

PRAXE

5

STŘEDNÍ PRŮMYSLOVÁ ŠKOLA CHOMUTOV

Slapnička

5.4.2022

VÝTAH

V.4

Zadání

Vytvořte program ovládání výtahu, za pomoci vnějších signálů. Výtah musí být schopen se pohybovat mezi patry za pomoci tlačítek kabiny a tlačítek na patrech, když je výtah obsazen rozsvítí se světlo kabiny a prioritu volání mají tlačítka kabiny. Při pohybu mezi patry nesmí dojít k přerušení pohyb, dokud výtah nedojede do patra. Sedmi segmentový displej ukazuje patro, kde se výtah nachází a LED šipky směr pohybu výtahu(při zaslavení zhasnou).

Teorie

Tvorba MakeFile- Makefile slouží jako spustitelný soubor zlehčující kompilaci souboru, to znamená že předem zahrnuje všechny potřebné věci, námi nastavené, pro programátor(typ procesoru, frekvence, COM,atd..). Makefile je vytvářen pomocí programu kde je možno vše nastavit pomocí pár kliknutí, nemusí být upravován text.

Pro komunikaci s ATMegou je využíván jtag-usb a programátor. Když je program nahrán je přes usb poslán na programátor, který tento usb signál o nastavené frekvenci přemění na jtag se stejnou frekvencí a může být nahrán na procesor a spuštěn.

Vnitřní fungování výtahu- všechny funkce jsou aktivní v log 0. Pro zjištění polohy obsahuje výtah senzory na každém patře, když výtah přeruší senzor je vyslána log 0, to samé platí pro tlačítka kabiny a tlačítka na patrech. Sedmi segmentový displej obsahuje předem nadefinované čísla aktivní také log 0(0b011- 1....), dále kabina obsahuje senzor podlahy(obsazeno 0) a senzor dveří(při otevření 0). Motor má dva směry otáčení, které zapříčiní pohyb dolu-0 nebo nahoru-1 a aktivace probíhá 0.

Popis programu

Program po spuštění inicializuje porty a následně je nastaví do stavu log1, potom je spuštěna první smyčka, která hledá první patro tím že jede dolů dokud není snímač prvního patra v log 0 a tím se přeruší. Dále je spuštěna hlavní smyčka, která volá funkce ledpatro. Tato funkce slouží pro zobrazení daného patra na sedmi segmentovém displeji pomocí předem nadefinovaných čísel, další je stavtlacitek kontrolující jestli jsou zmáčknuta tlačítka ať už na kabině nebo na patrech a připisuje proměnou patro_request. Funkce stav_vytahu slouží ke kontrole senzorů pater a poté přepisuje proměnou patro, při otevření tato funkce zastaví výtah. Dále funkce predpohyb volá pohyb s hodnotou podle velikosti proměnné patro a patro_request jestli je menší nebo větší, když je menší jede výtah nahoru a v případě že je větší jede dolu. Jako poslední je voláno obsazení rozsvěcí světlo, když je obsazeno a zhasíná když není.

Rozbor proměnných a funkcí (metod)

Typ	Název	popis
Int	preruseni	Slouží k přerušení cyklu mezi funkcemi
Int	stavP1	Hodnota pro port P1
Int	patro	Kde se nachází výtah
Int	patro_request	Kam má výtah jet
Int	mezipatro	Kontrolní proměnná jestli není mezi patry
void	zpozdeni	Funkce pro zpoždění pomocí nanosleep
Void	pohyb	Funkce pro ovládání motorů a zastavení při otevřených dveřích
Int	stav1	Stav portu P2 za použití funkce inb zjistí co je za signál na portech
Void	Nalezeniprizemi	Funkce inicializující program nalezením přízemí(patru 1)
void	obsazeni	Funkce kontroluje stav kabiny-dveřní a podlahový snímač.
void	stavtlacitek	Funkce kontroluje zda nejsou tlačítka zmáčknuta na portu P1, když ano změní proměnnou patro_request na dané číslo
void	Stav_vytahu	Funkce kontroluje kde se výtah nachází pomocí patrových snímačů a kde výtah je takovou hodnotu změní proměnné patro
Void	predpohyb	Funkce volá funkci pohyb na základě velikosti patro a patro_request, buď nahoru nebo dolů, když je shodné vymaže šipky pohybu a proměnné mezipatro=0
void	ledpatro	Funkce nastavující sedmi segmentový displej podle proměnné patro
void	main	Inicializuje porty a nastaví je do samých jedniček, potom začne smyčka obsahující nalezeniprizemi. Po skončení této smyčky začne smyčka hlavní.

Vývojový diagram

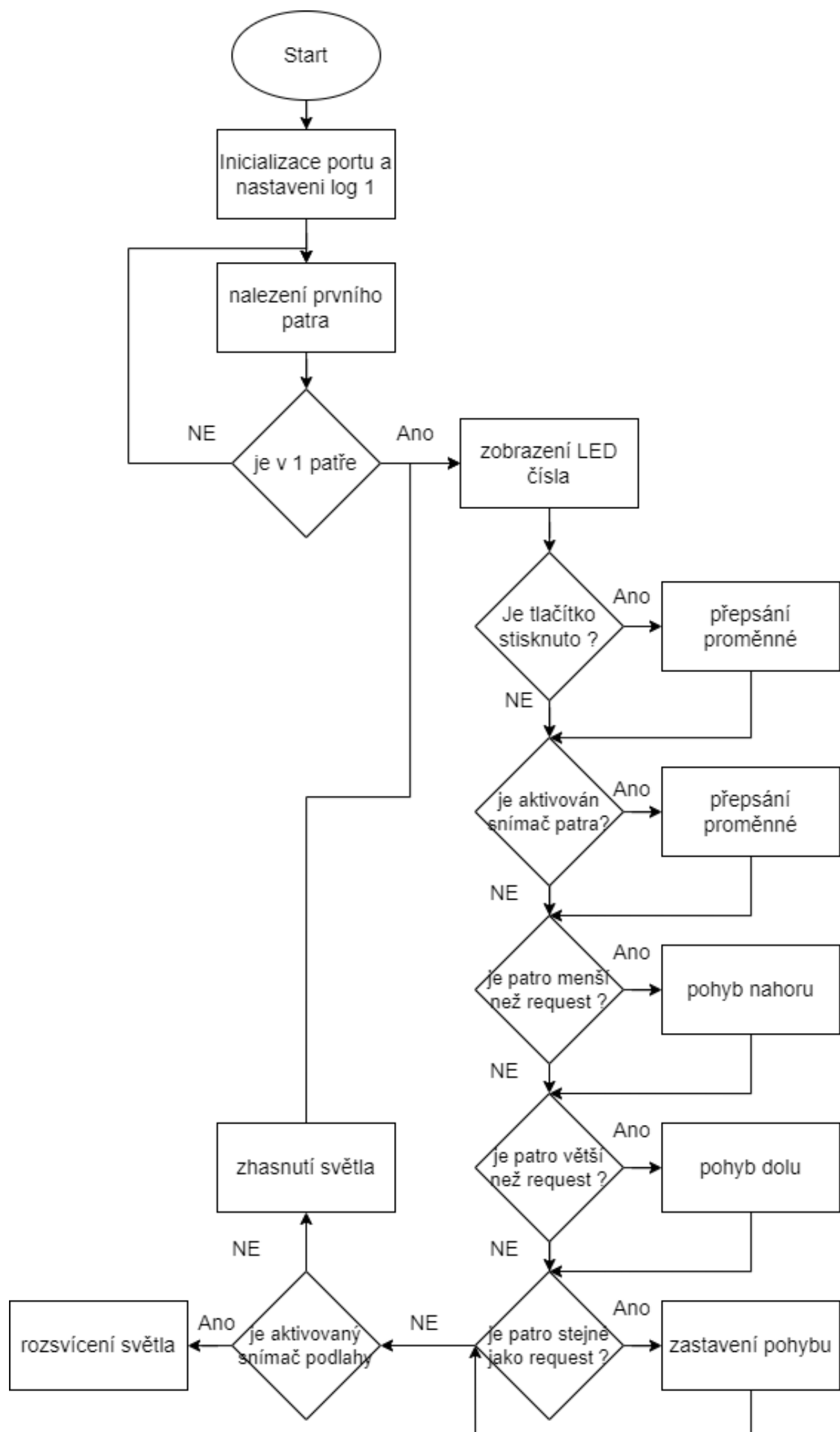
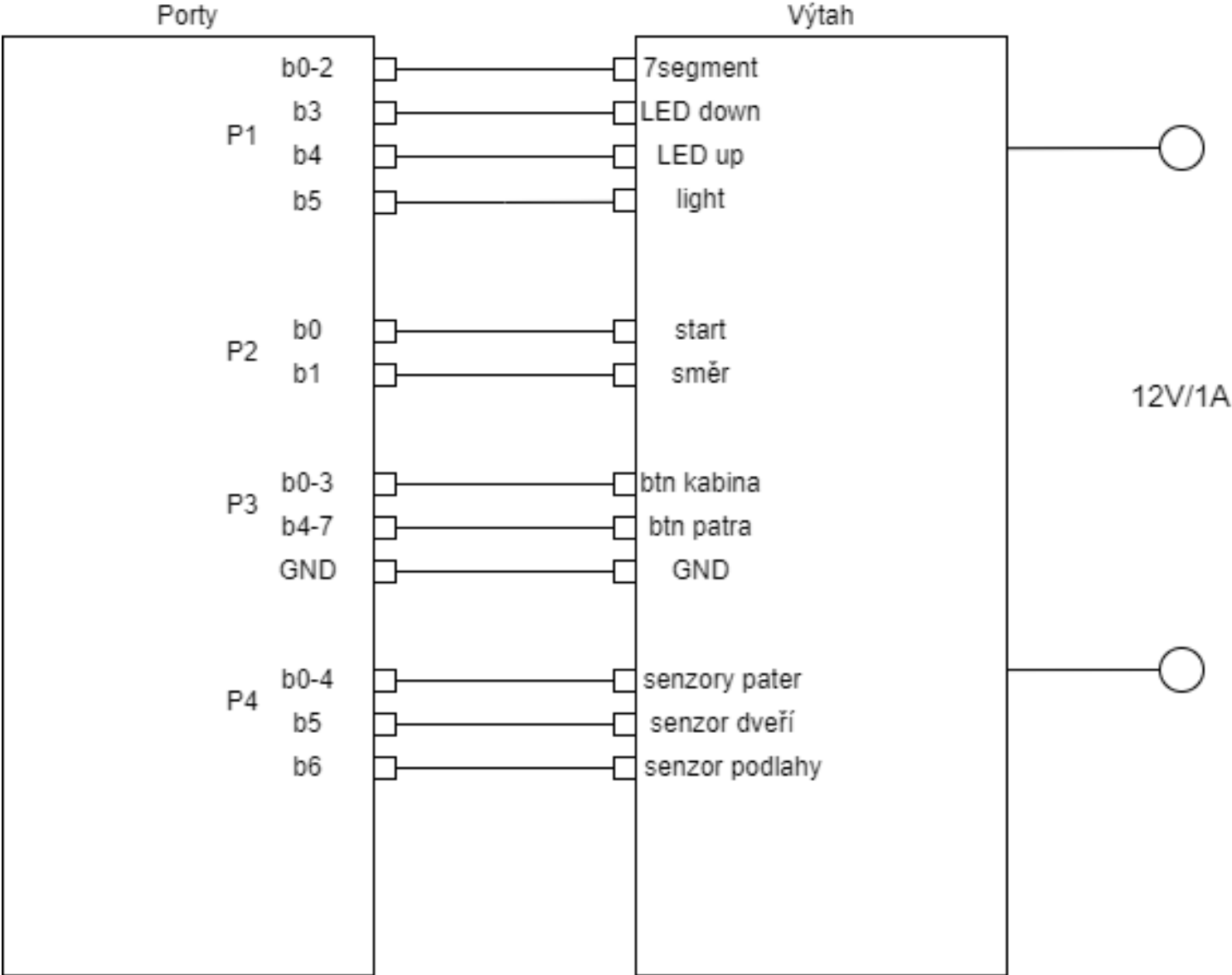


Schéma zapojení



Komentovaný výpis programu

```
# include <stdio.h>
# include <sys/io.h>
# include <time.h>

/*zapojení hw
P1 b0-2 7segmentovy displ.(111=7,110=6) b3 LED down, b4 LED up b5 cabin light b6 signal
P2 b0 motor start b1 smer
P3 b0-3 cabin buttons(jen 4) b4-7 floor buttons
P4 b0-4 senzory pater b5 senzor dveri b6 senzor podlahy
b7 snimac pohybu hridele
*/
static int preruseni = 0;
static int stavP1 = 255;//proměná pro port P1
static int patro = 1;//patro v kterém se nachází výtah
static int patro_request = 1;//volané patro
static int mezipatro = 1;

void zpozdeni(int sekundy, int nanosekundy)
{ //nastavení zpozdení pomocí nanosleep pro s a ns
struct timespec t;
t.tv_sec = sekundy;
t.tv_nsec = nanosekundy;
nanosleep(&t, NULL);
}

void pohyb(int smer)//fce pro pohyb výtahu(0-dolu, 1-nahoru)
{
    int stav1 = inb(0x301);
    if (smer == 0)
    { //smer dolu
        outb(0b11111100, 0x301);
        stavP1 = (stavP1 & ~(1 << 3)); //LED down
        stavP1 = (stavP1 | (1 << 4)); //reset bitu
        outb(stavP1, 0x300);
    }
    if (smer == 1)
    { //smer nahoru
        outb(0b11111110, 0x301);
        stavP1 = (stavP1 & ~(1 << 4)); //LED up
        stavP1 = (stavP1 | (1 << 3)); //reset bitu
        outb(stavP1, 0x300);
    }
    if (stav1 != (stav1 & ~(1 << 5))) //při otevření dveri zastavi
    {
        outb(255, 0x301); //vypnutí motoru
        zpozdeni(2, 0);
    }
    zpozdeni(0, 10000000);
}
```

```

void nalezeniprizemi()
{
    int stav1 = inb(0x301);
    if ((stav1 != (~(stav1 | (1 << 0))) && (stav1 != (stav1 | (1 << 5))))
    { //když není snímač 1 patra aktivní a výtah má zavřené dveře
        pohyb(0);
    }
    if (stav1 != (stav1 | (1 << 0))) //když je v 1 patře
    {
        preruseni = 1;
        outb(255, 0x301);
        zpozdeni(0, 100000);
    }
    if (stav1 != (stav1 & ~(1 << 5))) //při otevření dveří zastaví
    {
        outb(255, 0x301);
        zpozdeni(0, 100000);
    }
}

void obsazeni()
{ //fce pro zjistiení obsazeni kabiny (je světlo svítí, není nesvítí)
    int stav1 = inb(0x301);
    if (stav1 != (stav1 | (1 << 6))) //6bit floor sensor
    {
        stavP1 = (stavP1 & ~(1 << 5)); //změna 5 bitu p1 do 0 (aktivace)
        outb(stavP1, 0x300);
        zpozdeni(0, 100000);
    }
    if (stav1 != (stav1 & ~(1 << 6)))
    { //deaktivace když není sensor aktivní
        stavP1 = (stavP1 | (1 << 5)); //nastavení bitu do 1
        outb(stavP1, 0x300);
        zpozdeni(0, 100000);
    }
}

void stavtlacitek()
{ //fce pro detekci tlacitek
    int stav1 = inb(0x300);
    for (int i = 0; i < 4; i++)
    { //smyčka prověří zda jsou tlačítka kabiny aktivní
        if ((stav1 != (stav1 | (1 << i))) && (mezipatro == 0)) //vnitřní
        { //btn
            patro_request = (i + 1);
        }
    }
}

```

```

for (int i = 4; i < 8; i++)
{
    //smyčka prověří zda jsou tlačítka pater aktivní
    if ((stav1!=(stav1|(1<<i)))&&(patro==patro_request)&&(mezipatro==0))
        //vnější btn
        //když se výtah pohybuje není možno změnit patro_request
        patro_request = (i - 3);
}
}

void stav_vytahu()
{
    //fce na zjištění kde se nachází kabina
    int stav1 = inb(0x301);
    for (int i = 0; i < 4; i++)
    {
        //prověření všech senzorů pater
        if (stav1 != (stav1 | (1 << i)))
        {
            patro = (i + 1);
        }
    }
    if (stav1 != (stav1 & ~(1 << 5)))//při otevření dveří zastaví
    {
        outb(255, 0x301);
        zpozdeni(0, 1000000);
    }
}

void predpohyb()
{
    //určuje zda pojedě nahoru nebo dolů
    if (patro < patro_request)
    {
        //jede nahoru dokud se nerovnájí a patro je menší
        pohyb(1);
        mezipatro = 1;
    }
    if (patro > patro_request)
    {
        //jede dolů dokud se nerovnájí a patro je větší
        pohyb(0);
        mezipatro = 1;
    }
    if (patro == patro_request)//nulování LED šipek při zastavení
    {
        //v daném patře
        stavP1 = (stavP1 | (3 << 3));//vymazání šipek pohybu
        outb(stavP1, 0x300);
        mezipatro = 0;
    }
}

void ledpatro()
{
    //ovládání 7 segmentového displeje
    int pismenko[4] = { 0b011, 0b101, 0b001, 0b110 };//přednastavené
    stavP1 = (stavP1 & ~(pismenko[patro - 1] << 0)); //hodnoty hw
    outb(stavP1, 0x300);
    zpozdeni(0, 1000000);
}

```



```

void main()
{
    //přístup k dvěma adresám 0x301 a 0x300
    ioperm(0x300, 2, 1);
    outb(255, 0x301); //nastavení portu do log.1
    outb(255, 0x300);
    while (1)
    {
        //začátek programu nalezení přízemí
        nalezeniprizemi();
        obsazeni();
        ledpatro();
        if (preruseni == 1)
        {
            preruseni = 0;
            break;
        }
    }
    while (1)
    { //hlavní smyčka
        int stav1 = inb(0x300);
        int stav2 = inb(0x301);
        for (int i = 0; i < 10; i++) //smyčka aby to nebylo moc rychle
        {
            ledpatro();
            stavtlacitek();
            stav_vytahu();
            predpohyb();
            obsazeni();
        }
        outb(255, 0x301); //nulování P2(motor)
        stavP1 = (stavP1 | (0b111 << 0)); //nulování LED
        outb(stavP1, 0x300);
    }
}

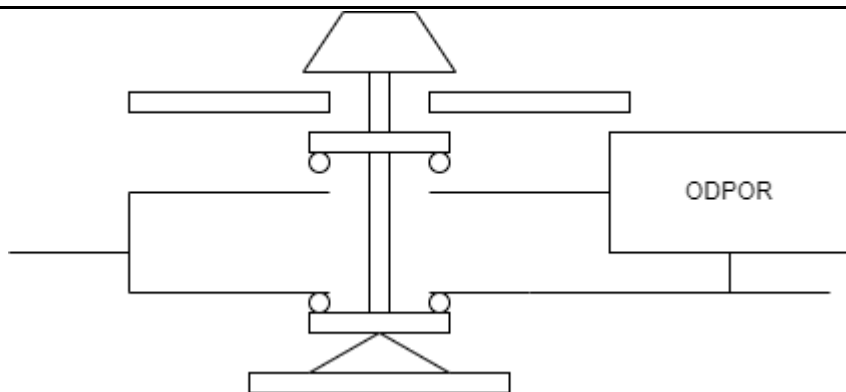
```

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.

RS-485 je sériová komunikace obsahující dva vodiče(A a B), může mít délku až 1.2km, ale to má vliv na přenosovou rychlost např při 10m je přenos 10Mib/s. Logická 1 nastává když je rozdíl A a B -200mV a log 0 +200mV, přenos dat se uskutečňuje pomocí 8bitových rámců se startbitem, jedním nebo více stopbity a případně i paritním bitem.
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).

Na každém patře by bylo tlačítko s unikátním odporem(třeba by rostl s každým dalším patrem), které by při stisku přerušilo obvod a zapojilo odpor do obvodu. Při zapojení odporu by došlo k poklesu napětí, které by bylo převedeno pomocí ADC na číslo, které se porovná s nějakým rozsahem pro každé patro.



Závěr

Dokumentace úlohy dopadla docela dobře, až na jednu otázku která nebyla zodpovězena. Kód byl upraven aby splnil chybějící požadavky,

Odkazy

<https://cs.wikipedia.org/wiki/RS-485>