



# MIKROPROCESOROVÉ A VESTAVĚNÉ SYSTÉMY

## 2017/2018

### **Hodiny s budíkem na bázi modulu Real Time Clock**

# Úvod

Táto práca sa zaoberá vstavanou aplikáciou pre mikrokontrolér Freescale K60 na doske platformy FITkit3. Účelom aplikácie je využitie modulu RTC (Real Time Clock) pre zobrazenie aktuálneho času a možnosti nastavenia alarmu. Ďalšími funkciami aplikácie sú zapnutie a vypnutie funkcie budenia, výber z 3 vstavaných zvukových a 3 svetelných signalizácii alarmu, možnosť nastavenia počtu opakovaných pokusov o budenie a časových odstupov medzi nimi.

## Ovládanie

Aplikácia sa ovláda pomocou terminálových volieb a HW tlačidla na doske. Pre ovládanie pomocou terminálu je nutné spojenie USB káblom medzi počítačom a doskou (port USB-B). Nastavenia sériovej komunikácie sú: rýchlosť 115 200 Baudov, 8 data bitov, 1 stop bit, bez parity a bez kontroly toku.

### Terminálové voľby:

- 1
  - Zobrazí aktuálny čas vo formáte hh:mm:ss, pričom zobrazenie času sa aktualizuje každú sekundu, až kým sa nepreruší jeho zobrazenie ľubovoľnou klávesou.
- 2
  - Nastavuje aktuálny čas. Po zvolení tejto voľby sa očakáva vstup od užívateľa vo formáte hh:mm:ss. Oddelovač jednotiek času „:“ je doplňovaný automaticky.
- 3
  - Nastavuje čas pre alarm. Po zvolení tejto voľby sa očakáva vstup od užívateľa vo formáte hh:mm:ss. Oddelovač jednotiek času „:“ je doplňovaný automaticky.
- 4
  - Zobrazí nastavený čas pre alarm vo formáte hh:mm:ss.
- 1
  - Voľba svetelnej signalizácie alarmu 0-3 (0 = žiadna svetelná signalizácia). Svetelnú signalizáciu je možné meniť aj počas jej behu.
- s
  - Voľba zvukovej signalizácie alarmu 0-3 (0 = žiadna zvuková signalizácia). Zvukovú signalizáciu je možné meniť aj počas jej behu.
- i
  - Nastavenie intervalu medzi jednotlivými pokusmi o budenie. Po zvolení tejto voľby sa očakáva vstup od užívateľa vo formáte hh:mm:ss, čo predstavuje daný interval. Napr. 00:10:00 znamená, že jednotlivé pokusy o budenie sa budú opakovať každých 10 minút (kým sa alarm nevypne). Oddelovač jednotiek času „:“ je doplňovaný automaticky.
- a
  - Nastavenie počtu opakovaných pokusov o budenie, 0-5.
- h
  - Zobrazenie nápovedy o ovládaní aplikácie.

## HW voľby:

- tlačidlo SW6
  - Vypína práve prebiehajúcu signalizáciu alarmu.
  - Zapnutie/vypnutie funkcie budenia (stand-by režim alarmu)

## Realizácia

Aplikácia bola implementovaná v jazyku C pomocou Kinetis Design Studio IDE. Program nie je založený na Kinetis Software Development Kit, ani žiadnom inom frameworku.

Pre sfunkčnenie sériovej komunikácie medzi FITkitom a počítačom s terminálom je využitý modul UART, kanál 5 na pinoch 8,9 portu E. Cieľová rýchlosť komunikácie je 115 200 Baudov – pomocou registrov BDL a C4 je dosiahnutá s pomerne nízkou chybou – 115 246 Baudov.

Na zobrazenie času je využitý modul RTC so sekundovými prerušeniami. Pre spustenie alarmu sa využíva prerušenie povolené registrom IER, spustené po dosiahnutí rovnosti hodnôt v registri času (TSR) a registri času alarmu (TAR). Zapnutie a vypnutie funkcie budenia je riešené práve pomocou povolenia daného prerušenia v registri IER. Ak príde prerušenie pre alarm, v jeho obsluhu sa povoľujú časovače PIT0 – pre svetelnú signalizáciu, PIT1 – pre zvukovú signalizáciu a PIT2 – pre vypnutie všetkých PIT časovačov po stanovenej dobe (7000 ms). Zapnutie a vypnutie funkcie budenia sa prepína v obsluhu prerušenia od tlačidla SW6 (port E, pin 11). Pohotovosť funkcie budenia je indikovaná LED diódou D12. V obsluhu prerušenia od tlačidla SW6 sa rieši aj vypnutie práve prebiehajúcej signalizácie alarmu, a to zakázaním časovačov PIT0-2, zodpovedných za prebiehajúcu signalizáciu.

Svetelná signalizácia je riešená pomocou MCU LED diód D9-D12 (port B, piny 2-5). Každý zo zvolených spôsobov svetelnej signalizácie je realizovaný striedavým preblikávaním rôznych diód v pravidelných 250 ms intervaloch. Interval je sledovaný pomocou čítača Periodic Interrupt Timer 0, ktorý v daných intervaloch pravidelne generuje prerušenia. Obsluhou prerušenia sa vždy rozsvieti iná LED dióda.

Zvuková signalizácia je realizovaná bzučiakom (port A, pin 4). Generovanie rôznych frekvencií je zaistené striedavým privádzaním oboch logických úrovní na jeho vstup, pričom dĺžka periódy sa rieši aktívnym čakaním procesora. Bzučiak v pravidelných 3000 ms intervaloch zahrá niekoľko tónov. Intervaly sú sledované pomocou časovača PIT 1 – generovaním prerušení.

Časovač PIT2 má okrem úlohy vypínania všetkých časovačov po dobe 7000 ms aj úlohu naplánovania ďalšieho spustenia alarmu – pokusu o budenie – po užívateľom nastavenom intervale. To sa realizuje zapísaním novej hodnoty do registra TAR, vypočítanej ako súčet aktuálnej hodnoty času z registru TSR a užívateľom stanoveného intervalu.

Keďže jeden deň má 86400 sekúnd a hodnota v registri času (TSR) sa každú sekundu inkrementuje – t.j. hodnotu 86400 môže viacnásobne prevýšiť, je potrebné vypočítať najprv relatívnu hodnotu času vzhľadom na deň, ak chceme zobrazit' aktuálny čas. Ak užívateľ zmení čas alarmu alebo aktuálny čas alebo zapne funkciu budenia, je nutné vypočítať a zapísať novú hodnotu do registru času alarmu (TAR). Táto hodnota musí byť vždy väčšia ako hodnota v registri TSR, aby sa alarm mohol v budúcnosti spustiť. Treba pri tom zohľadniť aj fakt, či má byť alarm spustený ešte v dnešný deň, alebo až na ten ďalší.

# Záver

Pri implementácii aplikácie je nutné vychádzať z dokumentácie pre mikrokontrolér K60, popisujúcej význam jednotlivých registrov, takisto aj zo schémy zapojenia čipu na doske. Bez nich by totiž realizácia aplikácie nebola možná. Napr. spoznanie komunikácie UART zabralo veľmi dlhú dobu, keďže nebolo hneď jasné, že k portu USB-B vedú práve piny 8, 9 z portu E, ktoré bolo treba nastaviť ako UART5 Tx/Rx. Riešenie daného problému by zabralo oveľa menej času pri pozornejšom čítaní schémy zapojenia.

Ďalším problémom, ktorý vznikol, bolo nepredvídateľné chovanie modulu RTC na doske FITkitu3. V registri TSR sa sekundy niekedy inkrementovali, inokedy zase nie. Daný problém bol vyriešený výmenou FITkitu za iný, na ktorom už modul RTC pracoval správne.

Výsledné riešenie bolo realizované so snahou o šetrenie zdrojov a čo najväčšiu modularitu – viacero krátkych funkcií, riešiacich 1 problém. Aplikáciu je preto možné s nevelkým úsilím v budúcnosti rozšíriť aj o ďalšie funkcie, ktoré by sa mohli zdať užitočné.