

# PRAXE

5	STŘEDNÍ PRŮMYSLOVÁ ŠKOLA CHOMUTOV	<i>Patrik Příhoda</i>
28.3.2022	Výtah	V4

# Zadání

- Vytvořte program, který bude ovládat výtah
- Program by měl umět:
  - Když je potřeba rozsvěcovat světlo v kabině
  - Výtah jezdí do patra po přivolání tlačítkem
  - V kabině se zobrazuje LED šipka směru ve kterém výtah jede
  - Zobrazení daného patra
  - Výtah jezdí pouze se zavřenými dveřmi
  - Zvukový signál po přijetí do patra

## Teorie

- Napájení – 12Vss/1A
- Vstupy a Výstupy: všechny jsou v úrovních TTL, aktivní v log. 0
- Motor je řízen dvěma signály, kdy jeden ze signálů spouští motor a druhý volí směr
- Je potřeba číst z portů a i na ně zapisovat (inb, outb)
- Pro správnou funkčnost je potřeba práce s bity (zjištění pozice, přepsání pouze daného bitu...)

## Popis programu

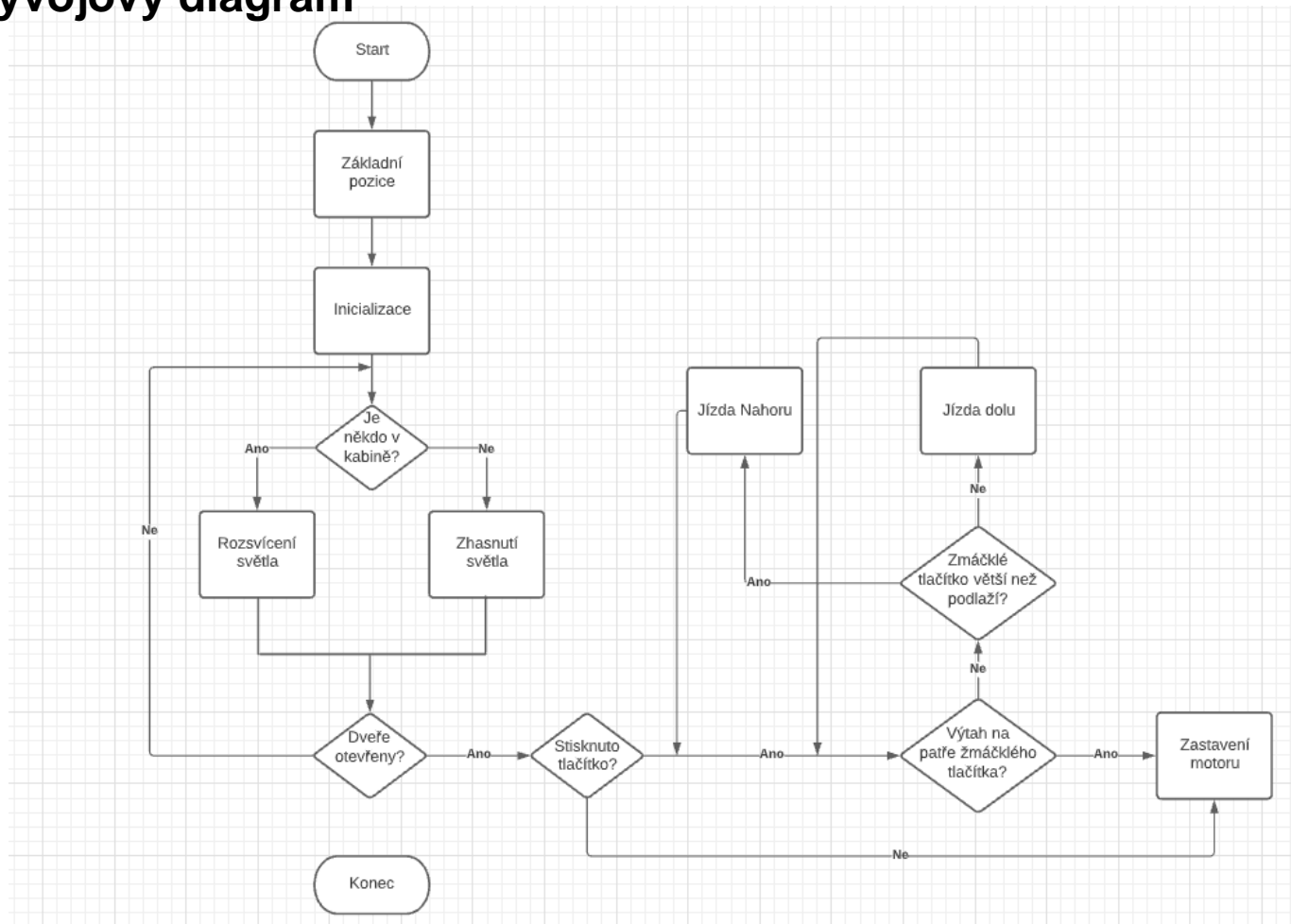
- Po spuštění programu se kabina výtahu nastaví do základní pozice (zakladniPozice())=> 1. patro. Výtah jede nahoru 4 vteřiny, zastaví, a poté jede dolů než najde patro. Zjistí jestli je v prvním patře a pokud ne tak opět jede dolů, dokud nezastaví v prvním patře. Proběhne Inicializace. Program čte adresu P4 a bit, kterým se zjišťuje pozice dveří. Jsou-li otevřené dveře výtah nereaguje na stisknutí tlačítek. Pomocí metody zjistíKabinu() program přečte adresu P4 a pokud je daný bit nastavený v 0 (člověk je v kabině) rozsvítí se světlo pomocí metody rozsvitSvetlo(), nebo naopak zhasne zhasniSvetlo(). Tlačítka fungují tak, že program čte adresu P3 a čte jednotlivé bity (zjistíTlacitka()) a stejně tak senzory podlahy (zjistíPodlazi()). Hodnoty se uloží do pomocných proměnných a porovnávají se. Pokud je tlačítko větší než podlaží tak výtah pojede nahoru (jizdaNahoru()), pokud menší tak dolů (jizdaDolu()) a pokud se rovnají výtah zastaví (jizdaStop()).

## Rozbor proměnných a funkcí (metod)

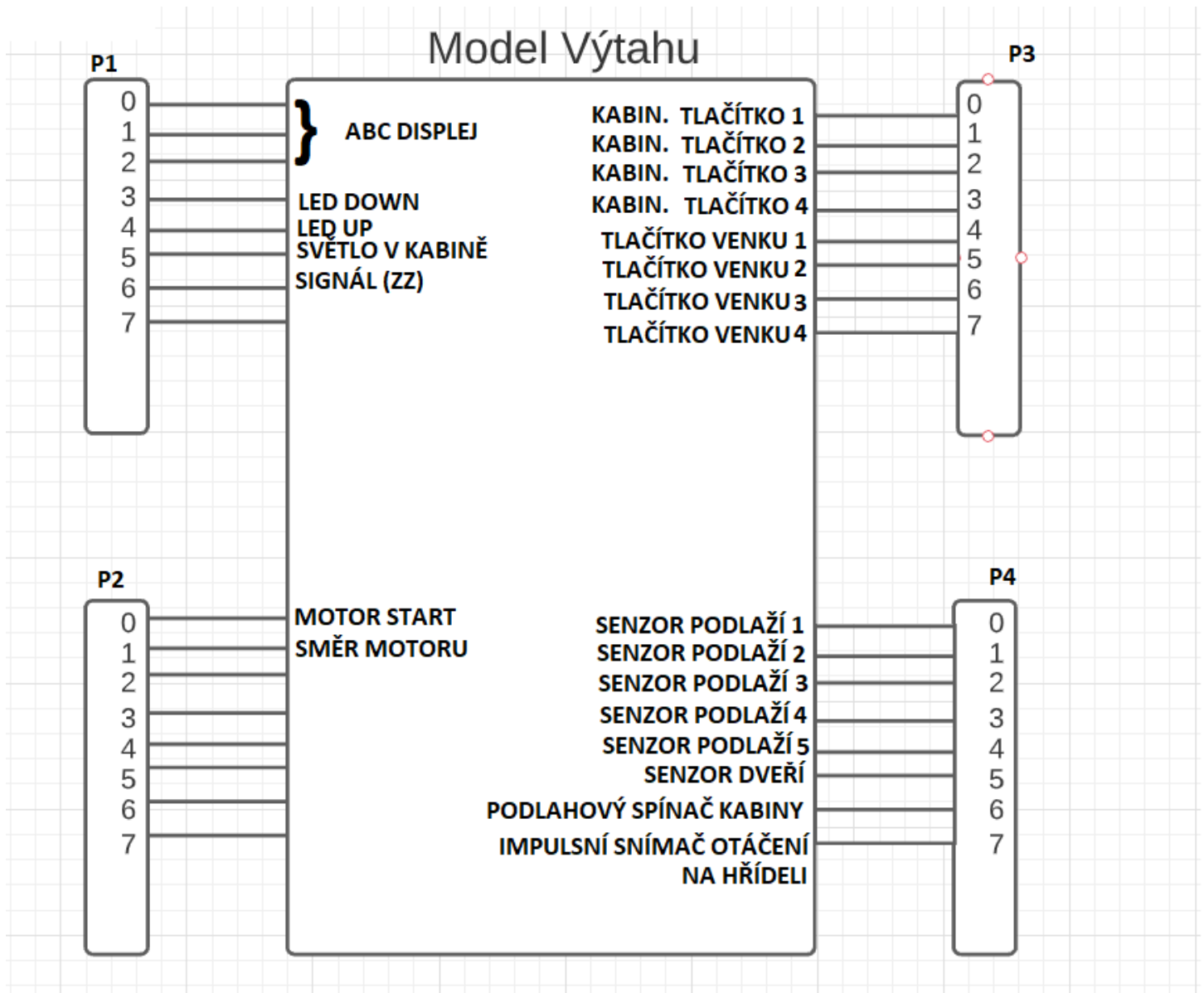
int	bitlsClear(int hodnota, int bitno)	Zjistí hodnotu bit na určité pozici
int	hodnota	Hodnota bitu
int	bitno	Pozice bitu
int	zjistíTlacitko()	Zjistí bity na pozicích kde jsou tlačítka v kabině i venku na adrese P3
void	inicializace	Inicializace
int	otevriDvere()	Zjistí bity na pozici dveří v adrese P4
int	zjistíPodlazi()	Zjistí bity na pozicích kde jsou senzory podlaží na adrese P4
void	zapniSirenu()	Zapne sirénu
void	vypniSirenu()	Vypne sirénu
void	zhasniSvetlo()	Zhasne světlo v kabině
void	rozsvitSvetlo()	Rozsvítí světlo v kabině
void	zjistíKabinu()	Zjistí hodnotu bitu na pozici kde je senzor v kabině na adrese P4
void	jizdaNahoru()	Zapne motor a směr otáčení nahoru
void	jizdaDolu()	Zapne motor a směr otáčení dolů
void	delay(int sec)	Zpodění

void	zakladniPozice()	Nastaví výtah do základní pozice => 1. patro
void	jizdaStop()	Vypne motor
void	main()	Hlavní část programu
int	vytahJede	Pomocná proměnná, pokud výtah jede je 1, jinak je 0
void	jizdaStop()	Vypne motor
int	main(void)	Hlavní část programu
int	podlazi	Hodnota podlaží
int	tlacitko	Hodnota tlačítka

## Vývojový diagram



# Schéma zapojení



## Komentovaný výpis program

```
#include"sim_io.h"
#include"sim_io.c"
#include"virtual_hw.c"
#include"vh-elevator.c"
#include"time.h"
#include<stdio.h>

static int motornahoru = 0b1111110;
static int motordolu = 0b11111100;
static int stop = 0xFF;
static int sirena = 0b10111111;
static int vytahJede = 0;
static int podlazi = 0;
static int tlacitko = 0;
static const int adresaP1 = 0x300;
static const int adresaP2 = 0x301;
static const int adresaP3 = 0x300;
static const int adresaP4 = 0x301;
```

```

// zjistí hodnotu bitu na pozici
int bitIsClear(int hodnota, int bitno)
{
    if ((hodnota & (1 << bitno)) == 0)
        return 1;
    else
        return 0;
}

// nastaví požadovaný bit do 1
int setBit(int hodnota, int bitno)
{
    hodnota |= (1 << bitno);
    return hodnota;
}

// nastaví požadovaný bit do 0
int clearBit(int hodnota, int bitno)
{
    hodnota &= ~(1 << bitno);
    return hodnota;
}

int zjistiTlacitko() // zjistí bity na pozicích kde tlačítka v kabine i venku na
adrese P4
{
    int stav = inb(adresaP3); // tlačítka

    if (bitIsClear(stav, 0) | bitIsClear(stav, 4))
        return 1;
    else if (bitIsClear(stav, 1) | bitIsClear(stav, 5))
        return 2;
    else if (bitIsClear(stav, 2) | bitIsClear(stav, 6))
        return 3;
    else if (bitIsClear(stav, 3) | bitIsClear(stav, 7))
        return 4;
    return 0;
}

void inicializace()
{
    outb(0xFF, 0x301);
    outb(0xFF, 0x300);
}

int otevriDvere() // zjistí bity na pozici dveří na adrese P4
{
    int stav = inb(adresaP4);

    if (bitIsClear(stav, 5))
        return 1;
    else
        return 0;
}

int zjistipodlazi() // zjistí bity na pozicích kde jsou senzory podlahy na adrese
P4

```

```

{
    int stav = inb(adresaP4); //senzory

    if (bitIsClear(stav, 0))
        return 1;
    else if (bitIsClear(stav, 1))
        return 2;
    else if (bitIsClear(stav, 2))
        return 3;
    else if (bitIsClear(stav, 3))
        return 4;
    return 0;
}

void zapniSirenu()
{
    int stav = inb(adresaP1);
    outb(clearBit(stav, 6), adresaP1);
}

void vypniSirenu()
{
    int stav = inb(adresaP1);
    outb(setBit(stav, 6), adresaP1);
}

void rozsvitSvetlo()
{
    outb(0b11011111, adresaP1);
}

void zhasniSvetlo()
{
    outb(setBit(0b11111111, 5), adresaP1);
}

int zjistKabinu() // zjist bit co ovlada senzor v kabine na adrese P4
{
    int stav = inb(adresaP4);

    if (bitIsClear(stav, 6))
        return 1;
    else
        return 0;
}

// ledky se svetlem zaroven se sekaj
/*
void ledVypnout()
{
    outb(setBit(0b11101111, 4), adresaP1);
    outb(setBit(0b11101111, 3), adresaP1);
}

void smerDolu()
{
    outb(clearBit(0b11111111, 3), adresaP1);
}

```

```

void smerNahoru()
{
    outb(clearBit(0b11111111, 4), adresaP1);
}
*/

void jizdaNahoru()
{
    outb(motornahoru, adresaP2);
    //smerNahoru();
}

void jizdaDolu()
{
    outb(motordolu, adresaP2);
    //smerDolu();
}

void jizdaStop()
{
    outb(stop, adresaP2);
    //ledVypnout();
}

void delay(int sec)
{
    struct timespec t;
    t.tv_sec = sec;
    t.tv_nsec = 0;
    nanosleep(&t, NULL);
}

void zakladniPozice() // vytah sjede do prvnio patra po zacatku
{
    jizdaNahoru();
    delay(4);
    jizdaStop();
    jizdaDolu();

    while(podlazi == 0)
    {
        if (zjistiPodlazi() != 0)
        {
            jizdaStop();
            podlazi = zjistiPodlazi();
        }
    }

    jizdaDolu();

    while(1)
    {
        if (zjistiPodlazi() != 0)
        {
            podlazi = zjistiPodlazi();
        }
        if (!vytahJede && podlazi > 1)

```

```

        {
            jizdaDolu();
            vytahJede = 1;
        }
        else if (podlazi == 1)
        {
            jizdaStop();
            vytahJede = 0;
            break;
        }
    }
}

int main(void)
{
    if (ioperm(0x300, 2, 1))
    {
        printf("Neni pristup k portum P1 a P3");
        return -1;
    }
    if (ioperm(0x301, 2, 1))
    {
        printf("Neni pristup k portum P2 a P4");
        return -1;
    }

    zakladniPozice();
    //outb(0xFF, adresaP3);
    inicializace();

    podlazi = 1;
    tlacitko = 1;
    while(1)
    {
        if (otevriDvere() == 1)
        {
            if (zjistiKabinu() == 1)
            {
                rozsvitSvetlo();
            }

            if (zjistiKabinu() == 0)
            {
                zhasniSvetlo();
            }

            if (zjistiTlacitko() != 0)
            {
                tlacitko = zjistiTlacitko();
            }
            if (zjistiPodlazi() != 0)
            {
                podlazi = zjistiPodlazi();
            }

            if (tlacitko > podlazi)
            {

```



```

        jizdaNahoru();
        vytahJede = 1;
    }
    else if (tlacitko < podlazi)
    {
        jizdaDolu();
        vytahJede = 1;
    }
    else
    {
        //tlacitko = 0;
        jizdaStop();
        //zapniSirenu();
        //delay(2);
        //vypniSirenu();
        vytahJede = 0;
    }
}
}
}

```

## Odpovědi na otázky

- Pro řízení pohonů větších výkonů není ekonomické vymýšlet vlastní řešení ovládání. Jaké komerčně dostupné zařízení byste pro číslicové ovládání takových motorů použili?**
- 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.**
  - CAN - Controller-Area Network je standardně navržený speciálně pro automobilový průmysl, ale nyní se používá také v dalších oblastech, jako jsou průmyslové automatizace a lékařské vybavení.
  - CAN. Tento protokol definuje jen vlastní fyzickou vrstvu a je typický pro několikanásobné zabezpečení rychlého přenosu dat v linkové vrstvě. Nemá definovanou aplikační vrstvu a aplikaci ponechává plně na uživateli.  
(viz<sup>1</sup>)
- 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

- Úloha se mi v rámci možností podařila. Jsem si vědom chyb např. není ošetřené zmáčknutí více tlačítek a zmáčknutí tlačítka venku, když je osoba uvnitř, nebo to že není možné spustit věci na P1 současně

---

<sup>1</sup>