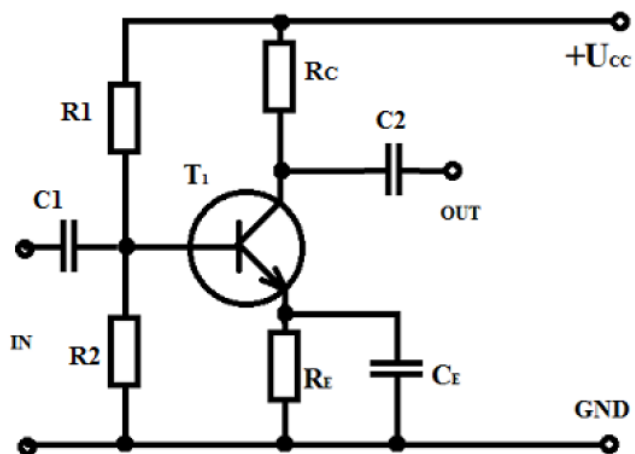


1. Lineárny napájací zdroj – bloková schéma, výstupné napätia z jednotlivých blokov, súčiastky pre jednotlivé bloky

2. Nízkofrekvenčný zosilňovač, funkcie jednotlivých prvkov:

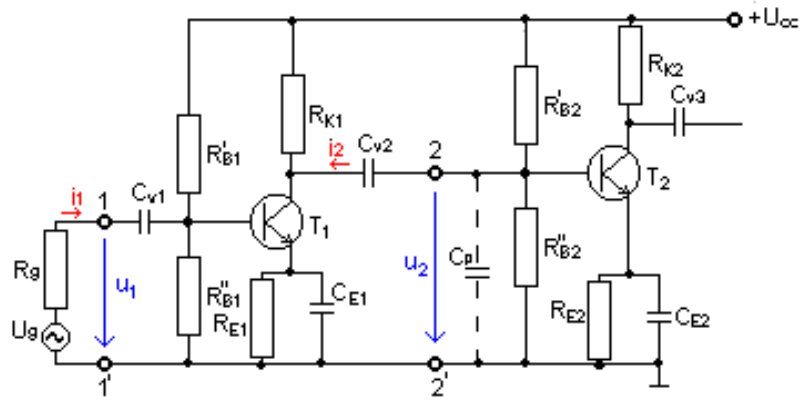
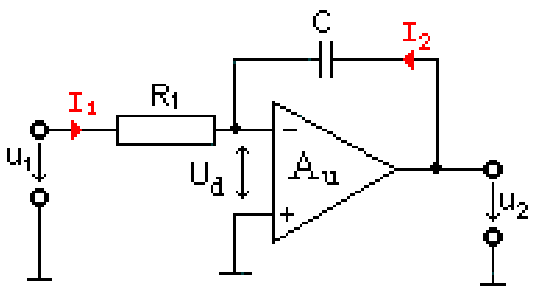
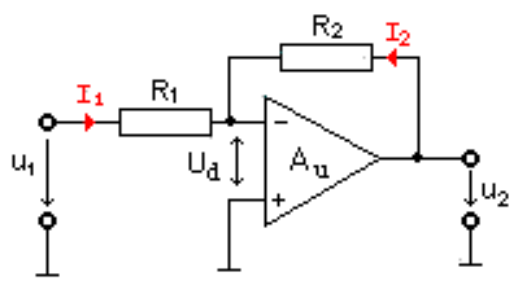
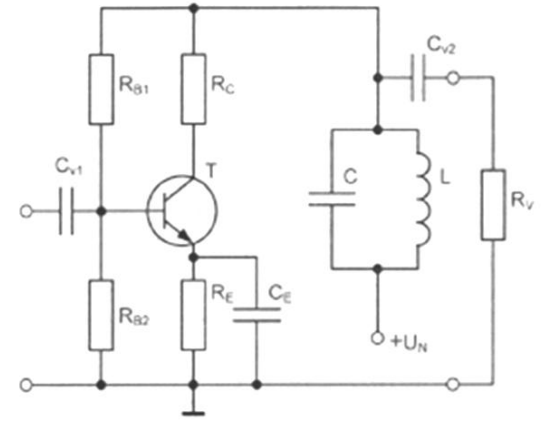
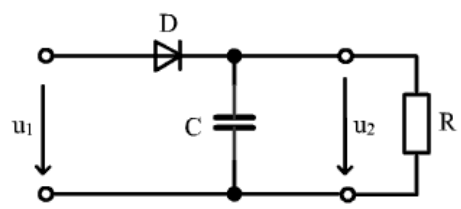
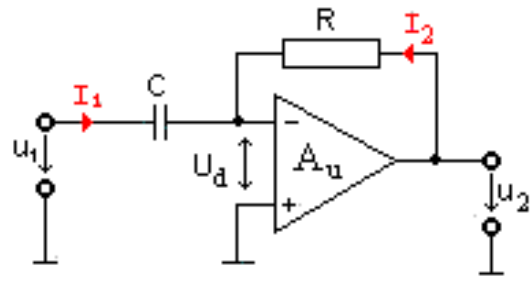
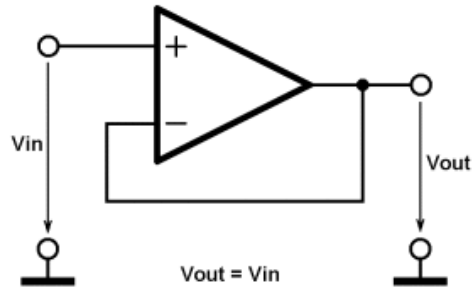
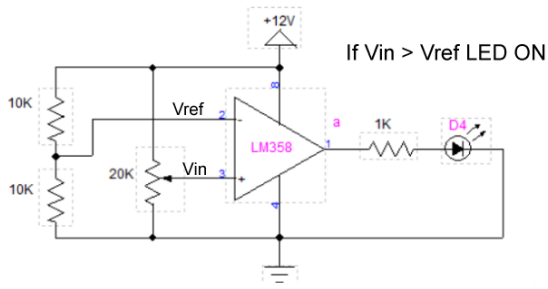


3. Ideálny operačný zosilňovač, vlastnosti

4. Bloková schéma vysielača

5. Bloková schéma prijímača

Pomenuj jednotlivé zapojenia

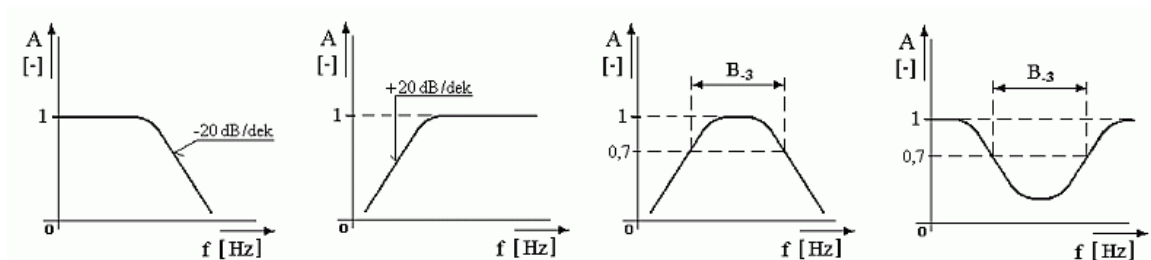


7. Definuj logickú funkciu AND, OR, XOR, XNOR

8. VA charakteristika tyristora

9. Triak, značka, popis, použitie

10. Pomenuj jednotlivé frekvenčné filtre:

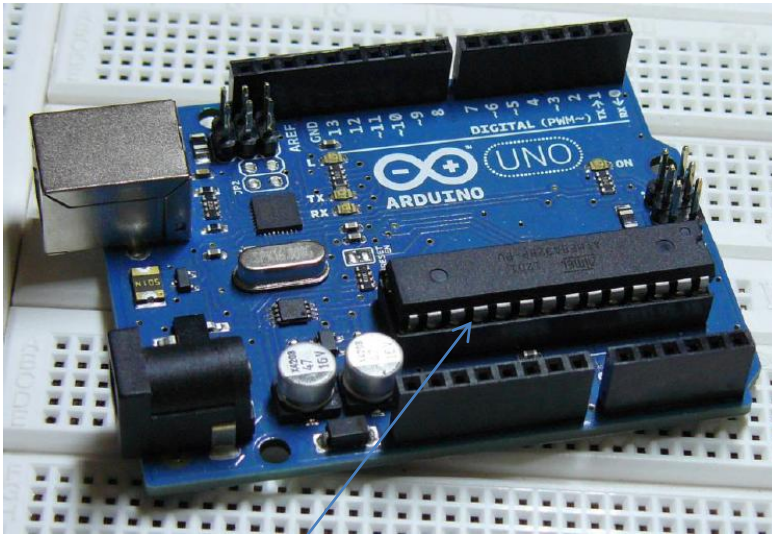


11. Definuj zosilnenie

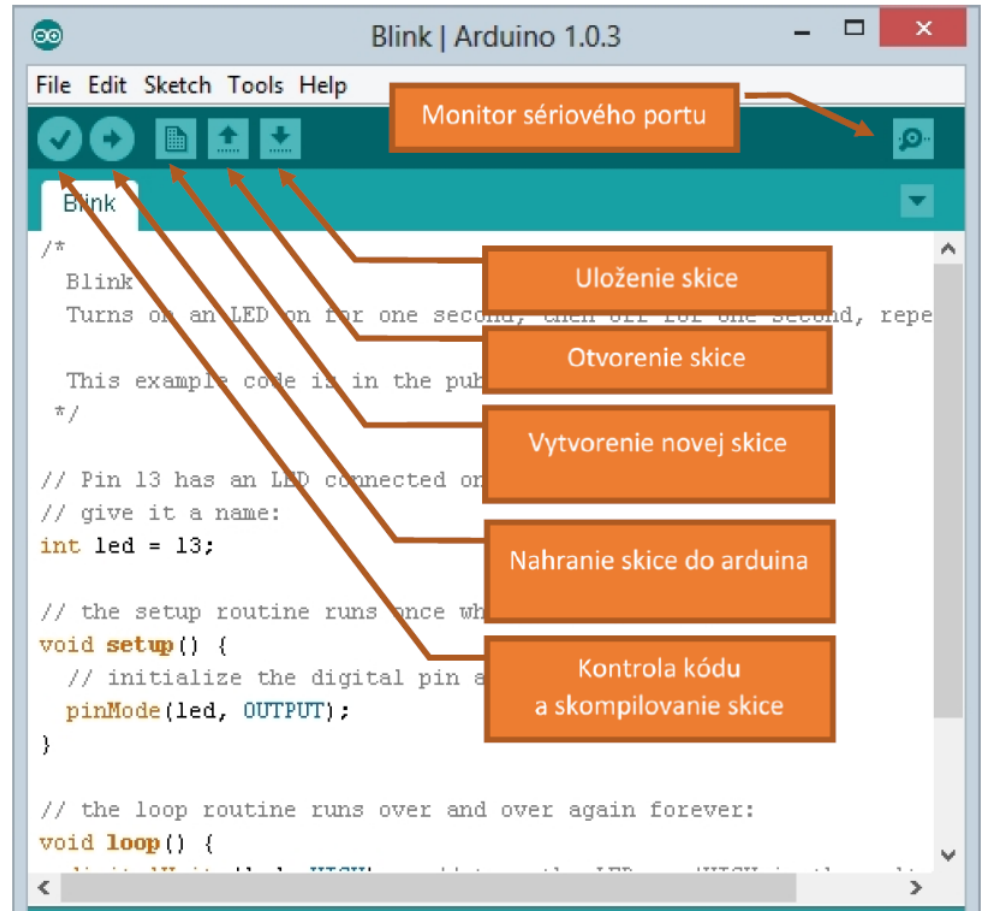
12. Rozdelenie zosilňovačov

Čo je to 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)

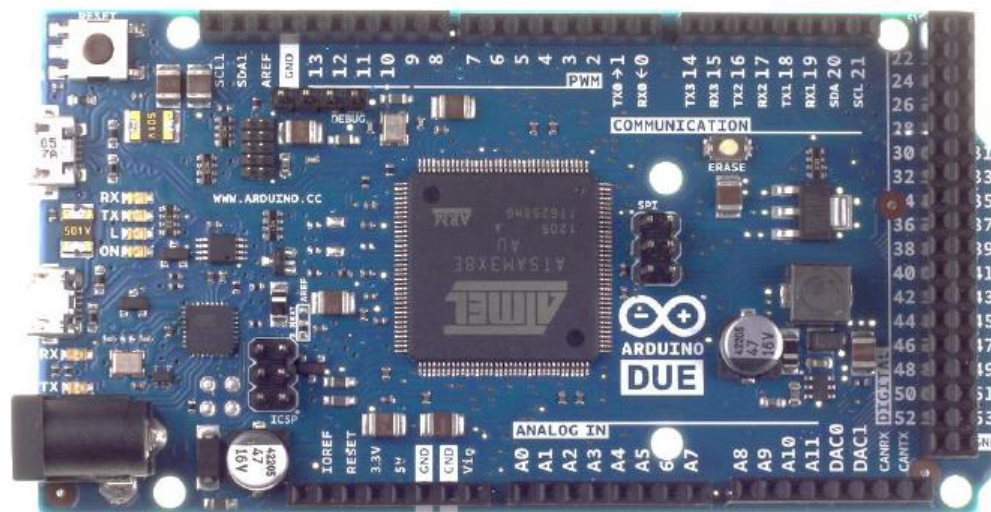


8 bitový mikrokontrolér



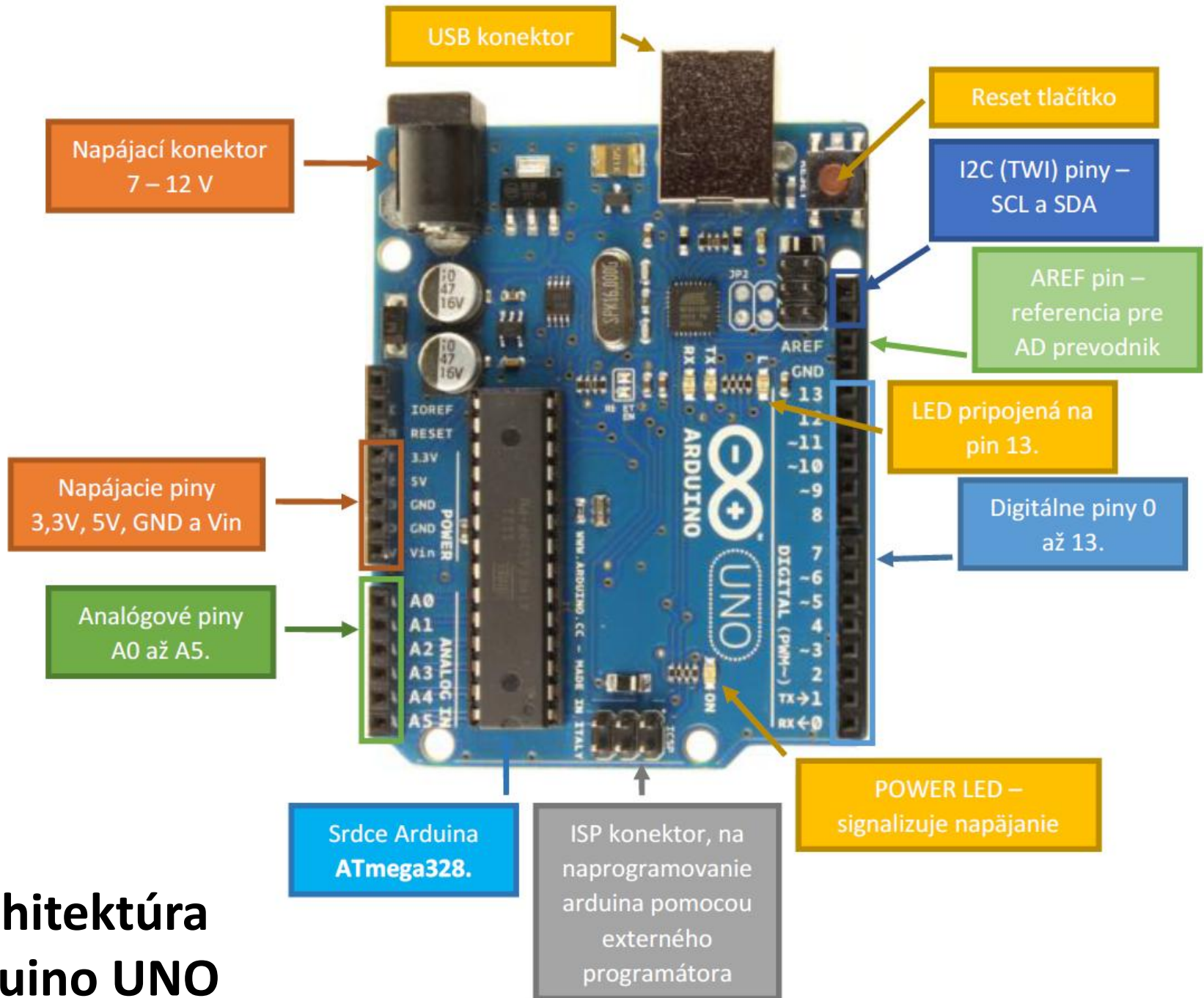
Základná štruktúra Arduino programu

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

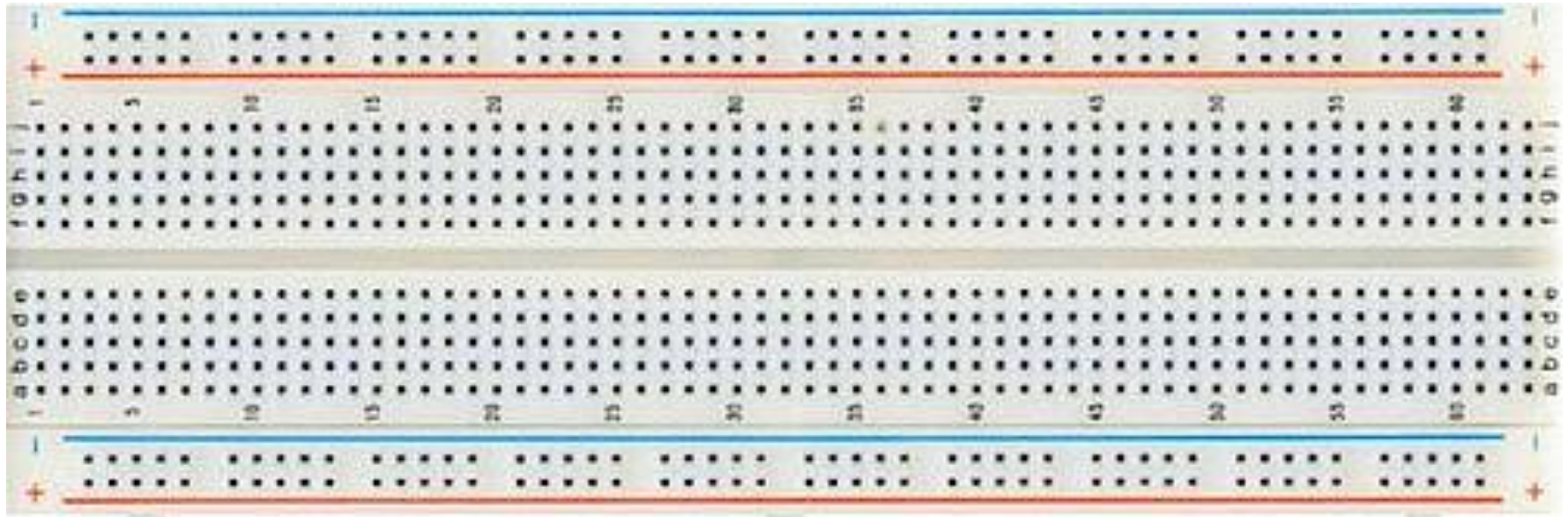
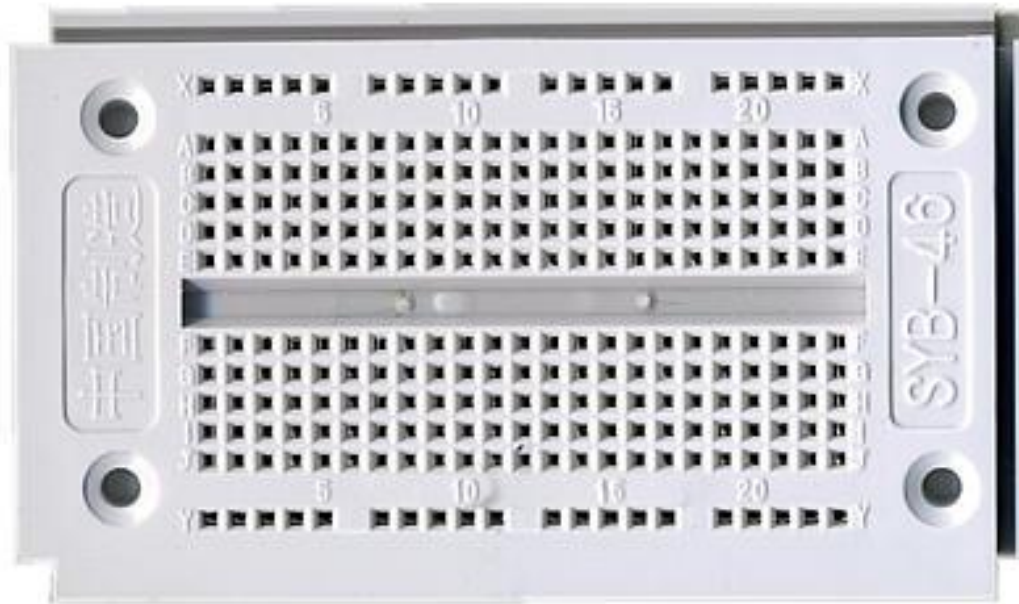


Zdrojové programy: www.majstrissmt.eu – učebné dokumenty, 1. ročník, Učebnica Arduina (pdf)

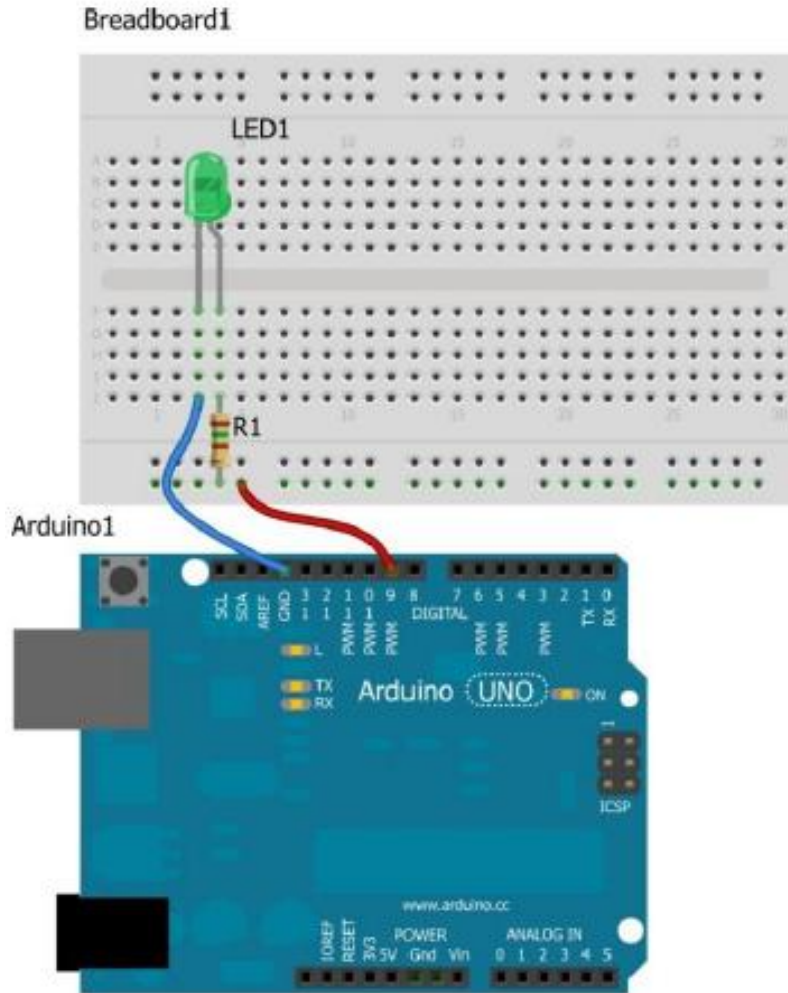
Architektúra Arduino UNO



Kontaktné polia



Pulzujúca LED

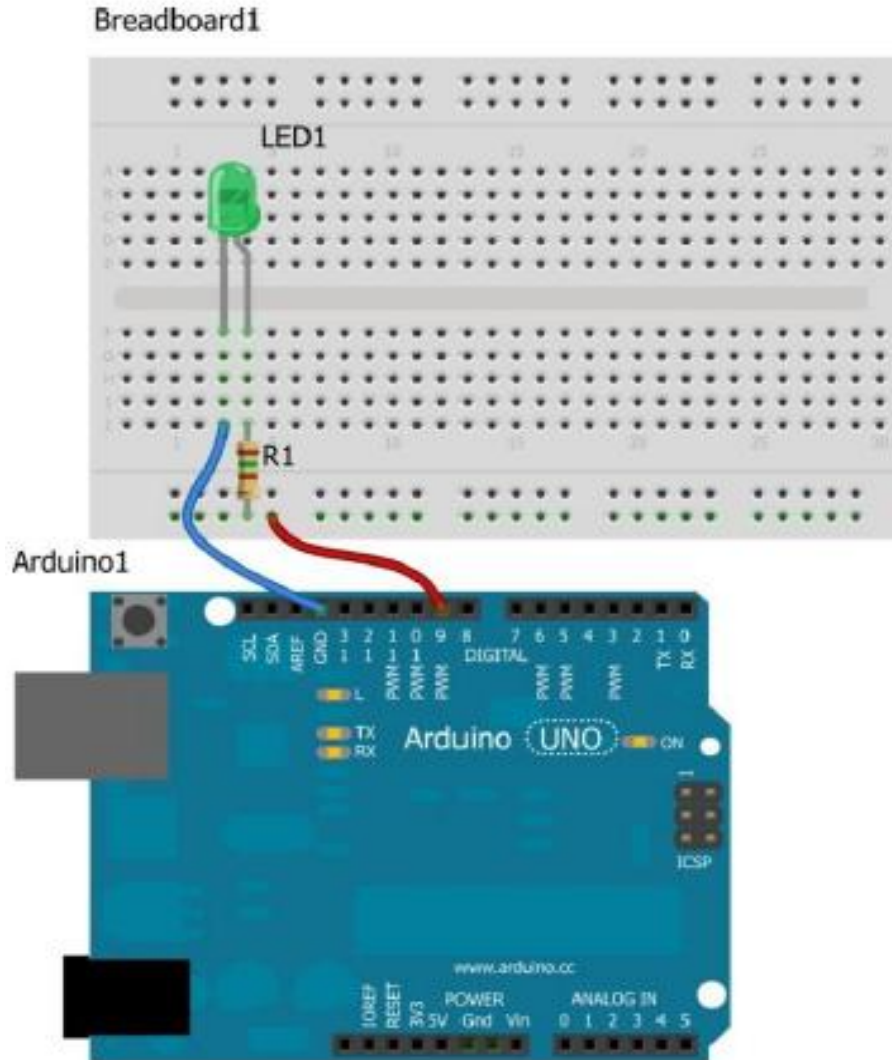


```
int ledPin = 9; // číslo pinu na ktorý je pripojená LEDka
int brightness = 0; // jas LEDky
int stepValue = 5; // veľkosť kroku na nastavenie jasu LEDky
int direction = 1; // Smer zvyšovanie alebo znižovanie jasu
```

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

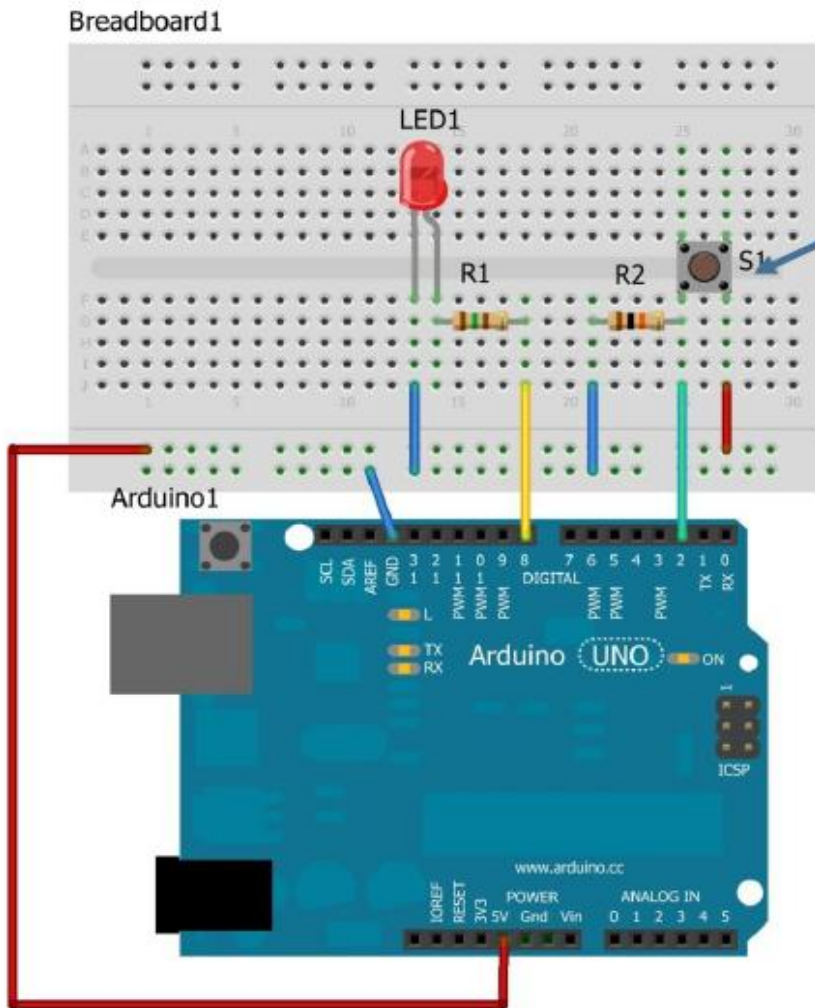
void loop() {
  // nastavenie jasu LEDky
  analogWrite(ledPin, brightness);
  // ak jas dosiahne maximálnej hodnoty, zmeníme smer na
  // znižovanie
  if(brightness == 255){
    direction = 2;
  }
  // ak jas dosiahne minimálnej hodnoty, zmeníme smer na
  // zvyšovanie
  if(brightness == 0) {
    direction = 1;
  }
  switch(direction){
    case 1: // zvyšovanie jasu
      brightness = brightness + stepValue;
      break;
    case 2: // znižovanie jasu
      brightness = brightness - stepValue;
      break;
  }
  delay(40);
}
```


SOS s LED



```
int ledPin = 9;
void setup() {
  // nastavíme ledPin (pin číslo 9) 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
}
```

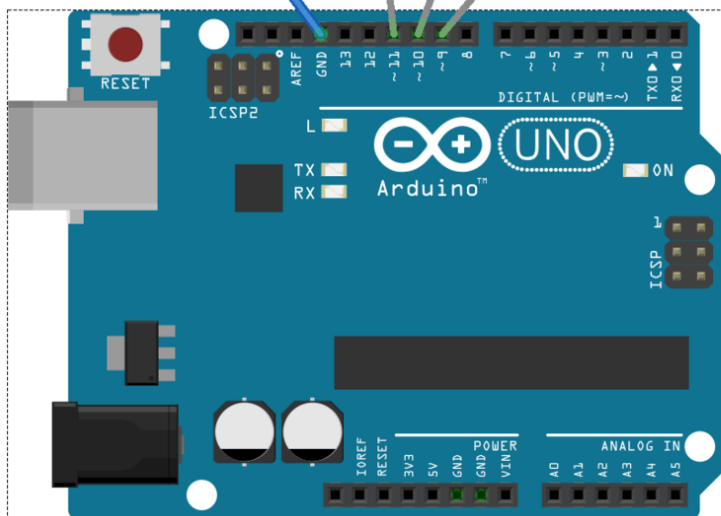
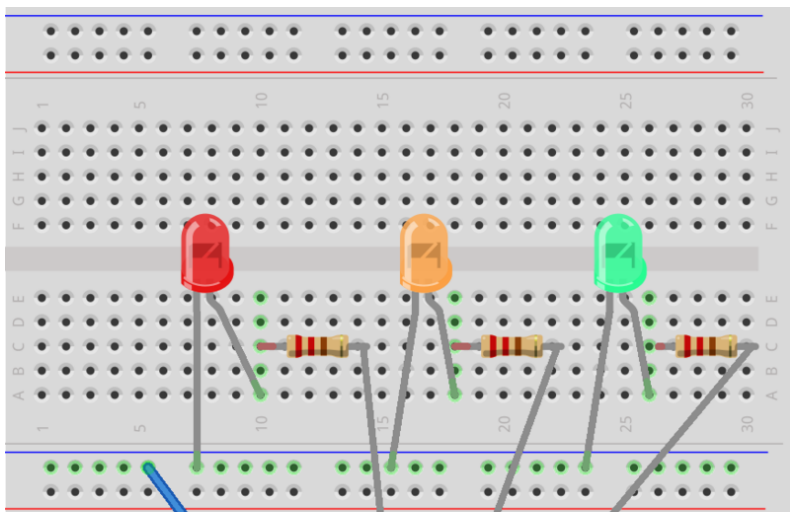
Interaktívna LED



R1 = 150Ohm, R2 = 10kOhm

```
//číslo pinu pre tlačítko
int buttonPin = 2;
//číslo pinu pre LEDku
int ledPin = 8;
//premenná uchovávajúca stav tlačítka
int buttonState = 0;
void setup() {
  // nastavenie pinu pre LEDku ako výstupný
  pinMode(ledPin, OUTPUT);
  // nastavenie pinu pre tlačítko ako vstupný
  pinMode(buttonPin, INPUT);
}
void loop(){
  // načítame stav tlačítka
  buttonState = digitalRead(buttonPin);
  // skontrolujeme či je tlačítko stlačené alebo nie
  // ak je, tzn. premenná buttonState má hodnotu HIGH
  if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH); // zapneme LEDku
  } else {
    // ak tlačítko nie je zopnuté (stlačené), LEDku vypneme
    digitalWrite(ledPin, LOW);
  }
}
```

Program s LED

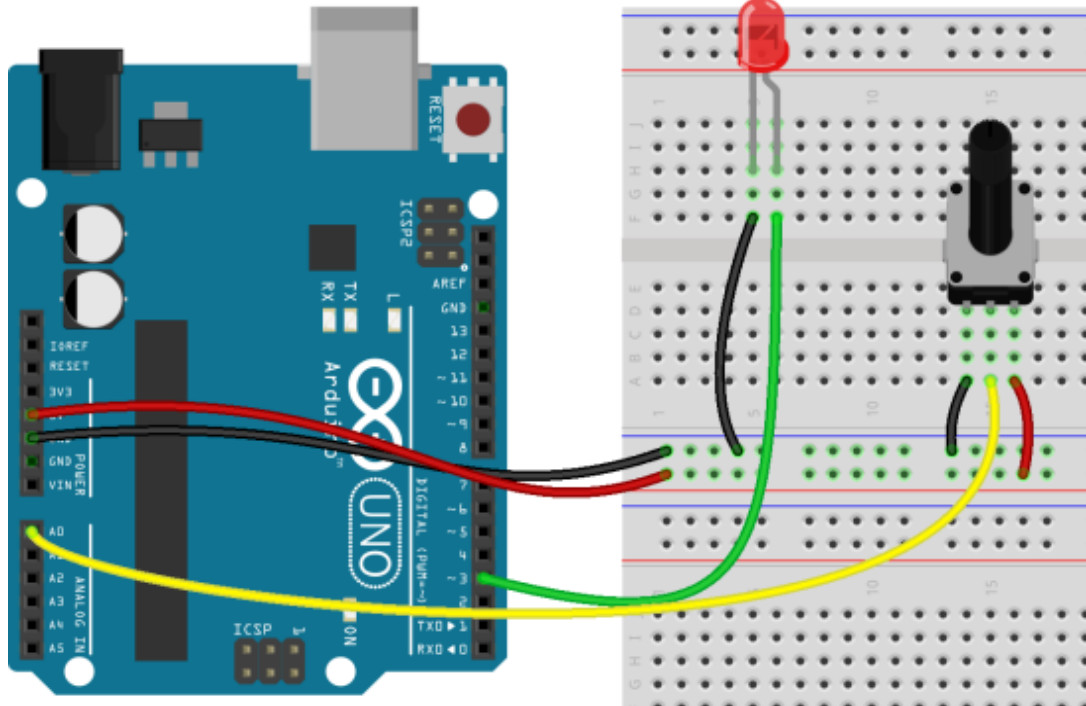


3x rezistor 220Ω alebo 330Ω

```
//Nastavíme piny pripojených LED diod  
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);  
}
```

Blikač riadený potenciometrom



Potenciometer: 100kOhm

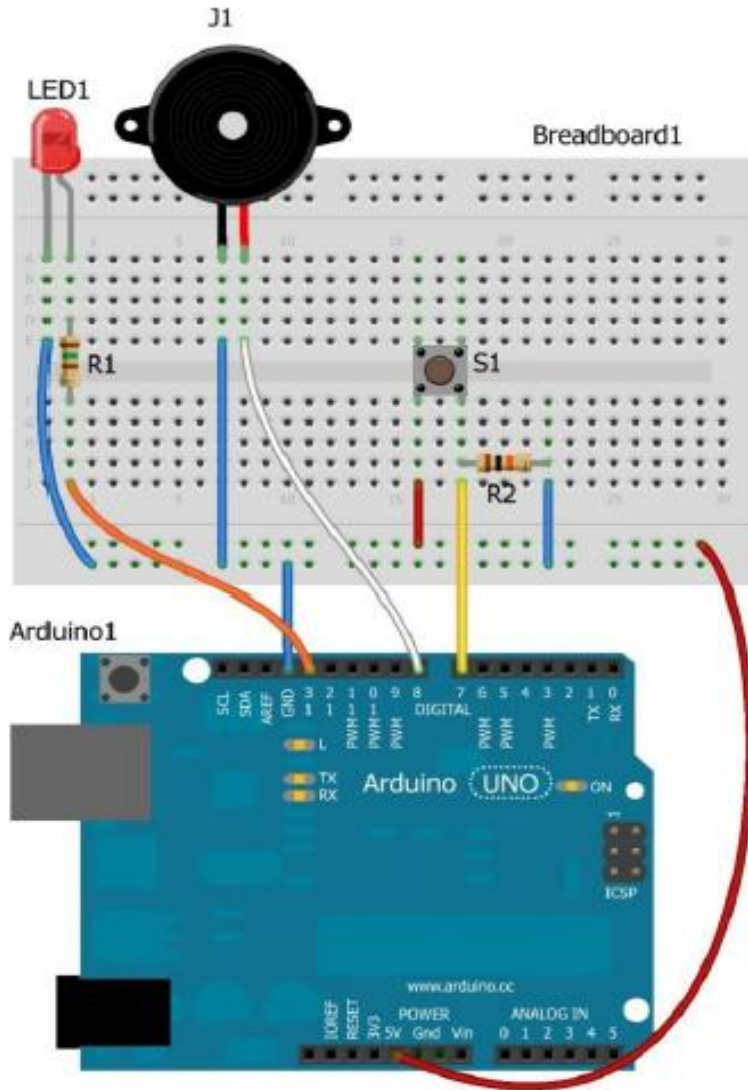
```
int potPin = A0; // číslo pinu pripojeného  
potenciometra  
int ledPin = 3; // číslo pinu pripojenej LED  
diody
```

```
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);  
}
```

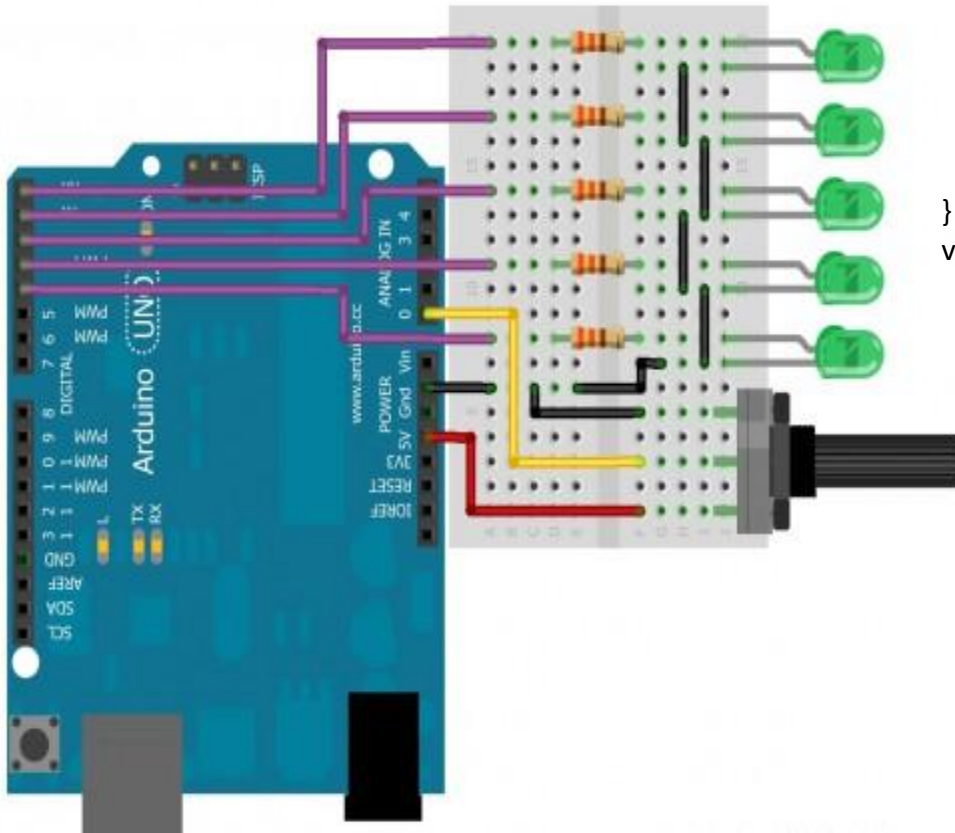
Generátor zvuku



R1= 150Ohm, R2= 10kOhm

```
// číslo pinu pre LED
int ledPin = 13;
// číslo pinu pre tlačidlo
int buttonPin = 7;
// číslo pinu pre piezo
int piezoPin = 8;
// premenná uchovávajúca stav tlačidla
int buttonState = 0;
void setup() {
// nastavenie pinu pre LEDku ako výstupný
pinMode(ledPin, OUTPUT);
// nastavenie pinu pre piezo ako výstupný
pinMode(piezoPin, OUTPUT);
// nastavenie pinu pre tlačidlo ako vstupný
pinMode(buttonPin, INPUT);
}
void loop(){
// načítame stav tlačidla
buttonState = digitalRead(buttonPin);
// skontrolujeme či je tlačidlo stlačené
// ak je, tzn. premenná buttonState má hodnotu HIGH
if (buttonState == HIGH) {
// zapneme LEDku
digitalWrite(ledPin, HIGH);
// zapneme generovanie tónu
tone(piezoPin, 1000, 100);
delay(100);
// ak nie je, tzn. premenná buttonState má hodnotu LOW
} else {
// vypneme LEDku
digitalWrite(ledPin, LOW);
// vypneme generovanie tónu
noTone(piezoPin); }
}
```

Čítanie polohy potenciometra

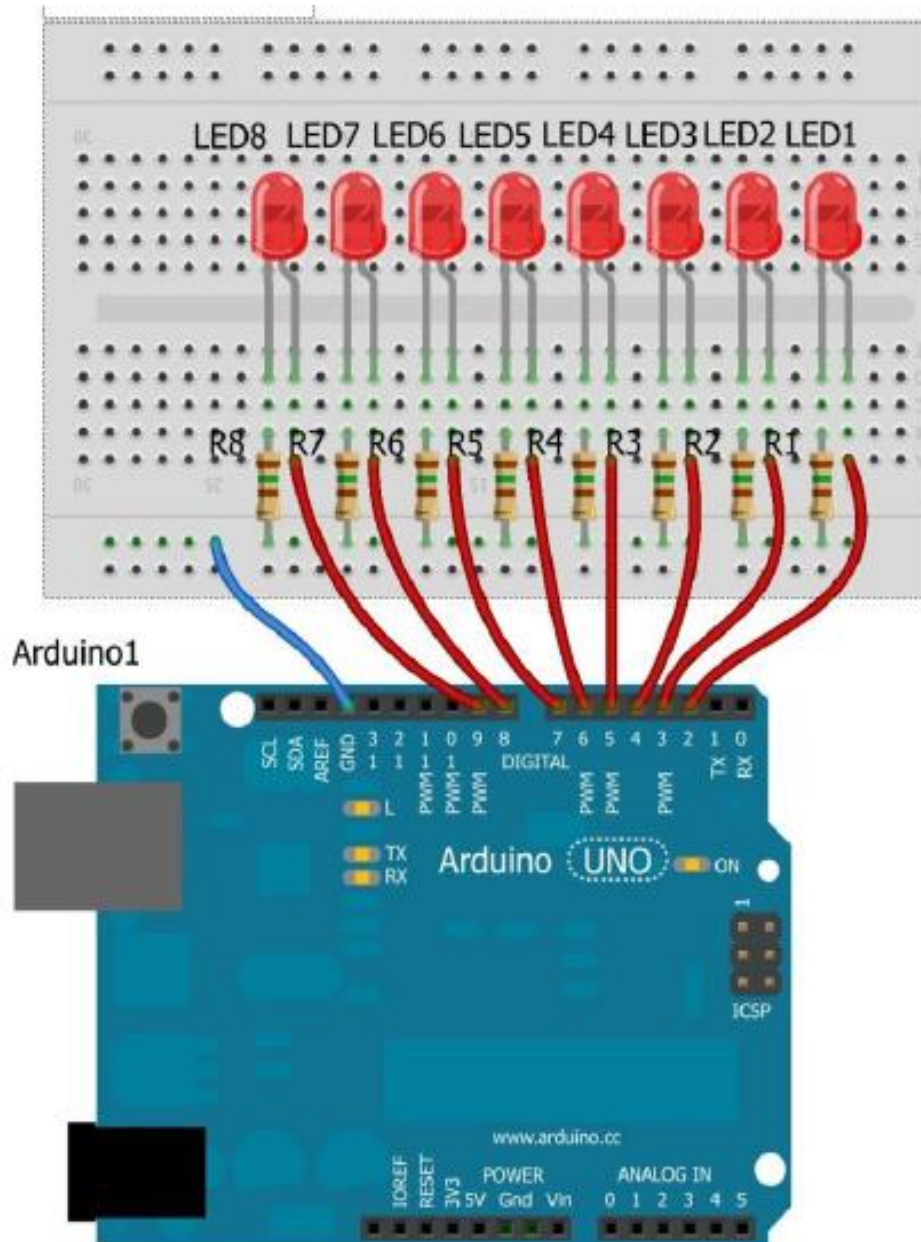


5x rezistor 220Ω alebo 330Ω

```
byte led[] = {1,2,3,4,5};  
//pole s pinmi pripojených LED diód  
byte pot = A0; //pin potenciometra pripojený na analógový vstup A0
```

```
int val;  
void setup() {  
  pinMode(led[0], OUTPUT);  
  pinMode(led[1], OUTPUT);  
  pinMode(led[2], OUTPUT);  
  pinMode(led[3], OUTPUT);  
  pinMode(led[4], OUTPUT);  
}  
void loop() {  
  val = analogRead(pot);  
  if(val > 800){  
    digitalWrite(led[0],HIGH);  
  }  
  else if(val > 600){  
    digitalWrite(led[1],HIGH);  
  }  
  else if(val > 400){  
    digitalWrite(led[2],HIGH);  
  }  
  else if(val > 200){  
    digitalWrite(led[3],HIGH);  
  }  
  else{  
    digitalWrite(led[4],HIGH);  
  }  
  delay(250);  
  digitalWrite(led[0],LOW);  
  digitalWrite(led[1],LOW);  
  digitalWrite(led[2],LOW);  
  digitalWrite(led[3],LOW);  
  digitalWrite(led[4],LOW);  
}
```

Knight rider efekt

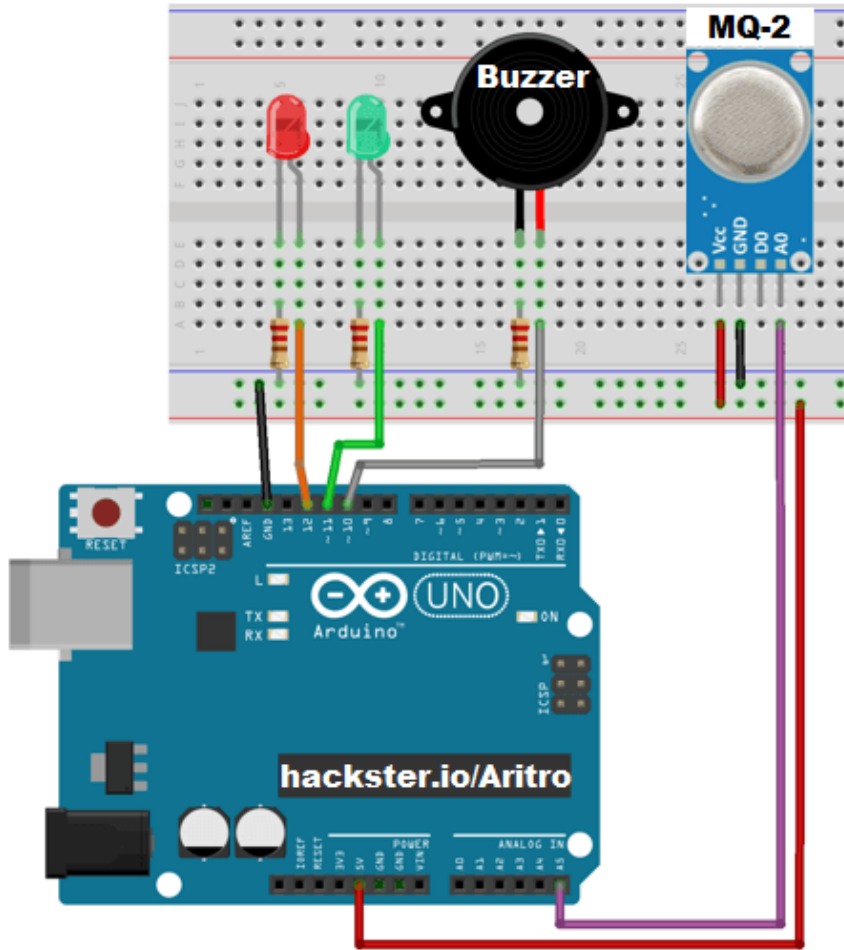


```
byte outputPins[] = {2, 3, 4, 5, 6, 7, 8, 9};
```

```
void setup()  
{  
  for(int i=0; i<8;i++)  
  {  
    pinMode(outputPins[i], OUTPUT);  
    digitalWrite(outputPins[i], LOW);  
  }  
}
```

```
void loop()  
{  
  for(int i=0; i<7;i++)  
  {  
    digitalWrite(outputPins[i], HIGH);  
    delay(75);  
    digitalWrite(outputPins[i], LOW);  
  }  
  for(int i=7; i>0;i--)  
  {  
    digitalWrite(outputPins[i], HIGH);  
    delay(75);  
    digitalWrite(outputPins[i], LOW);  
  }  
}
```

Detektor dymu a alkoholu MQ-4

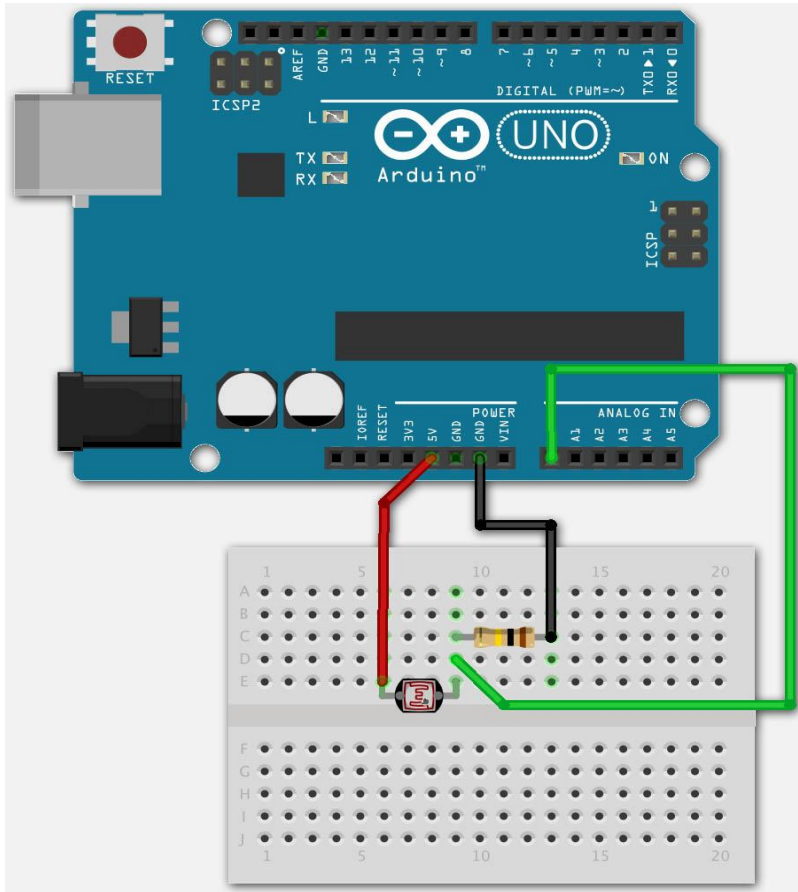


3 x rezistor 220 Ohm alebo 300 Ohm

```
int redLed = 12;
int greenLed = 11;
int buzzer = 10;
int smokeA0 = A5;
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);

  if (analogSensor > sensorThres)
  {
    digitalWrite(redLed, HIGH);
    digitalWrite(greenLed, LOW);
    tone(buzzer, 1000, 200);
  }
  else
  {
    digitalWrite(redLed, LOW);
    digitalWrite(greenLed, HIGH);
    noTone(buzzer);
  }
  delay(100);
}
```


Optický snímač



```
#define LDRpin A0
// pin where we connected the LDR and the resistor
int LDRValue = 0;
// result of reading the analog pin
int redLed = 12;
int greenLed = 11;
void setup() {
  Serial.begin(9600);
  // sets serial port for communication
  pinMode(redLed, OUTPUT);
  pinMode(greenLed, OUTPUT);
}
void loop() {
  LDRValue = analogRead(LDRpin);
  // read the value from the LDR
  Serial.println(LDRValue);
  // print the value to the serial port
  if (LDRValue > 1000)
  {
    digitalWrite(redLed, HIGH);
    digitalWrite(greenLed, LOW);
  }
  else
  {
    digitalWrite(redLed, LOW);
    digitalWrite(greenLed, HIGH);
  }
  delay(100);
  // wait a little
}
```

Zadanie:

1. Zapoj snímač vzdialenosti s dvoma LED. Testuj na dve vzdialenosti.
2. Zapoj snímač magnetického poľa. Pri detekcii rozsvieť LED.
3. Zapoj čidlo prekážky, pri detekcii rozsvieť LED.
4. Zapoj servomotor, otestuj polohy.
5. Zapoj detektor pohybu, pri detekcii rozsvieť LED.
6. Zapoj servomotor s potenciometrom.
7. Zapoj detektor ohňa, pri detekcii rozsvieť LED.
8. Zapoj snímač teploty s LED.
9. Zapoj tónový generátor s troma LED a troma tlačidlami.

Alternatívne úlohy:

1. Pripoj snímač vzdialenosti, zelenú LED a červenú LED, pri detekcii 10cm sa rozsvieti zelená LED, pri detekcii 20 cm sa rozsvieti červená LED.
2. K doske pripoj detektor ohňa, buzzer a LED, pri detekcii sa rozsvieti LED a zaznie akustický signál.
3. K doske pripoj čidlo prekážky, pri detekcii prekážky sa rozsvieti LED a zaznie tónový generátor.
4. Uprav program pre Knight rider efekt s 12 LED, pripoj k doske.
5. Pripoj čidlo prekážky a servo, pri detekcii prekážky sa servo otočí o 90 stupňov. Servo sa vráti do pôvodnej polohy o 5 sekúnd.
6. Pripoj magnetické čidlo a servo, pri detekcii sa servo otočí o 90 stupňov. Servo sa vráti do pôvodnej polohy o 5 sekúnd.

Zadanie 19.10.2017:

1. Andrej - Michal: detektor dymu a alkoholu, snímač teploty a vlhkosti
2. Marek – Jakub: servomotor a čidlo prekážky, magnetický spínač s LED
3. Roman – Matej: servomotor a čidlo prekážky, detektor pohybu s LED
4. Filip – Dominik: servomotor a čidlo prekážky, servomotor a potenciometer

Andrej – 2 interaktívne LED s dvoma tlačidlami + tónový generátor

Michal – spínač prekážky so servomotorom

Marek – 2 potenciometre s dvoma servomotormi

Jakub – detektor dymu s LED a servomotor

Roman – magnetický spínač s LED a so servomotorom

Matej – snímač teploty a vlhkosti s LED pri určitej teplote

Filip – čidlo prekážky a servomotor

Dominik - magnetický spínač s LED a tónovým generátorom

Zadanie 20.10.2017:

Dominik, Michal, Jakub, Roman – čidlo prekážky so servomotorom

Andrej, Marek – generátor s 1 LED, 3 Button a s 1 buzzerom

Filip – magnetický spínač so servomotorom

Matej M. – potenciometer s 5 LED, čítanie potenciometra

Matej O. - 2 interaktívne LED s dvoma tlačidlami

Programovanie Arduina, programové vybavenie mikroprocesora:

- **definičná časť** – definícia konštánt a premenných (napr: `int led = 13;`)
- **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  
sestnactkove soustave)  
• 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

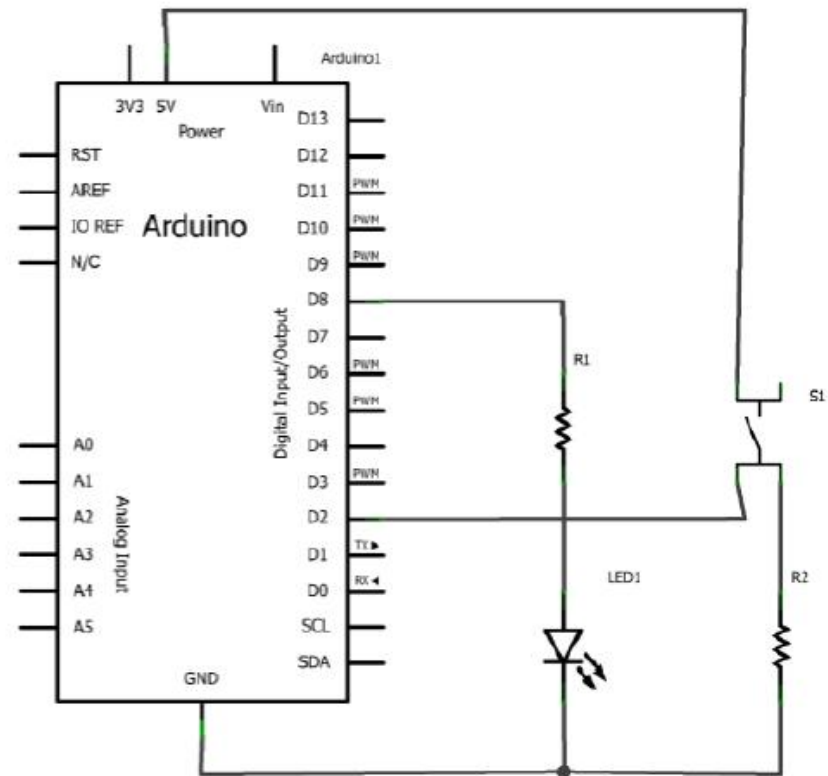
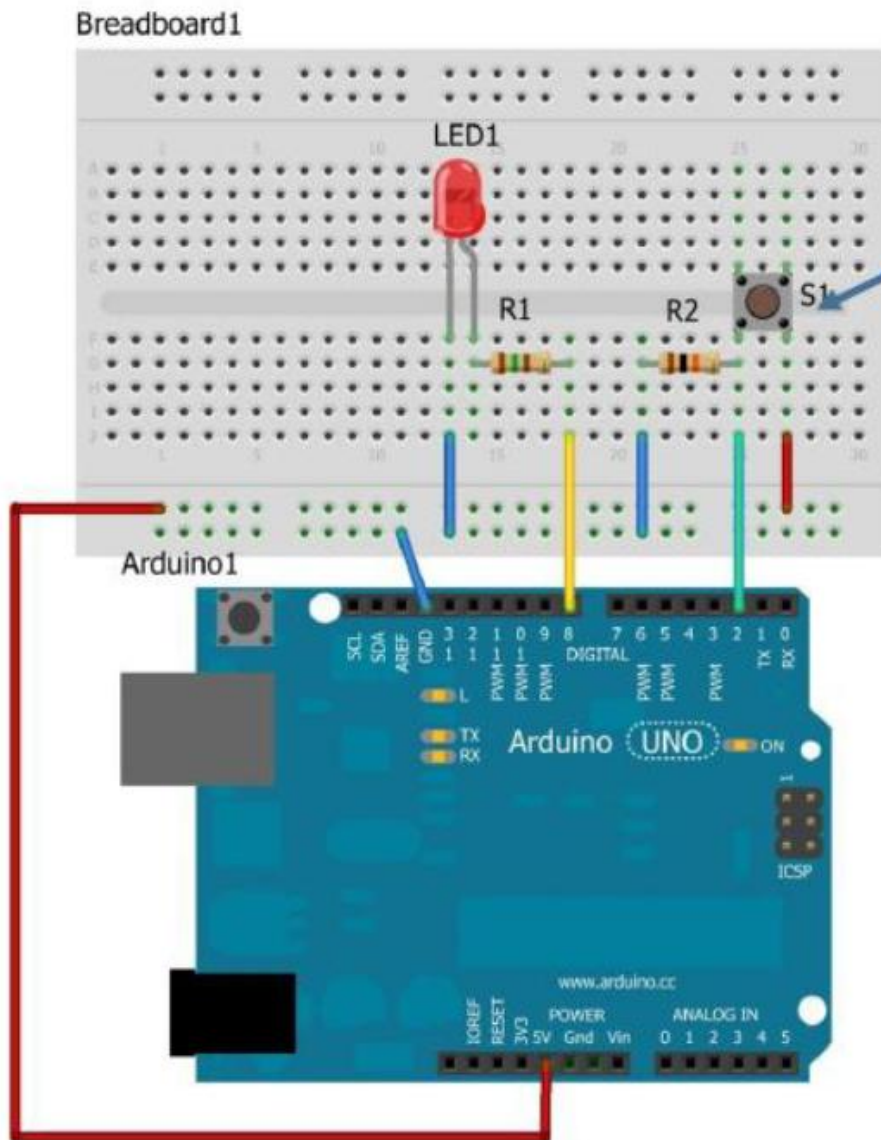
Generovanie náhodných čísiel:

- random(min, max)

Generovanie tónov:

- tone()
- notone()

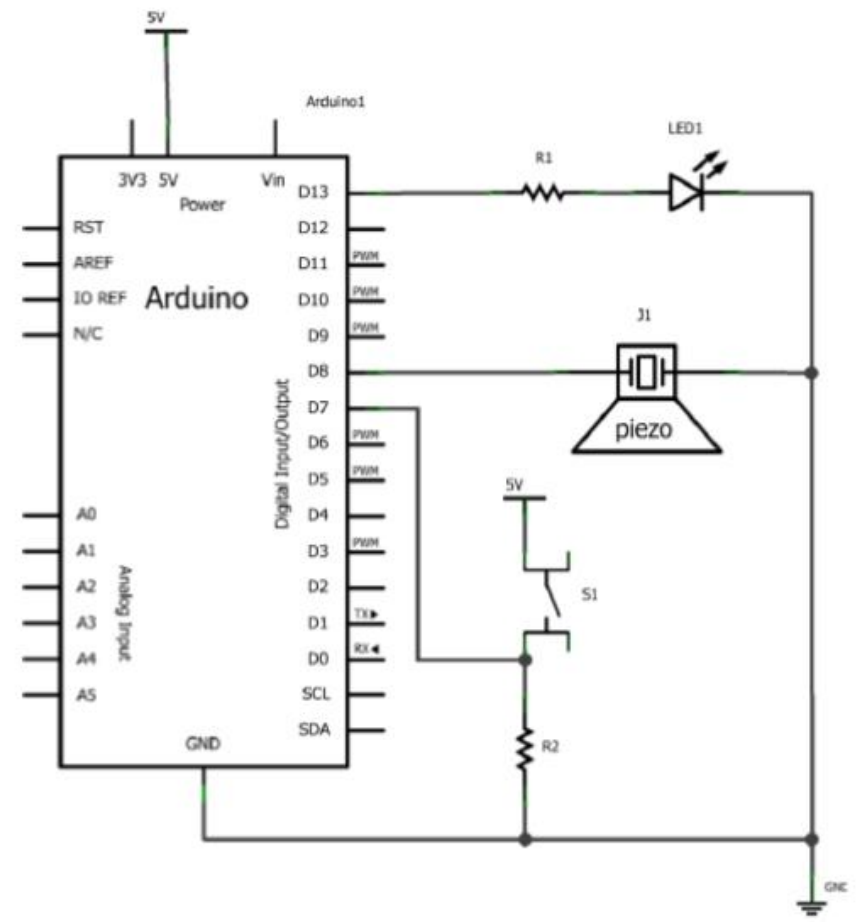
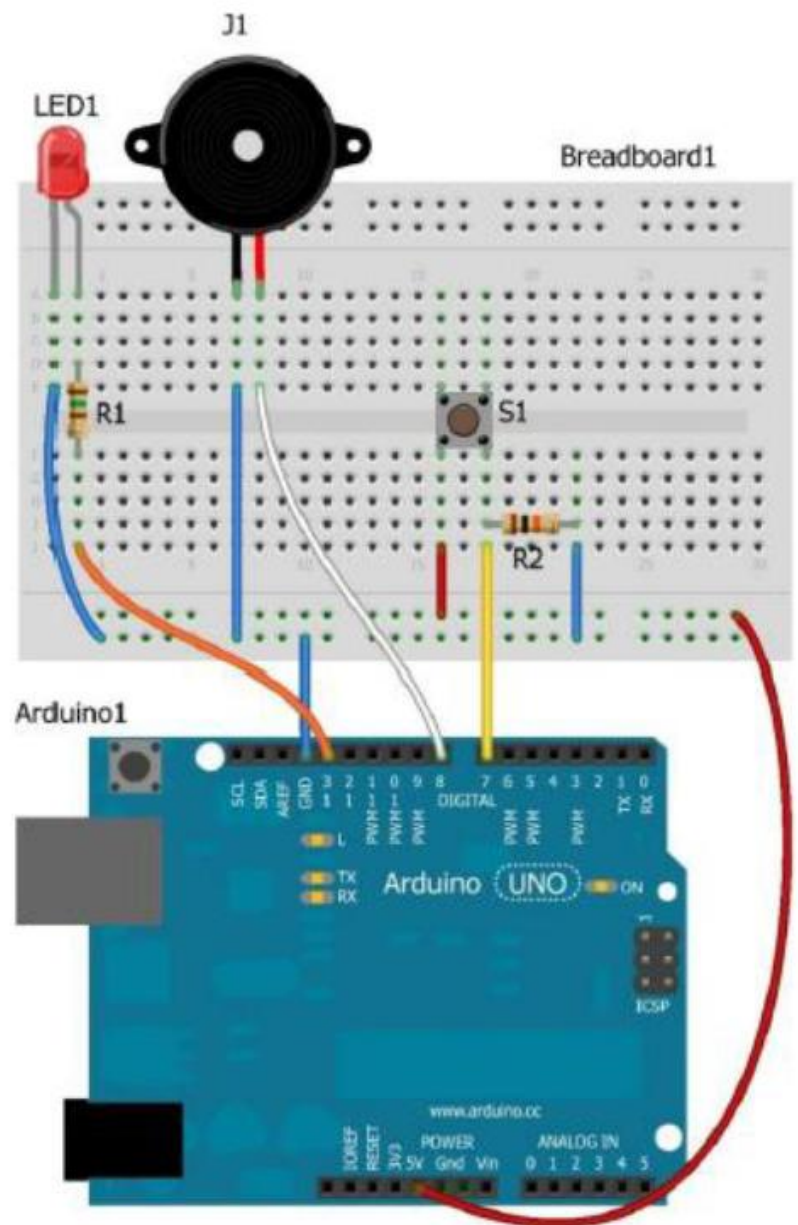
Uprav schému, kontaktnú plochu a program pre 2 interaktívne LED



Uprav program pre 2LED a 2 button

```
//číslo pinu pre tlačítka
int buttonPin = 2;
//číslo pinu pre LEDku
int ledPin = 8;
//premenná uchovávajúca stav tlačítka
int buttonState = 0;
void setup() {
// nastavenie pinu pre LEDku ako výstupný
pinMode(ledPin, OUTPUT);
// nastavenie pinu pre tlačítka ako vstupný
pinMode(buttonPin, INPUT);
}
void loop(){
// načítame stav tlačítka
buttonState = digitalRead(buttonPin);
// skontrolujeme či je tlačítka stlačené alebo nie
// ak je, tzn. premenná buttonState má hodnotu HIGH
if (buttonState == HIGH) {
digitalWrite(ledPin, HIGH); // zapneme LEDku
} else {
// ak tlačítka nie je zopnuté (stlačené), LEDku vypneme
digitalWrite(ledPin, LOW);
}
}
```

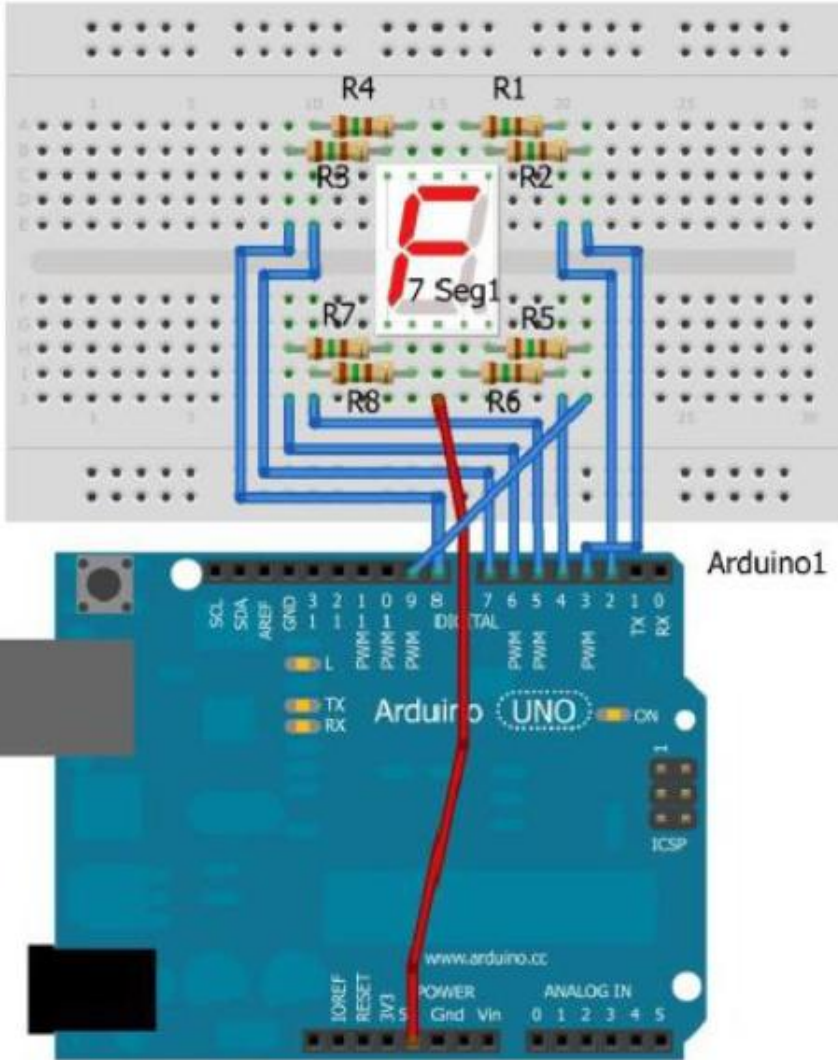

Uprav schému, kontaktnú plochu a program pre 3 tlačidlá, 1 LED a 3tóny, 1x buzzer



Uprav program pre 1LED, 3BUTTON, 1BUZZER

```
// číslo pinu pre LED
int ledPin = 13;
// číslo pinu pre tlačidlo
int buttonPin = 7;
// číslo pinu pre piezo
int piezoPin = 8;
// premenná uchovávajúca stav tlačidla
int buttonState = 0;
void setup() {
// nastavenie pinu pre LEDku ako výstupný
pinMode(ledPin, OUTPUT);
// nastavenie pinu pre piezo ako výstupný
pinMode(piezoPin, OUTPUT);
// nastavenie pinu pre tlačidlo ako vstupný
pinMode(buttonPin, INPUT);
}
void loop(){
// načítame stav tlačidla
buttonState = digitalRead(buttonPin);
// skontrolujeme či je tlačidlo stlačené
// ak je, tzn. premenná buttonState má hodnotu HIGH
if (buttonState == HIGH) {
// zapneme LEDku
digitalWrite(ledPin, HIGH);
// zapneme generovanie tónu
tone(piezoPin, 1000, 100);
delay(100);
// ak nie je, tzn. premenná buttonState má hodnotu LOW
} else {
// vypneme LEDku
digitalWrite(ledPin, LOW);
// vypneme generovanie tónu
noTone(piezoPin); }
}
```

LED displej



```

byte segmentPins[] =
{2/*A*/,/3/*B*/,/4/*C*/,/5/*D*/,/6/*E*/,/7/*F*/,/8/*G*/,/9/*BODKA*/};
byte segmentDigits[10][8] = {
//{ A , B , C , D , E , F , G , BODKA }
{LOW, LOW, LOW, LOW, LOW, LOW, HIGH, HIGH}, // číslica 0
{HIGH, LOW, LOW, HIGH, HIGH, HIGH, HIGH, HIGH}, // číslica 1
{LOW, LOW, HIGH, LOW, LOW, HIGH, LOW, HIGH}, // číslica 2
{LOW, LOW, LOW, LOW, HIGH, HIGH, LOW, HIGH}, // číslica 3
{HIGH, LOW, LOW, HIGH, HIGH, LOW, LOW, HIGH}, // číslica 4
{LOW, HIGH, LOW, LOW, HIGH, LOW, LOW, HIGH}, // číslica 5
{LOW, HIGH, LOW, LOW, LOW, LOW, LOW, HIGH}, // číslica 6
{LOW, LOW, LOW, HIGH, HIGH, HIGH, HIGH, HIGH}, // číslica 7
{LOW, LOW, LOW, LOW, LOW, LOW, LOW, HIGH}, // číslica 8
{LOW, LOW, LOW, LOW, HIGH, LOW, LOW, HIGH} // číslica 9
};

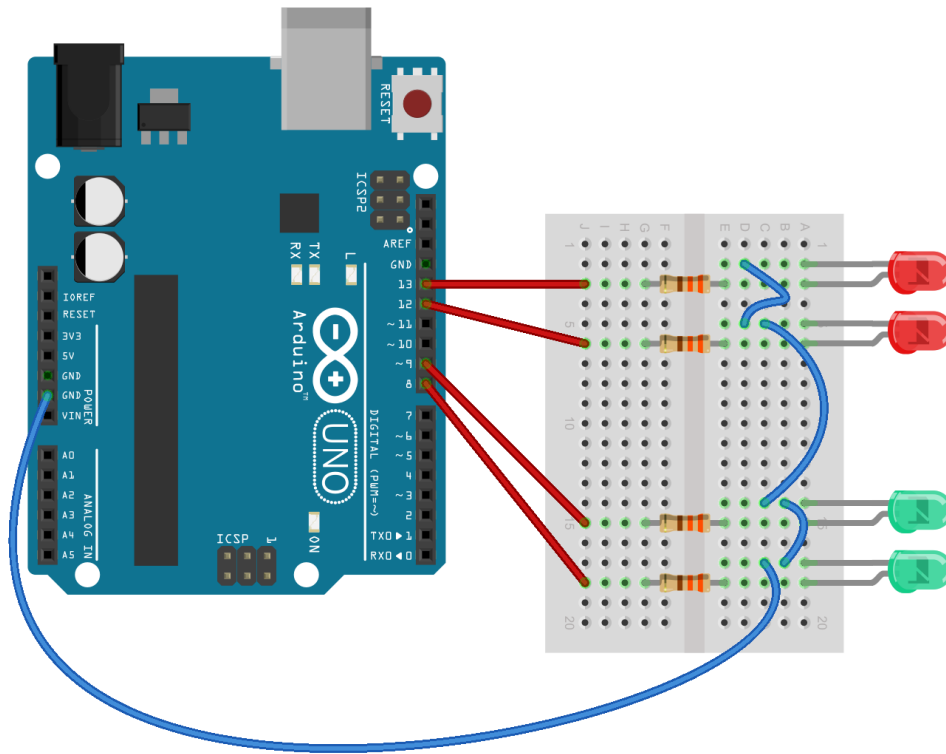
void segmentDisplay(int digit);
void setup(){
for(int i=0;i<8;i++){
pinMode(segmentPins[i], OUTPUT);
}
for(int i=0;i<8;i++){
digitalWrite(segmentPins[i], HIGH);
}
}

void loop(){
for(int n=0; n<10; n++){
segmentDisplay(n);
delay(500);
}
for(int m=9; m>=0; m--){
segmentDisplay(m);
delay(500);
}
}

void segmentDisplay(int digit){
for(int s=0; s<8; s++){
digitalWrite(segmentPins[s], segmentDigits[digit][s]);
}
}

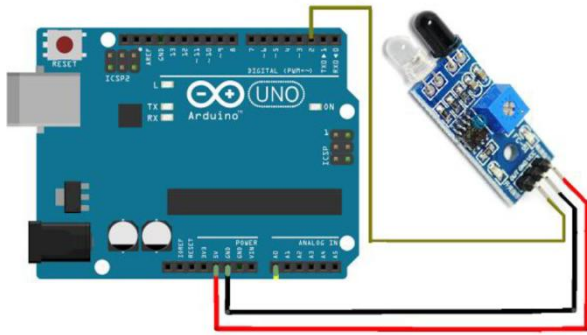
```

Blikač so 4 led



```
const int cas = 100;
void setup()
{
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
}
void loop()
{
  digitalWrite(9, HIGH);
  digitalWrite(13, HIGH);
  delay(cas);
  digitalWrite(9, LOW);
  digitalWrite(13, LOW);
  delay(cas);
  digitalWrite(8, HIGH);
  digitalWrite(12, HIGH);
  delay(cas);
  digitalWrite(8, LOW);
  digitalWrite(12, LOW);
  delay(cas);
}
```

Detektor překážky:



```
/*
```

```
IR Proximity Sensor interface code
```

```
Turns on an LED on when obstacle is detected, else off.
```

```
blog.circuits4you.com 2016
```

```
*/
```

```
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
```

```
{
```

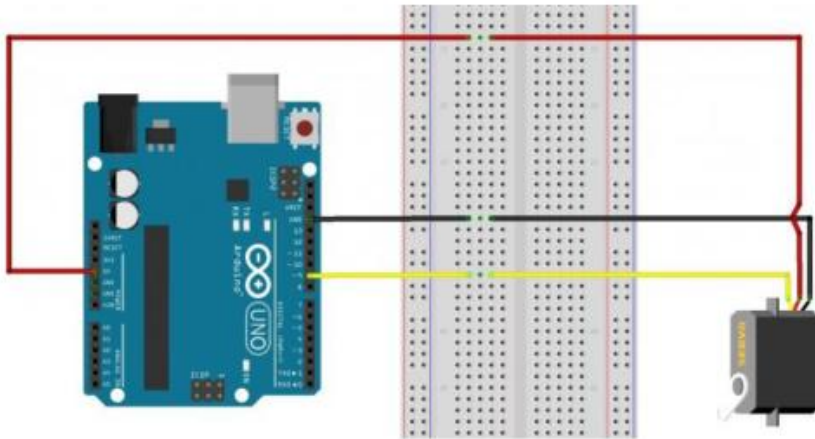
```
digitalWrite(13, LOW); // set the LED off
```

```
}
```

```
delay(100); // wait for a second
```

```
}
```

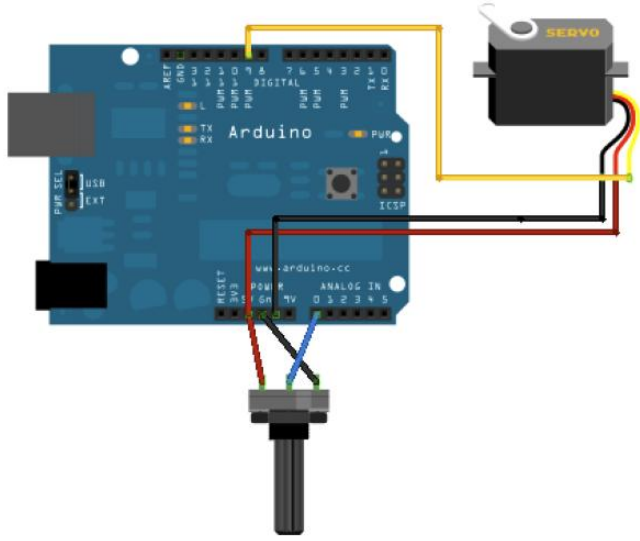
Servomotor



```
#include <Servo.h>
int servoPin = 9;
Servo servo;
int servoAngle = 0; // pozícia serva v uhloch
void setup()
{
  Serial.begin(9600);
  servo.attach(servoPin);
}

void loop()
{
  //ovládame smer servopohonu a polohu motora
  servo.write(45); // otočí servo doľava na 45 stupňov
  delay(1000); // čaká 1 sekundu
  servo.write(90); // otočí servo späť na 90 stupňov
  delay(1000); // čaká 1 sekundu
  servo.write(135); // otočí servo doprava na 135 stupňov
  delay(1000); // čaká 1 sekundu
  servo.write(90); // otočí servo späť na 90 stupňov
  delay(1000);
}
```

Ovládanie servomotora potenciometrom



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

```
Servo myservo;
```

```
// vytvorte servoobjekt na ovládanie servopohonu
```

```
int potpin = 0;
```

```
// analogový kolík určený na pripojenie potenciometra
```

```
int val;
```

```
// premenná na čítanie hodnoty z analógového kolíka
```

```
void setup() {
```

```
myservo.attach(9);
```

```
// pripojí servo na kolík 9 k servoventilu
```

```
}
```

```
void loop()
```

```
{
```

```
val = analogRead(potpin); // číta hodnotu potenciometra (medzi 0 a 1023)
```

```
val = map(val, 0, 1023, 0, 180);
```

```
// upravme ho tak, aby sme ho používali so servom (hodnota medzi 0 a 180)
```

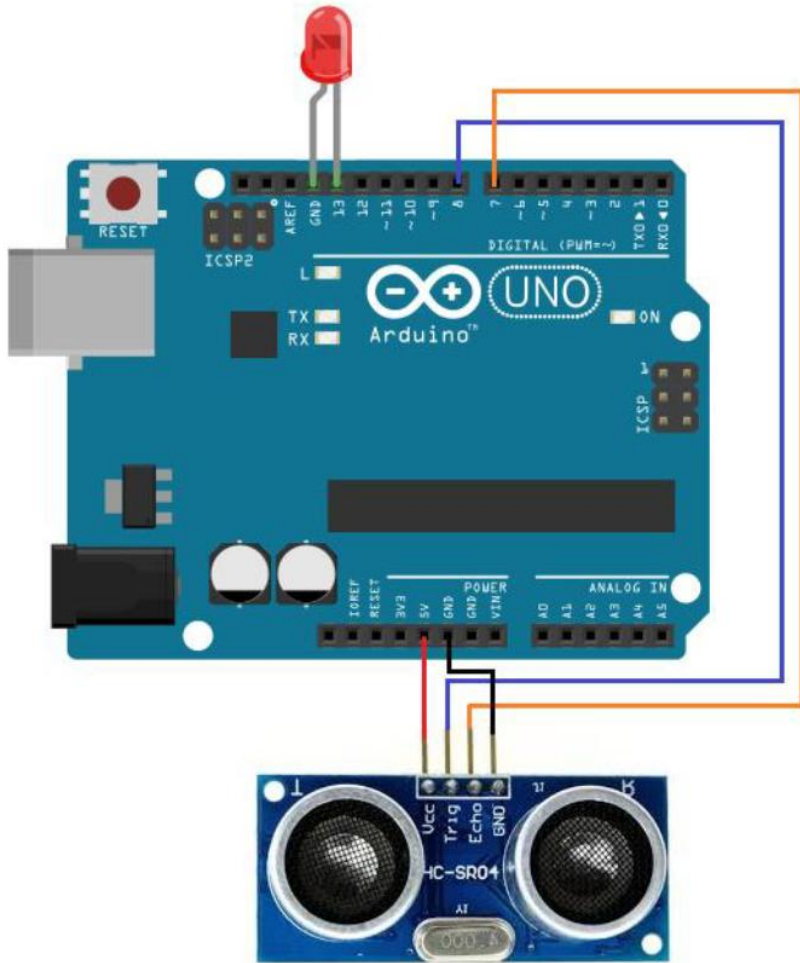
```
myservo.write(val);
```

```
// nastaví polohu servopohonu podľa stupnice
```

```
delay(15); // čaká na servo, aby sa tam dostal
```

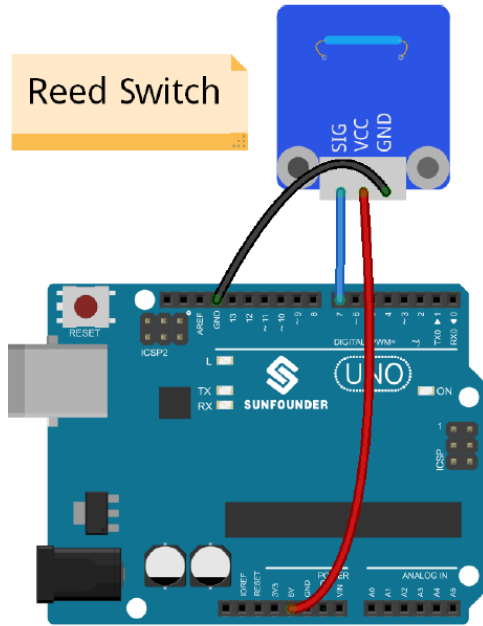
```
}
```

Merač vzdialenosti



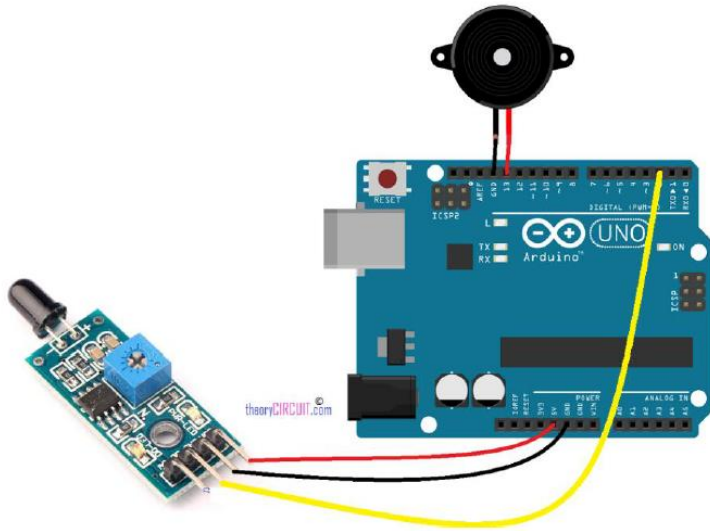
```
const int signalPin= 7;
const int triggerPin= 8;
const int LEDPin= 13;
const int LEDPin2= 12;
long signal, inches, centimeters;
void setup() {
  Serial.begin(9600); }
void loop() {
  pinMode (signalPin, INPUT);
  pinMode (triggerPin, OUTPUT);
  pinMode (LEDPin, OUTPUT);
  pinMode (LEDPin2, OUTPUT);
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  signal= pulseIn(signalPin, HIGH);
  inches= signal/148;
  centimeters= inches * 2.54;
  Serial.print(inches);
  Serial.print("in, ");
  Serial.print(centimeters);
  Serial.print("cm");
  Serial.println();
  delay(500);
  if (inches < 5) {
    digitalWrite(LEDPin, HIGH); }
  else {
    digitalWrite(LEDPin, LOW); }
  if (inches < 10) {
    digitalWrite(LEDPin2, HIGH); }
  else {
    digitalWrite(LEDPin2, LOW); }
  delay(100); }
```


Magnetický spínač



```
const int digitalInPin = 7;
// jazýčkový spínač pripojiť k pin7
const int ledPin = 13;
// pin13 vstavaný pre LED
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 so stat
  if (stat == HIGH)
    // ak vysoko
    {
      digitalWrite (ledPin, LOW);
      // potom vypnúť viedla
    }
  else
    // iný
    {
      digitalWrite (ledPin, HIGH);
      // zapnutie LED
    }
}
```

Detektor ohňa



```
int Buzzer = 13; // pin pre buzzer
int FlamePin = 2; // pin pre detektor
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);
  }
}
```

Zadania projektu ARDUINO:

1. Zapoj čidlo prekážky s indikáciou LED červená pri detekovaní prekážky, keď nedekuje prekážku indikuj LED zelená.
2. Zapoj detektor prekážky, pri detekovaní prekážky otoč servo na polohu 90°, keď nedekuje prekážku, otoč servo na 0°.
3. Zapoj Knight rider efekt, pri zasvietení LED spusti tónový generátor inej frekvencie.
4. Zapoj detektor vzdialenosti – pri detekovaní viac ako 20 cm vzdialenosti zasviet zelenú LED, pri detekovaní 30 cm vzdialenosti zasviet červenú LED.
5. Zapoj 3 interaktívne LED s tlačidlami, pri stlačení tlačidla tónový generátor generuje inú frekvenciu.
6. Zapoj snímač teploty s detekciou nad 20°C červenou LED.
7. Zapoj 2 servomotory regulované s dvoma potenciometrami.
8. Naprogramuj semafor – červená 3 s, oranžová s červenou 1s, zelená s oranžovou 2s.
9. Zapoj detektor ohňa s indikáciou ohňa s červenou LED a tónovým buzzerom.
10. Zapoj štyri LED, 1. bude svietiť 2 sekundy, 2. bude svietiť 3 sekundy, 3. bude svietiť 4 sekundy, 4. bude svietiť 5 sekúnd. Vždy bude svietiť iba 1 LED.
11. Zapoj 7 segmentový displej a 2 LED (červená, zelená). Červená bude svietiť pri nepárnom čísle, zelená bude svietiť pri párnom čísle.