

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Mikroprocesorové a vestavěné systémy – projekt
Sada demo aplikací nad FreeRTOS

Obsah

1	Úvod	2
2	Použití	2
3	Implementace	2
3.1	Setup	2
3.2	Úloha tlačítka	3
3.3	Úloha LED	4

1 Úvod

Zadáním projektu bylo vypracovat aplikaci, která používala operační systém pro mikroprocesory FreeRTOS¹. Projekt měl obsahovat následující části:

- Minimálně jedna vstupní a výstupní periférie
- Každá periférie musí být ve vlastní **úloze** (tásku)²
- **Synchronizační službu** pro předávání dat mezi úlohami
- **Službu pro práci s časem** pro aktivní čekání v úlohách

2 Použití

Samotné použití je jednoduché. Po zapojení na napájené začne na ESP blikat vestavěná LED v intervalu 500ms. Pokud člověk stiskne tlačítko a dokud ho drží, LED bliká jednou za vteřinu a indikuje tak, jak dlouho je tlačítko stisknuté. Po puštění tlačítka začne LED blikat tolikrát, kolik vteřin bylo tlačítko stisknuté. Toto dělá dokud člověk opět nestiskne tlačítko a nenastaví jiný počet bliknutí.

3 Implementace

Projekt byl vypracován s použitím **Arduino frameworku**. Ten obsahuje dvě základní části. Funkci **setup** (nastavení) a funkci **loop** (cyklus). Funkce setup běží pouze jednou a to úplně na začátku. Používá se na výchozí nastavení hardwaru a inicializaci proměnných. Poté se zavolá funkce loop, která se vykonává do nekonečna. Ta je zde ale nevyužitá, jelikož má každá část aplikace svojí vlastní úlohu.

3.1 Setup

Ve funkci setup se nejdříve nastaví mód pinů, které jsou použity pro výstup (vestavěná LED) a vstup (tlačítko na pinu 23). Následně se inicializuje fronta o velikosti 1 *int*, která se používá na posílání dat mezi úlohou tlačítka a LED. Jako poslední se zde vytvoří jednotlivé úlohy se svojí specifickou funkcí, jménem, velikostí zásobníku a prioritou. Ty se také hned spustí.

```
#define BUTTON (23)
QueueHandle_t queue; //queue for sending data between tasks
.
.
.
void setup() {
    pinMode(LED_BUILTIN, OUTPUT); //use builtin led as output
    pinMode(BUTTON, INPUT_PULLUP); //use button on pin 23 as input

    queue = xQueueCreate(1, sizeof(int)); //create queue with size of 1 int

    //create tasks
    xTaskCreate(ledTask, "led_task", 100000, NULL, 1, NULL);
    xTaskCreate(buttonTask, "button_task", 100000, NULL, 1, NULL);
}
```

¹<https://freertos.org/>

²<https://www.freertos.org/taskandcr.html>

3.2 Úloha tlačítka

V úloze tlačítka je `while(true)` cyklus, aby se úloha vykonávala do nekonečna a neukončila se. Pokud se zjistí, že je tlačítko stisknuté, nahraje se do fronty hodnota -1. Poté dokud je tlačítko stisknuté, každou vteřinu se do proměnné `cnt` přičte 1. Po puštění tlačítka se poté proměnná `cnt` nahraje do fronty. `vTaskDelay(1)` je zde použito pouze kvůli *Watchdogu*³, jelikož se někdy může stát, že dojde k vypršení jeho časovače, protože ho úloha nestihla vyresetovat. Tímto se zajistí, že ho úloha stihne vyresetovat.

```
void addToQueue(int data) {
    xQueueSend(queue, &data, 0);
}

void buttonTask(void *params) {
    int cnt = 0;

    while(true) {
        //on button press
        if (!digitalRead(BUTTON)) {
            addToQueue(-1); //send -1 to start blinking every second while button is pressed

            //count up every second while the button is pressed
            while(!digitalRead(BUTTON)) {
                cnt++;
                vTaskDelay(1000);
            }

            addToQueue(cnt); //send count to queue
            cnt = 0; //reset queue
        }

        vTaskDelay(1); //task is alive for watchdog
    }
}
```

³https://en.wikipedia.org/wiki/Watchdog_timer

3.3 Úloha LED

Úloha pro LED obsahuje stejně jako úloha pro tlačítko `while(true)` cyklus. Následně se zde jako první věc čte z fronty, ale pouze pokud obsahuje nějaká nová data. Ta se uloží do proměnné `cnt`. Pokud jsou data menší než 0 (-1 od úlohy tlačítka), začne LED blikat každou vteřinu, aby indikovala přičítání do proměnné `cnt` v úloze tlačítka, které také nastane každou vteřinu. Pokud je hodnota `cnt` rovna 0, bliká LED jednou za půl vteřiny, aby indikovala připravenost zařízení. Pokud je hodnota `cnt` větší než 0, bliká LED tolikrát, kolik je hodnota `cnt`. Mezi každým rozsvícením je zdržení 200ms. Po dokončení blikání čeká vteřinu a poté opět bliká tolikrát, kolik je hodnota `cnt`. Toto se opakuje, dokud se opět nestlačí tlačítko a nenahraje se nová hodnota.

```
void ledTask(void *params) {
    int cnt = 0;

    while(true) {
        //read from queue when something is waiting
        if (uxQueueMessagesWaiting(queue))
            xQueueReceive(queue, &cnt, 0);

        //blink every second while waiting for button to be depressed
        if (cnt < 0) {
            digitalWrite(LED_BUILTIN, HIGH);
            vTaskDelay(500);
            digitalWrite(LED_BUILTIN, LOW);
            vTaskDelay(500);
        }

        //blink every half second on start
        else if (cnt == 0) {
            digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN));
            vTaskDelay(250);
        }

        //blink as many times as for how long the button was pressed
        else {
            for (int i = 0; i < cnt*2; i++) {
                digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN));
                vTaskDelay(100);
            }
            vTaskDelay(1000);
        }
    }
}
```