

NSWI170 – Počítačové systémy

Tomáš Faltín

Příklad DU1 & DU2

Základní typy v C++

- Příznaky – `bool`
- Čísla
 - Znaménková – `int`, `float`, `double`
 - Nezáporná – `unsigned (int)`, `size_t`
- Znaky – `char`
- Výčtové typy – `enum class` (jindy)

Pole v C++

```
constexpr int leds[] = { led1, led2, led3, led4 };  
constexpr size_t leds_size = sizeof(leds) / sizeof(leds[0]);  
  
void print(const int array[], size_t array_length) {  
    for(size_t i = 0; i < array_length; ++i) {  
        printf(" %d", array[i]);  
    }  
}
```

Struktury v C++

```
struct S {  
    int value;  
  
    S() { // ctor - simple init of all attributes  
        value = 0;  
    }  
  
    S(int value_init) {  
        value = value_init;  
    }  
  
    int get_value() { return value; }  
    void  
};  
  
S s1; S s2(10);  
s1.attribute = 10;  
s.get_value();
```

Struktury v C++

```
struct Led {
    int pin;
    size_t turn_on_time;
    bool is_on;

    Led(int led_pin) { // ctor - simple init of data props
        pin = led_pin;
        turn_on_time = 0;
        is_on = false;
    }

    void setup() {
        pinMode(leds[i], OUTPUT);
        turn_off();
    }

    void turn_on() {
        if (!is_on) {
            digitalWrite(pin, ON);
            is_on = true;
            turn_on_time = now();
        }
    }

    void turn_off() {
        if (is_on) {
            digitalWrite(pin, OFF);
            is_on = false;
        }
    }
};
```

```
struct Leds {
    Led leds[] = {
        Led(led1_pin),
        Led(led2_pin),
        Led(led3_pin),
        Led(led4_pin),
    };

    size_t size() {
        return sizeof(leds) / sizeof(leds[0]);
    }

    void setup() {
        for(size_t i = 0; i < size(); ++i) {
            leds[i].setup();
        }
    }

    void turn_on_next() { ... }
    void turn_on_all() { ... }
    void show_binary_number(size_t number) { ... }
};

Leds leds;
constexpr size_t TIMEOUT_MS = 1000;
size_t last_turn_on;

void setup() {
    leds.setup();
    last_turn_on = now();
}

void loop() {
    if (last_turn_on + TIMEOUT_MS >= now()) {
        leds.turn_on_next();
        last_turn_on = now();
    }
}
```

Funkce - výstupní parametry

- žádný – void
- jeden – return XYZ
- více - & (reference)
 - Nedělá funkce víc věcí najednou?
 - ~~get_max_and_min()~~

```
int get_max(int array[], size_t array_length) {
    assert(array_length > 0);
    int max = array[0];
    for(size_t i = 1; i < array_length; ++i) {
        if (array[i] > max) {
            max = array[i];
        }
    }
}

size_t get_max_index(int array[], size_t array_length,
    int &found_max_value) {
    if (array_length == 0) {
        return 0;
    }

    size_t max_index = 0;
    for(size_t i = 1; i < array_length; ++i) {
        if (array[i] > array[max_index]) {
            max_index = i;
        }
    }
    found_max_value = array[max_index];
    return max_index;
}
```

Příklad na opravu

```
int f(int a[], int b) {  
    int x = 0;  
    for(int i = 0; i < b; ++i)  
        x += a[i];  
    return sum;  
}
```


Příklad na opravu: odsazení

```
int f(int a[], int b) {  
    int x = 0;  
    for(int i = 0; i < b; ++i) {  
        x += a[i];  
    }  
    return x;  
}
```

Příklad na opravu: pojmenování proměnných

```
int f(int array[], int array_length) {  
    int sum = 0;  
    for(int i = 0; i < array_length; ++i) {  
        sum += array[i];  
    }  
    return sum;  
}
```

Příklad na opravu: typy proměnných

```
int f(int array[], size_t array_length) {  
    int sum = 0;  
    for(size_t i = 0; i < array_length; ++i) {  
        sum += array[i];  
    }  
    return sum;  
}
```

Příklad na opravu: jméno funkce

```
int sum_array(int array[], size_t array_length) {  
    int sum = 0;  
    for(size_t i = 0; i < array_length; ++i) {  
        sum += array[i];  
    }  
    return sum;  
}
```

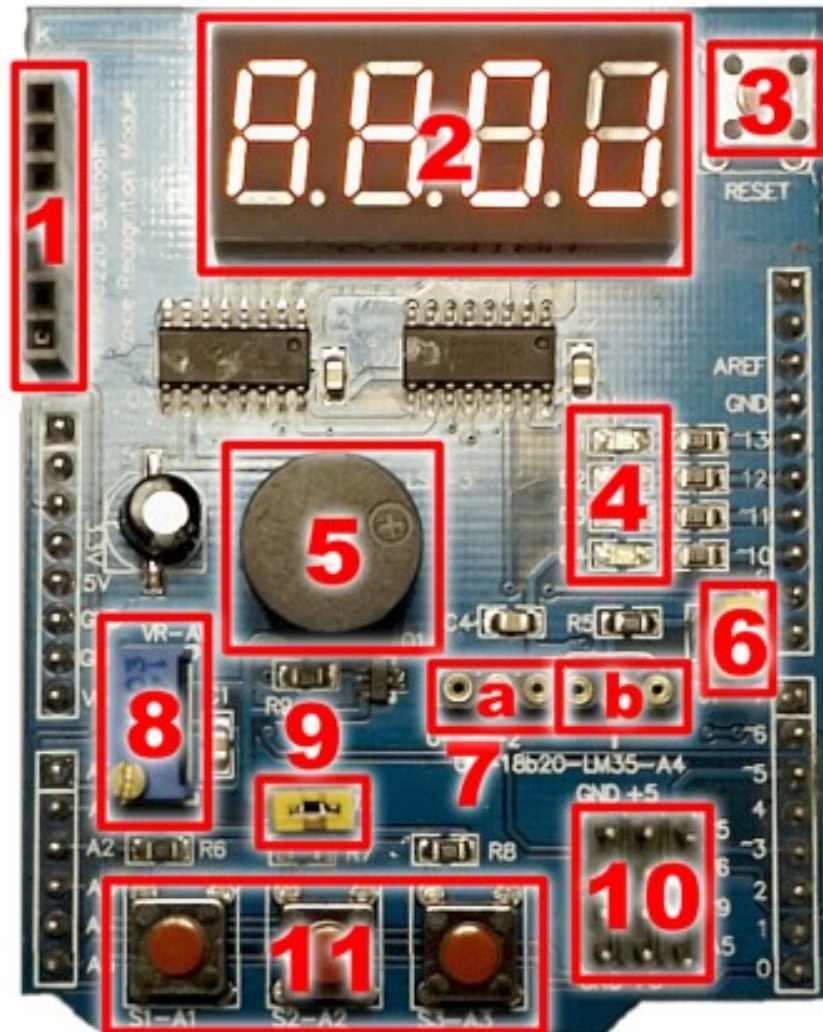
Tlačítka

- `pinMode(button1_pin, INPUT);`
 - inicializace
- `int value = digitalRead(button1_pin);`
 - `if (!value) { ... } // tlačítko je zmačknuto`
- Odkazy:
 - <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/>

Arduino

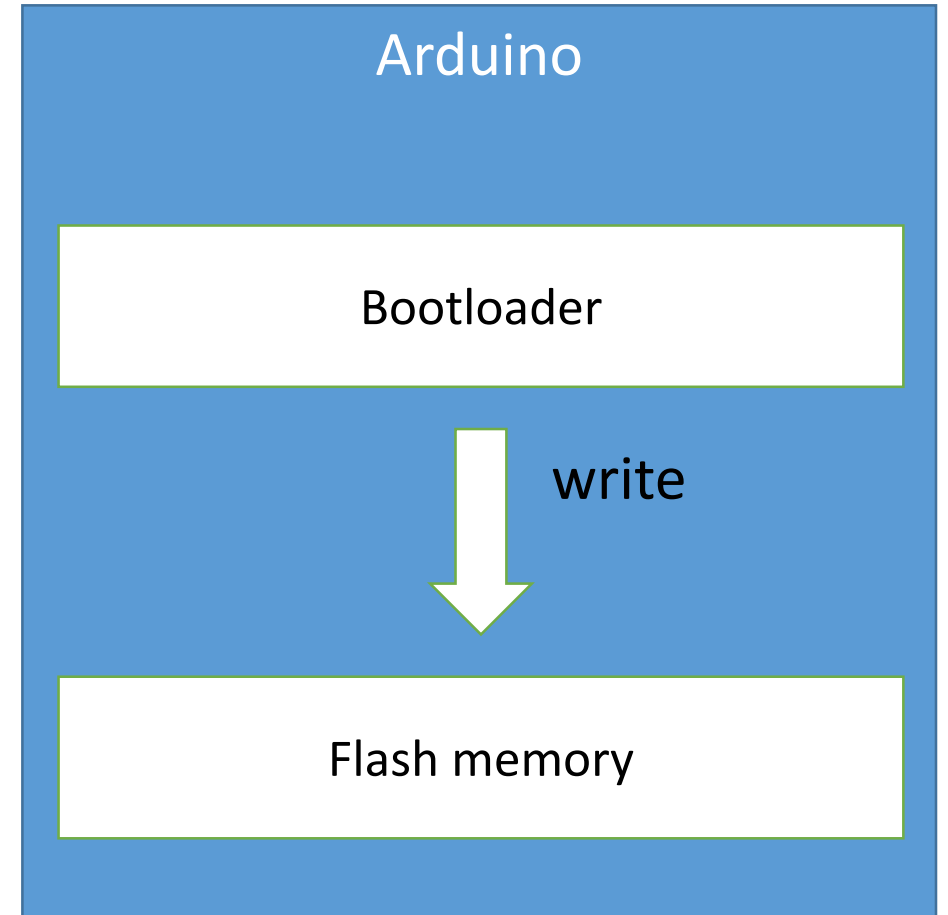
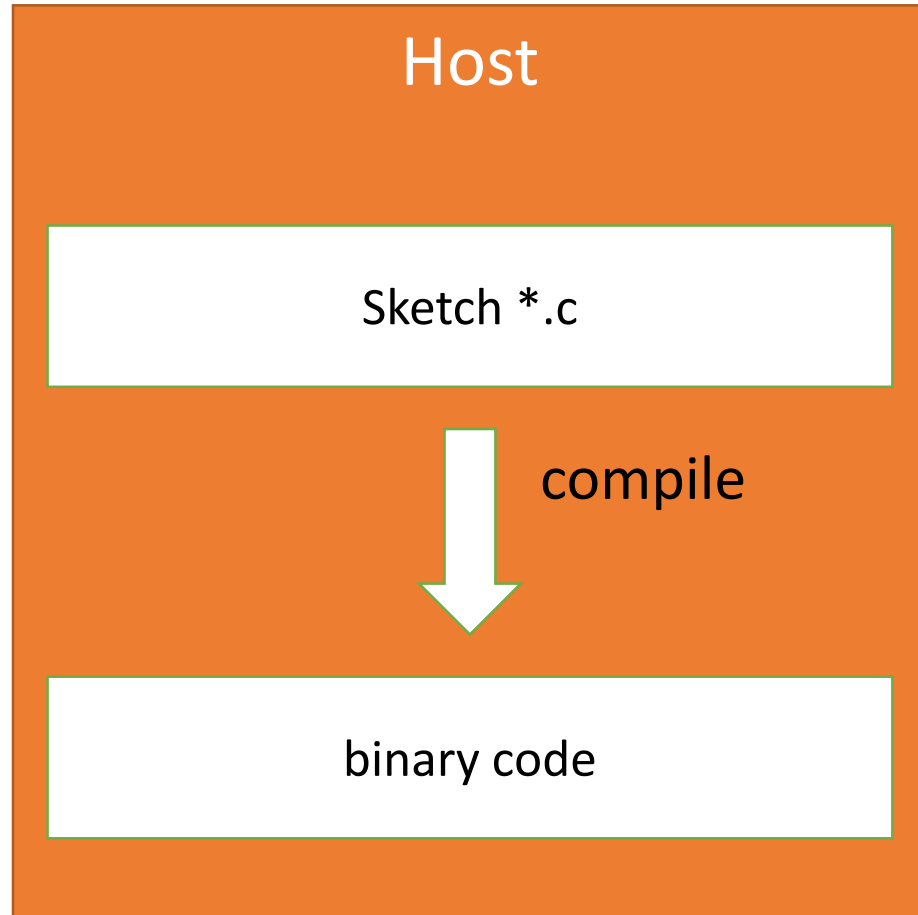
- Open-source HW SW projekt
- Arduino board + expansion board (shield)
- Arduino IDE

Arduino HW



- (1) Konektor pro Bluetooth
- (2) LED
- (3) Reset
- (4) Signalizační LED
- (5) Piezo-bzučák
- (6), (9) Propojka
- (7) Konektor pro IR
- (8) Potenciometr
- (10) Konektor pro čidla
- (11) Vstupní tlačítka

Kompilace

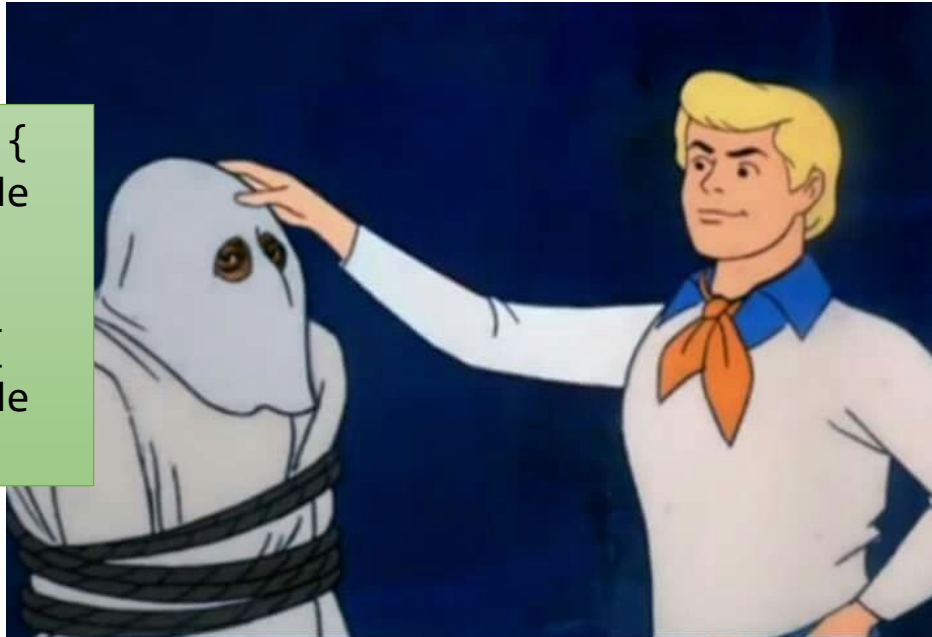


Arduino IDE

```
void setup() {  
    // put your setup code here, to run once:  
}
```

```
void loop() {  
    // put your main code here, to run repeatedly:  
    // called ~1000/s  
}
```

```
void setup() {  
  // init code  
}  
  
void loop() {  
  // main code  
}
```



```
int main() {  
  setup();  
  while(true) {  
    loop();  
  }  
}
```



Dobré programátorské zvyky

- Zapnout warnings překladače
- Nepoužívat copy&paste
 - Funkce, pole, ...
- Používat konstanty
 - `#include "funshield.h"`

Úkoly

1. Inicializovat LEDky
 - `pinMode(pin, OUTPUT/INPUT)`
2. Blikat vybranou LEDkou
 - `digitalWrite(pin, HIGH/LOW)`
 - `delay(ms)`
3. Blikat všemi ledkami najednou
 - ne C&P (co kdyby LEDek bylo 1M)
4. Blikat bez delay
 - `millis(ms)`
5. Had délky 2
6. Had libovolné délky

Odkazy

- https://www.ksi.mff.cuni.cz/teaching/nswi170-web/#@tab_links

Komunikace

- Bud'te proaktivní
- Web
 - https://fan1x.github.io/computer_systems.html
 - <https://www.ksi.mff.cuni.cz/teaching/nswi170-web/>
- Mattermost
- Mail
- Zoom 😊

Průběh cvičení

- Účel předmětu NSW170 – Počítačové systémy
 - Vysvětlit, co informatik potřebuje vědět o hardware a systémovém software
 - Seznámit se s jazykem, který je pravým opakem Pythonu
 - Vyzkoušet si programování v těsném kontaktu s hardware
- Obsah přednášky (Jakub Yaghob nebo Lubomír Bulej)
 - 1..2 – základy jazyka C
 - 3..14 – operační systémy, překladače, ...
- Obsah cvičení
 - Předmět je sice 2/2, ale cvičení je pouze jednou za 14 dní
 - Druhou dvouhodinu strávíte u domácích úkolů (a vaši učitelé při jejich kontrole)
 - 1 – první kroky v C++
 - 2..6 – programování pro Arduino
- Od třetího týdne přednáška se cvičením nesouvisí
 - Ani zápočet se zkouškou

Zápočet

- Před druhým cvičením (22.3.) si zajistěte prostředí k práci
 - **Arduino (v knihovně nebo koupit on-line)**
 - Nainstalujte si na vašem počítači **Arduino IDE** (pro řešení domácích úkolů)
- Na cvičení budou zadávány úlohy
 - Odevzdání do ReCodexu
 - **1 týden na řešení**
 - Arduinovské úlohy na sebe navazují, řešení tedy budete sami potřebovat
- Na šestém cvičení bude zadána hlavní domácí úloha

IDE

- V čem tedy budete programovat?
 - Technicky to bude C++
 - C++ je (téměř) nadmnožina C
 - U některých C-konstrukcí má C++ o něco přísnější pravidla, tím včas odhalíte některé chyby
 - Půjčíme si z C++ několik drobností usnadňujících život
 - Parametry předávané odkazem, prázdné závorky v deklaraci funkce bez parametrů, ...
 - Složitější vlastnosti C++ nejsou v nízkoúrovňovém prostředí příliš užitečné
 - Často ani nejsou dostupné kvůli omezené kapacitě hardware
- Kde?
 - 1. cvičení: coliru.stacked-crooked.com
 - Webový editor schopný zkompileovat a spustit jednoduchý program v C++
 - Kdo to umí, může používat jakýkoliv jiný editor a překladač C++
 - Zbytek cvičení: Arduino IDE - www.arduino.cc/en/main/software
 - Aplikace pro Windows/Linux/macOS
 - Editor, překladač, dálkový (USB) ovladač Arduina

Čas na hraní 😊

Hello World

```
#include <stdio>
```

```
int main()  
{  
    printf("Hello World :)\n");  
}
```

Tajemná funkce 1

```
#include <stdio>
```

```
int fn1(int array[], int length) {  
    int res = 0;  
    for(int i = 0; i < length; ++i) {  
        res += array[i];  
    }  
    return res;  
}
```

```
int main()  
{  
    int array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
    int res = fn1(array, 9);  
    printf("Result: %d", res);  
}
```

Tajemná funkce 2

```
#include <stdio.h>
```

```
int fn2(int array[], int length, int number) {  
    int i = 0;  
    while(i < length && array[i] != number) {  
        ++i;  
    }  
  
    return i;  
}
```

```
int main()  
{  
    int array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  
    static constexpr int SIZE = 100;  
    int res = fn2(array, SIZE, 4);  
    printf("Result: %d", res);  
}
```

Tajemná funkce 3 (1/2)

```
#include <stdio.h>

int fn3(int array[], int length) {
    int j = 0;
    int k = 0;
    for(int i = 0; i < length; ++i) {
        if (array[i] % 2 == 0) {
            ++j;
        } else {
            ++k;
        }
    }

    if (j > k) {
        return j;
    } else if (k > j) {
        return -k;
    } else {
        return 0;
    }
}

int main()
{
    int array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
    static constexpr int SIZE = 9;
    int res = fn3(array, SIZE);
    printf("Result: %d", res);
}
```

Tajemná funkce 3 (2/2)

```
int count_and_compare_odd_even(int array[], int length) {  
    int even_count = 0;  
    int odd_count = 0;  
  
    for(int i = 0; i < length; ++i) {  
        if (array[i] % 2 == 0) {  
            ++even_count;  
        } else {  
            ++odd_count;  
        }  
    }  
  
    if (even_count > odd_count) {  
        return even_count;  
    } else if (odd_count > even_count) {  
        return -odd_count;  
    } else {  
        return 0;  
    }  
}
```

Úkoly

1. Hello World
2. Nakreslit trojúhelník
3. Nakreslit vánoční stromeček
4. Vypsát průměr hodnot v poli
5. Nakreslit graf hodnot v poli
6. Nakreslit klouzavý průměr hodnot v poli
 - a) Pro fixní N
 - b) Obecně pro N po sobě jdoucích hodnot
7. Histogram

```
      *
    ***
  *****
*****
```

```
      *
    ***
  *****
*****
```

