Dokumentace k projektu do předmětu IMP

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

Maxim Plička(xplick04)

14. prosince 2022

Obsah

1 Úvod **2**

2 Implementace **2**
2.1 Inicializace modulů 2
2.2 Hlavní smyčka programu 2
2.3 Stavový automat 3

3 Ovládání aplikace **3**
3.1 Nastavení budíku 3
3.1.1 Nastavení světel a zvuku při buzení 3
3.1.2 Nastavení počtu a odkladu opakovaní buzení 3
3.1.3 Nastavení aktuálního času a času budíku 4
3.2 Budík 4

4 Závěr **4**

1 Úvod

Zadáním tohoto projektu bylo vytvořit vestavěnou aplikaci digitální hodiny s budíkem v jazyce C prostředky modulu Real Time Clock (RTC) dostupného na čipu Kinetis K60 z desky platformy FITkit 3. Projekt byl napsán v programu Kinetis Design Studio na systému Windows 10. Celý tento budík funguje ve 24-hodinovém formátu a nepodporuje zadávání dat.
Tato aplikace musí dále, dle požadavků uživatele umožňit:

- nastavení času pro hodiny a budík
- zapnutí/vypnutí funkce buzení
- zvukovou a světelnou signalizaci při buzení, přičemž si uživatel může vybrat alespoň jednu ze tří vestavěných zvukových melodií a alespoň jeden ze tří způsobů světelné signalizace
- opakování pokusu o buzení s nastavitelným počtem pokusů a časovým odstupem mezi pokusy

2 Implementace

Celý program je implementován v souboru `main.c`. Ve vrchní části programu se nachází definice a deklarace potřebných proměnných a makra. Dále se zde nachází implementaci podpůrných funkcí pro běh programu. Tyto funkce jsou z větší části ispirovány funkcemi ze cvičení, připadně demonstračních cvičení dostupných v E-LEARNINGu. Nakonec je zde implmenetována funkce `main()`, ve které je hlavní tok programu. Během implementace bylo využito schéma kontroleru ([3]) a také referenčního manuálu ([1]).

2.1 Inicializace modulů

Na začátku funkce `main()` se zavolají funkce `MCUInit()`, `pinInit()`, `UARTInit()` a `RTCInit()`. V těchto funkcích probíhá příprava pro práci s mikrokontrolerem. Všechny tyto funkce až na `RTCInit()` jsou inspirovány funkcemi ze cvičení. Funkce `RTCInit()` byla vytvořena na základě slajdu z přednášky ([2], slajd 15) a s pomocí refrenčního manuálu ([1]).

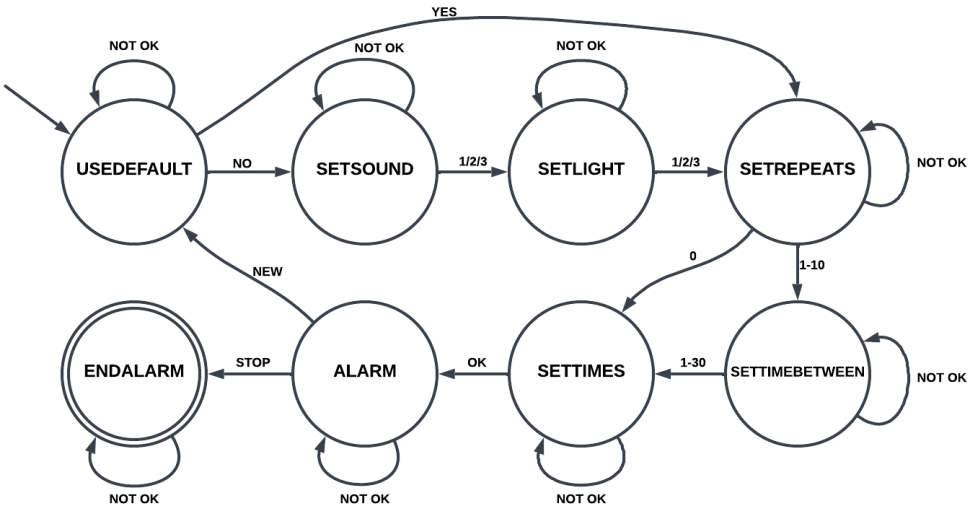
2.2 Hlavní smyčka programu

Hlavní smyčka programu je implemenotvána tak, aby z ní nešlo odejít. Dále se zde nachází jednotlivé stavy, mezi kterými se přechází. Při implemtaci nastavování jednotlivých dat od uživatele se vždy prvně zkontroluje formát vstupu a následně je tento vstup buď uložen do proměnné a pokračuje se dle stavového automatu (2.3), nebo je uživatel znovu dotázán na tento vstup.

Při nastavování časů do budíku je potřeba tyto časy převést na sekundy, jelikož RTC registry podporují čas buzení v sekundách. Nastavování času buzení je provedeno pomocí manipulace s registrem `RTC_TAR`. Jelikož je čítač času (`RTC_TSR`) od začátku běhu aktivní, nastavuji do registru `RTC_TAR` hodnotu čítače `RTC_TSR` + čas než by měl budík začít zvonit.

Jakmile se `RTC_TAR` rovná `RTC_TSR` do registru `RTC_SR` (Status registr) se na druhou pozici (Time alarm flag) nastaví logická 1. Toto díky mému nastavení RTC vyvolá přerušení, které má přednost před hlavní funkcí programu a je tedy ihned obslouženo ve funkci `RTC_IRQHandler()`. Následně se tok programu vrací do funkce `main()`, kde je na základě uživatele rozhodnuto, jestli se budík ukončí (zůstane ve stavu ENDALARM) a nebo začne od znovu.

2.3 Stavový automat



3 Ovládání aplikace

K přeložení a nahrání programu na mikrokontroler je zapotřebí Kinetis Design Studio. Celý běh aplikace se ovládá za pomocí terminálu PuTTY. V tomto terminálu je nutné nastavit si rychlost přenášení na 115200 baudů, data byty na 8, stop bity na 1, žádnou paritu, žádnou kontrolu toku a dále nastavit přislušný port na kterém se mikrokontroler nachází. Po nahrání programu na mikrokontroler se uživateli v terminálu objeví dotazy, na které může za pomocí klávesnice odpovídat. Předpokládaná odpověd je vymezena v hranatých závorkách za dotazem.

3.1 Nastavení budíku

Na začátku je uživatel dotázán, jestli chce využít základní nastavení pro světla a zvuk budíku. V případě, že uživatel zvolí základní variantu, je rovnou přesměrován na nastavení počtu opakování buzení. V opačném případě je mu zobrazeno nastavení světel.

3.1.1 Nastavení světel a zvuku při buzení

Uživateli jsou prvně přehrány varianty světel a může si z těchto variant vybrat. Následně jsou uživateli přehrány tóny zvuku při buzení, kde si může také vybrat.

3.1.2 Nastavení počtu a odkladu opakovaní buzení

Uživatel je dotázán na počet opakování. V případě, že zadá číslo 0, je přesměrován na nastavení času. V opačném připadě je dotázán na dobu v minutách, která značí interval odkadu buzení.

3.1.3 Nastavení aktuálního času a času budíku

Jako poslední věc, co uživatel nastavuje je aktuální čas, který se bude zobrazovat během buzení, a čas buzení. V případě, že je čas buzení menší nebo roven aktuálnímu času, je k tomuto času připočteno 24 hodin (budík se nastaví na následující den).

3.2 Budík

V této části se aktivně čeká na odpověd uživatele, kde se v případě zadání řetězce STOP ukončí celý budík. V případě zadání NEW se přeruší aktuální budík a uživatel musí budík nastavit znovu od začátku. Pokud uživatel nic nezadá, tak se budík po určité době spustí. Během přehrávání specifického buzení není možné budík přerušit. Toto lze udělat až poté, co buzení skončí.

4 Závěr

S řešením jsem začal včas, naneštěstí jsem podstatnou část z celkového času strávil u vyhledánáví určitých věcí v manuálu, které jsem nakonec nepoužil (např. FPGA LED, tlačítka). Z pohledu funkčnosti jsem v projektu zvládl splnit všechny body vymezené zadáním. Ohledně kvality výsledného řešení si myslím, že se snadně ovládá, zdrojový kód je přehledný, okomentovaný a dobře dekomponován na podproblémy. Co se týče dokumentace, tak si myslím, že je zde kvalitně sepsán postup práce s výsledným řešením. Vytknul bych si nejspíše, že zde podrobně nepopisuji práci s ostatními moduly. Ovšem přišlo mi, že bude lepší podrobněji popsat pouze modul RTC, který se nevyskytoval na cvičeních a musel jsem si o informace něm zjišťovat sám. Celkově bych si za projekt udělil 10/14. Body bych si strhnul za dokumentaci, kde si nejsem jistý, zda je zde vše dostatečně pospáno.

Reference

[1] freescale: K60 Sub-Family Reference Manual. 2012. Dostupné z: https://moodle.vut.cz/pluginfile.php/497392/mod_label/intro/k60p144m100sf2v2rm.pdf

[2] Růžička, R.: Čítače a časovače v mikrokontrolérech. [online], 2022. Dostupné z: https://moodle.vut.cz/pluginfile.php/500873/mod_folder/content/0/2022IMP08.pdf?forcedownload=1

[3] Šimek, V.: Minerva.PrjPcb. 2013. Dostupné z: https://moodle.vut.cz/pluginfile.php/497392/mod_label/intro/FITkit3_schema.pdf