

# Vysoké učení technické v Brně

## Fakulta informačních technologií



IMP - Mikroprocesorové a vestavěné systémy  
ARM-FITkit3: Jednoduchý elektronický klavír

## O projektu

Cílem tohoto projektu bylo navrhnout a implementovat systém pro přehrávání melodií v rozsahu alespoň jedné oktávy (jednoduchý klavír). Tento projekt je součástí studia na VUT FIT v Brně, předmět IMP 2019/2020. Projekt je realizován pro FITkit 3, s tím, že mapování jednotlivých kláves na tóny je vyřešeno přes zadávání písmen skrze terminál putty. Samotný program je implementován v jazyce C, v programovacím prostředí Kinetis Design Studio.

## Vlastnosti, omezení a vstup

Tento projekt nepotřebuje žádné externí vybavení protože je ovládán z příkazové řádky přes Putty (USB A → USB B). Jsou implementovány tóny v rámci 2 oktáv, přesněji C4 - C6. Projekt disponuje ukázkovou skladbou, ke které bylo využito více tónů než je dostupných v uvedeném rozmezí, slouží tedy pro prezentační účely funkcionality. Jedná se o úryvek ze skladby Megalovania od Toby Fox ze hry Undertale. Uživatel může po spuštění zadávat znaky **q - p** & **a - g**, kde každý znak náleží tónu C4 - C6 (q = C4, w = D4, ..., g = C6), název tónu se mu vypíše do konzole, s tím, že tyto tóny mají předem stanovenou délku, se kterou program pracuje. Při ukázkové skladbě je délka tónů jiná, pro zajištění realističtější melodie. Ukázkovou skladbu lze zapnout stisknutím mezerníku, všechny ostatní znaky program zachytí jako nový řádek.

## Implementace

Aplikace je rozdělena do dvou částí: inicializace a nekonečná smyčka čekající na input od uživatele. První část se skládá z několika inicializací jednotlivých “komponent”. Nejdříve se inicializuje samotné **MCU**, což je v podstatě základní jednotka celého FITkitu 3. Dále je na řadě inicializace portů/pinů. Zde se nachystají všechny prerekvizity pro samotné používání komponent potřebných v tomto projektu (základ UART5, FTM, PWM, Buzzer..). Podle schématu FITkit 3 byl buzzer přiřazen na 0x03.

UART5 je nutné inicializovat tak, aby šlo zadávat znaky z klávesnice přes Putty, zde se velice vyplatilo první cvičení předmětu IMP. Program ještě musí inicializovat **FTM**, které je použito v projektu pro samotné generování zvuku v kombinaci s **PWM**. Je přesněji použito FTM0, s tím, že využívá základní děličku 8.

Po provedení těchto přípravných funkcí se program dostane do stavu “čekání”, což je implementováno skrze `while(1)`, ve kterém program zachytává uživatelem zadané znaky a převádí je na zvuk. Jednotlivé znaky jsou mapovány na zjištěné frekvence s pomocí `switch`, ve kterém se volá funkce `playTone(hz, length)`, kde `hz` značí frekvenci tónu, který chceme, aby byl vygenerován, a `length` je doba, po kterou tón bude slyšen.

Pro přehrání tónu jsou k dispozici proměnné `FTM0_MOD` a `FTM0_C1V`, kde `FTM0_MOD` značí modulo signálu a `FTM0_C1V` střidu. Délka přehrávání je implementována skrze funkci `delay`.

Lze tudíž použít následující výpočet pro přiřazení:

```
FTM0_MOD = (50Mhz / 8) / Freq  
FTM0_C1V = FTM0_MOD / 2
```

`Freq` značí frekvenci tónu, který chceme přehrát. Tento výpočet se provede pro každý znak od uživatele (pokud spadá do **q - p** a **a - g**), zajistí vygenerování samotného tónu pod uvedenou frekvencí. Po vypršení funkce `delay` se nastaví střida na 0 a program opět čeká na další znak.

## Shrnutí a nedostatky

Tento projekt funguje jako jednoduchý elektrický klavír s ukázkovou skladbou.

V rámci projektu bylo zadáno v rámci možností na poskytnutém reproduktoru zajistit “odezvu” jednotlivých tónů, ovšem u téhle funkcionality jsem nepřišel na to, jakým způsobem modifikovat signál, aby odezva zněla realisticky. I přesto, že lze upravovat střidu, tak výsledek se ani z daleka nepodobá realistickému odeznění. Tuto část jsem zkoušel implementovat přes cyklus `for`, ve kterém jsem si rozdělil průběh tónu na části, a v každé části jsem zkoušel střidu zmenšovat, ovšem bez žádného kladného výsledku.