

```

/*
 * Ridici software robotu "AlcBot 1.0"
 * Robot slouzi k rozlevani napoju a vznikl jako diplomova prace na FSI
VUT Brno
 * Robot je majetkem firmy CUTTER Systems spol. s r. o.
 * Autor diplomove prace - Michaela Tejchmanova
 * Verze software - v1.4
*/

#include <Servo.h>
#include <Keypad.h>

//----- nastaveni promennych -----

//promenna pro cteni aktualniho casu (v ms)
long curTime;
long prevTimeC = 0;
long prevTimeZ = 0;
long prevTimeX = 0;
long prevTimeA = 0;

//promenne ridici zda-li je dokoncen HOME cyklus
int stepDone;
int homeDone = 1;
int startDone = 1;

int homeCdone = 0;
int homeZdone = 0;
int homeXdone = 0;

int senzorCon;
int senzorZon;
int senzorXon;

int stopON = 0; //promenna pro funkci stop()

long z; //promenne pro funkci start()
long x;
long c;
long a;

//promenne pro RS funkci otaceni motoru
int motorC = 0;
int motorZ = 0;
int motorX = 0;
int motorA = 0;

//prodleva mezi kroky (v microsec) - ridi rychlost otaceni krokovych motoru
long delayC = 1000;
long delayZ = 150;
long delayX = 100;
long delayA = 300;

//definovani servomotoru
Servo servoUchop;
int servoAngle;

//definovani klavesnice
const byte rows = 3; //pocet radku na klavesnici
const byte cols = 4; //pocet sloupcu na klavesnici
char Keys[rows][cols] = {

```

```

    {'C', 'c', 's', 'k'},
    {'Z', 'z', 'S', 'x'},
    {'X', 'x', 'H', 'x'}
};

byte rowPins[rows] = {2, 3, 4};
byte colPins[cols] = {5, 6, 7, 8};
Keypad myKeypad = Keypad(makeKeymap(Keys), rowPins, colPins, rows, cols);

//souradnice lahve a panaku
int sourX[1] ;
int sourY[1] ;

//definovani IO pinu
int dirC = 40; //STEP a DIR piny pro osy
int stepC = 30;

int dirZ = 42;
int stepZ = 32;

int dirX = 44;
int stepX = 34;

int dirA = 46;
int stepA = 36;

int enableC = 22; //piny pro zapinani a vypinani motoru
int enableZ = 24;
int enableX = 26;
int enableA = 28;

int servoControl = 47; //ovladani serva

int ledB = 10; //LED piny
int ledG = 11;
int ledW = 12;

int interruptPin = 20; //interrupt pin pro STOP tlacitko

int senzorC = 48; //piny indukcnich cidel
int senzorZ = 50;
int senzorX = 52;

// ----- setup -----

void setup() {
  interrupts(); //povoli funkci Interrupt
  analogWrite(ledW,15); //zapnuti bile LED

  //krokove motory
  pinMode(dirC, OUTPUT); //dir osa C
  pinMode(stepC, OUTPUT); //step osa C

  pinMode(dirZ, OUTPUT); //dir osa Z
  pinMode(stepZ, OUTPUT); //step osa Z

  pinMode(dirX, OUTPUT); //dir osa X
  pinMode(stepX, OUTPUT); //step osa X

  pinMode(dirA, OUTPUT); //dir osa A
  pinMode(stepA, OUTPUT); //step osa A

```

```

pinMode(enableC, OUTPUT);
pinMode(enableZ, OUTPUT);
pinMode(enableX, OUTPUT);
pinMode(enableA, OUTPUT);

digitalWrite(enableC, LOW);
digitalWrite(enableZ, LOW);
digitalWrite(enableX, LOW);
digitalWrite(enableA, LOW);

pinMode(23, OUTPUT); //mikrokrokovani
pinMode(25, OUTPUT);
digitalWrite(23, HIGH); //mikrokrokovani zapnout
digitalWrite(25, HIGH);

//cidla
pinMode(senzorC, INPUT);
pinMode(senzorZ, INPUT);
pinMode(senzorX, INPUT);

//servomotor
servoUchop.attach(servoControl); //uchop - defaultne otevreno
servoAngle = 0;
servoUchop.write(servoAngle);

//klavesnice
myKeypad.setHoldTime(500); //cas po kterem se detekuje stav HOLD v milisek
myKeypad.addEventListener(keyEvent); //EventListener pro klavesnici

pinMode(ledG, OUTPUT);
pinMode(ledB, OUTPUT);
pinMode(ledW, OUTPUT);

//interrupt - STOP tlacitko
pinMode(interruptPin, INPUT_PULLUP);
attachInterrupt(digitalPinToInterrupt(interruptPin), stop, FALLING);
//pri detekci na interrupt pinu, spust funkci stop()

//seriova komunikace - otevri (pro monitoring)
Serial.begin(9600);

}

// ----- funkce zapnuti motoru -----
void stepperOn(){
    digitalWrite(enableC, HIGH);
    digitalWrite(enableZ, HIGH);
    digitalWrite(enableX, HIGH);
    digitalWrite(enableA, HIGH);
}

// ----- funkce vypnuti motoru -----
void stepperOff(){
    digitalWrite(enableC, LOW);
    digitalWrite(enableZ, LOW);
    digitalWrite(enableX, LOW);

```

```

    digitalWrite(enableA, LOW);
}

//----- funkce krokovani motoru -----
void stepperC(int delayC) {
    stepDone = 0;
    while (stepDone == 0){
        curTime = micros();
        if (curTime - prevTimeC >= delayC){
            prevTimeC = curTime;
            if (digitalRead(stepC) == LOW){
                digitalWrite(stepC, HIGH);
            }
            else {
                digitalWrite(stepC, LOW);
            }
            stepDone = 1;
        }
    }
    return;
}

void stepperZ(int delayZ) {
    stepDone = 0;
    while (stepDone == 0){
        curTime = micros();
        if (curTime - prevTimeZ >= delayZ){
            prevTimeZ = curTime;
            if (digitalRead(stepZ) == LOW){
                digitalWrite(stepZ, HIGH);
            }
            else {
                digitalWrite(stepZ, LOW);
            }
            stepDone = 1;
        }
    }
    return;
}

void stepperX(int delayX) {
    stepDone = 0;
    while (stepDone == 0){
        curTime = micros();
        if (curTime - prevTimeX >= delayX){
            prevTimeX = curTime;
            if (digitalRead(stepX) == LOW){
                digitalWrite(stepX, HIGH);
            }
            else {
                digitalWrite(stepX, LOW);
            }
            stepDone = 1;
        }
    }
    return;
}

void stepperA(int delayA) {

```

```

stepDone = 0;
while (stepDone == 0){
  curTime = micros();
  if (curTime - prevTimeA >= delayA){
    prevTimeA = curTime;
    if (digitalRead(stepA) == LOW){
      digitalWrite(stepA, HIGH);
    }
    else {
      digitalWrite(stepA, LOW);
    }
    stepDone = 1;
  }
}
return;
}

//----- funkce HOME -----

int home(){
  homeDone = 0;
  homeCdone = 0;
  homeZdone = 0;
  homeXdone = 0;

  stepperOn();

  digitalWrite(dirC,HIGH);
  digitalWrite(dirZ,HIGH);
  digitalWrite(dirX,HIGH);
  servoUchop.write(0);

  int homeTime = 0;

  while (homeZdone == 0){
    curTime = micros();
    if (curTime - homeTime > 0){
      while (digitalRead(senzorZ) == LOW && stopON == 0){
        stepperZ(delayZ/2);
      }
      if (stopON == 0){
        digitalWrite(dirZ,LOW);
        for (int i=0; i<=1600; i++){
          stepperZ(delayZ*8);
        }
      }
      homeZdone = 1;
    }
  }

  homeTime = micros();

  while (homeXdone == 0){
    curTime = micros();
    if (curTime - homeTime >= 250){
      while (digitalRead(senzorX) == LOW && stopON == 0){
        stepperX(delayX/2);
      }
      if (stopON == 0){
        digitalWrite(dirX,LOW);
        for (int j=0; j<=4800; j++){

```

```

        stepperX(delayX*8);
    }
}
homeXdone = 1;
}
}

homeTime = micros();

while (homeCdone == 0){
    curTime = micros();
    if (curTime - homeTime >= 250){
        while (digitalRead(senzorC) == LOW && stopON == 0){
            stepperC(delayC*2);
        }
        if (stopON == 0){
            digitalWrite(dirC, LOW);
            for (int k=0; k<=565; k++){
                stepperC(delayC*8);
            }
        }
        homeCdone = 1;
    }
}

homeTime = micros();
if (homeDone == 0){
    analogWrite(ledB, 25);
}
homeDone = 1;

a=0;
c=0;
x=0;
z=0;

stepperOff();

return homeDone;
}

//----- funkcje STOP -----
void stop(){
    digitalWrite(stepC, LOW);
    digitalWrite(stepZ, LOW);
    digitalWrite(stepX, LOW);
    digitalWrite(stepA, LOW);

    stepperOff();

    motorC = 0;
    motorZ = 0;
    motorX = 0;
    motorA = 0;
    homeDone = 1;
    homeXdone = 1;
    homeZdone = 1;
    homeCdone = 1;
    startDone = 1;
    stepDone = 1;
}

```

```

    senzorXon = 1;
    senzorCon = 1;
    senzorZon = 1;
    stopON = 1;
    return;
}

//----- funkce START -----
//prozatim pouze na 3 panaky ve vezri v1.4

int start(){
    homeDone = 0;
    while (homeDone == 0){
        homeDone = home();
    }

    stepperOn();

    digitalWrite(ledB, LOW);
    long prevTime = 0;

    digitalWrite(dirZ, LOW);
    while (startDone == 0){ //najezd do souradnice lahve v ose Z-
        curTime = micros();
        if (curTime - prevTime >= 50000){
            for (z; z<=96000; z++){ //6400kr*15 celych otacek
                if (stopON == 0){
                    stepperZ(delayZ/2);
                }
            }
            prevTime = curTime;
            startDone = 1;
        }
    }
    startDone = 0;

    if (stopON == 0){
        digitalWrite(dirX, LOW);
        while (startDone == 0){ //najezd do souradnice lahve v ose X+
            curTime = micros();
            if (curTime - prevTime >= 50000){
                for (x; x<=128000; x++){ //6400kr*20 celych otacek
                    if (stopON == 0){
                        stepperX(delayX/2);
                    }
                }
                prevTime = curTime;
                startDone = 1;
            }
        }
        startDone = 0;
    }

    if (stopON == 0){
        if (servoAngle == 0){ //uchopeni lahve
            uchop();
        }
        else {
            servoUchop.write(0);
            uchop();
        }
    }
}

```

```

    }
}

prevTime = micros();

if (stopON == 0) {
digitalWrite(dirZ, HIGH); //zvednuti lahve v ose Z+
while (startDone == 0){
    curTime = micros();
    if (curTime - prevTime >= 50000){
        for (z; z>=19200; z--){ //6400*12 celych otacek
            if (stopON == 0){
                stepperZ(delayZ*1.5);
            }
        }
        prevTime = curTime;
        startDone = 1;
    }
}
startDone = 0;
}

if (stopON == 0) {
digitalWrite(dirC, LOW); //pootoceni s lahvi v ose C+
while (startDone == 0){
    curTime = micros();
    if (curTime - prevTime >= 50000){
        for (c; c<=3000; c++){
            if (stopON == 0){
                stepperC(delayC*1.5);
            }
        }
        prevTime = curTime;
        startDone = 1;
    }
}
startDone = 0;
}

if (stopON == 0) {
digitalWrite(dirZ, LOW); //pokles k panaku v ose Z-
while (startDone == 0){
    curTime = micros();
    if (curTime - prevTime >= 50000){
        for (z; z<64000; z++){ //6400kr*7 celych otacek
            if (stopON == 0){
                stepperZ(delayZ);
            }
        }
        prevTime = curTime;
        startDone = 1;
    }
}
startDone = 0;
}

nalej();

if (stopON == 0){
digitalWrite(dirC, LOW); //pootoceni k panaku c2 v ose C+

```



```

while (startDone == 0){
    curTime = micros();
    if (curTime - prevTime >= 150000){
        for (c; c<=3500; c++){ //
            if (stopON == 0){
                stepperC(delayC*2);
            }
        }
        prevTime = curTime;
        startDone = 1;
    }
}
startDone = 0;
}

nalej();

if (stopON == 0){
    digitalWrite(dirC, LOW); //pootoceni k panaku c3 v ose C+
    while (startDone == 0){
        curTime = micros();
        if (curTime - prevTime >= 150000){
            for (c; c<=4000; c++){ //
                if (stopON == 0){
                    stepperC(delayC*2);
                }
            }
            prevTime = curTime;
            startDone = 1;
        }
    }
    startDone = 0;
}

nalej();

if (stopON == 0){
    digitalWrite(dirC, HIGH); //pootoceni k panaku c3 v ose C+
    while (startDone == 0){
        curTime = micros();
        if (curTime - prevTime >= 100000){
            for (c; c>=1000; c--){ //
                if (stopON == 0){
                    stepperC(delayC);
                }
            }
            prevTime = curTime;
            startDone = 1;
        }
    }
    startDone = 0;
}

if (stopON == 0) {
    blinkG();
}

stepperOff();

startDone = 1;
return startDone;

```

```
}
```

```
//----- funkcce zablikej zelene -----
```

```
void blinkG(){
    int blinkDone;
    long blinkTime = 0;
    for (int b = 0; b <= 25; b++){
        while (blinkDone == 0){
            curTime = millis();
            if (curTime - blinkTime >= 100){
                if (digitalRead(ledG) == LOW){
                    digitalWrite(ledG, HIGH);
                }
                else {
                    digitalWrite(ledG, LOW);
                }
            }
            blinkTime = curTime;
            blinkDone = 1;
        }
    }
    blinkDone = 0;
}
digitalWrite(ledG, LOW);
return;
}
```

```
//----- funkcce nalej -----
```

```
void nalej(){
    a=0;
    int nalejDone = 0;
    long prevTime = 0;
    if (stopON == 0){
        digitalWrite(dirA, LOW); //naklopeni lahve v ose A+
        while (nalejDone == 0){
            curTime = micros();
            if (curTime - prevTime >= 50000){
                for (a; a<=2500; a++){ //
                    if (stopON == 0){
                        stepperA(delayA*2.5);
                    }
                }
                prevTime = curTime;
                nalejDone = 1;
            }
        }
        nalejDone = 0;
    }

    if (stopON == 0){
        digitalWrite(dirA, LOW); //naklopeni lahve v ose A+
        while (nalejDone == 0){
            curTime = micros();
            if (curTime - prevTime >= 5000){
                for (a; a<=4000; a++){ //
                    if (stopON == 0){
                        stepperA(delayA);
                    }
                }
            }
            prevTime = curTime;
        }
    }
}
```

```

        nalejDone = 1;
    }
}
nalejDone = 0;
}

if (stopON == 0){
digitalWrite(dirA, HIGH); //naklopeni lahve v ose A-
while (nalejDone == 0){
    curTime = micros();
    if (curTime - prevTime >= 3000000){
        for (a; a>=750; a--){ //
            if (stopON == 0){
                stepperA(delayA);
            }
        }
        prevTime = curTime;
        nalejDone = 1;
    }
}
nalejDone = 0;
}

if (stopON == 0){
digitalWrite(dirA, HIGH); //naklopeni lahve v ose A-
while (nalejDone == 0){
    curTime = micros();
    if (curTime - prevTime >= 5000){
        for (a; a>0; a--){ //
            if (stopON == 0){
                stepperA(delayA*1.5);
            }
        }
        prevTime = curTime;
        nalejDone = 1;
    }
}
nalejDone = 0;
}

}

//----- funkcce uchop -----
void uchop(){
    servoAngle = servoUchop.read();
    if (servoAngle == 0){
        servoAngle = 75;
    }
    else {
        servoAngle = 0;
    }
    servoUchop.write(servoAngle);
    Serial.println(servoAngle);
    return;
}

//----- LOOP -----
void loop() {

```

```

stopON = 0;

char klavesa = myKeypad.getKey(); //nacteni klavesy

if (klavesa){
    Serial.println(klavesa); //vypis stisknute klavesy na seriový monitor
}

//vyhomovani
if (homeDone == 0){
    homeDone = home();
}

if (startDone == 0){
    startDone = start();
}

if (motorC == 1){
    stepperC(delayC);
}

if (motorZ == 1){
    stepperZ(delayZ);
}

if (motorX == 1){
    stepperX(delayX);
}

if (motorA == 1){
    stepperA(delayA);
}

}

//----- funkce keypad -----
void keyEvent(KeypadEvent klavesa){
    switch (myKeypad.getState()){

        case PRESSED:
            stepperOn();
            switch (klavesa){
                case 'C':
                    digitalWrite(dirC,LOW); break;
                case 'c':
                    digitalWrite(dirC,HIGH); break;
                case 'Z':
                    digitalWrite(dirZ,HIGH); break;
                case 'z':
                    digitalWrite(dirZ,LOW); break;
                case 'X':
                    digitalWrite(dirX,LOW); break;
                case 'x':
                    digitalWrite(dirX,HIGH); break;
                case 'H':
                    homeDone = 0; break;
                case 's':
                    uchop(); break;
                case 'S':

```

```

        startDone = 0; break;
    }
    break;

case HOLD:
    switch (klavesA){
        case 'C':
            analogWrite(ledB, 0);
            motorC = 1; break;
        case 'c':
            analogWrite(ledB, 0);
            motorC = 1; break;
        case 'Z':
            analogWrite(ledB, 0);
            motorZ = 1; break;
        case 'z':
            analogWrite(ledB, 0);
            motorZ = 1; break;
        case 'X':
            analogWrite(ledB, 0);
            motorX = 1; break;
        case 'x':
            analogWrite(ledB, 0);
            motorX = 1; break;
        case 'k':
            analogWrite(ledB, 0);
            motorA = 1; break;
    }
    break;

case RELEASED:
    switch (klavesA){
        case 'C':
            stop(); break;
        case 'c':
            stop(); break;
        case 'Z':
            stop(); break;
        case 'z':
            stop(); break;
        case 'X':
            stop(); break;
        case 'x':
            stop(); break;
        case 'k':
            stop();
            digitalWrite(dirA, !digitalRead(dirA));
            break;
    }
    break;
}
}

```