

UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A
INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2023

Dominik Tomášek

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Minimalistická bezpečnostní kamera
Bakalářská práce

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Dominik Tomášek**
Osobní číslo: **I20031**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Komunikační a mikroprocesorová technika**
Téma práce: **Minimalistická bezpečnostní kamera**
Zadávající katedra: **Katedra elektrotechniky**

Zásady pro vypracování

Cílem práce je návrh síťové kamery s minimalistickými požadavky na HW. Teoretická část práce popíše problematiku kamerových snímačů, s ohledem na využívaná rozhraní pro přenos dat z kamery do mikroprocesoru. Na základě tohoto rozboru bude provedena rešerše dostupných procesorů, případně SoM vhodných pro řešení zadání. Praktická část práce provede návrh a konstrukci kamery schopné monitorovat vybraný prostor, buď ve formě kamery streamující kontinuálně obraz, nebo ve formě fotopastí, odesílající snímek například při detekci pohybu.

Rozsah pracovní zprávy: 30-50
Rozsah grafických prací:
Forma zpracování bakalářské práce: tištěná/elektronická

Seznam doporučené literatury:

- [1] VÁŇA, V. Mikrokontroléry ATMEL AVR: popis procesoru a instrukční soubor. Praha: BEN technická literatura, 2003. 336 s. ISBN 978-80-7300-083-0.
- [2] VÁŇA, V. Mikrokontroléry ATMEL AVR: programování v jazyce C. Praha: BEN technická literatura, 2003. 216 s. ISBN 978-80-7300-102-0.
- [3] VLACH, J. Řízení a vizualizace technologických procesů. Praha: BEN technická literatura, 2002. 160 s. ISBN 978-80-86056-66-X.
- [4] BRTNÍK, B. Základní elektronické obvody. Praha: BEN technická literatura, 2011. 156s. ISBN 978-80-7300-408-8
- [5] RIPKA, P.; TIPEK, A. Master Book of Sensors. Praha : BEN, 2003. ISBN 0-12-752184

Vedoucí bakalářské práce: Ing. Pavel Rozsival
Katedra elektrotechniky

Datum zadání bakalářské práce: 15. listopadu 2022
Termín odevzdání bakalářské práce: 12. května 2023

Ing. Zdeněk Němec, Ph.D. v.r.
děkan

L.S.

Ing. Jan Pidanič, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 31. ledna 2023

Prohlašuji:

Práci s názvem Minimalistická bezpečnostní kamera jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 27.4. 2023

Dominik Tomášek

PODĚKOVÁNÍ

Touto cestou bych chtěl poděkovat vedoucímu mé práce Ing. Pavlovi Rozsivalovi, za všechny jeho rady, velkou trpělivost, ochotu a vždy vřelé uvítání s úsměvem na rtech za jakékoliv situace při vedení mé práce.

Dále bych chtěl poděkovat celému kolektivu svých přátel, jak z univerzitního prostředí, tak i domovského prostředí za jejich podporu při všech sociálních akcích během mého studia.

Zvláštní poděkování patří mému spolubydlícímu za tvorbu příjemného prostředí pro psaní této práce.

Jako poslední díky posílám své rodině a to členům, kteří mi dodávali podporu během studia, a i těm členům, kteří o mě pochybovali, „Dostali jste mě tak daleko“.

ANOTACE

Práce je zaměřena na provedení rešerše vhodných jednotlivých kamerových modulů spolu s rešerší na mikroprocesory vhodné pro zpracování obrazu k následujícímu sestavení minimalistické bezpečnostní kamery v podobě kamery či popřípadě fotopasti.

KLÍČOVÁ SLOVA

Kamera, Fotopast, ESP 32 – CAM, Kamerový modul OV2640

TITLE

Minimum HW camera

ANNOTATION

The work is focused on a search for suitable individual camera modules together with a search for microprocessors suitable for image processing for the subsequent assembly of a minimalist security camera in the form of a camera or in the case of a photo trap.

KEYWORDS

Camera, Photo trap, ESP32 – CAM, Camera Module OV2640

OBSAH

| | |
|---|----|
| SEZNAM ILUSTRACÍ A TABULEK | 10 |
| SEZNAM ZKRATEK A ZNAČEK..... | 13 |
| ÚVOD..... | 15 |
| 1.TEORETICKÁ ČÁST..... | 16 |
| 1.1 Kamera..... | 16 |
| 1.1.1 Princip činnosti kamery..... | 17 |
| 1.1.2 O – Objektiv | 17 |
| 1.1.3 IF – Infračervený filtr..... | 17 |
| 1.1.4 S – Kamerový snímač (obrazový snímač)..... | 17 |
| CCD a CMOS..... | 18 |
| 1.1.5 BZ – Blok zpracování obrazu..... | 21 |
| 1.1.6 R – Komunikační rozhraní | 21 |
| 1.2 Přenos dat z kamery do mikroprocesoru..... | 22 |
| 1.2.1 Sériové rozhraní..... | 22 |
| Asynchronní sériové rozhraní | 22 |
| Synchronní sériové rozhraní | 23 |
| SPI rozhraní (SPI interface) | 23 |
| UART rozhraní (UART interface) | 25 |
| SCCB rozhraní (SCCB interface)..... | 27 |
| I2C rozhraní (I2C interface) | 29 |
| SMBus | 30 |
| PMBus | 31 |
| 1.2.2 Paralelní rozhraní..... | 31 |
| Parallel port LPT interface..... | 31 |
| PATA interface..... | 32 |
| PCI interface..... | 33 |
| 1.3 Kamerové komunikační rozhraní | 34 |
| Camera link interface..... | 34 |
| Kompozitní video (CVBS)..... | 36 |
| DVI interface..... | 36 |
| CPI rozhraní (Camera Parallel interface)..... | 37 |
| 1.4 Výběr kamerového modulu..... | 38 |

| | |
|---|----|
| Kamerový modul OV2640..... | 38 |
| Kamerový modul OV7670..... | 39 |
| Kamerový modul PTC08 | 40 |
| Kamerový modul ArduCam-Mini OV5642(Arduino camera) | 41 |
| Kamerový modul JPEG COLOR CAMERA C328..... | 42 |
| 1.5 Výběr hardwaru pro kamerový modul..... | 43 |
| Arduino | 43 |
| ESP 32 / ESP 8266 | 44 |
| Raspberry Pi | 45 |
| STM32 | 46 |
| 2.Praktická část | 48 |
| 2.1 Kamera..... | 48 |
| Seznam použitých součástí..... | 49 |
| Návrh desky plošného spoje (DPS)..... | 50 |
| Specifické části kódu | 52 |
| Uvedení kamery do provozu | 59 |
| 2.2 Fotopast..... | 67 |
| Seznam použitých součástí..... | 67 |
| Návrh desky plošného spoje..... | 68 |
| Specifické části kódu | 68 |
| Uvedení fotopasti do chodu | 71 |
| 3. Naměřené hodnoty a budoucí vylepšení obou zařízení | 74 |
| 3.1 Naměřené hodnoty..... | 74 |
| 3.2 Budoucí vylepšení obou zařízení..... | 77 |
| ZÁVĚR..... | 78 |
| POUŽITÁ LITERATURA | 79 |
| Bibliografie | 79 |

SEZNAM ILUSTRACÍ A TABULEK

| | |
|---|----|
| Obrázek 1. Blokové schéma kamery, zdroj [2]..... | 16 |
| Obrázek 2. Blokové schéma CCD snímače, zdroj [1]..... | 18 |
| Obrázek 3. Blokové schéma CMOS snímače, zdroj [1]..... | 18 |
| Obrázek 4. Princip fungování CCD snímače, zdroj [5]..... | 20 |
| Obrázek 5. Paralelní rozhraní a Sériové rozhraní, zdroj [7] | 22 |
| Obrázek 6. Packet dat pro přenos v asynchronním rozhraní, zdroj [8] | 22 |
| Obrázek 7. SPI rozhraní (konfigurace Master a Slave), zdroj [9]..... | 23 |
| Obrázek 8. Regular mode pro SPI rozhraní, zdroj [9]..... | 24 |
| Obrázek 9. Daisy chain metode pro SPI rozhraní, zdroj [9]..... | 25 |
| Obrázek 10. UART komunikační rozhraní, zdroj [8] | 26 |
| Obrázek 11. UART rozhraní s připojenými datovými sběrnicemi, zdroj [8]..... | 26 |
| Obrázek 12. 2-wire SCCB interface provedení, zdroj [10] | 27 |
| Obrázek 13. 3-wire SCCB interface provedení, zdroj [10] | 28 |
| Obrázek 14. I ² C komunikace mezi hlavním zařízením (Main) a vedlejším zařízením (Node, slave), zdroj [11] | 29 |
| Obrázek 15. Pull-up rezistory a jejich zapojení, zdroj [11]..... | 30 |
| Obrázek 16. Schéma SMBUS topologie s pull-up rezistory a rozdíly v napětí, zdroj [11] | 30 |
| Obrázek 17. Kabeláž pro komunikace LPT interface, zdroj [12] | 32 |
| Obrázek 18. Kabel pro spojení periférií za pomoci PATA interface, zdroj [13]..... | 33 |
| Obrázek 19. PCI Bus a připojované periferie, zdroj [16]..... | 34 |
| Obrázek 20. Přehled portů pro CLI, zdroj [17]..... | 35 |
| Obrázek 21. Vzorek signálu kompozitního videa, zdroj [20]..... | 36 |
| Obrázek 22. Konektory pro DVI kabely, zdroj [21] | 37 |
| Obrázek 23. CPI interface komunikace mezi přijímačem a vysílačem, zdroj [24] | 38 |
| Obrázek 24. Zapojení s OV2640 jako Slave obvodem, zdroj [26]..... | 39 |
| Obrázek 25. Kamerový modul OV2640, zdroj [27]..... | 39 |
| Obrázek 26. Kamerový modul OV7670, zdroj [28]..... | 40 |
| Obrázek 27. Kamerový modul PTC08, zdroj [31]..... | 41 |
| Obrázek 28. Kamerový modul Arducam – mini OV5642, zdroj [34] | 42 |
| Obrázek 29. Kamerový modul JPEG COLOR CAMERA C328, zdroj [35] | 43 |
| Obrázek 30. Vývojový kit Arduino UNO, zdroj [38] | 44 |
| Obrázek 31. Vývojový kit ESP32 WROOM, [43]..... | 45 |
| Obrázek 32. Vývojový kit Raspberry Pi 4, zdroj [46] | 46 |
| Obrázek 33. Kamerový kit od STM32F7, zdroj [49]..... | 47 |
| Obrázek 34. Schéma zapojení obvodu pro kameru..... | 48 |
| Obrázek 35. Cvičné zapojení kamery na nepájivém poli | 49 |
| Obrázek 36. Deska plošného spoje | 51 |
| Obrázek 37. Červená horní strana a modrá dolní strana DPS | 51 |
| Obrázek 38. Seznam použitých knihoven v programu..... | 52 |
| Obrázek 39. Definování maker pro GPIO piny | 52 |
| Obrázek 40. Nastavení GPIO pinů do režimu OUTPUT | 53 |
| Obrázek 41. Nastavení pinu pro externí probuzení..... | 53 |
| Obrázek 42. String s údaji pro spuštění CallmeBota | 53 |
| Obrázek 43. Funkce Void uspávání(); pro uspávání zařízení..... | 53 |
| Obrázek 44. Void print_wakeup_reason() pro výpis důvodu probuzení | 54 |

| | |
|---|----|
| Obrázek 45. Void print_reset_reason(RESET_REASON reason) pro výpis důvodu restartu | 54 |
| Obrázek 46. Aktivace služby CallmeBot | 55 |
| Obrázek 47. Funkce pro posílání zprávy na WhatsApp aplikaci | 55 |
| Obrázek 48. Definování vláken | 56 |
| Obrázek 49. Vytvoření Vlakyko1_fce | 56 |
| Obrázek 50. Obsah prvního vlákna | 57 |
| Obrázek 51. Vytvoření Vlakyko2_fce | 57 |
| Obrázek 52. Obsah druhého vlákna | 58 |
| Obrázek 53. Připojení k internetu pomocí WiFiManager knihovny | 58 |
| Obrázek 54. Teoretický obrázek propojení Arduina Uno a ESP32-CAM pro účely programování (na obrázku jsou znázorněny piny na propojení) | 59 |
| Obrázek 55. Vypsání hlášek do sériového terminálu po spuštění | 60 |
| Obrázek 56. Automatické připojení do sítě | 61 |
| Obrázek 57. Nezdařené připojení do sítě a spuštění access pointu pro konfiguraci připojení | 61 |
| Obrázek 58. Ukázka Wi-Fi sítě spolu s vytvořením AutoConnectAP | 61 |
| Obrázek 59. Přihlašovací údaje pro přihlášení do access pointu | 62 |
| Obrázek 60. Stránka WiFiManageru pro konfiguraci přihlašovacích údajů sítě | 62 |
| Obrázek 61. Stránka WiFiManageru pro připojení k síti | 63 |
| Obrázek 62. Úspěšné uložení Wi-Fi údajů do ESP32-CAM | 64 |
| Obrázek 63. Vypsání IP adresy do sériového portu a odeslání zprávy do WhatsAppu, na které je možno sledovat obraz z kamery | 64 |
| Obrázek 64. Zpráva na WhatsAppu s IP adresou kamery | 64 |
| Obrázek 65. Snímek z přenosu kamery na webserveru | 65 |
| Obrázek 66. Oznámení příchozího usnutí do režimu deepsleep | 66 |
| Obrázek 67. Postupné uspávání kamery | 66 |
| Obrázek 68. Cvičné zapojení fotopasti na nepájivém poli | 67 |
| Obrázek 69. Použité knihovny pro fotopast | 69 |
| Obrázek 70. Definování ukazatele vlákna | 69 |
| Obrázek 71. Funkce pro vypsání důvodů probuzení | 69 |
| Obrázek 72. Funkce pro vypsání důvodů pro resetování zařízení | 70 |
| Obrázek 73. Definování externího probuzení na pinu GPIO 13 | 70 |
| Obrázek 74. Obsah vlákna | 70 |
| Obrázek 75. Funkce pro uspávání fotopasti | 70 |
| Obrázek 76. Teoretický obrázek propojení Arduina Uno a ESP32-CAM pro účely programování (na obrázku jsou znázorněny piny pro propojení) | 71 |
| Obrázek 77. Výpis důvodu probuzení a spuštění fotopasti | 72 |
| Obrázek 78. Výpis sériového terminálu od fotopasti | 72 |
| Obrázek 79. Načtená microSD karta s uloženými fotografiemi | 73 |
| Obrázek 80. Obrázky pořízené fotopastí | 73 |
| Obrázek 81. Měření odběru proudu na cvičném zapojení kamery | 75 |
| Obrázek 82. Měření odběru proudu na fotopasti z ESP32-CAM | 75 |
| Obrázek 83. Měření odběru proudu na cvičném zapojení fotopasti | 75 |
| Obrázek 84. Měření času pro fotopast | 76 |
| Tabulka 1. Tabulkové porovnání CCD a CMOS, zdroj [1] | 19 |
| Tabulka 2. Porovnání jednotlivých kamerových modulů | 43 |

| | |
|--|----|
| Tabulka 3. Porovnání jednotlivých hardwarových desek..... | 47 |
| Tabulka 4. Seznam použitých součástek | 50 |
| Tabulka 5. Seznam použitých součástek | 68 |
| Tabulka 6. Měření odběru proudu na kameře z ESP32-CAM..... | 74 |
| Tabulka 7. Měření času pro fotopast..... | 77 |

SEZNAM ZKRATEK A ZNAČEK

\overline{CS} – Chip Select

SCLK – Select Clock

MOSI – Main out Slave in

MISO – Main in Slave out

Rx – Receiver

Tx – Transmitter

SPI – Serial Peripheral Interface

UART – Universal Asynchronous Receiver Transmitter

I2C – Inter – Integrated Controller

SCCB – Serial Camera Control Bus

CMOS – Complementary Metal Oxid Semiconductor

CCD – Charge – Coupled Device

EEPROM – electrically erasable programmable read-only memory

A/D převodník – Analogově/ Digitální převodník

D/A převodník – Digitálně / Analogový převodník

SoM – System on Module

BLC - Backlight Compensation

WDR - Wide Dynamic Range

HLC - Highlight Compensation

PMBus - Power Management Bus

SMBus – System Management Bus

UXGA – Ultra Extended Graphics Array

CIF – Common Intermediate Format

SVGA – Super Video Graphics Array

VGA – Video Graphics Array

HD – High Definition quality resolution

TTL – Transistor – Transistor – logic

MSB – Most Significant bit

LSB – Least significant bit

LPT – Line Print Terminal

PATA – Parallel Advanced Technology Attachment

PCI – Peripheral Component Interconnect

CVBS – Composite Video Baseband Signal ; Color, Video, Blanking and sync

CPI – Camera Parallel Interface

DVI – Digital Visual Interface

PIR- Passive InfraRed Sensor

CRT – Cathode – Ray Tube

LCD – Liquid Crystal Display

ÚVOD

Vzhledem ke zvýšenému množství krádeží a následně vzniklým škodám na majetku v posledních letech výrazně roste poptávka po bezpečnostních kamerových systémech, které umožňují monitorování soukromých objektů či pozemků.

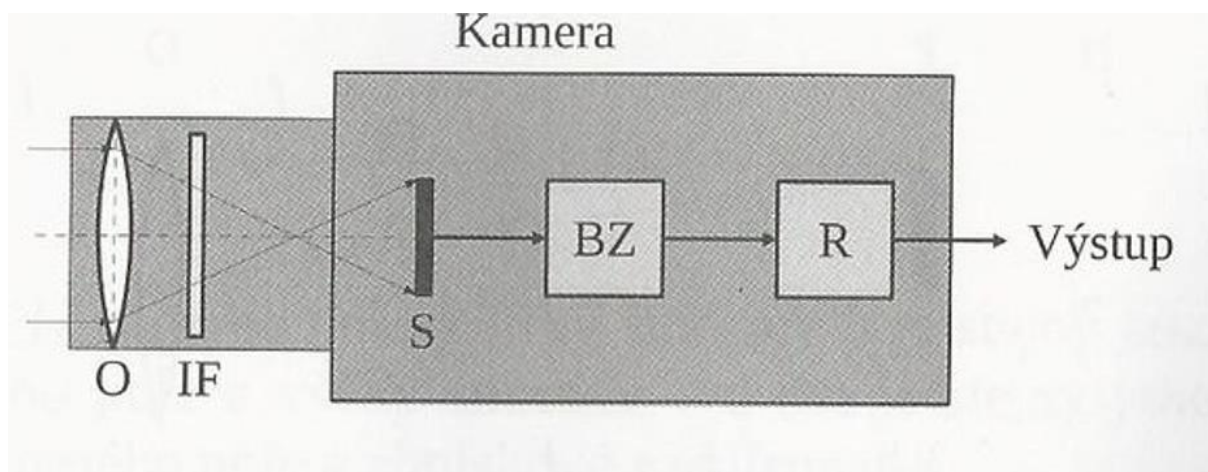
Cílem práce je navrhnout a zkonstruovat minimalistickou bezpečnostní kameru, určenou pro monitorování okolí. Zařízení je tvořeno dvěma samostatnými jednotkami. První jednotkou je kamera určená pro monitorování dění ve vnitřních prostorech budovy a také ve venkovním prostředí na lokální síti. Druhou jednotkou je fotopast, určená ke sledování dění života lesní zvěře a současně k její detekci, s možností ukládání zjištěných dat na microSD kartu.

1. TEORETICKÁ ČÁST

V teoretické části se budu zabývat problematikou kamerových snímačů s ohledem na využívaná rozhraní pro přenos dat z kamery do mikroprocesoru. Na základě tohoto rozboru bude provedena rešerše dostupných procesorů, případně SoM vhodných pro řešení zadání.

1.1 Kamera

Kamera je elektrické zařízení, jehož úkolem je převést světelné záření vycházející ze sledovaného prostoru na elektrický signál, který se následně pomocí určité technologie posílá na sledovací zařízení. Podívejme se na obrázek blokového schématu (viz obrázek 1.) kamery, kde můžeme vidět, že se kamera skládá z pěti částí, každou z těchto částí popíši.



Obrázek 1. Blokové schéma kamery, zdroj [2]

O – objektiv

IF – infračervený filtr

S – kamerový snímač nebo jinak nazýván obrazový snímač

BZ – blok zpracování obrazu

R – komunikační rozhraní

1.1.1 Princip činnosti kamery

Světelné záření se soustředí přes objektiv O na obrazový snímač S. Pro případ denního režimu se před S snímač přidává IF infračervený filtr, který je zde od toho, aby znemožnil průchod fotonů infračerveného záření, jenž by vedl k následnému zkreslení obrazu. Obrazový snímač převede dopadající světelné záření na elektrický signál, následně jde do BZ, tedy bloku zpracování. V bloku zpracování je tento signál převeden na obraz, poté podle potřeby upraven a zkomprimován. Výsledný obraz je pomocí R či komunikačního rozhraní opět převeden na signál vhodný pro přenos a vyšle jej do cílového sledovacího zařízení.

1.1.2 O – Objektiv

Objektiv je čočka nebo v některých případech soustava čoček, kterou se světelné záření ze sledovaného prostoru přenáší světelné paprsky na matici na světlo citlivých buněk obrazového snímače. [2]

1.1.3 IF – Infračervený filtr

Infračervený filtr se dává před obrazový snímač pro znemožnění průchodu fotonů infračerveného záření za denního používání. Pokud by kamera neobsahovala tento filtr, výsledný obraz by byl zkreslen za pomoci právě zmíněných infračervených fotonů. Ovšem jinak se to řeší při nočním režimu kamery, kdy je tento filtr dán stranou, jelikož se v noci světelné fotony nevyskytují, na druhou stranu je dostatek infračervených fotonů, čehož využívá kamerový snímač CMOS. [2]

1.1.4 S – Kamerový snímač (obrazový snímač)

Snímač přeměňuje obraz z optického elektromagnetického záření ze spektra elektromagnetického záření na elektrický signál, který posílá dále do systému. Snímač má v sobě uspořádány jednotlivé buňky do tzv. rastru. Nejčastěji do dvojrozměrného maticového tvaru. Jednotlivé buňky jsou odborně nazývány pixely, jež měří dopadající záření v časovém intervalu.

Kamerový snímač se řadí do skupiny aktivních snímačů, což znamená, že snímače se chovají při snímání dané veličiny jako zdroj elektrické energie. [3] [4] V našem případě vytváří snímač elektrickou energii pomocí vnitřního fotovoltického jevu na fotodiodě.

V současné době máme dva nejrozšířenější obrazové snímače, a to CMOS a CCD. Oba dva pracují na principu fotovoltického jevu. Foton po vzájemném působení s atomem je schopný přenést některý z jeho elektronů ze základního stavu do excitovaného stavu. V polovodiči se dokáže přitáhnout uvolněný valenční elektron k jejich pozičně přiloženým kladným elektrodám, jež jsou od polovodiče izolovány. U CCD snímače je tato elektroda izolována tenoučkou vrstvou oxidu křemičitého. Tím, že jsou valenční elektrony v pohybu, vzniknou potenciální jamky, tzv. miniaturní obrazové buňky neboli pixely, a ty jsou poté vyčítány, zesíleny a digitalizovány [2].

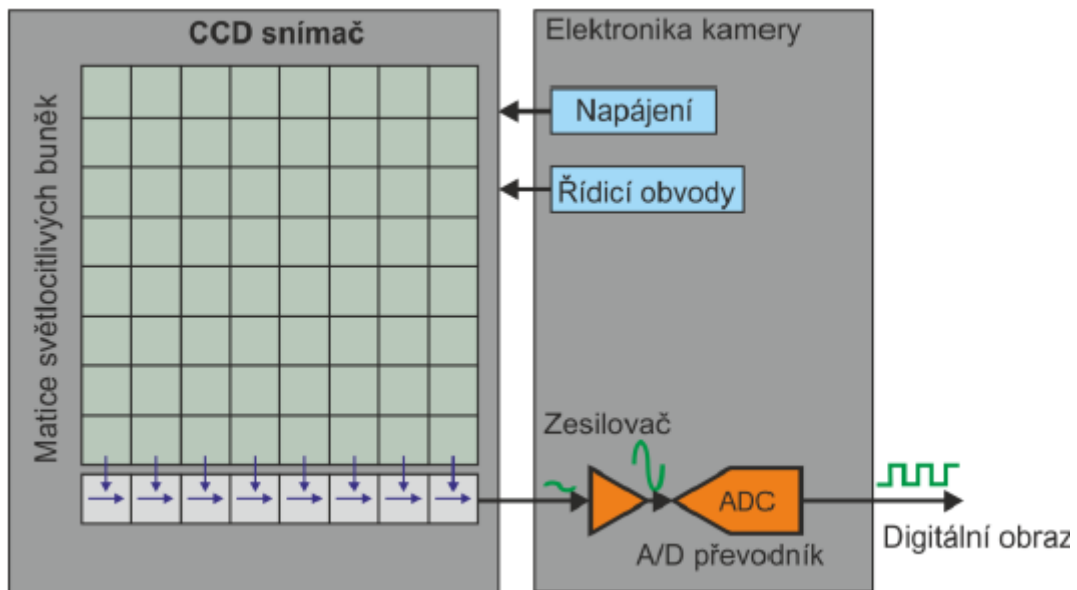
CCD a CMOS

CMOS – celým názvem Complementary Metal Oxid Semiconductor active pixel sensor

CCD – celým názvem Charge Coupled Device

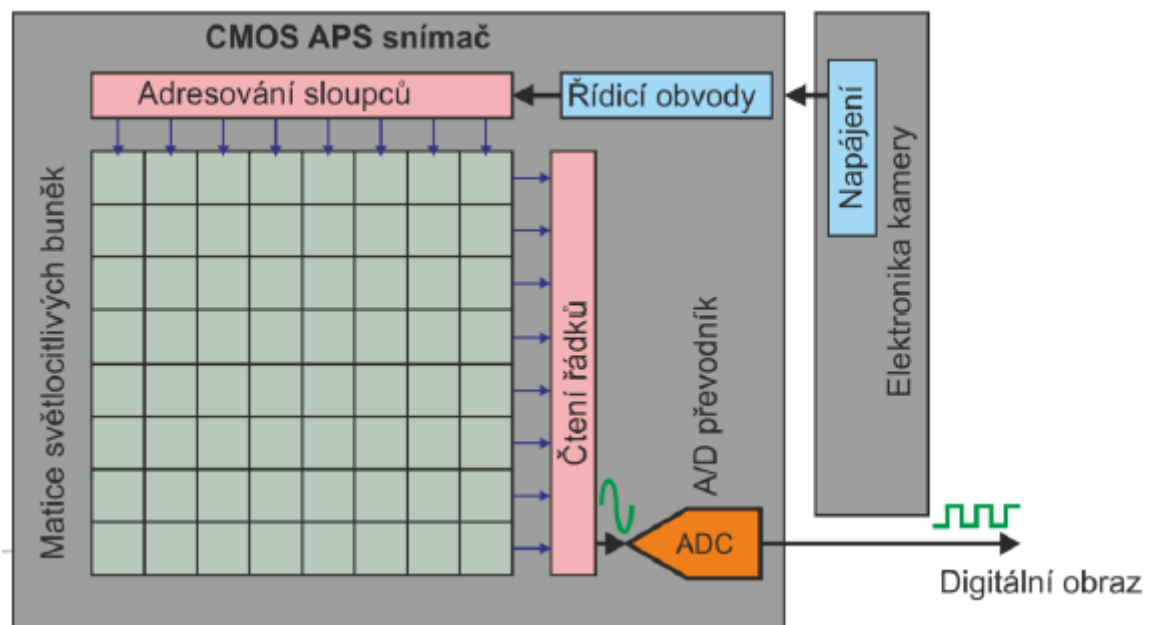
Hlavní rozdíl mezi CCD a CMOS spočívá ve vyčítání elektronů.

Pro výčet elektronů u CCD je použit posuvný registr, kde jsou buňky postupně vyčítány, zesilovány a za pomoci A/D převodníku následně zpracovány do digitální podoby. Viz obrázek 2.



Obrázek 2. Blokové schéma CCD snímače, zdroj [1]

CMOS vyčítá buňky přímo, každá buňka má svůj zesilovač a A/D převodník, jenž nám dovoluje jejich přímé adresování a čtení pomocí souřadnic X a Y na snímači. Viz obrázek 3.



Obrázek 3. Blokové schéma CMOS snímače, zdroj [1]

Dříve spíše dominovaly CCD, jenže v současné době jsou vytlačeny CMOS. CMOS nabízí zejména vysokou rychlost snímání obrazu s menší spotřebou za přijatelnou cenu.

Pro srovnání se podívejme na tabulku, kde se porovnávají jednotlivě snímače CMOS a CCD

| Parametr | CCD | CMOS APS |
|-----------------------------------|--|---|
| Signál vystupující z pixelu | Paket elektronů | Napětí |
| Signál vystupující z čipu | Napětí (analogový signál) | Bity (digitální signál) |
| Rychlost | Nízká, sekvenční čtení | Vysoká, maticově adresované buňky |
| Citlivost | Velmi dobrá, lze dosáhnout kvalitního obrazu, lepší barevná věrnost snímků | Za snížených světelných podmínek může být problém dosáhnout kvalitního zobrazení, horší rozlišení barev |
| Dynamický rozsah | Vysoký | Průměrný |
| Šum | Malý, velká kvalita obrazu | Je větší s ohledem na nižší fill factor (část světlocitlivé buňky je zakryta elektronikou) |
| Odběr energie | Odebírá více (uvádí se až 50 x více než CMOS APS) | Relativně malý |
| Složitost čipu a náklady na vývoj | Menší | Velké |
| Složitost systému | Velká, mnoho obvodů je mimo čip | Menší, většina obvodů je přímo na čipu |
| Cena | Vysoká, jedná se o specializovanou výrobní technologii | Levnější, je využíváno standardní technologie výroby logických obvodů |

SENZORIKA PRO VIZUALIZACI TECHNOLOGICKÉ SCÉNY| ZS 2019

Tabulka 1. Tabulkové porovnání CCD a CMOS, zdroj [1]

Princip činnosti CCD snímače

Činnost CCD snímače je složena ze tří částí: Příprava CCD, Expozice obrazu a Snímání obrazu.

Příprava CCD

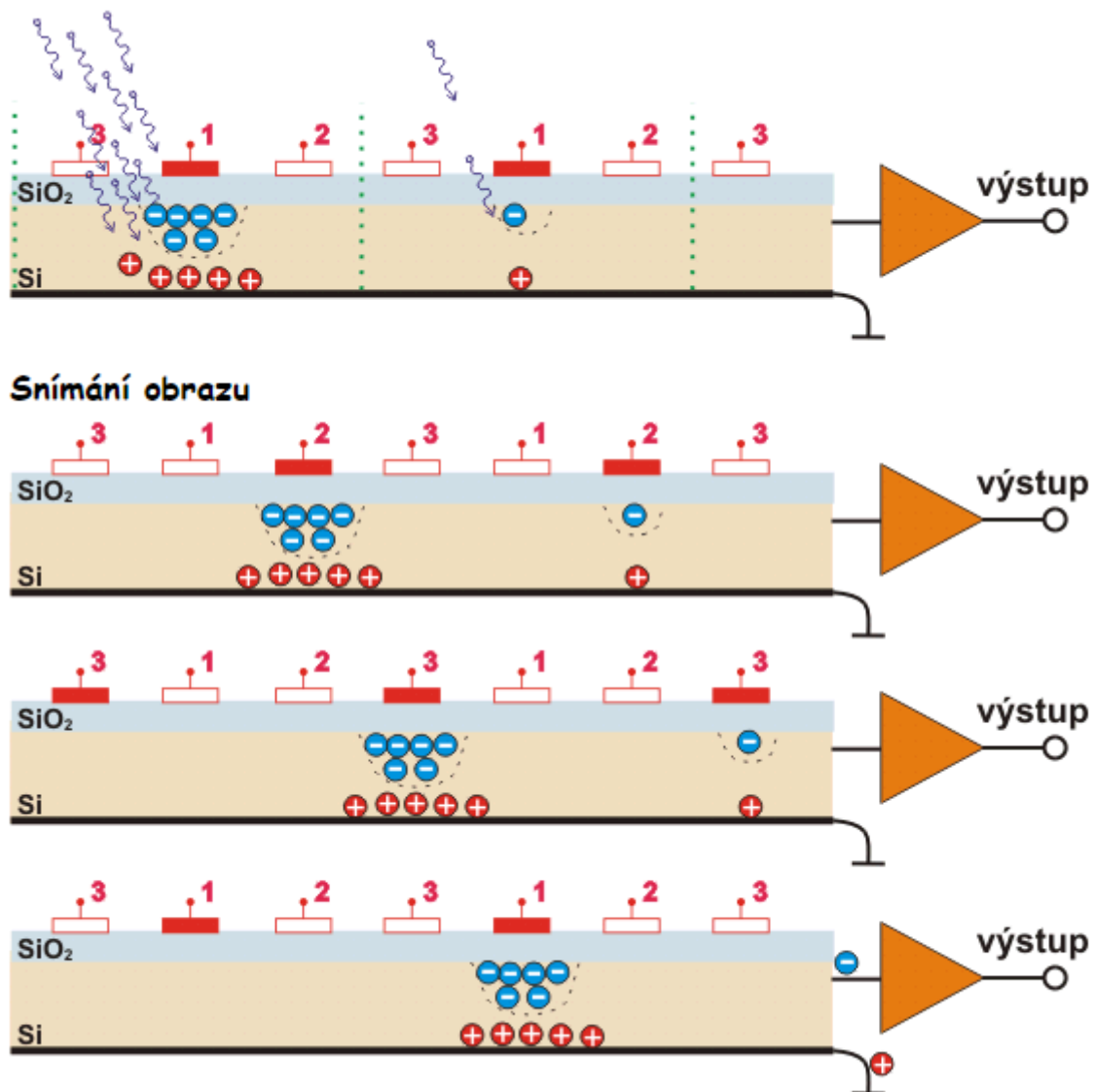
V této fázi jsou v CCD snímači odstraněny všechny zbylé volné elektrony za pomoci toho, že je mu odebrán přístup k světlu, a tím se smaže jakýkoliv přebytek z posledního nasnímaného obrazu. Zkráceně řečeno dojde k smazání posledního obrazu.

Expozice obrazu

Na elektrody označené číslem 1 se přivede kladné napětí a CCD snímač se vystaví světlu, jinak řečeno necháme na něj dopadat fotony, a otevře se tím závěrka. Fotony, které dopadají na senzor v polovodičích, excitují elektrony, jež jsou následně přitahovány ke kladně nabitým elektrodám. Po těchto uvolněných elektronech zůstanou v polovodiči díry, které mají vůči svému okolí kladně nabitý náboj, a tyto díry jsou obráceně přitahovány k elektrodě nacházející se ve spodní části CCD snímače.

Závěrka může být buď rolovací, anebo celková. Rolovací závěrka přenáší data z celého snímače najednou, myšleno, že přenáší data z horní, střední i spodní části skoro v jeden okamžik, jakmile na ni zasvítí světlo. Rolovací závěrka se liší tím, že postupně nahrává

jednotlivé řádky od shora dolů, což se nevyplácí třeba pro kamery s pohybem či pro kamery snímající pohyb aut. Ovšem je to výrazně levnější varianta než celková závěrka.



Obrázek 4. Princip fungování CCD snímače, zdroj [5]

Snímání obrazu

Jakmile se závěrka zavře, spustí se třífázový hodinový impuls, který je přiveden na skupiny elektrod 1, 2 a 3, z čehož plyne, že na elektrodách označené 2 se začne zvyšovat napětí a na elektrodách označené 1 se začne snižovat napětí, vše se děje zároveň. To nám zaručí, že skupinky elektronů jsou přitahovány k elektrodám označené 2. Poté se celý děj opakuje mezi elektrodami 2 a 3, 3 a 1 atd. Skupinky elektronů z jednotlivých pixelů se tak přesouvají z jednoho sousedního pixelu do druhého k výstupnímu zesilovači. Zesilovač následně zesílí malý proud, který se shoduje s počtem příchozích elektronů z jednotlivých pixelů, na úroveň napětí, jež jsou vhodné pro další zpracování obrazu. [5]

1.1.5 BZ – Blok zpracování obrazu

Výstup ze snímače se zpracovává na bloku zpracování obrazu, který tvoří digitální signálový procesor, v němž se provádí interpolace, komprese obrazu, korekce barev a další pokročilé funkce, jako třeba BLC, WDR a HLC.

Interpolace je operace, kdy se odhadují hodnoty neznámých barevných složek z hodnot odpovídajících barevných složek sousedních buněk.

BLC tzv. kompenzace protisvětla, anglicky Backlight Compensasion. Tato funkce zajišťuje potřebnou kvalitu obrazu v případě protisvětla. Protisvětlo je světlo, jehož zdroj se nachází za zorným polem kamery.

WDR nebo taky jasová adaptace, anglicky Wide Dynamic Range, je funkce, jež zajišťuje dostatečnou kvalitu obrazu v případě, kdy se v zorném poli kamery nacházejí současně velmi světlé a velmi tmavé části. Postupně nahrazuje BLC.

HLC kompenzace přsvícení, anglicky Highlight Compensation. Funkce zajišťuje dostatečnou kvalitu obrazu, pokud se v zorném poli kamery nacházejí zdroje intenzivního světla jako, jsou třeba reflektory na domu nebo reflektory na vozidle. [2]

1.1.6 R – Komunikační rozhraní

Rozhraní je zařízení či program, jenž zajišťuje správný přenos dat a komunikaci mezi jednotlivými zařízeními nebo programy. [6]

Úkolem komunikačního rozhraní je převést obraz na signál, jenž se přenáší na sledovací jednotku, jako je třeba mobilní telefon nebo monitor či televize, v dohledovém centru. Pro způsob přenosu máme dvě možnosti, buď použít metalické vedení, jako třeba vhodně vybraná kroucená dvojlinka, koaxiální kabel, anebo bezdrátovou komunikaci v podobě Wi-Fi.

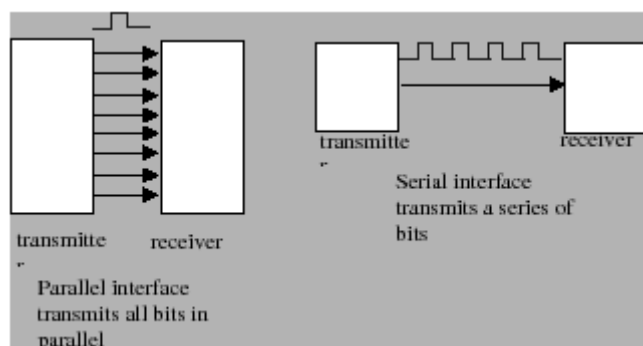
Máme dvě hlavní komunikační rozhraní, digitální a analogové.

Analogové rozhraní je převod obrazu na analogový signál, protože z bloku zpracování obrazu vyjdou data v digitální podobě. Až v tomto rozhraní se data převedou pomocí vhodného kodéru na analogový signál HD anebo již skoro vytlačený SD. V analogovém rozhraní se využívá koaxiálního kabelu s bajonetovou koncovkou. Délka kabeláže může být podle variant HD systému od 400 m až do 500 m.

Digitální rozhraní je převod obrazu z digitální podoby na digitální signál, jako přenosová zařízení se používají nejznámější rozhraní, např. Ethernet se standardem 100BASE-TX, jenž poskytuje rychlost přenosu až 100 Mb/s na dosah 100 m. Dále lze použít mnohem jednodušší řešení v podobě Wi-Fi, ta ovšem dosahují mnohem menších dosahů a rychlosti přenosu dat v rádech 54 Mb/s na 30 m. [2]

1.2 Přenos dat z kamery do mikroprocesoru

Data z kamery nebo jinak řečeno z kamerového modulu přenášíme pomocí rozhraní. Jinými slovy použijeme mezi dvěma digitálními systémy komunikační rozhraní v digitální formě. Pro tento přenos máme dva druhy rozhraní, sériové a paralelní. Hlavní rozdíl je ten, že sériové rozhraní používá k přenosu pouze jeden vodič, ale zato paralelní rozhraní má na přenos více vodičů. Sériové rozhraní vyše najednou sérii bitů. Paralelní rozhraní najednou vyše všechny bity paralelně.



Obrázek 5. Paralelní rozhraní a Sériové rozhraní, zdroj [7]

1.2.1 Sériové rozhraní

Sériové rozhraní umožňuje komunikaci mezi dvěma digitálními systémy za pomoci sérii napěťových impulsů TTL logiky na jednom vodiči. 1 pro vysokou hodnotu, tedy od 2,5 V do 5 V, a 0 pro nízkou hodnotu napětí. Už bylo řečeno, že sériové rozhraní vyše sérii bitů po vodiči tak, jak je to znázorněno na obrázku. Dále je důležité říci, že sériové rozhraní rozlišujeme na asynchronní a synchronní. Příklady sériových rozhraní jsou SPI, I²C, SCCB a UART. Hlavní potíží se sériovým rozhraním je to, že nevíme, kde nám leží naše potřebné bity na vodiči, příklad může být takový, že vyšleme tři série bitů za sebou jdoucích, ale jak určíme, kde začínají a kde končí, co je LSB a MSB, přišly nám všechny bity? Právě kvůli těmto otázkám jsou zavedené tzv. protokoly.

Asynchronní sériové rozhraní

Zkratkou nazváno SCI. SCI obsahuje paralelně/sériový převodník, který slouží jako vysílač. Následně obsahuje sériově/paralelní převodník sloužící jako přijímač. SCI může pracovat v poloduplexním režimu, kdy současně je pouze vysílač, anebo pouze přijímač, nebo také může pracovat v plněduplexním režimu, kdy současně pracuje jako vysílač a přijímač. Data jsou vysílána v tzv. frames do češtiny přeloženo rámce. Frame je celistvý a nerozdělitelný packet bitů. Frame v sobě obsahuje kontrolní bity a informační bity. Kontrolní bity jsou Start bit, Stop bit a Parity a informační bity jsou ty, jež přenášejí data. Start bit je signalizace začátku framu, Stop bitem značíme konec framu. Parity bit je speciální bit, který používáme pro detekci přenosových chyb.



Obrázek 6. Packet dat pro přenos v asynchronním rozhraní, zdroj [8]

Čtení dat se spustí Start bitem, jakmile Rx zaznamená Start bit, začne číst data, která mu přicházejí v pravidelných intervalech, jež jsou nastaveny pomocí přijímacích hodin na Rx. Rx a Tx se předem musí domluvit na přenosové rychlosti komunikace, která se udává v Baudech.

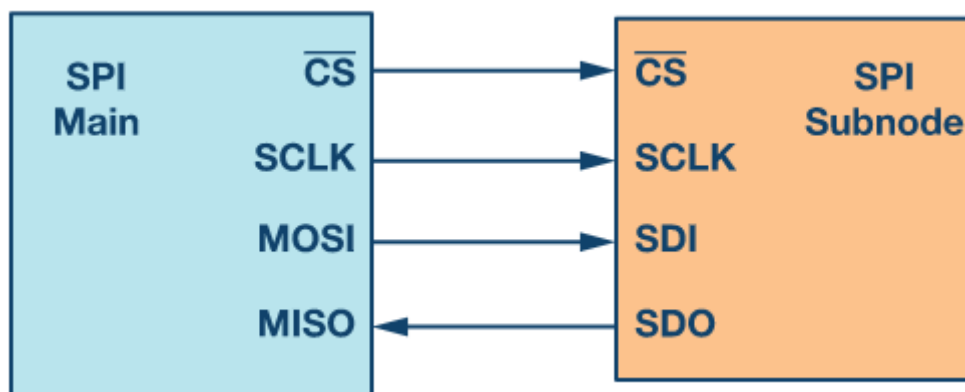
Podle SCI není potřeba synchronizovat své hodiny před zahájením komunikace. To znamená, že Tx vysílá data bez dohody na synchronizaci hodin s Rx. Rx přijímá data v asynchronním pořadí od Tx, pouze čeká až na Start bit.

Synchronní sériové rozhraní

Na rozdíl od asynchronních sériových rozhraní synchronní sériové rozhraní nemá žádné vnitřní hodiny, což znamená, že Rx nemůže nezávisle synchronizovat čtení dat z vodiče podle přenosové rychlosti Tx. Rx v tomto případě čeká na hodinový signál od Tx. Hodinový signál říká, kdy Rx může číst data z vodiče, z čehož vyplývá, že Tx a Rx se musí synchronizovat pomocí hodinového impulsu, aby byl uskutečněn přenos dat.

SPI rozhraní (SPI interface)

SPI rozhraní se používá pro komunikaci, když mikroprocesor je jako Master neboli hlavní jednotkou a musí vysílat data do zařízení, jež nemají vnitřní hodiny, tato zařízení se nazývají Slave či jinak nazývány ostatní obvody, jako ostatní obvody jsou brány EEPROM paměti, A/D převodník, D/A převodník, displeje, posuvný registr a senzory. Jedná se zde o jedno z nejrozšířenějších rozhraní, máme buď provedení se třemi vodiči, anebo provedení se 4 vodiči, ale minimálně potřebují dva vodiče, a to jsou data a hodinový impuls. Vodič s daty obsahuje bity a vodič s hodinovým impulsem udává Slave, kdy může číst a zapisovat data. SPI je plně duplexní, znamená to tedy, že data z Master a Slave se mohou vysílat a přijímat současně. SPI zařízení podporují větší frekvenci při porovnání s I^2C .



Obrázek 7. SPI rozhraní (konfigurace Master a Slave), zdroj [9]

Na obrázku vidíme SPI rozhraní ve 4vodičovém provedení. V tomto provedení máme 4 vodiče. První vodič je označen zkratkou \overline{CS} (chip select), druhý vodič je SCLK, který reprezentuje

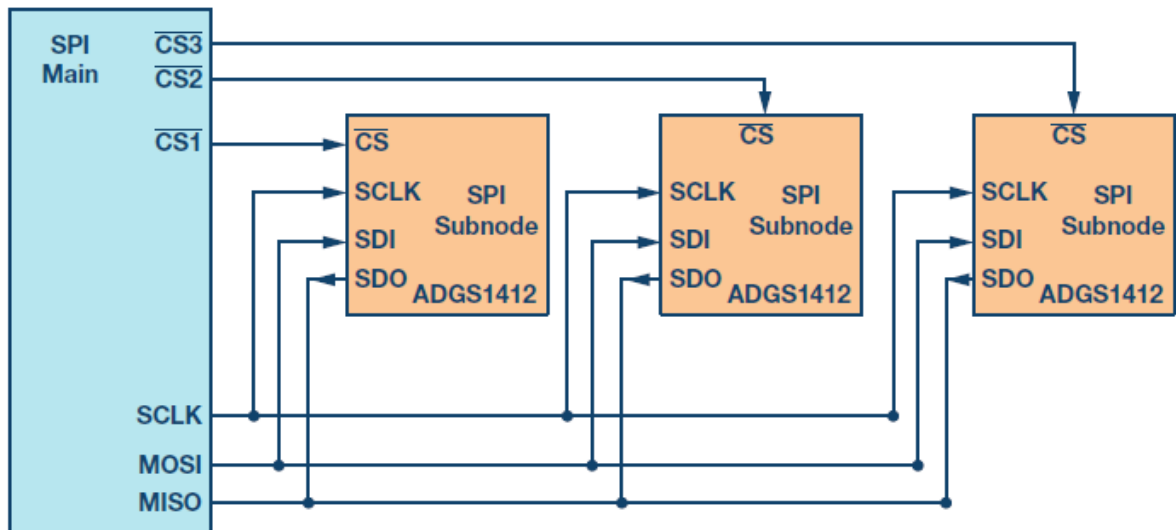
hodinový signál. Poslední dva vodiče jsou MOSI a MISO a slouží jako datové vodiče pro přenos dat mezi Master a Slave.

Princip datového přenosu funguje následovně:

Master zařízení pomocí \overline{CS} vodiče vybere Slave zařízení, následně na toto vybraní pošle po vodiči SCLK hodinový impuls, aby se spolu synchronizovaly buď na náběžnou hranu hodinového impulsu, nebo na sestupnou hranu hodinového impulsu. Pokud je více Slave zařízení, každé z nich dostane individuální \overline{CS} signál od Master zařízení. Poté, co si Master zvolí Slave zařízení, mohou si mezi sebou posílat a přijímat informace současně, a to díky MISO a MOSI vodičům. Během komunikace jsou data současně sériově vysílána po MISO/SDI datové sběrnici a přijímána na datovou sběrnici MISO/SDI, kde jsou vzorkována nebo načítána.

SPI rozhraní nám dovoluje mít vícero Slave zařízení, ale vždy pouze jedno Main zařízení. Mnohonásobné zapojení Slave zařízení se buď provádí v tzv. Regular mode, anebo v Daisy-chain method.

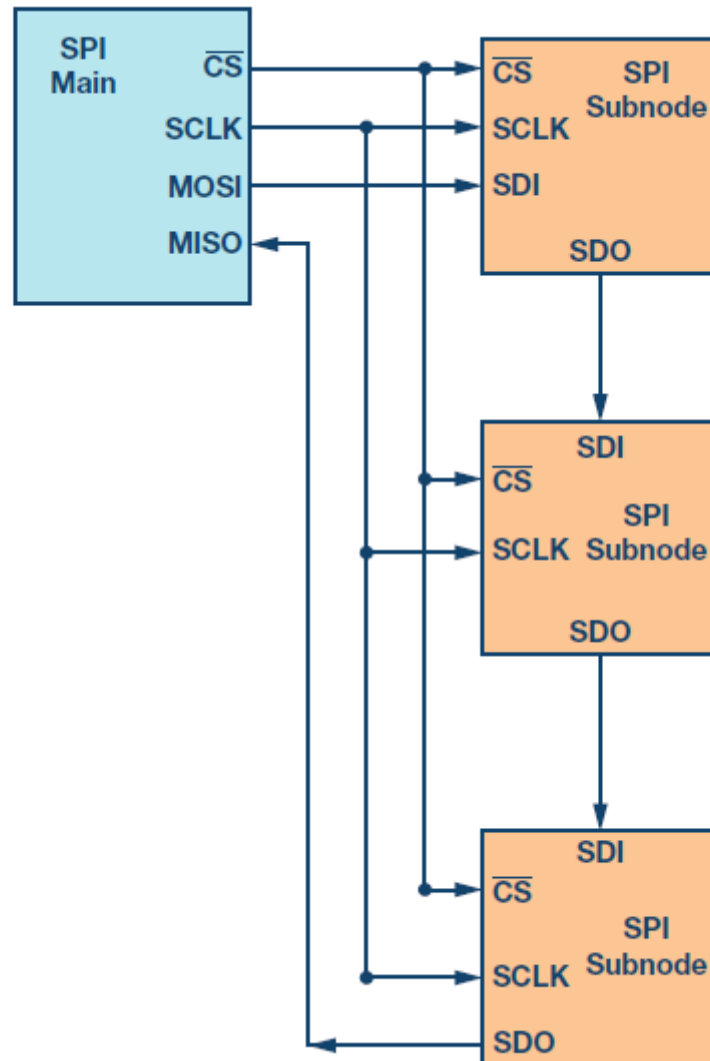
U Regular modu je zapotřebí, aby každé Slave zařízení mělo individuální vodič vedoucí od Main zařízení. Vysílat by měl v jeden moment pouze jenom jeden \overline{CS} vodič. Pokud tomu tak není a je najednou spuštěno vícero \overline{CS} vodičů, dojde ke zkažení dat na vodiči MISO a nebude již možné zjistit, kdo ze Slave zařízení odesílal data. S rostoucím počtem Slave zařízení roste i počet vodičů \overline{CS} . Velký počet \overline{CS} vodičů zapříčiní zvětšený počet vstupů a výstupů potřebných z Main zařízení, a tím se omezí počet Slave zařízení, která můžeme použít. Tomuto problému jde předejít tím, že použijeme multiplexor pro generování \overline{CS} signálů.



Obrázek 8. Regular mode pro SPI rozhraní, zdroj [9]

Daisy chain method je řešena paralelně na rozdíl od Regular modu, který je řešen sériově. V této metodě jsou všechny \overline{CS} signály sloučeny do jednoho a data si předávají z jednoho Slave zařízení do druhého. Veškeré Slave zařízení dostanou SPI clock signál ve stejný čas. Data z Mainu jsou přímo připojena na první Slave a první Slave je přeposílá dále po zařízeních. Pro úspěšný přenos dat z jednoho Slave zařízení do druhého Slave zařízení nám hraje klíčovou roli počet hodinových impulsů. Počet hodinových impulsů potřebných k přenosu dat je přímo

úměrný pozici Slave zařízení v řetězci. Pokud je zařízení druhé v řetězci je potřeba dvou hodinových impulsů pro přenos dat. Ovšem pozor, metoda Daisy chain není podporována u všech SPI zařízení.

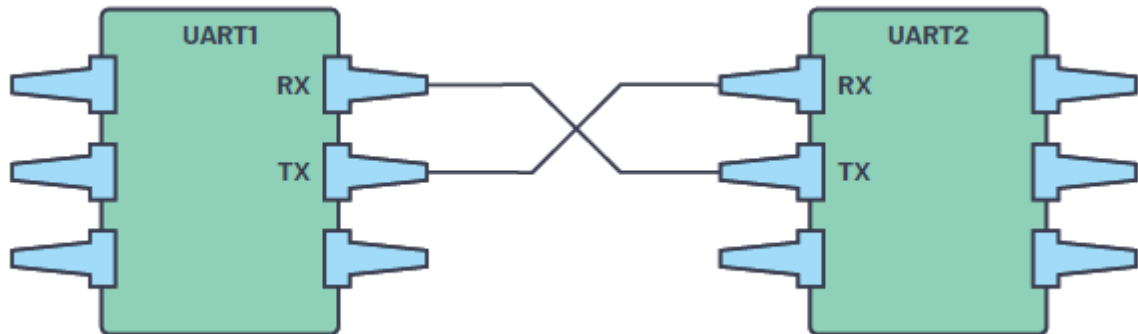


Obrázek 9. Daisy chain methode pro SPI rozhraní, zdroj [9]

UART rozhraní (UART interface)

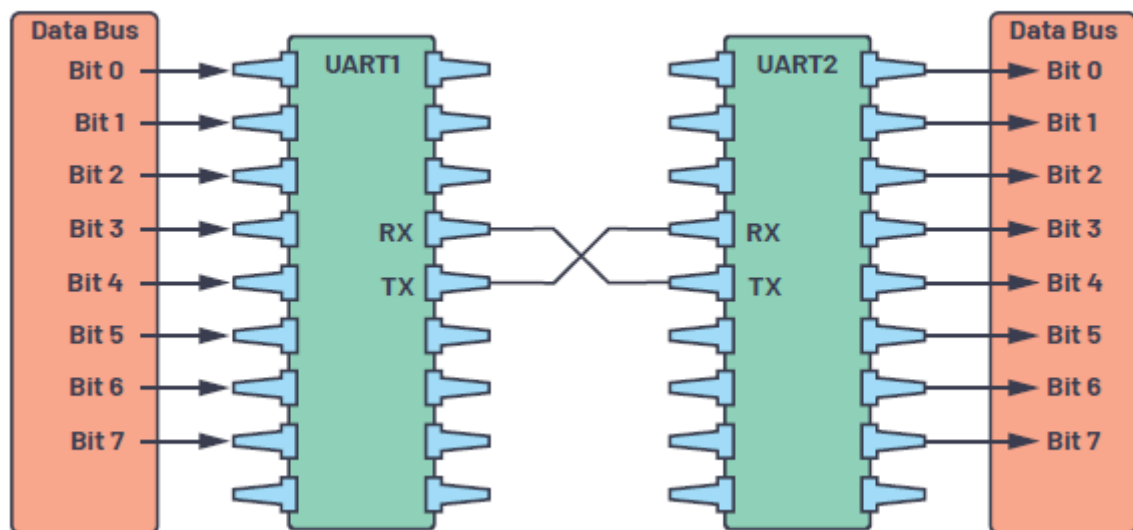
Embedded systémy, mikroprocesory a počítače většinou používají UART jako device-to-device komunikační protokol. Mezi všemi dostupnými protokoly, jež máme, UART využívá jenom dva vodiče pro vysílání a přijímání dat. Navzdory tomu, jak je rozšířený, není zcela optimalizovaný. UART, celým názvem Universal Asynchronous Receiver-Transmitter, je asynchronní sériové rozhraní, což znamená, že nepoužívá vnitřní hodiny pro synchronizaci, ale použije pro synchronizaci rychlost přenosu baud. Klasickým případem přenosové rychlosti baud rate je 9 600, tato rychlost je typická pro desky Arduino a příkladem druhé nejpoužívanější

rychlosti je 115 200 baud rate, která se nastavuje pro čipy od společnosti Esprif. Baud rate se musí na obou zařízeních nastavit stejně, jinak by přenos nebylo možno provést ve správném pořadí. U UART rozhraní je vždy pouze jeden Master a jeden Slave.



Obrázek 10. UART komunikační rozhraní, zdroj [8]

Přenos dat bude vysvětlen na případě, kdy ke každému UART rozhraní je připojena datová sběrnice. Vzhledem k tomu, že je to asynchronní sériová komunikace, tak se data budou odesílat v paketech. Datové sběrnice jsou připojeny paralelně na UART rozhraní, odkud budou posílána sériově bit po bitu na druhé UART zařízení, kde je zase paralelně připojena datová sběrnice – viz obrázek 11.



Obrázek 11. UART rozhraní s připojenými datovými sběrnicemi, zdroj [8]

První krok přenosu u UART rozhraní spočívá v tom, že UART přijme data z paralelně připojené datové sběrnice. **Druhým krokem** je přidání Start bitu, Parity bitu a Stop bitu k datovému framu ze sběrnice. Tímto nám vznikne celý data packet, který je připraven k odeslání na Tx. **Třetím krokem** je odesílání celého packetu z vysílajícího UART rozhraní do přijímacího UART rozhraní, tedy na Rx druhého UART rozhraní. Na přijímacím UART rozhraní jsou data vzorkována podle předem konfigurované rychlosti baud rate. **Čtvrtým krokem** je odstranění Start bitu, Parity bitu a Stop bitu z celého datového packetu a připraví

data k odeslání na datovou sběrnici. **Pátým krokem** je převod přijatého framu ze sériových dat zpět na paralelní a jejich přenos na datovou sběrnici.

Frame protokol je jednou z klíčových vlastností pro UART rozhraní, jenže ještě není plně využívána. Hlavní využití pro tento protokol je zvýšení bezpečnosti a ochrany každého zařízení. Vysvětlit se to hodí na případu, když mají dvě různá zařízení stejné frame protokoly, může vzniknout riziko, že po připojení na stejný UART rozhraní bez předchozí kontroly a konfigurace zařízení se může připojit na jiná piny a způsobí nechtěné problémy na zařízení nebo v programu. Právě aby se tomuto problému vyhnulo, každý frame protokol je vytvořen tak, aby byl jedinečný, jinými slovy vytvoří se vždy unikátní frame protokol pro jedno zařízení. [8]

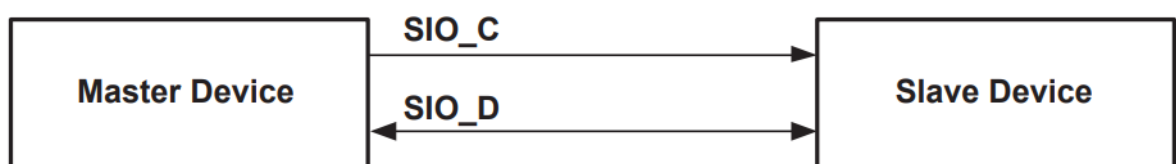
SCCB rozhraní (SCCB interface)

Je specificky navržené sériové synchronní rozhraní pro komunikaci výrobků od společnosti OmniVision. Interface se dělá v provedeních 2-wire anebo 3-wire. Rozdíl v těchto dvou provedení je následující. V provedení 2-wire může být na sběrnici pouze jedno Master zařízení a pouze jeden Slave obvod. Je-li ovšem zvoleno 3-wire interface provedení, dostaneme možnost mít vícero Slave zařízení na jedno Master zařízení. Vodiče pro tento interface jsou SIO_D, SIO_C a SCCB_E.

SIO_D je obousměrná vodičová linka pro datový přenos mezi Master a Slave zařízením. Nachází-li se sběrnice bez přenosu dat, zůstává v plovoucím stavu nebo trojstavovém stavu. Za udržování signálu zodpovídá Master a Slave, oba mají za úkol zabránit šíření neznámého stavu sběrnice.

SIO_C je jednosměrná komunikační linka, která je aktivní v High stavu. Zastupuje hodinový signál. Signál je vyslán z Master obvodu, kde funguje pouze jako output, a na Slave zařízeních jako input. Pokud je sběrnice nečinná, nastaví se na hodnotu log. 1 a přenos dat začne, když se SIO_C nastaví na log.0, jakmile je opět na SIO_C nastavená log.1, je nám oznámeno, že byl přenesen 1 bit.

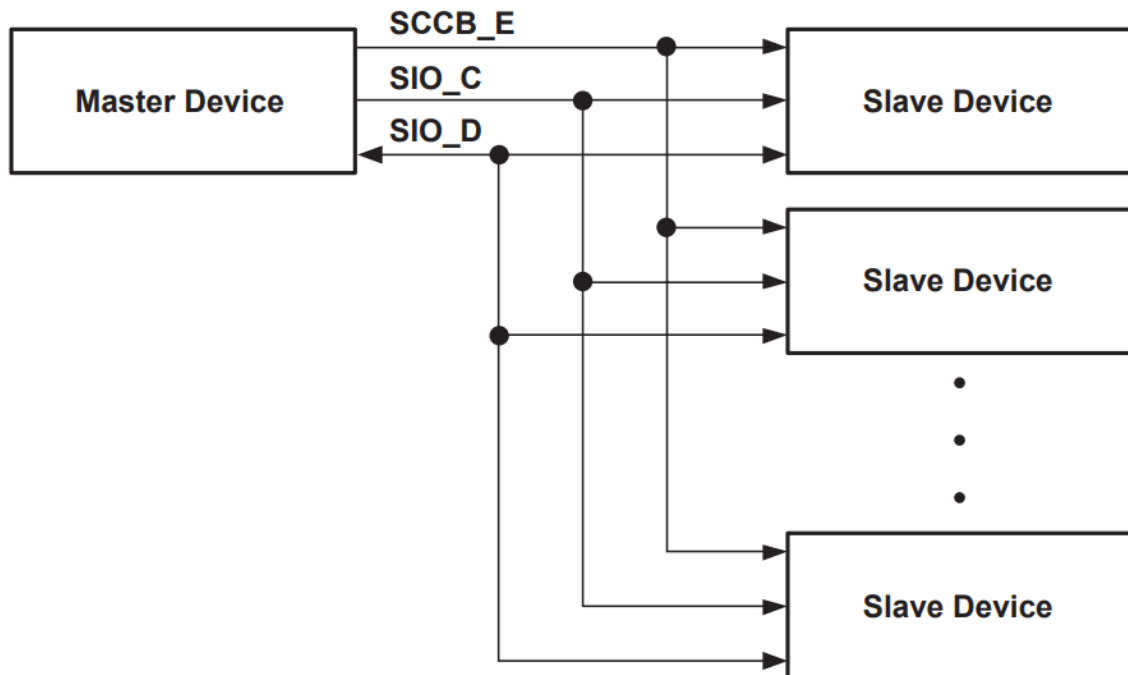
SCCB_E je také jednosměrná komunikační linka, aktivní v Low stavu. Signál musí vycházet z Master zařízení. Slouží k vybírání Slave zařízení. Signál nám indukuje Start a Stop datového přenosu. Přejít z High stavu do Low stavu značí start přenosu dat a obrácený proces. Tedy z Low stavu na High stav značí konec přenosu. Pokud není na Camerachipu SCCB_E obsažen, nastaví se tento signál ve výchozím nastavení na High stav a je tak držen.



Obrázek 12. 2-wire SCCB interface provedení, zdroj [10]

Jak již bylo řečeno 2-wire provedení dovoluje mít pouze jeden Master obvod a jeden Slave obvod. Dvou vodičové provedení je implementováno v produktech od Camerachipu, což je výrobek OmniVision, kde je vyžadovaný snížený počet pinů, a kvůli tomu není dostupný vodič se signálem SCCB_E. Master obvod musí zvládat třístavový režim a být schopen ho udržet na sběrnici dat. Pokud to nezvládne, musíme řídit linku tak, aby byl zaznamenán přestup mezi High a Low stavem, a tím se potvrdila komunikace mezi Master a Slave zařízením.

3 - Wire interface obsahuje SCCB_E vodič pro vybírání určitých Slave obvodů.



Obrázek 13. 3-wire SCCB interface provedení, zdroj [10]

3 – Wire interface je vhodný pro použití v případě, že máme pouze jeden Master obvod a máme k němu připojeno několik Slave zařízení. SCCB_E tu slouží pro výběr jednotlivých Slave obvodů.

Přenos dat po SCCB interface se stejně jako u ostatních interface dělí na Read a Write, tedy čtení a zápis. Start datového přenosu začne přechodem z High stavu na Low stav na SCCB_E lince. SIO_D se nastaví do logické 1 pro zabránění nepožadovaného stavu na sběrnici. Zápis nebo write probíhá ve třífázovém cyklu, ale může být i dvoufázový cyklus. Tento třífázový cyklus nám slouží pro zápis jednoho bytu dat. Fáze jsou tři.

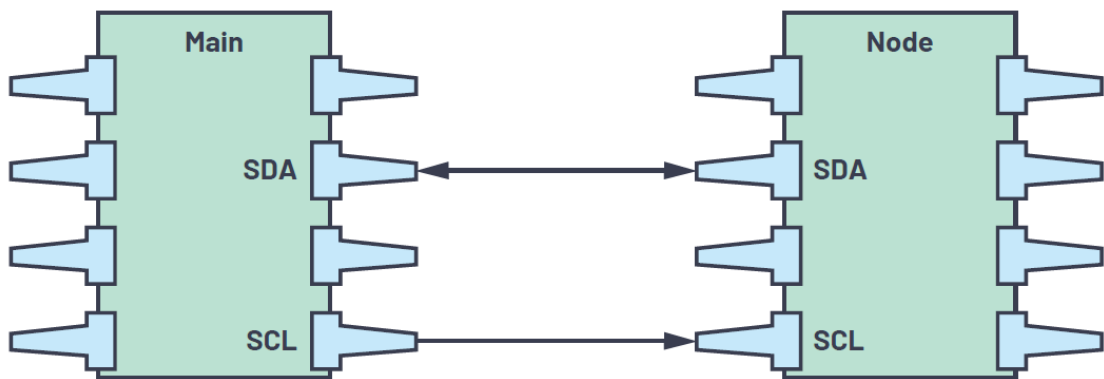
1. fáze: ID. adresa pro vybrání Slave obvodu, na který chce Master zapsat data, je aktivován Master obvodem.
2. fáze: Sub. adresa identifikuje registr na Slave obvodu, do kterého se mají zapsat data.
3. fáze: Write data zapíše 8 bitů, opět tuto fázi dokáže jenom Master zařízení.

Před každým čtením musí proběhnout třífázový nebo dvoufázový cyklus write. Dvoufázový cyklus čtení neumí identifikovat subadresu registru. Slave zařízení mu samo poví, z jakého registru má data odesílat pro Master obvod, aby je mohl přečíst. Dvoufázový cyklus čtení je

schopen potvrdit, jak Master obvod, tak i Slave obvod. Jako poslední je na řadě Don't Care Bit, který signalizuje, že přenos dat je kompletní, a může nastat ukončení přenosu dat. Konec přenosu dat je označen přechodem ze stavu Low na stav High. [10]

I²C rozhraní (I²C interface)

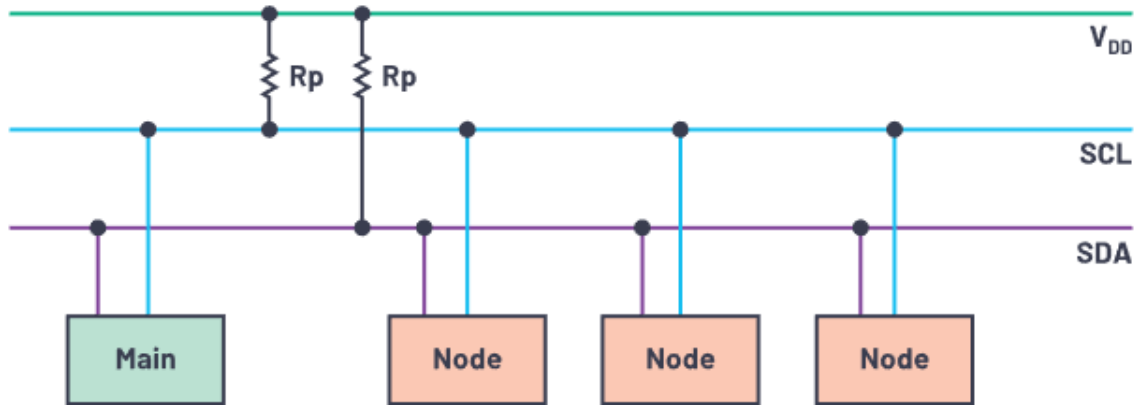
I²C Je běžně používáno jako sériová komunikace mezi zařízeními, která se skládají z různých obvodů. Jedná se o synchronní sériovou komunikaci, která podporuje propojování několika uzlů do jednoho hlavního zařízení a následně také z několika hlavních zařízení do několika uzlů a taky na sériovou komunikační sběrnici. Tato funkce je užitečná, pokud chceme maximalizovat použití jednoho mikrokontroleru, který zaznamenává data na jednu paměťovou kartu nebo na jeden displej. Mimo běžně používaného I²C jsou ještě varianty, které se zaměřují na využití výkonových aplikací. Varianty se nazývají SMBus a PMBus.



Obrázek 14. I²C komunikace mezi hlavním zařízením (Main) a vedlejším zařízením (Node, slave), zdroj [11]

Teoreticky máme možnost mít 1008 Slavu (nodu), to vychází z toho, že můžeme mít maximálně 10 bitů a $2^{10} = 1024$, jenže posledních 16 adres je rezervováno pro speciální případy. Počet Main zařízení je neomezený.

Komunikace probíhá následovně, každé zařízení je spojeno v I²C pomocí dvou vodičů SDA a SCL. Tyto dva vodiče zajišťují veškerou komunikaci na I²C spolu se společnou zemí. SDA je oboustranný vodič sloužící pro přijímání a vysílání dat mezi Main zařízením a Slave zařízením. SCL je jednosměrný vodič vycházející z Main zařízení pro generování hodinového impulsu. Nejdůležitější částí z hardwarových součástí jsou pull-up rezistory, jež se používají na vodičích SDA a SCL. Nejčastěji používaná hodnota pro pull-up rezistory je 4700 Ω . Tato hodnota je typická pro napětí 5 V na vodiči V_{DD} , v jiných textech či datasheetech nazývaném V_{CC} . Pull-up rezistory jsou používány, protože I²C zařízení se připojují ke sběrnici s otevřeným kolektorem nebo s drain piny v otevřeném stavu. Je-li linka bez přenosu, dat její stav je pasivně High. Ovšem pokud se začnou přenášet data, pull-up rezistory stáhnou linku do Low stavu a může se zahájit přenos dat.



Obrázek 15. Pull-up rezistory a jejich zapojení, zdroj [11]

Data jsou přenášena ve zprávách a dále dělí do datových rámců. Ty obsahují protokoly pro čtení nebo zápis a v nich se nachází adresní rámec s binární adresou uzlu a další datový rámec, který obsahuje přenášená data, podmínky start a stop, opakované start bity, bity pro čtení/zápis a potvrzovací/nepotvrzovací bity mezi jednotlivými datovými rámci.

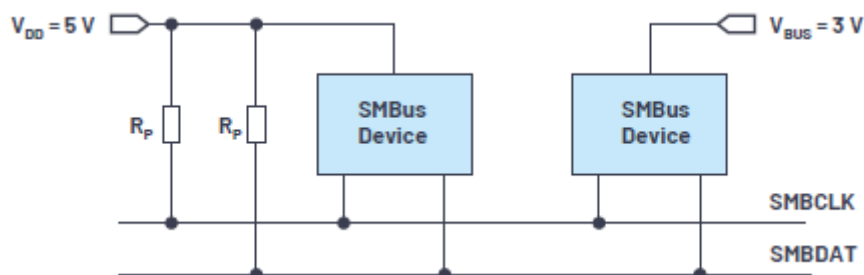


Obrázek 16. Mapa datové zprávy na I²C sběrnici, zdroj [11]

SMBus

SMBus se používá v aplikacích, kde je zapotřebí kritické monitorování parametrů. Běžně se používá na základních deskách počítačů a v embedded systémech. Obsahuje další funkce, jako třeba pro sledování teploty, dohled na napájecí napětí a řízení integrovaných čipů. Má opět pouze jen dva vodiče, a to SMBCLK a SMBDAT. Stejně jako u I²C používá pull-up rezistory a pro přenos dat má protokoly.

Velkou výhodou je to, že k SMBusové sběrnici mohou být připojena zařízení s různým napájecím napětím od 1,8 V, 3,3 V až 5 V.



Obrázek 17. Schéma SMBUS topologie s pull-up rezistory a rozdíly v napětí, zdroj [11]

PMBus

Má hlavní úkol a tím je dohled nad správou napětí. Také proto je to otevřený standard protokol pro správu napájení. Díky PMBusu máme umožněnu komunikaci mezi zařízeními založenými na digitální a analogové technologii. PMBus sběrnice se používají pro digitální správu napájecích zdrojů a pro správu napájení zařízení. Dále nám poskytuje příkaz pro sledování napájecího napětí, jestli nenastalo přepětí nebo podpětí, což by se dalo označit jako jeden z jeho základních parametrů. Je to taky způsob, jak dostat bity nebo bajty z jednoho zařízení do druhého. [11]

1.2.2 Paralelní rozhraní

Stejně jako sériové rozhraní zprostředkovává komunikaci mezi dvěma digitálními systémy, totéž dělá i paralelní rozhraní jenom jinou technologií. Paralelní rozhraní kóduje bity podle jejich pozicí na vodičích. Příklady paralelních rozhraní jsou LPT, PATA a PCI.

Parallel port LPT interface

Původně bylo toto rozhraní určeno pro připojení tiskárny k počítači, jenže postupem času se začal využívat i pro jiná zařízení. Často se nazývá Parallel Port LPT. V roce 1997 byl standardizován jako IEEE 1284.

Periferie s počítačem mohou být propojeny kabeláží o délce 2 m a 5 m. Parallel port LPT používá paralelní přenos bitů po 17 vodičích, které můžeme rozdělit na 8 datových vodičů a 9 kontrolních komunikačních vodičů. Rychlost datového přenosu může být až 5 Mb/s. Výstupní signály používají TTL logiku HIGH stav pro 3,5 V až 5,5 V a 0 V až 0,4 V pro LOW stav. Na pinech se pohybuje proud v maximálním rozsahu od 4 mA do 20 mA. Konektory se používají ve dvou variantách, samec a samice. První provedení je 25pinový D-sub konektor a druhé provedení je 36pinový Centronics konektor.

Parallel port LPT má pět aktivních módů. SPP, Nibble Mode, Byte Mode, EPP Mode a ECP Mode.

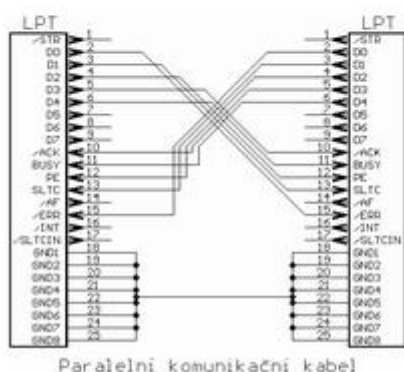
SPP – je to běžný mód pro paralelní datovou komunikaci s přenosovou rychlostí 150 kb/s. Jedná se o jednosměrnou komunikaci,

Nibble Mode – je to způsob, jak získat obousměrnou komunikaci a reverzní kanál mezi periferií a počítačem přímo z SPP bez hardwarových úprav, pouze se provede softwarová úprava, jenže tím se omezí přenosová rychlost na 100 kb/s.

Byte Mode – dosahuje přenosové rychlosti až 200 kb/s a má obousměrnou 8bitovou komunikaci.

EPP Mode – byl vytvořen pro výkonnější připojení přes paralelní port a stále dokázal být kompatibilní LPT. Díky němu se přenosová rychlost zvedla na rozmezí mezi 500 kb/s až 2 Mb/s.

ECP Mode – byl vytvořen jako pokročilý režim pro komunikaci s periferiemi, jako je tiskárna a skener. Rychlost přenosu může být 800 kb/s bez použití DMA, tedy přímého přístupu 3Mb/s nebo 8Mb/s [12]



Obrázek 18. Kabeláž pro komunikace LPT interface, zdroj [12]

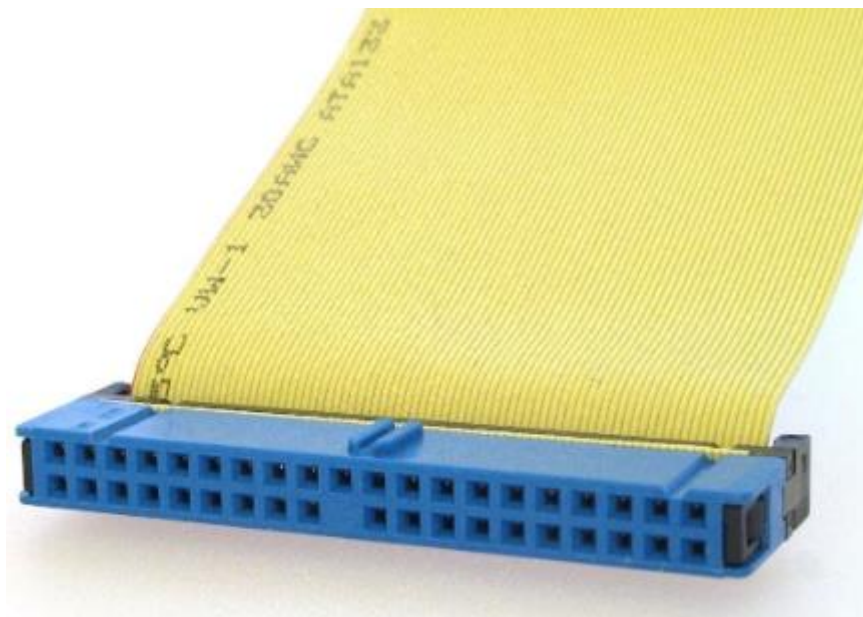
PATA interface

Je rozhraní pro připojování harddisků k počítači. PATA je zkratka pro Parallel Advanced Technology Attachment, jinde se to také píše jako Parallel ATA. PATA rozhraní se začalo používat v roce 1980. Od roku 2003 je přístupná technologie SATA, což je sériová verze ATA.

Původně PATA měla 40žilové páskové kabely, které později byly nahrazeny 80žilovými kabely, u nichž je každý druhý kabel uzemněn, což zmenšuje kapacitu mezi kabely. Maximální délka této kabeláže byla 46 cm, takže se hodily jenom pro zapojení s malou vzdáleností. Přenos dat PATA používá 5 V signalizaci na kabelech.

V průběhu let vzniklo několik variant PATA rozhraní, které se dlouhou dobu označovaly ATA, dokud nepřišlo SATA rozhraní. ATA 1, která byla jako první, používala Master/Slave konfiguraci, buď byla varianta se 40pinovými konektory a kabely, anebo se 44pinovými konektory a kabely. Varianta se čtyřmi piny navíc měla lepší napájení. Přenosová rychlost ATA 1 se pohybovala okolo 4 MB až 16 MB/s. ATA 1 poskytovala časování signálu pro přímý přístup do paměti a programované IN/OUT funkce, ve zkratce jednotka posílala informace přímo do paměti a PIO usnadňoval procesoru počítače řízení informačního přenosu. ATA 2 zvedla přenosovou rychlost dat na stálých 16 MB/s. Větší skok umožnila ATA 4, která zvětšila rychlost přenosu na 33 Mb/s a díky ATA packet interface (ATAPI) dokázala připojovat různé zařízení jako CD-ROM. ATA 4 podporovala 80pinové konektory nebo 40pinové konektory.

ATA 6 /ATAPI podporovala přenosovou rychlost přes 100 Mb/s a přišly HDD s kapacitou přes 10 Gb/s. [13]



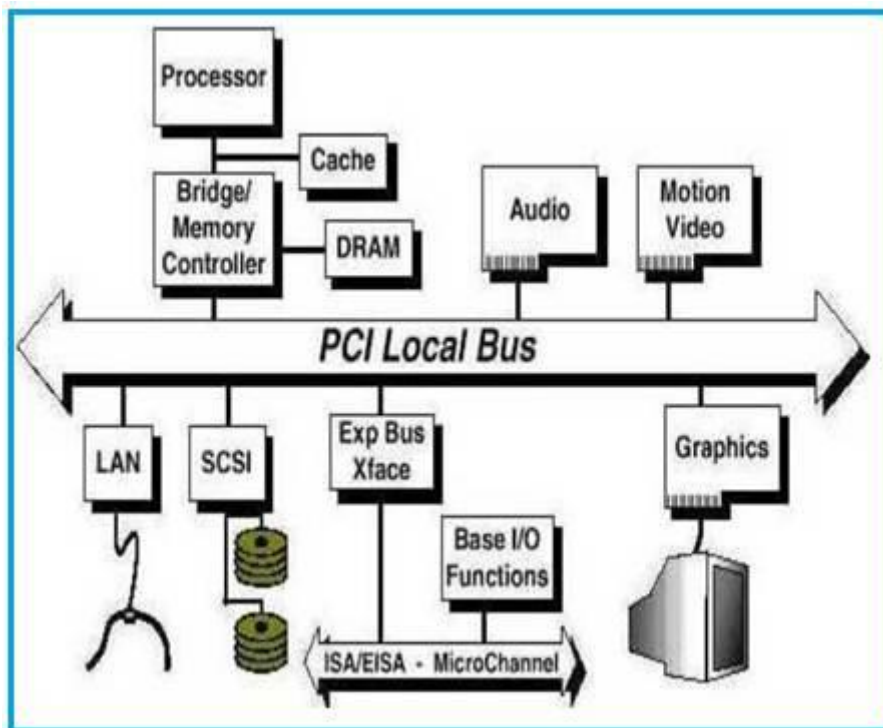
Obrázek 19. Kabel pro spojení periférií za pomoci PATA interface, zdroj [13]

PCI interface

Celým jménem Peripheral Component Interconnect, proto zkratka PCI. Největší popularitě se těšil mezi lety 1995 až 2005 pro připojování zvukových karet, síťových karet a video karet. Jednotkou PCI je PCI bus, což je název pro spoj mezi počítačem a komponentem. Lidé si často pletou PCI s PCI compliance, která se zabývá platebními kartami. PCI se postupně nahradilo novějším PCI Express.

Díky PCI bus můžeme měnit různé periferie, které chceme připojit k počítači. Počítače mohou mít jeden nebo více PCI busů. Do PCI busů se připojují PCI karty, které podporují 5 V a 3,3 V a mají 47 pinů. Datová šířka přenosu může být 32 bitů nebo 64 bitů, také může mít dvě taktovací frekvence 33Mhz anebo 66Mhz.

Komunikace procesoru a periferie připojené k PCI sběrnici je podrobně popsán v normě PCI busu. Dodržení všech podmínek standardu je velmi komplikované, proto se přidává na připojované periferie obvod s čipem S5933 od společnosti AMCC, jenž zajistí veškerou komunikaci pro přenos dat. Mezi možnosti přenosu dat patří jednosměrný přenos jednotlivých bytů přes specializované registry, přímé čtení nebo zápis do paměti připojené na desku a poslední možností je blokový přenos dat pře dvě jednosměrné fronty typu FIFO. [14] [15] [16]



Obrázek 20. PCI Bus a připojované periferie, zdroj [16]

1.3 Kamerové komunikační rozhraní

Camera link interface

Kamerové komunikační rozhraní Camera Link Interface vzniklo na základě toho, že na trhu byla spousta různých zařízení s množstvím různými konektory, proto se rozhodla firma National Semiconductor rozhodla vytvořit tento komunikační standard pro kamerové dohledové systémy. Dále v textu bude Camera Link Interface uváděna zkratkou CLI.

Technologie CLI vychází z technologie Channel Link, i s touto technologií přišla společnost National Semiconductor, jenže její původní využití bylo pro Flat displeje. Channel Link byl rozšířen metodou pro univerzální přenos dat. Channel link byl založen na technologii LVDS pro fyzickou vrstvu. LVDS je nízkonapěťové vysokorychlostní standardní rozhraní pro všeobecné účely. Channel Link má pár – ovladač a přijímač. Ovladač akceptuje 28 jednostranných signálů a jednostranné hodiny. Přijímač přijímá čtyři datové proudy LVDS spolu s hodinami LVDS a následně řídí 28 bitů a hodiny na desce. Technologie Channel Link nám dává výhodu v podobě menších konektorů a kabeláže, protože pro přenos dat potřebuje pět párů kabelů, což nám zapříčiní přenos až 28 bitu dat. Druhou výhodou je vysoká rychlost přenosu dat, která může dosahovat až 2 Gb/s. Konektor pro technologii CLI je MDR26, byl vybrán pro svou robustnost a hlavně vysokofrekvenční přenosovou rychlost pro Channel Link. Problém s CLI je ten, že používá 28bitové Channel Link čipy od National Semiconductor, které nejsou kompatibilní s jinými výrobci.

Kabeláž CLI poskytuje Camera Controls Signals (kamerové kontrolní signály), Video data (video data) a sériovou komunikaci.

Camera Controls Signals mají rezervovány čtyři páry LVDS pro všeobecné použití. Výrobci si můžou specifikovat tyto signály pro potřeby výrobku. Signály jsou: CC1, CC2, CC3 A CC4. CC1 je zkratka pro Camera Control 1.

Video data pro přenos dat videa potřebuje Channel Link a jeho sběrnici, na kterou se přenáší obrazová data a obrazová povolení. Obrazová povolení slouží k povolení přenosu obrazu na Channel Link čip. Obrazová povolení jsou čtyři a to: FVAL jako frame valid, LVAL jako line valid, DVAL data valid, SPARE je připraven pro budoucí využití. Všechna čtyři povolení musí být přístupná pro jeden Channel Link čip nebo pro všechny čipy Channel Linku, které používá kamera.

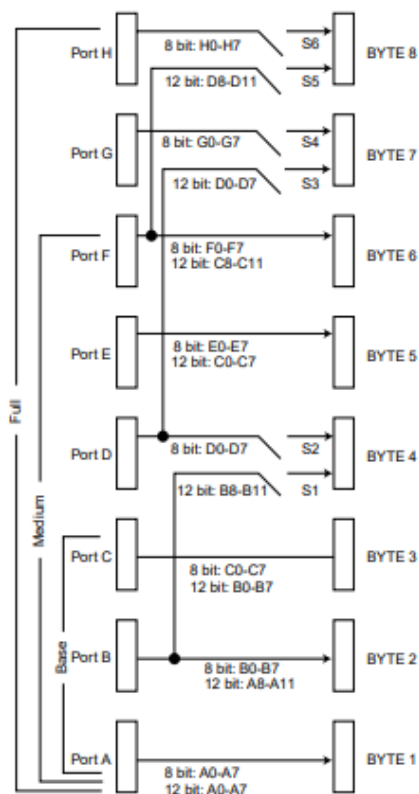
Sériová komunikace se provádí po dvou párech LVDS, na kterých běží asynchronní přenos dat z kamery a do frame grabberu a obráceně. Frame grabber a kamera by měly oba dva zvládnout alespoň 9600 baud. Frame grabber je nahrávací zařízení pro snímky nebo videa z kamery. Páry pro komunikaci máme SerTFG - pár pro sériovou komunikaci s frame grabberem a SerTC -pár pro sériovou komunikaci s kamerou.

CLI má tři možné konfigurace: BASE, MEDIUM a FULL. Tři různé verze konfigurace existují proto, že jeden čip od Channel Linku je limitován na 28 bitů, jenže některé kamery potřebují více čipů pro kvalitní přenos dat. CLI používá tzv. porty A–H, kdy jeden port je 8bitové slovo. Pomocí těchto portů řešíme již zmíněné konfigurace.

Base má 1 Channel Link čip a 1 kabelový konektor

Medium má 2 Channel Link čipy a 2 kabelové konektory

Full má 3 Channel Link čipy a 2 kabelové konektory [17]



Obrázek 21. Přehled portů pro CLI, zdroj [17]

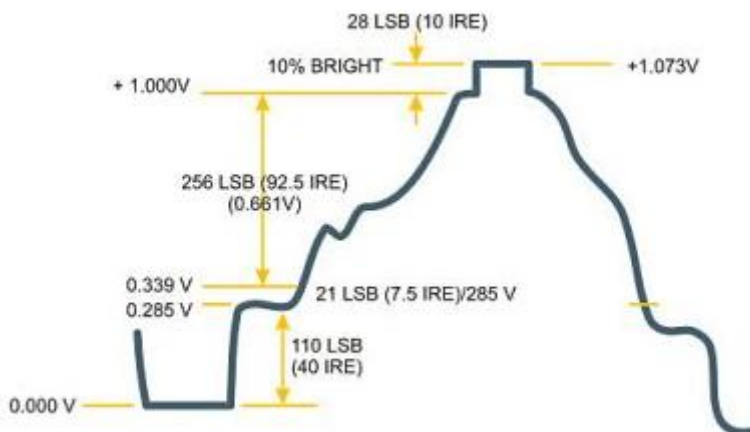
Kompozitní video (CVBS)

Převážně našlo využití pro přenos analogového obrazového datového signálu u analogových kamer. Druhou věcí je to, že kompozitní video je nejběžnějším rozhraním pro přenos dat analogového video signálu. Zkratka pro kompozitní video je CVBS, tedy Composite Video Blanking and Synchronization.

V jediném kabelu je kombinace informace pro vysílání barevného videa a synchronizační impulsy snímků a řádků. Video data se skládají ze signálu Y a C. Signál Y je určen světlostí a černobílými hodnotami, Signál C pro chrominanci nebo barvy. Barva video signálu je lineární kombinací jasů obrazu a modulární subnosné chrominance přenášející informaci o barvě a kombinaci sytosti a odstínu. Kombinace chrominance a jasů je složitější systém frekvenčního dělení. Signál CVBS obsahuje signál ColourBurst, který se označuje jako barevný synchronizační signál, jímž na začátku každého snímáčiho řádku dokáže přijímač obnovit zeslabené nosné chrominanci signály a následně dekodovat barevnou informaci. Kvalita signálu Y je snížena ve slabě osvětlených místnostech. Zhoršený signál C dokáže mít za následek zbarvení, které se liší od svého původního zdroje.

K přenosu dat používáme dva druhy konektorů: BNC a RCA. U RCA konektorů jsou tři barvy na tři jacky, jelikož CVBS přenáší jenom video data a my potřebujeme zajistit ještě přenos pro zvuk. Tři barvy, které používáme u RCA konektorů jsou, žlutá pro video data a červená s bílou pro audio data. CVBS kamery používají standardy rozlišení Pal (960 x 576) a NTSC (960 x 480).

Nejnovější verzí kompozitního videa je HDMI, kde není zapotřebí mít několik různých kabelů zvlášť, ale všechny kabely jsou spolu spojeny do jednoho jednotného kabelu, ve kterém jsou všechny kabely jak na zvuk, tak i na video. [18] [19] [20]



Obrázek 22. Vzorek signálu kompozitního videa, zdroj [20]

DVI interface

Jedná se o rozhraní pro přenos nekomprimovaného videa, analogového nebo digitálního signálu. Jsou tři množství: DVI, a to DVI – A pro analogový signál, DVI – D pro digitální signál a DVI-I pro oba druhy signálů. DVI vzniklo jako průmyslový standard pro přenos videa do sledovací jednotky.

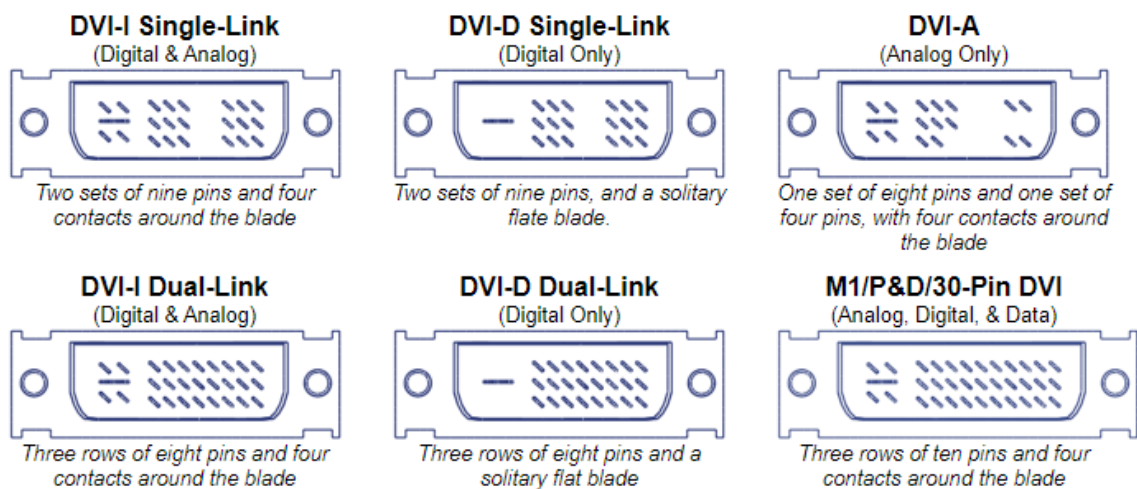
DVI-D je přímé digitální propojení mezi zdrojem videa a sledovací jednotkou. Poskytuje rychlejší přenos a větší kvalitu obrazu než analogový DVI. Výstupem je analogový VGA signál. Má Single-Link a Dual-Link. Rozlišení je 2560 x 1600 s 60hz nebo 1920 x 1080 s 144hz.

DVI-A se používal k připojení k CRT monitorům, ale hlavně se využívá k připojení k VGA rozlišením.

DVI-I dokáže přenést oba signály, jak digitál do digitálu, tak i analog do analogu. Takže má mnohem větší využití. Ovšem pro připojení jednoho konce, který je analogový, a druhého konce, který je digitální, bude potřeba použít VGA to DVI převodník nebo obráceně. Rozlišení pro Single-Link je 1920 x 1200 s 60hz a pro Dual-Link je 2560 x 1600.

U DVI je digitální formát ve dvou typech: Single-Link a Dual-Link. DVI kabely přenášejí digitální data pomocí formátu TMDS (Transition Minimized Differential Signaling). Single-Link využívá jeden vysílač 165 Mhz a Dual-Link využívá dva vysílače. Dual-link piny zdvojnásobují přenosový výkon, a tím se zajistí zvýšená kvalita a rychlost signálu. Single-Link zvládá rozlišení 1920 x 1200 při 60hz a Dual Link zvládne 2560 x 1600.

DVI přenáší obrazy v digitální formě takže odpadá nutnost konvertovat obraz z analogu do digitálu, jako bylo potřeba u VGA. Obraz je přímo přenesen do koncového zařízení bez ztráty obrazu. Digitální data se přenášejí díky standardu TMDS, který konvertuje tři barevné kanály R, G a B do jednoho sériového signálu, jenž má v sobě všechny tři kanály. [21] [22] [23]



Obrázek 23. Konektory pro DVI kabely, zdroj [21]

CPI rozhraní (Camera Parallel interface)

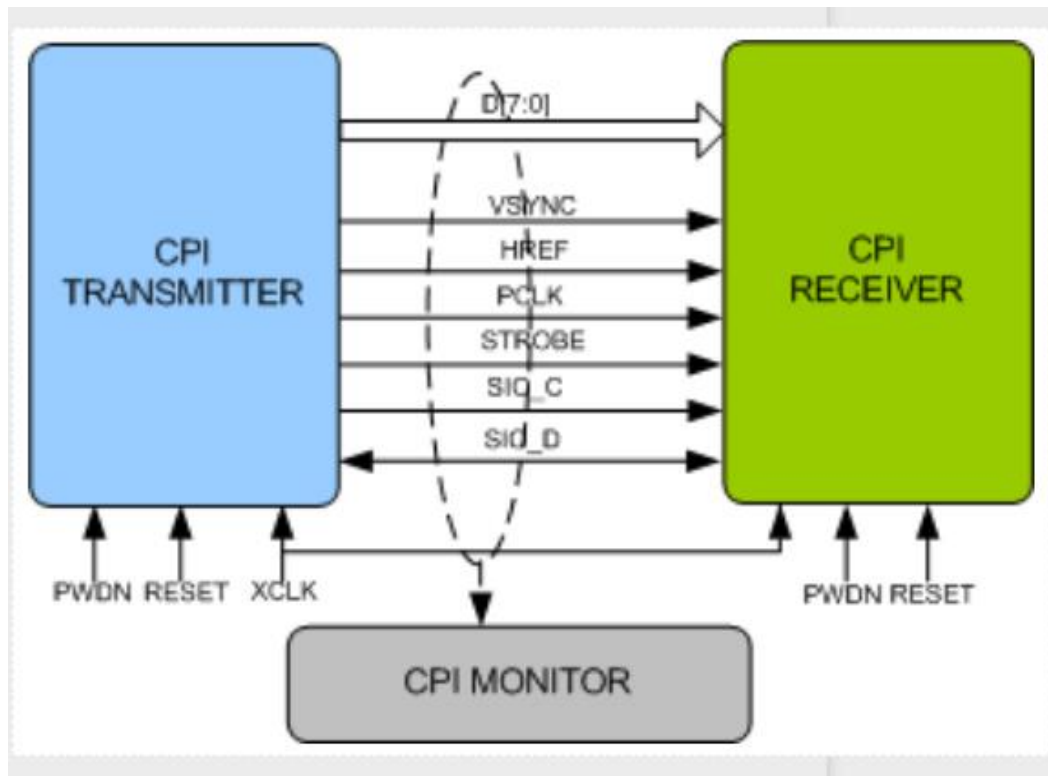
CPI je jedno z nejstarších původních rozhraní pro rozhraní obrazových snímačů specifikovaných organizací MIPI Alliance. Spolu s I^2C je CPI základním stavebním kamenem pro MIPI kamery. CPI může mít 8bitové nebo 10bitové paralelní přenosové linky pro připojení k datové sběrnici.

V základu se CPI skládá z 8bitové paralelní linky, která připojuje k datové sběrnici videa ze snímače obrazu. Snímač zajišťuje VSYNC (vertikální synchronizaci), HREF (horizontální referenci) a časování hodin pixelu (PCLK). Všechny paralelní signály jsou vysílány modulem kamery a přijímány řídicí jednotkou.

VSYNC je funkce, která synchronizuje obnovovací frekvenci snímače s jiným zařízením. To zaručí zachycení obrazu v určitém okamžiku s velkou přesností.

HREF je funkce, která oznamuje, že dokončil snímání jednoho řádku a jde na další řádek.

PCLK je funkce, která řídí rychlost dat z obrazového snímače, a PCLK určuje, jak rychle se data vykreslují z obrazového snímače do monitoru. Obrazová data jsou vypouštěna na každé vzestupné hrany PCLK. [24] [25]



Obrázek 24. CPI interface komunikace mezi přijímačem a vysílačem, zdroj [24]

1.4 Výběr kamerového modulu

Kamerový modul by se měl vybrat podle toho, aby splňoval naše požadavky. Ty se mohou různě lišit. Záleží jenom na našich preferencích, jako pixely, rozlišení, FPS, technologie pro obrazový snímač a cena, která funguje vždy jakonejvětší rozhodovací činitel.

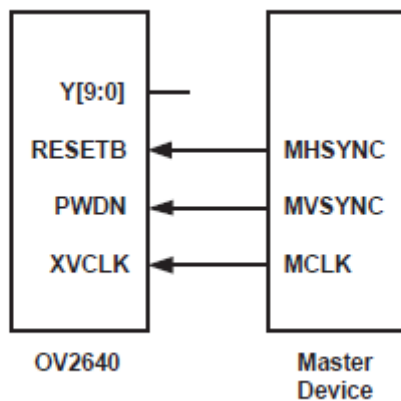
Kamerový modul OV2640

Jde o malý kamerový modul od společnosti OmniVision, který se těší velké oblibě díky své nízké ceně a dostupnosti. Velikostí je menší než průměrný lidský palec. Kamerový modul je založen na technologii CMOS. Své využití tato malá kamera našla především v telefonech okolo roku 2006, následně se uplatnil v hračkách a jako poslední uplatnění našel v digitálních statických fotoaparátech.

Kamerový modul zvládá rozlišení 1600 x 1200, což po přepočtu jsou 2 megapixely. Pokud jde o rychlost přenosu obrazu, tedy o FPS, OV2640 nabízí celkem tři možnosti. První je 15 FPS při rozlišení UXGA, tedy 1600 x 1200, druhá volba je 30fps při rozlišení SVGA 800 x 600 a poslední možností je pro 60FPS, a to je rozlišení CIF 400 x 300. Samozřejmě musíme brát

v potaz to, že se zvyšujícím počtem FPS se klesá kvalita obrazu. Modul je napájen stejnosměrným napětím od 2 V do 3 V. Zajímavým faktem je to, že by měl být schopen pracovat v teplotním rozsahu od -30 °C až do 70 °C, jenže stabilní obraz poskytuje jenom v rozsahu teplot od 0 °C do 50 °C. Ve standby modu je odběr proudu okolo 600 μA.

Kamerový modul je ovládán pomocí SCCB interface, jenž byl popsán již dříve. OV2640 má 10bitový digital video port. V továrním nastavení je modul připraven jako Master zařízení. Ovšem pro použití jako Slave, které je často vyžadováno, chceme-li připojit kamerový modul k mikrokontroleru, je zapotřebí zajistit uvedení registrů COM7[3] (0X12), CLKRC [6] (0x11), COM2[2] do logické 1. To zajistí nastavení do Slave režimu. Proto aby mohli Master a Slave spolu komunikovat, musí poskytnout Master signály pro hodiny, vertikální a horizontální synchronizaci. OV2640 použije piny PWDN a RESETB jako vertikální a horizontální synchronizaci, k tomu na pin XVCLK přivede Master zdroj hodin pro řízení Slave obvodu. Za pomoci těchto tří signálů řídí Master obvod Slave obvod. [26]



Obrázek 25. Zapojení s OV2640 jako Slave obvodem, zdroj [26]



Obrázek 26. Kamerový modul OV2640, zdroj [27]

Kamerový modul OV7670

I druhý kamerový modul je od rodiny OmniVision. Stejně jako u kamerového modulu OV2640 i kamerový modul OV7670 využívá technologii CMOS pro obrazový snímač. Opět našel využití v mobilních telefonech, v hračkách, počítačové kamery a v digitálních fotoaparátech. V

dnešní době ho spousta obchodů prodává na dps desce s vývody na připojení. Cenou se pohybuje pod 100 Kč.

Rozlišení tohoto modulu je VGA 640 x 480. Po přepočtu se dostaneme k tomu, že modul má rozlišení 0,3megapixelů. Rychlost přenosu obrazu je 30 FPS. U tohoto modulu nedokážeme zvětšit počet FPS tím, že by se zmenšilo rozlišení. Modul je napájen od 2,5 V do 3Vstejnsměrným napětím. Teplotní rozmezí pro správný provoz a stabilní obraz je od 0 °C do 50 °C, ovšem měl by fungovat i v rozmezí teplot od -30 °C do 70 °C podle uvedených hodnot v datasheetu. Ve standby modu má odběr proudu menší než 20 μ A.

Vzhledem k tomu, že se jedná o výrobek firmy OmniVision, používá pro komunikaci SCCB interface mezi mikrokontrolerem a kamerovým modulem. Mikrokontroler řídí modul přes SCCB a data jsou posílána po 8bitové paralelní sběrnici. Modul opět potřebuje externí signály od Master obvodu, tedy mikrokontroleru, a těmi signály jsou VSYNC, HREF/HSYNC a PCLK. [28] [29]



Obrázek 27. Kamerový modul OV7670, zdroj [28]

Kamerový modul PTC08

Tento modul nepochází od společnosti OmniVision, ale své využití našel ve stejných oblastech jako předchozí kamerové moduly. PTC08 je založen na technologii CMOS. Použitý CMOS čip v tomto kamerovém modulu je MT9V011, výrobcem je firma MICRON. Cena na zahraničních webech se pohybuje okolo 700 Kč. Velikost modulu je 32 mm x 32 mm.

Kvalita rozlišení je VGA, tedy 640 x 480 při 30 FPS, jenže samotný CMOS obrazový snímač se dá naprogramovat, aby dosahoval při CIF 60 FPS a při QVGA až 90 FPS. Doporučené napájecí napětí je 5 V, ale minimální napětí jsou 3,3 V. Co se týče teplotního rozmezí, je na to PTC08 mnohem lépe než předešlé moduly. Jeho teplotní rozmezí je -20 °C do 85 °C. Minimální elektrický proud pro funkčnost je 60 mA, maximální je 85 mA, pro optimální fungování je doporučeno 70 mA.

PTC08 jako komunikační rozhraní používá UART s maximální rychlostí 115 200 bps. Výstupem je obrázek ve formátu JPEG. [30] [31]



Obrázek 28. Kamerový modul PTC08, zdroj [31]

Kamerový modul ArduCam-Mini OV5642(Arduino camera)

Společnost ArduCam využila kamerové moduly společnosti OmniVision OV2640 a OV5642. Vzhledem k tomu, že se jedná o rodinu OmniVision, kamera má CMOS snímač obrazu. Cena tohoto modulu je 1134 Kč, sice je ze všech zmíněných modulů nejvyšší, ale její výše je oprávněná. Využití našel v Iot kamerách, kamerách k robotům a fotopastech.

Kamera disponuje 5megapixelovým snímačem OV5642. Rozlišení je 2592 x 1944, tedy 5Megapixelů, při 15 FPS. Samozřejmostí je možnost snížit kvalitu a zvýšit počet FPS. Rozlišení 1080p poskytne 30 FPS, rozlišení VGA a 720p poskytne 60 FPS a QVGA rozlišení nám poskytne 120 FPS. Doporučené napětí je 5 V stejnosměrným napětím. Teplotní rozmezí je podobné -10 °C až do +55 °C. Co se týče proudu, při normálním používání by měl mít modul odběr proudu při napětí 5 V 390 mA a při režimu spánku klesne odběr proudu až na 20 mA při 5 V.

Pro komunikaci s mikroprocesorem používá ArduCam SPI interface, kdy kamera je Slave obvod a mikroprocesor je Master obvod. Výstupem z modulu je obraz ve formátu JPEG. Ovšem jde s ním komunikovat i s I^2C interface.

Kamerový modul nám poskytuje dva režimy. V prvním režimu se jedná o možnost o Single capture mode, kde jde o zachycení jednoho snímku, jež přidá je do vyrovnávací paměti. Druhý režim je režim vícenásobného snímání, kde uživatel si zvolí počet snímků, které budou zachyceny. [32] [33] [34]



Obrázek 29. Kamerový modul Arducam – mini OV5642, zdroj [34]

Kamerový modul JPEG COLOR CAMERA C328

Kamerový modul je se svojí velikostí 20 mm x 28 mm nejmenší ze všech zmíněných modulů. Používá obrazový snímač OV7640 od OmniVision společnosti. Bohužel se ale již tento kamerový modul neprodává.

Rozlišení je VGA 640 x 480. Přenos obrazu zvládá ve dvou stavech, VGA při 30 FPS a QVGA při 60 FPS. Modul má provozní doporučené napětí 3,3 V a odběr proudu 60 mA. Napětí i proud jsou stejnosměrné. Teplotní rozsah pro tento modul je 0 °C do 25 °C.

Kamerový modul disponuje i možností tzv. sleepmodu, kdy se zařízení tzv. uspí pro menší spotřebu energie. Dále je vybaven EEPROM pamětí pro poskytnutí příkazového rozhraní uživateli. Paměť uživateli poskytuje sadu uživatelských příkazů pro navázání propojení mezi uživatelem a kamerou.

Komunikace s mikroprocesorem je uskutečněna za pomoci UART interface s maximální rychlostí až 115 200 bps. Zařízení je vybaveno funkcí pro automatické detekování rychlosti přenosu dat, tzv. baud rate. Výstupem z kamerového modulu je obrázek ve formátu JPEG přenesen přes sériový port k uživateli. [35]



Obrázek 30. Kamerový modul JPEG COLOR CAMERA C328, zdroj [35]

| Tabulka pro porovnání jednotlivých kamerových modulů | | | | | | |
|--|-------------|-------------------|-----------------|---------------------------|------------|------------------|
| název kamerového modu | rozlišení | fps | napájecí napětí | odběr proudu | interface | cena modulu v Kč |
| OV2640 | 1600 x 1200 | 15 fps až 60 fps | 3 V | standby mode: 600 μ A | SCCB a DVI | 30 Kč |
| OV7670 | 640 x 480 | 30 fps | 3 V | standby mode: 20 μ A | SCCB a DVI | 88 Kč |
| PTC08 | 640 x 480 | 30 fps až 90 fps | 5 V | 60 mA až 85mA | UART | 700 Kč |
| ARDUCAM- MINI | 2592 x 1944 | 15 fps až 120 fps | 5 V | 20 mA až 390 mA | SPI | 1134 Kč |
| JPEG COLOR CAMERA | 640 x 480 | 30 fps až 60 fps | 3,3 V | 60 mA | UART | již se neprodává |

Tabulka 2. Porovnání jednotlivých kamerových modulů

1.5 Výběr hardwaru pro kamerový modul

Existuje spousta mikroprocesorů, které se dají využít pro připojení ke kamerovému modulu. Záleží opět jenom na uživateli, jaké má podmínky, a na ceně. Především by se mělo hledět na to, zda je vůbec možné spustit komunikace mezi deskou mikroprocesoru a kamerovým modulem, dále na jaké vlastnosti a funkce by měla deska mít a v poslední řadě by se měla zvažovat cena za danou desku.

Arduino

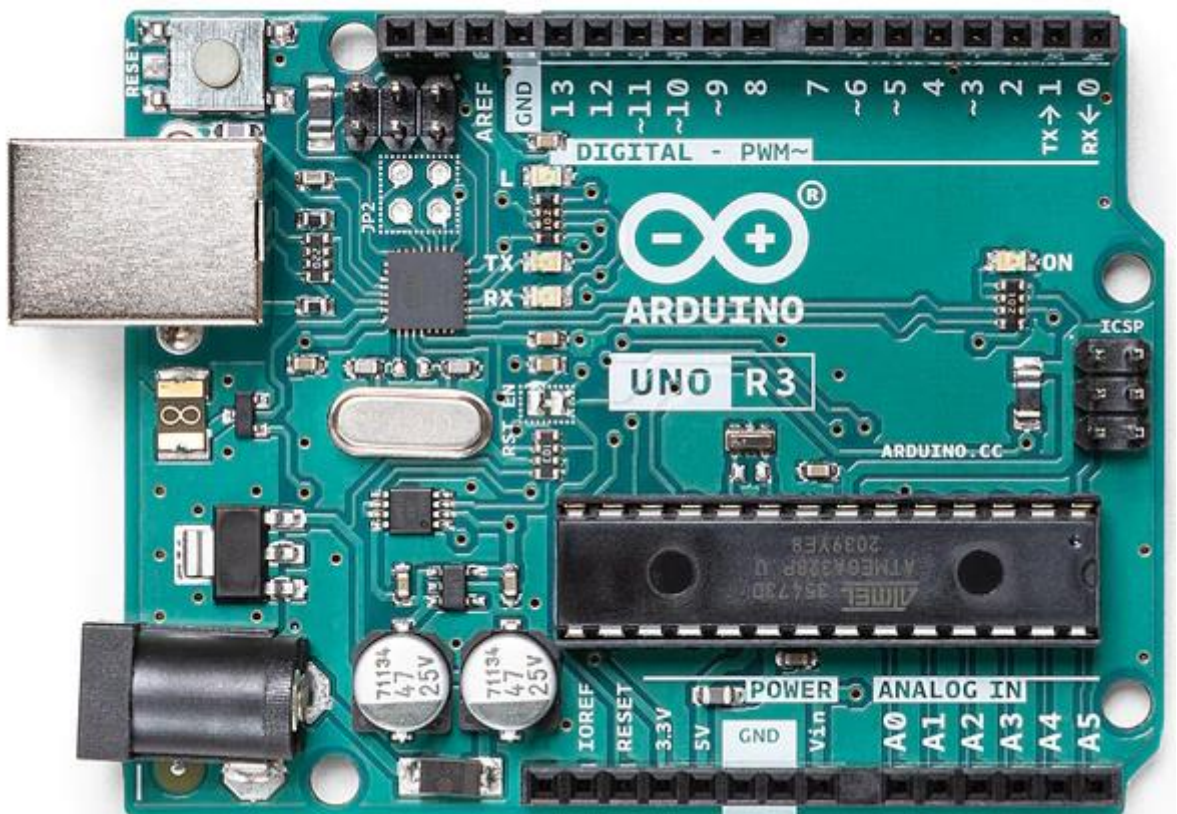
Velmi populární desky Arduino a jejich čínské klony se často používají pro kamerové moduly. Mezi hlavní typy patří Arduino Uno, Arduino Mega a Arduino Nano, všechny tři modely desky jsou založené na čípech od společnosti Atmega 328, Atmega 1280 a Atmega 2560. Ovšem tyto čipy nemají vhodný kamerový port nebo jej nemají vůbec. Deska Arduino Nano je deska menší velikosti, Arduino Uno je deska střední velikosti a Arduino Mega je rozměrově největší ze všech zmíněných desek.

Arduino Nano podporuje komunikaci pro UART, I^2C a SPI. Napájecí napětí je mezi 5 V až 12 V pro operační napětí 5 V, odebíraný proud se pohybuje okolo 19 mA. Cena za originální desku je okolo 500 Kč a cena čínského klonu je 200 Kč. Čip použitý na této desce je Atmega 328.

Arduino Uno podporuje komunikaci pro UART, I^2C a SPI. Napájecí napětí je 5 V až 12 V pro operační napětí 5 V a odebíraný proud je 50 mA. Cena pro originální desku je od 530 až 800 Kč a cena za čínský klon je 360 Kč. Čip použitý na této desce je Atmega 328 p.

Arduino Mega podporuje komunikaci UART, I^2C a SPI. Napájecí napětí je mezi 5 V až 12 V pro operační napětí 5 V a odebíraný proud je 50 mA. Cena za originální desku je od 900 až 1100 Kč a cena za čínský klon je 700 Kč. Čip, který byl použit pro tuto desku, je Atmega 2560

Všechny desky mají EEPROM paměť, SRAM a Flash memory, jenže v některých případech jsou požadavky kamerových modulů mnohem větší, než čím disponuje deska. [36] [37] [38] [39]



Obrázek 31. Vývojový kit Arduino UNO, zdroj [38]

ESP 32 / ESP 8266

Dalšími velmi populárními deskami s mikroprocesory jsou desky od společnosti Espressif. Tato firma přišla s průkopnickým čipem, který obsahoval Wi-Fi modul. To byl první čip s názvem ESP 8266. O pár let později přišel další, lepší čip od společnosti Espressif s názvem ESP 32, čip obsahoval Wi-Fi modul, Bluetooth modul a spoustu dalších možností.

Firma Espressif vyvinula několik variant svých desek, jak v běžné verzi, ale i ve speciálních variantách, např. ESP32 – CAM, která je přímo určena pro kamerové moduly.

ESP 8266 je tedy první ze skupiny ESP čipů. Je to levná deska, jejíž cena se pohybuje okolo 150 Kč na osazeném tištěném spoji. Bohužel této desce již skončila podpora a mnoho prodejců pouze doprodává zásoby, jež měli na skladě. Komunikaci přes GPIO piny podporuje SPI, I^2C

a dále má klasicky UART. Napájení je 5 V a odebíraný proud je od 10 μ A do 80 mA až po maximálních 350 mA.

ESP 32 je druhým čipem od typu ESP. Cena desky za tu kvantitu funkcí je adekvátních 210 Kč. Komunikace opět podporuje I^2C a SPI spolu s UART. Má moduly Wi-Fi 2,5 GHz a Bluetooth 4.2, poté kapacitní senzory, externí probuzení a Hallův senzor, nabízí možnosti pro usnutí do dvou modů Light sleep a Deep sleep. Napájení je 5 V a odebíraný proud je 80 mA.

ESP 32 – CAM je malá destička osazená čipem ESP32 a kamerovým modulem OV2640. Jedinou nevýhodou je to, že deska nemá USB/UART převodník, bez kterého je proces nahrávání značně komplikovaný. Cena za tento malý modulek je 310 Kč. Modul nabízí možnost připojení externí Wi-Fi antény, pokud by nevystačovala ta, která je již součástí čipu ESP32. K tomuto modulu se dá dokoupit shield, na kterém je připojen převodník UART/USB. Komunikace, které podporuje, je stejná jako u ESP 32. Na pájení je 5 V, odběr proudu se liší při použití různých funkcí, ale rozsah je 6 mA do 310 mA. Existuje ještě varianta ESP32 – CAMEYE, kde je na desce plošného spoje rovnou připojen i UART/USB převodník.

[40] [41] [42] [43]



Obrázek 32. Vývojový kit ESP32 WROOM, [43]

Raspberry Pi

Na řadu přichází nejdražší deska, kterou můžeme připojit ke kamerovému modulu. Raspberry pi jsou minipočítače s obrovským výkonem. Její cena se pohybuje od stovek korun až po tisíce korun. Bohužel se v současnosti ale těžce shánějí. Prodávají se v různých variantách od malých desek, které spíše připomínají mikroprocesorovou desku od Arduina nebo ESP, až po desku o velikosti lidské dlaně.

Aktuálně je na trhu Raspberry Pi 4. generace, které poskytuje Wi-Fi 2,5 a 5 GHz, Bluetooth 5.0, připojení pro USB 3.0 a 2.0, ethernetový port, audio 3,5 jack, Camera port, microHdmi pro připojení k monitoru, 64 bit procesor s frekvencí 1,5 GHz. Podporuje I^2C , SPI a sériovou komunikaci. Paměť Ram se liší podle variant, kdy můžeme mít 1 GB, 2 GB, 4 GB a 8 GB. Výrobce doporučuje používat originální zdroj napětí, který je na 5 V/ 3 A dc. Cena za tuto desku se pohybuje v rozmezí od 1600 Kč do 3000 Kč. Bohužel se v současnosti potýkají s celosvětovou nedostupností.

Raspberry Pi Pico je nejmenší možná varianta desek z řady Raspberry. Tato deska neposkytuje moduly, jako je Wi-Fi a Bluetooth. Má pouze 264 KB SRAM paměti a 2 MB Flash paměti. Podporuje komunikaci přes GPIO piny po I^2C , SPI a UART, dále disponuje sleep mody a teplotním čidlem. Napájení se pohybuje od 1,8 V do 5,5 V a odebíraný proud je 140 mA. Cena za tuto malou desku je 150 Kč. [44] [45] [46] [47] [48]



Obrázek 33. Vývojový kit Raspberry Pi 4, zdroj [46]

STM32

Jsou vysoce výkonné mikrokontrolery, které se hodí pro velikou škálu aplikací v různých odvětvích, jako jsou lékařství, domácí zabezpečovací systémy, tiskárny a skenery. S příchodem řady STM32 přišlo DCMI (Digital Camera Interface), což odstranilo i problémy s připojením paralelních kamerových modulů. Vhodnými mikrokontrolery jsou STM32H7 a STM32F7, které mají vysoký výkon a paměť.

Podporují komunikaci pro I^2C (SMBus a PMBus), UART a SPI. Dále disponují třemi režimy pro nízkou spotřebu. Flash paměť o velikosti 512 KB, SRAM s velikostí 256 KB. Napájení se pohybuje v rozmezí od 1,7 V do 3,6 V. Teplota pro provoz je od $-40\text{ }^{\circ}\text{C}$ do $125\text{ }^{\circ}\text{C}$, jenže adekvátní teplota pro provoz je $-40\text{ }^{\circ}\text{C}$ do $85\text{ }^{\circ}\text{C}$.

STM32F7 se prodává v Camera development kitu, který má k dispozici kameru s rozlišením VGA s možností výměny, 4,3 LCD – TFT palcový displej reagující na dotyk, video interface, Ethernet připojení, slot na micro SD a další funkce. Napájení je od 3,3 V do 5 V. Tento kit je vhodný pro vytvoření domácího zabezpečovacího systému pro kutily ve třech provedeních. [49] [50]



Obrázek 34. Kamerový kit od STM32F7, zdroj [49]

| Tabulka pro porovnání jednotlivých hardwarových desek | | | | | | |
|---|--------------|------------------------|-----------------|----------------------|----------------------------|--------------------|
| Název hardwarových desek | rozměr | podporovaná komunikace | napájecí napětí | odběr proudu | implementované moduly | cena desky v Kč |
| Arduino uno | 53,4 x 68,6 | I2C, SPI, UART | 5 V až 12 V | 19 mA | / | 360 Kč až 800Kč |
| Arduino Nano | 18 x 45 | I2C, SPI, UART | 5 V až 12 V | 50 mA | / | 200 Kč až 500 Kč |
| Arduino Mega | 53,3 x 101,5 | I2C, SPI, UART | 5 V až 12 V | 50 mA | / | 700 Kč až 1100 Kč |
| ESP8266 | 57 x 31 | I2C, SPI, UART | 5 V | 10 μ A až 350 mA | Wi-Fi modul | 150 Kč |
| ESP32 | 55 x 28 | I2C, SPI, UART | 5 V | 80mA | Blue. a Wi-Fi modul | 210 Kč |
| ESP32 - CAM | 40 x 27 | I2C, SPI, UART | 5 V | 6 mA až 310 mA | Blue. a Wi-Fi modul | 310 Kč |
| Raspberry Pi | 85,6 x 56,5 | I2C, SPI, UART | 5 V | 3 A | Blue., Wi-Fi, Ether. modul | 1600 Kč až 3000 Kč |
| Raspberry Pi Pico | 21 x 51 | I2C, SPI, UART | 1,8 V až 5,5 V | 140 mA | / | 150 Kč |
| STM32F7 Camera Dev. Kit | / | I2C, SPI, UART | 3,3 V až 5 V | NENÍ UVEDENO | / | 2108 Kč |

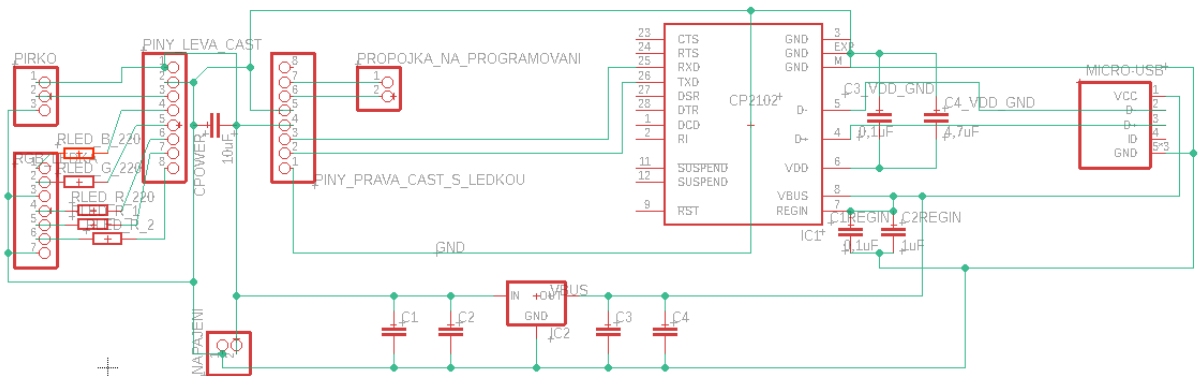
Tabulka 3. Porovnání jednotlivých hardwarových desek

2.Praktická část

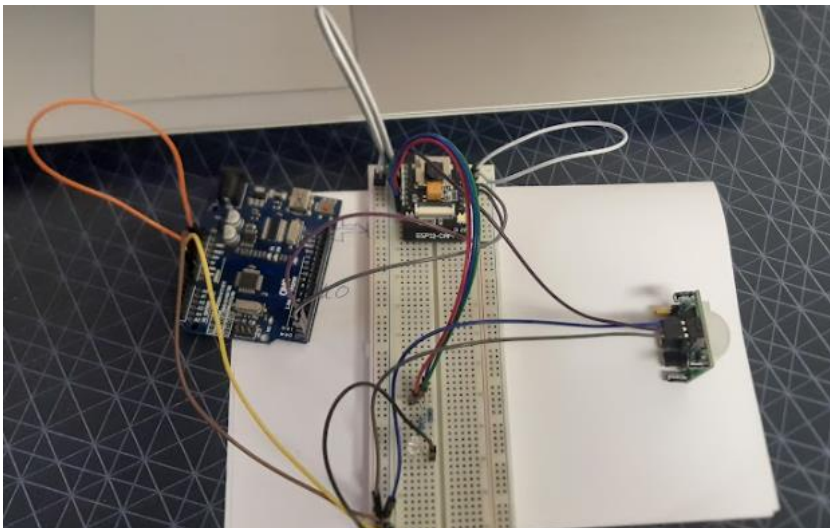
Druhá část práce se soustředí na konstrukci a konkrétní postupy, které jsem použil při realizaci mnou navržené kamery a fotopasti. Pro lepší přehlednost a lepší čtení práce jsem se rozhodl rozdělit praktickou část na dvě dílčí, z nichž každá se soustředí na specifický problém zvlášť. V první podkapitole se zaměříme na to, jak byla mnou navržená kamera konstruována, jaké součástky byly použity, specifické části kódu a uvedení kamery do chodu. Druhé podkapitola je věnována fotopasti od návrhu přes zhotovení, využití součástky při konstrukci, specifické části kódu až po její uvedení provozu.

2.1 Kamera

Kamera je založena na principu fotopasti, kdy v normální stavu je kamera uspaná pro úsporu energie, jakmile PIR senzor zaznamená pohyb nebo je aktivován, probudí kameru ze spánku a rozsvítí zelenou led diodu. Přes čtení sériové linky uživatel zjistí, jeli kamera připojena na internet nebo zda je nutno provést přihlášení. Pokud je kamera úspěšně připojena k síti, dostaneme do sériového terminálu vypsanou IP adresu, na kterou se můžeme připojit a sledovat přenos z kamery, jakmile je kamera připojena k internetu pošle, uživateli zprávu na aplikaci WhatsApp s IP adresou, kde je možno sledovat přenos z kamery. Po určité době se kamera opět uspí pro úsporu energie. To je signalizováno pomocí červené led diody, která opakovaně zabliká, a poté se kamera uspí.



Obrázek 35. Schéma zapojení obvodu pro kameru



Seznam použitých součástek

U sestavení kamery jsem se v některých částech inspiroval od kamer, které se volně prodávají na trhu. Kamera byla sestavena s použitím následujících součástek:

ESP32-CAM – tento modul jsem zvolil kvůli tomu, že splňuje moje požadavky na hardware, disponuje kamerovým modulem OV2640, čipem ESP32, jenž má Wi-Fi a Bluetooth a v poslední řadě disponuje slotem na microSD kartu. Navíc cena byla více než výhodná.

RGB Led dioda – díky třem základním barvám jsem schopný zprovoznit světelnou signalizaci pro lepší komunikaci mezi člověkem a kamerou.

PIR senzor – vybral jsem jej díky inspiraci ze světel reagujících na pohyb. Také je to skvělý způsob, jak probudit ESP32 z režimu deepsleep pomocí externího přerušení.

Anténa Wi-Fi 2,4 GHz – ESP32 – CAM má na sobě slot pro připojení externí antény pro zlepšení dosahu signálu. Zvolil jsem tedy možnost připojit externí anténu pro připojení k Wi-Fi, jelikož mám obavy z vestavěné antény na ESP32.

Rezistory – dva rezistory jsou hodnoty 220Ω pro RGB led diodu.

Piny samec a samice – pro jednodušší opravu zařízení dávám na DPS piny samice, abych mohl desku ESP32-CAM kdykoliv sundat a vyměnit za jinou. Opět pro jednoduchost zařízení a opravu zařízení.

Stabilizátor na 5 V – pro stabilizování napětí na lince Vbus

Zdroj pro napájení 12 V / 2 A – tento zdroj mi zůstal ze zakázky na montáž kamerového systému a skvěle se hodí pro napájení kamery.

Kondenzátory – pro stabilizátor a proti poklesu napájení.

Deska plošného spoje tedy (DPS) - desku jsem navrhl v programu EAGLE.

Krabice pro kameru – byla navržena mnou a také mnou vytištěna na 3D tiskárně

Svorkovnice – pro připojení kabelů ze zdroje napájení.

Převodník cp2102 – ESP32 – CAM nedisponuje USB/UART převodníkem pro komunikaci s počítačem. Pro nahrání je potřeba zapojit ještě Arduino UNO jako převodník a správně pozapojovat piny a opět rozpojit.

| Seznam použitých součástek | |
|----------------------------|--------|
| Název součástky | Cena |
| ESP32-CAM | 310 Kč |
| PIR senzor | 40 Kč |
| zdroj 12 V / 2 A | 0 Kč |
| RGB led dioda | 5 Kč |
| Ant. Wi-Fi 2,4 GHz | 33 Kč |
| Rezistory | 5 Kč |

| | |
|---------------------|--------|
| Piny samec a samice | 20 Kč |
| Stabilizátor na 5 V | 10 Kč |
| Kondenzátory | 30 Kč |
| Svorkovnice | 3 Kč |
| Převodník cp2102 | 100 Kč |
| DPS | 0 Kč |
| Krabice | 0 Kč |
| Cena celkem | 526 Kč |

Tabulka 4. Seznam použitých součástek

Návrh desky plošného spoje (DPS)

Jak jsem již zmiňoval na předchozí stránce. DPS jsem navrhl v programu Eagle. Pro jednoduchost při opravě nebo výměně kamerového modulu OV2640 za modul jiný jsem zvolil možnost dát na DPS piny samice, do které se zasouvá celá samotná deska ESP32 – CAM. Tím jsem docílil toho, že mám lepší přístup k tlačítku RESET a slotu na microSD kartu, jež se nachází na zadní straně ESP32-CAM.

K převodníku CP2102 je připojeno microUSB B pro komunikaci s počítačem. Zapojení celého převodníku vychází z datasheetu od výrobce [51], kondenzátory v okolí převodníku jsou přesně podle hodnot uvedených ve zmíněném datasheetu. Z převodníku vychází spoj na RX a TX přes piny, které slouží pro komunikaci mezi CP2102 a ESP32-CAM. RX a TX jsou přivedeny piny samice, do které se zasouvá ESP32-CAM, a jsou spojeny s piny UnR a U0T, tedy RX na UnR a U0T na TX. Tímto způsobem je zajištěna komunikace mezi počítačem a kamerou.

Stabilizátor na 5 V je v obvodu pro stabilizaci napětí na lince VBUS.

Na druhé straně jsou vyvedeny spoje pro rezistory na RGB led diodu a spoj pro PIR senzor.

Rezistory pro led diodu jsem zvolil z výpočtu výstupního napětí a výstupního proudu na pinu. Hodnoty pro napětí a proud na pinu jsem vyčetl z datasheetu od výrobce [43]. Proud na pinech se pohybuje od 5 mA až do 40 mA. Napětí na výstupu je 5 V a led dioda má s každou barvou odlišný úbytek napětí, proto tedy odečítám napětí ve výpočtu.

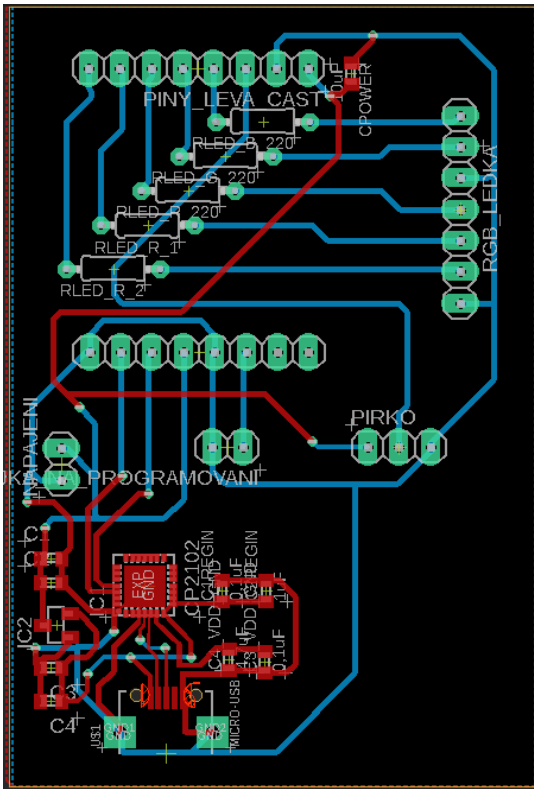
$$R_Z = \frac{U}{I} = \frac{U-U_d}{I} = \frac{5V-2,6V}{40mA} = 60 \Omega \quad R_{\check{c}} = \frac{U}{I} = \frac{U-U_d}{I} = \frac{5V-1,8V}{40mA} = 80 \Omega$$

Výsledek tedy je 220 Ω , zvolil jsem tedy nejbližší hodnotu odporu, kterou jsem zrovna měl k dispozici v dílně. Zároveň je to jedna z nejběžnějších hodnot odporu pro led diodu.

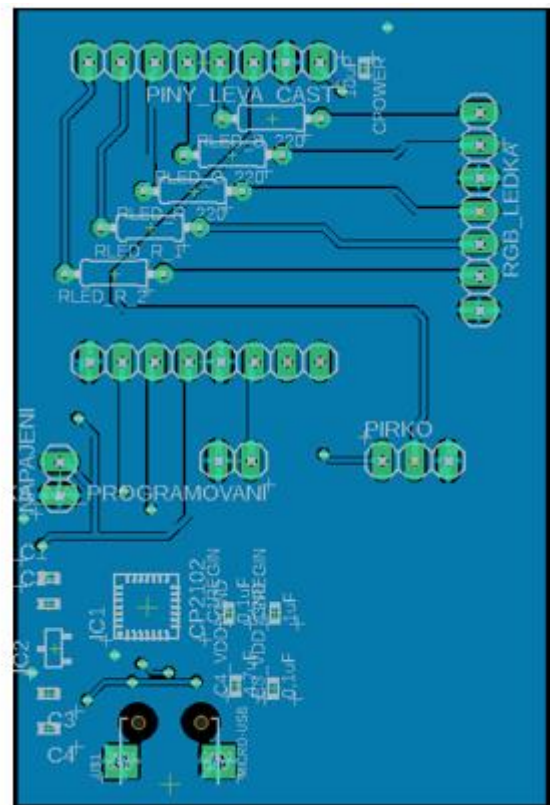
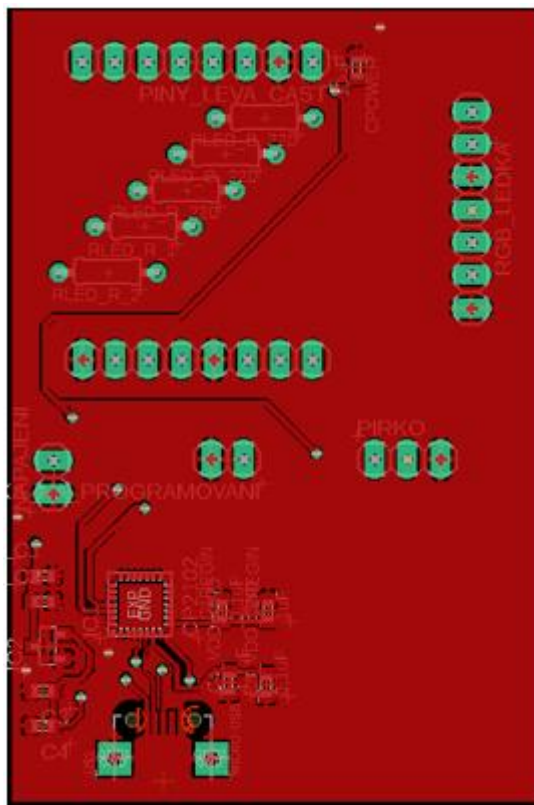
Pro připojení externí antény jsem odpájel rezistor ne ESP32-CAM, který byl připojen před implementovanou anténou na této desce.

Vzhledem k tomu, že počítám s budoucím rozšířením periférií na kameře, rozhodl jsem se proto umístit na DPS už rovnou přípravu v podobě možných rezistorů a celkem sedmi vyvedených pinů pro připojení periférii. Mohlo by se třeba jednat o vstupní periférie, jako je mikrofon pro zaznamenávání zvuku z okolí, teploměr pro měření teploty v monitorované oblasti a senzor pro měření tlaku v okolí. Dále bych chtěl přidat i nějaké výstupní periférie, jako je reproduktor pro přenos mého hlasu z monitorovacího zařízení do mikrofonu na kameře, pro komunikaci s lidmi v pozorované oblasti. Popřípadě i servomotor by nemusel být špatný pro natáčení úhlu zorného

pole kamery. Zkrátka bych tu svoji kamerku chtěl dostat na úroveň bezpečnostních kamer, jež si pořizují lidé z čínských e-shopů.



Obrázek 37. Deska plošného spoje



Obrázek 38. Červená horní strana a modrá dolní strana DPS

Specifické části kódu

Pro svou práci vycházím z obecného kódu pro kameru ESP32-CAM, který je k dostání v tutoriálových příkladech v Arduino. IDE. Tento kód slouží jako tutoriál k prvotnímu spuštění kamery a spuštění web serveru s aplikací, kam se přenáší obraz z kamery.

K tomuto základnímu kódu jsem přidal nastavení pinů pro výstupní a vstupní periferie spolu s následujícími funkcemi, které poskytují mnou požadované modifikace. Funkce pro uspávání do režimu deepsleep, jež ESP32 podporuje, funkce pro výpis důvodu probuzení a funkce pro výpis důvodu restartu zařízení i funkce pro posílání zprávy do aplikace WhatsApp. Dále dvě vlákna, jež každé z nich má jinou úlohu v podobě uspávání a vypisování údajů. Externí probuzení je realizováno pomocí RTC pinu na GPIO12, ke kterému je připojen PIR senzor, a jako poslední je tu připojení k internetu pomocí WiFiManager knihovny

Použité knihovny pro tento program jsou vidět na obrázku č.38

```
3  #include "esp_camera.h"
4  #include <WiFi.h>
5  #include <WiFiManager.h>
6  #include <rom/rtc.h>
7  #include "soc/soc.h"
8  #include "soc/rtc_cntl_reg.h"
9  #define CAMERA_MODEL_AI_THINKER
10 #include "camera_pins.h"
11 #include <HTTPClient.h>
12 #include <UrlEncode.h>
```

Obrázek 39. Seznam použitých knihoven v programu

Knihovna “esp_camera.h“ slouží ke konfiguraci kamerového modulu s čipem ESP32.

Knihovna “WiFi.h“ umožňuje připojení k internetu pomocí implementovaného Wi-Fi modulu na čipu esp32.

Knihovna “WiFiManager“ řeší ukládání přihlašovacích údajů od Wi-Fi sítě.

Knihovny od“rom/rtc“ až "soc/rtc_cntl_reg.h" zpřístupňují RTC piny pro použití externího probuzení.

Knihovny “camera_pins.h“ je definice pro piny na kamerovém modulu.

Knihovna “ESPmDNS.h“ je použita pro nastavení přejmenování ip adresy na použitelnější jméno pro uživatele.

Knihovny HTTPClient.h a UrlEncode jsou zde pro zprovoznění odesílání zprávy na WhatsApp.

Jako první jsem si definoval názvy pinů pro přehlednost za pomocí maker.

```
14 #define RED_PIN 2
15 #define GREEN_PIN 15
```

Obrázek 40. Definování maker pro GPIO piny

V dalším kroku jsem k těmto GPIO pinům nastavil OUTPUT stav, díky kterému je možno k nim připojit periferie, já jsem použil RGB led diodu.

```
70 pinMode(RED_PIN,OUTPUT);
71 pinMode(GREEN_PIN,OUTPUT);
```

Obrázek 41. Nastavení GPIO pinů do režimu OUTPUT

Nastavení pinu pro externí probuzení z deepsleepu modu, tedy RTC pin.

```
81 esp_sleep_enable_ext0_wakeup(GPIO_NUM_12, HIGH);
```

Obrázek 42. Nastavení pinu pro externí probuzení

Jako poslední jsou dva stringy, kde je telefonní číslo a apikey pro komunikaci s CallmeBotem a odesílání zpráv do WhatsApp aplikace.

```
18 String phoneNumber = "+420723505598";
19 String apiKey = "5385767";
```

Obrázek 43. String s údaji pro spuštění CallmeBota

Funkce **Void uspávání()**; slouží pro upozornění, že se zařízení za chvíli uspí do režimu deepsleep. To provádí v následujícím pořadí:

1. vypne zelenou led diodu, aby se nedošlo ke současnému svícení zelené led diody a červené led diody, což by mohlo vést k pomatění uživateli.
2. vypíše na sériovou komunikaci hlášku " jdu spát=====", čímž zařízení dává uživateli vědět, že se hodlá uspat do úsporného režimu.
3. Spustí se for cyklus, ve kterém 7x zabliká červená led dioda. To slouží jako světelná signalizace pro uspání zařízení.
4. Aktivuje se režim deepsleep a zařízení je uspáno.

```
255 void uspavani()
256 {
257     digitalWrite(GREEN_PIN,LOW);
258     Serial.println("jdu spat=====");
259     for (size_t i = 0; i < 7; i++)
260     {
261         digitalWrite(RED_PIN,HIGH);
262         delay(500);
263         digitalWrite(RED_PIN,LOW);
264         delay(500);
265     }
266     Serial.flush();
267     esp_deep_sleep_start();
268
269 }
```

Obrázek 44. Funkce Void uspávání(); pro uspávání zařízení

Funkce **Void print_wakeup_reason()**;; úkolem této funkce je vypsání důvodu probuzení mikroprocesoru z režimu deepsleep do sériové komunikace.

```
29 void print_wakeup_reason()
30 {
31     esp_sleep_wakeup_cause_t wakeup_reason;
32     wakeup_reason = esp_sleep_get_wakeup_cause();
33     switch(wakeup_reason) {
34         case ESP_SLEEP_WAKEUP_EXT0 : Serial.println("Wakeup caused by external signal using RTC_IO"); break;
35         case ESP_SLEEP_WAKEUP_EXT1 : Serial.println("Wakeup caused by external signal using RTC_CNTL"); break;
36         case ESP_SLEEP_WAKEUP_TIMER : Serial.println("Wakeup caused by timer"); break;
37         case ESP_SLEEP_WAKEUP_TOUCHPAD : Serial.println("Wakeup caused by touchpad"); break;
38         case ESP_SLEEP_WAKEUP_ULP : Serial.println("Wakeup caused by ULP program"); break;
39         default : Serial.printf("Wakeup was not caused by deep sleep: %d\n",wakeup_reason); break;
40     }
41 }
```

Obrázek 45. Void print_wakeup_reason() pro výpis důvodu probuzení

Funkce **Void print_reset_reason(RESET_REASON reason)** má za úkol vypsát do sériové komunikace důvod restartu mikroprocesoru po probuzení z deepsleepu .

```
43 void print_reset_reason(RESET_REASON reason)
44 {
45     switch ( reason)
46     {
47         case 1 : Serial.println ("POWERON_RESET");break; /**<1, Vbat power on reset*/
48         case 3 : Serial.println ("SW_RESET");break; /**<3, Software reset digital core*/
49         case 4 : Serial.println ("OWDT_RESET");break; /**<4, Legacy watch dog reset digital core*/
50         case 5 : Serial.println ("DEEPSLEEP_RESET");break; /**<5, Deep Sleep reset digital core*/
51         case 6 : Serial.println ("SDIO_RESET");break; /**<6, Reset by SLC module, reset digital core*/
52         case 7 : Serial.println ("TGW0WDT_SYS_RESET");break; /**<7, Timer Group0 Watch dog reset digital core*/
53
54         case 8 : Serial.println ("TG1WDT_SYS_RESET");break; /**<8, Timer Group1 Watch dog reset digital core*/
55         case 9 : Serial.println ("RTCWDT_SYS_RESET");break; /**<9, RTC Watch dog Reset digital core*/
56         case 10 : Serial.println ("INTRUSION_RESET");break; /**<10, Intrusion tested to reset CPU*/
57         case 11 : Serial.println ("TGW0WDT_CPU_RESET");break; /**<11, Time Group reset CPU*/
58         case 12 : Serial.println ("SW_CPU_RESET");break; /**<12, Software reset CPU*/
59         case 13 : Serial.println ("RTCWDT_CPU_RESET");break; /**<13, RTC Watch dog Reset CPU*/
60         case 14 : Serial.println ("EXT_CPU_RESET");break; /**<14, for APP CPU, reseted by PRO CPU*/
61         case 15 : Serial.println ("RTCWDT_BROWN_OUT_RESET");break;/**<15, Reset when the vdd voltage is not stable*/
62         case 16 : Serial.println ("RTCWDT_RTC_RESET");break; /**<16, RTC Watch dog reset digi. core and rtc module*/
63         default : Serial.println ("NO_MEAN");
64     }
65 }
```

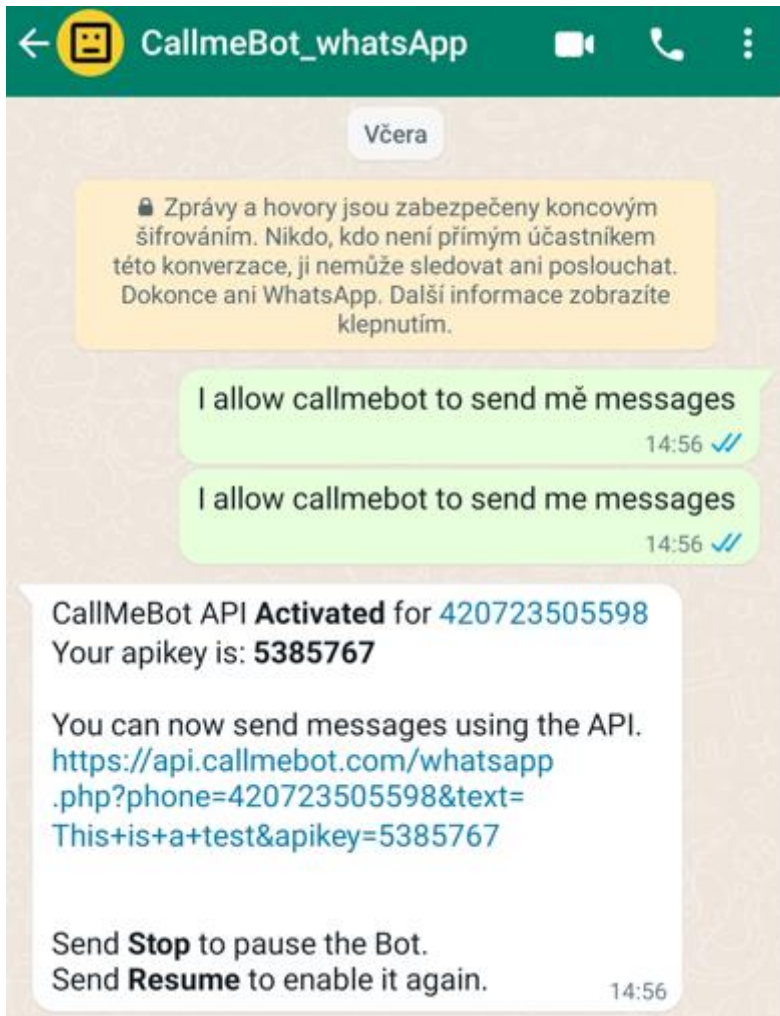
Obrázek 46. Void print_reset_reason(RESET_REASON reason) pro výpis důvodu restartu

Aby funkce **Void print_reset_reason(RESET_REASON reason)** a **Void print_wakeup_reason()** mohly fungovat v našem programu, je potřeba přidat do programu knihovny, které podporují použití RTC pinů.

Funkce **Void sendMessage(String message)** je v programu pro odesílání upozornění uživateli o probuzení kamery z režimu deepsleep. Pro zprovoznění této funkce je zapotřebí se připojit do služby CallmeBot přes aplikaci WhatsApp. Uživatel musí mít ve svých kontaktech přidane telefonní číslo +34 644 51 95 23. Na toto číslo uživatel musí napsat přes službu WhatsApp přihlašovací hlášku „I allow calmebot to send me messages“, tím se uživatel zaregistruje do této služby a dostane od CallmeBota tzv „apikey“, který se později používá pro posílání zpráv.

Uvnitř funkce je vytvořené URL s apikey, telefonním číslem, textem. Zpráva je zašifrována pomocí UriEncode knihovny. Pro posílání zprávy je potřeba vytvoření HTTP POST request,

jež následně čeká kontrolu podmínky s odpovědí v podobě kódu 200. Pokud přijde od http POST requestu kód 200, zpráva byla úspěšně poslána



Obrázek 47. Aktivace služby CallmeBot

```
266 void sendMessage(String message){
267     // Data to send with HTTP POST
268     String url = "https://api.callmebot.com/whatsapp.php?phone=" + phoneNumber + "&apikey=" + apiKey + "&text=" + urlEncode(message);
269     HttpClient http;
270     http.begin(url);
271
272     // Specify content-type header
273     http.addHeader("Content-Type", "application/x-www-form-urlencoded");
274
275     int httpResponseCode = http.POST(url);
276     if(httpResponseCode==200){
277         Serial.print("zpráva pro WhatsApp se úspěšně poslala");
278     }
279     else{
280         Serial.println("chyba v posílání zprávy");
281         Serial.print("HTTP response code");
282         Serial.println(httpResponseCode);
283     }
284     http.end();
285 }
```

Obrázek 48. Funkce pro posílání zprávy na WhatsApp aplikaci

Dvě použitá vlákna v programu **Vlakyngo1_fce (void * parameters)** a **Vlakyngo2_fce (void * parameters)** jsem se rozhodl použít, protože jsem potřeboval monitor hlídání času pro uspání

zařízení a také pro opakovatelné vypisování IP. adresy do sériové komunikace. Použitím vláken jsem dosáhl lepší přehlednosti kódu a nebylo nutno využívat smyčku loop, která je k dispozici.

```
24 TaskHandle_t Vlakyngo1_hdl;
25 TaskHandle_t Vlakyngo2_hdl;
26 void Vlakyngo1_fce (void * pvParameters);
27 void Vlakyngo2_fce (void * pvParameters);
```

Obrázek 49. Definování vláken

TaskHandle_t Vlakyngo1_hdl; a TaskHandle_t Vlakyngo2_hdl; jsou takzvané ukazatelé vláken, jakmile máme vytvořené tyto ukazatele, je možno zavolat funkci pro vytvoření vlákna.

Vlakyngo1_fce (void * parameters)

Na obrázku č. 50 je popsáno vytvoření vlákna. S tím přichází i možnost vytvoření funkce, kde bude běžet vlákno, jeho jméno, velikost, jeho ukazatel a taky možnost, na kterém jádře poběží vlákno, jelikož ESP32 je dvoujádrový mikroprocesor.

```
85 xTaskCreatePinnedToCore(
86     Vlakyngo1_fce, /* Function to implement the task */
87     "Vlakno1", /* Handle - Name of the task */
88     10000, /* Stack size in words */
89     NULL, /* Task input parameter */
90     0, /* Priority of the task */
91     &Vlakyngo1_hdl, /* Task handle. */
92     tskNO_AFFINITY); /* Core where the task should run */
```

Obrázek 50. Vytvoření Vlakyngo1_fce

Základní činností je udržet zařízení zapnuté, aby mohlo proběhnout připojení na internet nebo uživatel mohl pozorovat přenos videa přes webový server, a po uplynutí doby 5 minut se zařízení uspí do deepSleep režimu.

Činnost funkce Vlakyngo1_fce se dá popsat do těchto kroků:

1. Jako první věc vykoná výpis na sériovou linku, aby uživatel věděl, proč se zařízení probudilo.
2. Na sériovou linku vypíše hlášku “ vlakno bylo vytvořeno“, čímž dává vědět, že se vlákno vytvořilo a provádí se jeho obsah.
3. Zapne zelenou led diodu a vypíše na sériovou komunikaci “ ROZSVITIL JSEM ZELENOU“.
4. Následně čeká po dobu 5 minut zhasne, led diodu a zavolá funkci uspavani();.


```

219 void Vlakynko1_fce (void * pvParameters)
220 {
221
222     while(1)
223     {
224         print_wakeup_reason();
225         Serial.println("vlakno bylo vytvořeno");
226         digitalWrite(GREEN_PIN,HIGH);
227         Serial.println("ROZSVITIL JSEM ZELENOU");
228         delay(180000);
229         digitalWrite(GREEN_PIN,LOW);
230         uspravani();
231
232     }
233 }
234 }

```

Obrázek 51. Obsah prvního vlákna

Vlakynko2_fce (void * parameters)

Na obrázku 44. je ukázka vytvoření druhého vlákna a jeho vlastností.

```

205     xTaskCreatePinnedToCore(
206         Vlakynko2_fce, /* Function to implement the task */
207         "Vlakno2", /* Handel - Name of the task */
208         10000, /* Stack size in words */
209         NULL, /* Task input parameter */
210         0, /* Priority of the task */
211         &Vlakynko2_hndl, /* Task handle. */
212         tskNO_AFFINITY); /* Core where the task should run */
213
214
215     }

```

Obrázek 52. Vytvoření Vlakynko2_fce

Druhé vlákno slouží pro opakované vypisování IP. Adresy do sériové komunikace, jelikož byl problém s prvotním připojením na síť po prvním zapnutí. IP. Adresa se ukázala pouze jednou a uživatel musel čekat znovu, než se zařízení uspí a celé připojení začne znovu, aby mohl číst sériovou linku. Proto jsem přidal druhé vlákno, které za mě jednou za půl minuty vypíše aktuálně používanou IP adresu.

```

236 void Vlakynko2_fce (void * pvParameters)
237 {
238 while(1)
239 {
240     delay(30000);
241     Serial.println("druhe vlakno bylo vytvořeno ");
242     Serial.print("Camera Ready! Use 'http://");
243     Serial.println(WiFi.localIP());
244     Serial.println("' to connect");
245 }
246
247 }

```

Obrázek 53. Obsah druhého vlákna

Připojení k internetu pomocí **WiFiManager** knihovny

Připojení k internetu jsem vyřešil pomocí knihovny WiFiManager, která za mě řeší automatické připojování do sítě a ukládání přihlašovacích údajů do paměti mikroprocesoru.

Jak jsem již, zmínil tato knihovna za mě učiní automatické připojování do sítě, a to pokaždé, kdy je kamera probuzena z režimu deepsleep, jestliže zařízení bylo už jednou připojeno na síť, udrží si přihlašovací údaje v paměti pro opětovné připojení. Funguje to, i pokud zařízení připojíme na jinou síť, která již byla připojena na kameru. Kamera se vždy automaticky připojí na uložené síť.

```

176 bool res;
177 res = wm.autoConnect("AutoConnectAP","password");
178
179 if(!res)
180 {Serial.println("Failed to connect...");
181 ESP.restart();
182 }
183
184 while (WiFi.status() != WL_CONNECTED) {
185     delay(500);
186     Serial.print(".");
187 }
188
189 startCameraServer();
190
191
192 Serial.print("Camera Ready! Use 'http://");
193 Serial.print(WiFi.localIP());
194 Serial.println("' to connect");
195 sendMessage("kamera byla spuštěna obraz je na této adrese");
196 sendMessage(WiFi.localIP().toString().c_str());

```

Obrázek 54. Připojení k internetu pomocí WiFiManager knihovny

Připojení kamery na neznámou síť se dá popsat v těchto krocích.

1. Za prvé jsem provedl vytvoření objektu wm.

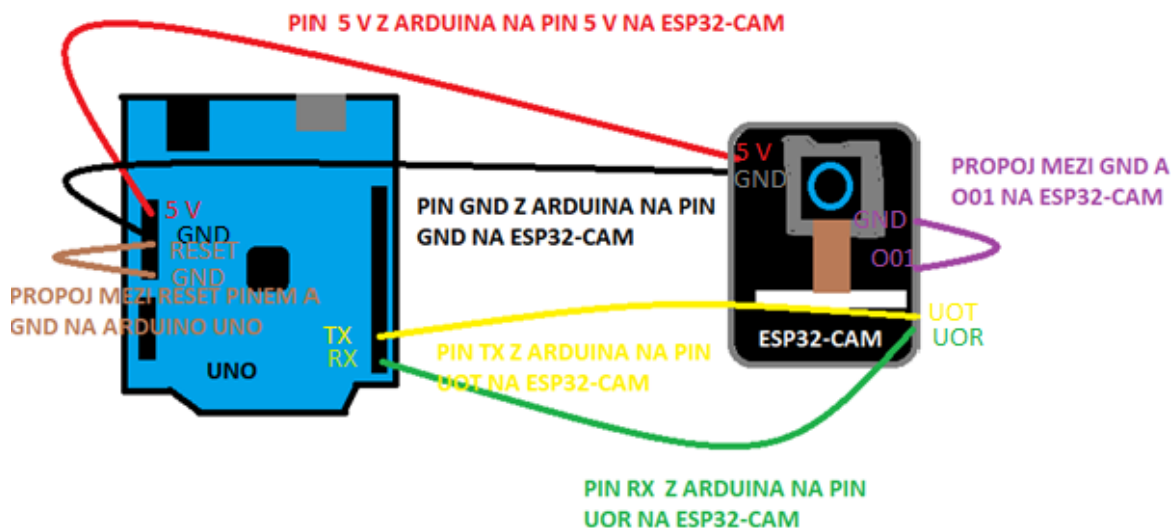
2. Druhým krokem bylo vytvoření přístupového bodu a udělení mu jeho přihlašovacích údajů, pokud se přihlášení nepovede, vypíše se hláška na sériový terminál.
3. Cyklus while pro vypisování teček do sériového terminálu, kvůli signalizaci připojování.
4. Zahájení webového serveru, na kterém je přenos z kamery.
5. Vypsání IP adresy do sériového terminálu na tuto adresu se uživatel připojí pro sledování přenosu z kamery.
6. Odeslání IP adresy do WhatsApp aplikace.

Uvedení kamery do provozu

Kamera je uvedena do provozu v 5 krocích, jež zde budou rozepsány a vysvětleny na obrázcích. Zprovoznění kamery je těžší než zprovoznění fotopasti hlavně kvůli rozdílným kódům a tím, jak fungují obě zařízení. Nahrávání programu do ESP32-CAM je u obou zařízení totožné se stejnými problémy, na které si uživatel musí dávat pozor.

1. KROK – nahrání programu do ESP32-CAM

Nahrávání programu do fotopasti je komplikované z toho důvodu, že deska na sobě nemá převodník, proto jsem při cvičném zapojení na nepájivém poli byl donucen použít Arduino Uno jako převodník pro USB/UART, které bylo konstantně vyresetováno - viz obrázek níže.



Obrázek 55. Teoretický obrázek propojení Arduina Uno a ESP32-CAM pro účely programování (na obrázku jsou znázorněny piny na propojení)

Zapojení Arduina Una a ESP32 – CAM je následující: jako první věc musíme zapojit napájení 5 V a GND, to vysvětlují červená a černá čára na obrázku, červená pro 5 V a černá pro GND. Druhou věcí je zapojit UART komunikaci mezi ESP32 – CAM a Arduino Uno, to docílíme

spojením pinů TX z Arduina Una a UoT na ESP32-CAM pro vysílání dat a pro příjem dat dáme spoj mezi pinem RX z Arduina a UoR na ESP32 – CAM pro přijímání dat. Třetím krokem je propojení pinu RESET a GND na Arduinu a na ESP32-CAM. Pokud jsou tyto kroky splněny, začne se nahrávat kód z počítače do ESP32-CAM. Člověk by měl mít na paměti, že někdy je potřeba pro zahájení nahrávání zmáčknout resetovací tlačítko na spodku desky pro zahájení nahrávání. Tento problém by se dal vyřešit připojením 10 µF kondenzátoru na pin ENABLE, jenže tento pin není na tomto modelu mikroprocesoru vyveden.

Pokud jde tedy o tu variantu, kdy se nahrává program na mnou navrženou desku, je potřeba pouze dávat na pozor na dvě věci a to jsou: za prvé propoj mezi pinem GPIO01 a GND a za druhé, jestli je potřeba zmáčknout tlačítko RESET pro zahájení nahrávání. Na mnou navržené desce je už převodník USB/UART pro přímé připojení k počítači.

2. KROK – spuštění kamery

Druhým krokem je samotné spuštění kamery, jelikož se dokončilo nahrávání programu do ESP32-CAM, zbývá udělat ještě dvě věci: vytáhnout propojku mezi pinem GPIO01 a GND, zmáčknout RESET tlačítko na spodní straně desky.

Zmáčknutím RESET tlačítka se zařízení restartuje a spustí nahraný kód a začne se provádět od začátku. To, že se spustí kamera, je signalizováno rozsvícením zelené led diody. Na sériovém terminálu se nám vypíše důvod probuzení, tedy po resetování je to reset reason a dále to bude vypsání dvou hlášek. Hláška „vlakno bylo vytvořeno“ oznamuje, že se spustilo první vlákno **Vlakynko1_fce (void * parameters)**. Druhá hláška „ROZSVITIL JSEM ZELENOU“ oznamuje, že zařízení je aktivní a je spuštěna doba 5 minut pro přihlášení do sítě a monitorování pozorované oblasti. Dále následuje přihlášení do sítě.

```
18:02:03.781 -> rst:0x5 (DEEPSLEEP_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
18:02:03.781 -> config: 0, SPIWP:0xee
18:02:03.781 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
18:02:03.781 -> mode:DIO, clock div:1
18:02:03.781 -> load:0x3fff0030,len:1344
18:02:03.828 -> load:0x40078000,len:13864
18:02:03.828 -> load:0x40080400,len:3608
18:02:03.828 -> entry 0x400805f0
18:02:04.299 ->
18:02:04.299 -> Wakeup caused by external signal using RTC_IO
18:02:04.340 -> vlakno bylo vytvořeno
18:02:04.340 -> ROZSVITIL JSEM ZELENOU
```

Obrázek 56. Vypsání hlášek do sériového terminálu po spuštění

3. KROK – připojení do sítě

Přihlášení do sítě probíhá automaticky, jestliže byla kamera někdy už do sítě připojena - viz obrázek níže.

```

18:04:23.068 -> Wakeup was not caused by deep sleep: 0
18:04:23.068 -> vlakno bylo vytvořeno
18:04:23.068 -> ROZSVITIL JSEM ZELENOU
18:04:23.301 -> *wm:AutoConnect
18:04:23.394 -> *wm:Connecting to SAVED AP: TendaAC10U
18:04:23.908 -> *wm:connectTimeout not set, ESP waitForConnectResult...
18:04:25.212 -> *wm:AutoConnect: SUCCESS
18:04:25.212 -> *wm:STA IP Address: 192.168.2.135
18:04:25.212 ->
18:04:25.212 -> WiFi connected
18:04:25.212 -> Camera Ready! Use 'http://192.168.2.135' to connect

```

Obrázek 57. Automatické připojení do sítě

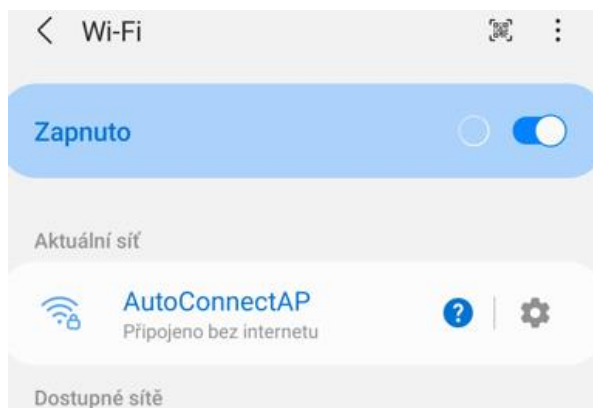
Pokud se kamera připojuje do sítě, kterou nezná, spustí pro ESP32-CAM access point pro konfiguraci přihlašovacích údajů. Tento access point najdeme v nabídce dostupných Wi-Fi sítí v našem okolí. V mém případě jsem access point pojmenoval AutoConnectAP a do sériového terminálu jsem dostal IP adresu, na kterou se mám připojit kvůli konfiguraci.

```

18:02:04.299 -> Wakeup caused by external signal using RTC_IO
18:02:04.340 -> vlakno bylo vytvořeno
18:02:04.340 -> ROZSVITIL JSEM ZELENOU
18:02:04.559 -> *wm:AutoConnect
18:02:04.590 -> *wm:Connecting to SAVED AP: Tenda_1D4610 spodni
18:02:05.104 -> *wm:connectTimeout not set, ESP waitForConnectResult...
18:02:07.605 -> *wm:AutoConnect: FAILED
18:02:07.605 -> *wm:StartAP with SSID: AutoConnectAP
18:02:08.116 -> *wm:AP IP address: 192.168.4.1
18:02:08.116 -> *wm:Starting Web Portal
18:02:37.237 -> *wm:1 networks found

```

Obrázek 58. Nezdařené připojení do sítě a spuštění access pointu pro konfiguraci připojení



Obrázek 59. Ukázka Wi-Fi sítí spolu s vytvořením AutoConnectAP

K připojení na AutoConnectAP je zapotřebí přihlašovacích údajů, které jsou na obrázku xx. na řádce 174. Na první pozici v závorce je jméno access pointu a na druhé pozici je heslo pro přihlášení do access pointu.

```

174   res = wm.autoConnect("AutoConnectAP", "password");

```

Obrázek 60. Přihlašovací údaje pro přihlášení do access pointu

Po úspěšném přihlášení do AutoConnectAP bude uživatel buď automaticky přesměrován na stránku, kde se provádí konfigurace Wi-Fi, nebo bude muset ručně zadat IP adresu 192.168.4.1 pro nasměrování na stránku konfigurace. Já osobně raději zadávám IP adresu ručně.

WiFiManager

AutoConnectAP

Configure WiFi

Info

















Exit

Update

Not Connected to Chrudim
AP not found

Obrázek 61. Stránka WiFiManageru pro konfiguraci přihlašovacích údajů sítě

Pro konfiguraci přihlašovacích údajů se musí kliknout na první tlačítko „Configure WiFi“, ta nás odkáže na další stránku, kde je k dispozici seznam dostupných Wi-Fi sítí v našem okolí, ze kterého si můžeme vybrat připojení. Dále se tu nachází kolonka SSID, je pro zapsání jména sítě, ale pokud se klikne na síť v seznamu dostupných sítí, bude tato kolonka pro nás předvyplněna. Pod kolonkou SSID je kolonka Password pro zapsání hesla, které se musí vždy zadávat ručně uživatelem, ještě máme možnost pro zobrazení hesla kvůli správnému zapsání. Následují dvě tlačítka „Save“ a „Refresh“. Tlačítko „Save“ slouží pro uložení přihlašovacích údajů a tlačítko „Refresh“ je pro znovunačtení okolních sítí. Na obrázku je ještě poslední okénko pro oznámení, že nemůže najít poslední síť, ke které bylo zařízení připojeno.

| | |
|--------------------------|---|
| Dominik |   |
| upa-zam |   |
| eduroam |   |
| guest |   |
| Hrob |   |
| Tenda_0AF690 |   |
| Global-Sport-kram |   |
| O2-Internet-771 |   |

SSID

Password

Show Password

Save

Refresh

Not Connected to Chrudim
AP not found

Obrázek 62. Stránka WiFiManageru pro připojení k síti.

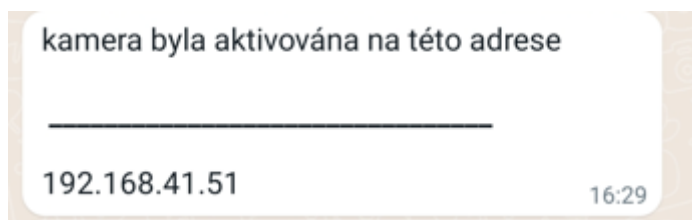
Proběhlo-li úspěšné připojení do sítě, kterou si uživatel zvolil, ukáže se mu nová stránka, na níž se zobrazí nápis z obrázku 63 (viz obrázek níže). Poté, kdy byly uloženy přihlašovací údaje do paměti mikroprocesoru, bude na sériový terminál vypsána IP adresa pro sledování záznamu z kamery. Pozor, přenos funguje pouze na lokální síti. Takže sledující musí být na stejné síti jako kamera. Pro někoho to může být problém, který se dá vyřešit tím, že by se implementoval RTSP protokol a server na nějaký veřejný server pro vysílání a sledování obrazu, nebo je tu zajímavější možnost, a to použít open platformu Home assistant. Pro použití řešení platformou Home assistant by bylo potřeba ještě zahrnout mDNS do kódu, jelikož by byl problém s IP adresou kamery, jestliže by se změnila síť, ze které vysílá obraz, a tím by kamera dostala novou IP adresu.

Saving Credentials
Trying to connect ESP to network.
If it fails reconnect to AP to try again

Obrázek 63. Úspěšné uložení Wi-Fi údajů do ESP32-CAM

```
16:29:09.680 -> *wm:connectTimeout not set, ESP waitForConnectResult...
16:29:11.531 -> *wm:Connect to new AP [SUCCESS]
16:29:11.531 -> *wm:Got IP Address:
16:29:11.531 -> *wm:192.168.41.51
16:29:12.547 -> *wm:config portal exiting
16:29:12.547 -> Camera Ready! Use 'http://192.168.41.51' to connect
16:29:14.164 -> zpráva pro WhatsApp se úspěšně poslala zpráva pro WhatsApp se úspěšně poslala druhe vlakno bylo vytvořeno
```

Obrázek 64. Vypsání IP adresy do sériového portu a odeslání zprávy do WhatsAppu, na kterém je možno sledovat obraz z kamery

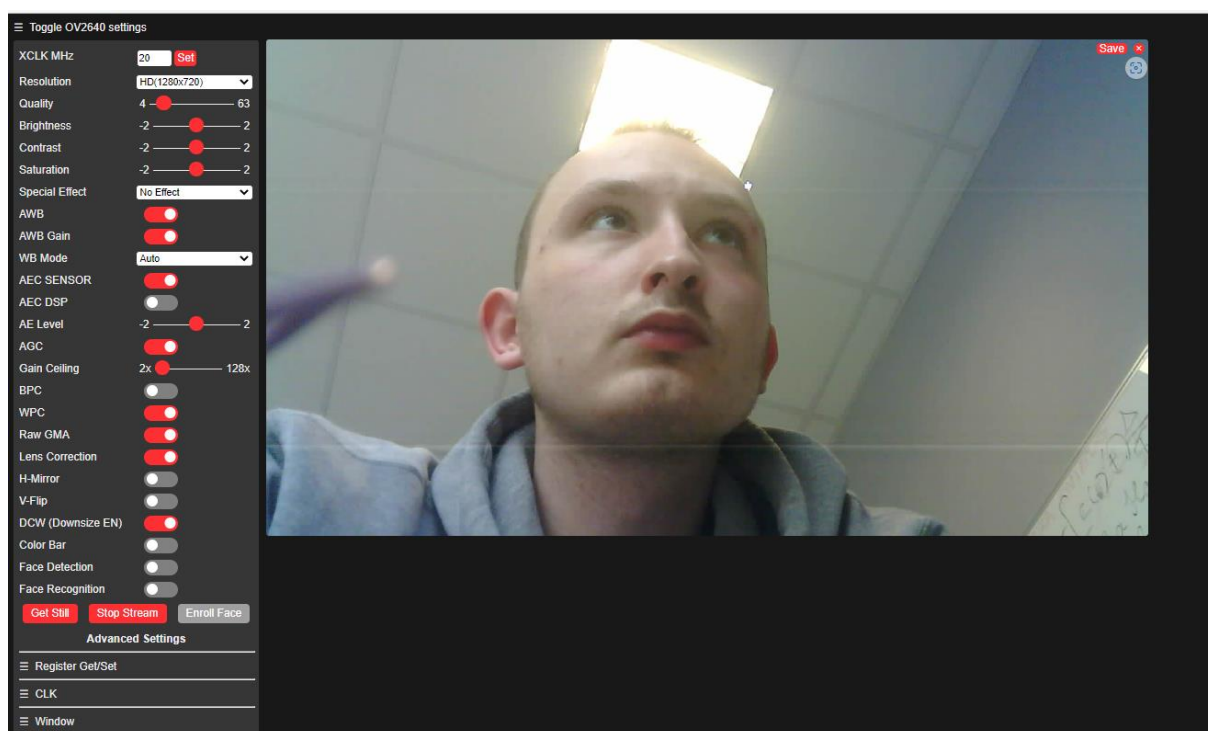


Obrázek 65. Zpráva na WhatsAppu s IP adresou kamery

4. KROK – spuštění přenosu z kamery

Z aplikace WhatsApp na mobilním telefonu nebo ze sériového terminálu se získala IP adresa pro připojení na sledování přenosu z kamery. Zadá-li se IP adresa do internetového vyhledávače, dostaneme se na webserver na obrázku 65.

Obraz kamery není okamžitě přenášen, je potřeba zmáčknout tlačítko pro zahájení přenosu „Start Stream“ a po jeho zmáčknutí se začne přenášet obraz z kamerového modulu. Pokud chceme ukončit přenos, zmáčkneme tlačítko „Stop Stream“, to vede k ukončení přenosu videa.



Obrázek 66. Snímek z přenosu kamery na webserveru.

Webserver nabízí spoustu úprav pro přenos videa, jak od změny rozlišení, přes úpravy jasu, kontrast, různé filtry, rozpoznávání obličejů až pořízení fotografie ve formátu .jpg. Přenos obrazu může být sledován, dokud svítí zelená led dioda, která signalizuje to, že je kamera v aktivním režimu, jakmile ale začne blikat červená led dioda, oznamuje nám kamera, že se uspí a obraz v režimu deepsleep nebude moct být sledován pro úsporu energie.

5. KROK – uspání kamery a znovuprobuzení kamery

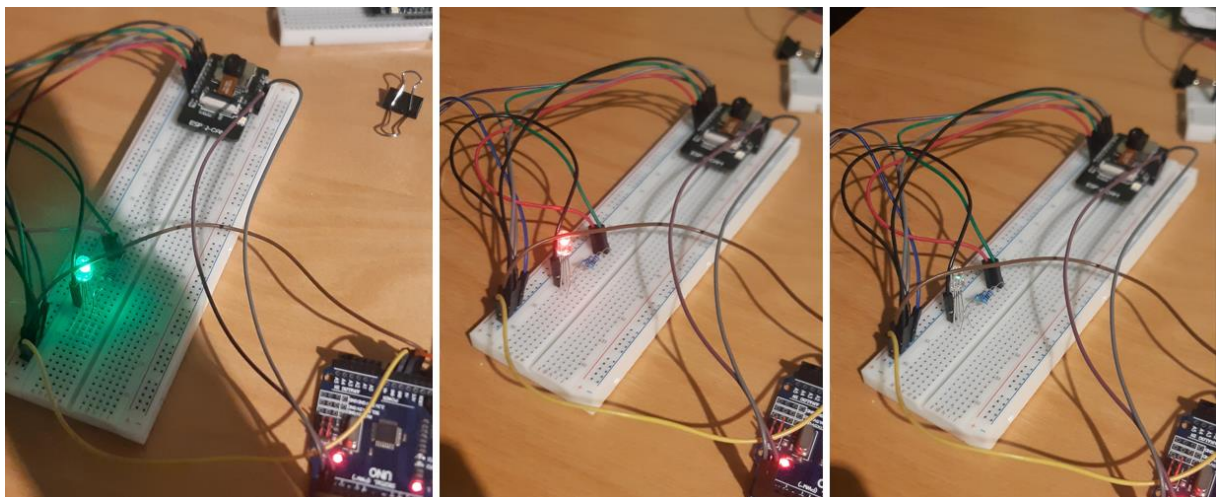
Na kameře začala blikat červená led dioda, která oznamuje, že má kamera poslední vteřiny před uspáním, ještě se vypíše do sériového terminálu hláška „jdu spát====“oznamující, že kamera se brzy uspí. Stihne se ještě jednou vypsát IP adresa pro sledování obrazu a kamera se uspí. To lze poznat tak, že na kameře nesvítí ani jedna barva na led diodě. Kamera je uspaná, dokud ji neprobudí externí probuzení ze signálu od PIR senzoru. Signál nepřijde, dokud PIR senzor

nezaznamená rozdílnou teplotu od původní teploty svého okolí ve svém zorném poli neboli nezaznamená pohyb v zorném poli kamery.

```
22:14:52.774 -> druhe vlakno bylo vytvořeno
22:14:52.774 -> Camera Ready! Use 'http://10.0.1.111
22:14:52.774 -> ' to connect
22:15:22.738 -> druhe vlakno bylo vytvořeno
22:15:22.738 -> Camera Ready! Use 'http://10.0.1.111
22:15:22.782 -> ' to connect
22:15:52.761 -> druhe vlakno bylo vytvořeno
22:15:52.761 -> Camera Ready! Use 'http://10.0.1.111
22:15:52.761 -> ' to connect
22:16:22.734 -> druhe vlakno bylo vytvořeno
22:16:22.769 -> Camera Ready! Use 'http://10.0.1.111
22:16:22.769 -> ' to connect
22:16:30.420 -> jdu spat=====
22:16:43.919 -> ets Jun 8 2016 00:22:57
```

Obrázek 67. Oznámení příchozího uspání do režimu deepsleep

Zelená led dioda signalizuje aktivní stav kamery, červená led dioda značí příchozí stav uspání a led dioda, jež nesvítí, signalizuje režim deepsleep.

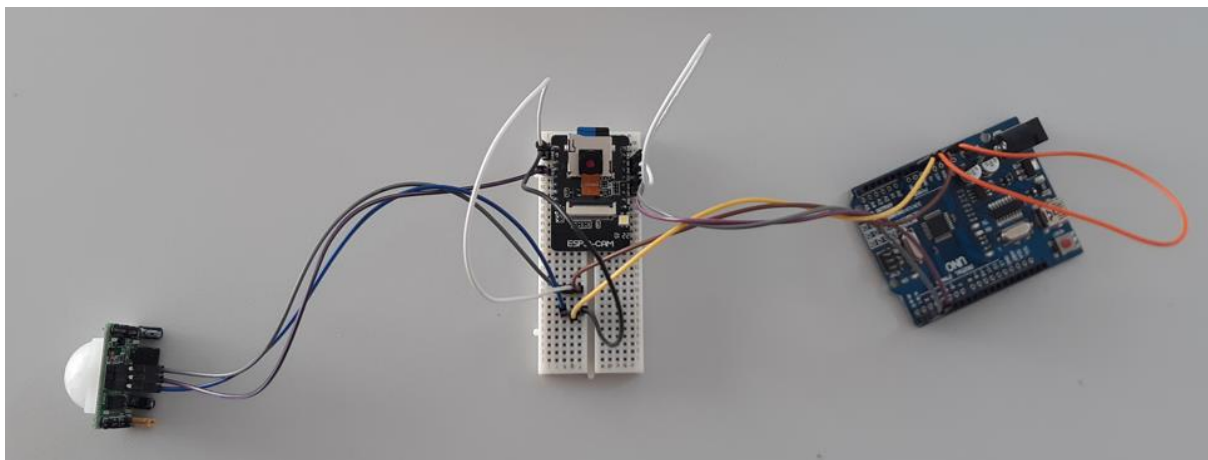


Obrázek 68. Postupné uspávání kamery

2.2 Fotopast

Fotopast bude fungovat na stejném principu jako kamera, ale až na pár výjimek. Fotopast bude umístěna v lese pro fotografování zvěře nebo pohybu života v lese u mě na vesnici. Data nebudou žádným způsobem posílat po síti kvůli nedostačujícímu pokrytí internetového signálu nebo mobilního pokrytí u nás na vesnici, proto je tedy budu ukládat na microSD kartu a jednou za čas ji přijdu zkontrolovat.

Princip fotopasti je následující. Fotopast bude zavěšena ve vhodné výšce asi do 2 m na stromě. Zařízení bude v konstantně deepsleep režimu pro úsporu energie na bateriích. Jakmile se na dosah PIR senzoru dostane nějaké zvíře nebo člověk, zkrátka zaznamená pohyb, probudí ESP32-CAM z úsporného režimu a kamera pořídí snímek, který je následně uložený do paměti na microSD kartě.



Obrázek 69. Cvičné zapojení fotopasti na nepájivém poli

Seznam použitých součástek

Fotopast je postavena ze stejných komponentů jako kamera, až na pár rozdílů, jelikož slot na microSD kartu zabírá většinu volných GPIO pinů, bylo tím omezeno použití periférií na minimum, tedy jedno PIR čidlo pro externí připojení.

ESP32 – CAM – tento modul jsem zvolil kvůli tomu, že splňuje moje požadavky na hardware, disponuje kamerovým modulem OV2640, čipem ESP32, jenž má Wi-Fi a Bluetooth a v poslední řadě disponuje slotem na microSD kartu. Navíc cena byla více než výhodná.

PIR senzor – pro detekci pohybu.

Stabilizátor na 5 V – pro stabilizaci linky VBUS.

MicroSD karta – pro ukládání fotografií.

Piny samec a samice – pro připojení ESP32 – CAM a PIR senzoru.

Kondenzátory – vybrané ke stabilizátoru a k převodníku cp2102 podle datasheetu [51].

Deska plošného spoje (DPS)- byla použita stejná jako u předchozí kamerové části.

Svorkovnice – pro připojení kabeláže od baterií.

Převodník Cp2102 – kvůli chybějícímu převodníku na desce ESP32-CAM.

Baterie GeB LiFePO4 Baterie IFR18650 1800mAh 3.2 V – vhodný zdroj napájení do lesa.

Navržená krabice vytištěná na 3D tiskárně.

Držák na baterie – pro spojení baterií za účelem napájení zařízení.

| Seznam použitých součástek | |
|----------------------------|----------|
| Název součástky | Cena |
| ESP32-CAM | 310 Kč |
| PIR senzor | 40 Kč |
| Baterie | 236 Kč |
| Držák na baterie | 57 Kč |
| Piny samec a samice | 20 Kč |
| Stabilizátor na 5 V | 10 Kč |
| Kondenzátory | 30 Kč |
| Svorkovnice | 3 Kč |
| Převodník cp2102 | 100 Kč |
| microSD karta | 269 Kč |
| Krabice | 0 Kč |
| DPS | 0 Kč |
| Cena celkem | 1 075 Kč |

Tabulka 5. Seznam použitých součástek

Návrh desky plošného spoje

Desku plošného spoje použiji stejnou desku, jako jsem navrhl pro kameru. Hodí se na tuto práci vzhledem k tomu, že je založena na stejných produktech jako již zmíněná kamera. Ovšem rozdíl bude v tom, že nebudou připojeny periférie, jako je led dioda nebo bzučák. Hlavní důvod, proč se nepřipojují k zařízení žádné další periférie, je ten, že slot pro microSD kartu zabere většinu GPIO pinů na levé straně mikroprocesoru ESP32 – CAM.

Z vlastní zkušenosti jsem zjistil, že většina RTC pinů nebude reagovat na externí probuzení a následné pořízení snímku i jeho uložení. Po několika pokusech a hledání na internetu jsem přišel na pin GPIO 13, který je vhodný pro probuzení z režimu deepsleep režimu pomocí externího probuzení.

Specifické části kódu

Při vytváření kódu pro fotopast jsem vycházel z kódu na stránkách RANDOM NERD TUTORIALS, kde byl ukázkový kód pro pořízení fotografie a jeho následné uložení do paměti microSD karty. Kód jsem upravil, abych k němu mohl přidat externí probuzení za použití PIR senzoru. Vlákno pro paralelní provedení hlídání usnutí a vypisování hlášek ke komunikaci s uživatelem. Hlavní část programu se provede v setup části, tedy pořízení snímku a jeho uložení do microSD karty, a vlákno se postará o zbytek potřebných částí programu usnutí a udržení vzhůru pro potřebnou dobu, aby se data mohla v klidu uložit. [52]

Na obrázku 70 jsou ukázány knihovny, jež byly použity pro kód.

```

2  #include "esp_camera.h"
3  #include "Arduino.h"
4  #include "FS.h"           // SD Card ESP32
5  #include "SD_MMC.h"     // SD Card ESP32
6  #include "soc/soc.h"    // Disable brownout problems
7  #include "soc/rtc_cntl_reg.h" // Disable brownout problems
8  #include "driver/rtc_io.h"
9  #include <EEPROM.h>     // read and write from flash memory
10 #include <rom/rtc.h>

```

Obrázek 70. Použité knihovny pro fotopast

Knihovna “esp_camera.h” je pro inicializaci pinů ke kameře a knihovna „Arduino.h“ je pro psaní v programu Arduino.IDE. Knihovny “FS.h” a “SD_MMC.h” jsou pro práci s microSD kartou. Knihovny “soc/soc.h”, “soc/rtc_cntl_reg.h”, “rom/rtc.h” a “driver/rtc_io.h” jsou pro rtc piny a vypnutí brownout registru. Knihovna “EEPROM.h” je pro práci s eeprom pamětí.

```

15  TaskHandle_t Vlakynko1_hndl;
16  void Vlakynko1_fce (void * pvParameters);
17

```

Obrázek 71. Definování ukazatele vlákna

Funkce **Void print_wakeup_reason()** je vidět na obrázku 64. , kde jsou znázorněny všechny hlášky, jež se vypisují do sériového terminálu, jakmile bylo zařízení probuzeno z režimu deepsleep. Úkolem této funkce je vypsání důvodů probuzení z deepsleepu -externí probuzení, probuzení pomocí časovače nebo díky senzoru pro dotyk.

```

29  void print_wakeup_reason()
30  {
31  esp_sleep_wakeup_cause_t wakeup_reason;
32  wakeup_reason = esp_sleep_get_wakeup_cause();
33  switch(wakeup_reason) {
34  case ESP_SLEEP_WAKEUP_EXT0 : Serial.println("Wakeup caused by external signal using RTC_IO"); break;
35  case ESP_SLEEP_WAKEUP_EXT1 : Serial.println("Wakeup caused by external signal using RTC_CNTL"); break;
36  case ESP_SLEEP_WAKEUP_TIMER : Serial.println("Wakeup caused by timer"); break;
37  case ESP_SLEEP_WAKEUP_TOUCHPAD : Serial.println("Wakeup caused by touchpad"); break;
38  case ESP_SLEEP_WAKEUP_ULP : Serial.println("Wakeup caused by ULP program"); break;
39  default : Serial.printf("Wakeup was not caused by deep sleep: %d\n",wakeup_reason); break;
40  }
41  }

```

Obrázek 72. Funkce pro vypsání důvodů probuzení

Funkce **Void print_reset_reason(RESET_REASON reason)** je vidět na obrázku 65. , kde jsou znázorněny všechny hlášky, které se vypíší do sériového terminálu, jakmile bylo zařízení resetováno probuzení z deepsleep režimu. Jeho úkolem je vypsání důvodů restartu mikroprocesoru.

```

43 void print_reset_reason(RESET_REASON reason)
44 {
45     switch ( reason)
46     {
47     case 1 : Serial.println ("POWERON_RESET");break; /**<1, Vbat power on reset*/
48     case 3 : Serial.println ("SW_RESET");break; /**<3, Software reset digital core*/
49     case 4 : Serial.println ("OWDT_RESET");break; /**<4, Legacy watch dog reset digital core*/
50     case 5 : Serial.println ("DEEPSLEEP_RESET");break; /**<5, Deep Sleep reset digital core*/
51     case 6 : Serial.println ("SDIO_RESET");break; /**<6, Reset by SLC module, reset digital core*/
52     case 7 : Serial.println ("TGWDT_SYS_RESET");break; /**<7, Timer Group0 Watch dog reset digital core*/
53
54     case 8 : Serial.println ("TG1WDT_SYS_RESET");break; /**<8, Timer Group1 Watch dog reset digital core*/
55     case 9 : Serial.println ("RTCWDT_SYS_RESET");break; /**<9, RTC Watch dog Reset digital core*/
56     case 10 : Serial.println ("INTRUSION_RESET");break; /**<10, Intrusion tested to reset CPU*/
57     case 11 : Serial.println ("TGWDT_CPU_RESET");break; /**<11, Time Group reset CPU*/
58     case 12 : Serial.println ("SW_CPU_RESET");break; /**<12, Software reset CPU*/
59     case 13 : Serial.println ("RTCWDT_CPU_RESET");break; /**<13, RTC Watch dog Reset CPU*/
60     case 14 : Serial.println ("EXT_CPU_RESET");break; /**<14, for APP CPU, reseted by PRO CPU*/
61     case 15 : Serial.println ("RTCWDT_BROWN_OUT_RESET");break;/**<15, Reset when the vdd voltage is not stable*/
62     case 16 : Serial.println ("RTCWDT_RTC_RESET");break; /**<16, RTC Watch dog reset digi. core and rtc module*/
63     default : Serial.println ("NO_MEAN");
64     }
65 }

```

Obrázek 73. Funkce pro vypsání důvodů pro resetování zařízení

```

79     esp_sleep_enable_ext0_wakeup(GPIO_NUM_13, HIGH);

```

Obrázek 74. Definování externího probuzení na pinu GPIO 13

Funkce **void Vlakynko1_fce(void * pvParameters)** - funkcí tohoto vlákna je po probuzení zařízení z deepsleepu vypsát do sériového terminálu důvod probuzení, tedy v mém případě to je externí probuzení. Poté se vypíší další hlášky do terminálu a po 10 vteřinách se zavolá funkce **void uspavani()**.

```

178 void Vlakynko1_fce (void * pvParameters)
179 {
180
181     while(1)
182     {
183         print_wakeup_reason();
184         Serial.println("vlakno bylo vytvořeno");
185         Serial.println("TEDKA POCKAM CHVILKU A PUJDU SPAT");
186         delay(10000);
187         uspavani();
188     }
189 }

```

Obrázek 75. Obsah vlákna

Funkce **void uspavani()** za zapíná režim deepsleep na desce ESP32-CAM. Před tím, než se zařízení uspí, bude vypsána hláška na sériový terminál, jež oznamuje spuštění deepsleep.

```

191 void uspavani()
192 {
193     Serial.println("jdu spat=====");
194     Serial.flush();
195     esp_deep_sleep_start();
196 }

```

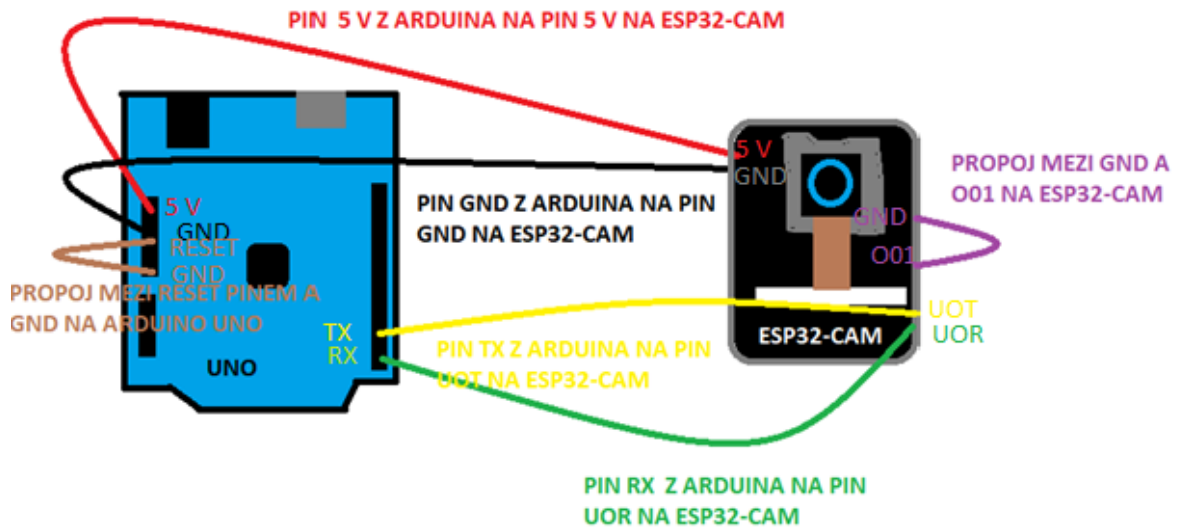
Obrázek 76. Funkce pro uspávání fotopasti

Uvedení fotopasti do chodu

Fotopast je uvedena do provozu o dost jednodušším způsobem než kamera, vzhledem k jejímu jednoduššímu programu a i fungování.

1. KROK – nahrání programu do ESP32-CAM

Nahrávání programu do fotopasti je komplikované z toho důvodu, že deska na sobě nemá převodník, proto jsem při cvičném zapojení na nepájivém poli byl donucen použít Arduino Uno jako převodník pro USB/UART, které bylo konstantně vyresetováno-viz obrázek níže.



Obrázek 77. Teoretický obrázek propojení Arduina Uno a ESP32-CAM pro účely programování (na obrázku jsou znázorněny piny pro propojení)

Zapojení Arduina Una a ESP32 – CAM je následující, jako první věc musíme zapojit napájení 5 V a GND, to vysvětlují červená a černá čára na obrázku, červená pro 5 V a černá pro GND. Druhou věcí je zapojit UART komunikaci mezi ESP32 – CAM a Arduino Uno, to docílíme spojením pinů TX z Arduina Una a UoT na ESP32-CAM pro vysílání dat a pro příjem dat dáme spoj mezi pinem RX z Arduina a UoR na ESP32 – CAM pro přijímání dat. Třetím krokem je propojení pinu RESET a GND na Arduinu a na ESP32-CAM. Pokud jsou tyto kroky splněny, začne se nahrávat kód z počítače do ESP32-CAM. Člověk by měl mít na paměti, že někdy je potřeba pro zahájení nahrávání, zmáčknout resetovací tlačítko na spodku desky pro zahájení nahrávání. Tento problém by se dal vyřešit připojením 10 μ F kondenzátoru na pin ENABLE, jenže tento pin není na tomto modelu mikroprocesoru vyveden.

Pokud jde tedy o tu variantu, kdy se nahrává program na mnou navrženou desku, je potřeba pouze dávat na pozor na dvě věci a to jsou: za prvé propoj mezi pinem GPIO01 a GND a za druhé, jestli je potřeba zmáčknout tlačítko RESET pro zahájení nahrávání. Na mnou navržené desce je už převodník USB/UART pro přímé připojení k počítači.

2. KROK – spuštění fotopasti

Druhým krokem je samotné spuštění fotopasti. Uživatel by měl mít na mysli, že po dokončení programu je nezbytné pro vytažení spoje mezi GND a GPIO1 na desce ESP32-CAM, jinak by se mikroprocesor pořád držel v režimu nahrávání. Jakmile je vypojený propoj mezi GND a GPIO1, zbývá poslední akt, a tím je zmáčknutí RESET tlačítka. To platí jak pro cvičné zapojení na nepájivém poli, tak i pro zapojení na DPS. Tím vlastně končí veškerá práce pro nahrávání a přes sériový terminál můžeme sledovat, co zařízení zrovna vypisuje.

```
12:41:27.309 -> rst:0x5 (DEEPSLEEP_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
12:41:27.355 -> configsip: 0, SPIWP:0xee
12:41:27.355 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
12:41:27.355 -> mode:DIO, clock div:1
12:41:27.355 -> load:0x3fff0030,len:1344
12:41:27.355 -> load:0x40078000,len:13924
12:41:27.355 -> ho 0 tail 12 room 4
12:41:27.355 -> load:0x40080400,len:3600
12:41:27.355 -> entry 0x400805f0
12:41:27.849 -> Wakeup caused by external signal using RTC_IO
```

Obrázek 78. Výpis důvodu probuzení a spuštění fotopasti

3. KROK – první fotografie na fotopasti

Třetím krokem je pořízení samotné fotografie. V minulém kroku jsme si provedli restart po nahrávání kódu, což zapříčinilo první spuštění programu, a tedy to provedlo i první fotografii z fotopasti, následně její uložení a následně se celé zařízení do 10 vteřin uspalo do režimu deepsleep. V tomto režimu bude čekat, dokud jej někdo neprobudí spuštěním PIR senzoru.

```
12:41:27.309 -> rst:0x5 (DEEPSLEEP_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
12:41:27.355 -> configsip: 0, SPIWP:0xee
12:41:27.355 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
12:41:27.355 -> mode:DIO, clock div:1
12:41:27.355 -> load:0x3fff0030,len:1344
12:41:27.355 -> load:0x40078000,len:13924
12:41:27.355 -> ho 0 tail 12 room 4
12:41:27.355 -> load:0x40080400,len:3600
12:41:27.355 -> entry 0x400805f0
12:41:27.849 -> Wakeup caused by external signal using RTC_IO
12:41:27.849 -> vlakno bylo vytvořeno
12:41:27.849 -> TEDKA POCKAM CHVILKU A PUJDU SPAT
12:41:28.220 -> pořídil jsem obrázek
12:41:28.220 -> Picture file name: /picture60.jpg
12:41:28.600 -> Saved file to path: /picture60.jpg
12:41:37.829 -> jdu spat=====
12:41:37.829 -> ets Jun 8 2016 00:22:57
```

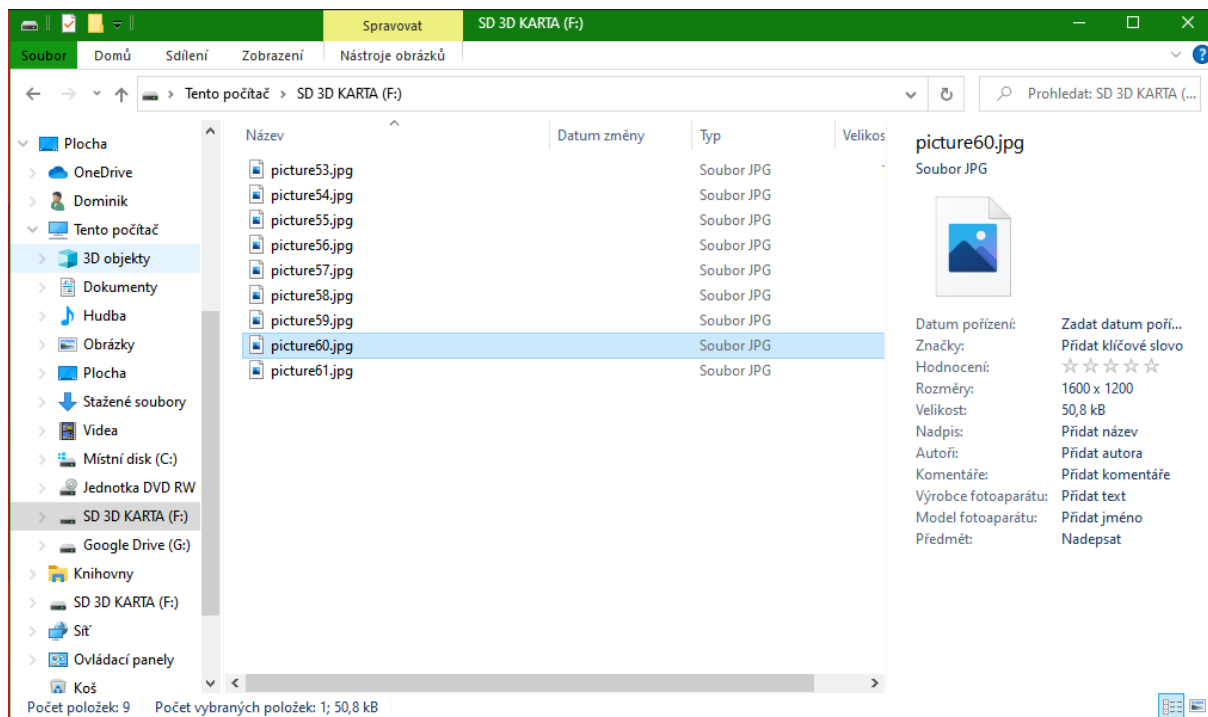
Obrázek 79. Výpis sériového terminálu od fotopasti

Na obrázku 71. je ukázka výpisu sériového terminálu pro komunikaci s fotopastí. Zařízení bylo probuzeno pomocí externího probuzení, následně spustilo vlákno pro hlídání času na uspání a vypsalo hlášky pro komunikaci s uživatelem. Poté zařízení pořídilo obrázek, přidělilo mu jméno s číslem a uložilo obrázek na microSD kartu. Jako poslední akci provedlo vypsání hlášky o uspání a zařízení přešlo do režimu deepsleep. Opět z režimu deepsleep je probuzeno externím probuzením. Fotopast je aktivní po dobu 10 vteřin. V těchto 10 vteřinách fotopast pořídí snímek, přidělí mu jméno i s pořadovým číslem a po uplynutí doby 10 vteřin je zařízení uspáno.

Fotografie jsou pořizovány s velikostí desítek kB až do 150 kB s rozměrem fotografie 1600 x 1200. Čas od pořízení fotografie až po její uložení na kartu se pohybuje 380 milisekund.

4. KROK – načtení microSD karty do počítače pro prohlížení fotografií

Čtvrtým a posledním krokem je vyjmutí microSD karty z ESP32-CAM a následné prohlížení v počítači. Obrázky jsou ukládány ve formátu JPG, můžeme si je tedy prohlédnout, jak na počítači, v mobilním telefonu, anebo na chytré televizi. Někdy je dobré mít čtečku na SD karty pro načtení karty v počítači.



Obrázek 80. Načtená microSD karta s uloženými fotografiemi



Obrázek 81. Obrázky pořízené fotopastí

3. Naměřené hodnoty a budoucí vylepšení obou zařízení

Ve třetí kapitole se budu věnovat naměřeným hodnotám, které jsem získal během měření na obou zařízeních při jejich zhotovování. Dále se budu věnovat možným vylepšením obou zařízení.

3.1 Naměřené hodnoty

Provedl jsem několik měření na kameře za účelem zjistit odběr proudu při různých režimech, v nichž se zařízení nacházelo, měření odběru proudu pro různá rozlišení obrazu a následně některé z nich porovnal s údaji od výrobce. Na fotopasti jsem měřil rychlost pro uložení obrazu a odběr proudu při různých režimech jako u kamery a opět provedl porovnání. Výsledky jsou uvedeny v tabulce níže.

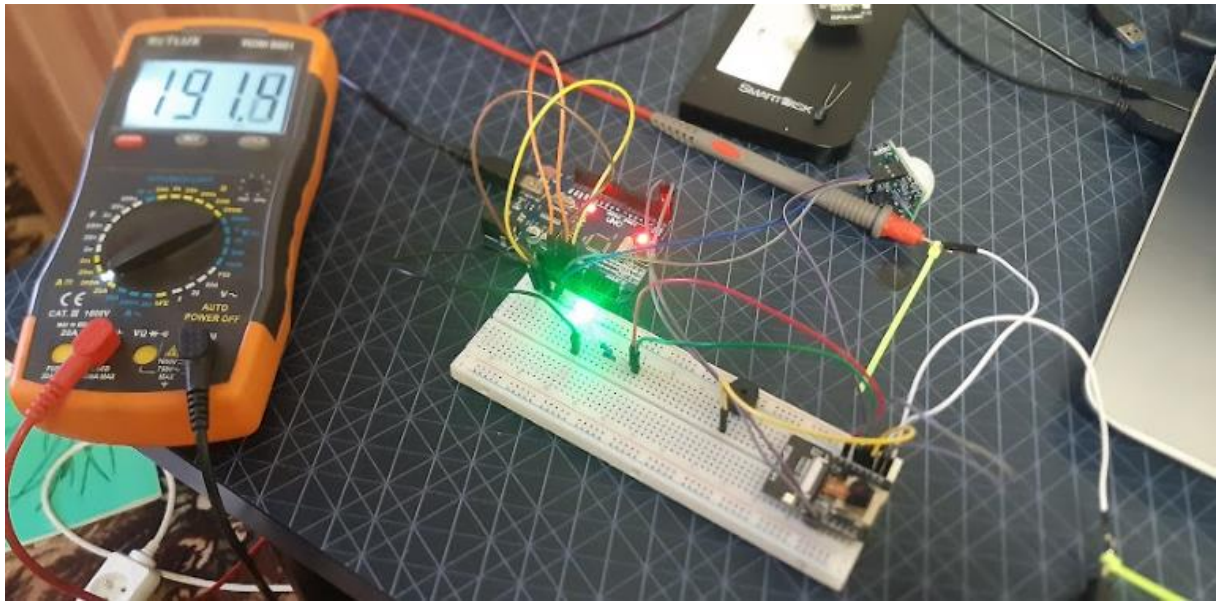
| Měření odběru proudu na kameře z ESP32-CAM | | |
|--|-----------------------|----------------------|
| stav zařízení při měření | naměřené hodnoty v mA | hodnoty v datasheetu |
| v deepsleep režimu | 6,04 mA | 6 mA |
| před zahájením vysílání | 80 mA- 120 mA | 80 mA |
| Vysílání v rozlišení QVGA | 90 mA - 170 mA | není uvedeno |
| Vysílání v rozlišení VGA | 180 mA | není uvedeno |
| Vysílání v rozlišení HD | 190 mA - 200 mA | není uvedeno |

Tabulka 6. Měření odběru proudu na kameře z ESP32-CAM

V datasheetech byl pouze uveden odběr proudu při vypnuté vestavěné led diodě, a to 180 mA, a druhou informaci byl odběr při zapnuté vestavěné led diodě, hodnota by měla odpovídat 270 mA.

Žádný z procházených datasheetů neměl uvedené hodnoty o odběru proudu při různých rozlišeních obrazu. [41] [43] [53] [54]

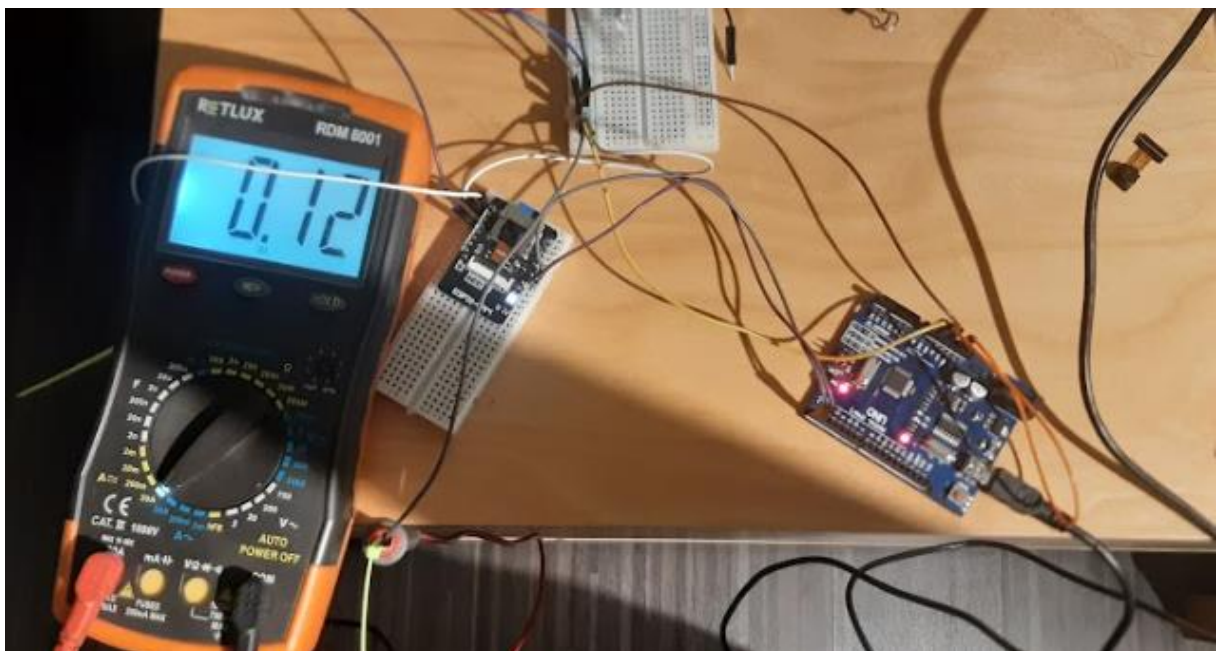
Hodnoty odběru proudu při různých rozlišení, se držely ve vypsáných mezích, jež jsem uvedl v tabulce, a nijak se neměnily. Rozlišení QVGA mělo chvilkově velké skoky odběru proudu v uvedených mezích, ale nikdy je nepřekročilo. Při rozlišení VGA byla hodnota konstantních 180 mA bez velkých skokových změn. Co se týče posledního měřeného rozlišení HD, zaznamenal jsem po dobu 5 vteřin skok na 200 mA, jenž následně klesl na 191,4 mA, jak je vidět na obrázku 82.



Obrázek 82. Měření odběru proudu na cvičném zapojení kamery

| Měření odběru proudu na fotopasti z ESP32-CAM | | |
|---|-------------------------|----------------------|
| stav zařízení při měření | naměřené hodnoty v mA | hodnoty v datasheetu |
| v režimu deepsleep | 2,9 mA - 3,2 mA -8,9 mA | 6 mA |
| při pořízení snímku | 160 mA - 170 mA | 180 mA |
| po pořízení snímku | 112 mA - 133,3 mA | není uvedeno |

Obrázek 83. Měření odběru proudu na fotopasti z ESP32-CAM



Obrázek 84. Měření odběru proudu na cvičném zapojení fotopasti

Pro druhé zařízení se mi povedlo dohledat aspoň více než polovinu údajů pro porovnání údajů z datasheetu a mých výsledků.

Při měření na fotopasti jsem zaznamenal problém při nastavení rozsahu ampérmetru. Byla-li zvolena pozice pro měření až maximálních 20 A, měření probíhalo bez problémů. Jakmile jsem přenastavil rozsah ampérmetru na 200mA, fotopast měla časté problémy se vůbec spustit anebo nechtěla uložit data na microSD kartu.

Měření času, během kterého jsem pořídil fotografii a uložil ji na kartu, jsem realizoval pomocí vypisování do sériového terminálu.

Provedl jsem dva různé způsoby měření času za účelem zjistit, za jak dlouho od probuzení PIR senzorem se pořídí obrázek, a druhé měření bylo zaměřeno na čas od pořízení obrázku až po jeho uložení dat na microSD kartu.

```
12:41:27.309 -> rst:0x5 (DEEPSLEEP_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
12:41:27.355 -> configsip: 0, SPIWP:0xee
12:41:27.355 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
12:41:27.355 -> mode:DIO, clock div:1
12:41:27.355 -> load:0x3fff0030,len:1344
12:41:27.355 -> load:0x40078000,len:13924
12:41:27.355 -> ho 0 tail 12 room 4
12:41:27.355 -> load:0x40080400,len:3600
12:41:27.355 -> entry 0x400805f0
12:41:27.849 -> Wakeup caused by external signal using RTC_IO
12:41:27.849 -> vlakno bylo vytvořeno
12:41:27.849 -> TEDKA POCKAM CHVILKU A PUJDU SPAT
12:41:28.220 -> pořídil jsem obrázek
12:41:28.220 -> Picture file name: /picture60.jpg
12:41:28.600 -> Saved file to path: /picture60.jpg
12:41:37.829 -> jdu spat=====
12:41:37.829 -> ets Jun 8 2016 00:22:57
```

Obrázek 85. Měření času pro fotopast

Zeleně označená hláška

„rst:0x5 (DEEPSLEEP_RESET), boot:0x13 (SPI_FAST_FLASH_BOOT)“ oznamuje probuzení zařízení z režimu deepsleep. Jedná se tedy o moment probuzení PIR v čase 12:41:27.309

Žlutě označená hláška „pořídil jsem obrázek“ oznamuje pořízení obrázku v čase 12:41:28.220

Červeně zakroužkovaná hláška „Saved file path to: /picture60.jpg“ oznamuje uložení obrázku na kartu v čase 12:41:28.600

Z čehož vyplývá, že časový rozdíl červeného a žlutého času je přesně 380 milisekund při velikosti obrázku v 50,8kB. Druhé měření ukazuje, že fotografii dokáže od probuzení pořídít do 911 ms, což je necelá 1 vteřina.

Dále jsem přidal tabulku pro ukázkou dalších obrázků a jejich času pro uložení

| měření času pro fotopast | | |
|--------------------------------|--|--|
| velikost obrázků v kB | čas pro uložení fotografie na kartu | čas od probuzení PIR senzorem do pořízení obrázku |
| obrázek o velikosti 50,8 kB | 380 ms | 911 ms |
| obrázek o velikosti 69 kB | 358 ms | 890 ms |
| obrázek o velikosti 96,9 kB | 618 ms | 913 ms |

Tabulka 7. Měření času pro fotopast

Z tabulky můžeme vidět, že pokud je obrázek veliký do 70 kB, je jeho čas pro uložení okolo 400 ms, blíží-li se velikostně ke 100 kB, čas jeho uložení se zdvojnásobí. Obrázky jsou vždy při rozměrech 1600 x 1200. Dále z tabulky jde vyčíst, že čas od probuzení do pořízení fotografie je pokaždé okolo 900 ms.

3.2 Budoucí vylepšení obou zařízení

Jak jsem již zmiňoval, pro kameru bych do budoucna rozhodně zvážil přidání vstupních periférií jako je mikrofón a teploměr, a výstupních periférií, jako je piezo buzák, servomotor 360 ° pro otáčení a zvětšení zorného pole. Tyto vyjmenované periférie, beru jako vedlejší, a ne tolik potřebné jako výměnu kamerového modulu OV2640 za OV5640 pro zlepšení obrazu z 2 megapixelů na 5 megapixelů. To by bylo první hlavní vylepšení, druhé hlavní vylepšení by byla možnost sledovat přenos z kamery z venku lokální sítě, tedy použití buď to RTSP protokolu a zakoupení serveru, na který bych obraz mohl odesílat, a já jsem se poté připojoval pouze na tento můj server. Ovšem přichází zde v potaz ještě možnost nastavení Home Assistant služby, která by celý problém řešila za mě, což by znamenalo pořízení Raspberry Pi 4, jenže je problém s náročným sehnáním tohoto produktu. Ještě bych zvažoval přidání záložních zdrojů na kameru, Raspberry Pi a WiFi router, protože může nastat nečekaný výpadek distributorské sítě elektřiny u monitorovanému objektu, a tím se naskytne možnost pro zloděje bez identifikace v klidu vykrást objekt hlídáný kamerami. Úložný prostor pro kameru není potřeba měnit, protože bude kamera používána ve vnitřních prostorách, ovšem pokud bych kameru umístoval ven, předělal bych krabici stejně jako u fotopasti za účelem ochrany proti venkovním vlivům.

Vylepšení fotopasti by se týkalo v oblastech připojení na mobilní síť a posílání obrázku na server nebo Google disk. Osobně bych zvolil Google disk pro uživatelský komfort. Bylo by tedy zapotřebí zřejmě změnit desku pro mikroprocesor ESP32. To by zahrnovalo přidání nových periférií, jako jsou slot na SIM kartu, teploměr pro monitorování teploty v okolí, slot na kamerový modul, který bych určitě vyměnil místo OV2640 za OV5640. Všechna tato data bych posílal pomocí SIM karty na svůj osobní Google disk. Moje představa je taková, že by na každé fotografii byla vypsána data o teplotě v okolí a čas pořízení fotografie. Vylepšil bych i krabici, do které je zařízení uloženo, jelikož vím, že krabička vytištěná na 3D tiskárně a nijak chráněná proti vlhkosti není zrovna dobrý způsob pro sledování zvěře v lese, proto bych tedy přidal vylepšení materiálu v podobě nějakého těsnění nebo jiného řešení, jež se naskytne. Bohužel v současné době nemůžu zvažovat posílání dat po mobilní síti, jelikož u mě ve vesnici není dobrý mobilní signál.

ZÁVĚR

Cílem práce bylo ověřit možnost postavit z dostupných komponent řešení webové kamery, případně fotopasti a ověřit funkčnost takového řešení. V teoretické práci jsem se zabýval teorií nutnou pro správnou volbu kombinace komponent a přístupu k řešení. V praktické části práce jsem pak navrhl kameru na základě ESP32 + kamera combo kitu. Řešení bylo doplněno o senzor pohybu (PIR senzor), který je schopný detekovat pohyb při velmi nízké spotřebě v porovnání detekce pohybu za použití detekce v obraze. Kamera bylo ověřena ve dvou režimech, a to klasická bezpečnostní webkamera, připojená přes vestavěné Wi-Fi, která navíc odesílá informace o detekci pohybu formou WhatsApp zpráv na telefon. Druhý režim je režim fotopasti, kdy kamera nevyžaduje připojení k síti a je napájena z baterií. Pro úsporu energie je trvale v režimu hlubokého spánku, dokud není detekován pohyb, poté se sejme obrázek, uloží na SD kartu a kamera opět usíná.

Obě navržená řešení se ukázala jako funkční. Webkameru běžící za NATem je potřeba zpřístupnit nějakou formou tunelu (např. VPN), pokud k ní chceme přistupovat mimo domácí síť, díky hlášení událostí pomocí zpráv na telefon umožňuje spolehlivou kontrolu hlídané oblasti. U fotopasti jsou kritické dva parametry, a to spotřeba, a tedy teoretická výdrž, tu mám díky spotřebě cca 170 mA a kapacitě dvou baterií 1800 mAh teoreticky 2 dny. Druhým parametrem je prodleva mezi detekcí pohybu PIR a faktickým sejmutím snímku, zde se pohybují okolo 0,9 sec od probuzení po získání snímku, což snese srovnání s low end profesionálními fotopastmi.

Praktickým testováním se povedlo ověřit funkčnost řešení, tedy splnit cíle práce, ale zároveň odhalilo některé problémy, pokud by mělo dojít k laickému nasazení, jako například zviditelnění zařízení v domácí síti.

POUŽITÁ LITERATURA

Bibliografie

[1 V. Hotař, „Senzorika pro vizualizaci,“ TUL, Liberec, 2019.
]

[2 K. Burda, Základy elektronických zabezpečovacích systémů, Brno: Akademické
] nakladatelství CERM, 2017.

[3 A. Guldan, Mikroelektronické senzory, Bratislava: Vydavateľstvo technickej a
] ekonomickej literatúry Bratislava, 1987.

[4 P. Rozsival, „Aktivní senzory,“ UPCE, Pardubice, 2022.
]

[5 Z. K. a. kolektiv, Roboty a robotizované výrobní technologie, Brno : Nakladatelství
] VUTIUM, 2016.

[6 „IT SLOVNÍK,“ © 2008 - 2022 IT-Slovník.cz team., [Online]. Available: [https://it-
\] slovník.cz/pojem/rozhrani](https://it-slovník.cz/pojem/rozhrani). [Přístup získán 21 2 2023].

[7 M. Lemmon, „What is a serial interface?,“ [Online]. Available:
] <https://www3.nd.edu/~lemmon/courses/ee224/web-manual/web-manual/lab9/node4.html>.
[Přístup získán 27 2 23].

[8 M. G. L. Eric Peña, „Analogue Dialogue,“ 12 2020. [Online]. Available:
] [https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-
protocol.html](https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html). [Přístup získán 27 2 23].

[9 P. Dhaker, „Analog Dialogue,“ 9 2018. [Online]. Available:
] [https://www.analog.com/en/analog-dialogue/articles/introduction-to-spi-
interface.html#:~:text=SPI%20is%20a%20synchronous%2C%20full,%2Dwire%20or%20
4%2Dwire..](https://www.analog.com/en/analog-dialogue/articles/introduction-to-spi-interface.html#:~:text=SPI%20is%20a%20synchronous%2C%20full,%2Dwire%20or%204%2Dwire..) [Přístup získán 27 2 23].

[1 OmniVision, „Serial Camera Control Bus Functional Specification,“ OmniVision,
0] Sunnyvale, CA USA, 2003.

[1 E. P. Mary Grace Legaspi, „I2C Communication Protocol:Understanding I2C
1] Primer,PMBus, and SMBus,“ Analog Dialogue, 2021.

[1 A. Vojáček, „hw-server.com,“ 15 4 2007. [Online]. Available: [https://hw-
2\] server.com/parallel-port-lpt-ieee-1284](https://hw-server.com/parallel-port-lpt-ieee-1284). [Přístup získán 27 3 23].

- [1 A. Raquel, „Recoverit,“ [Online]. Available: <https://recoverit.wondershare.com/hard-3-drive/what-is-pata.html>. [Přístup získán 27 3 23].
- [1 M. C. Ryan Perian, „LifeWire tech for humans,“ 17 10 2021. [Online]. Available: 4] <https://www.lifewire.com/what-is-pci-2640242>. [Přístup získán 27 3 2023].
- [1 P. Tišnovský, „ROOT.CZ,“ 16 10 2008. [Online]. Available: 5] <https://www.root.cz/clanky/podrobnejsi-popis-sbernice-pci/>. [Přístup získán 27 3 2023].
- [1 R. W. World, „Advantages of PCI bus | Disadvantages of PCI bus,“ [Online]. Available: 6] <https://www.rfwireless-world.com/Terminology/Advantages-and-Disadvantages-of-PCI-bus.html>. [Přístup získán 29 3 2023].
- [1 N. Semiconductor, „Camera Link,“ National Semiconductor, 2000. 7]
- [1 R. Kayne, „EasyTechJunkie,“ 22 2 2023. [Online]. Available: 8] <https://www.easytechjunkie.com/what-is-composite-video.htm>. [Přístup získán 27 3 2023].
- [1 Camius, „Camius security redefined,“ 2023. [Online]. Available: 9] <https://www.camius.com/cvbs-ahd-hd-tvi-hd-cvi-analog-camera-connection/>. [Přístup získán 27 3 2023].
- [2 „Delta.eu,“ [Online]. Available: https://shopdelta.eu/cvbs-video-signal-transmission-0-standard_12_aid771.html. [Přístup získán 7 3 2023].
- [2 DATAPRO, „Datapro,“ [Online]. Available: 1] https://www.datapro.net/techinfo/dvi_info.html. [Přístup získán 27 3 2023].
- [2 C. Hope, „Computer Hope,“ 12 4 2021. [Online]. Available: 2] <https://www.computerhope.com/jargon/d/dvi.htm>. [Přístup získán 27 3 2023].
- [2 Spo-Comm, „spo-comm,“ 21 10 2019. [Online]. Available: <https://www.spo-3-comm.de/en/blog/know-how/nice-to-know-what-is-dvi>. [Přístup získán 27 3 2023].
- [2 SmartDV, „SmartDV,“ [Online]. Available: <https://www.smart-dv.com/vip/cpi.html>. 4] [Přístup získán 27 3 2023].
- [2 FTDI, „What is the Camera Parallel Interface,“ FTDI, 23.3.2015. 5]
- [2 OmniVision, „OV2640 Color CMOS UXGA,“ OmniVision, Sunnyvale, CA USA, 2006. 6]
- [2 L. kit, „UXGA CMOS kamera TY-OV2640 V2.0 2MP pro Ai-Thinker ESP32-CAM,“ 7] [Online]. Available: <https://www.laskakit.cz/kamera-ty-ov2640-v2-0-2mp-pro-ai-thinker-esp32-cam/?gclid=Cj0KCQjww4-hBhCtARIsAC9gR3bF->

yxAVwLAW8uJAgGpnVnr8mkaCYOWvvEedXiVeJriXXkfoYGIFwksaAmK3EALw_wcB. [Přístup získán 29 3 2023].

[2 laskakit, „CMOS VGA kamera OV7670 640x480 bez paměti,“ [Online]. Available: 8] <https://www.laskakit.cz/cmos-vga-kamera-ov7670-640x480--bez-pameti/#relatedFiles>. [Přístup získán 27 3 2023].

[2 OmniVision, „OV7670/OV7171 CMOS VGA (640x480),“ OmniVision, 2006. 9]

[3 L. Guangzhou Putal Communication Technology Co., „PTC08 Serial Camera Module 0] Specification“.

[3 Adafruit, „TTL Serial JPEG Camera with NTSC Video,“ [Online]. Available: 1] <https://www.adafruit.com/product/397>. [Přístup získán 27 3 2023].

[3 BOTLAND, „BOTLAND ArduCam-Mini OV5642 5MPx,“ [Online]. Available: 2] <https://botland.cz/fotoaparaty-pro-arduino-a-raspberry-pi/6558-arducam-mini-ov5642-5mpx-2592x1944px-120fps-spi-kamerovy-modul-pro-arduino-uno-mega2560-raspberry-pi-pico-5904422375133.html>. [Přístup získán 27 3 2023].

[3 Arducam, „Arduino Camera: SPI Camera Module from Arducam,“ [Online]. Available: 3] <https://www.arducam.com/spi-arduino-camera/>. [Přístup získán 27 3 2023].

[3 Arducam, „ArduCAM-M-5MP Camera Shield,“ Arducam, 2015. 4]

[3 C. Ltd., „C328-7640 JPEG Camera Module W/ UART Interface,“ Nan Fung Ctr, Castle 5] Peak Rd, Tsuen Wan, NT., Hong Kong..

[3 Arduino, „Arduino Mega 2560 Rev3,“ [Online]. Available: 6] <https://store.arduino.cc/products/arduino-mega-2560-rev3>. [Přístup získán 27 3 2023].

[3 A. Team, „Machine vision with low-cost camera modules,“ 24 6 2020. [Online]. 7] Available: <https://blog.arduino.cc/2020/06/24/machine-vision-with-low-cost-camera-modules/>. [Přístup získán 27 3 2023].

[3 Arudino, „Arduino Uno Rev3,“ [Online]. Available: 8] <https://store.arduino.cc/products/arduino-uno-rev3>. [Přístup získán 27 3 2023].

[3 Arduino, „Arduino Nano,“ [Online]. Available: [https://store.arduino.cc/products/arduino-](https://store.arduino.cc/products/arduino-9] nano) nano. [Přístup získán 27 3 2023].

[4 M. Hruba, „POPIS NodeMCU - NodeMcu Lua WI-FI ESP8266 (ESP-12E WIFI) 0] CP2102,“ 2019. [Online]. Available: <http://nodemcu.tode.cz/popis-nodemcu/>. [Přístup získán 27 3 2023].

[4 DFROBOT, „ESP32-CAM Development Board,“ DFROBOT, 2019. 1]

- [4 H. T. MECHATRONICS, „Interfacing 5MP SPI Camera with NodeMCU ESP8266,“ 25
2] 12 22. [Online]. Available: <https://how2electronics.com/interfacing-5mp-spi-camera-with-nodemcu-esp8266/>. [Přístup získán 27 3 2023].
- [4 E. Systems, „ESP32-WROOM-32 Datasheet,“ Espressif Systems, 2019.
3]
- [4 R. Pi, „Raspberry Pi hardware,“ [Online]. Available:
4] <https://www.raspberrypi.com/documentation/computers/raspberry-pi.html>.
- [4 Laskakit, „Raspberry Pi Pico,“ [Online]. Available: <https://www.laskakit.cz/raspberry-pi-5-pico/>.
- [4 RPiShop, „Raspberry Pi 4 Model B - 4GB RAM,“ [Online]. Available:
6] https://rpishop.cz/raspberry-pi-4/1598-raspberry-pi-4-model-b-4gb-ram-765756931182.html?utm_source=seznam&utm_medium=cpc&utm_campaign=Zbo%c5%be%c3%ad.cz:+RPiShop.cz&utm_content=V%c5%a1echny+produkty&utm_term=.
- [4 R. P. Foundation, „Getting started with the Camera Module,“ [Online]. Available:
7] <https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/1>.
- [4 L. Pounder, „Raspberry Pi Pico Review: ‘Pi Silicon’ Debuts on \$4 Microcontroller,“ 22 1
8] 2021. [Online]. Available: <https://www.tomshardware.com/reviews/raspberry-pi-pico-review>.
- [4 A. SILICIA, „STM32F7 camera development kit,“ [Online]. Available:
9] <https://www.avnet.com/wps/portal/silica/products/product-highlights/2016/ev-stm32f7-camera-development-kit/>.
- [5 STlife.augmented, „STM32F722IC Datasheet,“ 4 2020. [Online]. Available:
0] <https://html.alldatasheet.com/html-pdf/1246911/STMICROELECTRONICS/STM32F722IC/2271/1/STM32F722IC.html>.
- [5 S. Labs, „SINGLE-CHIP USB-TO-UART BRIDGE CP2102/9,“ 2017.
1]
- [5 R. S. Sara Santos, „ESP32-CAM Take Photo and Save to MicroSD Card,“ [Online].
2] Available: <https://randomnerdtutorials.com/esp32-cam-take-photo-save-microsd-card/>.
[Přístup získán 8 4 2023].

PŘÍLOHA A – elektronické přílohy

- Text práce: TomasekD_MinimalistickaKamera_PR_2023.pdf
- Projekt EAGLE se schématem zapojení a návrhem desky plošných spojů
- Projekt Arduino. Ide s kódy pro kameru a fotopast
- Projekt FreeCad obsahující návrhy úložné boxy pro zařízení