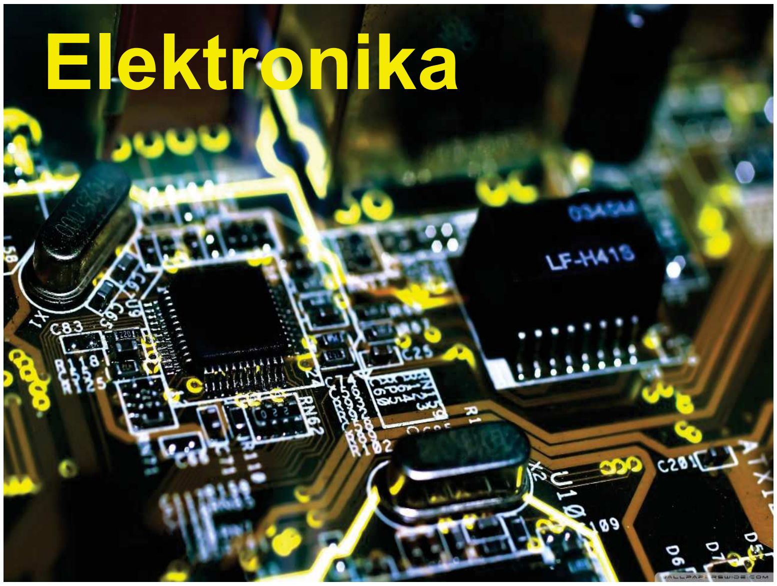
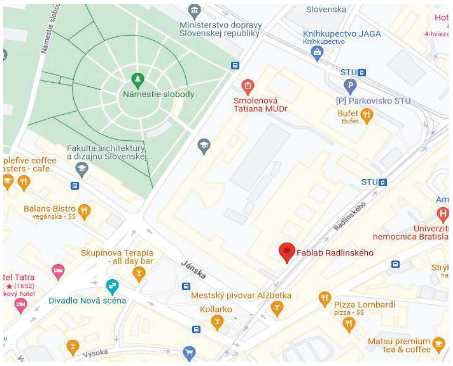
Elektronika



12. 4. 2023

- FabLab Radlinského, budova FCHPT STU
- od 9:30

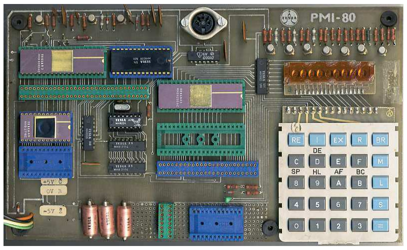


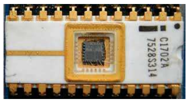
ARDUINO

open source

open hardware

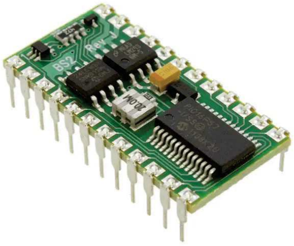
Predchodcovia





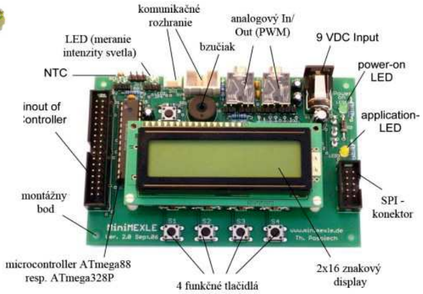
Predchodcovia

Parallax Basic Stamp 2



```
Main:
  HIGH 0
  PAUSE 500
  LOW 0
  PAUSE 500
  GOTO Main
  END
```

MiniMexle



História

Arduino created in 2005 by a team of professors at an Italian college led by a professor **Massimo Banzi.**

Massimo had taught his students previously using the BASIC STAMP but it was more expensive ($100 range).

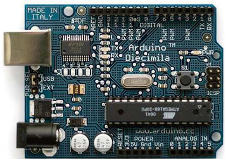
The name Arduino has an unusual origin: "Casa di Re Arduino" - The name of a pub in Ivrea, Italy that Massimo and his associates frequented, but **Arduin of Ivrea** was In fact a king of Italy (1002–1014).

http://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino

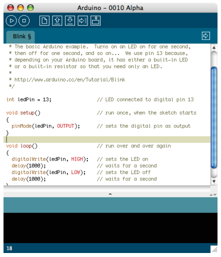
Čo je to Arduino?

Slovo "Arduino" môže znamenať:

A physical piece of hardware



A programming environment



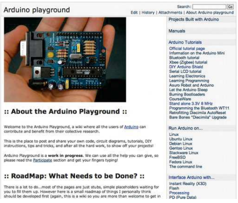
Arduino – 0010 Alpha

Blink §

```
 * The basic Arduino example.  Turns on an LED on for one second,
 * then off for one second, and so on...  We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;                // LED connected to digital pin 13

void setup()                    // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

void loop()                     // run over and over again
{
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(1000);                  // waits for a second
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(1000);                  // waits for a second
```

18

A community & philosophy



Arduino playground

Search: Go

Edit | History | Attachments | About Arduino playground

:: About the Arduino Playground ::

Welcome to the Arduino Playground, a wiki where all the users of Arduino can contribute and benefit from their collective research.

This is the place to post and share your own code, circuit diagrams, tutorials, DIY instructions, tips and tricks, and after all the hard work, to show off your projects!

Arduino Playground is a **work in progress**. We can use all the help you can give, so please read the Participate section and get your fingers typing!

:: RoadMap: What Needs to be Done? ::

There is a lot to do...most of the pages are just stubs, simple placeholders waiting for you to fill them up. However here is a small roadmap of things I personally think should be developed first (again, this is a wiki so you are more than welcome to get in

Projects Built with Arduino

Manuals

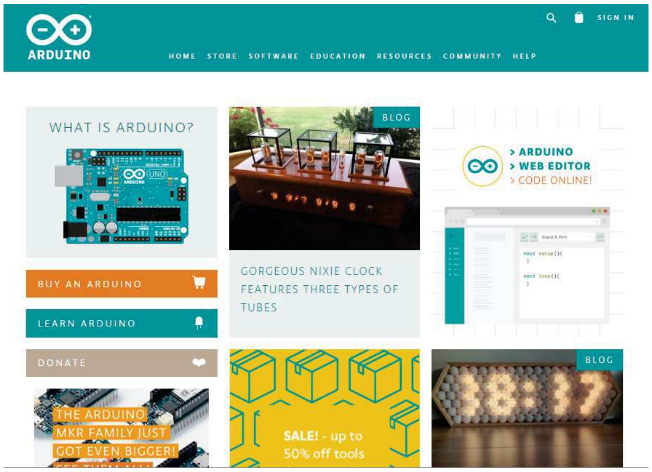
Arduino Tutorials
- Official tutorial page
- Information on the Arduino Mini
- Bluetooth tutorial
- Xbee (Zigbee) tutorial
- DIY Arduino Shield
- Serial LCD tutorial
- Learning Electronics
- Learning Programming
- Asuro Robot and Arduino
- Let the Arduino Sleep
- Burning Bootloaders
- CourseWare
- Stand alone 3.3V 8 MHz
- Programming the Bluetooth WT11
- Retrofitting Diecimila AutoReset
- Bare Bones "Diecimila" Upgrade

Run Arduino on...
- Linux
- Ubuntu Linux
- Debian Linux
- Gentoo Linux
- Slackware Linux
- FreeBSD
- Fedora Linux
- The command line

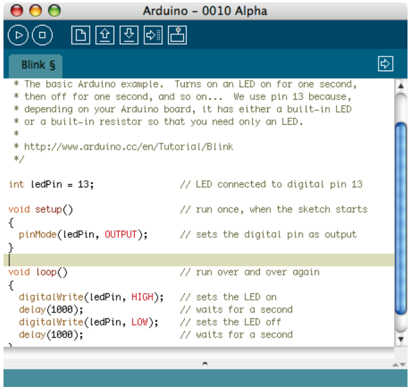
Interface Arduino with...
- Instant Reality (X3D)
- Flash
- Processing
- PD (Pure Data)

Arduino Web

https://www.arduino.cc/



Arduino Software



Arduino – 0010 Alpha

Blink §

```
 * The basic Arduino example.  Turns on an LED on for one second,
 * then off for one second, and so on...  We use pin 13 because,
 * depending on your Arduino board, it has either a built-in LED
 * or a built-in resistor so that you need only an LED.
 *
 * http://www.arduino.cc/en/Tutorial/Blink
 */

int ledPin = 13;                // LED connected to digital pin 13

void setup()                    // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

void loop()                     // run over and over again
{
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(1000);                  // waits for a second
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(1000);                  // waits for a second
```

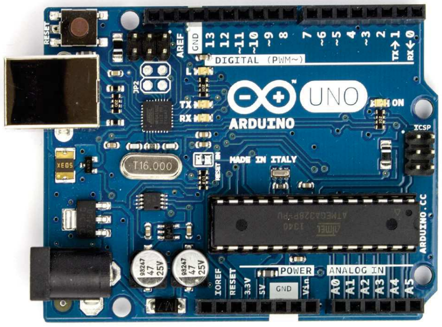
- Like a text editor
- View/write/edit **sketches**
- But then you program them into hardware

Arduino Software

Základná štruktúra programu

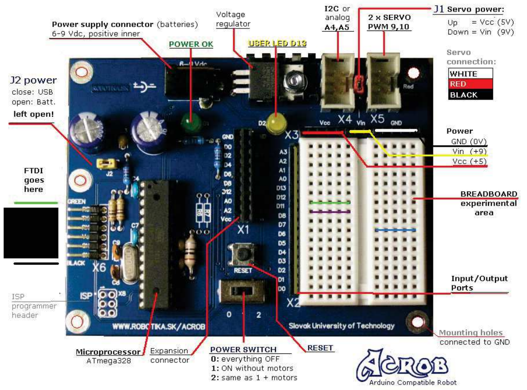
```
void setup()                      int main()
{                                 {
                                    setup();
}
                                    while(1)
                                    {
void loop()                          loop();
{                                    }
                                    return(0);
}                                 }
```

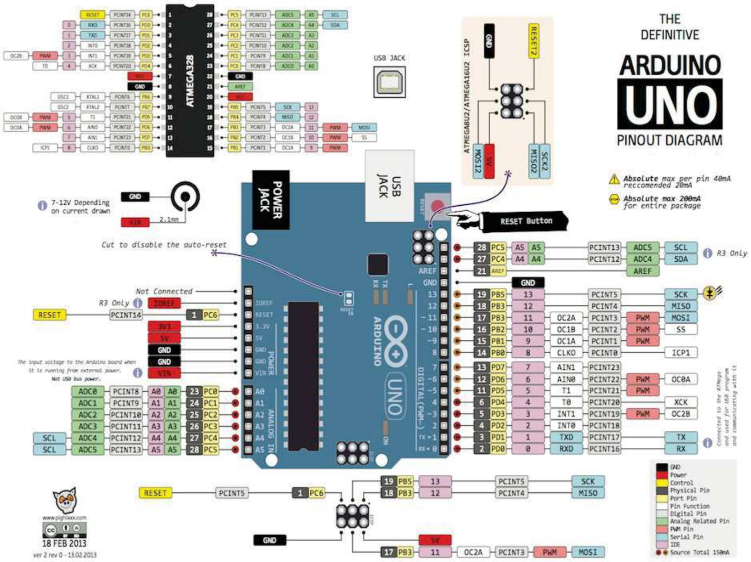
Arduino Hardware



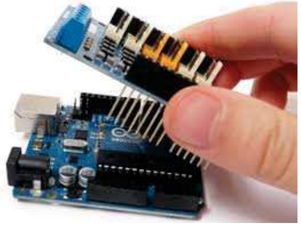
LilyPad

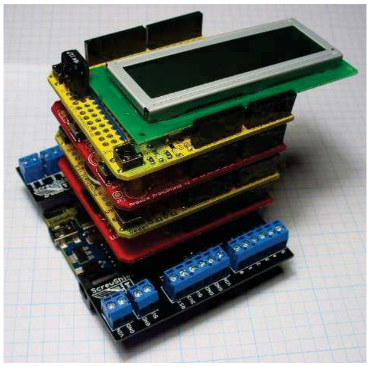
Acrob

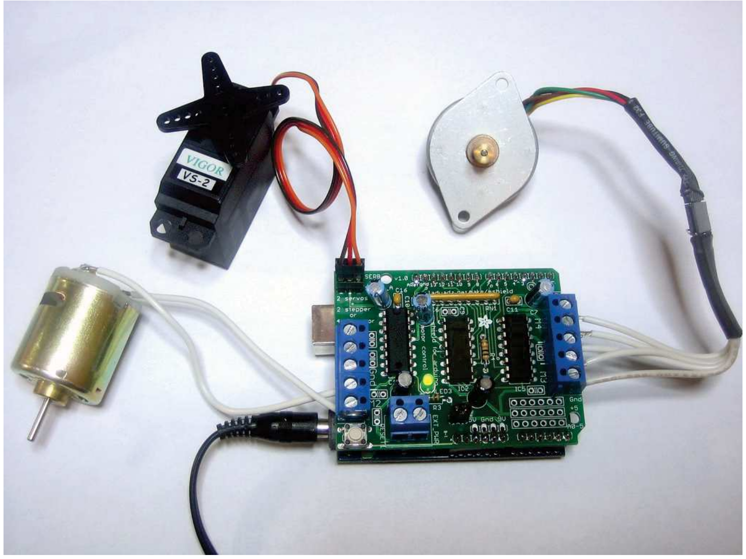


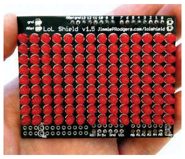


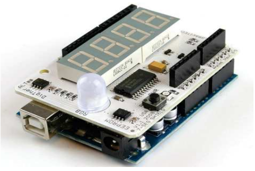
Shields

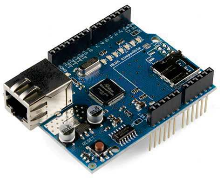


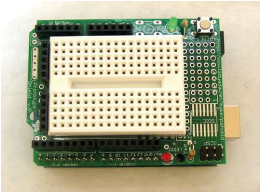












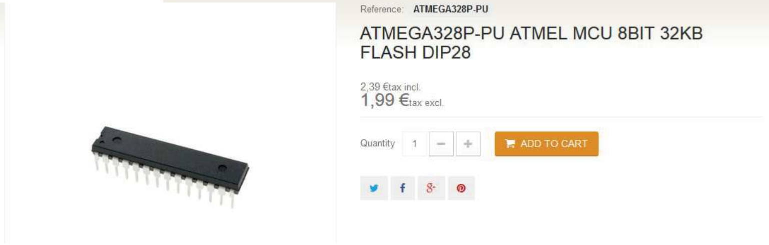
Reference: A000073

Arduino Uno R3 SMD - ORIGINAL ARDUINO (A000073)

19,13 €tax incl.
15,94 €tax excl.

Quantity 1 − + ADD TO CART

View larger

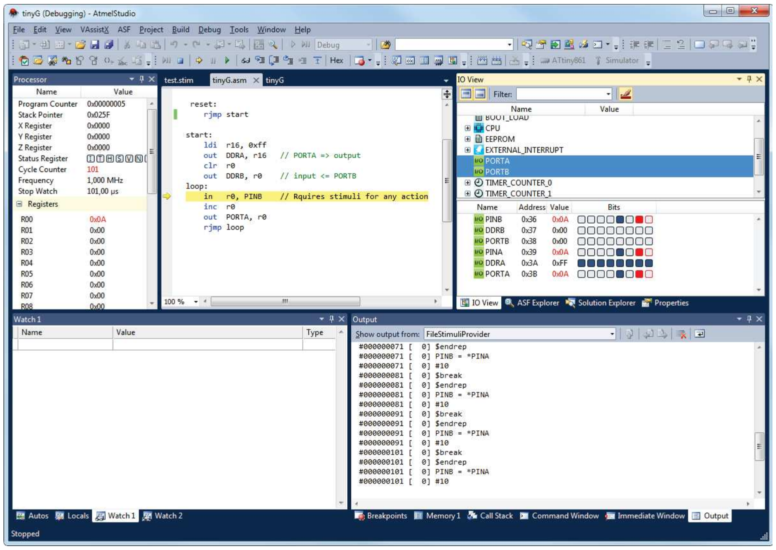


Reference: ATMEGA328P-PU

ATMEGA328P-PU ATMEL MCU 8BIT 32KB FLASH DIP28

2,39 €tax incl.
1,99 €tax excl.

Quantity 1 − + ADD TO CART



tinyG (Debugging) - AtmelStudio

File Edit View VAssistX ASF Project Build Debug Tools Window Help
Debug
Hex
ATtiny861
Simulator
test.stim
tinyG.asm
tinyG
100 %

Processor

Name	Value
Program Counter	0x00000005
Stack Pointer	0x025F
X Register	0x0000
Y Register	0x0000
Z Register	0x0000
Status Register	I T H S V N
Cycle Counter	101
Frequency	1,000 MHz
Stop Watch	101,00 µs
Registers	
R00	0x0A
R01	0x00
R02	0x00
R03	0x00
R04	0x00
R05	0x00
R06	0x00
R07	0x00
R08	0x00

```
reset:
    rjmp start

start:
    ldi  r16, 0xff
    out  DDRA, r16   // PORTA => output
    clr  r0
    out  DDRB, r0    // input <= PORTB
loop:
    in   r0, PINB    // Rquires stimuli for any action
    inc  r0
    out  PORTA, r0
    rjmp loop
```

IO View

Filter:

Name	Value
BOOT_LOAD	
CPU	
EEPROM	
EXTERNAL_INTERRUPT	
PORTA	
PORTB	
TIMER_COUNTER_0	
TIMER_COUNTER_1	

Name	Address	Value	Bits
PINB	0x36	0x0A	□□□□■□■□
DDRB	0x37	0x00	□□□□□□□□
PORTB	0x38	0x00	□□□□□□□□
PINA	0x39	0x0A	□□□□■□■□
DDRA	0x3A	0xFF	■■■■■■■■
PORTA	0x3B	0x0A	□□□□■□■□

Watch 1

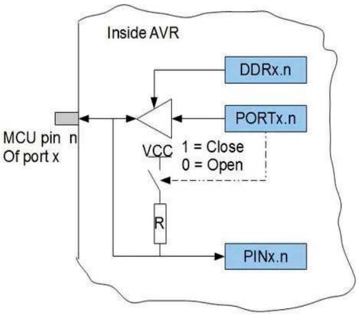
Name	Value	Type

Output

Show output from: FileStimuliProvider

```
#000000071 [  0] $endrep
#000000071 [  0] PINB = *PINA
#000000071 [  0] #10
#000000081 [  0] $break
#000000081 [  0] $endrep
#000000081 [  0] PINB = *PINA
#000000081 [  0] #10
#000000091 [  0] $break
#000000091 [  0] $endrep
#000000091 [  0] PINB = *PINA
#000000091 [  0] #10
#000000101 [  0] $break
#000000101 [  0] $endrep
#000000101 [  0] PINB = *PINA
#000000101 [  0] #10
```

Input/Output



```
#include <avr/io.h>
int main(void)
{
DDRA=0b11111111;//set all pins as output
PORTA=0b10101010;//write ones and zeros to pins
return 0;
}
```

```
#include <avr/io.h>
int main(void)
{
uint8_t x;
//set lower nibble to input
//This operation clears bits 0,1,2,3 of DDRA
DDRA &=~((1<<PA0)|(1<<PA1)|(1<<PA2)|(1<<PA3));
//Setting pulls-up
PORTA |=(1<<PA0)|(1<<PA1)|(1<<PA2)|(1<<PA3);
//set higher nibble to output
/*bit shift "<<"operation and logical OR "|"
used to set bits 4,5,6,7 to "1"*/
DDRA |=(1<<PA4)|(1<<PA5)|(1<<PA6)|(1<<PA7);
//write value to higher nibble
PORTA |=(1<<PA4)|(1<<PA5)|(1<<PA6)|(1<<PA7);
//read value from lower nibble
//and filter out high nibble
x=(0b00001111)&PINA;
return 0;
}
```

```
void setup() {
  pinMode(13, OUTPUT);  // digital pin 13 as outpu
  pinMode(7, INPUT);    // digital pin 7 as input
}

void loop() {
  val = digitalRead(7);   // read the input pin
  digitalWrite(13, HIGH);  // LED on
}
```

Arduino

```
void setup() {
  pinMode(13, OUTPUT);
}

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

2,918 bytes

Atmel Studio

```
#define F_CPU 16000000UL

#include <avr/io.h>
#include <util/delay.h>

int main(void)
{

    DDRB |= (1<<0);
    while (1)
    {
        PORTB |= (1<<0);
        _delay_ms(1000);
        PORTB &= ~(1<<0);
        _delay_ms(1000);
    }
}
```

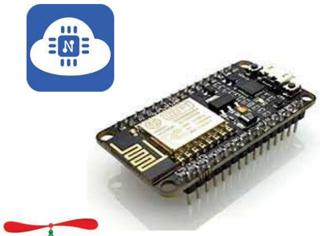
508 bytes

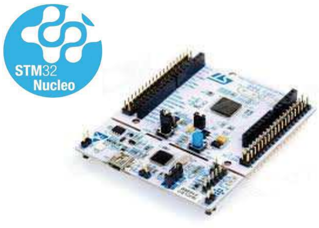
digitalWrite();

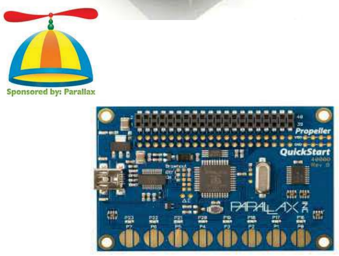
```
void digitalWrite(uint8_t pin, uint8_t val)
{
    uint8_t timer = digitalPinToTimer(pin);
    uint8_t bit = digitalPinToBitMask(pin);
    uint8_t port = digitalPinToPort(pin);
    volatile uint8_t *out;

    if (port == NOT_A_PIN) return;

    // If the pin that support PWM output, we need to turn
it off
    // before doing a digital write.
    if (timer != NOT_ON_TIMER) turnOffPWM(timer);

    out = portOutputRegister(port);

    uint8_t oldSREG = SREG;
    cli();

    if (val == LOW) {
        *out &= ~bit;
    } else {
        *out |= bit;
    }

    SREG = oldSREG;
}
```

PORTB |= (1<<0);

Ďalšie platformy



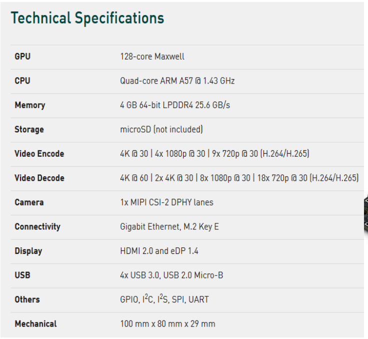




Ďalšie platformy

Low end computer, not a controller

Jetson Nano Developer Kit



Technical Specifications

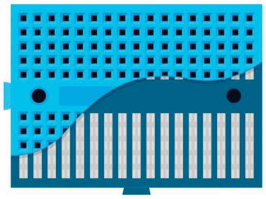
GPU	128-core Maxwell
CPU	Quad-core ARM A57 @ 1.43 GHz
Memory	4 GB 64-bit LPDDR4 25.6 GB/s
Storage	microSD (not included)
Video Encode	4K @ 30 | 4x 1080p @ 30 | 9x 720p @ 30 (H.264/H.265)
Video Decode	4K @ 60 | 2x 4K @ 30 | 8x 1080p @ 30 | 18x 720p @ 30 (H.264/H.265)
Camera	1x MIPI CSI-2 DPHY lanes
Connectivity	Gigabit Ethernet, M.2 Key E
Display	HDMI 2.0 and eDP 1.4
USB	4x USB 3.0, USB 2.0 Micro-B
Others	GPIO, I^2C, I^2S, SPI, UART
Mechanical	100 mm x 80 mm x 29 mm

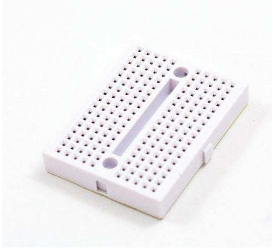
Číslicové (digitálne) výstupy: 0/1

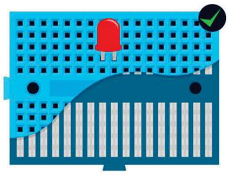
```
void setup() {
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // turn the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // turn the LED off
  delay(1000);              // wait for a second
}
```

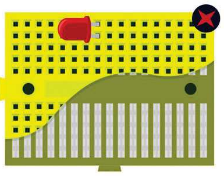
Breadboard

Nepájivé kontaktní pole (NKP)

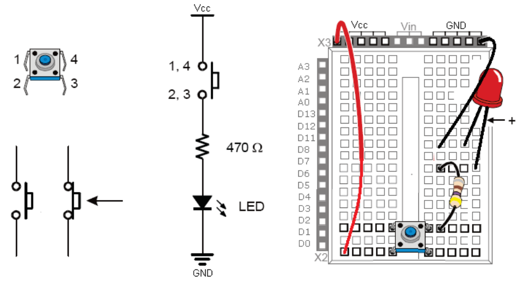




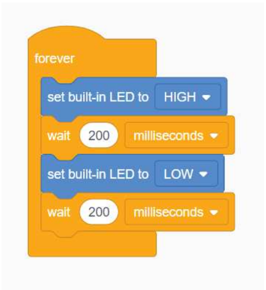




Číslicové (digitálne) vstupy: 0/1



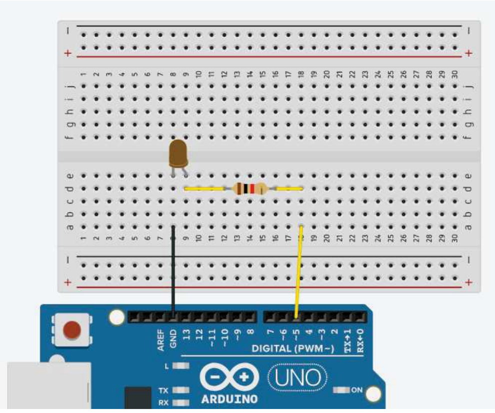
Číslicové (digitálne) vstupy: 0/1



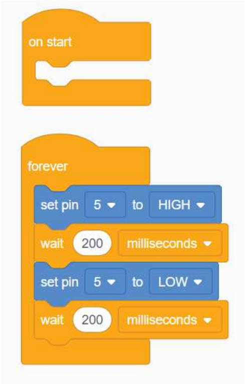
```
forever
  set built-in LED to HIGH
  wait 200 milliseconds
  set built-in LED to LOW
  wait 200 milliseconds
```

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

void loop()
{
  digitalWrite(LED_BUILTIN, HIGH);
  delay(200); // Wait for 200 ms
  digitalWrite(LED_BUILTIN, LOW);
  delay(200); // Wait for 200 ms
}
```

Číslicové (digitálne) vstupy: 0/1



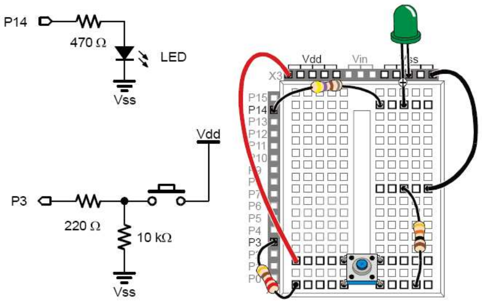
Číslicové (digitálne) vstupy: 0/1



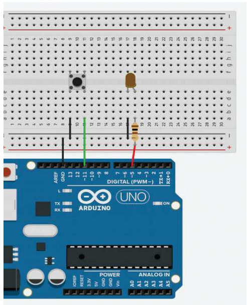
```
on start

forever
  set pin 5 to HIGH
  wait 200 milliseconds
  set pin 5 to LOW
  wait 200 milliseconds
```

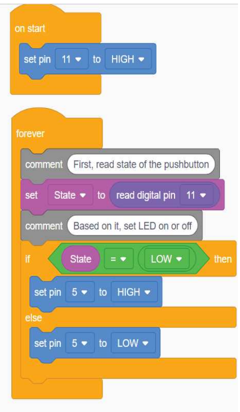
```
void setup()
{
  pinMode(5, OUTPUT);
}


void loop()
{
  digitalWrite(5, HIGH);
  delay(200); // Wait for 200 ms
  digitalWrite(5, LOW);
  delay(200); // Wait for 200 ms
}
```

Číslicové (digitálne) vstupy: 0/1



Číslicové (digitálne) vstupy: 0/1



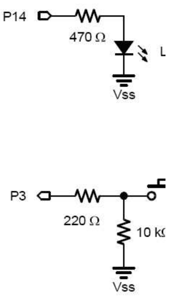


```
on start
  set pin 11 to HIGH

forever
  comment First, read state of the pushbutton
  set State to read digital pin 11
  comment Based on it, set LED on or off
  if State = LOW then
    set pin 5 to HIGH
  else
    set pin 5 to LOW
```

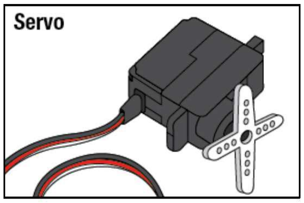
```
int State = 0;

void setup()
{
  pinMode(11, INPUT);
  pinMode(5, OUTPUT);
}


void loop()
{

  State = digitalRead(11);

  if (State == LOW)

    digitalWrite(5, HIGH);

  else

    digitalWrite(5, LOW);
}
```

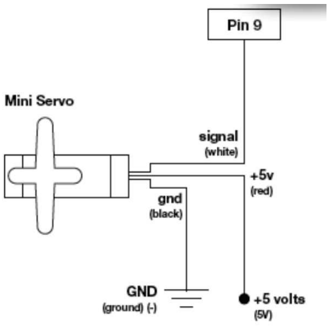
Číslicové (digitálne) vstupy: 0/1

```
#define SWITCH  3               // select the pin for Switch
#define LED_Red 7               // select the pin for LED

int State = 0;                  // variable to store the

void setup()
{
 pinMode(LED_Red, OUTPUT );     // this pin is an OUTPUT
 pinMode(SWITCH, INPUT );       // this pin is an INPUT
}

void loop()                     // endless loop
{
 State =  digitalRead(SWITCH);
 if (State == HIGH )            // notice 2x = !!!
   digitalWrite(LED_Red, HIGH);   // make it visible
 else
   digitalWrite(LED_Red, LOW);    // turn off
 delay(250);                    // wait 1/4s for another
}
```

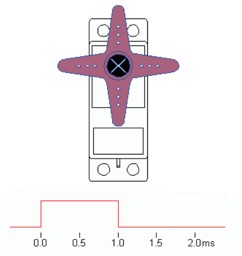


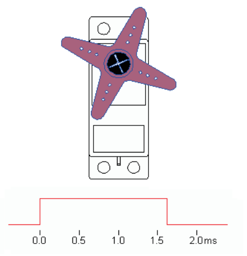
Analógové výstupy: servo





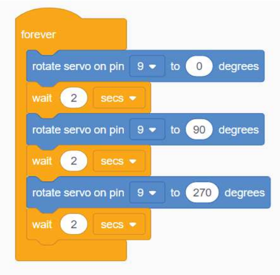
Analógové výstupy: servo





Analógové výstupy: servo

```
#include <Servo.h> // this program uses the Servo library
Servo LeftServo;   // create servo object to control both servos

 void setup()
 {
   LeftServo.attach(9);    // attaches the servo on pin 9 to servo object
 }

  void loop()
{
  LeftServo.write(0);
     delay(2500);
  LeftServo.write(90);
     delay(2500);
  LeftServo.write(270);
     delay(2500);
}
```

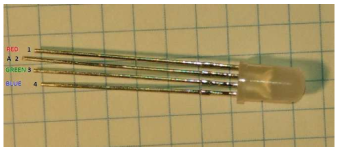


```
forever
  rotate servo on pin 9 ▾ to 0 degrees
  wait 2 secs ▾
  rotate servo on pin 9 ▾ to 90 degrees
  wait 2 secs ▾
  rotate servo on pin 9 ▾ to 270 degrees
  wait 2 secs ▾
```

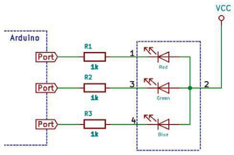
Analógové vstupy: senzor

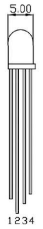
```
void loop()
{
    val = analogRead(SENSOR_1);     // read the value from the sensor

  Serial.print("Sensor = ");
  Serial.println(val, DEC);
  delay(200);

}   /* End of Loop */
```

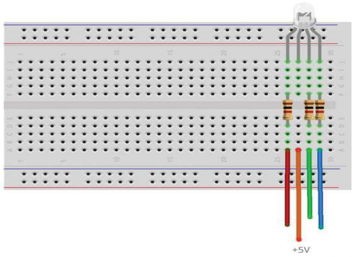
RGB dióda



```
void setup() {
  pinMode(3, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
}

void loop() {
  digitalWrite(3, HIGH);    // turn the LED on
  delay(1000);              // wait for a second
  digitalWrite(3, LOW);     // turn the LED off
  delay(1000);              // wait for a second
}
```







RGB dióda

```
const int REDpin = 6;
const int GRNpin = 3;
const int BLUpin = 5;

void setup()
{
  pinMode(REDpin, OUTPUT);
  pinMode(GRNpin, OUTPUT);
  pinMode(BLUpin, OUTPUT);
}
```

```
void setColor(int R, int G, int B)
{
  analogWrite(REDpin,255-R);
  analogWrite(GRNpin,255-G);
  analogWrite(BLUpin,255-B);
}
```

```
void loop()
{

    // Red=100%, Green=100%, Blue=0 == ORANGE
  digitalWrite(REDpin,LOW);   // Red ON
  digitalWrite(GRNpin,LOW);   // Grn ON
  digitalWrite(BLUpin,HIGH);  // Blu OFF

  delay(1000); // keep the color 1 second

    // HTML color code #D733FF
  analogWrite(REDpin, 0xD7);
  analogWrite(GRNpin, 0x33);
  analogWrite(BLUpin, 0xFF);

  delay(1000); // keep the color 1 second


  setColor(0, 201, 204);

  delay(1000); // keep the color 1 second

}
```