

Sada demo aplikací nad FreeRTOS

Hardware: ARM-FITkit3

Vedoucí: Dr. Strnadel

Funkcionalita

Cíl: Každému ze čtyř hlavních tlačítek (*SW2*, *SW3*, *SW4*, *SW5*) je přiřazena jedna LED dioda (*D9*, *D10*, *D11*, *D12* respektive). Po stisknutí tlačítka se příslušná dioda rozsvítí na právě jednu sekundu. Nezáleží na časování stisknutí tlačítek ani na jejich pořadí. Tlačítka lze mačkat opakovaně. Každé jedno stisknutí odpovídá jedné sekundě světla příslušné diody.

Implementace: Toto je implementováno za pomoci systému FreeRTOS. Pro rosvícení každé z diod je využit právě jeden Úkol (*Task*). Pro obsluhu tlačítek je využita takzvaná *Interrupt Service Routine*. Jedná se o handler obsluhující GPIO přerušení na portu *GPIO E*.

Pro komunikaci mezi handlerem a úkoly jsou využity Počítající Semaforey (*Counting Semaphore*). Ty zajišťují:

- konzistenci dat při změnách kontextu
- bezpečné přecházení z mezi úrovněmi obsluhy
- tím, že je počítající, lze počítat počet přerušení, které proběhla a vytvořit tak efektivně frontu pro tato přerušení. Úkoly pak proběhnou přesně tolikrát, kolikrát bylo tlačítko zmáčknuto.

Pro časování diod je použito časované blokování úkolů. Každý úkol je uspán na určitý počet takzvaných *ticků*, což jsou v podstatě periodická přerušení generovaná systémem FreeRTOS. Jejich frekvence je nastavena v konfiguračním souboru *FreeRTOSConfig.h*, pro určení správného počtu ticků lze využít makro pro převod z milisekund.

Problematika: Hlavními problémy, které bylo potřeba pro vytvoření projektu řešit byly:

- správná konfigurace GPIO portů pro obsluhu periférií
- správné vytvoření úkolů
- správné vytvoření a nakonfigurování přerušení
- správná implementace komunikace mezi ISR a úkoly.

Soubory

Pro vývoj aplikace bylo využito IDE MCUXpresso s kompletní sadou jejich nástrojů. Proto je součástí projektu spousta souborů, které nebyly psány ručně, ale byly generovány pomocí těchto nástrojů. Je proto obtížné odhadovat procentuální vlastnictví těchto souborů. Práci s nástroji jsem prováděl já, ale soubory jsem nepsal.

Jedná se o soubory ve složce *board*, až na dvě výjimky. Soubory *peripherals.c* a *peripherals.h* byly napsány ručně, protože MCUXpresso Configuration Tools měl problémy rozpoznat periférii GPIO a proto kód pro nastavení přerušení nešel vygenerovat. Právě tento problém značně zpomalil tvorbu projektu.

Důležité soubory:

- **source/MK60DN512xxx10_Project.c**
- vše v **board/...**
- **freertos/FreeRTOSConfig.h**

Instalace

Pro instalaci (překlad, naflashování...) lze využít například právě MCUXpresso IDE. Je třeba archiv importovat jako projekt, poté sestavit a naflashovat.

Skripty pro ruční build nejsou k dispozici.

Verze částí softwaru mohou být velmi důležité a je na ně třeba dát pozor.

- MCUXpresso IDE - v11.0.1
- **MCUXpresso SDK - v2.2.0** pro MK60DN512xxx10

Při tvorbě projektu bylo právě tohle také problémem. Škola využívá Kinetis Design Studio v3.0.0, které ale spoustu funkcí nepodporuje. Také nepodporuje některá SDK, která NXP poskytují. Sehnat ale staré SDK je velmi obtížné. Demo kódy, které škola poskytuje, jsou zase psány pro jinou verzi SDK (i než je například na návodu na Fitušce) a pro nové verze je potřeba je potřeba upravit. Ani návody na zprovoznění samotného kitu a tvorbu nových projektů nejsou 100% odpovídající.

Zdroje

- <https://www.freertos.org/> - guide, documentation, forums
 - <https://community.nxp.com/> - forums
 - http://www.lirmm.fr/~bosio/HMEE209/TP/161204_Mastering_the_FreeRTOS_Real_Time_Kernel-A_Hands-On_Tutorial_Guide.pdf
 - http://cache.freescale.com/files/32bit/doc/ref_manual/K60P144M100SF2V2RM.pdf
 - <https://www.nxp.com/docs/en/user-guide/MCUXIDEUG.pdf>
- a další...