

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií

Mikroprocesorové a vestavěné systémy
2021/2022

Stopky

1. Cíl projektu	3
1.1 Hlavní funkce	3
1.2 Ovládání	3
2. Implementace	4
2.1 Knihovny	4
2.2 Periferie	4
4. Testování	5
3. Závěr	5
7. Použitá literatura	6

1. Cíl projektu

Cílem projektu je vytvořit aplikaci jednoduchých stopek na FitKit 2.0 [5] s použitím mikrokontroleru rodiny MSP430 [4]. Pro dosažení těchto cílů je nutné využít nějaké vstupní zařízení pomocí kterého uživatel bude moci aplikaci ovládat a nějaké výstupní zařízení pomocí kterého uživatel uvidí stav systému. Pro tyto účely byly použity klávesnice fitkitu a LCD displej fitkitu. Způsob zobrazení na LCD je navržen ve formátu MM:SS.ss

1.1 Hlavní funkce

- spustit/zastavit stopky
Umožňuje spustit stopky, nezávisle na jejich stavu. Stejné tlačítko slouží na zastavení stopek, když stopky běží. Pokud jsou stopky pozastaveny a mají jinou hodnotu než 00:00.00, aplikace po stisknutí tlačítka bude pokračovat v měření
- uložit mezičas
Umožňuje uložit N mezičasů do paměti, při spuštěných stopkách. Z implementačních důvodů N omezeno na 9
- restart systému
Umožňuje restart systému - zastaví stopky, vynuluje je a odstraní všechny mezičasy
- listování mezi mezičasy
Umožňuje listovat mezi uloženými mezičasy. Přičemž tlačítkem se posouváme o 1 mezičas dále a po mezičasu 9. následuje opět mezičas 0. který není mezičasem ale absolutním časem stopek

1.2 Ovládání

Ovládání aplikace je implementováno pomocí klávesnice Fitkitu a mapování kláves je následující:

Tlačítko	Funkce
A	Spuštění/Zastavení stopek
B	Listování mezi mezičasy
#	Restart systému
*	Uložení mezičasu, na LCD zobrazí * pokud se podaří (dostatek místa), jinak zobrazí E

2. Implementace

Pro implementaci bylo využito periférií dostupných na kitu a knihoven dostupných ve vývojovém prostředí QDevKit [3], při použití SVN Fitkitu [2]. Stěžejní částí pro aplikaci je TIMER, konkrétně TIMER_A, který v aplikaci generuje periodické přerušení, které se vyvolá každých 1/64s. Každé přerušení je zkontrolován stav systému a následně zda stav povolí, bude přidána 1 do proměnné, které implementuje současný naměřený čas v sekundách, které je uložen pomocí FIXED POINT formátu, kde posledních 6 bitů je použito jako desetinných. Po dokončení přičtení je vyvolána pomocná funkce, která vypíše současný čas na LCD. Led5 je v tomto přerušení použita jako identifikace životu timeru. Toto bude ale provedeno pouze pokud jsou stopky spuštěné (Stavová proměnná counting je true) v opačném případě se nic neprovede. Obsluha klávesnice je implementována pomocí while smyčky, která čeká na příchod tlačítka (Aktivní čekání), pokud nějaké přijde nejdříve bude ověřeno zda opravdu k stisku došlo, pokud ano bude volána obslužná rutina pro jednotlivá tlačítka. Tlačítko A invertuje proměnnou stavovou counting a nastavuje současnou obrazovku na 0. Tlačítku B zvyšuje číslo stránky o 1 a následně provádí mod 10. Stránka 0 se používá pro zobrazení aktuálního času a ostatní se používají pro zobrazení uložených časů. Po stisku tlačítka B je automaticky stránka zobrazena pomocí pomocné funkce. Data pro zobrazení jsou čtena z array o velikosti 10 obsahující uint32_t, kde číslo na pozici 0 je rezervováno pro současný čas stoppek. Tlačítko # nuluje počet dostupných stránek a současný čas stoppek, zároveň stopky zastavuje. Tlačítko * kopíruje v array s časy z pozice 0 na pozici která je rovna stránce která ještě není použita po dokončení na LCD na krátký okamžik zobrazí *, pokud nebude volná stránka zobrazí E. Po dokončení subrutin aplikace čeká na uvolnění tlačítka. Při dopočítání na číslo 99:59.98 bude počítání automaticky zastaveno, protože by se čas již nevlezl správně na display. Tento čas by se dal zapsat ve formátu uloženém v aplikaci jako $(99 \cdot 60 + 59 \ll 6) \mid 0b111111$.

2.1 Knihovny

- fitkitlib
- mcu/libs/keyboard/
- mcu/libs/lcd/

2.2 Periferie

- TIMER A (MCU)
- SPI_adc (FPGA - VHDL) 2x
- keyboard_controller (FPGA - VHDL)
- lcd_raw_controller (FPGA - VHDL)
- keyboard 4x4 (fitkit)
- LCD displej (fitkit)

4. Testování

Aplikace byla testována, zda měření odpovídá jiným stopkám, a dále byly testovány všechny funkce aplikace. Odkaz na video z testování:

[Primární]

<https://youtu.be/e6cAL4MOVYM>

[Sekundární]

<https://drive.google.com/file/d/1IleTWHuAZXRyceqaVcwIWeaMWYZggsIT/view?usp=sharing>

3. Závěr

Aplikace byla implementována tak aby splnila všechny cíle aplikace a tak aby byla schopna chodu, nicméně některé aspekty nejsou implementovány tak dobře jak dobře by implementovány být mohly. Obsluha klávesnice by mohla být udělána pomocí přerušení a tím by mohl být ušetřen nějaký mikrokontrolerový čas a tím by se snížila spotřeba energie. Přesčas kdy procesor pouze čeká na přerušení od timeru nebo na příchozí tlačítko z klávesnice by procesor mohl být v sleep modu.

7. Použitá literatura

[1] Mikroprocesorové a vestavěné systémy - IMP. *IMP* [online]. Brno: VUT FIT, 2021, 2021 [cit. 2021-12-17]. Dostupné z: <https://www.fit.vut.cz/study/course/14662/>

[2] SVN Repository - FITkit. *FITkit* [online]. Brno: VUT FIT, 2021, 2021 [cit. 2021-12-17]. Dostupné z: <https://merlin.fit.vutbr.cz/FITkit/svn.html>

[3] Návod - FITkit. *FITkit* [online]. Brno: VUT FIT, 2021, 2021 [cit. 2021-12-17]. Dostupné z: <https://merlin.fit.vutbr.cz/FITkit/navody.html>

[4] MSP430x2xx Family User's Guide. MSP430x2xx Family User's Guide [online]. Dallas, Texas: 2009, 2013 [cit. 2021-12-17]. Dostupné z: https://www.ti.com/lit/ug/slau144j/slau144j.pdf?ts=1639723415945&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FMSP430G2302

[5] schematic_v20.pdf. *FITkit* [online]. Brno: VUT FIT, 2021, 2021 [cit. 2021-12-17]. Dostupné z: http://merlin.fit.vutbr.cz/FITkit/download/schematic_v20.pdf