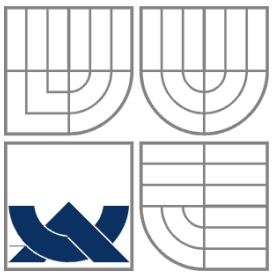
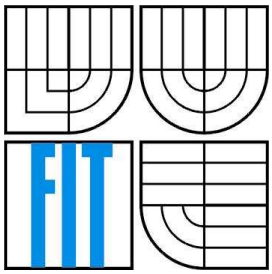


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

FITKIT JAKO ŘÍDÍCÍ JEDNOTKA KROKOVÉHO MOTORU

FITKIT AS STEP MOTOR CONTROLLER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ŠTĚPÁN ČEJKA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JOSEF STRNADEL, Ph.D.

BRNO 2011

Zadání bakalářské práce/11640/2010/xcejka02

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů Akademický rok 2010/2011

Zadání bakalářské práce

Řešitel: **Čejka Štěpán**
Obor: Informační technologie
Téma: **FITkit jako řídicí jednotka krokového motoru**
FITkit as Step Motor Controller
Kategorie: Vestavěné systémy

Pokyny:

1. Seznamte se s principy činnosti a řízení krokových motorů. Nastudujte parametry zadaného typu krokového motoru.
2. Seznamte se s FITkitem a diskutujte možnosti řízení krokového motoru pomocí MCU a FPGA dostupných na FITkitu.
3. Pro FITkit navrhněte jednotku pro řízení krokového motoru z bodu 2. Jednotka bude pracovat ve 2 režimech - A a B. V režimu A bude uživatel moci zadávat rychlost a směr otáčení motoru a volit typ řízení. V režimu B bude možno zadat úhel, o který se má pootočit hřídel motoru. Pro ovládání jednotky a komunikaci s uživatelem využijte periferie dostupné na FITkitu a terminálové okno.
4. Jednotku navrženou v bodě 3 implementujte ve variantě pro MCU a ve variantě pro FPGA. Implementaci doplňte o dokumentaci ve formátu umožňujícím její bezproblémové umístění na stránkách FITkitu.
5. Funkčnost jednotky prakticky ověřte.

Literatura:

- Dle pokynů vedoucího.

Při obhajobě semestrální části projektu je požadováno:

- Splnění bodů 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Strnadel Josef, Ing., Ph.D.**, UPSY FIT VUT
Datum zadání: 1. listopadu 2010
Datum odevzdání: 18. května 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
612 66 Brno, Božetěchova 2

L.S.

doc. Ing. Zdeněk Kotásek, CSc.
vedoucí ústavu

Abstrakt

Tato práce se zabývá problematikou krokových motorů. Čtenář získá základní přehled o typech, principech funkce a některých metodách řízení těchto motorů. Praktická část této práce spočívala v navržení a implementaci dvou řídících jednotek pro mikrokontrolér MSP430F2617 a hradlové pole FPGA obsažené na platformě FITkit. Řídící jednotky jsou navrženy tak, aby bylo možné zvolit mezi dvěma způsoby řízení a různými provozními režimy použitého motoru. Vlastnosti řídících jednotek byly ověřeny na krokovém motoru SMR 300-100-RI/24 s využitím zkonstruovaného rozdělovače impulsů.

Abstract

This work deals with problems related to design and implementation of a stepper motor controller based on the MSP430F2617 microcontroller and the Spartan3 field-programmable gate array available on the FITkit platform. In the work, details related to types, principles, functions and stepper-motor control methods are described in detail. Proposed control units are designed to work in two different ways of control and operating modes of motor. Features of the control units have been tested with the SMR 300-100-RI/24 step motor by the means of a custom-designed pulse generator.

Klíčová slova

Krokový motor, FITkit, řízení, mikrokontrolér, FPGA,VHDL

Keywords

Stepper motor, FITkit, control, microcontroller, FPGA,VHDL

Citace

Čejka Štěpán: FITkit jako řídící jednotka krokového motoru, bakalářská práce, Brno, FIT VUT v Brně, 2011

FITkit jako řídící jednotka krokového motoru

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Josefa Strnadela, Ph.D a uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Štěpán Čejka
12. května 2011

Poděkování

Rád bych poděkoval Ing. Josefu Strnadelovi, Ph.D za cenné rady, které mi poskytl při řešení praktické části, ale i při psaní této práce.

© Štěpán Čejka, 2011
Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Úvod

Současný stav techniky pohonů v oblasti výpočetních, řídících a regulačních zařízení je velmi často založen na aplikaci krokových motorů. Hlavní důvod proč krokové motory dnes tak hojně nacházejí uplatnění ve zmíněných oblastech je ten, že u těchto motorů lze přesně řídit jak jejich otáčky, tak i konkrétní polohu rotoru motoru. Nejčastěji se s tímto druhem motorů můžeme setkat u různých druhů tiskáren a zapisovacích zařízení. Dále pak ve strojírenství lze tyto motory mnohdy nalézt v kombinaci s převodovkou jako pohonnou jednotku u jemných a přesných polohovacích zařízení. Například v obráběcích CNC strojích slouží tyto pohony k nastavování polohy jak obráběného materiálu, tak i pojezdů jednotlivých os obráběcího nástroje. Velmi široké možnosti pro nasazení krokových motorů se otevřely rozvojem robotiky.

Jak již bylo zmíněno výše, princip krokového motoru umožňuje přesné řízení úhlové rychlosti, směru a velikosti úhlu, o který se má rotor motoru pootočit. K tomu řízení se nejčastěji používá vhodně navržený vestavěný systém. O návrhu takového systému pojednává tato práce. Existuje několik způsobů řízení krokových motorů. Druhy řízení a potřebné teoretické informace o krokových motorech jsou popsány v kapitole 2.

Funkci řídící jednotky čtyřfázového krokového motoru SMR 300-100-RI/24, který byl pro účely této práce zvolen, obstarává platforma FITkit, která byla navržena na Fakultě informačních technologií VUT v Brně. Vývojový kit generuje na svém výstupním rozhraní elektrické impulsy o nízké napěťové úrovni. Tyto impulsy pak vhodně ovládají tranzistorový rozdělovač impulsů, který vybudí příslušné vinutí krokového motoru. Bližší informace o rozdělovači impulsů a konkrétním typu krokového motoru jsou uvedeny v kapitole 3, FITkit je popsán v kapitole 4.

Implementace a funkcionalita řídící jednotky pro mikrokontrolér v jazyce C je uvedena v kapitole 5. V kapitole 6 pak následuje popis řídící jednotky pro programovatelné hradlové pole FPGA.

2 Krokový motor

2.1 Definice krokového motoru

Krokový motor je zařízení, které mění elektrickou energii na točivý pohyb. Tento pohyb ovšem není spojitý, ale probíhá po jednotlivých krocích, proto se tento druh motoru nazývá krokový. Krokový motor má 2 hlavní části, kterými jsou rotor a stator. Rotor se skládá z hřídele a několika permanentních magnetů. Stator se skládá z elektromagnetických cívek, které mají za úkol přitáhnout rotor - pootočit s ním. Toto „přitáhnutí“ způsobí elektrický proud, který při procházení cívkou statoru vytváří magnetické pole. Tímto způsobem dochází k otáčení rotoru. Důležitou vlastností a předností krokového motoru je, že lze přesně definovat a realizovat úhel, o který se rotor má otočit. Pokud bude krokovému motoru dodáváno napětí, je schopen rovněž stát na dané pozici a udržet ji i při působení vnějšího kroutícího momentu na rotor. Tyto vlastnosti krokového motoru lze v praktických aplikacích také velmi dobře využít [6].

2.2 Princip činnosti krokového motoru

Základní pohyb, který krokový motor vyvine, je otočení rotoru o jeden krok, kdy rotor opustí jednu ustálenou magnetickou polohu a přejde do polohy sousední. Této změny polohy se dosáhne přivedením řídícího impulsu na příslušné vinutí statoru. Po přivedení tohoto impulsu rotor nuceně zaujme takovou pozici, při které je výsledný magnetický odpor minimální. Pokud je motor nezatížen, tak se přesně sesouhlasí ozubení, kterým jsou opatřeny pólové nástavce statoru a rotoru. Pro vykonání jedné celé otáčky musí rotor vykonat definovaný počet kroků jako reakci na sled řídících impulsů přiváděných s určitým kmitočtem, který udává úhlovou rychlost otáčení rotoru. Počet těchto kroků záleží na typu a provedení krokového motoru.

2.3 Základní parametry krokových motorů

Krokové motory obecně disponují řadou různých statických a dynamických parametrů, které je potřeba znát při návrhu polohovacích zařízení pro bezchybný chod motoru. Nejvýznamnější parametry a pojmy jsou uvedeny níže. Informace zde publikované pocházejí z [3] a [6].

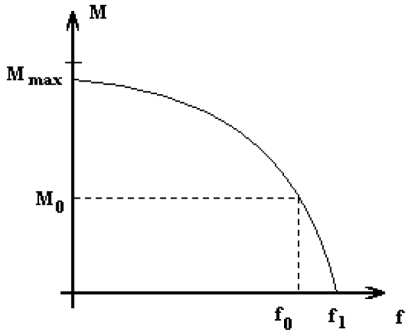
- **krok**- je pohyb rotoru z jedné výchozí pozice do nejbližší sousední pozice, který je vyvolán působením magnetického pole. Pootočení motoru o jeden krok je odezvou na příchod řídícího impulsu.
- **velikost kroku**- je úhel, o který se pootočí rotor při provedení jednoho kroku. Tento úhel je dán počtem kroků, který motor vykoná během jedné celé otáčky, tj. konstrukcí motoru.
- **tolerance úhlu kroku**- je maximální odchylka úhlu vůči jednomu kroku nezatíženého motoru. Tato odchylka může nabývat kladných i záporných hodnot.
- **otáčky rotoru**- jsou určeny počtem kroků rotoru za vteřinu. Tento počet je závislý na kmitočtu kroků f_k. Počet otáček rotoru je dán následujícím vztahem:

$$n = \frac{60 * f_k * \alpha}{360}$$

kde:

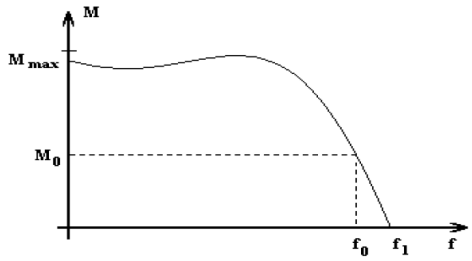
- n- počet otáček za minutu
- f_k- kmitočet kroků, udává se v Hz
- α- velikost kroku, udává se ve stupních

- **řídící kmitočet**- je kmitočet, kterým je krokový motor řízen. Pokud nedojde k chybě, je roven kmitočtu kroku.
- **maximální rozběhový kmitočet**- je kmitočet, při kterém se musí nezatížený motor rozběhnout i zastavit bez ztráty kroku a to i v případě, že se tento kmitočet skokově změní z nulové hodnoty na tento maximální kmitočet.
- **maximální provozní kmitočet**- je maximální kmitočet, kterým lze přivádět řídící impulsy do motoru, aniž by došlo k chybě v pootočení rotoru.
- **přídržný moment**- je největší kroutící moment, kterým může být zatížen rotor vybuzeného motoru, který se nepohybuje (je sepnuto jedno vinutí), aniž by došlo k vychýlení od požadované pozice.
- **vlastní přídržný moment**- je největší možný kroutící moment, kterým může být zatížen rotor nevybuzeného motoru (rotor je ve své poloze udržován působením permanentních magnetů), při kterém nedojde ke změně polohy rotoru.
- **statický úhel zátěže**- je úhel, o který se rotor vychýlí z magnetické klidové polohy při působení zátěže o určité velikosti na rotor vybuzeného motoru.
- **rozběhový moment**- je maximální moment, kterým lze motor zatížit při rozběhu s daným řídícím kmitočtem, aniž by nastala chyba v krokování.



Obrázek 2.1: Rozběhový moment [6]

- **momentová charakteristika**- vyjadřuje závislost momentu působícího na rotor na řídící kmitočet a uplatňuje se v oblasti tzv. řízeného zrychlování krokového motoru. V této oblasti je pro bezchybný chod motoru nutné rychlost zvyšovat plynule, s každým krokem o určitou hodnotu. Skoková změna rychlosti by vedla ke ztrátám kroku nebo dokonce k zastavení a rezonanci motoru.



Obrázek 2.2: Momentová charakteristika [6]

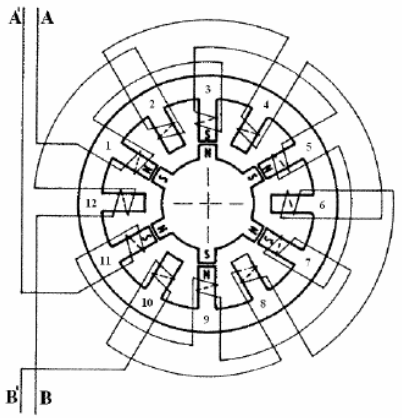
2.4 Klasifikace krokových motorů

Krokové motory se dělí do tří základních skupin. Kritériem rozdělení je jejich konstrukční provedení. První skupinou jsou krokové motory s aktivním rotorem, do druhé patří krokové motory s pasivním rotorem a hybridní krokové motory tvoří třetí skupinu. Informace čerpány z [13] a [14].

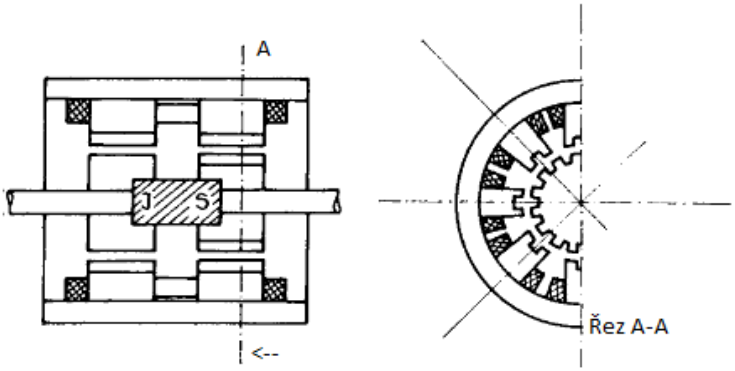
- **Krokové motory s aktivním rotorem** jsou konstruovány tak, že rotor motoru je opatřen sledem permanentních magnetů, podélně uložených po obvodu rotoru. V případě komutátorových motorů je rotor tvořen vlastním vynutím motoru a kotvou.

Pokud je pro konstrukci rotoru využito permanentních magnetů, lze tyto motory dělit dále podle směru polarizace magnetů na motory s radiálně polarizovaným rotorem (obrázek 2.3), nebo motory s axiálně polarizovaným rotorem (obrázek 2.4). U obou druhů motorů musí být počet pólových nástavců, které jsou tvořeny magnety, vždy sudý. Pohledem na řez motoru na obrázku 2.3 lze spatřit uspořádání permanentních magnetů, kdy se vždy střídá jejich polarita. Aby se zabránilo tomu, že budou vedle sebe dva magnety se stejnou polaritou, musí být počet těchto magnetů vždy sudý. Znázorněný motor má vinutí zapojena dvoufázově. Proud protékající těmito vinutími vytváří

elektromagnetické pole takové polarity, která je dána směrem toku elektrického proudu a způsobem navinutí cívky na odpovídající pólový nástavec.



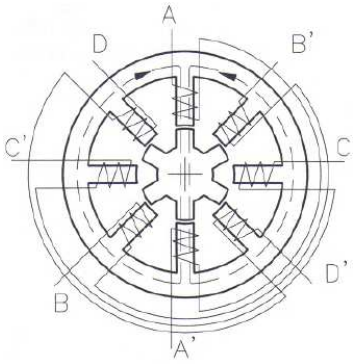
Obrázek 2.3: Dvoufázový krokový motor s aktivním, radiálně polarizovaným rotorem [13]



Obrázek 2.4: Konstrukce krokového motoru s axiálně polarizovaným rotorem [13]

- **Krokové motory s pasivním rotorem** (relační, reluktanční) jsou charakteristické tím, že rotor je tvořen vhodným kovovým profilem nebo několika plechy nalisovaných na hřídeli rotoru. Rotor je vyroben z magneticky měkkého (feromagnetického) materiálu a při pohledu na průřez motoru na obrázku 2.5 lze vidět, že rotor má po svém obvodu výrazně vyčnívající pólové nástavce, které mají stejnou šířku, ale je jich jiný počet ve srovnání s pólovými nástavci statoru. Vinutí u těch typů motorů jsou zapojena tak, že dvě protilehlé cívky jsou zapojeny vždy v sérii. K činnosti těchto motorů se využívá velký rozdíl mezi magnetickou vodivostí v podélné a příčné ose.

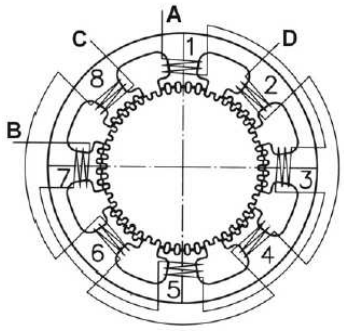
Pro oba doposud zmíněné typy krokového motoru je podstatné, že počet pólových nástavců statoru je vždy rozdílný od počtu nástavců rotoru. Z této skutečnosti pak plyne rozdílná rozteč těchto nástavců na obou součástech motoru a nemůže tedy nastat situace, že by pólové nástavce rotoru i statoru vůči sobě zaujaly přesně protilehlou souběžnou pozici, a tím bylo znemožněno otáčení motoru.



Obrázek č. 2.5: Řez krokovým motorem s pasivním rotorem [13]

- **Hybridní krokové motory** jsou v současné době nejrozšířenějším typem krokových motorů. Z obou výše uvedených typů motorů kombinuje tento druh určité vlastnosti, které umožňují dosáhnout velmi jemného krokování rotoru.

Stator těchto motorů se skládá z pólových nadstavců, na kterých je navinuta cívka. Rotor je tvořen hřídelí z oceli s vystouplými pólovými nástavci. Mezi těmito nástavci jsou umístěny permanentní magnety. Na pólových nástavcích statoru jsou vytvořeny malé drážky. Takto vzniklé ozubení umožňuje lepší magnetický tok ve vzduchové mezeře mezi statorem a rotorem. Díky tomu pak tyto motory prokazují lepší statické a dynamické vlastnosti, než motory s aktivním či pasivním rotorem.



Obrázek 2.6: Řez hybridním krokovým motorem [14]

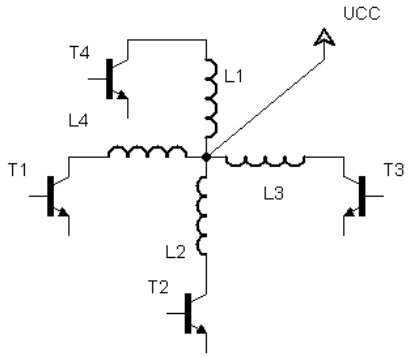
2.5 Způsoby řízení krokových motorů

V této kapitole jsou popsány základní typy řízení krokových motorů. Jednotlivé typy řízení lze mezi sebou kombinovat, čímž se např. dosáhne zvětšení přídržného momentu nebo zvýšení přesnosti polohování rotoru. Informace zde publikované jsou čerpány z [5] a [13].

- unipolární vs. bipolární řízení,
- jednofázové vs. dvoufázové řízení,
- řízení s poloviční krokem,
- mikrokrokování.

Unipolární řízení

Unipolární řízení je charakteristické tím, že proud prochází v jednom časovém okamžiku pouze jednou cívkou (v kombinaci s řízením s polovičním krokem může el. proud procházet v jednom okamžiku střídavě jednou a následně dvěma sousedícími cívkami). Tento druh řízení s sebou přináší výhody i nevýhody. Nevýhodnou je malý kroutící moment. Ten je vyvážen nízkým odběrem elektrického proudu v motoru a jednoduchou řídící elektronikou, kdy pro ovládání motoru postačuje jeden spínací prvek (většinou tranzistor) na každou cívku.

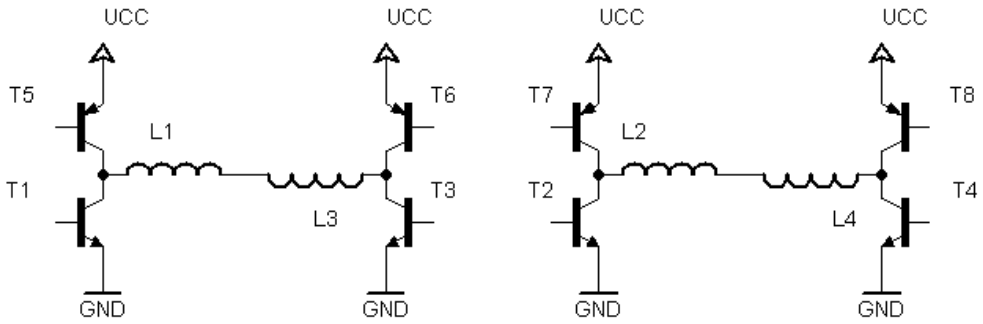


Obrázek 2.7: Unipolární řízení

Bipolární řízení

V případě bipolárního řízení prochází proud v jednom časovém okamžiku vždy dvěma protilehlými cívkami. Z těchto cívek má každá opačně orientované magnetické pole vůči druhé. Při srovnání s unipolárním způsobem řízení odebírá tento motor více elektrického proudu, ale je schopen vyvinout dvojnásobný kroutící moment.

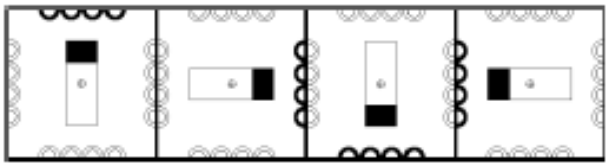
Tento způsob řízení je také náročnější na řídící elektroniku. Pro spínání jedné fáze jsou potřeba vždy dva spínací prvky pro navození obou směrů magnetického pole. Pokud roli spínacích prvků plní tranzistory, je nutné pro každou protilehlou dvojici cívek použít čtyři tranzistory, kdy dva jsou typu NPN a dva typu PNP. Toto zapojení se nazývá tzv. H-můstek a je znázorněno na obrázku 2.8.



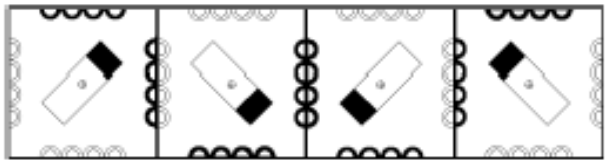
Obrázek 2.8: Bipolární řízení

Jednofázové vs. dvoufázové řízení

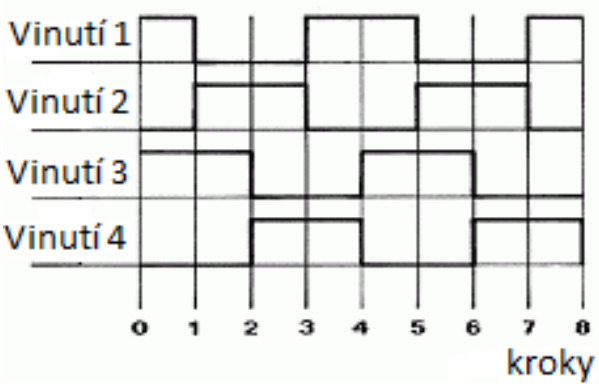
Při jednofázovém řízení je v jednom časovém okamžiku aktivní vždy pouze jedna cívka, u dvoufázového řízení jsou aktivní vždy cívky dvě. U obou způsobů řízení dochází k provedení stejného počtu kroků v rámci jedné otáčky, ale motor řízený dvoufázově dosahuje dvojnásobného kroutícího momentu, přičemž spotřebuje i dvojnásobně více elektrické energie ve srovnání s motorem řízeným jednofázově. Oba typy řízení jsou někdy označovány jako čtyřtaktní.



Obrázek 2.9: Unipolární jednofázové řízení [5]



Obrázek 2.10: Unipolární dvoufázové řízení [5]



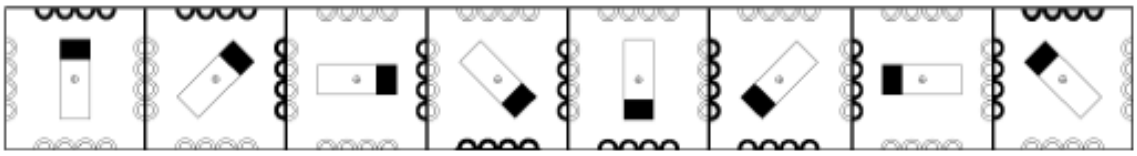
Obrázek 2.11: Časový průběh řídících impulsů přiváděných do krokového motoru při dvoufázovém řízení

Řízení s polovičním krokem

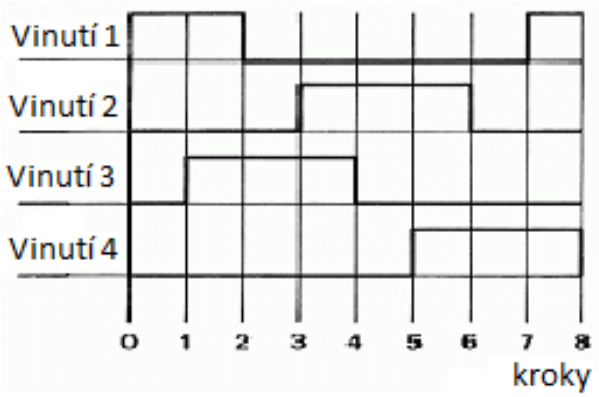
Řízení s polovičním krokem, nebo-li osmitaktní řízení, střídavě kombinuje jednofázové a dvoufázové řízení. Tímto způsobem řízením lze dosáhnout dvojnásobného počtu kroků v rámci jedné otáčky oproti jedno/dvoufázovému řízení, což znamená, že rotor motoru lze polohovat s dvojnásobnou přesností.

Při návrhu polohovacího zařízení poháněného osmitaktně řízeným krokovým motorem je nutné počítat s tím, že přídržný moment se liší vzhledem k počtu aktivních vinutí motoru. Je tedy nutné celé

zařízení dostatečně dimenzovat, aby nedošlo k samovolné změně polohy rotoru v době, kdy je motor zastaven a aktivní je pouze jedna fáze motoru.



Obrázek 2.12: Princip osmitaktně unipolárně řízeného krokového motoru [5]



Obrázek 2.13: Časový průběh řídících impulsů přiváděných do krokového motoru při osmitaktním řízení

Mikrokrokování

Mikrokrokování je poměrně nová metoda řízení krokového motoru. Vychází z metody dvoufázového řízení, kdy jsou v jednom okamžiku sepnuté vždy dvě sousední fáze motoru a obě fáze jsou napájeny stejně velkým proudem.

U této metody dochází k rozdělení jednoho kroku na několik menších stejně velkých kroků. Tohoto rozdělení lze dosáhnout tak, že každá z cívek bude napájena elektrickým proudem jiné velikosti. Tím dojde k vytvoření různě silných elektromagnetických polí kolem každé cívky, a tím lze dosáhnout téměř jakékoliv polohy rotoru.

Metoda mikrokrokování umožňuje dosáhnout velikosti kroku v řádech desetin až setin úhlových stupňů. Takové přesnosti polohování nelze dosáhnout jiným způsobem z důvodu mechanické realizace krokového motoru (opravdu zřídka je možné pracovat s motorem, jehož poloviční krok, dosažitelný osmitaktním řízením, je menší než 1°).

3 Periferní části řídící jednotky

3.1 Tranzistorový rozdělovač impulsů

Krokový motor nemůže být k FITkitu připojen napřímo, protože mikrokontrolér MSP430F2617 obsažený na platformě FITkit může poskytnout na jeden vývod maximální proud cca 40 mA a napětí 3.3 V. Krokový motor SMR 300-100-RI/24 vyžaduje napájecí napětí 24 V a jednou fází (vinutím) může odebírat proud až 250 mA. Na první pohled je tedy zřejmé, že napěťové a proudové úrovně si neodpovídají, FITkit není schopen dodat motoru dostatečný výkon a při přímém napojení motoru na FITkit by mohlo dojít ke zničení mikrokontroléru. Z tohoto důvodu bylo nutné mezi FITkit a motor vřadit výkonový spínací prvek, který je schopen po přivedení impulsů s malou napěťovou úrovní poskytnout danému vinutí odpovídající výkon pro správný chod motoru bez nebezpečí poškození mikrokontroléru a dále je schopen přepínat mezi jednotlivými fázemi s maximálním provozním kmitočtem použitého motoru.

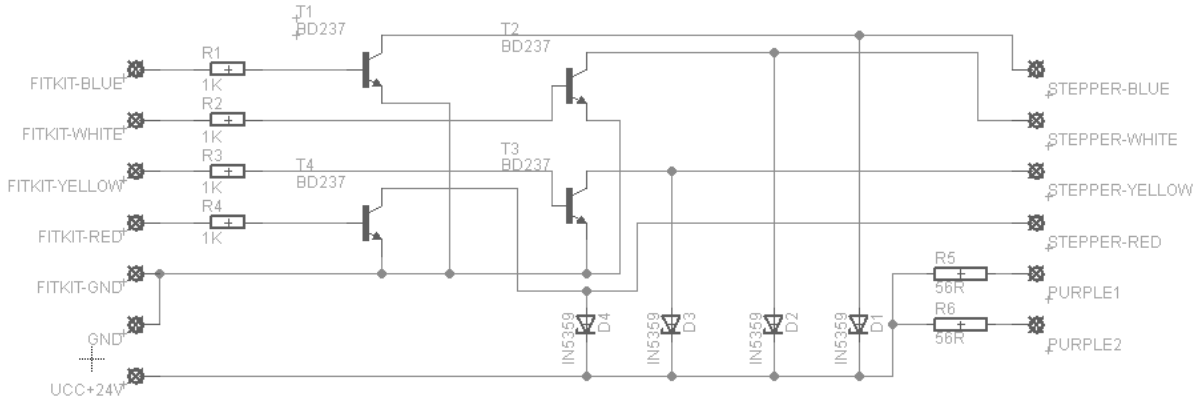
3.1.1 Konstrukce rozdělovače impulsů

Autorem práce navržený rozdělovač impulsů je založen na čtyřech výkonových unipolárních tranzistorech BD237 a slouží k unipolárnímu řízení krokového motoru. Každý tranzistor slouží ke spínání jedné cívky krokového motoru. Tyto tranzistory byly použity pro jejich schopnost propouštět přes přechod kolektor-emitor velký proud. Katalogový list tohoto tranzistoru uvádí hodnotu kolektorového proudu I_C 2 A, což je skoro desetkrát větší hodnota, než jakou potřebuje použitý motor, a napětí UCE 100 V.

3.1.2 Popis zapojení

Při pohledu na obrázek 3.3 reprezentující schéma vnitřního zapojení použitého motoru lze odvodit, že k ovládání dvou sousedních cívek slouží tři vodiče. Jeden vodič představuje společný střed cívek určený pro přivedení kladného napájecího napětí na obě cívky a zbylé dva vodiče slouží pro přepínání cívek na záporný pól napájecího zdroje. Na navrženém rozdělovači impulsů se tyto dva vodiče připojí ke kolektoru příslušného tranzistoru.

V zapojení figurují také čtyři Zenerovy diody, kdy každá dioda je připojena paralelně k jednomu vinutí motoru a slouží pro spínání indukční zátěže. Bližší popis významu těchto diod a princip činnosti samotného rozdělovače impulsů bude popsán v kapitole 3.1.3.



Obrázek 3.1: Schéma zapojení rozdělovače impulsů

3.1.3 Princip činnosti rozdělovače impulsů

Princip činnosti tohoto zapojení je vysvětlen pouze pro jednu fázi (vinutí) motoru, jelikož u ostatních fází je zapojení identické.

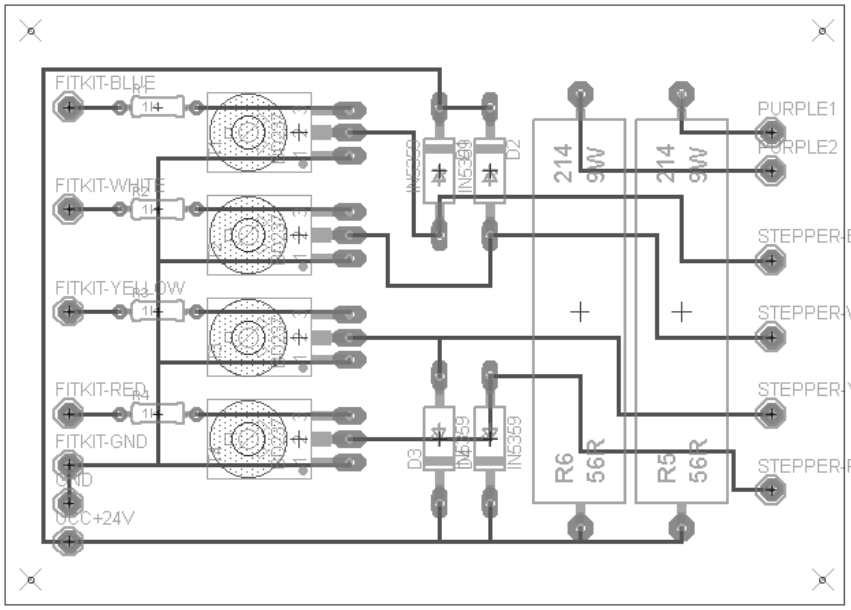
Po přivedení kladného napěťového impulsu z FITkitu na bázi tranzistoru T1 se na tomto tranzistoru otevře přechod kolektor-emitor po dobu trvání toho impulsu. Tím, že se otevře přechod kolektor-emitor, dojde vlastně podle způsobu zapojení navrženého rozdělovače a použitého motoru k sepnutí dané cívky na záporný pól napájecího zdroje a přes tuto cívku začne procházet elektrický proud. Proud tekoucí přes cívku vyvolá elektromagnetické pole, které zapříčiní pootočení rotoru do takové polohy, kdy je dosaženo nejmenšího magnetického odporu.

Jak bylo zmíněno v kapitole 3.1.2, je ke každému vinutí motoru paralelně připojena Zenerova dioda sloužící jako ochranný prvek tranzistoru. Po přivedení kladného napětí na bázi tranzistoru začne kolektorem protékat proud, který je odebírán vinutím motoru. Pokud se ale přechod kolektor-emitor náhle uzavře, odpor tranzistoru prudce vzroste na vysokou hodnotu. Proud tekoucí cívkou motoru se však nemůže skokově zmenšit na nulovou hodnotu. Hrozil by lavinový průraz přechodu kolektor-emitor, a tím zničení tranzistoru. Proto se k indukční zátěži paralelně připojuje Zenerova dioda v závěrném směru z pohledu kladného napájecího napětí, která v případě uzavření tranzistoru dokáže uvolnit cestu nahromaděné energii.

3.1.4 Praktická realizace

V prvních fázích vývoje řídící jednotky byl rozdělovač impulsů zapojen pouze na nepájivém kontaktním poli. Z důvodu, aby se rozdělovač impulsů nepoškodil při manipulaci, nebo aby nedošlo k náhodnému odpojení některé součástky, byla pro rozdělovač vytvořena deska plošných spojů. Návrh této desky byl vytvořen v programu Eagle 5.11.0 na základě navrženého elektrického schématu. Navržená deska je zobrazena na obrázku 3.2. Po zhotovení desky byla tato osazena součástkami a byly připojeny signální a napájecí vodiče. Vytvořený rozdělovač byl spolu s motorem

připevněn na dřevěný hranol pro snadnou přenositelnost a ochranu propojovacích kabelů mezi motorem a rozdělovačem implulsů. Výsledný celek je patrný z obrázku 4.4 v kapitole 4.



Obrázek 3.2: Deska plošných spojů a její osázení součástkami

3.2 Krokový motor SMR 300-100-RI/24

Pro praktické odzkoušení navržené řídící jednotky bylo možno využít jakýkoliv krokový motor. Byla zvolena kroková reverzační pohonná jednotka SMR 300-100-RI/24, která byla poskytnuta vedoucím práce Ing. Josefem Strnadelem, Ph.D.

3.2.1 Popis motoru

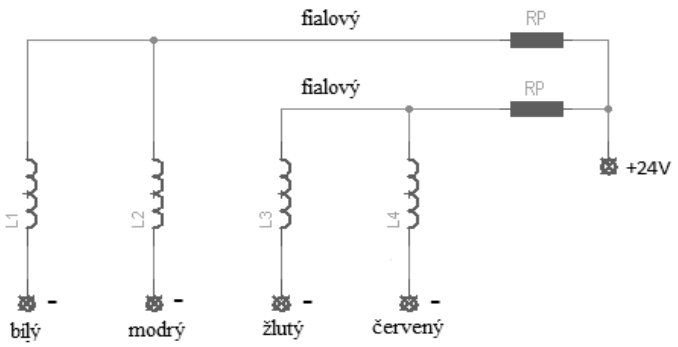
Jedná se o unipolární, čtyřfázový krokový motor s aktivním rotorem z permanentních magnetů, vyráběný firmou REGULACE – AUTOMATIZACE BOR, spol. s. r. o [1]. Motor lze ovládat pomocí čtyřtaktního nebo osmitaktního řízení. Ze způsobu řízení vyplývá počet kroků na jednu otáčku a úhel pootočení rotoru v jednom kroku. Počet kroků na jednu otáčku je ale především stanoven roztečí a počtem pólových nástavců na statoru motoru. U čtyřtaktního řízení lze dosáhnout 40 kroků po 9° na jednu otáčku. Při osmitaktním řízení lze krokování zjemnit, a to na hodnotu 80 kroků po 4,5° v rozsahu jedné otáčky rotoru. Veškeré ostatní parametry tohoto motoru lze nalézt v tabulce 3.1. Tabulka je převzata z katalogového listu výrobce motoru [2].

TECHNICKÉ ÚDAJE	čtyřtaktní řízení	osmitaktní řízení
Počet kroků na 1 otáčku	40	80
Úhel kroku	9°	4,5°
Tolerance úhlu kroku	0,27°	0,75°
Maximální rozběhový kmitočet	280 Hz	560 Hz
Maximální provozní kmitočet	800 Hz	1 700 Hz
Statický zatěžovací úhel	2,25° /10 m Nm	2,25° /5 m Nm
Přídržný moment	28 m Nm	22 m Nm
Vlastní přídržný moment	2 m Nm	
Amplituda proudu jedné fáze	0,25 A	
Napájecí napětí rozdělovače	24 V ss ± 10%	
Odpor vinutí fáze	30 ohmů ± 7%	
Indukčnost vinutí fáze	52 mH	
Předřadný odpor R_p	62 Ω/6 W	
Maximální radiální zatížení ložiska	1 N	
Maximální axiální zatížení ložiska	1,5 N	
Váha	0,165 kg	
Stabilizační zatěžovací moment	2 m Nm	
Moment setrvačnosti rotoru	$13{,}8.10^{-7}$kg.m^2	
Rozběhový moment	10 m Nm při 100 Hz	5 m Nm při 400 Hz
Rozběhový moment setrvačnosti	100.10^{-7}kg.m^2 při 100 Hz	100.10^{-7}kg.m^2 při 200 Hz

Tabulka 2.1: Technické parametry krokového motoru SMR 300-100-RI/24

3.2.2 Vnitřní zapojení krokového motoru

U motoru SMR 300-100-RI/24 mají vždy dvě sousedící vinutí společně vyvedeny tři vodiče k ovládání motoru. Jeden vodič je společný a slouží pro připojení k napájecímu zdroji na kladný pól. Tento vodič je označen fialovou barvou. Zbylé dva vodiče se pomocí rozdělovače impulsů vhodně přepínají na záporný pól, čímž se dosáhne požadovaného pohybu rotoru motoru. Schéma vnitřního zapojení motoru SMR 300-100-RI/24 je znázorněno na obrázku 3.3. Na obrázku je vidět předřadný odpor R_P, který slouží pro eliminaci tepelných výkonových ztrát vzniklých uvnitř v motoru. Tyto odpory dodává k motoru výrobce a jejich hodnoty stanovil na R_P = 68 Ω/6 W.



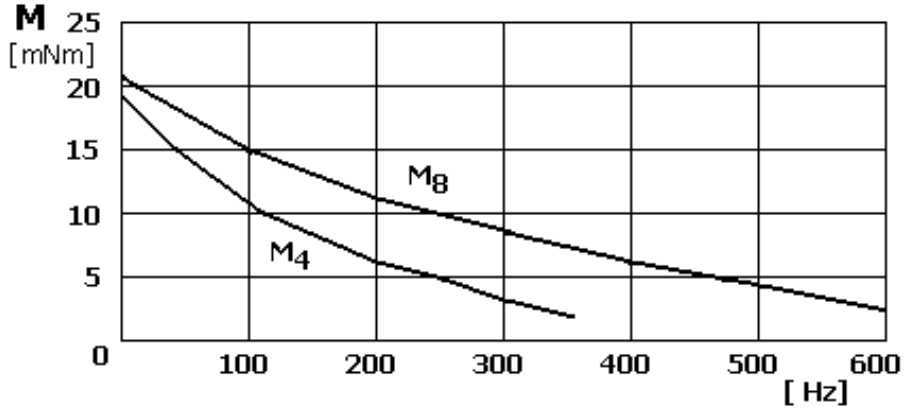
Obrázek 3.3: Vnitřní zapojení motoru s předřadnými odpory a s barevným značením vývodů z motoru SMR 300-100-RI/24

3.2.3 Momentové charakteristiky motoru SMR 300-100-RI/24

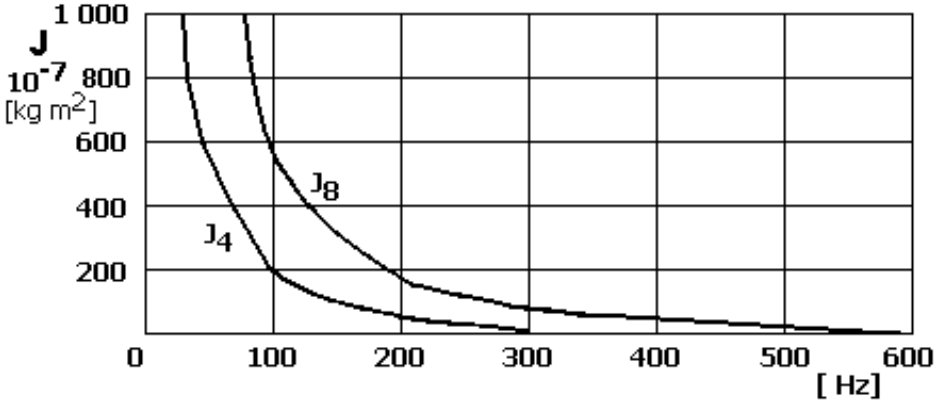
Momentové charakteristiky popisují dynamické jevy, které nastávají při provozu motoru. Tyto charakteristiky je potřeba vzít na vědomí při návrhu řídící jednotky. Níže uvedené charakteristiky jsou převzaty z katalogového listu výrobce motoru [2] a zobrazují průběhy při čtyřtaktním a osmitaktím řízení tohoto motoru.

Na obrázku 3.4 lze spatřit charakteristiku rozběhového momentu `M`, která určuje maximální možné zatížení rotoru vzhledem ke zvolenému rozběhovému kmitočtu. Meze této charakteristiky nesmí být překročeny, jinak by docházelo ke ztrátám kroku, nebo by se motor vůbec nerozběhl.

Na dalším obrázku je pak zobrazena charakteristika rozběhového momentu setrvačnosti `J`, se kterým je motor schopen se bezchybně roztočit bez zatížení zatěžovacím momentem a při definovaném řídícím kmitočtu.



Obrázek 3.4: Rozběhový moment motoru SMR 300-100-RI/24



Obrázek 3.5: Rozběhový moment setrvačnosti motoru SMR 300-100-RI/24

4 Platforma FITkit

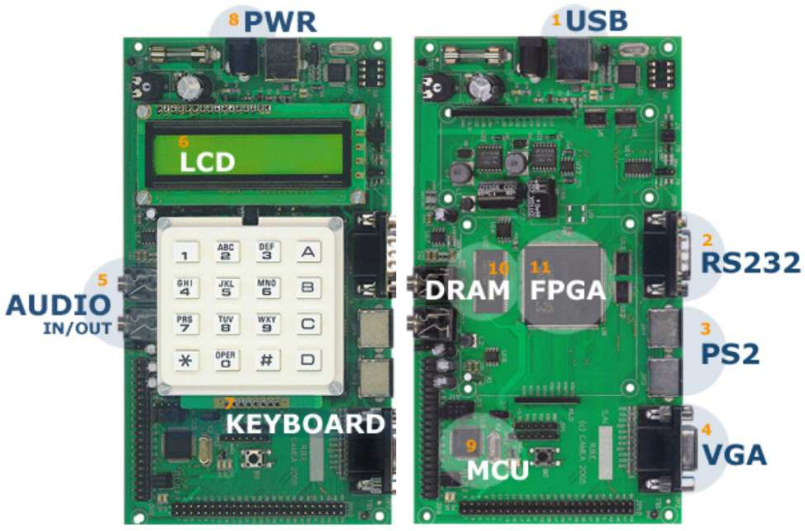
FITkit byl vyvinut na Fakultě informačních technologií VUT v Brně za účelem poskytnout studentům vhodnou učební pomůcku, pomocí které lze prakticky ověřit teoretické znalosti v oblasti vestavěných systémů. V této práci FITkit představuje jednotku, která řídí a provádí potřebné výpočty pro bezchybný chod krokového motoru.

V kapitole budou popsány význačné periferie a vlastnosti kitu, které jsou využívány při implementaci řídící jednotky. Veškeré informace uvedené v této kapitole byly čerpány z internetových stránek FITkitu [16].

4.1 Popis vývojového kitu

FITkit verze 2.0 je vybaven mikroprocesorem MSP430F2617, který tvoří základní jednotku celého kitu. K tomuto mikroprocesoru jsou pak prostřednictvím FPGA hradlového pole SPARTAN 3 připojeny další periferie, jako jsou například LCD displej nebo klávesnice. Tyto periferie mezi sebou komunikují prostřednictvím rozhraní SPI. FITkit obsahuje i další periferie, které však v této práci nebudou podrobněji popisovány vzhledem k tomu, že pro potřeby práce nebudou využívány. Pro úplnost je však uveden seznam nejdůležitějších periferií, které kit obsahuje:

- USB převodník FT2232C,
- audio rozhraní,
- konektory PS2,
- rozhraní VGA,
- konektor RS232,
- DRAM 8x8Mbit.



Obrázek 4.1: FITkit a popis základních periferií [17]

4.1.1 Komunikační rozhraní SPI

Rozhraní SPI umožňuje vytvořit interní sběrnici, ke které lze připojit potřebné periferie, a tím je zajištěna jejich vzájemná komunikace. Periferie mezi sebou komunikují podle SPI protokolu, který byl navržen právě pro tento účel. Základními jednotkami potřebnými pro komunikaci přes SPI je SPI řadič a SPI dekodér.

SPI řadič provádí konverzi SPI protokolu na vnitřní sériovou reprezentaci, která je dále dostupná na vytvořené interní sběrnici. K této sběrnici jsou jednotlivé komponenty připojeny přes SPI dekodér. SPI dekodér detekuje a analyzuje z interní sběrnice data určená pro konkrétní periferii a převádí sériově přenášené informace na paralelní rozhraní dané komponenty. Obecně platí, že každá komponenta připojená k navržené SPI interní sběrnici by měla mít svůj vlastní SPI dekodér. Každá komponenta je potom jednoznačně identifikovatelná podle toho, jaký bázový adresní prostor má přidělen SPI dekodér příslušné periferie. Informace čerpány z [9].

4.1.2 Mikroprocesor MSP430F2617

FITkit je osazen 16-bitovým nízkopříkonovým mikrokontrolérem rodiny MSP430 firmy Texas Instruments s těmito parametry [7]:

- nízké napájecí napětí (rozmezí 1.8 V až 3.6 V),
- nízký příkon (330 µA v aktivním režimu při 1 MHz a 2.2 V, 1.1 µA ve stand-by režimu, 0.2 µA v režimu vypnuto),
- 16-bitová RISC architektura, instrukční cyklus 125 ns,
- paměť FLASH 48 KB, paměť RAM 2 kB (resp. 92 kB FLASH a 8 kB RAM v případě FITkitu 2.0),
- moduly na čipu:
    - 3-kanálové DMA,
    - 16-bitové časovače Timer_A, Timer_B,
    - komparátor,
    - sériové komunikační rozhraní USART0 (asynchronní UART, synchronní SPI, I2C), USART1 (asynchronní UART, synchronní SPI),
    - 12-bitové A/D a D/A převodníky.

4.1.3 FPGA hradlové pole řady SPARTAN 3

Platforma FITkit obsahuje mj. programovatelné hradlové pole (anglicky Field Programmable Gate Array) XC3S50-4PQ208C firmy Xilinx. Toto hradlové pole umožňuje navrhnout libovolnou hardwarovou součástku a přiřadit jí požadované chování. Tato vlastnost představuje značný

pokrok, snadnou rekonfigurovatelnost, úsporu času a financí oproti fyzické realizaci tohoto hardwaru např. na desce plošných spojů. K popisu navrhovaného hardwaru slouží programovací jazyk VHDL. FPGA hradlové pole obsažené na FITkitu má následující prvky [8]:

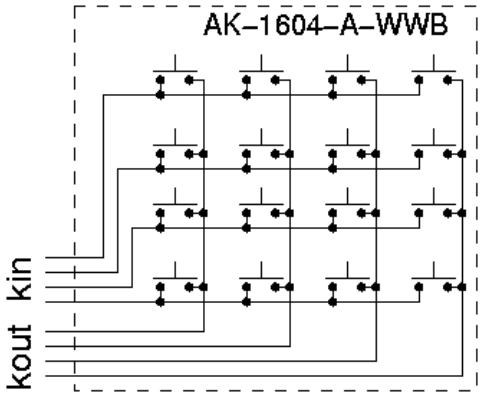
- rozhraní: až 124 uživatelských vstupů/výstupů (I/O), podpora až 23 různých I/O standardů,
- 192 konfigurovatelných logických bloků (CLBs) uspořádaných do matice o 16 řádcích a 12 sloupcích, 1728 logických buněk, 50000 logických hradel,
- paměť RAM 12 kb distribuovaných, 72 kb v jednom bloku,
- 2 jednotky pro správu hodin (DCMs),
- blokové dvouportové paměti BRAM s kapacitou 2 kB,
- násobičky 18x18 bitů.

4.1.4 LCD displej

Fitkit verze 2.0 obsahuje dvouřádkový LCD displej, na kterém je možno v jednom řádku zobrazit až 16 znaků. LCD displej je připojen přímo k FPGA, kde je implementován řadič tohoto displeje. Řadič obsahuje paměť ROM, která obsahuje bodové předlohy ASCII znaků, a dále dvě paměti RAM potřebné pro definování vlastních znaků a uchování aktuálního stavu, v němž se displej nachází.

Ve vytvořené aplikaci řídící jednotky je displej využit pro komunikaci s uživatelem, resp. pro zobrazení vstupních hodnot zadávaných uživatelem. Komunikace mezi mikroprocesorem a LCD je realizována pomocí rozhraní SPI.

4.1.5 Klávesnice



Obrázek 4.2: Zapojení klávesnice [19]

Pro zadání vstupních parametrů, které definují režim otáčení motorku, jsem využil maticové alfanumerické klávesnice, která je obsažena na FITkitu. Tato klávesnice je rovněž připojena k FPGA a pro správnou funkci je nutné používat vhodně navržený SPI řadič a dekodér.

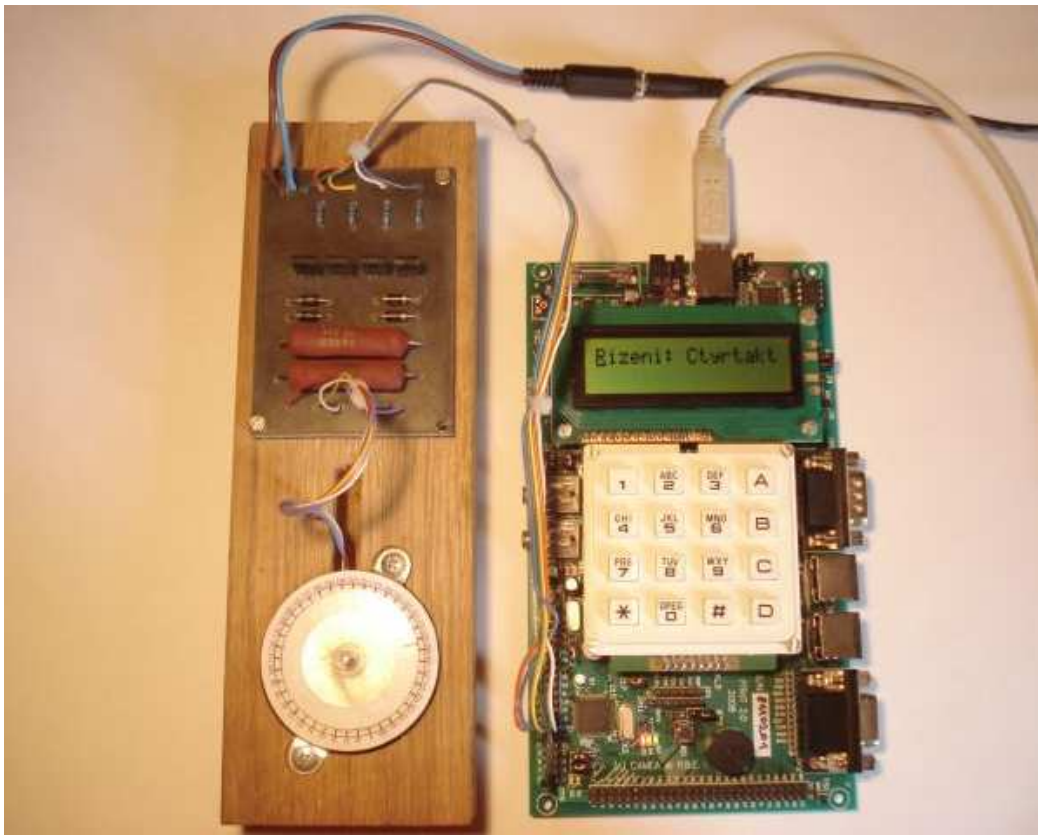
Implementace SPI řadiče a dekodérů pro práci s LCD displejem a klávesnicí byly převzaty z ukázkových aplikací z repozitáře pro FITkit.

4.2 Propojení FITkitu s rozdělovačem impulsů

Rozdělovač impulsů je k FITkitu připojen prostřednictvím rozhraní JP9. Na toto rozhraní je vyvedeno několik portů z mikrokontroléru (MCU). Rozdělovač je konkrétně připojen na port P3M z toho důvodu, že tento port je sdílen mezi MCU a FPGA viz [18]. Není tedy nutné, aby v případě volby řízení motorku z MCU či FPGA byl rozdělovač přepojován na jiný port či konektor. Zapojení jednotlivých vodičů rozdělovače impulsů k jednotlivým vývodům rozhraní JP9 na FITkitu je uvedeno v tabulce 4.1.

Vývod na JP9	Barva vodiče
1 (GND)	zelená
13	bíla
14	žlutá
15	modrá
16	oranžová

Tabulka 4.1: Mapování vývodů JP9 na vodiče rozdělovače impulsů



Obrázek 4.4: Pohled na FITkit s připojeným rozdělovačem a motorem osazeným úhlovou stupnicí

5 Řídící jednotka pro MCU

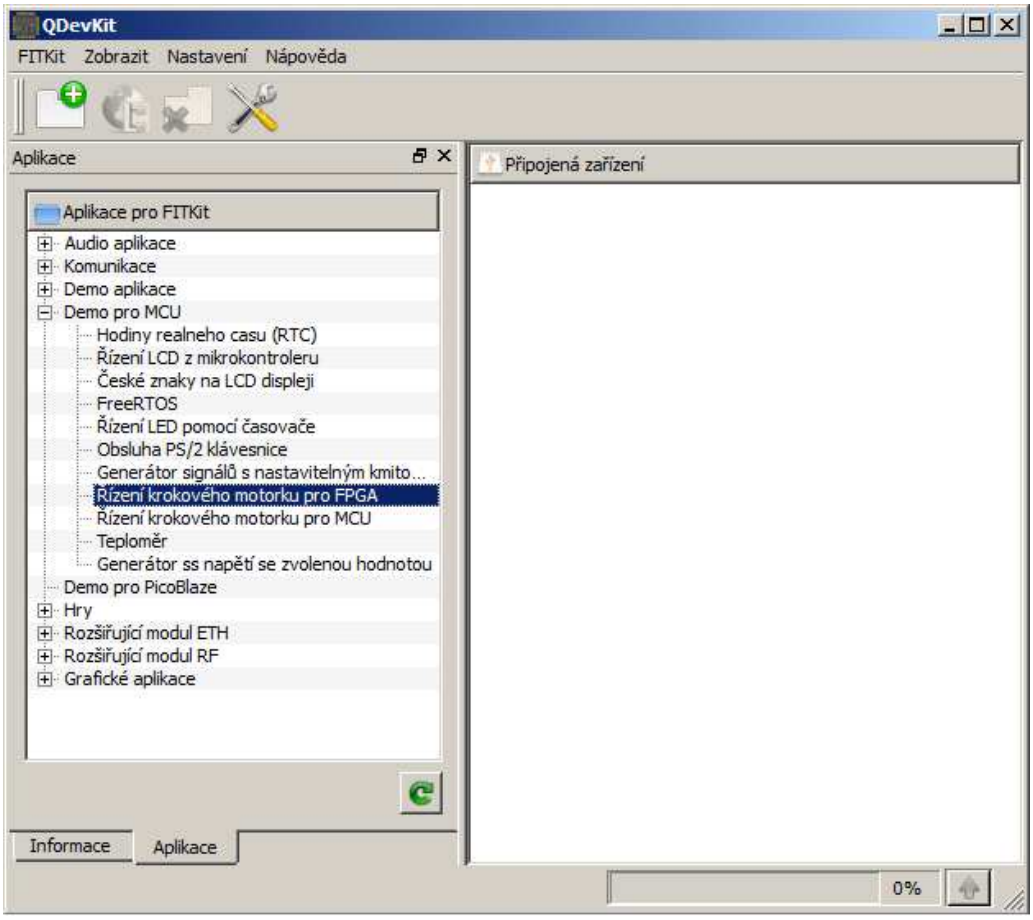
V této kapitole bude popsán návrh a implementace řídící jednotky v programovacím jazyce C pro mikrokontrolér MSP430F2617. Budou zde dále popsány jednotlivé fáze návrhu a různé problémy, které bylo nutné při implementaci řešit, aby byla zajištěna bezchybná činnost připojeného krokového motoru.

Řídící jednotka slouží výhradně pro demonstraci řízení krokového motoru, při které je kladen důraz na možnost naprosto libovolně periodicky měnit vstupní parametry, aniž by došlo k chybě krokování.

5.1 Získání uživatelských vstupů

Uživatelskými vstupy se rozumí hodnoty, které uživatel zadá mikrokontroléru ke zpracování. U FITkitu bylo možno volit ze dvou možností pro vstup uživatelských parametrů:

- LCD displej a klávesnice na FITkitu,
- terminálové okno skriptovatelného terminálu QdevKit.



Obrázek 5.1: Terminálové okno aplikace QdevKit

Pro vstup hodnot byla zvolena možnost využití klávesnice a LCD displeje. Rozhodnutí pro tuto volbu bylo učiněno s ohledem na to, aby bylo možné řídící jednotku ovládat i v době, kdy nebude FITkit připojen pomocí USB kabelu k počítači. Dalším argumentem, který tuto volbu odůvodnil, bylo rovněž to, že v dnešní době je naprostá většina vestavěných systémů, řízených pomocí jednočipových mikrokontrolérů, ovládána právě pomocí těchto hardwarových prostředků.

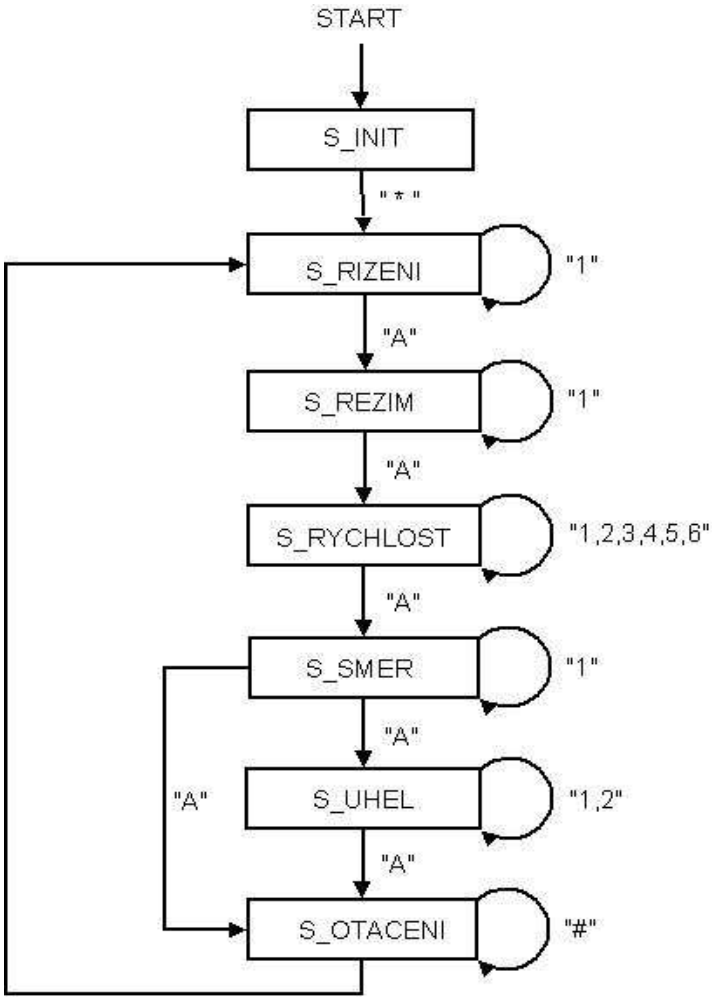
Aby se předešlo neočekávaným chybám při řízení motorku, nebo aby se nepřesáhly reálné parametry, kterých je schopen použitý motorek dosáhnout, bylo nutné všechny zadávané parametry důkladně kontrolovat, respektive těmto chybám předcházet. Toto opatření lze chápat tak, že v programu na úrovni zdrojového kódu jsou předdefinované konstanty. Program potom pouze detekuje stisk příslušného tlačítka na klávesnici a do interní proměnné přiřazuje tyto konstanty. Nedochází tedy k tomu, aby uživatel prostřednictvím klávesnice zadal na vstup nevalidní hodnoty, na kterých by pak řídící jednotka mohla zhavarovat. Detailní popis procesu načítání parametrů od uživatele bude popsán v podkapitole 5.1.2.

Řídící jednotka je navržena tak, aby po zadání vstupních hodnot od uživatele bylo započato otáčení motorku podle zadaných parametrů. Až rotor motorku dosáhne požadované polohy, nebo pokud bude otáčení zastaveno stisknutím příslušné klávesy, bude uživatel znovu vyzván k zadání nových vstupních parametrů. Oba procesy se tedy opakovaně střídají až do okamžiku, kdy bude mikrokontrolér odpojen od napájecího napětí. Proces získávání vstupních parametrů probíhá podle stavového automatu a řídící jednotka požaduje zadání těchto parametrů v uvedeném pořadí:

- **typ řízení**- možnost volby čtyřtaktního nebo osmitaktního řízení,
- **režim otáčení**- trvalé otáčení nebo otočení o zadaný úhel,
- **rychlost**- rychlost otáčení motorku v Hz, tj. počet kroků za vteřinu,
- **směr**- směr otáčení motorku, možnost volby vpravo či vlevo,
- **úhel**- volba velikosti úhlu natočení. Tato volba se zpřístupní pouze v režimu otáčení o zadaný úhel.

5.1.1 Stavový automat procesu získání parametrů

Celý proces získání parametrů je složen z několika jednotlivých voleb. Pro snadný přechod mezi jednotlivými volbami bylo navrženo jednoduché rozhraní s využitím klávesnice, kdy se po stisknutí potvrzovacího tlačítka přejde k volbě dalšího parametru. Přechod mezi jednotlivými volbami lze pro ilustraci popsat pomocí stavového automatu, kde jsou zmiňované volby jednotlivých parametrů označeny jako stavy tohoto automatu.



Obrázek 5.2: Stavový automat znázorňující proces získávání uživatelských vstupů

Na obrázku 5.2 je zobrazeno schéma stavového automatu. Toto schéma obsahuje sedm základních stavů (`S_INIT`, `S_RIZENI`, `S_REZIM`, `S_RYCHLOST`, `S_SMER`, `S_UHEL`, `S_OTACENI`). Ve schématu jsou dále znázorněna, jako symboly u přechodů mezi jednotlivými stavy, označení tlačítek, které způsobí přechod z jednoho stavu do dalšího (`*`, `A`). Dále lze ze schématu vyčíst seznam tlačítek, které určitým způsobem manipulují s právě získávaným parametrem (`1, 2, 3, 4, 5, 6, #`). Tento seznam tlačítek je umístěn vpravo od konkrétního stavu.

Na obrázku stavového automatu jsou zakresleny dva stavy, které nesouvisí s volbou parametrů otáčení motorku- `S_INIT`, `S_OTACENI`. Stav `S_INIT` znázorňuje vyzvání uživatele ke stisku tlačítka `*` pro počáteční inicializaci polohy motoru. Nastavení počáteční polohy se provede pouze jednou po naprogramování FITkitu. Stav `S_OTACENI` modeluje okamžik, kdy se zrovna motor otáčí. Otáčení lze zastavit stiskem tlačítka #.

Stav	Typ parametru	Manipulující tlačítka a jejich význam
S_INIT	Inicializace polohy motoru	*- počáteční inicializace polohy rotoru
S_RIZENI	Typ řízení	1- volba čtyřtaktní/osmitaktní
S_REZIM	Režim otáčení	1- volba trvalé otáčení/natočení o úhel
S_RYCHLOST	Rychlost v Hz	1- inkrementace po stovkách 2- inkrementace po desítkách 3- inkrementace po jednotkách 4- dekrementace po stovkách 5- dekrementace po desítkách 6- dekrementace po jednotkách
S_SMER	Směr	1- volba vpravo/vlevo
S_UHEL	Úhel natočení	1- inkrementace o velikost jednoho kroku 2- dekrementace o velikost jednoho kroku
S_OTACENI	Probíhá otáčení motoru	#- zastavení otáčení

Tabulka 5.1: Doplňující tabulka ke stavovému automatu

5.1.2 Postup při zpracování parametrů

Inicializace polohy

Ihned po naprogramování mikrokontroléru je uživatel vyzván ke stisku tlačítka *. Stiskem tohoto tlačítka dojde k pootočení motoru na výchozí pozici. Mikrokontrolér dostane směrodatnou informaci o aktuální pozici rotoru, čímž se zabrání ztrátě kroku při rozběhnutí motoru. Inicializace proběhne pouze jedenkrát ihned po naprogramování FITkitu, jak lze odvodit ze schématu stavového automatu.

Volba způsobu řízení

Dále je uživatel vyzván k výběru způsobu řízení motoru (čtyřtaktní nebo osmitaktní řízení). Tento výběr je realizován tak, že po provedení předchozí volby (inicializace polohy motorku) se na LCD displeji zobrazí předdefinovaná hodnota- Ctyrtakt. Uživatel může tuto hodnotu buď potvrdit stiskem klávesy A, nebo může hodnotu změnit na hodnotu Osmitakt stiskem tlačítka 1. Tuto

hodnotu lze pak libovolně měnit vícenásobným stiskem klávesy 1 z jednoho způsobu řízení na druhý, dokud nebude volba definitivně potvrzena.

Jak je patrno na obrázku stavového automatu, volba řízení je již vyžadována periodicky po dosažení cílové polohy motorem nebo úmyslným zastavením motoru klávesou #.

Výběr režimu a směru otáčení

Volby týkající se režimu otáčení a směru otáčení jsou řešeny stejně jako volba typu řízení, tzn. na displeji je zobrazena výchozí hodnota, kterou lze měnit na jinou hodnotu stisknutím tlačítka 1.

Volba rychlosti otáčení

Získání rychlosti otáčení je oproti předešlým mechanismům řešeno odlišně z toho důvodu, že uživateli musí být umožněno zadat jakoukoliv hodnotu z rozsahu, jakým disponuje připojený krokový motor SMR 300-100-RI/24.

Jak je patrné z tabulky 5.1 výše, pro volbu rychlosti je využito šest tlačítek. Program detekuje stisky jednotlivých tlačítek a potom provádí inkrementaci či dekrementaci o hodnotu jedna v číselném řádu (stovky, desítky, jednotky), kterému odpovídají příslušné klávesy. Program dále musí kontrolovat, aby se hodnota nedostala do oblasti záporných čísel, nebo aby nepřesáhla maximální možnou hodnotu rychlosti, která je dána způsobem řízení motoru.

Volba úhlu natočení

Tato volba se zpřístupní pouze tehdy, pokud byl dříve vybrán režim natočení rotoru o zvolený úhel. Pokud je volba zpřístupněna, uživatel může stiskem klávesy 1 inkrementovat, nebo stiskem klávesy 2 dekrementovat hodnotu úhlu o velikost jednoho kroku. Tato velikost kroku je opět závislá na připojeném krokovém motoru a zvoleném způsobu řízení. Pro použitý motor SMR 300-100-RI/24 je velikost jednoho kroku při čtyřtaktním řízení 9°, při osmitaktním řízení 4,5°.

5.1.3 Vnitřní reprezentace a uchování parametrů

Všechny parametry, které mohou nabývat pouze dvou předvídatelných hodnot (typ řízení, směr otáčení, režim otáčení), jsou v programu reprezentovány jako číselné konstanty jazyka ANSI C.

```
#define CTYRTAKT 2
#define OSMITAKT 3
#define VLEVO 4
#define VPRAVO 5
#define OTACENI_UHEL 6
#define OTACENI_TRVALE 7
```

Jakmile jsou potvrzovány jednotlivé volby parametrů, jsou tyto parametry ukládány do datové struktury `T_PARAMS`, kde jsou uchovány pro pozdější zpracování v řídící části navržené jednotky.

```
typedef struct{
  int rezim_otaceni;       //trvale otaceni, natoceni o uhel
  int smer;                //vpravo, vlevo
  int rizeni;              //ctyrtaktni, osmitaktni
  int rychlost;            //rychlost otaceni v HZ
  int poc_pul_stupnu;      //zadany uhel v poctu pulstupnu
  int pozice;              //pozice motoru
} T_PARAMS;
```

V této struktuře také figuruje položka `poc_pul_stupnu`. Tato položka představuje zadaný úhel v počtu půl stupňů. Úhel je tedy reprezentován celočíselným typem `integer`. Toto řešení bylo zvoleno z toho důvodu, že v případě osmitaktního řízení může hodnota zadaného úhlu nabývat i desetinné části (násobky hodnoty 4,5). Tuto hodnotu pak bylo nutné nějakým způsobem zobrazit na displeji pro informovanost uživatele. Funkce `LCD_append_string(char *data)` pro zobrazení textového řetězce na displeji FITkitu vyžaduje, aby se jí předal její jediný parametr (zobrazovaný řetězec) jako pole hodnot typu `char`. Bylo tedy nutné proměnnou, která reprezentuje zadaný úhel, do tohoto pole nakopírovat. Pro takový převod existuje v knihovně jazyka C *stdio.h* funkce `sprintf(char * str, const char * format,…)`, která umožňuje do alokované paměti nakopírovat obsah proměnné, která může být libovolného datového typu. Bohužel má tato funkce na většině jednočipových mikrokontrolérů problémy s převodem proměnné typu `float` či `double` na výše zmíněné pole hodnot typu `char` a korektně pracuje pouze tehdy, je-li převáděná hodnota celočíselná nebo rovněž typu `char`. Z tohoto důvodu je hodnota úhlu uvnitř programu reprezentována v počtu půl stupňů a pro korektní zobrazení hodnoty úhlu v úhlových stupních se v programu využívá tohoto algoritmu:

```
pul_stupne+=9;
sprintf(prvni, "%d", pul_stupne/2);
sprintf(druhy, "%d", (pul_stupne%2)==1?5:0);
strcat(msg,prvni);
strcat(msg,".");
strcat(msg,druhy);
LCD_clear();
LCD_append_string(msg);
```

Tento algoritmus pracuje tak, že se do řetězce (ve skutečnosti pole typu `char`) `prvni` nakopíruje celá část hodnoty úhlu. Do řetězce `druhy` se nakopíruje desetinná část této hodnoty. Tato desetinná část se vypočte pomocí operace zbytek po celočíselném dělení. Dále se pak tyto dílčí řetězce pomocí operace konkatenace řetězců postupně přidávají do jednoho výsledného řetězce `msg`, kde jsou dílčí řetězce odděleny desetinnou tečkou. Po této operaci je už hodnota úhlu ve správném tvaru a lze ji bezpečně zobrazit na LCD displeji v korektní podobě.

5.1.4 Obsluha LCD a klávesnice

Jak bylo zmíněno dříve, pro získání uživatelských vstupů a pro informovanost uživatele, který řídící jednotku ovládá, bylo využito klávesnice a LCD displeje umístěných na FITkitu. Vývojový tým FITkitu implementoval pro obsluhu těchto komponent řadu užitečných funkcí a maker. Tyto funkce jsou v programu využity pro snadnou a efektivní práci s použitými komponentami.

Obsluha displeje

Obsluha displeje na navržené řídící jednotce spočívala pouze v zobrazování aktuální hodnot zadávaných uživatelem, nebo ve vymazání displeje před zobrazením nové hodnoty. K tomu je v programu využito funkcí `LCD_append_string(char *data)` a `LCD_clear()`. Tyto funkce již samy zajišťují bezpečnou a korektní práci s displejem.

Obsluha klávesnice

Pro detekci stisku tlačítka na klávesnici a příjem dat se nabízely dvě možnosti:

- aktivně sledovat obsah řadiče klávesnice a přes rozhraní SPI z něj vyčítat,
- pracovat s klávesnicí pod přerušením.

V programu je využito první varianty. Z hlediska efektivity je pro vstup parametrů od uživatele naprosto postačující. Pro každou jednotlivou volbu parametrů je implementována samostatná funkce, která obsahuje nekonečný cyklus. V tomto nekonečném cyklu program detekuje a dekóduje stisknutou klávesu a ukládá příslušný symbol do proměnné `znak`. V případě, že není stisknuté žádné tlačítko, bude proměnná `znak` obsahovat nulu. Pokud je potom detekována jedna z požadovaných kláves, provede se uvnitř cyklu odpovídající operace (inkrementace hodnoty, dekrementace hodnoty,…). Tyto operace se ale neprovedou ihned. Po detekci tlačítka se odpovídající symbol uloží i do pomocné proměnné a tato pomocná proměnná se porovnává s obsahem proměnné `znak`. Pokud obě proměnné mají jiný obsah, znamená to, že tlačítko již bylo uvolněno, čímž se zabrání vícenásobnému provedení operací uvnitř cyklu v rámci jednoho stisku tlačítka.

Jakmile program detekuje stisk potvrzovacího tlačítka, okamžitě se nekonečný cyklus opustí a chod programu přejde do následujícího stavu podle stavového automatu na obrázku 5.2. Algoritmus

detekce stisku tlačítka je demonstrován na následujícím fragmentu zdrojového kódu a na vývojovém diagramu.

```
while(1){
  znak = key_decode(read_word_keyboard_4x4());
  if (znak != poslední_znak){
    last_ch = znak;
    //operace pod prislusnymi tlacitky
    if(znak=='A'){      //opusteni cyklu
      break;
    }
  }
}
```

Obrázek 5.3: Vývojový diagram znázorňující detekci stisku tlačítka na klávesnici

5.2 Řídící část

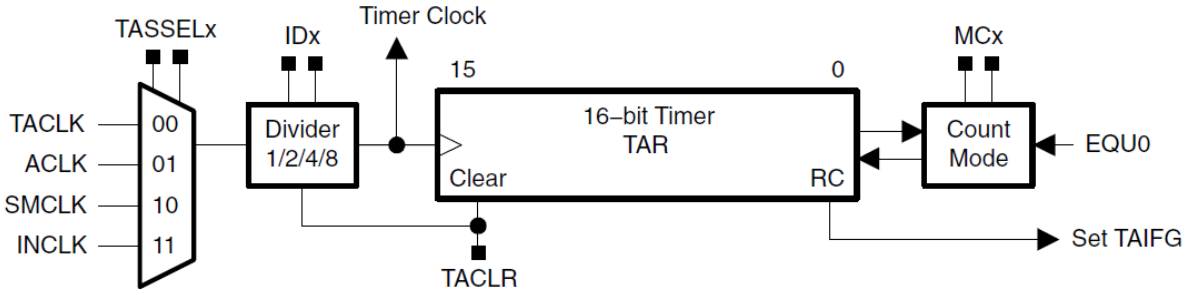
V této kapitole bude podrobně popsán vlastní algoritmus řízení připojeného krokového motoru. Bude vysvětleno, jaké techniky byly použity pro časování řídící jednotky, pro výpočet nové polohy hřídele motoru, a jakým způsobem se člení chod programu v závislosti na požadovaném režimu otáčení motoru. Zdrojový kód řídící části lze nalézt v modulu *motor_controler.c*.

5.2.1 Hlavní programová smyčka

Tento nekonečný cyklus slouží pro periodické volání podprogramů zajišťujících zpracování uživatelských vstupů, nastavení časovače a vlastní řízení motoru. Volání těchto podprogramů je umístěno v cyklu z toho důvodu, aby bylo možné opakovaně provádět řízení motoru bez nutnosti reprogramování mikrokontroléru. Chod programu se do této smyčky dostane ihned po provedení inicializace všech potřebných programových komponent. Programová smyčka je umístěna v souboru *main.c*.

5.2.2 Časování řídící jednotky

Pro přesné odměřování časových prodlev mezi generováním výstupních řídících impulsů bylo využito služeb 16-bitového časovače `TIMER_A`, který je umístěn na mikrokontroléru.



Obrázek 5.4: Architektura časovače

Na obrázku 5.4 je zobrazeno základní schéma použitého časovače. Toto schéma není kompletní, jsou vynechány některé nepoužité registry a komponenty. Celé schéma je možné si prohlédnout ze zdroje [15].

Pomocí registru `TASSELx` lze zvolit mezi čtyřmi vstupními kmitočty, které určují periodu čítání časovače. Pro navrženou řídící jednotku bylo využito frekvence `ACLK`, jejíž hodnota je stanovena na 32768 Hz. Tato frekvence je ve zdrojových kódech reprezentována proměnnou `TASSEL_1`.

Časovač dále obsahuje programovatelnou děličku `IDx` vstupního kmitočtu, pomocí níž lze vhodně upravit periodu čítání.

Další důležitou komponentou je šestnáctibitový registr `TACCR0`, který obsahuje hodnotu, do které má časovač čítat. Časovač umožňuje zvolit mezi třemi způsoby čítání, které lze nastavit pomocí registru `MCx`. Hodnota `MC_1` znamená, že se časovač po dosažení hodnoty v `TACCR0` zastaví. `MC_2` představuje režim nepřetržitého čítání, kdy se časovač při dosažení požadované hodnoty vynuluje a začne čítat znovu. Mód `MC_3` definuje režim čítání do hodnoty `TACCR0` a následně dekrementaci k nule. `MC_0` způsobí okamžité zastavení časovače.

Jakmile je dosaženo požadované hodnoty, nastaví se příznak přerušení `TAIFG` na hodnotu 1 a v programu se vykoná obslužná rutina přerušení od časovače `interrupt (TIMERA0_VECTOR) Timer_A (void)`.

5.2.3 Nastavení časovače

Nastavení časovače je závislé na zvolené rychlosti otáčení. Pro časovač byla zvolena vstupní frekvence 32768 Hz. Při této frekvenci trvá časovači čítání k hodnotě 32768 přesně jednu sekundu. Pro řídící jednotku to tedy znamená, že by každou sekundu na výstupní port generovala řídící impuls, a tím by motor vykonal jeden krok.

Aby bylo možné dosáhnout vyšší rychlosti krokování, je nutné snížit hodnotu v registru TACCR0. Požadovaná hodnota čítání je nepřímo úměrná zvolené rychlosti v Hz, a lze ji určit pomocí jednoduchého výpočtu:

$$T_{TACCR0} = f_{ACLK}/v$$

kde:

- T_{TACCR0} - hodnota, do které bude časovač čítat,
- f_{ACLK} - vstupní kmitočet, v našem případě 32768 Hz,
- v- zvolená rychlost otáčení v Hz.

Příklad výpočtu:

Pokud je zvolena rychlost otáčení motorku 200 Hz, registr TACCR0 se nastaví na hodnotu 163 podle výpočtu:

$$T_{TACCR0} = f_{ACLK}/v = 32768/200 = 163.84$$

Při nastavení časovače je třeba brát v úvahu také charakteristiku rozběhového momentu. Zadanou hodnotu rychlosti otáčení je nutné porovnat s hodnotou maximálního rozběhového kmitočtu, která je dána velikostí zátěže působící na hřídele motoru. Pro testování řídící jednotky nebyla hřídel zatížena. Proto se zvolená rychlost porovnává s hodnotou maximálního rozběhového kmitočtu nezatíženého motoru, kterou výrobce použitého motoru stanovil na 280 Hz při čtyřtaktním řízení a 560 Hz při osmitaktím řízení. Obě tyto hodnoty jsou uloženy v modulu *params.h* jako celočíselné konstanty `STARTFREQMAX_4` a `STARTFREQMAX_8`.

Pokud zvolená rychlost přesáhne tyto hodnoty, nastaví se rozběhový kmitočet na maximální možný vzhledem k typu řízení a dále se tato rychlost s každým krokem inkrementuje o hodnotu 1 až do požadované hodnoty.

5.2.4 Princip řídícího algoritmu

Jakmile jsou úspěšně získány uživatelské vstupy, jsou uloženy do struktury a inicializován časovač, program započne generování řídících impulsů pro řízení připojeného motoru.

Nejdříve je nutné zjistit, jaký byl zvolen typ řízení - čtyřtaktní/osmitaktní a podle toho se chod programu dále rozděluje. Jakmile je tato informace známa, řídící jednotka vypočte, kolik kroků je nutné vykonat pro dosažení požadovaného úhlu natočení hřídele motoru. Jak bylo uvedeno v kapitole 5.1.3, úhel je v programu reprezentován jako počet půl stupňů. Z této hodnoty se počet kroků určí podle následujícího algoritmu:

```
if(parametry_motorku->rezim_otaceni==OTACENI_UHEL)
  pocet_kroku4=(parametry_motorku->poc_pul_stupnu/2)/9;
else
  pocet_kroku4=-1;
```

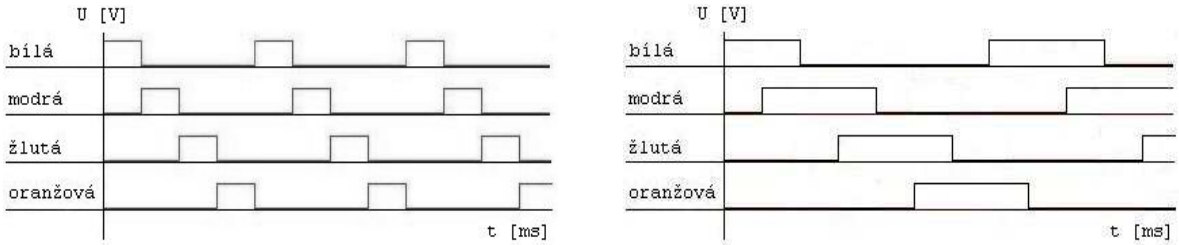
Tento algoritmus se použije při čtyřtaktním řízení, kdy je úhel odpovídající jednomu kroku roven hodnotě 9°. Při osmitaktním řízení je algoritmus mírně upraven v závislosti na velikosti jednoho kroku, která činí 4,5°. Pokud je zvolen režim trvalého otáčení motorku, nastaví se proměnná, která představuje počet kroků na hodnotu -1.

Jakmile je známa informace o počtu kroků, které je nutno vykonat, chod programu vstoupí do cyklu, který se opustí, až jsou provedeny všechny kroky, to znamená, až je dosaženo požadované polohy. Pokud je zvolen režim trvalého otáčení, nelze cyklus opustit jinak, než stisknutím tlačítka pro zastavení otáčení- #.

První akcí, která se provede po vstupu do tohoto cyklu, je dekrementace počtu kroků o hodnotu 1. Následuje velmi důležitý výpočet nové polohy rotoru, na jehož výstupu je číslo, které odpovídá hodnotě, kterou je nutné nastavit na výstupní port, aby se docílilo pootočení motorku o jeden krok ve správném směru beze ztrát kroku. Hodnoty nastavované na výstup jsou v programu uloženy v poli v hexadecimální podobě, takže výsledkem zmiňovaného algoritmu je index do tohoto pole. Tím je známa nová poloha motoru a je možno přikročit k nastavení odpovídající hodnoty na port P3M.

Před vlastním nastavením hodnoty na výstup se aktivuje časovač. Poté program čeká v nekonečném cyklu na vyvolání přerušení od tohoto časovače. Jakmile je detekováno přerušení, nastaví se hodnota na výstup, vynuluje se příznak přerušení, v obslužné rutině přerušení se časovač zinicializuje pro další použití a dojde k opuštění tohoto čekacího cyklu. Dojde k pootočení motoru a započne se nová série výpočtů pro provedení dalšího kroku.

Průběh řídících impulsů přiváděných na výstupní port je zobrazen na obrázku 5.5. V levé části obrázku je zobrazen průběh při čtyřtaktním řízení, v pravé pak při osmitaktním řízení. U každého průběhu je uvedena barva, která odpovídá barevnému značení jednotlivých vinutí použitého motoru.



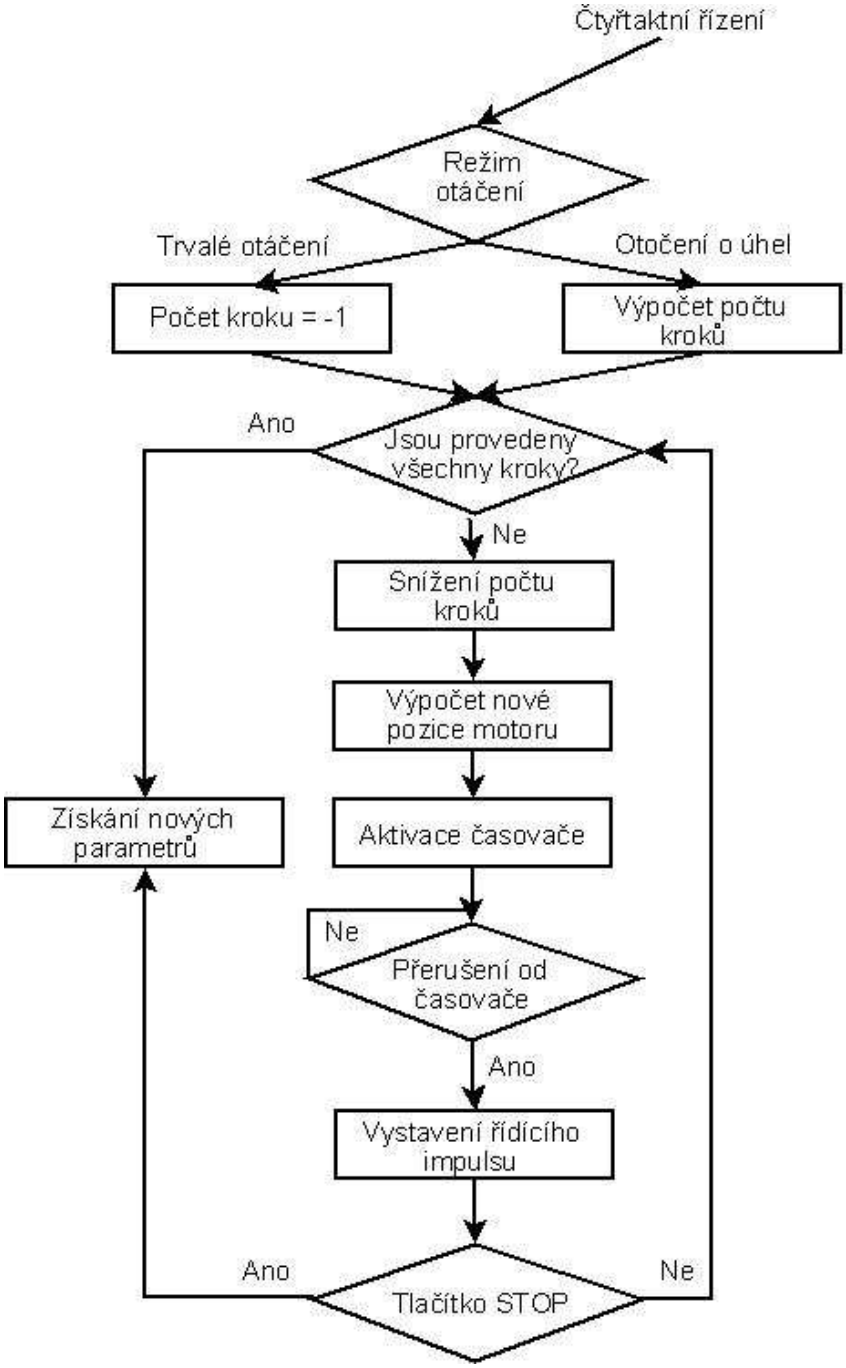
Obrázek 5.5: Průběhy řídících impulsů

Pro názornost a dodatečné vysvětlení je možné si prohlédnout hlavní části řídícího algoritmu na následujícím fragmentu zdrojového kódu a vývojovém diagramu:

```
int vystup[8]= {0x01, 0x03, 0x02, 0x06,0x04, 0x0C, 0x08, 0x09} }; //pole hodnot na vystup

while(pocet_kroku4!=0){
  pocet_kroku4--;
  if(parametry_motorku->smer==VLEVO){    //OTACENI DOLEVA
    parametry_motorku->pozice=(parametry_motorku->pozice % 8) + 1;
    if(parametry_motorku->pozice >=8)
      parametry_motorku->pozice=0;
  }
  else{                                  //OTACENI DOPRAVA
    parametry_motorku->pozice=(parametry_motorku->pozice % 8) - 1;
    if(parametry_motorku->pozice <0)
      parametry_motorku->pozice=7;
  }

  TACTL = TASSEL_1 + MC_1;     //aktivace casovace
  while(1) {       //cekani, dokud nedojde k preruseni od casovace
    if(timer_G==1){
    P3OUT= vystup [parametry_motorku->pozice];  //ridici impuls
    timer_G=0;
    break;
  }
 }
}
```



5.2.5 Rozšíření a omezení řídící jednotky

Rozšíření

Řídící jednotka podporuje dva typy řízení motoru. Při implementaci této jednotky bylo na výběr, zda se volba řízení provede jednou a definitivně, a při dalších procesech získávání parametrů se tato volba již nebude provádět, nebo bude druhým řešením umožněno měnit typ řízení vždy. Vytvořená řídící jednotka podporuje druhou možnost. Toto řešení pak sebou nese určitou nevýhodu, že v případě

čtyřtaktního řízení je vždy nutné kontrolovat, jakým typem řízení byl prováděn předchozí proces otáčení motoru.

Pokud totiž bylo předchozí řízení prováděno osmitaktně, a motor se zastavil v okamžiku, kdy byly aktivní dvě vinutí motoru (viz kapitola 2.5-Způsoby řízení), mohlo dojít při nově započatém čtyřtaktním řízení ke ztrátě jednoho kroku. Pokud nastane situace, že se motorek při osmitaktním řízení zastaví v této kolizní poloze, je tato skutečnost při nově započatém čtyřtaktním řízení detekována podle aktuální polohy rotoru, tedy podle indexu do pole výstupních hodnot, ze kterého byla použita odpovídající hodnota. Sudé indexy (včetně nultého) odpovídají stavu motoru, kdy je v daném okamžiku aktivní vždy jedna cívka (použito pro čtyřtaktní řízení), kdežto liché indexy odpovídají stavu, kdy jsou aktivní cívky dvě. Jestliže se tedy tato skutečnost úspěšně detekuje, bude se při započatém čtyřtaktním řízení přistupovat pouze na liché indexy místo sudých, to znamená, že budou v každém okamžiku sepnuty vždy dvě cívky. Tím se předejde chybám při krokování a dále toto řešení s sebou nese tu skutečnost, že se potom jedná o řízení čtyřtaktní dvoufázové.

Omezení

Navržená řídící jednotka má jednu nevýhodu. Nelze dosáhnout maximálního možného provozního kmitočtu, který garantuje výrobce použitého motoru. Tento nedostatek je způsoben časovou náročností všech výpočtů potřebných pro výpočet polohy hřídele motoru a výpočtů souvisejících s postupným zvyšováním rychlosti v případě, že zadaná rychlost je větší než maximální povolená rozběhová rychlost.

Při vyšších kmitočtech řízení se řídící jednotka dostane do stavu, kdy je perioda příchozích přerušení od časovače, na základě kterých dochází k vystavení řídícího impulsu na výstupní port, menší než doba potřebná pro výpočet všech důležitých informací. Vlivem setrvačnosti roztočeného rotoru a malého přídržného momentu použitého motoru potom dochází k překračování předpokládané polohy, a tím dochází k desynchronizaci mezi generováním řídících impulsů a touto předpokládanou polohou. Akumulací této chyby dojde později k úplnému zastavení motoru, kdy motor pouze rezonuje.

Garantované hodnoty lze dosáhnout s rozdělovačem impulsů RI 250-24-4/8, který dodává přímo výrobce k použitému motoru. S využitím tohoto rozdělovače není nutné provádět výpočty spjaté s polohou rotoru, ale stačí na jeho vstup přivádět pouze jeden řídící impuls, který určuje rychlost krokování, informaci o směru otáčení a typu řízení. Rozdělovač potom na úrovni slaboproudé elektroniky sám řídí, které vinutí má být sepnuto. Parametry tohoto rozdělovače lze nalézt v katalogovém listu výrobce [4].

6 Řídící jednotka pro FPGA

Tato kapitola informuje o návrhu a funkcionalitě řadiče krokového motoru, který je implementován v jazyce VHDL. V závěru kapitoly bude nastíněno, k jakým problémům došlo při realizaci této jednotky.

Při návrhu řadiče bylo uvažováno odlišení od verze pro mikrokontrolér při zachování stejných provozních režimů připojeného motoru. Toto odlišení spočívá v jiném způsobu vložení vstupních parametrů, přičemž toto řešení se více podobá reálně využívaným aplikacím v oboru automatizace, na rozdíl od čistě demonstrativní verze řídící jednotky pro mikrokontrolér.

6.1 Získání uživatelských vstupů

Parametry definující režim otáčení připojeného krokového motoru je pro tuto řídící jednotku nutno zadat přímo na úrovní zdrojového kódu. Jak bylo uvedeno výše, vytvořená jednotka by se měla co nejvíce podobat skutečně používané aplikaci, a z tohoto pohledu tedy parametry zadané na úrovni zdrojového kódu představují pouze inicializační hodnoty, na základě kterých je určeno, jakým způsobem má motor začít pracovat. Tyto parametry by pak mohly být měněny dynamicky, za chodu motoru, na základě příchodu elektrických impulsů od optických bran či inkrementálních snímačů otáček, přičemž by tyto snímače byly umístěny na zkonstruovaném polohovacím zařízení.

6.1.1 Vnitřní reprezentace a uchování parametrů

Tato řídící jednotka umožňuje volit mezi stejnými režimy řízení krokového motoru jako verze určená pro mikrokontrolér. Jednotlivé hodnoty jsou uchovány jako celočíselné hodnoty v proměnných typu `integer` (mapování jednotlivých parametrů na hodnotu, která je v programu detekována, je uvedeno níže, v tabulce 6.1). Výjimku tvoří parametr definující rychlost krokování a velikost úhlu. Rychlost lze zadat jako binární hodnotu, kterou nese 32-bitový signál typu `std_logic_vector`. Tento signál je potom využíván při časování vystavení řídících impulsů pro motor.

```
--tento signal predstavuje pozadovanou rychlost otaceni
signal rychlost : std_logic_vector(31 downto 0):= pozadovana_rychlost;

variable smer : integer := 5;          -- 4->vlevo, 5->vpravo
variable rizeni : integer := 2;        -- 2->ctyrtakt, 3->osmitakt
variable rezim : integer :=6;          -- 6-> o uhel, 7->trvale otaceni
variable poc_kroku : integer := 160;   --pocet pozadovanych kroku
```

Typ parametru	Hodnoty	Režim
Typ řízení	2	Čtyřtaktní
	3	Osmitaktní
Směr otáčení	4	Chod vlevo
	5	Chod vpravo
Režim otáčení	6	Otočení o úhel
	7	Trvalé otáčení

Tabulka 6.1: Mapování hodnot k režimu otáčení, který poskytuje daný parametr

Velikost úhlu, o který se má rotor pootočit, je zadán odpovídajícím počtem kroků. Toto řešení je zvoleno z toho důvodu, aby nebylo nutné uchovávat násobky základní velikosti jednoho kroku, a nemohlo tedy dojít k chybě.

6.2 Řídící část

V řídící části bylo oproti verzi pro mikrokontrolér provedeno minimum změn. Jedná se prakticky o přepis programu v jazyku C do jazyka VHDL. Bylo nutné implementovat vlastní jednoduchý časovač, podle kterého se odvíjí generovaní řídících impulsů pro motor.

6.2.1 Časování řídící jednotky

Jelikož FPGA obvod na FITkitu neobsahuje žádný časovač, bylo nutné vytvořit vlastní. Obvod FPGA obsahuje čtyři DCM bloky, které umožňují na základě hodinového signálu `SMCLK` vytvořit signál, který je v programu označován jako `CLK`, s jednou ze čtyř předdefinovaných hodnot frekvence. Pro účely řídící jednotky byl zvolen implicitní typ signálu `CLK` s frekvencí 25 MHz.

Program s každou náběžnou hranou signálu `CLK` zvyšuje hodnotu pomocného 32-bitového signálu `casovac` o hodnotu 1. Jakmile se dosáhne hodnoty, která je rovna hodnotě signálu představující rychlost krokování, vystaví se vhodný řídící impuls na výstupní port P3M.

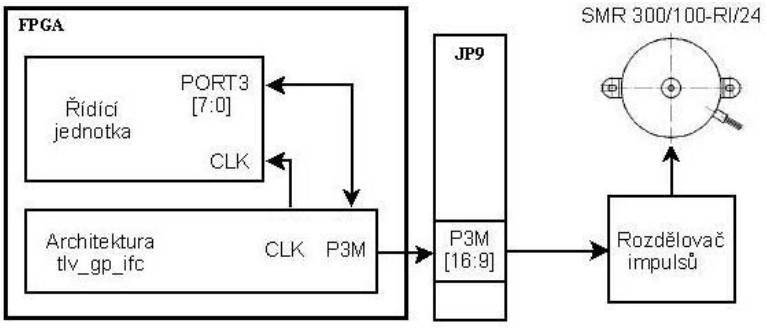
```
if ( (CLK'Event) and (CLK = '1') ) then
  casovac <= casovac + "00000000000000000000000000000001";  --casovac
  if (casovac >= vysledna_rychlost) then --kontrola hodnoty casovace
    casovac <= "00000000000000000000000000000000";
    --sekvence prikazu pro vlastní rizeni motorku
  end if;
end if;
```

V programu jsou také definovány dva signály, které nesou hodnotu maximálního možného rozběhového kmitočtu v závislosti na typu řízení. Pokud bude zvolená rychlost větší než tento výrobcem stanovený limit, provede se první krok s maximálním možným kmitočtem a s každým dalším krokem se tento kmitočet zvýší o předdefinovaný přírůstek, který lze ve zdrojovém kódu změnit pro pozvolnější, či prudší náběh rychlosti krokování.

6.2.2 Vlastní řídící jednotka

Řídící jednotku lze chápat jako prvek, do kterého vstupuje hodinový signál `CLK`. V závislosti na tomto signálu a na zvoleném režimu otáčení jsou potom na výstup tohoto prvku vystavovány řídící impulsy, které se pomocí signálu `PORT3` přivedou do tzv. top_level architektury `tlv_gp_ifc`. Tato architektura je propojena k portu P3M, který je vyveden na rozhraní JP9 a k tomuto portu je potom přes rozdělovač impulsů připojen motor. Komunikace mezi top_level architekturou a vlastní řídící jednotkou probíhá prostřednictvím navrženého komunikačního rozhraní, kdy jsou obě zmíněné části propojeny pomocí vhodně navržených signálů. Rozhraní řídící jednotky je zobrazeno níže – obrázek 6.1.

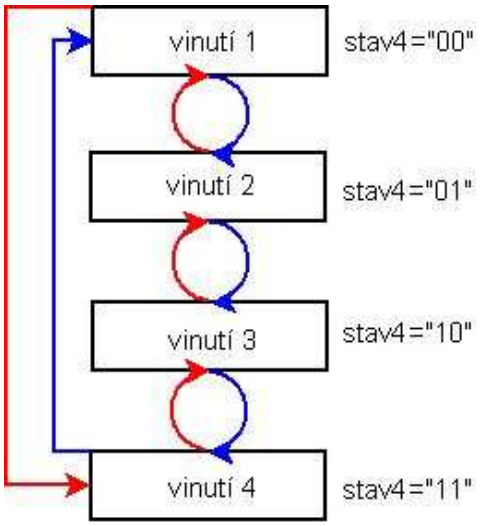
```
entity motcontr is
   port(
      CLK   : in std_logic;                        -- synchronizace
      PORT3 : inout std_logic_vector(7 downto 0) --ovladani portu P3M
    );
end motcontr;
```



Obrázek 6.1: Připojení motoru k řídící jednotce

Řídící algoritmus a větvení chodu programu se téměř neliší od verze řídící jednotky pro mikrokontrolér. Jakmile se detekuje konec čítání časovače, zjistí se, jakým typem řízení má být motor ovládán. Následuje upravení hodnoty časovače v případě, že požadovaná rychlost otáčení je nižší než aktuální. Jakmile je tento úkon proveden, je zapotřebí určit, kterou cívku je potřeba sepnout v závislosti na zvoleném směru otáčení a typu řízení.

Algoritmus, který rozhoduje o tom, na které vinutí motoru přivést řídící impuls vzhledem k aktuální pozici při čtyřtaktním řízení, lze popsat pomocí jednoduchého stavového automatu, který je uveden na obrázku 6.2. Na tomto obrázku představují červené šipky směr otáčení motoru vpravo a modré šipky symbolizují směr otáčení vlevo. Vpravo od každého stavu je uveden identifikátor `stav4`, který je v programu reprezentován jako 2-bitový signál. Na základě zadaného směru otáčení a hodnoty, kterou signál `stav4` nese, se potom na port P3M vystaví adekvátní hodnota.



Obrázek 6.2: Stavový automat reprezentující nastavení vhodného řídícího impulsu vzhledem k aktuální pozici rotoru

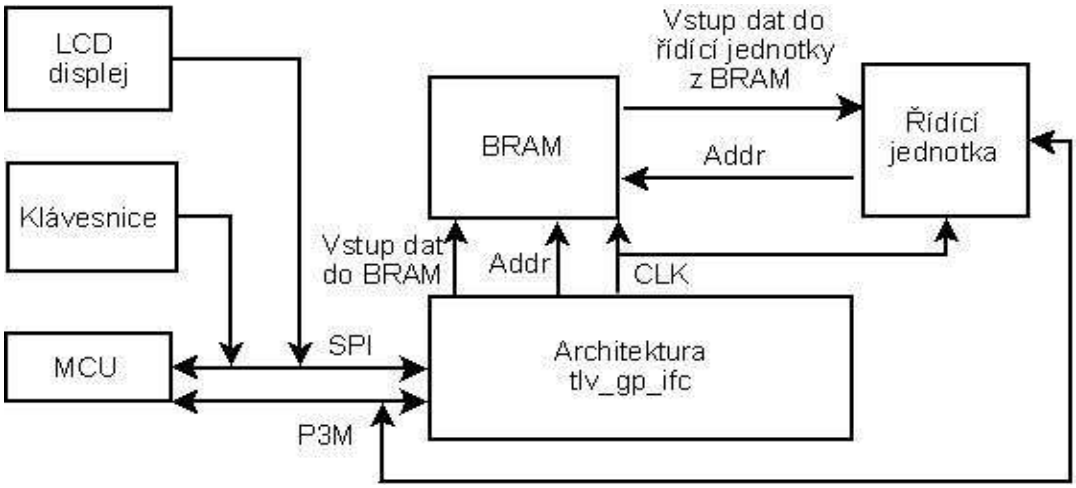
V případě zvoleného režimu natočení o úhel se po vystavení řídícího impulsu dekrementuje hodnota reprezentující počet požadovyných kroků. Pokud se následně zjistí, že již byly provedeny všechny kroky, znamená to, že bylo dosaženo cílové polohy a nastaví se pomocná proměnná `stop` na hodnotu 1. Při příchodu náběžné hrany signálu CLK se tato skutečnost detekuje, a tím dojde k zastavení činnosti řídící jednotky.

6.3 Možná rozšíření

Jakmile byla aktuální podoba řídící jednotky připravena, byl učiněn pokus implementovat ve zbývajícím čase její rozšířenou verzi. Princip této rozšířené verze spočíval ve využití mikrokontroléru, kdy mikrokontrolér měl za úkol zpracovávat parametry zadávané prostřednictvím klávesnice a displeje.

Jakmile byly parametry vloženy, mikrokontrolér tyto uložil pomocí sériového rozhraní SPI do dvouportové BRAM paměti, umístěné na FPGA. Poté bylo snahou aktivováním jednoho vývodu na úroveň log. 1 portu P3M, který je mezi mikrokontrolérem a hradlovým polem FPGA sdílený, dát druhé zmíněné komponentě povolení k započnutí vyčítání uložených hodnot a následně podle

získaných hodnot provést řízení připojeného motoru. Záměrem bylo, aby jakmile řídící jednotka pro FPGA dokončí řízení, dá zpětně mikrokontroléru stejným způsobem informaci, že může být započat nový proces načítání parametrů. Na obrázku 6.3 je uvedeno zjednodušené blokové schéma této rozšířené jednotky. V tomto schématu nejsou pro přehlednost a snadné pochopení uvedeny SPI dekodéry pro jednotlivé komponenty a signály pro propojení komponent jsou uvedeny ve zjednodušené a sjednocené podobě bez vyobrazení jejich datových šířek.



Obrázek 6.3: Blokové schéma plánovaného rozšíření řídící jednotky

Bohužel se z neznámých důvodů nepodařilo detekovat na portu P3M přivedený vstupní impuls, a proto nemohly být obě komponenty řádně synchronizovány.

Následně bylo vyzkoušeno jiné řešení, které spočívalo rovněž ve využití paměti BRAM, ale mírně se lišilo ve způsobu synchronizace mezi mikrokontrolérem a FPGA. Bylo zamýšleno propojit jeden vývod portu P4M, ke kterému má přístup pouze mikrokontrolér, s jedním vývodem portu X, ke kterému má přístup FPGA. U obou portů se podařilo detekovat příchozí impuls, ale nastal jiný problém, kdy došlo k naprosté desynchronizaci obsluhy klávesnice. Tento problém byl nejspíše způsoben neustálým analyzováním portu X, přičemž nedošlo k obsluze klávesnice, připojené k FPGA, ve správný okamžik.

Z těchto důvodů byla ponechána řídící jednotka ve stavu, kdy lze parametry zadávat pouze na úrovni zdrojového kódu.

7 Závěr

Teoretická část této bakalářské práce pojednává o principech činnosti a typech řízení krokových motorů. Součástí práce bylo navrhnout a implementovat dvě varianty řídící jednotky krokového motoru pro platformu FITkit. První varianta je určená pro řízení krokového motoru z mikrokontroléru, kdežto druhá verze řídící jednotky je určená pro hradlové pole FPGA.

Obě řídící jednotky se po praktickém odzkoušení na krokovém motoru SMR 300-100-RI/24 ukázaly jako funkční. Pro test funkčnosti řídících jednotek byl navržen a zkonstruován elektronický rozdělovač impulsů. Nad rámec zadání byl proveden pokus, kdy se na řízení motoru měly současně podílet jak mikrokontrolér, tak i hradlové pole. Bohužel toto rozšíření se nepodařilo uskutečnit, z důvodu nedostatku času a problémů, které byly popsány v závěru kapitoly 6.

U obou variant řídících jednotek vyvstal problém, kdy nebylo možné dosáhnout maximálního provozního kmitočtu. Tento nedostatek vznikl důrazem kladeným na preciznost polohování krokového motoru. Řešením by bylo použití rozdělovače impulsů, který dodával k motoru přímo výrobce. S využitím tohoto rozdělovače by bylo možné dosáhnout požadovaných hodnot rychlosti krokování, a práce by se značně zjednodušila. Tímto zjednodušením by však nebylo nutné realizovat téměř celou řídící část, čímž by práce podstatně ztratila na smyslu.

Navržené řídící jednotky poskytují dobrou základnu pro rozšíření jejich aktuálních verzí. Vývoj by se mohl ubírat směrem současného řízení několika krokových motorů, umístěných na zkonstruovaném polohovacím systému. Na tomto zařízení by byla současně umístěna skupina optických bran a inkrementálních snímačů otáček. Tyto prvky by potom zasílaly řídící jednotce informace o aktuální poloze pohyblivého jezdce a řídící jednotka by na základě těchto informací dynamicky měnila provozní režim připojených motorů.

Literatura

[1] spol. s r.o. REGULACE AUTOMATIZACE BOR. Regulace - automatizace Bor, spol. s r.o. [online], [cit. 14-01-2011], URL: http://www.regulace.cz/CZ.

[2] spol. s r.o. REGULACE AUTOMATIZACE BOR. Kroková reverzační pohonná jednotka SMR 300-100-RI/24. [online], [cit. 14-01-2011], URL:http://www.regulace.cz/DOWNLOADS /PDF/kl_km_300-100ri.pdf.

[3] spol. s r.o. REGULACE-AUTOMATIZACE BOR. Krokové motory – všeobecné údaje. [online], [cit. 14-01-2011], URL: http://www.regulace.cz/DOWNLOADS/PDF/kl_km_vu.pdf.

[4] spol. s r.o. REGULACE AUTOMATIZACE BOR. Rozdělovač impulzů RI 250-24-4/8. [online], [cit. 14-01-2011], URL: http://www.regulace.cz/DOWNLOADS/PDF/kl_km_ri.pdf.

[5] ŘEZÁČ, Kamil: Krokové motory. 2002-10-28, [online], [cit. 14-01-2011], URL:http://robotika.cz/articles/steppers/cs.

[6] HÁJEK, Josef: Řídící jednotka krokového motoru založená na mikrokontroléru HC08. Brno, 2007. Bakalářská práce. 43 s. Vysoké učení technické v Brně Fakulta informačních technologií.

[7] VAŠÍČEK, Zdeněk, STRNADEL, Josef: HADWARE / MCU -Základní parametry. 2010-01-04. [online], [cit. 25-04-2011], URL: http://merlin.fit.vutbr.cz/FITkit/docs/hardware/hw_mcu.html.

[8] VAŠÍČEK, Zdeněk, STRNADEL, Josef: HADWARE / FPGA -Základní parametry. 2009-03-18. [online], [cit. 25-04-2011], URL: http://merlin.fit.vutbr.cz/FITkit/docs/hardware/hw_fpga.html.

[9] VAŠÍČEK, Zdeněk: FIRMWARE / KOMUNIKAČNÍ SYSTÉM -SPI. 2009-09-17. [online], [cit. 25-04-2011], URL: http://merlin.fit.vutbr.cz/FITkit/docs/firmware/fpga_interconect.html.

[10] Katedra telekomunikační techniky, FEL ČVUT v Praze: Základní seznámení s jazykem VHDL. 2010. [online], URL: http://is.sssep9.cz/podklady/davidek/VOS2/VHDL/ jazyk%20VHDL_v2.1_20100901.pdf.

[11] SEKANINA, Lukáš: Studijní opora do předmětu INP-Úvod do jazyka VHDL. 2006. [online], URL: http:// https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/INP-IT/texts/inp_vhdl.pdf.

[12] HEROUT, Pavel: Učebnice jazyka C: 1. díl. Páté vydání. České Budějovice: KOPP, 2008.
.

[13] RYDLO, Pavel: Krokové motory a jejich řízení. Liberec, 2000. Studijní texty. Technická univerzita v Liberci, Fakulta mechatroniky a mezioborových inženýrských studií. [online], URL: http://www.mti.tul.cz/files/ats/krok2.pdf.

[14] BĚLÍČEK, David: Řízení krokového motoru mikrokontrolérem. Brno, 2009. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. [online], URL: http://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=15536.

[15] WANG, Yin: MSP430 Clock System and Timer. Northeastern University, 2007. [online], URL: http://www.ccs.neu.edu/home/noubir/Courses/CSU610/S07/MSP430-Clock-Timers.pdf.

[16] Vysoké učení technické v Brně, Fakulta informačních technologií: FITkit- Úvod. rev. 2011-03-05. [online], URL: http://merlin.fit.vutbr.cz/FITkit/.cs.

[17] Vysoké učení technické v Brně, Fakulta informačních technologií: FITkit- HARWARE. rev. 2011-03-05. [online], URL: https://merlin.fit.vutbr.cz/FITkit/private/ web/ index.php?pg=hardware.

[18] Vysoké učení technické v Brně, Fakulta informačních technologií: FITkit- schéma FITkit 2.0. [online], URL: http://merlin.fit.vutbr.cz/FITkit/download/schematic_v20.pdf.

[19] SLANÝ, Karel: Firmware / Klávesnice 4x4. 2009-03-19. [online], [cit. 25-04-2011], URL: https://merlin.fit.vutbr.cz/FITkit/private/web/index.php?pg=firmware&cl=fpga_keyboard.

Seznam příloh

Příloha 1

Obsah přiloženého CD nosiče

Adresáře:

- **stepperMCU** - obsahuje zdrojové soubory řídící jednotky pro mikrokontrolér, dále obsahuje podadresář *doc*, ve kterém je umístěna dokumentace k této jednotce ve formátu .rst,
- **stepperFPGA** - obsahuje zdrojové soubory řídící jednotky pro FPGA, dále obsahuje podadresář *doc*, ve kterém je umístěna dokumentace k této jednotce ve formátu .rst,
- **bp -** obsahuje vlastní text práce ve formátu .doc a .pdf.