Kamerový manipulátor pro stereovizi

Bc. Libor Odstrčil

Diplomová práce
2015

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2014/2015

ZADÁNÍ DIPLOMOVÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Libor Odstrčil**
Osobní číslo:	**A13425**
Studijní program:	**N3902 Inženýrská informatika**
Studijní obor:	**Informační technologie**
Forma studia:	**prezenční**
Téma práce:	**Kamerový manipulátor pro stereovizi**
Téma anglicky:	**A Stereovision Camera Rig**

Zásady pro vypracování:

1. **Prostudujte možnosti řízení krokových motorů pro ovládání pohybu kamer ve dvou osách.**
2. **Analyzujte možnosti zachytávání obrazu kamer z výstupu analogového videa, USB nebo HDMI.**
3. **Zpracujte možnosti vzdáleného ovládání spouště a ostření kamer.**
4. **Navrhněte schema zapojení celého systému a plošný spoj.**
5. **Implementujte firmware pro ovládání držáku pomocí dotykového LCD displeje.**

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **NOVÁK, Petr. Mobilní roboty: pohony, senzory, řízení. Vyd. 1. Praha: BEN – technická literatura, 2005, 247 s. ISBN 80-730-0141-1.**
2. **Control of Stepping Motors. The University of Iowa [online]. 1995 [cit. 2015-02-01]. Dostupné z: http://homepage.cs.uiowa.edu/ jones/step**
3. **HDMI [online]. 2003-2015 [cit. 2015-02-01]. Dostupné z: http://www.hdmi.org**
4. **ŠONKA, Milan, Václav HLAVÁČ a Roger BOYLE. Image processing, analysis, and machine vision. 3rd ed. Toronto: Thomson, 2008, 829 s. ISBN 978-0-495-08252-1.**
5. **HLAVÁČ, Václav. Zpracování signálů a obrazů. 1. vyd. Praha: Vydavatelství ČVUT, 2001, 220 s. ISBN 80-010-2114-9.**
6. **POYNTON, Charles. Digital video and HDTV: algorithms and interfaces. San Francisco: Morgan Kaufmann Publishers, 2003, 692 s. ISBN 15-586-0792-7.**
7. **VÁŇA, Vladimír. ARM pro začátečníky. Praha: BEN – technická literatura, 2009, 195 s. ISBN 978-80-7300-246-6.**
8. **CATSOULIS, John. Designing embedded hardware. 2nd ed. Sebastopol: O'Reilly, 2005, 377 s. ISBN 05-960-0755-8.**

Vedoucí diplomové práce: **Ing. Tomáš Dulík, Ph.D.**
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **6. února 2015**

Termín odevzdání diplomové práce: **15. května 2015**

Ve Zlíně dne 6. února 2015

L.S.

doc. Mgr. Milan Adámek, Ph.D.
děkan

doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl jsem seznámen s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

ABSTRAKT

Cílem této diplomové práce je vytvořit zařízení, které bude sloužit pro účely stereovize. Toto zařízení tvoří konstrukce, která umožňuje dvěma upevněným kamerám se pohybovat ve dvou osách, kterými jsou horizontální posun a rotace kolem svislé osy. Srdcem celého zařízení je řídicí deska s dotykovým LCD displejem. Za pomoci tohoto displeje lze ovládat zmíněný pohyb, přičemž samotný pohon tvoří dva krokové motory. Na displeji lze sledovat náhled z jedné či druhé kamery, popřípadě lze zobrazit obraz z obou kamer jako anaglyf 3D. Doplňující funkcí je možnost detekce bodu, kdy se jedna z kamer zaměří na zvolený bod a snaží se ho udržet ve svém zorném poli. Konektivitu řídicí desky tvoří Bluetooth modul, konektor pro trigger, 2 USB porty a další.

Klíčová slova: Stereovize, krokový motor, kamera, ARM, kompozitní video, LCD

ABSTRACT

The goal of this thesis is to create a device that will serve for stereovision. This device is formed by a construction that allows that two mounted cameras can move in two axes, which are horizontal motion and rotation around the vertical axis. The heart of the device is a control board with touch LCD display. With the help of this display we can control the said motion, while the drive consists of two stepper motors. On the display we can watch a preview of first or second camera, or we can watch the images from both cameras as anaglyph 3D. Additional feature is ability to detect a point where one of the cameras focuses on any point in trying to keep him in his field of vision. Connectivity control board consists of Bluetooth module, trigger connector, 2 USB ports, and more.

Keywords: Stereovision, stepper motor, camera, ARM, composite video, LCD

Rád bych na tomto místě poděkoval svému vedoucímu diplomové práce panu Ing. Tomáši Dulíkovi, Ph.D. za odborné vedení, cenné rady a připomínky, ochotnou pomoc a vstřícný přístup při vypracování mé diplomové práce.

Motto:

„Non schoale, sed vitae discimus.

- Neučíme se pro školu, ale pro život.“

Seneca

OBSAH

ÚVOD

Pokud se zaměříme na termín stereovize, neboli stereo vidění, tak se jedná o snímání a zpracovávání obrazu ze dvou snímacích zařízení, a to buď videokamery, nebo fotoaparátu. Důvodem takového způsobu snímání může být využití v oblasti počítačového vidění. V tomto případě lze zmínit použití pro výpočet vzdálenosti určitého objektu od kamer. Další oblastí, kde se dá setkat se stereovizí, je v současné době pravděpodobně oblast tvorby 3D obsahu. Zde se může jednat o tvorbu 3D fotek nebo samotného 3D videa.

Tato práce se zabývá tvorbou zařízení, které umožňuje jednoduché ovládání polohy kamer při aplikaci stereovize. V teoretické části se zaměřím na několik témat, které souvisí s konstrukcí zařízení. Protože budeme potřebovat pohon pro pohyb kamer, tak se zmíním o možných typech krokových motorů a o způsobech jejich řízení. Současně uvedu několik integrovaných obvodů, které se dají v současnosti sehnat, za účelem ovládání krokových motorů. Další kapitolou je zachytávání video signálu z kamer. Zde se zmíním o třech způsobech, jak je možné přenášet video obraz z kamer a následně ho zpracovat pro použití mikropočítačem. Mezi tyto tři způsoby přenosu patří analogové video, HDMI nebo USB. Vždy se budu snažit popsat základy takového přenosu a uvést, zda jsou potřeba pro zachytávání integrované obvody a jaké. Poslední kapitolou teoretické části je rozebrat, jaké existují možnosti pro vzdálené ovládání automatického ostření a spouště kamer. Zde provedu rozdělení na drátové a bezdrátové ovládání, kde zmíním několik technik a uvedu základní principy funkčnosti.

V praktické části se zaměřím především na samotnou tvorbu celého zařízení. Nejdříve popíši samotnou konstrukci, kterou tvoří krokové motory, nástavce pro upevnění kamer, převody a zabezpečovací prvky. Následně se zaměřím na řídicí desku, která bude celé zařízení ovládat. Nejdříve se zmíním o samotných funkcích, které by měla tato řídicí deska umět. Na základě tohoto bude proveden návrh hardwaru. Budu se snažit popsat jednotlivé funkční části obvodu a popsat, jak byly řešeny a za využití jakých součástek. Po návrhu hardwaru se budu snažit popsat také firmware řídicí desky. V této kapitole se opět zaměřím na hlavní funkční části, mezi které patří samotné mikropočítače, LCD displej, video převodník nebo ovladač krokových motorů. Nakonec popíši celou strukturu programu u každého z obou mikropočítačů.

I. TEORETICKÁ ČÁST

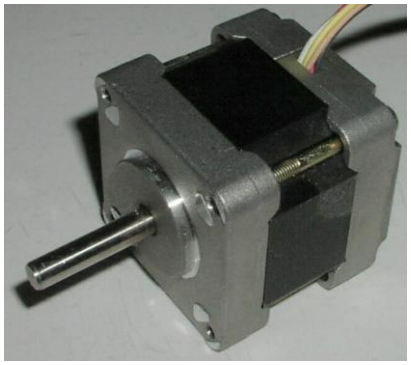
1 KROKOVÉ MOTORY

V praxi se naskytne potřeba využít pohonu, který umí přesně nastavit svoji polohu a v této poloze, i přes působící síly, se udržet. V tomto případě se uplatní krokový motor. Jedná se o synchronní, impulsně buzený motor, jehož funkční pohyb je nespojitý, protože je prováděn po jednotlivých úsecích (krocích).

Krokový motor, jako každý jiný motor, se skládá ze statoru a rotoru. Stator je v případě krokového motoru obvykle vyroben z ocelových lamel opatřených drážkami, v nichž se nacházejí měděná vinutí. V některých případech jsou vinutí nahrazena cívkami z důvodu levnějšího navíjení. Rotor krokového motoru může být tvořen železným jádrem, permanentními magnety nebo permanentní magnet může být vložen do pevného nebo laminovaného železného jádra.

K řízení krokového motoru se využívají ovladače, přímo určené pro ovládání krokových motorů. Tyto ovladače řídí funkční pohyb a režimy chodu krokového motoru. Příslušný ovladač řídí daný krokový motor tak, že budí jednotlivé fáze vinutí krokového motoru v jisté časové posloupnosti. Ovladač musí obvykle splnit dva požadavky. Jednak musí zajistit výkonové buzení fází motoru a dále vytvořit předepsanou posloupnost buzení jednotlivých fází motoru. Základními částmi ovladače je výkonová část a komutátor. Výkonová část je obvykle tvořena výkonovými spínacími prvky, jejichž počet odpovídá počtu fází krokového motoru. Funkcí komutátoru je na základě vstupních informací řídit spínání výkonových prvků tak, aby každému řídicímu impulsu odpovídalo natočení krokového motoru o jeden krok.

Krokové motory jsou používány především díky jejich snadné obsluze a nákladům na jejich pořízení a provoz. Avšak jejich hlavními nevýhodami jsou vibrace a hluk při provozu, společně s omezeným rozsahem rychlosti. Mezi nejzávažnější nevýhodu se pravděpodobně považuje trvalý odběr proudu, i když se motor neotáčí. [2, 4, 5, 6]



Obr. 1. Krokový motor. [3]

1.1 Druhy krokových motorů

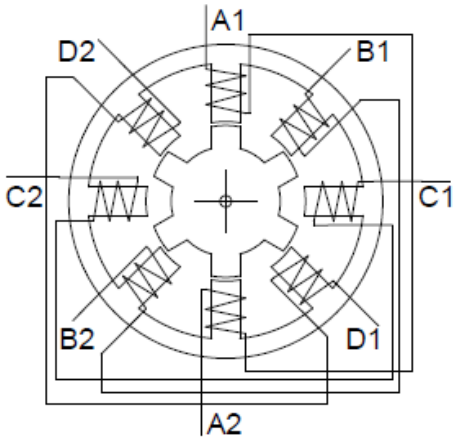
Podle konstrukčního uspořádání dělíme krokové motory na dvě základní skupiny:

- Krokové motory s pasivním rotorem
- Krokové motory s aktivním rotorem

1.1.1 Krokové motory s pasivním rotorem

Krokové motory s pasivním rotorem jsou také označovány jako reakční či reluktanční nebo krokové motory s proměnnou reluktancí. Z konstrukčního hlediska je rotor tvořen pouze svazkem plechů, nalisovaných na hřídel. V tomto případě je rotor bez vinutí. Stator tohoto typu motoru je tvořen určitým počtem pólů s navinutými cívkami. Příkladem může být krokový motor, jehož řez je zobrazen na obrázku (Obr. 2), zde je stator tvořen 8 póly. Jednu fázi zde tvoří dvojice cívek na protilehlých pólech, které jsou vzájemně spojeny. Krokový motor má tudíž 4 fáze, které označme A, B, C a D. Podle buzení jednotlivých fází v určeném pořadí (dáno způsobem řízení), kdy fází protéká budící stejnosměrný proud, je aktivováno magnetické pole v jednom páru pólů a nejbližší statorové zuby jsou přitahovány k pólu statoru.

Tento krokový motor může dosáhnout vysoké rychlosti, ale s relativně nízkým točivým momentem. V současnosti se tento typ motoru používá velmi zřídka. [1, 2, 6]

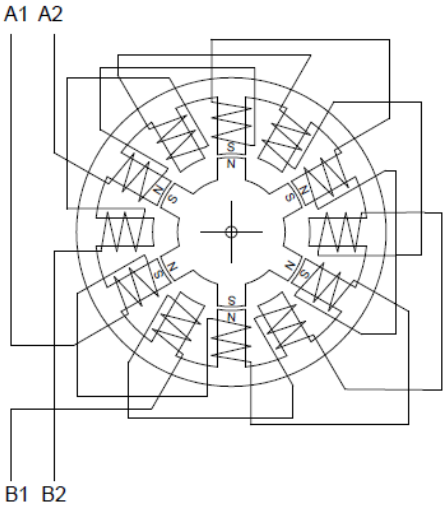


Obr. 2. Řez čtyřfázového krokového motoru s pasivním rotorem. [7]

1.1.2 Krokové motory s aktivním rotorem

U tohoto typu krokového motoru je rotor tvořen permanentním magnetem, který je radiálně polarizovaný. To znamená, že na obvodu rotoru se střídají severní a jižní pól, přičemž jejich počet je poloviční než počet pólů statoru, který je dále dělitelný čtyřmi. Statorové vinutí je navinuto dvoufázově, proto je nutné při spínání jednotlivých fází měnit směr protékajícího proudu (jedná se o bipolární řízení). Příkladem může být krokový motor, jehož řez je zobrazen na obrázku (Obr. 3). Pokud fáze tohoto motoru budeme budit podle dané posloupnosti, dojde k pootočení rotoru.

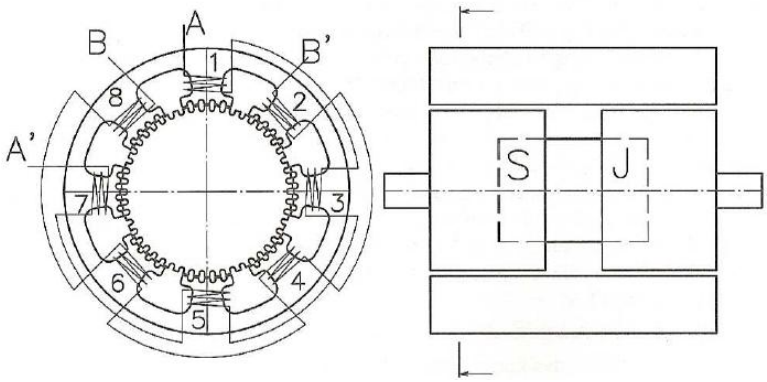
Vzhledem ke složitějšímu magnetickému obvodu jsou tyto krokové motory dražší. Díky malé časové konstantě vinutí lze s těmito motory dosáhnout vyšších rychlostí než u krokových motorů s pasivním rotorem. [6]



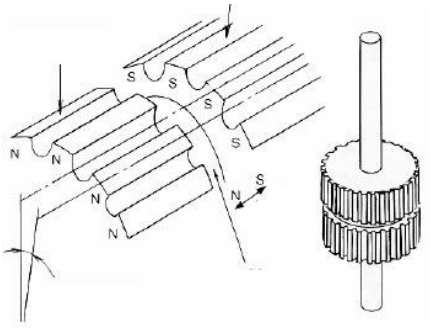
Obr. 3. Řez dvoufázového krokového motoru s radiálně polarizovaným magnetem. [7]

1.1.3 Hybridní krokové motory

Tento typ krokového motoru je v současnosti pravděpodobně nejpoužívanější typ. Hybridní krokový motor má rotor tvořen hřídelí z nemagnetického materiálu, na kterém jsou nalisovány dva pólové nástavce skládající se z plechů. Mezi těmito nástavci je vložen permanentní magnet, který je axiálně polarizovaný. Tímto získá každý pólový nástavec jinou magnetickou polaritu. Každý nástavec má po obvodu zuby, které svým počtem určují velikost kroku. Obvyklý počet zubů je 50, čemuž odpovídá velikost kroku 1,8°. Pro samotnou činnost je důležité, aby pólové nástavce byly vzájemně, v osovém směru, natočeny tak, že osy zubů jednoho nástavce jsou proti osám drážek druhého nástavce, neboli jsou natočeny o polovinu rotorové drážkové rozteče. Stator tvoří pólové nástavce s drážkami a dvoufázovým vinutím. Jednotlivé fáze vinutí jsou buzeny v předepsaném pořadí (dáno způsobem řízení), která vytváří točivé statorové magnetické pole, přičemž rotor se snaží toto magnetické pole sledovat tak, že se vždy nejbližší zuby rotoru nastaví do magneticky klidové polohy. [1, 2]



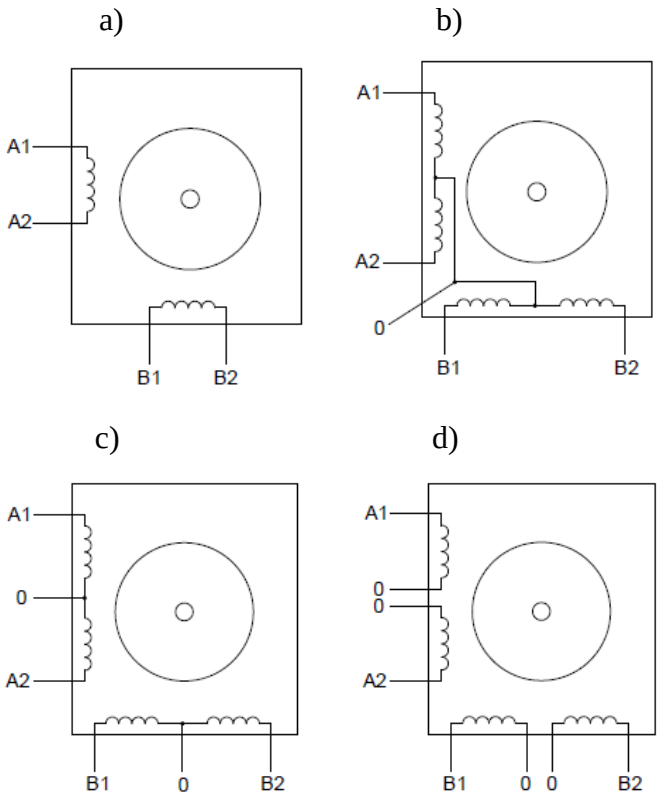
Obr. 4. Řez dvoufázového hybridního krokového motoru. [1]



Obr. 5. Pólové nástavce rotoru hybridního krokového motoru. [2]

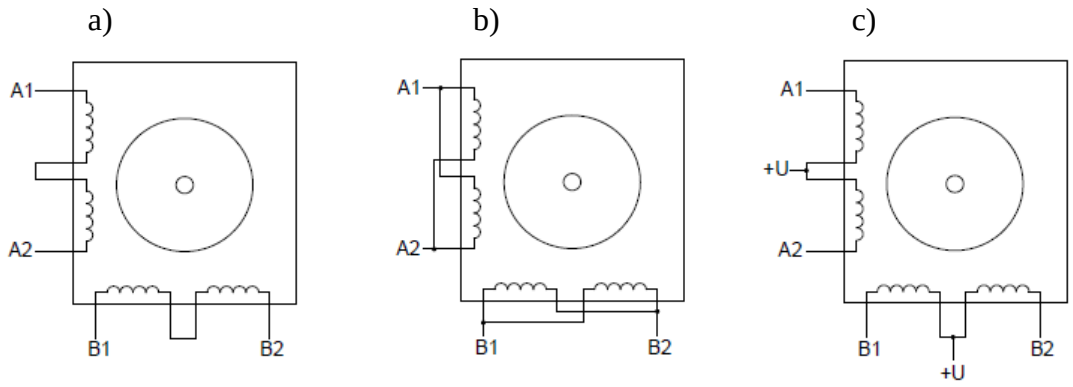
Podle způsobu a počtu vyvedených vodičů z motoru, rozdělujeme dvoufázové hybridní krokové motory: [1]

- 4 vodiče – vyvedeny začátky a konce vinutí
- 5 vodičů – navíc je vyveden společný střed obou vinutí
- 6 vodičů – navíc jsou vyvedeny samostatné středy vinutí
- 8 vodičů – každá polovina vinutí má vyvedeny oba konce



Obr. 6. Možné vyvedení vodičů u dvoufázového hybridního krokového motoru, a) 4 vodiče, b) 5 vodičů, c) 6 vodičů, d) 8 vodičů. [7]

Nejuniverzálnější možností je varianta s 8 vodiči, kdy jsou vyvedeny všechny části vinutí. Tato volba nám umožní si vybrat, zda krokový motor budeme budit bipolárně nebo unipolárně, přičemž bipolární řízení může být s paralelním nebo sériovým zapojení. Sériové zapojení má vyšší krouticí moment při nižších krokovacích frekvencích, oproti tomu paralelní zapojení má vyšší krouticí moment při vysokých krokovacích frekvencích. [1]



Obr. 7. Zapojení dvoufázového hybridního krokového motoru, a) bipolární sériové, b) bipolární paralelní, c) unipolární. [7]

1.1.4 Lineární krokové motory

Speciální typ krokového motoru, který poskytuje lineární pohyb, nikoliv rotační. Tyto krokové motory jsou založeny na podobných principech, jako výše popsané, a tvoří dvě skupiny: [1]

- Reakční lineární krokové motory
- Hybridní lineární krokové motory

Rotor je reprezentován tzv. běžcem, který se pohybuje nad statorem, který je tvořen drážkami a vinutím. U lineárního krokového motoru se nevytváří otáčivé magnetické pole, místo toho zde vzniká posuvné magnetické pole. [1]

1.2 Způsoby řízení krokových motorů

V této části budou rozebrány základní typy řízení krokových motorů. Způsobů řízení je hned několik a vždy závisí na tom, jakou zvolíme přesnost polohy, požadovaný krouticí moment a rychlost otáčení. Podle toho kolik fází budíme v jednotlivých krocích, rozlišujeme dva základní typy řízení, a to unipolární a bipolární řízení. [1]

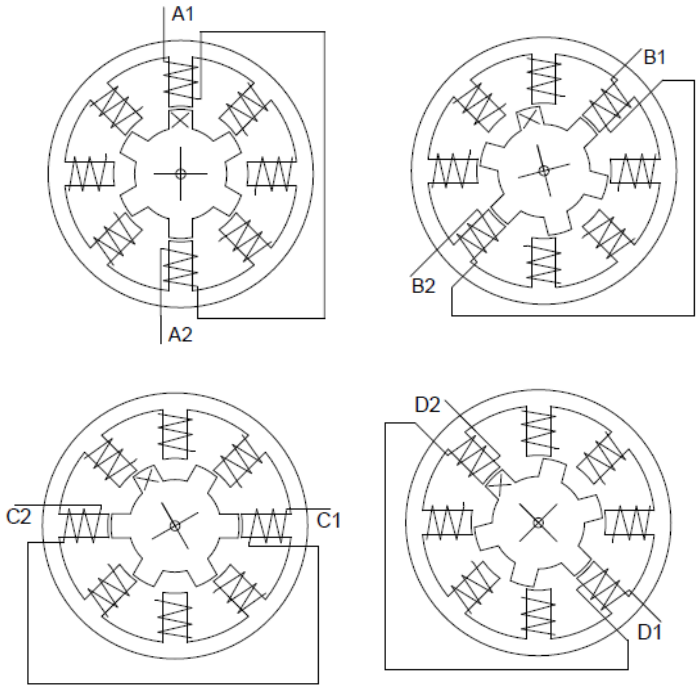
Při použití unipolárního řízení krokového motoru, prochází budící proud v jednom okamžiku pouze jednou cívkou. Krokové motory s tímto typem řízení mají sice menší odběr a zapojení řídící elektroniky je jednodušší, kdy stačí pouze jeden spínací prvek na každou cívku, ale oproti tomu mají nižší krouticí moment. Při bipolárním řízení krokového

motoru, prochází budící proud vždy dvěma protilehlými cívkami, které jsou zapojené tak, že navzájem mají opačně orientované magnetické pole. Oproti unipolárnímu řízení má takto řízený krokový motor větší odběr a zapojení řídící elektroniky je složitější, avšak máme k dispozici možnost většího krouticího momentu. [3]

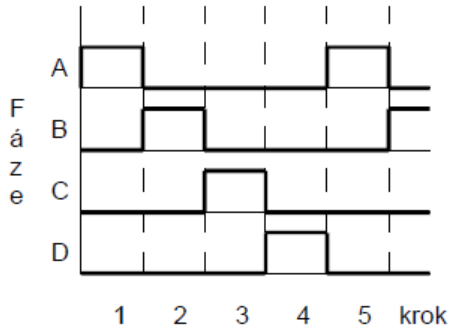
1.2.1 Čtyřtaktní řízení s magnetizací jedné fáze

Tento způsob řízení je považován za nejjednodušší způsob řízení krokového motoru. Je používán pro řízení čtyřfázového krokového motoru s pasivním rotorem nebo pro řízení hybridního krokového motoru v zapojení pro unipolární buzení fází.

Principiálně se jedná o to, že pokud je buzena fáze A (Obr. 8), dojde ke vzniku magnetického pole v ose pólových nástavců nesoucích vinutí fáze A. Následně dojde k pootočení rotoru, kdy nejbližší pár pólových nástavců rotoru se snaží dostat do místa největší intenzity magnetického pole, vytvořeným fází A. V další kroku je fáze A odpojena a je připojena fáze B. V tomto okamžiku se rotor otočí o ¼ zubové rozteče statoru do nové klidové polohy. Při dalším kroku, kdy se fáze B odpojí a připojí se fáze C, se motor otočí opět do klidové polohy. Totéž se bude opakovat i pro fázi D. Spínáním fází v sekvenci A-B-C-D-A (Obr. 9) docílíme toho, že se rotor otáčí proti směru hodinových ručiček. Pro změnu směru stačí změnit pořadí předchozí sekvence na tvar A-D-C-B-A. [1, 6]



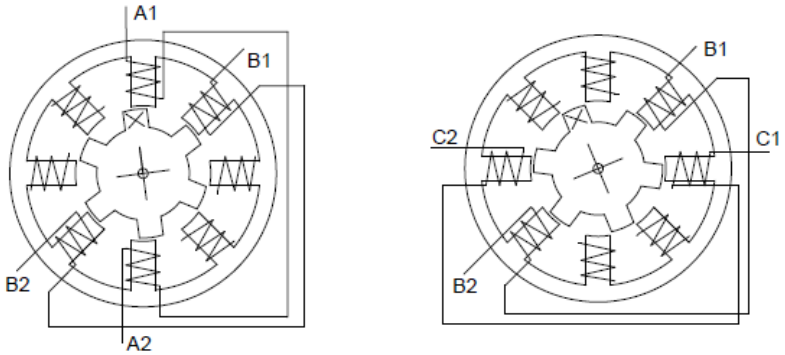
Obr. 8. Čtyřtaktní řízení čtyřfázového krokového motoru s magnetizací jedné fáze. [7]



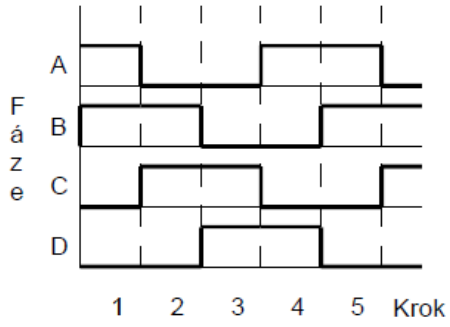
Obr. 9. Sekvence buzení jednotlivých fází. [7]

1.2.2 Čtyřtaktní řízení s magnetizací dvou fází

Při použití čtyřtaktního řízení s magnetizací dvou fází dochází k současnému buzení dvou sousedních fází. Oproti čtyřtaktnímu řízení s magnetizací jedné fáze je klidová poloha vychýlena a nachází se vždy mezi vybuzenými sousedními pólovými nástavci statoru. Velikost kroku i nadále zůstává stejná. Pro docílení otáčení rotoru, opět dochází k opakující se sekvenci buzení jednotlivých fází (Obr. 10) a to AB-BC-CD-DA (Obr. 11). Pro docílení opačného směru stačí změnit pořadí sekvence. [1, 6]



Obr. 10. Čtyřtaktní řízení čtyřfázového krokového motoru s magnetizací dvou fází (dva kroky). [7]

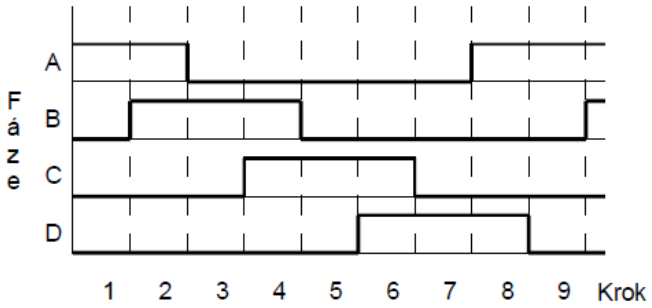


Obr. 11. Sekvence buzení jednotlivých fází. [7]

Oproti předešlému způsobu řízení je vyšší vazební moment a lze dosáhnout vyšších krokovacích frekvencí. [1]

1.2.3 Osmitaktní řízení

Osmitaktní řízení je vznikem sloučení čtyřtaktního řízení s magnetizací jedné a dvou fází. Při buzení jednotlivých fází se tudíž střídá buzení jedné a dvou sousedních fází a to v sekvenci A-AB-B-BC-C-CD-D-DA (Obr. 12) pro jeden směr a v opačném pořadí pro směr druhý. Rozdíl oproti předchozím řízením je, že docílíme poloviční kroku. Proto je často tento typ řízení označován jako řízení s poloviční krokem. Toto je považováno za výhodu a to i díky tomu, že není potřeba jakákoliv úprava stávajícího obvodu. Vzhledem k tomu, že v určitých okamžicích je buzena pouze jedna fáze, je provozní moment menší než v případě čtyřtaktního řízení s magnetizací dvou fází. [1, 6]



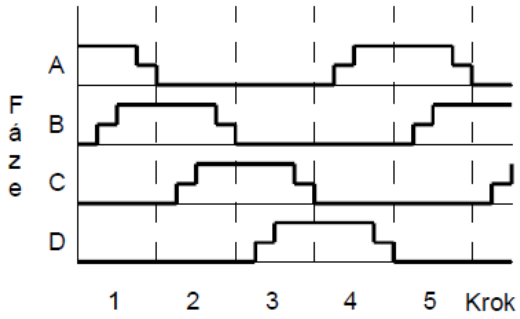
Obr. 12. Sekvence buzení jednotlivých fází. [7]

1.2.4 Mikrokrokování

Při určitých aplikacích je potřeba docílit velmi jemné rozlišitelnosti polohy krokového motoru. Úhel kroku krokového motoru je dán konstrukčně a jen výjimečně jde o úhel menší než 1°. Zmenšení kroku lze docílit zvětšením počtu fází nebo počtem rotorových zubů. Obojí je obtížné a nákladné na výrobu. Proto se využívá způsob řízení známy jako mikrokrokování, který zvyšuje počet kroků na otáčku, tím že je možné rozdělit jeden krok na několik stejně velkých mikrokroků, kdy jejich počet v dnešní době může být i 128 mikrokroků.

Princip vychází z metody magnetizace dvou fází, která byla dříve popsána. Oproti magnetizaci dvou fází, kde byly obě fáze buzeny stejnou velikostí proudu, tak zde jsou velikosti proudů různé. Vhodnou volbou velikostí těchto proudů v jednotlivých fázích můžeme dosáhnout libovolné klidové polohy mikrokroku mezi dvěma sousedními normálními kroky. K tomuto účelu je zapotřebí vícehladinový napájecí zdroj a kvalitní spínací obvody. V současné době jsou k dispozici speciální integrované obvody pro řízení krokových motorů, které nabízí možnost mikrokrokování, díky zabudovaným DA převodníkům.

Mikrokrokování slouží ke dvěma účelům. Prvním je, jak již bylo zmíněno, zastavení a držení polohy mezi celým a polovičním krokem. Druhým účelem je snížení mechanických rezonancí krokového motoru plynoucích z jeho poměrně nízkých tlumících vlastností. Tyto rezonance mohou způsobit ztrátu synchronizace rotoru, kdy důsledkem je ztráta kroku. Mimo toto může mikrokrokování eliminovat hluk. [1, 5, 6]



Obr. 13. Sekvence buzení jednotlivých fází. [7]

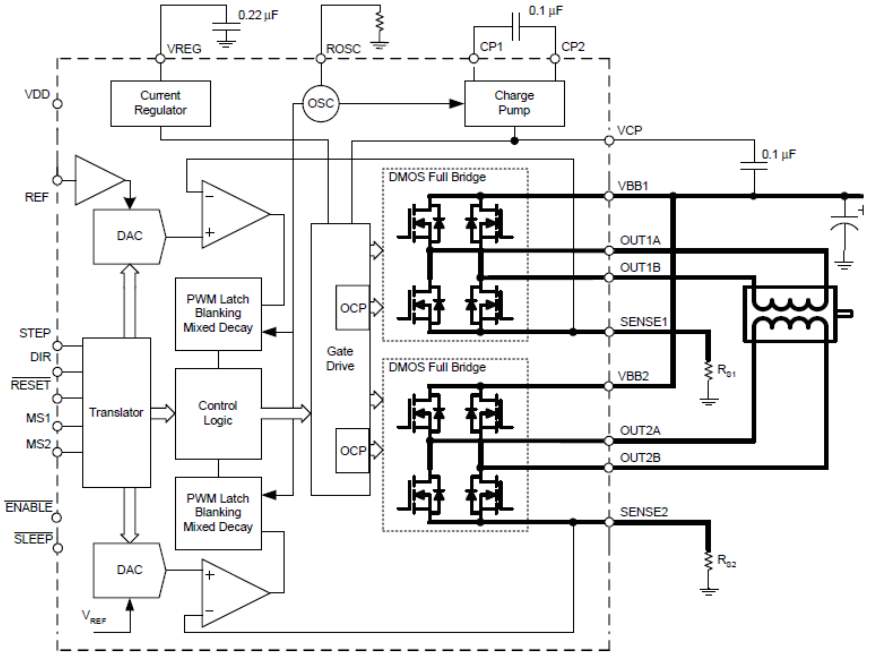
1.3 Integrované obvody pro řízení krokových motorů

V současné době je dostupná celá řada integrovaných obvodů určených pro řízení krokových motorů. A to od jednoduchých, které plní pouze funkci spínačů, přes regulátory proudu až po složité ovladače. [1]

1.3.1 A4982

A4982 je ovladač krokových motorů s vestavěnými tranzistory a překladačem pro jednodušší ovládání s plně funkčním mikrokrokováním, který nabízí společnost Allegro MicroSystems. Tento ovladač je navržen pro práci s dvoufázovým bipolárním krokovým motorem s celým, polovičním, čtvrtinovým nebo šestnáctinovým krokem. Výstupní napětí může být až 35 V a výstupní proud ±2 A. Napájecí napětí ovladače může být 3,3 V nebo 5 V. Ovladač také obsahuje nadproudovou ochranu, ochranu proti přehřátí, zámek při nízkém napětí a ochranu proti zkratům.

Překladač uvnitř ovladače je klíčem k jednoduché implementaci tohoto ovladače. Jednoduše stačí přivádět jednotlivé impulzy na vstupní pin STEP pro vytvoření jednoho mikrokroku. Proto je tento ovladač vhodný pro aplikace, kde nechceme zbytečně zatěžovat mikroprocesor. [22]

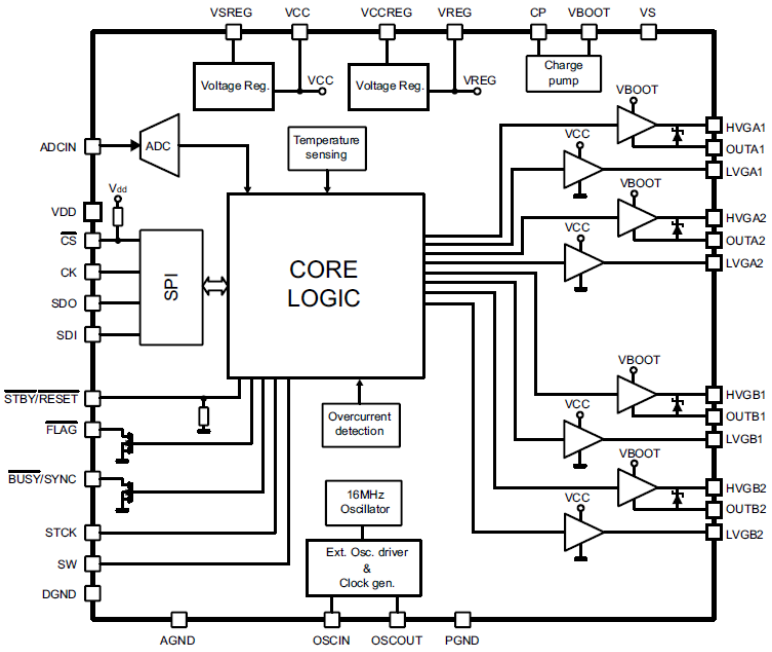


Obr. 14. Architektura ovladače A4982. [22]

1.3.2 L6480

Ovladač krokových motorů L6480 je moderní a plně integrované řešení pro ovládání dvoufázového bipolárního krokového motoru, který nabízí společnost ST Microelectronics. Výstupní napětí může být až 85 V. Napájecí napětí ovladače může být 3,3 V nebo 5 V.

Toto zařízení obsahuje Gate ovladač pro N-kanálové MOSFET tranzistory tvořící dvojí úplný můstek. Součástí jsou obvody pro nadproudovou ochranu. Díky unikátní technice řízení, lze dosáhnou mikrokrokování s velikostí kroku 1/128. Vnitřní digitální řídicí jádro generuje uživatelsky definovaný pohybový profil s akcelerací, rychlostí nebo cílovou pozicí, které je programováno jednoduše přes registry. Aplikační příkazy a data jsou posílána za využití SPI rozhraní. Kromě nadproudové ochrany, je zde implementována ochrana pro nízké napětí nebo teplotní ochrana. [23]



Obr. 15. Architektura ovladače L6480. [23]

1.3.3 DRV8821

Ovladač krokových motorů DRV8821 poskytuje řešení pro ovládání dvou krokových motorů s mikrokrokováním. Zařízení obsahuje obvody pro nezávislé ovládání dvou motorů za pomoci čtyř H-můstkových ovladačů a logiky pro možnost mikrokrokování. Obvodová část pro každý z motorů je tvořena N-kanálovými MOSFET tranzistory, které jsou konfigurovány jako H-můstek pro ovládání vinutí motorů. Výstupní napětí může být až 32 V a výstupní proud ±1,5 A. Napájecí napětí ovladače je 3,3 V.

Jednoduché krok/směr rozhraní umožňuje jednoduché ovládání řídicích obvodů. Je zde možnost konfigurace celého, poloviční, čtvrtečního a osminového kroku. V zařízení se také nachází obvody pro ochranu proti nadproudu, zkratům, nízkému napětí a přehřátí. [24]

2 ZACHYTÁVÁNÍ VIDEA

Při práci s kamerami se můžeme setkat s dvojím způsobem video výstupu. Výstup může být ve formě analogové nebo digitální.

V případě analogového videa máme na výběr několik způsobů přenosu video signálu. Prvním a nejběžnějším způsobem je použití jednoho signálu, který nese všechny složky videa. V tomto případě se jedná o kompozitní video signál označovaný také jako CVBS nebo jednoduše Video. Dalším způsobem je využití dvou signálů. Tato varianta se označuje jako S-Video. Jeden signál zde tvoří jasová složka a druhý barevná složka, které jsou v případě kompozitního video signálu spojené dohromady. Posledním způsobem je využití tří signálů. Zde hovoříme o komponentním videu a jednotlivé signály jsou tvořeny složkami barevného modelu RGB. Pokud se zaměříme na kamery a digitální fotoaparáty, tak v současné době se setkáme nejpravděpodobněji s použitím kompozitního video výstupu. Proto se v další části zaměřím právě na tento způsob přenosu analogového videa. [10]

Kromě analogového video výstupu se v dnešní době setkáme čím dál více s možností digitálního video výstupu a to ve formě HDMI. Mimo HDMI je možné video přenést i za pomoci USB rozhraní, ale za předpokladu, že námi zvolená kamera nebo fotoaparát má funkci Live Stream, která umožňuje právě přímý přenos snímaného obrazu přes rozhraní USB. Tato funkce bohužel není příliš rozšířená a pouze u některých kamer se s ní můžeme setkat.

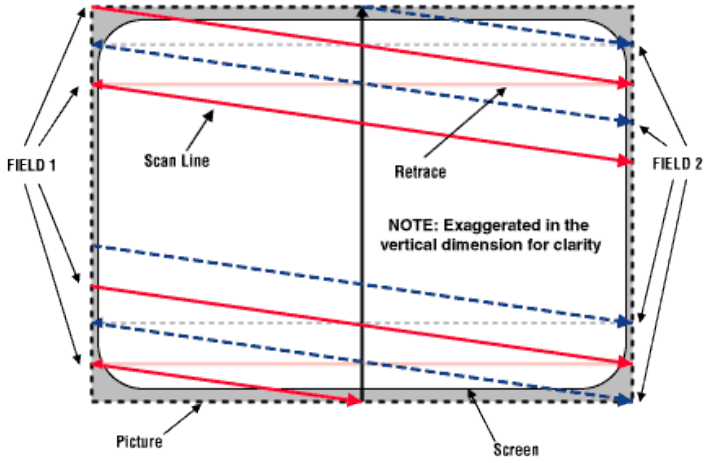
2.1 Způsob skenování obrazu

Dříve než nerozebereme jednotlivé způsoby přenosu videa, je nutné si uvědomit, že existují dva způsoby skenování obrazu, které jsou vzájemně nekompatibilní, proto je potřeba konverze při převodu mezi sebou.

2.1.1 Prokládané skenování

Při prokládaném skenování jde o rozdělení obrazového rámce na liché a sudé řádky. Tím nám vzniknou dvě pole z jednoho obrazového rámce. Prokládaný obraz je na displej vykreslen ve dvou krocích, kdy se vykreslí první pole a v druhém kroku se vykreslí

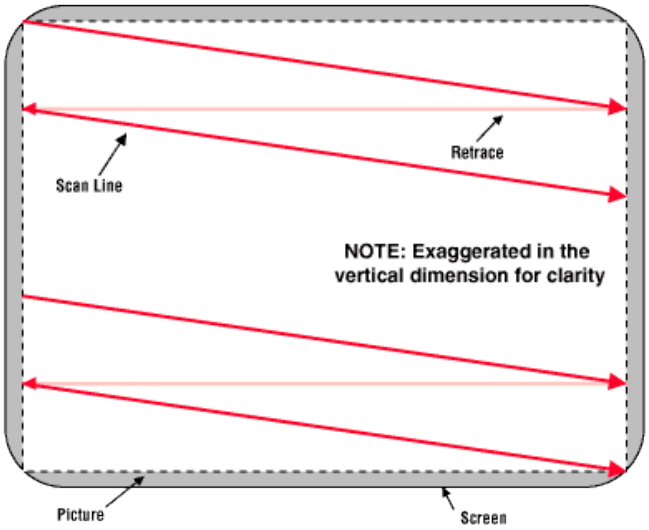
druhé pole mezi jednotlivé řádky prvního pole. Pokud tedy přenášíme prokládaný obraz, tak přenášíme vždy jen půlku obrazu, tím dochází k redukci na polovinu. Všechny analogové formáty používají prokládané skenování. Avšak u prokládaného skenování vzniká problém, kdy kvůli prodlevě mezi lichými a sudými řádky vzniká efekt rozmazání, kdy jedna polovina obrazu se obnovila s pohybujícím se objektem, zatímco druhá polovina čeká na obnovení. Toto je obzvláště zřetelné při zastavení videa a analyzování jednoho obrazového rámce. [10]



Obr. 16. Prokládané skenování. [10]

2.1.2 Progresivní skenování

Progresivní skenování, také označováno jako neprokládané skenování, vykresluje na displej současně všechny řádky obrazu. Neboli nedochází k rozdělení obrazu na poloviny. Oproti prokládanému skenování zde nedochází k efektu blikání obrazovky. Pokud video zastavíme a budeme analyzovat jeden obrazový rámec, tak oproti prokládanému skenování je obraz podrobnější. [10]



Obr. 17. Progresivní skenování. [10]

2.2 Kompozitní video

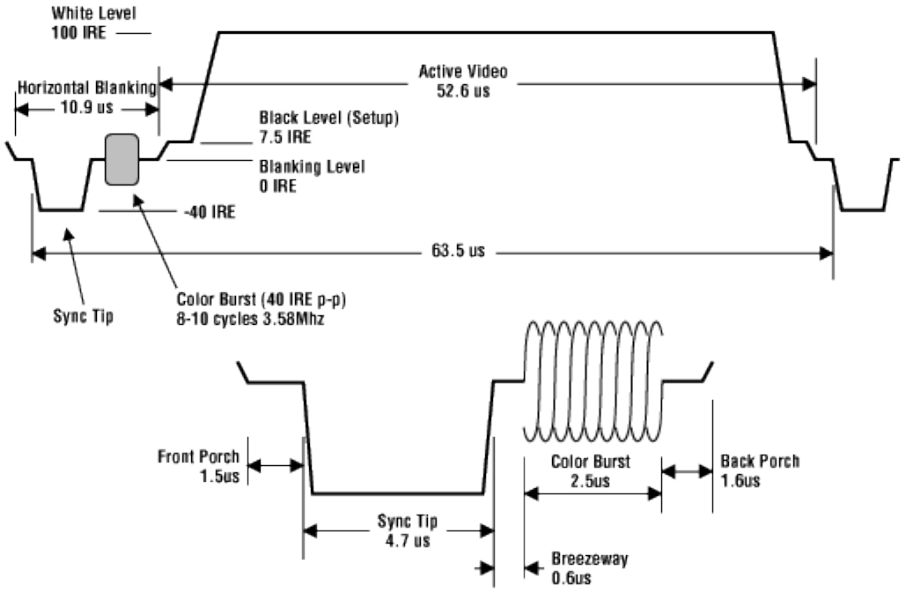
Jedná se o nejběžnější způsob pro přenos analogového videa, kdy pro přenos barevného videa (bez zvuku) nám stačí pouze jeden signál. Pro připojení se používá kabel s RCA konektorem.

Obr. 18. Konektor RCA.

2.2.1 Struktura kompozitního video signálu

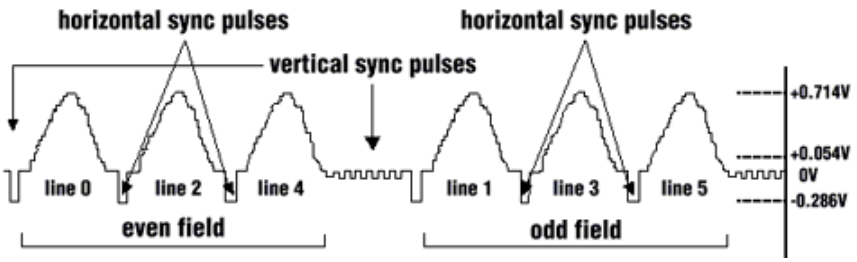
Jak již bylo řečeno, kompozitní video tvoří jeden signál, který je složen ze tří hlavních složek. Tyto složky tvoří složka, která nese informaci o intenzitě jasu (luma), další tvoří složka s informací o barvě (chroma). Poslední složkou je synchronizační signál. Pokud budeme uvažovat pouze černobílé video, tak signál tohoto videa bude složen pouze z jasové složky a synchronizačního signálu. Tento signál je často označován jako Y signál. V případě barevného videa, je přidána barevná složka označována jako C signál.

Při zaměření se na jeden horizontální obrazový řádek, je tento řádek složen z několika částí a to horizontálního synchronizačního signálu, Back-Porch, oblasti aktivních pixelů a Front-Porch. Avšak je důležité mít na paměti, že části, mimo část aktivních pixelů, jsou umístěny v čase tak, že nejsou vidět na displeji. [8, 10]



Obr. 19. Struktura jednoho horizontálního řádku (standard NTSC). [10]

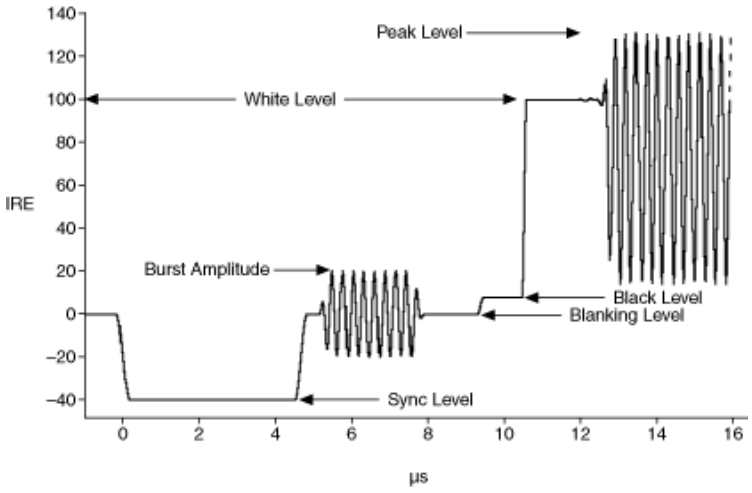
Horizontální synchronizační signál (HSYNC) je na začátku každého nového obrazového řádku. Za tímto signálem se nachází Back-Porch. V této části se nachází Color-Burst, který poskytuje informace, jako jsou reference fáze a amplitudy, pro dekódování barevného obsahu signálu. Dalším aspektem video signálu je vertikální synchronizační pulz (VSYNC). Jedná se o sérii pulzů, které se vyskytují mezi jednotlivými obrazovými rámci. Příklad synchronizačních pulzů je zobrazen na obrázku (Obr. 20), kde je pro jednoduchost zobrazen obrazový rámec s šesti řádky. [8, 10]



Obr. 20. Synchronizační impulzy (prokládané skenování). [8]

Zakódování informace o barvě do signálu je provedeno za pomoci kvadraturní amplitudové modulace (QAM). Modulace musí být zpětně dekódovatelná, proto se vkládá referenční signál označován jako Color-Burst, který se umisťuje na začátek každého obrazového řádku hned za horizontální synchronizační signál. Toto platí pro standardy PAL a NTSC. Pro standard SECAM se využívá frekvenční modulace za použití dvou sub-nosných frekvencí. Mimo to nepotřebuje Color-Burst signál.

Se signálem souvisí také video úrovně a rozsahy pro různé části video signálu. Jednotka používaná pro definování video úrovní je IRE (Institude of Radio Engineers). Zatemňovací úroveň je rovna hodnotě 0 IRE a bílá úroveň odpovídá hodnotě +100 IRE. Zatemňovací úrovni, jež je referenční úroveň pro video signál (většinou 0 V), odpovídá černá úroveň pro standard PAL a SECAM. V případě standardu NTSC je černá úroveň posunuta na hodnotu +7,5 IRE. Analogový kompozitní video signál je definován jako zdroj napětí s výstupní impedancí 75 Ω, kde mezi synchronizační a bílou úrovní je normálně 1 $V_{pk\text{-}pk}$. [8, 10]



Obr. 21. Znázornění video úrovní. [8]

Rozdílné hodnoty jednotlivých úrovní pro jednotlivé standardy jsou uvedeny v tabulce (Tab. 1).

Tab. 1. Rozdílné hodnoty jednotlivých úrovní pro jednotlivé standardy.

Standard	Sync level [IRE]	Blanking level [IRE]	Black level [IRE]	White level [IRE]	Peak level [IRE]	Burst amplitude [IRE]
NTSC	-40	0	+7,5	+100	+120	20,0
PAL	-43	0	0	+100	+133	21,5
SECAM	-43	0	0	+100	+130	N/A

2.2.2 Standardy analogového videa

Existuje několik běžně používaných standardů pro formáty analogového videa, které se většinou odlišují tím, že kromě technických parametrů se používají v odlišných státech světa. Takovým příkladem může být standard NTSC, který je používán v Severní Americe nebo Japonsku. Oproti tomu, v Evropě je běžný formát PAL, který vznikl po NTSC a je jeho vylepšením a nabízí lepší obrazovou kvalitu. Můžeme se také setkat se standardem SECAM, který je používán ve Francii. Je potřeba si uvědomit, že existuje kolem 15 různých sub-formátů těchto třech zmíněných. Většinou tyto standardy nejsou navzájem kompatibilní. Avšak mají základy na stejném způsobu skenování obrazu a reprezentaci barvy pomocí fázové modulace. Rozdíly jsou ve specifických skenovacích frekvencích, počtu skenovaných řádků a technice modulace barvy. Srovnání parametrů mezi jednotlivými standardy je uvedeno v tabulce (Tab. 2). [8, 10]

Tab. 2. Srovnání parametrů jednotlivých standardů.

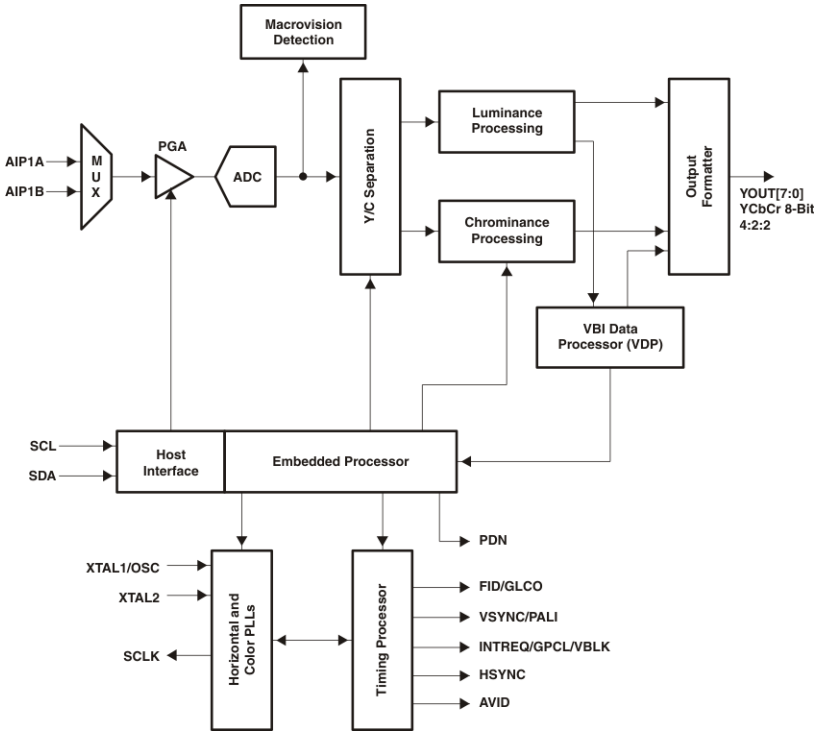
Standard	FPS [rámec/s]	Celkový počet řádků [px]	Aktivní počet řádků [px]	Horizontální rozlišení [px]	Doba jednoho řádku [µs]	Doba části aktivních pixelů [µs]
NTSC	29,97	525	480	320 – 650	63,55	52,2
PAL	25,00	625	576	320 – 720	64,00	52,0
SECAM	25,00	625	576	320 – 720	64,00	52,0

2.2.3 Integrované obvody pro zachytávání kompozitního videa

V současné době nabízí několik výrobců integrované obvody pro převod analogového videa na digitální formu.

TVP5150AM1

TVP5150AM1 je NTSC/PAL/SECAM video převodník od společnosti Texas Instruments. Buď lze připojit dva kompozitní vstupy, nebo jeden S-Video vstup. Signál je vzorkován 9bitovým AD převodníkem s analogovým procesorem. Pro všechny standardy je připojen pouze jeden krystal o frekvenci 14,31818 MHz. Toto zařízení převádí NTSC, PAL a SECAM video signál na 8bitový digitální výstup ve formátu YCbCr 4:2:2. Kromě digitálního video výstupu, převodník generuje synchronizační a hodinový signál. Převodník TVP5150AM1 má optimalizovanou architekturu pro snížení spotřeby a je napájen napětím 3,3 V a 1,8 V. Proto se jedná o zařízení vhodné pro aplikace fungující na bateriích. Programování registrů převodníku je zajištěno I^2C rozhraním a lze tak nastavit různé video charakteristiky jakým je odstín, jas, ostrost nebo sytost. [25]

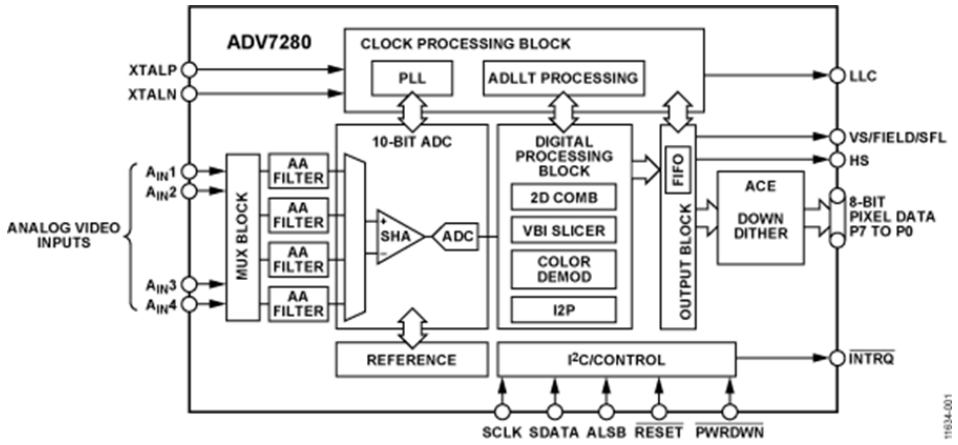


Obr. 22. Architektura převodníku TVP5150AM1. [25]

ADV7280

ADV7280 je NTSC/PAL/SECAM video převodník od společnosti Analog Devices. Buď lze připojit čtyři kompozitní vstupy, nebo dva S-Video vstupy. Signál je vzorkován 10bitovým AD převodníkem. Pro všechny standardy je připojen pouze jeden krystal o frekvenci 28,636336 MHz. Toto zařízení převádí NTSC, PAL a SECAM video signál na 8bitový digitální výstup ve formátu YCbCr 4:2:2, přičemž je zajištěna automatická detekce

vstupního video standardu. Kromě digitálního video výstupu, převodník generuje synchronizační a hodinový signál. Převodník ADV7280 je napájen napětím 3,3 V a 1,8 V. Programování registrů převodníku je zajištěno I^2C rozhraním a lze tak nastavit opět různé video charakteristiky. [26]



Obr. 23. Architektura převodníkuADV7280. [26]

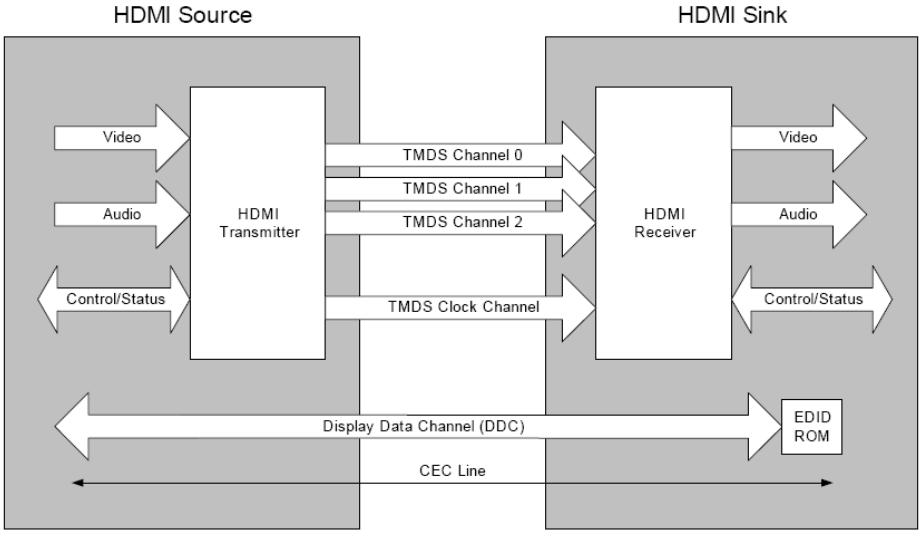
2.3 HDMI

HDMI je v dnešní době považováno za jeden z nejlepších způsobů pro přenos digitálního obrazu. HDMI přenáší jak nekomprimované video ve vysokém rozlišení, tak i vícekanálový digitální zvuk, a to vše pomocí jednoho kabelu. Navíc může posílat základní řídicí kódy ze zařízení na zařízení. HDMI je také zpětně kompatibilní s rozhraním DVI až na to, že neumožňuje přenos zvuku. Tudíž lze použít zařízení s DVI výstupem a připojit k zařízení s HDMI vstupem. Opačná kompatibilita avšak nemusí být, protože zvuk je přenášen na stejných kabelech jako obraz a zařízení s DVI vstupem si nemusí poradit s těmito daty.

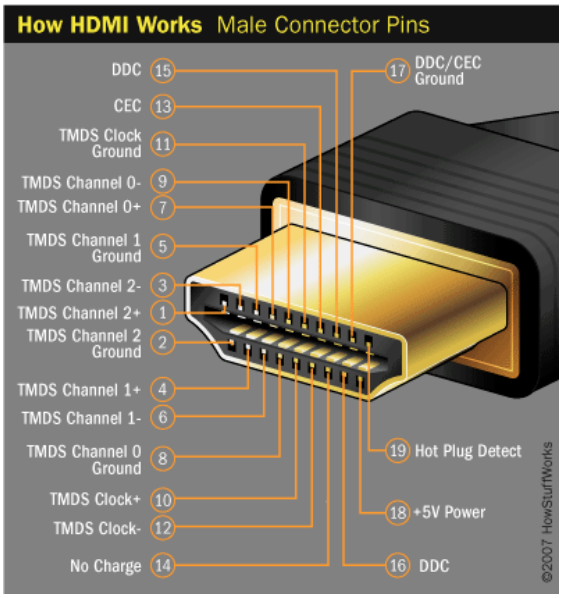
HDMI implementuje mechanizmus nazývaný HDCP, který slouží jako ochrana proti kopírování digitálního obsahu. Je proto důležité při návrhu HDMI přijímače si uvědomit, z jakého zdroje budeme zachytávat obrazová data, jinak si budeme muset pořídit klíče pro dešifrování těchto dat.

Co se týče celkové architektury HDMI, tak tu tvoří zdroj vysílaní (Source), který obsahuje HDMI vysílač, a přijímací zařízení (Sink), které obsahuje HDMI přijímač. Samotné propojení je vytvořeno kabelem s HDMI konektorem typu A, B, C nebo D. Konektor typu A je tvořen 19 piny, oproti tomu typ B má 29 pinů a používá se především pro přenos videa s velkým rozlišením. Konektory typu C a D mají stejný počet pinů jako

typ A, ale mají menší rozměr a jsou určeny pro přenosná zařízení. Samotná data jsou poté přenášena pomocí protokolu TMDS. HDMI využívá sériového diferenciálního přenosu, to znamená, že se využívá dvou vodičů nesoucí signál a současně inverzní. V přijímacím zařízení se měří rozdíl mezi těmito signály. Proto pro samotná data je vyhrazeno 6 vodičů, dva vodiče pro jeden kanál. Další dva vodiče jsou určeny pro časování přenosu. Mezi další vodiče patří CEC, umožňující ovládání připojených zařízení, DDC, které tvoří I^2C rozhraní pro přenos informací o zařízení a je také použito pro autentizaci u HDCP, Hot Plug Detect, určený pro vytvoření události, že nastalo například odpojení kabelu. Nakonec je v kabelu ještě zdroj napětí 5V a zemnící vodič pro CEC a DDC. Nesmíme ještě opomenout vodiče stínění TMDS kanálů, které stíní kroucené páry vodičů a minimalizují tak EMI emise, tyto vodiče jsou připojeny na zem na obou koncích kabelu. [11, 13, 14]



Obr. 24. Struktura HDMI. [11]



Obr. 25. HDMI konektor. [13]

2.3.1 TMDS protokol

HDMI využívá pro přenos dat protokol TMDS (Transition Minimized Differential Signaling). Jedná se o minimalizační technologii pro robustní přenosy digitálních dat za minimalizace EMI (Electromagnetic Interference). Za normálních okolností, kdy se data posílají vysokou rychlostí, narůstá obtížnost zpětného obnovení dat, dalším faktorem může být také délka kabelu. TMDS obsahuje techniky, které elegantně řeší tyto komunikační problémy, takže přijímací zařízení může přesně rekonstruovat data zasílaná ze zdroje.

V TMDS, se vysílající zařízení snaží signál zakódovat tak, aby se redukoval počet změn mezi jedničkami a nulami. Jakmile je signál zakódován, přidá se značka, jaký typ transformace byl použit pro minimalizaci, jedná se o funkci XOR nebo XNOR. Přijímací zařízení dekóduje tato zjednodušená data a obnoví původní digitální signál. Jednou z výhod tohoto postupu je, aby přijímací zařízení jasně vymezilo, kde každý bajt dat začíná a končí, čímž zajistil správný příjem dat. Tato unikátní technika v TMDS je to co umožňuje HDMI dosáhnout takových rychlostí.

TMDS protokol konvertuje v první fázi 8bitové vstupní obrazová data do 10bitového streamu. Ve druhé fázi je prvních 8 bitů invertováno postupně na sudý počet nul a jedniček k zajištění průměrné hodnoty stejnosměrného napětí. Pro každou 8bitovou barevnou složku RGB je vyčleněna jedna dvojice pinů zajišťující sériový diferenciální přenos. Další pár pinů je vyčleněn pro zajištění správného časování přenosu (Clock).

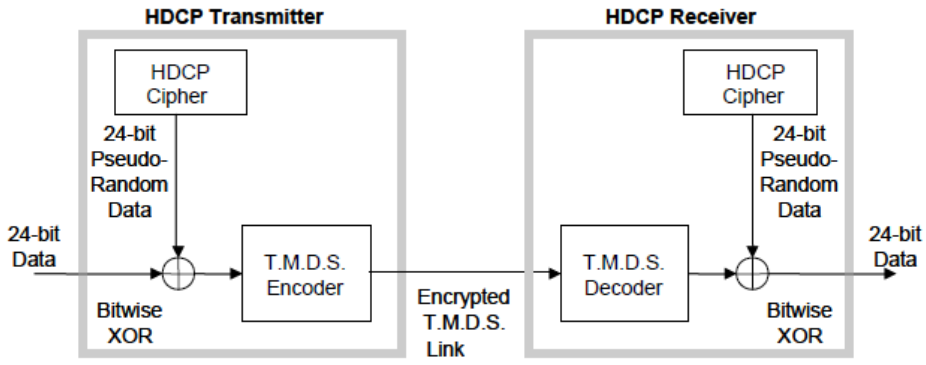
Stejně jako u analogového videa, tak i zde jsou synchronizační signály, přesněji jde o horizontální a vertikální synchronizační signály. Tyto signály jsou zakódovány v TMDS kanálu číslo 0. Celkem jde o 4 kombinace a ty jsou zobrazeny v tabulce níže (Tab. 3). Avšak oproti analogovému videu, je zde nejdůležitější pouze první ze synchronizačních pulzů. Zbylý prostor, který odpovídá zatemňovací části, byl použit pro přenos zvukových dat. [12, 14]

Tab. 3. Zakódování synchronizačních signálů.

HSYNC	VSYNC	10bitové kódové slovo
0	0	1101010100
1	0	0010101011
0	1	0101010100
1	1	1010101011

2.3.2 HDCP

HDCP (High-bandwidth Digital Content Protection) je navrženo k ochraně přenášeného audiovizuálního obsahu pomocí HDMI a popisuje mechanizmy pro jeho ochranu, jako je autentizace přijímače a šifrování audiovizuálního obsahu. Autorizované zařízení je pak takové, které má povolený přístup k HDCP obsahu. Vysílací zařízení testuje úspěšným dokončením autentizačního protokolu složeného z autentizace a výměny klíčů, zda připojený přijímač je autorizované zařízení. Komunikace mezi oběma zařízeními je za pomoci I^2C rozhraní, které je v případě HDMI specifikováno jako DDC (Digital Display Channel). HDCP šifrování je aplikováno na vstup TMDS kódování a dešifrování je aplikováno na výstupu TMDS dekódovače. HDCP šifrování je složeno z funkce XOR, která je aplikována na jednotlivé bity HDCP obsahu a pseudonáhodného proudu generovaného HDCP šifrou. HDCP šifra generuje nový 128bitový proud klíče pro každý pátý 24bitový pixel. [15]



Obr. 26. Schéma HDCP šifrování a dešifrování. [15]

Zatímco HDCP je volitelné ve specifikaci HDMI, tak téměř každé zařízení, které je navrženo tak, aby vysílalo nebo přijímalo chráněný obsah, tak je začleněno do HDCP. Mezi tato zařízení patří TV, DVD přehrávače atd. Jedinými zařízeními, která nejsou regulárně zařazena do HDCP jsou ta, která nejsou určena pro vysílání a příjem chráněného obsahu a patří mezi ně videokamery a digitální fotoaparáty. Proto je nutné si uvědomit, z jakého zařízení budeme audiovizuální obsah přijímat. [16]

2.3.3 Kódování pixelů

Kódování obrazových pixelů u HDMI zahrnuje RGB nebo YCbCr barevné modely.

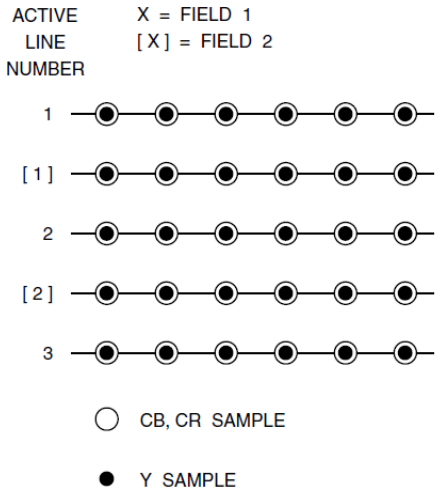
RGB

RGB barevný model je široce používaný napříč počítačovou grafikou. Červená, zelená a modrá jsou tři primární aditivní barvy. RGB barevný model je nejvíce převládající volba pro počítačovou grafiku, protože barevné displeje používají červenou, zelenou a modrou barvu pro vytvoření barvy. Většinou se jedná o variantu RGB 4:4:4, kdy jsou všechny složky vzorkovány stejnou frekvencí při 24 bitech na pixel. Avšak v určitých aplikacích není RGB barevný model tak efektivní. Když budeme potřebovat upravit intenzitu nebo barvu určitého pixelu, musíme vzít všechny tři složky RGB a intenzitu a barvu přepočítat. Proto se z tohoto a jiných dalších důvodů u mnoha video standardů používá jasový signál a další dva signály odpovídající barevným diferencím. Mezi nejznámější patří YCbCr barevný prostor. [17]

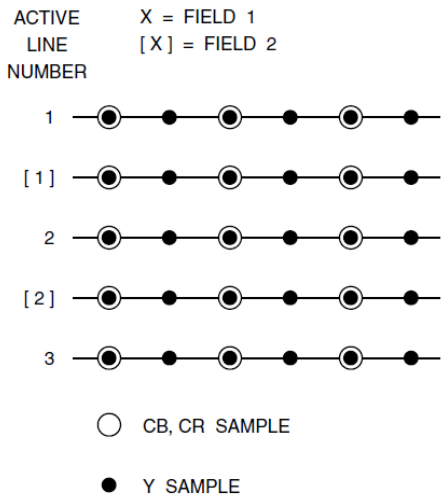
YCbCr

YCbCr je barevný prostor, kde Y je 8bitová složka definující intenzitu jasu a je v rozsahu 16 – 235. Cb a Cr jsou definovány v rozsahu 16 – 240 a tvoří barevnou složku (chrominanci). Cb je rozdíl mezi jasovou složkou a modrou barvou a Cr je rozdíl mezi jasovou složkou a červenou barvou. Studie ukázaly, že lidské oko je citlivější na jas, ale ne tak citlivé na barvu (chrominanci). Z tohoto důvodu se využívá kompresní techniky sub-vzorkování chromatické složky. Existuje několik verzí sub-vzorkování, jako je 4:4:4, 4:2:2, 4:1:1 nebo 4:2:0.

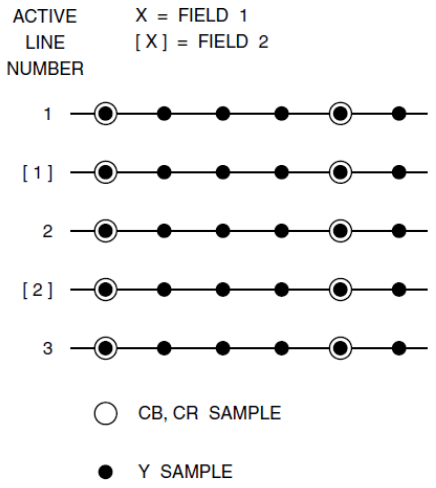
Pokud si rozebereme jednotlivé sub-vzorkovací varianty, tak 4:4:4 znamená, že všechny složky jsou vzorkovány stejnou frekvencí a tudíž každý vzorek má hodnotu Y, Cb a Cr. Když si vezmeme variantu 4:2:2, tak zde jsou složky Cb a Cr vzorkovány poloviční frekvencí. Proto pro dva horizontální vzorky Y odpovídá jeden Cb a Cr vzorek. 4:1:1 vzorkuje složky Cb a Cr čtvrtinovou frekvencí oproti složce Y, proto čtyřem horizontálním vzorkům Y odpovídá jeden vzorek Cb a Cr. Další možná varianta se od variant 4:2:2 a 4:1:1 liší v tom, že zde nedochází k redukci pouze v horizontálním směru, ale je zde redukce i ve směru vertikálním a je označována jako 4:2:0. Tady jeden vzorek Cb a Cr odpovídá čtyřem vzorkům Y, které jsou rozmístěny jak v horizontálním, tak i vertikálním směru. Pokud budeme chtít zobrazit na displeji formáty, u kterých byla provedena redukce, vždy je tento formát převeden na 4:4:4 a použitím interpolace jsou zbylé vzorky Cb a Cr dopočítány. U všech variant se počítá s 24 bity na pixel. [17]



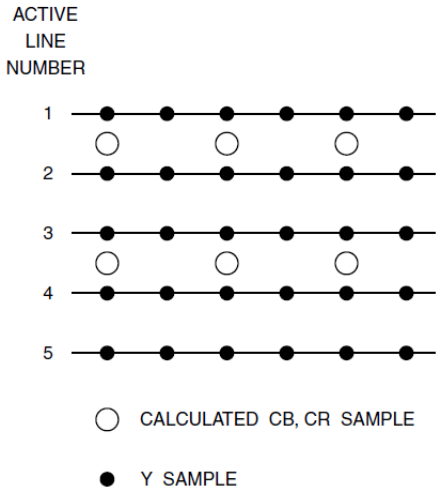
Obr. 27. YCbCr 4:4:4 (prokládané skenování). [17]



Obr. 28. YCbCr 4:2:2 (prokládané skenování). [17]



Obr. 29. YCbCr 4:1:1 (prokládané skenování). [17]



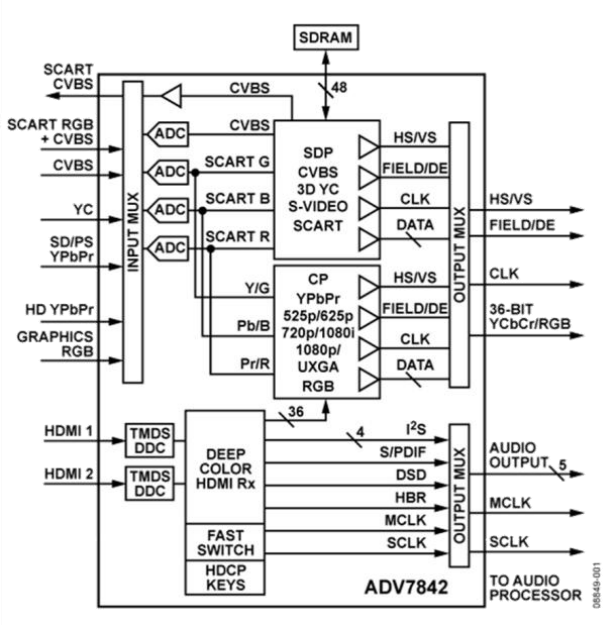
Obr. 30. YCbCr 4:2:0 (progresivní skenování). [17]

2.3.4 Integrované obvody pro zachytávání HDMI

Stejně jako v případě integrovaných obvodů pro převod analogového videa, jsou k dispozici integrované obvody pro příjem HDMI. Jsou k dispozici také varianty v kombinaci společně s analogovým video vstupem.

ADV7842

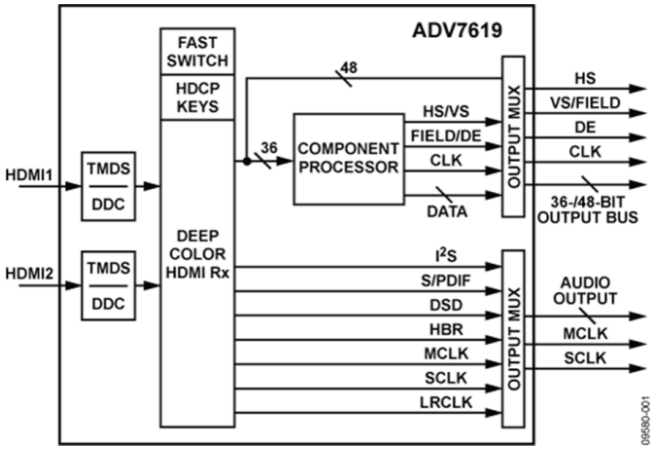
ADV7842 je dvojí HDMI 1.4 přijímač a více formátový analogový převodník, který je nabízen od společnosti Analog Devices. S tímto přijímačem lze přijímat všechny HDTV formáty až do rozlišení 1080p včetně 3D TV formátu. Přijímač disponuje 12 analogovými vstupy, které podporují NTSC, PAL a SECAM standardy. Analogový signál je vzorkován za pomoci 12bitového ADC. Součástí HDMI přijímače je hardwarově implementována ochrana proti kopírování HDCP. Díky funkci Xpressview lze přepínat mezi oběma HDMI vstupy během pouhé jedné sekundy. Tento přijímač také disponuje audio výstupem, který je dekódován z HDMI signálu. Každý HDMI port má 5 V detekci a Hot Plug detekci. Tento přijímač má až 36bitový digitální výstup, kde výstupní data jsou ve formě RGB 4:4:4 nebo YCbCr 4:2:2. Celé zařízení je dodatečně programovatelné za pomoci I^2C rozhraní a napájení je 3,3 V a 1,8 V. [27]



Obr. 31. Architektura HDMI přijímače ADV7842. [27]

ADV7619

ADV7619 je vysoce kvalitní dvojí HDMI 1.4 přijímač od společnosti Analog Devices. Přijímač je k dispozici v profesionální (bez HDCP klíčů) nebo v komerční verzi. Je umožněna podpora všech definovaných HDTV formátů až do rozlišení 1080p při 36bitové barevné hloubce. Integrován je CEC ovladač, který umožňuje ovládání připojených zařízení. K dispozici je také funkce Xpressview umožňující rychlé přepínání mezi oběma HDMI vstupy. Každý HDMI port má 5 V a Hot Plug detekci. Digitální výstup tvoří 24bitová sběrnice, kde data jsou posílána ve formě RGB 4:4:4 nebo YCbCr 4:2:2. Celé zařízení je dodatečně programovatelné za pomoci I^2C rozhraní a napájení je 3,3 V a 1,8 V. [28]

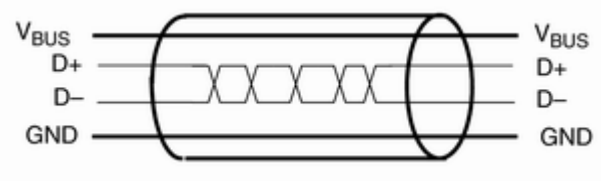


Obr. 32. Architektura HDMI přijímače ADV7619. [28]

2.4 USB

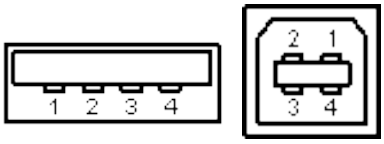
USB je univerzální sériová sběrnice, která využívá dva signálové vodiče s využitím diferenciálního kódování, podobně jako jeden kanál u TDMS při HDMI. Na fyzické úrovni jsou jednotlivá komunikující zařízení propojena systémem point to point, neboli jeden kabel propojuje pouze dva sousední uzly. Uzly tvoří samotná koncová zařízení, rozbočovače a jeden řídicí uzel nazývaný jako Host. Na logické úrovni se jedná o skutečnou sběrnici, protože řídicí uzel může komunikovat s jakýmkoliv zařízením, bez ohledu na to, v jakém místě stromové struktury se toto zařízení nachází.

Pro propojení dvou komunikujících uzlů se používá kabel se čtyřmi vodiči. Dva vodiče V_{BUS} a GND slouží pro napájení připojených periferních zařízení. Další dva, označované jako D+ a D-, jsou určeny pro sériový diferenciální přenos. Přenášené bity jsou kódovány metodou NRZI a díky vkládaným bitům není potřeba hodinového signálu. [18]



Obr. 33. Struktura USB kabelu. [18]

K propojení se používají USB konektory typu A a B, přičemž existují i jejich menší verze, jako je mini a mikro, určené pro malá přenosná zařízení.



Obr. 34. USB konektor typu A (vlevo) a typu B (vpravo). [18]

Pokud budeme chtít zachytávat video data z videokamery nebo z fotoaparátu, je potřeba si nejdříve ověřit, zda toto zařízení umožňuje funkci pro přímé vysílání obrazu přes USB, která se nazývá Live Stream. Pokud bychom zjistili, že touto funkcí naše zařízení nedisponuje a my bychom stále požadovali zachytávání obrazových dat za pomoci USB, tak je zde jedna možnost. Touto možností je využití zařízení označovaného jako Grabber, který disponuje vstupy, jakými mohou být vstup analogového videa nebo HDMI. Na USB výstupu jsou poté obrazová data vysílána ve formátech RGB nebo YCbCr, které jsou popsány v kapitole o HDMI u kódování pixelů. Kromě tohoto jsou kladeny také požadavky na straně příjemce, pokud budeme chtít data zachytávat za pomoci mikropočítače. Současné mikropočítače již disponují možností USB rozhraní, avšak je potřeba si uvědomit, že musíme vytvořit zařízení, které bude zastávat úlohu řídicího uzlu, neboli Host.

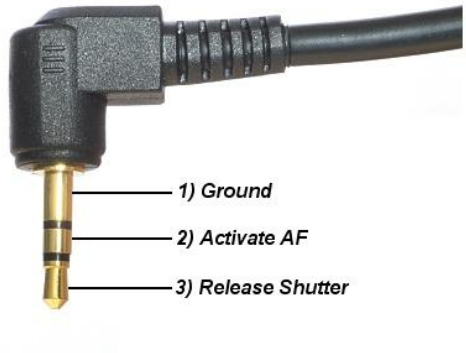
3 VZDÁLENÉ OVLÁDÁNÍ KAMER

Mnoho fotoaparátů má řadu možností pro vzdálené ovládání, které si ani neuvědomujeme, že existují. Existuje mnoho důvodu pro použití vzdáleného ovládání. Pokud potřebujeme dělat foto-pasti, autoportréty nebo fotografie z různých míst, tak můžeme využít drátových nebo bezdrátových možností, jak ovládat časování, ostření a spoušť fotoaparátu. Výhodou tohoto použití je také to, že celý proces focení je méně náchylný na chyby, které často vznikají při stlačení tlačítka spouště fotoaparátu.

3.1 Drátové ovládání

Existují dvě možnosti jak vzdáleně ovládat fotoaparát za pomoci připojeného kabelu.

První možností je použití tří-pinového konektoru, který je umístěn přímo na fotoaparátu a je určen pro připojení Triggeru. Ve většině případů se jedná o konektor typu 2,5 mm JACK. Tři piny představují zem, automatické ostření a uvolnění uzávěrky. Celý proces spouště je, že v první fázi dojde k propojení země a vodiče automatického ostření dohromady. V druhé fází dojde ke spojení země a vodiče uvolnění uzávěrky dohromady. Není nezbytné, aby bylo aktivováno automatické ostření před uvolněním uzávěrky. [19]



Obr. 35. 2,5mm Jack konektor. [19]

Druhou možností je použití USB rozhraní. V moderních fotoaparátech může být USB rozhraní použito nejen pro přenos fotek, ale také použito pro posílání řídicích příkazů fotoaparátu. Je často možné poslat příkaz pro automatické ostření, uvolnění spouště, nastavení zařízení a nastavení časové spouště. Je potřeba si uvědomit, že různé fotoaparáty

od různých výrobců mají různé příkazy a možnosti ovládání. Struktura protokolu je popsána v protokolu MTP (Media Transfer Protocol), který je známý pod jménem jeho předchůdce PTP (Picture Transfer Protocol). [20]

3.2 Bezdrátové ovládání

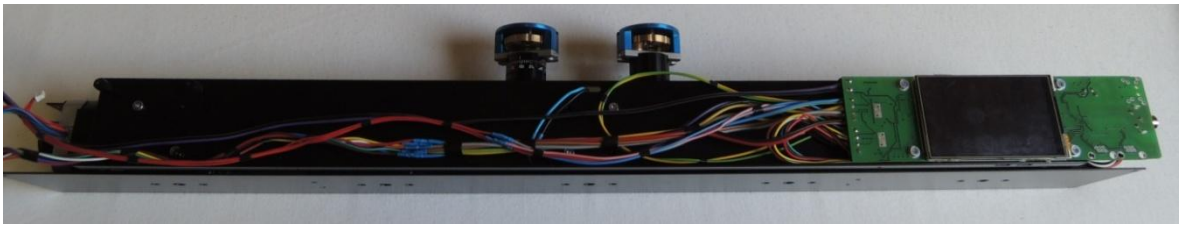
Fotoaparát lze vzdáleně ovládat i bezdrátově. Pro tento účel lze využít optického ovládání, přesněji využít infračervené světlo. Některé fotoaparáty mají již vestavěný infračervený přijímač. Pro různé typy fotoaparátu se opět liší sekvence signálu a jejich časování. Většinou se jedná jen o automatické ostření a uvolnění závěrky, ale v některých případech je možné nastavit i časovou spoušť.

Dalším způsobem může být použití RF (Radio Frequency). Jedná se o rádiové vlny, avšak ne každý fotoaparát má rádiový přijímač. Tento způsob je velmi vhodný pro venkovní použití i na větší vzdálenosti. [21]

II. PRAKTICKÁ ČÁST

4 KONSTRUKCE MANIPULÁTORU

Pro účely stereovize byla vytvořena konstrukce, která má dva nástavce pro uchycení dvou videokamer nebo fotoaparátů. Tyto nástavce jsou přes převody a hřídele napojeny na dva krokové motory, které umožňují pohyb těchto nástavců. Jeden krokový motor slouží pro horizontální pohyb, kdy se nástavce pohybují směrem k sobě nebo směrem od sebe. Druhý krokový motor je pro rotační pohyb kolem svislé osy nástavce. Tímto lze dosáhnout toho, že připevněná zařízení se mohou natáčet směrem k sobě nebo od sebe. Oba nástavce jsou navzájem provázány, tudíž nemusíme řešit o jaký úhel je každý nástavec pootočen, protože je každý natočen o stejný úhel. Pokud na nástavec připevníme kamery, může se stát, že jedna z těchto kamer nebude mít stejný úhel jako druhá. Pro tento účel je jeden z nástavců opatřen prstencem pro úpravu směru natočení. Pro samotné ovládání krokových motorů a zachytávání videa z obou kamer byla ke konstrukci připevněna řídicí deska s potřebnou elektronikou a dotykovým LCD displejem.



Obr. 36. Odkytovaná konstrukce manipulátoru.

Obr. 37. Detail dvou nástavců.

4.1 Pohon

Pro pohon byly použity krokové motory od společnosti Microcon. Jde o hybridní dvoufázové krokové motory řady SX v přírubě NEMA17. Tyto krokové motory se

vyznačují vysokými momenty při zachování malých rozměrů. Velikost kroku je 1,8° s možností dalšího elektronického zmenšování. Přesněji se jedná o typ SX17-1005D, který má vyvedenou hřídel na obě strany. Tento způsob vyvedení hřídele nám umožňuje jednak ovládat převody a hřídel pro pohyb nástavců na jedné straně a také připojit zařízení označované jako inkrementální čítač na druhou stranu. [29]

Technické parametry krokového motoru: [29]

	Sériové / paralelní zapojení
Statický moment [Nm]:	0,5
Jmenovitý proud [A]:	1 / 2
Indukčnost [mH]:	10,8 / 2,7
Odpor [Ω]:	5,40 / 1,35
Zbytkový moment [Nm]:	0,022
Moment setrvačnosti rotoru [gcm^2]:	54
Hmotnost [kg]:	0,3

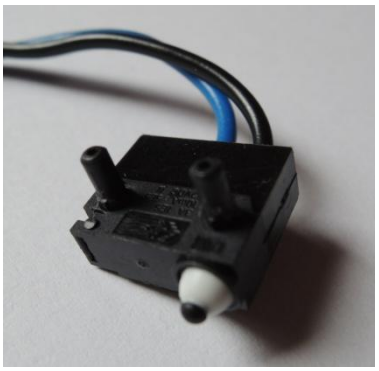
4.2 Zabezpečovací prvky

Abychom byli schopni sledovat a mít pod kontrolou chod obou krokových motorů a bezproblémový pohyb nástavců, byla konstrukce opatřena bezpečnostními prvky.

Jako první takový prvek můžeme jmenovat zarážky. Funkci zarážky zde zaujímají dva mikrospínače, které se starají o to, aby mikropočítač dostal signál o dojetí nástavců na samotné kraje při horizontálním pohybu, následkem čehož mohl vypnout krokový motor a nedošlo k poškození. Pro rotační pohyb, kolem svislé osy, byl použit optosenzor s rotační vačkou s definovaným úhlem rotace. Tímto je zajištěno, že kamery budou snímat pouze vymezený prostor před sebou a nebudou se moci otáčet kolem celé osy.

Dalším zabezpečovacím prvkem je optický inkrementální snímač. Tento snímač převádí rotační pohyb na dvoukanálový digitální výstup. Díky dvěma výstupním kanálům jsme také schopni rozeznat směr otáčení. Primární funkcí tohoto snímače v naší konstrukci

je zajistit detekci zaseknutí při pohybu nástavců, přičemž dojde k vypnutí příslušného krokového motoru, aby do něj nadále nešel proud a nedošlo k poškození.



Obr. 38. Mikrospínač (zarážka).

5 FUNKCE ŘÍDICÍ DESKY

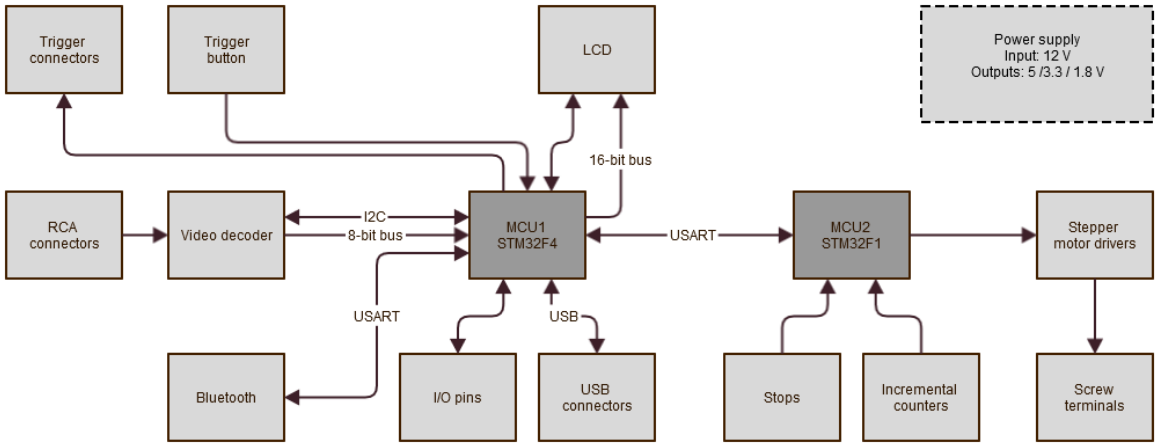
Hlavní funkcí řídicí desky by mělo být ovládání dvou krokových motorů. Za tímto účelem je potřeba navrhnou obvody, které na základě příkazů z MCU umožní spínat jednotlivé fáze krokového motoru. Abychom mohli tyto příkazy vyvolat, je potřeba vytvořit rozhraní pro uživatele. Pro tento a jiné další účely by měl být použit dotykový LCD displej. Další funkcí, kterou by měla řídicí deska schopna dělat, je zachytávání dvou kompozitních video signálů, které budou vysílány z obou kamer, s následným zobrazením na displeji jako náhled. Tímto by mělo být umožněno podat uživateli informaci o snímané expozici. Pro případ vzdáleného ovládání automatického ostření a spouště kamer je potřeba, aby řídicí deska měla možnost dvoufázového tlačítka pro vyvolání těchto funkcí, přičemž je potřeba poskytnou pro připojení dvou kabelů konektor typu 3,5 mm JACK. Dalšími doplňujícími funkcemi by měla být možnost komunikovat s tímto zařízením za pomoci Bluetooth, možnost USB konektivity a doplňující konektory pro případné připojení senzorů.

6 HARDWARE ŘÍDICÍ DESKY

Před samotným návrhem schémat a pozdějším návrhem plošného spoje, bylo potřeba shrnout potřebné informace. Mezi tyto informace patřila volba součástek, komunikace s jednotlivými obvody, omezení na typ součástek nebo omezení na velikost výsledného plošného spoje.

Co se týče omezení typu součástek, nebylo možné použít součástky s průchozí montáží, protože z jedné strany řídicí desky bude umístěn displej. Z tohoto důvodu a důvodu minimalizace byly použity výhradně SMD součástky. Pro návrh plošného spoje platila omezení pro výslednou šířku a výšku z důvodu umístění do omezujícího prostoru celé konstrukce manipulátoru.

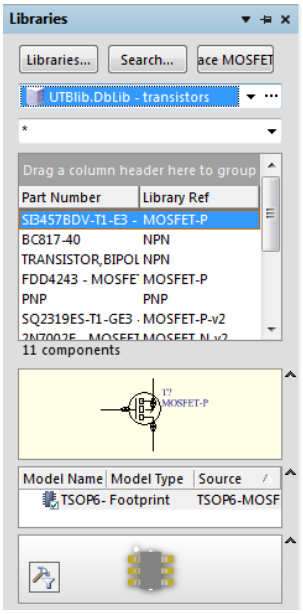
Po pečlivém výběru všech součástek jsem teprve mohl začít vytvářet schémata a navrhnou plošný spoj.



Obr. 39. Blokové schéma hardwaru řídicí desky.

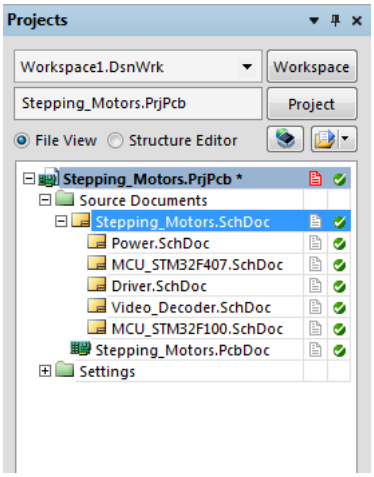
6.1 Program pro návrh plošných spojů

Pro návrh schémat zapojení a následnou tvorbu plošného spoje byl zvolen program Altium Designer 13. Tento program nabízí mnoho funkcí pro kompletní návrh plošných spojů. Základním prvkem je knihovna součástek. Celá knihovna tvoří obsáhlý nástroj pro vytváření strukturovaného obsahu, kde lze součástky dělit do různých skupin. Každá součástka má přiřazenou schematickou značku, footprint a pro možnost zobrazit si navrhnutou desku plošného spoje ve 3D, je možné k součástce přiřadit i 3D model. Pro každou součástku lze navíc vytvořit kompletní popis dané součástky.



Obr. 40. Panel knihovny součástek.

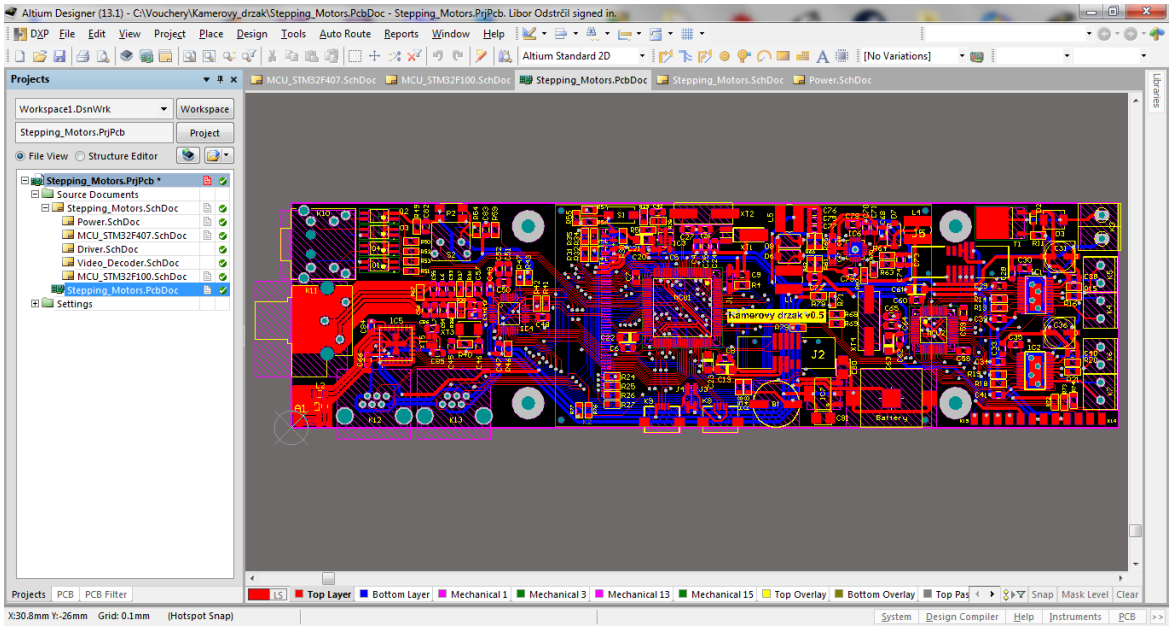
Před samotným návrhem bylo potřeba vytvořit nový PCB projekt. Pokud byl projekt vytvořen, bylo možné do něj následně snadno vkládat nové soubory pro tvorbu schémat nebo pro kreslení PCB.



Obr. 41. Panel souborů projektu.

Mezi některé z mnoha funkcí programu, které stojí za zmínku, patří kontrola uživatelsky nastavených pravidel nebo kontrola zda máme propojeny všechny piny. Mezi uživatelská pravidla může patřit například velikost mezer mezi vodiči a součástkami nebo velikosti prokovů či děr. Hlavně kontrola nepropojených spojů je velmi užitečná funkce a to v případě pokud se jedná o složitější projekt.

Navrhnutý plošný spoj lze vyexportovat do mnoha různých formátů, které podporuje mnoho výrobců plošných spojů. Kromě toho lze celý projekt uložit do formátu PDF. Potom je v jednom souboru k dispozici veškerá dokumentace zahrnující schémata, soupis součástek, náhled desky a jednotlivé její vrstvy.



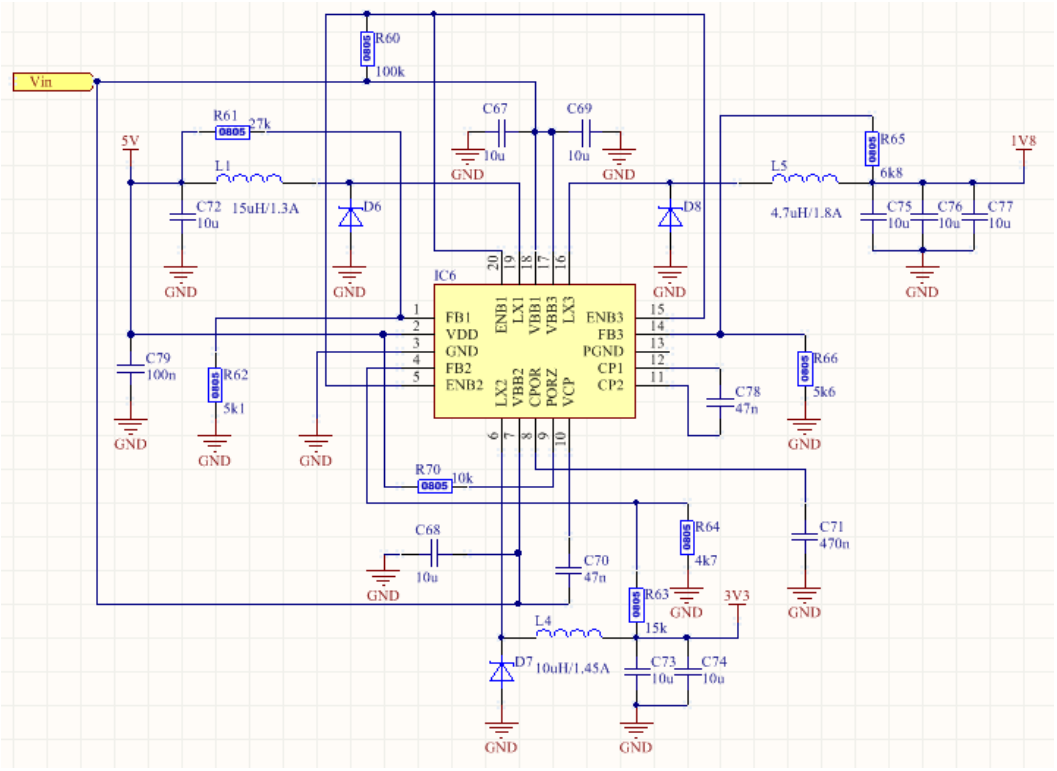
Obr. 42. Kreslení plošného spoje v programu Altium Designer 13.

6.2 Napájení

Jako každé elektronické zařízení, tak i toto zařízení potřebuje zdroj napájení. Protože bylo potřeba kvalitního zdroj, který umožňuje ze vstupního napětí vytvořit tři různá výstupní napětí, zvolil jsem spínaný regulátor A4490EES-T od společnosti Allegro Microsystems.

Tento regulátor umožňuje přivést na vstup napětí v rozmezí 4,5 – 34 V. V mém případě se uvažuje s napětím 12 V a to z důvodu využití tohoto napětí pro krokové motory. Vstupní napětí může být sníženo na tři různá napětí. Podle zvolených součástek jsou výstupní napětí nastavena na 5 V / 3,3 V / 1,8 V. Je potřeba uvést, že napětí pro krokové motory je použito přímo ze vstupního napětí, nikoliv z tohoto spínaného regulátoru.

Co se týče samotného regulátoru, tento regulátor má ochranu proti přetížení, nízkému vstupnímu napětí a proti přehřátí. Regulátor má pouzdro typu QFN s 20 piny a chladící podložkou pro lepší odvod tepla. [30]



Obr. 43. Schéma zapojení spínaného zdroje.

6.3 Mikropočítač (MCU)

Tato řídicí deska je osazena dvěma mikropočítači. První mikropočítač (MCU1) je STM32F407VE s 32bitovým procesorem ARM Cortex-M4 běžícím na frekvenci 168 MHz s DSP funkcemi a FPU. Tento MCU má FLASH paměť o velikosti 512 KB a paměť SRAM o velikosti 192 KB. Dalšími funkcemi je přítomnost FSMC paměťového ovladače, čtyř rozhraní typu USART a dvou typu UART, dvou rozhraní USB s funkcí OTG, kamerového rozhraní DCMI, 16 AD a dvou DA převodníků a 82 vstupně výstupních pinů. Napájení je zajištěno pomocí napětí 3,3 V. MCU je nabízeno v pouzdru LQFP se 100 piny. [31]

Primárními funkcemi tohoto MCU1 je ovládání dotykového LCD displeje, zpracování obrazových dat a ovládání video převodníku. Dalšími funkcemi jsou komunikace s MCU2, ovládání triggeru, komunikace přes Bluetooth, obsluha dvou USB rozhraní.

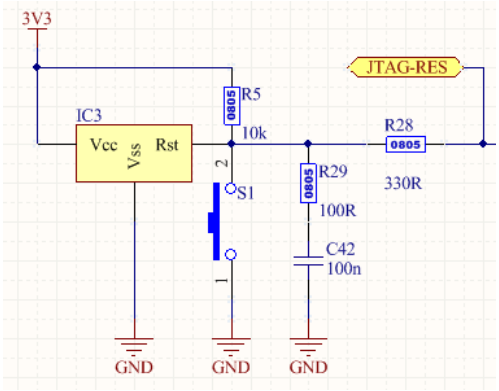
Druhý mikropočítač (MCU2) je STM32F100C4 s 32bitovým procesorem ARM Cortex-M3 běžícím na frekvenci 24 MHz. Tento MCU má FLASH paměť o velikosti 16 KB a paměť SRAM o velikosti 4 KB. Dalšími funkcemi je přítomnost dvou rozhraní

typu USART, jednoho AD a dvou DA převodníků a 37 vstupně výstupních pinů. Napájení je zajištěno pomocí napětí 3,3 V. MCU je nabízeno v pouzdru LQFP se 48 piny. [32]

Primární funkcí tohoto MCU2 je ovládání krokových motorů. Dalšími funkcemi jsou komunikace s MCU1, detekce zarážek a čtení inkrementálních čítačů pro detekci zaseknutí.

Důvodem použití dvou MCU bylo zapříčiněno nedostatkem pinů při použití jednoho MCU se 100 piny. Tato velikost pouzdra byla limitující, protože větší MCU bych na daný rozměr plošného spoje neumístil. Další důvod, který se posléze vyskytl, bylo snížení zátěže u MCU1, který bude převážně zaměstnávat zpracování obrazu a vykreslování na displej.

Součástí obou MCU je resetovací obvod. Tento obvod slouží pro účely resetování MCU1, kdy z příčiny manuálního vyvolání resetu za pomoci tlačítka nebo z důvodu nižšího napětí než 2,63 V, které hlídá integrovaný obvod STM1001RWX6F, je na resetovací pin MCU1 přivedena napěťová úroveň logické 0 po určitou dobu. MCU2 není na tento obvod napojen a je za pomoci pull-down rezistoru držen v resetu, až po samotné inicializaci MCU1 vyvolá tento MCU1 napěťovou úroveň logické 1 na resetovacím pinu MCU2 a tím je uvolněn z resetu a proběhne následně jeho inicializace.



Obr. 44. Schéma resetovacího obvodu.

6.4 LCD displej

Za účelem ovládání a zobrazení náhledu videa z obou upevněných kamer je řídicí deska osazena LCD displejem s rezistivní dotykovou vrstvou. Displej má úhlopříčku 3,2 palců a má rozlišení 240 x 400 pixelů. Maximální počet barev, který je displej schopen zobrazit je 262 tisíc barev. Za pomoci dvou pinů BS0 a BS1 displeje, lze volit šířku

sběrnice pro přenos obrazových dat, která může být 8, 16 nebo 18 bitů. V mém případě byla zvolena šířka sběrnice 16 bitů. Kromě pinů samotné sběrnice jsou zde také řídicí piny označované jako RS, CS, WR a RD. Tyto piny slouží pro signalizaci zápisu, čtení, a zda jde o přenos adresy nebo samotných dat. Displej má integrovaný ovladač HX8352-A. Tento ovladač lze přes registry programovat a spravovat tak celý displej, jako je způsob příjmu obrazových dat, způsob vykreslování a rotace displeje. Protože tento ovladač obsahuje paměť typu GRAM, kde se uchovávají data celého obrazu, lze využít pro ovládání displeje funkci MCU1 a to FSMC (Flexible Static Memory Controller). Pokud se připojí datová sběrnice a řídicí piny na příslušné piny MCU1, které podporují funkci FSMC, lze do paměti displeje přistupovat, jako kdyby byla součástí MCU1, přičemž se nemusím starat o řídicí signály displeje. Celý displej je napájen napětím 3,3 V. [35]

Součástí displeje je rezistivní dotyková vrstva. Vývody této vrstvy jsou připojeny na piny MCU1, které umožňují funkci AD převodníku.

Obr. 45. LCD displej.

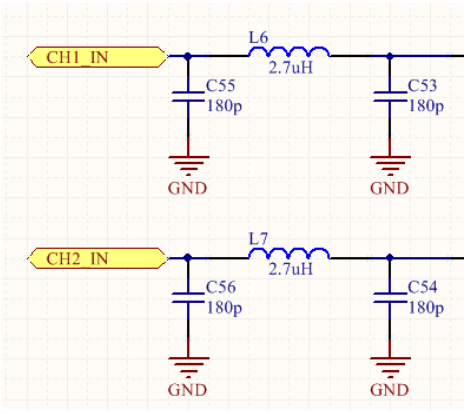
6.5 Video převodník

Pro zachytávání a převod kompozitního videa na digitální formát jsem použil video převodník TVP5150AM1. Jedná se o převodník, který dokáže zachytávat hned dva kompozitní video signály a přepínat se mezi nimi. Je zde podpora video standardů NTSC, PAL a SECAM. Kompozitní signál je vzorkován 9bitovým AD převodníkem a digitální forma videa je ve formátu YCbCr 4:2:2 posílána přes 8bitovou sběrnici do MCU1. Současně s datovou sběrnicí jsou k MCU1 připojeny piny vysílající synchronizační signály a hodinový signál. Použití převodníku vyžaduje přítomnost externího krystalu o frekvenci 14,31818 MHz. Další doporučenou přídavnou části je přidání antialiasingového filtru. Komunikace s převodníkem a jeho programování je zajištěno pomocí I^2C rozhraní. Díky

tomu lze kompletně upravovat nastavení převodníků, vybírat vstupní signál a případně upravovat video charakteristiky, jakými jsou odstín, jas, ostrost nebo sytost. [25]

MCU1 disponuje funkcí nazývanou DCMI (Digital Camera Interface). Jedná se o vestavěné kamerové rozhraní, které může být propojeno s kamerovými moduly nebo CMOS senzory. Je možné využít 8 až 14bitovou sběrnici. Přicházející data lze zpracovávat ve formátu YCbCr 4:2:2 nebo ve formátu RGB 565. V mém případě budu pracovat s formátem YCbCr 4:2:2, protože tento formát nabízí mnou zvolený video převodník. Abych mohl využívat této funkce, bylo potřeba správně připojit datové vodiče sběrnice a synchronizační signály na příslušné piny MCU1, které jsou určeny pro funkci DCMI.

Pro samotné napájení bylo potřeba na převodník přivést napětí 1,8 V a 3,3 V.



Obr. 46. Schéma antialiasingového filtru.

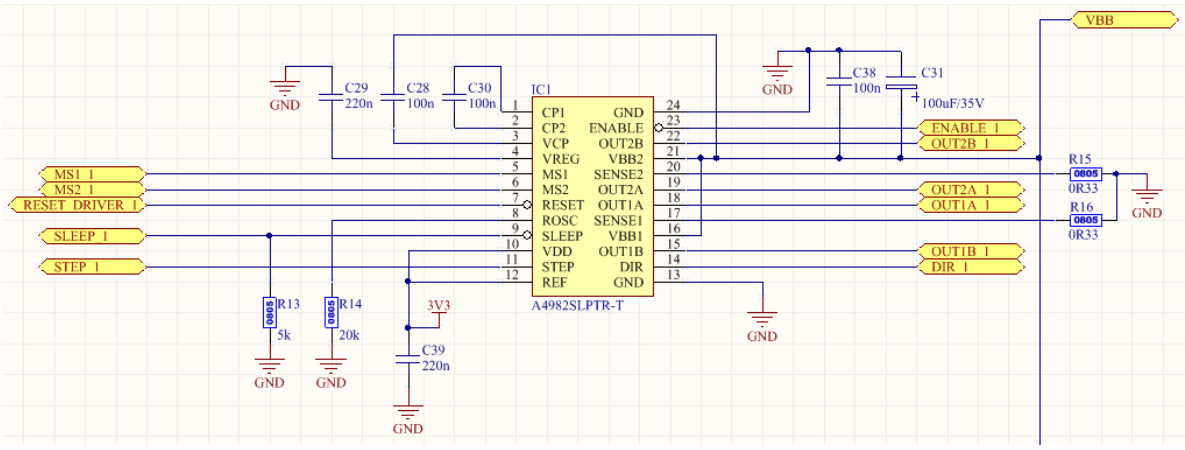
Pro připojení kompozitního kabelu k řídicí desce je použit konektor RCA s dvojím vstupem.

Obr. 47. RCA konektor.

6.6 Ovladač krokových motorů

Za účelem ovládat krokové motory, byly na řídicí desku umístěny dva ovladače krokových motorů A4982. Tento ovladač je určen pro ovládání dvoufázových bipolárních krokových motorů. Vhodnost tohoto ovladače zaručuje funkce mikrokrokování, kdy lze volit mezi celým, polovičním, čtvrtinovým nebo šestnáctinovým krokem. Samotná volba velikosti je zajištěna přes dva piny MS1 a MS2, kde jedna kombinace vstupních hodnot náleží jedné velikosti kroku. Dalšími rozhodujícími faktory bylo, že výstupní napětí může dosáhnout až 35 V a výstupní proud ±2 A nebo to, že uvnitř ovladače je překladač, díky kterému lze vytvořit mikrokrok, a tudíž pootočení rotoru krokového motoru, pouze jedním impulzem na patřičný vstupní pin STEP tohoto ovladače. Tím lze jak snížit zátěž procesoru MCU, tak i zjednodušit práci s tímto ovladačem. Kromě již zmíněných pinů, je zde i pin pro určení směru otáčení DIR nebo pin pro uspání ovladače SLEEP. Možnost uspání ovladače je vhodná pro snížení spotřeby v případě nečinnosti krokového motoru, kdy dojde k jeho odpojení od proudu. Kromě toho se sníží zahřívání ovladače. Avšak bylo potřeba si uvědomit, že pokud nebudou fáze krokového motoru buzeny proudem, tak nelze udržet pevnou polohu rotoru. Všechny řídicí piny ovladače jsou připojeny k MCU2. Jako napájení je přivedeno napětí 3,3 V. [22]

Jak bylo zmíněno, ovladače se při buzení fází krokového motoru silně zahřívají a je potřeba je ochlazovat. Proto byla ze spodu součástky na desce vytvořena chladící ploška, která je několika otvory propojena s druhou stranou desky, kde byla opět vytvořena chladící ploška. Tímto je zajištěno částečné chlazení celého ovladače.

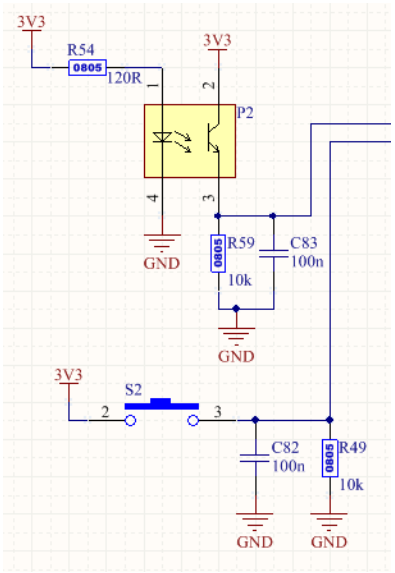


Obr. 48. Schéma zapojení ovladače krokového motoru.

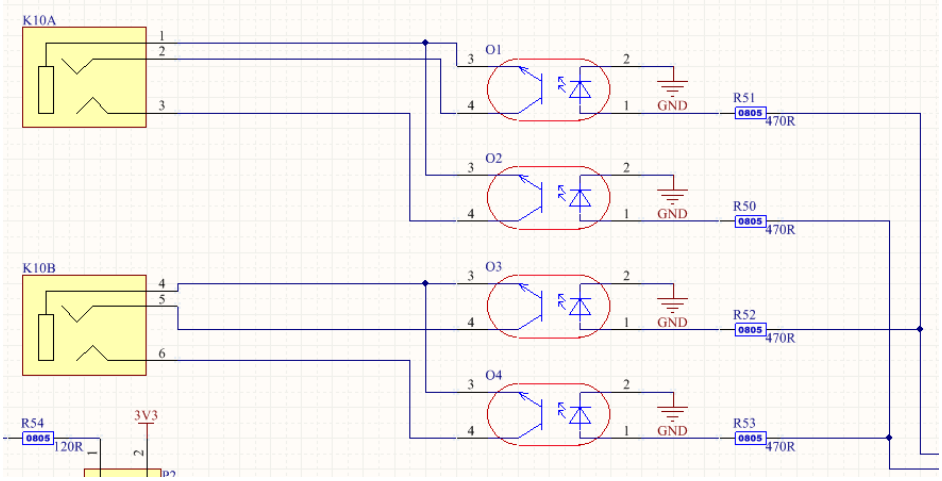
6.7 Trigger

Trigger se zde skládá ze dvou částí. První část tvoří dvou fázové tlačítko, které bylo vytvořeno ve smyslu tlačítka ostření a spouště u fotoaparátu. První fáze stisku je tvořena optosenzorem a druhá je tvořena mikrotlačítkem s natočením o 90°. Signály těchto dvou prvků jsou připojeny na piny MCU1, který se snaží detekovat přerušení.

Druhou částí je obvod, který tvoří konektor se dvěma vstupy typu JACK 3,5 mm a optočleny umožňující galvanické oddělení obvodů řídicí desky a elektroniky kamery, aby nedošlo k případnému poškození. Optočleny jsou ovládány přes výstupní piny MCU1. Pokud se na optočlen přivede napětí, vytvoří se průchozí cesta, kdy dojde ke spojení vodiče automatického ostření nebo spouště se zemnícím vodičem kamery.



Obr. 49. Schéma dvoufázového tlačítka.

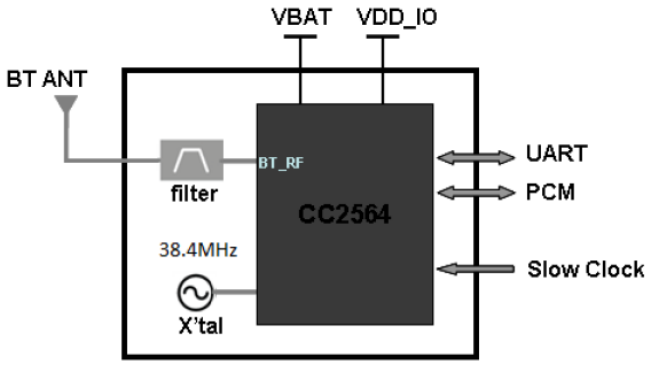


Obr. 50. Schéma zapojení optočlenů a konektoru pro trigger.

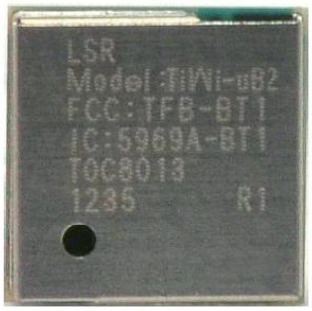
Obr. 51. Konektor JACK 3.5mm.

6.8 Bluetooth

Pro komunikaci za pomoci Bluetooth byl zvolen modul TiWi-uB2 od společnosti LS Research. Jedná se o modul s nízkou spotřebou napětí, který podporuje Bluetooth ve verzi 4.0. Bluetooth modul komunikuje s MCU1 přes rozhraní USART. Kromě toho bylo potřeba k tomuto modulu připojit anténu pro frekvenci 2,4 GHz. Modul obsahuje vnitřní oscilátor o frekvenci 38,4 MHz, kromě toho byl ještě přiveden pomalý hodinový signál o frekvenci 32,768 kHz za pomoci oscilátoru. Tento modul potřebuje pro funkčnost napájení s napětím 1,8 V a 3,3 V. [33]



Obr. 52. Architektura Bluetooth modulu. [33]

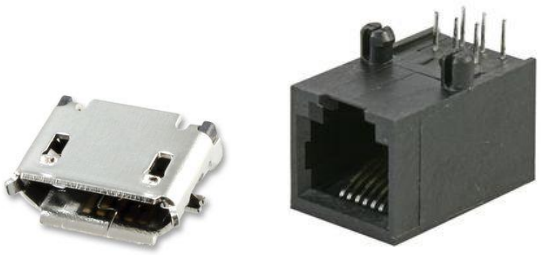


Obr. 53. Bluetooth modul. [33]

6.9 Doplňující konektory

Mezi doplňující konektory, které byly na desku umístěny, patří dva konektory typu RJ11 s 6 vývody. Oba tyto konektory mají vyvedeno napětí 3,3 V a zem, přičemž ostatní vývody jsou připojeny na piny MCU1 a lze je použít jako klasické vstupně výstupní piny.

Také jsou k dispozici dva konektory microUSB typu B pro USB rozhraní. Obě USB rozhraní jsou připojena k MCU1 na příslušné piny umožňující USB přenos s možností OTG. U každého konektoru je navíc pro případ, že budeme chtít připojené zařízení napájet napětím 5 V, vyveden kolíkový konektor, kde stačí použít jumper.



Obr. 54. Konektor microUSB (vlevo) a RJ11 (vpravo).

7 SOFTWARE ŘÍDICÍ DESKY

Samotný firmware řídicí desky, který započítává naprogramování obou mikropočítačů, se nachází ve fázi, kdy není využit celý potenciál navržené desky. Avšak, co je již v současné verzi firmwaru zahrnuto, splňuje požadované body zadání této práce, přičemž obsahuje ještě několik funkcí navíc.

7.1 Vývojové prostředí

Pro psaní kódu firmwaru bylo použito vývojové prostředí TASKING. Jedná se o vývojový nástroj pro embedded software. TASKING integruje v sobě vývojové prostředí, kompilátor a debugger nabízející podporu pro široký rozsah 8, 16 a 32bitových mikroprocesorů.

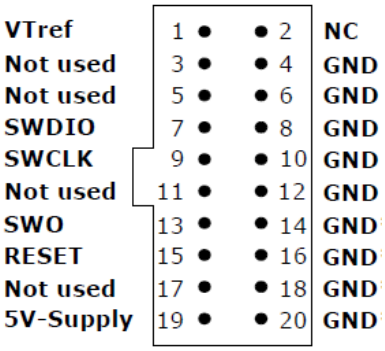
Firmware byl napsán v jazyce C a za využití základních knihoven od samotného výrobce použitých mikropočítačů.

7.2 Programátor

Pro programování a debugging bylo použito zařízení J-Link EDU od společnosti Segger. Jedná se o JTAG emulátor navrhnutý pro ARM procesory. S PC je propojen za pomoci USB. Dalším konektorem je samotný 20pinový JTAG konektor, který je kompatibilní s SWD (Serial Wire Debug), který v některých případech je náhradníkem JTAGu. Zvolené mikropočítače se programovaly přes rozhraní SWD. Toto rozhraní tvoří piny: [34]

Vref	Slouží pro detekci, zda cílové zařízení je napájeno
SWDIO	Obousměrný datový pin
SWCLK	Hodinový signál pro cílový procesor
SWO	Serial Wire Output Trace port
RESET	Resetovací signál pro cílový procesor
5 V	Napájecí napětí

Obr. 55. J-Link EDU. [34]



Obr. 56. 20pinový JTAG konektor. [34]

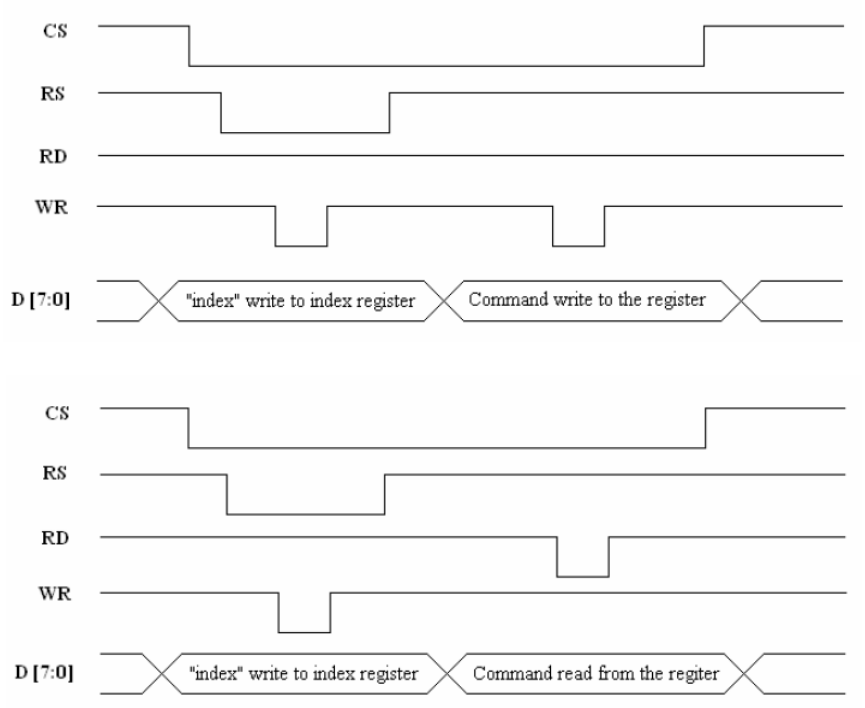
7.3 Práce s dotykovým LCD displejem

Práce s displejem zahrnuje dvě části. První část je tvořena nástroji pro vykreslování na displej, které zahrnují i samotné nastavení registrů ovladače displeje. Druhou část tvoří nástroj pro detekování stisku a určení souřadnice stisku.

V obou případech byla použita knihovna se základními funkcemi pro práci s displejem, kde bylo potřeba provést dodatečné úpravy.

7.3.1 Zobrazování na displeji

Komunikace s ovladačem displeje je řízena za pomoci čtyř signálů, a to signály CS, RS, RD a WR. Samotná sběrnice je nastavena na šířku 16 bitů. Průběhy signálů pro zápis a čtení jsou zobrazeny na obrázku (Obr. 57).



Obr. 57. Průběhy signálů pro zápis (nahoře) a čtení (dole). [35]

Aby se nemuselo složitě řešit časování a ovládání příslušných signálů, využilo se funkce FSMC, kterou nabízí MCU STM32F407 ve variantě se 100 a více piny. Při nastavení FSMC lze docílit toho, že do paměti GRAM displeje lze přistupovat stejně, jako kdyby se jednalo o interní paměť samotného MCU.

Nejdříve bylo potřeba definovat datovou strukturu, která bude obsahovat prvek s adresou registru a prvek s adresou pro data:

```
typedef struct
{
 __IO uint16_t LCD_REG;
 __IO uint16_t LCD_RAM;
} LCD_TypeDef;
```

a následně provést definici proměnné *LCD*, která je připojena k FSMC Bank1:

```
#define LCD ((LCD_TypeDef *)((0x60000000 | 0x00000001<<17) - 2))
```

V další fázi bylo možné přistoupit ke konfiguraci samotného hardwaru, kdy bylo potřeba přivést hodinový signál na příslušné GPIO porty a pro FSMC, nastavit GPIO piny do funkce FSMC a nakonec provést nastavení samotného FSMC:

```
FSMC_NORSRAMDeInit(FSMC_Bank1_NORSRAM1);
FSMC_NORSRAMInitTypeDef  FSMC_NORSRAMInitStructure;
FSMC_NORSRAMTimingInitTypeDef  p;

// FSMC_Bank1_NORSRAM1 timing configuration
p.FSMC_AddressSetupTime = 0x05; //One unit == 8.3ns (2units 16.6ns)
p.FSMC_AddressHoldTime = 2;
p.FSMC_DataSetupTime = 0x08;
p.FSMC_BusTurnAroundDuration = 0;
p.FSMC_CLKDivision = 0;
p.FSMC_DataLatency = 0;
p.FSMC_AccessMode = FSMC_AccessMode_A;

/* FSMC_Bank1_NORSRAM1 configured as follows:
  - Data/Address MUX = Disable
  - Memory Type = SRAM
  - Data Width = 16bit
  - Write Operation = Enable
  - Extended Mode = Disable
  - Asynchronous Wait = Disable
*/
FSMC_NORSRAMInitStructure.FSMC_Bank = FSMC_Bank1_NORSRAM1;
FSMC_NORSRAMInitStructure.FSMC_DataAddressMux =
 FSMC_DataAddressMux_Disable;
FSMC_NORSRAMInitStructure.FSMC_MemoryType = FSMC_MemoryType_SRAM;
FSMC_NORSRAMInitStructure.FSMC_MemoryDataWidth =
 FSMC_MemoryDataWidth_16b;
FSMC_NORSRAMInitStructure.FSMC_BurstAccessMode =
 FSMC_BurstAccessMode_Disable;
FSMC_NORSRAMInitStructure.FSMC_AsynchronousWait =
 FSMC_AsynchronousWait_Disable;
FSMC_NORSRAMInitStructure.FSMC_WaitSignalPolarity =
 FSMC_WaitSignalPolarity_Low;
FSMC_NORSRAMInitStructure.FSMC_WrapMode = FSMC_WrapMode_Disable;
FSMC_NORSRAMInitStructure.FSMC_WaitSignalActive =
 FSMC_WaitSignalActive_BeforeWaitState;
```

```
FSMC_NORSRAMInitStructure.FSMC_WriteOperation =
 FSMC_WriteOperation_Enable;
FSMC_NORSRAMInitStructure.FSMC_WaitSignal = FSMC_WaitSignal_Disable;
FSMC_NORSRAMInitStructure.FSMC_ExtendedMode =
 FSMC_ExtendedMode_Disable;
FSMC_NORSRAMInitStructure.FSMC_WriteBurst = FSMC_WriteBurst_Disable;
FSMC_NORSRAMInitStructure.FSMC_ReadWriteTimingStruct = &p;
FSMC_NORSRAMInitStructure.FSMC_WriteTimingStruct = &p;

FSMC_NORSRAMInit(&FSMC_NORSRAMInitStructure);

// Enable FSMC_Bank1_NORSRAM1
FSMC_NORSRAMCmd(FSMC_Bank1_NORSRAM1, ENABLE);
```

Po dokončení všech nastavení mohla probíhat komunikace s displejem za pomoci dvou funkcí *LCD_Driver_Write_Command* a *LCD_Driver_Write_Data*, kdy se přistupuje k prvkům proměnné *LCD*, která je typu struktury *LCD_TypeDef*. Pro poslání adresy registru se použila funkce:

```
void LCD_Driver_Write_Command(u16 data)
{
 LCD->LCD_REG = data;
}
```

a pro poslání dat se použila funkce:

```
void LCD_Driver_Write_Data(u16 data)
{
 LCD->LCD_RAM = data;
}
```

Při využití těchto funkcí probíhala komunikace s displejem. V první řadě bylo potřeba nastavit displej, kdy se provedl zápis do několika registrů. V tomto případě se postupovalo podle instrukcí v dokumentaci displeje. Po nastavení bylo možné využít několik funkcí pro práci s displejem:

`void LCD_Driver_Set_Window(unsigned int x1, unsigned int y1, unsigned int x2, unsigned int y2)`

- Nastavení vykreslovacího okna, kde parametry `x1` a `y1` značí souřadnici levého horního rohu okna a parametry `x2` a `y2` jsou souřadnice pravého dolního rohu.

`void LCD_Driver_Paint(u8 r, u8 g, u8 b)`

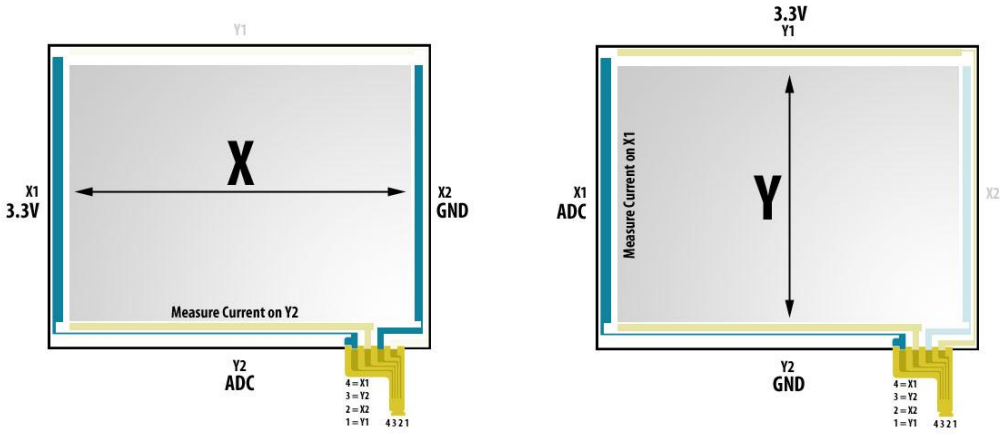
- Nastavení jednobarevného pozadí celého displeje, kde `r` je hodnota červené barvy, `g` je hodnota zelené barvy a `b` je hodnota modré barvy RGB barevného modelu.

`void LCD_Driver_Picture_RLE(GUI_image_t *obr, int x, int y)`

- Vykreslení komprimovaného obrázku RLE kompresí. `*obr` je ukazatel na strukturu s daty a informacemi o obrázku. Souřadnice levého horního rohu umístění na displeji jsou předány v parametrech `x` a `y`.

7.3.2 Dotyková vrstva

Dotyková vrstva je tvořena čtyřmi vodiči, a to ±Y a ±X. Pro zjištění souřadnice Y se přivede napětí 3,3 V na +Y a zem na -Y, poté se připojí +X na AD převodník. Naměřená hodnota se posléze použije pro přepočet na výslednou souřadnici. Pro zjištění souřadnice X se přivede napětí 3,3 V na +X a zem na -X, poté se připojí +Y na AD převodník, kde změřená hodnota se opět použije pro přepočet na výslednou souřadnici.



Obr. 58. Určování souřadnice X a Y. [36]

Před využitím AD převodníku, bylo potřeba nakonfigurovat příslušné piny a samotný AD převodník, jehož nastavení je následující:

```
ADC_CommonInitTypeDef ADC_CS;
ADC_CommonStructInit (& ADC_CS);
ADC_CS.ADC_Mode = ADC_Mode_Independent;
ADC_CS.ADC_Prescaler = ADC_Prescaler_Div2;
ADC_CS.ADC_DMAAccessMode  = ADC_DMAAccessMode_Disabled;
ADC_CS.ADC_TwoSamplingDelay  = ADC_TwoSamplingDelay_5Cycles;
ADC_CommonInit (& ADC_CS);

ADC_InitTypeDef ADC_S;
ADC_StructInit (& ADC_S);
ADC_S. ADC_Resolution = ADC_Resolution_12b;
ADC_S. ADC_ScanConvMode = DISABLE;
ADC_S. ADC_ContinuousConvMode = DISABLE;
ADC_S. ADC_ExternalTrigConvEdge = ADC_ExternalTrigConvEdge_None;

ADC_S. ADC_ExternalTrigConv = ADC_ExternalTrigConv_T2_CC2;
ADC_S. ADC_DataAlign = ADC_DataAlign_Right;
ADC_S. ADC_NbrOfConversion = 1;
ADC_Init (ADC1, & ADC_S);

ADC_Cmd (ADC1, ENABLE);
```

Podle aktuálně zjišťované souřadnici se přivedlo napětí a zem na příslušné piny a provedla se konverze AD převodníkem, pro kterou byla použita funkce:

```
uint16_t adc_convert()
{
 ADC_SoftwareStartConv(ADC1);//Start the conversion
 while(!ADC_GetFlagStatus(ADC1, ADC_FLAG_EOC));//Processing
 return ADC_GetConversionValue(ADC1); //Return the converted data
}
```

která vrací převedenou hodnotu. Po získání výsledných hodnot pro osu X a Y, se tyto hodnoty přepočítávají na velikost displeje.

Pro účel, zda byl proveden stisk a tudíž AD převodník vrací určitou hodnotu, bylo použito přerušení od časovače. V pravidelných intervalech se poté volá funkce:

```
int scan(int *ax, int *ay)
```

Tato funkce zjistí, zda byl proveden stisk tím, že se provede konverze AD převodníkem. Pokud je převedená hodnota nad určitou hranici, tak to znamená, že je stisk a provede se zjištění přesných souřadnic X a Y. Kromě toho, se v přerušení od časovače aktivuje další časovač, a to pouze v případě, kdy byl proveden stisk, protože je potřeba zajistit, aby tento stisk nebyl detekován vícekrát. Proto je zde delší prodleva.

7.3.3 GUI

Když už byly k dispozici funkce pro zobrazování na displeji a možnost detekovat stisk s určením souřadnice stisku, bylo možné vytvořit základní prvky pro vytváření GUI. To zahrnovalo vytvoření výčtových typů, jako souhrn možných pohledů (obrazovek) a všech možných událostí, které mohou vzniknout stiskem na displeji:

enum GUI_viewEnum_t

- Výčtový typ se souhrnem všech možných pohledů

enum GUI_eventCode_t

- Výčtový typ se souhrnem všech možných událostí

Následovalo použití struktury pro jeden objekt pohledu, zahrnující souřadnici umístění, ukazatel na vykreslovaný obrázek a typ události, kterou daný objekt vytvoří po stisku. Další použitá struktura definuje jeden pohled, který obsahuje pole s ukazateli na jednotlivé objekty umístěné v daném pohledu a označení pohledu:

typedef struct GUI_widget_t

- Struktura definující jeden objekt

typedef struct GUI_view_t

– Struktura definující jeden pohled

Pro práci s obrázky byla k dispozici také struktura, která zahrnuje údaje o daném obrázku, jako je výška, šířka, počet pixelů, počet barev, ukazatel na paletu barev a jednotlivé pixely, nebo zda jde o komprimovaný obrázek za pomoci RLE komprese. Tato struktura má název *GUI_image_t*.

Pro práci s GUI bylo k dispozici několik funkcí:

*GUI_view_t *GUI_view_create(uint8_t view)*

- Funkce pro vytvoření pohledu, kde parametr *view* obsahuje kód pohledu.

*void GUI_widget_create(GUI_view_t *v, int x, int y,*
*GUI_image_t *on, GUI_image_t *off, int EC)*

- V pohledu *v* vytvoří objekt na souřadnici dané parametry *x* a *y*. **on* a **off* jsou ukazatelé na obrázek, který se vykreslí, pokud je obrázek aktivní po stisku nebo neaktivní. Současně se předává informace o události, která je vytvořena po stisku na daný objekt a je předána jako parametr *EC*.

*void GUI_view_changeView(GUI_view_t *v)*

- Funkce, která změní aktuální pohled na pohled *v*.

*void GUI_view_draw(GUI_view_t *v)*

- Funkce vykreslí pohled *v*.

*void GUI_widget_draw(GUI_widget_t *w)*

- Funkce vykreslí objekt *w*.

*int GUI_returnEventCode(GUI_view_t *v, int x, int y)*

- Funkce vrací kód události, která se vyvolá v pohledu *v* na souřadnicích *x* a *y*.

Pro překreslování jednotlivých pohledů slouží přerušení od časovače. V tomto přerušení se porovná předchozí a současný pohled. Pokud se liší, tak dojde k novému vykreslení již aktuálního pohledu. Aktuální pohled se nastavuje v hlavní smyčce programu, kde se po zjištění dotyku zjistí, zda se na souřadnicích stisku vyskytuje nějaká událost. Podle události se změní pohled.

7.4 Komunikace mezi mikropočítači

Komunikace mezi oběma MCU je zprostředkována za pomoci USART rozhraní. Rychlost je nastavena na 115 200 Bd. Přesná konfigurace je stejná u obou MCU a je následující:

```
USART_InitTypeDef USART_InitStruct;

USART_InitStruct.USART_BaudRate = 115200;
USART_InitStruct.USART_WordLength = USART_WordLength_9b;
USART_InitStruct.USART_StopBits = USART_StopBits_1;
USART_InitStruct.USART_Parity = USART_Parity_Even;
USART_InitStruct.USART_HardwareFlowControl =
 USART_HardwareFlowControl_None;
USART_InitStruct.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
USART_Init(USART2, &USART_InitStruct);

USART_Cmd(USART2, ENABLE);
USART_ITConfig(USART2, USART_IT_RXNE, ENABLE);
```

Po přijetí znaku se vyvolá přerušení, pro které byla vytvořena funkce:

```
void USART2_IRQHandler(void)
{
 if(USART_GetITStatus(USART2, USART_IT_RXNE) != RESET)
 {
     RxChar = (uint8_t)(USART_ReceiveData(USART2));
     USART_ClearITPendingBit(USART2, USART_IT_RXNE);
 }
}
```

kde jde o to, že se přečte přijatý byte a uloží do proměnné `RxChar`.

Na základě přijatého znaku se provede jedna odpovídající operace. Seznam všech možných znaků je zobrazen v tabulce (Tab. 4).

Tab. 4. Seznam znaků.

Znak	Operace
0x30	Zastavení obou motorů
0x11	Motor1 se otáčí doprava. Nástavce se otáčí k sobě
0x12	Motor1 se otáčí doleva. Nástavce se otáčí od sebe
0x21	Motor2 se otáčí doprava. Nástavce se pohybují od sebe
0x22	Motor2 se otáčí doleva. Nástavce se pohybují k sobě
0x31	Nastavení celého kroku pro oba motory
0x32	Nastavení polovičního kroku pro oba motory
0x33	Nastavení čtvrtečního kroku pro oba motory
0x34	Nastavení šestnáctinového kroku pro oba motory
0x50	Kalibrace. Motor1 byl nastaven do výchozí pozice.
0x60	Detekce. Objekt je ve středu zorného pole.
0x61	Detekce. Objekt je vlevo.
0x62	Detekce. Objekt je vpravo.

7.5 Ovladače krokových motorů

Zvolený ovladač pro ovládání krokových motorů obsahuje překladač, který překládá vstupní impulzy na generování buzení jednotlivých fází krokového motoru, aby se vytvořil jeden mikrokrok.

Vstupní piny ovladače tvoří: [22]

ENABLE

- Deaktivace výstupů pro buzení fází. Vstupní piny překladače zůstávají aktivní.

SLEEP

- Deaktivace většiny vnitřních obvodů za účelem snížení spotřeby.

RESET

– Nastavení tranzistorů do předdefinovaného stavu. Všechny výstupy a vstupy jsou deaktivovány.

MS1/MS2

– Kombinace na těchto vstupech definuje velikost kroku, a to podle tabulky (Tab. 5). Změna na těchto pinech nemá efekt do dalšího impulzu na pinu STEP.

DIR

– Určuje směr otáčení motoru. Změna na tomto pinu nemá efekt do dalšího impulzu na pinu STEP.

STEP

– Změna z úrovně logické 0 na logickou 1 zapříčiní, že motor vykoná jeden krok. Velikost kroku závisí na stavu pinů MS1 a MS2.

Tab. 5. Kombinace na pinech MS1 a MS2 pro nastavení velikosti kroku. [22]

MS1	MS2	Velikost kroku
L	L	Celý
H	L	Poloviční
L	H	Čtvrteční
H	H	Šestnáctinový

Pro samotnou práci bylo definováno několik výčtových typů, které usnadňují a zpřehledňují informace o jednotlivém motoru:

```
typedef enum {APART = 0, TOGETHER = !APART} Direction;
```

- Definování směru otáčení.

```
typedef enum {SLEEP = 0, READY, RUN, STEP} DriverStatus;
```

- Definování možných stavů motoru.

```
typedef enum {DRIVER1 = 1, DRIVER2 = 2} Driver;
```

- Definování, o jaký ovladač se jedná.

```
typedef enum {FULL = 0, HALF, QUARTER, SIXTEENTH} Step_Value;
```

- Definování jednotlivých velikostí kroku.

Také byla vytvořena struktura, která definuje jednotlivý motor a skládá se z položky určující ovladač, směr otáčení, stav motoru, velikost kroku a pozici:

```
typedef struct
{
 Driver driverx;
 Direction direction;
 DriverStatus status;
 Step_Value stepValue;
 int position;
} Motor;
```

Nakonec byly vytvořeny funkce, které umožňují práci s ovladači a ovládání krokových motorů:

```
void Driver_Initialization(Motor* motorx, Driver driverx)
```

- Inicializační funkce, která motoru *motorx* přiřadí příslušný ovladač *driverx*. Následně je motoru nastaven počáteční směr, velikost kroku a stav je změněn na stav *READY*.

```
void Driver_Step(Motor* motorx)
```

- Funkce, která motoru *motorx* vyšle jeden impulz pro vytvoření jednoho kroku.

```
void Driver_Step_Value(Step_Value value)
```

- Nastavení velikosti kroku pro oba motory dohromady.

```
void Driver_Direction(Motor* motorx, Direction direction)
```

- Nastavení směru otáčení *direction* pro motor *motorx*.

void Driver_Sleep(Motor motorx, FunctionalState state)*

- Uvedení nebo uvolnění ovladače pro motor *motorx* do nebo ze spánku. Parametr *state* obsahuje buď *ENABLE* nebo *DISABLE*.

Celkově dohromady celé ovládání funguje tak, že po stisku tlačítka na displeji, určující pohyb, se vyvolá událost v hlavní smyčce programu MCU1. Následkem toho se pošle k MCU2 odpovídající příkaz a uloží se signalizace, že některý z motorů je v pohybu. Tímto se zamezí vícenásobnému zasílání příkazů pro pohyb motoru. V hlavní smyčce programu je navíc testování, že v případě, že stisk byl uvolněn a je signalizace o tom, že některý z motorů je v pohybu, tak se vyšle příkaz pro zastavení motoru a signalizace se změní na stav, že motory jsou v klidu. Na straně MCU2, který obdrží příkaz, se vykoná příslušná akce. V případě, že budu chtít, aby se krokový motor začal otáčet a uvedl tak nástavce do pohybu, tak se nastaví příslušnému motoru směr, uvolní se jeho ovladač z režimu spánku a nastaví se do režimu *STEP*. Současně se aktivuje časovač, který vytvoří prodlevu mezi uvedením motoru do stavu *RUN*. Tudíž, po stisku tlačítka se vyšle ovladači několik impulzů pro vytvoření malého pootočení a po menší prodlevě se teprve začnou nepřetržitě vysílat impulzy pro vytvoření kroku, dokud neuvolníme stisk.

Protože se nástavce pohybují po přesně definovaném úseku, tak mezi krajními zarážkami lze vyslat pouze omezený počet impulzů na pin STEP. Ve struktuře daného motoru se proto inkrementuje a dekrementuje prvek *position*, podle směru pohybu. Tímto se dá odvodit aktuální pozice nástavce.

7.6 Zachytávání kompozitního videa

Možnost zachytávání kompozitního videa za pomoci video převodníku obnášelo zprovoznit komunikační rozhraní mezi MCU1 a samotným převodníkem. Poté bylo potřeba provést základní nastavení registrů. Pokud byl převodník nastaven, vysílal na 8bitové výstupní sběrnici obrazová data ve formátu YCbCr 4:2:2 a současně synchronizační signály HSYNC a VSYNC na přídavných pinech. Další fáze zahrnovala patřičné nastavení samotného MCU1.

7.6.1 Video převodník

Pro komunikaci a následné programování registrů video převodníku se používá rozhraní I^2C. To znamenalo nakonfigurovat příslušné piny MCU1 a poté následně samotné rozhraní I^2C:

```
I2C_InitTypeDef I2C_InitStruct;

I2C_InitStruct.I2C_Mode = I2C_Mode_I2C;
I2C_InitStruct.I2C_DutyCycle = I2C_DutyCycle_16_9;
I2C_InitStruct.I2C_OwnAddress1 = 0x00;
I2C_InitStruct.I2C_Ack = I2C_Ack_Enable;
I2C_InitStruct.I2C_AcknowledgedAddress =
 I2C_AcknowledgedAddress_7bit;
I2C_InitStruct.I2C_ClockSpeed = 400000;
I2C_Init(I2C2, &I2C_InitStruct);

I2C_Cmd(I2C2, ENABLE);
I2C_ITConfig(I2C2, I2C_IT_ERR, ENABLE);
```

Následně byly vytvořeny funkce pro posílání a příjem dat:

```
uint8_t I2C_MasterSend(I2C_TypeDef* I2Cx, uint8_t
 slave_address, uint8_t register_address, uint8_t data)
```

- Na zařízení s adresou `slave_address` se uloží do registru `register_address` data předaná v parametru `data`. Funkce vrací 0 při správném odeslání. Pokud nastane problém, funkce vrací 1.

```
uint8_t I2C_MasterRead(I2C_TypeDef* I2Cx, uint8_t
 slave_address, uint8_t register_address)
```

- Ze zařízení s adresou `slave_address` se přečte obsah registru `register_address`. Funkce vrací přečtená data při správném přečtení. Pokud nastane problém, funkce vrací 1.

Pro komunikaci s video převodníkem byla adresa zařízení nastavena na hodnotu 0xB8.

S funkčním rozhraním I^2C jsem mohl začít s nastavováním převodníku. Nastavované registry a nastavované hodnoty jsou uvedeny v tabulce (Tab. 6). Jednalo se pouze o základní nastavení, přičemž si lze dodatečně nastavovat ostrost, světlost, kontrast atd.

Tab. 6. Nastavované registry a hodnoty.

Registr	Hodnota	Popis
R00	0x00	Zdroj video signálu na kanálu č. 1
R03	0x2F	YCbCr výstup / povolení HSYNC a VSYNC / povolení hodinového signálu
R04	0xF0	Zakázání možnosti se přepnout do standardu NTSC a SECAM
R0D	0x00	Nastavení rozsahu pro Y na 16 až 235, pro Cb a CR na 16 – 240 / nastavení offsetu na +128 / diskrétní synchronizační výstup
R0F	0x02	VBLK (Vertical Blanking Output) mód
R28	0x00	Automatické přepínání mezi standardy

Pro dodatečné ovládání převodníku byly vytvořeny funkce:

uint8_t TVP5150AM1_CheckConnection(void)

- Funkce přečte obsah registru R80, kde by měla být hodnota 0x51. Poté přečte obsah registru R81, kde by měla být uložena hodnota 0x50. Jedná se o ID registr převodníku. Proto, pokud se nepřečtou tyto hodnoty, nastal někde problém a převodník není připojen. Pokud je všechno v pořádku, je návratová hodnota 0, v opačném případě je návratová hodnota 1.

void TVP5150AM1_SelectVideoSource(VideoSource source)

- Podle toho, zda parametr *source* obsahuje *INPUT1* nebo *INPUT2*, se nastaví zdroj video signálu na příslušný kanál.

uint8_t TVP5150AM1_CheckVideoSignal(void)

- Funkce přečte obsah registru R88. Pokud se na přijatá data použije maska 0x0E a výsledkem je hodnota 0x0E, tak to znamená, že je detekován video vstup a funkce vrací 1. V opačném případě není na vstupu detekováno žádné video a funkce vrací 0.

7.6.2 Zpracování obrazových dat

Na straně MCU1 bylo potřeba provést určitá nastavení, abych byl schopen zpracovat přicházející obrazová data po 8bitové sběrnici. Za tímto účelem jsem využil jednu z funkcí, kterou nabízí daný MCU. Jedná se o funkci DCMI (Digital Camera Interface), která slouží právě pro daný účel. DCMI je schopné pracovat až s 14bitovou sběrnicí a hlavně je schopné přijmou data ve formátu YCbCr 4:2:2.

Než jsem začal s konfiguracemi, definoval jsem několik struktur a proměnných, které jsem následně použil již při konfiguracích. První takovou strukturou je struktura, která definuje tvar přicházejících dat po sběrnici:

```
typedef struct
{
 uint8_t Y0;
 uint8_t Cb;
 uint8_t Y1;
 uint8_t Cr;
} YCbCr_t;
```

Následně jsem vytvořil datové pole, které bude sloužit jako buffer pro DMA. Toto datové pole bude obsahovat jeden přijatý obrazový řádek. Uvažuje se s video standardem PAL o šířce řádku 720 pixelů, ale vzhledem k variantě YCbCr 4:2:2 a použité struktuře `YCbCr_t`, je délka poloviční:

```
YCbCr_t YCbCr_DMA_Buffer [PAL_WIDTH / 2] = {{0}}; //DMA buffer
```

Další struktura, kterou jsem definoval, obsahuje jednotlivé prvky již výsledného jednoho obrazového pixelu:

```
typedef struct
{
 uint8_t Y;
 uint8_t Cb;
 uint8_t Cr;
} VideoData;
```

Vzhledem k tomu, že přijatá data jsou ve formátu YCbCr a displej přijímá data ve formátu RGB 565, byl potřeba přepočet každého obrazového pixelu. Vzhledem k době použité pro přepočet, jsem nemohl provést přepočet ihned po příjmu jednoho řádku, kdy se vyvolá přerušení od DMA, že buffer je již plný. Pokud bych tento přepočet aplikoval, tak to by způsobilo, že než přepočítám jeden řádek, tak druhý bych mohl minout. Z toho důvodu a důvodu pozdější aplikace detekce bodu, jsem se rozhodl pro uložení celého jednoho obrazového rámce do paměti MCU1. Avšak byl zde problém s velikostí dostupné paměti.

Uvažuje se, že výsledný obraz, vykreslovaný na displej, bude mít 240 pixelů na výšku a 360 pixelů na šířku. Protože budu pracovat se standardem PAL o šířce 720 pixelů, tak jsem provedl redukci na polovinu. Co se týče výšky u standardu PAL, tak ta je 576 pixelů. Když jsem vzal v úvahu, že se jedná o prokládané video, tak stačí vykreslovat vždy jen jednu polovinu obrazu, složenou z lichých nebo sudých řádků. Kromě toho, jsem ještě musel aplikovat ořez, abych se dostal na konečnou výšku 240 pixelů. Jeden pixel má velikost 3 B, při obrazu 240 x 360 pixelů jsem dostal, že potřebuji 259 200 B. Celková velikost paměti MCU1 je 192 KB, přičemž maximální alokovaný blok může mít velikost maximálně 65 535 B. Proto jsem přistoupil k řešení, že jsem provedl kompresi jednoho řádku na polovinu a celý obraz rozdělil na dvě pole. Po této aplikaci jsem dostal, že potřebuji celkem 129 600 B, což odpovídá 64 800 B na jedno datové pole (jednu polovinu obrazu). Vzhledem k použité struktuře `VideoData`, byla vytvořena následující datová pole a k nim ukazatele pro snadnější práci:

```
VideoData BufferData1 [21600] = {{0}}; //prvni polovina dat
VideoData BufferData2 [21600] = {{0}}; //druha polovina dat

VideoData *pBufferData1 = &BufferData1[0];
VideoData *pBufferData2 = &BufferData2[0];
```

Pro využití DCMI bylo potřeba nastavit všechny potřebné piny do funkce DCMI. Poté jsem mohl provést konfiguraci samotného DCMI:

```
DCMI_InitTypeDef DCMI_InitStruct;
DCMI_CROPInitTypeDef DCMI_Crop;

DCMI_DeInit();
DCMI_InitStruct.DCMI_SynchroMode = DCMI_SynchroMode_Hardware;
DCMI_InitStruct.DCMI_CaptureMode = DCMI_CaptureMode_SnapShot;
DCMI_InitStruct.DCMI_PCKPolarity = DCMI_PCKPolarity_Falling;
DCMI_InitStruct.DCMI_VSPolarity = DCMI_VSPolarity_High;
DCMI_InitStruct.DCMI_HSPolarity = DCMI_HSPolarity_High;
DCMI_InitStruct.DCMI_CaptureRate = DCMI_CaptureRate_All_Frame;
DCMI_InitStruct.DCMI_ExtendedDataMode = DCMI_ExtendedDataMode_8b;
DCMI_Init(&DCMI_InitStruct);

DCMI_Crop.DCMI_CaptureCount = 720 * 2 - 1;
DCMI_Crop.DCMI_HorizontalOffsetCount = 148 - 1; //offset zleva
DCMI_Crop.DCMI_VerticalLineCount = 240 - 1; //počet zachycenych radku
DCMI_Crop.DCMI_VerticalStartLine = 24; //offset shora (625 - 576) / 2
DCMI_CROPConfig (&DCMI_Crop);
DCMI_CROPCmd (ENABLE);
```

Následně jsem také provedl nakonfigurování DMA (Direct Memory Access). DMA slouží pro přesun mezi různými lokacemi v adresovém prostoru bez účasti procesoru. DMA přenos je definován zdrojovou a cílovou adresou. Zdrojovou adresu jsem nastavil na DCMI a cílovou adresu jsem nastavil jako adresu mnou vytvořeného datového pole `YCbCr_DMA_Buffer`, určeného pro uložení jednoho přijatého obrazového řádku. Výsledná konfigurace samotného DMA je:

```
DMA_InitTypeDef DMA_InitStruct;

DMA_Cmd(DMA2_Stream1, DISABLE);
DMA_DeInit(DMA2_Stream1);

DMA_InitStruct.DMA_Channel = DMA_Channel_1;
```

```
DMA_InitStruct.DMA_PeripheralBaseAddr = (uint32_t)DCMI_BASE + 0x28;
DMA_InitStruct.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
DMA_InitStruct.DMA_PeripheralDataSize = DMA_PeripheralDataSize_Word;
 //32bit
DMA_InitStruct.DMA_PeripheralBurst = DMA_PeripheralBurst_Single;
DMA_InitStruct.DMA_DIR = DMA_DIR_PeripheralToMemory;


DMA_InitStruct.DMA_Memory0BaseAddr = (uint32_t)&YCbCr_DMA_Buffer[0];
DMA_InitStruct.DMA_BufferSize = dmaBufferLength; //dmaBufferLength
 720/2
DMA_InitStruct.DMA_MemoryInc = DMA_MemoryInc_Enable;
DMA_InitStruct.DMA_MemoryDataSize = DMA_MemoryDataSize_Word;
DMA_InitStruct.DMA_MemoryBurst = DMA_MemoryBurst_Single;


DMA_InitStruct.DMA_Mode = DMA_Mode_Circular;
DMA_InitStruct.DMA_Priority = DMA_Priority_High;
DMA_InitStruct.DMA_FIFOMode = DMA_FIFOMode_Enable;
DMA_InitStruct.DMA_FIFOThreshold = DMA_FIFOThreshold_Full;


DMA_Init(DMA2_Stream1, &DMA_InitStruct);
```

Pro ovládání DCMI a DMA byly vytvořeny funkce:

`void TVP5150AM1_StartCapture(void)`

- Tato funkce aktivuje DMA a následně aktivuje zachytávání DCMI.

`void TVP5150AM1_StopCapture(void)`

- Tato funkce deaktivuje DMA a následně deaktivuje zachytávání DCMI.

Celý proces zachycení obrazových dat zajišťují dvě přerušení. První přerušení je od DMA, kdy je naplněno datové pole `YCbCr_DMA_Buffer`, které obsahuje právě jeden obrazový řádek. Když je vyvoláno přerušení, vytvoří se ukazatel na datové pole `YCbCr_DMA_Buffer`. Podle toho, zda se zpracovává prvních 120 řádků nebo druhých 120 řádku, se ukládají data do datového pole `BufferData1` nebo `BufferData2`. Jeden prvek datového pole `YCbCr_DMA_Buffer` má tvar struktury `YCbCr_t`. Ta obsahuje prvky `Y0`, `Y1`, `Cb` a `Cr`. Vlastně jde hned o dva pixely, kde první pixel má jasovou složku

Y0 a druhý *Y1*. Barevné složky *Cb* a *Cr* jsou společné pro oba pixely. První redukce, která je provedena, je snížení šířky 720 pixelů na 360 pixelů. To je provedeno zprůměrováním jasových složek *Y0* a *Y1* do jednoho pixelu. Abych byl schopen uložit celý obraz do paměti MCU1, je potřeba provést ještě jednu redukci z 360 na 180 pixelů. Tato redukce je provedena opět zprůměrováním dvou sousedních pixelů.

Druhé přerušení je od DCMI, které znamená přijetí celého obrazového rámce. Do tohoto přerušení se dá dostat až po přijetí všech obrazových řádků a s již naplněnými datovými poli *BufferData1* a *BufferData2*. Opět se pracuje s prvním nebo druhým datovým polem, podle toho, který obrazový řádek se právě zpracovává. V obou případech je potřeba převést data z formátu YCbCr na formát RGB 565. Pro tento účel byla vytvořena datová pole:

```
signed short ValuesRCr[256];
signed short ValuesGCr[256];
signed short ValuesGCb[256];
signed short ValuesBCb[256];

signed short *pValuesRCr = &ValuesRCr[0];
signed short *pValuesGCr = &ValuesGCr[0];
signed short *pValuesGCb = &ValuesGCb[0];
signed short *pValuesBCb = &ValuesBCb[0];
```

která byla naplněna při inicializaci a to následujícím způsobem:

```
int i;
for (i = 0; i < 256; i++)
 ValuesRCr[i] = (i - 128) * 1400 / 1000;
for (i = 0; i < 256; i++)
 ValuesGCr[i] = (i - 128) * 711 / 1000;
for (i = 0; i < 256; i++)
 ValuesGCb[i] = (i - 128) * 343 / 1000;
for (i = 0; i < 256; i++)
 ValuesBCb[i] = (i - 128) * 1765 / 1000;
```

Výpočet jednotlivých složek RGB je následující:

```
pom = Y + *(pValuesRCr + Cr); // red
R1 = (pom < 0) ? 0 : (pom > 254) ? 254 : (uint8_t)pom;
pom = Y - *(pValuesGCr + Cr) - *(pValuesGCb + Cb); // green
G1 = (pom < 0) ? 0 : (pom > 254) ? 254 : (uint8_t)pom;
pom = Y + *(pValuesBCb + Cb); // blue
B1 = pom < 0 ? 0 : pom > 254 ? 254 : (uint8_t)pom;
```

Když už jsou známy jednotlivé složky RGB, stačí udělat přepočet z RGB 888 na RGB 565 a daný pixel vykreslit na displej:

```
R1 = R1 * 31 / 254;
G1 = G1 * 63 / 254;
B1 = B1 * 31 / 254;

LCD_Driver_Write_Data(((R1 & 0x1F)<<11) | ((G1 & 0x3F)<<5) | (B1 &
 0x1F)); //vykreslim pixel
```

Protože jsou v datovém poli uložena data pouze pro 180 pixelů na řádek, vykreslují se pixely způsobem, že mezi dva pixely se vykreslí jejich průměr. Tímto se docílí šířky 360 pixelů.

Se zachytáváním a zpracováním videa souvisí jeden časovač, který v intervalech po 1 sekundě kontroluje, zda je na vstupu video převodníku aktivní video vstup. Pokud dříve nebyl detekován video vstup a nyní již je detekován, tak se zavolá funkce `TVP5150AM1_StartCapture` pro aktivaci zachytávání obrazových dat. Pokud dříve byl detekován video vstup a nyní již není detekován, tak se zavolá funkce `TVP5150AM1_StopCapture` pro deaktivaci zachytávání obrazových dat.

7.7 Anaglyf 3D

Anaglyf je stereoskopická technika, která umožňuje uživateli vnímat prostorově obraz. Jde o metodu využívající barevné propustnosti různých barevných filtrů. Barevné

filtry propouštějí pouze světlo určité vlnové délky. Pokud sledovaný obraz nebude disponovat barvou s odpovídající vlnovou délkou, kterou filtr propouští, bude se tato barva jevit jako černá. Při snímání bílé barvy dojde ke zbarvení na barvu filtru.

V mém případě se uvažuje o brýlích s červeným filtrem pro jedno oko a s azurovým filtrem pro druhé oko. Proto jeden obraz bude mít nádech do červena, protože zde bude potlačena modrá a zelená složka. Druhý obraz bude mít nádech do azurové barvy, protože zde potlačím červenou složku. Je nutné aplikovat správně takto vytvořené obrazy na oči. Pokud vezmu obraz pro levé oko a dám mu nádech do azurové barvy, měl bych na toto oko použít červený filtr, na pravé oko je to obráceně. Teoreticky by měla barva filtru nahradit chybějící barevnou složku.

Při aplikaci anaglyfu bylo potřeba sejmout obraz z jednoho video vstupu, potlačit u něj červenou složku, pak se přepnout na druhý video vstup a následně potlačit modrou a zelenou složku. Nakonec oba obrazy spojit do jednoho. Z důvodu omezení vnitřní paměti a potřeby převodu na formát RGB 565 byla použita pouze jasová složka `Y`, čímž docílím černobílého obrazu. Pro uložení obrazových dat do paměti se využila datová pole určená pro normální obrazová data `BufferData1` a `BufferData2`. Prvky těchto datových polí tvoří struktura `VideoData` s prvky `Y`, `Cb` a `Cr`. Pro můj účel jsem s prvkem `Y` pracoval jako se složkou `R`, s prvkem `Cb` jako s `G` a s prvkem `Cr` jako s `B`. V prvním kroku jsem se přepnul na video vstup `INPUT1`. U tohoto obrazu jsem potlačil červenou složku, proto složka `R` je 0. Do složky `G` a `B` jsem uložil hodnotu jasové složky `Y`. Po uložení celého obrazového rámce, jsem se přepnul na video vstup `INPUT2`. Z důvodu příchodu špatných obrazových rámců, jsem byl donucen vynechat prvních 20 obrazových rámců, čímž došlo ke snížení rychlosti zobrazovaného videa. U druhého video vstupu jsem jasovou složku uložil jako složku `R`. Po příjmu celého obrazového rámce jsem vykreslil výsledný obraz uložený v datových polích `BufferData1` a `BufferData2`.

Dodatečná funkce, která byla pro práci s anaglyfem vytvořena, je možnost posunu červené vrstvy v horizontálním i vertikálním směru. Tato funkcionalita je implementovaná v přerušení od DMA, kdy jsem získal jeden obrazový řádek. Řešení je takové, že jsem vytvořil proměnné, které obsahují požadovaný offset od jednotlivých stran displeje. Na základě těchto offsetů se vynechají některé pixely nebo se posune uložení do datového pole `BufferData1` nebo `BufferData2`.

Obr. 59. Anaglyf 3D.

7.8 Detekce bodu

Funkce detekce bodu je určena pro udržování zvoleného bodu uprostřed zorného pole kamery. Detekce je založena na porovnávání barvy. Pokud jsem v režimu detekce, tak po stisku do plochy obrazu, se uloží do proměnné `colorSelected` barva, která se nachází na souřadnici stisku. Současně je okolo této souřadnice vykreslen červený čtvereček.

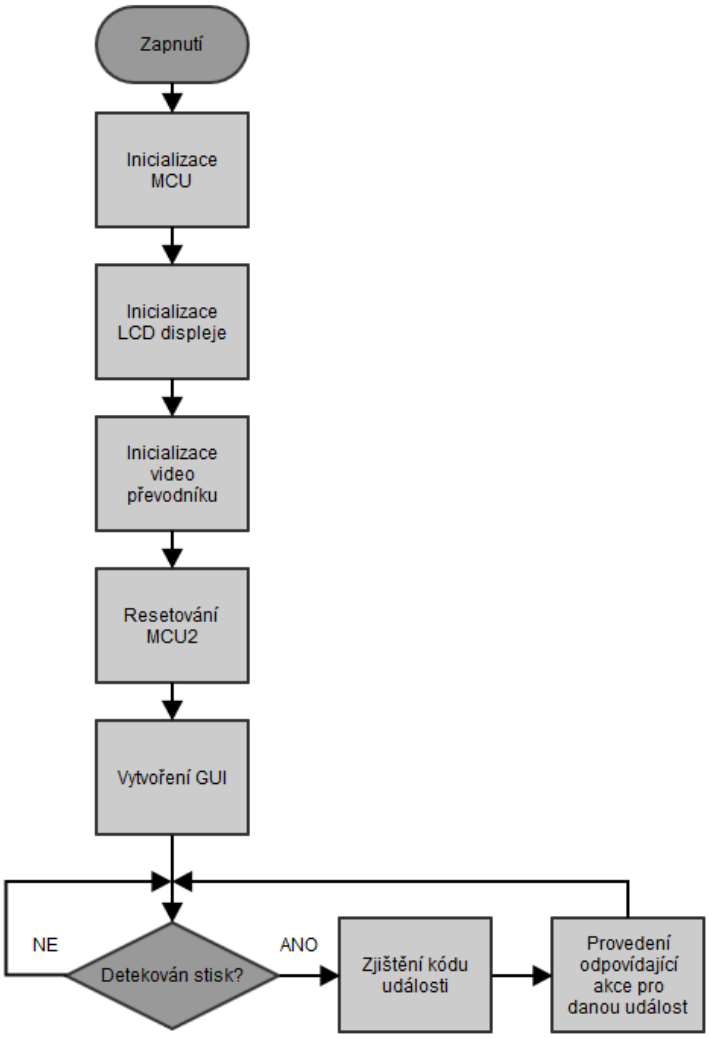
Pokud se tento čtvereček nachází vpravo od středu, tak se pošle příkaz k MCU2, aby se příslušný nástavec s kamerou pohyboval doprava. V případě, že se čtvereček nachází vlevo od středu, tak je k MCU2 poslán příkaz pro pohyb příslušného nástavce doleva. V případě, že je čtvereček ve vymezeném prostoru, ve středu displeje, nic se neděje.

Po zvolení barvy pro sledování, je aktivováno prohledávání. Prohledávání má tři fáze. V první fázi se prohledává v rámci čtverečku. Pokud se zde nenajde shoda v barvě, přičemž je zde určitá uživatelem definovaná tolerance rozdílu, tak se aktivuje další fáze. V druhé fázi se prohledává uživatelem definované okolí čtverečku. Pokud se najde shoda, dojde k překreslení čtverečku na nové souřadnice. V případě neshody se aktivuje třetí fáze. V poslední fázi se prohledává celý obraz. Pokud se nalezne shoda, tak dojde k přepočtu souřadnic čtverečku. V případě, že ani v celém obraze se již nenajde shoda, je čtvereček deaktivován, avšak stále probíhá prohledávání pro případ, že se po čase námi zvolená barva objeví v obraze. S přepočtem souřadnic čtverečku je spojeno i posílání příkazů pro MCU2, kdy podle umístění čtverečku na displeji se aktivuje příslušný pohyb nástavců, jak bylo popsáno výše.

Samotné srovnávání zvolené barvy není aplikováno na jeden pixel, ale na průměr oblasti 3 x 3 pixelů. Zde jsem to elegantně vyřešil tak, že v přerušení po přijetí celého obrazového rámce, kdy dochází k přepočtu na formát RGB 565, využiji datového pole `BufferData1`, kde poté co vykreslím obrazový pixel, mohu na toto místo uložit něco jiného. V mém případě zde budu ukládat průměry obrazových oblastí 3 x 3 pixelů, kde jednotlivé oblasti se překrývají pouze o jeden pixel. Stačí si vytvořit ukazatel a dodržovat správné posouvání v poli, a vždy po výpočtu nového pixelu ve formátu RGB 565 provést průměr barev. Tímto dostanu výsledné datové pole s jednotlivými průměrnými hodnotami oblastí 3 x 3 pixelů, na které mohu aplikovat prohledávání.

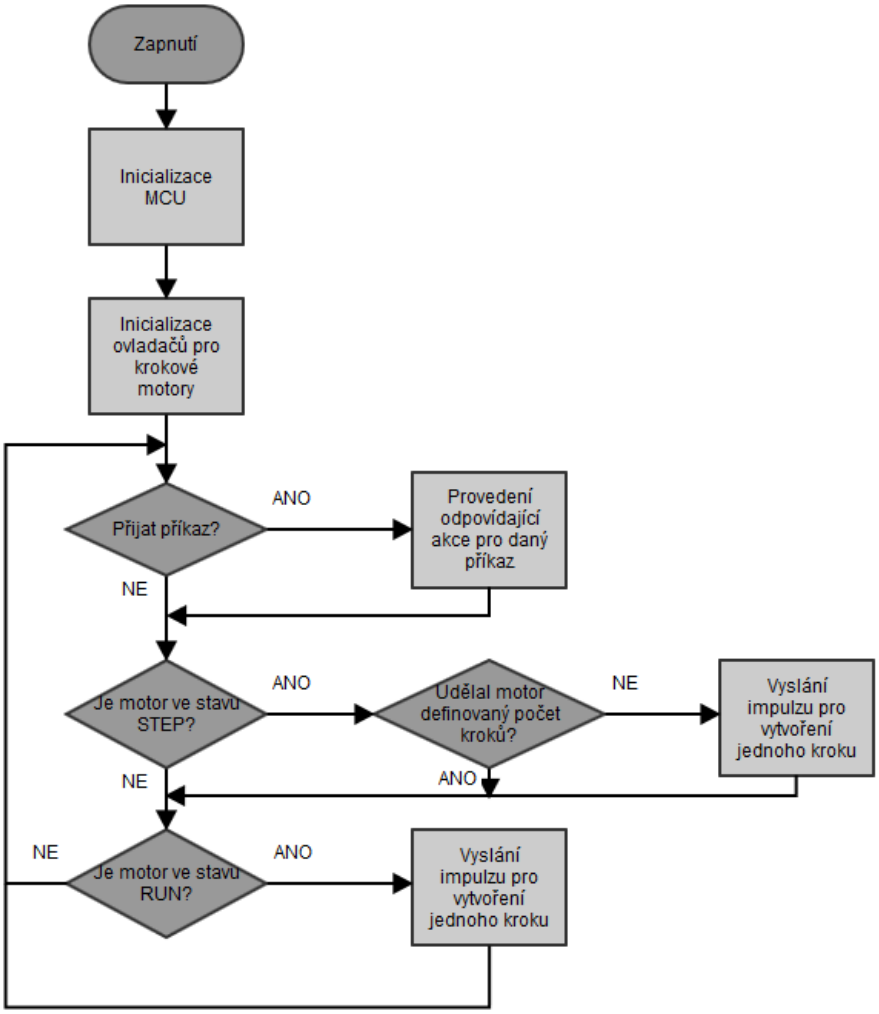
7.9 Struktura celého programu

První mikropočítač (MCU1) odpovídá za celou řadu funkcí, přičemž jeho hlavní programová smyčka zůstává jednoduchá. Poté co je MCU1 oživen, je provedena jeho inicializace, která zahrnuje nastavení systémových hodin, frekvence, GPIO pinů, rozhraní USART, rozhraní I^2C, AD převodníků, časovačů a jednotlivých přerušení. Následně je provedena inicializace LCD displeje. Poté následuje inicializace video převodníku a resetování MCU1. Ještě před tím, než se vstoupí do hlavní programové smyčky, je vytvořeno a uloženo celé GUI. Samotná hlavní programová smyčka se skládá z testování, zda byl detekován stisk. Pokud ano, tak se zjistí podle aktuálního pohledu, o jakou událost se jedná a získá se této události kód. Podle kódu události se vyvolá odpovídající akce.



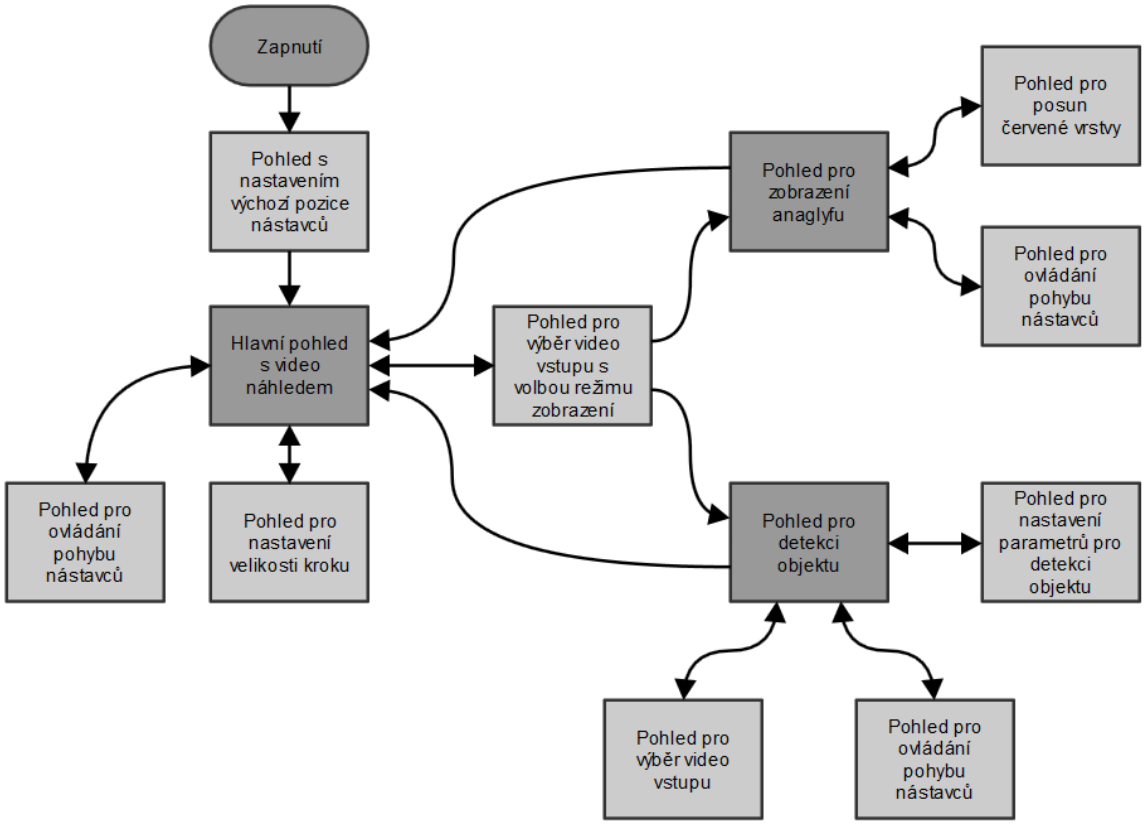
Obr. 60. Blokové schéma programu MCU1.

Druhý mikropočítač (MCU2) se stará pouze o ovládání krokových motorů. Poté co ho MCU1 uvolnění z resetu, dojde k inicializaci. Během inicializace se provede nastavení systémových hodin, frekvence, GPIO pinů, rozhraní USART, časovačů a přerušení. V další fázi se provede inicializace obou ovladačů pro krokové motory a motory se uvedou do režimu spánku. Po tomto následuje hlavní programová smyčka, ve které se po přijetí znaku od MCU1 vykoná odpovídající akce. Seznam těchto znaků s odpovídajícími akcemi, je uveden v tabulce (Tab. 4). V dalším kroku následuje testování, jestli se `motor1` nebo `motor2` nachází v režimu `STEP`. Pokud ano a přitom tento motor neučinil definovaný počet kroků, tak se vyšle impulz odpovídajícímu ovladači, aby následně krokový motor udělal jeden krok. V hlavní smyčce se nachází ještě jedno testování a to, zda je `motor1` nebo `motor2` v režimu `RUN`. Pokud ano, tak se vyšle impulz odpovídajícímu ovladači, aby následně krokový motor udělal jeden krok.

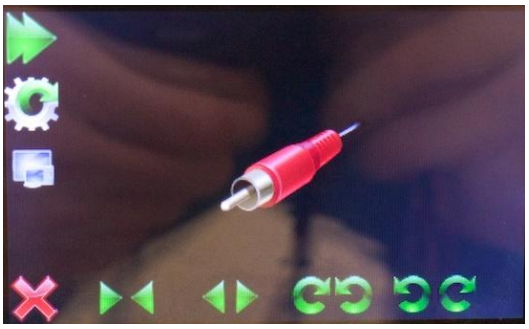


Obr. 61. Blokové schéma programu MCU2.

Pro celkové ovládání všech možností manipulátoru, bylo vytvořeno několik pohledů. Diagram propojenosti těchto pohledů je zobrazen na obrázku (Obr. 61).

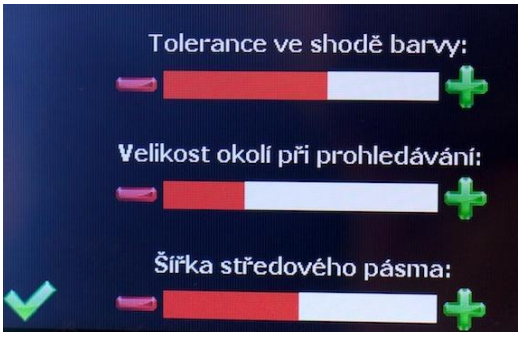


Obr. 62. Blokové schéma pohledů.



Obr. 63. Pohled pro ovládání pohybu nástavců.

Obr. 64. Pohled pro volbu video vstupu a režimu zobrazení.



Obr. 65. Pohled pro nastavení parametrů detekce bodu.

ZÁVĚR

Tato práce se zabývala tvorbou zařízení pro aplikaci stereovize. To zahrnovalo popsat v teoretické části, jaké existují typy krokových motorů a jakým způsobem lze tyto krokové motory ovládat. Součástí bylo uvedení příkladu integrovaných obvodů, které nám umožňují ovládání krokových motorů. V další kapitole byly probrány způsoby přenosu video signálu z kamer a následné možnosti zachytávání tohoto video signálu. Byly zmíněny tři způsoby. První způsob tvoří analogové video, kde jsem se zaměřil především na kompozitní video. Dalšími způsoby bylo HDMI a USB. U všech těchto způsobů jsem se zaměřil na základní principy těchto technik a zmínil jsem, jaké jsou potřeba dodatečné integrované obvody pro zachytávání takového video signálu s následným převodem do formátu vhodným pro mikropočítač. Poslední částí, kterou jsem rozebral v teoretické části, bylo uvedení možností, jak lze vzdáleně ovládat automatické ostření a spoušť u kamer. Byly popsány možnosti z oblasti drátového a bezdrátového ovládání.

Po shrnutí možností pro ovládání krokových motorů a zachytávání video signálu, jsem mohl přistoupit k praktické části. V této části jsem uvedl, jak byla řešena samotná konstrukce manipulátoru. Pro možnost ovládat tuto konstrukci, byla vytvořena řídicí deska. Samotnou tvorbu zahrnovalo, uvědomit si, jaké funkce by měla tato deska umožňovat. Po shrnutí všech požadovaných funkcí jsem přistoupil k návrhu hardwaru. Již zde se vyskytly problémy ve smyslu omezení na použitý typ součástek, kde bylo potřeba z důvodu umístění LCD displeje použít výhradně SMD součástky, nebo problém ve smyslu omezení na rozměry výsledné desky. Omezujícími rozměry byla výška, šířka a hloubka výsledné desky. Tyto rozměry byly pevně dány vymezeným prostorem v samotné konstrukci. Po volbě součástek byla navržena schémata a plošný spoj. U jednotlivých obvodových částí jsem se snažil popsat, jakým způsobem a s využitím jakých součástek byla provedena realizace.

Po návrhu hardwaru, bylo potřeba desku vyrobit a provést osazení s následným oživením. Oživení zahrnovalo vytvoření firmwaru. Firmware zahrnuje zprovoznění komunikace mezi oběma mikropočítači, vykreslování na LCD displej, ovládání video převodníku s následným zachytáváním kompozitního video signálu a samozřejmě také ovládání ovladačů pro krokové motory. V praktické části jsem se snažil popsat, jak byla provedena implementace zmíněných funkcí. Dodatečnými funkcemi, které byly implementovány, jsou zobrazení video obrazu z obou kamer jako anaglyf 3D nebo detekce zvoleného bodu. Při řešení firmwaru, jsem se setkal s celou řadou problémů, které nakonec

byly vyřešeny. Mezi největší problém lze jmenovat, jak provést přepočet obrazových pixelů z formátu YCbCr 4:2:2 na RGB 565, kde docházelo z časové náročnosti přepočtu ke ztrátě několika obrazových řádků. Problém byl vyřešen tak, že celý obraz byl uložen do interní paměti MCU. Zde ovšem také nastaly problémy, protože velikost celého obrazu přesahovala dostupnou paměť a navíc maximální alokovaný blok mohl mít velikost 65 KB. Nakonec tento problém byl docela elegantně vyřešen.

Díky této práci, jsem si mohl vyzkoušet práci na větším projektu, jehož výsledkem je funkční zařízení. Mohl jsem si vyzkoušet návrh složitějšího hardwaru a práci s mikropočítači, které mají 32bitový ARM procesor. Díky tomu jsem si mohl více prohloubit znalosti programování mikropočítačů v jazyce C.

Výsledné zařízení bylo přihlášeno do soutěže STOČ 2015, konané na FAI UTB ve Zlíně, kde v kategorii HW a SW aplikace bylo dosaženo 1. umístění.

Po shrnutí výsledné práce se domnívám, že všechny body zadání byly splněny.

SEZNAM POUŽITÉ LITERATURY

[1] NOVÁK, Petr. *Mobilní roboty: pohony, senzory, řízení*. Vyd. 1. Praha: BEN - technická literatura, 2004, 247 s. ISBN 80-7300-141-1.

[2] Krokový motor. *PohonnaTechnika.cz* [online]. 2007-2015 [cit. 2015-05-02]. Dostupné z: http://www.pohonnatechnika.cz/skola/motory/krokovy-motor

[3] ŘEZÁČ, Kamil. Krokové motory. *Robotika.cz* [online]. 2002 [cit. 2015-05-02]. Dostupné z: http://robotika.cz/articles/steppers/cs

[4] Základní principy krokových motorů. *E-Konstruktér* [online]. 2013 [cit. 2015-05-02]. Dostupné z: http://e-konstrukter.cz/novinka/zakladni-principy-krokovych-motoru

[5] JONES, Douglas. Control of Stepping Motors. *The University of Iowa* [online]. 1998 [cit. 2015-05-02]. Dostupné z: http://homepage.cs.uiowa.edu/~jones/step/

[6] RYDLO, Pavel. *Krokové motory a jejich řízení* [online]. 2000 [cit. 2015-05-02]. Dostupné z: https://www.pslib.cz/pe/skola/studijni_materialy/motory/krok_motor/krok_ucebnice.pdf

[7] KREJČÍ, Ondřej. *Realizace malého měniče pro krokové motory řízeného mikropočítačem* [online]. Brno, 2011 [cit. 2015-05-02]. Dostupné z: https://dspace.vutbr.cz/bitstream/handle/11012/94/BAKALARKA_FINAL.pdf. Bakalářská práce. VUT v Brně.

[8] Analog Video. *National Instruments* [online]. 2011 [cit. 2015-05-02]. Dostupné z: http://www.ni.com/white-paper/4750/en/

[9] Anatomy of a Video Signal. *National Instruments* [online]. 2006 [cit. 2015-05-02]. Dostupné z: http://www.ni.com/white-paper/3020/en/

[10] Video Basics. *Maxim Integrated* [online]. 2002 [cit. 2015-05-02]. Dostupné z: http://www.maximintegrated.com/en/app-notes/index.mvp/id/734

[11] TORRES, Gabriel. Inside HDMI. *HARDWARE Secrets* [online]. 2006 [cit. 2015-05-02]. Dostupné z: http://www.hardwaresecrets.com/article/Inside-HDMI-High-Definition-Multimedia-Interface/283/

[12] Video Display Signals. *Maxim Integrated* [online]. 2008 [cit. 2015-05-02]. Dostupné z: http://www.maximintegrated.com/en/app-notes/index.mvp/id/4306

[13] WILSON, Tracy. How HDMI Works. *HowStuffWorks.com* [online]. 2007 [cit. 2015-05-02]. Dostupné z: http://electronics.howstuffworks.com/hdmi.htm

[14] O´DONNELL, Bob. HDMI: The Digital Display Link. *HDMI.org* [online]. 2006 [cit. 2015-05-02]. Dostupné z: http://www.hdmi.org/pdf/whitepaper/SilicaonImageHDMIWhitePaperv73(2).pdf

[15] HDCP Specifications. *Digital Content Protection* [online]. 2015 [cit. 2015-05-02]. Dostupné z: http://www.digital-cp.com/hdcp_specifications

[16] Knowledge Base. *HDMI.org* [online]. 2003-2015 [cit. 2015-05-02]. Dostupné z: http://www.hdmi.org/learningcenter/kb.aspx

[17] Color Spaces. *Compression* [online]. [cit. 2015-05-02]. Dostupné z: http://compression.ru/download/articles/color_space/ch03.pdf

[18] TIŠNOVSKÝ, Pavel. Komunikační protokol USB. *ROOT.CZ* [online]. 2009 [cit. 2015-05-02]. Dostupné z: http://www.root.cz/clanky/komunikacni-protokol-universalni-seriove-sbernice/

[19] Triggering Multiple Cameras. *Breeze Systems* [online]. 2013 [cit. 2015-05-02]. Dostupné z: http://breezesys.com/MultiCamera/release.htm

[20] MAZUROV, Aleg. Digital Camera Control Using Arduino. *Circuits@Home* [online]. 2010 [cit. 2015-05-02]. Dostupné z: http://www.circuitsathome.com/camera-control/digital-camera-control-using-arduino-usb-host-shield-part-1-basics

[21] Methods to Trigger Off Camera Flash. *Scantips* [online]. 2009-2015 [cit. 2015-05-02]. Dostupné z: http://www.scantips.com/lights/trigger.html

[22] A4982. *Allegro MicroSystems* [online]. 2015 [cit. 2015-05-02]. Dostupné z: http://www.allegromicro.com/en/Products/Motor-Driver-And-Interface-ICs/Bipolar-Stepper-Motor-Drivers/A4982.aspx

[23] L6480. *ST.com* [online]. 2015 [cit. 2015-05-02]. Dostupné z: http://www.st.com/web/catalog/sense_power/FM142/CL851/SC1794/SS1498/LN1794/PF253950

[24] DRV8821. *Texas Instruments* [online]. 1995-2015 [cit. 2015-05-02]. Dostupné z: http://www.ti.com/product/DRV8821

[25] TVP5150AM1. *Texas Instruments* [online]. 1995-2015 [cit. 2015-05-02]. Dostupné z: http://www.ti.com/product/tvp5150am1

[26] http ADV7280. *Analog Devices* [online]. 1995-2015 [cit. 2015-05-02]. Dostupné z: http://www.analog.com/en/products/audio-video/video-decoders/adv7280.html

[27] ADV7842. *Analog Devices* [online]. 1995-2015 [cit. 2015-05-02]. Dostupné z: http://www.analog.com/en/products/audio-video/analoghdmidvi-interfaces/analog-hdmidvi-display-interfaces/adv7842.html

[28] ADV7619. *Analog Devices* [online]. 1995-2015 [cit. 2015-05-02]. Dostupné z: http://www.analog.com/en/products/audio-video/analoghdmidvi-interfaces/analog-hdmidvi-display-interfaces/adv7619.html

[29] Hybridní dvoufázové krokové motory řady SX. *Microcon* [online]. 2015 [cit. 2015-05-02]. Dostupné z: http://www.microcon.cz/pdf2015/13-20.pdf

[30] A4490. *Allegro MicroSystems* [online]. 2015 [cit. 2015-05-02]. Dostupné z: http://www.allegromicro.com/en/Products/Regulators-And-Lighting/Multiple-Output-Regulators/A4490.aspx

[31] STM32F407VE. *ST.com* [online]. 2015 [cit. 2015-05-02]. Dostupné z: http://www.st.com/web/catalog/mmc/FM141/SC1169/SS1577/LN11/PF252149

[32] STM32F100C4. *ST.com* [online]. 2015 [cit. 2015-05-02]. Dostupné z: http://www.st.com/web/catalog/mmc/FM141/SC1169/SS1031/LN775/PF216837

[33] TiWi-uB2. *LSR.com* [online]. 2012-2014 [cit. 2015-05-02]. Dostupné z: http://www.lsr.com/downloads/products/330-0100.pdf

[34] J-Link User Guide. *Segger* [online]. 2014 [cit. 2015-05-02]. Dostupné z: https://www.segger.com/cms/admin/uploads/productDocs/UM08001_JLink.pdf

[35] HX8352-A Datasheet. *Datasheet4U* [online]. 2008 [cit. 2015-05-02]. Dostupné z: http://www.datasheet4u.com/datasheet/H/X/8/HX8352-A_Himax.pdf.html

[36] ADC Example. *MicroBuilder.eu* [online]. 2010 [cit. 2015-05-02]. Dostupné z: http://www.microbuilder.eu/Tutorials/LPC2148/ADC/NDSTouchScreen.aspx

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

SPI	Serial Peripheral Interface
CVBS	Color Video Blanc Sync
RGB	Red, Green, Blue
HDMI	High-Definition Multimedia Interface
USB	Universal Serial Bus
RCA	Radio Corporation of America
HSYNC	Horizontal Sync
VSYNC	Vertical Sync
QAM	Quadrature Amplitude Modulation
PAL	Phase Alternating Line
NTSC	National Television System Committee
SECAM	Sequential Couleur Avec Memoire
IRE	Institute of Radio Engineers
ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
I^2C	Inter-Integrated Circuit
TMDS	Transition-Minimized Differential Signaling
CEC	Consumer Electronics Control
DDC	Display Data Channel
HDCP	High-bandwidth Digital Content Protection
EMI	Electromagnetic Interference
NRZI	Non Return to Zero Inverted
MTP	Media Transfer Protocol
PTP	Picture Transfer Protocol
RF	Radio Frequency

MCU	Microcontroller
LCD	Liquid-Crystal Display
SMD	Surface-Mount Device
DCMI	Digital Camera Interface
QFN	Quad-Flat No-leads package
ARM	Acorn RISC Machine architecture
RISC	Reduced Instruction Set Computing
DSP	Digital Signal Processing
FPU	Floating-Point Unit
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
UART	Universal Asynchronous Receiver/Transmitter
LQFP	Low profile Quad Flat Pack
GRAM	Graphics Random Access Memory
FSCM	Flexible Static Memory Controller
CMOS	Complementary Metal-Oxide-Semiconductor
OTG	On-The-Go
JTAG	Joint Test Action Group
SWD	Single Wire Debug
GPIO	General-Purpose Input/Output
GUI	Graphical User Interface
RLE	Run-Length Encoding
DMA	Direct Memory Access
HW	Hardware
SW	Software

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH