

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Mikroprocesorové a vestavěné systémy  
Hodiny s budíkem na bázi modulu (RTC)

# 1 Úvod

Cílem projektu bylo implementovat digitální hodiny s budíkem s využitím modelu RTC na platformě FitKit3.

## 1.1 Motivace

Jak už bylo popsáno, cílem je implementovat digitální hodiny s budíkem. K tomuto problému by se dalo přistupovat několika způsoby.

- Řešení pomocí elektrických(číslicové)systémů. Toto řešení by vypadalo jako připojení několika číslicových součástek k desce(čítač, dekodér atd.), realizace tohoto řešení by bylo dost velké, příliš nákladové na realizaci a odvedená práce by za výsledek nestála.
- Dalším řešením je naprogramovat budík software, toto řešení má hodně výhod: jednoduchá a rychlá implementace. Je však problém, na kterém zařízení tento program poběží.
- Třetí přístup a je ten který jsem v projektu použil, vestavěný systém (en. embedded system), který kombinuje software a hardware.

Pro dosažení cíle je vhodnější využít třetí přístup který má další výhody:

1. Jednoduchost vývoje, jelikož samotný program je napsán softwarově .
2. Hardware na kterém je tento software spuštěn.
3. Hardwarové komponenty které jsou použité (RTC atd.) a velkou výhodou je že dá se přidat další různé komponenty (obrazovka pro zobrazení času, atd.).

## 1.2 Použité nástroje

Pro řešení projektu bylo použito zařízení FitKit3, jako operační systém byl použit Windows 11, vývojové prostředí Kinetis Design Studio 3.0.0 IDE pomocí kterého je možné překládat zdrojový kód, nahrávat do FitKit3 a opravovat program. Pro komunikace se zařízením přes UART byly využity Putty.

## 1.3 Dekompozice zadání

Před začátkem řešení projektu byla provedena dekompozice problému:

- Konfigurace zařízení.
- Nastavení času.
- Konfigurace budíku.
  - Výběr zvukové signalizace.
  - Výběr světelné signalizace.
  - Nastavení opakování.
- Nastavení času budíku.
- Realizace vypnutí a zapnutí budíku.

## 1.4 Hardware

K realizaci projektu byly použity následující hardwarové komponenty: RTC, UART, diody, tlačítka a bzučák.

### 1.4.1 Obecná konfigurace MCU

Na začátku bylo provedeno nastavení hodinového pod systému(`reg. MCG_C4`), dal dělič byl nastaven na nulu(`reg. SIM_CLKDIV1`), vypnut watchdog(`reg. WDOG_STCTRLH`), pak zapnuty hodiny pro UART, RTC, port B, port E, port A. Pro zapínání hodin byly využity masky a registry `SIM_SCGC5`, `SIM_SCGC1`, `SIM_SCGC6`.

### 1.4.2 UART

Byl využit UART5. Pro konfigurace byly využity registry `UART5_C1`, `UART5_BDH`, `UART5_BDL`, `UART5_C4`, `UART5_C3`, `UART5_MA1`, `UART5_MA2`, `UART5_S2` a byla nastavena přenosová rychlost 115200Bd, přenos 8 bitu bez parity.

Za běhu byly využity registry: `UART5_S1` flagy `TDRE`, `TC` jsou využity pro sledování, zda je datový kanál zaneprázdněn, flag `RDRF` je využit pro sledování je-li datový registr naplněn. Registr `UART5_D` pro čtení a zápis dat.

### 1.4.3 RTC

Pro konfigurace byly využity registry: `RTC_CR` flag `SWR` pro resetování všech RTC registrů, `OSCE` pro zapnutí oscilátoru.

Za běhu byly využity registry: `RTC_SR` bit `TCE` zapnutí a vypnutí čítače sekund. Registr `RTC_TSR` pro nastavení a čtení času. Registr `RTC_TAR` pro nastavení a čtení času budíku. Registr `RTC_IER` flag `TAIE` pro zapnutí, vypnutí a přerušení budíku.

### 1.4.4 Porty

Pro UART5, diody, tlačítka a bzučák byly nastaveny porty. Pro nastavení byl použit registr `PORTx_PCRn` kde `x` je název portu a `n` je pin na který je připojen komponent.

U tlačítek byly nastaveny přerušení. Bzučák a diody za běhu programu se aktivují a deaktivují pomocí registru `GPIOx_PDOR` pro tento registr byla využita maska `PTx_PDOR` kde `x` je název portu.

## 1.5 Software

Pro účely projektu byla použita knihovna pro komunikaci s zařízením `MK60D10.h`. Pro práce s časem je použita knihovna `time` která umožní práce s Unix formátem času.

## 2 Implementace

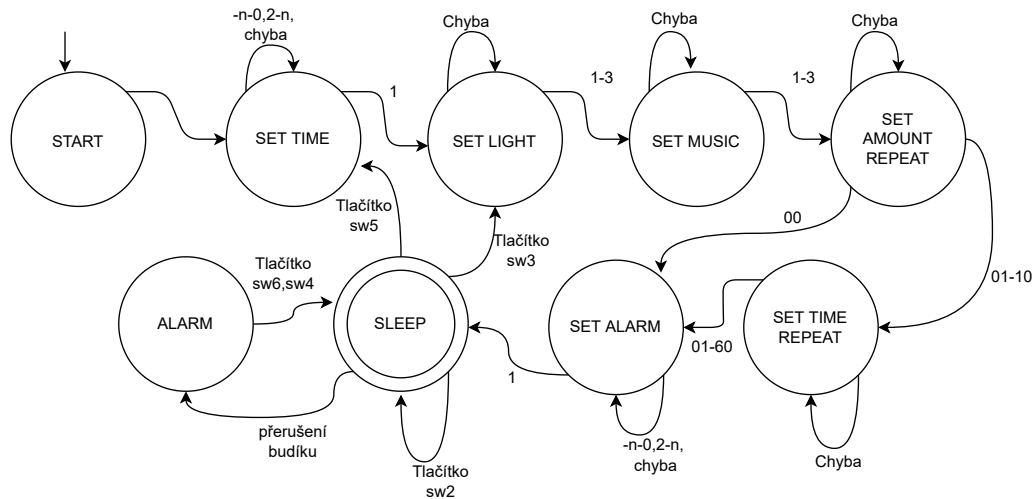
Na začátku podle dekompozice která je uvedená v sekci 1.3 byl navržen Konečný automat obrázek č:1. Pak byla provedena inicializace komponentů, při inicializaci kód byl inspirován příklady z laboratoře a příklady programů na `FitKit3` které byly zaznamenány na přednášce. Dál bylo implementováno řešení.

### 2.1 Rozdělení

Program byl rozdělen do jednotlivých souborů pro přehlednost. Každý soubor kromě `main.c` reprezentuje nějakou pomocnou část.

- `init` – reprezentuje inicializaci komponentu a MCU.
- `addons` – reprezentuje funkce blikání diod atd.
- `times` – reprezentuje funkci práce s časem.

- uart – reprezentuje funkci práce s UART.
- main – hlavní soubor



Obrázek 1: Konečný automat

## 2.2 Hlavní smyčka

V hlavní smyčce programu je implementován konečný automat.

Po zapnutí zařízení je potřeba nastavit čas. Pak jsou dvě možnosti: přejít do režimu spánku nebo nastavit budík. Režim spánku je nekonečná smyčka ze které můžete vystoupit pomocí přerušení (stisknutím tlačítka). Při nastavování budíku bude uživatel muset nastavit: zvukovou a světelnou signalizaci, a poté zvolit, kolikrát lze budík opakovat. Pak nastavit čas probuzení. Budík poté přejde do režimu spánku.

## 2.3 Práce s časem

Pro práci s časem bylo rozhodnuto použít unixový čas, který ukazuje počet sekund od roku 1970. Toto řešení má problémy, které jsou popsány v sekci 3.2.

### 2.3.1 Nastavení času

Uživatel zadá čas ve formátu „dd-mm-YYYY HH:MM“. Pomocí funkce `strptime` bude tento čas převeden do struktury `tm` a bude zkontrolována platnost dat, poté bude převeden na sekundy pomocí funkce `mktime` a zapsán do registru `RTC_TSR`.

### 2.3.2 Nastavení času buzení

Nastavení času budíku je podobné obecnému nastavení času. Uživatel zadá čas ve formátu „HH:MM“. Pak jsou dvě možnosti: buď bude budík nastaven na dnešek, pokud čas neuplynul, nebo na zítra. Porovnání času probíhá pomocí převodu času na zařízení do struktury `tm` a jeho porovnáním s nastaveným časem budíku.

Dále zkopírujeme aktuální čas na zařízení, a odečítáme tolik sekund, kolik je potřeba aby byl začátek dne(00:00) a pak se přidává počet sekund zbývajících do buzení, a nastavíme registr `RTC_TAR`.

## 2.4 Buzení

Pokud je budík zapnutý (je zapnuto přerušení budíku) když se čas ve dvou registrech `RTC_TAR` a `RTC_SRT` shoduje, bude vyvoláno přerušení. Kde flag přerušení bude vypnut a nastaven stav konečného automatu který reprezentuje buzení. Buzení probíhá zvukovou a světelnou signalizací, dokud uživatel nevypne budík nebo nezapne opakování. Stav buzení lze změnit jenom přerušením.

## 2.5 Zvukovou a světelnou signalizaci

Zvukovou a světelnou signalizaci jsou implementovány v souboru `addons`. Poté, co uživatel vybere signalizaci, bude jeho volba uložena v globální proměnné. V režimu buzení budou signalizace vybírány pomocí `switch`.

## 2.6 Přerušení

Přerušení může být zapnuté buď použitím tlačítek které jsou nastaveny nebo zapnutím buzení, je to popsáno v sekci 2.3.2.

### 2.6.1 Tlačítko SW2

Při stisknutí tlačítka `sw2` na obrazovku bude vypsán aktuální čas zařízení a jestli budík je nastaven bude vypsán i čas buzení. Z registru `RTS_TSR` bude zkopírován aktuální čas v sekundách a bude převáděn pomocí funkce `localtime_r` do struktury `tm` a pak už pomocí funkce `strftime` převeden do formátu řetězce.

### 2.6.2 Tlačítko SW6

Stisknutím tlačítka `sw6` lze odstranit všechny budíky nebo vypnout signalizaci budíku při buzení nebo když budík není nastaven, nastavit ho znovu na čas který už byl zadán uživatelem, nastavení je popsáno v sekci 2.3.2.

### 2.6.3 Tlačítko SW4

Při stisknutí tlačítka vypne se signalizace buzení a když je nastaven opakovaně nastaví se budík na nový čas který se počítá jako čas vypnutí budíku + čas za kolik minut budík musí znova zazvonit.

Při stisknutí tlačítka `sw4` přerušení bude vyvoláno ale pokud budík není ve stavu buzení nic se nestane.

### 2.6.4 Tlačítko SW5

Vyvolá se přerušení, hodiny a budík budou resetovány a vrátí se do začátečního stavu, konečný automat bude převeden do stavu nastavení času. Toto přerušení nebude obslouženo když budík ve stavu buzení.

### 2.6.5 Tlačítko SW3

Při stisknutí tlačítka `sw3`, automat přepne se do stavu nastavení budíku a umožní uživatelům nastavit budík na nový čas.

## 3 Shrnutí

### 3.1 Testování

Testování proběhlo manuálně. Bylo provedeno několik testů. Postup provádění testování: na začátku je nutné nastavit aktuální čas, dále nastavit signalizace a opakování, pak nastavit budík.

Při různých testech nastavení signalizace a opakování bylo zvolené různě. Na začátku bylo testováno funguje-li

buzení, dále bylo testováno opakování a vypínání/zapínání budíku, poslední etapou testování je testování různé světové a zvukové signalizace. Z testování nahoře jde říct že projekt je úplně funkční.

### **3.2 Problém unixového času**

Řešení s využitím unixového času má problém v tom že RTC\_TSR registr je 32 bitový a když čas se počítá z 1970 roku tak v roce 2038 bude přetečení. Které způsobí nefunkčnost budíku. Ale v současné době neovlivní chování zařízení.