

# PRAXE

2

STŘEDNÍ PRŮMYSLOVÁ ŠKOLA CHOMUTOV

Kindl Tomáš

22.3.2022

Robot

V4

# Zadání

Vytvořte program pro robota.

Program bude obsahovat:

- Ovládání robota přes ovladač
- Nalezení výchozí pozice
- Přehrát naučený program

## Teorie

Budeme pracovat v Linuxu. Použijeme bitovou knihovnu pro ulehčení práce. Robota budeme napájet +5V/0,5A pro logiku a +12V/1,5A pro motory. Maximální taktovací kmitočet je 450Hz (střída 1:1). Motory jsou aktivní v log. 0. Motorů může běžet více najednou, ale stejným směrem. Výstupy závor IR mají úroveň log. 0. a na panelu jsou indikační LED a ty svítí, pokud clonka prochází závorou. Výchozí stav je pokud jsou všechny závory zcloněny. Pokud aktivujeme základnu a na směr otáčení přivedeme log. 1 tak se pohybujeme po směru hodinových ručiček, pokud na směr otáčení přivedeme log.0 tak se otáčíme proti směru hod. ručiček. Když aktivujeme hlavní rameno tak směr otáčení v log. 1 bude dolů a v log. 0 nahoru. Po aktivaci motoru chapadla a přivedení log.1 na směr otáčení se chapadlo zavře a v log. 0 se otevře. A u ramena s chapadlem to funguje tak že v log.1 jde nahoru a v log. 0 dolu (směr otáčení).

## Popis programu

Po zapnutí programu se načte výchozí pozice ta se hledá tak, že nejprve zkusí jet doleva (určitý počet kroků vypočtený) jestli nenajde výchozí pozici a pokud nenajde tak pojedou druhým směrem, když najde výchozí pozici. Tak se spustí cyklus while, který běží nekonečně dlouho.

Poté máme další cykly while (podmíněné). První se spustí while pro  $i == 1$ . Kde ovládám základnu a hlavní rameno. Základna a hlavní rameno se ovládají pomocí ovladače. To znamená že pokud, program zaregistruje zmáčkнутé tlačítko (funkce if) tak udělá akci. Pokud se zmáčkne tlačítko doprava (tak se nastaví na výstupu motory základny do log. 0 dále nastavíme směr otáčení a jako poslední pulzy. Pulz je nastavení tak že odešle log.0 poté nastane časové zpoždění a pošle log.1 a pak zase časové zpoždění). Časové zpoždění je odvozeno z maximálního taktovacího kmitočtu. Stejně takhle to trvá v pohybu doleva. A i s ovládáním hl. ramena nahoru a dolu vždy se jen změní jaký motor zapínáme a nastavíme směr otáčení (kam se chceme hýbat). Pokud zmáčkne hlavní tlačítko tak se proměnná  $i$  nastaví na 2 a aktivuje se druhý cyklus.

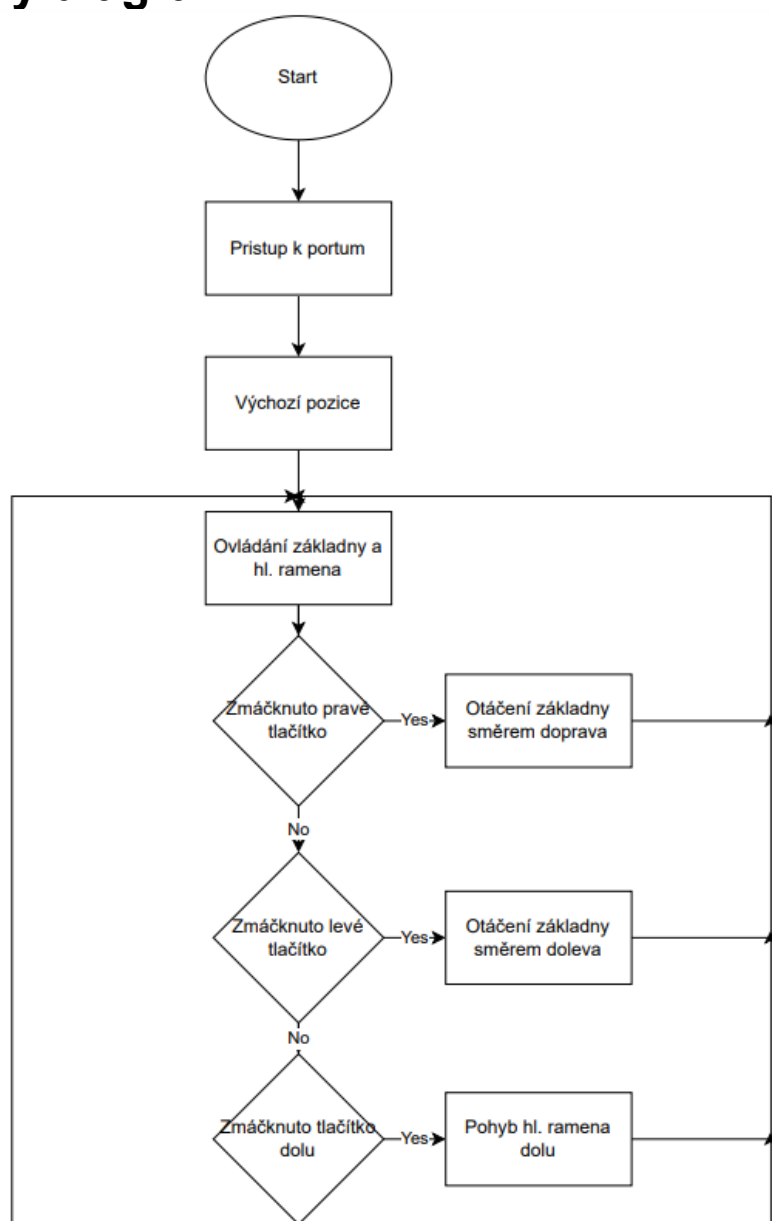
Cyklus while  $i == 2$ . V tomto cyklus ovládáme rameno s chapadlem. To funguje podobně jako v cyklu  $i == 1$ . Takže pokud zaregistrujeme že se zmáčklo tlačítko (pomocí if) nahoru tak se aktivují motory ramena s chapadlem nastavíme směr otáčení odešleme log. 0 na pulz poté časové zpoždění pak log.1 a pak zase časové zpoždění. Směr i motory necháváme ve stejné log. Úrovni. Když se zmáčkne tlačítko hlavní, tak se nastaví proměnná  $i = 3$ .

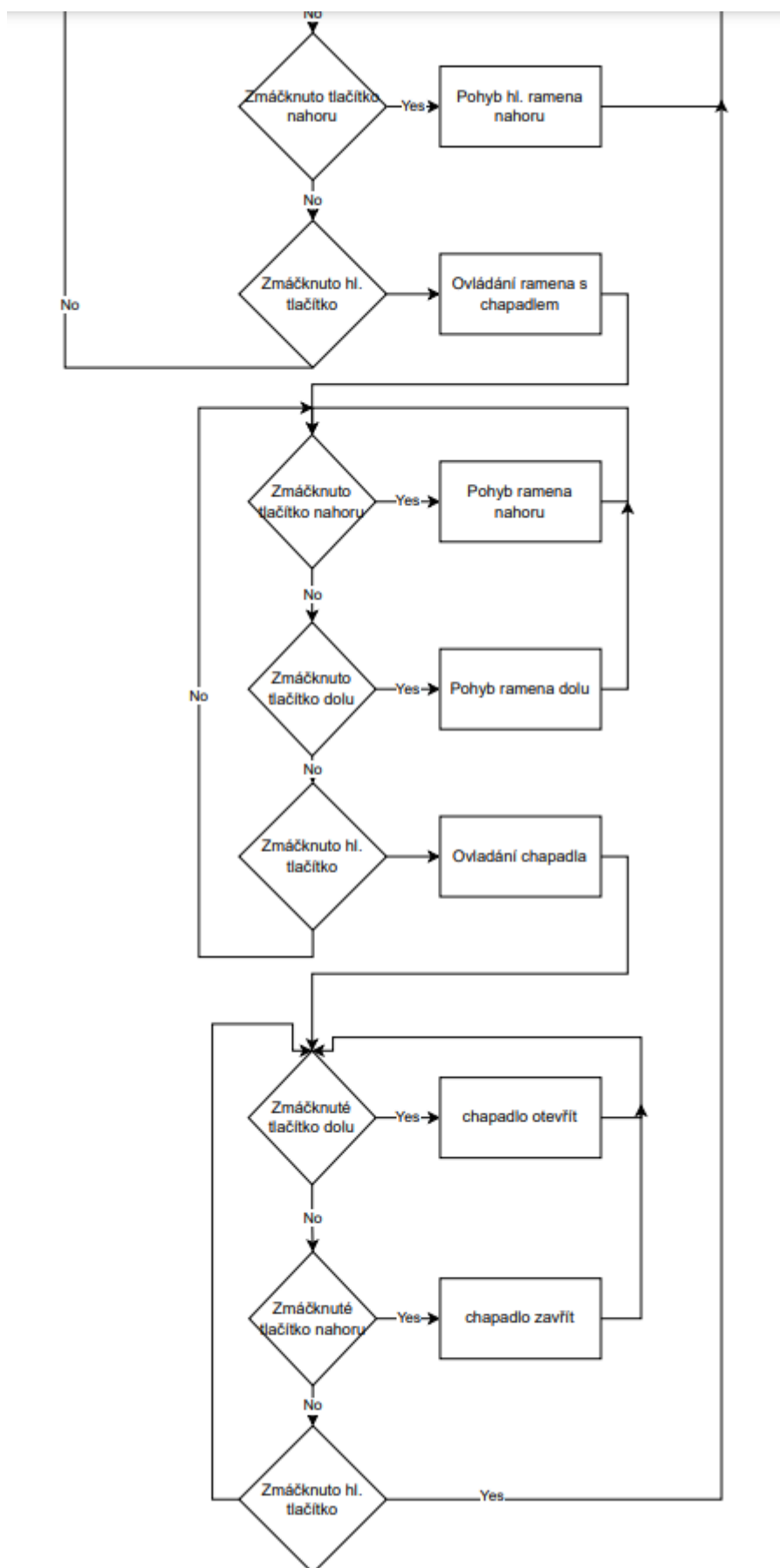
Cyklus while  $i == 3$ . Ovládá chapadlo, a to, jestli se zavře nebo otevře. Chapadlo se otevře tak že se zmáčkne na ovladači tlačítko nahoru(if) aktivuje se chapadlo a směr otáčení do log. 1 pulz do log. 0 poté zpoždění a pak pulz na log 1. To stejné s tlačítkem dolu akorát směr otáčení do log.0 a tím se chapadlo bude otevírat. Po zmáčknutí hlavní tlačítka je proměnná  $i = 1$  a znovu tedy můžeme ovládat základnu a hl. rameno.

## Rozbor proměnných a funkcí (metod)

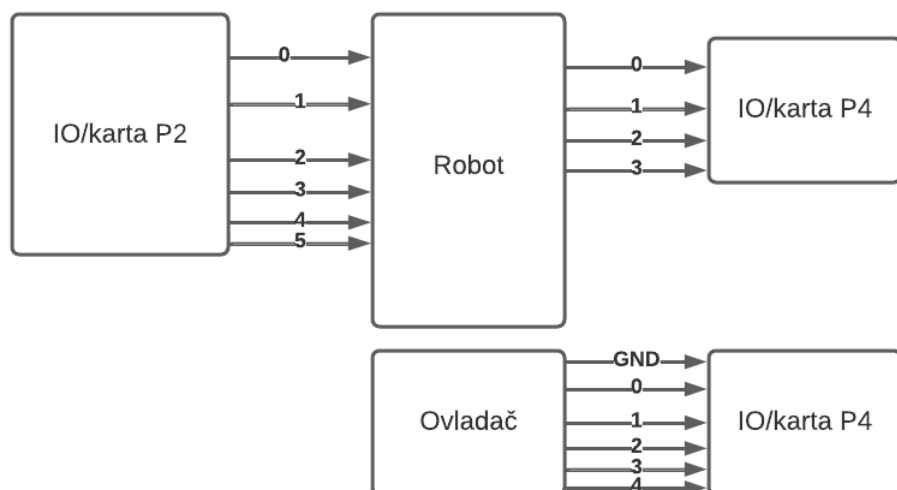
Funkce	proměnná	popis
void	main	hlavní cyklus programu
void	vych	Cyklus pro nalezení vych. pozice
Struct timespec	T,t2	Časová proměnná
Int	P4,p44	Čtení z portu p4
Int	lrz,lrzz	IR závora základny
Int	Pravetl,levetl,dolutl,nahorutl	Zjišťování, jestli jsou zmáčknutá tlačítka
Int	tl	Hlavní tlačítko zmáčknuté
int	i	Proměnná pro cykly while
Int	j	Proměnná pro cyklus for
int	P33	0x301

## Vývojový diagram





# Schéma zapojení



## P2

- 0 – motor hlavní rameno
- 1 – motor otáčení základny
- 2 – motor rameno chapadla
- 3 – motor chapadla
- 4 – směr otáčení
- 5 – taktovací pulzy

## P3

- 0 – pravý tl.
- 1 – levý tl.
- 2 – dolů tl.
- 3- nahoru tl.
- 4 – tlačítko hl.

## P4

- 0 – IR závora hl. rameno
- 1 – IR závora otáčení základny
- 2 - IR závora ramena chapadla
- 3 – IR závora chapadla

# Komentovaný výpis programu

```
#include <stdio.h>
#include <sys/io.h>
#include "bity.h" //bitová knihovna
#include <time.h>

int vych(void) //vychozi pozice
{
    struct timespec t;
    t.tv_sec = 0;
    t.tv_nsec = 2500000;
    struct timespec t2;
    t2.tv_sec=1;
    t2.tv_nsec = 0;
    int j;
    for(j=0;j<1200;j++) //počet kroků k nalezení výchozí pozice základny pokud začínáme úplně vlevo
    {
        int p4 = inb(0x301);
        int irz = bitisClear(p4,1);
        outb(0b11011101,0x301);
        nanosleep(&t,NULL);
        outb(0b11111101,0x301);
        if(irz == 1)
        {
            return 1; //ukončení funkce
        }
    }
    int c;
    for(c=0;c<2800;c++) //počet kroků k nalezení výchozí pozice základny pokud začínáme úplně vpravo
    {
        int p44 = inb(0x301);
        int irzz = bitisClear(p44,1);
        outb(0b11001101,0x301);
        nanosleep(&t,NULL);
        outb(0b11101101,0x301);
        if(irzz==1)
        {
            return 1; //ukončení funkce
        }
    }
}

int main(void)
{
    //definování časových zpoždění

    struct timespec t;
    struct timespec t2;
    t.tv_sec = 0;
    t.tv_nsec = 2500000;
    t2.tv_sec = 1;
    t2.tv_nsec = 0;

    //prístup k portum

    if(ioperm(0x300,2,1)!=0){
        printf("není přístup k portum 0x300");
        return 1;
    }
```

```

if(ioperm(0x301,2,1)!=0){
    ("neni pristup k portum 0x301");
    return 1;
}
vych(); //vychozí pozice
int i = 1;
while(1){ //hlavní cyklus
    while(i==1){ //podmineny cyklus
        int p4 = inb(0x301); //čtení z portu
        printf("hodnota p4 =%d ", p4);
        int p3 = inb(0x300);

        int pravetl = bitisClear(p3,0); //zmacknute prave tl.
        int dolutl = bitisClear(p3,2); //zmacknute dolu tl.
        int nahorutl = bitisClear(p3,3); //zmacknute nahoru tl.
        int levetl = bitisClear(p3,1); //zmacknute leve tl.
        int tl = bitisClear(p3,4); //zmacknute tlacitko hlavní
        int p33 = 0x301;

        if(pravetl ==1) //pohyb zaklady doprava
        {
//            printf("jedu doprava");
            outb(0b11011101,p33);
            nanosleep(&t,NULL);
            outb(0b11111101,p33);
        }
        else
        if(levetl==1) //pohyb zaklady doleva
        {
//            printf("jedu doleva");
            outb(0b11001101,p33);
            nanosleep(&t,NULL);
            outb(0b11101101,p33);
        }
        else
        if(dolutl ==1) //hl. rameno dolu
        {
//            printf("jedu dolu");
            outb(0b11011110,p33);
            nanosleep(&t,NULL);
            outb(0b11111110,p33);
        }
        else
        if(nahorutl ==1) //hl. rameno nahoru
        {
//            printf("jedu nahoru");
            outb(0b11001110,p33);
            nanosleep(&t,NULL);
            outb(0b11101110,p33);
        }
        else
        if(tl ==1) //zmacknute tlacitko
        {
            i = 2;
            nanosleep(&t2,NULL);
        }
    }
}

```

```

while(i==2) //ovladani ramena s chapadlem
{
    int p4 = inb(0x301);
    printf("hodnota p4 = %d", p4);
    int p3 = inb(0x300);
    int p33 = 0x301;
    int nahorutl = bitisClear(p3,3);
    int dolutl = bitisClear(p3,2);
    int tl = bitisClear(p3,4);
//    outb(0b11111111,p33);
        if(nahorutl==1) //rameno nahoru
        {
//            printf("chapadlo nahoru");
            outb(0b11011011,p33);
            nanosleep(&t,NULL);
            outb(0b11111011,p33);
        }
        else
        if(dolutl ==1) //rameno dolu
        {
//            printf("chapadlo dolu");
            outb(0b11001011,p33);
            nanosleep(&t,NULL);
            outb(0b111101011,p33);
        }
        else
        if(tl==1)
        {
            i = 3;
            nanosleep(&t2,NULL);
        }
    }
while(i==3) //ovladani chapadla
{
    int p4 = inb(0x301);
    printf("hodnota i = %d",p4);
    int p3 = inb(0x300);
    //zavrit
    int nahorutl = bitisClear(p3,3);
    //otevrit
    int dolutl =bitisClear(p3,2);
    int p33 = 0x301;
    int tl = bitisClear(p3,4);

    if(nahorutl==1) //chapadlo zavřít
    {
        outb(0b11010111,p33);
        nanosleep(&t,NULL);
        outb(0b11110111,p33);
    }
    else
    if(dolutl==1) //chapadlo otevřít
    {
        outb(0b11000111,p33);
        nanosleep(&t,NULL);
        outb(0b11100111,p33);
    }
}

```



```

else
    if(tl==1) //zmačknuté tlačítko
    {
        i=1; //zpět na ovladání základny
        nanosleep(&t2,NULL);
    }
}
}
}

```

## Odpovědi na otázky

1. Pro zvýšení přesnosti polohování se často používá tzv. mikrostepping. Pokuste se popsat základní princip této technologie. Jedná se o ovládání krokového motoru a mikrostepping znamená, že provádíme malé krůčky místo celých kroků, jak je zvykem u krokových motorů. Do motoru se odesílají částečné pulzy pomocí algoritmů a díky tomu se pohneme pouze o nějaký zlomek kroku.<sup>1</sup>
2. Pro pohodlnější řízení krokových motorů se často používají tzv. integrované drivery. Popište princip této rodiny obvodů a na jednom konkrétním zástupci vysvětlete nejdůležitější parametry.  
Využívá Spojení krokového motoru a komponenty pohodu do jednoho zařízení. To zajišťuje úsporný design, snížení kabeláže a nákladů (protože nekupujeme samostatné komponenty motoru a pohonu..) STM17Q-1AE - konkrétní motor s integrovaným driverem (12-48VDC, 2LED, 3 digitální vstupy, 1 dig. Výstup)<sup>2</sup>
3. V rámci 3D tiskárny máte pro svislé polohování použitý šroub se stoupáním závitu 1mm a krokový motor NEMA 17HS4401. Pokuste se odvodit nejmenší posun ve svislé ose této tiskárny (tip: nakreslete si obrázek se všemi parametry a podívejte se do katalogového listu uvedeného motoru).  
Motor má jeden krok 1,8 stupně. Šroub se stoupáním závitu 1mm. Nevím jaký by mohl být nejmenší posun.

## Závěr

Úlohu se mi podařilo udělat tak napůl, fungoval pohyb výchozí pozice jen základny a chybělo mi ukládání do souboru (přehrát naučený program).

<sup>1</sup> <https://www.motioncontroltips.com/faq-what-is-microstepping/>

<sup>2</sup> <https://www.applied-motion.com/products/integrated-steppers/stm17q-1ae>