

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Mikroprocesorové a vestavěné systémy

ARM-FITkit3: Hra HAD

Obsah

1	Úvod	2
2	Popis a návrh riešenia	2
2.1	Popis	2
2.2	Návrh	2
3	Implementácia	3
3.1	Detailnejší popis	3
4	Spustenie	3
5	Demonštrácia	3
6	Referenčná literatúra	4

1 Úvod

Zjednodušená implementácia hry HAD v jazyku C na platforme FitKit3 s maticovým displejom o veľkosti 16x8. Zmyslom je oboznámiť sa s princípom tvorby vstavaných aplikácií na mikrokontroléri Kinetis K60 s jadrom ARM Cortex-M4.

2 Popis a návrh riešenia

2.1 Popis

Aplikácia zobrazí na displeji hada predstavujúceho 4 za sebou zapnuté LEDky na displeji. Hada sa konštantne pohybuje na displeji, zatiaľ čo jeho smer sa ovláda tlačítkami.

2.2 Návrh

Aplikácia po inicializačných funkciách prejde do nekonečnej slučky, počas ktorej sa očakávajú prerušenia. Prvý typ prerušenia je od interného časovača, ktorý má úlohu zobraziť hada a jeho pohyb na displeji, druhý typ je od používateľa, ktorý stlačením smerových tlačidiel (konvenčne podľa polohy) vyvolá prerušenia a zmení smer hada.

Funkcie

- `SystemConfig()` : Konfigurácia príslušných port pinov, inputov a outputov
- `column_select(unsigned int col_num)` : Nastaví chcený stĺpec displeja
- `row_select(unsigned int row_num)` : Nastaví chcený riadok displeja
- `snakeInit()` : Nastaví 4 predvolené LEDky pri spustení aplikácie
- `delay(uint64_t bound)` : Filtrácia zákmitu po stlačení tlačítka
- `LPTMR0Init()` : Inicializácia LPTMR časovača
- `LPTMR0_IRQHandler(void)` : Obsluha prerušení časovača LPTMR
- `PORTE_IRQHandler(void)` : Obsluha prerušení tlačidiel

3 Implementácia

Aplikácia bola implementovaná pomocou cvičení, prednášok a manuálu.

3.1 Detailnejší popis

Aplikácia na začiatku nakonfiguruje funkciu `SystemConfig()` príslušné porty potrebné pre základnú funkčnosť. Nasledovná funkcia `snakeInit()` inicializuje pole predstavujúce hada s hodnotami jeho polohy. Nakoniec sa už len inicializuje časovač LPTMR funkciou `LPTMR0Init()` a aplikácia sa dostáva do nekonečnej slučky.

Časovač LPTMR vytvára v pravidelných intervaloch prerušenia, počas ktorých obsluha (`LPTMR0_IRQHandler()`) zobrazí jednotlivé segmenty hada na displeji a po istom počte prerušení (150 cyklov) sa had posunie (`move()`) v daný smer o jedno políčko. V prípade narazenia na okraj displeja sa had začne zobrazovať na protiľahlej strane.

Stlačenie ktoréhokoľvek zo 4 smerových tlačidiel vytvorí prerušenie ktoré sa obslúži v `PORTE_IRQHandler()` a nastaví nový smer hada. Táto funkcia kontroluje, ktoré tlačítko vyvolalo prerušenie a ešte zistí či je daný smer možný vzhľadom na momentálny smer.

Ďalšie použité funkcie sú `column_select(unsigned int col_num)` slúžiaca na výber stĺpca displeja, `row_select(unsigned int row_num)` na výber riadka displeja a `delay(uint64_t bound)` na filtrácia zákmitu po stlačení tlačítka.

4 Spustenie

Po otvorení *Kinetis Design Studio 3 IDE* vyberiete *File* → *Import* → *General* → *Archive file* a *Next*.

Vyberiete zip súbor s projektom a vyberiete projektový priečinok v ktorom ho chcete otvoriť (*Do existujúceho alebo vytvoríte nový*) a stlačíte *Finish*. Nakoniec ho skompilujete a spustíte, uistite sa, že FitKit3 zariadenie je správne napojené a rozpoznané.

5 Demonštrácia

Video-komentár k projektu: <https://nextcloud.fit.vutbr.cz/s/7DXEij4c5itJr9e>

6 Referenčná literatúra

<https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIMP-IT%2Fhwlabs&cid=13997>
<https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIMP-IT%2Flectures>
<http://www.fit.vutbr.cz/~simekv/schematics%20-%20FITkit%20v3.0.pdf>
<https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIMP-IT%2Fhwlabs%2FLab3%2Fkl05p48m48sf1rm.pdf&cid=13997>