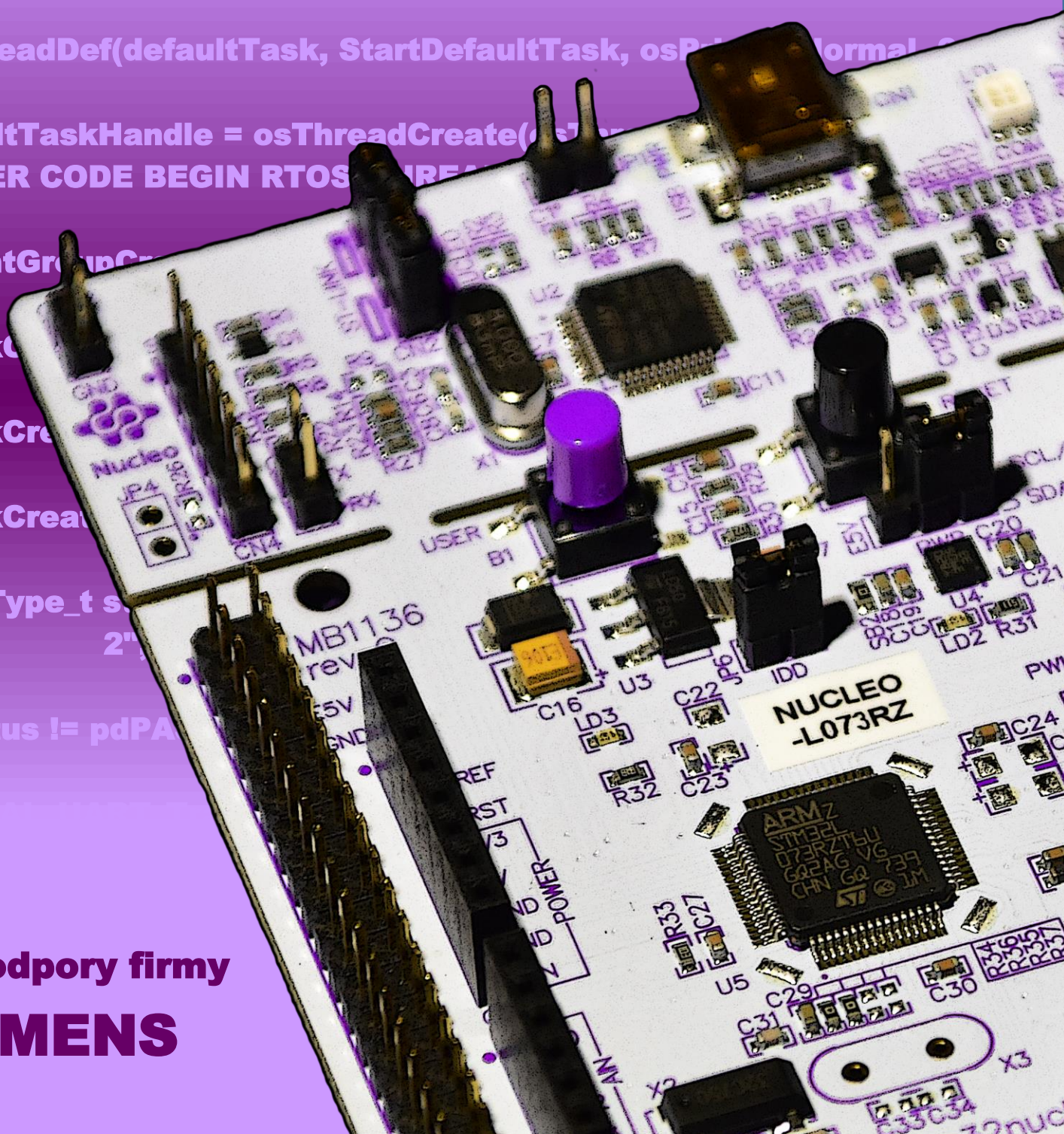
Používáme FreeRTOS

na mikrokontroleru STM32

Ing. Vojtěch Skřivánek

```
* Create the thread(s) */
/* definition and creation of defaultTask */

osThreadDef(defaultTask, StartDefaultTask, os

defaultTaskHandle = osThreadCreate(
/* USER CODE BEGIN RTOS

xEventGr

xTaskC

xTaskCre

xTaskCrea

BaseType_t s
                2"

if(status != pdPA
{
```



za podpory firmy

SIEMENS

Používáme FreeRTOS na mikrokontroleru STM32

Ing. Vojtěch Skřivánek

Vydání první 2023

ISBN: 978-80-7539-153-7 (PDF verze)

Poděkování

Děkuji firmě

SIEMENS

za podporu při psaní této knihy.

Obsah

Kapitola 1

Úvod

Tato kniha pojednává o operačním systému ***FreeRTOS***, který je v současné době velmi rozšířený. Je to především díky jeho volné licenci a dostupnosti pro velké množství mikrokontrolerů. V knize je popsáno, jak tento operační systém funguje, a jaké možnosti a funkce nabízí vývojáři k použití.

Kniha se nesnaží pokrýt všechny teoretické aspekty operačního systému. Soustředí se především na praktické využití nejčastěji používaných nástrojů. Proto se v knize nepopisují ani rozšířené, nepříliš často využívané funkce systému. Prostor je raději přenechán praktickým příkladům, v nichž jsou často ukázána skrytá nebezpečí. Ta mohou při špatném použití nástrojů operačního systému neblaze ovlivnit běh programu.

Příklady v této knize využívají jako platformu vývojovou desku ***NUCLEO***-*L073RZ*. (Použita v knižní sérii "Programujeme STM32".) Nicméně příklady jsou na hardwaru závislé pouze minimálně – pro blikání LED, použití sériové linky na komunikaci s PC. Proto je lze snadno převést i na jiné kontrolery či systémy, pro něž je ***FreeRTOS*** uzpůsobeno. Pro seznámení s operačním systémem a jeho nástroji navíc často stačí příklad pouze pročíst.

Kniha vychází především ze dvou dokumentů uvedených v závěru knihy. Teoretická část je v této knize značně zkrácena, ale není opomenuto nic důležitého. Naopak díky většímu důrazu na praktické příklady je občas názorně vidět problém, který při pouhém teoretickém popisu funkce nástrojů není zřejmý. A právě pomocí vysvětlení neočekávaného chování ukázkového programu si čtenář uvědomí, jak operační systém funguje a nač si při jeho používání dát pozor.

Předpokládá se, že čtenář již má zkušenosti s programováním mikrokontrolerů ***STM32***, ví, jak se pracuje s knihovními funkcemi ***HAL*** a umí používat ***STM32CubeIDE***.

1.1 Struktura knihy

Kniha se ve svých jedenácti kapitolách zabývá různými tématy operačního systému ***FreeRTOS***. V první, obecné kapitole je teoretický úvod o operačním systému a jeho struktuře.

Každá z následujících osmi kapitol se zabývá jedním z nástrojů (či skupinou podobných nástrojů), které může programátor využít. Každá tato kapitola má krátký teoretický úvod k danému nástroji.

Co je však důležitější, každá kapitola či podkapitola obsahuje příklady. Příklady se ve své komplexnosti liší v závislosti na náročnosti či důležitosti tématu. Často se v příkladech experimentuje, či se jejich program mění, aby byl vidět rozdíl v jeho chování. Důvodem změn je často intuitivní použití funkcí operačního systému, které však vede k neočekávanému běhu programu.

Téměř v každém příkladu se nachází jeden či více grafů názorně zobrazujících běh programu, které jsou velmi důležité k pochopení způsobu funkce operačního systému. Všechny knihovní funkce operačního systému použité v příkladech jsou v textu detailně popsány.

Každá kapitola předpokládá znalosti z kapitol předchozích, a to především u příkladů. Není tedy doporučeno kapitoly přeskakovat, zvláště pak v případě, kdy čtenář nemá předchozí zkušenosti s žádnými kontrolery a operačními systémy.

1.2 Fonty textu

Anglické zkratky, názvy funkcí a parametrů jsou v textu psány tučnou kurzívou - např. ***xTaskCreate***. Ač by bylo konzistentnější používat v knize buď pouze české, nebo anglické názvosloví, existuje-li český ekvivalent (např. ***queue*** = fronta), je použit, jelikož lépe zapadne do věty, která je pak srozumitelnější.

V obrázcích či grafech jsou některé části číselně označeny. Na tato označení je v textu odkazováno tučně psaným číslem v závorce – např. **(1)**.

Názvy vláken a funkcí jsou psány kurzívou a názvy vláken (***Task*** = vlákno) začínají velkým písmenem – např. *Vlakno1*, *uartOdesli*.

Kapitola 2

Vývojové nástroje

K programování mikrokontrolerů jsou zapotřebí tři základní nástroje.

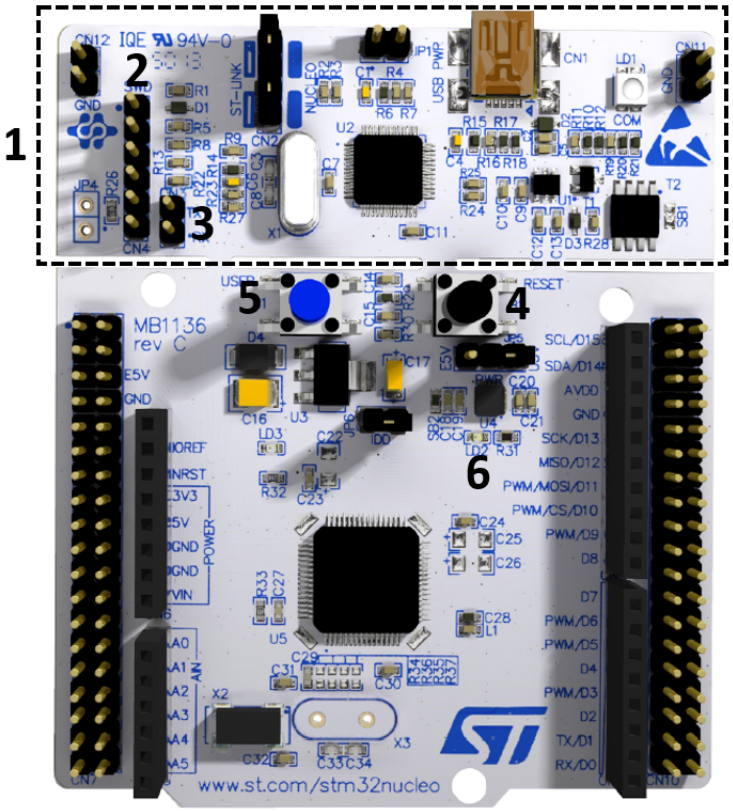
Prvním z nich je samozřejmě samotný mikrokontroler. Nejsnazším způsobem, jak mít k dispozici funkční mikrokontroler připravený k programování a používání jeho periferií, je opatřit si vývojovou desku. Firma STMicroelectronics nabízí širokou škálu cenově příznivých vývojových desek. Mezi nejznámější patří řady desek ***Nucleo*** a ***Discovery***. Ty se od sebe liší tím, že zatímco ***Nucleo*** obsahuje převážně pouze mikrokontroler a konektory, ***Discovery*** dost často skýtá také rozličné senzory, tlačítka, LED, displeje a jiné prvky, připojené přímo ke kontroleru a připravené k okamžitému použití.

Druhým potřebným nástrojem je programátor mikrokontroleru, který dokáže nahrát binární kód z PC do paměti programu čipu. Tento programátor je možné koupit samostatně, ale výhodou všech vývojových desek firmy STMicroelectronics je, že každá z nich má v sobě programátor a ***debugger*** (používaný k ladění programu) již integrovaný.

Zbývá poslední nástroj, ačkoliv jde vlastně o dva, které je však možno opět získat v jednom balíčku. Prvním je vývojové prostředí, ve kterém je možné snadno psát kód a spravovat projekt programu. Druhým je překladač zdrojového kódu, psaného v jazyce C, do strojového binárního kódu, kterému rozumí daný kontroler. Firma STMicroelectronics nabízí zdarma své vývojové prostředí ***STM32CubeIDE***, ve kterém je možné napsat zdrojový kód, přeložit jej do strojového a pomocí programátoru jej nahrát do programové paměti mikrokontroleru.

2.1 Vývojová deska

Vývojová deska, která je použita na všechny příklady v této knize, je z řady desek ***Nucleo*** s názvem ***NUCLEO-L073RZ***.



Vývojová deska NUCLEO-L073RZ [1]

Deska obsahuje, mimo mikrokontroleru ***STM32L073RZ***, také již zmíněný programátor a debugger sloužící k ladění programu **(1)**. Programátor se nachází v horní části desky, kterou je možné od spodní odlomit. K programování kontrolerů mimo vývojovou desku slouží konektor ***CN4*** v levé části **(2)**. Horní část také obsahuje převodník komunikace z ***USB*** na ***UART***, jehož vývody ***CN3*** jsou vpravo od programovacího konektoru **(3)**. Ani jeden z těchto konektorů není třeba používat, jelikož programátor i převodník jsou s kontrolerem spojeny můstky mezi horní a dolní částí desky. Převodník je spojen s periferií kontroleru ***USART2***, kterou budeme používat ve všech příkladech komunikujících s počítačem.

Kromě LED, které signalizují funkci programátoru a správné napájení, je na desce umístěna také jedna uživatelská LED ***LD2*** **(6)**, kterou lze pinem mikrokontroleru ovládat. Tuto LED budeme používat ve všech příkladech, kde se bude pracovat s nastavením výstupní hodnoty signálu. LED je připojena na pin ***PA5***.

Pro příklady v této knize není potřeba znát další detaily o této vývojové desce ani na ní nic zapojovat. Bude stačit ji pouze připojit.

2.2 Vývojové prostředí

Jak již bylo zmíněno, výrobce čipu poskytuje zdarma vývojové prostředí ***STM32CubeIDE***, ve kterém jsou vytvořeny všechny příklady této knihy.

Toto prostředí v sobě obsahuje konfigurátor periferií čipu, programování a správu projektu, nahrání programu do mikrokontroleru a možnost jeho ladění (debuggování).

S instalací vývojového prostředí se automaticky nainstaluje i ovladač programátoru, který je umístěný na Nucleo desce, a překladač ze zdrojového kódu na strojový.

STM32CubeIDE je možné po registraci na webových stránkách firmy STMicroelectronics stáhnout zcela zdarma. Nejsnazším způsobem nalezení odkazu ke stažení je zadat do internetového vyhledávače heslo "STM32CubeIDE"a pravděpodobně hned první odkaz bude mířit na správnou stránku, kde najdete také návod na instalaci a manuál k použití.

Snad není nutné popisovat instalaci vývojového prostředí, která je plně automatická a nainstaluje i všechny nutné ovladače.

2.3 Shrnutí

Po přečtení této kapitoly by měl mít čtenář k úspěšnému naprogramování příkladů uvedených v této knize připraveny následující nejnutnější věci:

- vývojovou desku **NUCLEO-L07RZ**,
- **USB kabel** k připojení desky v počítači a
- počítač s nainstalovaným vývojovým prostředím **STM32CubeIDE**.

S touto nezbytnou výbavou, je možné se směle pustit do čtení následujících kapitol, které odhalují používání operačního systému ***FreeRTOS***.

Kapitola 3

FreeRTOS obecně

FreeRTOS (***Free Real Time Operating System***) je operační systém pracující v reálném čase určený především pro programy běžící na mikrokontrolerech. Operační systém je upraven pro mnoho překladačů a několik desítek architektur mikrokontrolerů.

Operační systém funguje jako skupina knihoven, jejichž funkce umožňují multitasking. Multitasking je prováděn pomocí vláken, jejichž program může probíhat paralelně. Operační systém dále nabízí širokou škálu nástrojů na synchronizaci vláken, přenos dat mezi nimi a další užitečné možnosti.

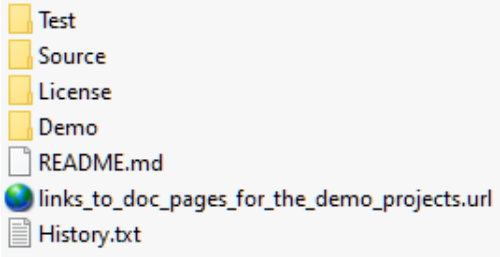
FreeRTOS má otevřenou licenci, je tedy zcela zdarma a je možné jej využít pro komerční aplikace.

Operační systém je možné stáhnout z webové stránky: https://www.freertos.org/index.html.

3.1 Struktura souborů

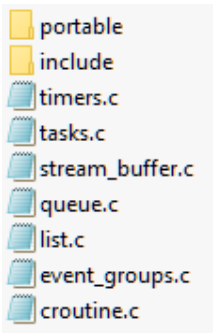
FreeRTOS funguje jako skupina knihoven psaných v jazyce C. Pokud chceme využívat operační systém, musíme se v našem programu odkázat na tyto knihovny, respektive jejich hlavičkové soubory.

Po stáhnutí archivu s operačním systémem z odkazu uvedeném výše v něm nalezneme tento obsah.



Ve složce ***Demo*** se nachází velké množství ukázkových funkčních projektů. Je možné se jimi nechat inspirovat při tvorbě vlastního projektu. Ale jak uvidíme, vývojové prostředí pro nás projekt již připraví.

Ve složce ***FreeRTOS/Source*** nalezneme následující složky a soubory:

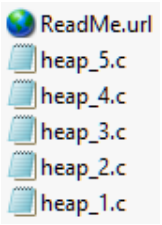


Knihovny se dělí na dvě kategorie. První skupina je závislá na architektuře kontroleru a je nutné zvolit správnou skupinu souborů, aby program fungoval. Na obrázku se jedná o soubory ve složce ***portable***. V této složce jsou různé verze knihovních funkcí závislých na hardwaru a překladači. V naší knize pracujeme s překladačem ***GCC*** a kontrolerem s jádrem ***ARM Cortex-M0+***. Proto budou v našich projektech použity soubory ze složky ***FreeRTOS/Source/portable/GCC/ARM_CM0***. Dále tato složka obsahuje schémata správy paměti, kterým se budeme věnovat v další kapitole.

Druhou kategorii tvoří knihovny, které nejsou závislé na hardwaru. Fungují stejně na všech kontrolerech, všech architekturách. Na obrázku jde o složku ***include*** a všechny soubory s příponou ***".c"***. Ne na všechny knihovny je nutné odkazovat z hlavního programu. Pokud například nebudeme využívat časovače, jsou soubory ***timers.h*** a ***timers.c*** zbytečné. Dva soubory jsou ale nutné vždy – ***tasks.c*** a ***list.c***. Bez nich operační systém fungovat nebude. Ostatní jsou volitelné podle toho, chceme-li využít jejich funkcí.

Každý projekt musí obsahovat hlavičkový soubor ***FreeRTOSConfig.h***. V tomto souboru se musí nacházet nezbytné konfigurační konstanty. Není doporučeno tento soubor připravovat vlastním úsilím. Lepším řešením je zkopírovat tento soubor z vhodného příkladu z archivu operačního systému. Ideální řešení nabízí vývojové prostředí ***STM32CubeIDE***, které za nás konfigurační soubor vytvoří.

Zvláštní pozornost ještě věnujeme složce ***FreeRTOS/Source/portable/MemMang***. Ta v sobě obsahuje hlavičkové soubory schémat správy paměti.



O schématech a správě paměti pojednává následující kapitola.