

Počítačové systémy

Adam Šmelko

smelko@ksi.mff.cuni.cz

NSWI170 - cvičení

LS 2020/2021

Jazyk C

- Globálne premenné

Životnosť premenných

- Miesto definície premennej určuje jej scope.
- Premenná definovaná mimo tela akejkoľvek funkcie => **Globálna premenná**
- **SCOPE** premennej:
 - Vo svojom bloku {}
 - V nižšom (zanorenom) bloku {}
 - **NIE** vo vyššom bloku

```
int global_a;

int f(int param_b) {
    int local_c;
    local_c = global_a;
    ...
    return local_c;
}

void g() {
    int local_d = local_c; //error!
    ...
}

int main() {
    global_a = 42;
}
```

```
void functionWithScopes() {

    int local_a;

    for (int i=0; i<1; i++) {

        local_a = local_a + i;
        int local_b;

    }

    int local_c = local_a;
    local_c = local_b; //error!

}
```

```
void f() {
    int local_a = global_b; //error!
}

int global_b;

void g() {
    int local_c = global_b; //OK
}

int main() {
    global_b = 42;
}
```

Arduino

LED

Arduino

- Arduino UNO
 - základní deska, digitální a analogové piny
- Expansion board (shield)
 - multifunction shield - funduino
 - tlačítka, LED, 4-místný displej, bzučák
- Arduino IDE
 - www.arduino.cc/reference
 - kabinet.fyzika.net/dilna/ARDUINO/funduino-popis.php
 - 1.8 / 2.0.beta3
 - manage libraries
 - Arduino AVR Boards (Uno)
 - Tools / Board / ... Arduino Uno
 - Tools / Port / COMx (Arduino Uno)
 - Sketch / Include Library / Add .ZIP Library [iba raz](#)
 - Sketch / Include Library / Funshield [každú sketch](#)

CPU ATmega328P
14 digital I/O pins
6 analog inputs
Clock speed 16 MHz
FLASH memory 32 KB
SRAM 2 KB
EEPROM 1 KB

Arduino

- sketch
 - save -> Documents/Arduino/**novak** -> **novak.ino**
 - compile ctrl-R
 - upload ctrl-U
- void **setup()**;
 - jednou při startu
 - nastavení módu PINů
 - počáteční hodnoty PINů / proměnných
- void **loop()**;
 - ~ 1000x /s
 - vlastní výkonný kód

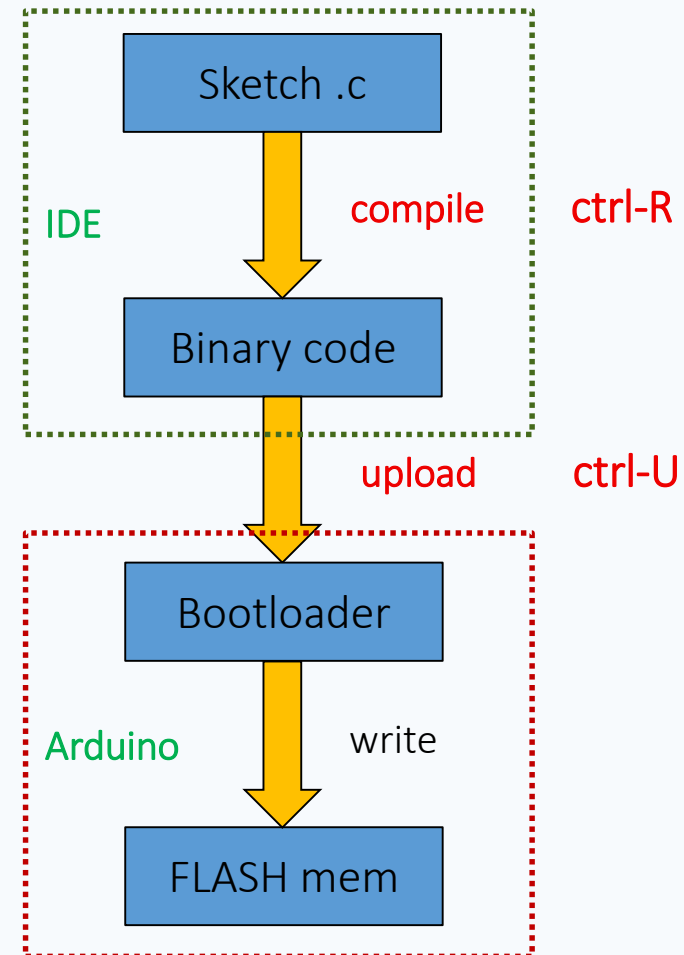
```
int main() {  
  init();  
  setup();  
  for(;;)  
    loop();  
}
```

zjednodušeno

```
void setup() {  
  ....  
}  
  
void loop() {  
  ....  
}
```

inicializace
provede se
jednou

smyčka
provádí se
cca 1000x /s



LED

- diody připojeny na PINy
- inicializace PINu
 - typicky v setup()
 - **pinMode**(pin, **OUTPUT**)
- ovládání LED ≈ zápis na PIN
 - **digitalWrite**(pin, HIGH/LOW)
 - HIGH ≈ ☺ nesvíí ⚡
 - LOW ≈ ☹ svítí
 - Funshield helper: ON/OFF

no copy-
and-paste

- ➡ rozsviďte i-tou LED
 - ostatní zhasněte
 - parametr

```
// my funshield library
constexpr int led[] {led1_pin, led2_pin, led3_pin, led4_pin};
constexpr int led_count = sizeof(led)/sizeof(led[0]);

// ....
digitalWrite( led[i], LOW);
```

low-level kód
nepoužívejte přímo, vytvořte si abstrakci

```
// my funshield library

constexpr int led[] {...};
constexpr int led_count = ...;

// my assignments

void led_on(..) {
    ....
}

// setup and loop

void setup() {
    ....
}

void loop() {
    ....
}
```

výkonný
kód

pinMode

inicializace
≡ zhasnutí

jen volání
vlastních funkcí

Blikání a časování

- unsigned long **millis()**
- void **delay**(unsigned long ms)

globální
musí přežít!

```
unsigned long last_time = 0;

void loop_delay( unsigned long interval) {
    auto cur_time = millis();
    if( cur_time >= last_time + interval) {
        last_time = cur_time;
        ....
    }
}
```

nastal čas
něco dělat!

➡ 2.1 blikající LED

- kdy něco dělat?
- co dělat?
 - (interní) stav vs. jeho vizualizace
- co si potřebuju pamatovat?
- kde to budu mít uložené?

teď
změna stavu

0	300	600	900
😊	😬	😊	😬

vizualizace

➡ 2.2 semafor na železničním přejezdu

- vždy jedna dvojice LED svítí, druhá ne
- parametr: čas jednoho bliku

```
void blik(....) {
    bool sviti = false;
    if( nastal čas) {
        sviti = ! sviti;
        // vizualizace
    }
}
```

funguje?

stav
musí přežít

změna stavu

```
bool sviti = false;

void blik(....) {
    if( nastal čas) {
        sviti = ! sviti;
        ....
    }
}
```

dekom
pozice

logická
negace

Kulička a had

- ➡ 2.4 běžající kulička

- 1. parametr:
 - čas zobrazení jedné kuličky v ms
- 2. parametr: způsob běhání
 - dokola: 01230123...
 - odráží se: 012321012321...

- ➡ 2.5 ♥ běžající had

- 2.5a několik kuliček (LED) za sebou
 - parametr: velikost hada = počet LED
 - kulička z 2.4 $\approx$ had velikosti 1
 - had postupně vylézá a zalézá
 - nejdříve se zobrazí jedna kulička, pak dvě, ...
- 2.5b aliasing
 - krajní kuličky část času s nižší intenzitou
 - svítí menší počet cyklů $\rightsquigarrow$ nižší intenzita
 - pokročilejší: rostoucí / klesající intenzita
- 2.5c PWM ➡ www.arduino.cc

- ➡ kultura kódu

- dekompozice
- parametrizace
- odstranění copy-and-paste
 - `loop_delay()`

Binární rozklad

→ 2.3 binární rozklad

- parametr: rozsvítit LED dle spodních 4 bitů
 - bit = 1 → 😊 rozsvítit LED
 - bit = 0 → 😐 zhasnout LED
- funkci použijte v count-up
 - jednoduchý čítač: 0 .. ∞
 - parametr: rychlost počítání (ms)
 - po každém uplynulém intervalu se inkrementuje čítač a zobrazí se LED

