

# Stopky

Ondřej Nedojedlý\*

25. Listopadu, 2023

## Obsah

<b>1</b>	<b>Zadání</b>	<b>2</b>
<b>2</b>	<b>Teoretický rozbor</b>	<b>2</b>
2.1	Popis přípravku . . . . .	2
2.1.1	LCD . . . . .	3
2.2	Popis softwaru . . . . .	3
2.2.1	RTOS . . . . .	4
2.3	Vývojový diagram . . . . .	4
<b>3</b>	<b>Řešení</b>	<b>5</b>
3.1	Zdrojový kód . . . . .	5
<b>4</b>	<b>Hodnocení</b>	<b>6</b>
4.1	Úspěchy/neúspěchy . . . . .	6
4.2	Další rozšíření . . . . .	7

## Seznam obrázků

1	STM32F4 discovery kit . . . . .	2
2	Vývojový diagram . . . . .	4

## Seznam tabulek

1	Připojení LCD k STM32F4 Discovery . . . . .	3
2	Připojení dalších pinů LCD . . . . .	3

---

\*SPŠ elektrotechnická, Makarenkova 1; Mgr. Tomáš Michalek

# 1 Zadání

Vytvoření programu na přípravku **STM32F407VGT6U** s **RTOS-RTX4**, jež bude fungovat jako jednoduché stopky. Program bude zobrazovat na prvním řádku LCD čas, který začne běžet od začátku programu. Čas bude ve formátu:

$$MM : SS : mmm$$

kde:

**M** jsou minuty,

**S** jsou sekundy,

**m** jsou milisekundy.

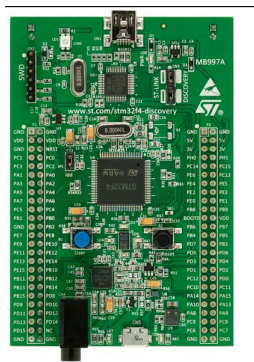
Po zmáčknutí uživatelského tlačítka, se na druhý řádek LCD dispeje zobrazí čas v okamžiku zmáčknutí tlačítka. Nová časová stopa se nahraje až při opětovném zmáčknutí uživatelského tlačítka; při držení se nenahrává stále nová časová stopa.

## 2 Teoretický rozbor

Pro pochopení běhu programu, jeho funkcionality a možných problémů je nutno mít solidní základy teoretického rázu, jinak bude zpracování úlohy problematické.

### 2.1 Popis přípravku

Jak avizováno výše, jedná se o přípravek STM32F407VGT6U od firmy STMicroelectronics. Přípravek je 32-bitový Arm® Cortex®-M4 architektury RISC, přípravek je zaobalen ve STM32F4-DISCOVERY kitu (Obrázek 1).



Obrázek 1: STM32F4 discovery kit

STM32F4xx podporuje:

- Analogový generátor náhodných čísel,
- 15 komunikačních sběrnic,
- Dva 12-bitové DA převodníky,
- Tři 12-bitové AD převodníky,
- 17 časovačů,
- 1-MB flash paměti,
- 132-KB RAM paměti,
- ST-LINK, pro nahrávání programů,
- ...

Jedná periferie, kterou je nutno připojit jest LCD.

### 2.1.1 LCD

Připojení LCD k STM32F4-Discovery je provedeno přes piny dle tabulky 1. Jedná se konkrétně LCD 1602A verze 1.3.

LCD	STM32F4	LCD	připojení
RS	PE3	VSS	GND
RW	PE4	VDD	+5v
E	PE5	D0	NC
D4	PE6	D1	NC
D5	PE7	D2	NC
D6	PE8	D3	NC
D7	PE9	A	+5v
		K	+5v
		V0	Napěťový dělič

Tabulka 1: Připojení LCD k STM32F4 Discovery

Tabulka 2: Připojení dalších pinů LCD

Pro funkčnost LCD je nutno ještě zapojit piny dle tabulky 2, jinak nebude nice zobrazeno.

## 2.2 Popis softwaru

Vývoj je činěn v  $\mu$ Vision IDE. Jako knihovny použijeme:

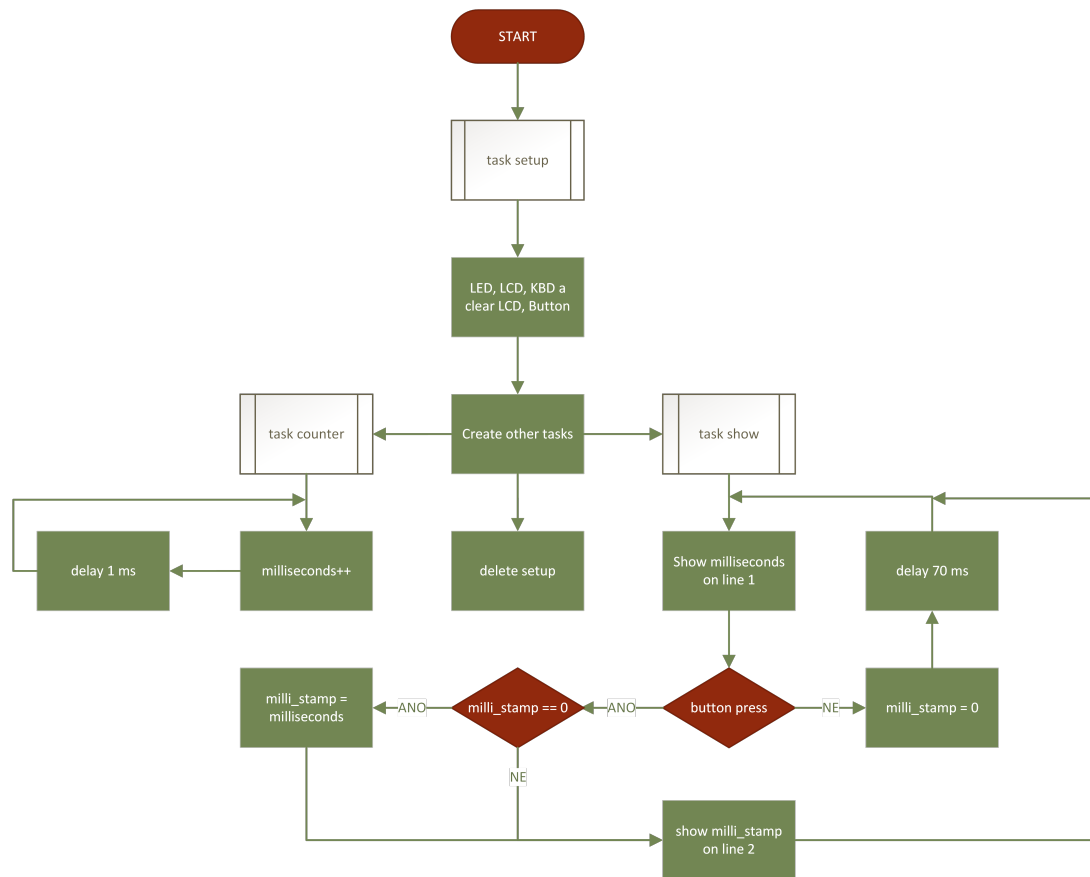
**stm32 kit** Knihovna nám umožní provést abstrakci nad CMIS;

**RTX4** Vytvoření RTOS, namísto superloopy;

### 2.2.1 RTOS

**Real-Time Operační Systém** je druh operačních systému který nám umožňuje práci s kriticky náročnými požadavky, jelikož u GPOS není zaručeně dáno (kromě IRS), že se činnost provede do času  $k$ . RTOS definuje tasky<sup>1</sup>, které jsou zaměňovány dle R-R algoritmu.

### 2.3 Vývojový diagram



Obrázek 2: Vývojový diagram

<sup>1</sup>Někdy je "task" chápán jako synonymum ke slovu "thread".

## 3 Řešení

Vytvořil jsem dva tasky, kde: první počítá oběhnuté milisekundy, druhý task zobrazuje na LCD uběhnutý čas, a kontroluje zmáčknutí uživatelského tlačítka.

### 3.1 Zdrojový kód

```
#include "stm32_kit.h"
#include "stm32_kit/lcd.h"
#include "stm32_kit/keypad.h"
#include "stm32_kit/button.h"
#include "stm32_kit/led.h"

#include <stdio.h>

OS_TID g_t[2];
size_t millicesocds = 0;

void showMilli(int millisecods, int line){
    char buff[LCD_COLS+1] = {0};
    snprintf(buff, LCD_COLS, "%02u:%02u:%02u",
        (millisecods/(1000*60))%60,
        (millisecods/1000)%60,
        millisecods%1000
    );

    if(line == 0) LCD_set(LCD_LINE1);
    if(line == 1) LCD_set(LCD_LINE2);

    LCD_print(buff);
}

__task void counter() {
    millicesocds = 0;

    while(1) {
        millicesocds++;
        delay_ms(1);
    }
}

__task void show() {
    size_t mili_stamp = 0;
    while(1)
    {
```

```

        showMilli(millicsocs , 0);

        if(io_read(USER_BUTTON))
        {
            if(!mili_stamp)
                mili_stamp = millicsocs;
            showMilli(mili_stamp , 1);

            continue;
        }
        mili_stamp = 0;

        delay_ms(70);
    }
}

__task void setup() {
    LCD_setup();
    LCD_set(LCD_CUR_NO_BLINK);
    LCD_set(LCD_CUR_OFF);
    KBD_setup();
    LED_setup();
    BTN_setup();

    g_t[0] = os_tsk_create(counter , 0);
    g_t[1] = os_tsk_create(show , 0);

    os_tsk_delete_self();
}

int main()
{
    os_sys_init(setup);
}

```

## 4 Hodnocení

Výsledek hodnotím kladně, podařilo se vyřešit zadání, v jednoduchém kódu.

### 4.1 Úspěchy/neúspěchy

Úspěšnost vidím ve splnění zadání, neúspěchy mohou být viděny v neoriginalitě řešení.

## 4.2 Další rozšíření

Jako další rozšíření vidím:

- Přidání historie časů, a jejich následné prohlížení;
- Počítání doby, jež uběhla mezi časy;
- Uživatelské tlačítko bude v separátním tasku.