

Programování

5	SPŠ Chomutov	Kapusňák Jiří
5.4.2022	Výtah	V4

Zadání

Vytvořte s využitím operačního systému Linux program pro naprogramovatelný výtah, který využívá techniky krokového motoru a jeho pohyb se určuje tlačítkama na patře či u kabiny. Tento program by měl obsahovat:

- Zvolení patra
- Základní pozice
- Ošetření chyb (otevření dveří uprostřed jízdy apod.)
- Povolení tlačítek podle toho zda člověk je v kabině (není, fungují tlačítka na patrech, opačně fungují tlačítka u kabiny)
- Zobrazování patra na segmentu
- Rozsvícování LED diod na patrech podle směru jízdy
- Zvuk po příjezdu na zvolené patro

Teorie

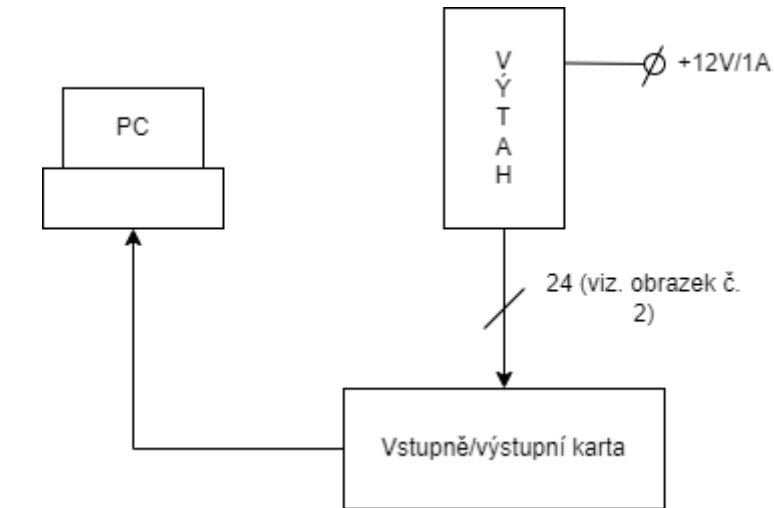
K samotnému programování používáme konzolovou aplikaci s textovým uživatelským rozhraním Midnight Commander. Konzolovou aplikaci máme propojenou vstupně/výstupní kartou se 4 porty. P1-P2 pouze čtení (inb 0x300 a 0x301) a P3-P4 pouze pro zápis (outb 0x300 a 0x301). V našem případě používáme všechny porty. Porty jsme zapojili podle obr. číslo 2.

Po spuštění počítače musíme naboootovat Linux podle cesty C:\TCE\BOOT\TCE.BAT a tím se dostaneme do rozhraní s kterým pracujeme. K použití máme 6 konzol, které přepínáme pomocí ALT + F1-F6. Otevřeme si správce souborů pomocí příkazu „mc“ a vytvoříme si nový adresář (F7). V námi vytvořeném adresáři ještě musíme vytvořit nový soubor (SHIFT + F4) a uložit (F2) jako main.c. Vrátime se zpět do jakékoliv konzole a pomocí příkazu „cd nasadresar/“ přejdeme do našeho adresáře aby jsme mohli zkompileovat náš kód příkazem „gcc main.c -o app -Wall“ (zkompiluj kód z main.c do aplikace, pojmenuj ji app a vyzkoušej všechny možné upozornění/chyby). Aplikaci zapneme ./app. Do samotného kódu ještě musíme zakomponovat funkci „ioperm“, která nám bude vyhazovat zda máme práva s portama na kartě pracovat, což mít nebudeme. Takže při spouštění programu musíme psát „sudo ./app“ (sudo = administrátorská práva na jeden příkaz).

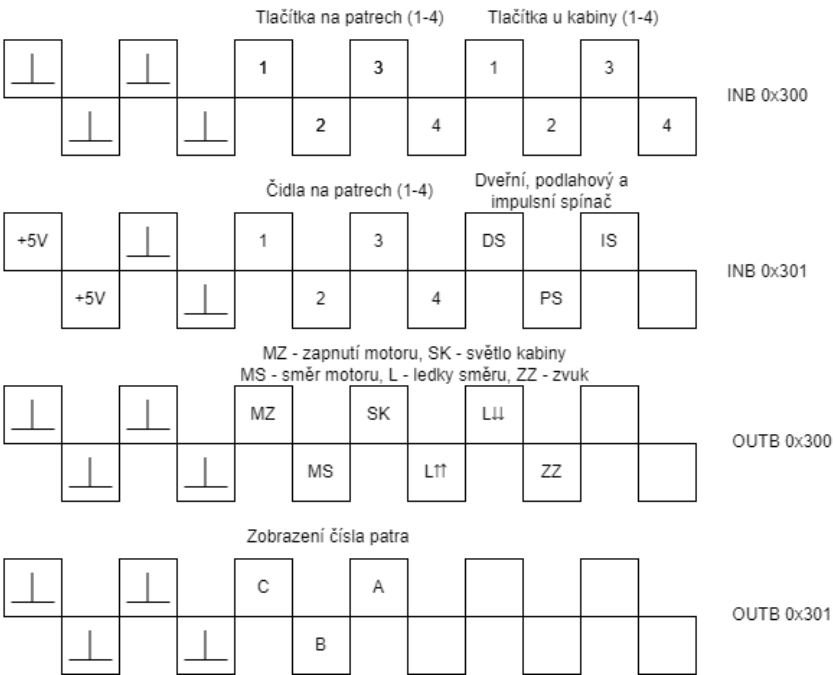
Výtah využívá k pohybu technologie krokového motoru, což znamená, že robot se hýbe po velmi malých krocích. Krokový motor obsahuje segmentovaný zmagnetovaný stator a rotor s určitým počtem cívek, které když jsou pod napětím tak vytvářejí severní a jižní póly, které buď tlačí nebo táhnou rotor tak aby se roztočil. Představme si to jako „impulz“ kde přivedeme napětí do protějších cívek a motor se posune ze své klidové polohy do nejbližší klidové polohy po vytvoření pólů. Tomu se říká krok a pro celý pohyb výtahu děláme těchto pohybů mnoho.

Výtah napájíme 12V a 1A (viz. obrázek č. 1), všechny vstupy a výstupy jsou aktivní v úrovních TTL v logické nule.

Schéma zapojení



(obrazek č. 1)



(obrazek č. 2)

Popis Programu:

Po zapnutí programu se výtah automaticky posune do své základní pozice, kterou jsme zvolili 1. patro (úplně dole). Poté výtah zkontroluje zda podlahový spínač je v logické nule a podle toho nastaví zda budou fungovat tlačítka na patrech nebo u kabiny.

Dále výtah kontroluje v cyklu while zda uživatel zvolil nějaké patro, vždy se nám uloží zvolené patro do proměnné zvolen, když je zvolen 0 výtah se nijak nehýbe a pouze napíše do konzole „Zvolte patro“.

Když uživatel nějaké patro zvolil, zvolené patro se opět uloží do proměnné zvolen a ještě se zkontroluje patro ve kterém výtah stojí podle čidel na inb 0x301. Když je zvolené patro vyšší než patro ve kterém výtah stojí bude se posouvat nahoru, v opačném případě dolů a když zvolené patro je stejné jak patro ve kterém stojí, nic se nestane.

Posun funguje následovně, určí se tedy jakým směrem výtah pojede, jestli nahoru nebo dolů, spustí se nový cyklus while, který pojede dokud výtah nedorazí na zvolené patro. Při každém cyklu se zkontroluje zda jsou otevřené dveře, když jsou, výtah se zastaví a bude do konzole vypisovat „Jsou otevřené dveře!!!!“. Podle směru se rozsvítí na outb 0x300 na 3. nebo 4. bitu (viz. obrázek č. 2 ve Schéma Zapojení), dále se rozsvítí v kabině světlo.

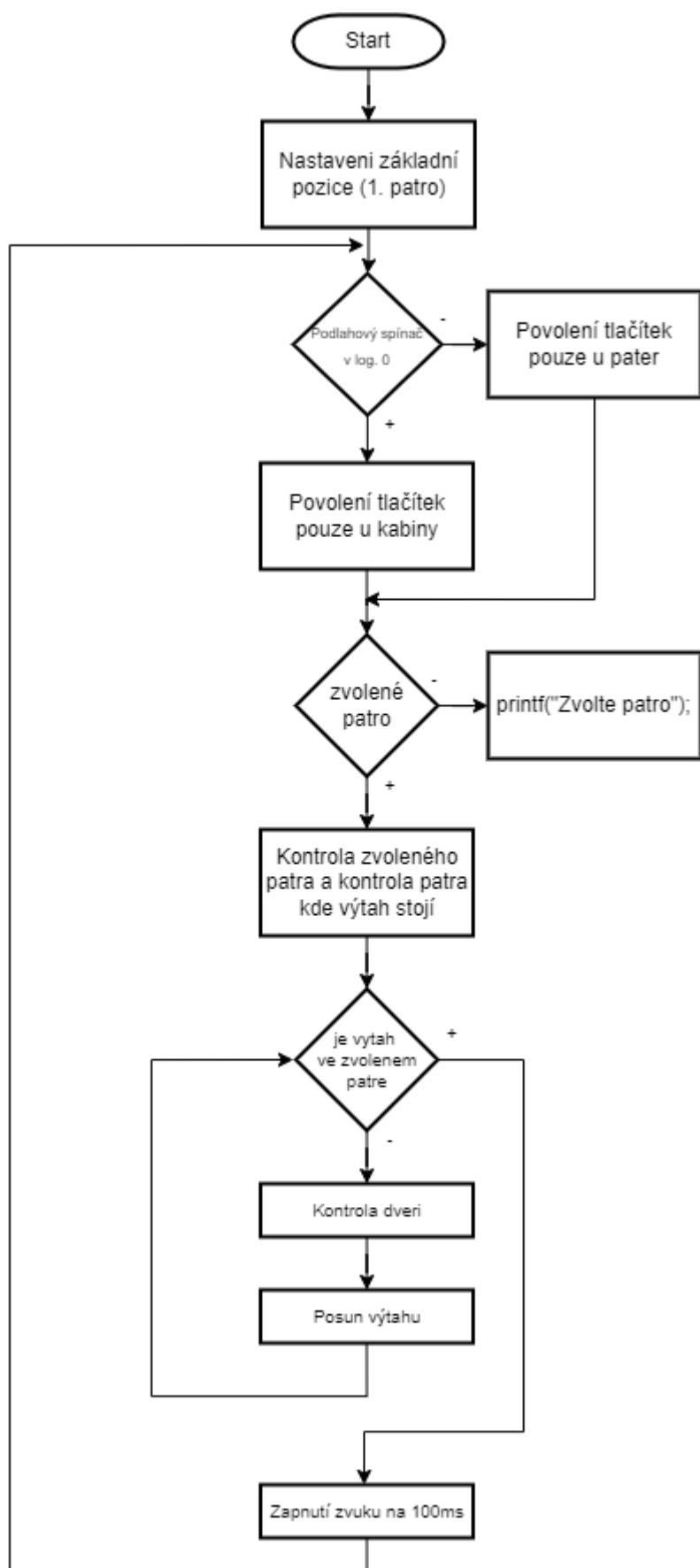
Když se výtah pohybuje tak se ještě kontroluje furt dokola čidla na patrech a podle toho se nastavuje segment na výtahu, který vypisuje v jakém patře momentálně výtah je, nebo kudy projíždí. Po dojezdu na zvolené patro výtah vydá na 100ms zvuk

Popis proměnných

Typ	Proměnná	Účel
int	nahoru	Bitové číslo pro posun nahoru
int	dolu	Bitové číslo pro posun dolů

Typ	Argumenty	Metoda	Účel
int		prava	Nastavení iopermu
void		zvuk	Zvuk na 100ms
int	int hodnota, int bit	kontrolabitu	Kontrola zda bit je set nebo clear
void		segment	Zvolení bit. čísla pro segment
void		nastavenisegmentu	Nastavení čísla patra
int		kontroladveri	Kontrola dveří
int		patro	Patro ve kterém výtah je
int		zvolenepatro	Patro co zvolil uživatel
int		vychozipozice	Posun na vychozi pozici
void		nastaveniportu	Zapsani samych jednicek na porty
void	int kam, int patro	posun	Posun vytahu
int		main	Hlavní smyčka

Vývojový diagram



Vypis programu

```
1. #include <stdio.h>
2. #include <sys/io.h>
3. #include <time.h>
4.
5. int dolu = 232;
6. int nahoru = 242;
7.
8. int prava() {
9.     if (ioperm(0x300,2,1)!= 0) {
10.         return 0;
11.     }
12.     else {
13.         return 1;
14.     }
15. }
16. void zvuk() {
17.     struct timespec t;
18.     t.tv_sec = 0;
19.     t.tv_nsec = 100000000;
20.     outb(223, 0x300);
21.     nanosleep(&t, NULL);
22. }
23.
24. int kontrolabitu(int hodnota, int bit) {
25.     int maska = (1 << bit);
26.     int vysledek = (hodnota & maska);
27.     if (vysledek == 0) {
28.         return 0;
29.     }
30.     else {
31.         return 1;
32.     }
33. }
34.
35. void segment(int cislo) {
36.     switch(cislo) {
37.         case 0:
38.             outb(252 ,0x301);
39.             break;
40.         case 1:
41.             outb(250 ,0x301);
42.             break;
43.         case 2:
44.             outb(254 ,0x301);
45.             break;
46.         case 3:
47.             outb(249 ,0x301);
48.             break;
49.     }
50. }
51.
52. void nastavenisegmentu() {
53.     int stavP4 = inb(0x301);
54.     for (int i = 0; i < 4; i++) {
55.         if (kontrolabitu(stavP4, i) == 0) {
56.             segment(i);
57.         }
58.     }
59. }
60.
61. int kontroladveri() {
62.     int stavP4 = inb(0x301);
63.     if (kontrolabitu(stavP4, 4) == 0) {
64.         return 1;
65.     }
66.     return 0;
```

```

67. }
68.
69. int patro() {
70.     int stavP4 = inb(0x301);
71.     for (int i = 0; i < 4; i++) {
72.         if (kontrolabitu(stavP4, i) == 0) {
73.             return i;
74.         }
75.     }
76. }
77.
78. int zvolenepatro() {
79.     int stavP3 = inb(0x300);
80.     for (int i = 0; i < 4; i++) {
81.         if (kontrolabitu(stavP3, i) == 0) {
82.             return i + 1;
83.         }
84.     }
85.     for (int i = 4; i < 8; i++) {
86.         if (kontrolabitu(stavP3, i) == 0) {
87.             return i - 3;
88.         }
89.     }
90.     return 0;
91. }
92.
93. void vychozipozice() {
94.     while(kontrolabitu(inb(0x301), 0) != 0) {
95.         outb(dolu, 0x300);
96.     }
97.     zvuk();
98.     outb(255, 0x300);
99.     nastavenisegmentu();
100. }
101.
102. void nastaveniportu() {
103.     outb(255, 0x300);
104.     outb(255, 0x301);
105. }
106.
107. void posun(int kam, int patro) {
108.     while (kontrolabitu(inb(0x301), patro) != 0) {
109.         if (kontroladveri() == 0) {
110.             printf("jsou otevrene dveře\n");
111.             outb(255, 0x300);
112.         }
113.         else {
114.             nastavenisegmentu();
115.             printf("vytah jede\n");
116.             outb(kam, 0x300);
117.         }
118.         nastavenisegmentu();
119.         zvuk();
120.         outb(255, 0x300);
121.     }
122. }
123.
124. int main() {
125.     if (prava() == 0) {
126.         return 0;
127.     }
128.     else {
129.         nastaveniportu();
130.         vychozipozice();
131.         while (1) {
132.             int zvolen = zvolenepatro();
133.             if (zvolen == 0) {
134.                 printf("zvolte patro\n");
135.             }
136.             else {

```

```
137.             int pravda = zvolen - 1;
138.             if (pravda > patro()) {
139.                 posun(nahoru, pravda);
140.             }
141.             else if (pravda < patro()) {
142.                 posun(dolu, pravda);
143.             }
144.             else {
145.                 outb(255, 0x300);
146.             }
147.         }
148.     }
149.
150.     return 1;
151. }
152. }
```


Odpovědi na otázky

1. Pro řízení pohonů větších výkonů není ekonomické vymýšlet vlastní řešení ovládání. Jaké komerčně dostupné zařízení byste pro číslicové ovládání takových motorů použili?
2. Při integraci různých jednotek v rámci jednoho systému je nutné přenášet mezi nimi informace. Pokuste se pojmenovat jeden vybraný průmyslový protokol (automotive, automatizace...) a popsat jeho základní vlastnosti.
3. S rostoucím počtem pater našeho výtahu by vyhrazení IO linky pro každé tlačítko začalo být neúnosné. Pokuste se navrhnout efektivnější způsob s využitím analogového rozhraní (s měřením napětí).

Závěr

V programu výtahu nám chybělo zakomponování podlahového spínače, takže nebylo možné jak určit zda se budou používat tlačítka na patrech či u kabiny, takže fungovali obě najednou. Jinak výtah fungoval bez problémů.¹
