**TERMINOLÓGIA**

Mikrokontrolér je jednočipový mikropočítač. Je to súčiastka (integrovaný obvod), ktorá sa dá naprogramovať, lebo obsahuje procesor, pamäte, atď.

Mikroprocesor je ...

Mikropočítač je ...

Vývojové prostredie (IDE) je program na pohodlné písanie programu (editor + kompilátor)

Vývojová doska je ...

Vývojový kit je ...

Arduino UNO je vývojová doska, ktorá obsahuje mikrokontrolér ATmega328, ktorý sa môže programovať vo vývojovom prostredí ARDUINO IDE. Existuje viacero modelov Arduino.

Na simuláciu využívame WOKWI.COM. Ide o klavdovú službu, v ktorej môžeme mať uložené projekty, vieme v nej simulovať rôzne vývojové dosky a rôzne súčiastky.

Základný predpis programu pre Arduino obsahuje dve funkcie:

**setup()** – kód v nej sa vykoná raz an začiatku

**loop()** – kód v nej sa vykonáva opakovane po funkcii setup

Ďalšie funkcie:

**pinmode(cisloPinu, INPUTaleboOUTPUT)** – nastaví pin ako VSTUP alebo VÝSTUP

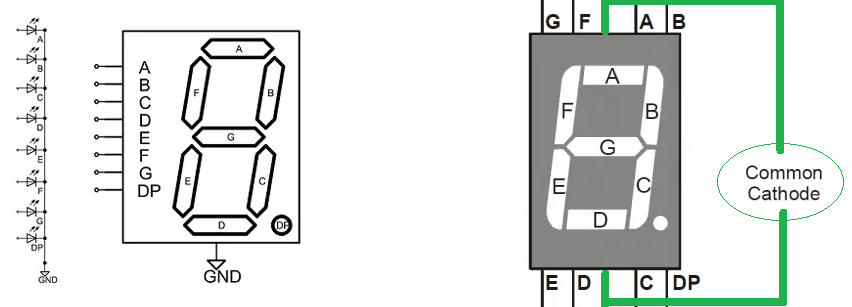
**digialWrite(cisloPinu, HIGHaleboLOW)** – prepojí pin s VCC alebo GND

**delay(CAS)** – pozastaví činnosť (v milisekundách)

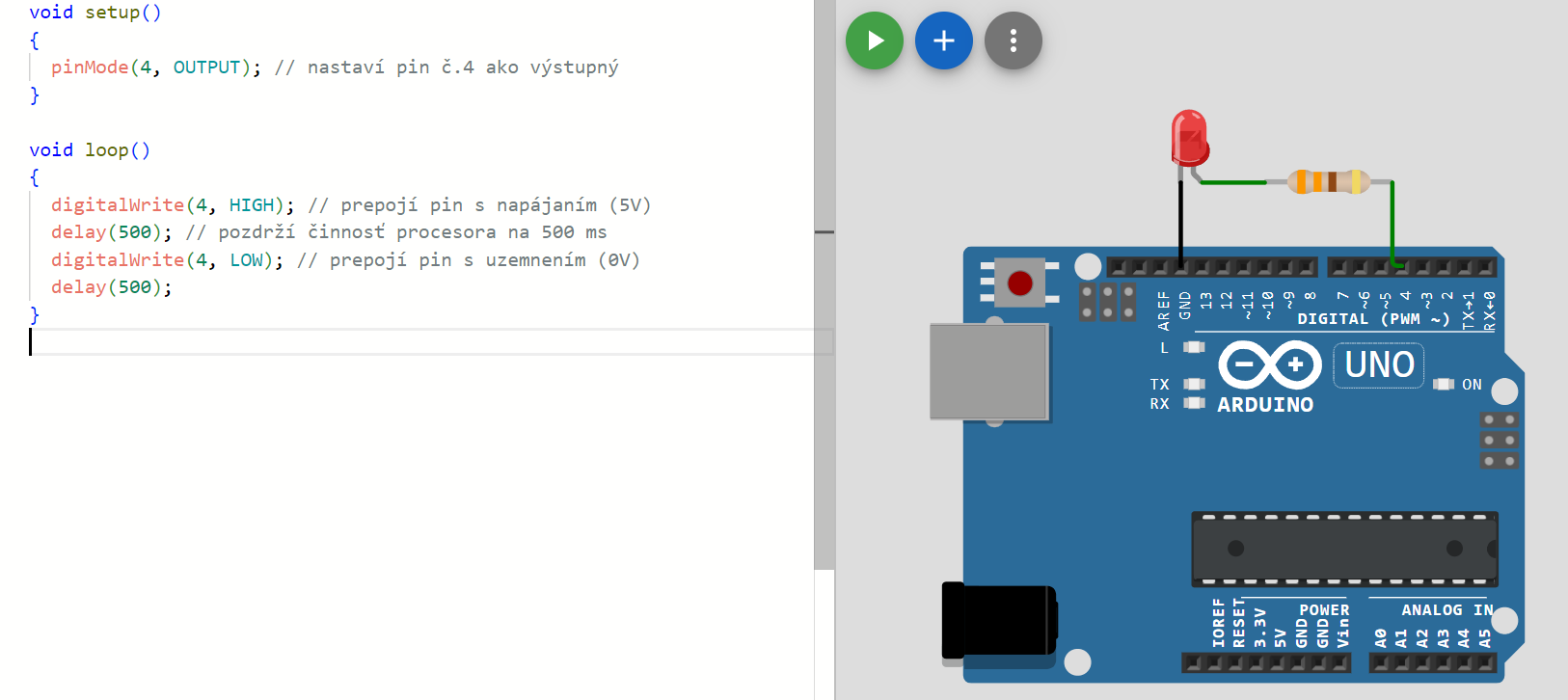
Aby LED dióda svietila, treba prepojiť anódu s kladným zdrojom (VCC / 5V) a katódu s uzemnením (GND / 0V).

RGB LED dióda dokáže vysvietiť rôzne farby kombináciou troch pinov. Nedokáže vysvietiť tmavé farby. Pozor, či je typu „spoločná anóda“ alebo „spoločná katóda“. Pri spoločnej anóde pripájame na COM kladný zdroj napätia VCC a uzemnením svietime (LOW pre zasvietenie).

7 segmentová dióda je v súčiastka, ktorá ma v sebe umiestnených 7 LED diód (občas s ôsmou desatinou bodkou). Môže byť so spoločnou anódou alebo katódou. Označenie pinov:



Príklad na blikanie:



ÚLOHA: príklad na blikanie semafóru

Ďalšie funkcie:

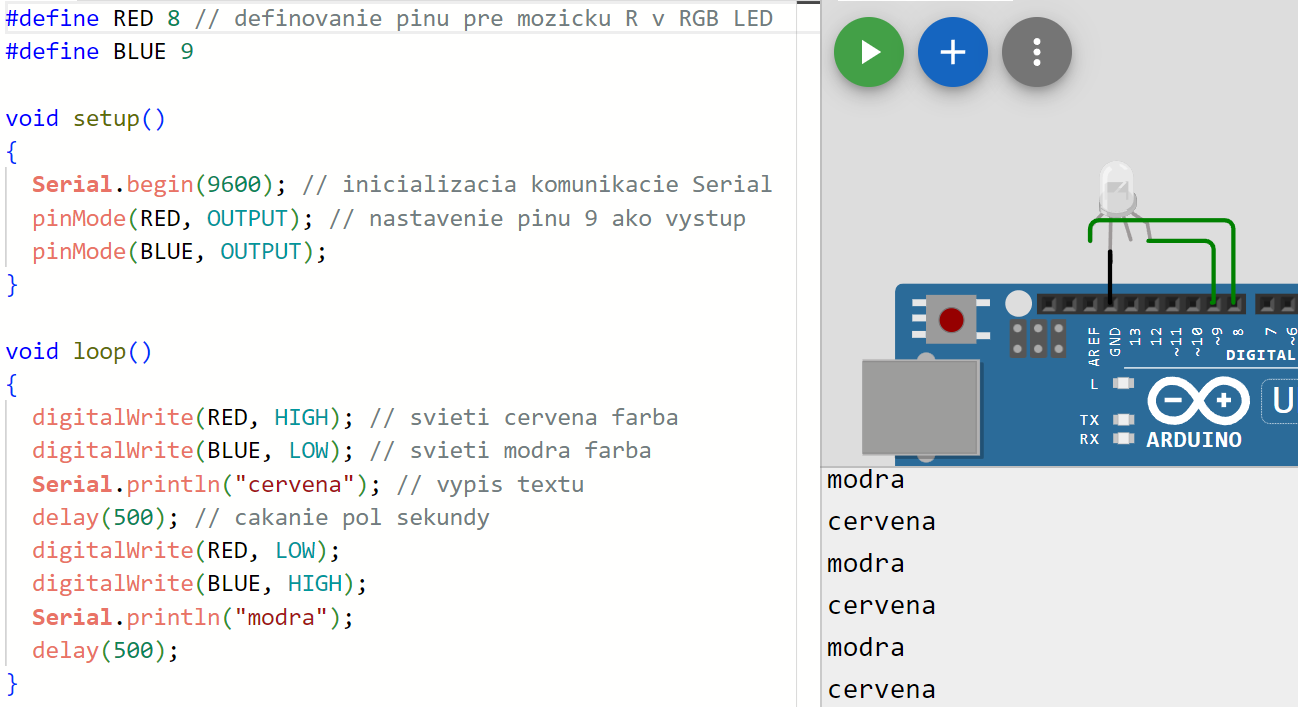
**Serial.begin(BAUDRATE)** – inicializácia Serialu (za BAUDRAE ide „konkrétne“ číslo, napr. 9600)

**Serial.print(“TEXT“)** – výpis/odoslanie textu TEXT do terminálu

**Serial.println(TEXT)** – výpis/odoslanie obsahu premennej TEXT do terminálu a nový riadok (\n)

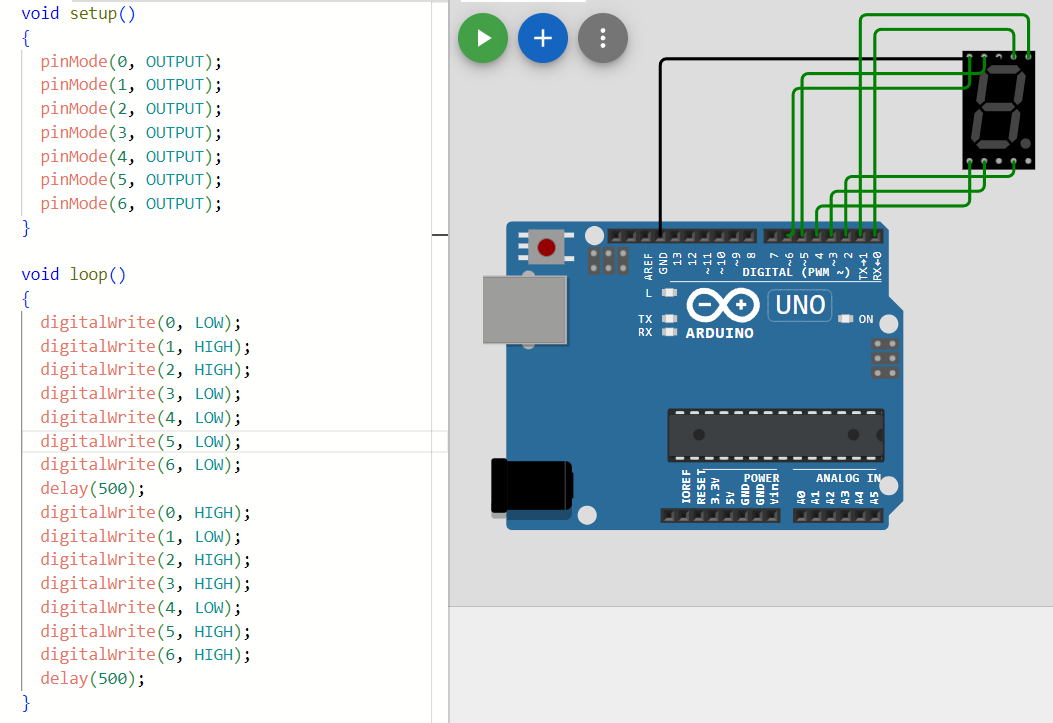
Pozor! Serial/UART využíva na komunikáciu nie len USB, ale aj pin 0 a 1, takže tam nič nepripájajte.

Príklad na blikanie RGB s výpisom do terminálu:

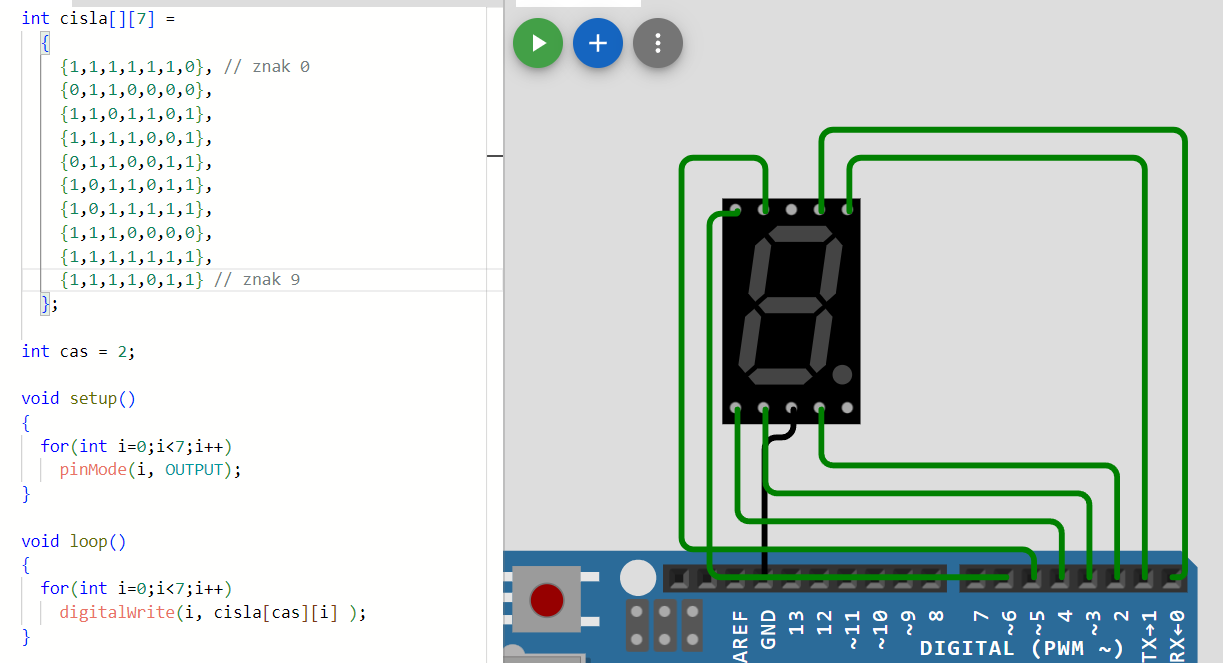


ÚLOHA: príklad na blikanie R-G-B-Y-W a výpis ich farieb

Príklad na blikanie 1-5 na 7-segmentovej dióde:



Príklad na zobrazenie jedného čísla (z 0-9) na 7-segmentovej dióde (s využitím poľa)



ÚLOHA: príklad na postupné zobrazenie 0-9 pri rôznom poradí pinov

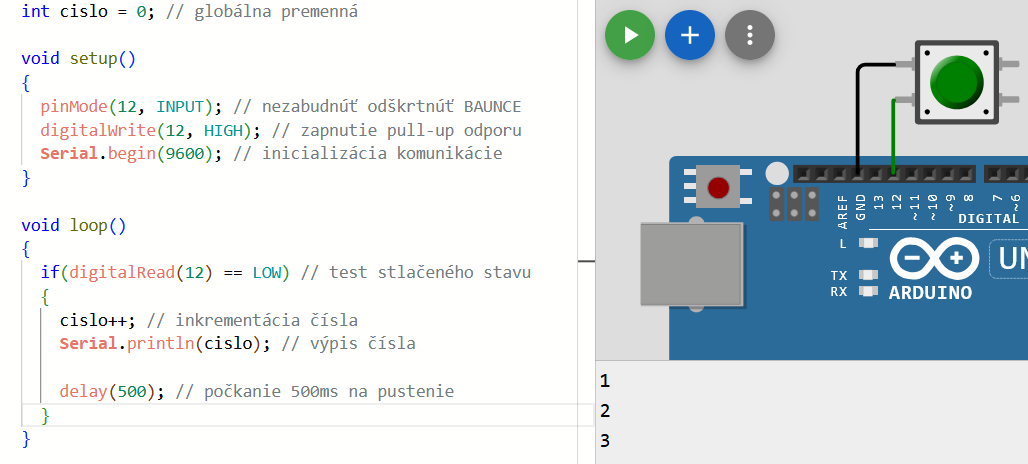
Ďalšie funkcie:

**random(CISLO)** – vygenerovanie náhodného čísla (môže sa použiť aj rozsah)

**randomSeed(analogRead(A0))** – vytvorenie náhodného seed-u pre random

**digitalRead(cisloPinu)** – načítanie napäťového stavu na pine (LOW / HIGH)

Príklad na testovanie stlačenia tlačidla:



Tlačidlo prepája dva piny na jednej strane. Pri nestlačenom stave je na voľnom pine FLOATING stav, nie HIGH! Aby tam bolo napätie, treba pripojiť buď externý pull-up (napätie cez rezistor) alebo interný. Ak sa pripojí tlačidlo na VCC, treba pripojiť externý pull-down – nie interný, ten neexistuje.

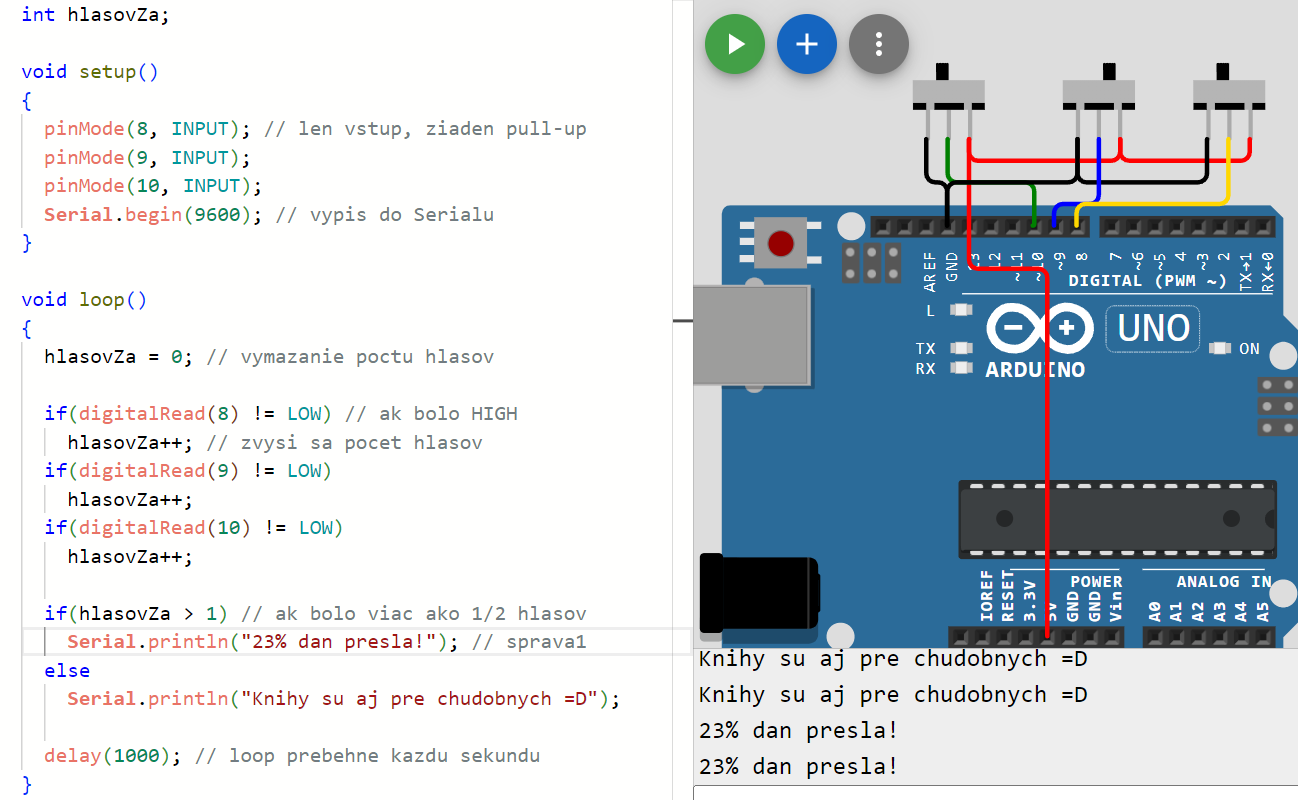
ÚLOHA: príklad na zvýšenie čísla na 7-segmentovom displeji po stlačení tlačidla

ÚLOHA: po každom stlačení tlačidla sa zvýši počet svietiacich diód na bargraphe (a rôzne modifikácie)

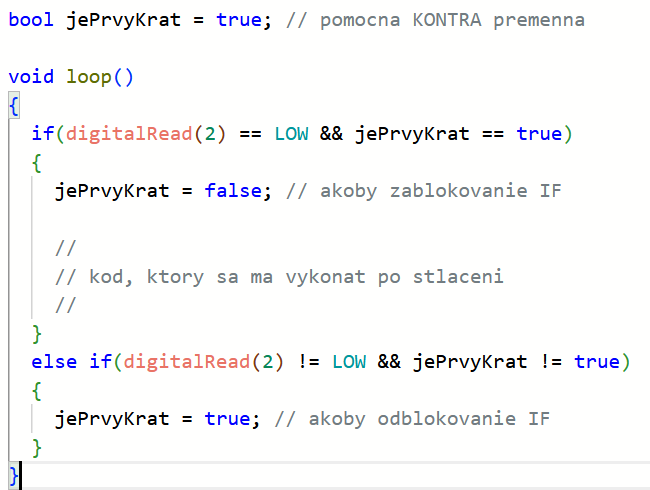
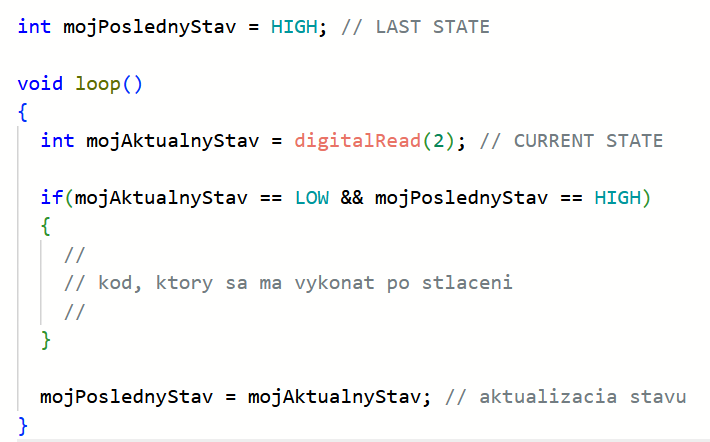
ÚLOHA: vygeneruje sa náhodné číslo 1-3, následne treba stlačiť správne tlačidlo 1-3, aby SCORE++

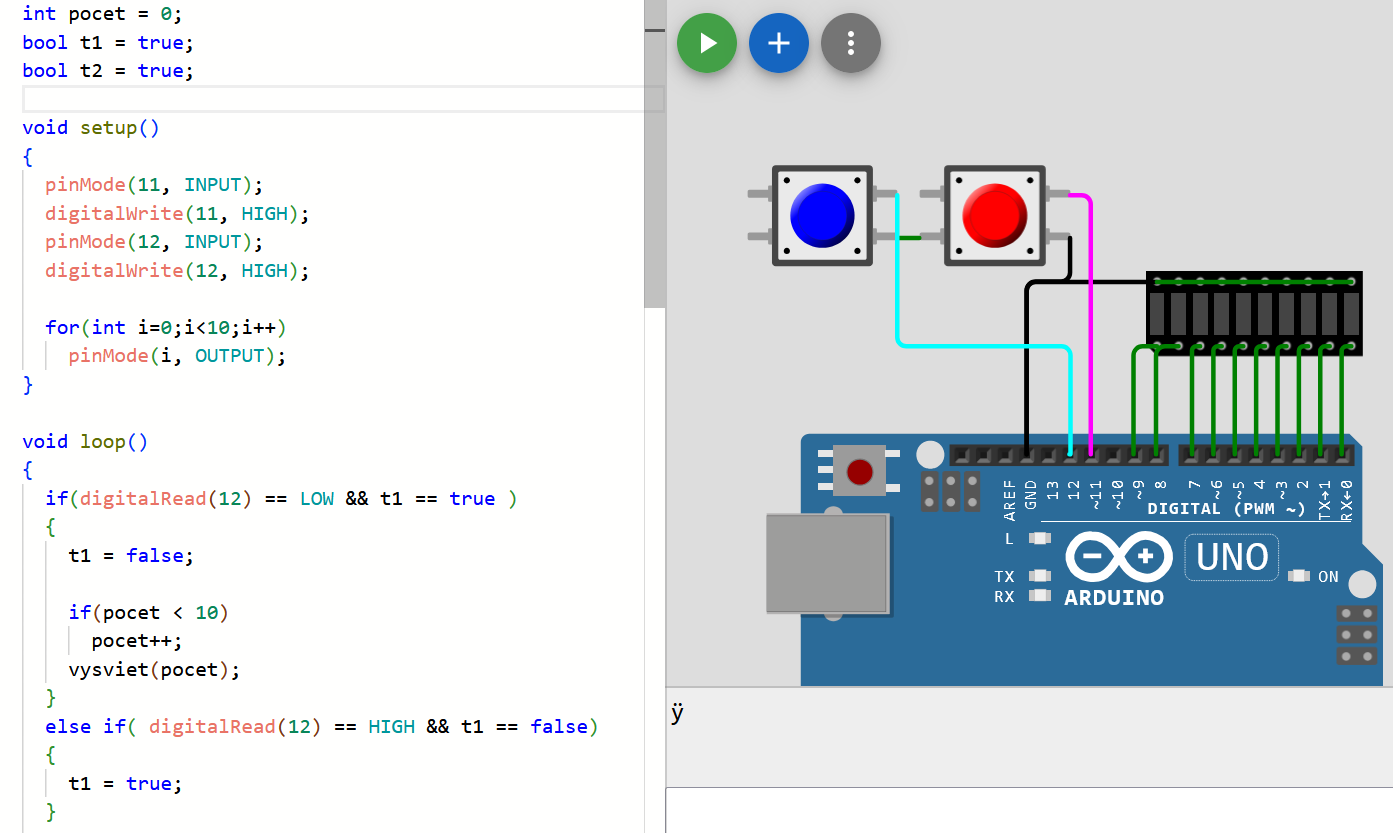
Tlačidlo (button) sa po pustení ĽTM navráti do nestlačenej polohy. Pre Zachovanie stavu sa používa prepínač (switch). Pokiaľ má tri nožičky, prepína ľavú a strednú versus pravú a strednú. Pokiaľ má dve (DIP switch 8), prepína stavy spojený a rozpojený.

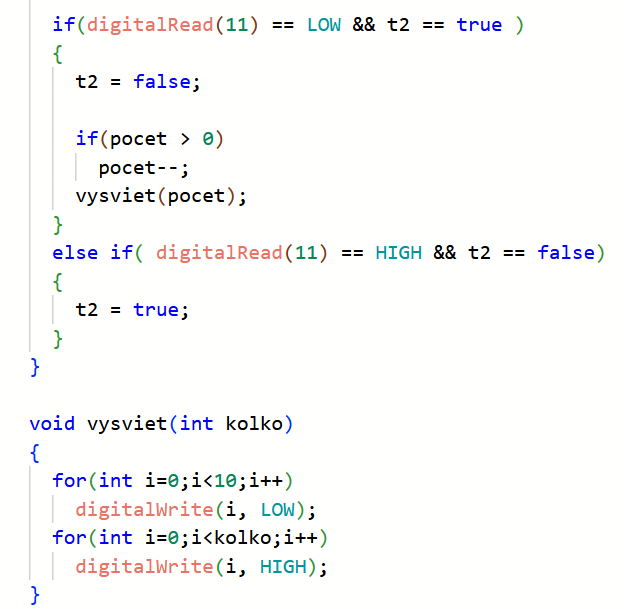
Príklad na 3 prepínače, pričom pri prepnutí väčšieho počtu zobrazí správu1, pri 0/1 zobrazí správu2



Riešenie s delay() nie je vhodné, lebo nereaguje na rýchle stláčanie a reaguje viackrát pri držaní. Pokiaľ má MCU reagovať na JEDNO STLAČENIE, použije sa buď konrta podmienka, alebo lastState.

Príklad na počítanie stlačenia a zobrazenie počtu pomocou bargraphu



ÚLOHA: príklad na prepínanie farieb a ich rozsvietenie na RGB LED dióde

Ďalšie funkcie:

**#include <LiquidCrystal.h>** – prlinkuje knižnicu pre prácu s LCD displejom

**LiquidCrystal lcd(rs,e,d4,d5,d6,d7)** – vytvorenie objektu, ktorý reprezentuje LCD

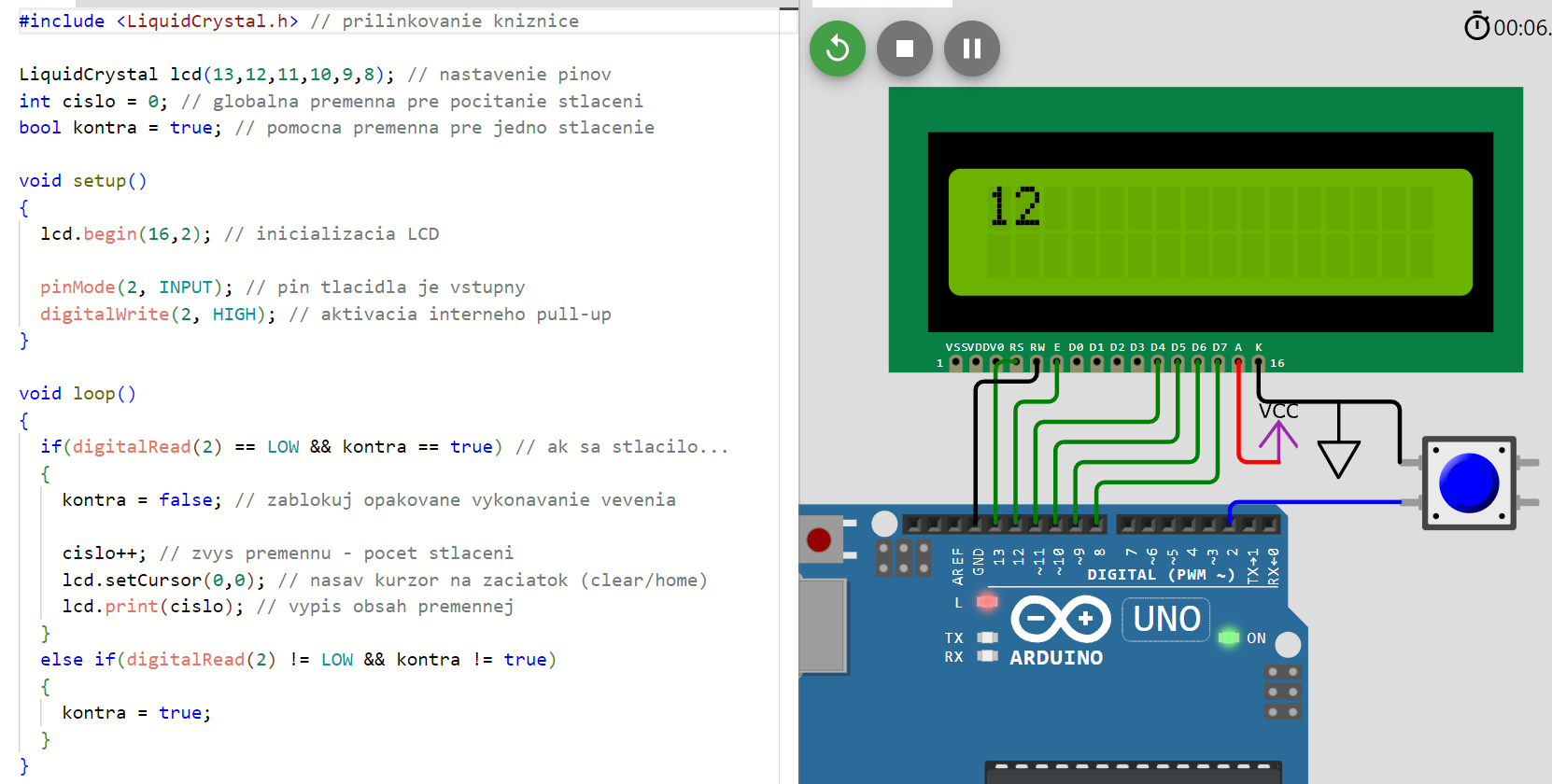
**lcd.begin(16,2)** – inicializácia LCD a nastavenie jeho „rozmerov“

**lcd.print(“TEXT“)** – výpis textu TEXT na LCD displej

**lcd.setCursor(s,r)** – presun kurzoru na súradnicu stĺpec / riadok

*viac funkcii*: <https://www.arduino.cc/reference/en/libraries/liquidcrystal/>

Príklad na prácu s LCD displejom:

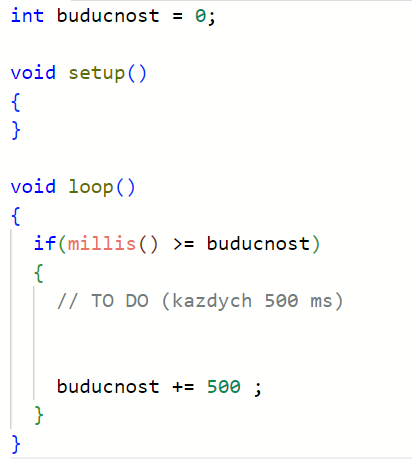


ÚLOHA: príklad na dve tlačidlá, jedno zvyšuje a druhé znižuje číslo v rozsahu 0 až 9.

ÚLOHA: príklad na jedno tlačidlo. Každé párne stlačenie vypíše ON a každé nepárne stlačenie OFF.

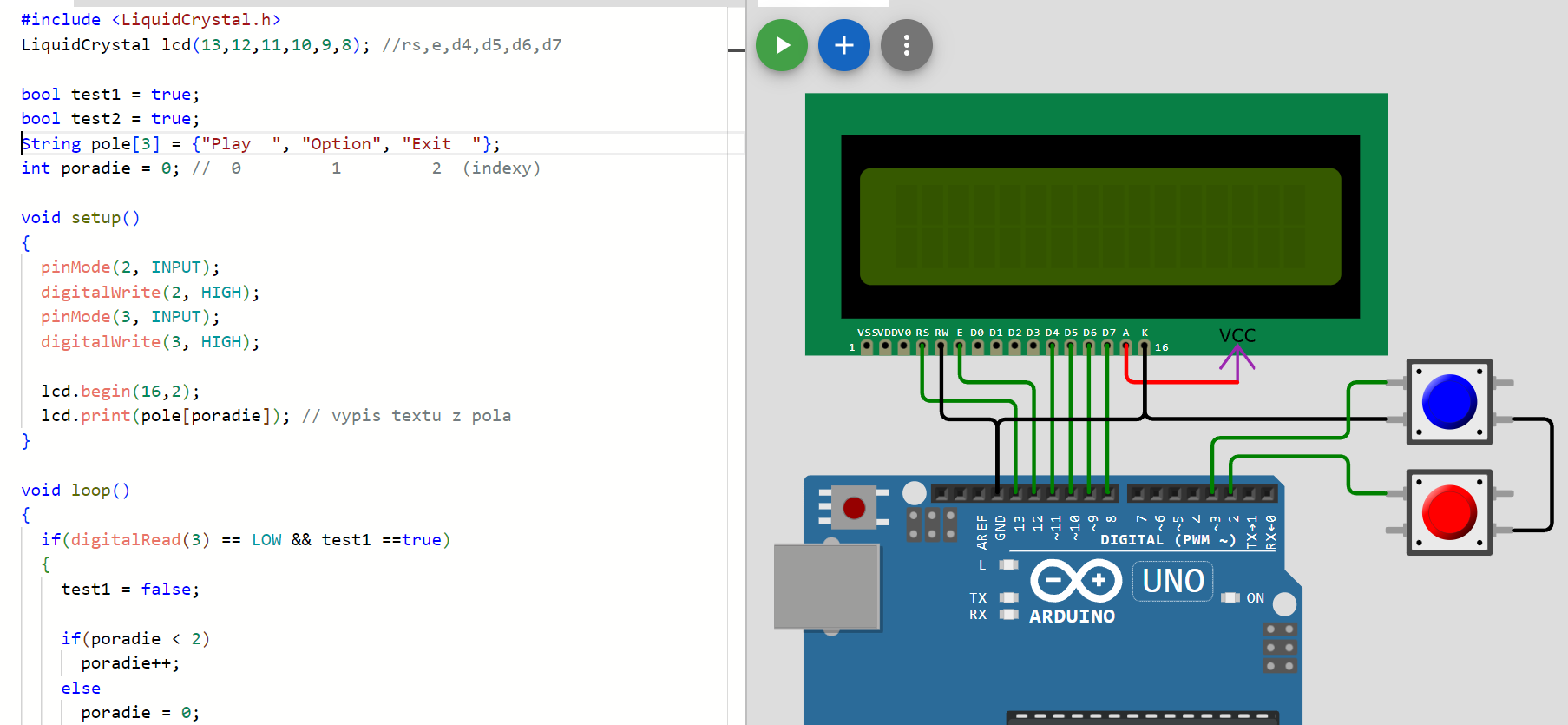
Ďalšie funkcie:

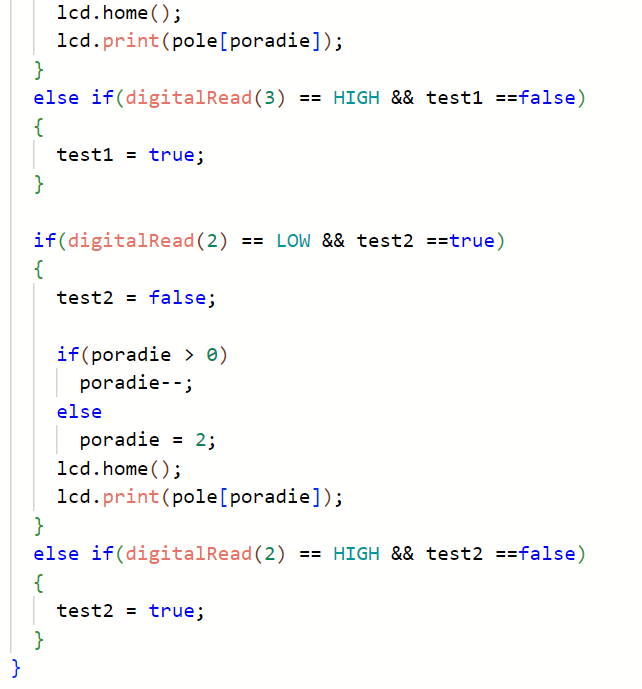
**millis()** – vráti počet milisekúnd od spustenia programu. Vhodné na vytváranie intervalov



ÚLOHA: blikanie LED diódy v 500ms intervale (bez delay) –> 500ms ON / 500ms OFF.

Príklad na MENU:





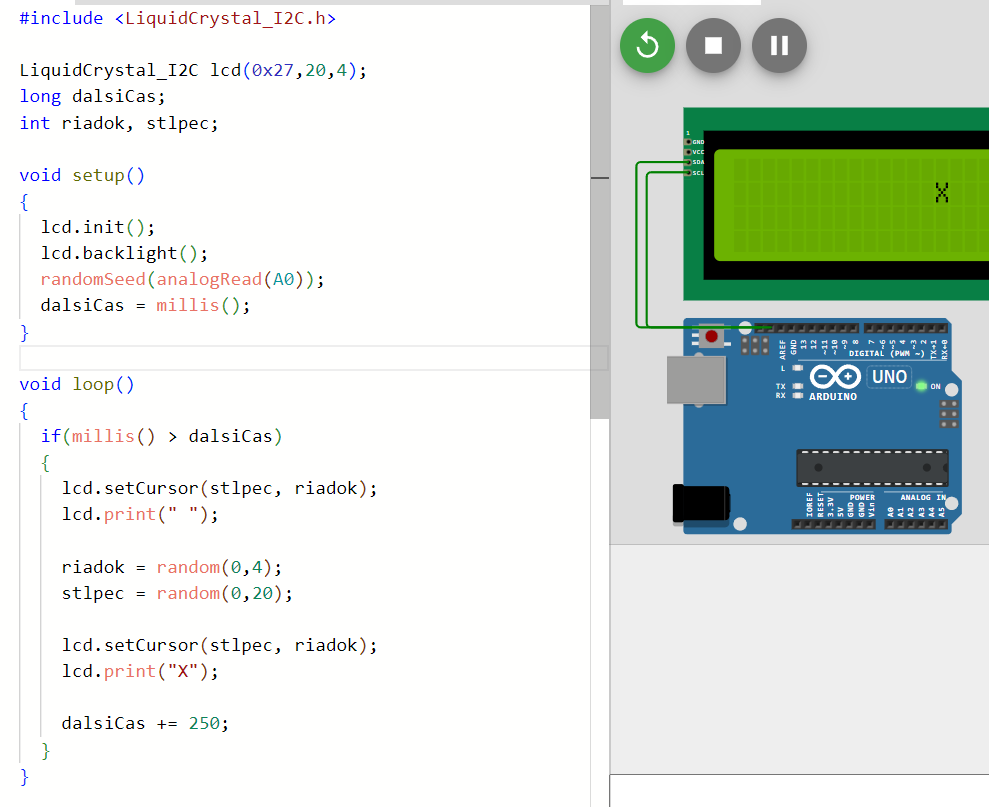
Ďalšie funkcie:

**#include <** **LiquidCrystal\_I2C.h>** – prlinkuje knižnicu pre prácu s LCD displejom s I2C prevodníkom

**LiquidCrystal\_I2C lcd(0x27, 16, 2)** – vytvorenie objektu lcd pre 16x2 displej s adresou 0x27

**lcd.init()** – inicializácia LCD

Príklad na I2C LCD:



ÚLOHA: vykreslenie panáčika a jeho pohyb pomocou 4x tlačidiel v rámci LCD.

ÚLOHA: posun v intervaloch obsahu každého riadku na nový a generovanie znakov “\*“ do prvého.

Ďalšie funkcie:

**Serial.available()** – navráti počet bajtov v zásobníku (vhodné pre test, či sú nejaké nové dáta)

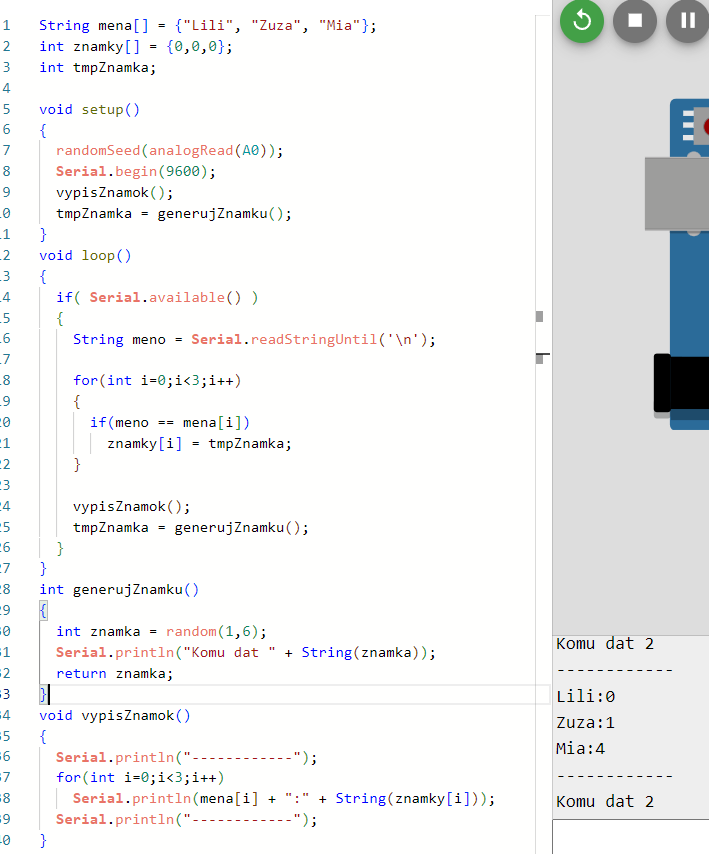
**Serial.read()** – načítanie 1 znaku / bajtu

**Serial.readString()** – načítanie textu (čaká 1 sek. na nové dáta)

**Serial.readStringUntil(‘#‘)** – načítanie textu (čaká 1 sek. na nové dáta alebo prijatie znaku #)

**Serial.parseInt()** – načítanie znakov a konverzia do čísla (čaká 1 sek., alebo na nečíselný znak)

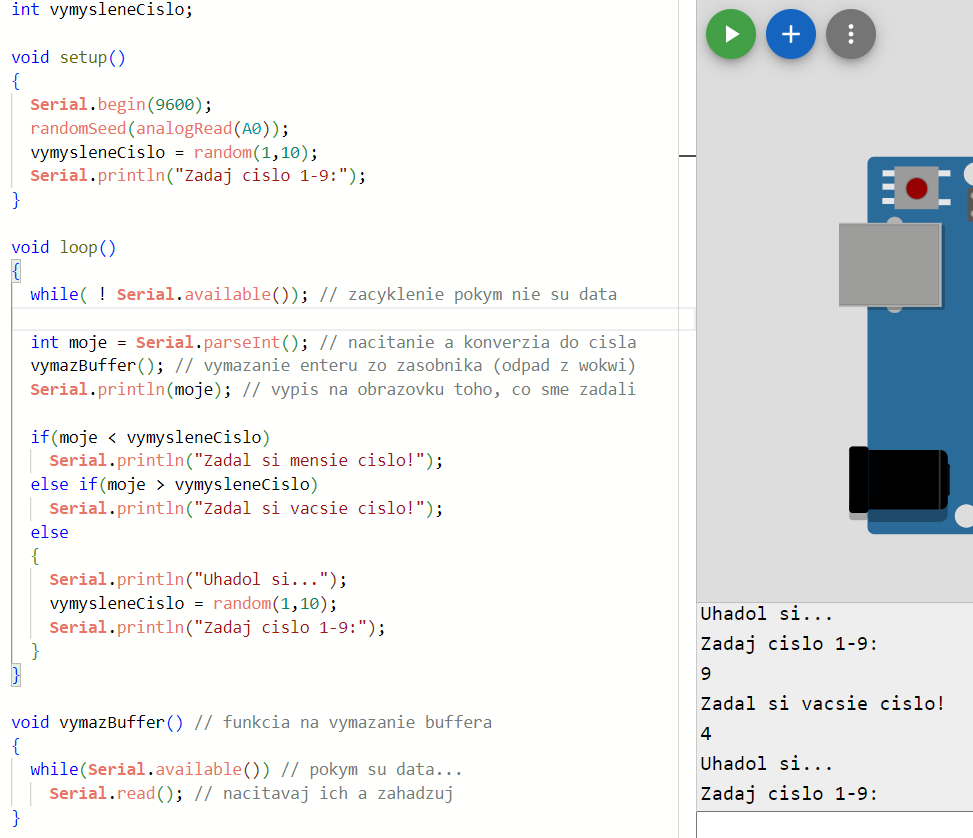
Príklad na načítanie zo Serial:



ÚLOHA: kalkulačka cez Serial (opakované načítavanie 2 čísel a jednej operácie a výpis výsledku).

ÚLOHA: bootovanie do konfig. režimu, keď je tlačidlo stlačené. Následne základný chod – blikanie.

Príklad na načítanie čísla zo Serialu a čakanie na AKCIU:



Analógový prevodník (ADC) konvertuje analógové napätie (0V – 5V) na 10 bitovú hodnotu (číslo 0 až 1023). Je 6 kanálový a využíva piny A0 - A5. Pracuje na princípe postupnej aproximácie. Pre načítanie sa používa funkcia:

**analogRead(cisloAxPinu)** – načítanie analógovej hodnoty z pinu A0-A5

Pre najjednoduchší spôsob získania regulovateľného analógového napätia je možné využiť potenciometer. Ten vnútorne vytvára napäťový delič, čiže nejde ani tak o jeho hodnotu odporu, ale o pomer odporu medzi GND-SIG a SIG-VCC (R1/R2).

Ďalšie funkcie:

**map(hodnota, zMIN, zMAX, doMIN, doMAX)** – premapovanie hodnoty z rozsahu Z do rozsahu DO.

Príklad na načítanie analógovej hodnoty a jej premapovanie:



ÚLOHA: porovnávanie dvoch potenciometrov a pokiaľ P1<P2 LED svieti, pokiaľ P1>P2 LED nesvieti.

ÚLOHA: na LCD displeji as zobrazuje toľko ‘X‘, nakoľko je potenciometer pootočený (progressbar).

Vonkajšie prerušenie (external interrupt) vyvoláva prerušenie programu potom, čo na pinoch INT0 / INT1 zachytí zvolený signál. Akcie, na ktoré vonkajšie prerušenie reaguje sú:

- výskyt nábežnej hrany (prechod z log.0 na log.1) – RISING EDGE

- výskyt dobežnej hrany (prechod z log.1 na log.0) – FALLING EDGE

- výskyt nábežnej alebo dobežnej hrany – CHANGE STATE

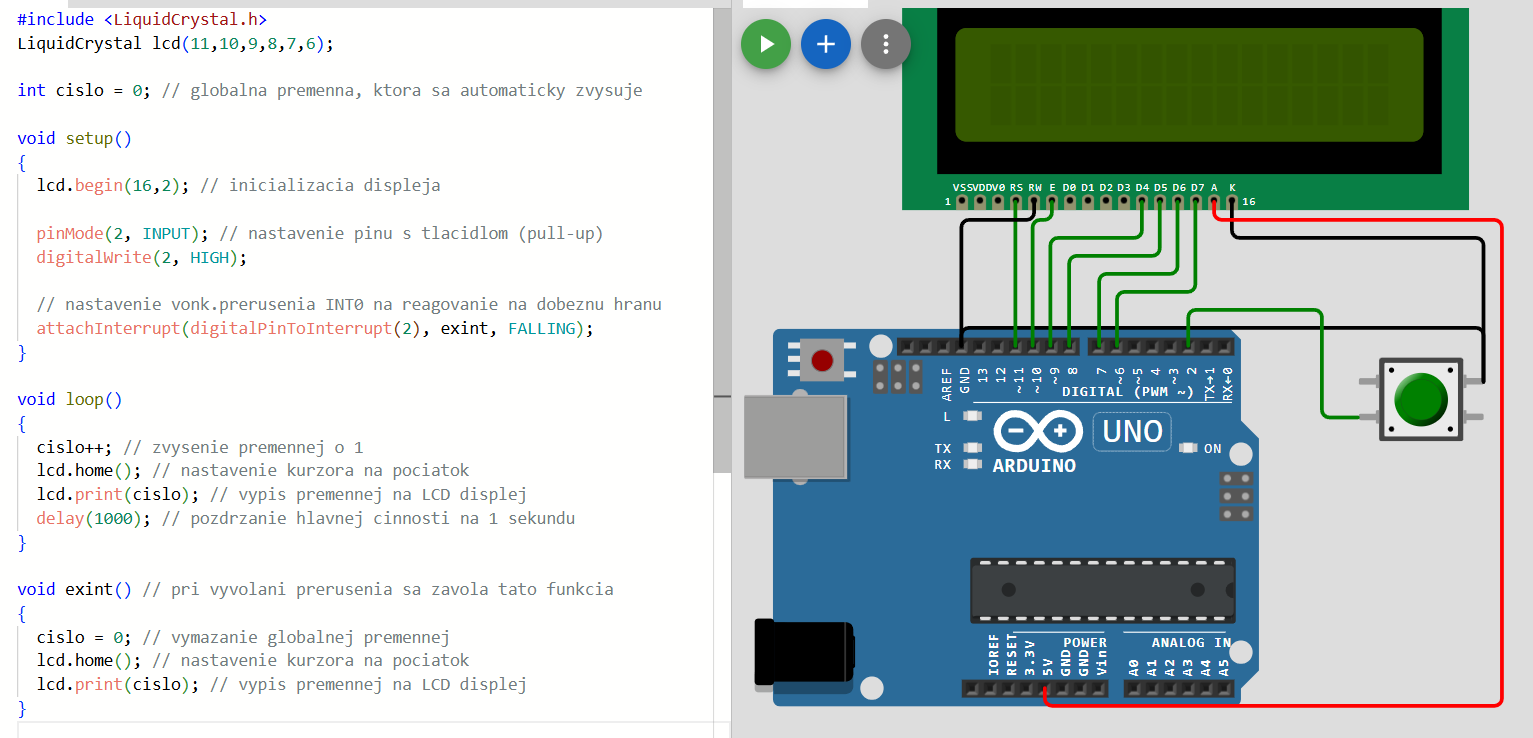
- výskyt napäťovej úrovne 0V – LOW LEVEL

Vonkajšie prerušenie na nachádza na pinoch 2 a 3 v Arduino Uno. Pre nastavenie sa používa funkcia:

**attachInterrupt(digitalPinToInterrupt(2), FUN, AKCIA)** – zavolanie funkcie FUN pri výskyte akcie na 2

Pozor, viacero systémov využíva vlastné prerušovače (napr. delay, millis, I2C, ...). Preto je zakázané volať tieto funkcie počas obsluhy iného prerušenia. Obsluha prerušenia (funkcia, ktorá sa volá po nastaní prerušenia) by mala byť krátka a rýchla, zložitejší kód by sa mal vykonávať v loop().

Príklad na resetovanie počítadla:



ÚLOHA: systém na kalibráciu – home position (dve dorazové tlačidlá ukončia otáčanie krokového m.).

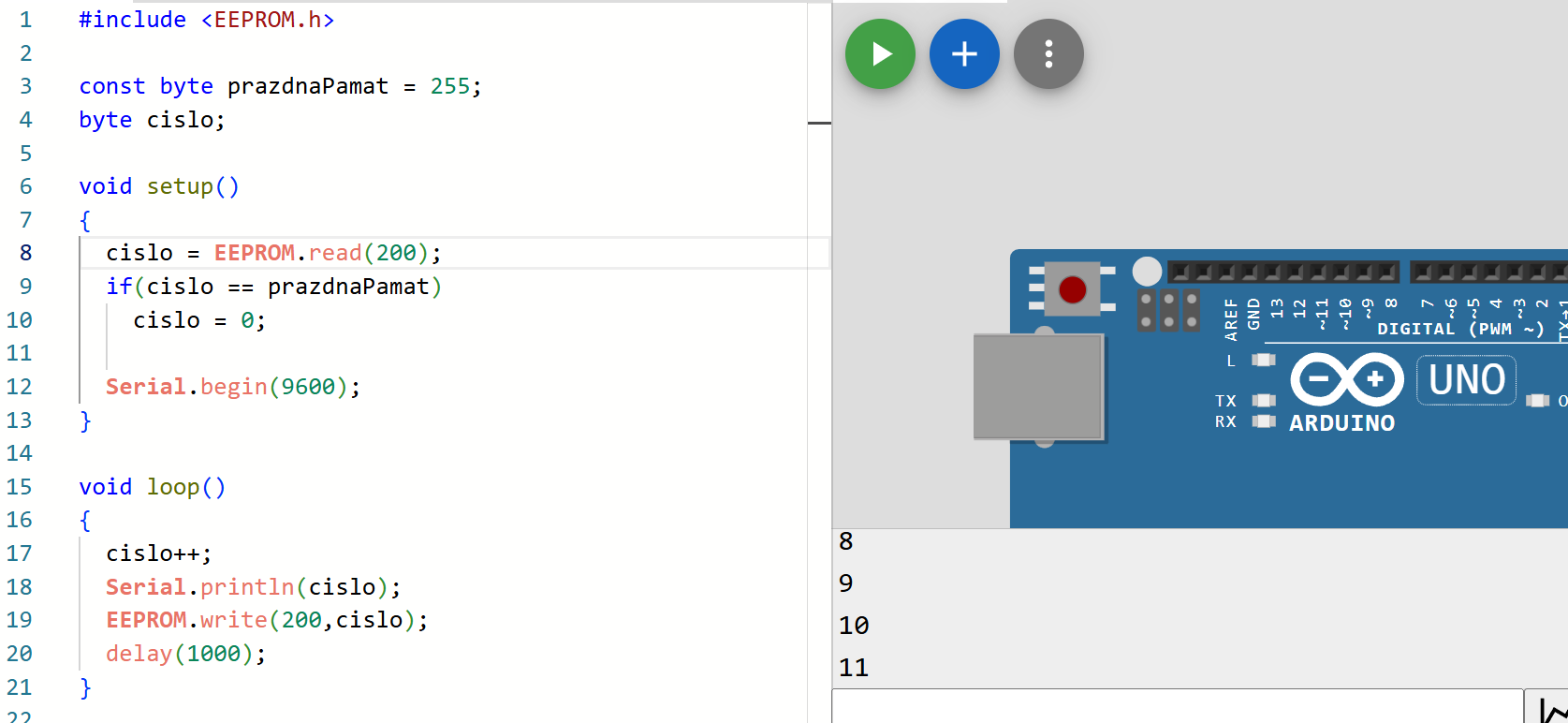
Arduino obsahuje EEPROM pamäť o veľkosti 512 B. Dáta v tejto pamäti ostanú uložené aj po odpojení elek. Zzdroja – ide o non volatívnu pamäť. Pre prácu s ňou treba knižnicu: EEPROM.h

Okrem klasických funkcií obsahuje funkcie put() a get(), ktoré dokážu zapísať celú štruktúru dát (napr float ako 4 po sebe idúce bajty v pamäti). Počet zápisov je cca 100 000

**EEPROM.write(ADRESA,HODNOTA)** – zápis bajtu (0-255) na adresu (0-511)

**byte a = EEPROM.read(ADRESA)** – načítanie bajtu z adresy eeprom pamäte

Príklad na EEPROM:



Pre resetovanie Arduina nepoužívajte PLAY/STOP tlačidlo, ale červený reset!

Častým komponentom v Arduino projektoch je prvok WS2812. Známy je pod označením NEOPIXEL. Ide o všefarebný zobrazovací komponent riadený sériovou linkou. Znamená to, že vie zobraziť akúkoľvek RGB farbu, pričom do Arduina sa zapája akékoľvek množstvo len pomocou jedného pinu.

Na prácu s ním je možné použiť knižnicu „Adafruit\_NeoPixel.h“. Pri vytváraní objektu treba uviesť počet prvkov v sérii, pin na ktorý sa prvý pripája, typ prvku (najčastejšie NEO\_GRB+NEO\_KHZ800).

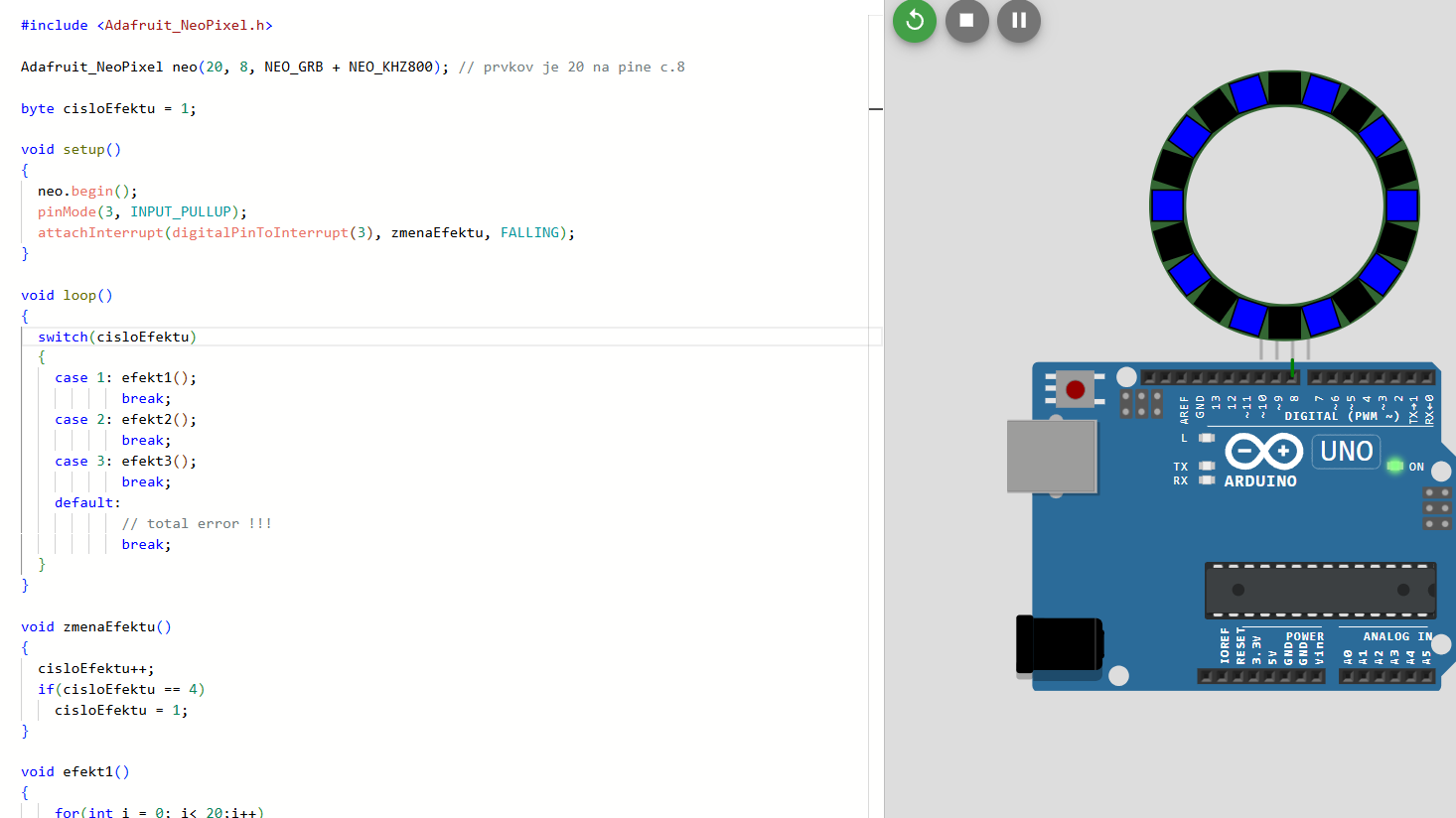
**neo.begin()** – inicializácia objektu

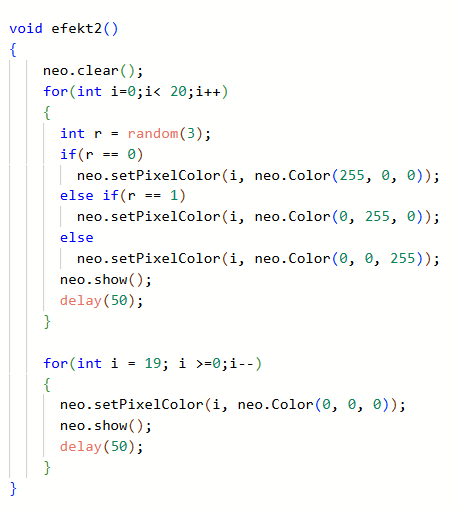
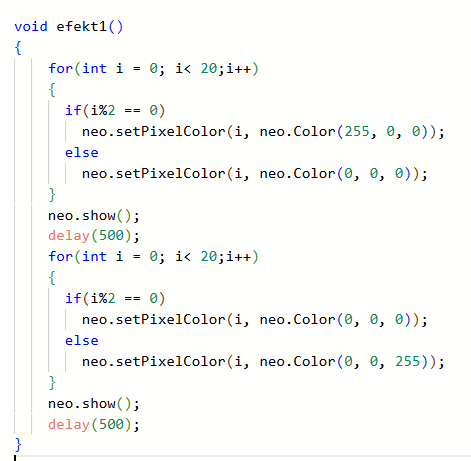
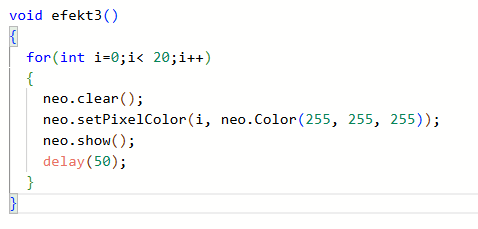
**neo.setPixelColor(KTORY, neo.Color(255, 0, 0))** – nastavenie RED farby na prvku s poradím KTORY

**neo.show()** – refresh farieb na prvkoch

<https://github.com/adafruit/Adafruit_NeoPixel/blob/master/Adafruit_NeoPixel.h> (zoznam funkcii)

Príklad na NEOPIXEL:



Najpoužívanejšie funkcie, a štruktúry vo Wiring*:* [*https://docs.arduino.cc/language-reference/*](https://docs.arduino.cc/language-reference/)

Najpoužívanejšie súčiastky vo Wowki*:* [*https://docs.wokwi.com/getting-started/supported-hardware*](https://docs.wokwi.com/getting-started/supported-hardware)

Bežný Arduino kit:

