

ROBOTICKÁ RUKA POMOCÍ ARDUINA – JOYSTICK

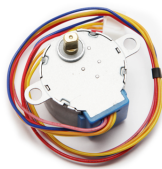
VĚTŠINA INDUSTRIÁLNÍCH I DOMÁCÍCH SYSTÉMŮ MÁ NĚJAKÉ VSTUPNÍ ZAŘÍZENÍ – OVLADAČ. V TÉTO KAPITOLE SE SEZNÁMÍTE S JOYSTICKEM, KTERÝ ZNÁTE Z POČÍTAČOVÝCH HER, JAKO VSTUPNÍM ZAŘÍZENÍM PRO OVLÁDÁNÍ ROBOTICKÉ RUKY.

CÍLE

- a** Pochopit princip joysticku a jeho využití jako vstupní zařízení pro konstrukce na bázi Arduina.
- b** Sestrojit jednoduchou robotickou ruku za pomoci krokového motorku, servo motorku a dílů vytištěných na 3D tiskárně.
- c** Tuto ruku naprogramovat a nastavit tak, aby uměla vymáchat pytlík čaje v připraveném hrníčku s vodou o požadované teplotě.



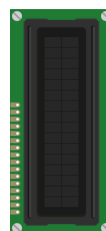
Servo motor



Krokový motor včetně



PS2 Joystick



LCD display



Potenciometr

POUŽITÉ SOUČÁSTKY

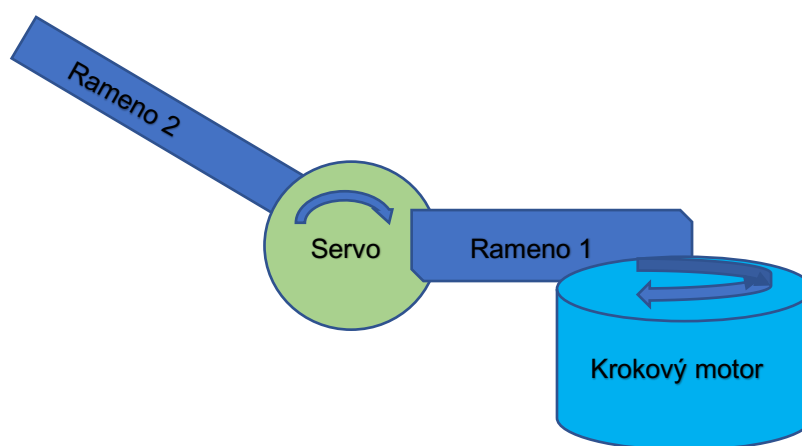
Čas: **135** m in

Úroveň: **□□□□□**

Vychází z: **5, 6, 7**

ROBOTICKÁ RUKA

Též zvaná *manipulátor*, patří mezi takzvané *sériové roboty* neboli řetězce. To jsou druhy robotů, charakteristické pravidelným střídáním pevných částí (zvaných *linky* nebo ramena) s pohyblivými částmi (zvanými *klouby*). Tyto klouby mohou být sférické, cylindrické nebo posuvné. V našem případě použijeme dva cylindrické (rotační) klouby sestavené za pomoci krokového motorku a serva známých z předchozích příkladů. Schéma viz následující obrázek.



Díly pro tuto robotickou ruku jsou vytištěné na 3D tiskárně a je možné si je stáhnout z WWW stránek. Jsou velmi jednoduché a jejich sestavení je patrné z přiložených obrázků. Předpokládám, že 3D tiskárnu již má většina škol k dispozici. V případě nouze, lze ramena sestavit např. z několika spojených kusů lepenkového papíru. S takovou rukou však řešte pouze první dvě úlohy. Pokud chcete řešit i závěrečnou úlohu této kapitoly – máchání čaje pomocí robotické ruky, pak je zakončení ruky nutné udělat z nějakého vodě (čaji) odolného materiálu anebo ruku testujte pouze „nasucho“ – bez vody.

Počet kloubů u robotické ruky u ní určuje takzvaný stupeň volnosti. Tato ruka má tedy dva stupně volnosti. Pokud byste někdy chtěli sestavit robotickou ruku, tak aby dokázala to, co umí ruka lidská potřebujete 6 stupňů volnosti. Průmyslově vyráběné a používané robotické ruce – manipulátory mají obvykle 5 až 7 stupňů volnosti.

JOYSTICK

Zvaný též pákový nebo křížový ovladač. Je vstupní zařízení, které se sestává z páky připevněné na základně, která se dokáže pohybovat ve směru x a y, přičemž je snímán

úhel a velikost vychýlení. Obvykle je rovněž vybaven jedním nebo více tlačítky, u kterých snímáno stlačení.

Joystick se používá pro hraní videoher, ale rovněž pro ovládání letadel, automobilů, robotických rukou atd.

Joystick obsažený ve vaší stavebnici má jedno tlačítko na vrcholu „kloboučku“. Pro připojení potřebuje pět vodičů. Dva jsou napájení. Jeden slouží jako indikace stlačení tlačítka a připojuje se na digitální vstup. Je nastaven tak, že implicitně proud prochází a při sepnutí nikoliv. Další dva vodiče se připojují na analogové vstupy a poskytují informaci o pohybu joysticku ve směru os x a y jako číslo v intervalu 0 až 1023. 512 znamená, že joystick je v prostředku a obě hraniční čísla pak znamenají maximální vychýlení. V následujícím příkladu se s joystickem blíže seznámíte.

POZNÁMKA

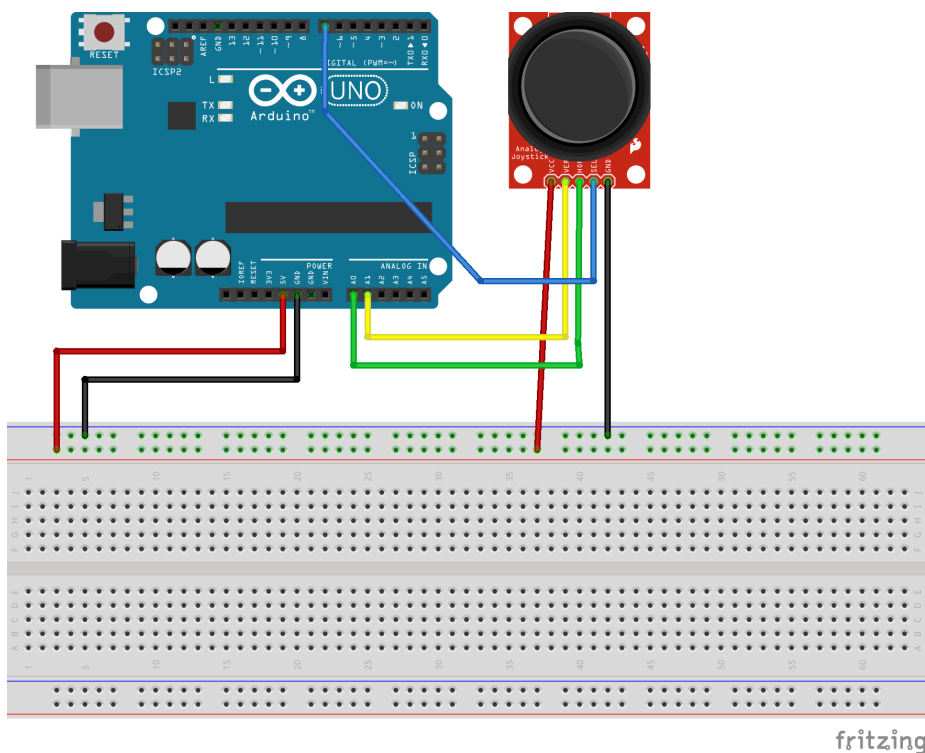
Následující úkoly na sebe navazují, a proto po vyřešení prvního úkolu není nutné obvod rozpojovat, ale naopak postupně k němu budete přidávat další součástky. Stejně tak i program je koncipován od jednoduššího ke složitějšímu.

ÚKOL 1

Zapojte joystick a monitorujte jeho stavy pomocí sériové komunikace.

ZAPOJENÍ OBVODU

Sestavte obvod dle následujícího obrázku:



fritzing

PROGRAMOVÝ KÓD

```
1  int JoyStick_X = 0; //Xová osa joysticku - analogový pin 0
2  int JoyStick_Y = 1; //Yová osa joysticku - analogový pin 1
3  int JoyStick_Z = 7; //Tlačítko joysticku - pin 7
4  int x,y,z;
5
6  void setup() {
7      Serial.begin(9600);
8      pinMode(JoyStick_Z, INPUT_PULLUP); //Nastavení tlačítka
9      joysticku
10     Serial.println("Test joysticku");
11 }
12
13 void loop() {
14     x=analogRead(JoyStick_X);
15     y=analogRead(JoyStick_Y);
16     z=digitalRead(JoyStick_Z);
17     Serial.print("X = ");
18     Serial.print(x);
19     Serial.print(", Y = ");
20     Serial.print(y);
21     Serial.print(", Z = ");
22     Serial.println(z);
23     delay(500);
24 }
```

a

b

c

- a** Nastavení připojení joysticku. Vodiče pro pohyb ve směru X a Y jsou připojeny na analogové piny na portech A0 a A1. Vodič pro osu Z (tlačítko joysticku) na pin 7 (digitální). Proměnné x, y a z jsou jejich instance v programu.
- b** Úvodní nastavení. Nastavení tlačítka joysticku a příprava sériového portu.
- c** Hlavní program s cyklickým opakováním. Načtení hodnot z joysticku a jejich výpis na sériový port. Aby se výpis neopakoval příliš často, čeká se před následujícím načtením půl vteřiny

Program přeložte a nahrajte do Arduina. V programu Arduino si otevřete Sériový monitor. Pohybujte joystickem a sledujte, jak se při tom mění výpis.



NEFUNGUJE JOYSTICK

Zapojení v desce – zkontrolujte, zda jsou vývody zapojeny do odpovídajících pinů desky Arduino.

Zapojení v joysticku – zkontrolujte, zda jsou vývody zapojeny do odpovídajících pinů joysticku.

NEJDE NAHRÁT KÓD DO DESKY

USB kabel – ujistěte se, že máte desku Arduino připojenou k počítači.

Správný port – ujistěte se, že máte vybraný správný port pro připojení k desce Arduino pomocí USB kabelu.

NEFUNGUJE SÉRIOVÝ MONITOR

Nezobrazuje se text – Zkontrolujte zapojení USB kabelu, zkontrolujte nastavení správné rychlosti – 9660 baudů.

Zobrazuje se nesmyslný text – Zkontrolujte nastavení správné rychlosti – 9660 baudů.



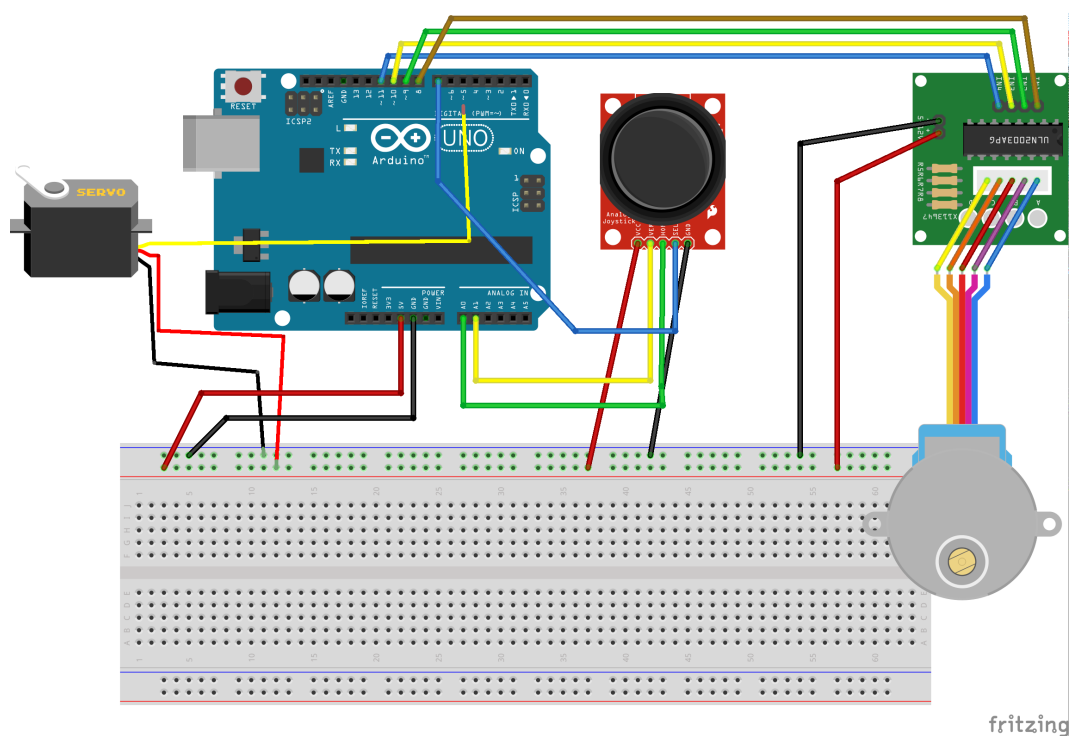
Zkuste úlohu řešit tak, že si místo sériového monitoru zapojíte LCD displej a pohyby joysticku si budete zobrazovat na něm. Můžete využít schéma zapojení a zdrojový kód z Úlohy 3 v této kapitole.

ÚKOL 2

Sestrojte robotickou ruku ovládanou pomocí joysticku. Součástky na výrobu ruky vytiskněte na 3D tiskárně.

ZAPOJENÍ OBVODU

Sestavte ze součástek robotickou ruku dle fotografie. Obvod zapojte dle následujícího schématu. K joysticku přibydou dva motorky. Krokový motor se zapojuje pomocí ovladače.



PROGRAMOVÝ KÓD

```
1  #include <Servo.h>
2
3  Servo myservo;//Vytvoření objektu pro řízení krokového
4  motoru
5  int poloha; //Svislá poloha ruky
6
7  // Piny pro krokový motor
8  const int in1 = 8;
9  const int in2 = 9;
10 const int in3 = 10;
11 const int in4 = 11;
12 // proměnná pro nastavení rychlosti,
13 // se zvětšujícím se číslem se rychlost zmenšuje
14 int rychlost = 8;
15
16
17 //Joystick
18 int JoyStick_X = 0; //Xová osa joysticku - analogový pin 0
19 int JoyStick_Y = 1; //Yová osa joysticku - analogový pin 1
20 int JoyStick_Z = 7; //Tlačítko joysticku - pin 7
21 int x,y,z;
22
23
24
25 void setup() {
26     myservo.attach(6);//Servo motor je na pinu 6
27     myservo.write(0);//Ruka do výchozí polohy
28     poloha = 0; //Pamatuj si tuto polohu
29     // inicializace digitálních výstupů pro krokový motor
30     pinMode(in1, OUTPUT);
31     pinMode(in2, OUTPUT);
32     pinMode(in3, OUTPUT);
33     pinMode(in4, OUTPUT);
34     //inicializace Joysticku
35     pinMode(JoyStick_Z, INPUT_PULLUP); //Nastavení tlačítka
36     joysticku
37 }
38
39 void loop() {
40     pohyb();
41     while(1);
42 }
43
44
45
46
```

a

b

c

d

e


```

47 void rotacePoSmeru(int uhel) {
48     for(int i=0;i<(uhel*64/45);i++){
49         krok(1,0,0,0);
50         krok(1,1,0,0);
51         krok(0,1,0,0);
52         krok(0,1,1,0);
53         krok(0,0,1,0);
54         krok(0,0,1,1);
55         krok(0,0,0,1);
56         krok(1,0,0,1);}
57 }
58 void rotaceProtiSmeru(int uhel) {
59     for(int i=0;i<(uhel*64/45);i++){
60         krok(1,0,0,1);
61         krok(0,0,0,1);
62         krok(0,0,1,1);
63         krok(0,0,1,0);
64         krok(0,1,1,0);
65         krok(0,1,0,0);
66         krok(1,1,0,0);
67         krok(1,0,0,0);}
68 }
69
70 void krok(int a, int b, int c, int d){
71     digitalWrite(in1, a);
72     digitalWrite(in2, b);
73     digitalWrite(in3, c);
74     digitalWrite(in4, d);
75     delay(rychllost);
76 }
77
78 void pohyb(){
79     int x,y,z;
80     z=1;
81     while (z) {
82         x=analogRead(JoyStick_X);
83         y=analogRead(JoyStick_Y);
84         z=digitalRead(JoyStick_Z);
85         if (x>550) { //doprava
86             rotacePoSmeru(5);
87         }
88         else if (x<480){ //doleva
89             rotaceProtiSmeru(5);
90         }
91         else if (y<480){ //dolů
92             if (poloha>=5) {
93                 poloha=poloha-5;

```

f

g

```

94         myservo.write(poloha);
95         delay(1000);
96     }
97 }
98 else if (y>550){ //nahoru
99     if (poloha<=170) {
100         poloha=poloha+5;
101         myservo.write(poloha);
102         delay(1000) ;
103     }
104 }
105 }
106 delay(100);
107 }

```

g

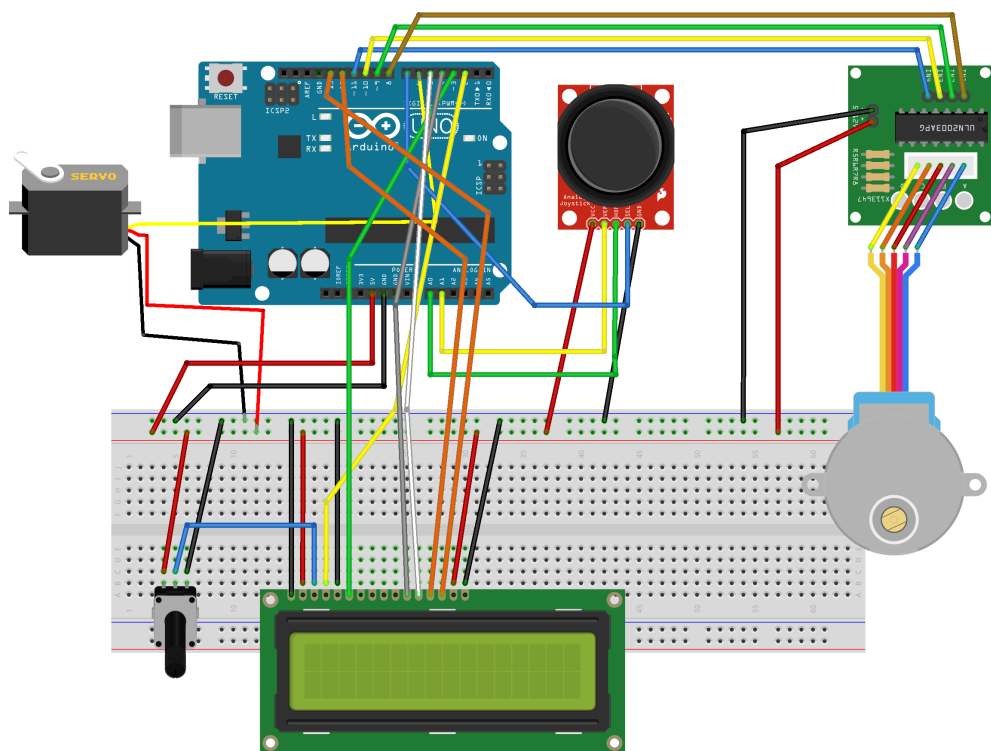
- a** Nastavení servo motoru (svislý pohyb). Proměnná poloha slouží k zapamatování aktuální svislé polohy ruky.
- b** Nastavení krokového motoru (vodorovný pohyb). Proměnná rychlost slouží pro nastavení pauzy mezi jednotlivými pohyby motoru.
- c** Nastavení pinů joysticku (viz předchozí příklad).
- d** Inicializace potřebných proměnných a knihoven.
- e** Hlavní program. Nedělá nic jiného, než spustí proceduru pohyb, která se stará o pohyb ruky, dle pohybu joysticku. Konstrukce while(1) je nekonečná smyčka ve které skončí program po stisku tlačítka joysticku. Pro opětovné spuštění je nutné stisknout tlačítko reset na Arduinu.
- f** Funkce řídící pohyb krokového motorku. Tyto funkce jsou opsané z manuálu výrobce motorku (dle manuálu k stavebnici, kterou máte k dispozici). Máte-li jiný motorek, než ten ze stavebnice je možné (ale nepravděpodobné), že bude tyto funkce nutné přepsat dle manuálu vašeho motorku.
- g** Hlavní funkce. Načítá pohyb joysticku do proměnných x a y a hlídá stisk tlačítka (proměnná z). Dle těchto hodnot pak řídí pohyb ruky pomocí obou motorků. Funkce končí stiskem tlačítka (v z je po stisku 0).

ÚKOL 3

Sestrojte robotickou ruku pro máchání čaje, ovládanou pomocí joysticku. Tato ruka by měla umět připevněný čajový pytlík máchat předem danou dobu v hrnečku a pak jej přesunout mimo hrneček.

ZAPOJENÍ OBVODU

Upravte zapojení z předchozího příkladu. Doplňte LCD panel, na který se budou zobrazovat informace a stavy ruky. Zkontrolujte opravdu pečlivě jeho zapojení vzhledem k velkému počtu vodičů.



fritzing

PROGRAMOVÝ KÓD

Následující kód vychází z předchozího příkladu a popsány proto budou pouze odlišnosti.

```
1  #include <LiquidCrystal.h>
2
3  #include <Servo.h>
4
5  Servo myservo; //Vytvoření objektu pro řízení krokového
6  motoru
7
8  int poloha; //Svislá poloha ruky
9
10 // Piny pro krokový motor
11 const int in1 = 8;
12 const int in2 = 9;
13 const int in3 = 10;
14 const int in4 = 11;
15
16 // Proměnná pro nastavení rychlosti,
17 // se zvětšujícím se číslem se rychlost zmenšuje
18 int rychlost = 8;
19 int uhel1;
20 int x,y,z;
21 int i,j;
22
23 //Joystick
24 int JoyStick_X = 0; //Xová osa joysticku - analogový pin 0
25 int JoyStick_Y = 1; //Yová osa joysticku - analogový pin 1
26 int JoyStick_Z = 7; //Tlačítko joysticku - pin 7
27
28 // Piny pro připojení displeje
29 LiquidCrystal lcd(2, 3, 4, 5, 12, 13);
30
31 int minut; //Počet minut pro máchání čaje
32
33 void setup() {
34     myservo.attach(6); //Servo motor je na pinu 6
35     myservo.write(0); //Ruka do výchozí polohy
36     poloha = 0; //Pamatuj si tuto polohu
37     // inicializace digitálních výstupů pro krokový motor
38     pinMode(in1, OUTPUT);
```

a

a

```

39     pinMode(in2, OUTPUT);
40     pinMode(in3, OUTPUT);
41     pinMode(in4, OUTPUT);
42     //incializace Joysticku
43     pinMode(JoyStick_Z, INPUT_PULLUP); //Nastavení tlačítka
44 joystick
45     lcd.begin(16, 2); // Počet sloupců a řádek LCD displeje
46
47 }
48
49 void loop() {
50     //main program
51     lcd.clear();
52     lcd.print("Pouzij joystick");
53     lcd.setCursor(0,2);
54     lcd.print("pro nastaveni");
55     pocatecni_nastaveni(); //Nastavení polohy ruky "nad
56 hrnek"
57     lcd.clear(); //Nastavení polohy pro připevnění pytlíku
58     myservo.write(poloha+15);
59     delay(1000);
60     rotateProtiSmeru(90);
61     delay(1000);
62     lcd.print("Ruka pripravena");
63     lcd.setCursor(0,2);
64     lcd.print("pripevni caj a potvrd"); //Potvrdit stiskem
65 joystick
66     delay(1000);
67     z=1;
68     while (z) {
69         x=analogRead(JoyStick_X);
70         y=analogRead(JoyStick_Y);
71         z=digitalRead(JoyStick_Z);
72         delay(100);
73     }
74     delay(1000);
75     minut=pocetMinut(); //Nastavení počtu minut pro máchání
76     delay(1000);
77     rotatePoSmeru(90); //Najedeme nad čaj
78     delay(1000);
79     myservo.write(poloha-5); //Máchání čaje
80     for (i=minut;i;i--)
81     {
82         lcd.clear();
83         lcd.print("Zbyva:");
84         lcd.setCursor(0,1);
85         lcd.print(i);

```

a

b

```

86     lcd.print(" minut");
87     for (j=1;j<10;j++){
88         myservo.write(poloha-7);
89         delay(3000);
90         myservo.write(poloha+7);
91         delay(3000);
92     }
93 }
94 lcd.clear(); //Konec máchání
95 lcd.print("Hotovo");
96 myservo.write(poloha+20);
97 delay(10000);
98 rotatePoSmeru(90); //Odjezd doprava
99 myservo.write(15);
100 while(1) { } //Nekonečná smyčka
101 }
102
103 // zde následují funkce pro volání jednotlivých
104 // kroků pro otočení po či proti směru hodinových
105 // ručiček
106 void rotatePoSmeru(int uhel) {
107     for(int i=0;i<(uhel*64/45);i++){
108         krok(1,0,0,0);
109         krok(1,1,0,0);
110         krok(0,1,0,0);
111         krok(0,1,1,0);
112         krok(0,0,1,0);
113         krok(0,0,1,1);
114         krok(0,0,0,1);
115         krok(1,0,0,1);}
116 }
117 void rotateProtiSmeru(int uhel) {
118     for(int i=0;i<(uhel*64/45);i++){
119         krok(1,0,0,1);
120         krok(0,0,0,1);
121         krok(0,0,1,1);
122         krok(0,0,1,0);
123         krok(0,1,1,0);
124         krok(0,1,0,0);
125         krok(1,1,0,0);
126         krok(1,0,0,0);}
127 }
128 // každý krok obsahuje výrobcem dané pořadí
129 // pro správné spínání motoru a následnou
130 // pauzu, kterou určujeme rychlost otáčení
131 void krok(int a, int b, int c, int d){
132     digitalWrite(in1, a);

```

b

```

133     digitalWrite(in2, b);
134     digitalWrite(in3, c);
135     digitalWrite(in4, d);
136     delay(rychlost);
137 }
138
139 void pocatecni_nastaveni(){
140     int x,y,z;
141     z=1;
142     while (z) {
143         x=analogRead(JoyStick_X);
144         y=analogRead(JoyStick_Y);
145         z=digitalRead(JoyStick_Z);
146         if (x>550) { //doprava
147             rotacePoSmeru(5);
148         }
149         else if (x<480){ //doleva
150             rotaceProtiSmeru(5);
151         }
152         else if (y<480){ //dolů
153             if (poloha>=5) {
154                 poloha=poloha-5;
155                 myservo.write(poloha);
156                 delay(1000);
157             }
158         }
159         else if (y>550){ //nahoru
160             if (poloha<=170) {
161                 poloha=poloha+5;
162                 myservo.write(poloha);
163                 delay(1000);
164             }
165         }
166     }
167     delay(100);
168 }
169
170 int pocetMinut(){
171     int m=2;
172     int x, y, z;
173     z=1;
174     lcd.setCursor(0,0);
175     lcd.print("Maximum 9 minut");
176     lcd.setCursor(0,1);
177     lcd.print("Louhovat: ");
178     lcd.print(m);
179     lcd.print(" min");

```

```

180     while (z) {
181         x=analogRead(JoyStick_X);
182         y=analogRead(JoyStick_Y);
183         z=digitalRead(JoyStick_Z);
184         if (y>550){ //dolu
185             if (m) {
186                 m=m-1;}
187             lcd.setCursor(10,1);
188             lcd.print(m);
189             }
190         if (y<480){ //nahoru
191             if (m<9) {
192                 m=m+1;}
193             lcd.setCursor(10,1);
194             lcd.print(m);
195
196         }
197         delay(300);
198     }
199 }
200 return m;
201 }

```

- a** Nastavení LCD panelu včetně pinů, na které je zapojen.
- b** Vlastní program, tentokrát složitější. Postupně se prochází následující kroky:
 - a. Počáteční nastavení polohy ruky – nad hrnek.
 - b. Přesun ruky na pozici, kde se připevní pytlík čaje.
 - c. Nastavení doby pro máchání pytlíku.
 - d. Vlastní máchání pytlíku.
 - e. Posun ruky na místo, kde je možné pytlík sejmout.
 - f. Nekonečná smyčka na závěr.
- c** Funkce pro nastavení počtu minut pro máchání čaje.

PRÁCE S ROBOTICKOU RUKOU

- a** Po spuštění programu je třeba jako první věc nastavit robotickou ruku nad hrnek. Doporučuji nastavovat na prázdný hrnek. Ruku nastavte trochu napravo od středu hrnku a částečně jí ponořte pod horní okraj (cca. 1 cm). Potvrďte stiskem joysticku.
- b** Ruka si nyní najede vlevo od hrnku na pozici, na které můžete připevnit pytlík s čajem. Snažte se při tom nepohybovat s rukou. Pokud se vám to nepodaří, připevněte pytlík, stiskněte reset Arduino a vraťte se na krok jedna. Alternativně začínejte rovnou s připevněným pytlíkem.

- c** Připravte si talířek nebo nějakou podložku, nad kterou ruka donese vylouhovaný pytlík. Umístěte jí na pozici asi 90° napravo od hrnku.
- d** Nastavte počet minut, po které se má čaj louhovat, nalejte do hrnku vodu s požadovanou teplotou a potvrďte joystickem.
- e** Robotická ruka provede vymáchání čaje a po nastavené době odnese pytlík napravo od hrnku nad připravenou podložku.
- f** Tímto celý cyklus končí, pro jeho opakování je nutné stisknout reset na Arduinu.

POZNÁMKY

Věnujte opravdovou pečlivost zapojení obvodu. Zejména zapojení LCD panelu je poměrně složité díky velkému počtu vodičů.

Vzhledem k tomu, že se zde pracuje s kapalinou (čajem) dávejte pozor, aby nedošlo k namočení Arduina, motorků nebo jiných elektronických součástí. Pokud k tomuto přece jen dojde, okamžitě odpojte Arduino od zdroje elektrického proudu a nechte vše dobře vyschnout, nejlépe do příštího týdne.

Můžete se setkat s nepřesnostmi v pohybu ruky, které jsou způsobeny tím, že ruka zavadí o nějaký předmět (nejčastěji hrnek). Pokud si tohoto všimnete v počátečních fázích, raději systém resetujte a začněte od začátku.

ZÁVĚR

V tomto příkladu jste se naučili zapojit obvod sestávající z dvou motorků, displeje a joysticku. Dozvěděli jste se něco málo o robotice a vyzkoušeli si ovládání jednoduché robotické ruky. Je třeba si uvědomit, že robotické ruce v praxi mívají nejméně pět stupňů volnosti – pět motorků a jejich ovládání a programování je podstatně složitější.