VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Mikroprocesorové a vestavěné systémy

Dokumentace k projektu

xhorky32 2021/2022

Obsah

1 Základní informace

Zvolil jsem si variantu projektu `B - LIBOVOLNÝ KIT: Dekodér Morseovy abecedy`. Tuto variantu vede M. Bidlo, zpracovat ji lze na libovolném kitu (FITKit, ESP32,... - bez OS). Jako doporučený programovací jazyk je jazyk C.

1.1 Zadání

Vytvořte systém umožňující dekódování symbolů abecedy, zadaných v podobě Morseova kódu prostřednictvím jednoho zvoleného tlačítka kitu, do podoby znaků latinky. Dekódovaný text vypisujte po zadání každého znaku postupně na terminál nebo na displej kitu (pokud jím je vybaven). Vhodně přizpůsobte ruční zadávání Morseovky uživatelem (délka obou značek, mezery mezi značkami i jednotlivými symboly abecedy). Při zadávání využívejte indikaci bzučákem kitu (pokud jím je vybaven), případně alespoň pomocí LED, pro snazší sledování délky zadávaných značek. (viz [3])

1.2 Hardware

Pro vypracování projektu jsem zvolil přípravek FITKit (verze 2.0). Ten je pomocí USB A-B kabelu připojen k mému počítači. Nic dalšího není pro práci s FITKitem v rámci tohoto projektu nutné, tedy nebyly použity žádné jiné externí periferie. FITKit není nijak upraven ani zvláštně nastaven. Zapojeny jsou piny tak jak byly zapojeny při převzetí z knihovny VUT FIT. Tj.: `J18, J8, J9, J6, J3` (uprostřed ne), `J5, J17, J11, J12` a `GND TD0` v `JP11`.

1.3 Software

Pro vývoj, překlad a spuštění mého projektu jsem použil program QDevKit. Ten je nainstalován na virtuálním stroji ve VirtualBoxu (viz [1]). Konkrétně používám virtuální stroj pro předměty INC, IVH, INP (plně dostačující i pro projekt do IMP), nejnovější verze (tedy 202103). Jedná se o rychlou a jednoduchou možnost jak ihned začít pracovat na projektu bez dalších komplikací, proto jsem zvolil tuto cestu. (a ne například přímou práci s kitem na hostovi-mém OS)

1.4 Hotové řešení

Dokončený projekt má pomocí zvoleného (jednoho) tlačítka dekódovat symboly Morseovy abecedy a vypisovat písmena na displej kitu. Toto tlačítko rozlišuje mezi dlouhým a krátkým stiskem, čím se pokryjí všechny (2) symboly Morseovy abecedy. Při implementaci tohoto řešení jsem však narazil na několik problémů, které bylo třeba vyřešit, taktéž mě napadly další možné funkce, které jsem implementoval. Vše bude popsáno níže v této dokumentaci. (symboly morseovy abecedy jsou uvedeny například ve [4])

2 Implementace

Projekt je implementován v jazyce C a VHDL. Pracuji s komponenty FPGA i MCU, primárně však s MCU.

2.1 FPGA

Projekt by nebyl funkční, pokud by nebyl implementován top level (FPGA) ve VHDL. Ten zajišťuje, že FITKit reaguje na stisk tlačítek a lze díky němu pracovat s displejem. Tato část je však plně převzatá z již implementovaných dem aplikací ve virtuálním stroji. Konkrétně byla převzata z `Demo aplikace: Demo klávesnice a LCD`. Vybral jsem konkrétně toto demo, jelikož v mém projektu pracuji s klávesnící i LCD současně. (viz [2])

2.2 MCU

2.2.1 Použité knihovny a soubory

Jedná se již o můj tvořený kód v jazyce C. Zde jsem opět využil již hotových řešení - použil jsem knihovnu `fitkitlib.h`. Dále pracuji s klávesnicí i displejem, pracuji tedy s funkcemi z `lcd/display.h` a `keyboard/keyboard.h`. Vše je opět již implementované ve FitkitSVN, resp. na virtuálním stroji, který používám. Ze standardních knihoven používám zejména knihovnu pro práci s řetězci.

2.2.2 Vlastní kód

Celá implementace je soustředěna v jednom souboru a to je `main.c`. Ten je dělen na části definic, globání proměnné a část funkcí. V první části se nachází potřebné definice konstant, booleovských proměnných, globálních proměnných a konstantní proměnná `morseCodes`, která obsahuje řetězce - posloupnosti symbolů morseovky seřazené od A do Z a poté od 0 do 9. Následuje část s funkcemi, samotná implementace projektu. Zmínil bych zde funkci `handle_keys`, která se stará o obsluhu tlačítek a přepínání do různých stavů či zobrazování textu na displeji.

2.2.3 Způsob řešení projektu

Nejprve bylo nutné vypnout watchdog a inicializovat hardware. Poté program vstoupí do hlavní nekonečné smyčky, kde se obsluhují tlačítka a terminál. Dále se zde rozeznává krátký a dlouhý stisk nuly, kontroluje se počet znaků displeje a případně se resetuje, pokud je displej zaplněn. V případě stisku nuly se dle délky stisku do jednoduchého pole přidá znak tečky či pomlčky, dle toho, zda jde o krátký či dlouhý signál.

Pokud uživatel zadávání posloupnosti symbolů skončí, stiskne hvězdičku, čímž se obsah tohoto pole bude ve funkci `convertMorseToChar` porovnávat s předdefinovaným seznamem řetězců `morseCodes`. Pokud se uživatelova posloupnost shoduje s jedním z řetězců, porovnávání skončí, zaznamená se index, na kterém srovnávání bylo ukončeno. Poté dle indexu funkce jednoduše vrátí požadovaný znak (hodnota z ASCII tabulky).

Zvlášními případy jsou, pokud není zadána žádná posloupnost - funkce vrátí znak mezery, nebo pokud srovnání skončí a nebyla nalezena shoda - funkce vrací znak otazníku. Díky této funkci je známý výsledný znak a už je možné jej vypsat na displej kitu. Před výpisem se znak uloží do pole výsledného textu. Samotný výpis textu na displej probíhá tak, že se do pole současného textu vloží část (32 znaků) pole výsledného textu.

Pokud vás implementace-řešení projektu zajímá podrobněji, doporučuji se podívat přímo do zdrojového kódu.

2.3 Řešené problémy při implementaci

Zadání přesně nespecifikuje všechny situace, detaily máme možnost vyřešit dle vlastního uvážení.

2.3.1 Zadávání symbolů morseovky

Pro zadávání symbolů Morseovy abecedy jsem zvolil tlačítko `'0'`. To dle délky stisku rozeznává, zda zadáváte symbol `.` (značí krátký signál) nebo − (značí dlouhý signál). Bylo však nutné vyřešit, jakým způsobem FITKit pozná, že uživatel ukončil zadávání sekvence symbolů a má se na displej vypsat dané písmeno nebo číslice. V případě, že vaše zadaná posloupnost není z Morseovy abecedy, na displej se vypíše otazník. Pokud naopak nezadáte nic, považuje se to za mezeru.

2.3.2 Tlačítko *

Funguje jako takový "Enter". Po zadání sekvence symbolů v Morseově abecedě stisknete tlačítko hvězdičky. Tím FITKit pozná, že jste ukončili zadávání znaků pro jedno písmeno a může být vypsáno na displej kitu a uživatel může pokračovat v zadávání dalších znaků (resp. písmen). Zvolené řešení mi přijde nejjednodušší a nejintuitivnější.

Napadlo mě však, že by FITKit mohl automaticky rozpoznat, že je ukončeno zadávání symbolů pomocí pauzy (doba, kdy by uživatel nic nezadával). Jenže toto řešení by celý proces překladu morseovky do latinky zpomalilo. Navíc by bylo nutné přidat možnost nastavení délky pauzy, jelikož pomalejší uživatel by vyžadoval, aby požadovaná pauza byla například delší. Toto řešení jsem tedy zavrhl pro zbytečnou složitost a neefektivitu.

Při mém řešení si uživatel symboly morseovky může zadávat jak rychle chce, jakmile ví, že zadal svůj požadovaný znak, stiskne hvězdičku a zadává dál. Dále toto tlačítko při zadávání znaků morseovy abecedy funguje jako mezerník, v případě, že nezadáte žádnou sekvenci symbolů.

2.3.3 Ideální délka dlouhého stisku

Tlačítko `'0'` se používá pro zadávání symbolů morseovky a rozlišuje mezi dlouhým a krátkým stiskem. Jak dlouho mám mít stisklé dané tlačítko, aby se to považovalo za dlouhý stisk? Tento problém jsem nechal na uživateli. Standardně je nastaven určitý interval, kdy rozeznávám mezi krátkým a dlouhým stiskem. Jedná se o jakési optimální nastavení, které považuji dle sebe za ideální. Avšak pro někoho jiného to ideální být nemusí. Proto jsem uvedl k funkčnosti další tlačítko.

2.3.4 Tlačítko

Přepne FITKit do tzv. módu Nastavení. Vypíše vám, jak je současně nastaven dlouhý stisk. Poté pomocí dalších kláves (A, B, C, D) můžete nastavit délku stisku dle libosti (původní, krátký, střední, dlouhý). Po nastavení-přepnutí délky stisku se vždy objeví obrazovka s nápisem Test, kde si uživatel může zkoušet, zda mu dané nastavení vyhovuje (vypisuje se . nebo - na displej kitu dle toho, zda bylo tlačítko nuly stisknuto dlouze či krátce). Pokud je nastavení dokončeno, stačí tlačítko mřížky stisknout znovu a FITKit je připraven na zadávání symbolů Morseovy abecedy.

2.3.5 Ostatní tlačítka klávesnice

Má FITKit reagovat na stisk ostatních tlačítek? Co se má v tom případě dít? Toto jednoduché "dilema"jsem vyřešil tak, že při stisku klávesy, která nemá nadefinované chování se zkrátka nic nestane.

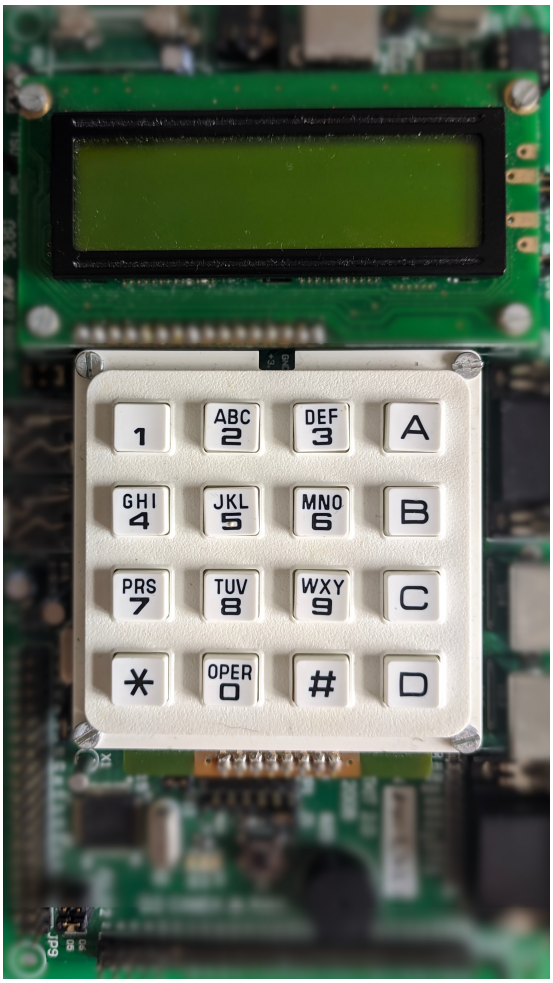
2.3.6 Výsledná zpráva

Jelikož výsledkem mojí aplikace je zpráva, resp. dekódované symboly morseovky v podobě písmen latinky, nastalo zde hned několik obtíží. První byla limitace počtu znaků na displeji (FITKit 2.0 má dvouřádkový displej s nejvíce 32 znaky). Jelikož mi přišlo nevhodné, aby výsledná zpráva byla takto omezena, rozhodl jsem se vytvořit buffer pro aktuálně zobrazovanou zprávu na displeji a zároveň buffer pro výslednou zprávu. Buffer pro výslednou zprávu je ovšem také limitován, a to hardwarem FITKitu. Jelikož jsem v implementaci zvolil variantu statických polí, buffer může být pouze tak velký, kolik je možno alokovat paměti v zásobníku kitu. Proto může buffer výsledné zprávy mít nejvýše 800 znaků. Je to však výrazné zlepšení oproti původní variantě s 32 znaky.

Dále bylo potřeba zajistit, aby si v případě dlouhé zprávy mohl uživatel zprávu celou přečist, proto jsem implementoval tzv. view mód, který umožní prohlížení výsledné zprávy a ”listování”v celém bufferu pomocí dvou tlačítek. (viz 3)

3 Popis ovládání

Na obrázku je zvýrazněno vše, s čím na FITKitu budete pracovat. Displej, klávesnice a diody D5 a D6.



Obrázek 1: FITKit 2.0

3.1 Displej

Na displeji se po nahrání aplikace zobrazí text: ”Zadejte kod v Morseově abecedě”. Poté uživatel může začít zadávat vstup nebo si nastavit délku stisku tlačítka či prohlížet výsledný text.

3.2 Klávesnice

Pracuje se s klávesami nuly, hvězdičky a mřížky. Klávesa nuly rozeznává krátký a dlouhý stisk a pomocí ní zadáváte sekvenci symbolů (krátký a dlouhý signál). Pokud je sekvence symbolů úspěšně zadaná, stisknete hvězdičku a FITKit přeloží tuto sekvenci dle Morseovy abecedy do latinky a dané písmeno zobrazí na displej. Poté pokračujete dále v zadávání. Klávesa mřížky slouží jako vstup do nastavení. Uživatel si zde nastaví požadovanou délku dlouhého stisku a tu případně otestuje. Dále je implementována funkčnost kláves A, B, C, D.

3.2.1 Klávesy pro zadávání symbolů a dekódování

Hlavní klávesy pro používání programu jsou `*` a `0`. Jejich funkčnost je podrobněji již popsána v kapitolách 2.3.2, 2.3.1.

3.2.2 Nastavení

Klávesa `#` přepne kit do módu nastavení. Zde se uživateli zobrazí aktuální nastavení délky stisku tlačítka pro zadávání symbolů morseovky. Pokud si toto nastavení přeje změnit, klikne na klávesy A - D. Jednotlivé klávesy nastaví délku stisku na jednu z variant: optimální, krátký, střední, dlouhý stisk. Poté následuje obrazovka Test:, kde si uživatel délku stisku ověří. Poté může opustit nastavení znovustisknutím mřížky nebo nastavení opět změnit. Tato klávesa je také popsána již dříve v kapitole 2.3.4.

3.2.3 View mód

Po stisknutí klávesy `D` se program přepne do tohoto módu. Zde je možné si prohlédnout celý výsledný dekódovaný text. Pomocí tlačítek `A` (dopředu) a `B` (zpátky) se uživatel v textu pohybuje. Pokud je uživatel na konci textu a stiskne tlačítkem, že chce jít dopředu, objeví se opět na začátku textu. Analogicky to tak funguje i pro opačný případ. V tomto módu nelze měnit nastavení ani jakkoliv měnit text. Mód uživatel opustí stiskem stejné klávesy, jako do něj vstoupil.

3.2.4 Clear a Backspace

Chyba se stát může. A proto jsem implementoval možnost stiskem tlačítka `C` smazat buffer pro symboly morseovy abecedy. Pokud je buffer prázdný a přesto se toto tlačítko zmáčkne, program to považuje za pokus o smazání celého výstupního textu. Uživateli se proto na displeji zobrazí hláška, zda chce zprávu smazat. Dle instrukcí na displeji potvrdí či odmítne (tlačítky hvězdičky či mřížky). V případě potvrzení se celá výstupní zpráva smaže. Dále je implementována funkčnost tlačítka `B`, které v případě stisku smaže jeden znak ze zprávy, funguje tedy jako backspace.

3.3 Diody

Dioda D5 indikuje stisknutou klávesu 0. D6 slouží zase jako indikátor neukončené posloupnosti. Rozsvítí se (červeně) pouze tehdy, pokud uživatel zadává více jak pět symbolů z morseovy abecedy. Morseova abeceda totiž sestává z posloupnosti symbolů, která není delší než 5 těchto symbolů. Pokud se tedy uživatel pokusí zadat šestý (a další) symbol, dioda se rozsvítí a upozorní tak uživatele na to, že je potřeba buď to smazat současnou posloupnost a začít znovu, nebo ji odeslat a přeložit do znaku latinky.

3.4 Pracovní postup po nahrání aplikace

Pokud máte moji aplikaci nahranou na FITKitu, stačí otevřít terminál FITKitu. Poté se zobrazí na displeji kitu text vyzývající k zadání vstupu. Uživatel poté začne zadávat symboly pomocí tlačítka nuly. Pokud jsou zadány korektně všechny symboly, stačí stisknout hvězdičku a znak latinky se objeví na displeji. Poté se postup jen opakuje. Pokud si chcete nastavit délku stisku, je možné kdykoliv přejít do režimu nastavení. Taktéž je možné smazat současnou sekvenci symbolů morseovky a začít znovu nebo smazat rovnou celý výsledný text a začít uplně odznovu. Zpráva se automaticky ukládá a pokud se nevejde na displej, můžete si ji poté celou prohlédnout pomocí view módu. Při práci doporučuji mít otevřený termínal v QDevKit, jelikož vám bude vypisovat v jakém jste módu či vámi zadané symboly morseovy abecedy.

4 Krátké dokumentační video

V tomto videu předvádím, jak se pracuje s FITKitem, pokud je v něm nahraný můj projekt. Shlédnout ho můžete zde: odkaz na Google Drive.

4.1 Komentář k videu

Video je delší než jedna minuta, jelikož zde ukazuji i další funkce, které nejsou v zadání tohoto projektu. Kvůli funkci, kterou nazývám view-mode jsem si musel navíc předchystat již výsledný text před začátkem videa. Proto zde uvádím důležité části videa (z čehož je patrné, že funkčnost požadovaná ze zadání se dá jednoduše odprezentovat za méně jak půl minuty):

POČÁTEK	KONEC	POPIS ČÁSTI
0:00	0:07	mazání jednoho znaku
0:08	0:20	view-mode
0:21	0:25	mazání celého textu (clear)
0:25	**0:49**	**základní funkčnost, zadávání symbolů morseovky a jejich překlad do latinky**
0:50	1:10	nastavení délky stisku, testování
1:11	1:23	mazání jednoho znaku (backspace)
1:24	1:33	zadání neplatné sekvence (nepatří do morseovy abecedy) - výpis otazníku
1:34	1:42	view-mode při prázdném výstupu

5 Závěr

Projekt není úplně dokonalý. Jednak jsem limitován hardwarem samotného FITKitu, kdy více jak 800 znaků nelze uložit do pole kvůli paměťové náročnosti (zásobník). Nicméně je to stále mnohem více, než pouze to, co se vleze na displej (32 znaků). Dále jsem zpozoroval, že kurzor se občas objeví, občas ne. Není to sice nijak podstatné pro chod aplikace, ale občas to může být matoucí, že se jednou objeví a jednou ne. Myslím si, že i přes nedokonalosti projekt splňuje to, co je od něj očekáváno a i něco navíc, což dle mého názoru přispěje k větší přívětivosti aplikace pro běžného uživatele.

Použitá literatura a zdroje

[1] FITKit - Download.
URL https://merlin.fit.vutbr.cz/FITkit/private/web/index.php?pg=download

[2] FITKit SVN - použitý top level v FPGA (navigujte do: trunk/apps/demo/keyboard_lcd).
URL https://merlin.fit.vutbr.cz/FITkit/docs/svn/svn_content.html?ref=&lng=cs

[3] IS FIT, zadání projektu IMP.
URL https://wis.fit.vutbr.cz/FIT/st/course-sl.php.cs?id=785554&item=86399&nc=1

[4] Wikipedia - Morse code.
URL https://en.wikipedia.org/wiki/Morse_code