PRAXE

8	STŘEDNÍ PRŮMYSLOVÁ ŠKOLA CHOMUTOV	Slapnička
1.2.2022	ROBOT	V.4

Zadání

Vytvořte program na ovládání stavového automatu, tak aby fungoval jako robot.

Funkce- Po spuštění nalezení základních pozic pomocí IR závor, když jsou aktivní, pro základnu, hl.rameno, vl.rameno a chapadlo. Dále vytvořit ovládání pomocí tlačítek ovladače, pohyb základny doprava a doleva. Pohyb ramen nahoru, dolu a zavíraní a otevíraní chapadla.

Teorie

Vnitřní fungování robota- Robot obsahuje krokové motory schopné otáčení na dvě strany(doleva log.0, doprava log.1) a spínané změnou taktovacího signálu z 0 na 1 nebo naopak(maximální takt je 450hz). Tyto motory jsou aktivovány log.0 a musí být aktivované před taktem, každá část obsahuje jeden motor, dohromady čtyři motory. Pro nalezení základní pozice jsou používány IR závory obsahující foto diodu, které vysílají log.1 dokud není přerušen signál clonkou nacházející se na každé části(když je přerušena tak vysílá log.0).

Kód je kompilován pomocí gcc a obsahuje knihovnu sys/io.h, v které používáme funkce outb a inb na vysílaní logických signálů. Po kompilaci musí být aplikace spuštěna jako správce, protože je ovládána vnější periferie.

Pro ovládání robota slouží ovladač s pěti tlačítky.

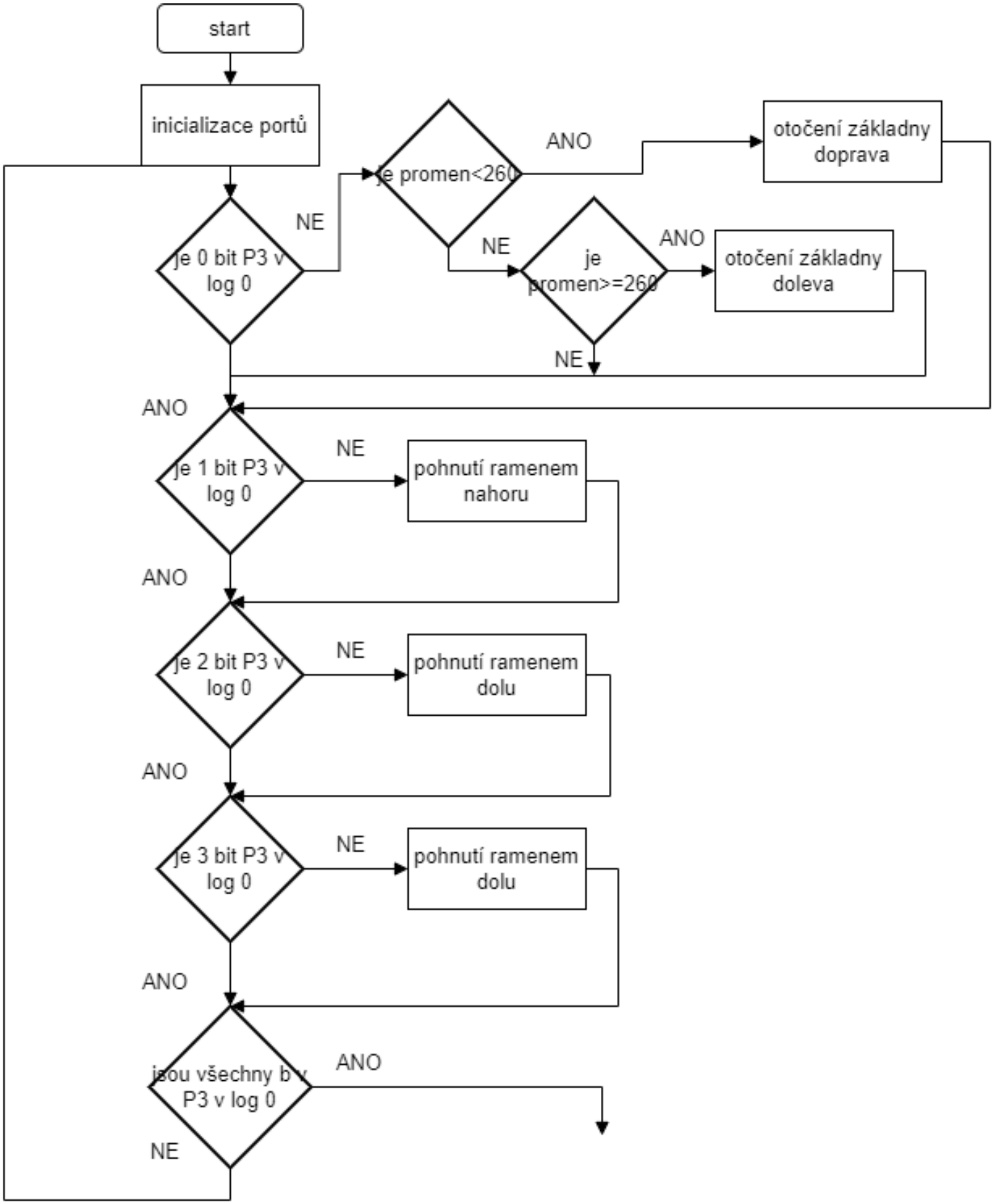
Popis programu

Po spuštění začne program deklarací vstupů a výstupů, poté začne smyčka pro nalezení základní polohy pomocí ir závor, tak aby všechny byli v log.0. Jako první je pohyb základny o 90 stupňů doprava a když není v tomto rozsahu, tak se základna otáčí doleva dokud nepřerušuje clonka závoru. Dále je hl.rameno, které se pohybuje směrem nahoru dokud také nepřeruší ir závoru, vl.rameno se pohybuje směrem dolu a chapadlo také směrem dolu. Po nalezení všech základních pozic je tato smyčka přerušena a začíná smyčka pro ovládání pomocí tlačítek. Ovladač obsahuje čtyři tlačítka(nahoru, dolu, doprava, doleva a attack), když není zmáčknuto tlačítko attack, tak je ovládána základna a hl.rameno. Tlačítkem doprava a doleva je ovládána základna a nahoru dolu hl.rameno, při stisknutí je ovládáno vl.rameno tlačítkem nahoru a dolu a chapadlo je ovládáno doleva(otevřít) a doprava(zavřít). Tahle smyčka běží furt dokola, než je aplikace ukončena.

Rozbor proměnných a funkcí (metod)

Typ	Název	popis
void	zpozdeni	Funkce pro zpoždění využívající nanosleep
void	CLK	Funkce pro taktovací signál, buď pro otočení motoru doprava nebo doleva
Int	promen	Proměnná pro otáčení základny o 90 stupňů doleva, podmínka běží dokud promen není v určité hodnotě(vyšlo cca 260 krát), tak aby se otočila základna relativně přesně.
Int	Stav1	Proměnná volající funkci inb(P3) a následná práce s ní pro nalezení základní polohy.
Int	Stav2	Proměnná volající funkci inb(P4) využívána ke zjištění stavu ovladače.
Int	maska	Proměnná uchovávající hodnotu 0b10000000
Int	main	Požádání o přístup k portům a spuštění první nekonečné smyčky pro nalezení základní pozice, když je nalezena je přerušena a začne druhá nekonečná smyčka pro práci s ovladačem.

Vývojový diagram



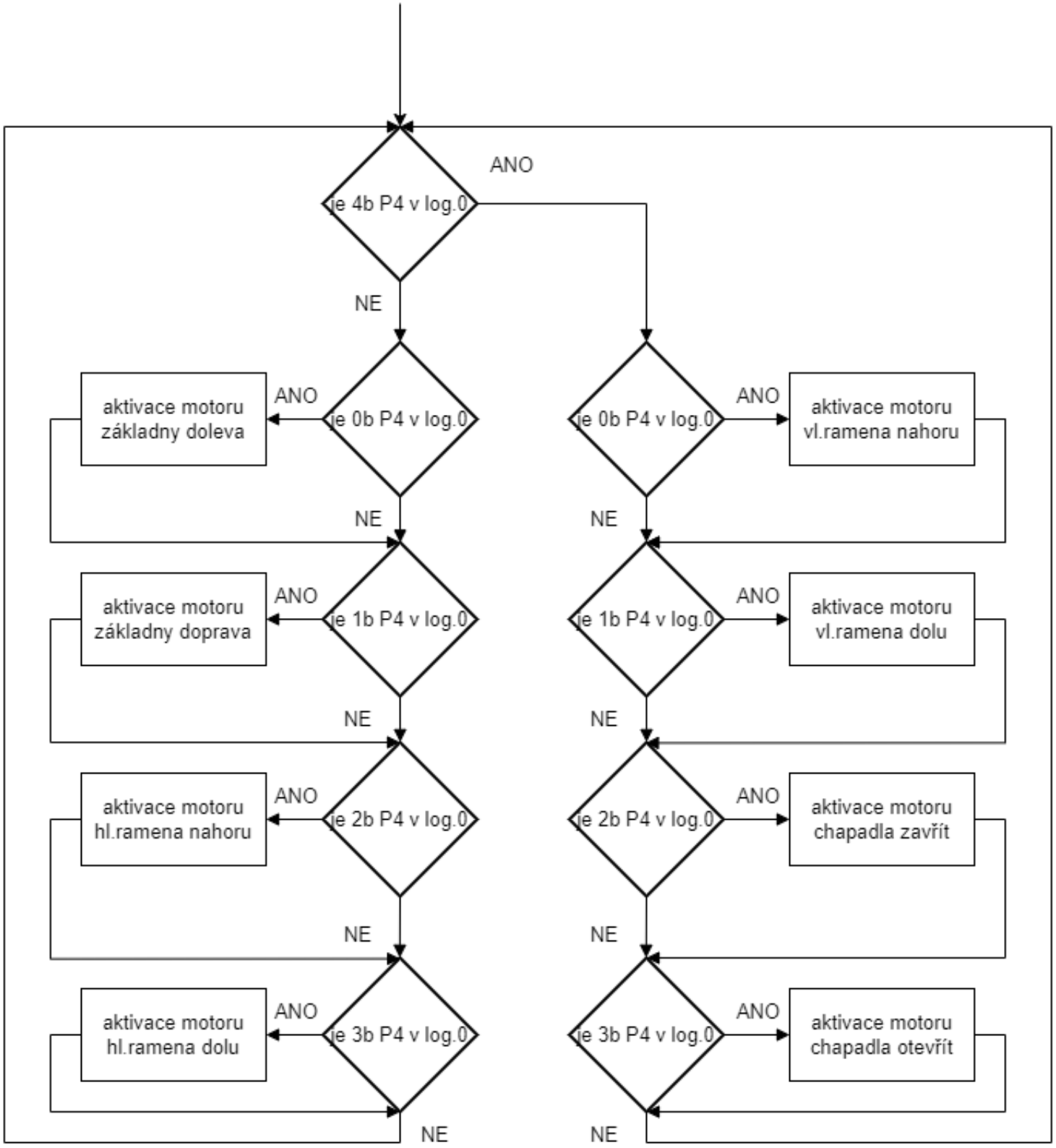
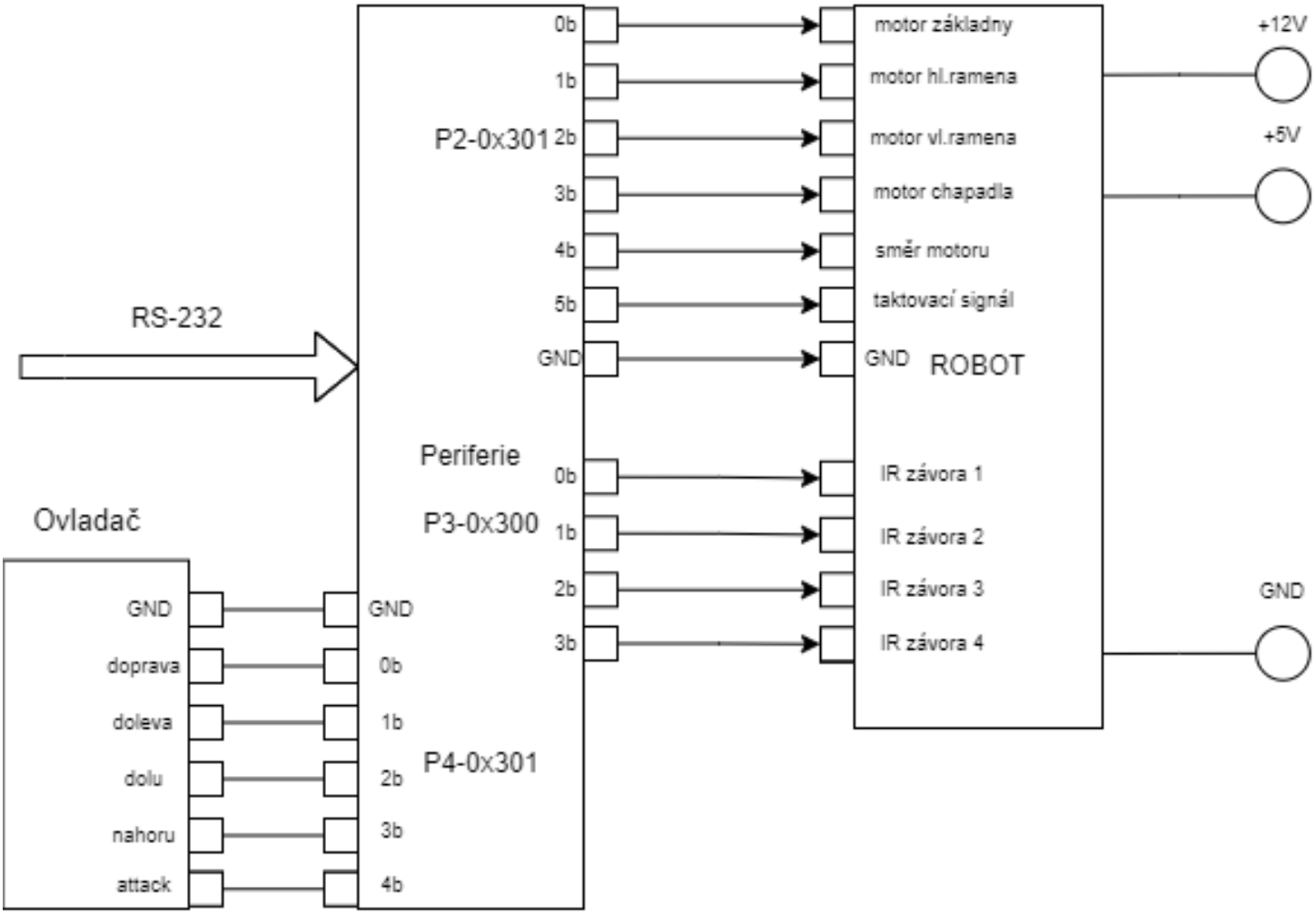

je 4b P4 v log.0
ANO
NE
je 0b P4 v log.0
aktivace motoru základny doleva
ANO
NE
je 0b P4 v log.0
aktivace motoru vl.ramena nahoru
ANO
NE
je 1b P4 v log.0
aktivace motoru základny doprava
ANO
NE
je 1b P4 v log.0
aktivace motoru vl.ramena dolu
ANO
NE
je 2b P4 v log.0
aktivace motoru hl.ramena nahoru
ANO
NE
je 2b P4 v log.0
aktivace motoru chapadla zavřít
ANO
NE
je 3b P4 v log.0
aktivace motoru hl.ramena dolu
ANO
NE
je 3b P4 v log.0
aktivace motoru chapadla otevřít
ANO
NE
NE

Schéma zapojení



Komentovaný výpis programu

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

//motor aktivni log0
//max kmitocet 450hz
//P2 b0-3 vyber casti b4 smer a b5 CLK
//P3 stav infradiod(0zakladna, 1 hl.rameno, 2 vl.rameno, 3 chapadlo)
//P4 pro ovladac 0b pravo, 1 levo, 2 dolu, 3 nahoru, 4 attack
//takt vychazi na 2.22ms

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

/*FILE *fptr;
fptr = fopen("stavy.txt","w+");*/

void CLK(int x)    //funkce pro takt na obě strany(doleva-doprava, nahoru-dolu)
{
       if(x==1)
       {
       outb(0b11011111,0x301);  //pátý bit v log.0 změna taktu a 4 v log.1 směr
                                  motoru
       zpozdeni(0,2420000);//sice teoretický delay vyšel 2,2ms(1000/450=2,222),
                           ale určil jsem
       }     //zpozdeni trochu delší, aby nedošlo k přeskakování kroku motoru
       if(x==0)
       {
       outb(0b11001111,0x301);//pátý bit v log.0 změna taktu a 4 v log.0 směr
                                 motoru
       zpozdeni(0,2420000);
       }
}

int main()
{
    ioperm(0x301,2,1);
    ioperm(0x300,2,1);
    int promen = 0;
    while(1)//smyčka pro nalezení základní pozice robota pozice se hledají
              najednou
    {
      int stav1 = inb(0x300);
      if(stav1 != (stav1 & (~(1 << 0))))//kontrola stavu základny dokud není v
      {                                        log.0
            if(promen>=260)//druhá fáze otočení když není clonka v 90 stupních
                           proti směru hodinových ručiček
            {      //tak je základna otáčena po směru dokud není nalezena
                  for(int i=0;i<5;i++)
                  {
                        outb(0b11101110,0x301);//aktivace motoru a směru
                        zpozdeni(0,2420000);
                        CLK(0);
                  }

            }
            if(promen<260)//otočení základny o 90 stupňů po směru jesli robot
            {               //není blízko clonky
```

```
                for(int i=0;i<5;i++)
                {
                        outb(0b11111110,0x301);
                        zpozdeni(0,2420000);
                        CLK(1);
                }
            }
    }

    if(stav1 != (stav1 & (~(1 << 1))))//nalezení clonky hl.ramena
    {//ir závora je nahoře, takže podmínka běží dokud se nezmění stav
                for(int i=0;i<5;i++)
                {
                        outb(0b11111101,0x301);//aktivace hl.ramena a směr nahoru
                        zpozdeni(0,2420000);
                        CLK(1);
                }
    }

    if(stav1 != (stav1 & (~(1 << 2))))//vl.rameno
    {//ir závora je vždy dole
                for(int i=0;i<5;i++)
                {
                        outb(0b11101011,0x301);//aktivace vl.ramena a směru dolu
                        zpozdeni(0,2420000);
                        CLK(0);
                }
    }
    if(stav1 != (stav1 & (~(1 << 3))))//chapadlo
    {
                for(int i=0;i<5;i++)
                {
                        outb(0b11100111,0x301);//aktivace chapadla a směru dolu
                        zpozdeni(0,2420000);
                        CLK(0);
                }
    }

    if((stav1 != (stav1 | (1 << 0))) && (stav1 != (stav1 | (1 << 1))) && (stav1 !=
(stav1 | (1 << 2))) && (stav1 != (stav1 | (1 << 3))))
    {//pokud jsou všechny ir závory v log 0 je smyčka while přerušena
      break;
    }
    promen++;//promena pro základnu
}

    while(1)//hlavní smyčka pro ovládání pomocí tlačítek
    {
      int stav2 = inb(0x301);
    if(stav2 !=( stav2 & (~(1<<4))))//kdyz neni aktivni tlacitko attack
    {//tak je ovládána základna a hl.rameno
      if((stav2 != (stav2 | (1 << 0))))//podminka pro
      {             //tlacitko do prava zakladny
            outb(0b11111110,0x301);//aktivace motoru základny a směr
                                   doleva
            zpozdeni(0,2420000);
            CLK(1);
      }
```

```
      if((stav2 != (stav2 | (1 << 1))))//podminka pro
      {            //tlacitko do leva zakladny
            outb(0b11101110,0x301);//směr doprava
            zpozdeni(0,2420000);
            CLK(0);
      }
      if((stav2 != (stav2 | (1 << 2))))//podminka pro
      {            //tlacitko dolu hl.ramena
            outb(0b11111101,0x301);//hl.rameno a směr nahoru
            zpozdeni(0,2420000);
            CLK(1);
      }
      if((stav2 != (stav2 | (1 << 3))))//podminka pro
      {                                  //tlacitko nahoru hl.ramena
            outb(0b11101101,0x301);//směr dolu
            zpozdeni(0,2420000);
            CLK(0);
      }
     }


    //kdyz je tlacitko attack stisknuto
    if(stav2 != ( stav2 | (1<<4)))
    {//je ovládáno vl.rameno a chapadlo
      if((stav2 != (stav2 | (1 << 0))))//podminka pro
      {     //tlacitko nahoru vl.ramena
            outb(0b11111011,0x301);//vl.rameno a směr nahoru
            zpozdeni(0,2420000);
            CLK(1);
      }
      if((stav2 != (stav2 | (1 << 1))))//podminka pro
      {     //tlacitko dolu vl.ramena
            outb(0b11101011,0x301);//směr dolu
            zpozdeni(0,2420000);
            CLK(0);
      }
      if((stav2 != (stav2 | (1 << 2))))//podminka pro
      {     //tlacitko dolu chapadla
            outb(0b11110111,0x301);//aktivace chapadla a zavření
            zpozdeni(0,2420000);
            CLK(1);
      }
      if((stav2 != (stav2 | (1 << 3))))//podminka pro
      {     //tlacitko nahoru chapadla
            outb(0b11100111,0x301);//otevření
            zpozdeni(0,2420000);
            CLK(0);
      }
    }
  }
}
```

Odpovědi na otázky

1. Pro zvýšení přesnosti polohování se často používá tzv. mikrostepping. Pokuste se popsat základní princip této technologie.

Každý krokový motor se posouvá po krocích a tyto kroky rozdělují jedno otočení motoru o 360 stupňů na určitý počet kroků. Tyto kroky je možné dále rozdělit na mikrosteps, aby došlo k hladšímu posunu motoru pomocí pulzně šířkové modulace. Když je proveden normální krok o určité době, tak motorem protéká proud, což vytvoří el.mag. pole a motor se otočí na případnou stranu.

Mikrostepping tento krok rozdělí a dochází k vypínaní a zapínání proudu během tohoto kroku, to zajistí

hladší posun motoru a zmenšení zvuku. Nevýhodou je že při zvyšování počtu mikrokroků dochází k exponenciálními poklesu točivého momentu motoru(2-70% původní, 3-38%), take není možno zvyšovat kroky donekonečna, protože ložisko motoru má také nějaké vlastní tření a kdyby nebylo el.mag. pole dostatečně silné motor by se nehýbal.

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.

   Používá se pro práci s proudem, který ním teče, nebo pro ovládání jiné součástky

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).

Tento motor má jeden krok 1,8stupňů(dohromady 200), při tomto stoupaní se pohne tiskárna o 0,005mm(1,8*1/360) při nejmenším kroku (stoupání 1mm po otočení o 360stupních na závitu).

Závěr

Úloha byla skoro vyřešena až na zapisování pohybů do textového souboru, případně čtení pohybů. Nalezení polohy není odzkoušené na reálném hardwaru, nemusí být přesné. Odpověď číslo dva není úplně zpracována a kód by mohl více využívat funkcí, také mě napadlo zda by nebylo možné využít Mikrostepping pro hladší pohyb(za cenu pomalejšího otáčení). Robot není programově ohraničen, aby zastavil před fyzickými zábranami.

Odkazy

https://www.linearmotiontips.com/microstepping-basics/
https://www.faulhaber.com/en/support/technical-support/motors/tutorials/stepper-motor-tutorial-microstepping-myths-and-realities/