IMP projekt

ARM-FITkit3: Demonstrace využití rozhraní USB

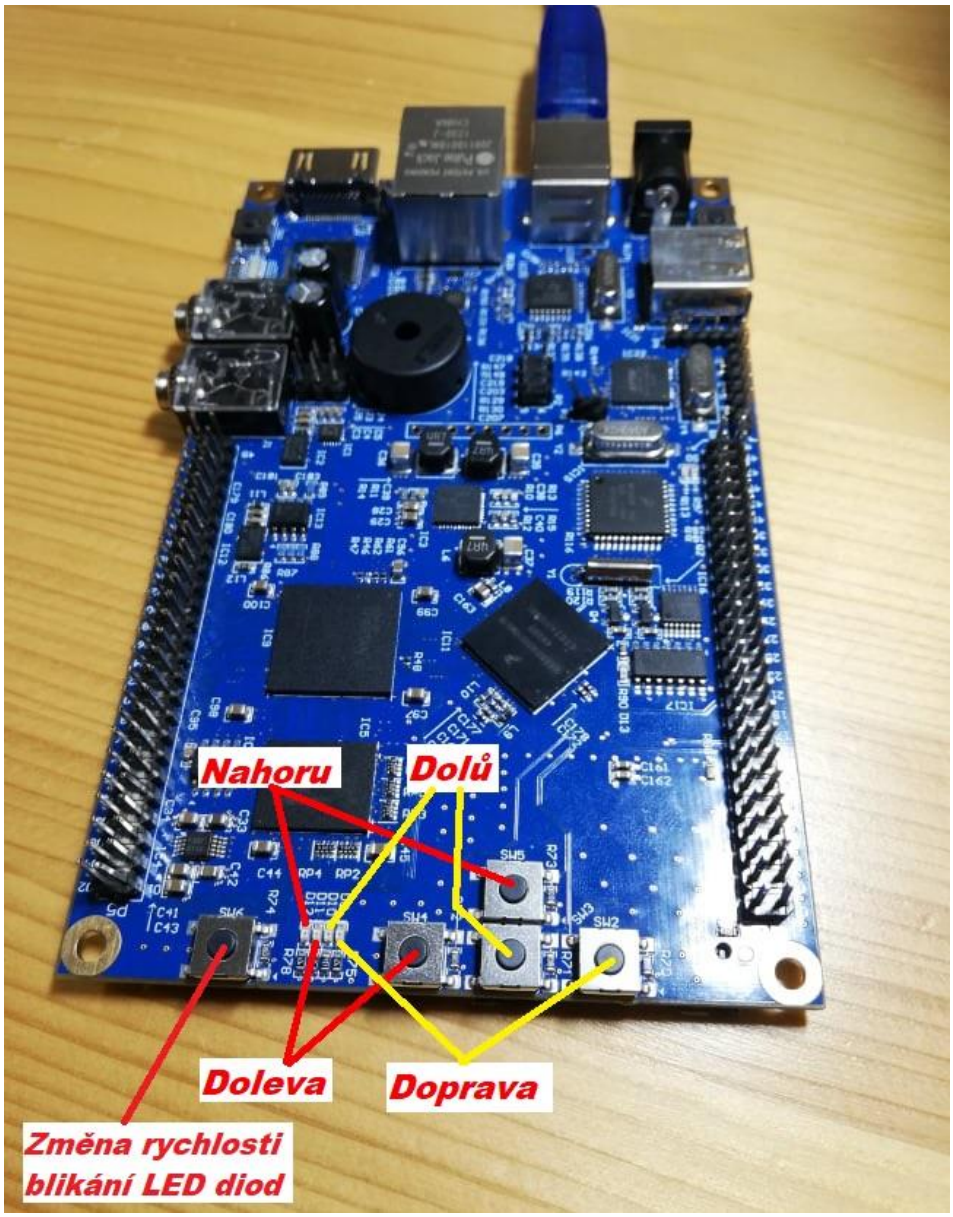
Petr Marek (xmarek69)

22.12. 2019

Popis projektu

Projekt má za cíl demonstrovat využití rozhraní USB. FITkit ovládá kurzor PC myši pomocí tlačítek, kterými je FITkit osazen. Každé ze 4 tlačítek pohybuje kurzorem myši v jednom směru. Při stisku kteréhokoliv tlačítka se pomalu začne pohybovat myš daným směrem. Pro zrychlení pohybu je potřeba tlačítko pustit, a opětovně (bezprostředně) znovu stisknout – tedy udělat „dvojklik“. Kromě pohybu myši se rovněž s každým stisknutím tlačítka rozsvítí některá z LED diod, rovněž již umístěných na desce FITkitu. Při pomalém pohybu příslušná dioda svítí, při rychlejším pohybu (při dvojkliku tlačítka) bliká. Rychlost blikání jde přepnout tlačítkem 5 na desce FITkitu (přepíná se mezi 3mi rychlostmi).

Ovládání



Schémata a spouštění

K FITkitu není potřeba nic připojovat.

Nejprve je potřeba vytvořit adresář:
C:\TEMP\test_mqx\Freescale_MQX_4_2_FITKIT_KDS300

Do adresáře Freescale_MQX_4_2_FITKIT_KDS300 je poté potřeba vložit jednotlivé složky MQX RTOS (build, config atd). Je potřeba sestavit knihovny MQX RTOS. Následně nahradit soubory z adresáře: „C:\TEMP\test_mqx\Freescale_MQX_4_2_FITKIT_KDS300\usb\device\examples\hid\mouse“ přiloženými soubory

Je potřeba importovat projekt (soubor xmarek69.wsd), importovaný projekt s názvem „hid_mouse_dev_fitkit“ sestavit a spustit (debugovat).

Způsob řešení a implementace

Projekt je založen na MQX RTOS. Jelikož jsem nenalezl žádný funkční způsob, jak založit nový MQX RTOS projekt, tak jsem jej vytvářel přímo z importované ukázky USB myši z MQX RTOS. 2 soubory v této ukázce slouží pro správné nastavení a funkci USB, ty jsem nijak neupravoval. Naopak soubory přímo implementující zpracování a zasílání dat přes USB (soubory *Sources/mouse.c* a *Sources/mouse.h*) Jsem značně upravil (přes 60% kódu z těchto souborů je můj vlastní). V těchto dvou souborech jsem rovněž využil malé části kódu z jiných ukázek MQX RTOS.

Program je implementován tak, že se při spuštění FITkitu vytvoří jedna úloha *Main_Task()* (má nastavený auto start). Ta zavolá funkci *TestApp_Init()*, ve které se provede konfigurace USB a inicializace globálních proměnných, tlačítek a LED diod. Následně se v nekonečném cyklu volá stále dokola funkce *checkButtons()*, která kontroluje (technikou pooling), která tlačítka jsou stisknuta a na jejich základě nastaví proměnné (*dx* a *dy*), které později ve funkci *move_mouse()* rozhodují o tom, jak se má ukazatel pohybovat (*move_mouse()* je volána v rámci callback funkce). V rámci funkce *checkButtons()* jsou rovněž aktivovány LED diody. Efektu blikání je dosaženo tím, že se počítá, kolikrát byla funkce *checkButtons()* zavolána s daným tlačítkem stisknutým – a každé N zvolání se stav (dioda svítí/nesvítí) přepne. Toto N je dáno proměnnou *blink_speed*, jejíž hodnota je rovněž upravována ve funkci *checkButtons()* (při každém stisku příslušného tlačítka se *blink_speed* přepíná mezi 3mi různými hodnotami – „rychlostmi“).

Závěr

Projekt má všechny popsané funkce plně funkční. Jeden z problémů, se kterými jsem se setkal byl, že je v MQX RTOS pro FITkit definované přerušení pouze na tlačítku 1 (přiřazené pohybu kurzoru doprava), tudíž jsem nemohl změnu rychlosti blikání implementovat pomocí přerušení na tlačítku 5, což by bylo příhodné, ale rovněž jsem musel použít opakovanou kontrolu stavu tlačítka jako u ostatních tlačítek. Druhým problémem bylo, že jsem (z časových důvodů) nemohl implementovat složitější způsob ovládání kurzoru (například pomocí akcelerátoru).