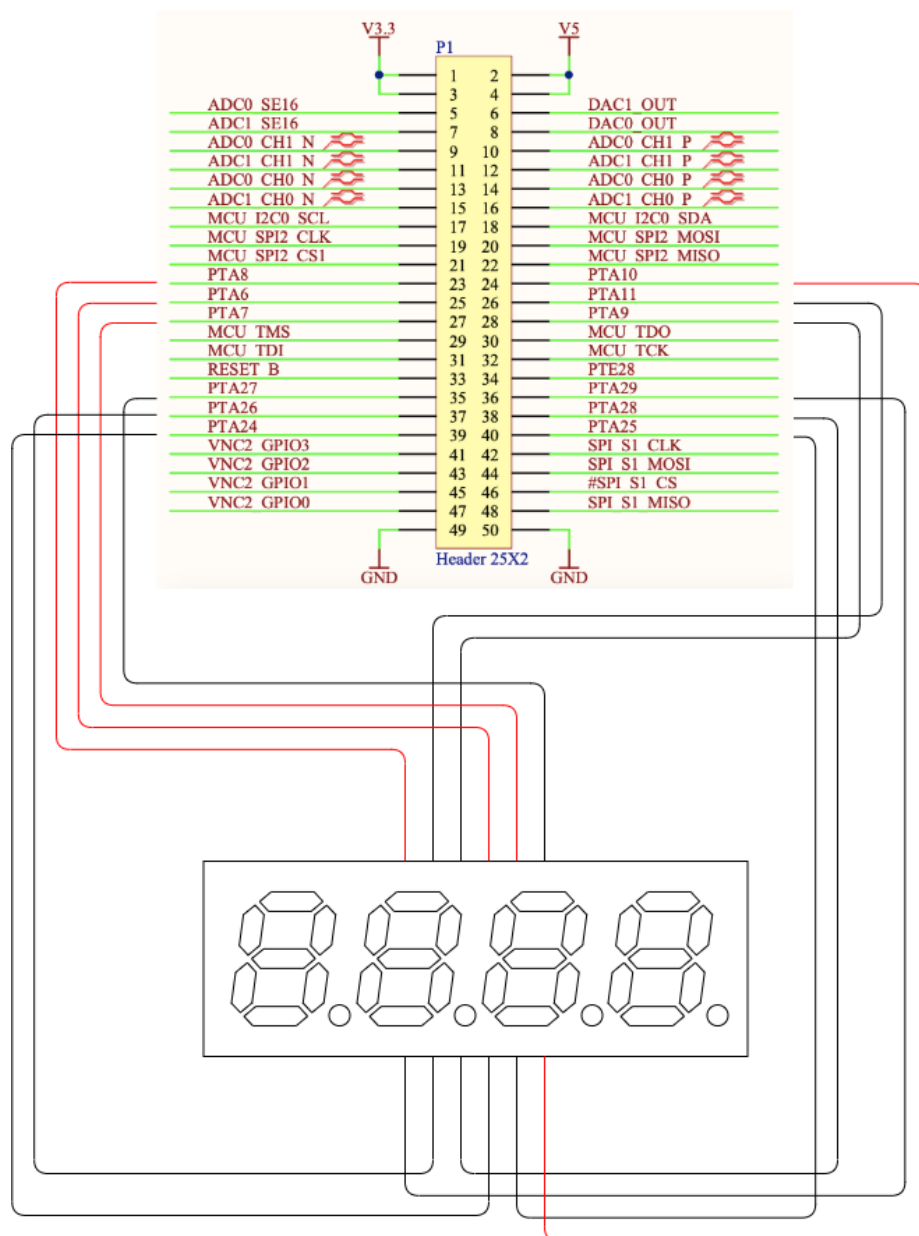
### 2.1 Chování 7-segmentového displeje

Náš 7-segmentový displej má opravdu velice špatnou dokumentaci. Nelze v ní nalézt smysluplné schéma zapojení, tudíž bylo třeba experimentovat. Experimentem jsem přišel na následující chování:



### 2.2 Zapojení displeje

Displej jsem připojil na volné porty PTA (podle platného schématku zapojení přípravku FitKit3) následujícím způsobem:



## 2.3 Programování aplikace stopky

Nejdříve jsem si zadefinoval pole, které obsahuje jednotlivé číslice pro zobrazení na displeji. Následně metodu, která na určitém displeji zobrazí určitou číslici. Inicializoval jsem porty na GPIO a definoval všechny jako výstupní.

Pro ovládání aplikace jsem zvolil tlačítka SW2, SW3, SW4, SW5 na PORTE.

Tyto tlačítka jsem nastavil jako GPIO vstup s pull-up rezistory tak, aby vyvolávaly přerušení. Podprogram pro obsluhu přerušení na základě tlačítka, které přerušení vyvolalo, provede logiku.

Dále bylo třeba mezi jednotlivými displeji rychle přepínat (v jeden konkrétní časový okamžik je možné zobrazit číslici pouze na jednom z displejů). Mezi přepínáním program chvíli čeká, jinak by displeje svítily nepatrně.

Hodnota `current_time`, která je zobrazená na displeji je v programu uložena ve formě struktury `T_TIME`. Ta obsahuje počet vřetin a desítek milisekund.

Intervaly 10ms jsem generoval pomocí čítače `LPTMR0` s nastavením komparačního registru `LPTMR0->CMR = 9`. Při vyvolání přerušení dojde k inkrementaci příslušných hodnot v `current_time`. Nastavení čítače bylo inspirováno kódem z dema.

Pro uchování dílčích časových okamžiků je vytvořeno pole s velikostí 100 záznamů. Tím pádem mé stopky podporují maximálně 100 zaznamenaných časových okamžiků.

## 2.4 Využití HW prostředky

- FitKit3 Minerva 1 13/123
- 7-segmentový displej HD-M514RD

## 3 Ovládání

Následující tabulka definuje chování programu po stisknutí příslušného tlačítka na přípravku.

SW2	Začne počítat. Pokud již program počítá, zaznamená časový okamžik do paměti.
SW3	Vypne počítání. Pokud již program nepočítá, restartuje čas a vyprázdňuje paměť časových okamžiků.
SW4	V případě, že program nepočítá, přeskočí na předchozí zaznamenaný časový okamžik.
SW5	V případě, že program nepočítá, přeskočí na následující zaznamenaný časový okamžik.

## 4 Závěr

Výsledný projekt funguje podle specifikovaného zadání, rozložení problému na podproblémy je přiměřené zadání a kód je rozšiřitelný.