

# Laboratorní práce

## Digitální technika

### 1 Úvod

1. Představení desky Schoolboard a jejích součástí.
2. Představení programu Visual Studio Code, program i s rozšířením PlatformIO je potřeba mít pro studenty nainstalovaný předem.
3. Ukázka nahrání programu (*test.h*) do čipu na desce a vysvětlení významu jednotlivých příkazů.
4. Vzory pro jednotlivé programy si studenti vždy nahrají do čipu na desce a potom je zkusí upravit podle zadání.<sup>1</sup>
5. Tato laboratorní práce počítá s tím, že studenti mají za sebou úvodní kurz programování a alespoň matně si pamatují, jak fungují proměnné, podmínky a cykly.
6. Úkoly označené hvězdičkou \* jsou volitelné.

### 2 Blikání LED

1. Do čipu nahrajte program *blink.h*.
2. Upravte program tak, aby blikala zelená LED s periodou 0,5 s. Video s dvěma možnými řešeními je [zde](#).

### 3 Čtení tlačítka a výpis do sériové linky

1. Do čipu nahrajte program *button\_serial.h*.
2. Otevřete terminál a sledujte činnost programu.
3. Ukončete terminál můžete stiskem Ctrl+C, napřed je potřeba kliknout myší do prostoru terminálu. Vždy před nahráním programu do čipu je potřeba ukončit terminál!

---

<sup>1</sup>Přitom je potřeba dodržovat pojmenování proměnných i souborů bez diakritiky a bez mezer a počítat s tím, že v jazyku C++ se rozlišují velká a malá písmena.

4. Upravte program tak, aby posílal data do terminálu přiměřenou rychlostí, např. 2x za sekundu. Jedna z možných variant je na tomto [videu](#).

## 4 Počítání stisků tlačítka

1. Do čipu nahrejte program *button\_counting.h*.
2. Stiskněte opakovaně tlačítko 0 a sledujte výpis na terminálu. Co pozorujete?
3. Upravte program tak, aby se při krátkém stisku tlačítka zvedla hodnota na terminálu vždy o 1.

## 5 Rozsvícení LED na stisk tlačítka

1. Do čipu nahrejte program *button\_LED.h*.
2. Upravte program tak, aby se LED při stisknutí tlačítka rozsvítila na 1,5 sekundy.
3. Upravte program z bodu 1 tak, aby se při stisku tlačítka rozsvítily všechny LED. Jedno z možných řešení je na videu [zde](#).
4. \* Do čipu nahrejte program *push\_LED.h*, který rozsvěcí jednu LED po druhé – viz [video](#). Proměnná *led* funguje jako tzv. **stavová** – uchovává informaci o tom, v jakém stavu se program právě nachází.
5. \* Upravte program z bodu 2 tak, aby se LED při stisknutí tlačítka rozsvítila na 1,5 sekundy, a to bez použití funkce *delay()* – program se nebude zastavovat. Pomůže vám funkce *millis()*<sup>2</sup>.

## 6 Změna jasu LED

1. Do čipu nahrejte program *PWM\_LED.h*. Pokuste se pochopit, jak program funguje.
2. Upravte program tak, aby se po každém krátkém stisku tlačítka jas LED zesiloval vždy po 1/4. Po dosažení maxima se LED vypne a cyklus se opakuje.

---

<sup>2</sup>Funkce *millis()* vrací počet milisekund od zapnutí nebo resetování čipu.

3. \* Upravte program tak, aby se po každém krátkém stisku tlačítka jas LED zesiloval vždy po 1/4 a po dosažení maxima zase po 1/4 zeslaboval.

## Další zdroje

- [1] *Výuka programování v C/C++* [online] Petr Bílek [cit. 2020-03-02] Dostupné z: <https://www.sallyx.org/sally/c>
- [2] *Další kurzy C/C++* [online] <https://www.itnetwork.cz> [cit. 2020-03-02] Dostupné z: <https://www.itnetwork.cz/cplusplus>
- [3] *ROBOTICKÝ MANUÁL* [online] Miroslav Burda a kol. [cit. 2020-03-02] Dostupné z: <https://roboticsbrno.github.io/RoboticsBrno-guides/RoboticsManual.pdf>

## 7 ŘEŠENÍ k vybraným příkladům

Kapitola 4, úkol 3:

```
void loop(){
    if ( !digitalRead(SW0) ) c=c+1;
    Serial.println(c);
    delay (200);
}
```

---

Kapitola 5, úkol 2:

```
void loop(){
    if (( digitalRead ( SW1 )) == LOW )
    { digitalWrite (L_R, HIGH );
      delay(2000);}
    else { digitalWrite (L_R, LOW );}
}
```

---

Kapitola 5, úkol 5:

```
int pushed = 0;
long time1 = 0;

void setup(){
    Serial.begin(115200);
    pinMode(SW1, INPUT_PULLUP);
    pinMode(L_R, OUTPUT);
    delay(10); // needed for stabilization
}

void loop(){
    if (( digitalRead ( SW1 )) == LOW ) {
        if ( pushed == 0 ) {
            digitalWrite (L_R, HIGH );
            pushed = 1;
            time1 = millis();
        }
    }
}
```

```

        if ( (millis() - time1 > 1500) and (time1 > 0) ) {
            digitalWrite (L_R, LOW );
            pushed = 0;
        }
    }
}

```

---

Kapitola 6, úkol 2:

```

void loop() // this part works in cycle
{
    if (( digitalRead ( SW3 )) == LOW ) x = x + 250;
    if (x > 1000) x = 0;
    ledcWrite (0, x);
    delay(200);
}

```

---

Kapitola 6, úkol 3:

```

int x = 0;
int up = 1; // bool up = true;

void setup(){
    pinMode(L_R, OUTPUT);
    pinMode(SW3, INPUT_PULLUP);
    ledcSetup (0, 1000 , 10); // create PWM
    // ledcSetup ( channel , freq , resolution )
    // channel = 0 - 15
    // resolution = 10 => 2^10 => 1024
    ledcAttachPin (L_R , 0); // ledcAttachPin (pin , channel)
}

void loop() // this part works in cycle
{
    if (( digitalRead ( SW3 )) == LOW ) {
        if (up) {
            x = x + 250;
            if (x == 1000) up = 0;
        }
        else {

```

```
        x = x - 250;
        if (x == 0) up = 1;
    }
    ledcWrite (0, x);
    delay(200);
}
```