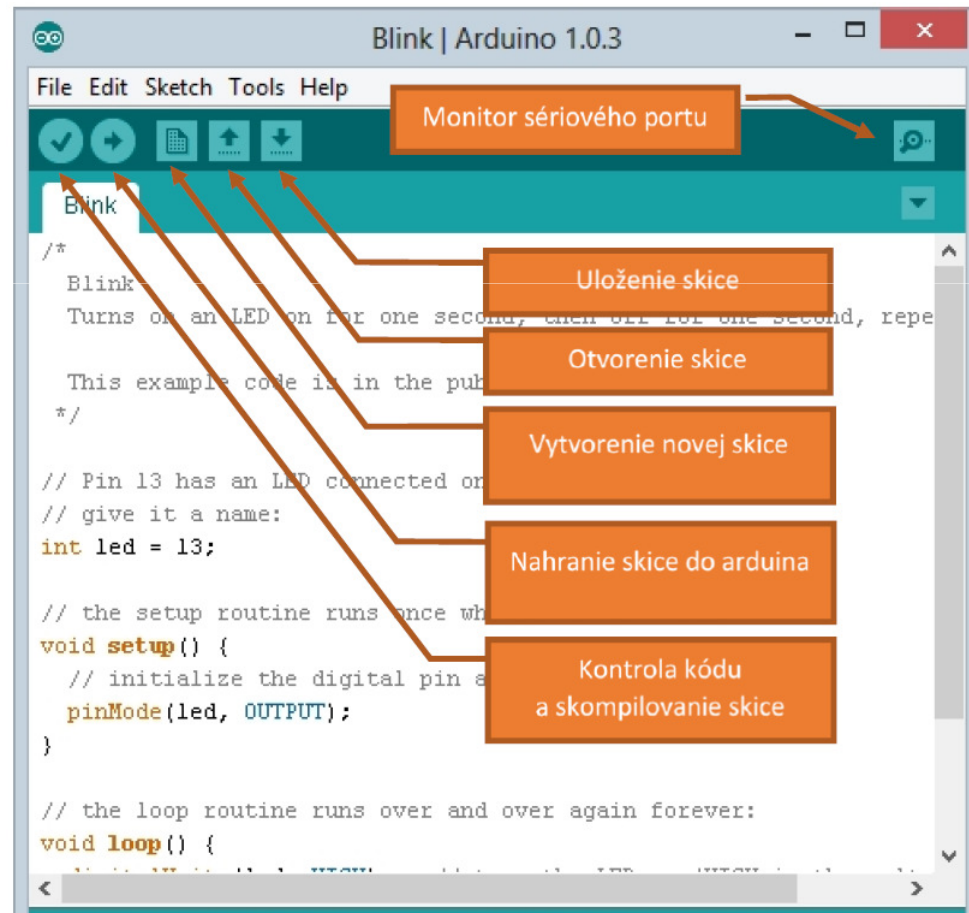


Čo je Arduino?

- **open-source projekt**, konkrétne sa jedná o elektronickú platformu
- sú dostupné elektronické schémy zapojenia Arduino dosky
- zdrojové kódy knižníc a vývojového prostredia
- tieto zdroje môžete ďalej upravovať a šíriť alebo predávať
- základom je **vývojová doska** (plošný spoj - hardvér) a **vývojové prostredie** (softvér)

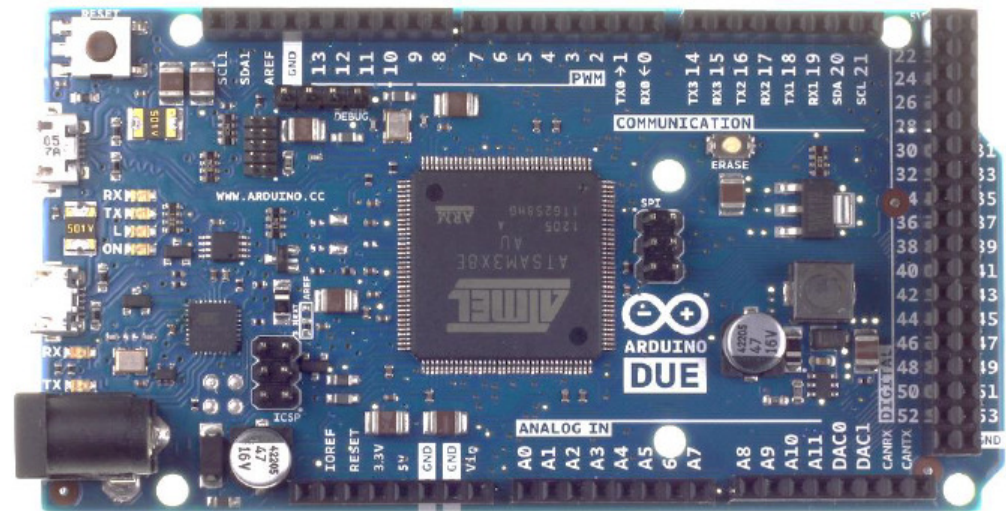
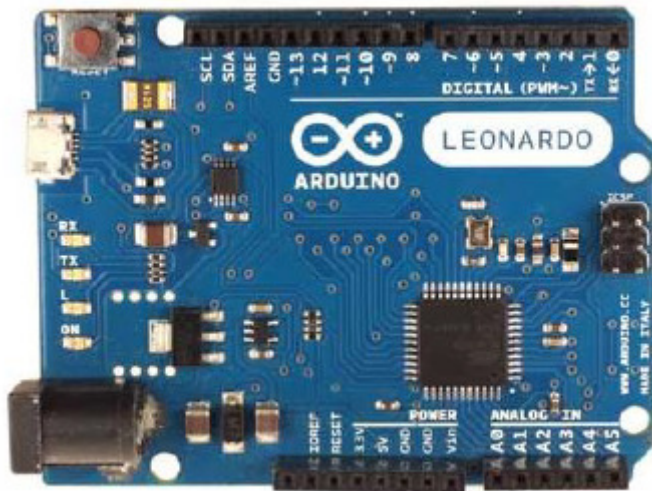
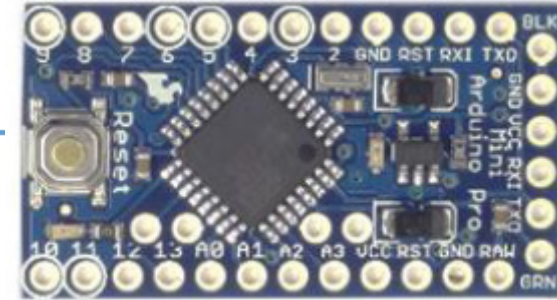


8 bitový mikrokontrolér

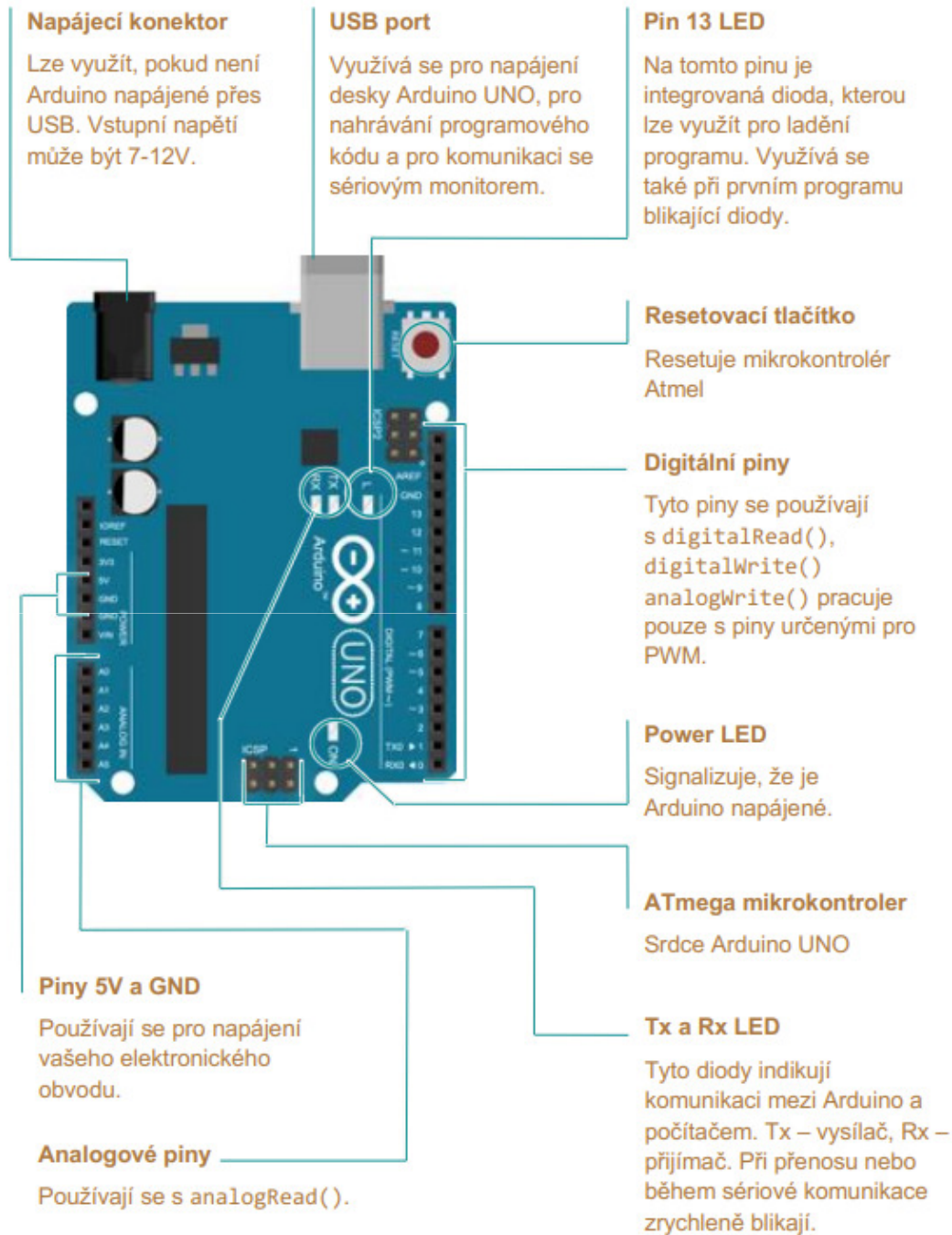


Základná štruktúra Arduino programu

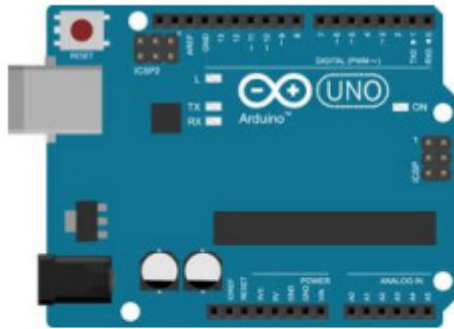
```
void setup() {  
    // nastavenie pinov, seriového portu, atď  
}  
  
void loop() {  
    // sem sa vpíše hlavný kód programu, ktorý sa bude vykonávať dookola  
}
```



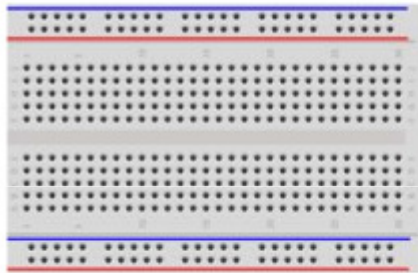
Arduino



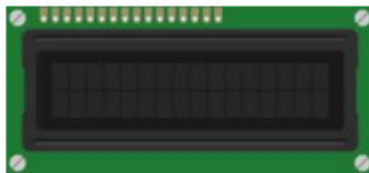
Hardware



Arduino Uno – základní vývojová deska mikrokontroleru která tvoří srdce všech projektů. Jedná se o jednoduchý počítač, který nemá prozatím rozhraní pro interakci s ním. Budete vytvářet elektronická zapojení, rozhraní pro interakci a volání mikrokontroleru pro komunikaci s ostatními komponentami.



Kontaktní pole – deska, která umožňuje sestavovat elektronické obvody bez nutnosti používat pájku. Součástky se zapojují do řádků a sloupců desky.



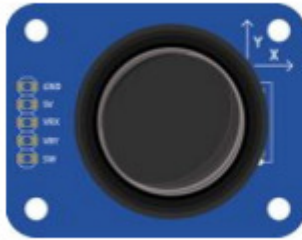
LCD display – jeden z typů displeje pro zobrazení alfanumerických znaků. Existuje celá řada LCD displejů . V sadě je displej se 2 řádky po 16 znacích.



1 bitový sedmissegmentový displej – je display pro zobrazování číslic a speciálních znaků. Lze jej využít pro zobrazení teploty



Fotorezistor – je součástka, která mění svůj odpor podle toho, jak intenzivní záření na ní dopadá.



Křížový ovladač, joystick – je herní modul, který umožňuje snímat pohyb ve dvou směrech. Samotný joystick je tvořen dvěma potenciometry, jejichž hodnota určuje směr.



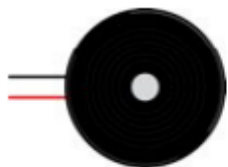
LED diody – jsou polovodičové součástky, schopné vyzařovat světlo. Používají se převážně k signalizaci. V současné době se využívají i k osvětlení.



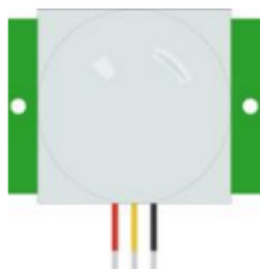
Usměrňovací diody - je elektrotechnická součástka, jejímž úkolem je v elektrickém obvodu propouštět elektrický proud jen jedním směrem.



Stejnoseměrný motor – je určený k pohybu robotických částí. Nejčastěji se používá v mobilních robotech nebo v zařízeních, kde je vyžadován plynulý rotační pohyb.



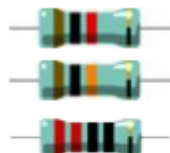
Pasivní bzučák – součástka, která vytváří zvuky na základě elektrických pulzů.



Pohybové čidlo – snímá své okolí pod úhlem přibližně 120 stupňů. Pokud v tomto snímaném prostoru proběhne větší změna teplot, tzn. například projde člověk či zvíře, je tato změna zaznamenána díky změně výstupního napětí.



Potenciometr – součástka, která při otáčení osou mění poměr odporu na její dráze. Lze je použít například při změně jasu displeje nebo LED diody.



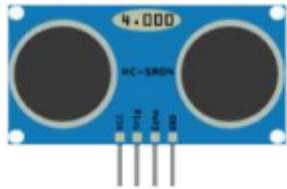
Rezistor – je pasivní elektrotechnická součástka, která je zařazována do obvodu z důvodu snížení elektrického proudu nebo získání úbytku napětí.



Tranzistor – je polovodičová součástka, kterou lze jednoduše používat jako zesilovač, spínač, stabilizátor aj.



Servomotor – slouží pro nastavení polohy ovládaného mechanismu a následné držení v této poloze. Stejnoseměrné servo motory se mohou využít pro ovládání robotické paže nebo pro nastavení kormidla u leteckých modelů.



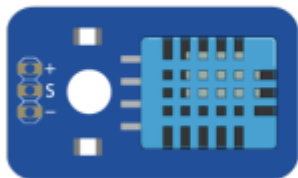
Sonar – je senzor k měření vzdálenosti. Měření je založeno na principu měření doby mezi vysláním akustického signálu a přijetím odraženého signálu.



Tlačítko – je jednoduchá mechanická součástka, která slouží ke spojení a rozpojení obvodu.

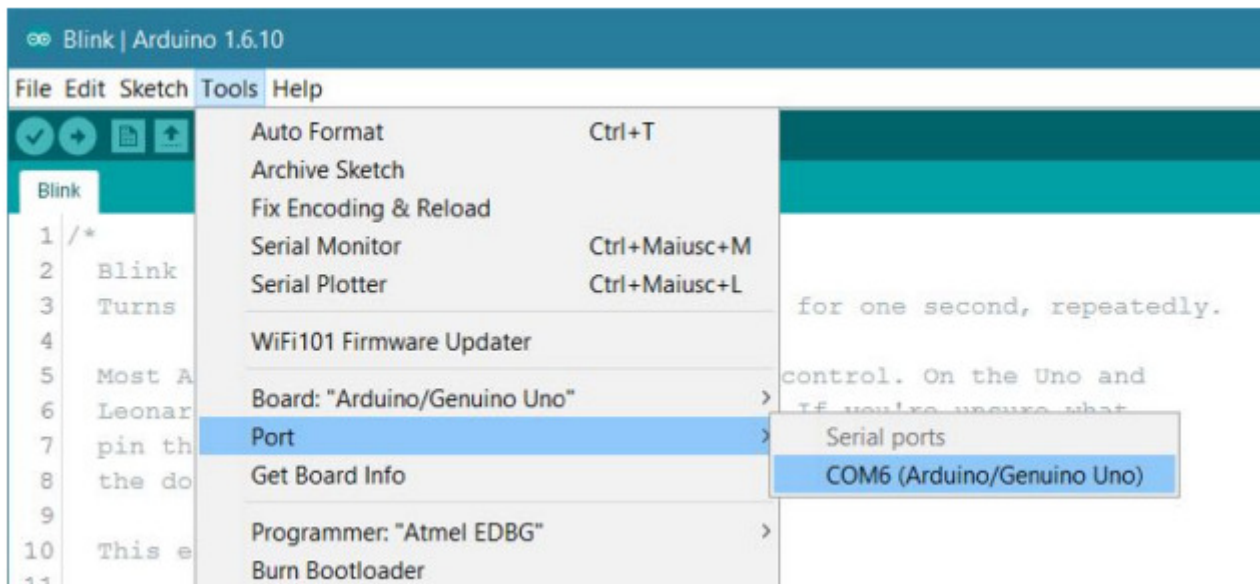
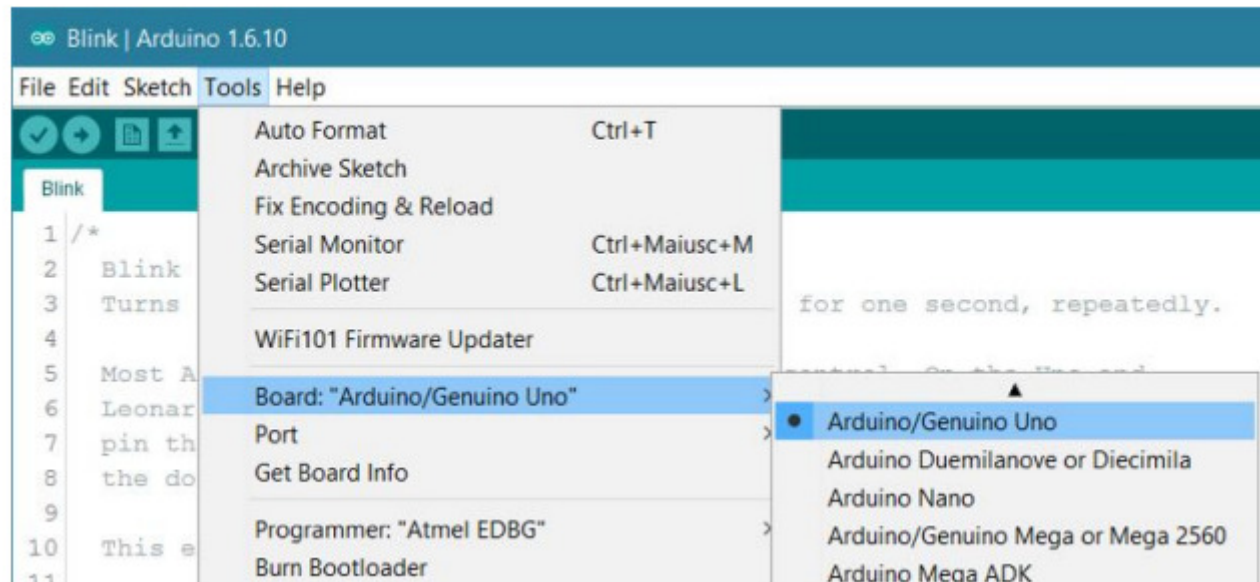


Termistor – se používá pro měření teploty. Jedná se o rezistor, který v závislosti na teplotě svůj odpor mění podle přesně definované charakteristiky.



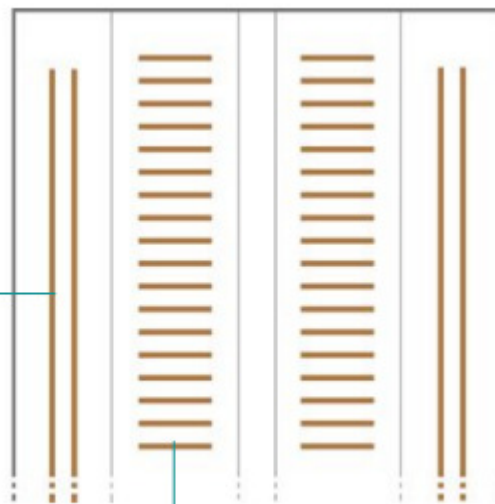
Teplotní a vlhkostní čidlo – obsahuje vše potřebné. Stačí ho připojit rovnou k desce Arduino a načítat hodnoty z okolí, které se obnovují přibližně každé 2 sekundy.

Software



Kontaktné pole

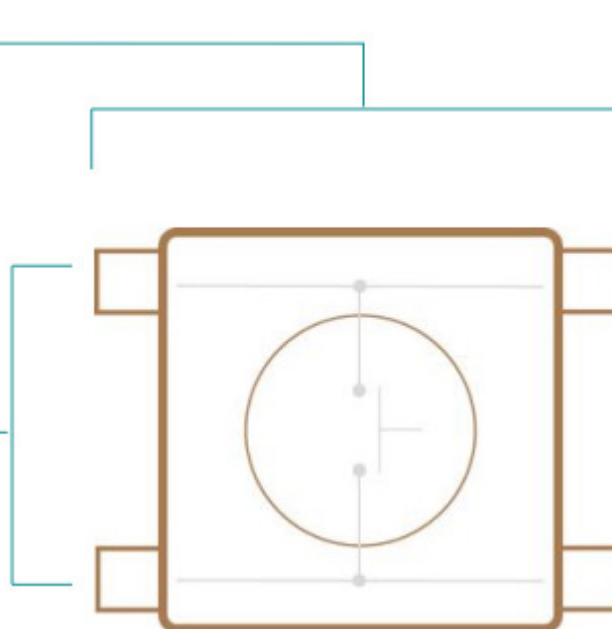
Uvnitř kontaktní desky jsou vodiče propojující zdičky, jak je vidět na obrázku.



Tlačidlo

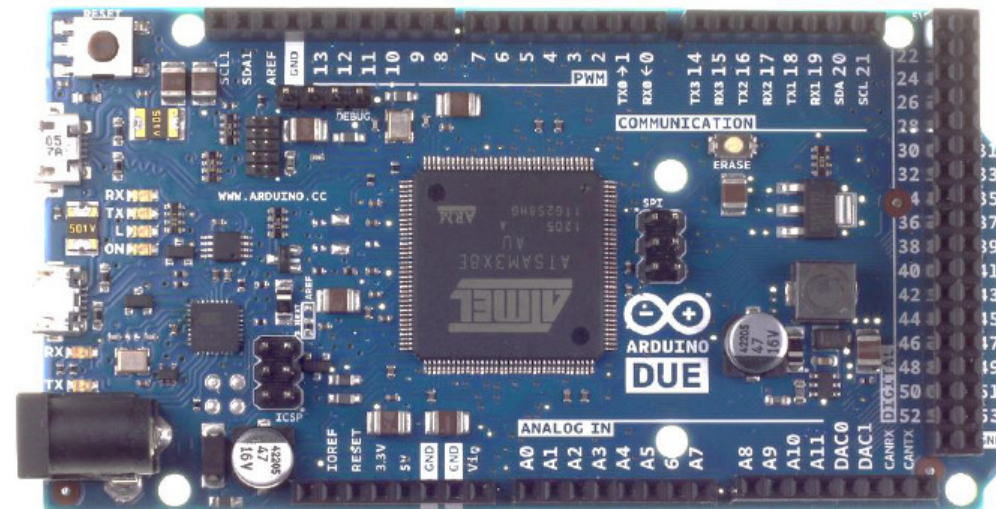
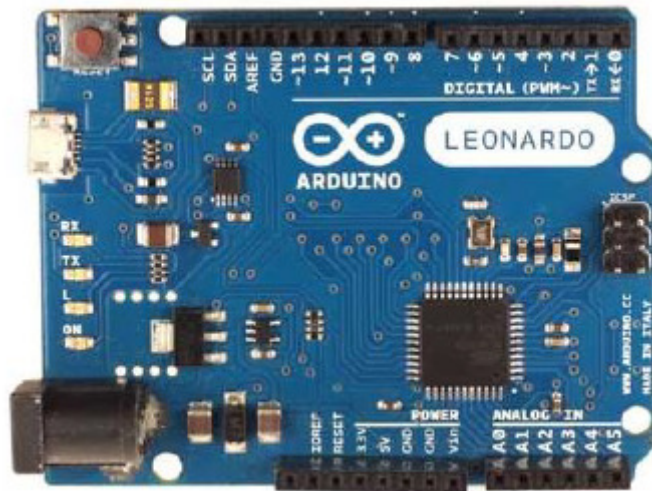
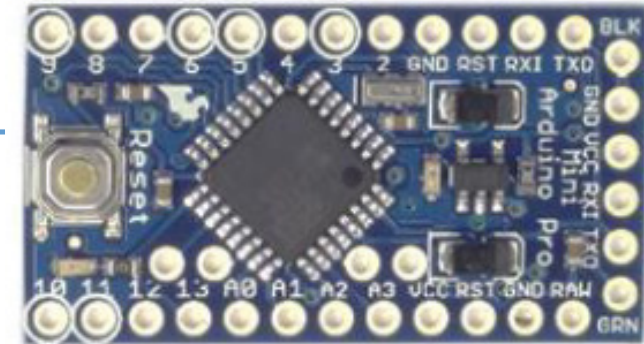
Tyto dva konektory jsou vzájemně propojeny

Tyto dva konektory nejsou vzájemně propojeny. Tvoří samotný spínač



Základná štruktúra programu

```
void setup() {  
    // nastavenie pinov, seriového portu, atď  
}  
  
void loop() {  
    // sem sa vpíše hlavný kód programu, ktorý sa bude vykonávať dookola  
}
```



Programovanie Arduina, programové vybavenie mikroprocesora:

- **definičná časť** – definícia konštánt a premenných (napr: `int led = 13;`), vloženie podprogramu
- **časť setup** – vykoná sa len 1x, určíme vstupy a výstupy (napr: `pinMode(led, OUTPUT);`)
- **časť loop** – slučka, príkazy sa vykonávajú dookola (napr: `digitalWrite(led, HIGH);`)

Príklad programu pre blikanie LED:

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

Podmienené príkazy if:

```
if ( podmienka ) { príkazy 1 }
else { príkazy 2 }
```

Pole:

```
string uzivatelia[5] = { Jozef, Peter, František,
Karol, Michal };
```

```
Serial.println(uzivatelia[0]); >>>Jozef
Serial.println(uzivatelia[4]); >>>Michal
Serial.println(uzivatelia[1]); >>>Peter
```

Komentáre:

```
// toto je jednoriadkový komentár,
```

```
/* toto je komentár na viac riadkov */
```

Cykly:

```
for (int i = 0; i < 10; i++)  
{ Serial.println(i); }
```

```
while (digitalRead(10))  
{ blikanie }
```

Dátové typy:

a) číselné

- byte (8 bitov)
- integer (celé číslo)
- long (celočíselné 32 bitové číslo)
- float (číslo s desatinnou čiarkou)

b) Boolean

- logická 0 (false)
- logická 1 (true)

c) Znakový dátový typ

- char – jeden znak textu
- string – viac znakov textu

Čítanie vstupu:

- digitalRead(vstup)

Vstupy a výstupy:

- pinMode(led, OUTPUT);
- pinMode(tlacitko, INPUT);

Podmienený príkaz

Case:

```
switch (premenna) {  
  case 1:  
    .....  
    break  
  case 2:  
    .....  
    break;  
  default:  
    ..... }  
}
```

Čakanie v

milisekundách:

```
delay()
```

Odosielanie dát:

```
• Serial.print(97);  
//vypise: 97  
• Serial.print(2.123456);  
//vypise: 2.12  
• Serial.print('a');  
//vypise: "a"  
• Serial.print("ABCDEF");  
//vypise: "ABCDEF,,  
• Serial.print(97, DEC);  
//vrati: 97  
• Serial.print(97, BIN);  
//vrati: 1100001  
• Serial.print(97, HEX);  
//vrati: 61 (=97 v  
sestnactkovej sustave)  
• Serial.print(4.56789, 0);  
//vrati: 5  
• Serial.print(4.56789, 1);  
//vrati: 4.6  
• Serial.print(4.56789, 3);  
//vrati: 4.568
```

Matematické operácie:

- +, -, *, /,
- min(), max(),
- pow(a,b) – mocnina a na b
- sqrt(a) – druhá odmocnina a

Generovanie náhodných čísiel:

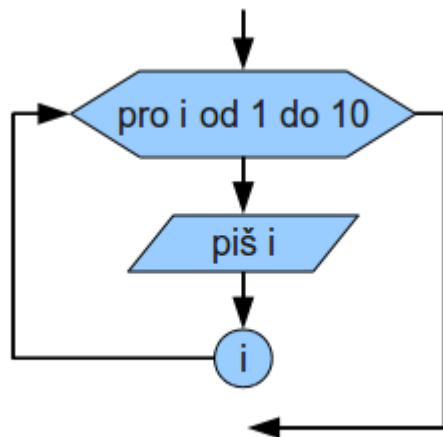
- random(min, max)

Generovanie tónov:

- tone()
- notone()

Cyklus - FOR

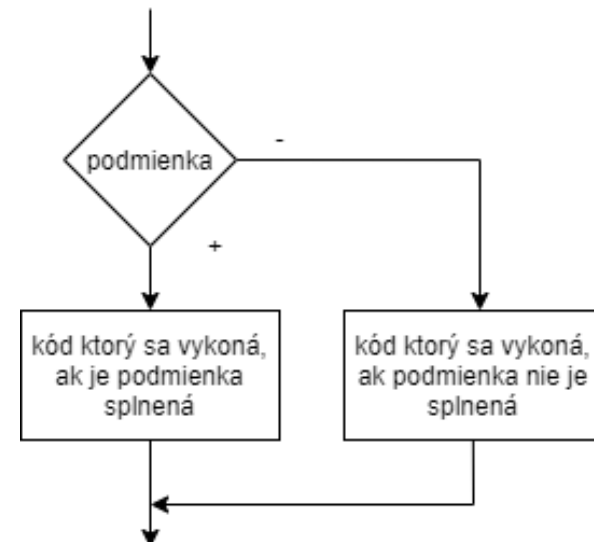
```
void setup() {  
  Serial.begin(9600);  
  
  for (int count=0; count<255; count++) {  
    Serial.println(count);  
    delay(10);  
  }  
}  
  
void loop() {  
  
}
```



- poznáme počet prechodov
- podmienka na začiatku cyklu
- najoblúbenejší typ cyklu

Podmienený príkaz - IF

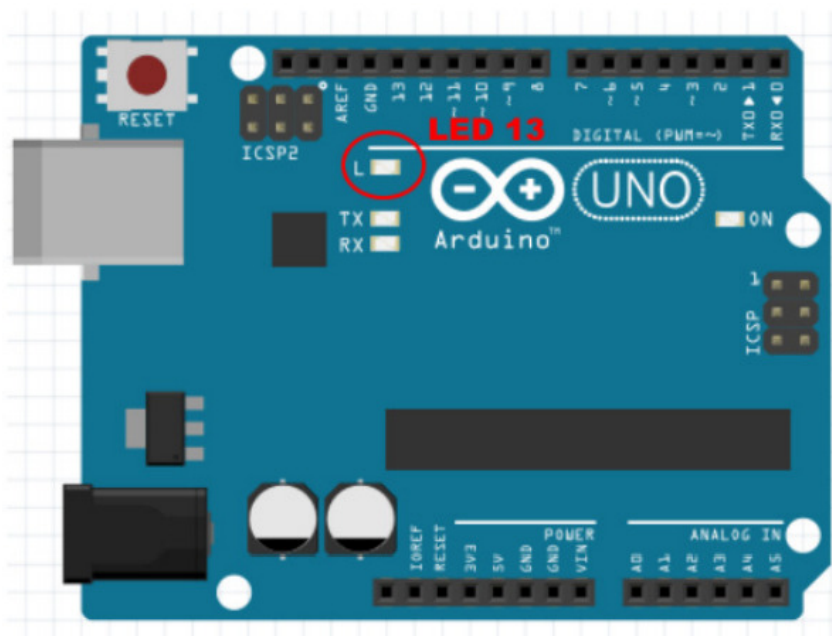
```
if (podmienka A) {  
  
  // blok príkazov A  
  
}else if (podmienka B) {  
  
  // blok príkazov B  
  
}else{  
  
  // blok príkazov C  
  
}
```



LED pin 13

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // turn the LED off by making the voltage LOW
  delay(1000);           // wait for a second
}
```

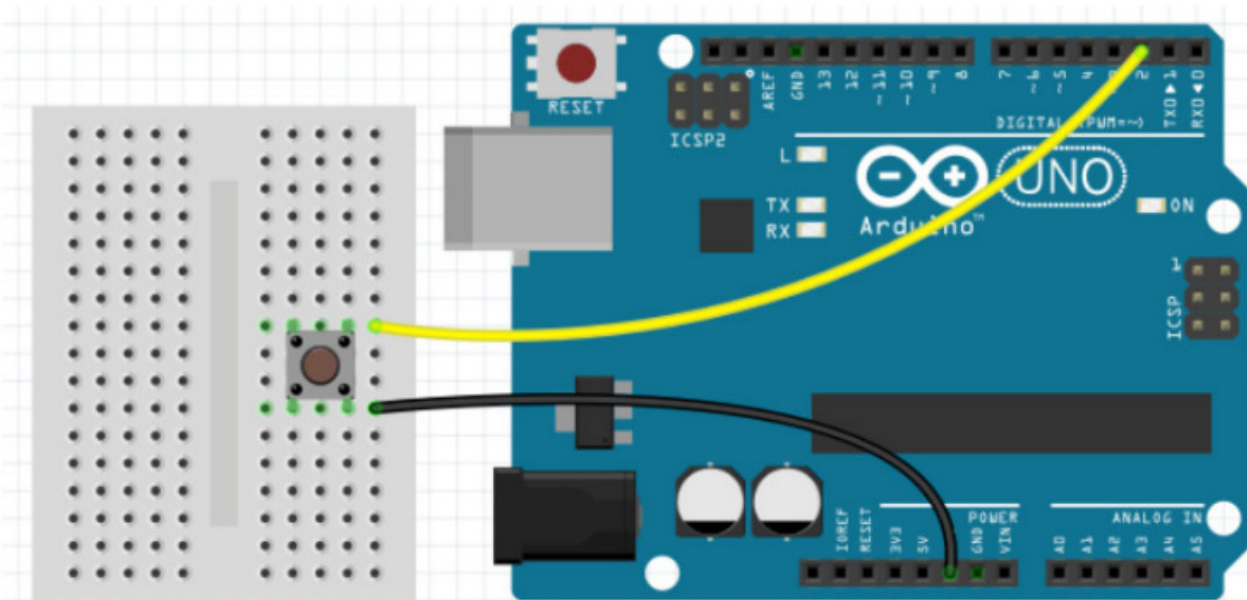


Tlačidlo

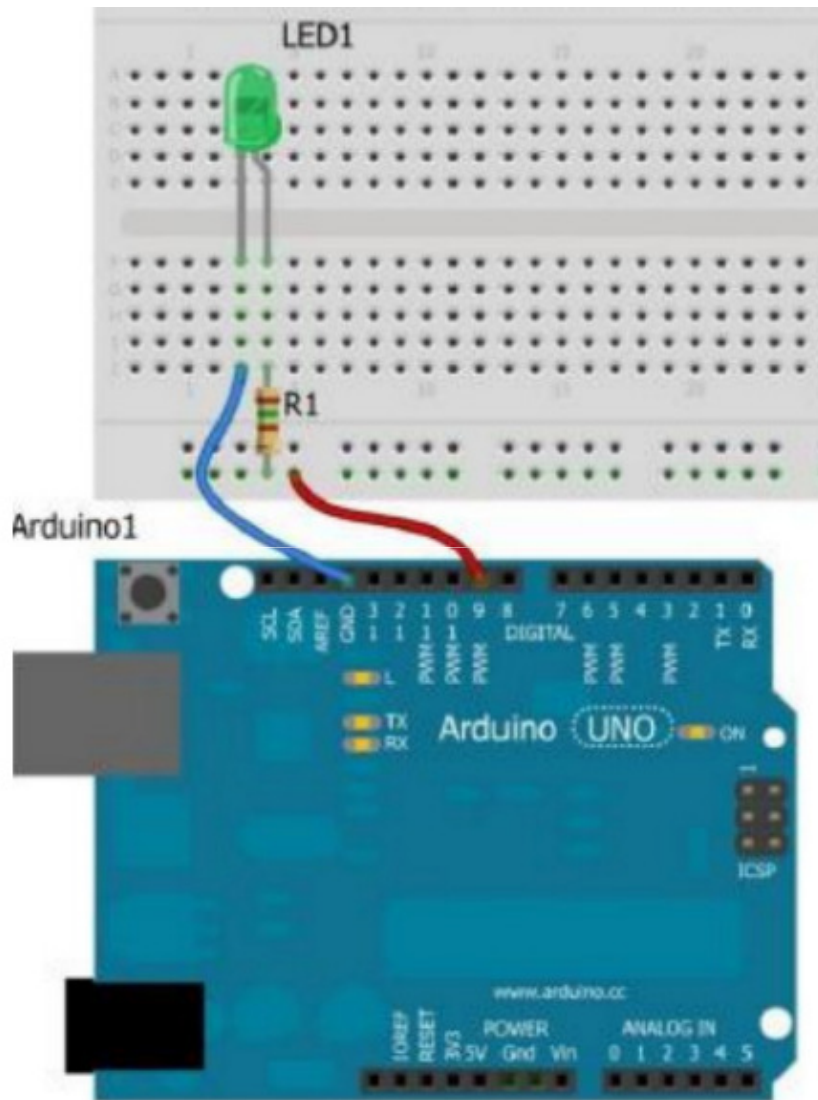
```
void setup() {  
  //configure pin2 as an input and enable the internal pull-up resistor  
  pinMode(2, INPUT_PULLUP);  
  pinMode(13, OUTPUT);  
}
```

```
void loop() {  
  //read the pushbutton value into a variable  
  int sensorVal = digitalRead(2);
```

```
  if (sensorVal == HIGH) {  
    digitalWrite(13, LOW);  
  } else {  
    digitalWrite(13, HIGH);  
  }  
}
```



SOS blikanie diód

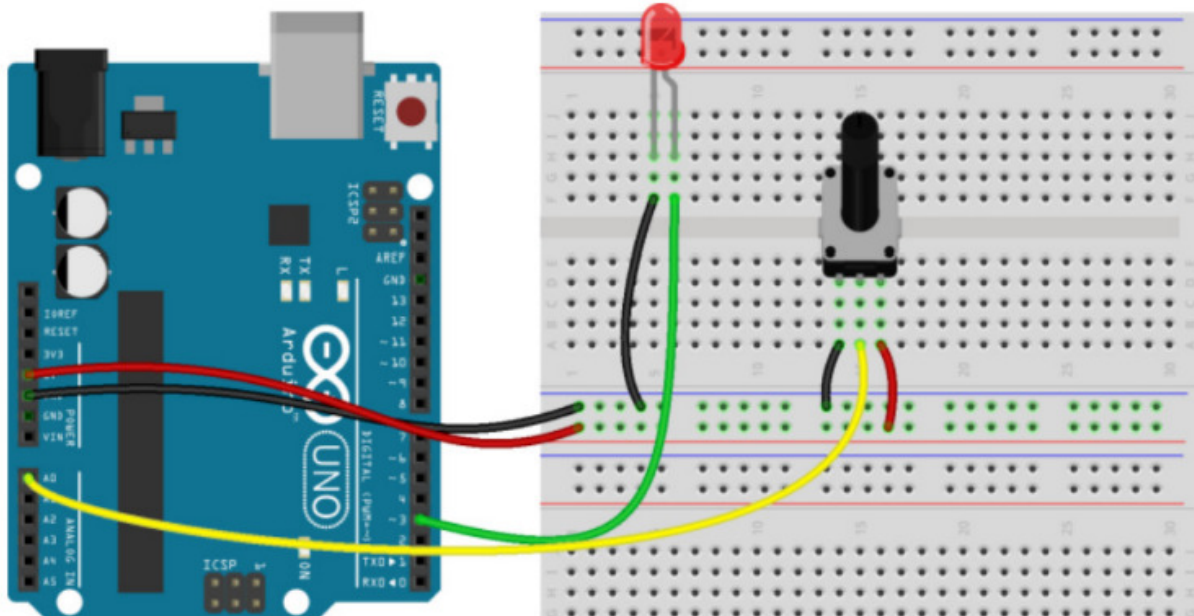


Uprav frekvencie na 100ms a 400ms

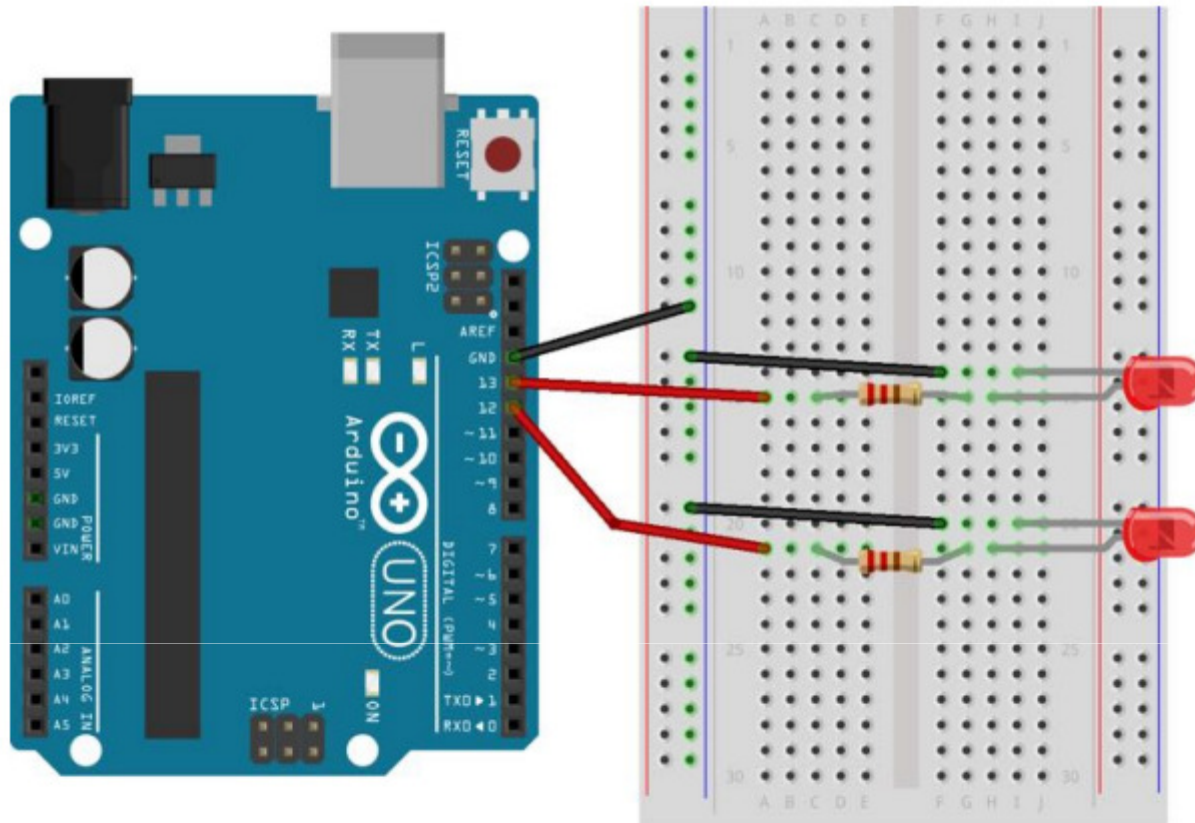
```
int ledPin = 9;
void setup() {
  // nastavíme ledPin (pin číslo 13) ako výstupný
  pinMode(ledPin, OUTPUT);
}
void loop() {
  // 3 krátke '...' v morseovej abecede písmeno 'S'
  for(int i=0; i<3; i++){
    digitalWrite(ledPin, HIGH);
    delay(120);
    digitalWrite(ledPin, LOW);
    delay(120);
  }
  delay(100); //čakáme 100 milisekúnd
  // 3 dlhé '---' v morseovej abecede písmeno 'O'
  for(int i=0; i<3; i++){
    digitalWrite(ledPin, HIGH);
    delay(350);
    digitalWrite(ledPin, LOW);
    delay(350);
  }
  delay(100);
  //zase čakáme 100 milisekúnd
  //a zase 3 krátke '...' teda 'S'
  for(int i=0; i<3; i++){
    digitalWrite(ledPin, HIGH);
    delay(120);
    digitalWrite(ledPin, LOW);
    delay(120);
  }
  delay(5000);
  //celý cyklus sa spustí znova po 5 sekundách
}
```

Blikač s potenciometrom

```
// Potenciometer
int potPin = A0; // číslo pinu pripojeného potenciometra
int ledPin = 3; // číslo pinu pripojenej LED diody
int potProm = 0; // premenná pre analógovú hodnotu potenciometra
void setup() {
  // nastavenie LED ako výstup
  pinMode(ledPin, OUTPUT);
  // nastavenie potenciometra ako vstup
  pinMode(potPin, INPUT);
}
void loop() {
  // načítanie analógovej hodnoty senzora a uloženie do premennej
  potProm = analogRead(potPin);
  // zapne LED diódu
  digitalWrite(ledPin, HIGH);
  // zastaví program na čas zodpovedajúci analógovej hodnote potenciometra
  delay(potProm);
  // vypne LED diódu
  digitalWrite(ledPin, LOW);
  // zastaví program na čas zodpovedajúci analógovej hodnote potenciometra
  delay(potProm);
}
```



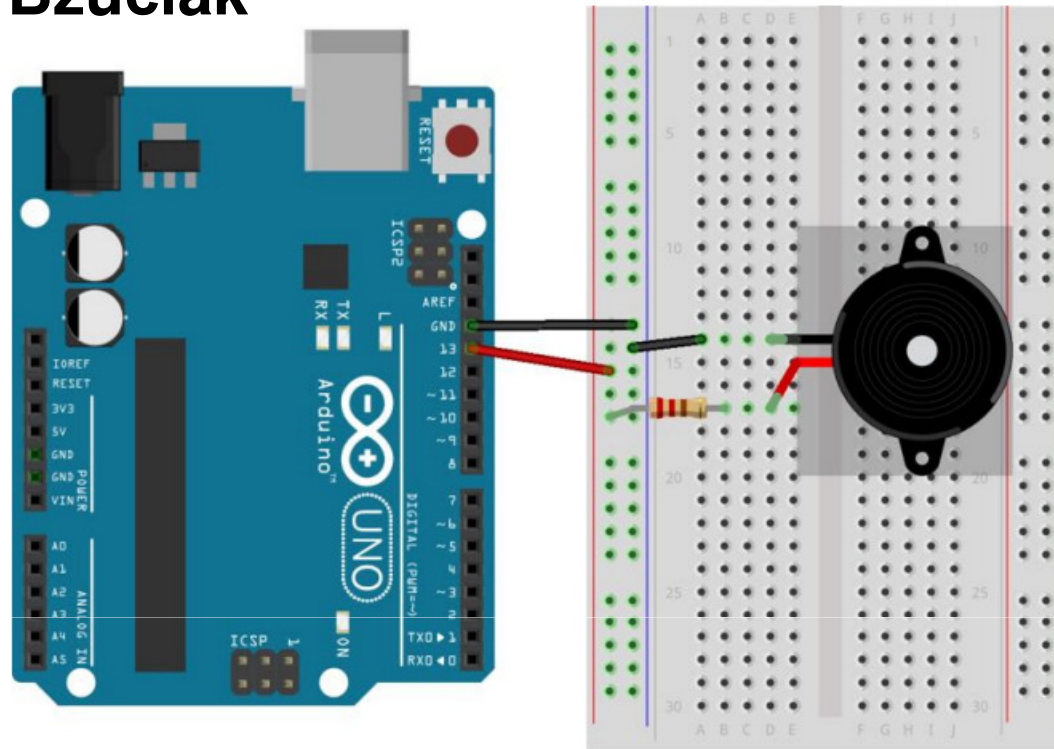
Dve led



Uprav program aby blikali súčasne

```
1 void setup() {  
2     pinMode(13, OUTPUT);  
3     pinMode(12, OUTPUT);  
4 }  
5  
6 void loop() {  
7     digitalWrite(13, HIGH);  
8     digitalWrite(12, LOW);  
9     delay(1000);  
10    digitalWrite(13, LOW);  
11    digitalWrite(12, HIGH);  
12    delay(1000);  
13 }
```

Bzučiak



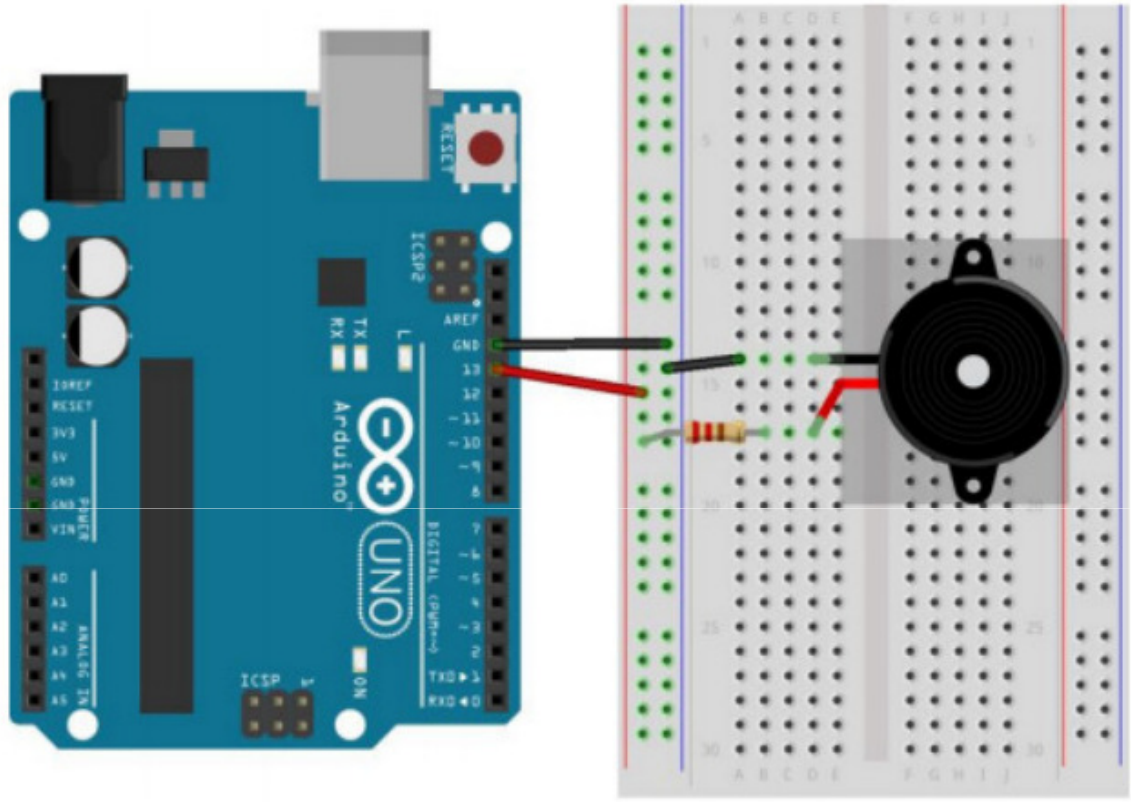
Uprav program aby bzučal inou frekvenciou.

```
const int bzucak=13; // definice proměnné bzucak

void setup() {
  pinMode(bzucak, OUTPUT); // nastavení pinu jako výstup
}

void loop() {
  tone(bzucak, 440); // funkce pro přehrání tónu
  delay(1000);
  noTone(bzucak); // funkce pro přerušení tónu
  delay(1000);
}
```

Bzučiak



Nahrad' frekvencie :

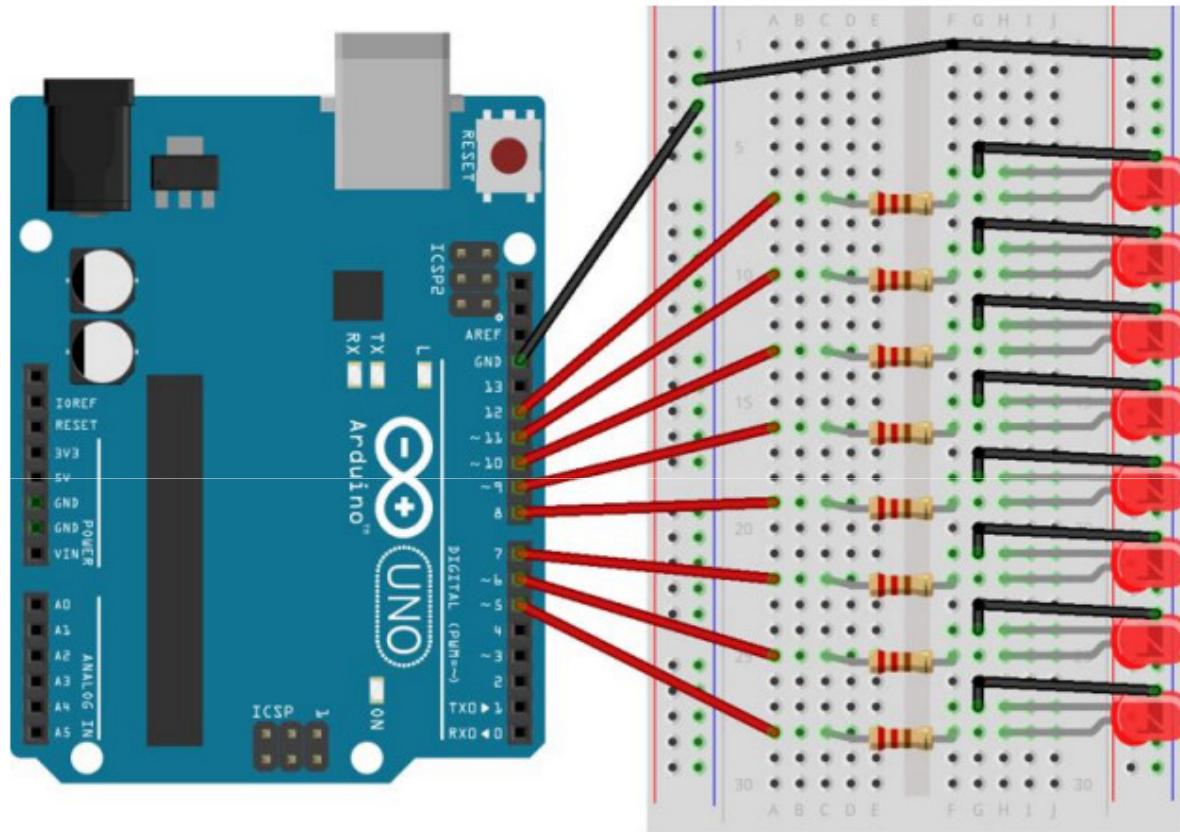
- 329 – 629
- 392 – 692
- 261 – 561
- 294 – 594

```
const int pinBzucak=13;
void setup() {
  pinMode(pinBzucak, OUTPUT);
}
void loop() {
  tone(pinBzucak, 329); delay(1000);
  noTone(pinBzucak); delay(100);
  tone(pinBzucak, 329); delay(1000);
  noTone(pinBzucak); delay(100);
  tone(pinBzucak, 329); delay(1000);
  noTone(pinBzucak); delay(100);
  tone(pinBzucak, 0); delay(1000);

  noTone(pinBzucak); delay(100);
  tone(pinBzucak, 329); delay(1000);
  noTone(pinBzucak); delay(100);
  tone(pinBzucak, 329); delay(1000);
  noTone(pinBzucak); delay(100);
  tone(pinBzucak, 329); delay(1000);
  noTone(pinBzucak); delay(100);
  tone(pinBzucak, 0); delay(1000);

  noTone(pinBzucak); delay(100);
  tone(pinBzucak, 329); delay(1000);
  noTone(pinBzucak); delay(100);
  tone(pinBzucak, 392); delay(1000);
  noTone(pinBzucak); delay(100);
  tone(pinBzucak, 261); delay(1000);
  noTone(pinBzucak); delay(100);
  tone(pinBzucak, 294); delay(1000);
  noTone(pinBzucak); delay(100);
  tone(pinBzucak, 329); delay(1000);
  tone(pinBzucak, 0); delay(1000);
}
```

Animácie

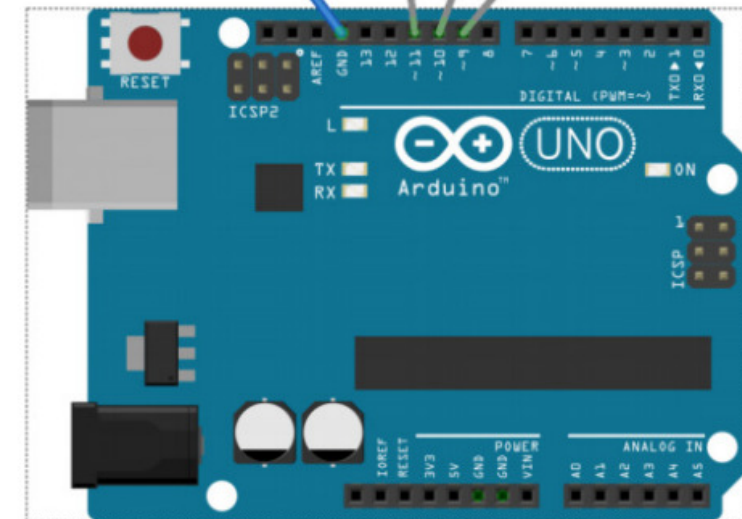
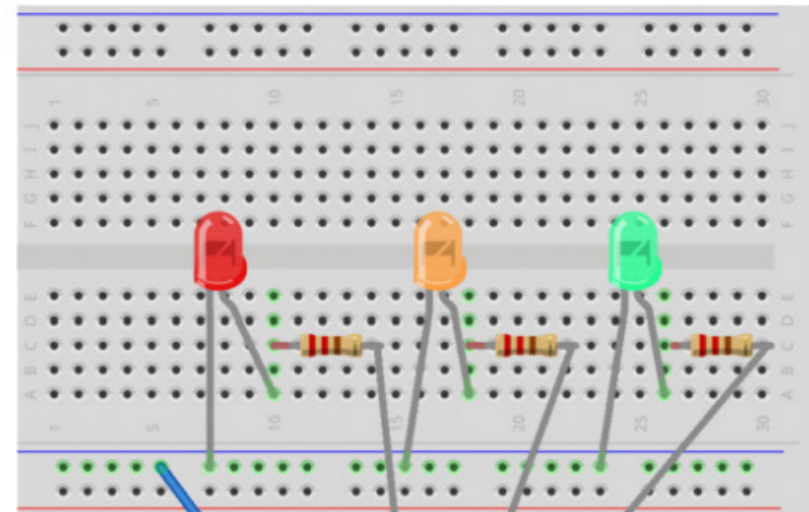


Uprav program aby led blikali v dvoch smeroch.

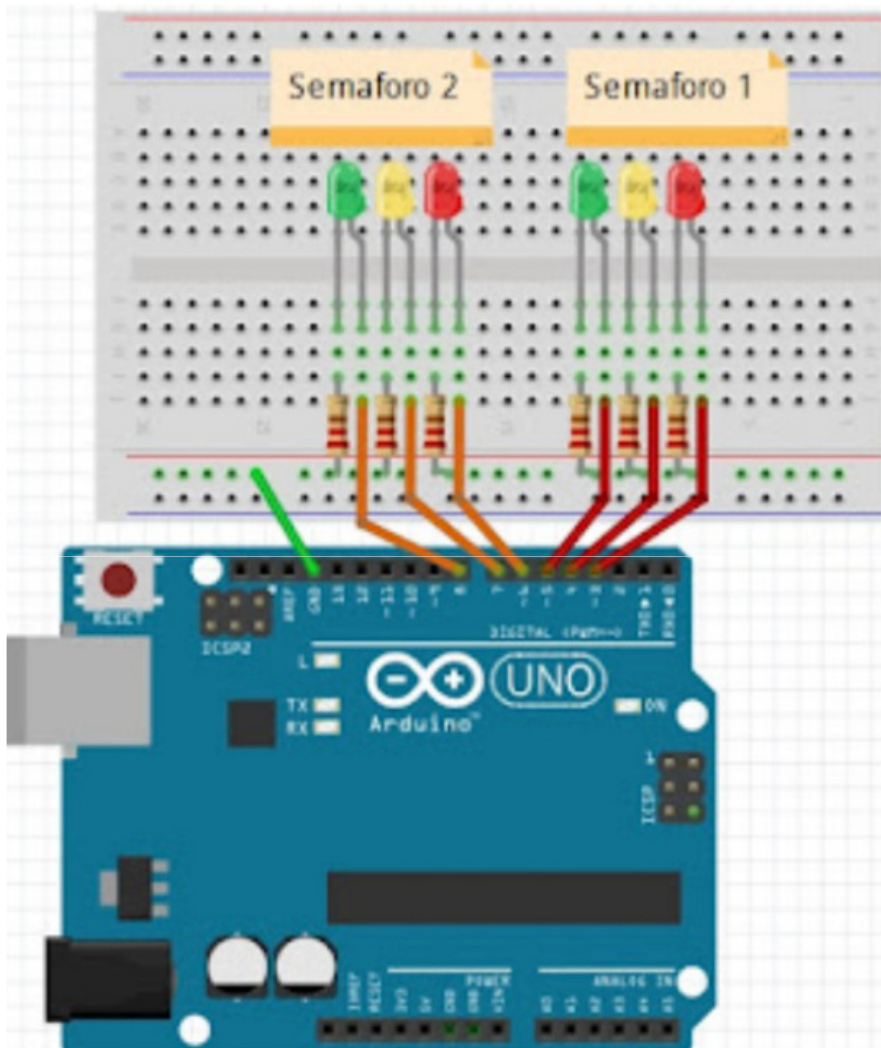
```
void setup() {  
  pinMode(5, OUTPUT);  
  pinMode(6, OUTPUT);  
  pinMode(7, OUTPUT);  
  pinMode(8, OUTPUT);  
  pinMode(9, OUTPUT);  
  pinMode(10, OUTPUT);  
  pinMode(11, OUTPUT);  
  pinMode(12, OUTPUT);  
}  
  
void loop() {  
  changeLED (5);  
  changeLED (6);  
  changeLED (7);  
  changeLED (8);  
  changeLED (9);  
  changeLED (10);  
  changeLED (11);  
  changeLED (12);  
}  
  
void changeLED(int pin) {  
  digitalWrite(pin, HIGH);  
  delay(50);  
  digitalWrite(pin, LOW);  
  delay(50);  
}
```

Semafór

```
//Nastavíme piny pripojených LED diod int cervenaled = 9;  
int cervenaled = 9;  
int oranzovaled = 10;  
int zelenaled = 11;  
void setup() {  
  //Nastavíme zapojené piny ako výstupné  
  pinMode (cervenaled, OUTPUT);  
  pinMode (oranzovaled, OUTPUT);  
  pinMode (zelenaled, OUTPUT);  
}  
void loop() {  
  //Nastavíme v akom poradí a akú dobu budú LED diódy svietiť  
  digitalWrite (cervenaled, 1);  
  delay(9000);  
  digitalWrite (oranzovaled, 1);  
  delay (1000);  
  digitalWrite (cervenaled, 0);  
  digitalWrite (oranzovaled, 0);  
  digitalWrite (zelenaled, 1);  
  delay (9000);  
  digitalWrite (zelenaled, 0);  
  digitalWrite (oranzovaled, 1);  
  delay (3000);  
  digitalWrite (oranzovaled, 0);  
}
```



Semafor



Uprav semafor 2 na piny 10, 11, 12

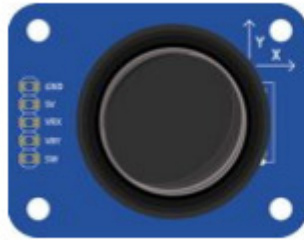
```
int leds[]={3,4,5,6,7,8}; //semafor 1 a 2 (cervena, oranzova, zelena)
int cas1=3000;
int cas2=500;
int n;
void setup() {
for (n=0;n<6;n++) {
pinMode (leds[n],OUTPUT);
}}

void sequencia(){
digitalWrite (leds[1],LOW); //oranzova1 off
digitalWrite (leds[0],HIGH); //cervena1 on
digitalWrite (leds[3],LOW); //cervena2 off
digitalWrite (leds[5],HIGH); //zelena2 on
delay (cas1);
digitalWrite (leds[5],LOW); //zelena2 off
digitalWrite (leds[4],HIGH); //oranzova2 on
delay (cas2);
digitalWrite(leds[0],LOW); //cervena1 off
digitalWrite (leds[2],HIGH); //zelena1 on
digitalWrite (leds[4],LOW); //oranzova2 off
digitalWrite (leds[3],HIGH); //cervena2 on
delay (cas1);
digitalWrite (leds[2],LOW); //zelena1 off
digitalWrite(leds[1],HIGH); //oranzova1 on
delay (cas2);
}

void loop () {
sequencia();
}
```



Fotorezistor – je součástka, která mění svůj odpor podle toho, jak intenzivní záření na ní dopadá.



Křížový ovladač, joystick – je herní modul, který umožňuje snímat pohyb ve dvou směrech. Samotný joystick je tvořen dvěma potenciometry, jejichž hodnota určuje směr.



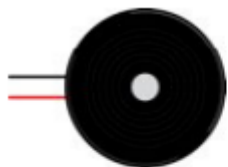
LED diody – jsou polovodičové součástky, schopné vyzařovat světlo. Používají se převážně k signalizaci. V současné době se využívají i k osvětlení.



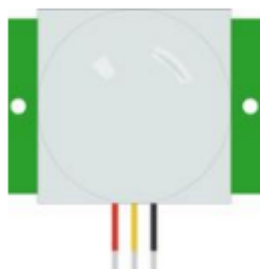
Usměrňovací diody - je elektrotechnická součástka, jejímž úkolem je v elektrickém obvodu propouštět elektrický proud jen jedním směrem.



Stejnoseměrný motor – je určený k pohybu robotických částí. Nejčastěji se používá v mobilních robotech nebo v zařízeních, kde je vyžadován plynulý rotační pohyb.



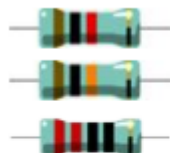
Pasivní bzučák – součástka, která vytváří zvuky na základě elektrických pulzů.



Pohybové čidlo – snímá své okolí pod úhlem přibližně 120 stupňů. Pokud v tomto snímaném prostoru proběhne větší změna teplot, tzn. například projde člověk či zvíře, je tato změna zaznamenána díky změně výstupního napětí.



Potenciometr – součástka, která při otáčení osou mění poměr odporu na její dráze. Lze je použít například při změně jasu displeje nebo LED diody.



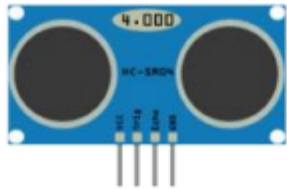
Rezistor – je pasivní elektrotechnická součástka, která je zařazována do obvodu z důvodu snížení elektrického proudu nebo získání úbytku napětí.



Tranzistor – je polovodičová součástka, kterou lze jednoduše používat jako zesilovač, spínač, stabilizátor aj.



Servomotor – slouží pro nastavení polohy ovládaného mechanismu a následné držení v této poloze. Stejnosměrné servo motory se mohou využít pro ovládání robotické paže nebo pro nastavení kormidla u leteckých modelů.



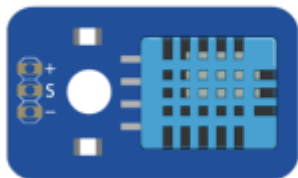
Sonar – je senzor k měření vzdálenosti. Měření je založeno na principu měření doby mezi vysláním akustického signálu a přijetím odraženého signálu.



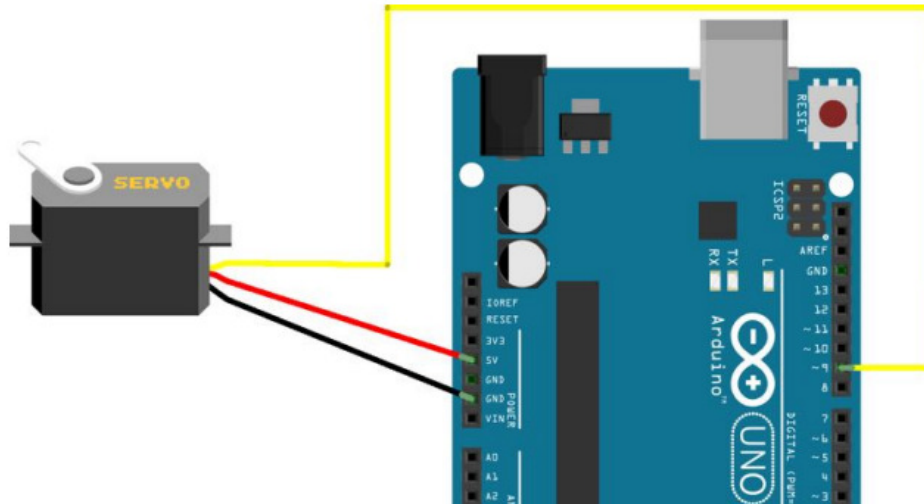
Tlačítko – je jednoduchá mechanická součástka, která slouží ke spojení a rozpojení obvodu.



Termistor – se používá pro měření teploty. Jedná se o rezistor, který v závislosti na teplotě svůj odpor mění podle přesně definované charakteristiky.



Teplotní a vlhkostní čidlo – obsahuje vše potřebné. Stačí ho připojit rovnou k desce Arduino a načítat hodnoty z okolí, které se obnovují přibližně každé 2 sekundy.

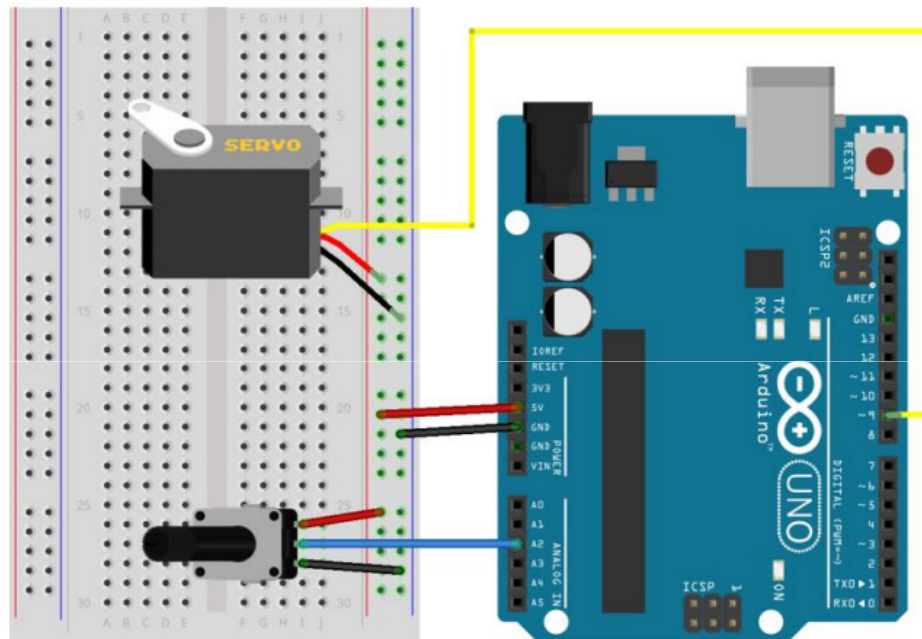


Servomotor

Uprav program aby bola pozícia 180°.

```
1  #include <Servo.h> // Připojení knihovny
2
3  Servo myservo; // Vytvoření instance třídy pro každý servomotor
4  int pos = 90; // Pozice servomotoru 90°
5
6  void setup()
7  {
8      // Definice pinu na který je připojen signální vodič
9      // servomotoru
10     myservo.attach(9);
11 }
12
13 void loop()
14 {
15     myservo.write(pos); // Nastavení pozice servomotoru
16     delay(15);
17 }
```

Servomotor



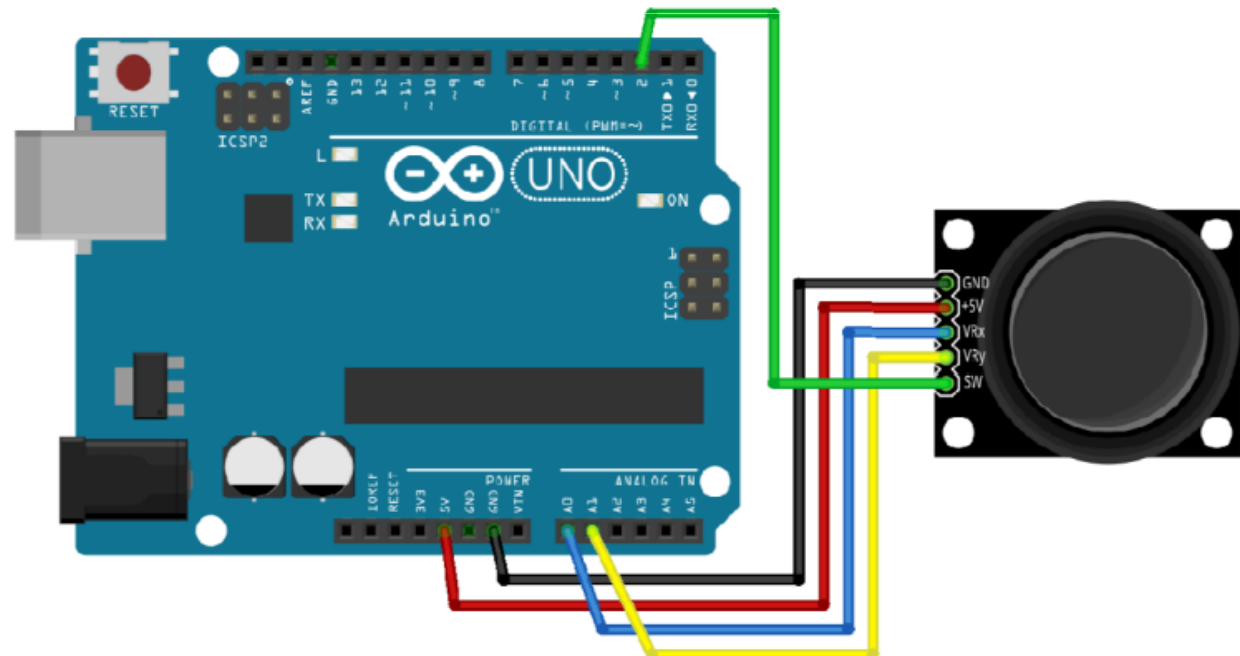
```
1  #include <Servo.h>
2
3  Servo myservo;
4  int pos = 0;
5
6  void setup()
7  {
8      myservo.attach(9);
9  }
10
11 void loop()
12 {
13     pos = analogRead(A2);
14     pos = map(pos, 0, 1023, 0, 179);
15     myservo.write(pos);
16     delay(5);
17 }
```

Uprav program aby ovládací signál potenciometra bol napájaný z pinu A1.

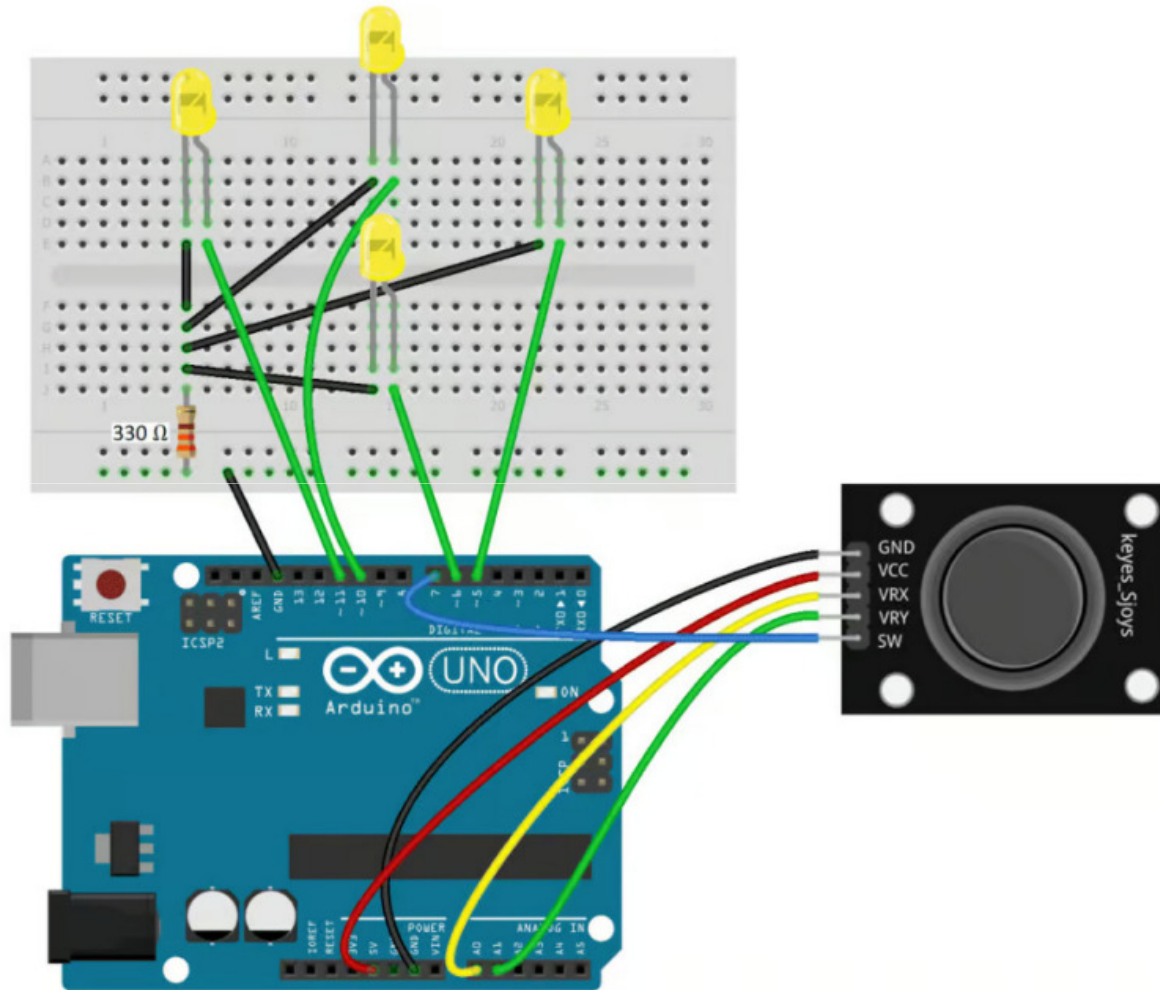
Joystick

Otestuj sériový monitor

```
const int xPin = A0; //X attach to A0
const int yPin = A1; //Y attach to A1
const int swPin = 2; //Bt attach to digital 8
void setup()
{
  pinMode(swPin,INPUT); //set bt pin as INPUT
  digitalWrite(swPin, HIGH); //and HIGH
  Serial.begin(9600); //initialize serial monitor
}
void loop()
{
  Serial.print("X: ");
  Serial.print(analogRead(xPin),DEC);
  Serial.print("|Y: ");
  Serial.print(analogRead(yPin),DEC);
  Serial.print("|Z: ");
  Serial.println(digitalRead(swPin));
  delay(500);
}
```



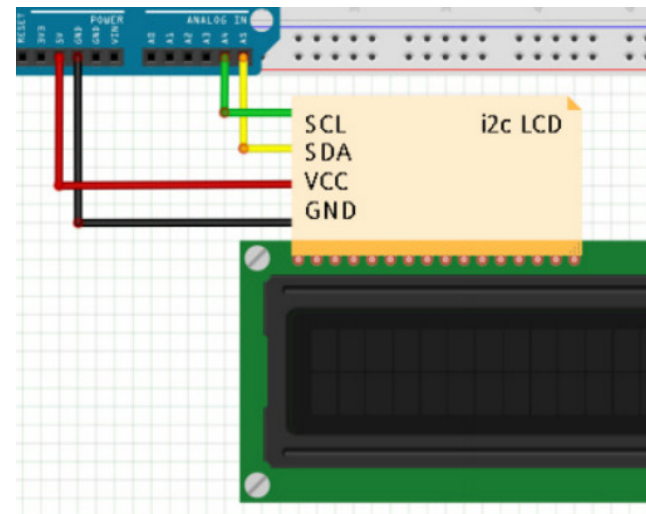
Joystick – zapojenie s LED



LCD – bežiaci text

```
/*  
LCD i2c  
bežiaci text  
*/  
#include <Wire.h> // zahrnutie kniznic do programu  
#include <LiquidCrystal_I2C.h>  
LiquidCrystal_I2C lcd(0x27,16,2); // inicializacia kniznice (adresa, znaky, riadky)  
int i=200; // premenna zdrzania pri posune textu  

```



Zmeňte premennú zdržania textu – hore a dole (50 - 400), zmeňte posúvaný text

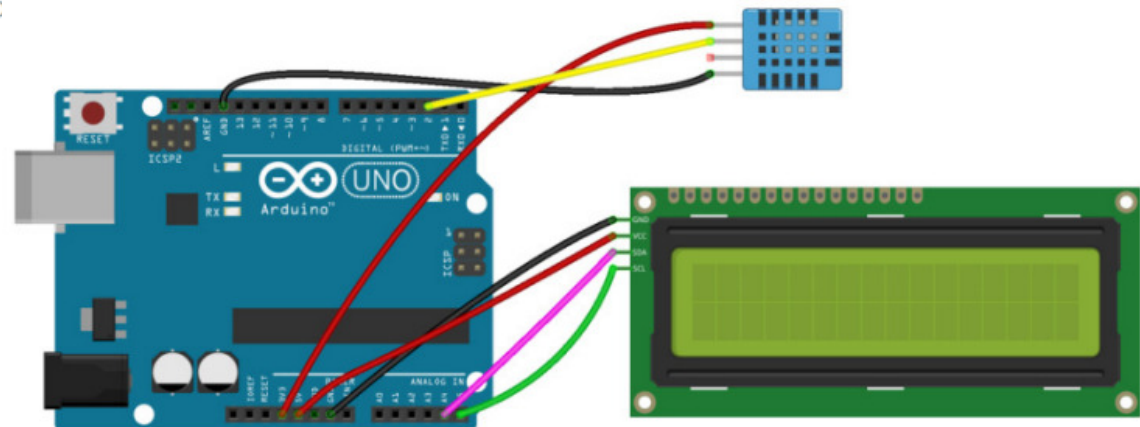
```

#include <LiquidCrystal_I2C.h>
#include "DHT.h"
LiquidCrystal_I2C lcd(0x27, 16, 2); // I2C address 0x3F, 16 column and 2 rows
DHT dht;
void setup()
{
  dht.setup(2); // data pin 2
  // dht.begin(); // initialize the sensor
  lcd.init(); // initialize the lcd
  lcd.backlight(); // open the backlight
}
void loop()
{
  delay(2000); // wait a few seconds between measurements
  float humi = dht.getHumidity(); // read humidity
  float tempC = dht.getTemperature(); // read temperature
  lcd.clear();
  // check if any reads failed
  if (isnan(humi) || isnan(tempC)) {
    lcd.setCursor(0, 0);
    lcd.print("Failed");
  } else {
    lcd.setCursor(0, 0); // start to print at the first row
    lcd.print("Temp: ");
    lcd.print(tempC); // print the temperature
    lcd.print((char)223); // print ° character
    lcd.print("C");
    lcd.setCursor(0, 1); // start to print at the second row
    lcd.print("Humi: ");
    lcd.print(humi); // print the humidity
    lcd.print("%");
  }
}

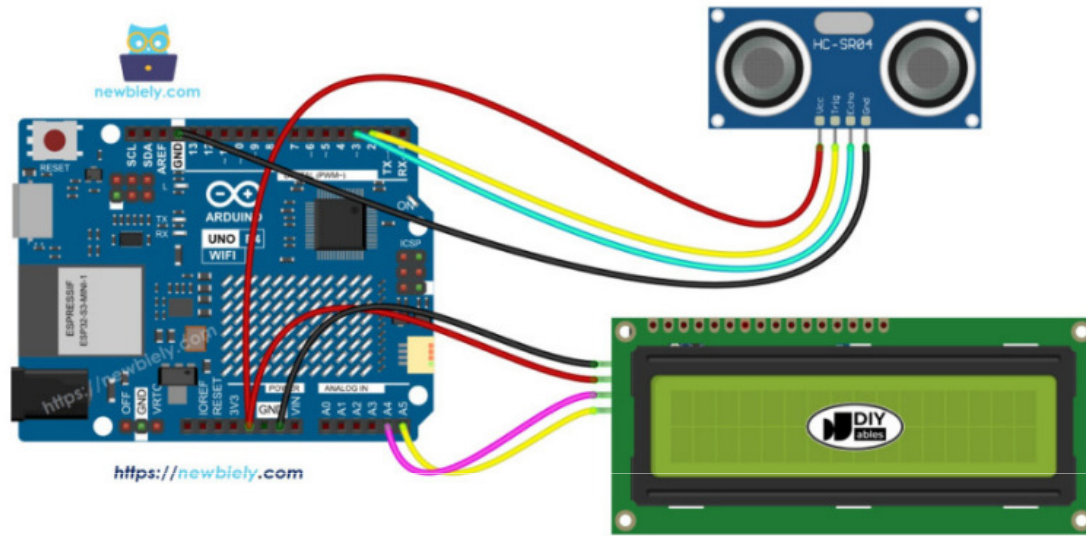
```

LCD displej so snímačom teploty a vlhkosti

Vymeňte riadky s hodnotami vlhkosť a teplota, uprav texty na teplota a vlhkosť



LCD displej so snímačom vzdialenosti

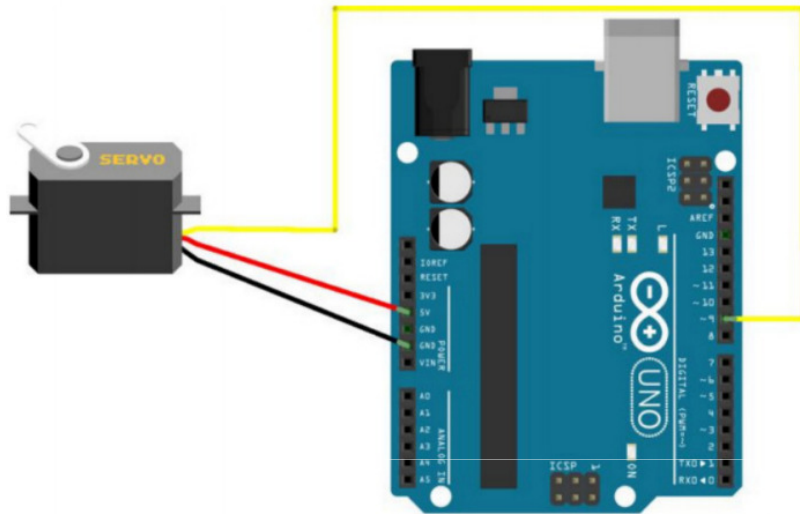


`distance_cm = 0.017 * duration_us;`

Vzdialenosť = rýchlosť zvuku * čas / 2

Zapojte pomocou kontaktnej plochy

Zapojenie servomotora – výchylka 180°

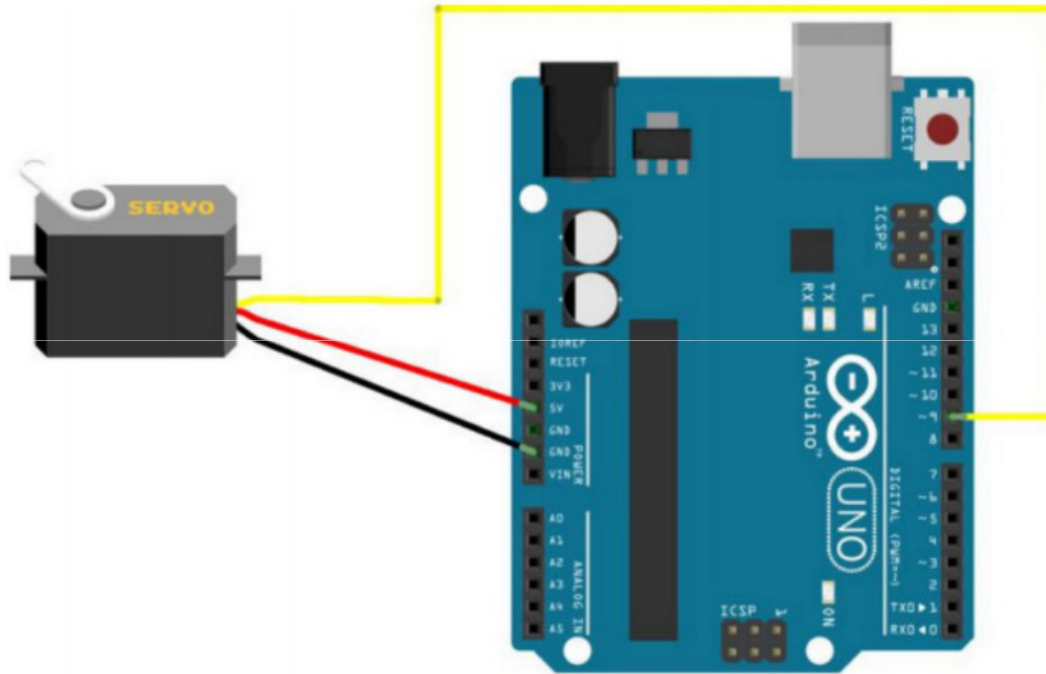


```
#include <Servo.h>
Servo myservo;
int pos = 0;
void setup()
{
  myservo.attach(9);
}
void loop()
{
  // Úprava mezní hodnoty natočení servomotoru na 180
  for(pos = 0; pos <= 180; pos += 1)
  {
    myservo.write(pos);
    // Úprava prodlevy, aby se servo ještě více zpomalilo
    delay(20);
  }
  // Úprava mezní hodnoty natočení servomotoru ze 180 na 0
  for(pos = 180; pos >= 0; pos -= 1)
  {
    myservo.write(pos);
    // Úprava prodlevy, aby se servo zrychlilo
    delay(5);
  }
}
```

Uprav program na výchylku 90°, otestuj

Zapojenie servomotora – výchylka 40°, 80°,120°, 160°

```
#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(160);
delay(1000);
}
```

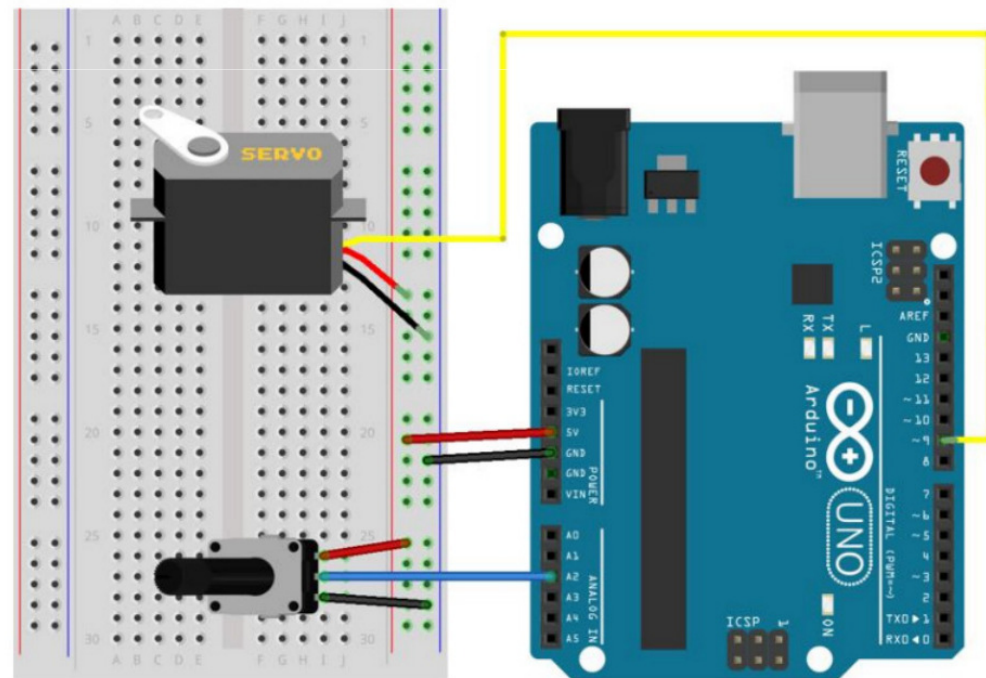


Zmeň výchylky na 45°, 90°, 135°, 180°

```
#include <Servo.h>
Servo myservo;
int pos = 0;
void setup()
{
  myservo.attach(9);
}
void loop()
{
  pos = analogRead(A2);
  pos = map(pos, 0, 1023, 0, 179);
  myservo.write(pos);
  delay(5);
}
```

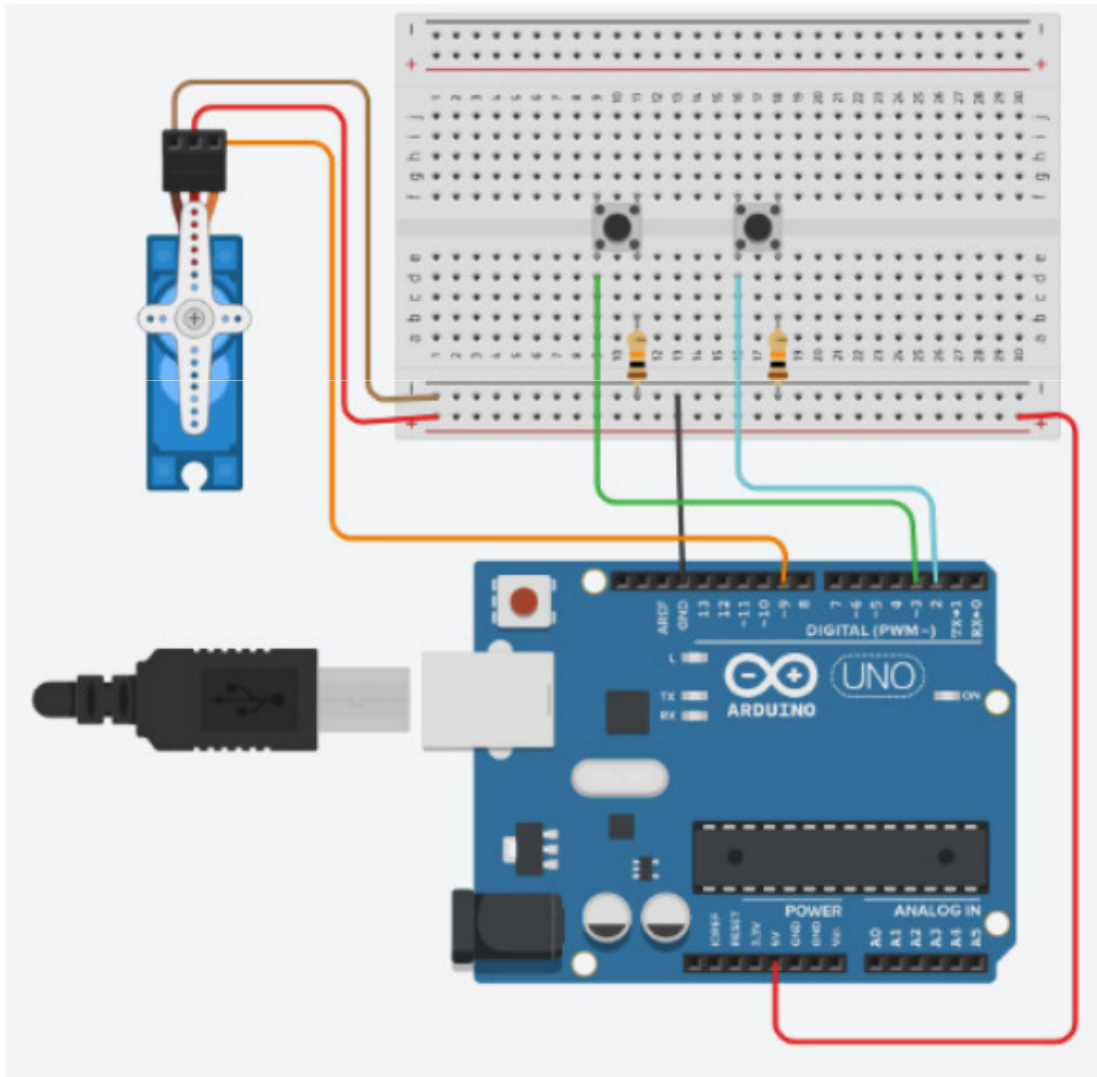
Servomotor s potenciometrom

Uprav výchylku na 90°



Servomotor s dvoma tlačidlami

Otestuj funkčnosť



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

```
Servo myservo;
```

```
int val;  
int ChE = 3 ; // pin to be connected in key left  
int ChC = 4 ; // pin to be connected in the central key  
int ChD = 5 ; // pin to be connected in the right key
```

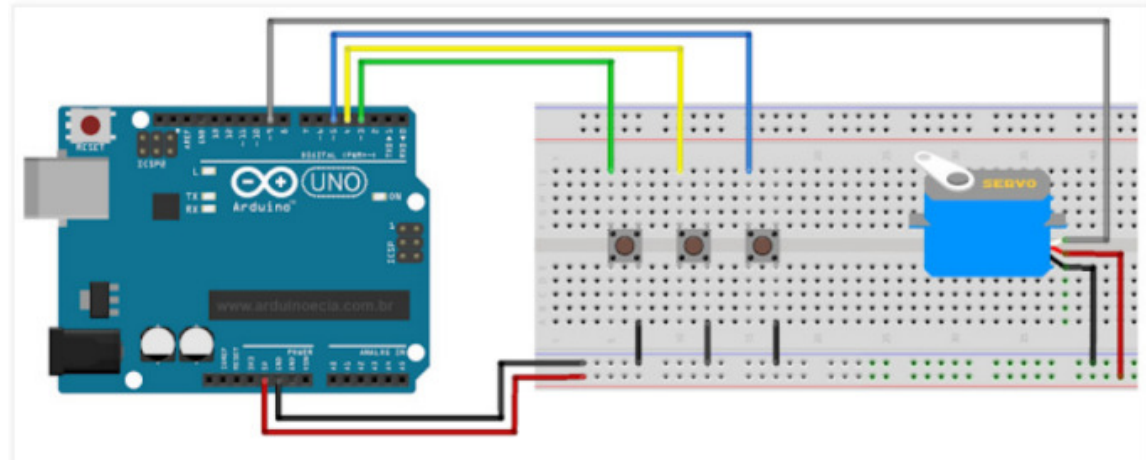
```
void setup ()
```

```
{  
  // Servo data pin connected to pin 9 of the Arduino  
  myservo.attach ( 9 );  
  // Set the pin as input  
  pinMode (ChE, INPUT);  
  // Triggers the internal pull-up resistor  
  digitalWrite (ChE, HIGH);  
  pinMode (ChC, INPUT);  
  digitalWrite (ChC, HIGH);  
  pinMode (ChD, INPUT);  
  digitalWrite (ChD, HIGH);  
}
```

```
void loop ()
```

```
{  
  // Le value of the Left Key (On / Off)  
  val = digitalRead (ChE);  
  // If the key is pressed, the servo moves  
  if (val!= 1 )  
  {  
    myservo.write ( 60 ); // Move the servo to the angle of 60 degrees  
    delay ( 15 ); // Delay for the servant reach the position  
  }  
  
  val = digitalRead (ChC);  
  if (val != 1 )  
  {  
    myservo.write ( 90 ); // Move the servo to the angle of 90 degrees  
    delay ( 15 );  
  }  
  
  val = digitalRead (ChD);  
  if (val != 1 )  
  {  
    myservo.write ( 120 ); // Move the servo to the angle of 120 degrees  
    delay ( 15 );  
  }  
}
```

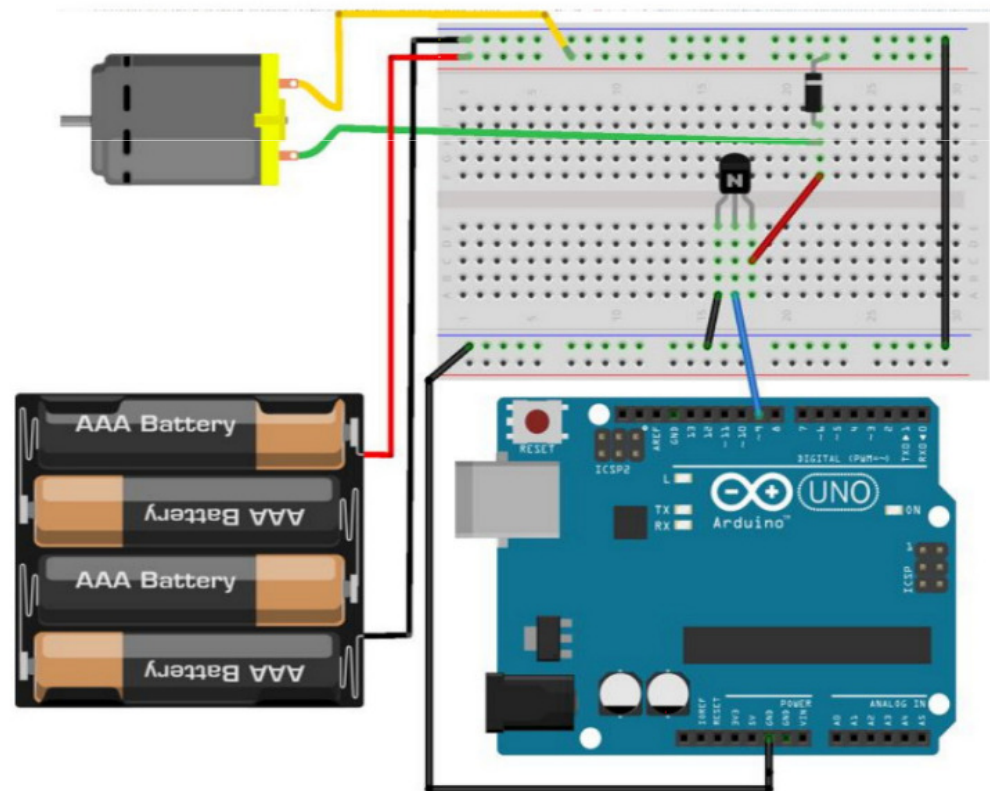
Servomotor s troma tlačidami



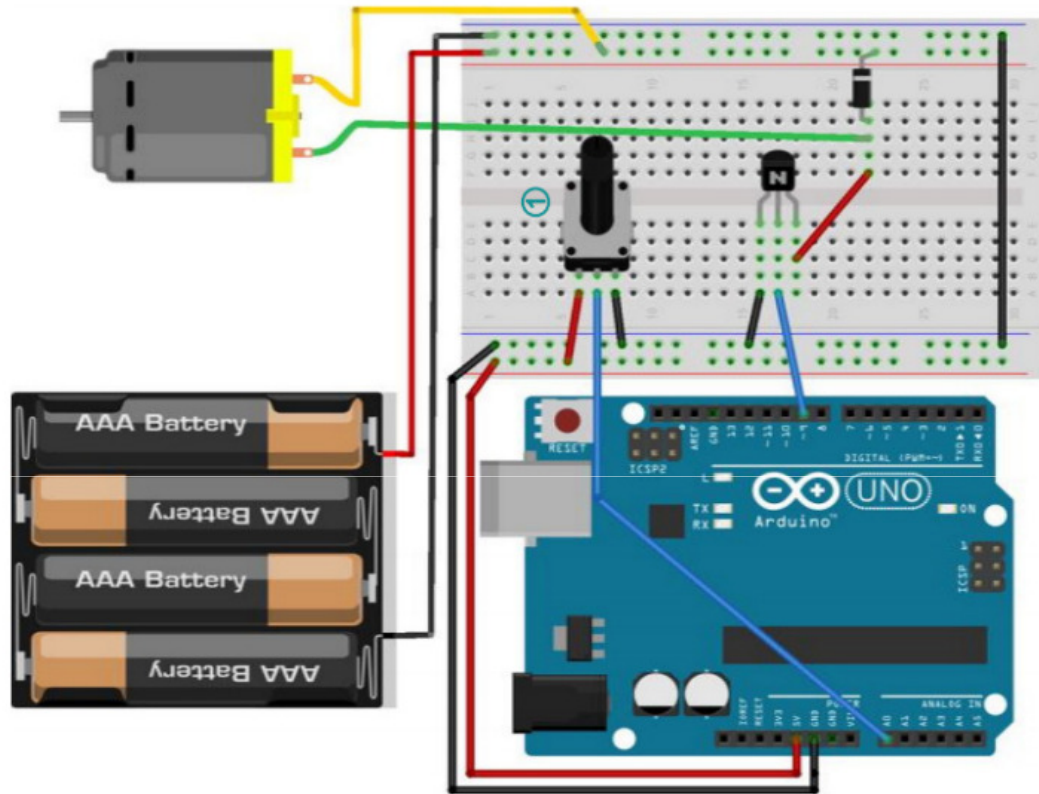
Uprav program na výchylky 45° , 90° , 135°

Jednosmerný motorček

```
const int transistorPin = 9;  
const int speedMotor = 200;  
void setup() {  
  pinMode(transistorPin, OUTPUT);  
}  
void loop() {  
  analogWrite(transistorPin, speedMotor);  
}
```



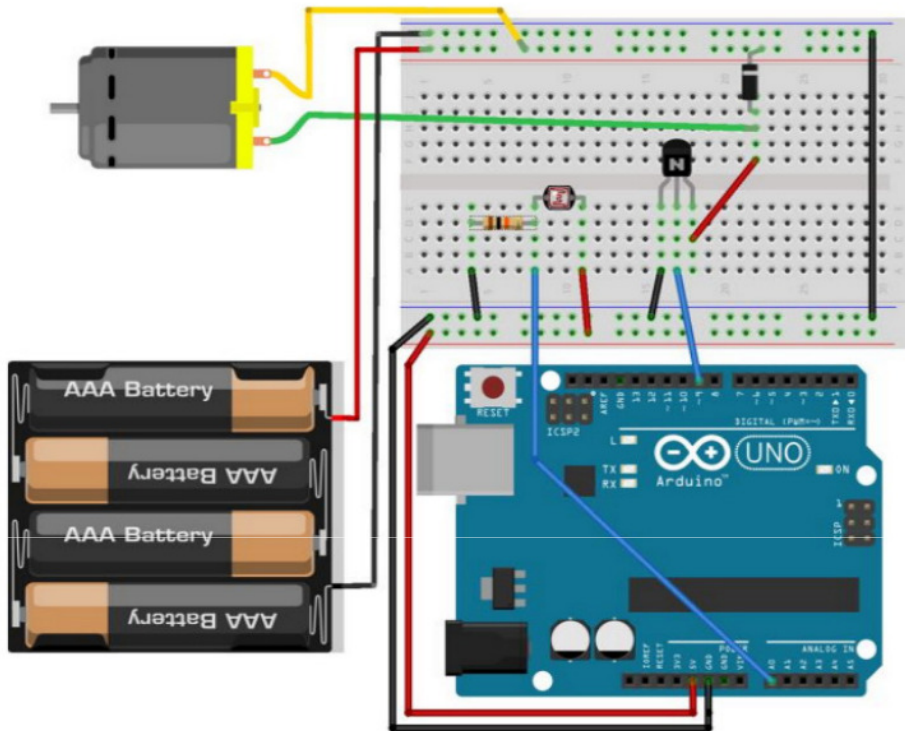
Regulácia otáčok potenciometrom



```
const int transistorPin = 9;
void setup() {
  pinMode(transistorPin, OUTPUT);
}
void loop() {
  int sensorValue = analogRead(A0);
  int outputValue = map(sensorValue, 0, 1023, 0, 255);
  analogWrite(transistorPin, outputValue);
}
```

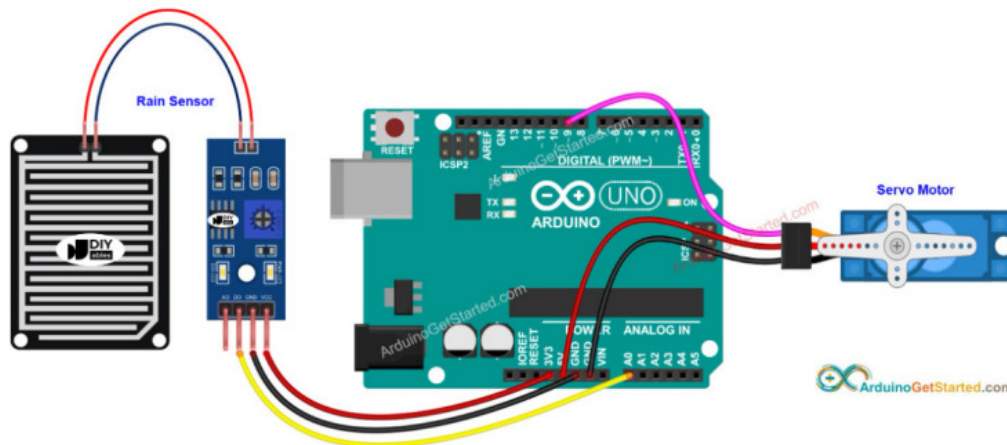
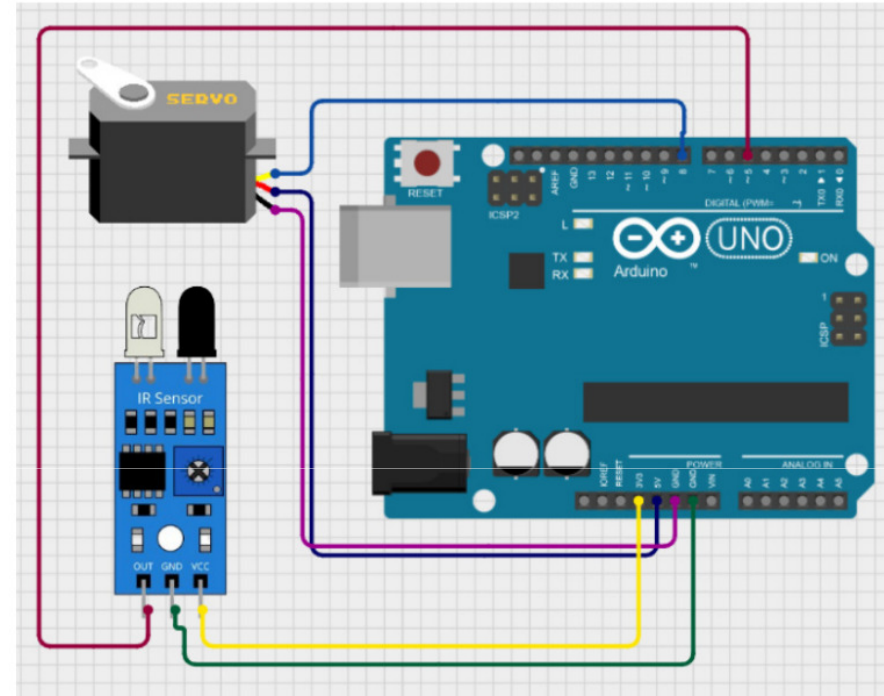
Motor so snímačmi

Regulácia otáčok fotorezistorom



Čidlo otáčok so snímačom prekážky

Uprav výchylku na 180°



Servomotor so snímačom dažďa

Uprav výchylku na 180°

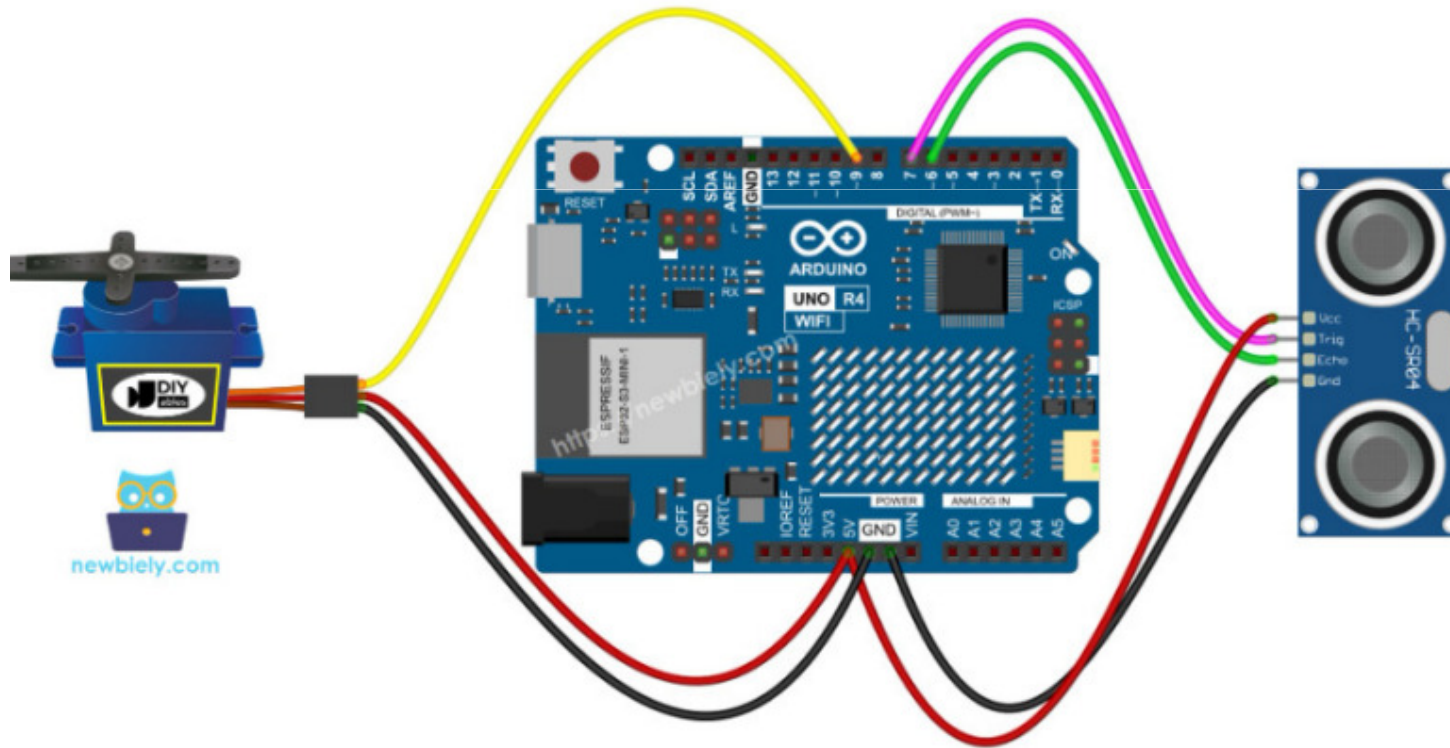
Servomotor so snímačom vzdialenosti

Otestuj zapojenie s prahovými hodnotami

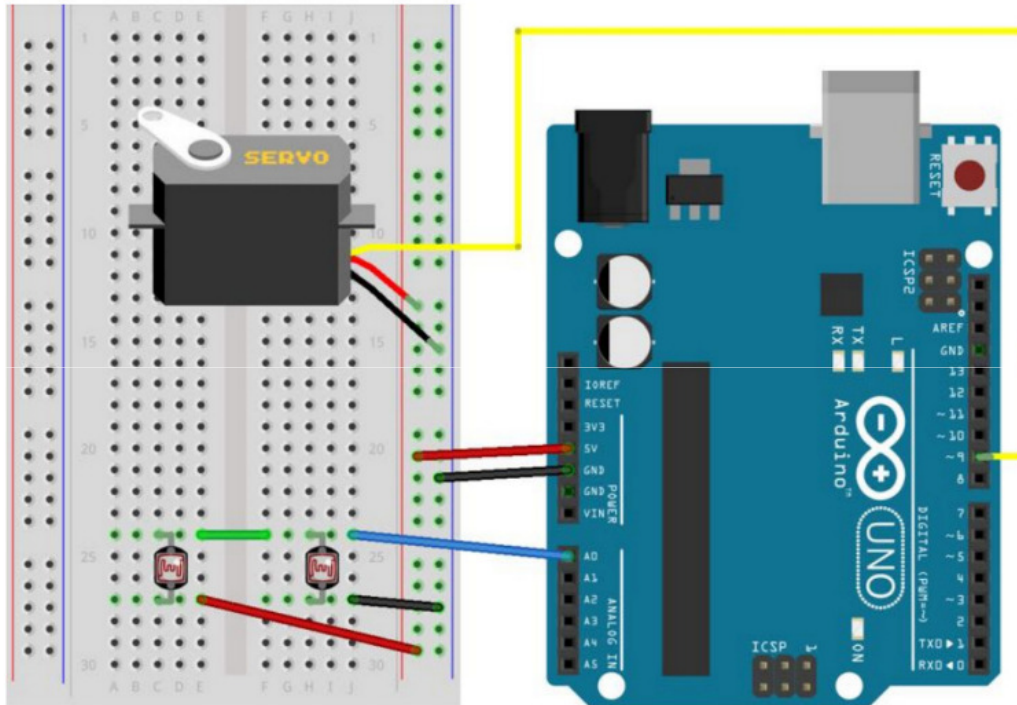
20 cm – výchylka 45°

30 cm – výchylka 90°

40 cm – výchylka 135°



Fotorezistor - servomotor



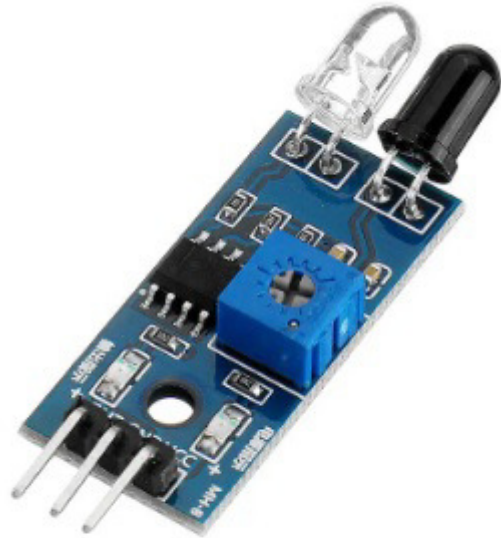
Otestuj zapojenie

```
#include <Servo.h>
int sensorPin = A0;
int servoPin = 9;
int sensorValue = 0;
int servoPos = 90;
Servo myservo;
void setup() {
  pinMode(sensorPin, INPUT);
  myservo.attach( servoPin );
}
void loop() {
  sensorValue = analogRead(sensorPin);
  if (sensorValue < (512) )
  {
    if (servoPos < 180)
    {
      servoPos++;
    }
  }
  if (sensorValue > (512) )
  {
    if (servoPos > 0)
    {
      servoPos--;
    }
  }
  myservo.write(servoPos);
  delay(100);
}
```

Ciel': Testovanie snímačov pre platformu Arduino:

- Čidlo prekážky
- Detektor dymu a alkoholu
- Snímač teploty a vlhkosti
- Pohybové čidlo
- Čidlo dažďa
- Snímač magnetického poľa
- Snímač ohňa
- Termistor
- Ultrazvukový senzor
- Projekt pre skleník

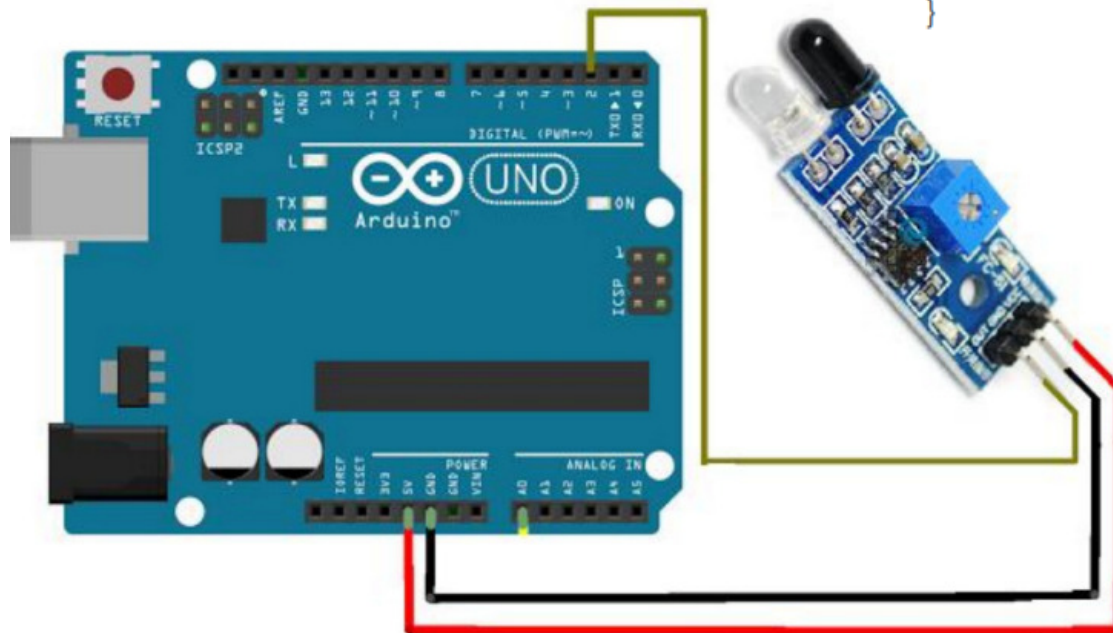
Čidlo prekážky



- Modul obsahuje 1 pár infračervených diód (1 vysielač a 1 prijímač) pomocou ktorých je možné merať vzdialenosť od prekážky od 2 cm – 30 cm.
- Na module je možné naladiť aj citlivosť.
- Senzor je možné použiť pri správnom zapojení aj pre robota sledujúceho čiernu čiaru na bielom podklade (line follower).
- Pracovné napätie senzora je 3.3V – 5V pre čip LM393.
- Modul má zabudované indikátorové LED-ky pre výstup a pre napájanie.

Čidlo prekážky

Doplň zapojenie led diódy a otestuj funkčnosť



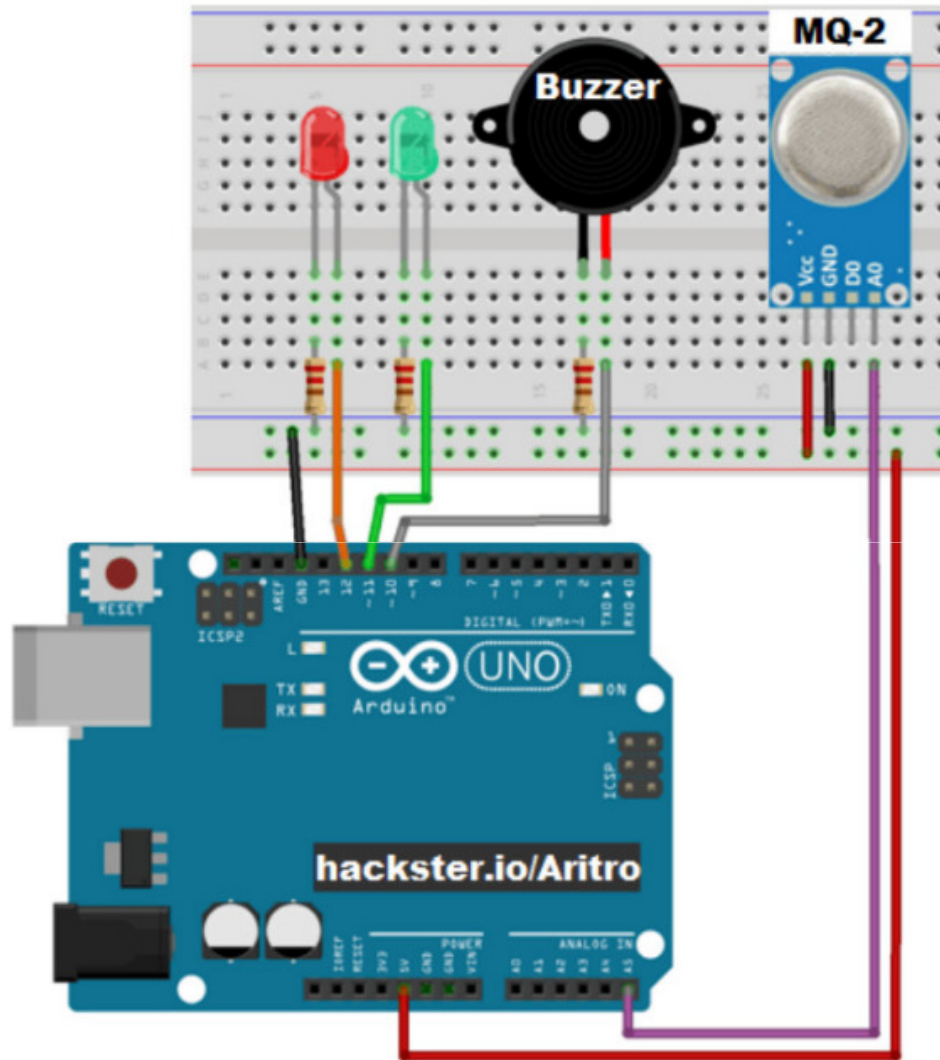
```
const int ProxSensor=2;
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
  //Pin 2 is connected to the output of proximity sensor
  pinMode(ProxSensor,INPUT);
}
void loop() {
  if(digitalRead(ProxSensor)==HIGH) //Check the sensor output
  {
    digitalWrite(13, HIGH); // set the LED on
  }
  else
  {
    digitalWrite(13, LOW); // set the LED off
  }
  delay(100); // wait for a second
}
```

Detektor dymu a alkoholu



- Sensory rôznych plynov zo série MQ pomocou ktorých dokážeme snímať od LPG cez uhoľné plyny až po etanol.
- Sensory snímajú nie len prítomnosť jednotlivých plynov, ale aj ich koncentráciu.
- Na každom type sa nachádza potenciometer a indikačná LED-ka.
- Pracovné napätie: 5V.
- Pre niektoré typy sa používa alternujúce napätie (približne od 1.3V – 5V). Pri iných napätiach sa potom dajú testovať ďalšie plyny.
- Digitálny aj analógový výstup

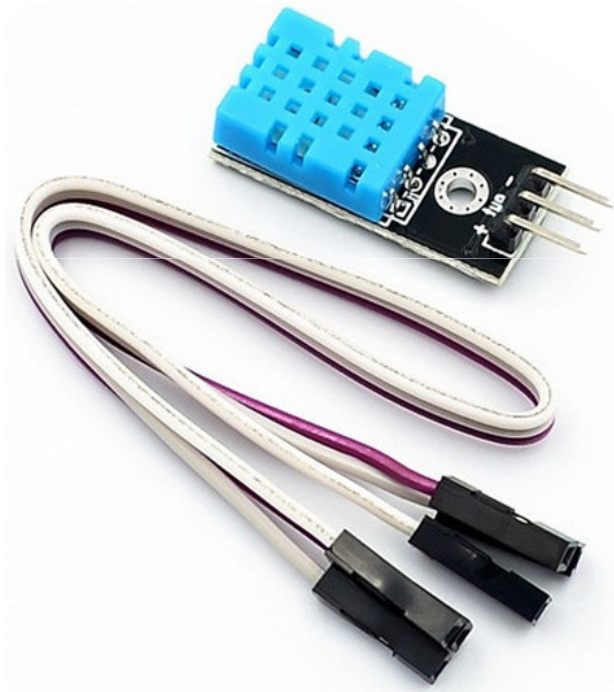
Detektor dymu a alkoholu



- Otestuj zapojenie
- Zmenši citlivosť

```
int redLed = 12;
int greenLed = 11;
int buzzer = 10;
int smokeA0 = A5;
// Vaša prahová hodnota
int sensorThres = 400;
void setup() {
  pinMode(redLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
  pinMode(buzzer, OUTPUT);
  pinMode(smokeA0, INPUT);
  Serial.begin(9600);
}
void loop() {
  int analogSensor = analogRead(smokeA0);
  Serial.print("Pin A0: ");
  Serial.println(analogSensor);
  // Kontrola či sa dosiahne prahová hodnota
  if (analogSensor > sensorThres)
  {
    digitalWrite(redLed, HIGH);
    digitalWrite(greenLed, LOW);
    tone(buzzer, 1000, 200);
  }
  else
  {
    digitalWrite(redLed, LOW);
    digitalWrite(greenLed, HIGH);
    noTone(buzzer);
  }
  delay(100);
}
```

Snímač vlhkosti a teploty

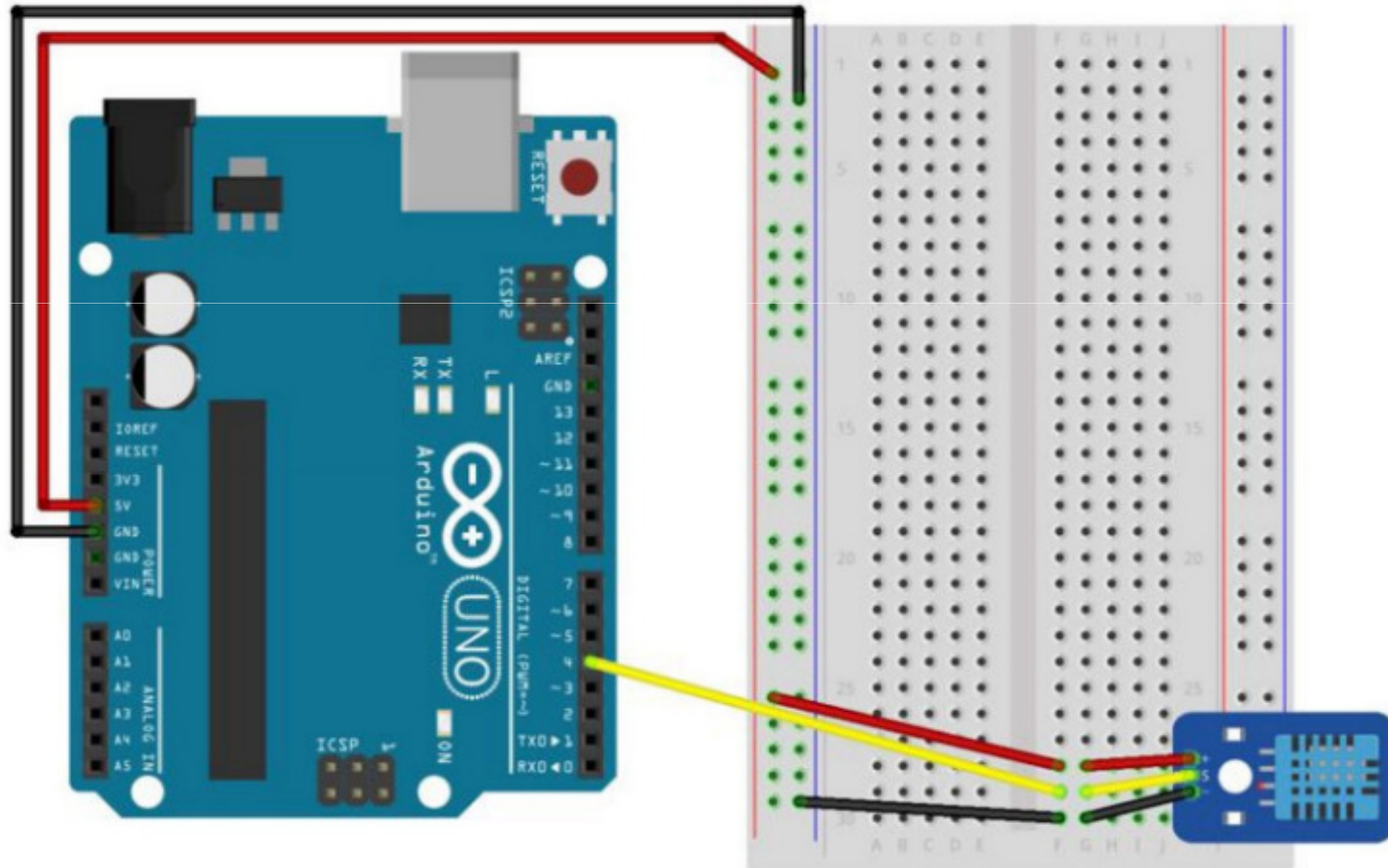


- Jednoduchý digitálny senzor na snímanie vlhkosti a teploty okolitého prostredia.
- Dá sa využiť na budovanie vlastnej meteostanice.
- Vhodný pre začiatočníkov. Pre získanie presnejšej teploty a vlhkosti s väčším rozsahom teplôt odporúčame zakúpiť DHT22.

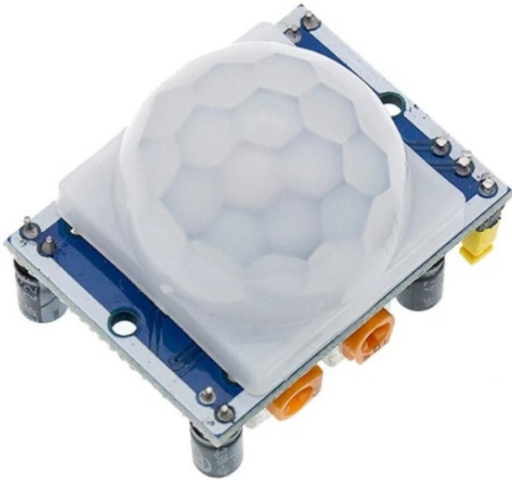
```
#include <DHT11.h>
DHT11 dht11(4);
int temperature = 0;
int humidity = 0;
void setup(){
  Serial.begin(9600);
}
void loop()
{
  int result = dht11.readTemperatureHumidity(temperature, humidity);
  if (result == 0) {
    Serial.print("Temperature: ");
    Serial.print(temperature);
    Serial.print(" °C\tHumidity: ");
    Serial.print(humidity);
    Serial.println(" %");
  } else {
    // Print error message based on the error code.
    Serial.println(DHT11::getErrorString(result));
  }
  delay(5000);
}
```

Snímač vlhkosti a teploty

- Otestuj zapojenie
- Prepíš do slovenčiny
- Skontroluj sériový monitor



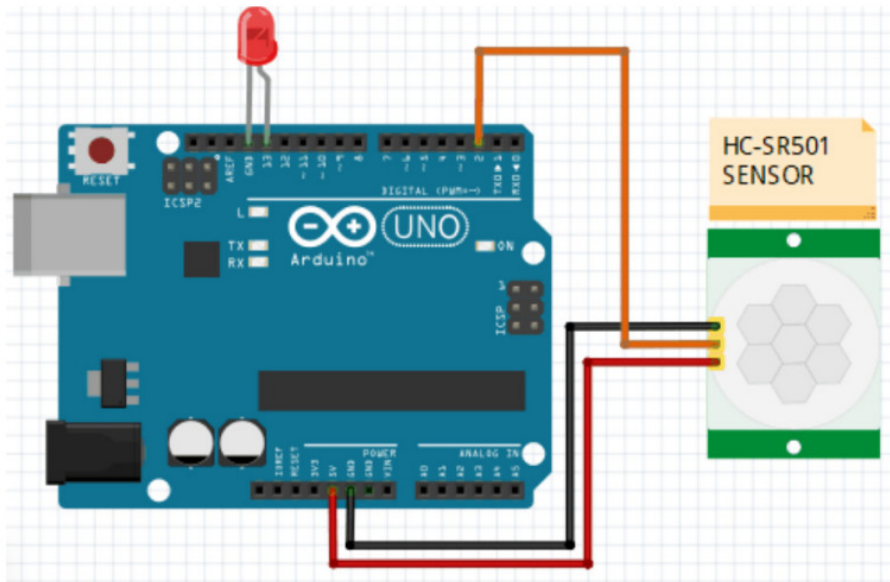
Pohybové čidlo



Skontroluj výpisy v sériovom monitore pri detekcii pohybu

- Sníma okolie pod uhlom približne 120 stupňov.
- Pokiaľ v tomto snímanom priestore prebehne väčšia zmena teplôt, tzn. napríklad prejde človek alebo zviera, je táto zmena zaznamenaná vďaka zmene výstupného napätia.
- Infračervený senzor pohybu.
- Dosah približne 7 metrov

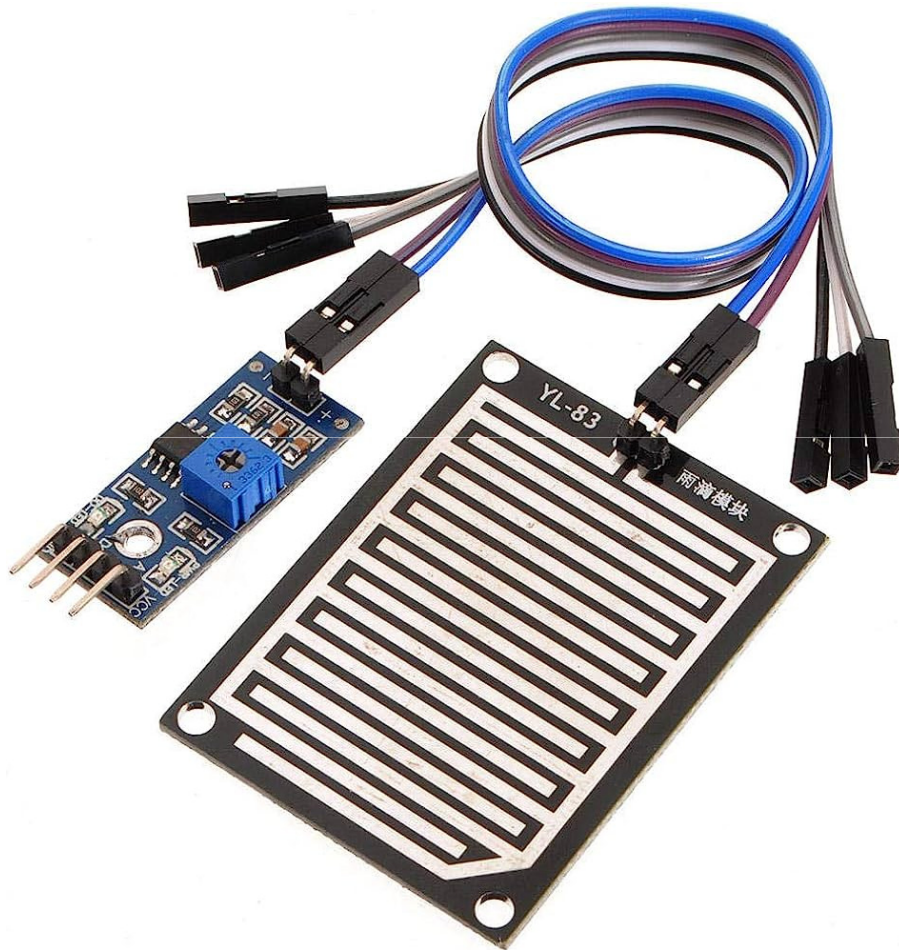
Pohybové čidlo



Skontroluj výpisy v sériovom monitore pri detekcii pohybu

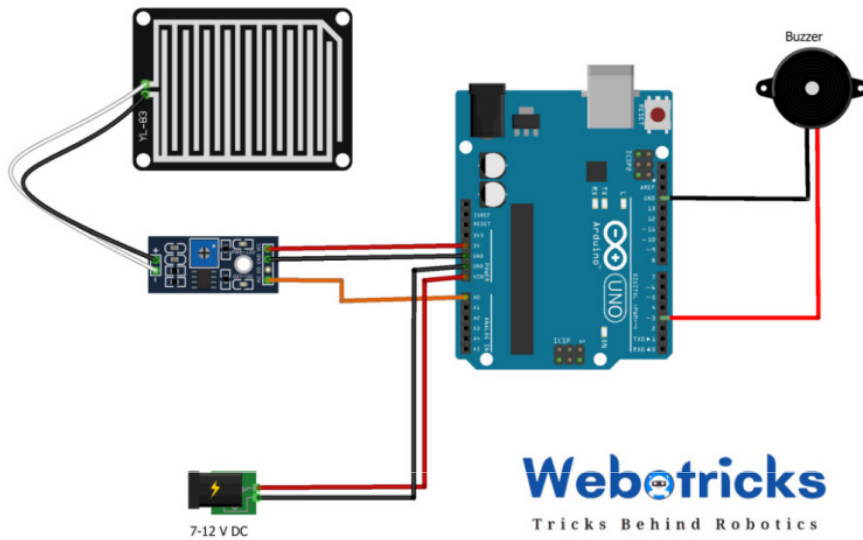
```
int led = 13;           // the pin that the LED is attached to
int sensor = 2;        // the pin that the sensor is attached to
int state = LOW;       // by default, no motion detected
int val = 0;           // variable to store the sensor status (value)
void setup() {
  pinMode(led, OUTPUT); // initialize LED as an output
  pinMode(sensor, INPUT); // initialize sensor as an input
  Serial.begin(9600);   // initialize serial
}
void loop(){
  val = digitalRead(sensor); // read sensor value
  if (val == HIGH) {        // check if the sensor is HIGH
    digitalWrite(led, HIGH); // turn LED ON
    delay(100);             // delay 100 milliseconds
    if (state == LOW) {
      Serial.println("Detekcia pohybu!");
      state = HIGH;        // update variable state to HIGH
    }
  }
  else {
    digitalWrite(led, LOW); // turn LED OFF
    delay(200);             // delay 200 milliseconds
    if (state == HIGH){
      Serial.println("Pohyb sa zastavil!");
      state = LOW;         // update variable state to LOW
    }
  }
}
```

Senzor dažďa



- Senzor dažďa dokáže detekovať zrážky za hranicou senzorov vlhkosti.
- Modul je vlastne variabilný resistor ktorý mení hodnoty od 100K ohm (keď je mokrý) až po 2M Ohm (keď je suchý).
- V skratke, čím viac vody na senzore, tým väčší prúd bude detekovaný.
- Na module je LED-ka indikujúca pripojenie na VCC aj výstup a potenciometer, ktorým je možné ladiť citlivosť.
- Je veľmi jednoducho programovateľný a je vhodný aj pre začiatočníkov.

Senzor dažďa



Otestuj prahovú hodnotu

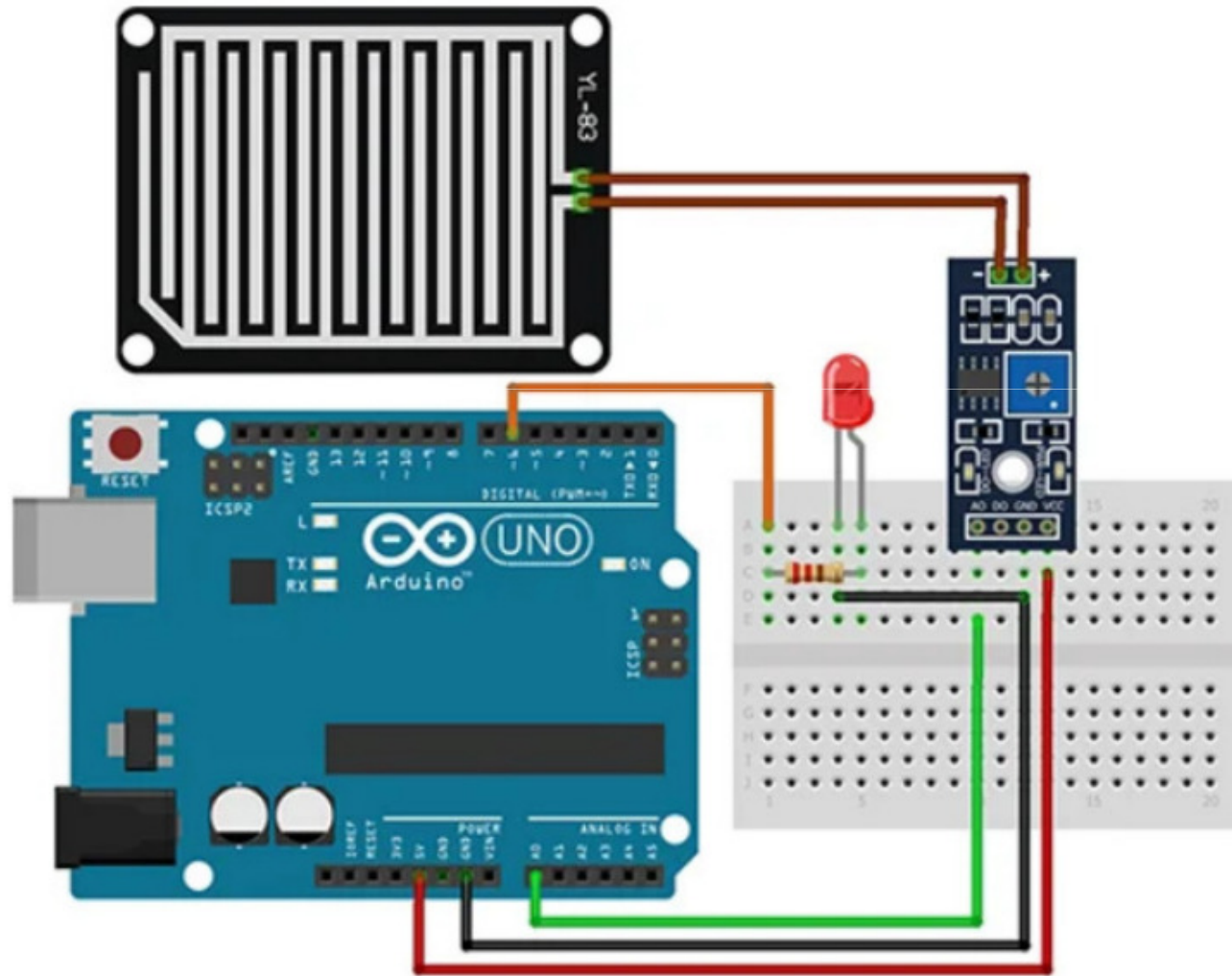
```
#define RAIN_SENSOR A0  
#define BUZZER 3
```

```
void setup() {  
  pinMode(RAIN_SENSOR, INPUT);  
  pinMode(BUZZER, OUTPUT);  
  Serial.begin(9600);  
}
```

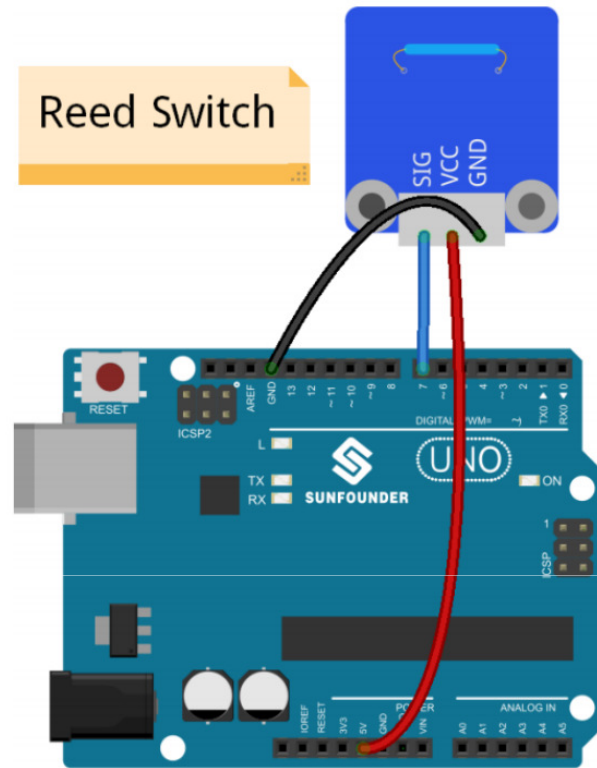
```
void loop() {  
  int sensorValue = analogRead(RAIN_SENSOR);  
  Serial.println(sensorValue);  
  
  if (sensorValue < 102) { // Threshold for rain detection  
    digitalWrite(BUZZER, HIGH);  
  } else {  
    digitalWrite(BUZZER, LOW);  
  }  
  delay(1000); // Delay for stability  
}
```

Senzor dažďa 2

Skontroluj údaje v sériovom monitore



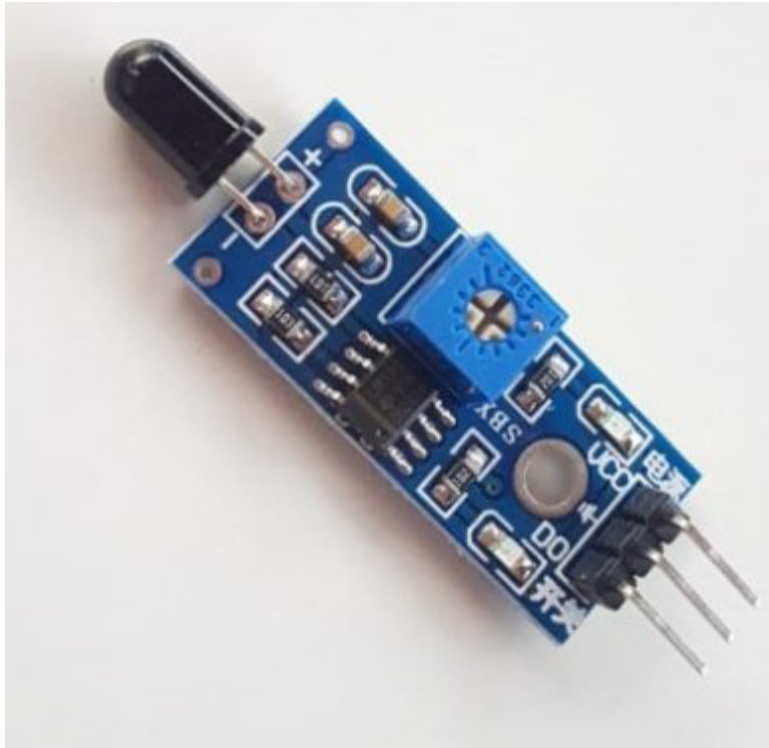
Snímač magnetického poľa



Doplň zapojenie s LED

```
const int digitalInPin = 7; // jazýčkový spínač pripojiť k pin7
const int ledPin = 13; // pin13 vstavaný viedol
void setup ()
{
  pinMode (digitalInPin, INPUT); // nastaví digitalInPin ako vstup
  pinMode (ledPin, OUTPUT); // nastaví ledPin ako výstup
}
void loop ()
{
  boolean stat = digitalRead (digitalInPin); // načítanie hodnoty pin7 sa stat
  if (stat == HIGH) // ak vysoko
  {
    digitalWrite (ledPin, LOW); // potom vypnúť viedla
  }
  else // iný
  {
    digitalWrite (ledPin, HIGH); // zapnutie LED
  }
}
```

Snímač ohňa



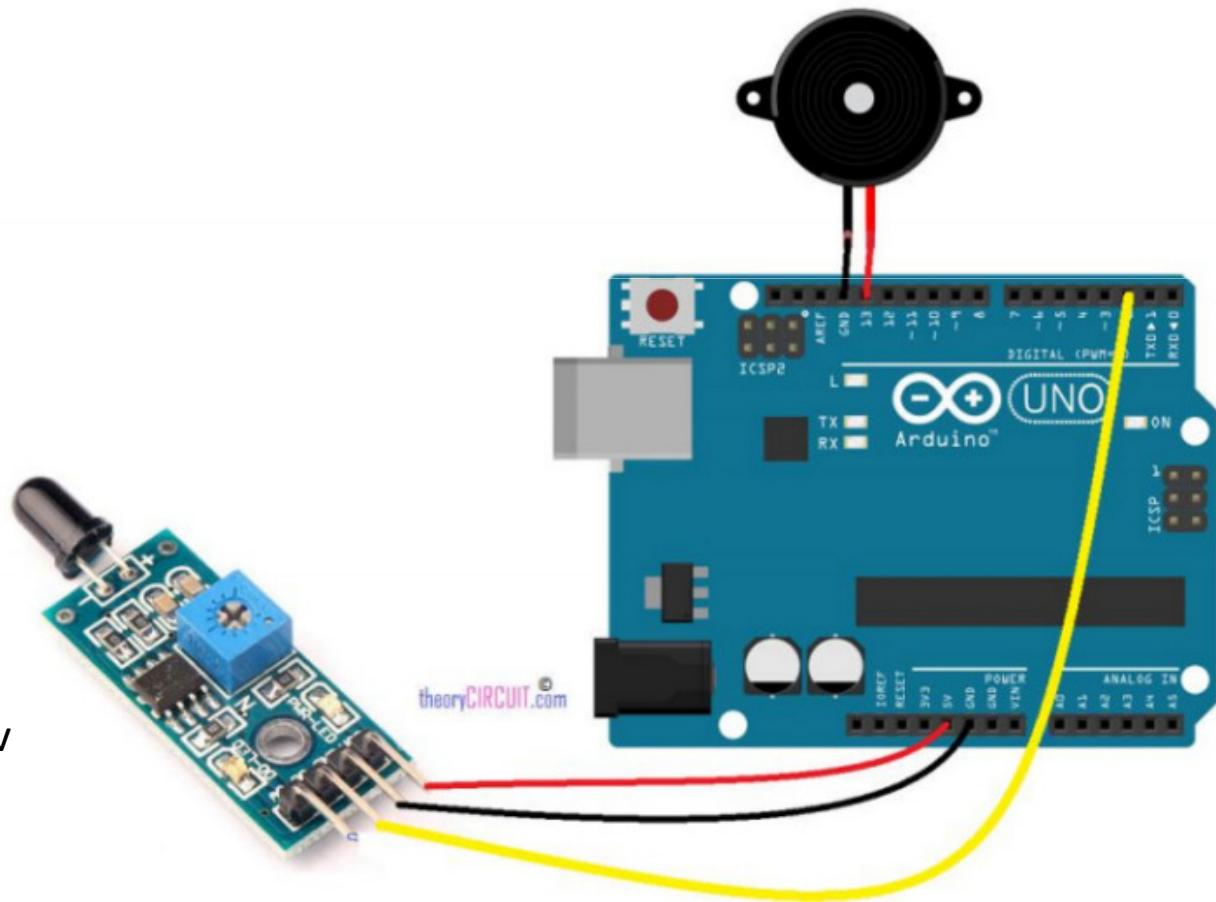
- Senzor je citlivý na svetlo o vlnovej dĺžke približne 760nm, teda v infračervenej časti spektra.
- Na jeho rozmery a cenu je pomere citlivý a aj malý plameň dokáže zaregistrovať na vzdialenosti rádovo v desiatkach centimetrov.
- Na module je možné ladiť citlivosť zabudovaným potenciometrom.

Doplň zapojenie s LED

Snímač ohňa

```
int Buzzer = 13; // Use buzzer for alert
int FlamePin = 2; // This is for input pin
int Flame = LOW; // HIGH when FLAME Exposed
void setup() {
  pinMode(Buzzer, OUTPUT);
  pinMode(FlamePin, INPUT);
  Serial.begin(9600);
}
void loop() {
  Flame = digitalRead(FlamePin);
  if (Flame== HIGH)
  {
    Serial.println("HIGH FLAME");
    digitalWrite(Buzzer, LOW);
  }
  else
  {
    Serial.println("No flame");
    digitalWrite(Buzzer, HIGH);
  }
}
```

Doplň zapojenie s LED, uprav program pre bzučiak pri detekcii ohňa



Termistor 2

NTC termistor MF5A-3 10K 3950 pre teploty -30°C – 150°C



Otestuj zapojenie a sériový monitor

```
int ThermistorPin = 0;
int Vo;
float R1 = 10000;
float logR2, R2, T, Tc, Tf;
float c1 = 1.009249522e-03, c2 = 2.378405444e-04, c3 = 2.019202697e-07;

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

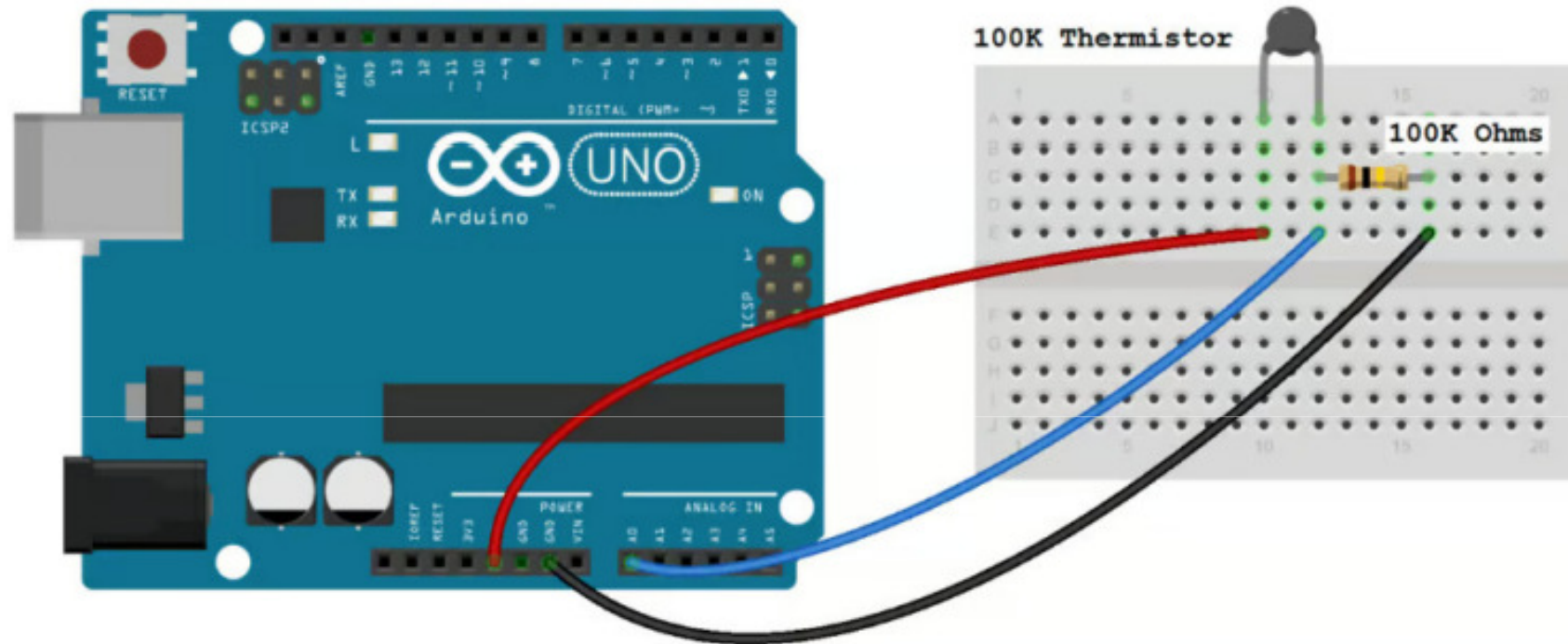
void loop() {

  Vo = analogRead(ThermistorPin);
  R2 = R1 * (1023.0 / (float)Vo - 1.0);
  logR2 = log(R2);
  T = (1.0 / (c1 + c2*logR2 + c3*logR2*logR2*logR2));
  Tc = T - 273.15;
  Tf = (Tc * 9.0) / 5.0 + 32.0;

  Serial.print("Temperature: ");
  Serial.print(Tf);
  Serial.print(" F; ");
  Serial.print(Tc);
  Serial.println(" C");

  delay(500);
}
```

Termistor 2



Otestuj zapojenie a sériový monitor

Ultrazvuk 3

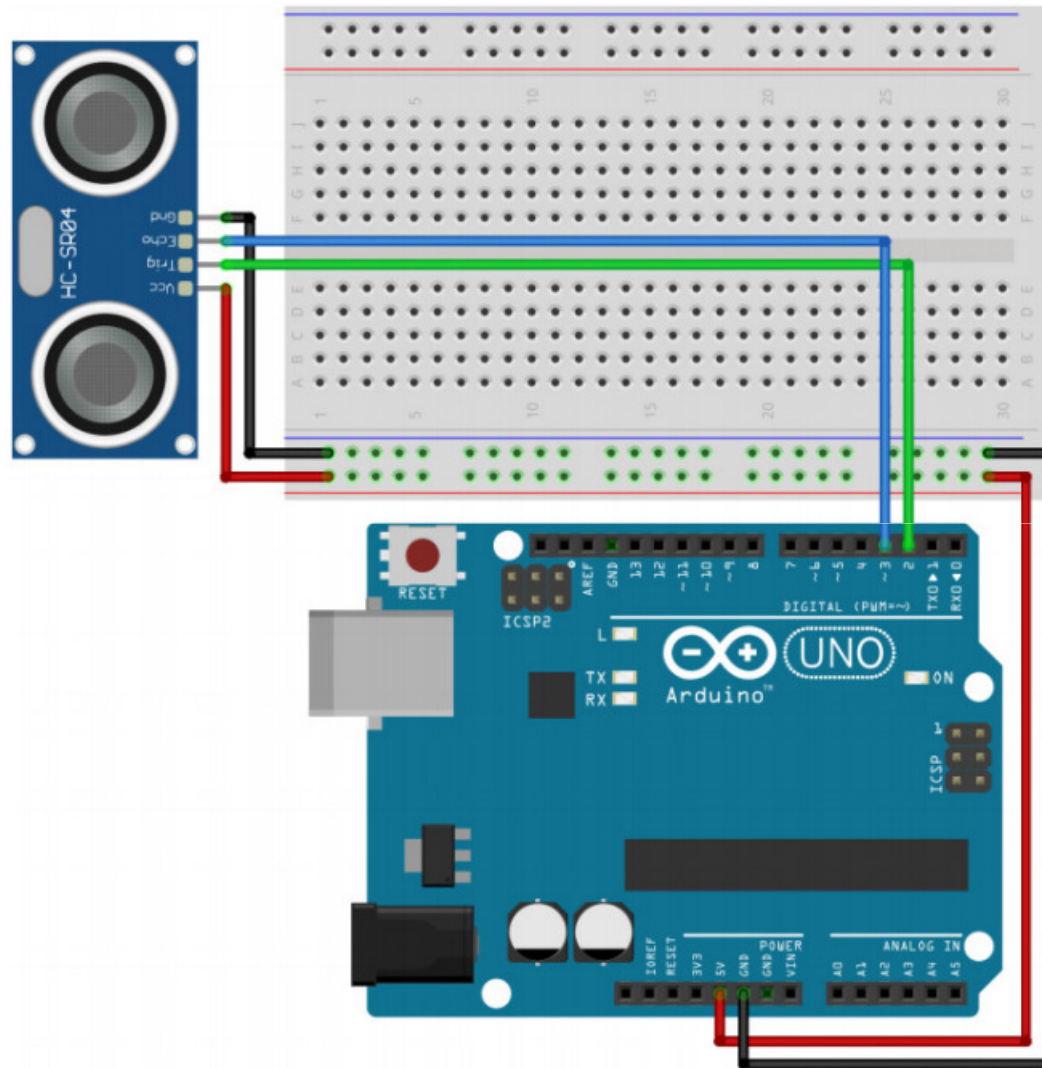


- Ultrazvukový senzor na meranie vzdialenosti.
- Detekčná vzdialenosť od 1 cm do 3 m s presnosťou na 0.3 cm.
- Pozorovací uhol do 15°.

Špecifikácie:

- Operačné napätie 3.8 – 5.5 V
- Spotreba 8 mA
- Frekvencia ultrazvuku 40 kHz
- Maximálna vzdialenosť 300 cm
- Minimálna vzdialenosť 1 cm
- Uhol detekcie 15°

Ultrazvuk 3

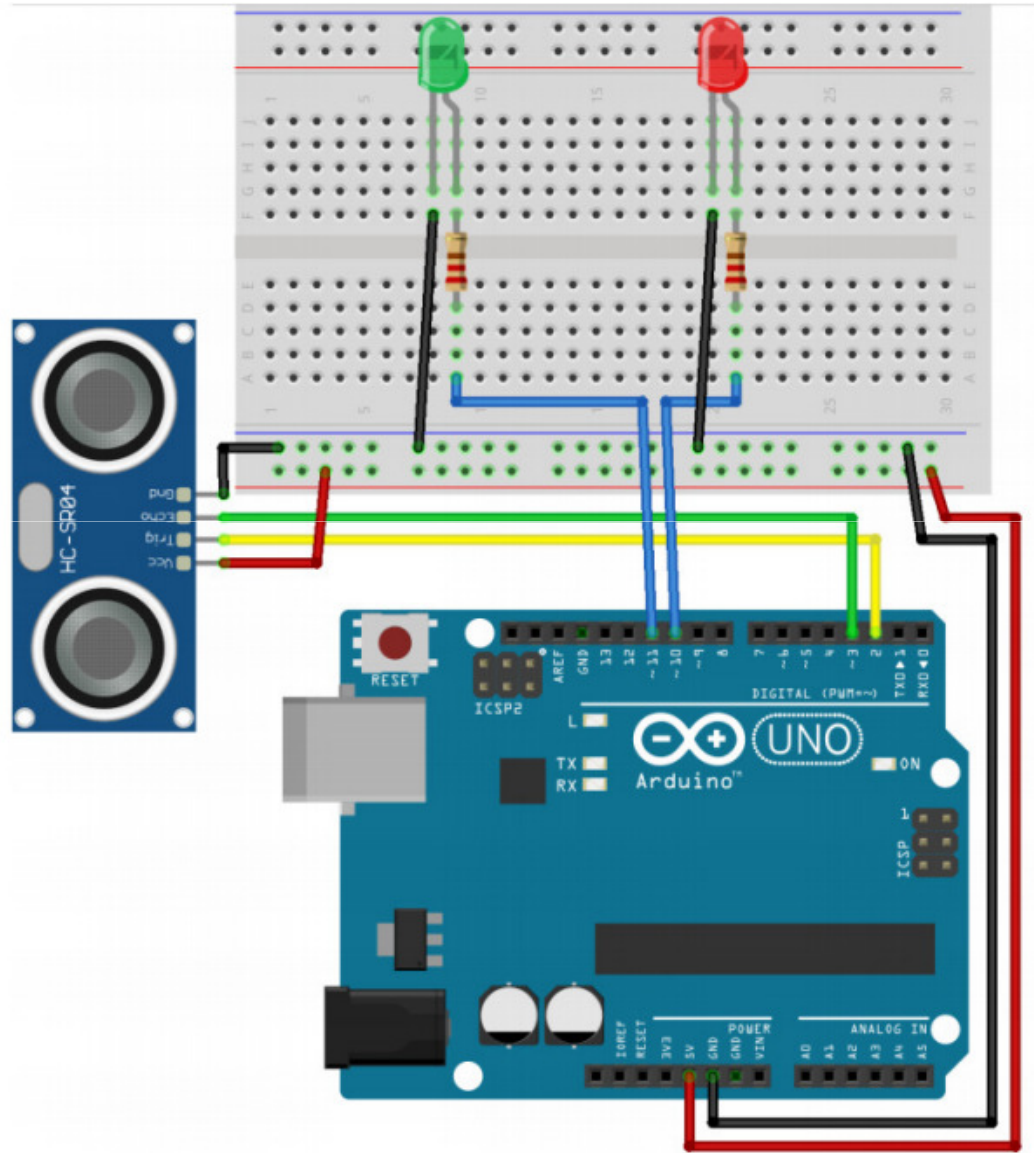


```
#define trigPin 2
#define echoPin 3
long duration;
int distance;
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}
void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2;
  Serial.print("Distance = ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(50);
}
```

Prelož do slovenčiny

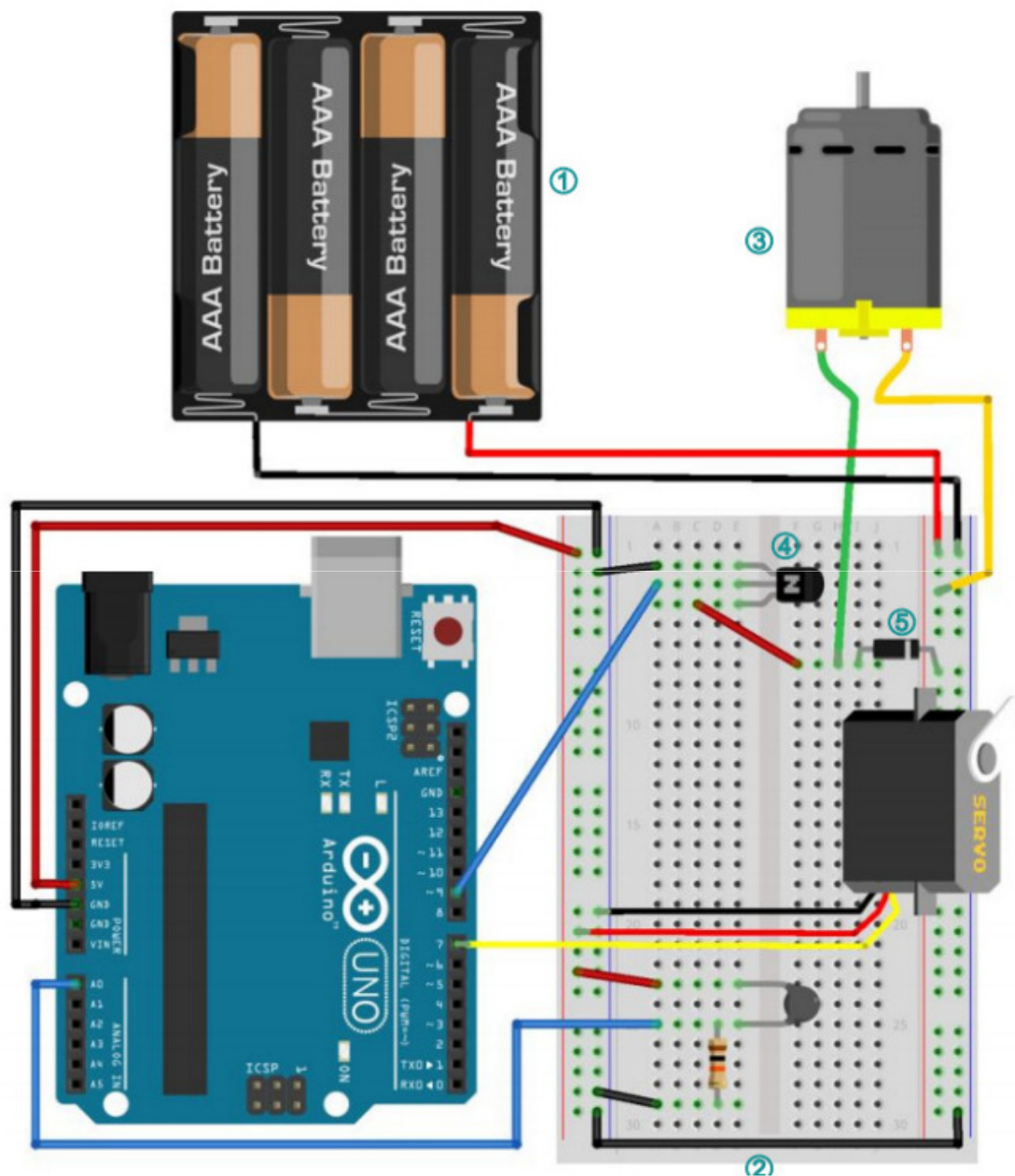
$$\text{Vzdialenosť} = \text{rýchlosť zvuku} * \text{čas} / 2$$

Ultrazvuk



Zmeň prahové hodnoty na 20 cm a 50 cm

Vetrák



Pri 30°C nastav servomotor na 180° a zapni motorček ventilátora

Príklady

- obsah obdĺžnika
- objem kvádra
- pytagorova veta
- faktoriál
- povrch a objem gule
- povrch a objem kužeľa
- aritmetický priemer 3 čísiel
- súčin troch čísiel
- Napíš program pre výpočet súčtu, rozdielu, súčinu a podielu 2 čísiel

```
int a;
int b;
int obsah;

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

void loop() {

  Serial.print("Vlozte rozmer a: ");
  while (Serial.available()==0) {
  }
  float a=Serial.parseFloat();
  Serial.println(a);

  while(Serial.available() > 0) {
    Serial.read();
  }

  delay(1000);

  Serial.print("Vlozte rozmer b: ");
  while (Serial.available()==0) {
  }
  float b=Serial.parseFloat();
  Serial.println(b);

  while(Serial.available() > 0) {
    Serial.read();
  }

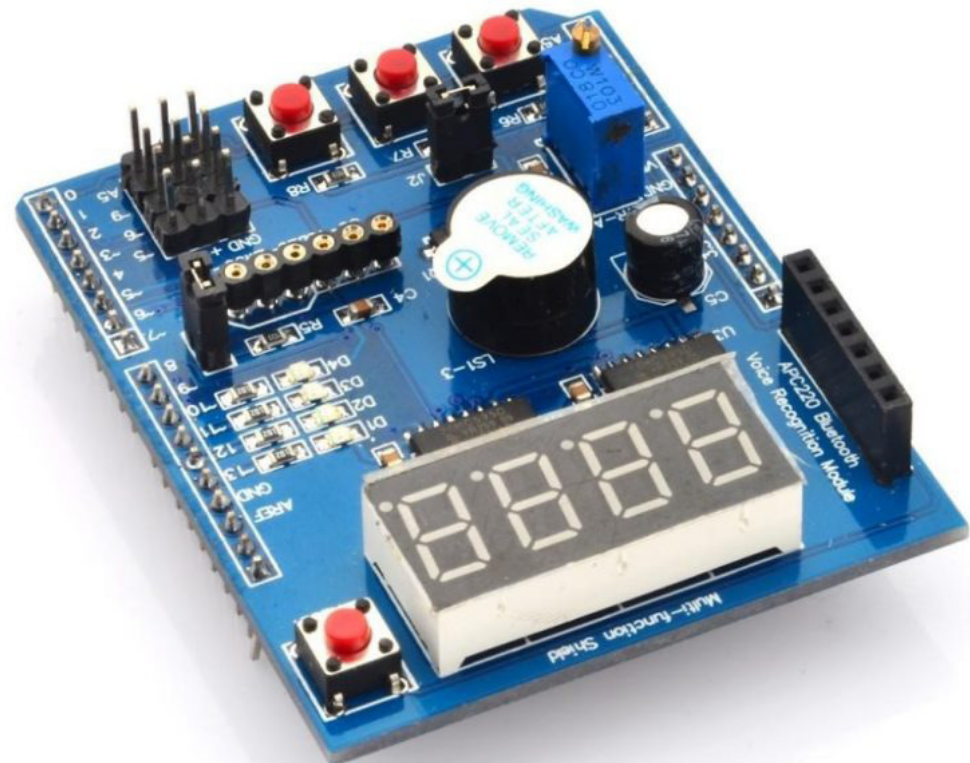
  delay(1000);

  Serial.println("");
  obsah = a*b;

  Serial.println("Obsah obdlznika : ");
  Serial.print( obsah );
  Serial.println("");
}
}
```

Výukový multishield

- blikanie 1
- striedavý blikáč
- tlačidlá – rôzne funkcie
- test tlačidiel
- siréna 1
- svetelný had
- pulzujúca led
- stopky
- 4 pulzujúce LED
- LED a SOS
- postupné spínanie LED
- bzučiak



Úlohy – animácia – 17.3.2026

1. Uprav program – [Svetelný efekt](#) so šiestimi LED diódami
2. Uprav program – [Generátor zvuku](#) – s tromi tlačidlami a frekvenciou 1000, 2000, 3000
3. Uprav program – [7 segmentový displej](#) – aby zobrazoval čísla len v jednom smere
4. Uprav program – [Policajná siréna](#) – o ďalšiu dvojicu LED
5. Uprav program – [Fotorezistor – led dióda](#) – s analógovou hodnotou A1
6. Uprav program – [Fotorezistor 3 LED](#) – na 2 LED
7. Zapoj podľa schémy – [Fotorezistor – bzučiak](#), otestuj iné parametre delay
8. Otestuj program – [Semafor 3](#)

Úlohy – Joystick – 18.3.2026

1. Uprav program – [Joystick – sériový monitor](#) s parametrom delay 2000ms
2. Zapoj podľa schémy – [Joystick – LED](#)

Úlohy – LCD displej – 18.3.2026

1. Uprav program [LCD bežiaci text](#), spomalenie rýchlosti, zmeň hodnotu zdržania textu
2. Uprav program [LCD displej so snímačom DHT](#), prelož popis do slovenčiny vymeň riadky
3. Uprav program [LCD displej s meračom vzdialenosti](#), prelož do slovenčiny, zapoj do kontaktnej plochy

Úlohy – Motorčeky – 18.3.2026

1. Uprav program [Servomotor 180°](#) na zmenu polohy na 90°
2. Zapoj podľa schémy [Servomotor – nastavenie](#) uprav polohy na 40°, 80°, 120°, 160°
3. Zapoj podľa schémy [Servomotor – potenciometer](#) – ovládací pin potenciometra na A1

4. Zapoj podľa schémy – [Servomotor 2 tlačidlá verzia 2](#)
5. Zapoj podľa schémy – [Servomotor ovládaný tromi tlačidlami](#) , výchylky 45° , 90° , 135°
6. Uprav program – [Roztočenie jednosmerného motorčeka](#) – otestuj rôzne rýchlosti otáčok od 0 do 200.
7. Otestuj program – [Reguláci otáčok jednosmerného motorčeka](#)
8. Otestuj program – [Regulácia otáčok jednosmerného motorčeka fotorezistorom](#)
9. Otestuj program – [Čidlo prekážky so servomotorom](#) uprav na 180°
10. Otestuj zapojenie – [Servomotor – snímač dažďa](#) , servomotor sa vychýli pri detekcii o 180°
11. Uprav program – [Servomotor – snímač vzdialenosti](#), prahová vzdialenosť – 20 cm, natočenie serva o 180°, prelož parametre do slovenčiny
12. Otestuj program – [Fotorezistor – servomotor](#)

Úlohy – Snímače – 19.3.2026

1. Uprav program - [Čidlo prekážky](#) – doplň zapojenie LED diódy
2. Uprav program – [Detektor dymu a alkoholu](#) - zmeň prahové hodnoty
3. Uprav program – [DHT11 – snímač teploty](#) – vymeň riadky a prelož do slovenčiny
4. Otestuj program – [Pohybové čidlo 2](#) – skontroluj výpisy v sériovom monitore
5. Uprav program – [Detektor dažďa](#) – zmeň prahové hodnoty
6. Otestuj program – [Detektor dažďa 2](#) – skontroluj údaje v sériovom monitore
7. Otestuj program – [Snímač magnetického poľa](#) – doplň zapojenie s LED
8. Uprav program – [Snímač ohňa](#) – doplň zapojenie s LED

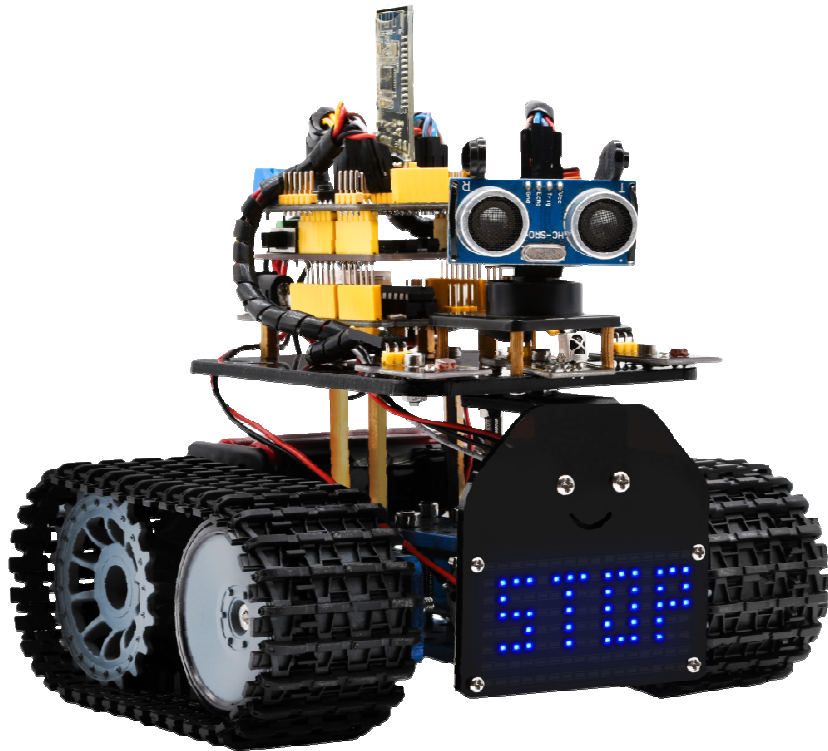
9. Otestuj program – [Termistor 2](#) – otestuj sériový monitor
10. Otestuj program – [Ultrazvuk 3](#) – otestuj zapojenie
11. Otestuj program – [Ultrazvuk](#) - zmeň prahové hodnoty na 20 cm a 50 cm
12. Uprav program – [Skleník](#) - zapni servomotor pri 25°C, nastav polohu na 180°
13. Uprav program – [Vetrák](#) - pri 30°C nastav servomotor na 180° a zapni motorček

Úlohy – Príklady – 19.3.2026

1. [Obsah obdĺžnika](#)
2. [Objem kvádra](#)
3. [Pytagorova veta](#)
4. [Faktoriál](#)
5. Vytvor program - Povrch a objem gule
6. Vytvor program - Povrch a objem kužeľa
7. Vytvor program - Aritmetický priemer 3 čísiel
8. Vytvor program - Súčin troch čísiel
9. Vytvor program pre výpočet súčtu, rozdielu, súčinu a podielu 2 čísiel

Cieľ: Naprogramovanie trasy robotického tanku

Minirobotický tank



Tím Keyestudio navrhol mini tankového robota. Má procesor, ktorý je programovateľný pomocou Arduino IDE, pričom jeho piny sú mapované na senzory a akčné členy pomocou shield prepojenia, do ktorého je zapojený procesor.

Ten číta senzory, ovláda akčné členy a rozhoduje sa, ako majú fungovať.

Môže vykonávať viacero funkcií, ako je vyhýbanie sa prekážkam, IR diaľkové ovládanie, ovládanie BT, sledovanie svetla a tak ďalej.

Cieľ: Naprogramovanie trasy robotického tanku

Úlohy – Robotický tank – 20.3.2026

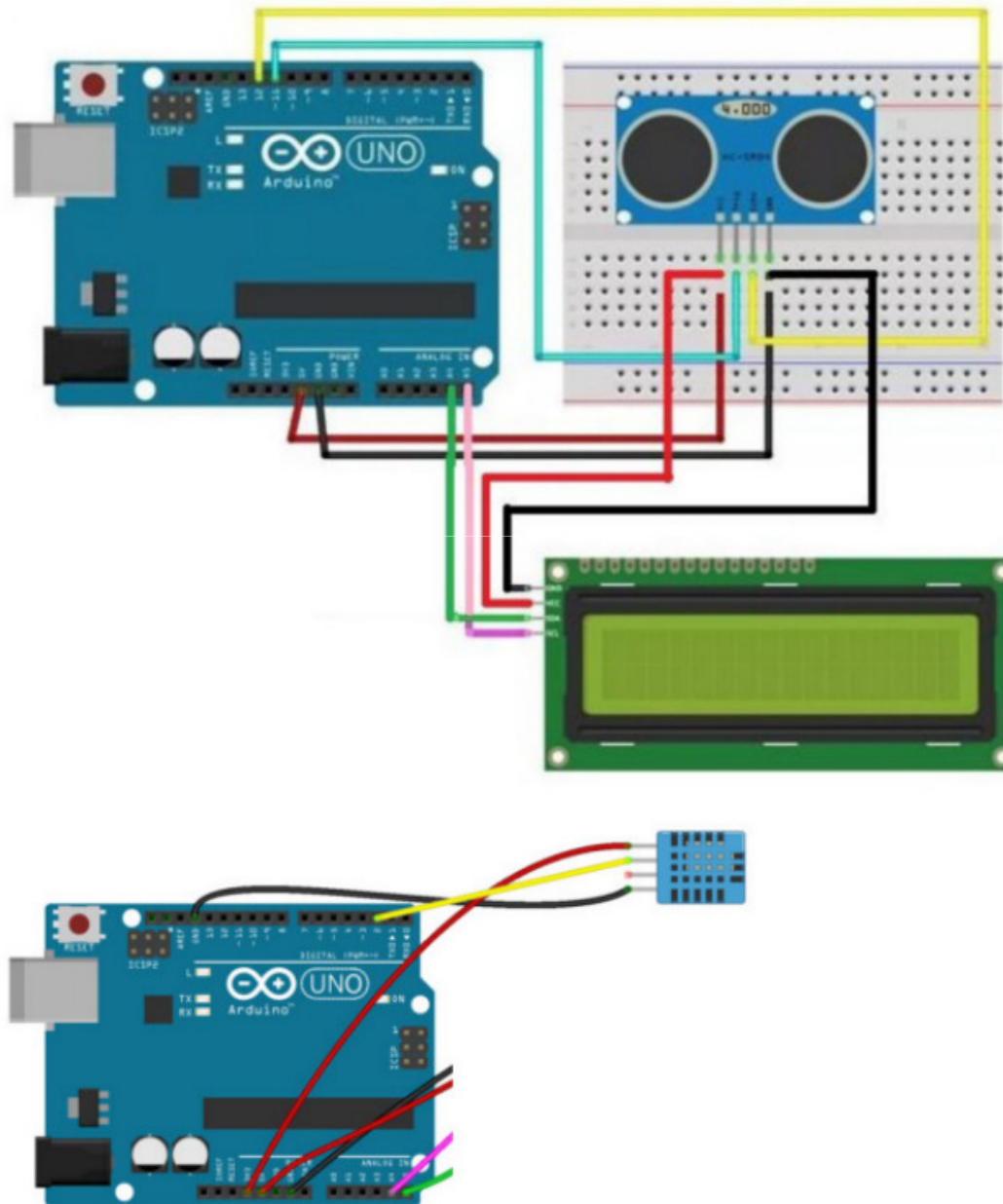
1. [Testovanie ovládania tanku pomocou bluetooth](#) – skúška aplikácie Tankcar
2. Naprogramovanie trasy robotického tanku:
 - A: [Trasa dopredu – dozadu](#)
 - B: [Otáčanie namieste](#)
 - C: [Pohyb do štvorca](#)

Naprogramuj trasu:

Dopredu 1 sekunda – stop 1 sekunda – dozadu 1 sekunda – stop 1 sekunda – otáčanie namieste vľavo 2 sekundy – stop 1 sekunda – otáčanie namieste vpravo 2 sekundy – stop 1 sekunda – pohyb do štvorca ([podľa podprogramu](#))

3. [Test pre sledovanie svetla](#)

Cieľ: LCD displej so snímačom teploty a snímačom vzdialenosti



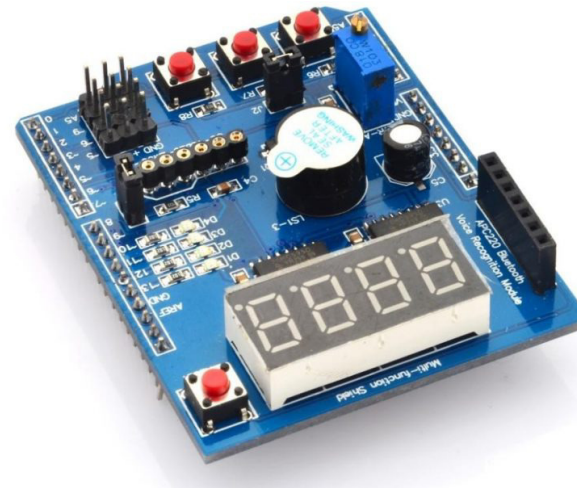
Snímač teploty a vzdialenosti

Otestuj program a vymeň riadky s údajmi

Výukový multishield

- Shield určený hlavne do škôl alebo ako výukový modul pre začiatočníkov.
- Vďaka tomu že je dizajnovaný ako shield (nadstavba), nie je nutné žiadne pájkovanie ani prepájanie káblíkmi.
- Modul stačí jednoducho nasunúť na vývojovú dosku
- Na shielde sa nachádzajú základné periférie ako:
 - 4 x indikačná LED
 - 3 x tlačidlo + 1 x reset
 - 1 x bzučiak
 - 1 x 7 segmentový displej s čipom 74HC595
 - 1 x potenciometer / odporový trimmer

Shield ja napájaný priamo z vývojovej dosky teda nie je nutné dokupovať žiadny zdroj. Je kompatibilný s Arduino UNO, MEGA, DUE, LEONARDO prípadne ich kópiami.



Cieľ: LCD displej so snímačom teploty a snímačom vzdialenosti

Úlohy – Snímač teploty a vzdialenosti - 20.3.2026

1. Uprav program pre [LCD displej so snímačom teploty a vzdialenosti](#) – premeň poradie riadkov na displeji
-

Cieľ: Výukový multishield

Úlohy – Výukový multishield – 20.302026

1. Otestuj program pre [animáciu led](#)
2. Otestuj program pre [striedavý blikáč](#)
3. Otestuj program pre [tlačidlá](#)
4. Otestuj program pre [tlačidlá 2](#)
5. Zapni [sirénu](#)
6. Uprav program pre [pulzujúcu led](#) – súčasne budú pulzovať dve led
7. Vytvor program pre [animáciu led](#) ako SOS
8. Otestuj program [svetelný had](#) so 4 led