



Fakulta elektrotechnická
Katedra aplikované elektroniky a telekomunikací

DIPLOMOVÁ PRÁCE

Miniaturní kamerový modul s bezdrátovým přenosem obrazu

Autor práce: Bc. Adam Zachář
Vedoucí práce: Ing. Petr Weissar, Ph.D.

Plzeň 2014

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta elektrotechnická
Akademický rok: 2013/2014

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Adam ZACHAŘ**
Osobní číslo: **E12N0028P**
Studijní program: **N2612 Elektrotechnika a informatika**
Studijní obor: **Elektronika a aplikovaná informatika**
Název tématu: **Miniaturní kamerový modul s bezdrátovým přenosem obrazu**
Zadávací katedra: **Katedra aplikované elektroniky a telekomunikací**

Z á s a d y p r o v y p r a c o v á n í :

Navrhněte a realizujte modul s kamerou a bezdrátovým přenosem obrazu.

1. Vyberte vhodný kamerový modul vzhledem ke kvalitě obrazu, rozhraní a hlavně velikosti modulu.
2. Zvolte vhodný typ bezdrátového přenosu vzhledem k rychlosti, odolnosti rušení a zpoždění.
3. Realizujte vzorové řešení a navrhněte příp. i přijímací modul.



Rozsah grafických prací: podle doporučení vedoucího

Rozsah pracovní zprávy: 30 - 40 stran

Forma zpracování diplomové práce: tištěná/elektronická

Seznam odborné literatury:

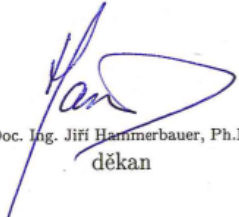
Student si vhodnou literaturu vyhledá v dostupných pramenech podle doporučení vedoucího práce.

Vedoucí diplomové práce: **Ing. Petr Weissar, Ph.D.**


Katedra aplikované elektroniky a telekomunikací

Datum zadání diplomové práce: **14. října 2013**

Termín odevzdání diplomové práce: **12. května 2014**


Doc. Ing. Jiří Hammerbauer, Ph.D.
děkan




Doc. Dr. Ing. Vjačeslav Georgiev
vedoucí katedry

V Plzni dne 14. října 2013

Abstrakt

Tato diplomová práce se zabývá návrhem a realizací bezdrátového kamerového modulu. Existují dva zásadně odlišné způsoby realizace - analogová a digitální realizace. Tato práce se zaměřuje na digitální realizaci - na snímání, zpracování a bezdrátový přenos v digitální podobě. Prvním klíčovým požadavkem byla volba vhodného kamerového modulu z hlediska kvality obrazu, rychlosti snímání a velikosti. Druhým klíčovým požadavkem byla volba vhodného bezdrátového modulu s dostatečnou přenosovou rychlostí a s minimálním zpožděním. Tyto požadavky byly splněny použitím kamerového modulu OV2640 s podporou výstupního formátu JPEG s rozlišením 2 Mpx, wifi modulu CC3000 s přenosovou rychlostí 6,8 Mb/s a procesoru STM32F407 s podporou kamerového rozhraní DCMI. Výsledné zařízení je tak schopné přenášet až 10 snímků za vteřinu v rozlišení 320x240 a se zpožděním řádově jednotek ms.

Klíčová slova

Bezdrátový kamerový modul, OV2640, CC3000, Smart config.

Abstract

Zachář, Adam. *Miniature camera module with wireless video transmission* [*Miniaturní kamerový modul s bezdrátovým přenosem obrazu*]. Pilsen, 2014. Master thesis (in Czech). University of West Bohemia. Faculty of Electrical Engineering. Department of Applied Electronics and Telecommunications. Supervisor: Petr Weissar

This thesis describes the design and implementation of a wireless camera module. There are two fundamentally different ways of implementation - analog and digital implementation. This thesis focuses on the digital implementation - on image capture, processing and wireless transmission in digital form. The first key requirement was the choice of a suitable camera module in terms of image quality, frame rate and size. The second key requirement was the choice of a suitable wireless module with sufficient data rate and with minimal delay. These requirements were met by using OV2640 camera module with support for JPEG output format with a resolution of 2 Mpx, wifi module CC3000 with a transmission rate of 6,8 Mb/s and STM32F407 processor with support for camera interface DCMI. The final device is able to transmit up to 10 frames per second with a resolution of 320x240 and a delay of the order of units of ms.

Keywords

Wireless camera module, OV2640, CC3000, Smart config.

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci, zpracovanou na závěr studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem svou závěrečnou práci vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 270 trestního zákona č. 40/2009 Sb.

Také prohlašuji, že veškerý software, použitý při řešení této diplomové práce, je legální.

V Plzni dne 9. května 2014

Bc. Adam Zachař

.....

Podpis

Obsah

Seznam obrázků	vii
Seznam tabulek	viii
Seznam symbolů a zkratek	ix
1 Úvod	1
2 Způsoby realizace	2
2.1 Analogové kamery	2
2.2 IP kamery	2
2.2.1 Raspberry Pi	3
3 Kamerový modul	6
3.1 Předpoklady pro výběr	6
3.2 Kamerový modul OV2640	7
3.2.1 Komunikace s OV2640	9
3.2.2 Zapojení kamery OV2640	10
4 Bezdrátový modul	11
4.1 Předpoklady pro výběr	11
4.2 Modul CC3000	12
4.2.1 Komunikace s CC3000	13
4.2.1.1 SPI	13
4.2.1.2 UART	15
4.2.1.3 I2C	15
4.2.2 Programování modulu CC3000	15
4.2.3 Smart Config CC3000	18
5 Procesor	20
5.1 Předpoklady pro výběr	20
5.2 Periferní zařízení a programování	22
5.2.1 CMSIS	23
5.2.2 DMA	25

5.2.3	DCMI	26
5.2.4	I2C	28
5.2.5	SPI	29
6	Praktická realizace	31
6.1	Vysílací modul	31
6.1.1	Napájení	32
6.1.2	Časování	32
6.1.3	Periférie	32
6.1.4	Hardwarové řešení	32
6.1.5	Popis programu vysílacího modulu	33
6.2	Přijímací modul	35
6.2.1	Stručný popis programu	35
7	Závěr	36
	Reference, použitá literatura	38
	Přílohy	41
A	Schémata zapojení	41
A.1	Schémata zapojení	41
B	Desky plošných spojů a seznam součástek	44
C	Použité skripty, zdrojové kódy	48

Seznam obrázků

2.1	Raspberry Pi Převzato z [19] 	3
2.2	Kamera pro Raspberry Pi Převzato z [20] 	4
3.1	Kamerový modul OV2640 + rozšiřující DPS Převzato z [3] 	7
3.2	Horizontální synchronizace Převzato z [1] 	9
3.3	SVGA časování Převzato z [1] 	10
4.1	Modul CC3000 Převzato z [16] 	12
4.2	Master-slave model SPI komunikace Převzato z [13] 	13
4.3	Záhlaví SPI Převzato z [13] 	14
4.4	Časování při prvním zápisu do modulu CC3000 Převzato z [13] 	14
4.5	Časování při zápisu do modulu CC3000 Převzato z [13] 	15
4.6	Model uart komunikace Převzato z [11] 	15
4.7	Organizace uživatelského rozhraní API	16
4.8	Smart config zdroj Převzato z [12] 	18
4.9	Program Smart Config pro chytré telefony	19
5.1	Rozložení pinů procesoru STM32f407	22
5.2	Struktura CMSIS Převzato z [7] 	23
5.3	DMA2 mapování Převzato z [6] 	25
5.4	Průběh I2C komunikace Převzato z [21] 	28
6.1	Blokové schéma zařízení	31
6.2	Funkční blokové schéma	34
6.3	Ukázka programu po příjem videa	35
A.1	Schéma zapojení modulu s kamerou OV2640	41
A.2	Schéma zapojení bezdrátového kamerového modulu 1. část.	42
A.3	Schéma zapojení bezdrátového kamerového modulu 2. část.	43
B.1	Deska plošného spoje bezdrátového kamerového modulu spodní vrstva, zvětšení 2,5x	44
B.2	Deska plošného spoje bezdrátového kamerového modulu vrchní vrstva, zvětšení 2,5x	45

B.3	Osazovací plánec desky plošného spoje vrchní vrstvy, zvětšení 2,5x	45
B.4	Osazovací plánec desky plošného spoje spodní vrstvy, zvětšení 2,5x	46
B.5	Umístění konstrukčních otvorů a rozměry DPS bezdrátového kamerového modulu.	46

Seznam tabulek

3.1	Porovnání technologií CCD a CMOS <small>[Převzato z [22]]</small>	6
3.2	Seznam pinů OV2640 a jejich význam	8
4.1	Přehled bezdrátových modulů a jejich vlastností	11
5.1	Základní parametry procesoru STM32F407	21
5.2	DCMI piny	26
B.1	Seznam součástek	47

Seznam symbolů a zkratek

FPS	Frame Per Second. Počet snímků za vteřinu.
JPEG	Joint Photographic Experts Group. Což je vlastně konsorcium, které tuto kompresi navrhlo.
CCD	Charge-Coupled Device. Zařízení s vázanými náboji (technologie kamerových čipů).
CMOS	Complementary Metal–Oxide–Semiconductor. Technologie výroby IO.
SCCB	Serial Camera Control Bus. Sériová kamerová řídicí sběrnice.
UART	Universal Asynchronous Receiver and Transmitter. Univerzální asynchronní přijímač a vysílač.
SPI	Serial Peripheral Interface. Sériové periferní rozhraní.
I2C	Inter-Integrated Circuit. Multi-masterová počítačová sériová sběrnice.
USB	Universal Serial Bus. Univerzální sériové rozhraní.
TCP/IP	Transmission Control Protocol/Internet Protocol. Primární přenosový protokol/protokol síťové vrstvy.
MCU	Microcontroller. Jednočipový počítač.
MAC	Media Access Control. Jedinečný identifikátor síťového zařízení.
SSID	Service Set Identifier. Jedinečný identifikátor každé bezdrátové (WiFi) sítě.
FPGA	Field Programmable Gate Array. Programovatelná hradlová pole.
MSB	Most Significant Bit. Nejvýznamnější bit.
DCMI	Digital CaMera Interface. Digitální kamerové rozhraní.
API	Application Programming Interface. Rozhraní pro programování aplikací.
DMA	Direct Memory Access. Přímý přístup do paměti.

1

Úvod

Cílem této diplomové práce je navrhnout miniaturní kamerový modul s bezdrátovým přenosem obrazu. Tento modul je určen pro model železnice, který slouží jako výuková pomůcka. Modul bude snímat obraz z pohledu strojvedoucího, takže si studenti budou moci lépe představit reálný obraz, který se strojvedoucímu naskýtá při řízení lokomotivy.

Existují dva zásadně odlišné způsoby řešení - analogová a digitální realizace. Tato práce se zaměřuje na digitální realizaci - snímání, zpracování a následný bezdrátový přenos v digitální podobě.

Navržené zařízení by mělo snímat obraz okolí v dostatečné kvalitě a s minimálním zpožděním přenést obraz do modulu přijímače. Nicméně je nutné zvolit kompromis mezi kvalitou obrazu, cenou a velikostí zařízení. Pro potřeby tohoto zařízení jsou rozhodujícími parametry velikost zařízení a cena, které tedy budou upřednostňovat před kvalitou obrazu.

Na trhu je dostupná celá řada miniaturních bezdrátových kamer s následujícími parametry :

- Rozlišení: 640x480
- Rychlost snímání videa (FPS): 30
- Snímač: CMOS
- Cena: 3000 - 6000 Kč

Cílem je co nejvíce se přiblížit těmto parametrům, nicméně největší důraz bude klade na velikost zařízení, a to i za cenu nižší rychlosti snímání videa.

2

Způsoby realizace

V této kapitole shrnuji možné způsoby realizace bezdrátového kamerového modulu. Existují dvě zásadně odlišné způsoby realizace - analogové (analogové kamery) a digitální (IP kamery) řešení. V této práci se zabývám především IP kamerovými systémy postavenými na mikrokontrolérech. V následujícím textu provedu srovnání těchto dvou technologií - jejich výhody, nevýhody a možnosti využití.

2.1 Analogové kamery

Tyto kamerové systémy jsou konstruovány tak, aby jejich výstup mohl být zobrazen na televizních monitorech, čemuž je přizpůsobeno rozlišení kamer v závislosti na použitém výstupním formátu (např. PAL). U PAL systému činí maximální velikost snímku 720x576 obrazových bodů (0,4 Mpx) a snímková frekvence 25 fps. Je také důležité poznamenat, že analogová kamera snímá obraz ve dvou cyklech, nejdříve liché a pak sudé půlsnímky. Tento proces snímání má za následek časový posun v jednom snímku, v případě pohybu obrazu pak dochází k rozmazání detailů. Snímání obrazové informace je zajištěno snímáním čipem a digitální signál je nutné převádět na analogový. To způsobuje zkreslení a snížení kvality obrazu. Analogové kamery se v dnešní době používají u leteckých modelů letadel a vrtulníků. Pomocí bezdrátových AV modulů, využívajících frekvenční pásmo 5,8 nebo 2,4 GHz, lze dosáhnout přenosu obrazu až na několik jednotek km.

2.2 IP kamery

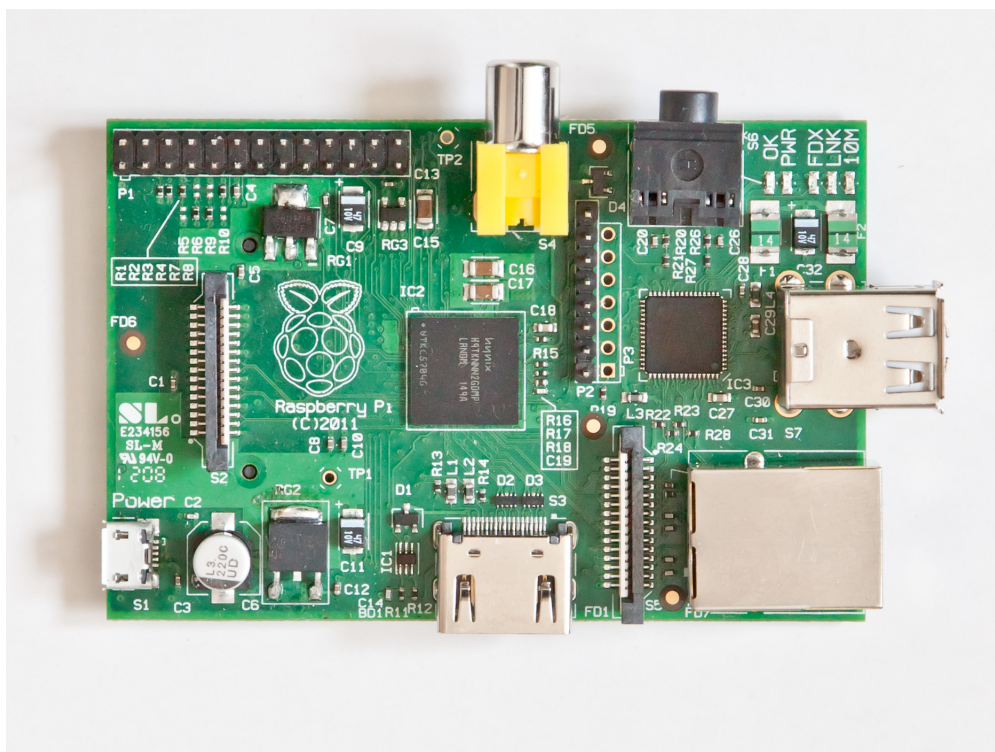
IP kamery jsou plně digitální a jejich rozlišení není nijak omezeno. Kvalita video obrazu je definována počtem odeslaných snímků za vteřinu a rozlišením snímku. IP kamery představují kamerové prvky pro připojení do síťové infrastruktury objektu. Každá kamera v síti má svou IP adresu, pomocí které je dosažitelná z internetu. Informace je přenášena v datové podobě v blocích podle pravidel IP protokolu. Snímková frekvence není omezena na 25 fps, jako je tomu u analogového kamerového systému. Může běžně dosahovat řádově

desítek snímků za vteřinu, což může být užitečné při sledování velmi rychlého pohybu. Tyto kamerové systémy se často používají v domácnostech a firmách jako bezpečnostní.

IP kamery lze realizovat mnoha způsoby, které se liší z hlediska způsobu zpracování obrazové informace a způsobu programování. IP kamerové moduly lze konstruovat na obvodech FPGA, mikrokontrolérech (zpravidla 32-bitové) nebo na procesorech s OS (např. Linux). V této práci se zabývám konstrukcí IP kamerového modulu na mikrokontroléru. Podrobný popis mého řešení je popsán v následujících kapitolách. Ještě před tím zde uvedu příklad jiného způsobu řešení IP kamery - na bázi OS Linux. Na trhu existuje celá řada miniaturních počítačů, pro tento příklad jsem si vybral Raspberry Pi.

2.2.1 Raspberry Pi

Raspberry Pi je jednodeskový počítač s deskou o velikosti zhruba platební karty. Vyvíjí ho britská nadace Raspberry Pi Foundation s cílem podpořit výuku informatiky ve školách. V době uvedení na trh se toto zařízení dle recenzí vyznačovalo dobrým poměrem cena/výkon. Výrobce nabízí k počítači jako operační systémy ARMové verze linuxových distribucí Debian a Arch. [wikipedia]



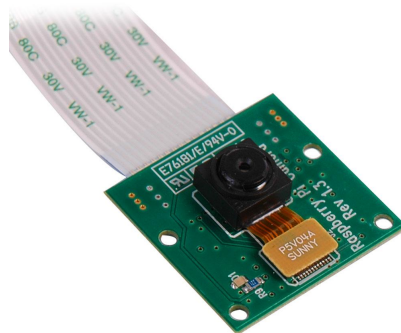
Obr. 2.1: Raspberry Pi |Převzato z [19]|

Základní parametry Raspberry PI Model A :

- Procesor – ARM1176JZF-S z rodiny ARM11 taktovaný na 700 Mhz
- Grafický procesor – VideoCore IV, podporující OpenGL, 1080p30, MPEG-4
- Operační paměť – 256 MB
- Obrazový výstup – Composite RCA, HDMI, DSI
- Zvukový výstup – 3,5 mm konektor, HDMI
- 8xGPIO, sběrnici UART, I2C, SPI
- slot pro SD nebo MMC kartu

K Raspberry PI je možné dokoupit kameru s rozlišením 5Mpx. Kamera je schopná pořizovat snímky v rozlišení 2592 x 1944 a natáčet video v rozlišení:

- 1080p na 30fps
- 720p na 60fps
- 640*480p na 60/90 fps



Obr. 2.2: Kamera pro Raspberry Pi |Převzato z [20]|

Jedná se o velmi malý kamerový modul o rozměrech 20mm x 25mm x 9mm připojený pomocí 15žilového CSI (Camera Serial Interface) kabelu.

Pomocí těchto modulů lze efektivně a poměrně levně zkonstruovat vlastní kamerový systém. Počítač se programuje přímo v Linuxu, kde je předinstalován programovací jazyk Python. Na internetu existuje celá řada návodů, v nichž jsou velmi detailně popsány a komentovány jednotlivé softwarové projekty. Například zprovoznění portů GPIO, které mohou být využity například k řízení servo motorů pro flexibilní natáčení kamery. Raspberry PI obsahuje dva základní příkazy pro práci s touto kamerou, které mohou obsahovat parametry - velikost, kvalita a formát uložení:

- `raspivid` – vytvoří video
- `raspistill` – vytvoří fotku

Přehled základních parametrů funkcí:

- `width, -w` – šířka obrazu (64 - 1920)
- `height, -h` – výška obrazu (64 - 1080)
- `bitrate, -b` – datový tok
- `output, -o` – jméno výstupního souboru
- `timeout, -t` – doba snímání obrazu
- `framerate, -fps` – počet snímků za vteřinu (2 - 30)

Rád bych zde demonstroval jednoduchost nastavení kamery a vytvoření RTSP streamu.

Před spuštěním konfiguračního rozhraní je vhodné provést aktualizaci systému:

- `sudo apt-get update`
- `sudo apt-get upgrade`

Dalším krokem je povolení kamery v konfiguračním nastavení, které spustíme pomocí příkazu:

- `sudo raspi-config`

Závěrem postačí zadat následující příkaz, který zahájí streamování videa pomocí VLC Media Playeru, a na přijímací straně zadat následující URL adresu :

```
rtsp://<IPadresaRaspberry>:1234/.
```

- `raspivid -o - -t 0 |cvlc -vvv stream:///dev/stdin --sout '#rtp{sdp=rtsp://:1234/}' :demux=h264`

, kde

- `o - -` – výstup má být zapsán do stdout
- `t 0` – vypnutí časového limitu
- `vvv` – specifikace zdroje signálu
- `sout` – specifikace výstupu

Značnou výhodou tohoto způsobu realizace je vysoká kvalita obrazu. Velkým nedostatkem je ale zároveň značná velikost zařízení.

3

Kamerový modul

V této kapitole shrnuji všechna kritéria, na základě kterých jsem vybíral vhodný kamerový modul. Dále detailně popisuji vybraný obrazový senzor.

3.1 Předpoklady pro výběr

Z hlediska využití jsou zásadní následně diskutovaná kritéria. Kamera bude sloužit pro snímání obrazu modelové železnice z pohledu strojvedoucího, z čehož vyplývají první dvě klíčová kritéria - velikost a cena zařízení. Vzhledem k bezdrátovému přenosu obrazu jsme limitováni přenosovou rychlostí bezdrátového modulu. Z tohoto důvodu nemá smysl upřednostňovat kvalitu snímaného obrazu nad ostatními parametry. Nicméně třetím klíčovým kritériem je schopnost kamerového modulu poskytovat kompresní výstupní formát (např. JPEG).

Je také nutné poznamenat, že snímací čipy v kamerách jsou konstruovány na technologii CCD nebo CMOS, které se liší svými vlastnostmi. Nebudu zde popisovat principy jednotlivých technologií, uvedu jen shrnutí důležitých parametrů.

Snímací čip	CCD	CMOS
Cena	vysoká	nízká
Rozměry řešení	vyšší	nízké
Spotřeba	vysoká	nízká
Kvalita obrazu	vysoká	nižší až nízká
Rozlišení	vysoké	střední
Digitální šum	nízký	vysoký
Rychlost	nižší až vysoká	vysoká
Dynamický rozsah	vysoký	nižší

Tab. 3.1: Porovnání technologií CCD a CMOS |Převzato z [22]|

Z výše uvedené tabulky 3.1 plyne, že technologie CMOS lépe odpovídá výše jmenovaným kritériím, a proto se dále budu věnovat pouze sensorům založeným na technologii

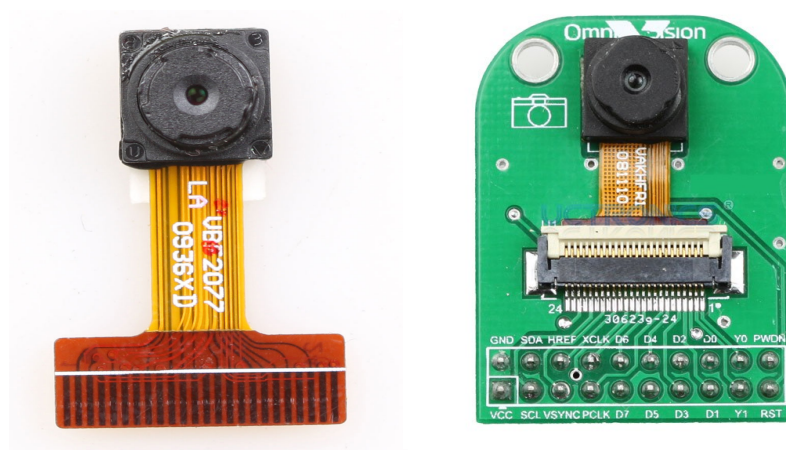
CMOS. Na trhu najdeme řadu výrobců CMOS kamerových modulů, mezi nejznámější patří společnosti OmniVision, Toshiba, Micron a Aptina. Osobně jsem zvolil kamerový modul od firmy OmniVision, jelikož nabízejí celou škálu modulů s rozlišením od VGA až po 16 Mpx kamery. Tyto snímací čipy se v dnešní době používají prakticky ve většině mobilních telefonů, notebooků, webových kamer, najdou uplatnění i v automobilovém průmyslu a v zabezpečovací technice. Mnou vybraný modul OV2640 představuje nejlevnější verzi s podporou kompresního formátu JPEG s rozlišením 2 Mpx.

3.2 Kamerový modul OV2640

Kamerový modul OV2640 je nízkonapěťový obrazový senzor od firmy OmniVision s nízkou spotřebou řádově desítky mW, který je zapouzdřen v malém BGA pouzdře. Senzor poskytuje maximální rozlišení UXGA (1632x1232) a obsahuje na jednom čipu jak kameru, tak i obrazový procesor pro předzpracování dat, díky kterému podporuje mnoho obrazových funkcí. Například:

- AGC (Automatic Gain control) – automaticky nastavuje zesílení, a tím zvyšuje citlivost kamery při nízké úrovni osvětlení.
- AEC (Automatic Exposure control) – automatické řízení expozice.
- AWB (Automatic White Balance) – automatické vyvážení bílé.
- Gamma korekce – programovatelná.

Řízení senzoru se provádí přes SCCB rozhraní (Serial Camera Control Bus), které je kompatibilní s I2C rozhraním (popsané v kapitole 3.2.1.).



Obr. 3.1: Kamerový modul OV2640 + rozšiřující DPS |Převzato z [3]|

Základní parametry OV2640 :

- Maximální rozlišení 1600x1200 UXGA
- Teplotní rozsah 0°C až 50°C
- Velikost čočky 1/4"
- Výstupní formáty (8-bit)
 - YUV422/420, YCbCr422
 - RGB565/555
 - 8-bit kompresní formát JPEG
 - 8/10-bit RAW RGB data
- Maximální snímková frekvence
 - UXGA 15fps
 - SVGA 30fps
 - CIF 60fps
- Velikost pixelu 2,2 μm x 2,2 μm
- Velikost pouzdra 5725 μm x 6285 μm

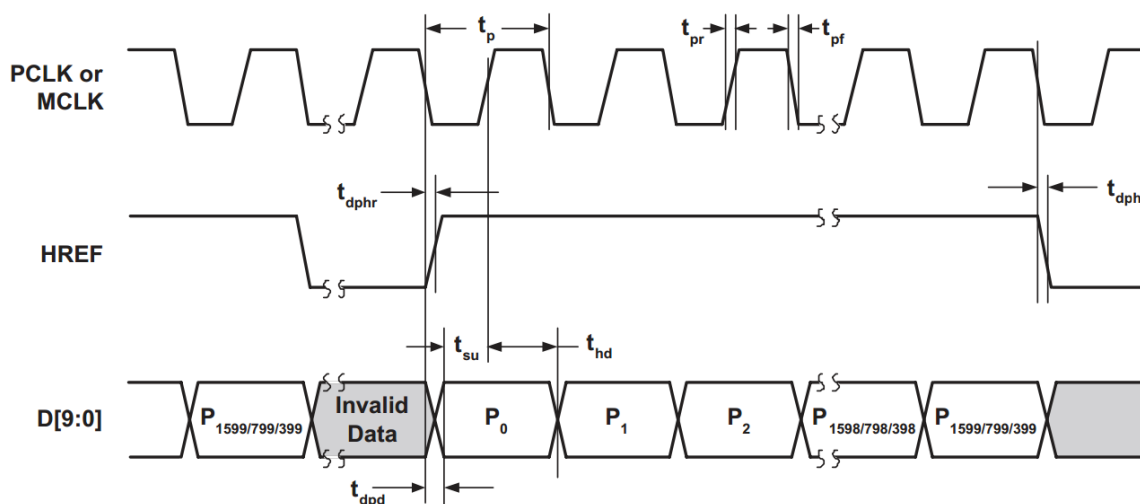
Pin	Typ	Popis
VDD	napájení	+3,3 V
GND	napájení	zem
SDIOC	vstup	SCCB hodiny
SDIOD	vstup/výstup	SCCB data
VSYNC	výstup	vertikální synchronizace
HREF	výstup	horizontální synchronizace
PCLK	výstup	Pixel clock
XCLK	vstup	systémové hodiny
D0-D7	výstup	paralelní výstup videa
RESET	vstup	reset(aktivní log. 0)
PWD	vstup	vypnutí napájení (aktivní log. 1)

Tab. 3.2: Seznam pinů OV2640 a jejich význam

3.2.1 Komunikace s OV2640

Senzor OV2640, jako jiné obrazové senzory od firmy OmniVision, podporuje komunikaci přes SCCB (Serial Camera Control Bus). Tato sériová sběrnice je hardwarově kompatibilní s I2C s maximální taktovací frekvencí 400 kHz. Důležitým parametrem pro nastavení registrů senzoru je samotná adresa zařízení na I2C sběrnici. Pro čip OV2640 jsou to adresy 0x60 pro zápis a 0x61 pro čtení. Senzor obsahuje kolem 100 osmibitových registrů, které slouží pro nastavení výstupního formátu, rozlišení, časování a identifikaci čipu. Data se přenášejí po osmibitové sběrnici spolu s dalšími pomocnými signály, jako jsou:

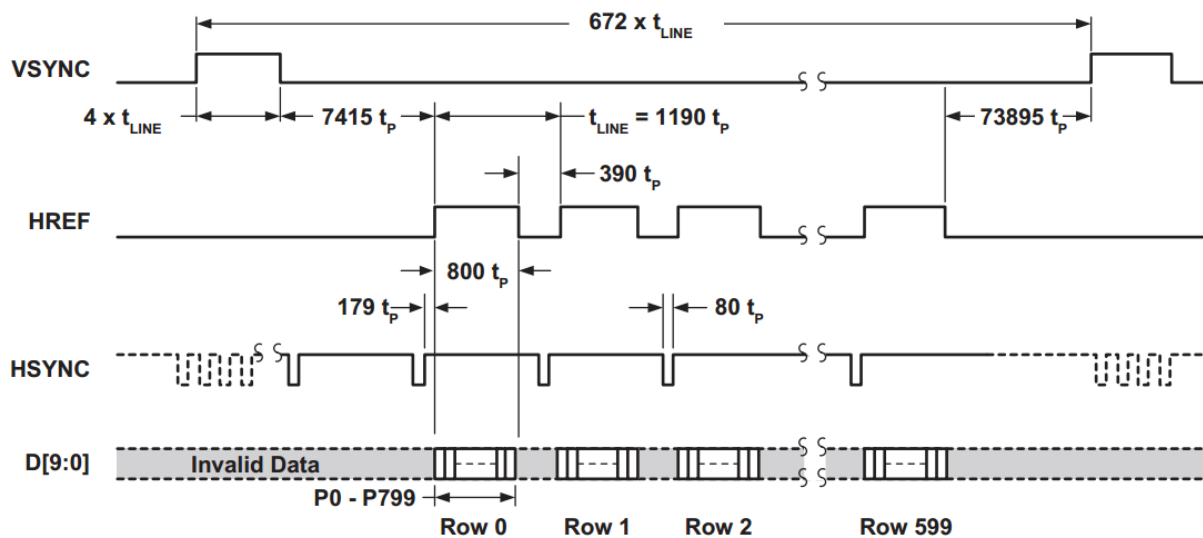
- HREF – horizontální synchronizace, označuje začátek řádku.
- VSYNC – vertikální synchronizace, označuje začátek snímku.
- PCLK – označuje taktovací frekvenci jednotlivých bytů.



Obr. 3.2: Horizontální synchronizace [Převzato z [1]]

Na obrázku 3.2 senzor OV2640 posílá data v paralelním synchronním formátu. Nejprve je potřeba připojit zdroj hodinového signálu k pinu XCLK, přičemž frekvence hodinového signálu podle katalogového listu činí obvykle 24 MHz. Po přivedení hodinového signálu dojde ke generování řídicích signálů VSYNC a HREF a ke generování dat D0 až D7. Z časového diagramu horizontální synchronizace lze za prvé vidět, že data D0 až D7 jsou vzorkována na náběžnou hranu signálu PCLK. Za druhé je možné vidět, že data jsou vzorkována, pouze v případě, že signál HREF je ve stavu logická "1". Signál HREF má význam horizontální synchronizace a určuje začátek a konec řádku. Je důležité si uvědomit, že jeden byte neodpovídá hodnotě právě jednoho pixelu. Počet bytů na pixel závisí na zvoleném výstupním formátu. Ve výchozím nastavení je formát YCbCr422, což znamená, že v průměru budu potřeba dva byty na jeden pixel.

Na obr 3.3 lze vidět časování při čtení snímku ve formátu SVGA (800x600). Při vysoké úrovni signálu HSYNC (log. 1) musíme zachytit 800 pixelů, což odpovídá jednomu řádku.



Obr. 3.3: SVGA časování [Převzato z [1]]

600 řádků, odpovídajících jednomu snímku, je zachyceno během nízké úrovně signálu VSYNC (log. 0). To znamená, že sestupnou hranou signálu VSYNC je signalizován začátek snímku a náběžnou hranou konec snímku. Je také nutné znát snímkovou frekvenci. Ve výchozím stavu hodiny PCLK mají stejnou frekvenci jako hodiny XCLK. Nicméně je možné pomocí děliček a fázového závěsu PLL změnit frekvenci hodin PCLK, a tak změnit snímkovou frekvenci. Fázový závěs je možné nastavit pomocí SCCB rozhraní. Snímková frekvence je také závislá na použitém výstupním formátu, jako je např. UXGA, SVGA, VGA a CIF.

3.2.2 Zapojení kamery OV2640

Čip OV2640 potřebuje pro svoji správnou funkci dvojí napájení. Pro napájení analogové části a I/O pinů je potřeba zdroj napětí o hodnotě 2,8 V a pro napájení samotného jádra čipu 1,8 V. Tato dvě napětí jsou získána pomocí napěťového regulátoru XC6206. Pro správnou funkci celého obvodu je potřeba připojit zdroj externích hodin na pin XCLK (viz. tabulka 3.2). Podle katalogového listu bývá jeho typická hodnota 24 MHz. Dále je nutné vybrat vhodný oscilátor s dobou náběhu a doběhu (t_r , t_f) menší jak 5 ns. Jako oscilátor jsem zvolil FXO-HC536R-24 na technologii CMOS s dobou náběhu a doběhu (t_r , t_f) menší jak 3 ns. Důležitým parametrem u CMOS oscilátorů je výstupní kapacitní zatížení, které činí 15 pF. Jelikož kamerový modul má na vstupním pinu XCLK kapacitu 10 pF, je nutné umístit oscilátor co nejbližší k čipu OV2640, a předejít tak chybám způsobeným špatným časováním. Schéma zapojení kamerového modulu viz. příloha A.

4

Bezdrátový modul

V této kapitole uvádím možné způsoby digitálního bezdrátového přenosu obrazu - stručný přehled dostupných bezdrátových modulů. Následně detailně popisuji jeden vybraný.

4.1 Předpoklady pro výběr

Pro výběr vhodného bezdrátového modulu jsou zásadní následující kritéria. Za prvé je žádoucí, aby modul disponoval dostatečnou přenosovou rychlostí. Je ovšem nutné stanovit, co považuji za dostatečnou přenosovou rychlost. Předpokládám, že budu přenášet snímky ve formátu JPEG z rozlišením 640x480 a se snímkovou frekvencí aspoň 10 fps. Velikost jednoho JPEG obrázku v rozlišení VGA je přibližně 100KB v závislosti na kvalitě kompresního procesu. Z toho vyplývá, že je potřeba, aby bezdrátový modul byl schopen přenášet data rychlostí až 8 Mb/s. Níže uvedená tabulka srovnává přenosové rychlosti známých bezdrátových přenosových technologií.

	Přenosová rychlost (kbit/s)	Komunikační dosah v řádech (m)
GPRS/GSM	64-128	1000 a více
Bluetooth	100-3000	10-100
ZigBee	10-250	10-100
RF	1-2000	10-100
Wifi	11 000 a více	10-100

Tab. 4.1: Přehled bezdrátových modulů a jejich vlastností

Se srovnáním výše uvedeného požadavku na přenosovou rychlost a přehledové tabulky vyplývá, že se dále budu věnovat pouze wifi modulům, jelikož jen tyto moduly dosahují dostatečné přenosové rychlosti.

Mezi nejznámější výrobce těchto wifi modulů patří:

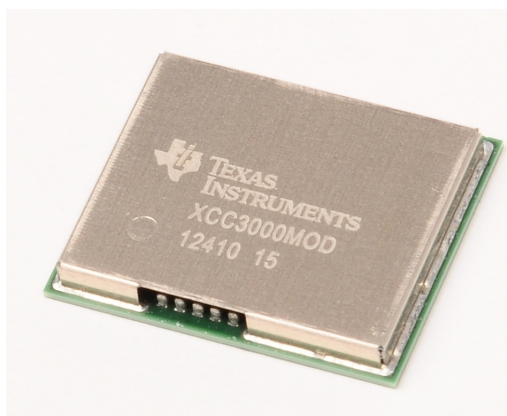
- WIZnet – WizFi210, WizFi220
- Texas Instruments – CC3000

- Microchip – RN171
- Lantronix – xPico Wifi
- Connect One – Nano Socket iWiFi, Nano WiReach

Při výběru vhodného bezdrátového wifi modulu nicméně není nejdůležitějším parametrem samotná maximální přenosová rychlost, ale komunikační rychlost s hostitelským mikrokontrolérem. Moduly podporují řadu rozhraní, jako např. UART, SPI, I2C či USB. Moduly jen s UART rozhraním jsou pro moji aplikaci očividně nedostačující vzhledem k jejich maximální komunikační rychlosti 3 Mb/s. Wifi moduly disponují mnoha funkcemi, jako je integrovaný TCP/IP stack, web server, AP (access point), a podporují mnoho protokolů. Pro bezdrátový přenos videa není nutné, aby modul obsahoval vlastní web server, musí však podporovat protokoly, jako jsou TCP, UDP a DHCP klientské pro snadné připojení do lokální sítě. Na základě těchto kritérií jsem zvolil wifi modul CC3000 od firmy Texas Instruments, který komunikuje přes SPI rozhraní s hodinovým kmitočtem až 16 MHz.

4.2 Modul CC3000

Modul CC3000 od firmy Texas Instruments je založen na standardu bezdrátových lokálních sítích IEEE 802.11b a 802.11g. Oba tyto nejčastěji se vyskytující standardy definují bezdrátové sítě pracující v pásmu 2,4 GHz. Liší se maximální dosažitelnou rychlostí (u standardu 802.11b činí nejvyšší dosažitelná rychlost 11 Mb/s, u 802.11g pak až 54 Mb/s). CC3000 podporuje všechny režimy zabezpečení sítě, jako je WEP, WPA a WPA2. Umožňuje rychlé automatické připojení k AP bez zásahu uživatele díky sedmi konfigurovatelným profilům, které jsou uloženy v EEPROM paměti. Dále podporuje funkci "Smart Config", což umožňuje připojit zařízení k WLAN síti pomocí tabletu, smartphonu nebo počítače.



Obr. 4.1: Modul CC3000 [Převzato z [16]]

Základní vlastnosti:

- Integrovaný TCP/IP stack, který umožňuje jednoduché internetové připojení s jakýmkoliv MCU.
- Podpora čtyř současně běžících síťových portů TCP (~4Mb/s) UDP (~7Mb/s) nebo UDP.
- Vestavěné síťové protokoly: ARP, ICMP, DHCP klient a DNS klient umožňující snadné připojení do sítě.
- Podporuje čtyřvodičové sériové periferní rozhraní SPI s taktem hodin až 16 MHz.

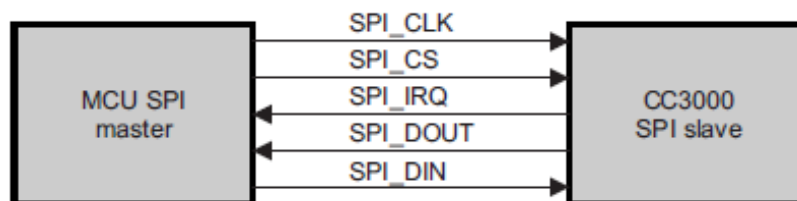
Je vhodné poznamenat, že modul CC3000 obsahuje integrovanou EEPROM paměť, ve které je uložen firmware, konfigurace sítě a MAC adresa zařízení. Paměť EEPROM je programovatelná pomocí I2C nebo přes API rozhraní.

4.2.1 Komunikace s CC3000

Modul CC3000 obsahuje tři různá periferní rozhraní SPI, UART a I2C. Níže uvádím důležité vlastnosti a využití těchto rozhraní v modulu CC3000.

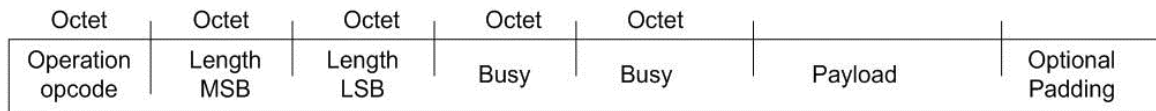
4.2.1.1 SPI

Jedná se o primární rozhraní pro komunikaci s hostitelským MCU, komunikující pomocí pěti signálů. Master-slave komunikační model je zobrazen na obrázku 4.2. Maximální frekvence hodin SPI_CLK je 16MHz.



Obr. 4.2: Master-slave model SPI komunikace |Převzato z [13]|

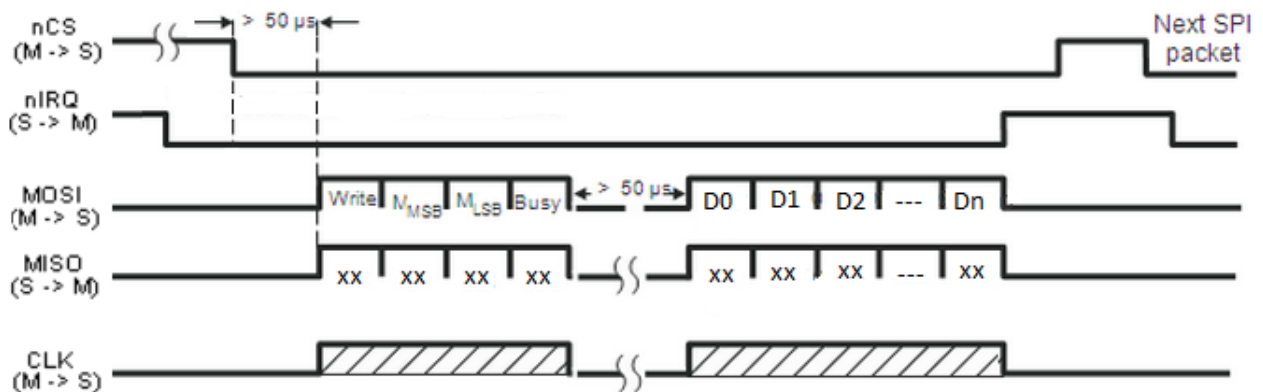
SPI protokol se používá pro komunikaci s modulem CC3000, který je řízen z hostitelského MCU, chovající se jako master zařízení, zatímco CC3000 se chová jako slave. Nejvýznamnější, a tudíž i první, bit je MSB. Nastavení hodin a fáze pro SPI je konfigurováno tak, že data jsou vzorkována na sestupnou hranu hodinového signálu. Z tohoto důvodu je CPOL (polarita hodinového signálu) nastavena na "0" a CPHA (fáze hodinového signálu) na "1". Je důležité, aby všechna odeslaná nebo přijatá data přes SPI rozhraní měla 16-bitovou délku. Každý paket, který je předáván na SPI sběrnici, se skládá ze záhlaví 5 bytů. Za ním následují uživatelská data, která se případně doplňují padding (výplň) bytem, aby celá délka byla zarovnána na 16-bitů, jak je vidět na následujícím obrázku.



Obr. 4.3: Záhlaví SPI [Převzato z [13]]

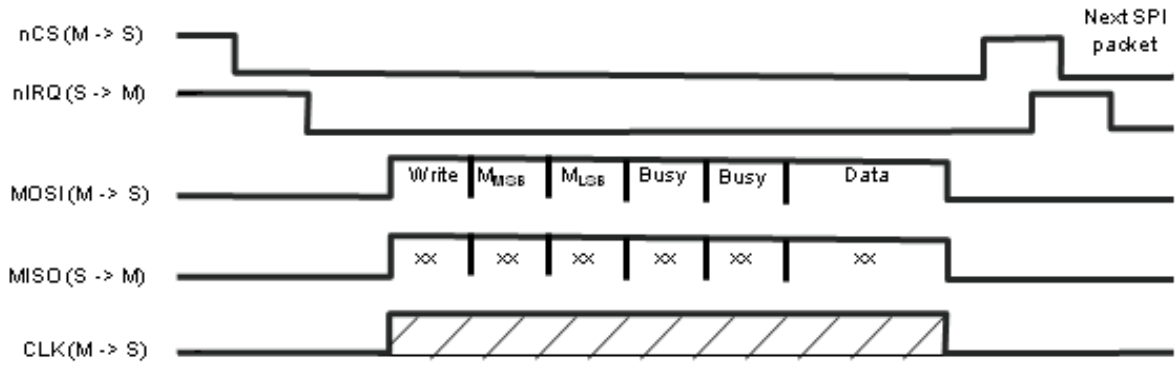
Prvním bytem hlavičky je operační kód Opcode (čtení nebo zápis), následován dvěma byty, které určují délku užitečných dat, a Padding bytem, který slouží jako zarovnávací byte. Hlavička je uzavřena dvěma Busy byty.

Je důležité zmínit, že první zápis do modulu CC3000 má trochu jiné časování než ostatní, jak lze vidět porovnáním obrázku 4.4 a 4.5. V prvním případě master zařízení (MCU) detekuje nízkou úroveň IRQ signálu, kde v tomto případě detekce IRQ neznamená, že zařízení je připraveno komunikovat s mastrem, ale že je připraveno po náběhu napájení. Master zařízení povolí signál nCS (not Chip Select) a zavede zpoždění nejméně 50 μ s před zahájením samotného přenosu dat. Jako první se vysílá hlavička SPI 4 byty a je vloženo další zpoždění 50 μ s, po kterém je vysílán zbytek paketu.



Obr. 4.4: Časování při prvním zápisu do modulu CC3000 [Převzato z [13]]

V ostatních případech komunikace probíhá následujícím způsobem. Nejdříve master zařízení povolí nCS a čeká na IRQ signál, který je generován obvodem CC3000 a informuje o připravenosti přijímat data. Zápis se opět skládá z hlavičky - tentokrát 5 bytů. Následně jsou vysílána data, která jsou doplněna padding bytem, aby celkový paket byl zarovnan na 16-bitů. Po posledním bytu přenášených dat master uvolní nCS a poté rovněž CC3000 uvolní signál IRQ, jak lze vidět na obr. 4.5.

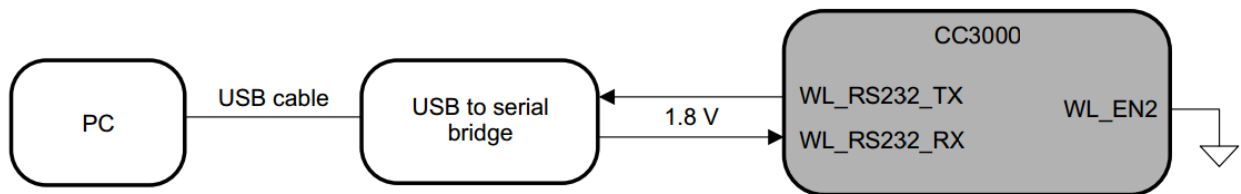


Obr. 4.5: Časování při zápisu do modulu CC3000 [Převzato z [13]]

Čtení je zahájeno přijetím IRQ signálu generovaného obvodem CC3000. Následně master zařízení povolí nCS a a vyšle 3 byty, které obsahují operační kód (čtení). Poté dojde k přijímání dat z CC3000, kde v prvních dvou bytech je obsažena délka dat. Po posledním přijatém bytu dojde k uvolnění nCS (MCU) a IRQ (CC3000). [13]

4.2.1.2 UART

Toto jednoúčelové WLAN sériové rozhraní slouží k připojení modulu CC3000 k PC, na kterém je spuštěn testovací nástroj, který slouží k ladění během fáze vývoje a testování. Je však potřeba přepnout CC3000 do testovacího režimu připojením pinu WL_EN2 na zem. Je také důležité zmínit, že napěťová úroveň signálů TX a RX je neobvyklých 1,8 V.



Obr. 4.6: Model uart komunikace [Převzato z [11]]

4.2.1.3 I2C

I2C rozhraní slouží jako jedna z možností pro aktualizaci firmware, který je uložen v EEPROM paměti. Tato paměť dále obsahuje informace o konfiguraci sítě a MAC adresu.

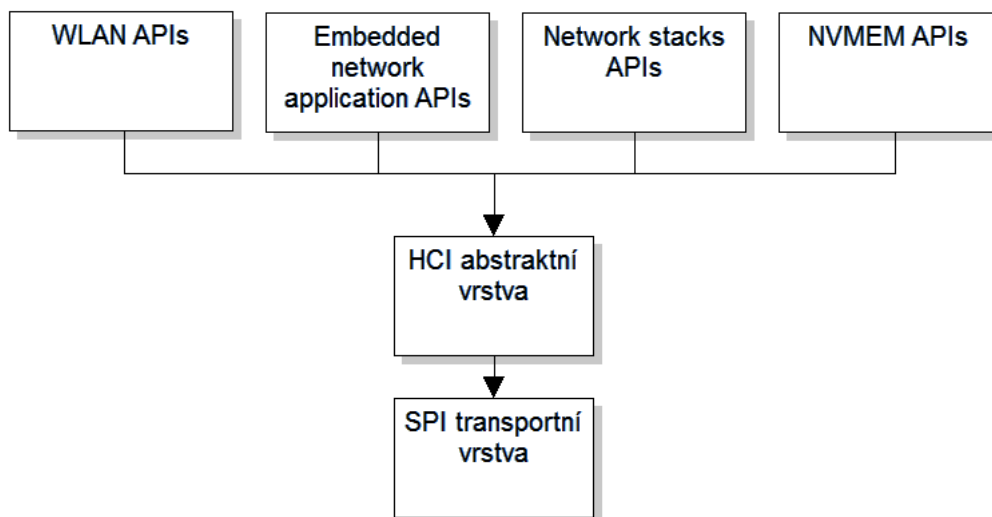
4.2.2 Programování modulu CC3000

K modulu CC3000 je dodáván softwarový balíček, obsahující sadu knihoven s cílem usnadnit a urychlit vývoj celého zařízení. Soubor knihoven pro modul CC3000 je rozdělen na dva hlavní celky:

- CC3000 uživatelské aplikační programové rozhraní (APIs)
- transportní vrstva API

Aplikační programové rozhraní API představuje vzájemnou interakci mezi modulem CC3000 a koncovým uživatelem. K interakci s přístrojem dochází přes SPI transportní vrstvu. API se nachází na vrcholu celé struktury, která je zapouzdřená vrstvou HCI (Host Controller Interface). Uživatelské rozhraní API je organizováno do čtyř různých entit :

- **WLAN APIs** – entita pro implementaci protokolu 802.11.
- **Network stacks APIs** – funkce API BSD soketu (Berkeley socket).
- **Embedded network application APIs** – zahrnuje základní síťové aplikace (Ping, DNS).
- **NVMEM APIs** – Nonvolatile memory, slouží pro přístup k EEPROM paměti, kde je uložena konfigurace zařízení CC3000.



Obr. 4.7: Organizace uživatelského rozhraní API

Každá skupina API obsahuje příkazy a události. Události mohou být dvojího druhu - nevyžádané nebo vyvolané připojeným zařízením (MCU). Nevyžádané události jsou generovány asynchronně a jsou volány zařízením CC3000. Mezi nevyžádané události patří :

- **HCI_EVT_WLAN_ASYNC_SIMPLE_CONFIG_DONE** – oznámení, že proces konfigurace je kompletní.
- **HCI_EVT_WLAN_UNSOL_CONNECT** – WLAN je připojeno.
- **HCI_EVT_WLAN_UNSOL_DISCONNECT** – zařízení cc3000 je odpojeno od AP.

- `HCI_EVNT_WLAN_UNSOL_DHCP` – byla přidělena dynamicky IP adresa zařízení `cc3000`.
- `HCI_EVNT_WLAN_UNSOL_INIT` – zařízení `cc3000` dokončilo inicializační proces.
- `HCI_EVNT_WLAN_ASYNC_PING_REPORT` – oznámení o výsledku ping.

Vrstva Network stacks APIs navazuje spojení TCP/IP mezi zařízeními za předpokladu, že byla provedena inicializace a připojení do sítě ve vrstvě WLAN APIs. Přenos dat probíhá přes rozhraní socket vrstvy pomocí standardních příkazů "send" a "sendto", které jsou součástí BSD socketu. Volba, který příkaz použít, závisí na typu použitého socketu UDP (send) nebo TCP (sendto). Seznam knihoven APIs a HCI vrstev dodávaných výrobcem:

- `cc3000_common`
- `evnt_handler`
- `hci`
- `netapp`
- `nvmen`
- `security`
- `socket`
- `wlan`

Wifi modul CC3000 poskytuje plnou podporu pro procesory MSP od Texas Instruments, jako je řada MSP430 či Stellaris. Pro jiné typy procesorů (např. řada STM32) je potřeba napsat vlastní knihovnu, která představuje SPI transportní vrstvu. Tato vrstva musí obsahovat následující funkce využívané ve vyšších vrstvách:

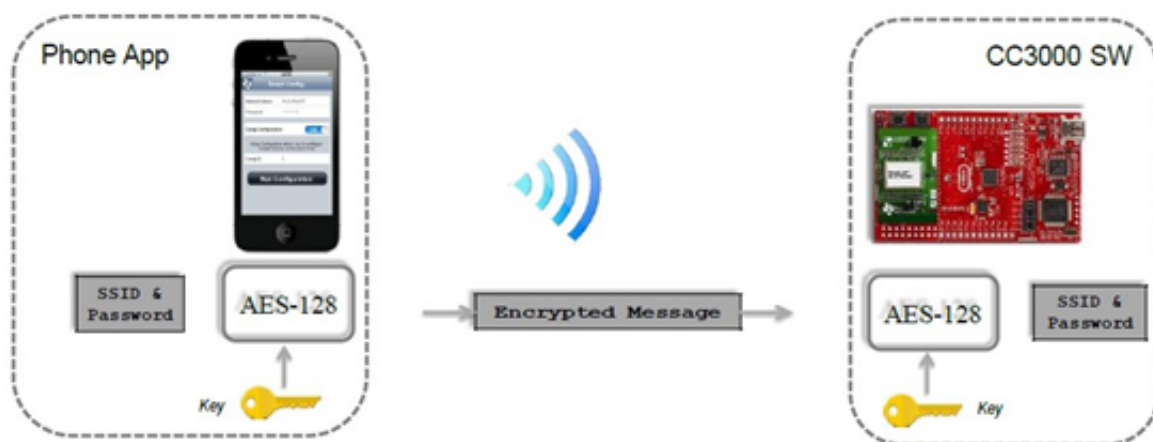
- `SpiOpen` – hlavním úkolem této funkce je inicializace SPI rozhraní a vykonávání obslužné rutiny SPI RX, která je volána při každém přijatém paketu od zařízení CC3000.
- `SpiWrite` – parametrem funkce jsou data, která mají být přenesena přes SPI rozhraní, a jejich délka.
- `SpiRead` – operace čtení se provádí při příjmu IRQ požadavku (aktivní log. 0) v klidovém stavu SPI.
- `SpiClose` – hardware SPI je uveden do původního nastavení, vypnutí zařízení `cc3000`.
- `SpiResumeSpi` – pokračování SPI komunikace.

- `sReadWlanInterruptPin` – přečte hodnotu SPI IRQ pinu na zařízení CC3000.
- `sWlanInterruptEnable` – povolení přerušení externího vstupu IRQ nebo DMA při příjmu dat.
- `sWlanInterruptDisable` – zakázání přerušení externího vstupu IRQ nebo DMA při příjmu dat.
- `sWriteWlanPin` – vyvolá zapsání hodnoty na pin ENABLE, který slouží jako reset pro zařízení cc3000.

Důležitým faktem je, že velikost vysílaného paketu je dána velikostí vyrovnávací paměti RX a TX zásobníku. Zásobníky RX a TX jsou vymezeny v rámci hranic proměnné `CC3000_MINIMAL_RX_SIZE` a `CC3000_MAXIMAL_RX_SIZE`, které jsou obsaženy v hlavičkovém souboru `cc3000/common`. Velikosti RX a TX vyrovnávací paměti může být změněna v závislosti na požadavku aplikace. Maximální velikost RX a TX zásobníku je 1520 bytů, ale maximální velikost užitečných dat v TX a RX zásobníku bez HCl a SPI záhlaví činí pouhých 1468 bytů.

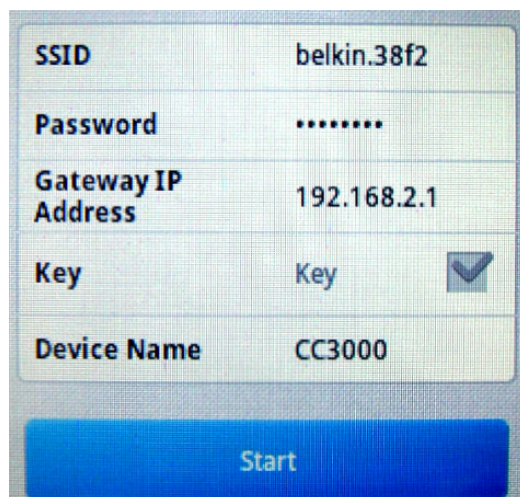
4.2.3 Smart Config CC3000

Prvním krokem před samotným použitím modulu ve wifi síti je jeho konfigurace. Je důležité znát informace o AP, jako je identifikátor SSID jedinečný pro bezdrátovou wifi síť a bezpečnostní kód za předpokladu, že je povoleno zabezpečení sítě pomocí WEP, WPA nebo WPA2. Vzhledem k tomu, že aplikace s těmito wifi moduly většinou postrádají uživatelské rozhraní (klávesnici nebo dotykovou obrazovku), proces připojení do sítě může být poměrně složitý. Proto firma TI vyvinula technologii Smart config pro snadné připojení do domácí sítě. Proces konfigurace je zabezpečen pomocí šifrování AES-128.



Obr. 4.8: Smart config zdroj |Převzato z [12]|

Pře samotným provedením procedury Smart config je nutné, aby modul CC3000 přešel do režimu konfigurace (vyvoláno externí událostí, např. stisknutím tlačítka). V tomto režimu dochází ke skenování konfiguračních UDP paketů, které obsahují SSID a klíč AP. Jakmile CC3000 obdrží veškeré informace, generuje asynchronní událost SIMPLE CONFIG DONE k hostitelskému MCU a dojde k uložení profilu připojení do vnitřní paměti EEPROM wifi modulu. Proces inteligentní konfigurace lze provádět pomocí šifrování AES, nebo bez něj. V zabezpečeném přenosu konfigurační zařízení (smartphone, tablet, pc) zakóduje informace o AP pomocí specifického AES klíče a začne tyto informace vysílat jako broadcast. Hostitelský MCU dekoduje informace a dojde k uložení profilu. Jakmile je proces inteligentní konfigurace hotový, CC3000 se resetuje a připojí k požadovanému AP. Na závěr je odeslán mDNS (Multicast DNS) paket s cílem oznámit ukončení konfigurace zařízení. Následující obrázek zobrazuje ukázkou konfiguračního programu pro mobilní telefon s operačním systémem Android. Název sítě SSID a IP adresa jsou automaticky generovány, stačí zadat heslo AP a zahájit proces konfigurace stisknutím tlačítka START.



Obr. 4.9: Program Smart Config pro chytré telefony

5

Processor

V této kapitole uvádím kritéria, na základě kterých jsem zvolil vhodný procesor. Následně popisují mnou používaná periferní rozhraní.

5.1 Předpoklady pro výběr

Výběr správného procesoru je důležitým předpokladem pro správnou funkci celého zařízení. Prvním kritériem pro výběr procesoru byl jeho dostatečný výpočetní výkon. Na základě předchozích zkušeností jsem zvolil mikrokontroléry od firmy STMicroelectronics, které nabízejí 32-bitové procesory řady STM32, založené na nejnovějším jádru ARM Cortex M4. Toto jádro je navrženo speciálně pro embedded aplikace vyžadující vysoký výkon, běh v reálném čase a nízký příkon. Další předností tohoto jádra jsou jeho nízké nároky na plochu čipu. Díky tomuto faktu je výsledná cena mikrokontroléru nízká, takže může konkurovat i některým výkonným 8-bitovým či 16-bitovým mikrokontrolérům. Druhým kritériem pro výběr procesoru byla jeho schopnost podporovat kamerové rozhraní DCMI (Digital CaMera Interface). Posledním kritériem byla vhodná velikost pouzdra procesoru a počtu jeho pinů. DCMI rozhraní je obsaženo v procesorech řady STM32 s nejmenším pouzdrům LQFP100. Seznam procesorů s DCMI rozhraním v pouzdře LQFP100:

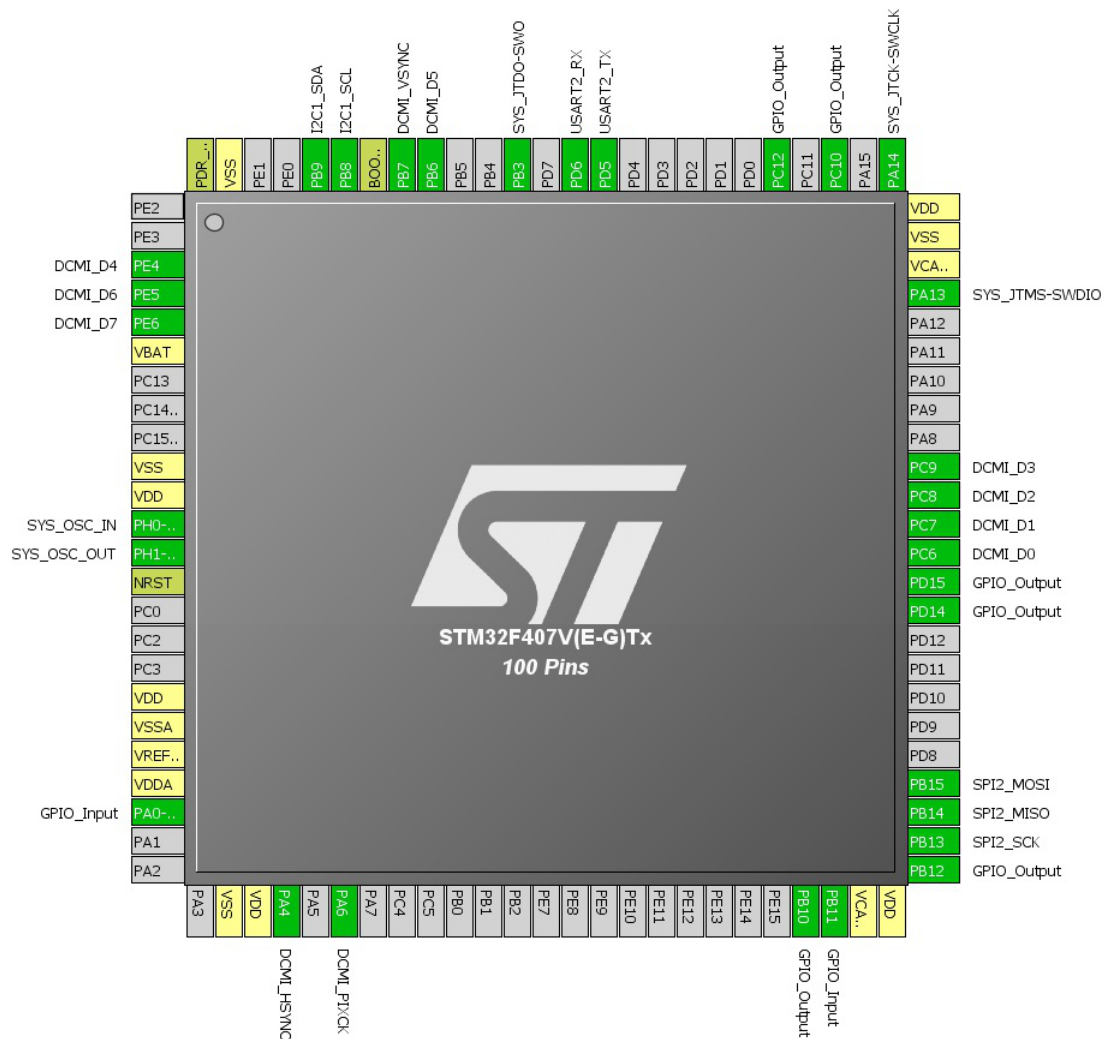
- STM32F207VxTx
- STM32F217VxTx
- STM32F407V(E-G)Tx
- STM32F417V(E-G)Tx
- STM32F427V(E-G)Tx
- STM32F429V(E-G)Tx
- STM32F437V(E-G)Tx

Levnější řada procesorů STM32F2 je založena na jádru ARM Cortex M3, a je tedy dostatečně výkonná. Pro moji aplikaci je ovšem nedostačující z hlediska velikosti paměti, a proto bylo nutné použít výkonnější řadu STM32F4 založenou na ARM Cortex M4 s větší SRAM pamětí. To mi umožnilo pracovat s větším rozlišením přenášeného obrazu. Konkrétně jsem zvolil procesor STM32F407VGT6, který je obsažen ve vývojovém kitu STM32F4-DISCOVERY, na kterém jsem celé zařízení vyvinul. Základní parametry procesoru STM32f407VGT6 jsou uvedeny v tabulce 5.1.

procesor STM32F407VGT6	
Jádro	ARM Cortex M4
Šířka datové sběrnice	32-bit
Velikost programové paměti FLASH	1 MB
Velikost datové paměti SRAM	192 KB
Provozní napájecí napětí	1,8 až 3,6 V
Periferní zařízení	DMA, I2S, LCD, PWM, WDT, ADC, DAC
Komunikační rozhraní	CAN, DCMI, Ethernet, I2C
	LIN, SPI, UART/USART, USB OTG
Počet programovatelných vstupů/výstupů	82
Pouzdro	100-LQFP
Provozní teplota	-40°C 85°C

Tab. 5.1: Základní parametry procesoru STM32F407

Pro návrh rozložení pinů procesoru STM32F407 jsem použil volně dostupný program MicroXplorer od Firmy STMicroelectronics. Díky tomuto programu jsem měl k dispozici rychlý přehled funkcí všech pinů, v případě překrytí funkcí na jednom pinu i možnost přemapování. Tento program je také schopen generovat zdrojový kód pro nastavení jednotlivých pinů.



Obr. 5.1: Rozložení pinů procesoru STM32f407

5.2 Periferní zařízení a programování

Procesor STM32F407 je naprogramován v Keil MDK verze 5. Jedná se o nejnovější verzi kompletního prostředí pro vývoj softwaru pro celou škálu procesorů ARM, jako je Cortex-M nebo Cortex-R. MDK obsahuje uVision IDE/Debugger a ARM C/C++ kompilátor. Použil jsem volně dostupnou verzi MDK-ARM Lite Edition, která má následující omezení:

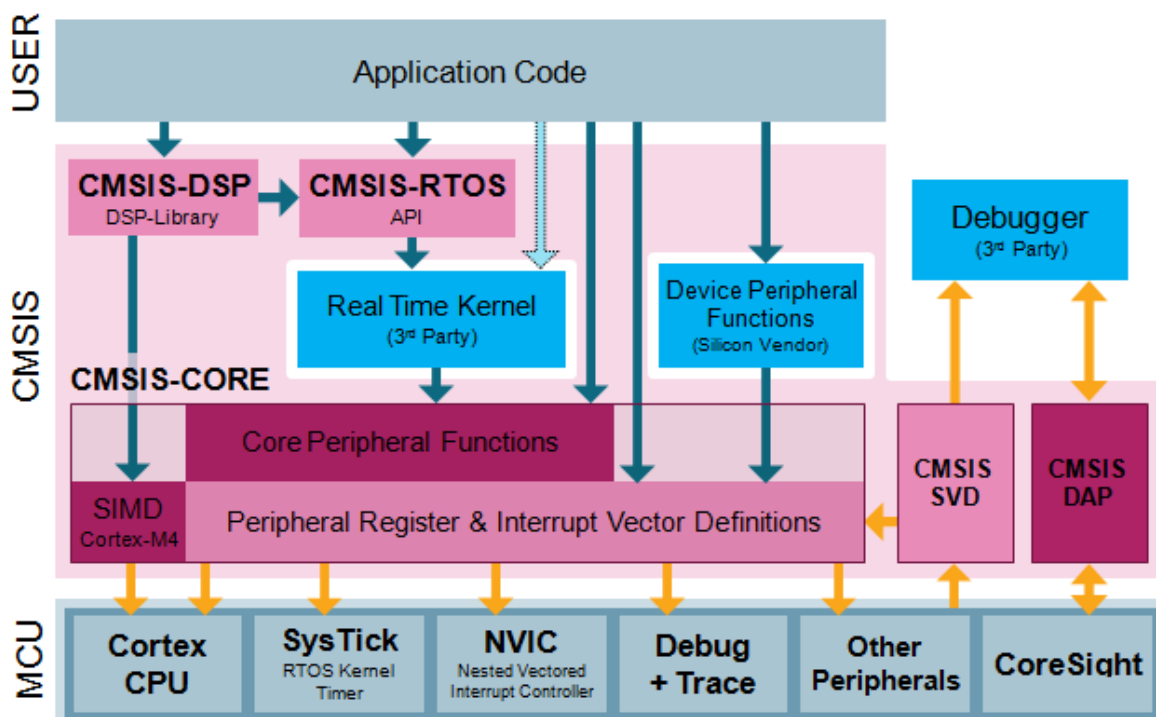
- Maximální velikost kódu je omezena na 32 kB.
- Debugger nepodporuje programy větší než 32 kB.
- Kompilátor neumožňuje generovat výpis strojového kódu -S, -ASM.

Hlavní součástí každého projektu je "startup" kód, který je spuštěn těsně po restartování mikrokontroléru, před spuštěním hlavního programu main. Kód je realizován jako

univerzální pro všechny typy mikrokontrolérů řady STM32F4 (např. "startup_stm32f4xx.s"). Linker Skript a "startup" se spouštějí společně, to znamená, že se automaticky inicializuje MCU podle dat uložených v linkeru. "startup" kód inicializuje proměnné, kopíruje definované proměnné z Flash do RAM paměti, inicializuje zásobník a pak předává prostředky hlavnímu programu (main). "startup" kódy jsou obvykle psány v assembleru nebo v jazyce C. Další důležitou částí projektu jsou CMSIS knihovny, které jsou popsány v následující podkapitole.

5.2.1 CMSIS

CMSIS (Cortex Microcontroller Software Interface Standard) rozhraní je nezávislé na dodavateli hardwaru abstraktní vrstvy pro řadu procesoru Cortex-M. Představuje jednoduché softwarové rozhraní pro procesor, které podstatně zkracuje dobu vývoje zařízení.



Obr. 5.2: Struktura CMSIS |Převzato z [7]|

Komponenty CMSIS :

- **CMSIS-CORE** API rozhraní pro jádro procesoru Cortex-M. Poskytuje standardizované rozhraní pro Cortex-M0, Cortex-M3, Cortex-M4 a pro Cortex-M4.
- **CMSIS-DSP** Soubor více než 60 funkcí pro různé datové typy s fixní přesností nebo s plovoucí desetinnou čárkou. Knihovna obsahuje zdrojový kód a příklad aplikace včetně společných DSP algoritmů (složité aritmetické a vektorové operace).
- **CMSIS-RTOS** API Společné API rozhraní pro operační systémy Real-Time. Poskytuje standardní programovací rozhraní, které je přenosné do mnoha RTOS systémů.

- CMSIS-SVD Popis periferních zařízení do souboru XML.
- CMSIS-DAP Debug Access Port. Standardizovaný firmware pro ladicí jednotku.

CMSIS je obrovská softwarová vrstva, která nedefinuje standardní periférie, ale usnadňuje přístup k těmto periferním zařízením. Umožňuje výrobcům vyrábět široké variace zařízení, které jsou založené na procesorech Cortex-M se společným standardem. CMSIS zlepšuje přenositelnost softwaru. Vývojáři mohou psát software rychleji prostřednictvím snadno použitelného a standardizovaného softwarového rozhraní. CMSIS je podporován všemi tradičními kompilátory (ARMCC, IAR a GNU).

Základní knihovny CMSIS použité v mém projektu:

- `system_stm32f4xx.c` – Tento soubor obsahuje konfiguraci systémových hodin pro procesory STM32F4xx.
- `core_cm4.h` Core Peripheral Access Layer – obsahuje funkce pro snadný přístup do jádra procesoru Cortex-M4.

Další důležitou částí jsou standardní periferní knihovny pro snadnou konfiguraci periferních obvodů:

- `misc` – funkce pro konfiguraci přerušování NVIC řadiče.
- `stm32f4xx_dcmi`
- `stm32f4xx_dma`
- `stm32f4xx_exti`
- `stm32f4xx_gpio`
- `stm32f4xx_rcc`
- `stm32f4xx_syscfg`
- `stm32f4xx_usart`
- `stm32f4xx_spi`
- `stm32f4xx_i2c`
- `stm32f4xx_tim`

V následujících podkapitolách se zmíním o periferních zařízeních, využívaných v bezdrátovém kamerovém modulu, a uvedu jednotlivá nastavení pomocí standardních periferních knihoven.

5.2.2 DMA

Jedná se o přímý přístup do paměti (Direct Memory Access), způsob přímého přenosu dat mezi paměťmi a periferními zařízeními. Data neprocházejí skrz procesor, a proto lze dosáhnout vyššího výkonu (během přenosu dat může procesor zpracovávat jiné strojové instrukce). DMA se používá pro přenos větších objemů dat.

Procesor STM32F407 obsahuje dva DMA řadiče, které jsou téměř identické. Přesto jsou mezi nimi určité rozdíly. DMA2 řadič může vykonávat přesuny z paměti do paměti, zatímco DMA1 nikoliv. Dále DMA1 je připojeno k APB1 sběrnici (142 MHz) a DMA2 k APB2 sběrnici (84 MHz). Každý DMA řadič má osm samostatných streamů a každý stream je spojen s jedním z osmi volitelných kanálů. Každý DMA port má možnost nastavení priority, která je až čtyřúrovňová (very high, high, medium a low). V případě stejné priority kanálu v daný okamžik, kanál s nižším má přednost vykonat danou instrukci. DMA přenos je definovaný zdrojovou a cílovou adresou a je schopen provádět tři různé režimy přenosu:

- z paměti do paměti
- z periférie do paměti
- z paměti do periférie

Peripheral requests	Stream 0	Stream 1	Stream 2	Stream 3	Stream 4	Stream 5	Stream 6	Stream 7
Channel 0	ADC1		TIM8_CH1 TIM8_CH2 TIM8_CH3		ADC1		TIM1_CH1 TIM1_CH2 TIM1_CH3	
Channel 1		DCMI	ADC2	ADC2				DCMI
Channel 2	ADC3	ADC3				CRYP_OUT	CRYP_IN	HASH_IN
Channel 3	SPI1_RX		SPI1_RX	SPI1_TX		SPI1_TX		
Channel 4			USART1_RX	SDIO		USART1_RX	SDIO	USART1_TX
Channel 5		USART6_RX	USART6_RX				USART6_TX	USART6_TX
Channel 6	TIM1_TRIG	TIM1_CH1	TIM1_CH2	TIM1_CH1	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	
Channel 7		TIM8_UP	TIM8_CH1	TIM8_CH2	TIM8_CH3			TIM8_CH4 TIM8_TRIG TIM8_COM

Obr. 5.3: DMA2 mapování |Převzato z [6]|

Z obr 5.3 lze vyčíst, že pro nastavení DMA přenosu z DCMI periferního zařízení je třeba nastavit Channel = 1 a Stream = 1. Další konfigurace je zřejmá z následujícího výpisu zdrojového kódu s komentáři.

```

1 DMA_InitTypeDef DMA_InitStructure;
2
3 /*zapnutí hodin pro DMA2*/

```

```

4  RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_DMA2, ENABLE);
5
6  /*definování parametrů DMA */
7  DMA_InitStructure.DMA_Channel = DMA_Channel_1;
8  DMA_InitStructure.DMA_PeripheralBaseAddr = DCMI_DR_ADDRESS;
9  DMA_InitStructure.DMA_Memory0BaseAddr = (uint32_t)frame_buffer;
10 DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralToMemory;
11
12 //nastavení velikosti bufferu pro rozlišení 320*240 ve formátu RGB565 (16-bit)
13 DMA_InitStructure.DMA_BufferSize = 320*240/2;
14 DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
15 DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Enable;
16 DMA_InitStructure.DMA_PeripheralDataSize = DMA_PeripheralDataSize_Word; 32-bit
17 DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_Byte; 8-bit
18 DMA_InitStructure.DMA_Mode = DMA_Mode_Circular;
19 DMA_InitStructure.DMA_Priority = DMA_Priority_High;
20 DMA_InitStructure.DMA_FIFOMode = DMA_FIFOMode_Enable;
21 DMA_InitStructure.DMA_FIFOThreshold = DMA_FIFOThreshold_Full;
22 DMA_InitStructure.DMA_MemoryBurst = DMA_MemoryBurst_Single;
23 DMA_InitStructure.DMA_PeripheralBurst = DMA_PeripheralBurst_Single;
24
25 /*konfigurace IRQ DMA kanálu*/
26 DMA_Init(DMA2_Stream1, &DMA_InitStructure);
27
28 /*nastavení priorit přerušeni DMA2_Stream1*/
29 NVIC_InitStructure.NVIC_IRQChannel = DMA2_Stream1_IRQn;
30 NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 2;
31 NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1;
32 NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
33 NVIC_Init(&NVIC_InitStructure);
34
35 DMA_Cmd(DMA2_Stream1, ENABLE); //povolení DMA

```

5.2.3 DCMI

Digitální kamerové paralelní rozhraní je schopné přijímat vysokorychlostní datový tok z externí 8-, 10-, 12- nebo 14-bitové CMOS kamery. Podporuje řadu různých formátů, jako je například YCbCr422/RGB565 nebo kompresní datový formát JPEG. Následující tabulka obsahuje seznam vstupních signálů DCMI rozhraní.

D[0:13]	vstupní data
HSYNC	vstup horizontální synchronizace
VSYNC	vstup vertikální synchronizace
PIXCLK	vstup hodinového signálu

Tab. 5.2: DCMI piny

Digitální kamerové rozhraní je schopné přijímat data rychlostí až 54 MB/s. Vstupní signál PIXCLK má programovatelnou polaritu, takže data mohou být zachycena buď na náběžnou, nebo sestupnou hranu signálu PIXCLK. Data jsou přijímána do 32-bitového registru (DCMI_DR) a následně poslána skrz DMA kanál. Důležitým faktem je, že frame buffer je řízen DMA kanálem, nikoliv kamerovým rozhraním. Data z kamery mohou být přijímána buď do řádku (YUV/RGB), nebo do bloků u JPEG formátu. Pokud chceme povolit příjem JPEG obrázku, je nutné povolit bit 3 v DCMI_CR registru. Tok dat je synchronizován hardwarově pomocí signálů HSYNC a VSYNC nebo pomocí synchronizačního kódu vloženého do toku dat (embedded). DMA přenos se musí povolit Capture bitem v

DCMI_CR registru. Požadavek na přenos DMA kanálem je generován vždy, když obdrží kompletní 32-bitový blok dat ve svém datovém registru.

V následující části zdrojového kódu je uvedena konfigurace DCMI rozhraní podle kapitoly 3.2.1 (Komunikace s OV2640). Podle této kapitoly jsou data z kamerového modulu OV2640 vzorkována na náběžnou hranu hodinového signálu, polarita signálu HSYNC je nastavena na vysokou úroveň (aktivní log. 0) a šířka datové sběrnice činí 8 bitů. U DCMI rozhraní je možné nastavit až pět zdrojů přerušení. Jak lze vidět z výpisu zdrojového kódu, nastavil jsem pouze tři - přerušení při přetečení, chybě synchronizace a VSYNC (generováno po zachycení celého snímku).

```

1  DCMI_InitTypeDef DCMI_InitStructure;
2
3  /*zapnutí hodin pro DCMI*/
4  RCC_AHB2PeriphClockCmd(RCC_AHB2Periph_DCMI, ENABLE);
5
6  /* DCMI konfigurace */
7  DCMI_InitStructure.DCMI_CaptureMode = DCMI_CaptureMode_Continuous;
8  DCMI_InitStructure.DCMI_SynchroMode = DCMI_SynchroMode_Hardware;
9  DCMI_InitStructure.DCMI_PCKPolarity = DCMI_PCKPolarity_Rising;
10 DCMI_InitStructure.DCMI_VSPolarity = DCMI_VSPolarity_High;
11 DCMI_InitStructure.DCMI_HSPolarity = DCMI_HSPolarity_Low;
12 DCMI_InitStructure.DCMI_CaptureRate = DCMI_CaptureRate_1of2_Frame;
13 DCMI_InitStructure.DCMI_ExtendedDataMode = DCMI_ExtendedDataMode_8b;
14
15 /*konfigurace DCMI registrů*/
16 DCMI_Init(&DCMI_InitStructure);
17
18 /*nastavení priorit přerušení IRQ*/
19 NVIC_PriorityGroupConfig(NVIC_PriorityGroup_1);
20 NVIC_InitStructure.NVIC_IRQChannel = DCMI_IRQn;
21 NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 1;
22 NVIC_InitStructure.NVIC_IRQChannelSubPriority = 2;
23 NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
24 NVIC_Init(&NVIC_InitStructure);
25
26 /*nastavení zdrojů přerušení DCMI*/
27 DCMI_ITConfig(DCMI_IT_VSYNC, ENABLE);
28 DCMI_ITConfig(DCMI_IT_OVF, ENABLE);
29 DCMI_ITConfig(DCMI_IT_ERR, ENABLE);
30
31 DCMI_Cmd(ENABLE); //povolení DCMI
32 DCMI_CaptureCmd(ENABLE); //povolení Capture - video

```

V obsluze přerušení, podle níže uvedeného zdrojového kódu od zdroje VSYNC, dochází k překopírování celého obrázku do dočasného bufferu pomocí funkce memcpy, která je definovaná v hlavičkovém souboru string.h. Dočasný buffer je postupně zpracováván v hlavním programu v nekonečné smyčce. Přerušení je generováno desetkrát za vteřinu a odpovídá snímkové frekvenci (fps).

```

1 void DCMI_IRQHandler(void)
2 {
3     if (DCMI_GetFlagStatus(DCMI_FLAG_VSYNCRI) == SET){
4         memcpy(send_buffer, frame_buffer, 320*240/2);}
5
6     else if (DCMI_GetFlagStatus(DCMI_FLAG_ERRRI) == SET){
7         debug_write("DCMI_FLAG_ERRRI\n");}
8
9     else if (DCMI_GetFlagStatus(DCMI_FLAG_OVFRI) == SET){
10        debug_write("DCMI_FLAG_OVFRI\n");}
11
12    DCMI_ClearFlag(DCMI_FLAG_VSYNCRI);
13    DCMI_ClearFlag(DCMI_FLAG_ERRRI);
14    DCMI_ClearFlag(DCMI_FLAG_OVFRI);
15 }

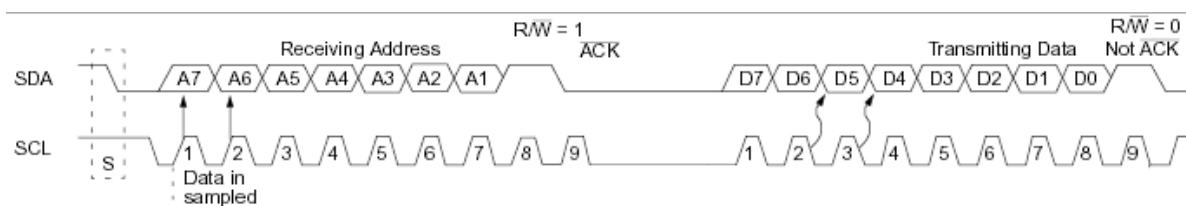
```

5.2.4 I2C

I2C sběrnice je použita pro konfiguraci kamerového modulu OV2640. I2C rozhraní může pracovat v jednom z následujících režimů:

- slave vysílač
- slave přijímač
- master vysílač
- master přijímač

Jelikož kamerový modul OV2640 pracuje v režimu slave, procesor musí pracovat v režimu master. V režimu master I2C rozhraní zahajuje přenos dat a generuje hodinový signál. Každému zařízení do módu slave, připojenému na sběrnici I2C, je přidělena unikátní celočíselná adresa, aby bylo možné jednoznačně určit uzel, se kterým má zařízení typu master komunikovat. Jelikož mám na sběrnici připojeno pouze jedno zařízení, využívám pouze 7-bitovou adresu přenášenu v jednom bytu, přičemž poslední přenesený bit má význam přepínače mezi módem čtení a zápisu dat.



Obr. 5.4: Průběh I2C komunikace [Převzato z [21]]

Na obr 5.4 je zobrazen průběh I2C komunikace při adresování uzlů pomocí 7-bitové adresy. Komunikace je zahájena start bitem, tj. přechodem signálu SDA z klidového stavu (log. 1) na datovém i hodinovém vodiči na úroveň logické nuly. Ihned po start bitu je možné zahájit přenos adresy. Zařízení master vyšle na sběrnici všech 7 bitů adresy, které jsou vzorkovány při náběžné hraně hodinového signálu. Poslední bit určuje, zda se mají data z cílového zařízení vysílat, či naopak zda má toto zařízení data přijímat. Po přenosu všech 8 bitů (7 bitů adresy + 8. bit s určením směru přenosu dat) provede každý uzel slave porovnání přijaté adresy se svoji vlastní adresou. Vybraný uzel musí potvrdit, že se na sběrnici skutečně nachází. Potvrzení (bit ACK – acknowledge) se v devátém cyklu hodin pošle zpět na zařízení typu master. Příjem logické nuly značí, že adresovaný uzel skutečně existuje a je možné s ním zahájit komunikaci. [21]

V následující části je uvedena inicializační sekvence pro rozhraní I2C.

```

1 I2C_InitTypeDef I2C_InitStruct;
2
3 /*povolení hodinového signálu na I2C*/
4 RCC_APB1PeriphClockCmd(RCC_APB1Periph_I2C1, ENABLE);
5

```

```

6  /* nastavení I2C struktury */
7  I2C_InitStruct.I2C_Mode = I2C_Mode_I2C;
8  I2C_InitStruct.I2C_DutyCycle = I2C_DutyCycle_2;
9  I2C_InitStruct.I2C_OwnAddress1 = 0xFE;
10 I2C_InitStruct.I2C_Ack = I2C_Ack_Enable;
11 I2C_InitStruct.I2C_AcknowledgedAddress = I2C_AcknowledgedAddress_7bit;
12 I2C_InitStruct.I2C_ClockSpeed = 30000;
13
14 /* inicializace I2C */
15 I2C_Init(I2C1, &I2C_InitStruct);
16
17 /* zapnutí I2C periférie */
18 I2C_Cmd(I2C, ENABLE);

```

Funkce pro zápis a čtení konfiguračních registrů kamery OV2640 :

- `OV2640_WriteReg(uint16_t Addr, uint8_t Data);`
- `OV2640_ReadReg(uint16_t Addr);`

5.2.5 SPI

Jedná se o sériovou synchronní periferní sběrnici, která umožňuje plně duplexní komunikaci jedné řídicí master jednotky s dalšími podřízenými slave jednotkami po společné sběrnici. Obvykle SPI je připojeno k externímu zařízení pomocí následujících signálů:

- MOSI (Master Out / Slave In data)
- MISO (Master In / Slave Out data)
- SCLK (Serial Clock)
- NSS (Slave Select)

Sběrnice SPI je v navrhovaném zařízení použita pro komunikaci s wifi modulem CC3000, který se chová jako slave zařízení. SPI je nastaveno do módu master s plně duplexním přenosem s délkou vysílaných dat 16-bitů. Mikrokontrolér STM32F407 obsahuje několik SPI rozhraní. Použité rozhraní SPI2 je připojeno na vnitřní sběrnici APB1 o kmitočtu 42 MHz. S nastavenou děličkou je frekvence SCLK signálu generovaného SPI rozhraním nastavena na 21 MHz. Polarita a fáze hodinového signálu je nastavena podle připojeného wifi modulu viz. kapitola 4.2.1 ("Komunikace s CC300").

- `CPOL = 0` – klidová úroveň hodinového signálu log. 0.
- `CPHA = 1` – hodnota je čtena při přechodu hodin z aktivní do klidové úrovně ("druhou" hranou).

```

1  SPI_InitTypeDef spiConfig;
2
3  //povolení hodinového signálu pro SPI2
4  RCC_APB1PeriphClockCmd(RCC_APB1Periph_SPI2, ENABLE);
5
6  spiConfig.SPI_Direction = SPI_Direction_2Lines_FullDuplex;
7  spiConfig.SPI_Mode = SPI_Mode_Master;

```

```
8 spiConfig.SPI_DataSize = SPI_DataSize_16b;
9 spiConfig.SPI_NSS = SPI_NSS_Soft;
10 spiConfig.SPI_BaudRatePrescaler = SPI_BaudRatePrescaler_2;
11 spiConfig.SPI_FirstBit = SPI_FirstBit_MSB;
12
13 // mód 1 (CPOL = 0, CPHA = 1)
14 spiConfig.SPI_CPOL = SPI_CPOL_Low;
15 spiConfig.SPI_CPHA = SPI_CPHA_2Edge;
16
17 //inicializace definované struktury
18 SPI_Init(SPI2, &spiConfig);
19
20 //povolení SPI2
21 SPI_Cmd(SPI2, ENABLE);
```

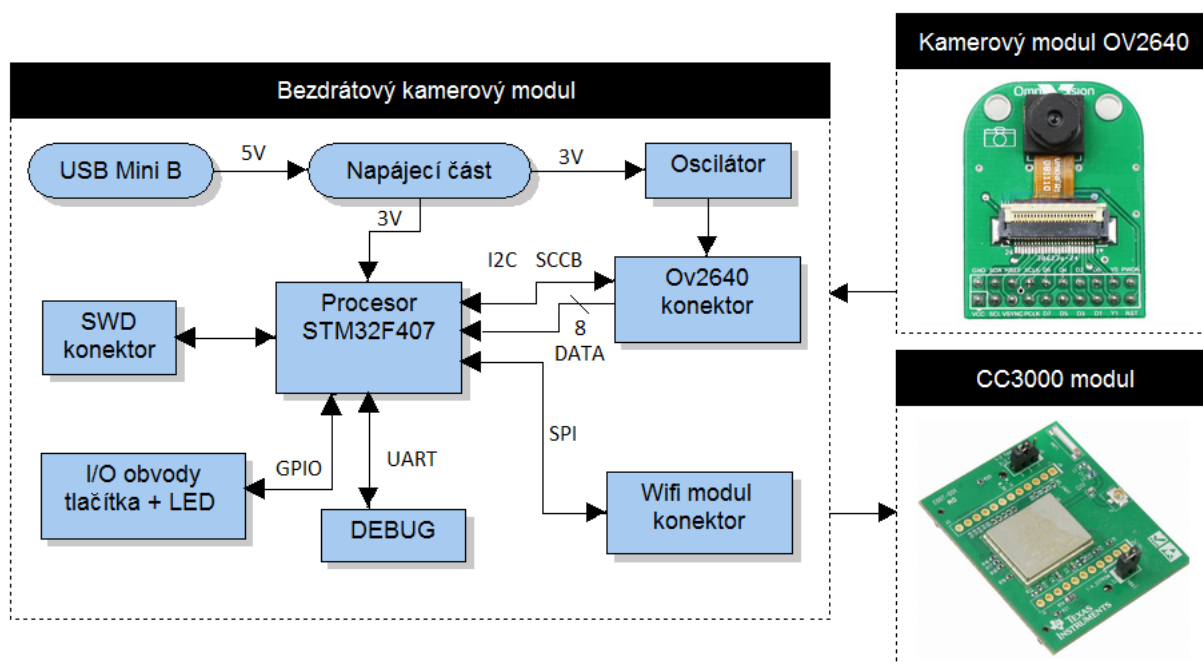
6

Praktická realizace

Předmětem této kapitoly je stručný popis funkce vysílacího i přijímacího modulu zařízení.

6.1 Vysílací modul

Celá softwarová část zařízení byla vyvinuta na STM32F4 Discovery kitu, na kterém byla také ověřena správná funkce zařízení. Na základě toho byl realizován návrh hardwarové části. Při návrhu hardwaru byl kladen velký důraz na spolehlivost, jednoduchost a zejména co nejmenší velikost zařízení. Pro naprogramování byl použit vestavěný programovací a ladící nástroj ST-LINK/V2 na STM32F4 Discovery kitu. Blokové schéma zařízení je zachyceno na obr. 6.1.



Obr. 6.1: Blokové schéma zařízení

6.1.1 Napájení

Pro napájení celého zařízení byl použit obvod TPS79333 od firmy Texas Instrument. Jedná se o regulátor s nízkým úbytkem napětí (LDO), nízkým šumovým napětím, rychlým náběhem (cca 50 μ s) a nízkým klidovým proudem (170 μ A), což je výhodné zejména pro bateriově napájené systémy. Pro zajištění potřebné stability výstupního napětí je nutné na výstup tohoto obvodu připojit keramický kondenzátor 2,2 μ F. Rodina obvodů TPS793xx je založena na BICMOS technologii, díky které je zajištěn malý úbytek napětí 112 mV při proudu 200 mA. Výstupní napětí obvodu TPS79333 činí konstantních 3,3 V, sníženo o úbytek na Schottkyho diodě sloužící k ochraně obvodu (Schéma zapojení viz. příloha A).

6.1.2 Časování

Časování mikrokontroléru STM32F407VGT6 je určeno připojeným externím krystalem o frekvenci 8 MHz, z něhož jsou pomocí vnitřních fázových závěsů (PLL) vytvořeny další kmitočty, např. pro samotné jádro Cortex-M4 pracující na frekvenci 168 MHz. Další kmitočty jsou např. využity pro sběrnice periferních obvodů APB2 (84 MHz) a APB1 (42 MHz).

Časování pro kamerový modul OV2640 je vytvořeno pomocí CMOS oscilátoru FXO-HC536R s frekvencí 24 MHz, s přesností ± 25 ppm a dobou náběhu a doběhu menší jak 3 ns. Důležitým parametrem u CMOS oscilátorů je výstupní kapacitní zatížení, které činí maximálně 15 pF. Jelikož má kamerový modul na vstupním pinu signálu XCLK kapacitu 10 pF, bylo potřeba umístit oscilátor co nejbližší ke konektoru, a předejít tak chybám způsobeným špatným časováním.

6.1.3 Periférie

Kromě připojeného kamerového (I2C-SCCB + DATA) a wifi modulu (SPI), které jsou popsány v kapitole 3 a 4, jsou k procesoru připojeny další periferní obvody - tlačítka a LED diody. Dále je v procesoru použito UART rozhraní připojené přes převodník (USB na RS232) do počítače, kde se vytvoří virtuální sériový port. Toto rozhraní s přenosovou rychlostí 115200 baud slouží k ladění programu a pro výpis informací, jako je vlastní MAC adresa a IP adresa sítě, ke které je bezdrátový kamerový modul připojen.

6.1.4 Hardwarové řešení

Celá hardwarová část kamerového bezdrátového modulu je navržena na dvouvrstvé desce plošného spoje o rozměrech 42x56 mm se čtyřmi montážními otvory. Deska byla navržena v návrhovém systému EAGLE Freeware verze 6.2 od firmy Cadsoft. Použitá bezplatná verze je určena pro návrh desek plošného spoje maximálně o dvou vrstvách a velikosti 100x80 mm. DPS je osazena jak THT, tak i SMD součástkami a konektory pro připojení

kamerového a wifi modulu. Minimální tloušťka čar je 0,3 mm a minimální šířka izolačních mezer 0,24 mm. DPS má rozlitou zem jak v horní, tak i ve spodní vrstvě. Veškerá výrobní dokumentace, jako jsou desky plošných spojů, osazovací plánek, rozmístění montážních otvorů a seznam použitých součástek, je v příloze B.

6.1.5 Popis programu vysílacího modulu

Celá struktura programu se skládá z několika zdrojových kódů - pro práci s wifi modulem (popsané v kapitole 4.2.2) a pro práci s mikrokontrolérem (popsané v kapitole 5.2.1). V následující části této podkapitoly je obsažen popis hlavního programu main.

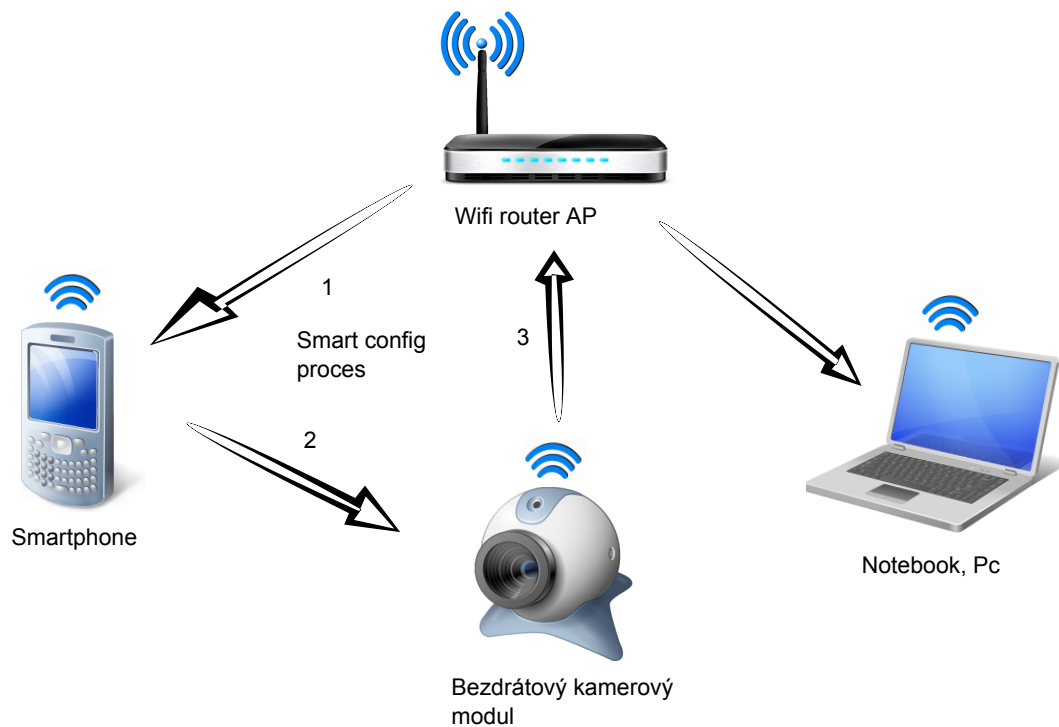
Nejprve je provedena inicializace SPI rozhraní a GPIO pinů, jako je IRQ a EN pro řízení modulu cc3000. Následně dochází k inicializaci pomocí funkce "wlan_init", která slouží pro inicializaci asynchronně volaných událostí (viz. kapitola 4.2.2), pro inicializaci firmwaru a ovladače zařízení. Samotný inicializační proces hardwaru včetně aktualizace softwaru CC3000 je spuštěn funkce "wlan_start", kdy dojde k povolení pinu EN a ke spuštění WLAN zařízení. Po skončení inicializace je nutné provést kontrolu zpětným vyčtením verze softwaru zařízení. Pro správnou funkci musíme zajistit, aby verze firmwaru a ovladače zařízení odpovídala správné verzi knihoven, které zajišťují funkci APIs a HCI vrstvy.

Dalším krokem je připojení zařízení do sítě, tedy k wifi routeru, ke kterému je také připojen smartphone a PC. Před spuštěním samotného procesu konfigurace je nutné smazat předešlý profil připojení a pomocí funkce "netapp_dhcp" nastavit síťové rozhraní, které je nakonfigurováno pro dynamické přidělování IP adresy. Samotný proces konfigurace je spuštěn funkcí "wlan_smart_config_start". V tomto režimu dochází ke skenování konfiguračních UDP paketů, které obsahují SSID a klíč AP. Konfigurační UDP pakety jsou vysílány smartphonem pomocí aplikace Smart Config (obr. 6.2, fáze 2). Jakmile CC3000 obdrží veškeré informace, generuje asynchronní událost SIMPLE CONFIG DONE k hostitelskému MCU a dojde k uložení profilu připojení do vnitřní paměti EEPROM wifi modulu. Po skončení procesu konfigurace zařízení čeká v nekonečné smyčce na přidělení IP adresy. Proces konfigurace je detailněji popsán v kapitole 4.2.3.

Z hlediska rychlosti přenosu byl pro přenos dat vybrán UDP protokol. Před samotným přenosem dat je potřeba vytvořit socket pomocí funkce "socket" s následujícími parametry:

- **domain** – specifikuje rodinu protokolů vytvořeného socketu, podpora pouze AF_INET (síťový protokol IPv4).
- **type** – pro UDP je použit SOCK_DGRAM - protokol založený na datagramech.
- **protocol** – specifikace IP protokolu - IPPROTO_UDP.

Nakonec dojde k připojení předem nastaveného socketu pomocí funkce "connect", jejímž parametrem je cílová adresa, která se skládá se z IP adresy a čísla portu. Tím je proces nastavení wifi modulu kompletní.



Obr. 6.2: Funkční blokové schéma

Poslední částí konfigurace zařízení je nastavení kamerového modulu. V první řadě jsou nastaveny GPIO piny (datové signály DCMI a řídicí VSYNC, HSYNC, PCLK a RST) a I2C rozhraní pro samotnou konfiguraci kamery. Pro ověření správného zapojení a nastavení I2C je vyčteno identifikační číslo kamery. Dalším krokem je inicializace DCMI a DMA rozhraní (popsané v kapitole 5.2) a nastavení konfiguračních registrů přes I2C sběrnici pro rozlišení QVGA (320x240) ve formátu RGB565. Důležitým faktem je, že data nejsou přenášena v JPEG formátu, jak bylo avizováno v úvodních kapitolách. Problém spočíval ve skutečnosti, že kamera, kterou jsem vybral, nebyla schopna provádět JPEG kompresi v reálném čase, tak jak bylo uvedeno v katalogovém listu. Pro rozlišení VGA (640x480) v JPEG formátu byla kamera schopna zpracovat maximálně dva snímky za vteřinu, což je pro potřeby celého zařízení nedostačující. Posledním krokem konfigurace kamerového modulu je povolení snímání obrazu přes DCMI rozhraní.

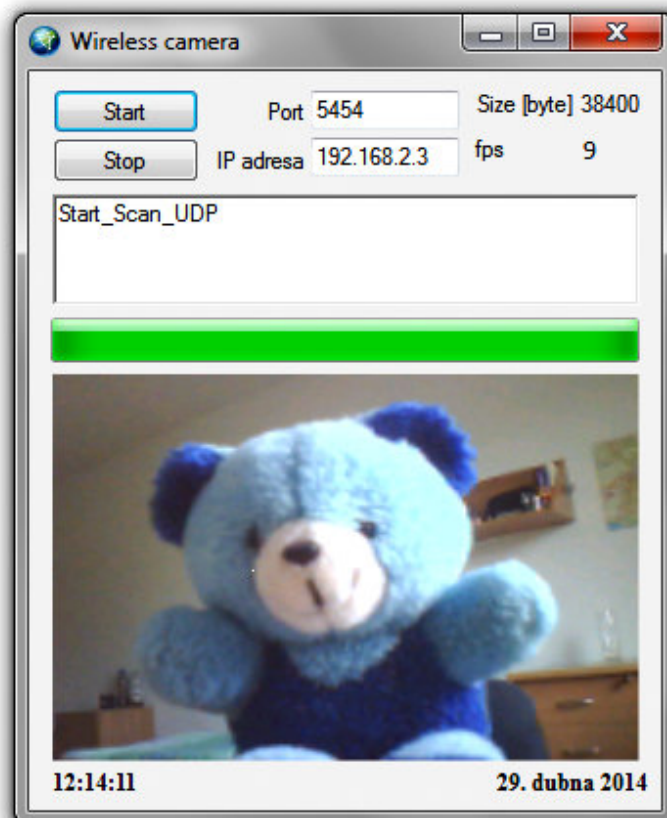
Při zachycení celého snímku DCMI rozhraní generuje přerušování, v jehož obsluze je celý snímek přepokopírován do dočasného bufferu a je nastaven příznak pro zpracování snímku v nekonečné smyčce hlavního programu. Zde jsou data posílána pomocí funkce "sendto", která je určena pro přenos UDP paketů. Celý vysílaný obrázek je rozdělena do paketů s délkou dat 1200 byte. Pro snadnou modifikatelnost jsou vysílaná data kamerového bezdrátového modulu uvozena stejnými příznaky pro začátek a konec obrázku jako u formátu JPEG (0xff 0xd8 - začátek, 0xff 0xd9 - konec). Před odesláním paketu je vždy nutné počkat, zdali byl předchozí paket odeslán.

6.2 Přijímací modul

Cílem této diplomové práce bylo navrhnout vysílací bezdrátový kamerový modul, avšak pro ověření správné funkce bylo nutné navrhnout i modul přijímací. První variantou pro příjem videa bylo použití volně dostupného video přehrávače s GPL licencí, např. VLC. V tomto případě by bylo potřeba implementovat do vysílacího modulu funkce pro RTSP (Real Time Streaming Protocol). Druhou variantou bylo napsání vlastního UDP serveru pro příjem videa. Z důvodu jednoduchého provedení jsem se rozhodl napsat vlastní program pro příjem videa v programu C#.

6.2.1 Stručný popis programu

V programu C# byl vytvořen jednoduchý klient pro příjem UDP paketů. Využití tohoto protokolu v prostředí .NET nám umožňuje třída "UdpClient". Při stisknutí tlačítka "Start" dojde k vytvoření instance třídy UdpClient a k nastavení IP adresy a portu, na kterém budeme přijímat data. Nakonec je zavolána metoda "UdpClient.BeginReceive", která asynchronně volá funkci pro zpracování přijatých dat. Po přijetí paketu, označující začátek obrázku (0xff 0xd8), dojde k postupnému uložení jednotlivých paketů, vždy s délkou 1200 bytů do bufferu. Data jsou přijímána ve formátu RGB565 (16-bitů na pixel), pro zobrazení pomocí třídy "pictureBox" je bylo potřeba převést do formátu bmp.



Obr. 6.3: Ukázka programu po příjem videa

7

Závěr

Úkolem této diplomové práce bylo navrhnout miniaturní bezdrátový kamerový modul. V úvodní pasáži jsou shrnuty možné způsoby realizace těchto kamerových modulů - IP a analogové kamery. Následně jsou pro účely navrhovaného zařízení zvoleny IP kamery postavené na mikrokontrolérech bez operačního systému. V následující kapitole byla stanovena tato kritéria pro výběr vhodného kamerového modulu z hlediska jeho využití: dostatečná kvalita obrazu, vysoká rychlost snímání, malá velikost zařízení. Z důvodu bezdrátového přenosu obrazu byl vybrán kamerový modul OV2640, který splňuje výše uvedená kritéria a vyznačuje se podporou kompresního formátu JPEG, díky čemuž došlo ke snížení nároků na přenosovou rychlost bezdrátového modulu. Kritéria pro výběr vhodného bezdrátového modulu byla následovná: dostatečně velká přenosová rychlost a minimální zpoždění. Pro bezdrátový přenos byl tedy vybrán wifi modul CC3000 s přenosovou rychlostí až 6,8 Mb/s a zanedbatelným zpožděním. Hlavním kritériem pro výběr procesoru byla jeho schopnost podporovat kamerové rozhraní DCMI, a proto byl vybrán odpovídající procesor STM32F407VGT6. Na základě těchto výše uvedených kritérií bylo předpokládáno, že výsledné zařízení bude vykazovat následující parametry: přenos 10 snímků za vteřinu v rozlišení 640x480 v JPEG formátu.

Výsledkem celé práce je návrh vysílacího modulu, jehož funkce byla ověřena na vývojovém kitu STM32F4 prostřednictvím vytvoření modulu přijímače v počítači. Přijímací modul byl realizován jako počítačový program pro příjem a zobrazení sekvence obrázků, napsaný v programovém prostředí C#. Navržený bezdrátový vysílací kamerový modul vykazuje následující parametry a vlastnosti:

- Rozlišení – 320x240
- Počet snímků za vteřinu – 10
- Zpoždění – řádově jednotky ms.
- Velikost – 42x56 mm
- Cena – cca 1400 bez DPS.

- Jednoduché připojení modulu do wifi sítě pomocí mobilní aplikace bez potřeby uživatelského rozhraní (klávesnice, dotyková obrazovka).

Z porovnání předpokládaných a skutečných parametrů navrženého zařízení lze vyčíst, že původního záměru nebylo zcela dosaženo. Hlavní problém představuje fakt, že kamerový modul OV2640 s rozlišením 2 Mpx neumožňuje provádět kompresi v reálném čase (max. 2 fps), tak jako bylo uvedeno v katalogovém listu zařízení. Z tohoto důvodu jsou data z kamery přenášena ve formátu RGB565 (16-bitů na pixel). Vzhledem k maximální přenosové rychlosti bezdrátového modulu dochází ke snížení maximálního rozlišení na pouhých 320x240 pixelů.

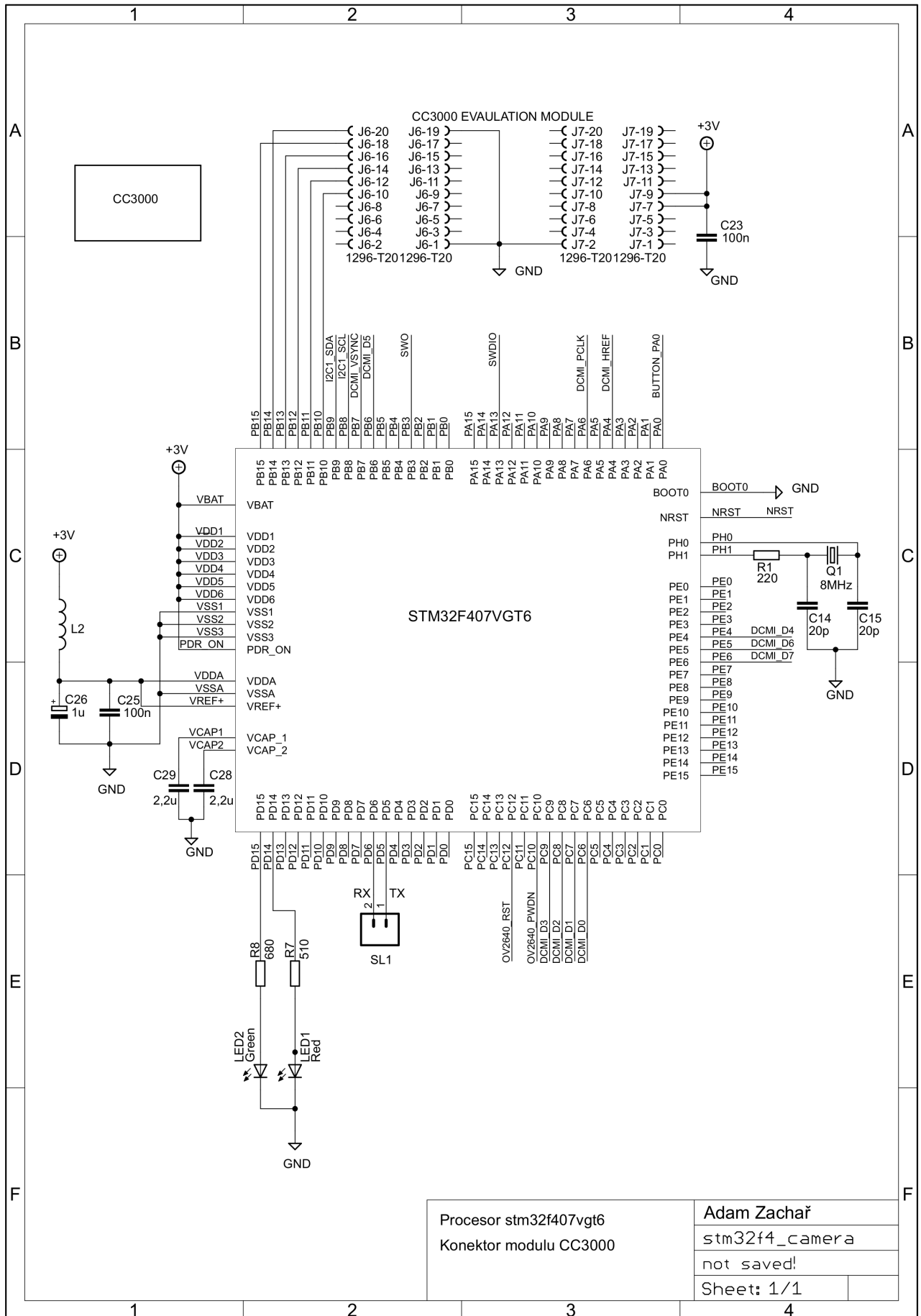
I přes tyto nedostatky vykazuje navržené zařízení velmi dobré parametry. Nicméně stále je zde prostor pro zlepšení. Prvním příkladem zlepšení může být použití lepšího kamerového modulu s výstupním formátem JPEG, který pracuje v reálném čase. To by umožnilo docílení lepší kvality přenášeného obrazu. Dalším příkladem zlepšení zařízení může být vylepšení přijímacího modulu. Místo vlastního programu v jazyce `c#` by bylo vhodnější použít nějaký komerčně dostupný video přehrávač, jako je např. VLC. V tomto případě by bylo nutné implementovat do vysílacího modulu funkce pro RTSP streamování videa.

Literatura

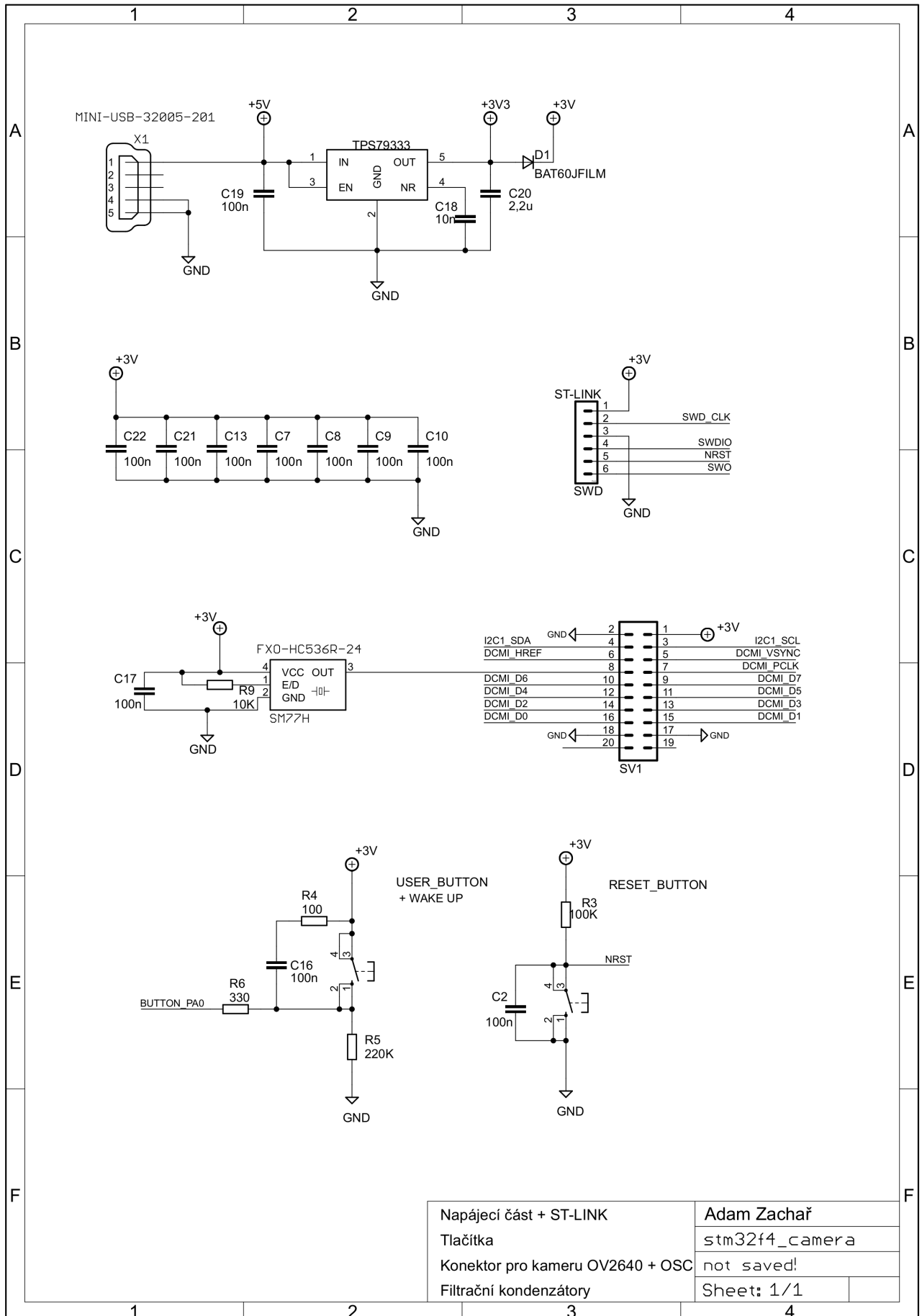
- [1] OV2640. *Color CMOS camera chip* [online]. United States: Copyright 2006, OmniVision. Poslední změna 28.02.2006 [cit. 7.5.2014]. Dostupné z: <http://www.uctronics.com/download/cam`module/OV2640`DS.pdf>
- [2] OV2640. *CMOS camera - SCCB interface* [online]. United States: Copyright 2007, OmniVision. Poslední změna 25.06.2007 [cit. 7.5.2014]. Dostupné z: <http://www.ovt.com/download`document.php?type=document&DID=63>
- [3] OV2640 module. *picture* [online]. [cit. 7.5.2014]. Dostupné z: <http://www.uctronics.com/static/images/20130222/ov2640-2mp-cmos-camera-module-adapter-board-jpeg-out-e00f0dbc-800x800.jpg>
- [4] STM32F407. *Processor - datasheet* [online]. Copyright 2012, STMicroelectronics. Poslední změna 04.06.2013 [cit. 7.5.2014]. Dostupné z: <http://www.st.com/web/en/resource/technical/document/datasheet/DM00037051.pdf>
- [5] STM32F407. *Processor - reference manual* [online]. Copyright 2012, STMicroelectronics. Poslední změna 03.02.2014 [cit. 7.5.2014]. Dostupné z: <http://www.st.com/web/en/resource/technical/document/reference`manual/DM00031020.pdf>
- [6] STM32F407. *Application note - DMA controller* [online]. Copyright 2014, STMicroelectronics. Poslední změna 04.02.2014 [cit. 7.5.2014]. Dostupné z: <http://www.st.com/st-web-ui/static/active/en/resource/technical/document/application`note/DM00046011.pdf>
- [7] CMSIS. *Cortex Microcontroller Software Interface Standard* [online]. ARM. Poslední změna 10.01.2014 [cit. 7.5.2014]. Dostupné z: <http://www.keil.com/pack/doc/CMSIS/General/html/index.html>
- [8] Cortex-M. *Mikrokontroléry STM32-Cortex-M* [online]. Poslední změna 29.12.2009 [cit. 7.5.2014]. Dostupné z: <http://www.fkt.cz/cz/clanky/soucastky/art`132/mikrokontrolery-stm32-cortex-m3.aspx>

- [9] IP kamery. *IP vs. analog kamery a základní pojmy* [online]. [cit. 7.5.2014]. Dostupné z: <http://software-dl.ti.com/ecs/simplelink/cc3000/public/doxygen/API/v1.12/html/index.html>
- [10] CC3000. *Wifi modul - overview of documentation* [online]. Texas Instruments. Poslední změna 13.05.2013 [cit. 7.5.2014]. Dostupné z: <http://processors.wiki.ti.com/index.php/CC3000>
- [11] CC3000. *Wifi modul - datasheet* [online]. Texas: Copyright 2013, Texas Instruments. Poslední změna 4.12.2013 [cit. 7.5.2014]. Dostupné z: <http://www.ti.com/lit/ds/symlink/cc3000.pdf>
- [12] CC3000. *Wifi modul - smart config* [online]. Texas: Copyright 2012, Texas Instruments. Poslední změna 24.04.2014 [cit. 7.5.2014]. Dostupné z: <http://processors.wiki.ti.com/index.php/CC3000`Smart`Config>
- [13] CC3000. *Wifi modul - spi interface* [online]. Texas: Copyright 2013, Texas Instruments. Poslední změna 18.04.2014 [cit. 7.5.2014]. Dostupné z: [http://processors.wiki.ti.com/index.php/CC3000`Serial`Port`Interface`\(SPI\)](http://processors.wiki.ti.com/index.php/CC3000`Serial`Port`Interface`(SPI))
- [14] CC3000. *Wifi modul - picture* [online]. [cit. 7.5.2014]. Dostupné z: <http://media.digikey.com/Photos/Texas%20Instr%20Photos/CC3000EM.JPG>
- [15] CC3000. *Wifi modul - API library* [online]. Texas: Copyright 2013, Texas Instruments. Poslední změna 27.02.2014 [cit. 7.5.2014]. Dostupné z: <http://software-dl.ti.com/ecs/simplelink/cc3000/public/doxygen/API/v1.12/html/index.html>
- [16] CC3000. *Wifi chip - picture* [online]. [cit. 7.5.2014]. Dostupné z: <http://www.embarcados.com.br/wp-content/uploads/2013/10/cc3000`klein.jpg>
- [17] Raspberry Pi. *RTSP stream* [online]. [cit. 7.5.2014]. Dostupné z: <http://www.mybigideas.co.uk/RPi/RPiCamera/>
- [18] Raspberry Pi. *Streaming Video Using VLC Player* [online]. [cit. 7.5.2014]. Dostupné z: <http://www.raspberry-projects.com/pi/pi-hardware/raspberry-pi-camera/streaming-video-using-vlc-player>
- [19] Raspberry Pi. *picture - board* [online]. [cit. 7.5.2014]. Dostupné z: <http://readwrite.com/files/modela.jpg>
- [20] Raspberry Pi. *picture - camera* [online]. [cit. 7.5.2014]. Dostupné z: <http://1.bp.blogspot.com/-rKqnSA5U6sw/UnBwMNTZckI/AAAAAAAAABfM/-n5d60ni6bY/s1600/ras`cam.jpg>

- [21] I2C. *Komunikace po sériové sběrnici* [online]. Pavel Tišnovský. Poslední změna 08.01.2009 [cit. 7.5.2014]. Dostupné z: <http://www.root.cz/clanky/komunikace-po-seriove-sbornici-isup2supc/>
- [22] CMOS vs. CCD. *Porovnání snímacích čipů* [online]. Milan Šurkala. Poslední změna 05.10.2009 [cit. 8.5.2014]. Dostupné z: <http://www.digimanie.cz/fotomobily-snimaci-cipy-cmos-vs-ccd/2885>



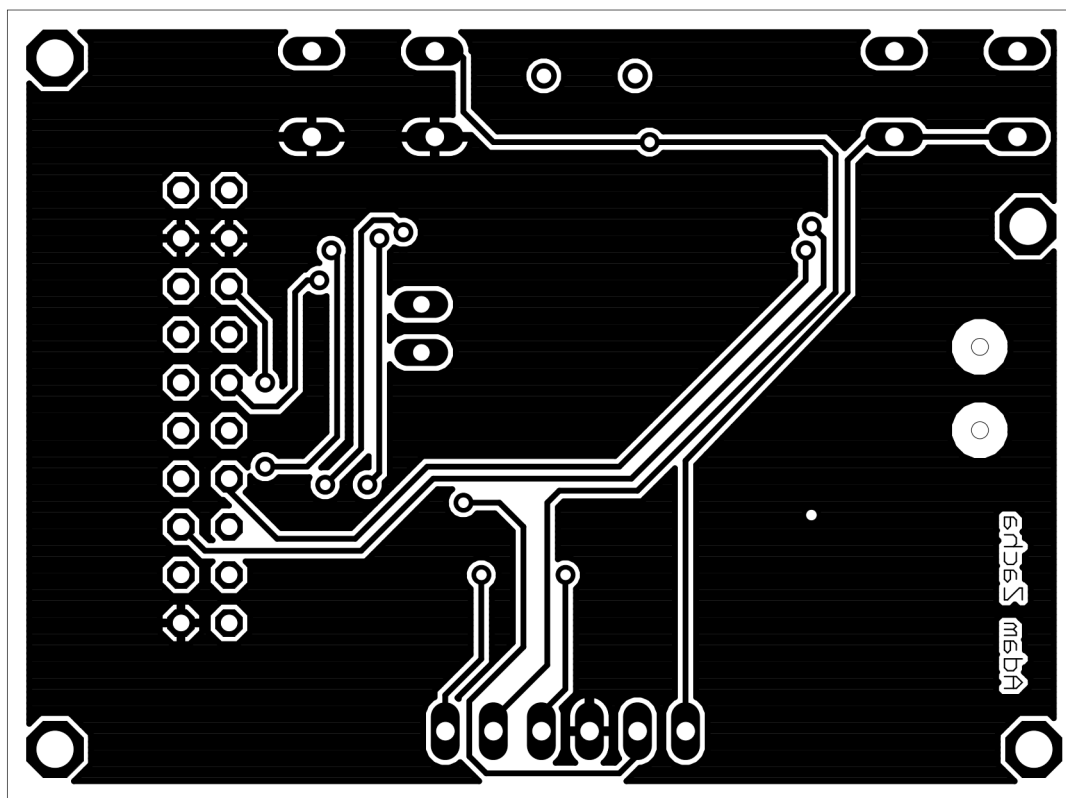
Obr. A.2: Schéma zapojení bezdrátového kamerového modulu 1. část.



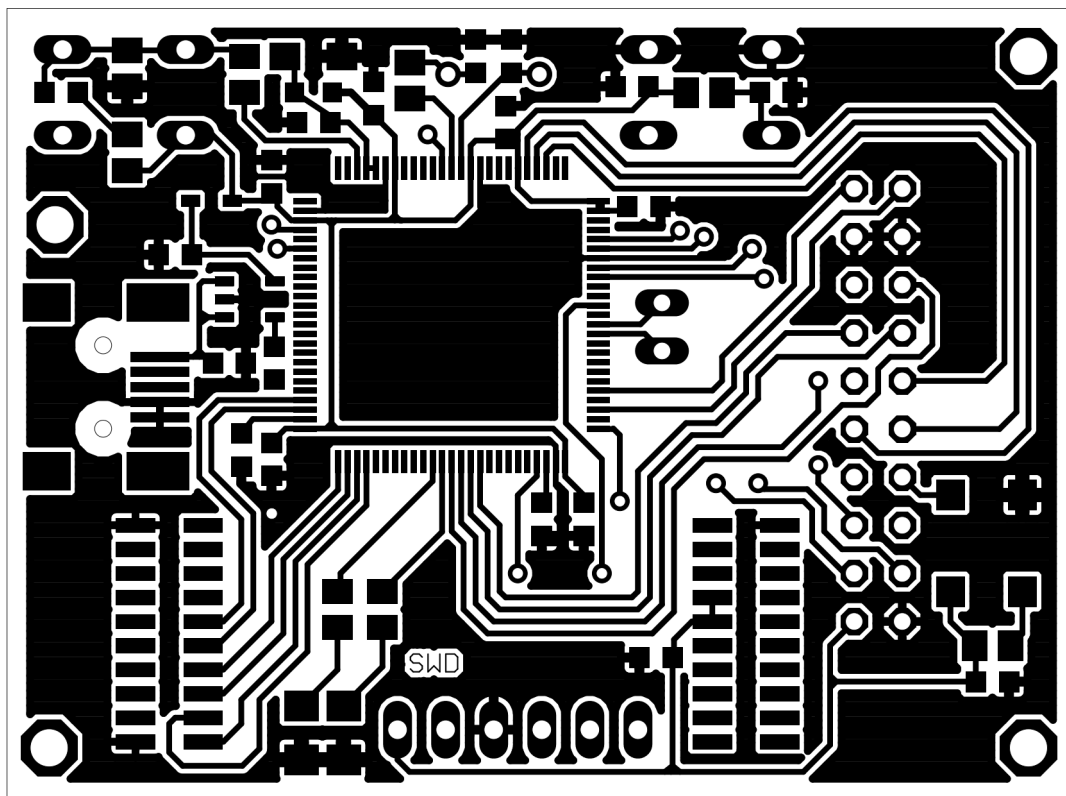
Obr. A.3: Schéma zapojení bezdrátového kamerového modulu 2. část.

Příloha B

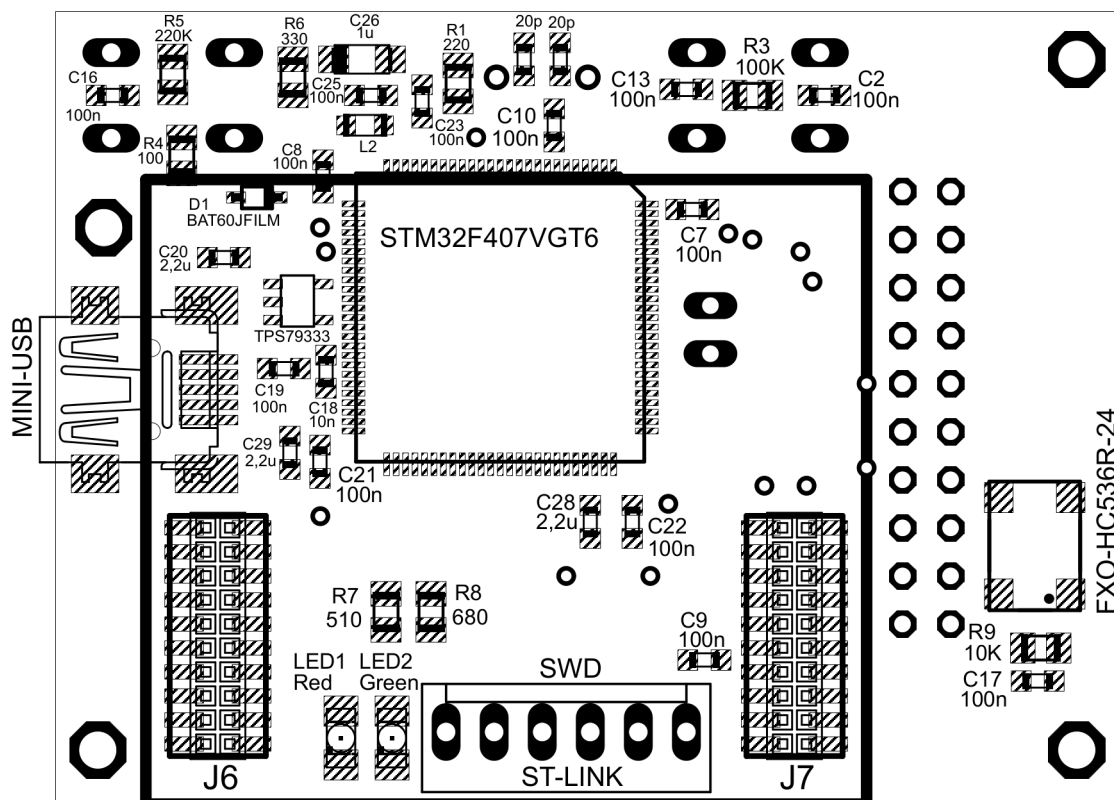
Desky plošných spojů a seznam součástek



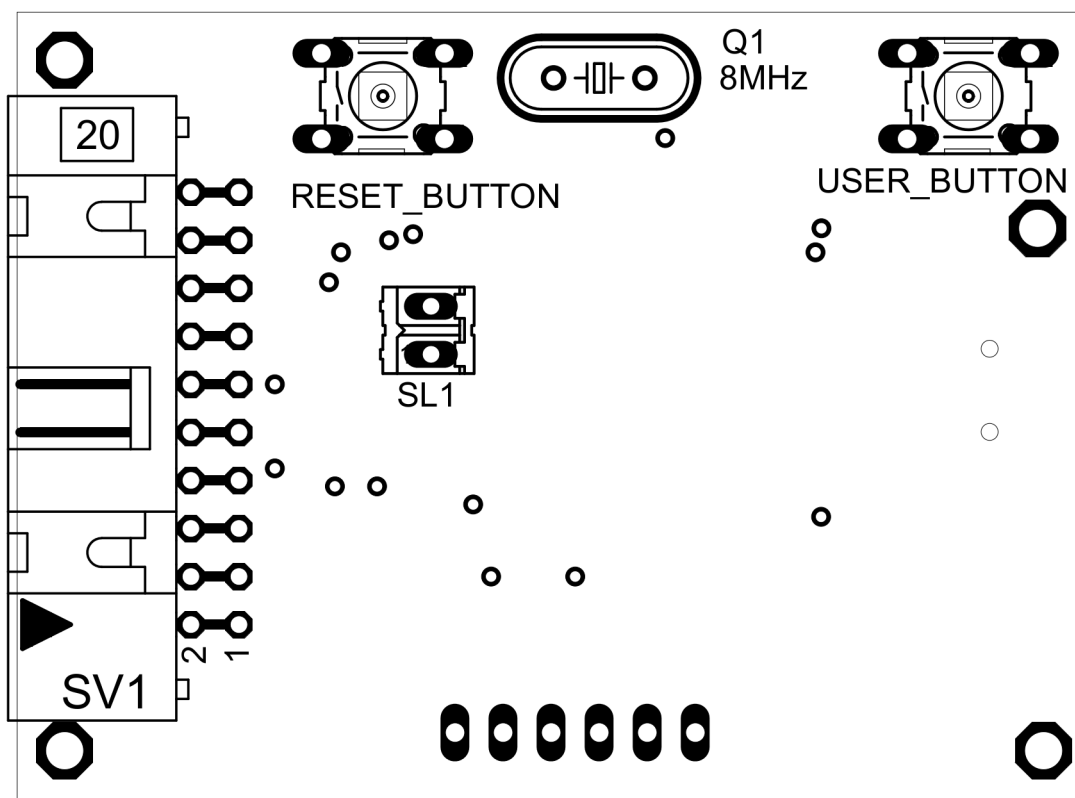
Obr. B.1: Deska plošného spoje bezdrátového kamerového modulu spodní vrstva, zvětšení 2,5x



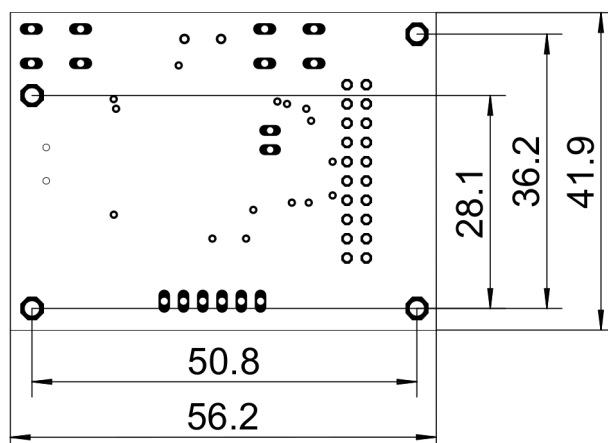
Obr. B.2: Deska plošného spoje bezdrátového kamerového modulu vrchní vrstva, zvětšení 2,5x



Obr. B.3: Osazovací pláněk desky plošného spoje vrchní vrstvy, zvětšení 2,5x



Obr. B.4: Osazovací plánec desky plošného spoje spodní vrstvy, zvětšení 2,5x



Obr. B.5: Umístění konstrukčních otvorů a rozměry DPS bezdrátového kamerového modulu.

Typ součástky	Hodnota	Pouzdro
C2, C7, C9, C10, C13	100nF	0603K
C14	20pF	0603K
C15	20pF	0603K
C16, C17	100nF	0603K
C18	10nF	0603K
C19	100nF	0603K
C20	2,2uF	0603K
C21, C22, C23, C25	100nF	0603K
C26	1uF	0603K
C28	2,2uF	0603K
C29	2,2uF	0603K
D1	BAT60JFILM	SOD323
FXO-HC536R	24MHz	SM77H
J6	konektor	2x10 pin, 1,27mm SMD
J7	konektor	2x10 pin, 1,27mm SMD
LED1	Red 10mA	1206
LED2	Green 10mA	1206
Q1	8MHz	HC49/S
L2	100nH	L2012C
R1	220	M0805
R3	100K	M0805
R4	100	M0805
R5	220K	M0805
R5	330	M0805
R7	510	M0805
R8	680	M0805
R9	10K	M0805
RESET BUTTON	tlačítko	TC-0102-T-A00
USER BUTTON	tlačítko	TC-0102-T-A00
TPS79333	LDO	5SOT-23
USB mini B	zásuvka	SMD
STM32F407VGT6	procesor	LQFP100
SV1	konektor	2x10 pin, 2,54mm
ST-LINK	konektor	1x6 pin, 2,54mm

Tab. B.1: Seznam součástek

Příloha C

Použité skripty, zdrojové kódy

Zdrojové kódy bezdrátového kamerového modulu jsou umístěny na CDROM.