

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



Dokumentácia k projektu pre predmet IMP  
Výťah opäť nechodí - Riadenie výťahu cez MCU  
rady MSP430x

8. Decembra 2016

Autor: Matej Marušák xmarus06@stud.fit.vutbr.cz

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Návrh riešenia problému</b>	<b>2</b>
2.1	Spracovanie signálov . . . . .	2
2.2	Riadenie výťahu . . . . .	2
2.3	Správanie výťahu . . . . .	3
2.4	Odlišnosť od zadania . . . . .	3
<b>3</b>	<b>Implementácia programu</b>	<b>3</b>
3.1	Ukladanie stavu programu . . . . .	3
3.2	Fronta s unikátnymi hodnotami . . . . .	3
<b>4</b>	<b>Práca s aplikáciou</b>	<b>4</b>
4.1	Vstup aplikácie . . . . .	4
4.2	Výstup aplikácie . . . . .	4
<b>5</b>	<b>Analýza pamäťových nárokov</b>	<b>5</b>
<b>6</b>	<b>Záver</b>	<b>5</b>

# 1 Úvod

Táto práca vznikla ako súčasť projektu v rámci predmetu Mikroprocesorové a vestavěné systémy vyučovaného na Fakulte informačných technológií Vysokého učení technického v Brne.

Náplňou práce je analýza problému riadenia výťahu a následná implementácia programu na riadenie výťahu v jazyku C pre mikrokontrolér rady MSP430x osadený na vývojovom kite FitKit 2.

## 2 Návrh riešenia problému

Samotné riadenie výťahu pozostáva z reagovania na dva druhy signálov a následné rozhodovanie ako, kam a kedy sa bude kabína výťahu pohybovať.

### 2.1 Spracovanie signálov

Signály rozlišujeme na signály od užívateľov výťahu (stlačenie tlačítka na privolanie výťahu, vybranie si poschodia kam chce ísť atď.) a signály od senzorov výťahu (kabína výťahu prišla na ďalšie poschodie, dvere sú otvorené/zavreté atď.). Tieto signály prichádzajú náhodne a môže sa stať že príde viac signálov súčasne. Je potrebné si pamätať poradie vytvorených požiadaviek a obsluhovať ich postupne. Rozhodovací algoritmus obsluhy požiadaviek od užívateľov je znázornený na obrázku 1. Na obrázku je viditeľné, ako sú niektoré signály úmyselne zahodené a v akom prípade sú signály zapamätané. 'Zastávka += X' znamená, že sa poznamená aby výťah na tomto poschodí zastavil. Jedná sa o frontu bez duplicitných hodnôt, to znamená, že hodnota je vložená na koniec v prípade, že už takáto hodnota sa v zozname nenachádza. 'PORUCHA' a 'OPRAVA' znamená uloženie požiadavky na pozastavenie prevádzky alebo jej opätovné spustenie.

Príjem signálov od senzorov nie je ničím zaujímavý. Každý signál je poznamenaný a algoritmus riadenia výťahu sa podľa týchto údajov dokáže rozhodnúť, akú akciu vykoná. Viac o riadiacom algoritme je v nasledujúcej kapitole.

### 2.2 Riadenie výťahu

Po získaní dát od užívateľov a senzorov výťahu, je potrebné na tieto dáta reagovať. Algoritmus môže byť popísaný stavovým automatom. Stavový automat je avšak príliš komplikovaný aby bol uvedený v tejto dokumentácii. Dovoľujem si preto radšej ho popísať textovo:

1. Ak nastala porucha, zastav všetky prebiehajúce akcie
2. Ak je potrebné zatvoriť dvere -> 'Zatvor dvere'
3. Ak je prijatý signál 'oprava' -> Zaháj 'Oprava'
4. Ak prebieha 'Oprava' počkaj kým kabína príde na 1 poschodie
5. Ak nastupujú/vystupujú užívatelia, čakaj
6. Ak je potrebné otvoriť dvere -> 'Otvor dvere'
7. Ak kabína stojí a dvere sú otvorené a existuje požiadavka -> 'Zatvor dvere'
8. Ak kabína stojí a dvere sú zatvorené -> 'Začni pohybovať kabínou k smeru želaného poschodia'
9. Ak sa kabína hýbe a prišla na poschodie na ktoré je zadaná požiadavka -> 'Zastav a otvor dvere'

Ak je ľubovoľná podmienka splnená, vykonávanie sa preruší a pokračuje sa od začiatku. Je tak riešená priorita, teda napríklad sa kabína nezačne pohybovať bez zatvorených dverí.

### 2.3 Správanie výťahu

Nakoľko pohyb výťahu je len simulovaný, dáta od senzorov výťahu musia byť taktiež simulované. Preto boli určené časové hodnoty, za ktoré by mali jednotlivé signály prísť a po tomto čase sa správame, ako keby daný signál prišiel.

Zatvorenie a otvorenie dverí trvá 1 sekundu. Pohyb kabíny z jedného poschodia na druhé trvá 5 sekúnd. Doba na výstup a nástup trvá 2 sekundy.

### 2.4 Odlišnosť od zadania

Zo zadania: *Po zastavení výťahu v daném patře se v tomto patře po krátkém čekání otevřou dveře umožňující přístup k výťahové kabině. Dveře zůstanou určitou dobu otevřené a poté se začnou pomalu zavírat.*

Z toho vyplýva, že výťah, ktorý stojí na určitom poschodí a nemá žiadne požiadavky na vybavenie, má zavreté dvere. Avšak reálne výťahy po zastavení na poschodí a otvorení dverí nechávajú dvere otvorené, ak nemajú žiadnu ďalšiu požiadavku. Rovnako teda je navrhovaný aj tento výťah.

## 3 Implementácia programu

Program bol podľa zadania implementovaný v jazyku C. Hlavná smyčka v tele main funkcie testuje zmeny stavu programu a podľa nich sa rozhoduje, akú funkcionálnu vyvolá, alebo akú zmenu vykoná. Táto smyčka sa opakuje každých 10ms.

Aby však prichádzajúce udalosti počas vykonávania inej činnosti sa nestratili, klávesnica je obsluhovaná za využitia prerušenia.

Program bol inšpirovaný demo aplikáciou Demo - klávesnice využívajúci prerušenie. V súbore main.c táto podobnosť je takmer nulová, avšak ostatné súbory sú takmer totožné so súborami z tejto demo aplikácie.

### 3.1 Ukladanie stavu programu

Na ukladanie stavu aplikácie bola využitá vlastnosť jazyka C, bitfield. Táto funkcionálna umožňuje efektívne ukladať stavy aplikácie na minimálne množstvo pamäte s dobrým prístupom.

Ďalej na ukladanie bolo použité pole char-ov a unsigned integer-y.

### 3.2 Fronta s unikátnymi hodnotami

Na udržiavanie prehľadu, ktoré poschodia majú byť navštívené bola implementovaná fronta s viacerými špeciálnymi vlastnosťami. Prvou je, že sa vkladajú hodnoty na koniec fronty len vtedy, ak sa žiadna položka s rovnakou hodnotou vo fronte už nenachádza. Táto vlastnosť je potrebná preto, lebo pri návšteve jedného poschodia sú vybavené všetky požiadavky.

Z tejto fronty môžu položky vypadávať z ľubovoľnej pozície (keď sa prechádza cez poschodie na ktorom existuje požiadavka, toto poschodie je vybavené aj keď nežiadalo ako prvé). Preto je potrebné aby sa položky zmršťovali dopredu fronty. Teda ak vypadne položka nie z konca zoznamu, všetky položky sa posunú vpred v zozname. Vďaka tejto vlastnosti môže pevne obmedziť veľkosť fronty na počet poschodí v budove.

## 4 Práca s aplikáciou

Aplikácia beží na FitKit-e 2. Ovládanie je simulované za pomoci tlačidiel klávesnice. Výstup je zobrazovaný v terminálovom okne, hlavne vďaka možnosti zobraziť viac údaj prehľadnejšie.

### 4.1 Vstup aplikácie

Aplikácia priíma vstupy z FitKit-u. Na obrázku 2 je vyfotený FitKit s označenými tlačidlami. 'VONKU' znamená tlačidlo na privolanie výťahu vonku na šachte výťahu (teda VONKU 2 je tlačidlo na druhom poschodí). 'VNÚTRI' označuje všetky tlačidlá vnútri kabíny na výber poschodia.

### 4.2 Výstup aplikácie

Výstup aplikácie je zobrazovaný v terminály za pomoci matice ohraničenej hviezdčikami. Nová matica je zobrazená pri každej zmene, ktorá je viditeľná pre používateľa výťahu. Jedna ukážková matica je zobrazená na obrázku 3. Je rozdelená na dve časti.

Prvá časť s hlavičkou 'ON FLOORS' obsahuje 4 riadky pre každé poschodie. Jedná sa o informácie, ktoré sa dajú zistiť na každom poschodí zvlášť. 'STATE' zobrazuje aktuálny stav výťahu. 'STAY' znamená že kabína stojí, 'UP'/'DOWN' kabína sa pohybuje hore/dole a 'ERR' ak nastala vo výťahu chyba. Položka 'CURR' zobrazuje aktuálne poschodie na ktorom sa kabína nachádza, resp. ktorým naposledy prechádzala. 'DOOR' znamená či sú dvere na danom poschodí otvorené ('O') alebo zavreté ('C').

Druhá časť, 'INSIDE CABIN' zobrazuje dáta, ktoré sú zistiteľné z kabíny. Teda 'STATE' obsahuje stav, rovnako ako je to aj v informáciách na poschodiach. Rovnako sa vypisuje aj poschodie pod položkou 'ACTUAL'. Posledná tabuľka 'VISITING' označuje, ktoré tlačidlo s poschodím svieti. Ak je pri niektorom '0', znamená to, že toto poschodie bolo vybrané (či už z vnútra kabíny, alebo zvonku).

```
*****
*=====ON==FLOORS=====*
*   | STATE | CURR | DOOR | *
* -1 | STAY  |   3   |   C   | *
*  1 | STAY  |   3   |   C   | *
*  2 | STAY  |   3   |   C   | *
*  3 | STAY  |   3   |   O   | *
*=====INSIDE=CABIN=====*
*   STATE :      STAY      *
*   ACTUAL :          3     *
* VISITING | -1 | 1 | 2 | 3 | *
*          |   |   |   |   | *
*****
```

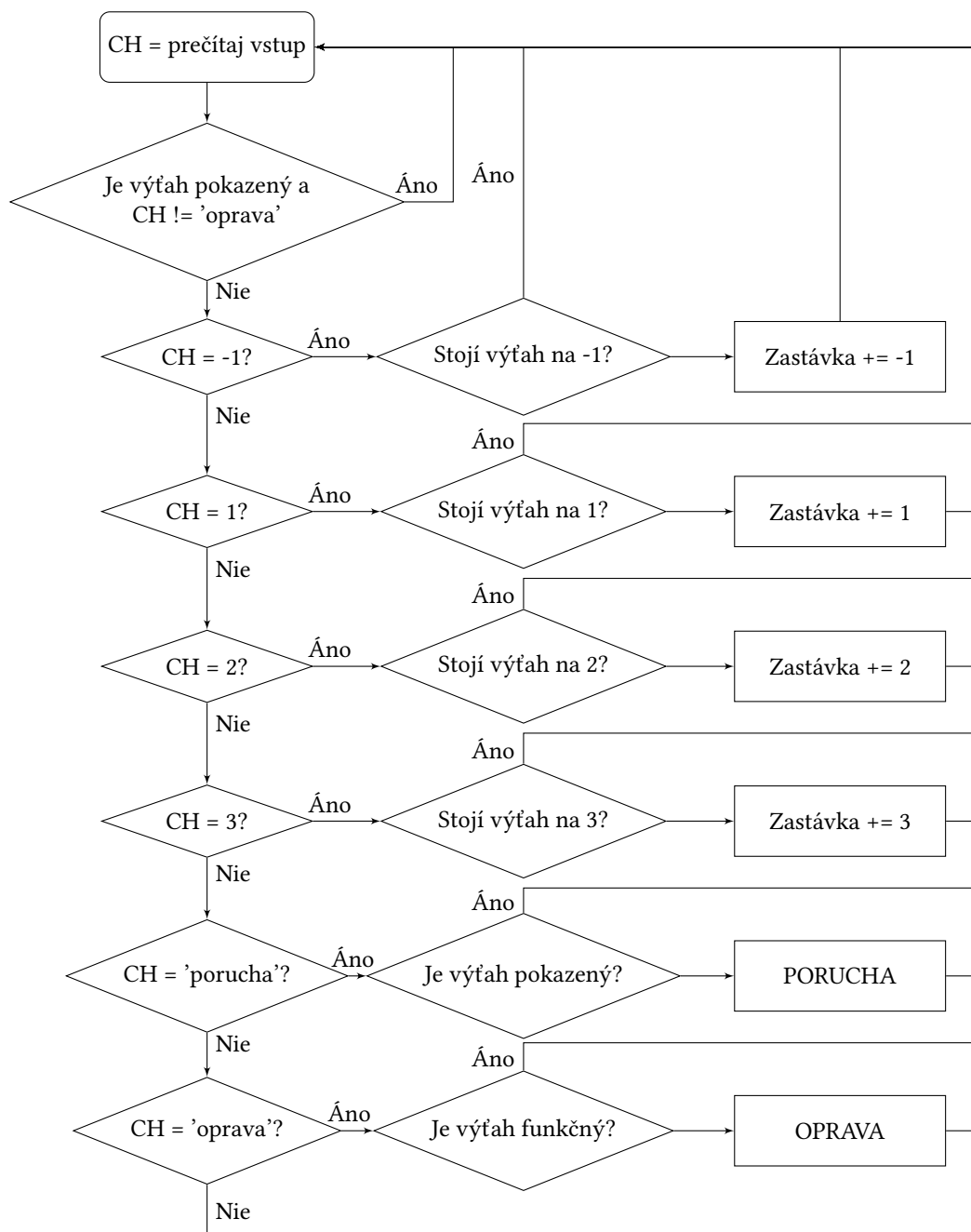
Obr. 3: Výpis stavu aplikácie.

## 5 Analýza pamäťových nárokov

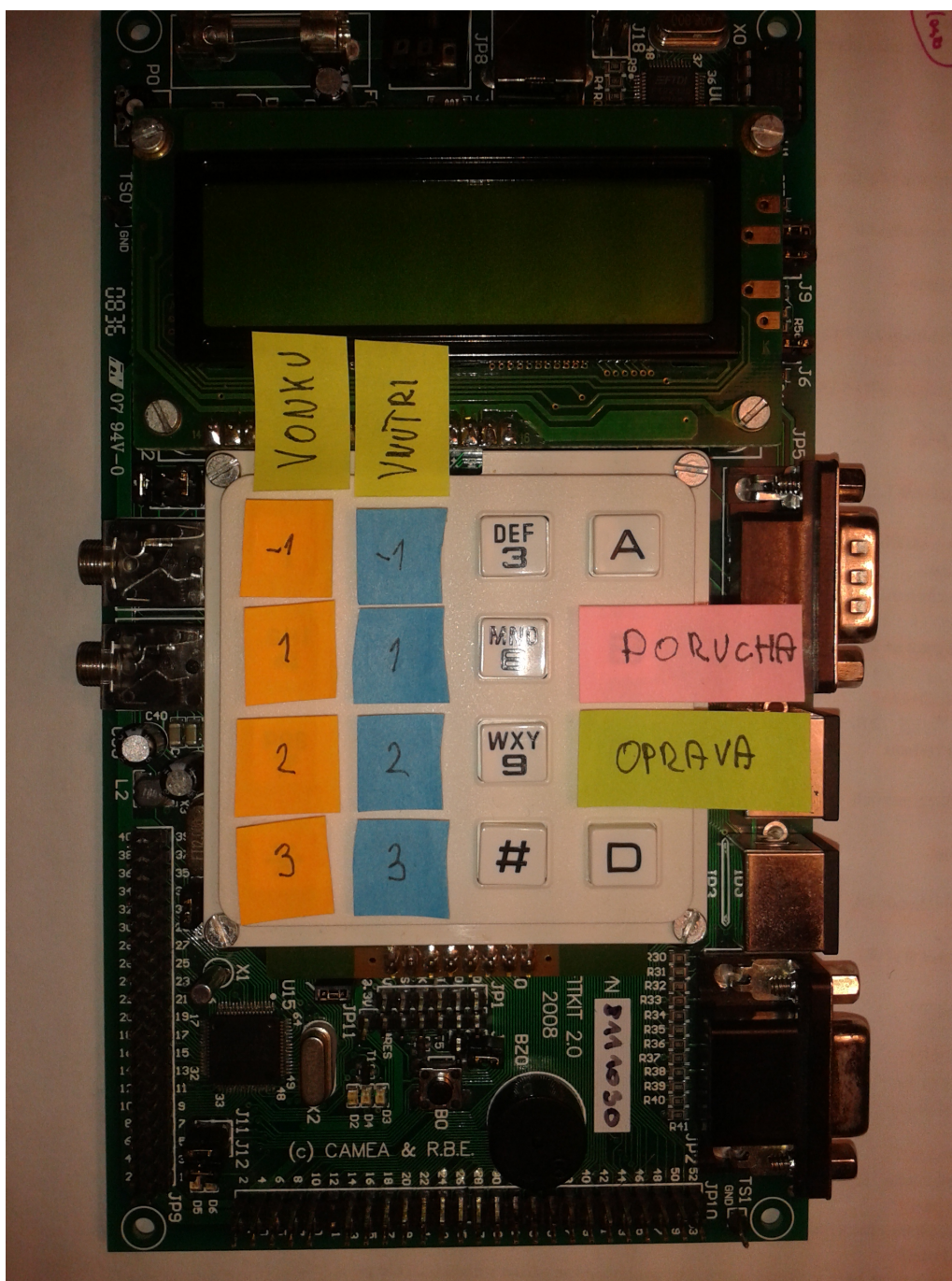
Nároky na RAM	22944b
Nároky na Flash pamäť	370b
Počet riadkov <code>main.c</code>	472

## 6 Záver

Aplikácia bola aktívne testovaná a správne fungovala pri všetkých testoch.



Obr. 1: Proces obsluhy užívateľských signálov



Obr. 2: Upravená verzia fitkitu pre ľahšie ovládanie aplikácie.