

PODROBNÝ PRŮVODCE TEORIÍ

PODROBNĚ ROZEPSANÉ PŘÍKLADY S POPISEM FUNKCIONALITY OBVODŮ A PROGRAMOVÉHO KÓDU, KTERÝ JE ZAMĚŘEN NA POUŽÍVÁNÍ SERVOMOTORŮ S VYUŽITÍM PODMÍNKOVÉHO PŘÍKAZU **if..else**. ŘEŠENÉ ÚKOLY ZOHLEDŇUJÍ NOVĚ PROBRANÉ VĚCI A STAVÍ NA PŘEDCHOZÍCH ZNALOSTECH.

OBSAH PRŮVODCE

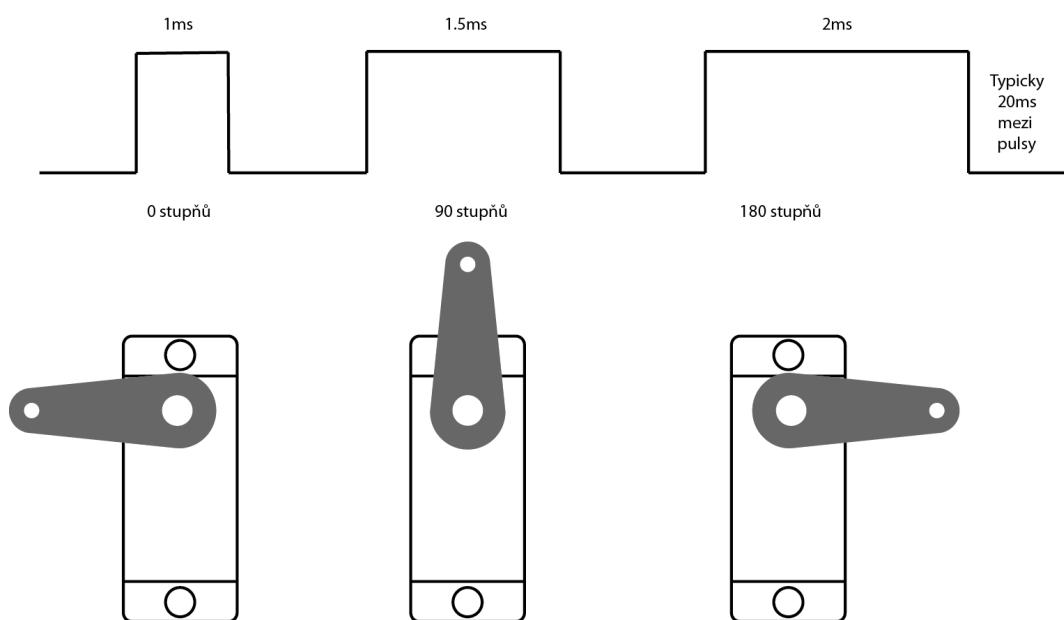
- ① Princip servomotorů.
- ② Vysvětlení podmínkového příkazu **if..else**.
- ③ Příklad zapojení servomotoru.
- ④ Princip a zapojení potenciometru v Arduino pro ovládání servomotoru.
- ⑤ Programové kódy pro vysvětlení používání podmínkového příkazu **if..else..**
- ⑥ Řešené problémy při zapojení servomotoru.
- ⑦ Technická část pro závěrečný projekt – slunečnice.
- ⑧ Vysvětlení řešení samostatných úkolů.

SERVOMOTORY

Servomotory jsou speciální typy motorů, které obvykle neumožňují otáčení neustále dokola, ale udržují nastavený úhel natočení. Využívají se například pro ovládání robotické ruky, změny pozice klapek, křídélek nebo kormidla u leteckých modelů. Jejich hlavní výhodou je malý rozměr a hmotnost s relativně velkou silou.

Servomotory jsou řízeny posíláním elektrického impulsu s proměnnou šířkou nebo **modulací šířky impulzu** (PWM). Impuls je předáván přes řídící kabel. Servomotor se obvykle může otáčet o 90° v obou směrech o celkovém rozsahu 180° . Neutrální poloha motoru je definována jako poloha, ve které má servomotor stejné množství potencionálního otáčení v obou směrech. PWM posílaná k motoru určuje polohu hřídele a na základě trvání impulsu odesланého řídícím vodičem. Rotor se vrátí do požadované polohy.

Servomotor očekává puls každých 20 milisekund (ms) a délka impulsu určuje, jak daleko se motor otáčí. Například 1,5ms puls přivede motor do polohy 90° . Kratší než 1,5 ms se pohybuje proti směru hodinových ručiček k poloze 0° a delší než 1,5 ms otáčí servomotorem ve směru hodinových ručiček směrem k 180° pozici.



Obr. 1 - Princip servomotoru

PODMÍNKOVÝ PŘÍKAZ IF..ELSE

Podmínkový příkaz **if...else** umožňuje větší kontrolu nad tokem kódu. Příkaz začíná klíčovým slovem **if** a první podmínkou. Když tato podmínka nastane, tak se vykoná blok příkazů. Při nesplnění první podmínky lze využít alternativu v podobě příkazu **else**, za kterým následuje opět blok příkazů. Příkaz **else** může být doplněn o další podmínku **if**. Program se prochází postupně přes jednotlivé podmínky a pokud není žádná splněna, tak skončí v samostatné části **else** bez podmínky.

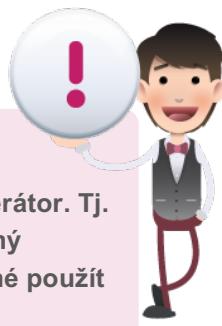
SYNTAXE

```
1  if (podmínka A) {  
2      // blok příkazů A  
3  }else if (podmínka B) {  
4      // blok příkazů B  
5  }else{  
6      // blok příkazů C  
7  }
```

Všimněte si, že blok **else if** může být použit s nebo bez ukončovacího bloku **else** a naopak. Je povolen neomezený počet takových **else if** větví.

Pro podmínky příkazu **if..else** se využívá porovnávacích operátorů.

```
1  x == y (x se rovná y)  
2  x! = y (x není rovno y)  
3  x < y (x je menší než y)  
4  x> y (x je větší než y)  
5  x <= y (x je menší nebo rovno y)  
6  x> = y (x je větší nebo roven y)
```



DEJTE SI POZOR

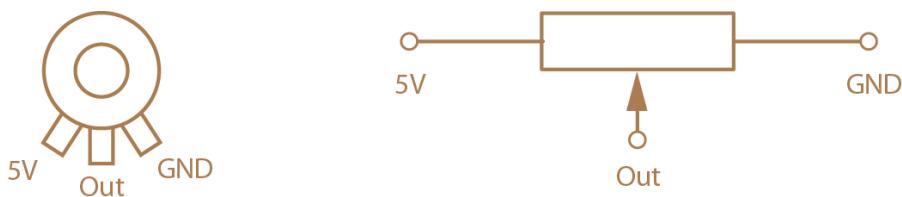
→ Musíte si dát pozor na to, abyste náhodou nepoužili přiřazovací operátor. Tj. znaménko jednoduchého rovná se. Např. `if(x=10)`. Takto uvedený operátor by proměnné `x` přiřadil hodnotu 10. Pro porovnání, je nutné použít dvojité rovná se. Např. `if(x==10)`.

POTENCIOMETR V ARDUINU

Potenciometr je součástka, poskytující proměnlivý odpor, který lze číst deskou Arduino jako analogovou hodnotu. Potenciometry mají zpravidla tři vývody.

- ① Jeden vývod je připojení k napětí.
- ② Druhý vývod (prostřední) je připojen do analogového vstupu desky Arduino.
- ③ Třetí vývod je připojen k zemnění.

Otáčením kolíku potenciometru měníme velikost odporu na obou stranách snímací plochy, která je připojena k prostřednímu vývodu potenciometru. Pokud bude hřídelí otočeno do krajní polohy (minimální), bude hodnota výstupního napětí 0V a přečteme hodnotu 0. Otočením hřídele na druhou stranu do krajní polohy (maximální) bude hodnota napětí 5V, ale na analogovém vstupu přečteme hodnotu 1023. Funkce `analogRead()` vrací číslo mezi 0 a 1023, které je úměrné hodnotě velikosti napětí.

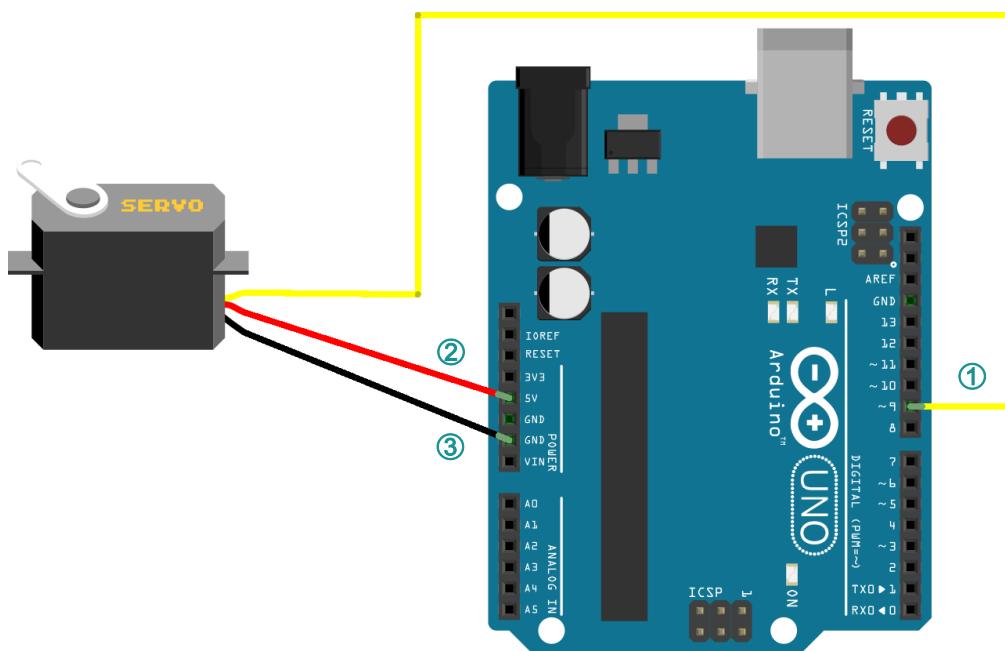


Obr. 2 - Vývody potenciometru

ZÁKLADNÍ PŘÍKLAD

ZAPOJENÍ OBVODU

Základní příklad se omezí pouze na jednoduché zapojení servomotoru bez přídavného potenciometru.



Obr. 3 – Základní zapojení servomotoru

- ① Řídící vodič servomotoru je připojen do desky Arduino na jeden z digitálních pinů, který umožňuje PWM. V příkladu to je pin číslo 9.
- ② Napájení servomotoru je zajištěno přes červený vodič, který je připojen na pin 5V. Pro proudově náročnější servomotory lze využít externí zdroj napájení.
- ③ Vodič zemnění je připojen na pin GND.

PROGRAMOVÝ KÓD

Základní programový kód využívá již známých konstrukcí, které byly použity v předchozích lekcích. Proto je vhodné si zopakovat především příkaz cyklu **for**.



Pro práci se servomotory se využívá již v Arduinu připravená knihovna **servo**. Pokud by nebyla nainstalovaná, musí se přidat podle návodu v úvodní kapitole.

ZÁKLADNÍ PŘÍKLADEM – ZMĚNA ÚHLU NATOČENÍ

Tento program postupně mění úhel natočení z 0° až 180° a zpět.

```
1 #include <Servo.h>
2
3 Servo myservo;
4 int pos = 0;
5
6 void setup()
7 {
8     myservo.attach(9);
9 }
10
11 void loop()
12 {
13     for(pos = 0; pos <= 180; pos++)
14     {
15         myservo.write(pos);
16         delay(15);
17     }
18     for(pos = 180; pos >= 0; pos--)
19     {
20         myservo.write(pos);
21         delay(15);
22     }
23 }
```

- ① Připojení knihovny pro ovládání servomotoru.
- ② Vytvoření instance **myservo** třídy pro každý motor.
- ③ Deklarace proměnné **pos**, obsahující pozici motoru – úhel natočení.
- ④ Definice pinu, na který je motor připojen. Zde je na pinu **9**.
- ⑤ Cyklus **for** zajišťující otáčení od 0° do 180° .
- ⑥ Nastavení motoru na aktuální úhel.
- ⑦ Přerušení běhu programu, aby měl motor čas se natočit.
- ⑧ Cyklus **for** zajišťující otáčení od 180° zpět na 0° .
- ⑨ Nastavení motoru na aktuální pozici definovanou v proměnné **pos**.
- ⑩ Přerušení běhu programu, aby měl motor čas se natočit do nové pozice.



(Př. 1) Naprogramujte servomotor tak, aby se otáčel od 0° do 120° pomaleji a při zpětném chodu aby se otáčel rychleji.



SERVOMOTOR NEFUNGUJE

Zapojení servomotoru – zkontrolujte zapojení signálního vodiče servomotoru. Tento vodič musí být na některém z pinů poskytujících PWM. Dále zkontrolujte, zda nemáte zapojený vodič napájené (červený) a zemnění (hnědý) naopak.

NEJDE NAHRÁT KÓD DO DESKY

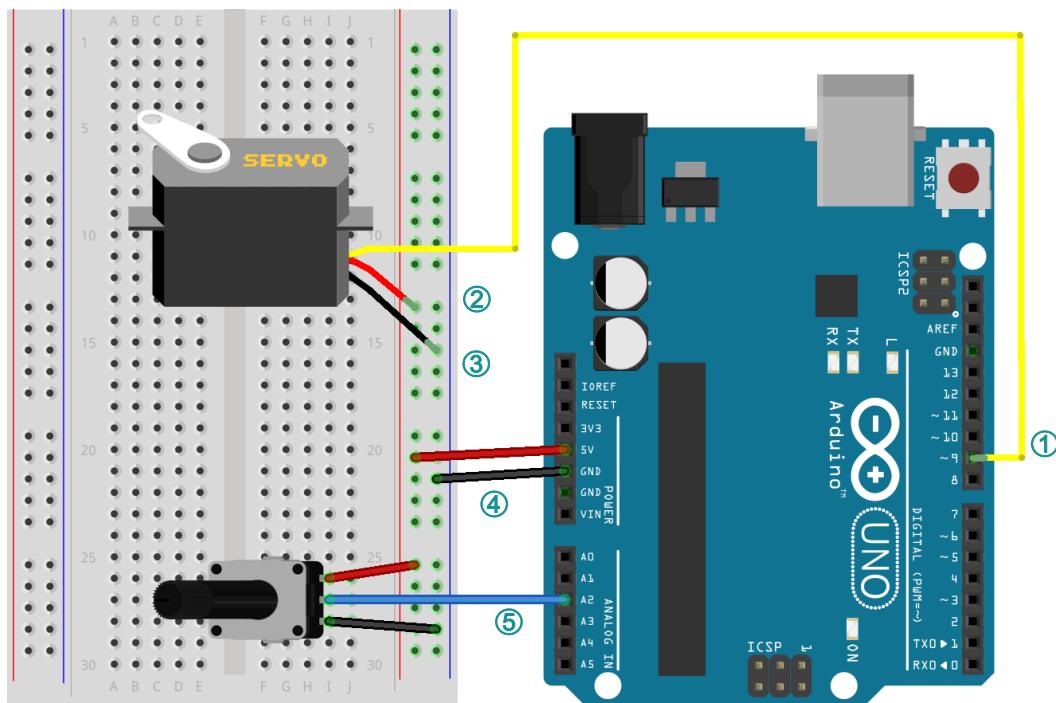
USB kabel – ujistěte se, že máte desku Arduino připojenou k počítači.

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

SERVOMOTOR OVLÁDÁNÝ POTENCIOMETREM

ZAPOJENÍ OBVODU

Základní příklad je rozšířen o zapojení potenciometru, pomocí kterého lze ovládat natočení servomotoru.



Obr. 4 - Zapojení potenciometru pro ovládání servomotoru

- ① Řídící vodič servomotoru je připojen do desky Arduino na jeden z digitálních pinů, který umožňuje PWM. V příkladu to je pin číslo **9**.
- ② Napájení servomotoru je zajištěno přes červený vodič, který je připojen do kontaktního pole. Pro proudově náročnější servomotory lze využít externí zdroj napájení.
- ③ Vodič zemnění je připojen také do kontaktního pole, do svislého bloku pro zem.
- ④ Do kontaktního pole je přivedeno napájení (červený vodič) a zemnění (černý vodič) přímo z desky Arduino.
- ⑤ Prostřední kontakt potenciometru je přiveden na analogový pin desky Arduino **A2**. Krajní vývody potenciometru jsou přivedeny do kontaktního pole, červený vodič do části pro napájení a černý vodič do části pro zemnění.

PROGRAMOVÝ KÓD

Programový kód pro tento příklad je jednodušší, protože nevyužívá cykly **for** pro nastavování pozice, ale přímo potenciometru.



Víme, že při čtení hodnoty potenciometru pomocí funkce `analogRead()`, bude hodnota na analogovém vstupu nabývat rozmezí 0 - 1023. Servomotor ale může být otáčen od 0° do 180°. Jazyk wiring nabízí funkci, která velmi jednoduše namapuje rozmezí hodnot potenciometru do rozmezí hodnot servomotoru. Jedná se o funkci `map()`.

`map(hodnota, zMin, zMax, doMin, doMax).`

OVLÁDÁNÍ SERVOMOTORU POTENCIOMETREM

Otáčením potenciometru se bude měnit poloha servomotoru.

```
1 #include <Servo.h>
2
3 Servo myservo;
4 int pos = 0;
5
6 void setup()
7 {
8     myservo.attach(9);
9 }
10
11 void loop()
12 {
13     pos = analogRead(A0);
14     pos = map(pos, 0, 1023, 0, 179);
15     myservo.write(pos);
16     delay(5);
17 }
```

- ① Připojení knihovny pro ovládání servomotoru.
- ② Vytvoření instance `myservo` třídy pro každý motor.
- ③ Deklarace proměnné `pos`, obsahující pozici motoru – úhel natočení.

- ④ Definice pinu, na který je motor připojen. Zde je na pinu **9**.
 - ⑤ Čtení hodnot potenciometru, které jsou v rozmezí 0 – 1023.
 - ⑥ Namapování hodnot potenciometru (0-1023) a rozmezí servomotoru (0-179).
 - ⑦ Natočení servomotoru do příslušné pozice uložené v proměnné **pos**.
 - ⑧ Krátké přerušení běhu programu, aby měl servomotor čas provést změnu natočení.

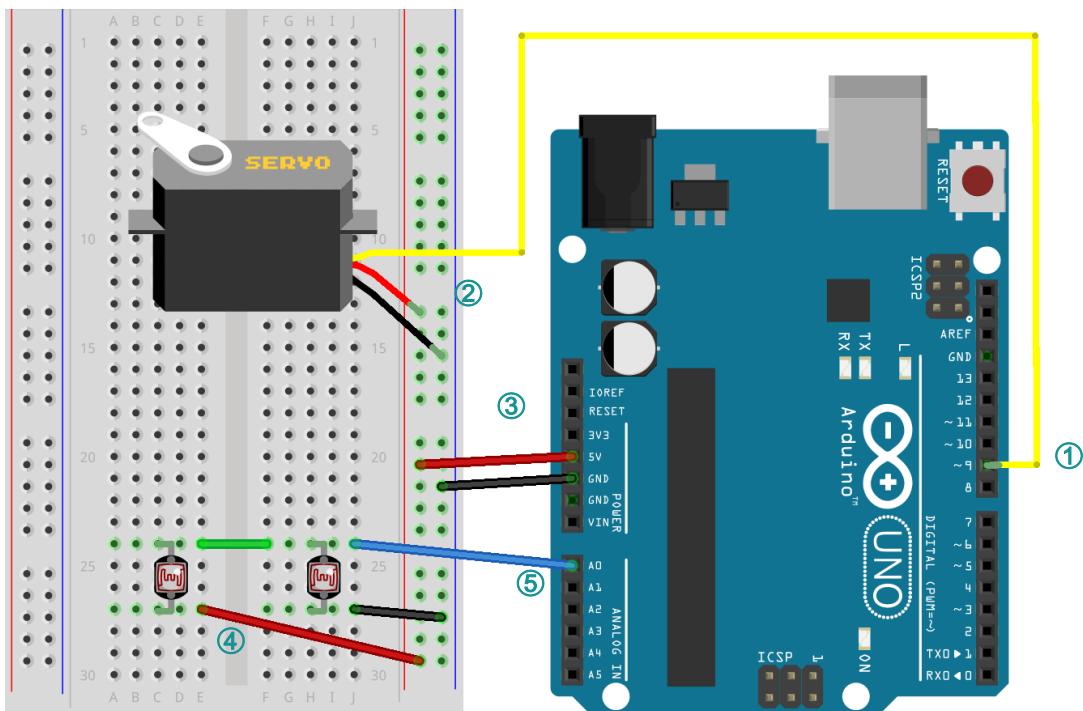


(Př. 2) Zkuste ovládat servomotor pomocí dvou tlačítek. Každé tlačítko bude ovládat servomotor v jednom směru. Dokud bude tlačítko stisknuto, tak se servomotor bude otáčet.

SERVOMOTOR OVLÁDANÝ FOTOREZISTORY

ZAPOJENÍ OBVODU

Tento příklad demonstруje ovládání servomotoru v závislosti na osvětlení fotorezistorů. Lze jej spojit s výrobou konstrukce, která představuje slunečnici, která se bude natáčet podle osvitu vždy jednoho z připojených fotorezistorů.



Obr. 5 - Zapojení fotorezistorů pro ovládání servomotoru

- ① Řídící vodič servomotoru je připojen do desky Arduino na jeden z digitálních pinů, který umožňuje PWM. V příkladu to je pin číslo **9**.
- ② Napájení servomotoru je zajištěno přes červený vodič, který je připojen do kontaktního pole. Pro proudově náročnější servomotory lze využít externí zdroj napájení. Do kontaktního pole je zapojen i vodič zemnění.
- ③ Do kontaktního pole je přivedeno napájení (červený vodič) a zemnění (černý vodič) přímo z desky Arduino.
- ④ Fotorezistory jsou zapojeny v kontaktním poli odděleně a propojeny vodičem podle schématu. (Takto zapojené fotorezistory představují tzv. proudový dělič.)
- ⑤ Hodnoty z fotorezistorů jsou předávány do desky Arduino prostřednictvím analogového pinu **A0**.

PROGRAMOVÝ KÓD

```

1 #include <Servo.h>                                ①
2
3 int sensorPin = A0;                                ②
4 int servoPin = 9;                                   ③
5
6 int sensorValue = 0;                               ④
7 int servoPos = 90;                                ⑤
8
9 Servo myservo;                                    ⑥
10
11 void setup() {                                     ⑦
12     pinMode(sensorPin, INPUT);                    ⑧
13     myservo.attach( servoPin );                  ⑨
14     myservo.write( servoPos );
15 }
16
17 void loop() {                                     ⑩
18     sensorValue = analogRead(sensorPin);          ⑪
19
20     if (sensorValue < (512) )                      ⑫
21     {
22         if (servoPos < 180)
23         {
24             servoPos++;
25         }
26     }
27 }
```

```

28
29     if (sensorValue > (512) )
30     {
31         if (servoPos > 0)
32         {
33             servoPos--;
34         }
35     }
36
37     myservo.write(servoPos);
38
39     delay(100);
40 }
```

- ① Připojení knihovny pro ovládání servomotoru.
- ② Deklarace proměnné pro číslo pinu, na který jsou připojeny fotorezistory.
- ③ Deklarace proměnné pro číslo pinu, na který je připojen datový vodič servomotoru.
- ④ Deklarace proměnné pro hodnoty z fotorezistorů.
- ⑤ Deklarace proměnné s hodnotou pro nastavení servomotoru.
- ⑥ Vytvoření instance **myservo** třídy pro každý motor.
- ⑦ Definice pinu, na který jsou připojeny fotorezistory jako vstup.
- ⑧ Definice pinu, na který je motor připojen. Zde je na pinu **9**.
- ⑨ Nastavení servomotoru do výchozí pozice – 90°.
- ⑩ Čtení hodnot fotorezistoru, které jsou v rozmezí 0 – 1023.
- ⑪ Testování hodnot fotorezistoru. Pokud je hodnota větší než polovina maxima (512), bude se servomotor otáčet od poloviny do 180°.
- ⑫ Průběžné otáčení servomotoru do příslušné pozice uložené ve zvětšující se proměnné **servoPos**.
- ⑬ Test na druhou část pohybu od 512 do 0.
- ⑭ Dekrementace proměnné **servoPos**, která představuje úhel natočení servomotoru.
- ⑮ Zajištění natočení servomotoru.



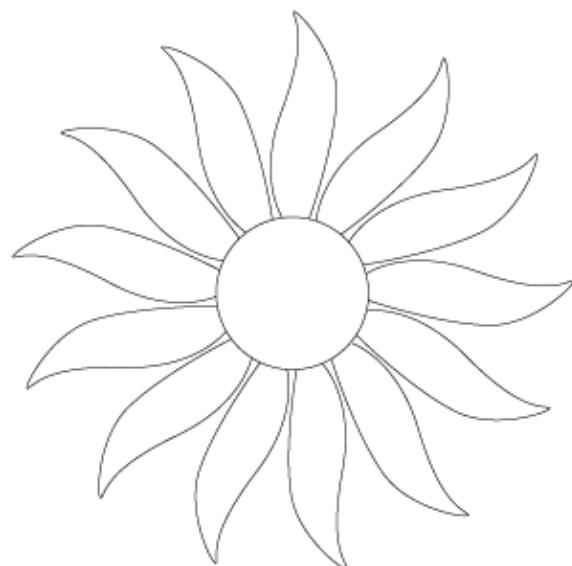
Pokud budete mít připravenou a funkční programovou část, můžete vytvořit květinu, která se otáčí za světlem. Návod je v následující kapitole.

OTÁČEJÍCÍ SE KVĚTINA

Pro vytvoření otáčející se květiny za světlem budeme potřebovat: karton (tvrdší papír), nůžky a tavnou pistoli (lepící pásku), brčko.

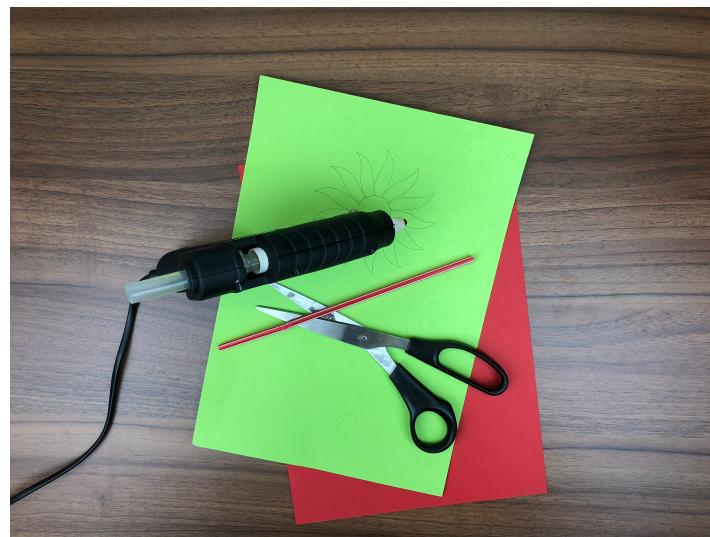
PAPÍROVÁ KONSTRUKCE

Konstrukce květiny je jednoduchá. Stačí si vytisknout na pevný papír přiloženou šablonu, která je na obrázku Obr. 6 - Šablona květiny.



Obr. 6 - Šablona květiny

Vytištěnou šablonu vystříhněte podle plných čar.



Obr. 7 - Vytiskněná šablona



Obr. 8 - Vystřížená šablona

Na vystříženou šablonu naneste tavnou pistoli lepidlo a následně vystřížené červené kolečko.



Obr. 10 – Přilepení středu



Obr. 9 – Lepení brčka

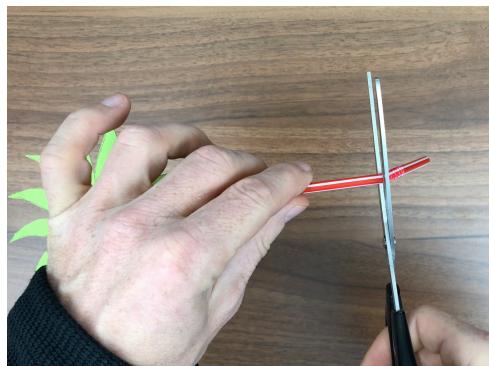
Naneste lepidlo z tavné pistole na brčko, které přilepte na květinu.



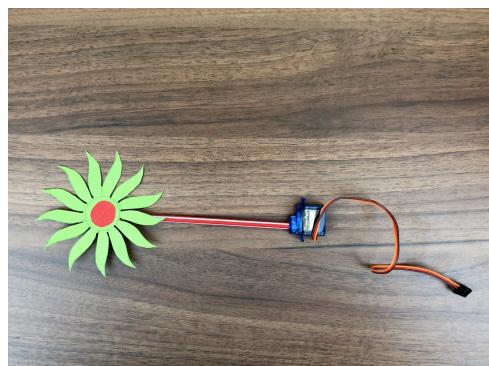
Obr. 11 – Přilepení brčka



Obr. 12 – Celý pohled na květinu



Obr. 13 – Zastřížení brčka



Obr. 14 – Připevnění květiny na servo

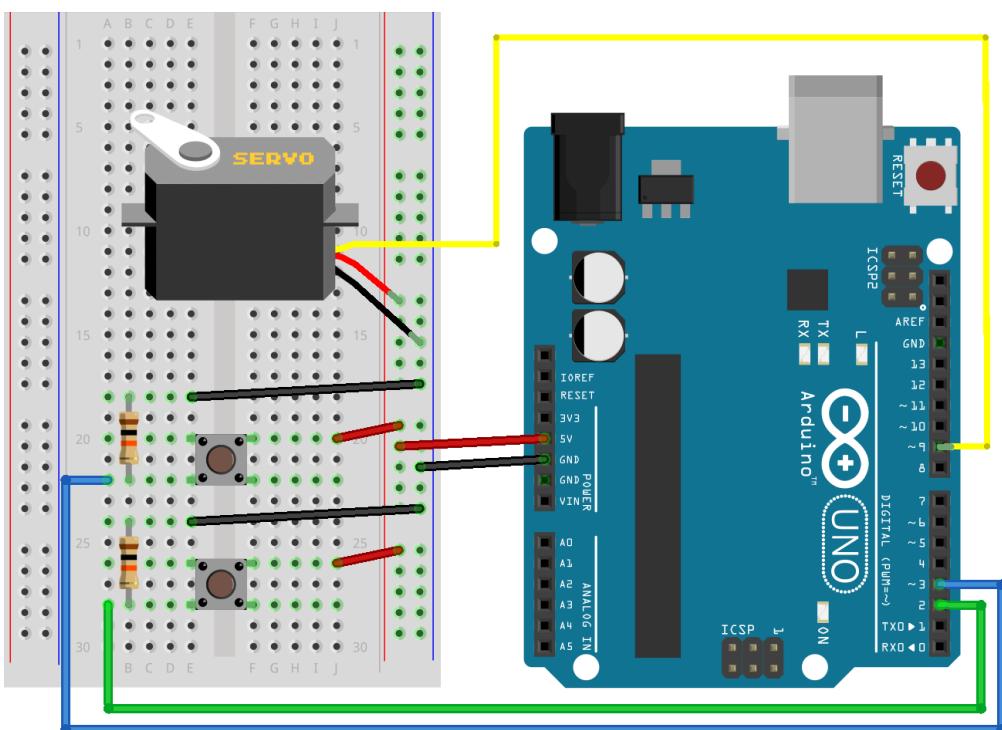
ŘEŠENÍ PŘÍKLADŮ

(Př. 1) Naprogramujte servomotor tak, aby se otáčel od 0° do 120° pomaleji a při zpětném chodu aby se otáčel rychleji. Řešení příkladu spočívá ve velmi jednoduché úpravě.

```
1 #include <Servo.h>
2
3 Servo myservo;
4 int pos = 0;
5
6 void setup()
7 {
8     myservo.attach(9);
9 }
10
11 void loop()
12 {
13     // Úprava mezní hodnoty natočení servomotoru na 120
14     for(pos = 0; pos <= 120; pos++)
15     {
16         myservo.write(pos);
17         // Úprava prodlevy, aby se servo ještě více zpomalilo
18         delay(20);
19     }
20     // Úprava mezní hodnoty natočení servomotoru ze 120 na 0
21     for(pos = 120; pos >= 0; pos--)
22     {
23         myservo.write(pos);
24         // Úprava prodlevy, aby se servo zrychlilo
25         delay(5);
26     }
27 }
```

(PŘ. 2) ZKUSTE OVLÁDAT SERVOMOTOR POMOCÍ DVOU TLAČÍTEK.

Tlačítka určují směr otáčení servomotoru.



Obr. 15 - Zapojení tlačítek pro ovládání servomotoru

```
1 #include <Servo.h>
2
3 Servo myservo;
4 int servoPin = 9;
5 int leftButton = 2;
6 int rightButton = 3;
7 int servoPos = 90;
8 int delayPeriod = 2;
9
10 void setup()
11 {
12     myservo.attach(servoPin);
13     myservo.write(servoPos);
14     pinMode(leftButton, HIGH);
15     pinMode(rightButton, HIGH);
16 }
17
18 void loop()
19 {
20     if(digitalRead(leftButton) == LOW)
21     {
22         if(servoPos > 0)
23         {
24             servoPos--;
25         }
26         myservo.write(servoPos);
27         delay(delayPeriod);
28     }
29
30     if(digitalRead(rightButton) == LOW)
31     {
32         if(servoPos < 180)
33         {
34             servoPos++;
35         }
36         myservo.write(servoPos);
37         delay(delayPeriod);
38     }
39 }
```