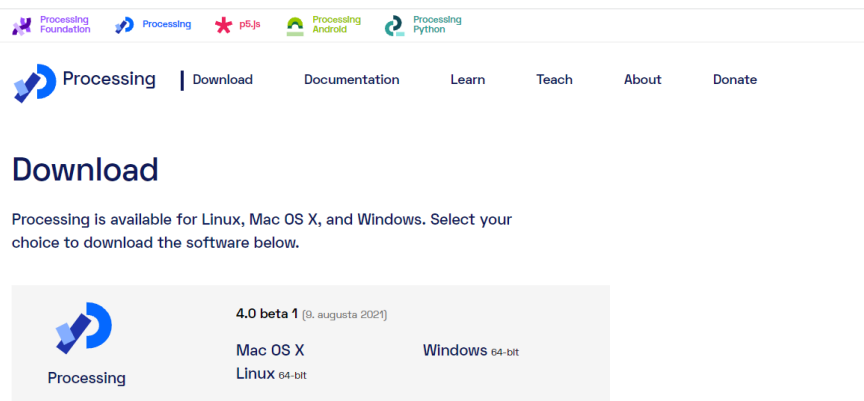


Arduino a Processing IDE

1. Program Processing

Program si stiahneme odtiaľto: <https://processing.org/download>



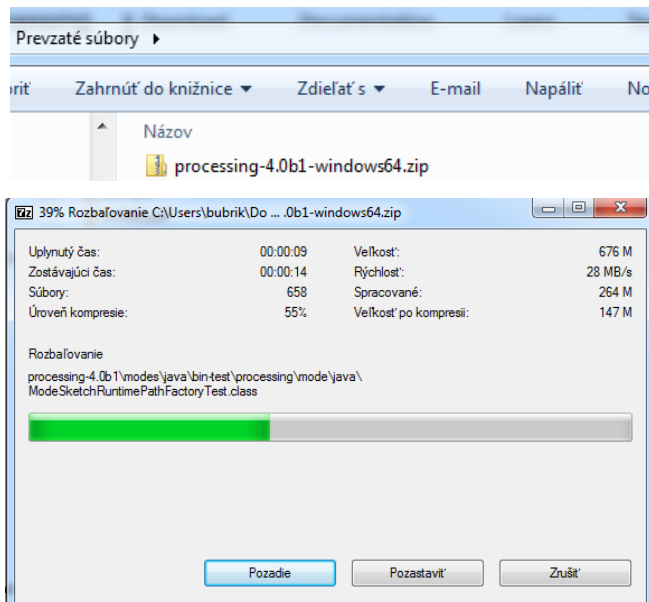
[GitHub](#) [Report Bugs](#) [Wiki](#) [Supported Platforms](#)

Read about the changes in 4.0. The list of revisions covers the differences between releases in detail.

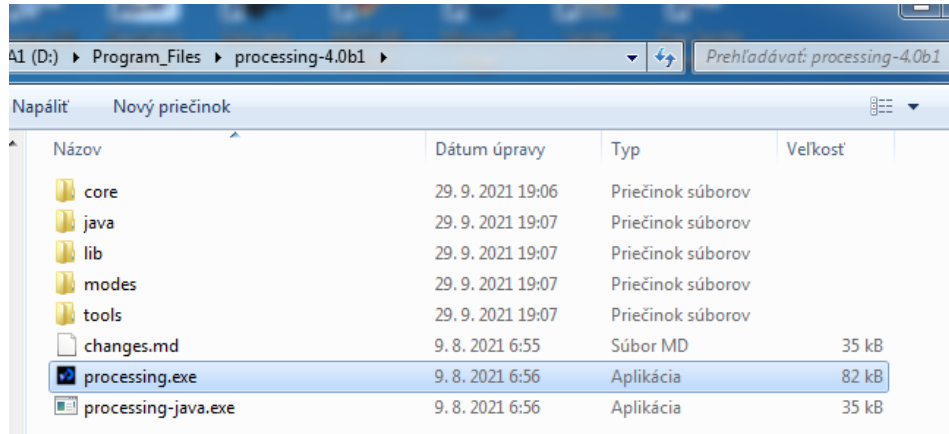
Stable Releases

4.0 beta 1	(9. augusta 2021)	Mac OS X / Windows 64-bit / Linux 64-bit
3.5.4	(17. januára 2020)	Mac OS X / Linux 64-bit / Windows 64-bit / Windows 32-bit
2.2.1	(31. júla 2014)	Linux 32-bit / Linux 64-bit / Windows 32-bit / Windows 64-bit / Mac OS X

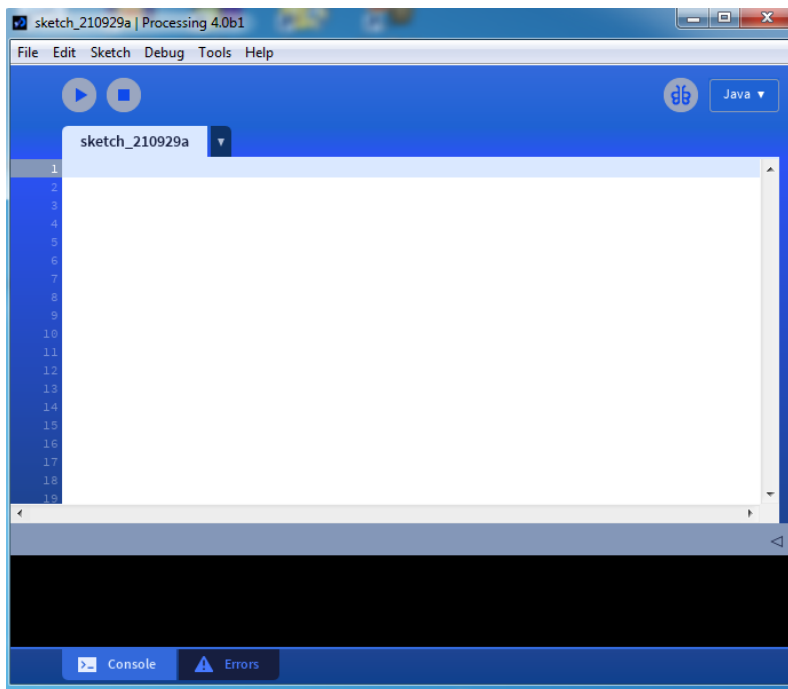
Stiahnutý zip súbor extrahujeme:



Spustíme túto aplikáciu:

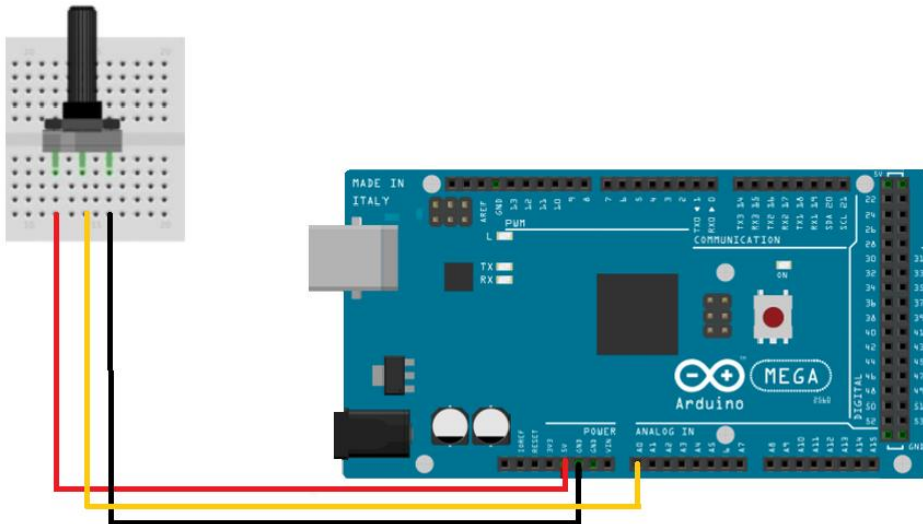


Spustená aplikácia:



2. Schéma zapojenia s Arduino

Na vyskúšanie zapojíme s ľubovoľným Arduino LED a potenciometer:



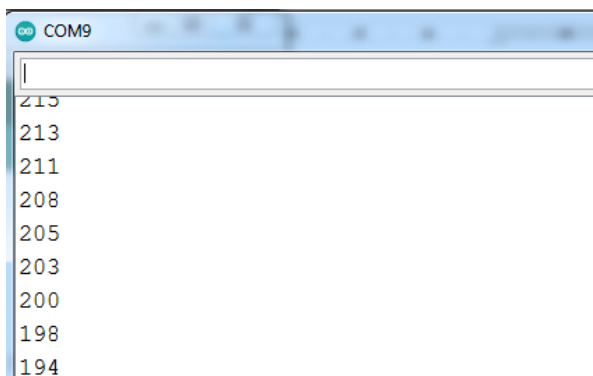
3. Program pre Arduino

```
int led_pin = 7; // LED je na pine 7
int pot_pin = A0; // potenciometer je na pine A0
int pot_output; // premenná na uloženie hodnoty potenciometra

void setup ( ) {
  pinMode(led_pin, OUTPUT); // pin, na ktorom je LED, bude výstupný
  Serial.begin(9600); // zapnutie sériovej komunikácie na 9600 baudov
}

void loop ( ) {
  pot_output = analogRead (pot_pin); // prečítanie analógovej hodnoty z potenciometra
  int mapped_output = map (pot_output, 0, 1023, 0, 255); // hodnota z potenciometra je v rozsahu
  0..1023 a potrebujeme ju namapovať do rozsahu 0..255
  Serial.println (mapped_output); // odoslanie hodnoty na sériový port
  if (Serial.available ( ) > 0) { // kontrola, či Processing IDE poslal na sériový port nejakú
  hodnotu
    char state = Serial.read ( ); // prečítanie hodnoty na sériovom porte
    if(state == '1') // pre hodnotu 1 zapneme LED
    {
      digitalWrite (led_pin, HIGH);
    }
    if (state == '0') { // pre hodnotu 0 vypneme LED
      digitalWrite (led_pin, LOW);
    }
  }
  delay(50);
}
```

Výstup na sériovom porte (hodnoty potenciometra namapované do rozsahu 0-255):



```
COM9
215
213
211
208
205
203
200
198
194
```

4. Program pre Processing IDE

```
import processing.serial.*; // import knižnice pre sériovú komunikáciu s Arduino
Serial myPort; // inicializácia premennej pre sériovú komunikáciu
float background_color; // premenná pre farbu pozadia

void setup ( ) {
  size (500, 500); // veľkosť okna
  myPort = new Serial (this, "COM3", 9600); // nastavenie COM portu, na ktorom máme pripojené
  Arduino
  myPort.bufferUntil ( '\n' ); // Prijem dát z Arduino
}

void serialEvent (Serial myPort) {
  background_color = float (myPort.readStringUntil ( '\n' ) );
  // zmena farby pozadia podľa prijatej hodnoty zo sériového portu
}

void draw ( ) {
  background ( 150, 50, background_color ); // úvodná farba pozadia po otvorení sériového okna
  if ( mousePressed && ( mouseButton == LEFT ) ) { // ak je stlačené ľavé tlačidlo myši
    myPort.write ( '1' ); // pošli 1 do Arduino
  }
  if ( mousePressed && ( mouseButton == RIGHT ) ) { // ak je stlačené pravé tlačidlo myši
    myPort.write ( '0' ); // pošli 0 do Arduino
  }
}
```

5. Ako to spolupracuje?

Arduino IDE a Processing IDE budú medzi sebou komunikovať prostredníctvom sériovej komunikácie. Processing IDE má sériovú knižnicu, ktorá uľahčuje komunikáciu s Arduino. Keď pohneme gombíkom potenciometra, Arduino odošle namapovanú hodnotu z <0; 255> na sériový port a teda aj do Processing IDE. Processing IDE potom zmení farbu sériového okna podľa tejto hodnoty (čiže podľa pohybu gombíka potenciometra).

Podobne, keď stlačíme tlačidlo myši v sériovom okne Processing IDE, Processing IDE odošle '1' alebo '0' na sériový port (1=ľavé, 0=pravé). V Arduino programe IDE je použitý príkaz `Serial.read ()`, ktorý túto hodnotu zo sériového portu prečíta a potom zapne alebo vypne LED podľa stlačeného tlačidla.

Komentár od [b1]:

V programe pre Processing IDE treba nastaviť sériový port COM, na ktorom máme pripojené Arduino. Monitor sériového portu z Arduino IDE musíme zavrieť (inak Processing IDE hlási chybu).

