

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV RADIOELEKTRONIKY

DEPARTMENT OF RADIO ELECTRONICS

KAMEROVÝ ZABEZPEČOVACÍ SYSTÉM S MINIPOČÍTAČEM RASPBERRY PI

CAMERA SECURITY SYSTEM WITH RASPBERRY PI MINICOMPUTER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Marek Vitula

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Jaromír Kolouch, CSc.

BRNO 2018

Bakalářská práce

bakalářský studijní obor **Elektronika a sdělovací technika**

Ústav radioelektroniky

Student: Marek Vitula

ID: 186596

Ročník: 3

Akademický rok: 2017/18

NÁZEV TÉMATU:

Kamerový zabezpečovací systém s minipočítačem Raspberry Pi

POKYNY PRO VYPRACOVÁNÍ:

Prozkoumejte možná řešení a vypracujte rámcový návrh kamerového zabezpečovacího systému založeného na využití minipočítače Raspberry Pi. Program běžící v minipočítači bude detekovat pohyb na scéně, ukládat fotografie na lokální úložiště a odesílat je na externí úložiště, kde je bude možno zobrazovat. Pro odesílání fotografií zvažte použití GSM modulu SIM900, případně navrhnete jiný vhodný modul s tím, že zvolené řešení zdůvodníte. Systém musí být schopen pořídit okamžitou fotografii na vyžádání, odeslat ji na server a e-mailem ji pak odeslat na specifikovanou adresu včetně údajů o stavu baterie a dobíjení. Dálkové řízení zabezpečovacího systému má být umožněno také pomocí SMS. Navržený systém má být energeticky autonomní s možností napájení ze solárního panelu, z čehož plyne požadavek na co nejnižší spotřebu systému. Vypracujte dále podrobný návrh části systému v rozsahu stanoveném konzultantem a vedoucím projektu. Minimálním požadavkem bude sestava minipočítače doplněného vhodným paměťovým systémem místo SD karty, kamery a externího úložiště s možností zobrazení včetně potřebného programového vybavení.

Navrženou sestavu realizujte a ověřte její funkčnost.

DOPORUČENÁ LITERATURA:

[1] MANN, B. C pro mikrokontroléry. Praha: BEN, 2003.

[2] KOSEK, J. PHP – tvorba interaktivních internetových aplikací. Praha: Grada, 1999.

Termín zadání: 5.2.2018

Termín odevzdání: 24.5.2018

Vedoucí práce: doc. Ing. Jaromír Kolouch, CSc.

Konzultant: Ing. Ondřej Pavelka, Honeywell, spol. s r.o. – HTS CZ o.z.

prof. Ing. Tomáš Kratochvíl, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato bakalářská práce se zabývá návrhem kamerového zabezpečovacího systému na platformě Raspberry Pi, který může být vzdáleně ovládán a napájen solárním panelem. Součástí je popis komponent a také návrh aplikace umožňující detekci pohybu ze scény a také pomocí PIR senzoru, následné zpracování fotografií a odeslání do cloudového úložiště.

KLÍČOVÁ SLOVA

Raspberry Pi, kamera, počítačové vidění, detekce pohybu, solární napájení, Python, Linux

ABSTRACT

This bachelor thesis describes a design of a camera surveillance system on the Raspberry Pi platform that can be controlled remotely and powered by a solar panel. It includes a description of the individual components to be used and also deals with the design of the application for motion detection using a camera and PIR motion sensor output and the following image processing and storage on cloud.

KEYWORDS

Raspberry Pi, camera, computer vision, motion detection, solar powered, Python, Linux

VITULA, Marek. *Kamerový zabezpečovací systém s minipočítačem Raspberry Pi*. Brno, 2018, 65 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky. Vedoucí práce: doc. Ing. Jaromír Kolouch, CSc.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Kamerový zabezpečovací systém s minipočítačem Raspberry Pi“ jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského 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, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu doc. Ing. Jaromírovi Kolouchovi, CSc. a odbornému konzultantovi panu Ing. Ondřejovi Pavelkovi za odborné vedení, trpělivost při konzultacích a jejich podnětné návrhy k mé práci.

Brno

.....

podpis autora(-ky)

OBSAH

Úvod	10
1 Teoretická část práce	11
1.1 Hardwarové vybavení	11
1.2 Programové vybavení	17
1.3 Detekce pohybu	18
1.4 Mobilní sítě	21
1.5 Solární napájení	21
2 Návrh aplikace	24
2.1 Architektura aplikace	24
2.2 Detekce pohybu	26
2.3 Řízení snímání fotografií	28
2.4 Detekce pohybu z PIR pomocí obsluhy přerušení	31
2.5 Ovládání aplikace	32
2.6 Konfigurace	34
2.7 Sledování stavu nabíjení	37
3 Návrh zapojení	38
3.1 Prototypování	38
3.2 Řešení napájení	38
3.3 Komunikace s GSM modulem SIM800	42
3.4 Blokové schéma návrhu zařízení	44
3.5 Seznam a cena komponent	45
4 Realizace a testování	46
4.1 Měření spotřeby zařízení	47
4.2 Zdrojový kód aplikace	47
5 Závěr	48
5.1 Možná vylepšení	48
Literatura	49
Seznam symbolů, veličin a zkratk	50
Seznam příloh	51
A Zapojení GPIO pinů na Raspberry Pi	52

B	Denní úhrn solárního záření v Brně	53
C	Ukázkové fotografie pořízené kamerou	54
D	Dokumentace aplikace	55

SEZNAM OBRÁZKŮ

1.1	Raspberry Pi Zero W, převzato z thepihut.com	11
1.2	Raspberry Pi GPIO Pinout, převzato z pinout.xyz	12
1.3	Raspberry Pi NoIR Camera V2, převzato z reichelt.de	13
1.4	Aproximovaná spektrální citlivost CMOS snímače Sony IMX219. Autorem je Koen Hufkenssem [7]	14
1.5	Schéma PIR modulu	16
1.6	Globální horizontální záření [6]	22
2.1	Architektura aplikace	24
2.2	Hledání vektorů pohybu	27
2.3	Informační panel aplikace	33
2.4	Panel aplikace obsahující log	34
2.5	Rozhraní pro konfiguraci	37
3.1	Prototyp zařízení sestavený na nepájivém poli	38
3.2	Schéma napájení RPi přes USB, zdroj raspberrypi.org	39
3.3	Schéma spínače výkonové LED	40
3.4	Schéma multiplexovacího protokolu GSM0710	43
3.5	Blokové schéma návrhu zařízení	44
4.1	Realizace kamerového zabezpečovacího systému	46
B.1	Solar Irradiance Brno [8]	53
C.1	Ukázka denní fotografie pořízené kamerou	54
C.2	Ukázka noční fotografie pořízené kamerou s IR přísvitem	54

SEZNAM TABULEK

1.1	Možnosti modulu HC-SR501	16
1.2	Parametry LED z katalogu	16
2.1	Příkazy SMS	33
2.2	Parametry konfiguračního souboru	36
3.1	Proudový odběr komponent	39
3.2	Seznam komponent	45
4.1	Měření spotřeby	47
A.1	Zapojení GPIO pinů na Raspberry Pi	52

ÚVOD

Tento dokument popisuje semestrální práci vytvořenou na VUT, fakultě elektrotechniky, v Brně.

Po krátkém úvodu a seznámení se zadáním práce následuje kapitola číslo 1, ve které je popsán minipočítač Raspberry Pi, na kterém bude práce dále stavět. Dále je popsán operační systém Raspbian a další programové vybavení, které bude použito. Je zde teoreticky popsán způsob detekce pohybu a také je teoreticky objasněn problém solárního napájení.

Kapitola č. 2 popisuje návrh aplikace pro detekci pohybu.

Návrh řešení se nachází v kapitole č. 3 a popis samotné realizace projektu je v kapitole č. 4.

Poslední kapitola krátce hodnotí celou práci a navrhuje případná zlepšení.

Úvod do problematiky

Práce si klade za cíl vytvořit zařízení, které bude schopné sloužit jako monitorovací systém, například pro ostrahu objektu, nebo pro monitorování zvěře.

Ve srovnání s komerčně dostupnými, kamerovými zabezpečovacími systémy, bude navrhovaný systém, vytvořený na platformě Raspberry Pi a Linux, disponovat větší variabilitou.

Monitorovací systém je navržen tak, aby mohl být po nějakou dobu energeticky soběstačný. Problém lze řešit napájením pomocí solárního panelu, kterým bude dobíjen akumulátor.

Systém by tedy měl být energeticky úsporný. Bude nutné vybrat vhodnou, spolehlivou metodu detekce obrazu, která by nevyžadovala pro svoji činnost příliš mnoho elektrické energie.

Konektivita monitorovacího systému do sítě Internet bude zajištěna pomocí přídatného modulu.

1 TEORETICKÁ ČÁST PRÁCE

1.1 Hardwarové vybavení

Raspberry Pi

Raspberry Pi je název malého, jednodeskového počítače vyvinutého britskou nadací Raspberry Pi Foundation v roce 2012 s cílem podpořit výuku informatiky a vytvořit počítač dostupný pro všechny. V roce 2017 byla dostupná již třetí generace tohoto minipočítače a také zmenšená verze Raspberry Pi Zero a Zero W.

Primárním operačním systémem je Raspbian, což je derivát Linuxu Debianu, avšak je možné použít i řadu jiných operačních systémů, například Windows IoT Core nebo RISC OS.[1]

Raspberry Pi Zero W

Základem Raspberry Pi Zero W je SoC (System on a chip) Broadcom BCM2835 založený na 32 bitové architektuře ARMv6. Taktovací frekvence procesoru je 1 GHz.

Počítač je vybaven 512 MB paměti typu SDRAM a grafickým procesorem Broadcom Videocore IV. Pro úložiště dat jsou použity microSD karty.

Doporučená cena v době uvedení na trh byla 10 USD.[1]



Obr. 1.1: Raspberry Pi Zero W, převzato z thepihut.com

Napájení a spotřeba

Zařízení je napájeno z micro USB portu napájecím napětím 5V. Dle technické dokumentace se proudový odběr ve stavu nečinnosti pohybuje kolem 100 mA. Maximální odběr zařízení včetně zapojené klávesnice, myši a obrazovky může být až 350 mA.

Nicméně lze různými postupy snížit proudový odběr až na 80 mA ve stavu nečinnosti. Například vypnutím HDMI portu se odběr sníží o 25 mA, vypnutí LED diod ušetří až 5 mA. Spotřeba elektrické energie také velmi záleží na počtu připojených periférií. Samostatná Python kamera má spotřebu přibližně 250 mA.

Velký vliv na spotřebu má také vytížení procesoru. Bude tedy vhodné optimalizovat algoritmus detekce obrazu tak, aby využíval co nejméně procesorového času a ve výsledku spotřeboval co nejméně elektrické energie.

GPIO

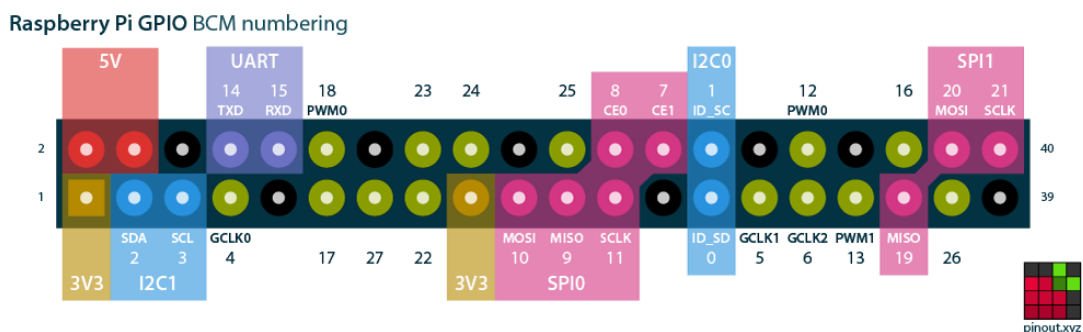
GPIO je zkratka pro General Purpose Input Output. Jsou to uživatelsky konfigurovatelné piny, které mohou být jak vstupní, tak výstupní. Raspberry Pi Zero, Raspberry Pi 3 a Raspberry Pi 2 disponuje 40 takovými piny. Pro srovnání - první verze Raspberry Pi A a B disponovala pouze 26 těmito piny.

Raspberry Pi používá 3,3voltovou logiku, je tedy nutné použít pro zapojení periférií využívajících 5 V převodník úrovní.

GPIO je možné nastavit například jako UART, SPI, I2C nebo PCM.

Pro ovládání GPIO lze použít Python modul GPIO, který ale v době psaní práce (verze 0.63) umožnil pouze softwarové GPIO. Nebylo tedy například umožněno hardwarové generování PWM signálu.

Pro hardwarové generování signálu lze využít modul RPIO, který umožňuje generování PWM pomocí DMA s přesností 1,2 mikrosekundy.



Obr. 1.2: Raspberry Pi GPIO Pinout, převzato z pinout.xyz

Úložiště

Raspberry Pi standardně používá jako úložiště (micro)SD karty. Maximální propustnost čtečky karet je 25 MB/s, je dána omezením sběrnice USB.

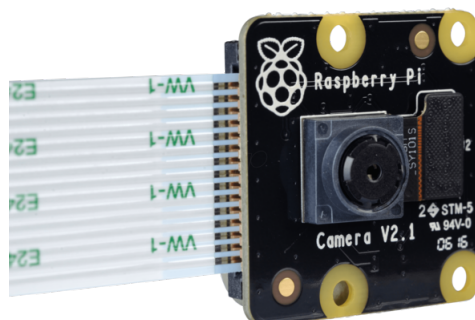
Spotřebitelské SD karty používají levné paměti typu TLC. Berme tedy v úvahu, že spotřebitelské SD karty mají kratší životnost než při užití pevného disku. K dostání jsou ale i karty pro průmyslové použití, které využívají paměti typu MLC či SLC, které mají několikanásobně vyšší životnost než TLC paměti.

Životnost paměti se dá zvýšit například použitím RAM disku tmpfs v Linuxu. Také je možné přesunout root na jiný typ úložiště a SD kartu ponechat pouze na bootování.

Při preferenci co nejnižší spotřeby elektrické energie monitorovacího zařízení, bude vhodné použít jako hlavní datové úložiště pouze SD kartu.

Raspberry Pi NoIR Camera Board V2

Vzhledem k požadavku na noční snímání je použita kamera bez infračerveného filtru Raspberry Pi NoIR Camera Board V2. Kvůli absenci IR filtru je možné využití okem neviditelného infračerveného přísvitu.

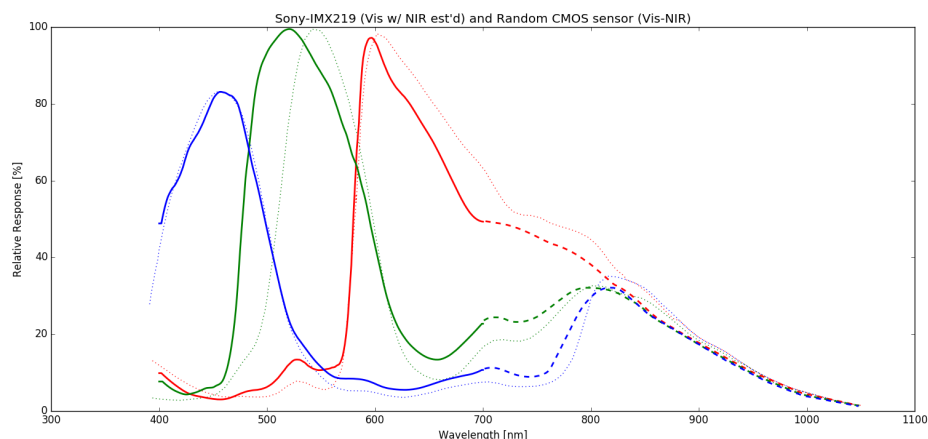


Obr. 1.3: Raspberry Pi NoIR Camera V2, převzato z reichelt.de

Kamera disponuje 8 megapixelovým obrazovým senzorem Sony IMX219 a fixním ohniskem. Video je možné natáčet až do rozlišení 1920x1080 pixelů při 30 fps a je možné pořizovat statické snímky až do rozlišení 3820x2464 pixelů.

Zorné pole kamery v horizontálním směru je 62 úhlových stupňů.

Kamera je připojena pomocí vysokorychlostního sériového rozhraní CSI, které má ve verzi 3 teoretickou maximální datovou propustnost 5,8 Gbps.



Obr. 1.4: Aproximovaná spektrální citlivost CMOS snímače Sony IMX219. Autorem je Koen Hufkenssem [7]

Data zobrazena plnou čarou zobrazují spektrální citlivost ve viditelném spektru a jsou extrahovány z katalogového listu produktu.

Data zobrazena přerušovanou čarou jsou aproximována porovnáním spektrální citlivosti podobného CMOS snímače od firmy Sony.

Z grafu tedy vyplývá, že citlivost snímače klesá se prodlužující se vlnovou délkou světla, které dopadá na snímač CMOS. Přibližně při vlnové délce světla 1050 nm je spektrální citlivost snímače rovna téměř nulová.

Tuto informaci můžeme využít k nákupu vhodné infračervené diody pro přisvit kamery, aby bylo zajištěno dobré snímání obrazu při nevhodných světelných podmínkách.

GSM/GPRS modul SIM800

SIM800 je modul od společnosti SIMCom pro bezdrátovou komunikaci pomocí technologie GSM a GPRS.

Podporuje frekvenční pásma 850/900/1800/1900 MHz, hlasovou komunikaci, SMS a GPRS data třídy 10.

Maximální teoretická přenosová rychlost pro CS4 je tedy 85,6 kbps pro downlink a 21,4 kbps pro uplink. Ke komunikaci je možné využít sériové rozhraní podporující AT příkazy, nebo je možné využít PPP stack pro internetové připojení.

Pro připojení k Raspberry Pi využijeme sériovou sběrnici UART. Modul podporuje baudovou rychlost od 1200 do 460800 Bd.

Point-to-Point Protocol

Point-to-Point Protocol, zkráceně PPP je název komunikačního protokolu linkové vrstvy, používaný pro přímé spojení mezi dvěma síťovými uzly.

V našem případě budeme tento protokol používat pro navázání spojení mezi GPRS modulem po sériové lince. K připojení je použit program PPPD, který naváže spojení pomocí tzv. chatovacích skriptů.

AT příkazy

AT příkazy jsou instrukce pro řízení modemu. AT je zkratka pro ATention.

Každý příkaz začíná „AT“. Jako příklad uvedme příkazy specifické pro komunikaci s GSM/GPRS modemem: AT+CMGS je příkaz pro odesílání SMS zprávy, AT+CGSN je příkaz pro získání IMEI čísla modemu.

APN

APN, neboli Access Point Name je jméno brány mezi sítí GSM/GPRS a sítí Internet. Název APN pro daného mobilního operátora nám musí být znám před připojením.

PIR modul HC-SR501

Modul HC-SR501 používá PIR element LHI778 a integrovaný obvod BISS0001 pro obsluhu signálu z PIR elementu.

Modul umožňuje nastavení citlivosti snímání v rozsahu od 3 do 7 metrů vzdálenosti snímaného objektu od kamery a také nastavení časového zpoždění mezi detekovaným pohybem a odeslaným signálem o detekci.

Nastavení citlivosti snímání je prováděno pomocí potenciometru na modulu.

Napájecí napětí je v rozmezí 5 až 20 V. Pokud je detekován pohyb, objeví se na výstupním pinu logická jednička (3,3 V), která zůstane aktivní po nastavenou dobu.

Nastavení časového zpoždění je též prováděno potenciometrem.

Po vypršení nastaveného zpoždění, zůstane modul ještě 2,5 sekundy ve stavu logické nuly.

V tomto 2,5 sekundovém časovém úseku nemůže být detekován další pohyb snímaného objektu.

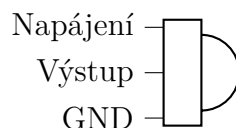
Modul může pracovat ve dvou režimech detekce pohybu, buď v jednoduchém nebo opakovaném.

V jednoduchém režimu dojde k zahájení odpočítávání časovače zpoždění ihned po detekci pohybu. K dalšímu restartování může dojít až po vypršení nastaveného času.

V opakovaném režimu je časovač restartován s každým dalším detekovaným pohybem.

Tab. 1.1: Možnosti modulu HC-SR501

Pin nebo nastavení	<i>Funkce</i>
Zpoždění	Jak dlouho má být na výstupním pinu logická 1
Citlivost	Rozsah nastavení vzdálenosti od 3 m až 7 m
Spouštění	Jednoduché nebo opakované zpoždění.
GND	Uzemnění
Výstup	Logický výstup 3,3 V, pokud je detekován pohyb
Napájení	5 až 20 V DC napětí



Obr. 1.5: Schéma PIR modulu

Výkonová IR LED dioda GT-P04IR4101

Parametry

Tab. 1.2: Parametry LED z katalogu

Parametr	<i>min.</i>	<i>typická</i>	<i>max.</i>
Ztrátový výkon		1 W	
Úbytek napětí	1,65 V		1,75 V
Proudový odběr		400 mA	420 mA
Vyzářený výstupní výkon		400 mW	
Vlnová délka světla		850 nm	
Tepelný odpor		12 °C/W	
Úhel osvětlení		120°	

1.2 Programové vybavení

Raspbian

Raspbian je linuxový operační systém založený na distribuci Debian. Je optimalizovaný pro hardware Raspberry Pi, určený pro architekturu armhf. První verze byla vytvořena v roce 2012. Raspbian bude použit ve verzi lite, tedy bez grafického rozhraní a pouze s nutnými programovými knihovnami.

Python

Python je vysokoúrovňový skriptovací jazyk navržený Guidem Van Rossem v roce 1991. Jazyk podporuje několik programovacích paradigmat včetně objektového, procedurálního a funkcionálního. V roce 2017 existovaly dvě navzájem částečně nekompatibilní odnože Pythonu a to 2.7.x a 3.x.x. Python je snadno rozšiřitelný o další programové knihovny – balíčky.

Picamera

Picamera je knihovna pro Python, která umožňuje ovládat kameru pro Raspberry Pi, včetně pořizování statických snímků a videí. Poskytuje poměrně bohaté programové API.

Pillow

Pillow je knihovna pro Python, která vznikla jako derivát staršího modulu PIL. Pillow rozšiřuje schopnosti Pythonu o pokročilé zpracování obrazu.

PyDrive

PyDrive je knihovna pro Python pro práci s cloudovým úložištěm Google Drive. Vznikla jako zjednodušení Google Drive REST API. Knihovna umožňuje jednoduchou manipulaci se soubory uloženými na Google Drive. K autentizaci uživatele je použit framework OAuth 2.

Python-gsmmodem

Python-gsmmodem je knihovna pro Python pro komunikaci s modemy pomocí AT příkazů. Podporuje odesílání a přijímání SMS zpráv a také hlasovou telefonii.

Epsolar-tracer

Epsolar-tracer je knihovna pro Python pro komunikaci se solárním regulátorem Epsolar využívající protokol Modbus a rozhraní RS-485.

Google Apps Script

Google Apps Script je skriptovací jazyk založený na JavaScriptu pro webové aplikace běžící pod Google Cloudem. Google poskytuje běhové prostředí a editor s debuggerem. V projektu je tento jazyk využíván pro vytvoření grafického rozhraní pro konfiguraci aplikace a pro sběr dat.

JSON

JSON je zkratka pro JavaScript Object Notation. Je to jednoduchý formát pro výměnu dat. Jeho výhoda spočívá v tom, že má jednoduchou strukturu. Je tedy lehce čitelný i lehce editovatelný a má také velmi dobrou podporu v programovacích jazycích. V projektu je tento formát využíván pro konfiguraci aplikace.

Závěr k výběru softwaru

Veškeré vybrané softwarové vybavení se řadí do kategorie Open-source. Takový software je vyvíjen především komunitně a jeho zdrojový kód je volně k dispozici. Cena softwaru je tedy nulová. Další kapitola se bude zabývat využitím vybraného softwaru.

1.3 Detekce pohybu

Detekce pohybu je proces detekce změny pozice zkoumaného objektu vůči pozadí. Zde se budeme zabývat především elektronickou detekcí pohybu.

Existuje několik metod elektronické detekce pohybu, například optická nebo akustická metoda. Mezi optické metody patří detekce infračerveného záření pomocí PIR senzoru. Příklad akustické metody je detekce ultrazvukem.

Pohyb

Pod pojmem pohyb rozumíme pohyb mechanický, což je změna polohy pozorovaného objektu. Pohyb nelze detekovat z jednoho statického obrazu, potřebujeme obrazový tok. Budeme tedy pracovat s dynamickým obrazem, což je posloupnost statických obrazů.

Detekce pohybu pomocí PIR senzoru

Zkratka PIR znamená „passive infrared“, jde tedy o pasivní infračervený detektor, který funguje na principu pyroelektrického jevu.

Pyroelektrický jev

Pyroelektrický jev je schopnost materiálu generovat dočasný elektrický potenciál při změně teploty. Vyskytuje se u dielektrik s jednou polární osou symetrie.

PIR Element

PIR element je základní funkční prvek PIR detektoru. Je to polovodičová součástka, citlivá na světlo, vyrobená ze sloučenin na bázi lithia a tantalu. PIR snímač je citlivý v širokém spektru záření, je proto nutné před snímač aplikovat filtr, který propouští pouze infračervené záření o vlnových délkách v rozsahu 8 až 14 μm . Lidské tělo emituje záření o vlnové délce 9,4 μm . Se změnou intenzity dopadajícího infračerveného záření na povrch pyroelektrického materiálu se změni hodnota elektrického povrchového náboje. Tato změna je měřena citlivým tranzistorem FET. [5]

Optika

Úkolem optické soustavy je soustředit infračervené záření z detekční zóny do PIR elementu. V praxi se používají Fresnelovy čočky nebo zrcadla.

Detekce pohybu z dynamického obrazu

Existuje několik metod pro detekci pohybu ze snímané scény. Metody jsou primárně založené na segmentaci pozadí a popředí obrazu. Pozadím nazýváme část obrazu ve které zjišťujeme změny a popředí nese právě ty změny. Pozadí ale většinou nebývá statické, je v něm obsažen například šum snímacího zařízení, nebo změny osvětlení.

Je několik algoritmů, které tyto problémy řeší s různými nároky na výpočetní výkon.

Zmiňme například:

- Porovnávání histogramu mezi snímky
- Sledování rozdílných bodů mezi snímky
- Porovnání snímků zpracovaných detektorem hran
- Metoda optického toku
- GMM – Gaussian Mixture Models
- Sigma-Delta filtrace

Porovnávání histogramu mezi snímky

Jedná se o jednu z nejjednodušších metod detekce pohybu. Porovnávají se histogramy dvou snímků - vyhodnocují se tedy jasové charakteristiky snímků. Tato metoda je výpočetně málo náročná a je vhodná pouze do podmínek se stálým osvětlením. Je nutné dobře zvolit referenční snímek podle kterého se bude porovnávat pohyb a průběžně ho aktualizovat podle změn v prostředí. Výpočetně nejméně náročné je průměrování histogramů několika snímků, ve kterých není označen pohyb. Také je dobré snímek zbavit šumu přes šumové filtry. [4]

Výhody

Výpočetně málo náročná metoda.

Nevýhody

Metoda je náchylná na šum a také na změny prostředí.

Sledování rozdílných bodů mezi snímky

Tato metoda spočívá v porovnávání snímků na úrovni pixelů. Je vhodné snímky nejprve převést do 8-bitového barevného prostoru a potom odečítat hodnoty. Obrázky od sebe v této podobě odečteme a získáme nový snímek, který nyní bude prezentovat rozdíl složek. Čím méně se od sebe korespondující pixely liší, tím se jejich hodnota v rozdílu bude více blížit hodnotě 0 (černá barva). Poté je nutné aplikovat metodu prahování abychom rozeznali informaci od šumu. [4]

Výhody

Schopnost určit místo ve scéně na kterém došlo k pohybu.

Nevýhody

Relativně výpočetně náročná operace porovnávání snímků na úrovni pixelů.

Metoda optického toku

Metoda optického toku dokáže sledovat pohyb pixelu mezi snímky ve video sekvenci a zjistit tak rychlost pohybu. Tímto způsobem se dá předpovídat trajektorie sledovaného objektu. Metoda je výpočetně velmi náročná pro sledování všech pixelů ve scéně. S určitou optimalizací je však použitelná i jako detektor pohybu.

1.4 Mobilní sítě

GSM

GSM (Groupe Spécial Mobile) popisuje protokoly pro mobilní telekomunikaci druhé generace. Poprvé bylo GSM použito v roce 1991 ve Finsku a následně se stalo standardem pro hlasovou komunikaci po celém světě. Standard GSM původně využíval pouze přepínání okruhů optimalizovaných pro full-duplex hlasovou telefonii, avšak příchod GPRS rozšířil síť o možnost paketového přenosu dat.

GPRS

Systém GPRS (General Packet Radio Service) je rozšíření systému GPS o datový paketový přenos. Umožňuje připojení do sítě Internet.

1.5 Solární napájení

Jde o způsob získávání elektrické energie přeměnou ze sluneční energie. Energii můžeme získávat buď pomocí fotovoltaických článků, které využívají fotovoltaický jev, nebo pomocí tzv. koncentrovaných solárních panelů. Nedílnou součástí napájecího obvodu se solárními články je akumulátor, který pokrývá napájení, když článek nedodává dostatečný proud, což se děje například za zhoršených světelných podmínek nebo v noci.

Fotovoltaický článek

Fotovoltaický článek využívá fotovoltaického jevu, což je jedna z forem vnitřního fotoelektrického jevu.

Princip funkce

Fotovoltaický jev vzniká v polovodičích, když foton s dostatečnou energií uvolní elektron z valenčního pásu do pásu vodivostního. Ve valenčním pásu zůstane „chybějící elektron“, tzv. díra, kterou lze považovat za elementární kladný náboj (díra se pohybuje tak, že se do ní přemístí valenční elektron sousedního atomu, čímž se díra přesune na původní místo tohoto elektronu).

Zjednodušeně lze prohlásit, že dopadem fotonu se vytvoří pár pohyblivých nábojů elektron-díra. Tyto náboje se difúzí nebo působením elektrického pole v okolí PN přechodu pohybují ve směru k elektrodě se stejnou polaritou – elektron k záporné a díra ke kladné. Při propojení elektrod vnějším obvodem putují elektrony k opačné elektrodě, kde rekombinují s děrami – vnějším obvodem prochází elektrický proud.[3]

Monokrystalický článek

Monokrystalický článek je vytvořen na jediném velkém krystalu křemíku. Dosahuje nepatrně lepší efektivity (14 - 18 %) než polykrystalický, a to pouze, pokud má panel ideální orientaci ke slunci.

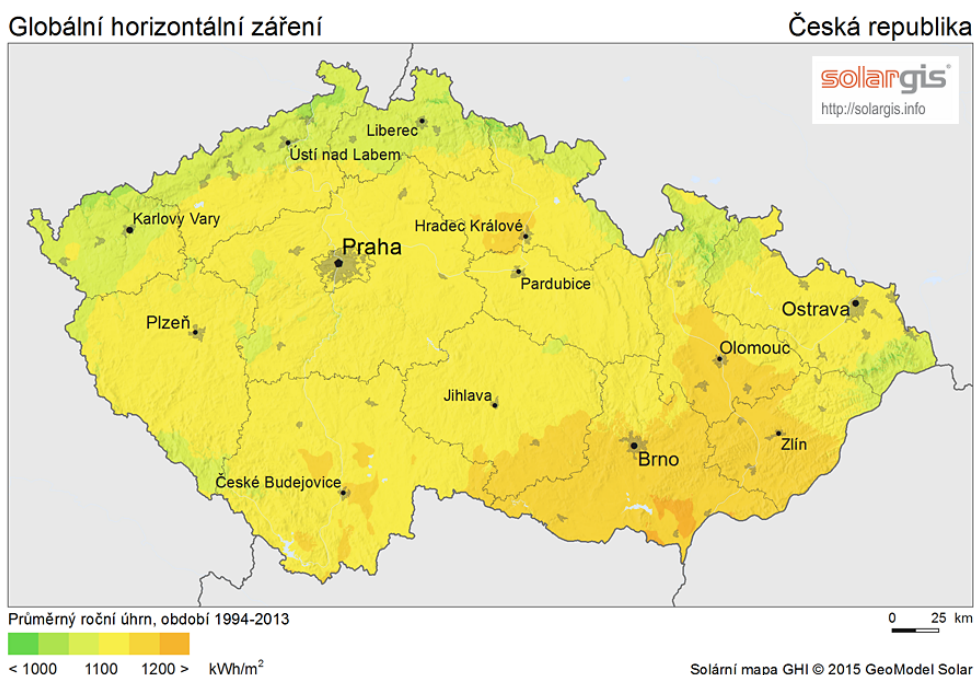
Polykrystalický článek

Je složen z většího počtu článků, které jsou pak navzájem propojeny. Panel bývá světlejší a může se na pohled jevit jako „flekatý“. Tento panel produkuje rovnoměrnější výkon, ale má nižší účinnost (asi 12 - 17 %). Jeho použití je vhodné, pokud budou na panel dopadat sluneční paprsky pod různým úhlem.

Efektivita solárních panelů

Efektivita solárních panelů bude závislá především na těchto faktorech:

- Zeměpisná poloha, kde je panel umístěn.
- Poloha panelu vůči slunci.
- Velikost solárního panelu.
- Typ solárního panelu.
- Účinnost obvodu pro přenos energie.



Obr. 1.6: Globální horizontální záření [6]

U solárních panelů se udávají tyto parametry:

- Výkon - jaký výkon je panel schopný podávat.
- Napětí naprázdno - jaké napětí se na svorkách panelu může objevit, pokud není panel zatížen (spotřebičem).
- Maximální napětí při plném výkonu - jaké napětí bude na svorkách panelu, pokud bude zatížen maximálním povoleným proudem.
- Maximální proud při plném výkonu - maximální povolený odebíraný protékající proud.

Akumulátor

Akumulátor je zařízení, které má schopnost uchovávat energii. Nejběžnější typy elektrických akumulátorů pracují na elektrochemickém principu. Mezi takové akumulátory patří například: nikl-kadmiové (NiCd), nikl-metal hydridový (NiMh), lithium-polymerový (Li-pol), lithium-iontový (Li-ion) a olověný (Pb).

Lithiové akumulátory

Lithiové akumulátory patří v současnosti k nejpoužívanějším typům akumulátorů. Najdeme je téměř v každé přenosné spotřební elektronice. Mezi lithiové akumulátory řadíme lithium-iontové (Li-ion) a novější lithium-polymerové (Li-pol).

Mezi výhody lithiových akumulátorů patří především velká energetická hustota, můžeme tedy mít baterii s vysokou kapacitou s relativně malou hmotností a objemem. Další výhodou je nízké samovybíjení, schopnost dodávat vysoké proudy a také relativně dlouhá životnost.

Mezi nevýhody zmiňme nutnost složitějšího obvodu pro nabíjení. Je nutné hlídat akumulátor před přílišným nabitím a vybitím a také před vysokou teplotou. Je zde také nebezpečí vznícení nebo výbuchu při zkratu.

Olověné akumulátory

Olověný akumulátor je typ sekundárního galvanického článku s elektrodami na bázi olova. Elektrolytem je kyselina sírová.

Výhodou je především jednodušší obvodové řešení pro nabíjení a také nižší cena vzhledem ke kapacitě akumulátoru.

Nevýhodou je zejména vyšší hmotnost akumulátoru.

2 NÁVRH APLIKACE

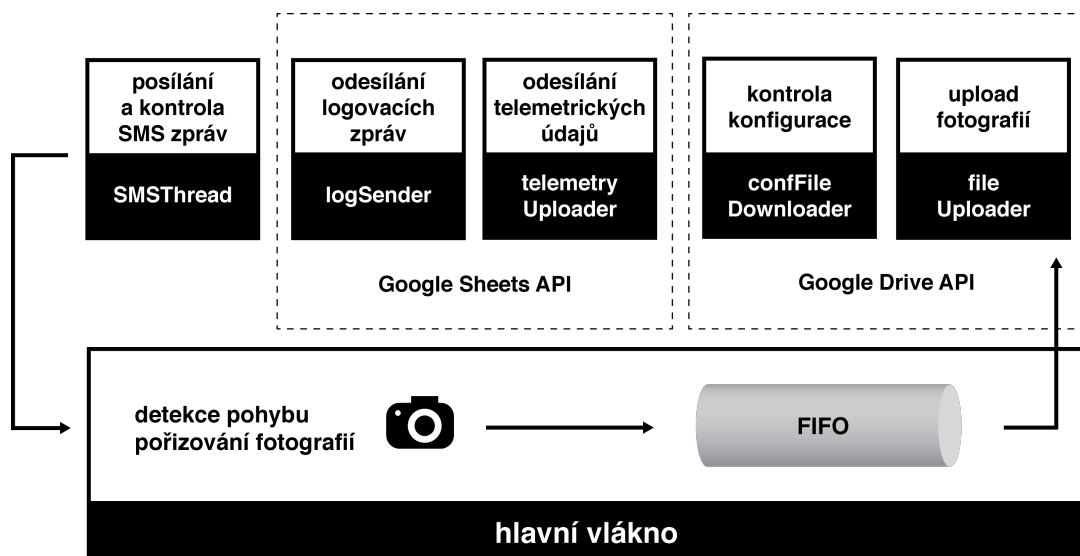
2.1 Architektura aplikace

Aplikace běžící na Raspberry Pi je napsána v programovacím jazyce Python verze 3.6.5. Aplikace se spustí po startu operačního systému a provede inicializaci vlákna pro stahování konfiguračního souboru v daných intervalech, poté provede inicializací vlákna pro upload fotografií.

Vícevláknové zpracování

Aplikace bude provádět současně několik činností.

Bude nutné nepřetržitě monitorovat pohyb, odesílat fotografie a současně vyčítat stav baterie. Také je důležité kontrolovat, zda nedošlo ke změně konfigurace. Z toho důvodu bude třeba program rozdělit do více na sobě nezávislých vláken.



Obr. 2.1: Architektura aplikace

Hlavní vlákno aplikace

Hlavní vlákno aplikace se stará o detekci pohybu a pořizování fotografií.

Jakmile je detekován pohyb, je provedena zkušební fotografie, ze které je odečtena celková jasová složka snímaného obrazu.

Pokud je hodnota jasu pod prahem denního snímání, jsou nastaveny parametry kamery pro noční režim a je pořízena fotografie.

Fotografie je zkomprimována, uložena na SD kartu a vložena do fronty pro zaslání do externího úložiště.

Pokud je aktivovaný režim echo, je provedeno ještě dalších 10 fotografií v sekundových intervalech.

Odesílání fotografií do úložiště

Odesílání fotografií je řešeno ve vlákne nazvaném fileUploader.

Toto vlákno umožňuje autentizaci s Google Drive pomocí OAuth 2.0. Následně je vyčítána fronta a fotografie jsou postupně odesílány do úložiště.

Pokud je aplikace v režimu batch, je odesílání fotografií aktivováno až ve stanovenou dobu.

Kontrola konfigurace

Vlákno kontroly a stahování konfiguračního souboru zvané configFileDownloader rovněž používá Google Drive API.

Po spuštění vlákna se inicializuje spojení s Google Drive a stáhne se JSON konfigurační soubor podle zadaného identifikátoru. JSON data v konfiguračním souboru jsou přečtena pomocí modulu JSON a naimportována do třídy UserConfig.

V rámci aplikace jsou konfigurační data uložena v modulu config ve třídách BaseConfig a UserConfig, přičemž třídu BaseConfig nelze konfigurovat pomocí JSON a třída UserConfig je určena pro uživatelskou konfiguraci.

Stahování konfiguračního souboru a kontrola změny probíhá ve stanovených intervalech, výchozí hodnota je každých 5 sekund. Pokud vlákno nepracuje, je uspáno.

Odesílání telemetrických údajů

Vlákno odesílání telemetrických údajů využívá Google Sheets API ke komunikaci s Google Sheets.

Vlákno vyčítá informace o stavu nabíjení akumulátoru ze solárního regulátoru a také informace o využití systémových prostředků jako vytížení CPU a využití operační paměti.

Data se převádí do podoby, kterou je možné vložit do tabulky Google Sheets. Vyčítání dat a odesílání do tabulky je prováděno každých 5 sekund.

Odesílání logovacích zpráv

Vlákno odesílání logovacích zpráv využívá Google Sheets API ke komunikaci s Google Sheets.

Vlákno je implementováno jako takzvaný handler pro modul logging, který je využíván pro logování zpráv. Namísto ukládání logovacích zpráv do souboru se zprávy ukládají do fronty a v daných intervalech jsou posílány do listu Google Sheets.

Odesílání a kontrola SMS zpráv

Vlákno SMSThread provádí kontrolu přijatých zpráv.

Pokud je přijata SMS zpráva s významem z autorizovaného čísla, tak se provede akce. Pokud je přijata zpráva z neautorizovaného čísla, tak je okamžitě vymazána.

2.2 Detekce pohybu

Zařízení umožňuje dvě metody detekce pohybu - detekce pohybu z obrazu a detekce pohybu pomocí PIR čidla.

Ve výchozím nastavení jsou zapnuty obě metody detekce, avšak detekci pohybu pomocí PIR senzoru je možné vypnout.

Detekční úhel je dán omezením horizontálního pozorovacího úhlu zvolené kamery. V našem případě výrobce kamery udává pozorovací úhel 62 úhlových stupňů v horizontálním směru a 48 úhlových stupňů ve vertikálním směru.

Detekce pohybu pomocí PIR

Výhodou detekce pohybu pomocí PIR senzoru je nezávislost na osvětlení scény. Je možné ji použít například v noci, kdy již nelze použít detekci pohybu z obrazu z důvodu necitlivosti kamery.

Další výhodou detekce pohybu PIR senzorem je nízká výpočetní náročnost, protože je kontrolována pouze jedna logická hodnota.

Efektivita detekce pohybu PIR senzorem je dána těmito parametry: citlivosti PIR senzoru, pozorovacím úhlem kamery a dosahem LED přísvitů.

PIR senzor použitý v této práci má udávanou spolehlivou detekční vzdálenost maximálně 5 metrů, z čehož vyplývá dosah LED přísvitů, tedy také 5 metrů.

Detekce pohybu z obrazu

Výhodou detekce pohybu z obrazu je kupříkladu možnost umístit kameru za sklo, respektive okno. V případě detekce pohybu pomocí PIR senzoru toto není možné.

Detekce pohybu z obrazu využívá algoritmus výpočtu vektorů pohybu z optického toku.

Tato softwarová metoda je výpočetně velmi náročná, ale patří mezi nejspolehlivější metody. Není tolik ovlivňována změnou osvětlení na snímané scéně a je odolná proti šumu.

V případě Raspberry Pi je kódér H.264 implementován přímo na GPU. Tento princip se využívá například v kódéru MPEG pro kompresi videa.

Hledání vektorů pohybu

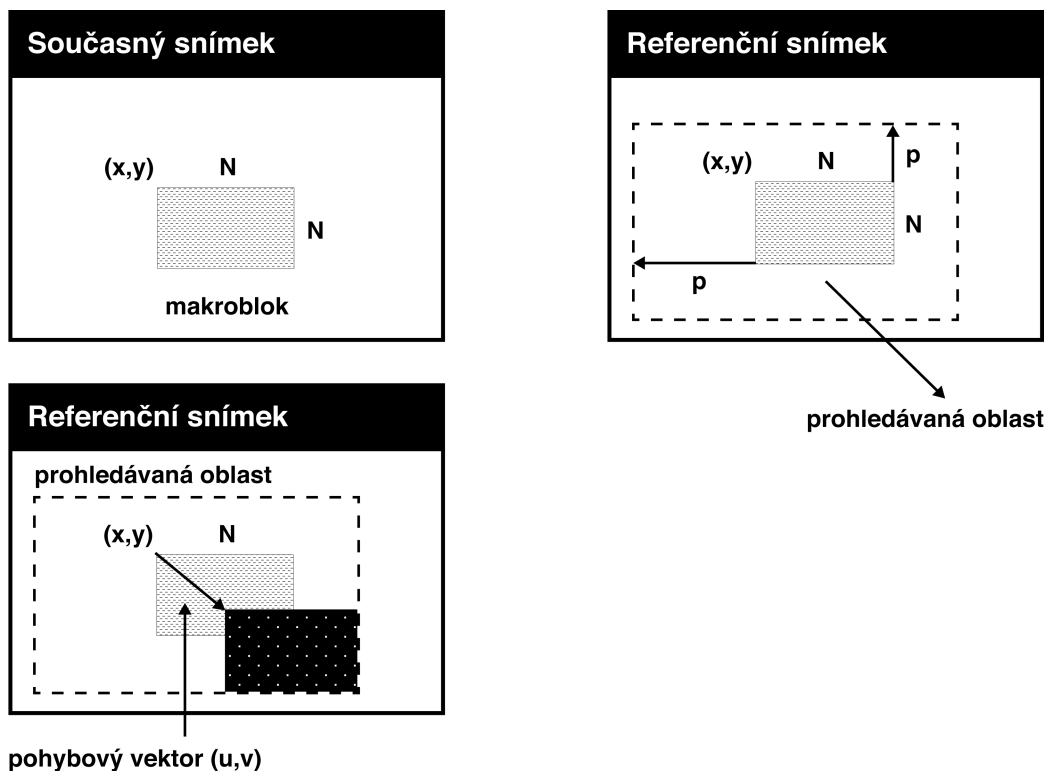
Data o vektorech pohybu je možné získat přímo z kódéru H.264.

V našem případě snímáme video v rozlišení 640x368 pixelů. Vektory pohybu jsou počítány přímo na úrovni makrobloků (makroblok má rozměry 16x16 pixelů).

V našem případě bude tedy mít matice vektorů pohybu rozměry 40+1 sloupců a 23 řádků. Matice obsahuje jeden sloupec dat navíc.

Data o pohybu mají velikost 4 bajty. Každá z proměnných vektoru pohybu $[x]$ a $[y]$ má velikost jeden bajt, zbývající dva bajty obsahují sumu absolutních diferencí (SAD).

Suma absolutních diferencí udává míru odlišnosti mezi referenčním blokem a současným blokem. Je vypočtena jako suma rozdílů absolutních hodnot mezi jednotlivými pixely v původním bloku a bloku, který porovnáváme.



Obr. 2.2: Hledání vektorů pohybu

Třída pro detekci pohybu z obrazu

PiMotionAnalysis je třída poskytnutá knihovnou PiCamera pro detekci pohybu z obrazu v reálném čase, využívající metodu počítání vektorů pohybu.

Třída implementuje metodu analyse, která je volána pro každý snímek. V této metodě je nutné implementovat logiku počítání velikosti vektorů a prahování.

Prah detekce je možné parametrizovat minimální velikostí vektoru, tj. o kolik se daný makroblok posunul. A také počtem vektorů splňující danou minimální velikost.

Níže je ukázka výpočtu velikosti pohybových vektorů za použití modulu Numpy.

```
class vectorMotionAnalysis(picamera.array.PiMotionAnalysis):
    def analyse(self, a):
        a = np.sqrt(
            np.square(a['x'].astype(np.float)) +
            np.square(a['y'].astype(np.float))
        ).clip(0, 255).astype(np.uint8)
        # Jestli je více jak 10 vektorů s velikostí
        # větší než 60, pohyb je detekován
        if (a > 60).sum() > 10:
            print('Pohyb detekován!')
```

2.3 Řízení snímání fotografií

Řízení snímání fotografií je poskytnuto třídou CaptureHandler a metodou tick.

Metoda tick je volána každou sekundu.

Pokud je zavolána metoda motion_detected a přitom současně není zpracovávána žádná jiná fotografie, provede se metoda tick.

Parametry snímaných fotografií

Ve výchozím nastavení se snímá v rozlišení 640x368 pixelů.

Pro denní režim je nastavení módu expozice a vyvážení bílé automatické. Rychlost uzávěrky je také automatická. V nočním režimu je použit expoziční režim nightpreview a doba uzávěrky je nastavena na 200 ms.

Fotografie jsou zkomprimovány a optimalizovány za použití modulu PIL ve výchozím nastavením komprese 75.

```

# Ukázka snímání fotografie v noci
with picamera.PiCamera() as camera:
    camera.resolution = (1280, 720)
    # Snímkovací frekvence je nastavena na 1 fps
    # Doba uzávěrky je nastavena na 200 milisekund a ISO na 800
    camera.framerate = 1
    camera.shutter_speed = 200000
    camera.exposure_mode = 'nightpreview'
    camera.iso = 800
    # Kamera potřebuje čas na změření AWB - Auto white balance
    sleep(10)
    camera.capture('dark.jpg')
    # Snímek je pořízen

```

Detekci denní doby

Detekce intenzity osvětlení (respektive denní doby) je řešena softwarově.

Využívá se výpočtu průměrné hodnoty pixelů vyjádřené v RGB.

Tato metoda není příliš přesná, ale je dostačující pro rozhodnutí, zda-li má být fotografie pořízena v nočním režimu s přisvitlením.

Měří se průměrná hodnota pixelu v testované matici.

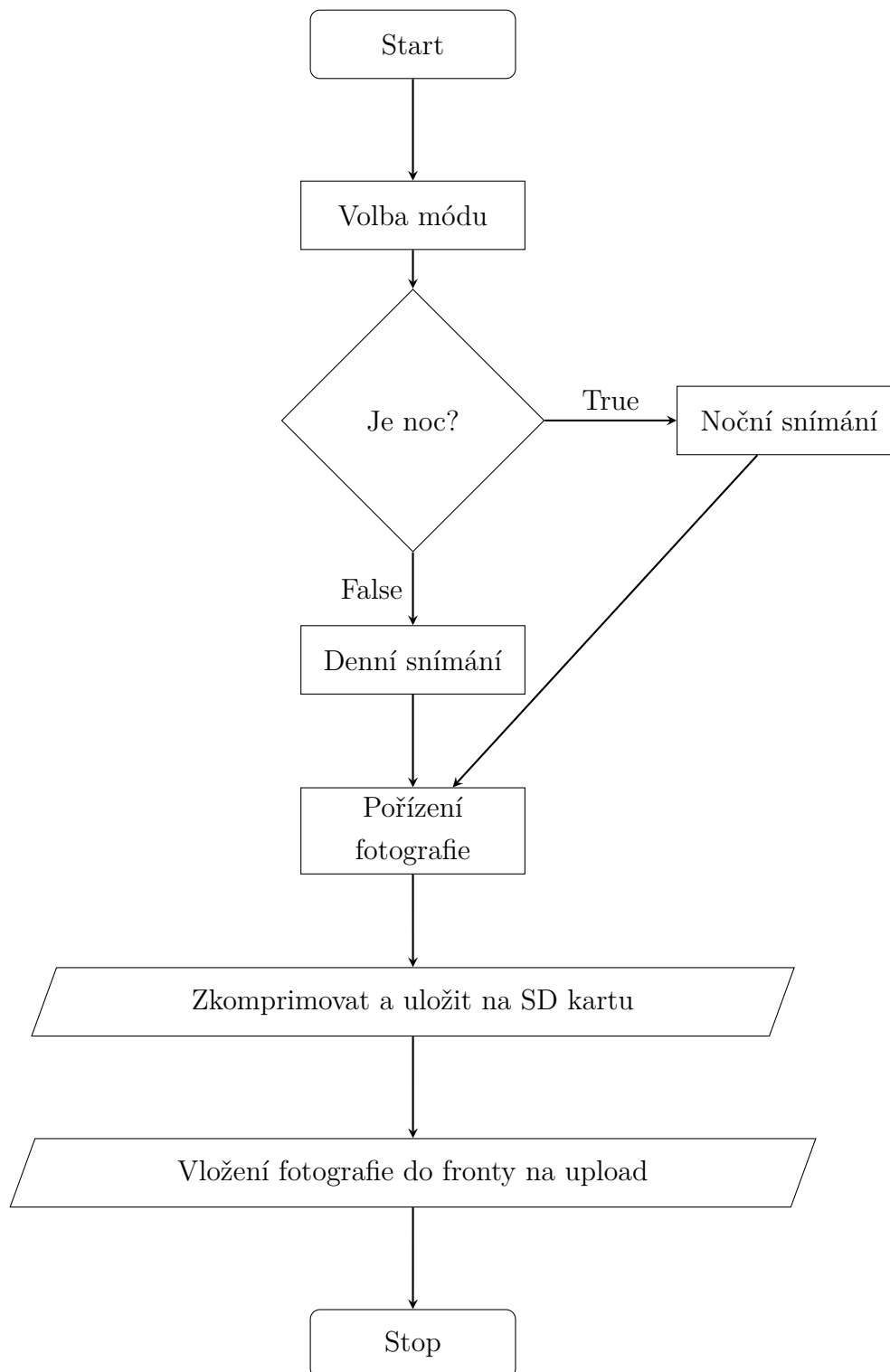
Experimentálně bylo zjištěno, že pokud je průměrná hodnota vyšší jak 50, můžeme prohlásit, že světelné podmínky jsou dostatečné pro použití parametrů pro denní snímání.

```

def scanIfDay():
    with picamera.PiCamera() as camera:
        camera.resolution = (testWidth, testHeight)
        with picamera.array.PiRGBArray(camera) as stream:
            camera.exposure_mode = 'auto'
            camera.awb_mode = 'auto'
            camera.capture(stream, format='rgb')
            pixAverage = int(np.average(stream.array[...],1))
    logging.info("Light Meter pixAverage=%i" % pixAverage)
    if (pixAverage > 50):
        return True
    else:
        return False

```

Vývojový diagram detekce pohybu



2.4 Detekce pohybu z PIR pomocí obsluhy přerušení

Ukázka řešení detekce pohybu ze senzoru PIR pomocí obsluhy přerušení.

Obsluha přerušení je řešena pomocí modulu GPIO.

PIR senzor je připojen k GPIO pinu 7.

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM) # nastav mod cislovani na BCM
PIR_PIN = 7             # nastaveni PIR na pin 7 (BCM)
GPIO.setup(PIR_PIN, GPIO.IN) # nastavime pin na vstupni

def motion(PIR_PIN):
    print("Detekovan pohyb!")

print ("Test PIR senzoru") # pockame na inicializaci
time.sleep(2)
print("Pripraven")

try:
    GPIO.add_event_detect(PIR_PIN, GPIO.RISING, callback=motion)

while 1:
    time.sleep(100)

except KeyboardInterrupt:
    print("Konec") # ukonceni na preruseni z klavesnice
    GPIO.cleanup()
```


Odesílání fotografií na cloudové úložiště

Po nasnímání jsou fotografie zkomprimovány a optimalizovány s použitím JPEG podle nastavené úrovně komprese a uloženy na lokální úložiště.

Jméno fotografie je časová známka v době, kdy byla pořízena.

Toto jméno souboru fotografie je následně vloženo do fronty, která je sdílána s vláknem pro odesílání fotografií, ve kterém je v daných časových intervalech vyčítána a nezávisle na běhu hlavního vlákna je posílána do vzdáleného úložiště.

2.5 Ovládání aplikace

Ovládací panel aplikace

Ovládací panel je řešen prostřednictvím aplikace Google Sheets pomocí Google API, která umožňuje integraci aplikací napsaných v jazyce Python s produkty Google.

K tomuto řešení bylo přistoupeno z důvodu požadavku na vzdálenou konfiguraci aplikace bez nutnosti provozovat nějaký webový server pro běh rozhraní.

Google API je tzv. REST, to znamená, že je implementováno na protokolu HTTP.

Z toho plynoucí výhody jsou jednoduchost implementace a malý přenášený objem dat. Jsou přenášena pouze ta data, která jsou vyžádána.

V neposlední řadě je to také spolehlivost a rychlost zajištěná servery Google a také možnost publikace celého dokumentu veřejně přístupnou HTML stránkou.

Rozhraní vytvořené pro aplikaci obsahuje pět sešitů.

Úvodní sešit nazvaný Dashboard obsahuje grafické shrnutí údajů o stavu nabíjení a napájení a také o stavu systémových prostředků jako je vytížení CPU a RAM.

Data, ze kterých se vychází, se ukládají do sešitu zvaném Log.

V dalším sešitu jsou zobrazeny náhledy fotografií nasnímaných při detekci pohybu.

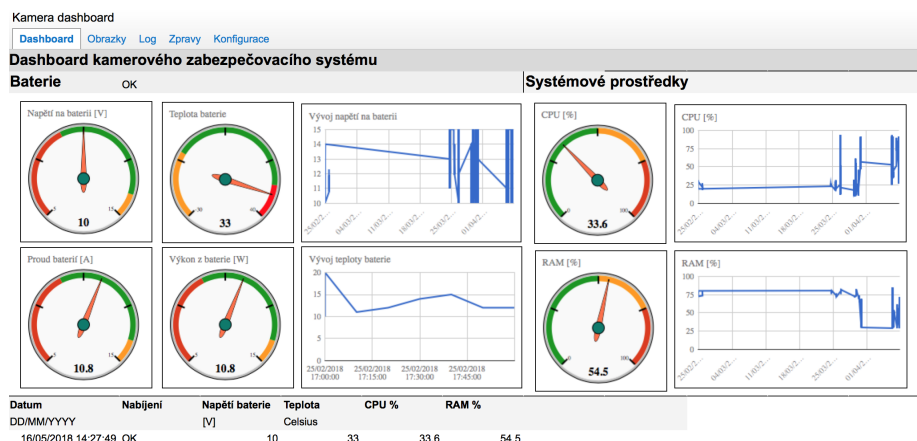
Následující sešit obsahuje log zpráv, označených jmény vláken, odkud zprávy přišly.

Zprávy jsou řazeny do čtyř kategorií:

- DEBUG - podrobné informace o chodu programu, je možné je vypnout.
- INFO - informační zprávy.
- WARNING - varování neohrožující chod programu.
- ERROR - chyby.
- CRITICAL - velmi závažné chyby, které znemožňují chod programu.

Poslední sešit umožňuje konfigurovat zařízení pomocí dialogového okna s formulářem.

Na základě zadaných dat je vygenerován konfigurační soubor ve formátu JSON, který je poté stažen ve vlákně pro kontrolu konfigurace.



Obr. 2.3: Informační panel aplikace

Ovládání a notifikace pomocí SMS

Aplikaci je možné ovládat pomocí SMS z jednoho autorizovaného čísla, pokud je v konfiguračním souboru nastavený parametr SMS control.

Tab. 2.1: Příkazy SMS

SMS příkaz	Popis	Odpověď
BATTERY	Informace o stavu baterie	Charging 12.3V
TAKEPHOTO	Okamžitá fotografie	-
SET MODE <x>	Nastav mód na x	Mode has been set to <x>
Q MODE	Dotaz na aktivní mód	Current mode is <x>
-	Nesprávný příkaz	Command not recognized

Notifikace pomocí emailu

Upozornění na detekci pohybu je také odesláno e-mailem prostřednictvím Google Cloud.

Kontrola nových fotografií na Google Drive je spouštěna každou minutu.

Pokud jsou detekovány nové fotografie, tak je odeslán e-mail na zvolenou e-mailovou adresu spolu s fotografiemi, které byly pořízeny. Následně se aktualizuje seznam fotografií v sešitu.

Kamera dashboard

Dashboard Monitoring Obrazky Log **Zpravy** Konfigurace

Datum	Úroveň	Vlákn	Zpráva
03-30 20:21	DEBUG	MainThread	Starting program
03-30 20:21	DEBUG	ConfFileDownlo	Initialization of Conf file checker thread
03-30 20:21	DEBUG	FileUploader	Initialization of file uploader thread
03-30 20:21	DEBUG	MainThread	File uploader started
03-30 20:21	WARNING	ConfFileDownlo	file_cache is unavailable when using oauth2client >= 4.0.0
03-30 20:21	WARNING	FileUploader	cannot import name 'file_cache'
03-30 20:21	INFO	ConfFileDownlo	URL being requested: GET https://www.googleapis.com/discovery/v1/apis/drive/v2/rest
03-30 20:21	INFO	FileUploader	URL being requested: GET https://www.googleapis.com/discovery/v1/apis/drive/v2/rest
03-30 20:21	DEBUG	MainThread	Starting camera
03-30 20:21	INFO	ConfFileDownlo	URL being requested: GET https://www.googleapis.com/drive/v2/files/1TBLQfJHsZYPXDvcpS_ysg6asxf-5_Oku?alt=json
03-30 20:21	DEBUG	MainThread	Capture of video started
03-30 20:21	INFO	TelemetrySender	URL being requested: POST https://sheets.googleapis.com/v4/spreadsheets/1XNmtg0NoCiU03NDZohIBwmOpFC-heHWbdRzY_
03-30 20:21	INFO	LogSender	URL being requested: POST https://sheets.googleapis.com/v4/spreadsheets/1XNmtg0NoCiU03NDZohIBwmOpFC-heHWbdRzY_
03-30 20:21	INFO	LogSender	URL being requested: POST https://sheets.googleapis.com/v4/spreadsheets/1XNmtg0NoCiU03NDZohIBwmOpFC-heHWbdRzY_
03-30 20:21	INFO	LogSender	URL being requested: POST https://sheets.googleapis.com/v4/spreadsheets/1XNmtg0NoCiU03NDZohIBwmOpFC-heHWbdRzY_
03-30 20:21	INFO	TelemetrySender	URL being requested: POST https://sheets.googleapis.com/v4/spreadsheets/1XNmtg0NoCiU03NDZohIBwmOpFC-heHWbdRzY_
03-30 20:21	INFO	LogSender	URL being requested: POST https://sheets.googleapis.com/v4/spreadsheets/1XNmtg0NoCiU03NDZohIBwmOpFC-heHWbdRzY_
03-30 20:21	INFO	Dummy-5	Motion detected!
03-30 20:21	INFO	Dummy-5	Motion detected!
03-30 20:21	INFO	Dummy-5	Motion detected!
03-30 20:21	INFO	Dummy-5	Motion detected!
03-30 20:21	INFO	Dummy-5	Motion detected!
03-30 20:21	INFO	Dummy-5	Motion detected!
03-30 20:21	INFO	Dummy-5	Motion detected!
03-30 20:21	INFO	Dummy-5	Motion detected!
03-30 20:21	DEBUG	MainThread	Capturing!
03-30 20:21	INFO	Dummy-5	Motion detected!
03-30 20:21	DEBUG	MainThread	Captured 2018-03-30T20:21:54.309269.jpg
03-30 20:21	DEBUG	MainThread	Echo mode enabled, capturing will continue
03-30 20:21	DEBUG	MainThread	Echo image captured

Obr. 2.4: Panel aplikace obsahující log

2.6 Konfigurace

Konfigurační data jsou v rámci aplikace uloženy ve dvou třídách - BaseConfig a UserConfig. Vzdáleně měnit je možné pouze konfigurace třídy UserConfig.

Konfigurace je načítána ze souboru ve formátu JSON, který je umístěn na úložišti Google Drive. Po stažení je zahájena kontrola a parsování souboru do třídy UserConfig.

Konfigurační soubor je v určitých intervalech dotazován na změnu. Pokud byl konfigurační soubor změněn, je stažen a načten. Toto je řešeno ve vlákně ConfFileDownloader.

Režim detekce

Aplikace pracuje ve čtyřech základních režimech.

- Real-time
- Batch
- On-demand
- Interval

Real-time režim

V tomto režimu se fotografie pořídí ihned, jakmile je detekován pohyb.

Fotografie je také ihned odeslána na úložiště. Volitelně je pak možné zapnout upozornění SMS zprávou na zadané telefonní číslo a tzv. „dozvuk“, kdy se bude pokračovat ve snímkování každou sekundu i po prvotní detekci pohybu po následujících deset sekund.

Režim má v JSON konfiguračním souboru pod parametrem *mode* přiřazenou hodnotu *realtime*.

Batch režim

Tento režim funguje obdobně jako režim Real-time. Snímky však nejsou odeslány ihned po detekci pohybu, ale pouze jednou denně ve zvolené době.

Volitelně je pak možné zapnout upozornění SMS zprávou na zadané telefonní číslo a tzv. „dozvuk“, kdy se bude pokračovat ve snímkování i po detekci pohybu v zadaném intervalu.

Režim má v JSON konfiguračním souboru pod parametrem *mode* přiřazenou hodnotu *batch*.

On-demand

On-demand, neboli na vyžádání, je režim, kdy se fotografie pořídí pouze na vyžádání pomocí SMS zprávy.

Režim má v JSON konfiguračním souboru pod parametrem *mode* přiřazenou hodnotu *demand*.

Interval

Tento režim umožňuje nastavení pravidelného snímkování v daném intervalu. Interval se nastavuje v minutách pomocí parametru *interval*.

Režim má v JSON konfiguračním souboru pod parametrem *mode* přiřazenou hodnotu *interval*.

Režim echo

Režim echo je volitelným parametrem pro nastavení „dozvuku“, tj. snímání dalších deseti fotografií po první detekci v sekundových intervalech.

Konfigurační soubor

Parametrem *mode* se nastavuje pracovní režim detekce, jsou povoleny čtyři hodnoty: *realtime*, *batch*, *demand* a *interval*.

Parametr *echo* je volitelným parametrem pro nastavení „dozvuku“, tj. snímání fotografií ve stanoveném intervalu po první detekci a uplatňuje se pouze pokud je zvolen režim real-time nebo batch.

Parametr *interval* nastavuje interval snímání pro režim interval. Je zadáván v sekundách.

Parametr *usePIR* udává zda se má k detekci pohybu použít i PIR senzor.

Ovládání a notifikace pomocí SMS zpráv je upravena parametry *SMSNotification*, *SMSControl* a *authorizedNumber*. Pole *authorizedNumber* obsahuje telefonní číslo, které může ovládat zařízení pomocí SMS příkazů a zároveň dostává notifikace.

K datovému typu string bylo přistoupeno z důvodu lepší serializace z HTML formulářů.

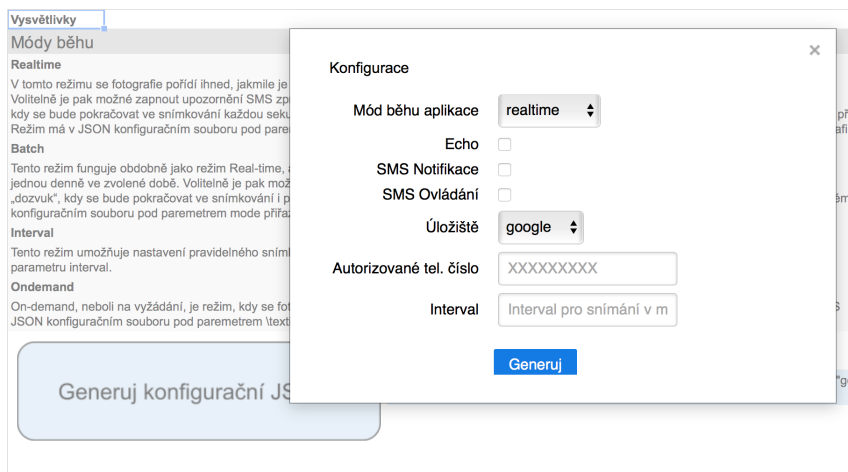
Tab. 2.2: Parametry konfiguračního souboru

Parametr	Datový typ	Popis
mode	string	Metoda detekce
echo	string	Použití dozvuku
interval	string	Interval
usePIR	string	aktivace PIR senzoru
SMSNotification	string	Notifikace pomocí SMS
SMSControl	string	Ovládání pomocí SMS
authorizedNumber	string	Autorizované číslo

```
{  
    "mode": "realtime",  
    "echo": "off",  
    "interval": "5",  
    "usePIR": "on",  
    "SMSNotification": "off",  
    "SMSControl": "on",  
    "authorizedNumber": "+420733122122"  
}
```

Rozhraní pro konfiguraci

Rozhraní pro konfiguraci je poskytnuto jako nadstavba nad Google Sheets vytvořená pomocí HTML a JavaScriptu. Samotná konfigurace se provádí v uživatelsky přívětivém formuláři. Po odeslání se pomocí JavaScriptu serializují data uložena ve formuláři do podoby JSON – JavaScript Object Notation a následně se uloží na Google Drive.



Obr. 2.5: Rozhraní pro konfiguraci

2.7 Sledování stavu nabíjení

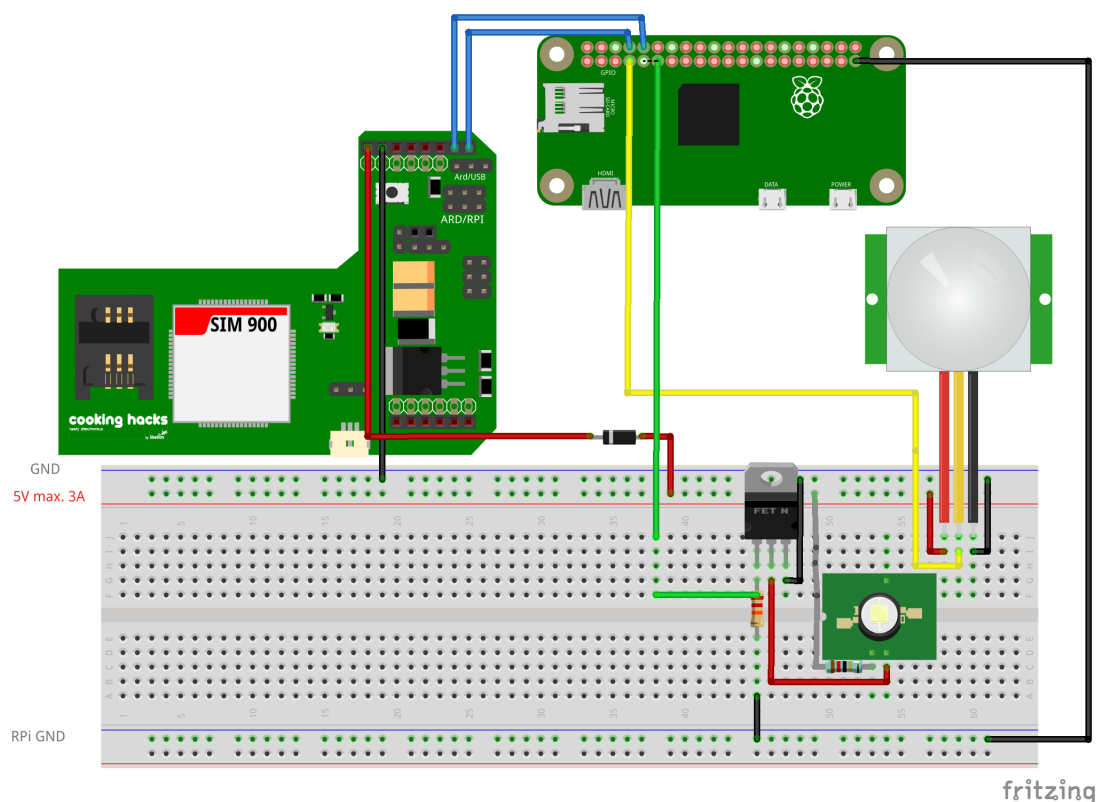
Vyčítání stavu akumulátoru a průběhu nabíjení ze solárního regulátoru Epsolar LS1024B je možné realizovat pomocí rozšiřujícího modulu epsolar-tracer, který umožňuje vyčítat následující informace: napětí baterie, napětí na solárním panelu, proud na zátěži, nabíjecí proud a teplotu baterie.

Komunikace je realizována pomocí protokolu Modbus.

3 NÁVRH ZAPOJENÍ

3.1 Prototypování

Před samotným návrhem výsledného zařízení, je vhodné nejprve komponenty sestavit na nepájivém poli abychom otestovali vzájemnou funkčnost a případně odstranili nedostatky.



Obr. 3.1: Prototyp zařízení sestavený na nepájivém poli

3.2 Řešení napájení

Spotřeba zařízení

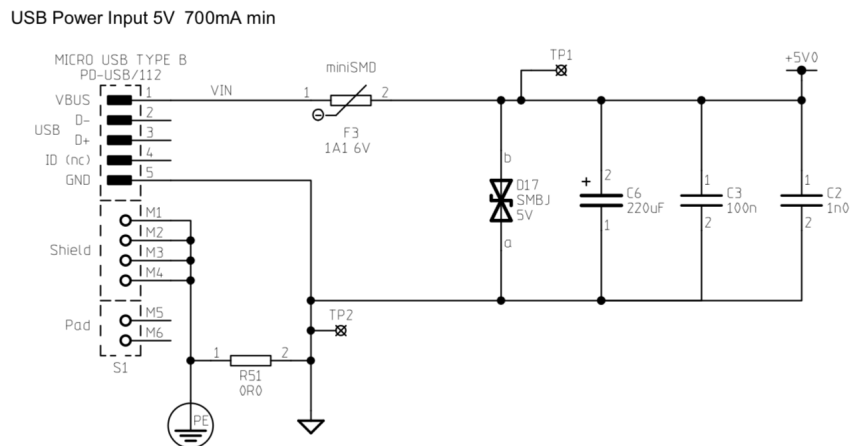
Před samotným návrhem napájecího obvodu je nutné nejprve zjistit proudový odběr jednotlivých komponent zařízení z katalogů.

Napájení Raspberry Pi Zero W

Špičkový proudový odběr samotného Raspberry Pi Zero W může dosahovat až 350 mA. Výrobce doporučuje použít zdroj, který je schopen dodávat 700 mA.

Raspberry Pi je možné napájet z USB portu, nebo přes GPIO 5 V pin. V případě GPIO je nutné zajistit stabilní napájení, protože pin je zapojen přímo na 5voltovou větev bez filtračního obvodu.

V našem případě můžeme Raspberry Pi napájet z GPIO, protože máme zajištěné stabilní napětí a ochranu z DC-DC měniče.



Obr. 3.2: Schéma napájení RPi přes USB, zdroj raspberrypi.org

Napájení modulu SIM800

Špičkový proudový odběr modulu SIM800 může dosahovat až 2 A při 5 V, zejména při registraci do sítě. Bude tedy nutné napájení dostatečně dimenzovat.

Tab. 3.1: Proudový odběr komponent

Proudový odběr @5V [mA]	<i>min.</i>	<i>typická</i>	<i>max.</i>
RPi Zero W	80	150	350
RPi Camera v2.1	100	100	?
SIM800	1,5	-	2000
PIR senzor	-	-	65
LED	-	400	420

Napájení výkonové LED

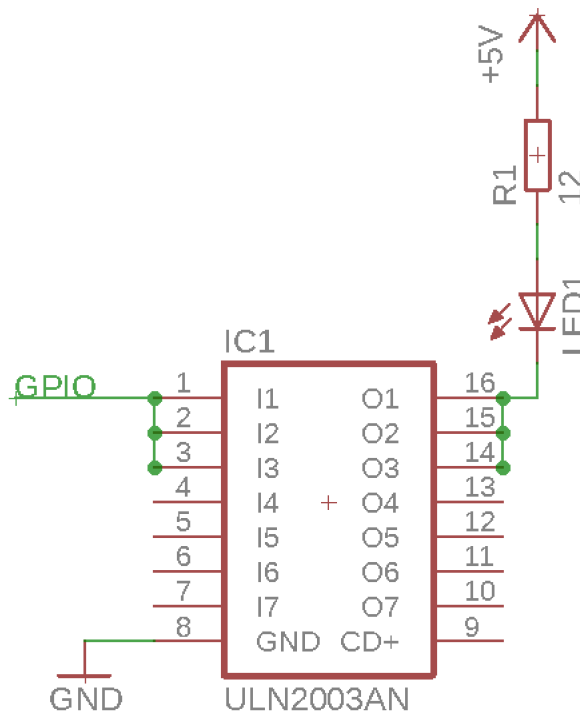
Použitá výkonová LED dioda má proudový odběr 400 mA. Takový proud není možné poskytnout přes GPIO, bude nutné diodu spínat tranzistorovým spínačem. V úvahu ještě připadá spínat diodu pulzně přes PWM, ale bylo by nutné vyřešit problém s

načasováním spouštění diody a pořizováním fotografií, aby nedocházelo ke stroboskopickému jevu.

Návrh spínače výkonové LED

Proudový odběr použité diody je 400 mA, úbytek napětí je 1,7 V. Diodu budeme spínat přes GPIO, které používá 3,3voltovou logiku. Prahové napětí pro sepnutí tranzistoru bude muset být minimálně 2,7 V, přičemž při napětím menším než 0,8 V musí být tranzistor zavřený. V úvahu připadá spínat LED diodu proudově řízeným bipolárním tranzistorem, nebo napětově řízeným tranzistorem typu MOSFET. U tranzistoru typu MOSFET dochází k menším energetickým ztrátám než u bipolárního tranzistoru, ale vzhledem k vyšší ceně a malé dostupnosti výkonových logických tranzistorů pro 3,3voltovou logiku bude lepší použít Darlingtonovo zapojení bipolárních tranzistorů. Takový tranzistor má velké proudové zesílení a malý vstupní proud.

V práci je použito tranzistorové pole ULN2003A sestávající ze sedmi Darlingtonových tranzistorů. K této součástce bylo přistoupeno z důvodu možného budoucího rozšíření o spínání více LED.



Obr. 3.3: Schéma spínače výkonové LED

Solární panel a akumulátor

Vzhledem k rozvaze o proudovém odběru jednotlivých komponent budeme vycházet z celkového průměrného proudového odběru 300 mA. Celkový příkon je tedy $0.3 \times 5 = 1.5$ W. Budeme uvažovat geografickou polohu Brna.

Solární panel

Dle Solargis.info dopadá v Brně 1150 kWh/m² solárního záření za rok.

$$1150000/8760 \approx 131 \text{ W/m}^2$$

Průměrně tedy v Brně dopadá 131 W/m², avšak efektivita solárních panelů se pohybuje kolem 15%, to je 20 W/m². Předpokládejme, že ztratíme dalších 33% při přenosu energie. Celkově tedy máme využitelnou energii pouhých 14 W/m².

Výpočet velikosti solárního panelu

$$\frac{1.5 \text{ W}}{14 \text{ W/m}^2} \approx 1071 \text{ cm}^2$$

Aby zařízení bylo možné provozovat celoročně, musíme provést výpočet velikosti solárního panelu pro měsíc s nejméně slunečnými dny, tedy pro prosinec. Využijeme kalkulátor Solar Electricity Handbook [8] pro zjištění úhrnu slunečního záření za měsíc prosinec. Bude tedy nutné velikost panelu přepočítat s ohledem na měsíc prosinec. V prosinci v Brně dopadá průměrně 1,41 kWh/m² za den. Obecně platí, že panel by měl být natočen směrem na jih pod úhlem 41 stupňů pro celoroční optimum efektivitu solárního panelu.

$$1410/24 \approx 59 \text{ W/m}^2$$

Po přepočtu efektivitu solárního panelu a ztráty při přenosu získáme 6 W/m²

Přepočet výpočtu velikosti solárního panelu pro měsíc prosinec

$$\frac{1.5 \text{ W}}{6 \text{ W/m}^2} \approx 2500 \text{ cm}^2$$

Pro celoroční napájení bude nutné zajistit panel velikosti alespoň 2500 cm²
Tomuto požadavku odpovídá solární panel WS-30/12V.

Akumulátor

Použitý akumulátor musí zajistit nepřetržité napájení zařízení po dobu dlouhých zimních nocí, zejména v měsíci prosinci a také po dobu špatného počasí. Budeme tedy předpokládat, že zařízení musí vydržet týden bez dobíjení.

S ohledem na nevýhody lithiových baterií uvedené v kapitole 1.5, bude zde vhodnější použít olověnou baterii.

Výpočítáme kapacitu akumulátoru pro noční provoz v prosinci, předpokládejme délku noci 16 h. Předpokládáme proudový odběr 300 mA při 5V.

$$16h \times 0.3A = 4.8 \text{ Ah}$$

Výpočet kapacity akumulátoru pro týden bez slunce. Předpokládáme proudový odběr 300 mA při 5V.

$$7dní \times 24 \text{ h} \times 0.3 \text{ A} \approx 50.4 \text{ Ah}$$

Mezi odpovídající komerčně nabízené akumulátory patří Westinghouse WA12200 12V/20Ah.

DC-DC Měnič 12/5V

Pro změnu 12 V ze vstupu akumulátoru na 5 V, které je vyžadováno pro napájení všech komponent zařízení, je použit DC-DC měnič CPT 120503 12/5V od firmy Current Logic. Maximální udávaný proudový odběr je 3 A. Udávaná účinnost je v rozmezí 90 až 94 procent. Měnič také obsahuje ochranu proti zkratu a také ochranu proti přetížení.

Solární regulátor Epsolar LS1024B

Solární regulátor řídí nabíjení akumulátoru ze solárního panelu a také chrání baterii před jejím přebitím i hlubokým vybitím. Existují dva typy solárních regulátorů - MPPT a PWM. PWM solární regulátory vyhovují svými parametry pro naši aplikaci, používají pulzně šířkovou modulaci k dosažení optimálního nabití baterie.

Požadavkem na solární regulátor je schopnost dodávat proud minimálně 3 A při 5 V, možnost připojení 12 V olověné baterie a 12 V solárního panelu. Dalším požadavkem je možnost vyčítání dat o stavu nabití baterie a nízká pořizovací cena.

Tyto požadavky splňuje PWM solární regulátor LS1024B od firmy Epsolar, který umožňuje odebírat až 10A ze solárního panelu a dodávat 10A do zátěže. Je také umožněna datová komunikace přes protokol Modbus, realizovaná sériovou sběrnicí RS-485.

Komunikace mezi Raspberry Pi a solárním regulátorem je možné realizovat knihovnou epsolar-tracer pomocí převodníku RS-485 na USB.

3.3 Komunikace s GSM modulem SIM800

Komunikace s GSM modulem je realizována na rozhraní UART. Raspberry Pi disponuje jednou hardwarovou periferií UART, která je dostupná na pinech 14 (TX) a

15 (RX). Modul SIM800 podporuje maximální bitovou přenosovou rychlost 460800 bit/s, ale při použití multiplexovacího protokolu GSM0710 je výrobcem doporučená maximální přenosová rychlost 115200 bit/s.

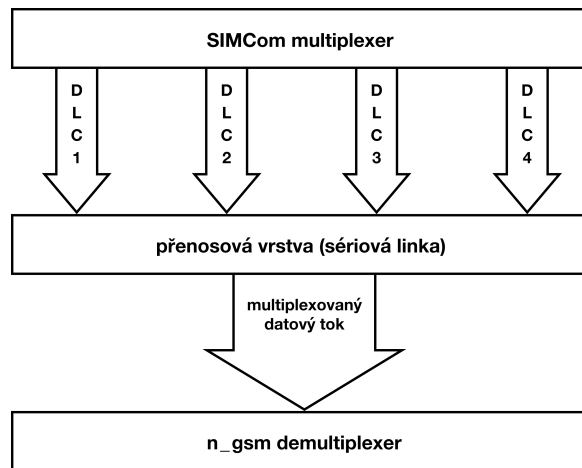
Multiplexovací protkol GSM0710

Modul SIM800 disponuje pouze jedním hardwarovým rozhraním UART, to znamená, že standardně můžeme otevřít pouze jedno spojení. V této aplikaci, ale bude nutné navázat minimálně dvě spojení – pro PPP a druhé spojení pro posílání a přijímání SMS zpráv.

3GPP definuje standard 27.010 (GSM0710), který umožňuje provádět multiplexování na přenosové vrstvě, které rozdělí datové toky do takzvaných DLC (Data Link Channel). Takový logický kanál je možné používat separátně, je tedy možné navázat PPP připojení na jednom kanálu a na druhém kanálu naslouchat zda-li nepřišla zpráva SMS.

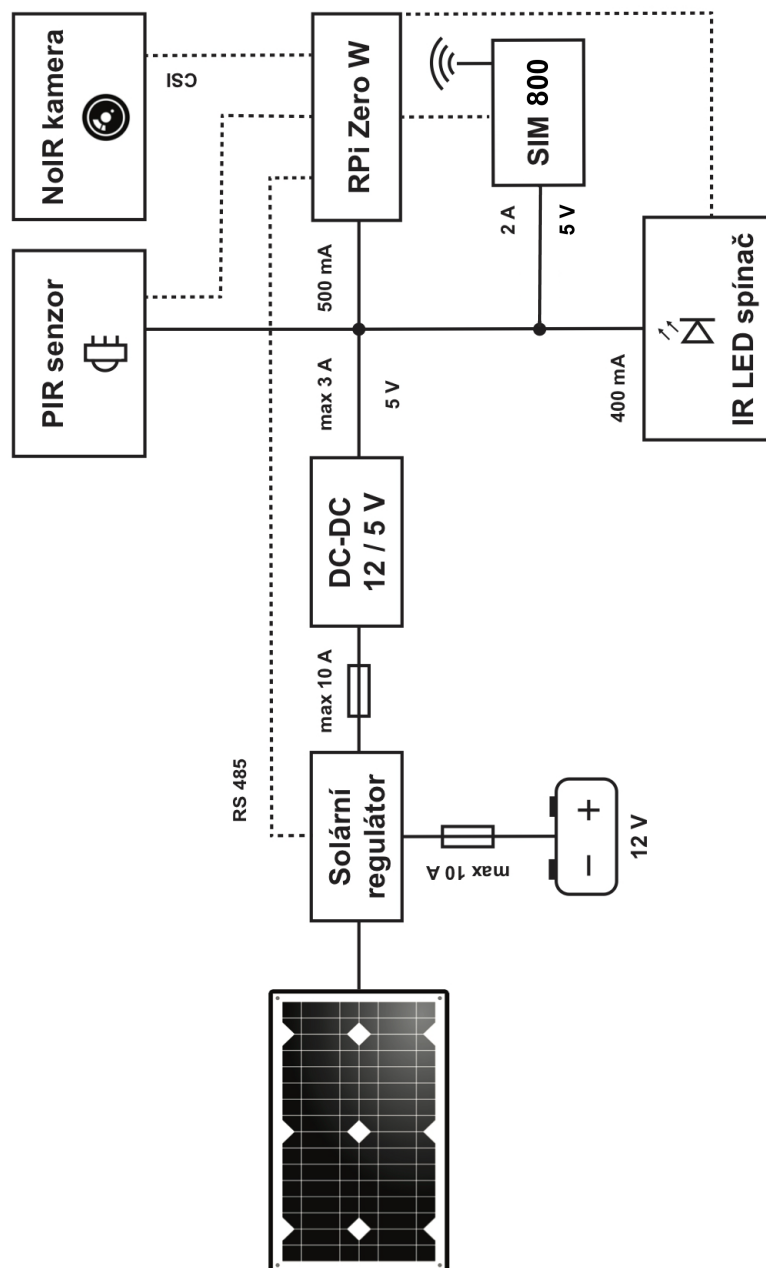
Podpora pro multiplexovací protokol GSM0710 standardně v Linuxu není, ale je možné ji zkompilovat a přidat do kernelu v podobě LKM (Loadable Kernel Module). V repozitáři Linux kernelu ji najdeme v podobě zdrojového kódu jazyka C – `n_gsm.c`. Po načtení modulu do kernelu je do adresáře `proc` přidán soubor zařízení `gsmtty`, který ale ještě není možný použít jako terminál.

Abychom mohli zařízení používat jako terminál je nutné připojit takzvanou „line discipline“, toto je řešeno přes program `cmux` (autorem je Nicolas Le Munchet), který byl upraven dle specifikací modulu SIM800.



Obr. 3.4: Schéma multiplexovacího protokolu GSM0710

3.4 Blokové schéma návrhu zařízení



Obr. 3.5: Blokové schéma návrhu zařízení

3.5 Seznam a cena komponent

Tab. 3.2: Seznam komponent

Jméno komponenty	Cena včetně DPH
Raspberry Pi Zero W	313 Kč (rpishop.cz)
Raspberry Pi NoIR Camera Board V2	858 Kč (rpishop.cz)
SanDisk microSD karta 16 GB class 10	229 Kč (czc.cz)
GSM/GPRS modul SIM800	250 Kč
PIR modul HC-SR501	110 Kč (gme.cz)
Výkonová IR LED dioda GT-P04IR4101	45 Kč (gme.cz)
Akumulátor Westinghouse WA12200	1290 Kč (gme.cz)
Solární panel WS-30/12V	1990 Kč (gme.cz)
Solární regulátor Epsolar LS1024B	739 Kč (gme.cz)
Převodník USB na RS-485	240 Kč (rpishop.cz)
DC/DC měnič 12V/5V CPT	157 Kč
ULN20003A	10 Kč (gme.cz)
Rezistor 22k	2,60 Kč (gme.cz)
Drátový rezistor RD 12R 5W	5,30 Kč (gme.cz)

Cena komponent je platná k prosinci 2017.

4 REALIZACE A TESTOVÁNÍ

Praktickým výstupem práce je samotná realizace a otestování zařízení bez solárního napájení. Vybrané komponenty byly umístěny do plastové konstrukční krabičky, která byla upravena s ohledem na rozmístění jednotlivých komponent.

Na přední straně krabičky je čočka PIR senzoru, infračervená LED dioda a kamera. Na vrchní straně se nachází anténa pro GSM modul.

Na pravém boku je 2,1 mm napájecí konektor. Je možné připojit akumulátor nebo stejnosměrný napájecí zdroj. Tolerované napájecí napětí je 8–20 V.

Dále se zde nachází tlačítko pro zapnutí či vypnutí zařízení. Vypínání je řešeno softwarově, takže po stisku nedochází k úplnému vypnutí zařízení ale pouze k přepnutí do režimu halt.



Obr. 4.1: Realizace kamerového zabezpečovacího systému

4.1 Měření spotřeby zařízení

Spotřeba zařízení byla měřena pomocí multimetru UNI-T UT70A před vstupem do DC-DC měniče.

Tab. 4.1: Měření spotřeby

měřená veličina	stav „idle“	běžící aplikace	stav „halt“
napájecí napětí [V]	12	12	12
typický proudový odběr [mA]	80	110	30
špičkový proudový odběr [mA]	130	210	50
typický příkon [W]	0,96	1,32	0,36

4.2 Zdrojový kód aplikace

Aplikace je vydána jako balíček pro Python 3 pod názvem „rpicameramon“. Balíček obsahuje pět modulů.

Zdrojové kódy aplikace jsou spolu s dokumentací a skripty zveřejněny na serveru GitHub.

<https://github.com/marekvi95/bakalarka>

```
/
├── rpicameramon ..... hlavní balíček
│   ├── __init__.py
│   ├── motion.py
│   ├── config.py
│   ├── filemanipulation.py
│   └── telemetry.py
├── google ..... skripty pro google API
│   └── scripts
│       ├── main.gs
│       └── index.html
├── init ..... spouštění po startu
│   ├── listen-shutdown.py
│   └── listen-shutdown.sh
├── docs ..... dokumentace aplikace
├── gsm ..... komunikace s gsm modulem
│   ├── ppp
│   ├── chatscripts
│   ├── cmux
│   └── n_gsm
└── latex ..... zdrojové kódy bakalářské práce
```


5 ZÁVĚR

V práci byla prozkoumána možná řešení a byl zpracován návrh kamerového zabezpečovacího systému založeného na minipočítači Raspberry Pi.

V teoretické části práce byly vybrány hardwarové komponenty vhodné pro aplikaci na detekci pohybu s ohledem na co nejnižší spotřebu elektrické energie. Při výběru softwaru byl dáván důraz na jeho aktuálnost, otevřenost a rozšiřitelnost. Z těchto důvodů byl pro práci vybrán jako hlavní programovací jazyk Python.

V druhé kapitole je popsán návrh samotné aplikace pro detekci z obrazu a též za pomoci PIR senzoru. Důraz je kladen na konfigurovatelnost aplikace.

Třetí kapitola se věnuje návrhu napájení jednotlivých komponent pomocí solárního panelu a akumulátoru.

Poslední kapitola se věnuje realizaci zařízení. Realizačním výstupem je funkční sestava minipočítače, kamery, PIR senzoru, IR LED přísvitů a GSM/GPRS modulu včetně napájecího modulu připraveného k připojení akumulátoru či zdroje.

Bylo ověřeno, že aplikace běžící v zařízení umožňuje detekovat pohyb metodou počítání vektorů pohybu a také pomocí PIR senzoru v rámci omezení úhlu záběru kamery, citlivosti PIR senzoru a dosahu LED přísvitů. Po detekci pohybu je pořízena fotografie, která je zkomprimována a následně odeslána do externího úložiště pomocí GSM/GPRS modulu, který zajišťuje konektivitu do sítě Internet. Na externím úložišti běží aplikace, která odešle upozornění o detekci pohybu na zvolenou mailovou adresu. Aplikace na externím úložišti umožňuje zobrazení fotografií a také umožňuje zařízení konfigurovat a číst zprávy o chodu aplikace. Zařízení je také možno ovládat pomocí definovaných SMS zpráv.

5.1 Možná vylepšení

Praktická realizace zařízení ukázala nedostatky při použití minipočítače Raspberry Pi. Vzhledem k požadavku na bateriové napájení se jako vhodnější jeví použití mikrokontroléru s podporou uspávání, které by pomohlo ušetřit energii v době neaktivity zařízení.

Za zvážení by také stálo použití kamery s lepším objektivem, který by umožnil pořizovat kvalitnější fotografie.

Pro připojení k síti Internet by bylo vhodnější použít modul podporující síť čtvrté generace. Nástupce modelu SIM800 je SIM7100 od firmy Simcomm, který podporuje i nový standard úzkopásmového LTE NB-IoT.

Co se týká aplikace pro detekci, tak jako možné vylepšení se nabízí lokalizace a klasifikace objektu, tak aby bylo možné rozeznávat, zda se jedná o člověka nebo o zvíře. Případná další vylepšení by mohla přinést aplikace neuronových sítí.

LITERATURA

- [1] Raspberry Pi Foundation: *Oficiální dokumentace k Raspberry Pi* [online]. Dostupné z URL: [<https://www.raspberrypi.org/documentation/>](https://www.raspberrypi.org/documentation/).
- [2] Camera Python interface documentation: *Oficiální dokumentace k Picamera* [online]. Dostupné z URL: <http://picamera.readthedocs.io>.
- [3] BECHNÍK B. *Stručná historie fotovoltaiiky* [online]. 2014, poslední aktualizace 1.9.2014 [cit. 18.11.2017]. Dostupné z URL: <http://oze.tzb-info.cz/fotovoltaiika/11652-strucna-historie-fotovoltaiiky>
- [4] ČVUT Praha - Fakulta biomedicínského inženýrství *Detekce pohybu ve videu* [online]. [cit. 1.10.2017]. Dostupné z URL: <http://fbmi.cvut.cz/files/predmety/3528/public/Detekce%20pohybu%20ve%20videu.pdf>
- [5] MICHALEC L. *PIR detektor: skvělý sluha, ale zlý pán* [online]. 2013, poslední aktualizace 18.3.2013 [cit. 1.10.2017]. Dostupné z URL: <https://vyvoj.hw.cz/automatizace/pir-cidlo-skvely-sluha-ale-zly-pan.html>
- [6] Solargis *Mapy globálního horizontálního solárního záření pro Českou republiku* [online]. Dostupné z URL: <https://solargis.com/products/maps-and-gis-data/free/download/czech-republic>.
- [7] HUFKENS K. *Spektrální citlivost Raspberry Pi kamery* [online]. 2016, poslední aktualizace 5.6.2016 [cit. 1.10.2017]. Dostupné z URL: <http://www.khufkens.com/2016/06/05/raspberry-pi-camera-v2-spectral-response-curve/>
- [8] BOXWELL M. *Tabulky solárního záření pro danou lokaci* [online]. 2017, [cit. 18.11.2017]. Dostupné z URL: <http://www.solarelectricityhandbook.com/solar-irradiance.html>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

IoT	Internet of Things, Internet věcí
SoC	System on a chip
HDMI	Rozhraní pro přenos videa a zvuku
FPS	Počet snímků za sekundu
GPIO	General Purpose Input Output
UART	Universal asynchronous receiver-transmitter
SPI	Serial Peripheral Interface Bus
I2C	Inter-Integrated Circuit
PCM	Pulzně kódová modulace
PWM	Pulzně šířková modulace
DMA	Direct Memory Access, tj. přímý přístup do paměti
NoIR	No Infrared, označení pro kameru bez IR filtru
CMOS	technologie výroby integrovaných obvodů
GSM	standard pro mobilní hlasovou komunikaci
GPRS	rozšíření GSM o paketová data
PPP	Point-to-Point Protocol
APN	Access Point Name
PIR	Passive Infrared Sensor
PIL	Python Imaging Library
API	Application Programmable Interface
BSD	Berkeley Software Distribution
JSON	JavaScript Object Notation
FET	Field-effect transistor
FPGA	Field-programmable gate array

SEZNAM PŘÍLOH

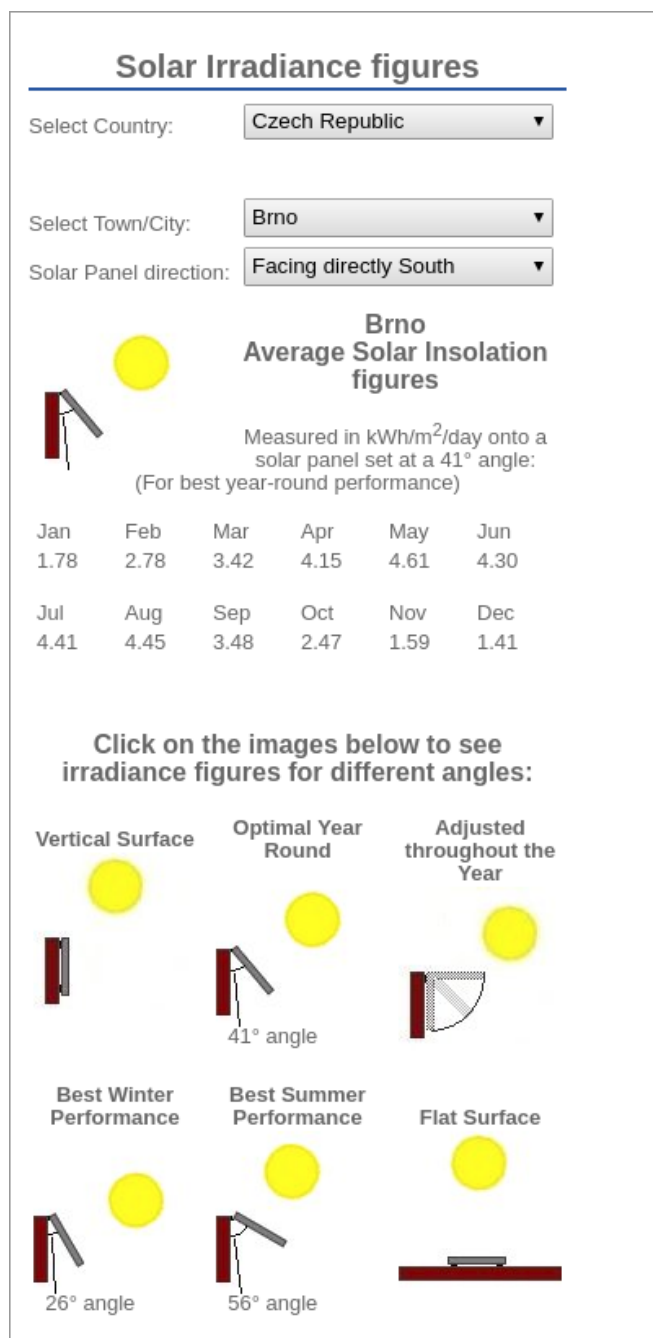
A	Zapojení GPIO pinů na Raspberry Pi	52
B	Denní úhrn solárního záření v Brně	53
C	Ukázkové fotografie pořízené kamerou	54
D	Dokumentace aplikace	55

A ZAPOJENÍ GPIO PINŮ NA RASPBERRY PI

Tab. A.1: Zapojení GPIO pinů na Raspberry Pi

zapojení	název	BCM	pin	pin	BCM	název	zapojení
-	3v3	-	1	2	-	5V	Napájení
-	SDA	2	3	4	-	5V	-
Tlačítko ON/OFF	SCL	3	5	6	-	GND	GND
SIM800 RST	GPCLK0	4	7	8	14	TXD	SIM800 Rx
-	GND	-	9	10	15	RXD	SIM800 Tx
-		17	11	12	18	PWM0	PIR signál
-		27	13	14	-	GND	-
-		22	15	16	23		LED signál
-	3v3	-	17	18	24		-
-	MOSI	10	19	20	-	GND	-
-	MISO	9	21	22	25		-
-	SCLK	11	23	24	8	CE0	-
-	GND	-	25	26	7	CE1	-
-	ID_SD	0	27	28	1	ID_SC	-
-		5	29	30	-	GND	-
-		6	31	32	12	PWM0	-
-	PWM1	13	33	34	-	GND	-
-	MISO	19	35	36	16		-
-		26	37	38	20	MOSI	-
-	GND	-	39	40	21	SCLK	-

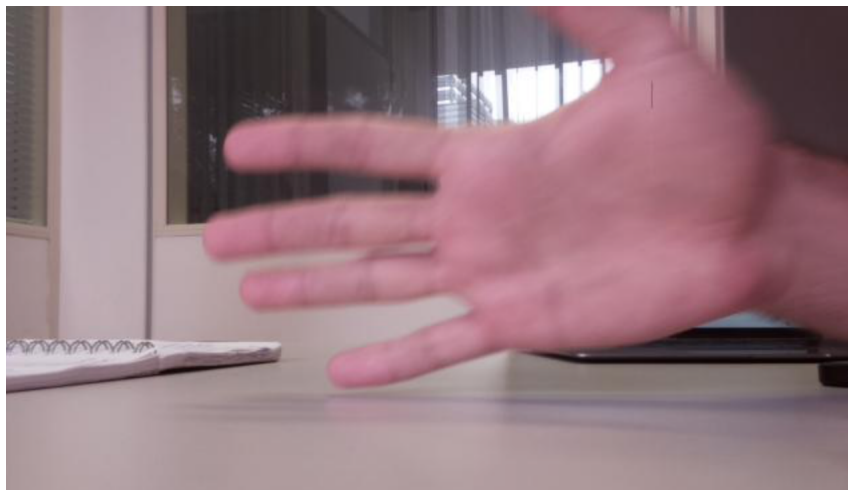
B DENNÍ ÚHRN SOLÁRNÍHO ZÁŘENÍ V BRNĚ



Obr. B.1: Solar Irradiance Brno [8]

C UKÁZKOVÉ FOTOGRAFIE POŘÍZENÉ KAMEROU

Fotografie jsou pořízeny v rozlišení 640x368 pixelů a zkomprimovány pomocí knihovny Pillow ve formátu JPEG v kvalitě 75 s dodatečnou optimalizací.



Obr. C.1: Ukázka denní fotografie pořízené kamerou



Obr. C.2: Ukázka noční fotografie pořízené kamerou s IR přísvitem

D DOKUMENTACE APLIKACE

K dokumentaci aplikace je použit generátor dokumentace Sphinx. Dokumentace ve zdrojovém kódu je realizována pomocí konvence Google docstring.

Dokumentace je napsaná v anglickém jazyce. Na následujících stránkách je přiložena dokumentace vygenerovaná pro \LaTeX . Dokumentace v HTML je dostupná v GitHub repozitáři.

Camera Security System with Raspberry Pi Minicomputer Documentation

Release 0.1

Marek Vitula

May 21, 2018

CONTENTS

1	config module	3
2	filemanipulation module	5
3	motion module	7
4	telemetry module	9

This package provides camera security system functionality for the Raspberry Pi minicomputer. It uses Raspberry Pi NoIR camera for the motion detection and also PIR sensor if available.

Capturing is handled by the picamera package and image processing is handled by the Pillow package.

Captured images are sent to the Google Drive storage.

Package itself consist of four submodules - config, filemanipulation, motion and telemetry. Each of them has several classes.

CONFIG MODULE

Config modules is used for storing of the application configuration internally. It consist of two classes - BaseConfig and UserConfig. UserConfig inherits from the BaseConfig class.

UserConfig class adds variables which can be loaded from the external JSON file.

class `rpicameramon.config.BaseConfig`

Bases: `object`

Basic configuration which cannot be changed online

imageDir

str – default image directory

imagePath

str – path to the directory + imageDir

imageWidth

int – width of taken pictures in pixels

imageHeight

int – height of taken pictures in pixels

imageVFlip

bool – flips image vertically if true

imageHFlip

bool – flips image Horizontally if true

imagePreview

bool – opens window with picture if true

imageQuality

int – JPEG quality 0...100

vecMagnitude

int – minimum magnitude of a motion vector

vecCount

int – minimal number of motion vectors

cameraFPS

int – framerate of the camera

confCheckTime

int – how long does it take between conf file checks in sec

PIRpin

int – GPIO pin (BCM) where is PIR sensor connected

LEDpin

int – GPIO pin (BCM) where is LED connected

pictureFolderID

str – Google Drive ID of folder with pictures

confFileID

str – Google Drive ID of JSON configuration files

dashboardFileID

str – Google Drive ID of dashboard panel

logRange

str – Range in sheets for telemetry messages

msgRange

str – Range in sheets for log messages

OAuthJSON

str – Name of the JSON file for Google OAuth authentication

fileUploadSleep

int – Sleep time in seconds for file uploader thread

batchUploadWindow

int – Allowed hour for file uploads in batch mode

GSMport

str – Device address of a GSM Module

GSMbaud

int – Baud rate for communication with the GSM module

GSMPIN

int – PIN for unlocking the SIM card

class `rpicameramon.config.UserConfig`

Bases: `rpicameramon.config.BaseConfig`

Extends BaseConfig class with the user defined configuration. This configuration can be updated from the external JSON.

mode

str – default runtime mode (realtime, interval, batch, ondemand)

echo

boolean – echo mode

interval

int – default interval time in seconds for interval mode

storage

str – gdrive or dropbox. Dropbox is only experimental

usePIR

boolean – use PIR sensor

SMSNotification

boolean – SMS SMSNotification

SMSControl

boolean – Allow control via SMS messages

authorizedNumber

int – Authorized number for control and notifications

classmethod `load_config` (*conf*)

This method loads configuration to UserConfig class from the JSON file.

Raises `KeyError` – If JSON cannot be parsed

FILEMANIPULATION MODULE

This module consist of classes for the manipulation with files in Google Drive.

```
class rpicameramon.filemanipulation.ConfFileDownloader (group=None, target=None,  
get=None, name=None,  
filename=None, *, dae-  
mon=None)
```

Bases: `threading.Thread`

`ConfFileDownloader` is a subclass of a `Thread` class First it loads the initial JSON configuration from Google drive and then it check if the configuration was changed. If yes, it loads it.

filename

string – file ID of JSON configuration file on Google drive

run ()

This method runs when `ConfFileDownloader` thread is started. First it initialize connection with Google Drive and starts loop for checking if JSON configuration file on Google Drive was modified. If yes, It downloads it and calls the method for loading it into `UserConfig` class.

File is checked in intervals specified in `confCheckTime` variable.

auth ()

Function for Google Drive authentication. It loads credentials from the file “mycreds.txt”.

Returns Drive object

Return type drive

download_file (drive)

This function downloads specified file from the Google Drive.

Parameters **drive** (*obj*) – Google Drive object

Returns Content of the file as string

Return type str

get_timestamp (drive)

This function fetches date of the last modification of specified file.

Returns Timestamp of last modification of the file

Return type timestamp

parse_json (jsonfile)

load_config (configdata)

```
class rpicameramon.filemanipulation.FileUploader (group=None, target=None,  
name=None, storage=None, *,  
daemon=None, q=None)
```

Bases: `threading.Thread`

This class provides a File uploader thread for uploading captured photos. It supports google drive and dropbox (experimental)

storage

str – storage type (gdrive or dropbox)

q

Queue obj – queue with filenames of pictures to be uploaded

dropbox_is_init

boolean – True if Dropbox is initialized

gdrive_is_init

boolean – True if Google Drive is initialized

run ()**gdrive_init ()****gdrive_upload** (*filename, drive*)**dropbox_init ()****dropbox_upload** (*filename, dbx*)

`rpicameramon.filemanipulation.stopwatch` (*message*)

MOTION MODULE

This module has functionality for the motion detection from the scene and also from the PIR sensor and also SMS handler. Optical flow method is used for the motion detection.

class `picameramon.motion.MotionAnalysis` (*camera, handler*)

Bases: `picamera.array.PiMotionAnalysis`

MotionAnalysis class extends PiMotionAnalysis class.

The array passed to `analyse()` is organized as (rows, columns) where rows and columns are the number of rows and columns of macro- blocks (16x16 pixel blocks) in the original frames. There is always one extra column of macro-blocks present in motion vector data.

analyse (*a*)

class `picameramon.motion.PIRMotionAnalysis` (*pin, handler*)

Bases: `object`

This class provides a handler for PIR sensor It detects if the input pin is in the high state and calls `motion_detected` handler

Parameters

- **pin** (*number*) – BCM number of GPIO pin where is PIR sensor connected
- **handler** (*obj*) – CaptureHandler object

is_detected ()

class `picameramon.motion.CaptureHandler` (*camera, post_capture_callback=None, q=None*)

Bases: `object`

It provides a handler for capturing the pictures. With the LED Switch functionality.

camera

obj – PiCamera object

callback

str – callback

q

obj – queue for passing captured photos

detected

bool – True if motion is detected

working

bool – True if the picture is saving

i

int – counter of captured photos

echoCounter

int – counter of taken pictures with echo mode

motion_detected()

tick()

This tick method provides a handler for capturing the pictures. It ticks every second after PiMotion.start() was called. If detected is True and method is not processing any capture (That is indicated by variable 'working'), it begins with processing. First, datetime method is called to obtain the actual datetime, then the scene is analyzed with scan_day method which returns true if light conditions appear to be daylight or false if the light level is too low.

If the echo mode is activated

For daylight: exposure compensation is set to 0 <-25;25> exposure mode is set to auto camera shutter speed is set to 0 (auto mode)

For bad light conditions: exposure compensation is set to 25 for maximum brightness exposure mode is set to night preview shutter speed is set to 200000 microseconds which is equal to 0,2s
Turn on the LED

scan_day()

This method captures picture as an RGB array and calculates average value of the pixels in the matrix. If the value pixAverage is more than 50 (scene is gray to black). It indicates that the light condition is poor and we can set up night params.

class rpicameramon.motion.**SMSHandler** (*handler*)

Bases: object

handle_sms (*sms*)

class rpicameramon.motion.**PiMotion** (*post_capture_callback=None, q=None*)

Bases: object

This class handles the camera and PIR sensor setup. It sets up resolution and framerate and starts capturing the video outputting it to the MotionAnalysis object.

Parameters

- **post_capture_callback** (*callback*) – not in use
- **q** (*Queue obj*) – queue for captured photos

Raises KeyboardInterrupt – Interrupt the program

start()

TELEMETRY MODULE

This module provides functionality for sending telemetry data to the Google Sheets.

class `rpicameramon.telemetry.GoogleHandler`

Bases: `object`

get_credentials ()

Gets valid user credentials from storage.

If nothing has been stored, or if the stored credentials are invalid, the OAuth2 flow is completed to obtain the new credentials.

Returns Credentials, the obtained credential.

get_sheets_service (*credentials*)

Gets Google Sheets Service v4

Returns Google Sheets service object

get_file_service (*credentials*)

Gets Google Drive Service v3

Returns: Google Drive service object

add_sheet_line (*service=None, line=None, spreadsheetId=None, rangeName=None*)

Append a line/lines to Google Sheet.

Parameters

- **service** (*obj*) – Google Sheets Service object
- **line** (*nested list*) – matrix of data to be put into Google Sheet
- **spreadsheetId** (*str*) – Id of spreadsheet
- **rangeName** (*str*) – Range in a spreadsheet

upload_file (*service, filename*)

Uploads a image/jpeg file to the Google Drive.

Parameters

- **service** (*obj*) – Google Drive Service object
- **filename** (*str*) – full path to the file.

class `rpicameramon.telemetry.LogSender` (*logQueue, googleHandler*)

Bases: `threading.Thread`

`LogSender` class is a subclass of a `Thread` class. It provides a functionality to send a captured log to Google Sheets. It should be used together with the `logging.handlers.QueueHandler` class as a handler for logging. `LogSender` works only in specified time intervals, dequeues everything in `logQueue`, sends it to the Google Sheets and goes sleep for the amount of time specified by the variable `fileUploadSleep`

logQueue

obj – `queue.Queue` object with `logRecord` (*obj*)

enqueued

googleHandler

obj – GoogleHandler instance object

run ()

class `rpicameramon.telemetry.TelemetrySender` (*googleHandler*)

Bases: `threading.Thread`

Telemetry class is a subclass of a Thread class. It provides a functionality for sending a telemetric data to Google Sheets.

Attributes:

run ()

getTelemetry ()