

# PRŮVODCE HODNOCENÍM

TENTO PRŮVODCE JE SPÍŠE MOŽNÝM NÁVODEM A PŘEHLEDEM VZOROVÝCH OTÁZEK NEBO ÚLOH, KTERÉ MOHOU BÝT VYUŽITY K HODNOCENÍ ŽÁKŮ FORMOU ZNÁMKOVÁNÍ.

Jsou zde uvedeny příklady, které mohou být námětem pro otestování nabytých znalostí žáků se zaměřením zejména na programování. Studenti by neměli být primárně zkoušeni z elektroniky. Sestavování obvodů je sice důležitou součástí robotiky, ale v rámci výuky programování robotických nebo vestavěných (embedded) systémů by neměla být hlavním kritériem pro hodnocení.

Úkoly by měli být realizovány v praktické rovině, tzn. že by se nemělo jednat o pouhé vyplňování testů, ale žák by měl mít možnost si své řešení ověřit. V tomto ohledu se zde nabízí možnost praktického ověřování přímo na obvodech. Obvody může mít učitel připraveny, zejména pokud se bude jednat o složitější konstrukce. Jednodušší může žák sestavit sám, ale primárně podle schématu zapojení.

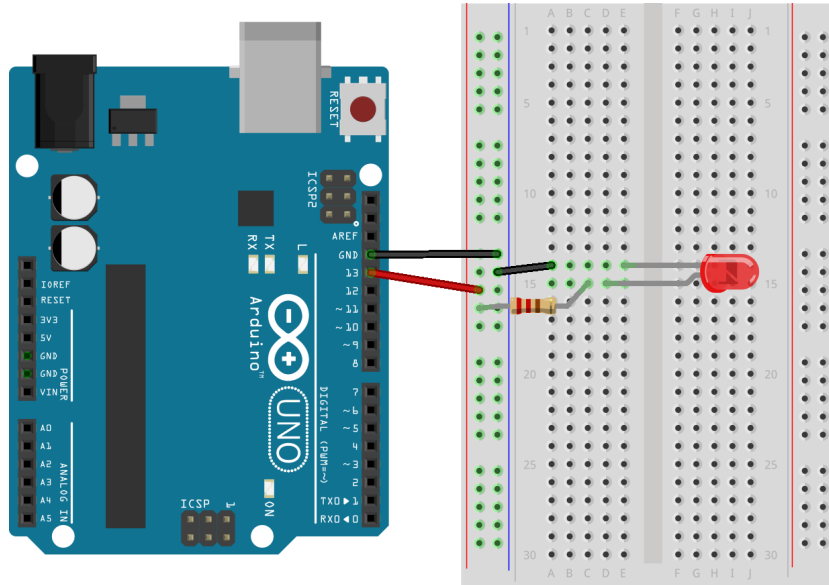
Otázky jsou rozděleny do témat, které kopírují strukturu učebnice. Z jednotlivých otázek lze poskládat rozsáhlejší dokument pro testování. Znalosti získané v jednotlivých tématech lze dále testovat i s jinými elektronickými komponentami z jiných kapitol a uvedené otázky by měli sloužit jako návod, jak vytvořit další. Většinou se jedná o poměrně jednoduché otázky, které vedou k doplnění chybějícího fragmentu programového kódu, nebo otestování kódu za účelem zjištění funkcionality.

Otázka nebo skupina otázek je rozdělena na následující části:

1. Zadání – obsahuje schéma zapojení, programový kód nebo obojí.
2. Otázka – na základě zadání je formulována otázka, která pracuje s tímto zadáním pracuje. K jednomu zadání může být uvedeno více otázek.
3. Odpověď – vzorová řešení pro zodpovězení otázky.

# OTÁZKY LED

## ZADÁNÍ



```
void setup() {  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);  
  delay(1000);  
  digitalWrite(13, HIGH);  
  delay(1000);  
}
```

## OTÁZKA

➔ Z uvedeného obvodu a programového kódu zjistěte, zda bude dioda v obvodu blikat.



## ODPOVĚĎ

➔ Dioda blikat nebude, protože v obou funkcích digitalWrite jsou uvedeny v druhém parametru hodnoty HIGH nebo 1.





### OTÁZKA

→ Jak změníte uvedený kód, aby dioda začala blikat?

### ODPOVĚĎ

V jedné z funkcí **digitalWrite** musí být nastaven druhý parametr na hodnotu **LOW** nebo **0**.

```
void setup() {  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);  
  delay(1000);  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```



### OTÁZKA

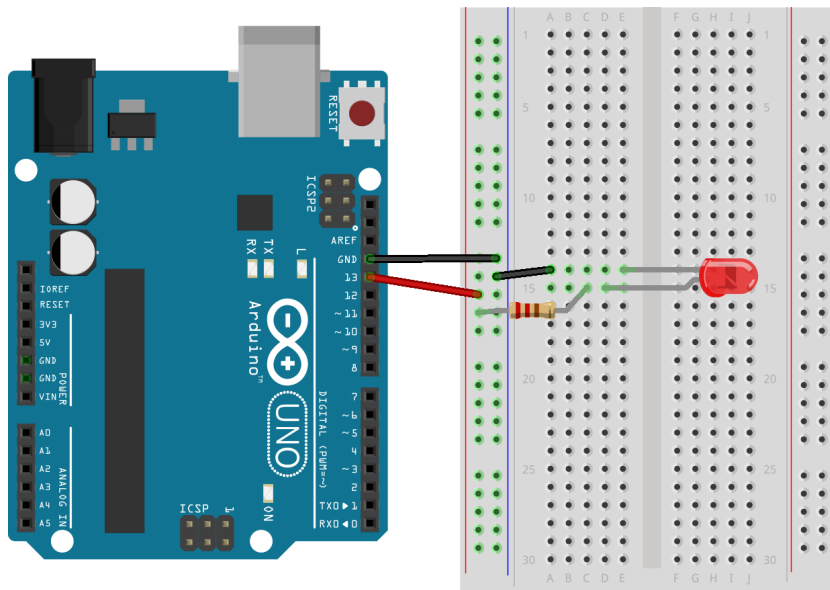
→ Jak upravíte program, aby dioda blikala pomaleji?

### ODPOVĚĎ

→ Dioda bude blikat pomaleji, jestliže se změní hodnota parametru funkce **delay** na vyšší číslo. Hodnota čísla je udávána v milisekundách.



## ZADÁNÍ



```
void setup() {  
  pinMode(12, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(12, HIGH);  
  delay(1000);  
  digitalWrite(12, LOW);  
  delay(1000);  
}
```

### OTÁZKA

➔ Bude dioda blikat nebo svítit?



### ODPOVĚĎ

➔ Dioda nebude svítit ani blikat, protože je zapojena na vstup 13, ale v programu je nastavena hodnota vstupního pinu na 12.





### OTÁZKA

→ Jak upravíte kód, aby byl příklad funkční?

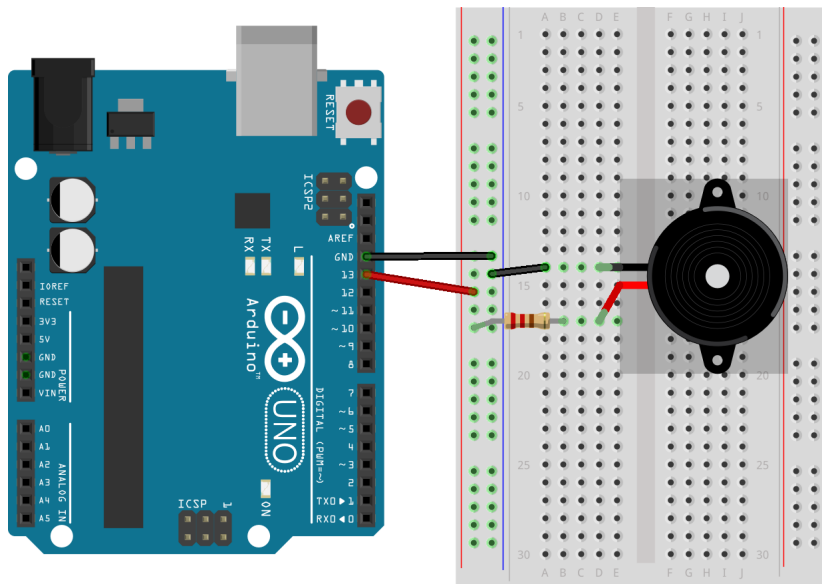


### ODPOVĚĎ

- Změní se číslo vstupního pinu na hodnotu 13.
- Druhým řešením může být připojení červeného vodiče v obvodu na pin 12.

# OTÁZKY PIEZO BZUČÁK

## ZADÁNÍ



```
const int bzucak=12;

void setup() {
  pinMode(bzucak, OUTPUT);
}

void loop() {
  tone(bzucak, 440);
  delay(1000);
  noTone(bzucak);
  delay(1000);
}
```

## OTÁZKA

→ Z uvedeného obvodu a programového kódu zjistěte, zda bude bzučák vydávat nějaký tón?



## ODPOVĚĎ

→ Bzučák nebude vydávat tón, protože vstupní pin je nastaven na číslo 12, ale v obvodu je zapojen na pin 13.





### OTÁZKA

→ Upravte programový kód tak, aby bzučák vydával nějaký zvuk.



### ODPOVĚĎ

→ V programovém kódu stačí změnit hodnotu konstanty bzucak na 13.

```
const int bzucak=13;
```



### OTÁZKA

→ Jak upravíte programový kód, aby bzučák vydával zvuk delší dobu?



### ODPOVĚĎ

→ V programovém kódu stačí změnit hodnotu funkce delay na hodnotu 3000.

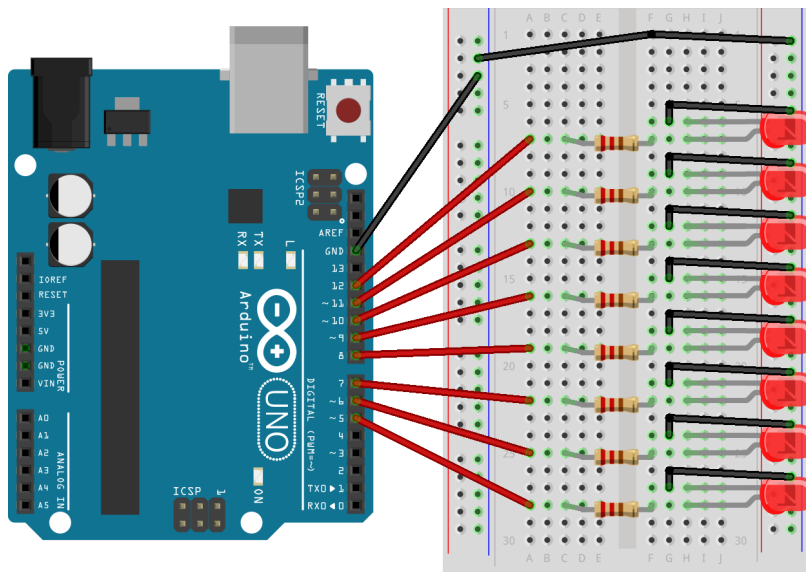
```
const int bzucak=13;
```

```
void setup() {  
  pinMode(bzucak, OUTPUT);  
}
```

```
void loop() {  
  tone(bzucak, 440);  
  delay(3000); // zde se zvýší hodnota  
  noTone(bzucak);  
  delay(1000);  
}
```

# OTÁZKY LED ANIMACE

## ZADÁNÍ



## OTÁZKA

→ Doplňte programový kód tak, abyste využili níže uvedenou funkci a světlo opakovaně běželo z jedné strany na druhou?

```
void setup() {  
  pinMode(5, OUTPUT);    // dioda 1  
  pinMode(6, OUTPUT);    // dioda 2  
  pinMode(7, OUTPUT);    // dioda 3  
  pinMode(8, OUTPUT);    // dioda 4  
  pinMode(9, OUTPUT);    // dioda 5  
  pinMode(10, OUTPUT);   // dioda 6  
  pinMode(11, OUTPUT);   // dioda 7  
  pinMode(12, OUTPUT);   // dioda 8  
}  
  
void changeLED(int pin) {  
  digitalWrite(pin, HIGH);  
  delay(50);  
  digitalWrite(pin, LOW);  
  delay(50);  
}
```





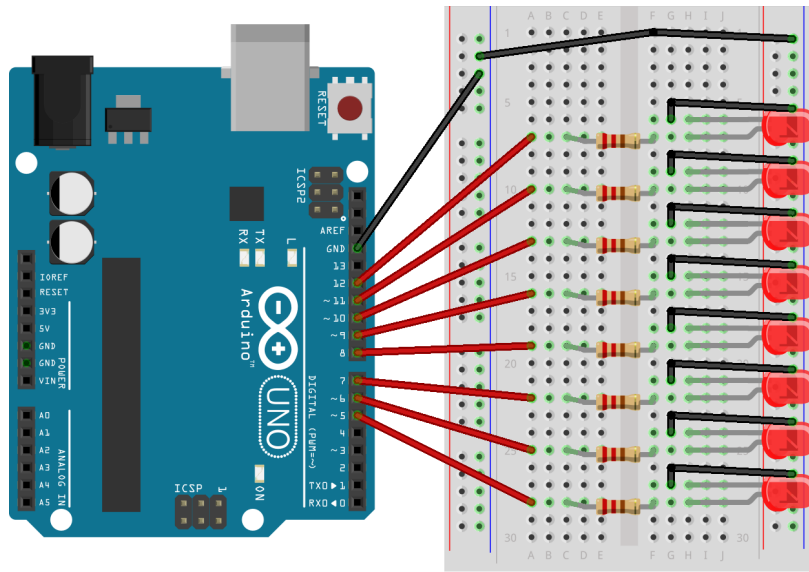


### ODPOVĚĎ

→ V programovém kódu stačí opakovaně volat deklarovanou funkci `changeLED`.

```
void loop() {  
  changeLED(5);  
  changeLED(6);  
  changeLED(7);  
  changeLED(8);  
  changeLED(9);  
  changeLED(10);  
  changeLED(11);  
  changeLED(12);  
}
```

## ZADÁNÍ



```
int pinArray[] = {5, 6, 7};
int count = 0;
int timer = 50;

void setup() {
  for (count=0;count<sizeof(pinArray);count++) {
    pinMode(pinArray[count], OUTPUT);
  }
}

void loop() {
  for (count=0; count<sizeof(pinArray); count++) {
    changeLED(pinArray[count]);
  }
}

void changeLED(int pin) {
  digitalWrite(pin, HIGH);
  delay(timer);
  digitalWrite(pin, LOW);
  delay(timer);
}
```

### OTÁZKA

→ Jak byste doplnili čísla pinů v poli `pinArray` tak, aby se diody opakovaně rozsvěcovali z prava do leva a zpět? Zapojení diod je vidět na schématu.

```
int pinArray[] = {5, 6, 7};
```



### ODPOVĚĎ

→ Pole `pinArray` se doplní o chybějící čísla pinů v následujícím pořadí.

```
int pinArray[] = {5,6,7,8,9,10,11,12,11,10,9,8,7,6 };
```



### OTÁZKA

→ Jak byste upravili programový kód, aby každá dioda v každém cyklu blikla 2x?



### ODPOVĚĎ

→ Prvním řešením je úprava pole `pinArray`.

```
int pinArray[] = {5,5,6,6,7,7};
```

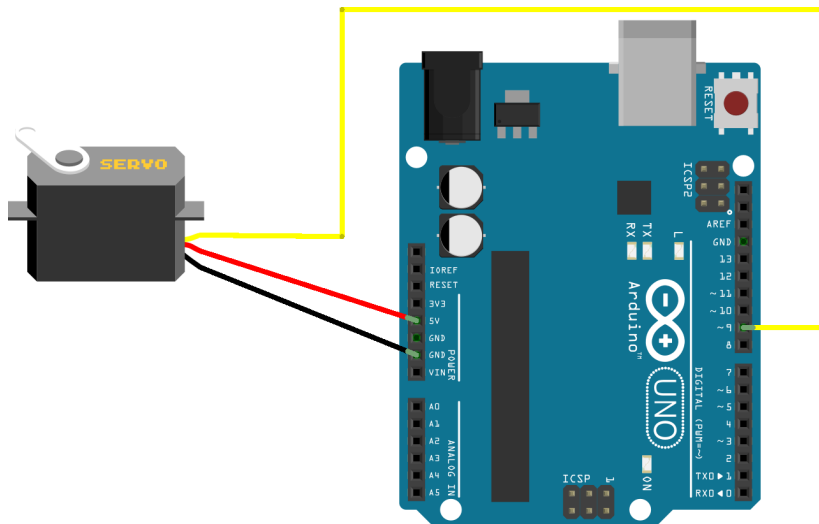
→ Druhým řešením je úprava funkce `changeLED`.

```
void changeLED(int pin) {  
    digitalWrite(pin, HIGH);  
    delay(timer);  
    digitalWrite(pin, HIGH);  
    delay(timer);  
    digitalWrite(pin, LOW);  
    digitalWrite(pin, LOW);  
    delay(timer);  
}
```



# OTÁZKY SERVO – FOR, IF

## ZADÁNÍ



```
#include <Servo.h>

Servo myservo;

void setup()
{
  myservo.attach(9);
}

void loop()
{
  myservo.write(pos); // Nastavení pozice servomotoru
  delay(15);
}
```

### OTÁZKA

→ Nadefinujte proměnou `pos` tak, aby se osa servomotoru otočila do pozice 90°.



### ODPOVĚĎ

→ Stačí napsat jediný řádek, který definuje proměnou `pos` s hodnotou 90°.

```
int pos = 90;
```



### OTÁZKA

→ Jak byste upravili programový kód, aby se osa servomotoru natáčela postupně na 10°, 40°, 80°, 120° a 180°?



## ODPOVĚĎ

- Prvním řešením může být postupné natáčení s využitím volání metody `write` pro každý úhel.

```
#include <Servo.h>
Servo myservo;
void setup()
{
    myservo.attach(9);
}

void loop()
{
    myservo.write(10);
    delay(1000);
    myservo.write(40);
    delay(1000);
    myservo.write(80);
    delay(1000);
    myservo.write(120);
    delay(1000);
    myservo.write(180);
    delay(1000);
}
```

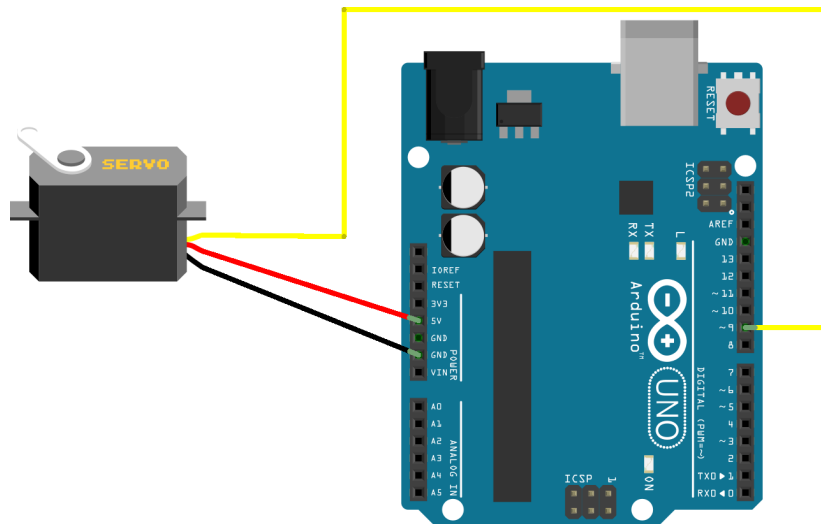
- Druhým řešením může být využití pole a cyklu `for`.

```
#include <Servo.h>
int degreeArray[] = {10,40,80,120,180};
Servo myservo;
void setup()
{
    myservo.attach(9);
}

void loop()
{
    for(pos = 0; pos <= 5; pos += 1){
        myservo.write(degreeArray[pos]);
        delay(1000);
    }
}
```



## ZADÁNÍ



```
#include <Servo.h>

Servo myservo;
void setup()
{
  myservo.attach(9);
}

void loop()
{
  for(pos = 0; pos <= 180; pos += 1){
    myservo.write(pos);
    delay(20);
  }
}
```

### OTÁZKA

→ Jak upravíte programový kód, aby se servomotor otáčel i zpět?



### ODPOVĚĎ

→ Do programového kódu se přidá cyklus `for`, který zajistí opačné natáčení servomotoru.

```
#include <Servo.h>
```

```
Servo myservo;
```

```
Int pos = 0;
```

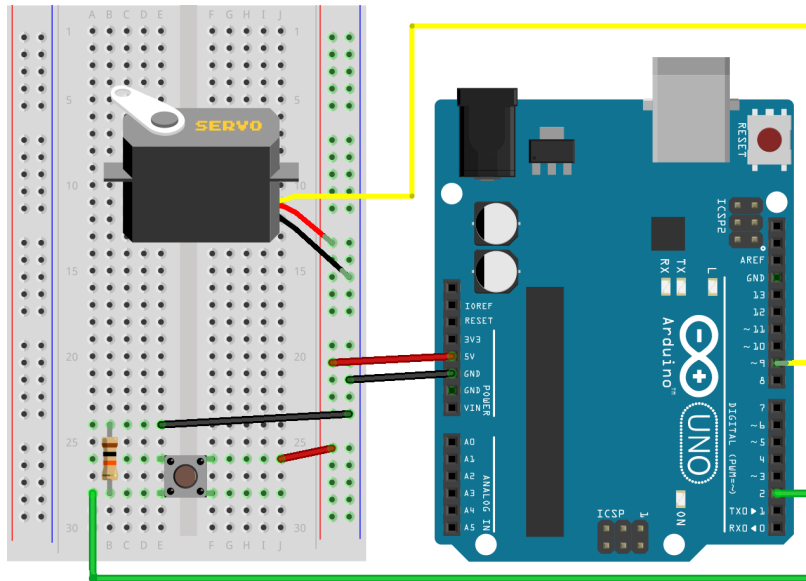
```
void setup(){  
  myservo.attach(9);  
}
```

```
void loop(){  
  for(pos = 0; pos <= 180; pos += 1){  
    myservo.write(pos);  
    delay(20);  
  }  
  for(pos = 180; pos >= 0; pos -= 1) {  
    myservo.write(pos);  
    delay(20);  
  }  
}
```





## ZADÁNÍ



```
#include <Servo.h>

int servoPin = 9;
int Button = 2;
int servoPos = 0;
int delayPeriod = 2;

Servo myservo;

void setup(){
  myservo.attach(servoPin);
  myservo.write(servoPos);
  pinMode(Button, INPUT);
}

void loop(){

  if(servoPos < 180){
    servoPos++;
  }
  myservo.write(servoPos);
  delay(delayPeriod);
}
```

### OTÁZKA

→ Jak upravíte programový kód, aby se servomotor otáčel pouze při stisknutém tlačítku?



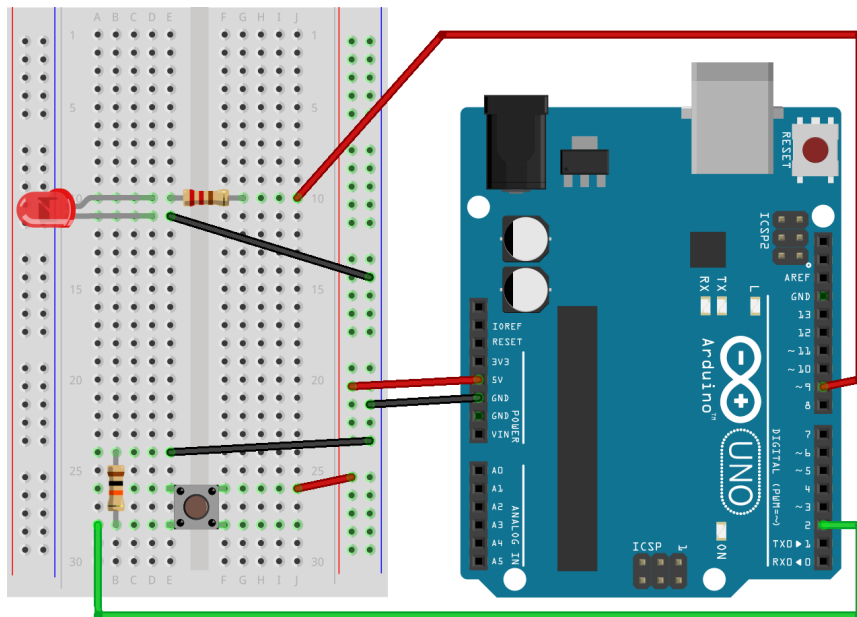
### ODPOVĚĎ

→ Do funkce `loop` se přidá podmínka pro test stisknutého tlačítka.

```
void loop(){  
  if(digitalRead(Button) == LOW){  
    if(servoPos < 180){  
      servoPos++;  
    }  
    myservo.write(servoPos);  
    delay(delayPeriod);  
  }  
}
```



## ZADÁNÍ



```
int pinLed = 9;
int Button = 2;

void setup(){
    pinMode(Button, INPUT);
    pinMode(pinLed, OUTPUT);
}

void loop(){
    digitalWrite(pinLed, HIGH);
    delay(1000);
    digitalWrite(pinLed, LOW);
}
```

### OTÁZKA

→ Jak upravíte programový kód, aby LED svítila pouze při stisknutém tlačítku?



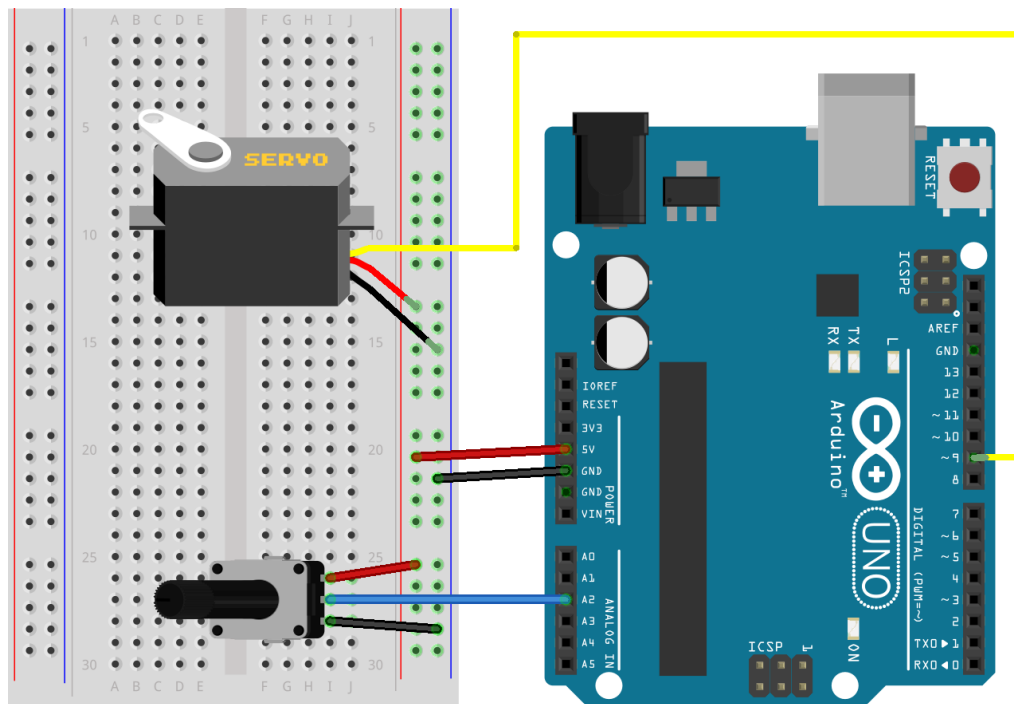
### ODPOVĚĎ

→ Funkce `loop` se upraví tak, že se přidá podmínka pro test stisknutého tlačítka.

```
void loop(){  
  if(digitalRead(Button) == LOW){  
    digitalWrite(pinLed, HIGH);  
  }else{  
    digitalWrite(pinLed, LOW);  
  }  
}
```



## ZADÁNÍ



```
#include <Servo.h>

Servo myservo;
int pos = 0;

void setup()
{
  myservo.attach(9);
}

void loop()
{
  pos = analogRead(A2);
  pos = map(pos, W, X, Y, Z);
  myservo.write(pos);
  delay(5);
}
```

### OTÁZKA

→ Jaké hodnoty doplníte ve funkci `map` za písmena W, X, Y, Z, aby při otáčení potenciometru se natáčel i servomotor v celém svém rozsahu?



### ODPOVĚĎ

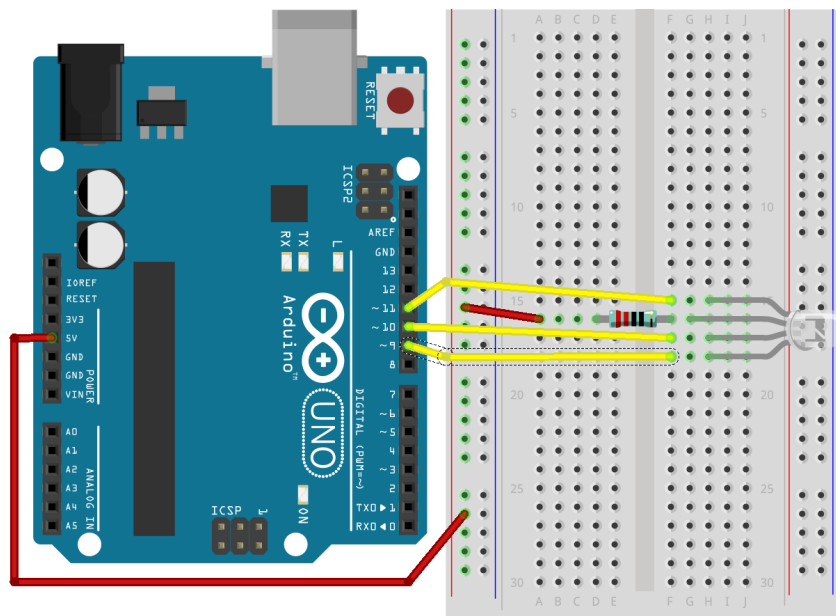
→ Parametry ve funkci `map` mohou vypadat následujícím způsobem.

```
pos = map(pos, 0, 1023, 0, 179);
```



# OTÁZKY RGB LED – VLASTNÍ FUNKCE

## ZADÁNÍ



```
void setup() {  
    pinMode(11, OUTPUT);    //červená  
    pinMode(10, OUTPUT);    //zelená  
    pinMode(9, OUTPUT);     //modrá  
}  
  
void loop() {  
    digitalWrite(11, X);    //červená  
    digitalWrite(10, Y);    //zelená  
    digitalWrite(9, Z);     //modrá  
}
```

### OTÁZKA

→ Jaké hodnoty doplníte ve funkcích `digitalWrite` za písmena X, Y, Z, aby dioda svítila modře?



### ODPOVĚĎ

→ Pokud se jedná o RGB diodu se společnou anodou, tak řešení je následující.

```
void loop() {  
  digitalWrite(11, HIGH); //nebo místo HIGH je 255  
  digitalWrite(10, HIGH); //nebo místo HIGH je 255  
  digitalWrite(9, LOW);   //nebo místo LOW je 0  
}
```



### OTÁZKA

→ Jaké hodnoty doplníte ve funkcích `digitalWrite` za písmena X, Y, Z, aby dioda svítila zeleně?



### ODPOVĚĎ

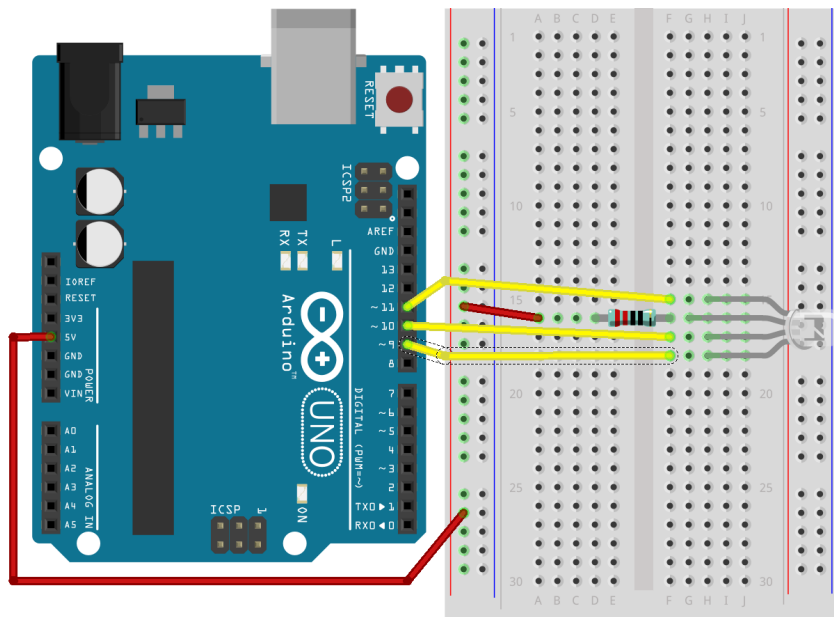
→ Pokud se jedná o RGB diodu se společnou anodou, tak řešení je následující.

```
void loop() {  
  digitalWrite(11, HIGH); //nebo místo HIGH je 255  
  digitalWrite(10, LOW);  //nebo místo HIGH je 0  
  digitalWrite(9, HIGH);  //nebo místo LOW je 255  
}
```





## ZADÁNÍ



```
void setup() {  
    pinMode(11, OUTPUT);    //červená  
    pinMode(10, OUTPUT);    //zelená  
    pinMode(9, OUTPUT);     //modrá  
}  
  
void loop() {  
    setColor(255,0,0);  
}
```

### OTÁZKA

- Naprogramujte funkci `setColor` tak, aby se v uvedeném programu svítila RGB dioda definovanou barvou v zadaných parametrech volané funkce.



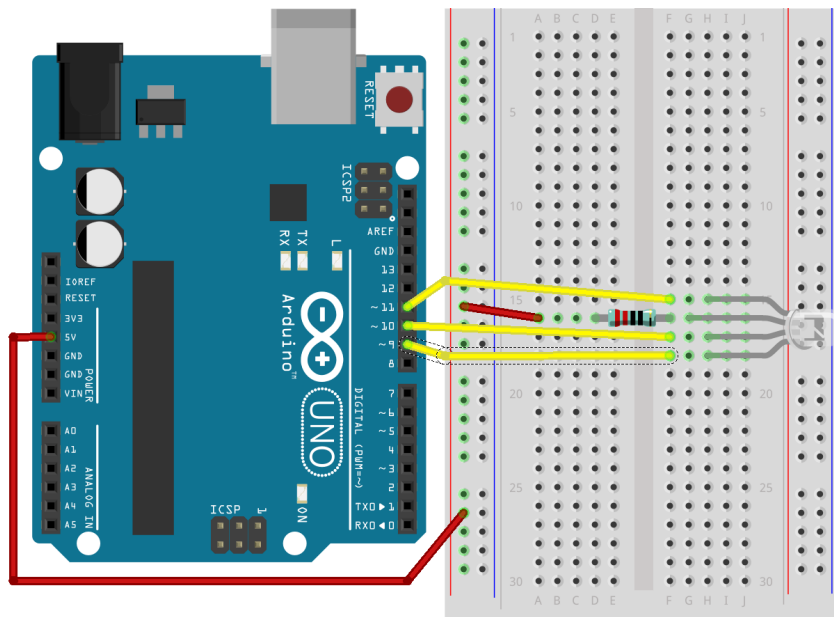
### ODPOVĚĎ

- Funkce `setColor` bude vypadat následovně.

```
void setColor(int redC, int greenC, int blueC ) {  
    digitalWrite(11, redC);  
    digitalWrite(10, greenC);  
    digitalWrite(9, blueC);  
}
```



## ZADÁNÍ



```
void setup() {  
    pinMode(11, OUTPUT);    //červená  
    pinMode(10, OUTPUT);    //zelená  
    pinMode(9, OUTPUT);     //modrá  
}  
  
void loop() {  
    setColor(255,0,0);  
    delay(1000);  
    setColor(0,0,255);  
    delay(1000);  
    setColor(0,250,0);  
    delay(1000);  
}
```

### OTÁZKA

- Naprogramujte funkci `setColor` tak, aby se v uvedeném programu střídali barvy, podle zadaných parametrů.



### ODPOVĚĎ

- Funkce `setColor` bude vypadat následovně. Řešení pro diodu se společnou anodou.

```
void setColor(int redC, int greenC, int blueC ) {  
    digitalWrite(11, redC);  
    digitalWrite(10, greenC);  
    digitalWrite(9, blueC);  
}
```



### OTÁZKA

- Jakými barvami bude RGB dioda svítit, při správně definované funkci `setColor`?



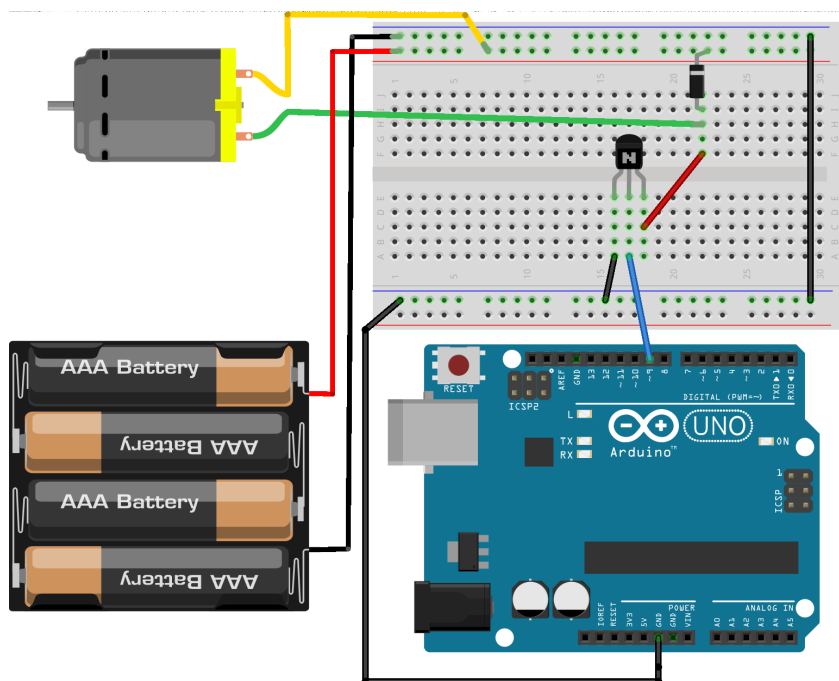
### ODPOVĚĎ

- Podle parametrů, které jsou ve funkci `setColor` bude RGB dioda postupně svítit barvami: tyrkysová, žlutá, fialová.



# MOTOR DC

## ZADÁNÍ



```
const int transistorPin = 9;
const int speedMotor = 200;

void setup() {
  pinMode(transistorPin, OUTPUT);
}

void loop() {
  analogWrite(transistorPin, speedMotor);
}
```

### OTÁZKA

➔ Jak upravíte program, aby se otáčky motoru postupně zrychlovaly od 0 do maximální rychlosti 255?



### ODPOVĚĎ

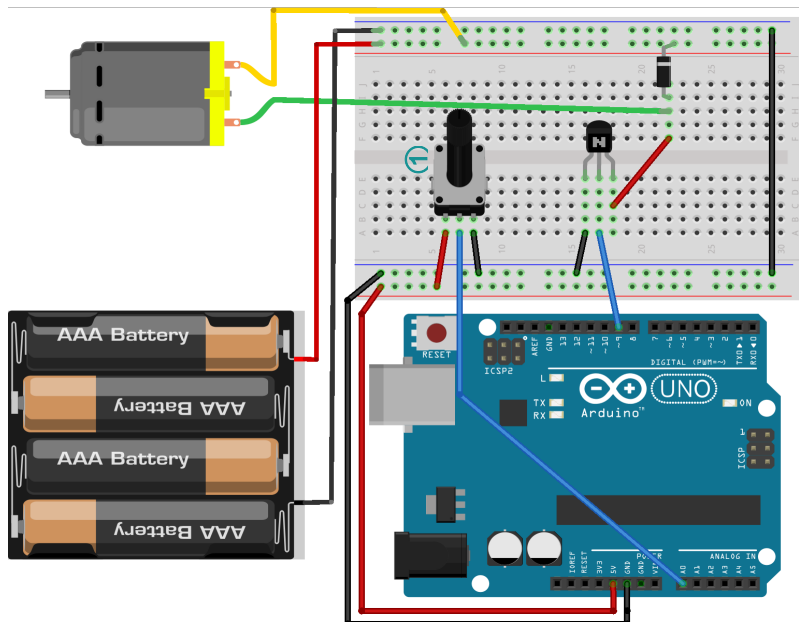
➔ Pro řešení lze použít příkaz cyklu `for`, který zajistí postupné zvyšování rychlosti v proměnné `speedMotor`.

```
void loop() {  
    for(speedMotor = 0; speedMotor <= 255; speedMotor += 1){  
        analogWrite(transistorPin, speedMotor);  
    }  
}
```



# MOTOR DC

## ZADÁNÍ



```
const int transistorPin = 9;

void setup() {
  pinMode(transistorPin, OUTPUT);
}

void loop() {
  int sensorValue = analogRead(A0);

  . . .

  analogWrite(transistorPin, outputValue);
}
```

### OTÁZKA

→ Jak upravíte program, aby se otáčky motoru regulovali pomocí připojeného potenciometru?



### ODPOVĚĎ

→ Pro řešení lze stačit doplnit funkci `map`, jejíž návratová hodnota se uloží do proměnné `outputValue`.

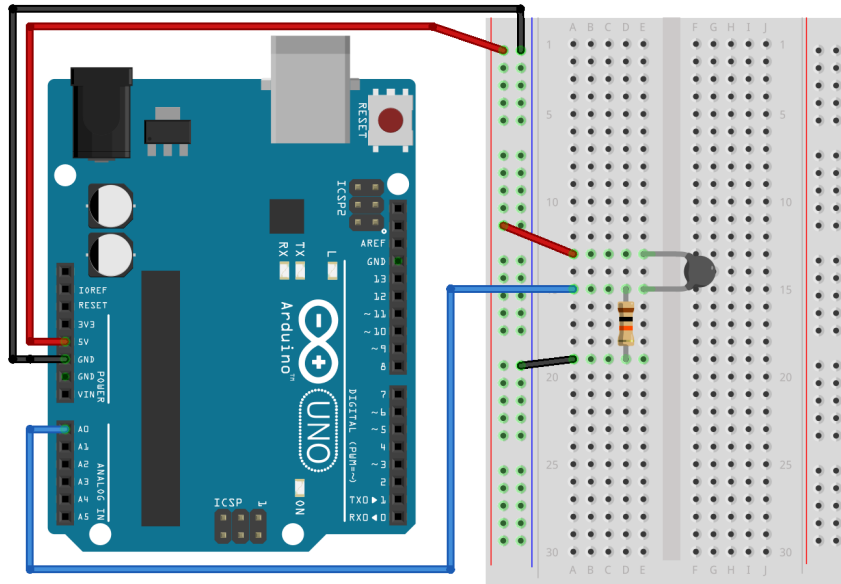
```
int outputValue = map(sensorValue, 0, 1023, 0, 255);
```





# MĚŘÍME TEPLOTU

## ZADÁNÍ



```
int termistorPin = 0;
int Vout;
float R2 = 10000;
float logR2, R1, T;
float c1 = 1.009249522e-03, c2 = 2.378405444e-04,
c3 = 2.019202697e-07;

void setup() {
  Serial.begin(9600);
}

void loop() {
  Vout = analogRead(termistorPin);
  R1 = R2 * (1023.0 / (float)Vout - 1.0);
  logR1 = log(R1);
  T = (1.0 / (c1 + c2*logR1 + c3*logR1*logR1*logR1));

  . . .

  delay(500);
}
```

### OTÁZKA

- Jak doplníte programový kód, abyste zjištěnou teplotu uloženo v proměnné `T` zobrazili pomocí sériového monitoru?



### ODPOVĚĎ

- Vzhledem k tomu, že ve funkci `setup` je inicializace pro sériovou komunikaci, měli by si žáci vzpomenout na funkce pro výpis v sériovém monitoru.

```
Serial.print("Teplota: ");  
Serial.println(T);
```



### OTÁZKA

- Vytvořte dvě funkce, které budou zajišťovat výpočet teploty ve stupních Celsia a Fahrenhaita?

Výpočet stupňů Fahrenhaita:  $(\text{Stupně celsia} * 9.0) / 5.0 + 32.0$

Výpočet stupňů Celsia:  $T - 273.15$



### ODPOVĚĎ

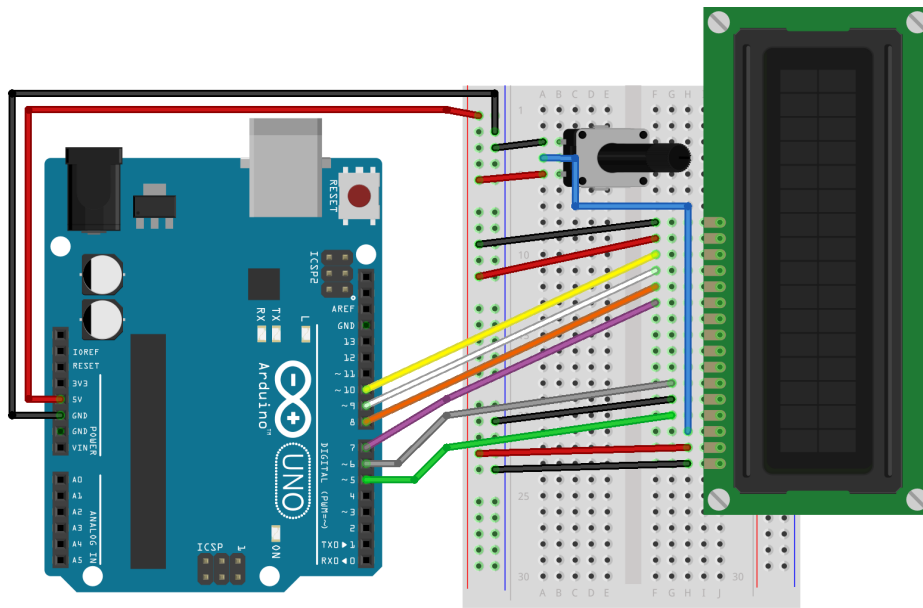
- Podle uvedených vzorců je výpočet pomocí funkcí velmi jednoduché. V první řadě se funkce musí deklarovat:

```
void getCelsius(T) {  
    return T - 273.15;  
}  
  
void getFahreIn(Tc) {  
    return (Tc * 9.0) / 5.0 + 32.0;  
}
```

Funkce se musí zavolat s odpovídajícími parametry:



## ZADÁNÍ



```
#include <LiquidCrystal.h>

int rsPin = 5;
int ePin = 6;
int d4Pin = 7;
int d5Pin = 8;
int d6Pin = 9;
int d7Pin = 10;
LiquidCrystal LCD(rsPin,ePin,d4Pin,d5Pin,d6Pin,d7Pin);

void setup(){

}

void loop(){

}
```

### OTÁZKA

→ Jak doplníte programový kód, aby se na prvním řádku displeje zobrazilo slovo Dobrý a na druhém řádku den?



### ODPOVĚĎ

→ Pro zobrazení požadovaného textu lze využít metod z instance třídy LCD.

```
void setup(){  
  LCD.begin(16,2);           // inicializace displeje  
  LCD.clear();               // vymazání displeje  
  LCD.setCursor(0,0);        // nastavení začátku kurzoru  
  LCD.print("Dobrý");        // vypsání textu  
  
  LCD.setCursor(0,1);        // nastavení kurzoru druhý řádek  
  LCD.print("den");          // vypsání textu  
}
```



### OTÁZKA

- Jak upravíte programový kód, aby se na LCD displeji zobrazovalo počítadlo od 1 do 100? Po dosažení hodnoty 100 se počítadlo vynuluje a začne opět od 1. Využijte podmínkový příkaz `if`.



### ODPOVĚĎ

- Při využití podmínkového příkazu `if`, stačí zadat do funkce `loop` inkrementátor v podobě proměnné `i`. Upravený kód bude vypadat následovně.

```
int i = 0;

void setup(){
  LCD.begin(16,2);           // inicializace displeje
}

void loop(){
  LCD.clear();               // vymazání displeje
  LCD.setCursor(0,0);        // nastavení začátku kurzoru
  LCD.print(i);              // vypsání čísla
  delay(100);

  if(i==100){
    i=1;
  }else{
    i++;
  }
}
```



### OTÁZKA

- Jak upravíte programový kód, aby se na LCD displeji zobrazovalo počítadlo od 1 do 100? Po dosáhnutí hodnoty 100 se počítadlo vynuluje a začne opět od 1. Využijte příkaz cyklu `for`.



### ODPOVĚĎ

- Při využití příkazu cyklu `for`, stačí zadat vypisovat hodnotu proměnné `i`. Upravený kód bude vypadat následovně.

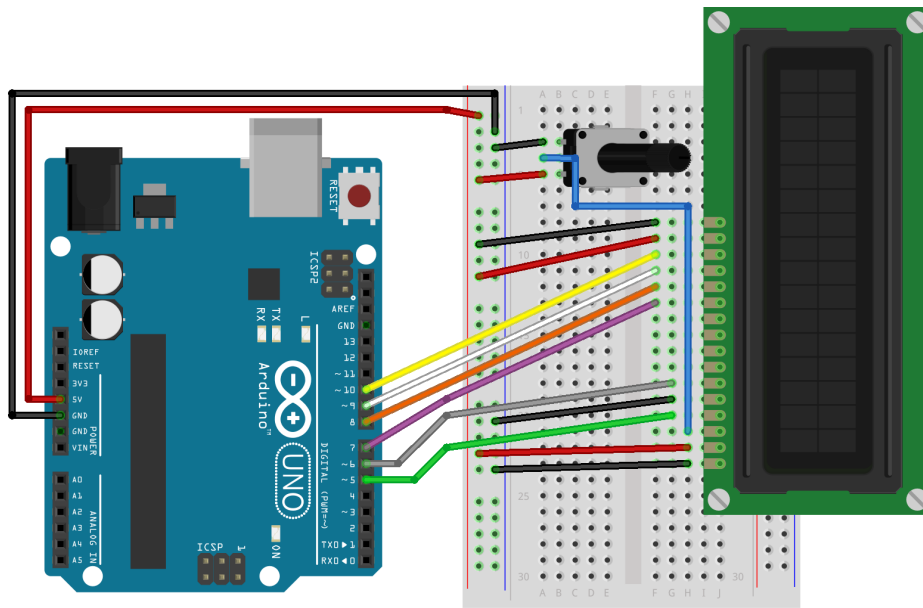
```
int i = 0;

void setup(){
  LCD.begin(16,2);          // inicializace displeje
}

void loop(){
  for(i = 0; i <= 100; i++){
    LCD.clear();            // vymazání displeje
    LCD.setCursor(0,0);     // nastavení začátku kurzoru
    LCD.print(i);           // vypsání čísla
    delay(100);
  }
  delay(5000);
}
```



## ZADÁNÍ



```
#include <LiquidCrystal.h>
```

```
int rsPin = 5;
int ePin = 6;
int d4Pin = 7;
int d5Pin = 8;
int d6Pin = 9;
int d7Pin = 10;
LiquidCrystal LCD(rsPin,ePin,d4Pin,d5Pin,d6Pin,d7Pin);
```

```
void setup(){
  LCD.begin(16,2);           // inicializace displeje
}
```

```
void loop(){
  for(i = 0; i <= 100; i++){
    LCD.clear();             // vymazání displeje
    LCD.setCursor(0,0);      // nastavení začátku kurzoru
    LCD.print(i);            // vypsání čísla
    delay(100);
  }
  delay(5000);
}
```

### OTÁZKA

→ Jak upravíte programový kód, aby se na LCD displeji zobrazovalo kromě počítadla od 1 do 100 i slovo Pocitadlo?



### ODPOVĚĎ

→ Do programového kódu se musí přidat nové nastavení pozice kurzoru. Jedno nastavení bude pro text Pocitadlo a druhé pro číslo v proměnné i.



```
int i = 0;

void setup(){
  LCD.begin(16,2);           // inicializace displeje
}

void loop(){
  for(i = 0; i <= 100; i++){
    LCD.clear();             // vymazání displeje

    LCD.setCursor(0,0);      // nastavení začátku kurzoru
    LCD.print("Pocitadlo");  // vypsání slova Pocitadlo

    LCD.setCursor(10,0);     // nastavení začátku kurzoru
    LCD.print(i);            // vypsání čísla
    delay(100);
  }
  delay(5000);
}
```



### OTÁZKA

→ Jak upravíte programový kód, aby se na LCD displeji zobrazovalo počítadlo od 100 do 1?



### ODPOVĚĎ

→ V cyklu `for` stačí provést obrácený odpočet proměnné `i`.



```
int i = 0;

void setup(){
  LCD.begin(16,2);           // inicializace displeje
}

void loop(){
  for(i = 100; i >= 1; i--){ // dekrementování proměnné i
    LCD.clear();             // vymazání displeje
    LCD.setCursor(0,0);      // nastavení začátku kurzoru
    LCD.print(i);            // vypsání čísla
    delay(100);
  }
  delay(5000);
}
```