

# Arduino Potenciometer & servo

Ing. Gabriel Valky, PhD.

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

# Potenciometer

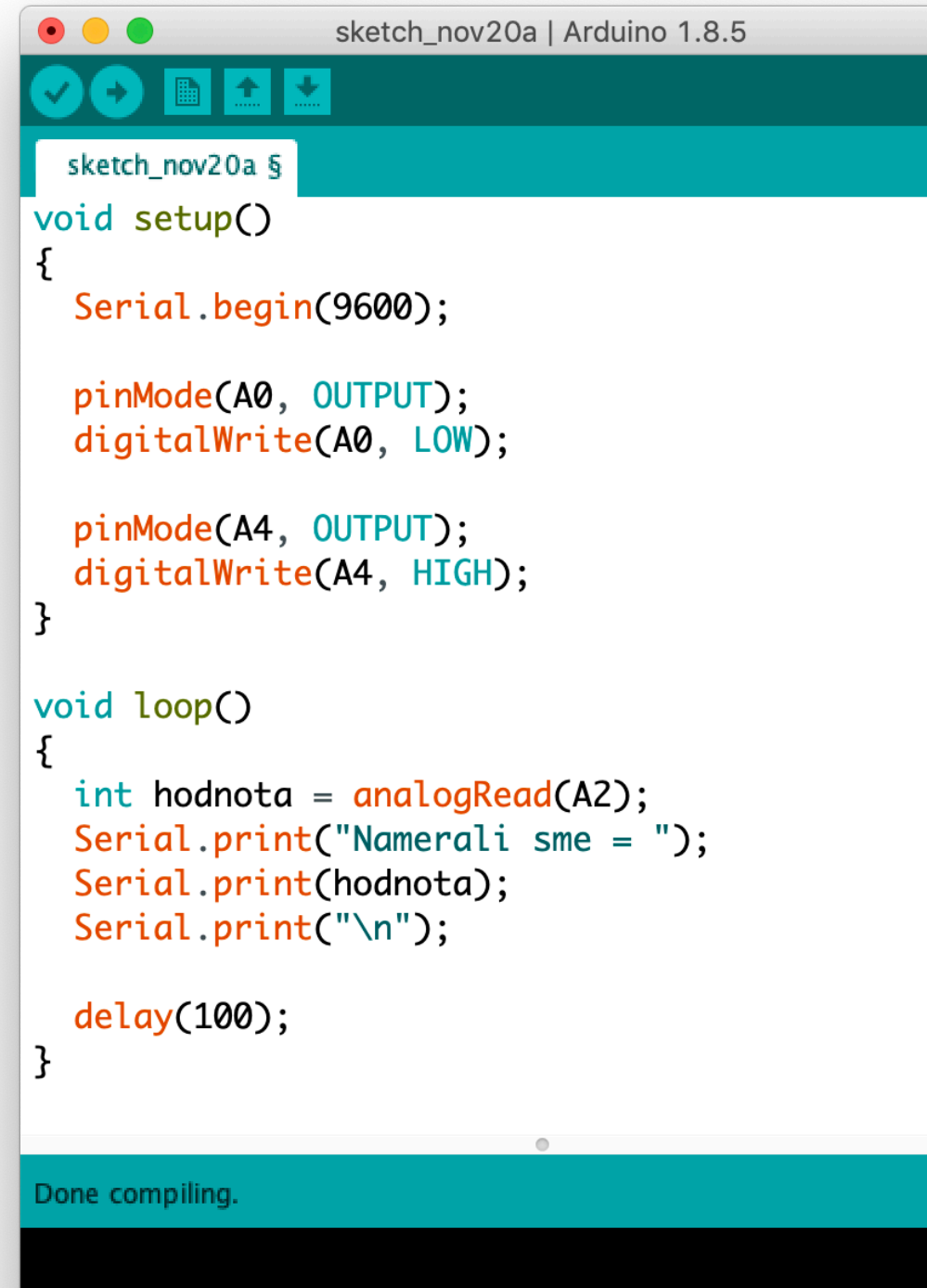
- Škonstrukcia

# Potenciometer - nahradna schema

- \$SCHEMA
- \$OBRAZOK ako priletujeme pinheader

# U0: Arduino - analogRead

- A0 – 5V
  - A1 – nic
  - A2 – analogRead
  - A3 – nic
  - A4 – 0V (GND)
- 
- V akom rozsahu ziskavame hodnoty?



```
sketch_nov20a | Arduino 1.8.5

void setup()
{
  Serial.begin(9600);

  pinMode(A0, OUTPUT);
  digitalWrite(A0, LOW);

  pinMode(A4, OUTPUT);
  digitalWrite(A4, HIGH);
}

void loop()
{
  int hodnota = analogRead(A2);
  Serial.print("Namerali sme = ");
  Serial.print(hodnota);
  Serial.print("\n");

  delay(100);
}
```

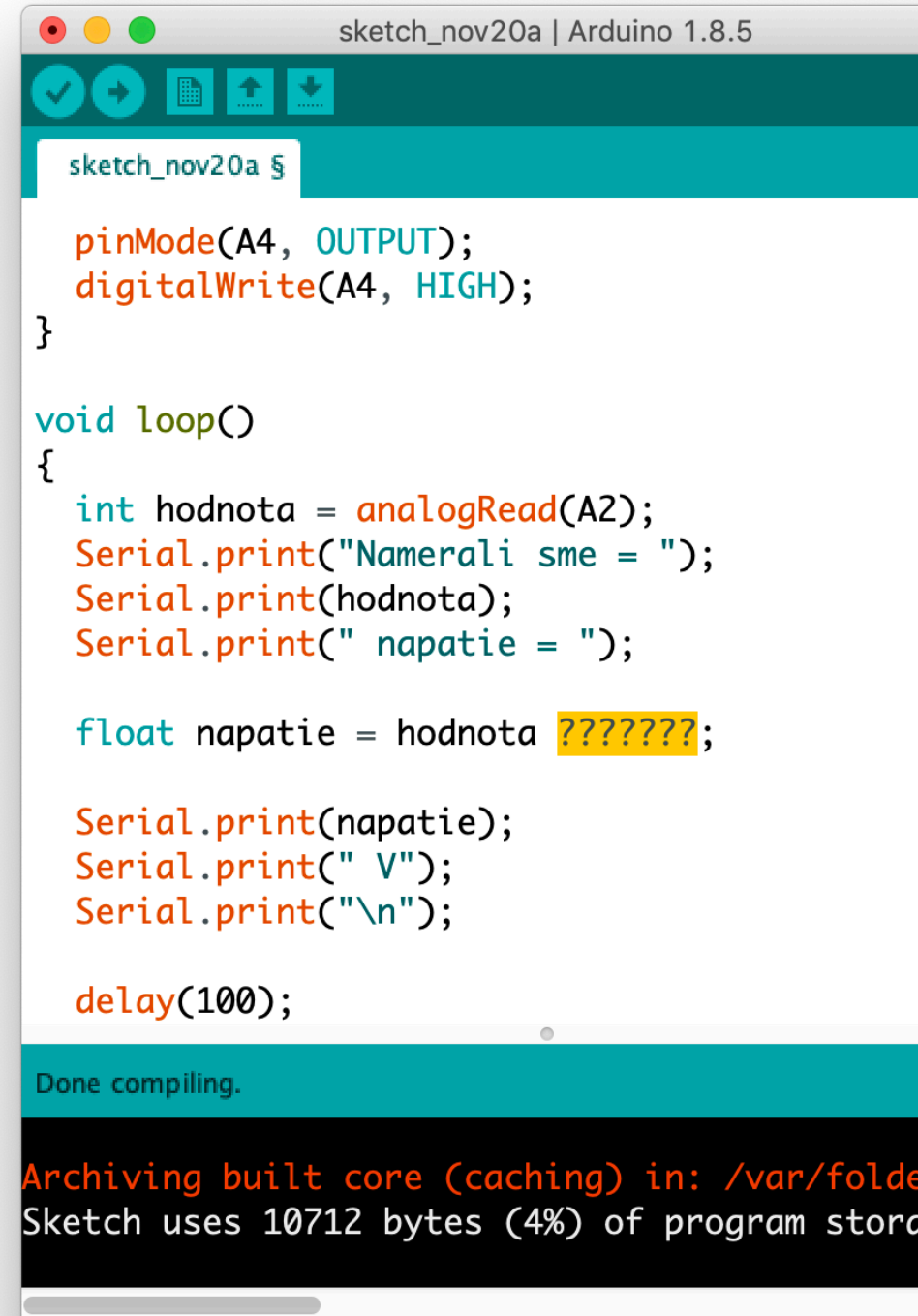
Done compiling.

# 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);`
- \$SKONTROLOVAT, TABULKU TYPOV?

# U1: Arduino - aritmetika

- Vypocitajte hodnotu napatia z vysledku AD konverzie



```
sketch_nov20a | Arduino 1.8.5

sketch_nov20a §

pinMode(A4, OUTPUT);
digitalWrite(A4, HIGH);
}

void loop()
{
  int hodnota = analogRead(A2);
  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 compiling.

Archiving built core (caching) in: /var/folder
Sketch uses 10712 bytes (4%) of program storage.
```

## U2: Linearna interpolacia

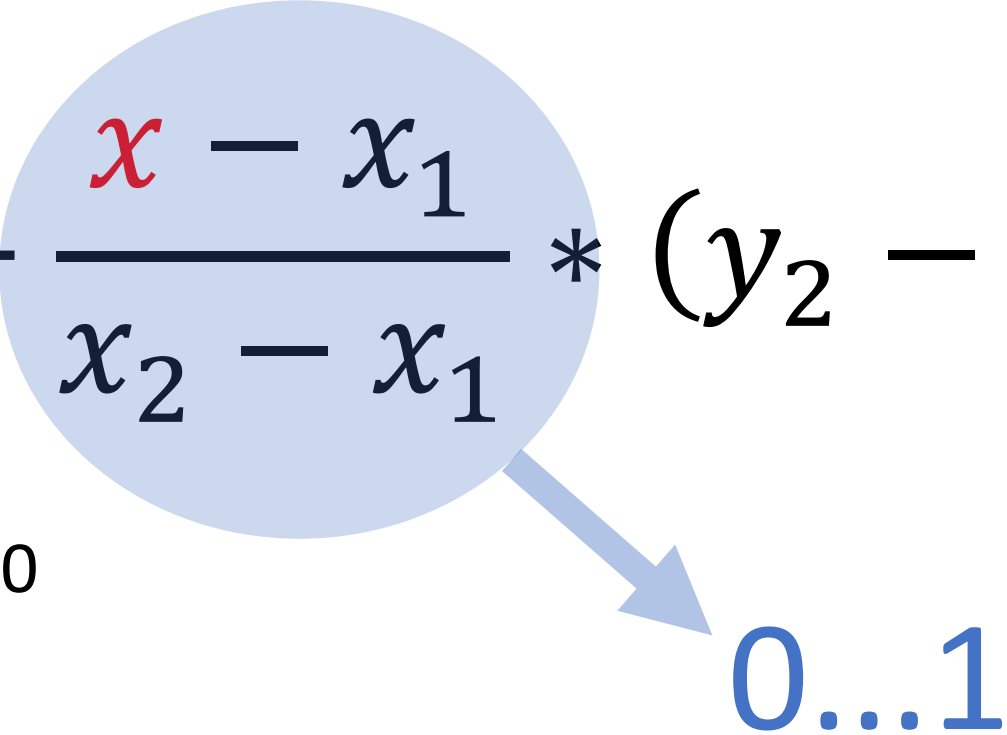
- Upravte program tak, aby vypisoval hodnoty v intervale  $\langle 10, 30 \rangle$

## U3: Globalna premenna

- Vypiste cislo iba vtedy, ked sa zmenilo



# 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

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

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

# Servo

- Motor
- Prevodovka
- Potenciometer
- Riadenie so spatnou vazbou
- Kontroler
- \$PRIDAT OBRAZOK

# Servo - riadenie

- \$CASOVACI DIAGRAM/ANIMACIA

# U4: Servo - arduino

- Nastavte servo do strednej polohy
- Vychadzajte z prikladu Blink (File -> Examples -> Basics -> Blink)
- `delayMicroseconds(us);`
- `delay(ms);`
- `pinMode(A0, OUTPUT);`
- `digitalWrite(A0, LOW);`
- `digitalWrite(A0, HIGH);`

1 s (sekunda) = 1000 ms (milisekund)

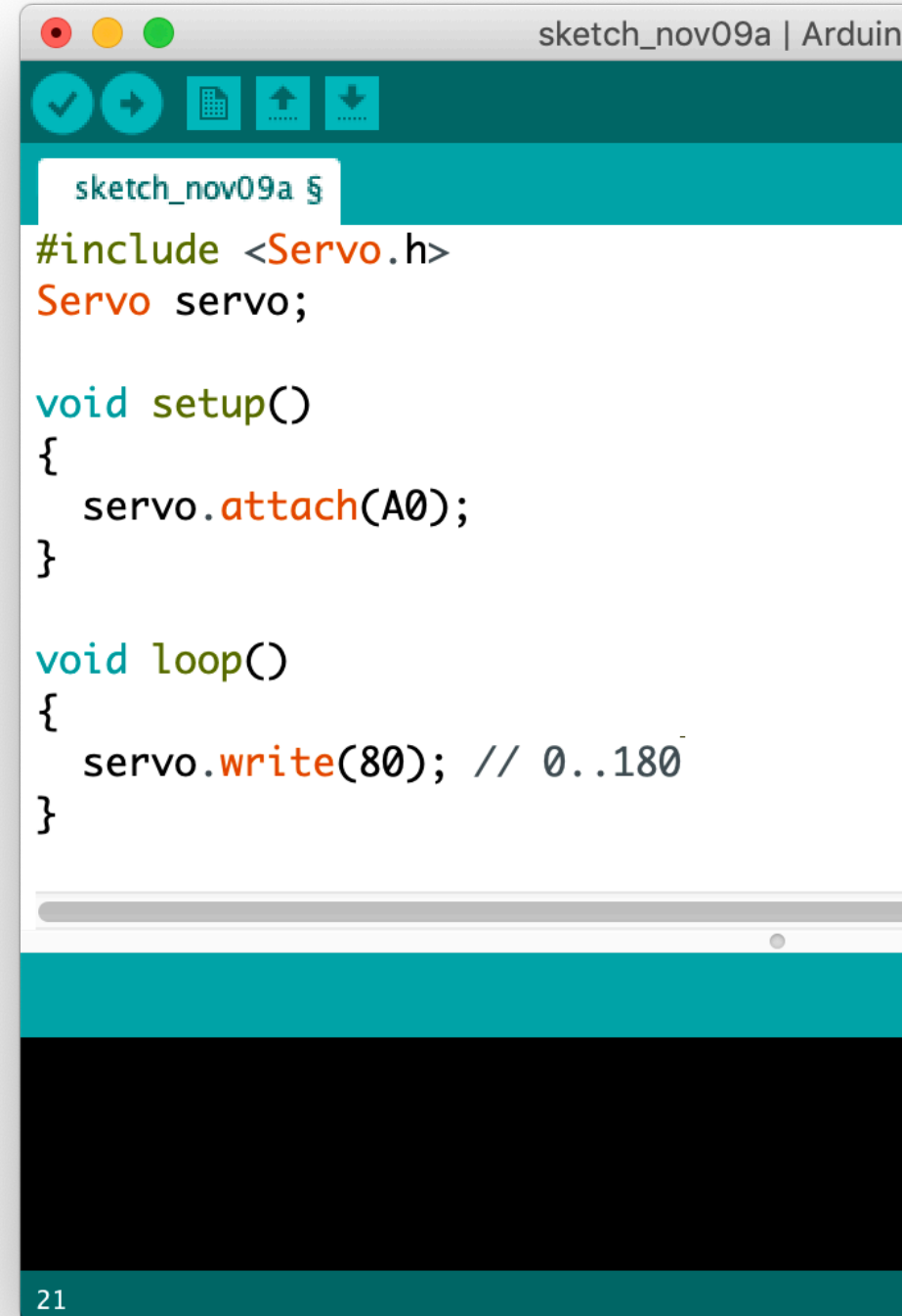
1 ms (milisekunda) = 1000 us (mikrosekund)

# U5: Servo + potenciometer

- Ovladajte servo v plnom rozsahu s použitím potenciometra
- Servo – pin A5, GND, 5V
- Potenciometer – piny A0-A2-A4

# U6: Servo + potenciometer

- Pouzite kniznicu **Servo**

A screenshot of the Arduino IDE interface. The title bar at the top reads "sketch\_nov09a | Arduino". Below the title bar is a toolbar with icons for checking, running, and saving. The main text area contains the following C++ code:

```
sketch_nov09a §  
#include <Servo.h>  
Servo servo;  
  
void setup()  
{  
  servo.attach(A0);  
}  
  
void loop()  
{  
  servo.write(80); // 0..180  
}
```

The code is color-coded: keywords are blue, comments are grey, and function names/variables are black. The bottom status bar shows the number "21".

```
sketch_nov09a | Arduino  
✓ → [Icons]  
sketch_nov09a §  
#include <Servo.h>  
Servo servo;  
  
void setup()  
{  
  servo.attach(A0);  
}  
  
void loop()  
{  
  servo.write(80); // 0..180  
}  
21
```

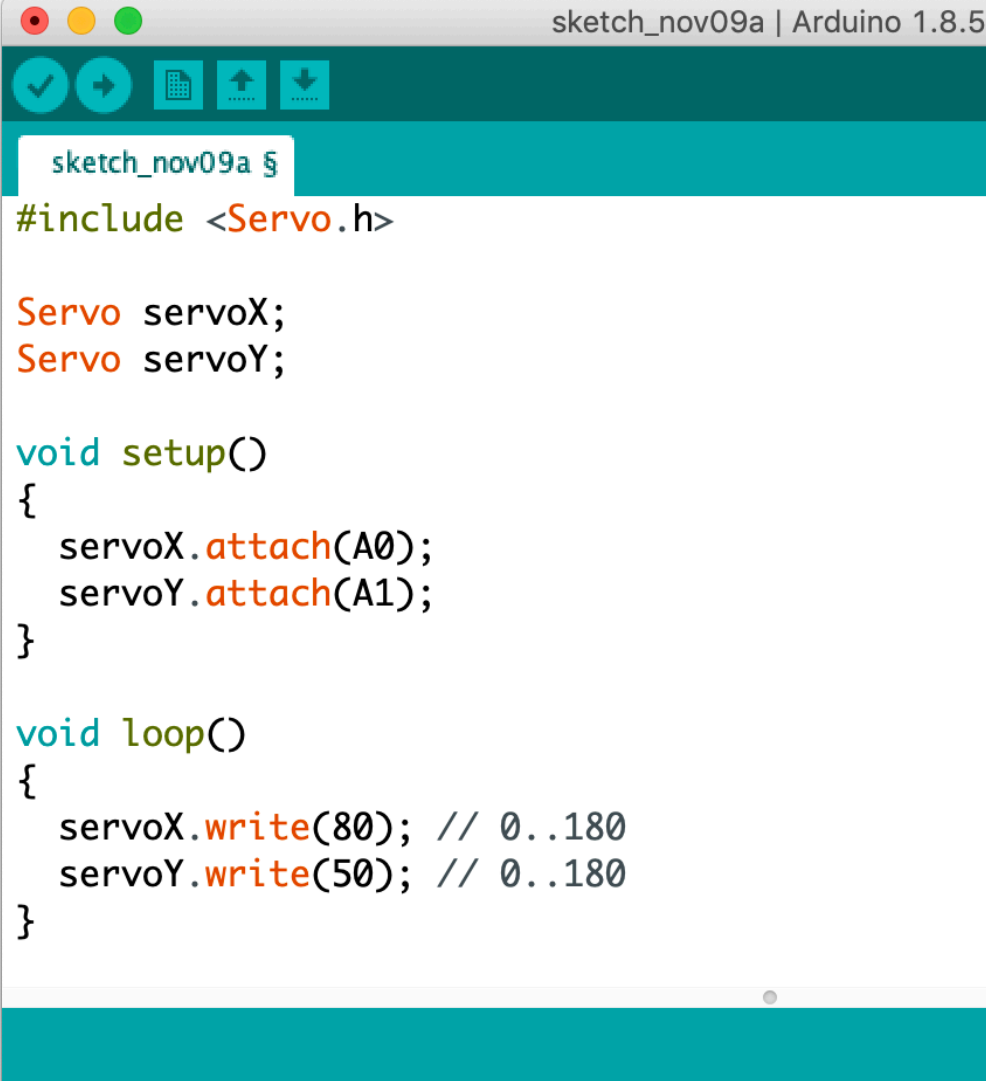
# Spajkujeme

- \$NAKRESLIT SCHEMU
- \$ODFOTIT SPOJKY



# U7: Kartezijský súradnicový systém

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



```
sketch_nov09a | Arduino 1.8.5

sketch_nov09a §
#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: Pismo

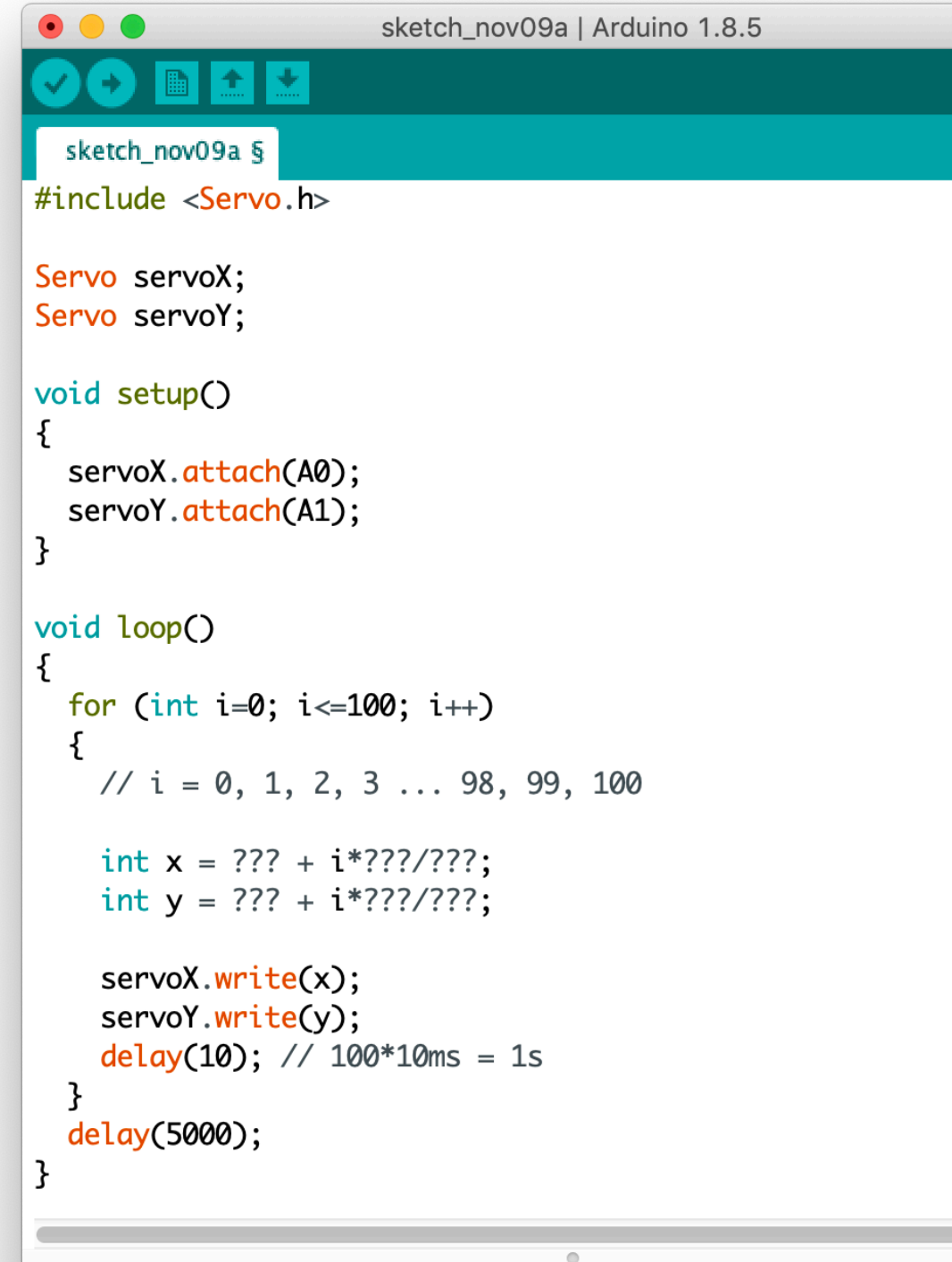
- Pohybujte laserom v tvare pismena I, L, T
- Bonus: E, H

# U9: Usecka

- Pohybujte bodom z laveho spodneho rohu papiera do praveho horneho
- Na konci pohybu 5 sekund cakajte

# U10: Usecka

- Pohybujte bodom z laveho spodneho rohu papiera do praveho horneho
- Na konci pohybu 5 sekund cakajte
- Parametricke vyjadrenie priamky
- Dlzka animacie 1s



```
sketch_nov09a | Arduino 1.8.5
sketch_nov09a §
#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 pismeno M

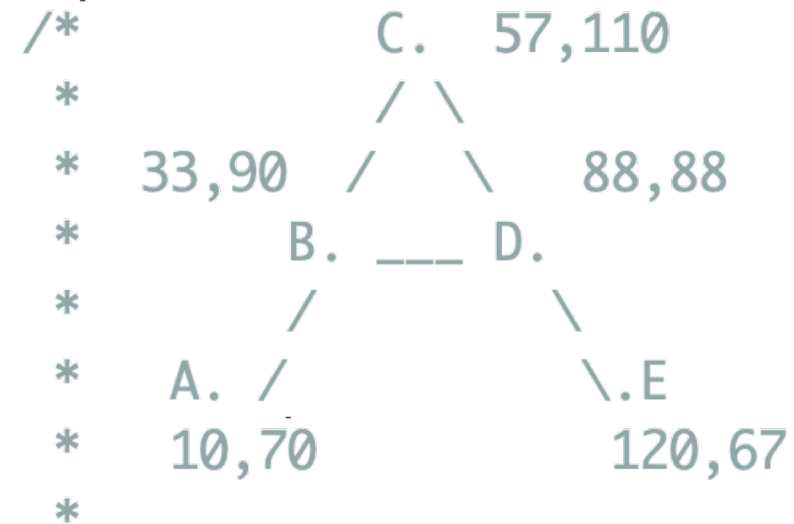
```
void ciara(int x1, int y1, int x2, int y2)
{
    int dlzka = sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
    for (int i=0; i<dlzka; i++)
    {
        int x = x1 + (x2-x1)*i/dlzka;
        int y = y1 + (y2-y1)*i/dlzka;
        servoX.write(x);
        servoY.write(y);
        delay(20);
    }
}
```

# U12: struktury 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 radianoch,  $2\pi = 360$ stupnov
- ŠOBRAZOK

# U13: Kruh

- Nakreslite pismeno 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;  
}
```



# Dakujem

Ing. Gabriel Valky, PhD.

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