Dílenská praxe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A4** | B. Model výtahu | | | |
| Macháček Daniel | |  | 1/ | Známka: |
| 28.9.2016 | | Datum odevzdání: | 19.10.2016 | Odevzdáno: |

Zadání:

Zpracujte program v programovacím jazyce C ovládající model výtahu tak, aby obsahoval nejméně tyto funkce:

1. ovládání pohybu kabiny pomocí tlačítek na patrech
2. ovládání pohybu kabiny pomocí tlačítek v kabině
3. ovládání pomocných funkcí výtahu
4. respektování funkcí tlačítek v závislosti na stavu výtahu (obsazená, případně plná kabina, . . . )
5. sledování provozních a chybových stavů

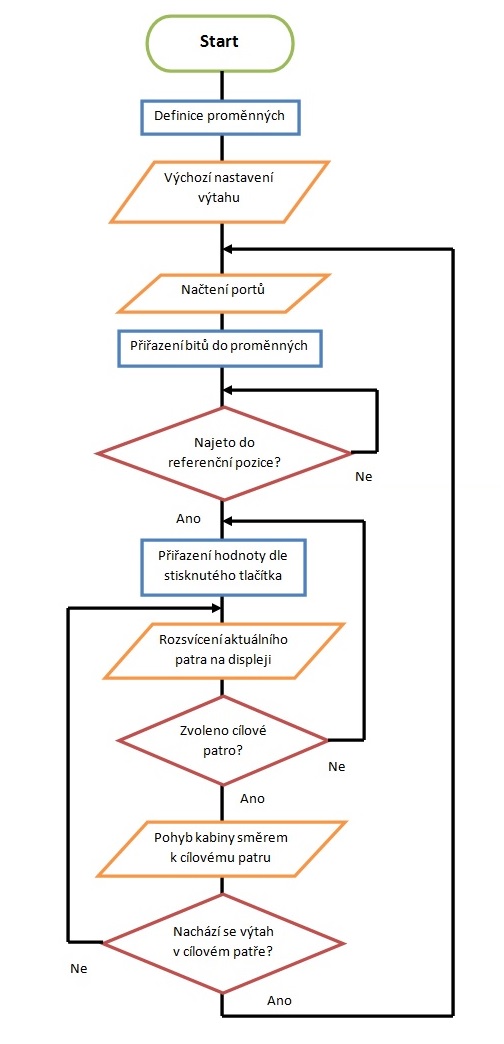
Závěr:

Navzdory mému snaze můj program nedokázal pohnout kabinou. Po důkladném prozkoumání programu bych ale chybu nepřičítal samotnému programu nýbrž zapojení. Při nedostatku času na konzultačních hodinách jsem narychlo zapojoval model výtahu (s nedostatkem kabelů – tudíž jsem musel přemýšlet, které vodiče si můžu dovolit vynechat, a navíc ještě špatně drželi, což mohlo vyústit v další chyby při jejich navracení) a nejspíš jsem někde udělal chybu.

Přílohy:

* Výpis programu – 4 strany
* Vývojový diagram – 1 strana

**Vývojový Diagram:**



**Výpis programu:**

#include <stdio.h> //Otevření knihovny

#include <dos.h> //Otevření knihovny

int TK1, TK2, TK3, TK4; //Definice proměnných

int TP1, TP2, TP3, TP4; //Definice proměnných

int C1, C2, C3, C4; //Definice proměnných

int DS, PS, IS; //Definice proměnných

int i300, i301; //Definice proměnných

int Cil = 0; //Definice proměnné a přiřazení hodnoty 0

int Pat; //Definice proměnné

int status = 0; //Definice proměnné a přiřazení hodnoty 0

void main() //Hlavní pole

{

outport(0x300, 0xFF); //Výchozí hodnota pro port

outport(0x301, 0xFF); //Výchozí hodnota pro port

while (1) //Nekonečný cyklus

{

i300 = inport(0x300); //Načtení hodnot z portu

i301 = inport(0x301); //Načtení hodnot z portu

TK1 = i301 & 0x40; //Rozpoznání bitu pro TK1 pomocí hodnoty masky

TK2 = i301 & 0x10; //Rozpoznání bitu pro TK2 pomocí hodnoty masky

TK3 = i301 & 0x04; //Rozpoznání bitu pro TK3 pomocí hodnoty masky

TK4 = i301 & 0x01; //Rozpoznání bitu pro TK4 pomocí hodnoty masky

TP1 = i301 & 0x80; //Rozpoznání bitu pro TP1 pomocí hodnoty masky

TP2 = i301 & 0x20; //Rozpoznání bitu pro TP2 pomocí hodnoty masky

TP3 = i301 & 0x08; //Rozpoznání bitu pro TP3 pomocí hodnoty masky

TP4 = i301 & 0x02; //Rozpoznání bitu pro TP4 pomocí hodnoty masky

C1 = i300 & 0x40; //Rozpoznání bitu pro C1 pomocí hodnoty masky

C2 = i300 & 0x10; //Rozpoznání bitu pro C2 pomocí hodnoty masky

C3 = i300 & 0x04; //Rozpoznání bitu pro C3 pomocí hodnoty masky

C4 = i300 & 0x01; //Rozpoznání bitu pro C4 pomocí hodnoty masky

DS = i300 & 0x80; //Rozpoznání bitu pro DS pomocí hodnoty masky

PS = i300 & 0x20; //Rozpoznání bitu pro PS pomocí hodnoty masky

IS = i300 & 0x08; //Rozpoznání bitu pro IS pomocí hodnoty masky

switch (status) //Vybírá proces podle hodnoty status

{

case 0: //Když je hodnota status 0 (Nájezd do referenční pozice - první partro)

{

if (C1 == 0) //Jestli najela kabina do referenční pozice

{

status = 1; //Nastaví status hodnotou 1

}

else //Kabina není v prvním patře

{

outport(0x300, 0xAF); //Pošle na port kombinaci bitů pro pohyb do prvního patra

status = 0; //Nastaví status hodnotou 0

}

}

break; //Konec procesu

case 1: //Když je hodnota status 1 (Test tlačítek)

{

if (DS == 0) //Dveře jsou zavřeny

{

if (PS == 0) //Pokud je někdo v kabině

{

if (TK1 == 0) Cil = 1; //Pokud je stisknuto tlačítko 1 v kabině nastaví Cil na 1

if (TK2 == 0) Cil = 2; //Pokud je stisknuto tlačítko 2 v kabině nastaví Cil na 2

if (TK3 == 0) Cil = 3; //Pokud je stisknuto tlačítko 3 v kabině nastaví Cil na 3

if (TK4 == 0) Cil = 4; //Pokud je stisknuto tlačítko 4 v kabině nastaví Cil na 4

}

else //Když není nikdo v kabině

{

if (TP1 == 0) Cil = 1; //Pokud je stisknuto tlačítko 1 na patře nastaví Cil na 1

if (TP2 == 0) Cil = 2; //Pokud je stisknuto tlačítko 2 na patře nastaví Cil na 2

if (TP3 == 0) Cil = 3; //Pokud je stisknuto tlačítko 3 na patře nastaví Cil na 3

if (TP4 == 0) Cil = 4; //Pokud je stisknuto tlačítko 4 na patře nastaví Cil na 4

}

}

status = 2; //Nastaví status hodnotou 2

}

break; //Konec procesu

case 2: //Když je hodnota status 2 (Test aktuálního patra + signalizace na segmentový displej)

{

if (C1 == 0) //Pokud je výtah v prvním patře

{

Pat = 1; //Nastaví se Pat na 1

outport(0x300, 0x04); //Pošle na segmentový displej kombinaci bitů pro rozsvícení příslušného patra

}

if (C2 == 0) //Pokud je výtah v druhém patře

{

Pat = 2; //Nastaví se Pat na 2

outport(0x300, 0x02); //Pošle na segmentový displej kombinaci bitů pro rozsvícení příslušného patra

}

if (C3 == 0) //Pokud je výtah v třetím patře

{

Pat = 3; //Nastaví se Pat na 3

outport(0x300, 0x06); //Pošle na segmentový displej kombinaci bitů pro rozsvícení příslušného patra

}

if (C4 == 0) //Pokud je výtah v čtvrtém patře

{

Pat = 4; //Nastaví se Pat na 4

outport(0x300, 0x01); //Pošle na segmentový displej kombinaci bitů pro rozsvícení příslušného patra

}

if (Cil > 0) //Bylo zvoleno patro

{

status = 3; //Nastaví status hodnotou 3

}

else //Nebylo zvoleno patro

{

status = 1; //Nastaví status hodnotou 1

}

}

break; //Konec procesu

case 3: //Když je hodnota status 3 (Pohyb motoru + světlo v kabině)

{

if (DS == 0) //Jestli jsou dveře kabiny zavřeny

{

if (Cil > Pat) //Když je kabina nad cílovým patrem

{

outport(0x300, 0xBB); //Poslání kombinace bitů pro pohyb kabinou příslušným směrem

status = 2; //Nastaví status na hodnotu 2

}

if (Cil < Pat) //Když je kabiny pod cílovým patrem

{

outport(0x300, 0xAB); //Poslání kombinace bitů pro pohyb kabinou příslušným směrem

status = 2; //Nastaví status na hodnotu 2

}

if (Cil == Pat) //Když je výtah v cílovém patře

{

outport(0x300, 0xFF); //Vypnutí pohonů

Cil = 0; //Nastaví Cil na 0

status = 1; //Nastaví status na hodnotu 1

}

}

else //Dveře otevřeny (možno za jízdy)

{

outport(0x300, 0xFF); //Vypnutí pohonů

Cil = 0; //Nastaví Cil na 0

status = 1; //Nastaví status na hodnotu 1

}

}

break; //Konec procesu

}

}

}