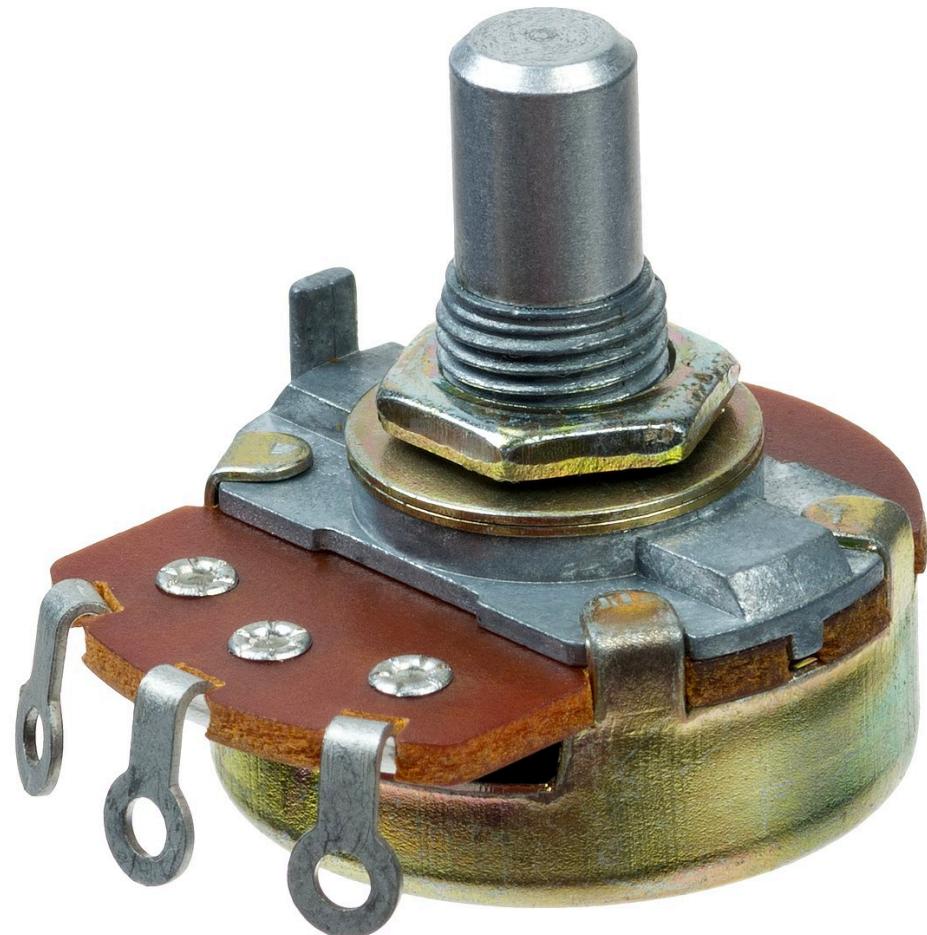
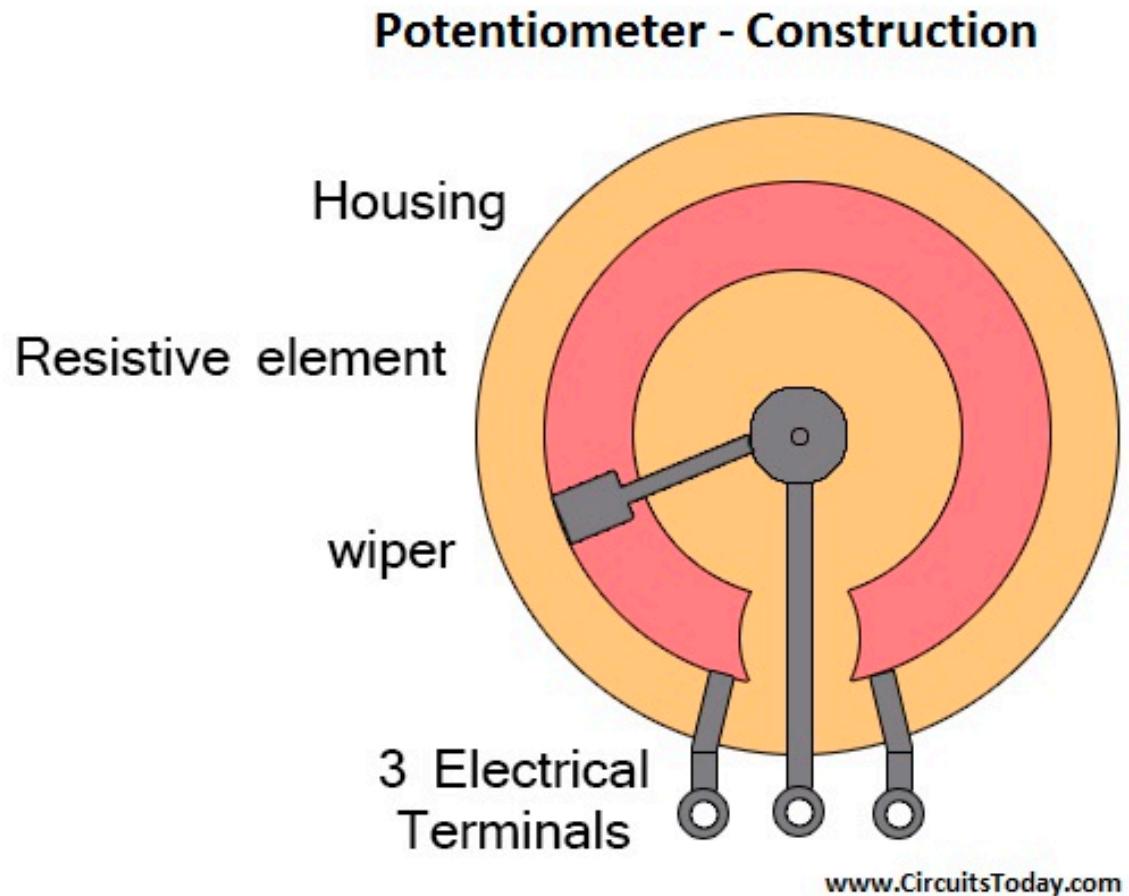


# Arduino Potenciometer & servo

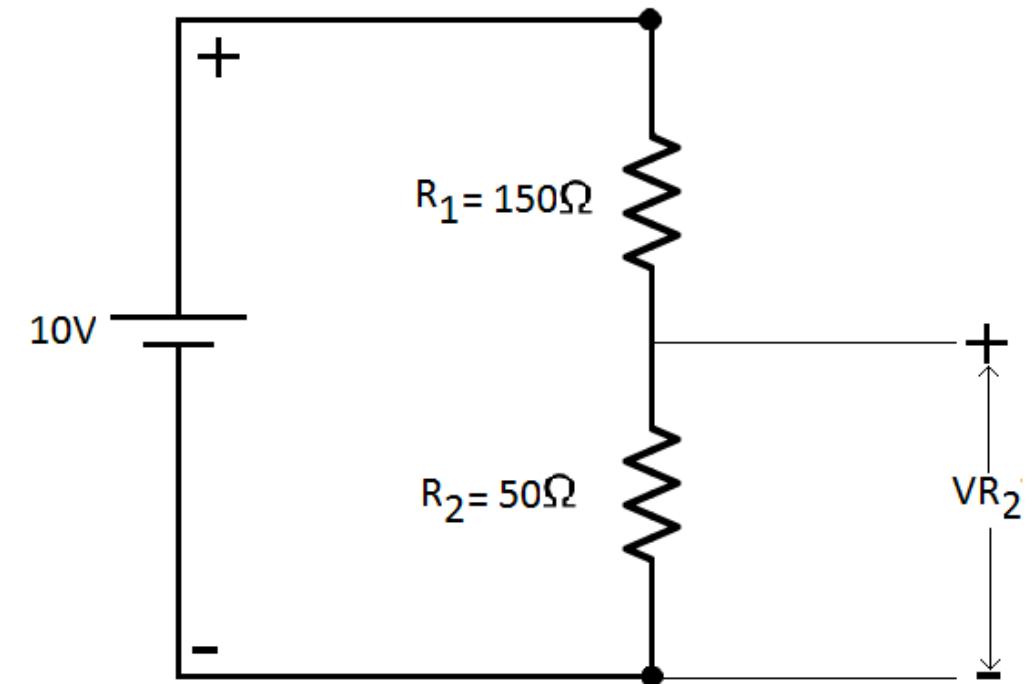
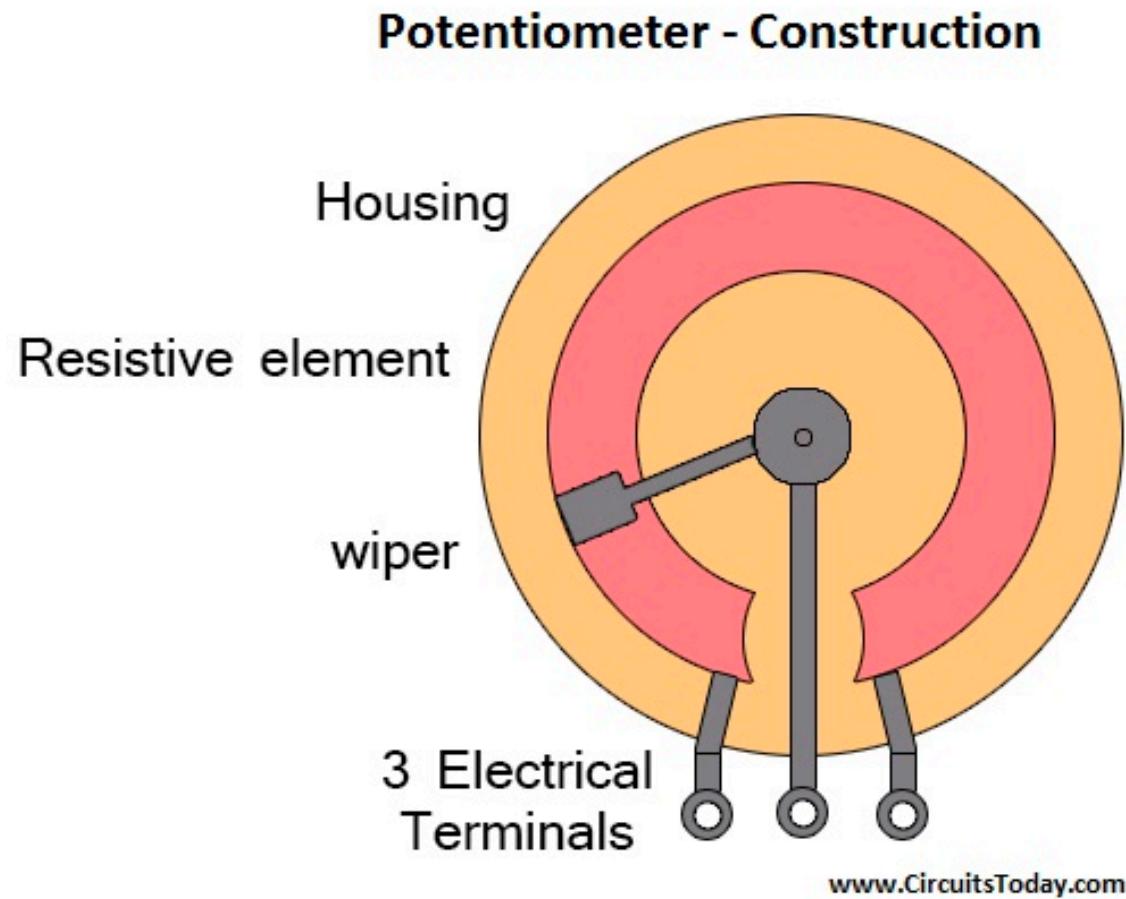
Ing. Gabriel Války, PhD.

<http://L.valky.eu/servo>

# Potenciometer - konštrukcia



# Potenciometer - náhradna schéma

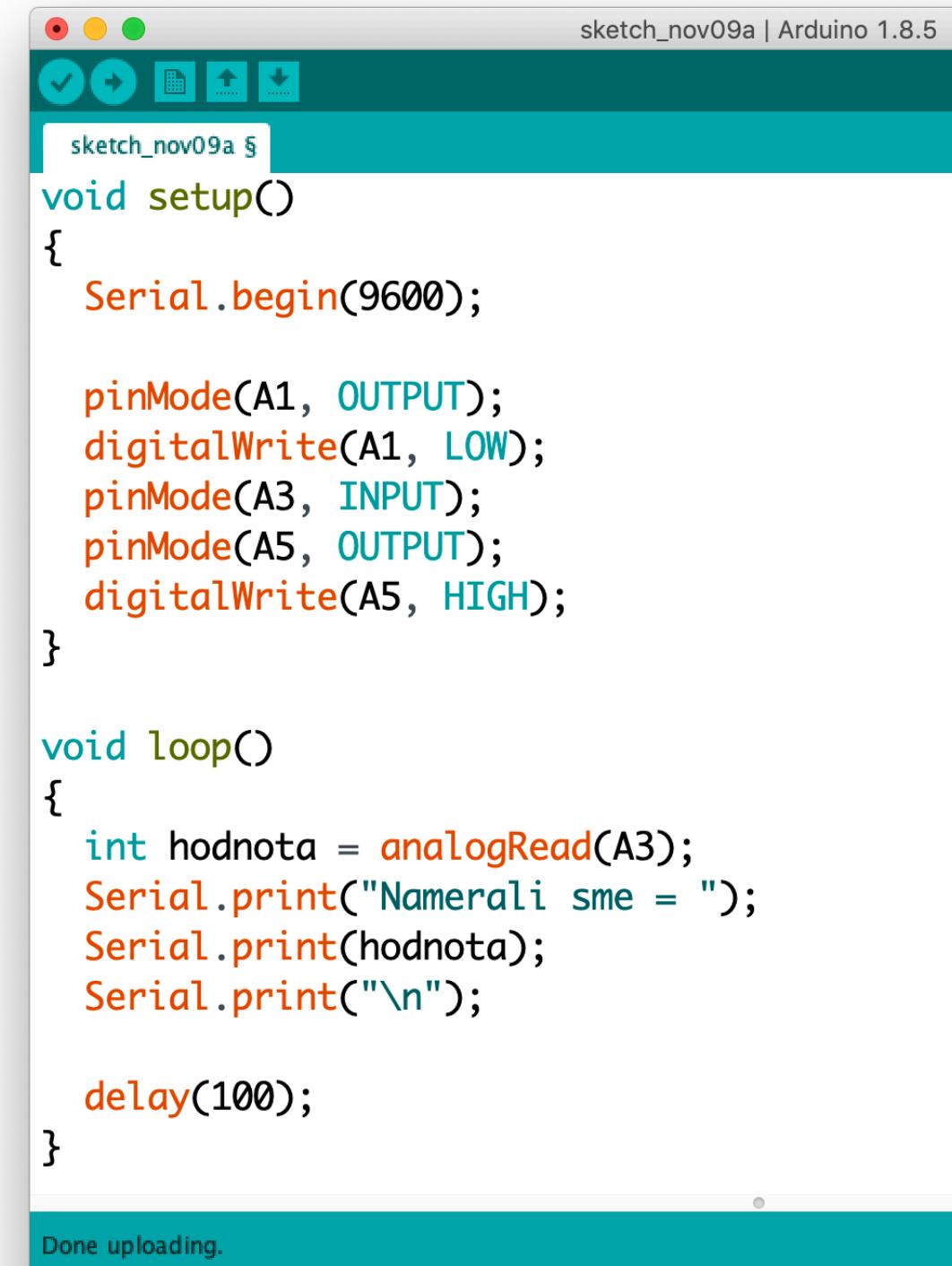


Potentiometer Working

[www.CircuitsToday.com](http://www.CircuitsToday.com)

# U0: Arduino - analogRead

- A5 – 5V
- A4 – nič
- A3 – analogRead
- A2 – nič
- A1 – 0V (GND)
- V akom rozsahu získavame hodnoty?



The screenshot shows the Arduino IDE interface with the title bar "sketch\_nov09a | Arduino 1.8.5". The code editor contains the following sketch:

```
sketch_nov09a §
void setup()
{
    Serial.begin(9600);

    pinMode(A1, OUTPUT);
    digitalWrite(A1, LOW);
    pinMode(A3, INPUT);
    pinMode(A5, OUTPUT);
    digitalWrite(A5, HIGH);
}

void loop()
{
    int hodnota = analogRead(A3);
    Serial.print("Námerali sme = ");
    Serial.print(hodnota);
    Serial.print("\n");

    delay(100);
}
```

The status bar at the bottom right says "Done uploading."

# U1: Arduino - aritmetika

- Serial.println(2/2);
- Serial.println(3/2);
- Serial.println(4/2);
- Serial.println(100\*100);
- Serial.println(300\*300);
- Serial.println(1000\*1000);
- Serial.println(500/1000\*5);
- Serial.println(500\*5/1000);

# U1: Arduino – integrálne typy

int8\_t  
uint8\_t  
int16\_t  
uint16\_t  
int32\_t  
uint32\_t

<b>data type</b>	<b>bytes</b>	<b>min</b>	<b>max</b>
<b>char</b>	<b>1</b>	<b>-128</b>	<b>127</b>
<b>byte</b>	<b>1</b>	<b>0</b>	<b>255</b>
<b>int</b>	<b>2</b>	<b>-32768</b>	<b>32767</b>
<b>unsigned int</b>	<b>2</b>	<b>0</b>	<b>65535</b>
<b>long</b>	<b>4</b>	<b>-2147483648</b>	<b>2147483647</b>
<b>unsigned long</b>	<b>4</b>	<b>0</b>	<b>4294967295</b>

# U1: Arduino - aritmetika

```
Serial.println((long)1000*1000);
```

```
Serial.println(1000L*1000);
```

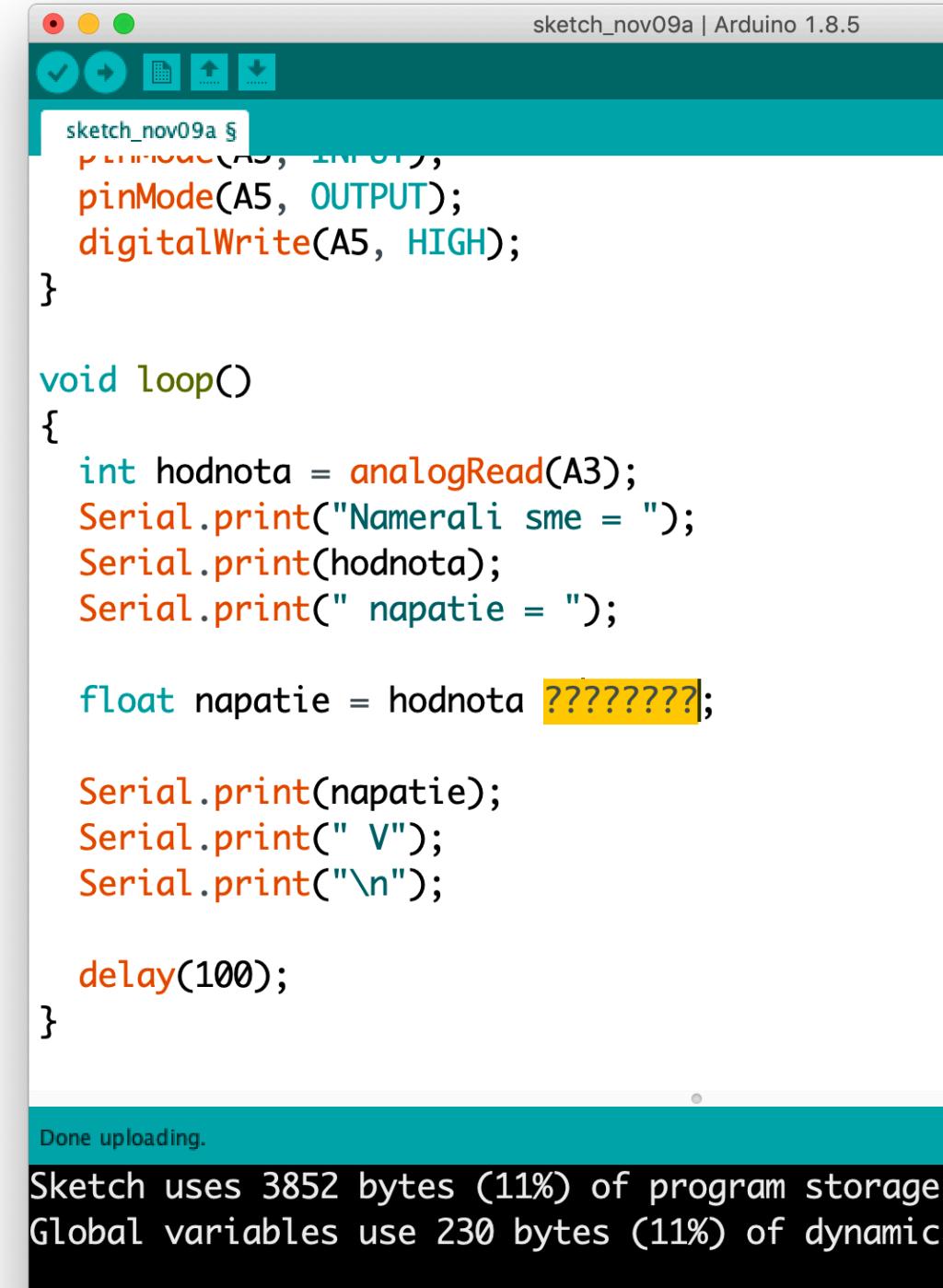
```
Serial.println(1000UL*1000);
```

```
Serial.println((float)500*5/1000);
```

```
Serial.println(500.0f*5/1000);
```

# U1: Arduino - aritmetika

- Vypočítajte hodnotu napäťia z výsledku AD konverzie



```
sketch_nov09a §
pinMode(A3, INPUT);
pinMode(A5, OUTPUT);
digitalWrite(A5, HIGH);

}

void loop()
{
    int hodnota = analogRead(A3);
    Serial.print("Namerali sme = ");
    Serial.print(hodnota);
    Serial.print(" napatie = ");

    float napatie = hodnota ??????????;

    Serial.print(napatie);
    Serial.print(" V");
    Serial.print("\n");

    delay(100);
}
```

Done uploading.

Sketch uses 3852 bytes (11%) of program storage  
Global variables use 230 bytes (11%) of dynamic

## U2: Linearna interpolacia

- Vystup z **analogRead** je v rozsahu 0..1023
- Upravte program tak, aby vypisoval hodnoty v intervale <0, 10>
- Upravte program tak, aby vypisoval hodnoty v intervale <0, 20>
- Upravte program tak, aby vypisoval hodnoty v intervale <10, 30>

# U3: Globálna premenná

- Vypíšte číslo iba vtedy, keď sa zmenilo
- Operátor zhody ==
- Operátor rozdielu !=
- Operátor priradenia =
- Negácia !
- Negácia podmienky **if (!(podmienka))**

```
if (a == b)
{
    // a je zhodne s b
} else
{
    // a nie je zhodne s b
}

if (a != b)
{
    // a nie je zhodne s b
} else
{
    // a je zhodne s b
}
```

# Linearna interpolacia

$$y = y_1 + \frac{x - x_1}{x_2 - x_1} * (y_2 - y_1)$$

- $y = (\text{float})x/1023*20+10$
- $x_1 =$
- $x_2 =$
- $y_1 =$
- $y_2 =$

0...1

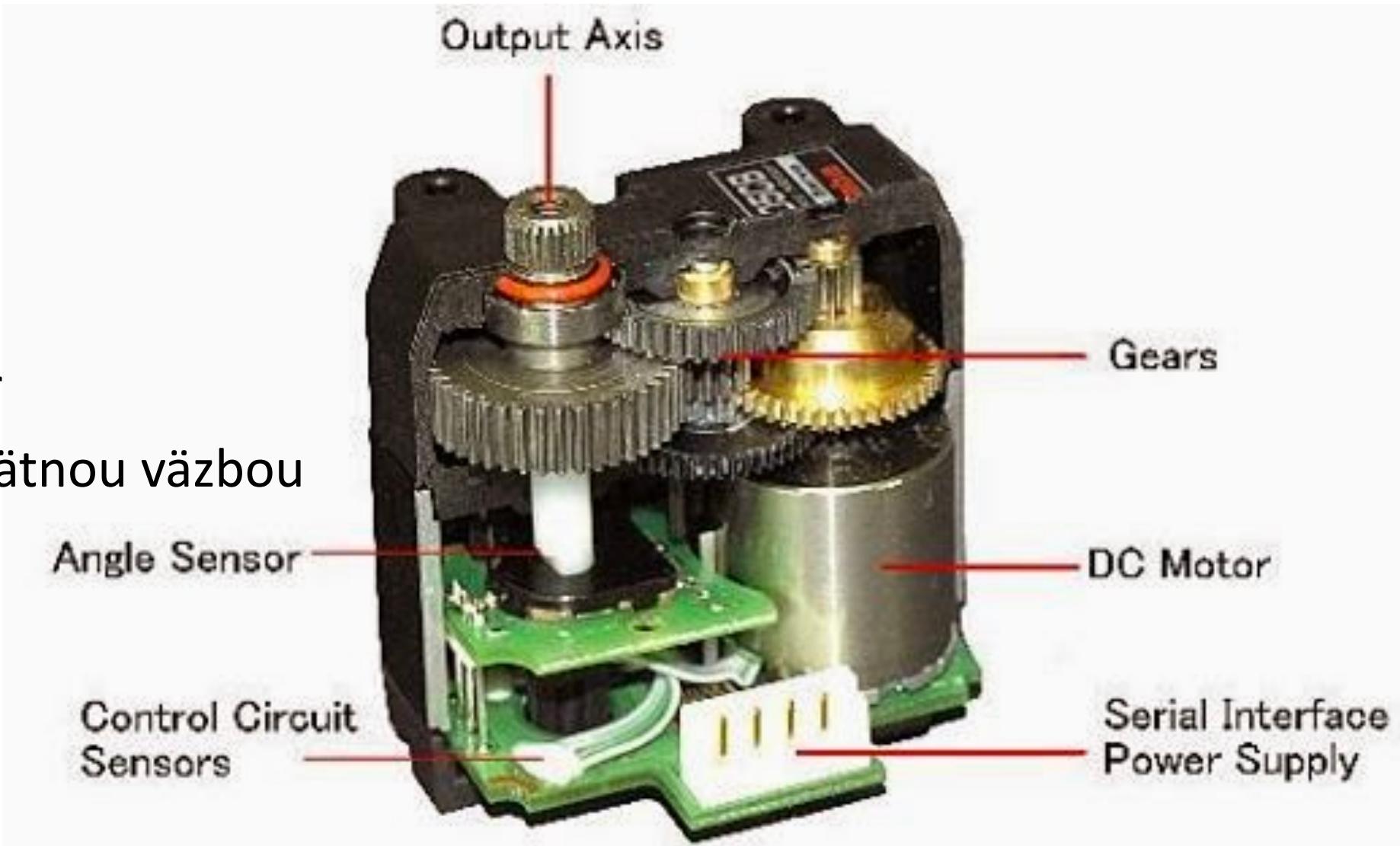
# Linearna interpolacia

$$y = y_1 + \frac{x - x_1}{x_2 - x_1} * (y_2 - y_1)$$

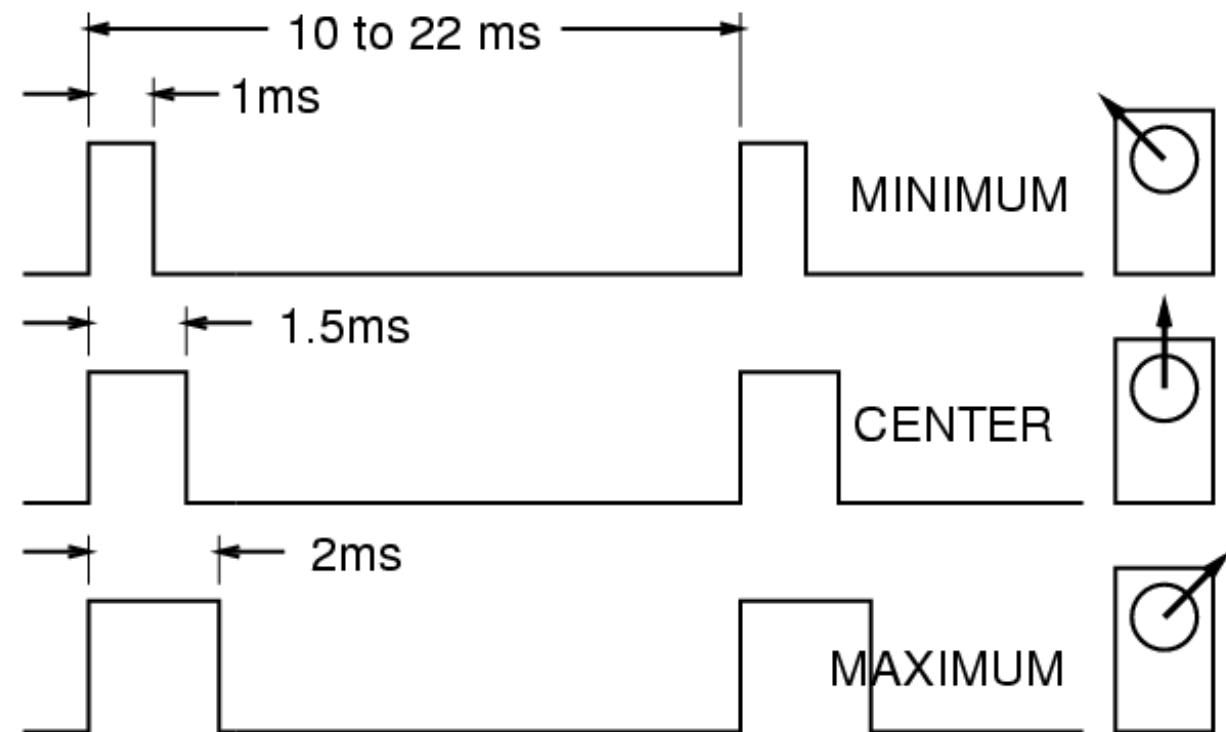
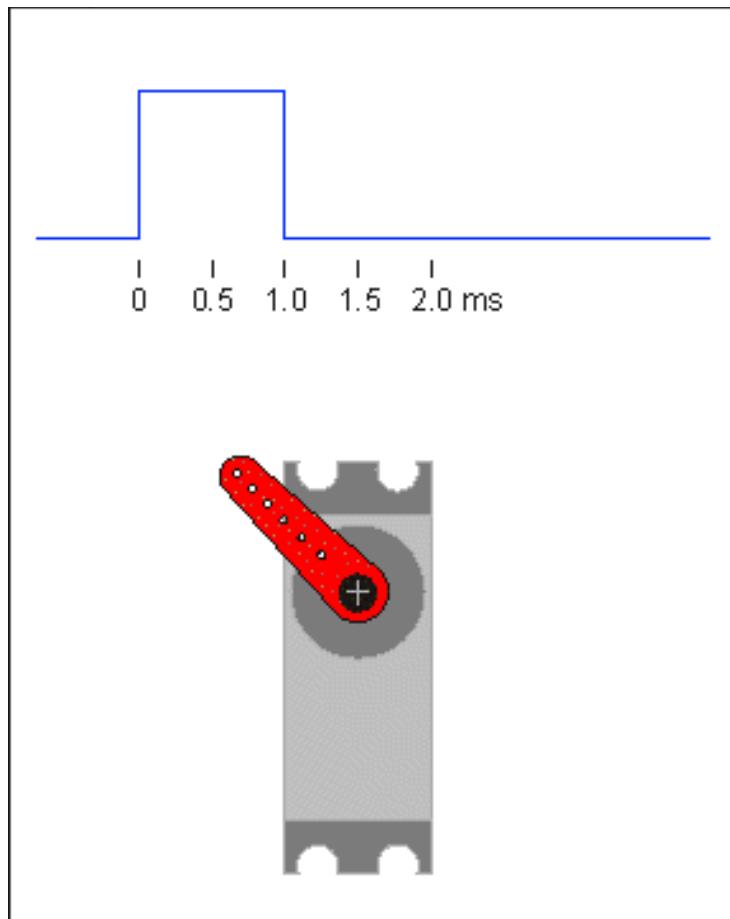
- $y = (\text{float})x/1023*20+10$
- $x_1 = 0$
- $x_2 = 1023$
- $y_1 = 10$
- $y_2 = 20$

# Servo

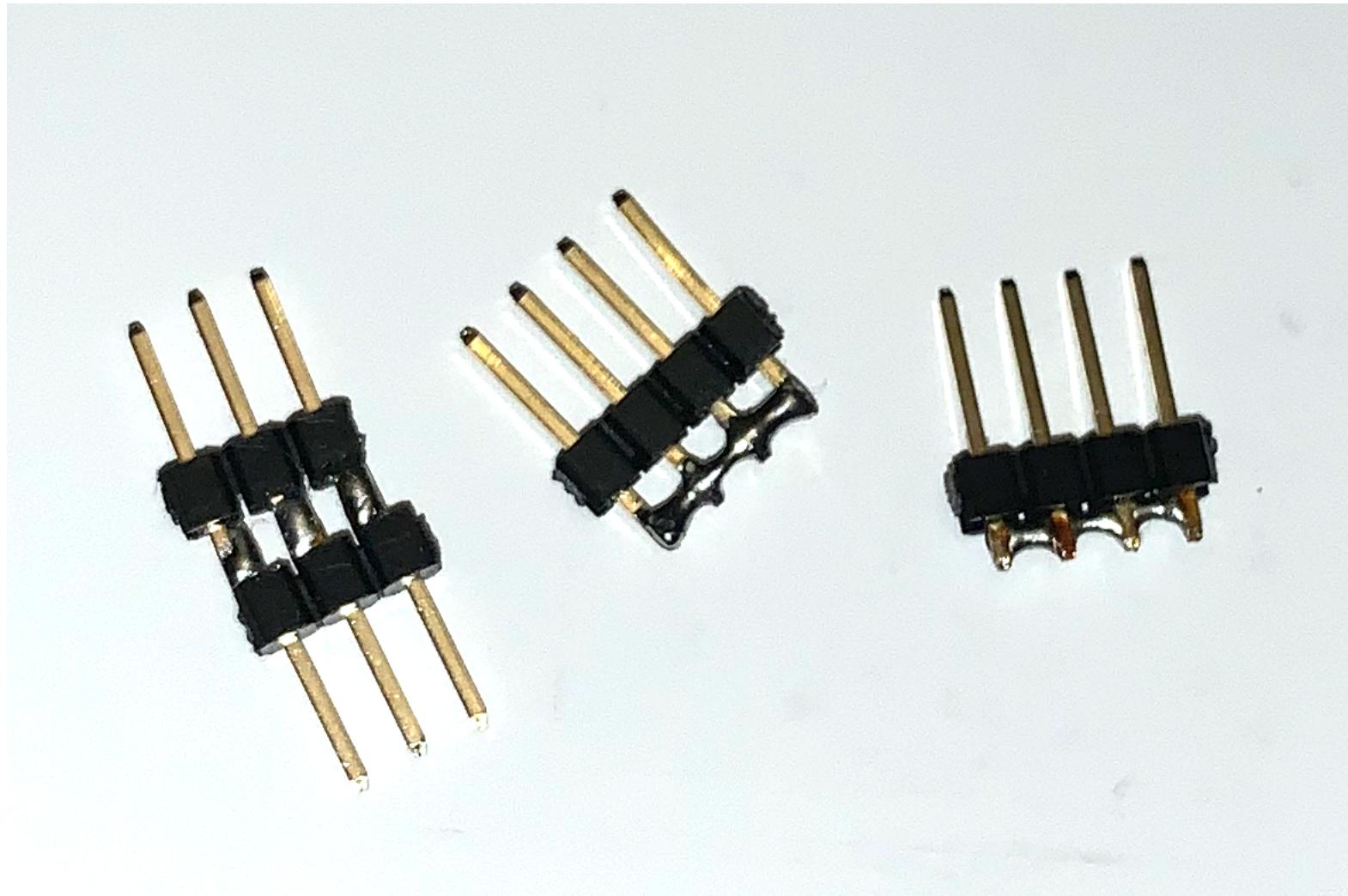
- Motor
- Prevodovka
- Potenciometer
- Riadenie so spätnou väzbou
- Kontrolér



# Servo - riadenie



Spájkujeme



# U4: Servo - arduino

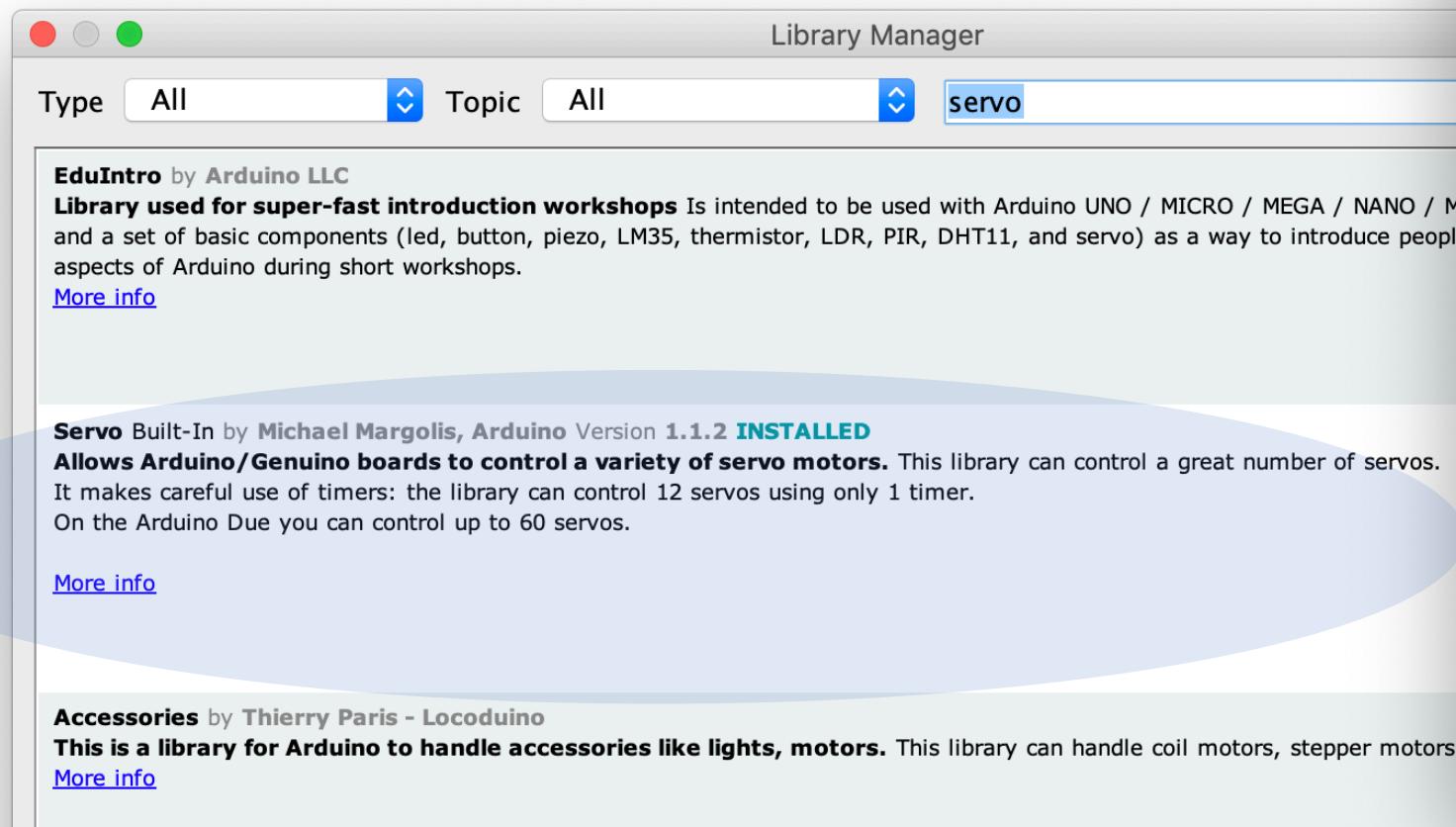
- Červený - 5V, Čierny - GND, Biely – riadiaci signál
  - Nastavte servo do strednej polohy
  - Vychadzajte z príkladu Blink (File -> Examples -> Basics -> Blink)
- 
- `delayMicroseconds(us);`
  - `delay(ms);`
  - `pinMode(A0, OUTPUT);`
  - `digitalWrite(A0, LOW);`      1 s (sekunda) = 1000 ms (milisekúnd)
  - `digitalWrite(A0, HIGH);`     1 ms (milisekunda) = 1000 us (mikrosekúnd)

# U5: Servo + potenciometer

- Ovladájte servo v plnom rozsahu s použitím potenciometra
- Servo – pin A0, GND, 5V
- Potenciometer – piny A1-A3-A5

# U6: Servo + potenciometer

- Použíte knižnicu **Servo**
- Sketch -> Include library -> Manage libraries



The screenshot shows the Arduino IDE with a sketch titled 'sketch\_nov09a'. The code is as follows:

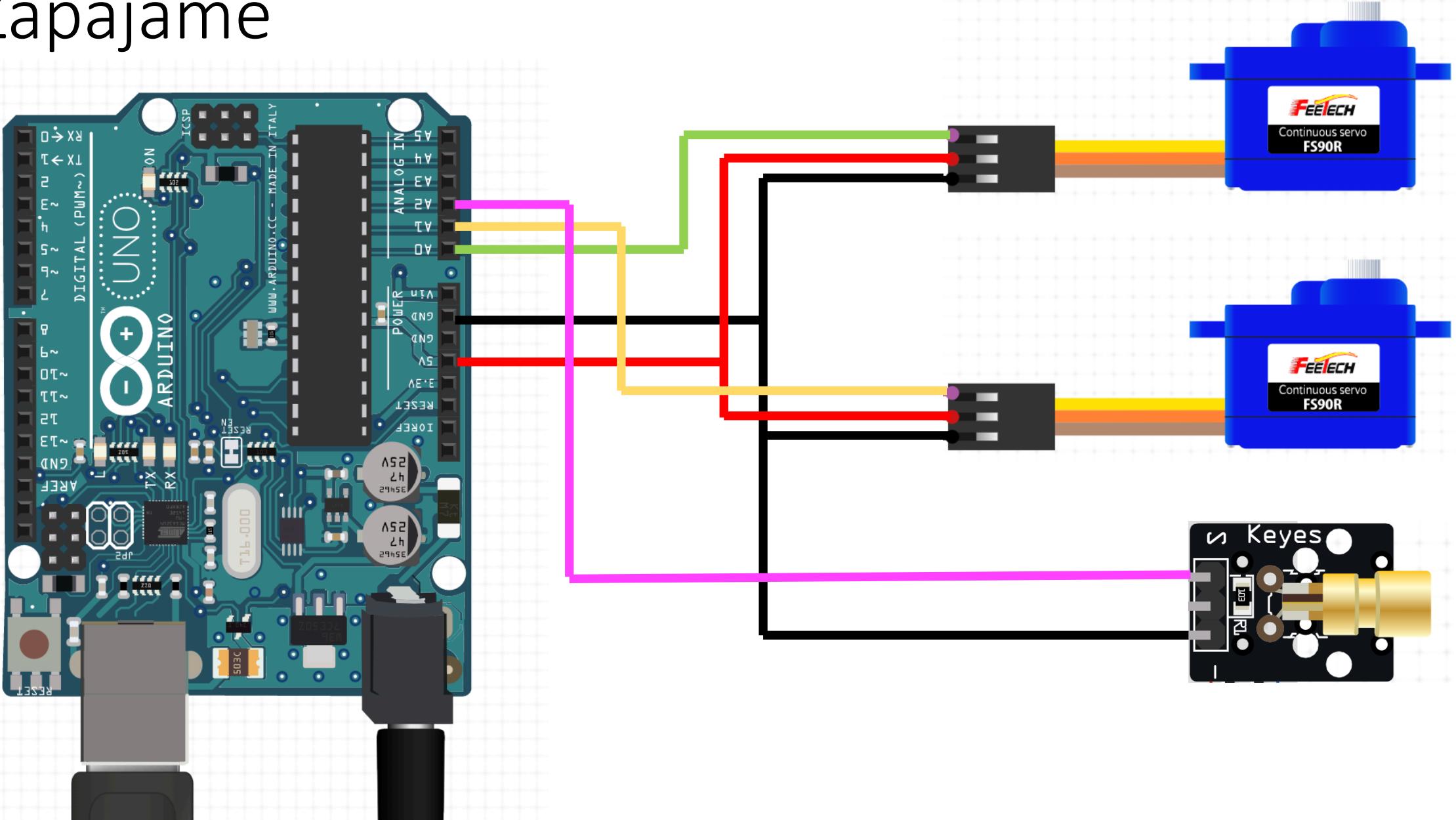
```
#include <Servo.h>
Servo servo;

void setup()
{
    servo.attach(A0);
}

void loop()
{
    servo.write(80); // 0..180
}
```

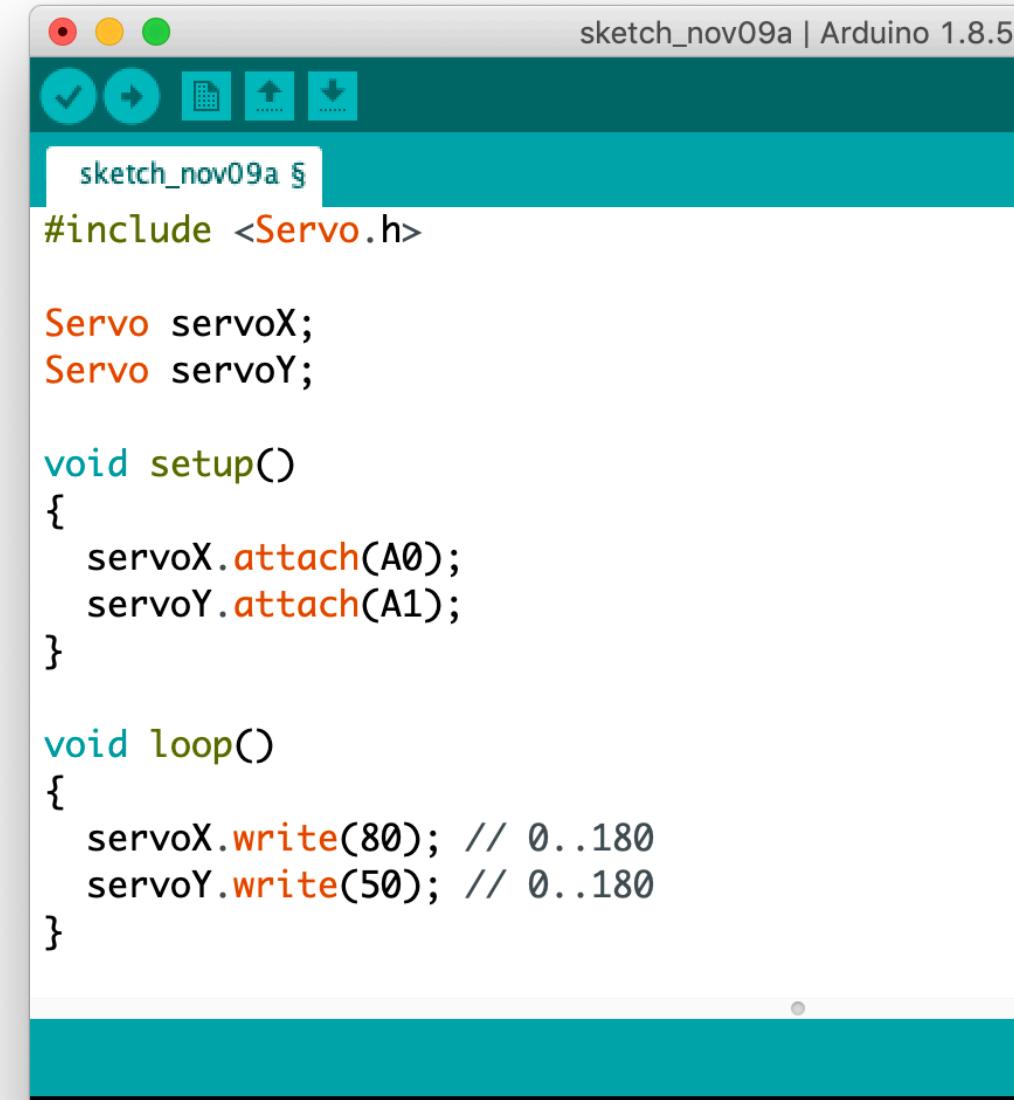
The code includes an include statement for the Servo library, defines a servo object, and sets up an attach call to pin A0. In the loop, it writes a value of 80 to the servo, which corresponds to a range of 0..180.

# Zapájame



# U7: Karteziánsky súradnicový systém

- servoX - horizontálny (vodorovný) pohyb
- servoY – vertikálny (zvisly) pohyb
- Nájdite súradnice rohov papiera A4
- Kreslite obvod papiera, pri každom bode počkajte 1 sekundu
- delay(1000)



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** sketch\_nov09a | Arduino 1.8.5
- Toolbar:** Includes icons for file operations (New, Open, Save, Print, Find, Copy, Paste, Undo, Redo).
- Sketch Name:** sketch\_nov09a §
- Code Area:** Contains the following C++ code for an Arduino sketch:

```
#include <Servo.h>

Servo servoX;
Servo servoY;

void setup()
{
    servoX.attach(A0);
    servoY.attach(A1);
}

void loop()
{
    servoX.write(80); // 0..180
    servoY.write(50); // 0..180
}
```

# U8: Písmeno

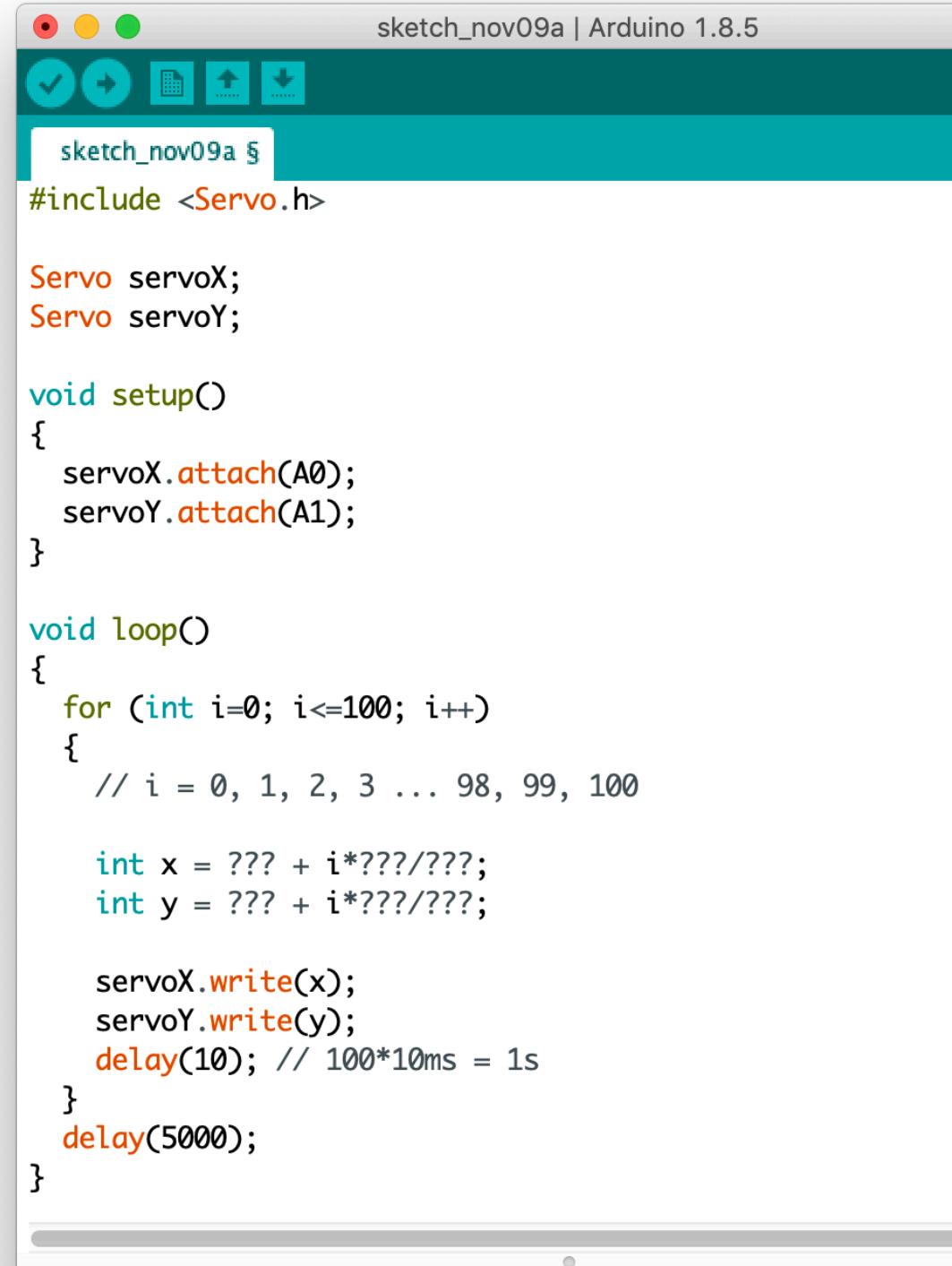
- Pohybujte laserom v tvare písmena I, L, T
- Bonus: F, E, H

# U9: Úsečka

- Pohybujte bodom z ľavého spodného rohu papiera do prvého horného
- Na koncoch úsečky 5 sekúnd čakajte

# U10: Úsečka

- Pohybujte bodom z ľavého spodného rohu papiera do prvého horného
- Na koncoch úsečky 5 sekúnd čakajte
- Parametrické vyjadrenie priamky
- Dĺžka animácie 1s



The screenshot shows the Arduino IDE interface with the title bar "sketch\_nov09a | Arduino 1.8.5". The code editor contains the following C++ code:

```
#include <Servo.h>

Servo servoX;
Servo servoY;

void setup()
{
    servoX.attach(A0);
    servoY.attach(A1);
}

void loop()
{
    for (int i=0; i<=100; i++)
    {
        // i = 0, 1, 2, 3 ... 98, 99, 100
        int x = ??? + i*???/???;
        int y = ??? + i*???/???;

        servoX.write(x);
        servoY.write(y);
        delay(10); // 100*10ms = 1s
    }
    delay(5000);
}
```

# U11: Polyline

- S pomocou funkcie **ciara** nakreslite písmeno M, A, X, Y

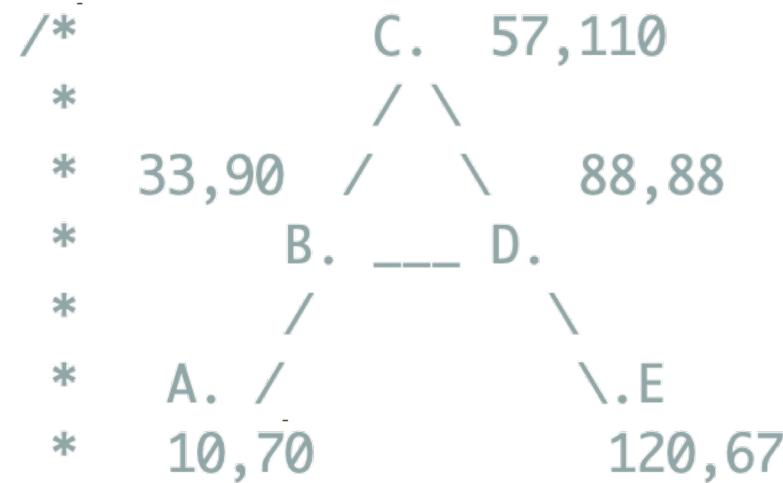
```
void ciara(int x1, int y1, int x2, int y2)
{
    int dlzka = sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
    for (int i=0; i<dlzka; i++)
    {
        int x = x1 + (x2-x1)*i/dlzka;
        int y = y1 + (y2-y1)*i/dlzka;
        servoX.write(x);
        servoY.write(y);
        delay(20);
    }
}
```

# U12: Štruktúry a polia

```
struct bod
{
    int x, y;
};

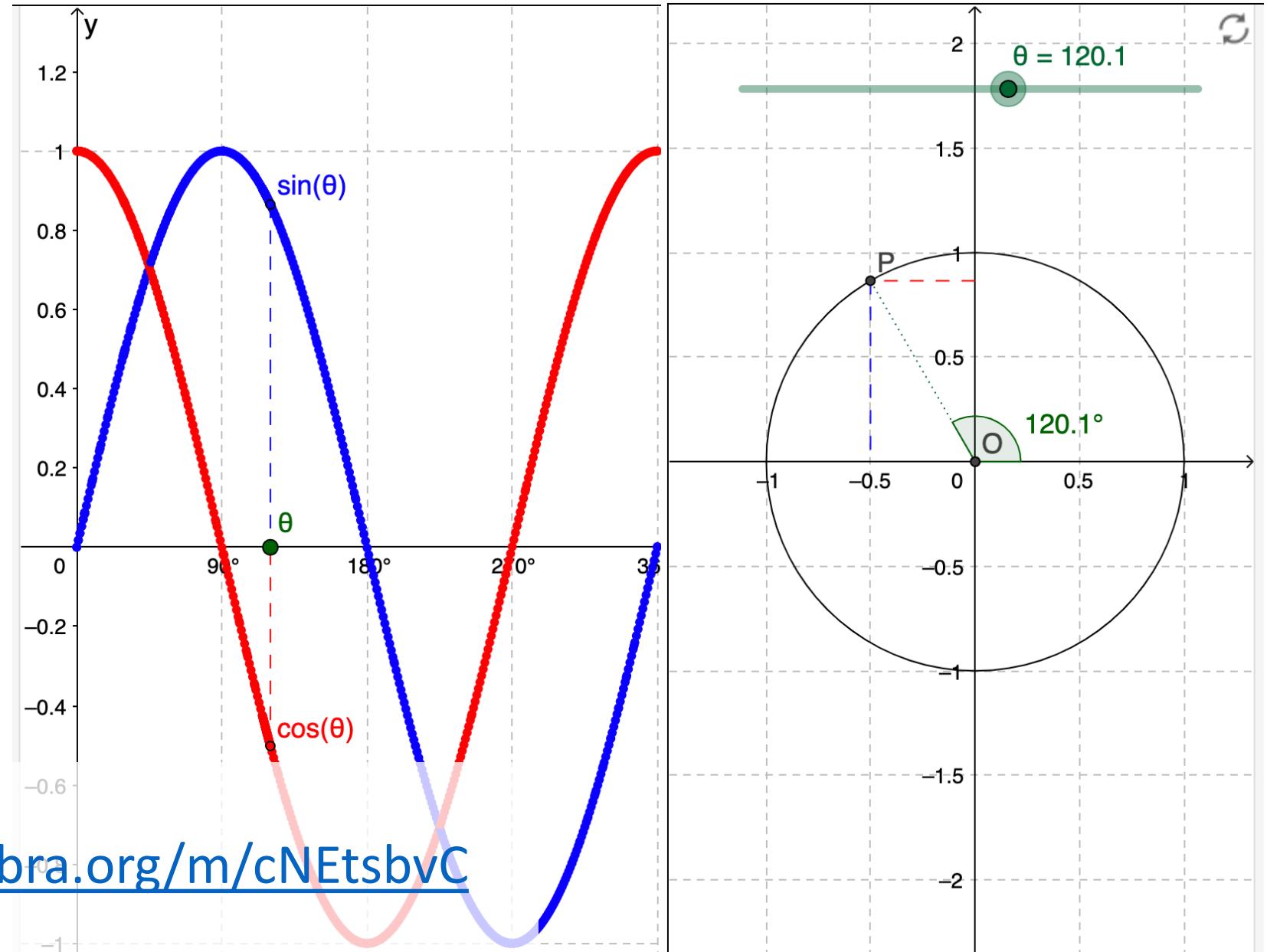
bod A{10, 70}, B{50, 95}, C{65, 110}, D{80, 95}, E{120, 67};*/
bod ciary[] = {A, B, C, D, E, D, B, A};

void loop()
{
    int pocet = sizeof(ciary)/sizeof(ciary[0]);
    for (int i=0; i<pocet-1; i++)
    {
        ciara(ciary[i].x, ciary[i].y, ciary[i+1].x, ciary[i+1].y);
    }
    delay(1000);
}
```



# U13: Kruh

- Funkcie cos, sin
- Argument v rad
- $2\pi = 360^\circ$
- <https://www.geogebra.org/m/cNEtsbvC>



# U13: Kruh

- Nakreslite písmeno O, C
- Bonus: Q, G

```
for (int uhol=0; uhol<360; uhol++)  
{  
    float uholRad = uhol*PI/180.0f;  
    int x = 100+cos(uholRad)*50;  
    int y = 100+sin(uholRad)*50;  
}
```

# Ďakujem

Ing. Gabriel Války, PhD.

<http://L.valky.eu/servo>