



**Předmět: IMP**

**ARM-FITkit3: Hodiny s budíkem na bázi modulu Real Time Clock (RTC)**

Autor: Ksenia Bolshakova (xbolsh00)

Brno 2018

## Zadání

V jazyce C vytvořte projekt s implementací vestavné aplikace "digitální hodiny s budíkem" pomocí modulu Real Time Clock (RTC) dostupného na čipu Kinetis K60 z desky platformy FITkit 3. Aplikace musí, dle požadavků uživatele, umožnit

- 1) nastavení času pro hodiny a budík,
- 2) zapnutí/vypnutí funkce buzení,
- 3) zvukovou a světelnou signalizaci při buzení, přičemž uživatel si bude moci vybrat jednu z alespoň tří vestavěných zvukových melodií a jeden z alespoň tří vestavěných způsobů světelné signalizace,
- 4) opakování pokusu o buzení s nastavitelným počtem pokusů a časovým odstupem mezi pokusy.

## Spuštění.

Pro spuštění projektu je potřeba ARM-FITkit3, KDS a Putty. V Putty je třeba nastavit Speed na 115200.

Po spuštění projektu má nejprve uživatel nastavit datum a čas ve formátu "DD-MM-YYYY HH:MM:SS", v případě chyby tento krok zopakuje. Dalším krokem je výběr melodie.

Aby se dalo poslechnout všechny melodie musí se stisknout „M[1-3]”, pro výběr melodie musí se stisknout [1-3].

Pak se musí vybrat světelná signalizace, pro možnost výběru musí se stisknout „L[1-3]”, pro výběr stiskne [1-3].

Dále se musí vybrat počet opakování od 0 do 3, v případě chyby se daný krok zopakuje.

Po výběru počtu opakování následuje výběr doby mezi opakováními v rozmezí od 10 do 100.

Dále se nastavuje datum a čas budíku ve formátu "DD-MM-YYYY HH:MM:SS".

Poslední krok je nastavení budíku – lze si vybrat mezi možnostmi [new, off, reboot, poweroff], kde **new** je zakládání nového budíku, **off** je vypnutí budíku, **reboot** je restart inicializace, **poweroff** je vypnutí systému.

## Implementace.

Pro implementaci byly využité příklady z cvičení a democvičení, dokumentace ke KDS a schéma FITkitu.

V souboru main.c v hlavní funkci main() je nejprve inicializace MCU, portů, UART5, RTC.

Dále je naimplementován stavový automat pro budík.

Ve stavu 0 se implementuje nastavování data a času, konvertuje se čas ze stringu do integeru pomocí funkce ConvertStrToTime() a knihovny time.h, zapíná se RTC. Funkce

ConvertStrToTime() parsuje string a kontroluje datum: rok, měsíc a den. Následuje kontrola počtu dní v měsíci a zda je rok přestupný a kontrola času.

Ve stavu 1 je využita funkce MusicChoose(), která kontroluje co uživatel zadal na vstup, a pokud si chce uživatel poslechnout všechny možnosti, zavolá se funkce Music(), která využívá funkce Beep() a Delay() v různých variacích pro různé melodie. Pokud uživatel vybere nějakou melodii, nastaví se do globální proměnné *chosen\_song*.

Ve stavu 2 se nastavuje světelná signalizace ve funkci LightChoose(), pro možnost výběru dané signalizace používá se funkce Light(), která nastavuje různé ledky a čekání. Pro nastavení signalizace se využívá globální proměnná *chosen\_light*.

Ve stavu 3 se nastavuje počet opakování ve funkci RepChoose(), kde se kontroluje počet opakování.

Ve stavu 4 se nastavuje delay ve funkci DelayChoose(), kde se kontroluje počet čekání na další opakování světové a zvukové signalizace.

Ve stavu 5 se nastavuje budík, parsuje se čas a datum ve funkci AlarmInit() pomocí funkce ConvertStrToTime().

Stav 6 je aktivování. Pomocí funkce ConvertTimeToStr() se konvertuje čas v RTC na datum a čas. Comand **new** vypne budík a vrátí konečný automat zas do stavu 1. Comand **off** vypne budík, nastaví RTC na 0. Command **reboot** nastaví RTC na 0, a vrátí se ke stavu 0.

Command **poweroff** vypne systém, přejde do stavu 7, kde bude čekat ve while smyčce.

**Literatura:**

<https://www.nxp.com/docs/en/user-guide/KDSUG.pdf>

přednášky IMP