

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



## IMP - Mikroprocesorové a vestavěné systémy 2019/2020

### ARM-FITkit3: Kodér Morseovy abecedy

## **Obsah**

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Návod na spustenie</b>	<b>1</b>
<b>3</b>	<b>HW zapojenie použitých komponent</b>	<b>1</b>
<b>4</b>	<b>Popis ovládania</b>	<b>2</b>
<b>5</b>	<b>Implementácia</b>	<b>2</b>
<b>6</b>	<b>Záver</b>	<b>3</b>

# 1 Úvod

Táto technická správa vznikla ako dokumentácia k projektu do predmetu Mikroprocesorové a vestavěné systémy. Mojou hlavnou úlohou bolo implementovať vstavanú aplikáciu v jazyku C pre *Kinetis K60*, ktorá by mala kódovať zadané znaky do Morseovej abecedy.

Aplikácia by mala podporovať:

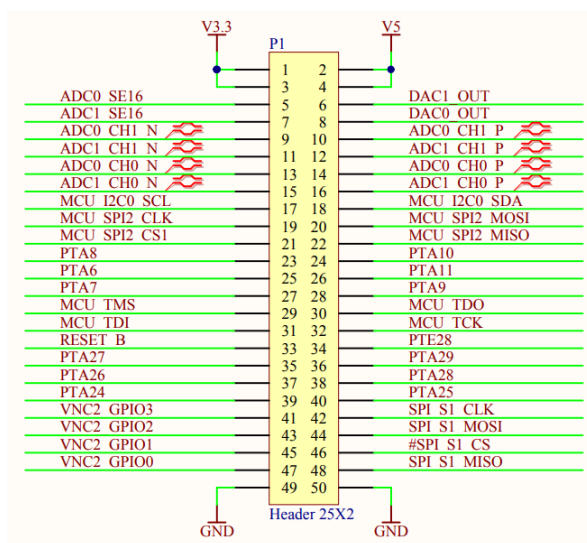
- Zadávanie znakov by malo prebiehať interaktívne prostredníctvom klávesnice pripojenej k vývojovej doske FITkit3.
- Výstup je realizovaný pomocou piezo bzučiaku na FITkit3, napr. znak 'A' (.-) bude zakódovaný jedným krátkym a jedným dlhým pípnutím, znak 'D' (-..) atď... [2]
- Aplikácia by mala umožniť zadávanie ďalšieho znaku a to aj v prípade že je stále bzučiacom reprodukováný predchádzajúci znak.
- Klávesami \* a # by mali byť implementované rôzne rýchlosti reprodukcie Morseovej abecedy
- Ako vstupnými symbolmi uvažujeme nad písmenami českej abecedy (bez háčikov a čiarok)

## 2 Návod na spustenie

Na spustenie kódéra Morseovej abecedy na zariadení FITkit3 sú potrebné nasledujúce softvérové produkty: Operačný systém *Windows 10*, aplikácia a *Kinetis Design Studio*. Pomocou aplikácie *KDS* je možné preložiť zdrojový kód a následne ho nahráť do zariadenia FITkit3. Taktiež je potrebné pripojiť externú klávesnicu k zariadeniu FITkit3. Presné zapojenie je popísané v sekcii 3.

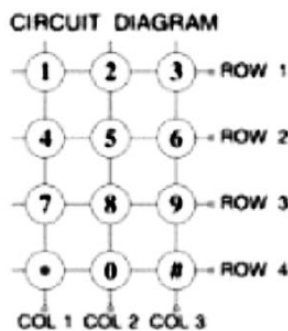
## 3 HW zapojenie použitých komponent

Ako som spomínal v sekcii 2, je potrebné k zariadeniu FITkit3 pripojiť externú klávesnicu. Klávesnicu je možné pripojiť pomocou siedmich GPIO pinov, ktoré sa nachádzajú na zariadení FITkit3. Konkrétne sa jedná o rozširujúce konektory P1.



Obr. 1: Rozširujúce konektory P1, zdroj: [3, strana 18]

Na zapojenie klávesnice som si zvolil piny PTA6–PTA11 a PTA27. Schéma klávesnice je nasledovná:



AK-207, 304, 507, 707, 804

OUTPUT ARRANGEMENT		
OUTPUT	PIN NO.	SYMBOL
1	COL 2	
2	ROW 1	
3	COL 1	
4	ROW 4	
5	COL 3	
6	ROW 3	
7	ROW 2	



Obr. 2: Schéma externej klávesnice, zdroj: [1, strana 5]

## 4 Popis ovládania

Po následnom pripojení klávesnice a nahratia programu do FITkit3 ovládanie aplikácie pozostáva len zo stlačenia vybranej klávesy, bohužiaľ som nestihol implementovať časovač *LPTMR*, čo malo za následok len čiastočnú funkčnosť aplikácie. Z mojej implementácie je možné vypípať len prvé písmeno na danom tlačidle. Napr. na tlačidlo 2 je možné vypípať len A, na tlačidlo 3 len písmeno D atď...

## 5 Implementácia

Aplikácia je implementovaná pomocou jazyka C v programe *KDS*. Vstupným bodom programu je funkcia `main()`, ktorá volá pomocné funkcie `MCUInit()` a `PortsInit()`. Funkcia `MCUInit()` je prevzatá z demo aplikácie `FITkit3-demo`, ktorá nám bola poskytnutá. Funkcia má na starosti inicializáciu MCU, nastavenie základného clocku a vypnutie watchdogu. Funkciu `PortsInit()` som tiež pôvodnej prevzal z dema ale bolo potrebné ju výrazne modifikovať. Rows (riadky) klávesnice mám nastavené ako output a cols (stĺpce)

```
void PortsInit(void)
{
    SIM->SCGC5 = SIM_SCGC5_PORTA_MASK; //Zapnutie clock portu
    PORTA->PCR[4] = PORT_PCR_MUX(0x01); //Reproduktor

    //Nastavenie korespondujúcich pinov
    int pulldown = (PORT_PCR_MUX(0x01) | PORT_PCR_PE(0x01) | PORT_PCR_PS(0x00));
    PORTA->PCR[6] = pulldown; //PTA6 --> 25 --> COL1 --> zeleny --> 0x40
    PORTA->PCR[7] = pulldown; //PTA7 --> 27 --> COL3 --> fialovy --> 0x80
    PORTA->PCR[8] = pulldown; //PTA8 --> 23 --> COL2 --> oranžovy --> 0x100
    PORTA->PCR[9] = PORT_PCR_MUX(0x01); //PTA9 --> 28 --> ROW3 --> sivy --> 0x200
    PORTA->PCR[10] = PORT_PCR_MUX(0x01); //PTA10 --> 22 --> ROW1 --> zlty --> 0x400
    PORTA->PCR[11] = PORT_PCR_MUX(0x01); //PTA11 --> 26 --> ROW4 --> modry --> 0x800
    PORTA->PCR[27] = PORT_PCR_MUX(0x01); //PTA27 --> 35 --> ROW2 --> biely --> 0x8000000
    //Nastavenie korespondujúcich PTA pinov ako output (vystup)
    PORTA->PDDR = GPIO_PDDR_PDD(SPK) | ROW[0] | ROW[1] | ROW[2] | ROW[3];
}
```

Obr. 3: Inicializácia GPIO pinov

ako input s pulldown. Ďalej vo funkcii `main()` mám nekonečný cyklus, v ktorom sa nachádzajú dva hlavné for cykly:

```
for (int rowIndex = 0; rowIndex < 4; rowIndex++)
```

```
a for (int columnIndex = 0; columnIndex < 3; columnIndex++). Cykly postupne inte-
```

rujú cez riadky a stĺpce a kontrolujú či nedošlo k stlačeniu tlačidla. V prípade stlačenia tlačidla 2 sa zavolajú funkcie `beepDash()` a `beepDot()`, ktoré slúžia na vypípanie Morseovho kódu. Medzi týmito funkciami sa volá funkcia `delay()`, ktorá zabezpečí krátku pauzu medzi vypípaním časťami písmena.

## 6 Záver

Aplikácia mi funguje len čiastočne, k implementácii časovača *LPTMR*, ktorý by mi pomohol dosiahnuť plnú funkčnosť nedošlo z časových dôvodov.

## Použitá literatúra

- [1] NEVORAL, J.: Pár slov k projektům... [online], rev. 26. októbra 2018. Dostupné z: [http://www.stud.fit.vutbr.cz/~xnevor02/Demo\\_k\\_projektum.pdf](http://www.stud.fit.vutbr.cz/~xnevor02/Demo_k_projektum.pdf)
- [2] WIKIPEDIE: Morseova abeceda. [online], rev. 22. novembra 2019. Dostupné z: [https://cs.wikipedia.org/wiki/Morseova\\_abeceda](https://cs.wikipedia.org/wiki/Morseova_abeceda)
- [3] ŠIMEK, V.: Schéma obvodového zapojenia výukového kitu Minerva. [online], rev. 2013. Dostupné z: <https://wis.fit.vutbr.cz/FIT/st/cfs.php.cs?file=%2Fcourse%2FIMP-IT%2Fexcs%2FFITkit3-schema.pdf&cid=13324>