

Lab 3 - Buttons

HW Feedback

- Pridajte sa na Mattermost

Buttons

Pin-y

- `BUTTON1_PIN` , `BUTTON2_PIN` , `BUTTON3_PIN`

Inicializácia

- `pinMode(pin, INPUT)`
 - `INPUT` znamená, že napätie je nastavované externými zdrojmi
 - default

Detekcia stavu

- `digitalRead(BUTTON_PIN)`
 - vracia `LOW` (`ON`) ak je stlačené
 - stlačením tlačítka ho pripojíme na GND, read prečíta 0 (== `LOW`)
 - inak je floating

Buttons cont.

Kmitanie (bouncing)

- chyba HW
- hodnota na pine tlačítka sa môže náhodne zmeniť
 - trvanie ~30 ms
- treba ošetriť v kóde
 - na HW3 netreba

Unforgivable Curse #3 - DRY

Don't Repeat Yourself

- Nepísať ten istý kód viacej krát
 - prehľadnosť
 - ľahšie úpravy
 - menej bugov
- Ako na to?
 - cykly
 - polia
 - funkcionálna dekompozícia

Príklad WET :(

```
void setup() {  
    pinMode(led_pin_1, OUTPUT);  
    pinMode(led_pin_2, OUTPUT);  
    pinMode(button_pin_1, INPUT);  
    pinMode(button_pin_2, INPUT);  
}  
  
void loop(){  
    if (digitalRead(button_pin_1) == DOWN) {  
        digitalWrite(led_pin_1, ON);  
    } else {  
        digitalWrite(led_pin_1, OFF);  
    }  
    if (digitalRead(button_pin_2) == DOWN) {  
        digitalWrite(led_pin_2, ON);  
    } else {  
        digitalWrite(led_pin_2, OFF);  
    }  
}
```

Príklad DRY :)

```
#include <funshield.h>
constexpr int LED_PINS[] = { LED1_PIN, LED2_PIN, LED3_PIN };
constexpr int LED_COUNT = sizeof(LED_PINS) / sizeof(LED_PINS[0]);
constexpr int BUTTON_PINS[] = { BUTTON1_PIN, BUTTON2_PIN, BUTTON3_PIN };
constexpr int BUTTON_COUNT = sizeof(BUTTON_PINS) / sizeof(BUTTON_PINS[0]);

void init_pins(const int pin_array[], const int pin_count, int mode) {
    for (int i = 0; i < pin_count; ++i) {
        pinMode(pin_array[i], mode);
    }
}

void setup() {
    init_pins(LED_PINS, LED_COUNT, OUTPUT);
    init_pins(BUTTON_PINS, BUTTON_COUNT, INPUT);
}

void loop() {
    for (int i = 0; i < BUTTON_COUNT; ++i) {
        bool is_down = digitalRead(BUTTON_PINS[i]) == DOWN;
        digitalWrite(LED_PINS[i], is_down ? ON : OFF);
    }
}
```

Pozor na if-y

- Ak musíme pridávať if-y do tela loopu aby sme odlíšili jednotlivé iterácie, pravdepodobne je niečo zle
- Pravdepodobne je potrebná zmena architektúry, dekompozícia

Zlé if-y

```
void loop() {  
    // ...  
    for (int i = 0; i < BUTTONS; ++i) {  
        if (digitalRead(BUTTON_PINS[i]) == ON) {  
            if (!button_was_down[i]) {  
                button_was_down[i] = true;  
                if (i == 0) {  
                    start_the_countdown();  
                } else if (i == 1) {  
                    stop_the_countdown();  
                } else {  
                    reset_counter();  
                }  
            }  
        } else {  
            button_was_down[i] = false;  
        }  
    }  
}
```

Opravené if-y

```
bool handle_button(int button_index) {
    if (get_pin_state(BUTTON_PINS[button_index]) == DOWN) {
        if (!button_was_down[button_index]) {
            button_was_down[button_index] = true;
            return true;
        }
    } else {
        button_was_down[button_index] = false;
    }
    return false;
}

void handle_counter_logic() {
    if (handle_button(0)) start_the_countdown();
    if (handle_button(1)) stop_the_countdown();
    if (handle_button(2)) reset_counter();
}

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

Cvičenie 1 - Binary LEDS

- Vytvorte funkciu na zobrazenie čísel v dvojkovej sústave pomocou LEDiek
 - napr. `void display_binary(int num)`
 - zavoláme raz v `setup()`, `loop()` prázdny
 - dekompozícia
- 4 LEDky → 16 čísel
 - modulo 16

Binárne operátory

- `&` = AND, `|` = OR, `~` = NOT
- `<<`, `>>` Left shift, right shift

```
5 & 7 // 0101 AND 0110 == 0100 == 4
6 >> 1 // 0110 -> 0011 = 3
```

Cvičenie 2 - Timed Counter

- Hodnota na LEDkách sa inkrementuje o 1 v danom intervale
 - `constexpr int PERIOD = 1000ms`
- Použite funkciu z predošlého cvičenia
- Modulo nieje potrebné

Cvičenia 3, 4 - Button counter

Cvičenie 3

- Hodnota na LEDkách sa bude zvyšovať len počas stlačenia tlačítka 1
 - → hodnota na tlačítku je LOW
 - reálny čas stále beží

Cvičenie 4

- Hodnota na LEDkách sa zvýši len keď stlačíme tlačítko
 - → stav tlačítka sa zmení
 - inkrementujeme raz
 - bonus: debounce

Cvičenie 5 - More buttons

- Tlačítko 2 zníži hodnotu na LEDkách
 - pozor na záporné hodnoty
- Tlačítko 3 zvýši hodnotu o 5
- Všetky tlačítka robia skoro to isté...
 - Myslite na DRY

OOP

- cheli by oddeliť logiku tlačítok od zvyšku programu
 - → funkcie
 - počet parametrov narastá
 - dáta niesú oddelené
- Riešenie: Struct

```
struct Button {  
    int pin;  
    bool is_pressed;  
};  
  
void processButton(Button &button) {  
    if (digitalRead(button.pin) == ON && !button.is_pressed) {  
        button.is_pressed = true;  
        ... // Handle click event  
    }  
}
```

Debugging

- Môžeme použiť Serial linku

```
void setup(){  
    Serial.begin(9600)  
}  
  
void loop(){  
    //..  
    Serial.println("Arduino writes back")  
    Serial.println(sensor_value, HEX)  
}
```

HW3

- Kombinácia timed a increment čítača
 - keď stlačíme tlačítko, hodnota sa zvýši o 1
 - ak ho stále držíme stlačené, tak po danej perióde sa znovu inkrementuje
 - pozor na korektnú dĺžku periódy
- nemusíte riešiť debounce