

**NSWI170: Počítačové systémy**

<http://www.ksi.mff.cuni.cz/~svoboda/courses/212-NSWI170/>

Cvičení 2

# **Arduino: LED**

**Martin Svoboda**

[martin.svoboda@matfyz.cuni.cz](mailto:martin.svoboda@matfyz.cuni.cz)

1. a 8. 3. 2022

**Univerzita Karlova, Matematicko-fyzikální fakulta**

# Arduino

## Platforma Arduino

- **Arduino UNO**
  - Základní deska, 14 digitálních a 6 analogových pinů
  - CPU ATmega328P, 16 MHz, FLASH memory 32 kB
- **Funduino**
  - Multifunction shield
  - Tlačítka, LED, segmentový displej, ...
- Dokumentace
  - <https://www.arduino.cc/en/reference/>
  - <http://kabinet.fyzika.net/dilna/ARDUINO/funduino-popis.php>

# Arduino

## Arduino IDE

- Ovládání
  - CTRL+S: uložení souboru (přípona \*.ino)
  - CTRL+R: kompilace programu
  - CTRL+U: upload do Arduina

## Struktura programu

- Funkce `void setup()` ;
  - Spustí se jednou při startu
  - Nastavení módů pinů (vstupní, výstupní)
  - Inicializace hodnot pinů a globálních proměnných
- Funkce `void loop()` ;
  - Vlastní výkonný kód
  - Spouští se přibližně 1000x za sekundu

# P1: Rozsvícení diody

## Rozsviďte konkrétní vybranou diodu

- Nejprve připojte soubor s definicemi potřebných konstant
  - `#include "funshield.h"`
  - <https://www.ksi.mff.cuni.cz/teaching/nswi170-web/downloads/Funshield.zip>
  - Archiv rozbalte a samotný hlavičkový soubor `funshield.h` umístěte do stejného adresáře jako program
- **Číslo portů diod**
  - Konstanty `led1_pin`, `led2_pin`, ...
    - Fakticky jde o čísla 13, 12, 11, 10 pro diody D1 až D4
  - Definujte pole `constexpr int led_pins[] = { led1_pin, led2_pin, led3_pin, led4_pin };`
  - V programu používejte **vlastní logická čísla diod 0 až 3**
    - Ta interně překládejte na čísla portů pomocí `led_pins`

# P1: Rozsvícení diody

Pokračování...

- **Inicializace programu**
  - `pinMode(pin, OUTPUT);`
  - Vypnutí všech diod
- **Ovládání diody**
  - Zápis LOW (zapnutí) / HIGH (vypnutí) na příslušný pin
  - `digitalWrite(pin, value);`
- Logické číslo diody předejte jako parametr vytvořené funkce
  - Při jejím volání použijte natvrdo jednu vybranou diodu

# P2: Blikající dioda

## Rozblikujte konkrétní vybranou diodu

- Parametry programu
  - Číslo diody opět natvrdo v kódu, např. 3
  - Délka intervalu jako konstanta, např. 300 ms
- **Kontrola časování**
  - `unsigned long now = millis();`
    - Vrátí aktuální systémový čas v milisekundách
    - Hodnota přeteče po přibližně 50 dnech (nemusí nás to trápit)
  - Naplánování času první / další události
    - Zapamatování si tohoto času v globální proměnné
  - Prohození stavu zapnutí v daný okamžik
    - `if (current_time >= next_time) { ... }`
    - Zapamatování si logického stavu v další globální proměnné

# P3: Železniční semafor

## Rozblikujte železniční semafor

- Aneb dvojice sousedních diod střídavě svítí a druhá ne

# P4: Binární rozklad

## Pomocí diod zobrazujte hodnotu inkrementovaného počítadla

- Počítadlo začíná na 0 a v každém kroku se inkrementuje o 1
- Pomocí diod vždy zobrazíte nejnižší 4 bity aktuálního čísla
  - Bit 1 diodu rozsvítí, bit 0 naopak zhasne
  - Např. pro číslo  $53_{10} = 110101_2$  zobrazíte 0101
    - Aneb dioda 0 nesvítí, 1 ano, 2 ne, 3 ano
- Menší nápověda
  - $x \& 1$
  - $x \ll 1$



# P5: Běhající kulička

## Naprogramujte běhající kuličku

- Aneb postupně po jedné rozsvěcujete diody 0, 1, 2 a 3 a následně opět od 0 dál

# R2: Odrážející se kulička

## **Domácí úkol** (zadání v ReCodExu)

- Kulička běží jedním směrem, na konci se odrazí a pokračuje směrem opačným