

Bakalárska práca



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra měření

System pre vzdialené monitorovanie procesov v laboratóriu

Jozef Dujava

Školiteľ: Ing. Vojtěch Petrucha, Ph.D.
Máj 2019

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Dujava** Jméno: **Jozef** Osobní číslo: **466177**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra měření**
Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Systém pro vzdálené monitorování procesů v laboratoři

Název bakalářské práce anglicky:

System for Remote Monitoring of Laboratory Processes

Pokyny pro vypracování:

- S využitím platformy 32F746G-DISCOVERY realizujte jednoduchý dohledový systém, který bude dovolovat vzdálený provoz a monitoring žihacích procesů.
- Systém bude umožňovat přenos obrazu z kamery a naměřených dat ze senzorů (teplota, průtok, proud, apod.).
- Systém bude disponovat i galvanicky oddělenými vstupy a výstupy a rozhraním RS232 pro komunikaci s PID regulátory teploty.
- Data budou prezentována formou www stránky nebo uživatelské aplikace, přenos dat bude vhodným způsobem zabezpečen.

Seznam doporučené literatury:

- [1] Steve Heath, S.: Embedded Systems Design, Newnes 2003
- [2] Fraden, J.: Handbook of Modern Sensors, Springer 2004
- [3] Montrose, Mark I.: EMC and the Printed Circuit Board: Design, Theory, and Layout Made Simple, IEEE 1999
- [4] Záhlava, V.: Návrh a konstrukce DPS, Ben 2010

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Vojtěch Petrucha, Ph.D., 13138

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **30.01.2019**

Termín odevzdání bakalářské práce: **24.05.2019**

Platnost zadání bakalářské práce:

do konce letního semestru 2019/2020

Ing. Vojtěch Petrucha, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Podakovanie

Chcel by som sa poďakovať môjmu školiteľovi, Ing. Vojtěchovi Petruchovi, Ph.D., za jeho cenné poznatky, rady a pripomienky, ktoré boli veľmi nápomocné pri tvorbe tejto práce.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval samostatne a že som uviedol všetky použité informačné zdroje v súlade s Metodickým pokynom o dodržovaní etických princípov pri príprave vysokoškolských záverečných prác.

V Prahe, 22. mája 2019

Abstrakt

V tejto bakalárskej práci bol navrhnutý elektronický monitorovací systém pre sledovanie žihacích procesov. Tie sú používané pri vývoji magnetických fluxgate senzorov v laboratóriu MagLab katedry merania. Tento systém meria hodnoty sledovaných fyzikálnych veličín (teplota, el. prúd, prietok, atď.) a taktiež zaznamenáva obraz z kamery. Vzdialený prístup k systému je umožnený pomocou internetového pripojenia a klientskej aplikácie vo forme webovej stránky. Toto rozhranie taktiež poskytuje možnosť vzdialeného ovládania niektorých častí žihacieho procesu. Systém je založený na vývojovom kite STM32-Discovery, ku ktorému bola vyrobená rozširujúca doska pre podporu jednotlivých vstupov (senzorov) a výstupov. V ďalšom texte sa nachádza bližší popis danej problematiky, ako aj popis samotného návrhu hardvéru a funkcie softvéru. Následne sú taktiež popísané testovacie procedúry a získané výsledky pre jednotlivé časti systému. Záver obsahuje zhodnotenie konečného výsledku práce a prehľad možných budúcich vylepšení systému.

Kľúčové slová: vzdialené monitorovanie, laboratórne procesy, STM32-Discovery

Abstract

In this thesis, an electronic monitoring system has been developed to oversee annealing processes. These processes are part of the development of new fluxgate magnetic sensors in the MagLab laboratory of the Department of Measurement. The monitoring system measures the values of selected physical quantities (temperature, electrical current, liquid flow rate, etc.) and also captures images using an attached camera. Remote access to the system is available using the Internet and a client application in the form of a website. This interface also allows the user to control certain parts of the annealing process. The system is based on an STM32-Discovery development board, for which an expansion board has also been developed. This expansion board provides additional circuitry and connectors to support various additional inputs (sensors) and outputs. Further text provides a more detailed insight into the subject matter, as well as a description of the hardware and software of the system. Afterwards, the testing procedures and results are described for each of its various parts. In the conclusion, the end result is evaluated and possible future improvements are proposed.

Keywords: remote monitoring, laboratory processes, STM32-Discovery

Title translation: System for Remote Monitoring of Laboratory Processes

Obsah

1 Úvod	1
1.1 Motivácia	1
1.2 Požiadavky na systém	2
1.3 Možné riešenia	3
2 Navrhnutý systém	5
2.1 Vývojový kit STM	6
2.2 Kamera	8
3 Rozširujúca doska	9
3.1 Napájanie	11
3.2 Digitálne vstupy a výstupy	12
3.3 Senzory teploty	13
3.4 Senzor prúdu	16
3.5 Senzor prietoku	17
3.6 Senzor pohybu	18
3.7 PID regulátor	20
4 Firmvér serveru	21
4.1 Použité softvérové prostriedky	21
4.2 Spôsob funkcie	22
4.3 HTTP server	23
4.3.1 Použitá technológia	23
4.3.2 Spracovanie požiadaviek	24
4.4 Komunikácia so vstupmi/výstupmi	26
4.4.1 Digitálne vstupy/výstupy	27
4.4.2 Senzory teploty	28
4.4.3 Senzor prúdu	30
4.4.4 Senzor pohybu	30
4.4.5 PID regulátor	31
4.4.6 Kamera	32
5 Klientska aplikácia	35
5.1 Používateľské rozhranie	36
5.2 Prijímanie dát zo senzorov	39
5.3 Nastavovanie výstupov	40
5.4 Prijímanie obrazu z kamery	40
6 Testovanie systému	41
6.1 PID regulátor	41
6.2 Senzor prúdu	41
6.3 Senzor teploty	44
7 Záver	45
Literatúra	47

Obrázky

1.1 Elektrická piecka používaná na žíhanie materiálov	2	4.5 Časový priebeh čítania z registra MAX31865 [5, s. 18]	29
2.1 Diagram navrhnutého monitorovacieho systému [1]	5	4.6 Konverzia riadku obrázka z formátu RGB-565 do RGB-888	33
2.2 Vývojový kit STM32F746NG-DISCOVERY [1]	7	4.7 Diagram spracovania snímok z kamery	34
2.3 Zapojenie Arduino konektora vývojového kitu STM [2]	8	5.1 Klientska aplikácia (graf)	35
2.4 Prídavný kit STM32F4DIS-CAM s kamerou	8	5.2 Klientska aplikácia (kamera)	36
3.1 Zapojenie Arduino konektora rozširujúcej dosky	9	5.3 Menu vstupov/výstupov v klientskej aplikácii	37
3.2 Osadená rozširujúca doska	10	5.4 Klientska aplikácia (graf) v responzívnom móde	38
3.3 Schéma napájania prídavnej dosky	11	6.1 Test komunikácie s PID regulátorom	41
3.4 Schéma zapojenia digitálnych vstupov/výstupov	12	6.2 Testovací obvod merania prúdu	42
3.5 Schéma 4-svorkového zapojenia odporového senzoru	14	6.3 Nameraná prevodná charakteristika senzoru LTS 15-NP	43
3.6 Typická schéma zapojenia prevodníka MAX31865 [5, s. 24]	15	6.4 Nameraná charakteristika merania prúdu	43
3.7 Schéma zapojenia odporového senzoru teploty	15	6.5 Nameraná charakteristika merania teploty	44
3.8 Senzor prúdu LTS 15-NP a jeho bloková schéma [6, s. 1, 4]	16		
3.9 Prevodná charakteristika senzoru prúdu LTS 15-NP [6, s. 4]	16		
3.10 Schéma zapojenia senzoru prúdu	17		
3.11 Schéma zapojenia senzoru prietoku	18		
3.12 Bloková schéma digitálneho PIR senzoru [7, s. 12]	18		
3.13 Časový diagram výstupu PIR senzoru EKMC1693112 [7, s. 7]	19		
3.14 Schéma zapojenia PIR senzoru	19		
3.15 Schéma zapojenia PID regulátora	20		
4.1 Diagram funkcie serveru	23		
4.2 Diagram spracovania požiadavky HTTP serverom	25		
4.3 Zobrazenie hodnôt sensorov na displeji kitu	27		
4.4 Časový priebeh zápisu do registra MAX31865 [5, s. 18]	29		

Tabuľky

4.1 Podporované požiadavky HTTP serveru	25
4.2 Popis registrov prevodníka MAX31865 [5, s. 13]	28
4.3 Popis kontrolného registra MAX31865 [5, s. 13]	28
4.4 Použité nastavenie kontrolného registra MAX31865	28
4.5 Použité nastavenie USART pre PID regulátor	31
4.6 Popis sériovej komunikácie OM 402PID [14]	31

Kapitola 1

Úvod

1.1 Motivácia

Laboratórium MagLab katedry meraní FEL vyvíja magnetické senzory typu fluxgate. Pre tieto senzory je potrebné vyrábať magnetické jadrá so špecifickými vlastnosťami, ktoré minimalizujú šumenie výsledných senzorov. Toho sa dosahuje žihaním magneticky mäkkých materiálov rôznymi tepelnými profilmi, často v prítomnosti externého magnetického poľa.

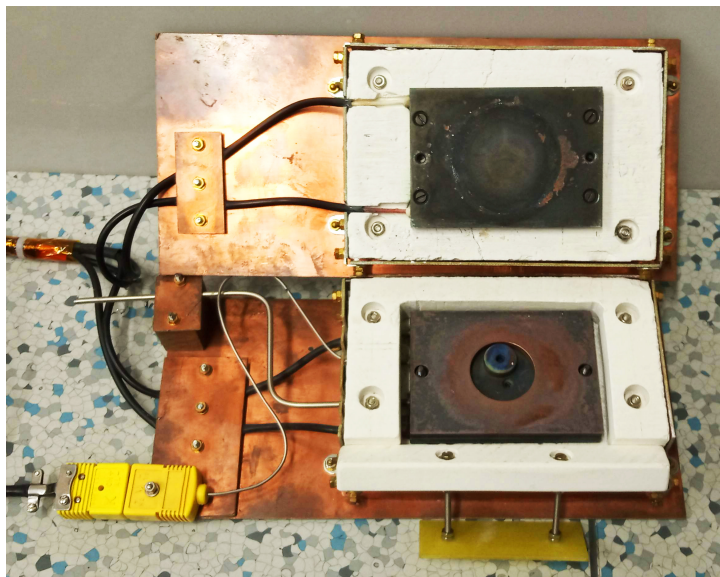
Na ohrev týchto materiálov sa používajú elektrické piecky s keramickými odporovými výhrevnými telesami, viď obr. 1.1. Teploty v nich sú merané pomocou zabudovaných termočlánkov typu K, pričom dosahujú hodnoty v rozsahu 100 °C až 500 °C. Na zabránenie oxidácie žíhaných materiálov pri takto vysokých teplotách majú tieto piecky taktiež možnosť napojenia na zdroj inertného plynu (typicky Argón). Teplotu piecky je potrebné presne regulovať za účelom zachovania špecifických teplotných profilov v čase, ktoré zabezpečia žiadané vlastnosti výsledného materiálu.

Pri potrebe žíhania materiálu za prítomnosti externého magnetického poľa sa celá výhrevná piecka obklopuje jarmom silného elektromagnetu. Ten je napájaný max. napätím cca 120 V, pričom ním tečie prúd až do 15 A. Veľkosť tohoto prúdu je nutné sledovať, keďže je od neho závislá veľkosť magnetickej indukcie poľa, čo ovplyvňuje vlastnosti výsledného materiálu.

Pri použití elektromagnetu v ňom vzniká značný stratový výkon, až v ráde jednotiek kW. Preto je potrebné jeho aktívne chladenie, ktoré sa dosahuje otvoreným okruhom vodného chladenia. Z rovnakého dôvodu je nutné monitorovať teplotu cievok elektromagnetu a výstupnej chladiacej kvapaliny. Týmto spôsobom je možné odhaliť zlyhanie chladiaceho systému a prípadné prehriatie cievok elektromagnetu, ktoré by mohlo viesť k ich poškodeniu. Taktiež je možné za týmto účelom merať prietok chladiacej kvapaliny.

Všetky tieto sledované fyzikálne veličiny je možné merať samostatnými elektronickými meracími prístrojmi, osobne v laboratóriu. Takýto spôsob merania je však časovo náročný a nepraktický, pretože vyžaduje fyzickú prí-

tomnosť poverenej osoby v laboratóriu. Lepším riešením by bol teda systém, pri ktorom by bolo možné vzdialené monitorovanie týchto veličín, prípadne aj ovládanie niektorých častí laboratórneho procesu.



Obrázok 1.1: Elektrická piecka používaná na žihanie materiálov

Riešením je teda elektronický merací systém, ktorý poskytuje vzdialený prístup. Pre čo najjednoduchšie použitie systému bol zvolený vzdialený prístup pomocou internetu. Monitorovací systém umiestnený v laboratóriu sa teda pripojí do fakultnej internetovej siete. Následne bude k nemu možný vzdialený prístup pomocou zariadení taktiež pripojených do rovnakej siete, za použitia špeciálnej aplikácie alebo formou webovej stránky. Pomocou nej bude možné sledovať hodnoty meraných veličín a ovládať časti laboratórneho procesu.

1.2 Požiadavky na systém

Konkrétne požiadavky na tento systém sú:

- Vstupy a výstupy:
 - 2x Senzor teploty
 - Meranie teploty cievok elektromagnetu, výstupnej chladiacej kvapaliny
 - Rozsah meraných teplôt cca 0 až 100 °C
 - Senzor prúdu
 - Meranie prúdu tečúceho elektromagnetom
 - Potrebné galvanické oddelenie od meraného obvodu
 - Rozsah veľkosti meraného prúdu približne 0 až 15 A

- Senzor prietoku
 - Sledovanie prietoku kvapaliny v chladiacom obvode elektromagnetu
 - Umožňuje odhalenie závady chladenia
- PID regulátor teploty
 - Externý regulátor, s ktorým je potrebné komunikovať
 - Reguluje teplotu vo výhrevnej piecke na potrebné hodnoty
 - Potreba čítať súčasnú hodnotu teploty a taktiež nastavovať novú požadovanú hodnotu
- 2x Digitálny vstup, 2x výstup
 - Vstupy na všeobecné použitie, výstupy napr. na zapínanie/vypínanie zariadení v laboratórnom procese
 - Potrebné galvanické oddelenie
- Senzor pohybu
 - Za účelom upozornenia na prítomnosť osôb v laboratóriu
- Záznam z kamery
 - Pre vizuálny dohľad nad žihacím procesom a laboratóriom
 - Nie je potrebný živý prenos, postačujú samostatné snímky na vyžiadanie
- Pripojenie k internetovej sieti
 - Káblové pripojenie k fakultnej sieti
 - Poskytuje možnosť vzdialeného prístupu k systému
- Vzdialené ovládanie klientskou aplikáciou
 - Formou webovej stránky alebo dedikovanej aplikácie
 - Zobrazovanie súčasných hodnôt vstupov a obrazu z kamery
 - Ovládanie výstupov

1.3 Možné riešenia

Existuje mnoho spôsobov ako vyrobiť takýto monitorovací systém. Každý z nich má svoje špecifiká, výhody a nevýhody pre danú aplikáciu. V tejto časti budú popísané niektoré vybrané možnosti implementácie takéhoto systému, avšak nie je zďaleka možné ich popísať všetky, alebo do väčšej hĺbky.

Jedným z možných riešení by bolo napríklad použiť notebook, ktorý by sa pripájal k internetovej sieti a taktiež zabezpečoval záznam obrazu pomocou zabudovanej webkamery. Zároveň obsahuje výkonný procesor, a takýto systém by teda dokázal zjavne bez problémov plniť funkciu webového serveru.

Problémom je, že bežné notebooky nemajú prakticky žiadne externé elektronické rozhrania, ktorými by bolo možné priamo sledovať senzory či ovládať výstupy systému. Na tento účel by museli byť použité externé zariadenia, napr. pripojené pomocou zbernice USB. To už ale navyšuje komplexnosť a cenu celého systému, v závislosti na počte a type potrebných zariadení. Taktiež by sa pripojením takýchto zariadení zväčšili už aj tak pomerne veľké fyzické rozmery tohto systému.

Ďalšou nevýhodou použitia notebooku v monitorovacom systéme je jeho veľká spotreba elektrickej energie, čo ho robí nevhodným na dlhodobú (resp. nepretržitú) prevádzku. K tomuto prispieva aj prítomnosť vysoko-úrovňového operačného systému (Windows, Linux...), ktorý je komplexný a náchylný na chyby a prípadné útoky z internetu. Zároveň by bežné funkcie OS, napr. aktualizácie a reštarty systému, mohli narušiť funkciu monitorovacieho systému a spôsobiť tak možné škody na zariadení v laboratóriu. Taktiež sa nejedná o operačné systémy reálneho času, a teda nie je možné garantovať časové odozvy monitorovacieho systému, pretože sú závislé od momentálneho vyťaženia OS.

Možnou alternatívou by bolo napríklad použitie cenovo dostupného mini-počítača Raspberry Pi. Ten má v porovnaní s notebookom viaceré výhody, ako napríklad kompaktnosť a oveľa menšiu spotrebu energie, pri zachovaní relatívne vysokého výpočtového výkonu. Raspberry Pi taktiež narozdiel od notebookov poskytuje GPIO piny plniace funkciu digitálnych vstupov a výstupov. Tieto poskytujú možnosť pripojenia rôznych prídavných dosiek (nazývané *HATs*), napr. podporujúcich pripojenie rôznych senzorov, indikátorov alebo aj motorov. Tie sú vyžadované pre pripojenie analógových senzorov, pretože samotné Raspberry Pi neobsahuje žiaden A/D prevodník. Raspberry Pi má ale aj svoje nedostatky, napr. oproti notebooku absenciu displeja a integrovanej kamery. Medzi nevýhody je možné taktiež počítať aj bežný Linuxový operačný systém, ktorý má rovnaké nevýhody ako OS notebookov popísané vyššie.

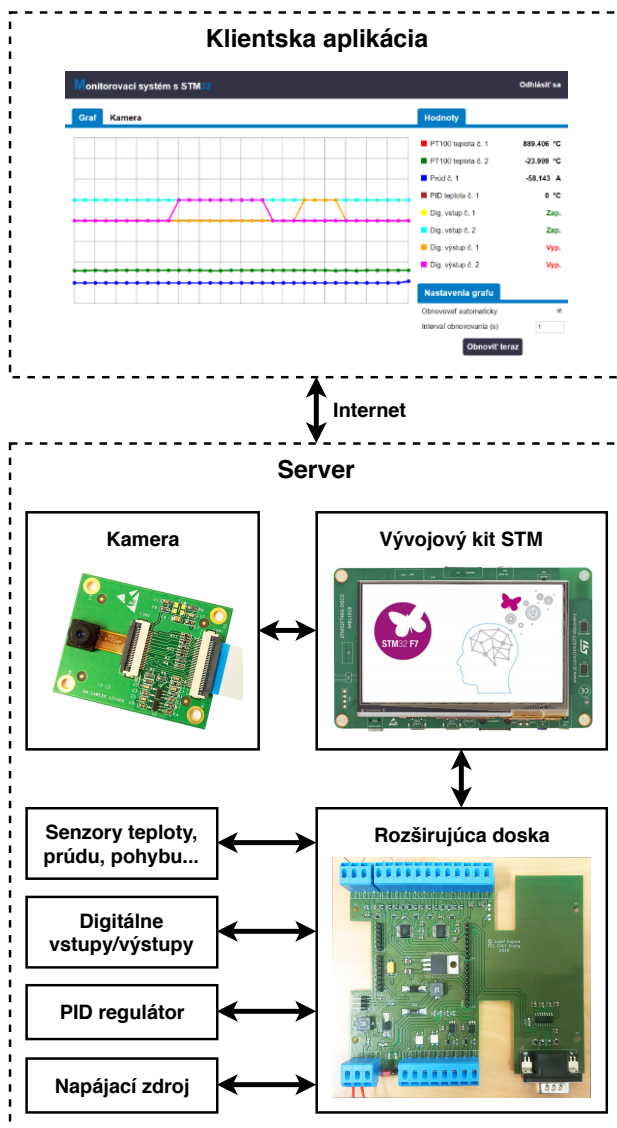
Zrejme najrobustnejším riešením by bolo použiť už nejaký hotový priemyselný monitorovací systém. Tie sú ale finančne náročné a často komplikované na inštaláciu a údržbu. Zároveň nie sú vždy presne prispôbené výslednému použitiu a poskytujú tak často funkcie, ktoré by neboli využité, alebo naopak nepodporujú niektoré požadované funkcie.

Z týchto dôvodov bola zvolená cesta návrhu vlastného monitorovacieho systému, ktorý by presne splňal vytýčené požiadavky. Tento systém už bol raz navrhovaný v rámci predmetu *Projekt v tímu*, ktorého som nebol súčasťou. Projekt sa však nikdy nedostal do funkčného štádia a nebol dokončený. Nový systém prebral niektoré časti návrhu elektrického zapojenia zo starého projektu, avšak aj tieto museli byť miestami opravované. Všetky ostatné časti systému (firmvér, návrh DPS, klientska aplikácia, atď.) boli navrhnuté nanovo od základov. V ďalších častiach textu sa nachádza bližší popis tohto systému.

Kapitola 2

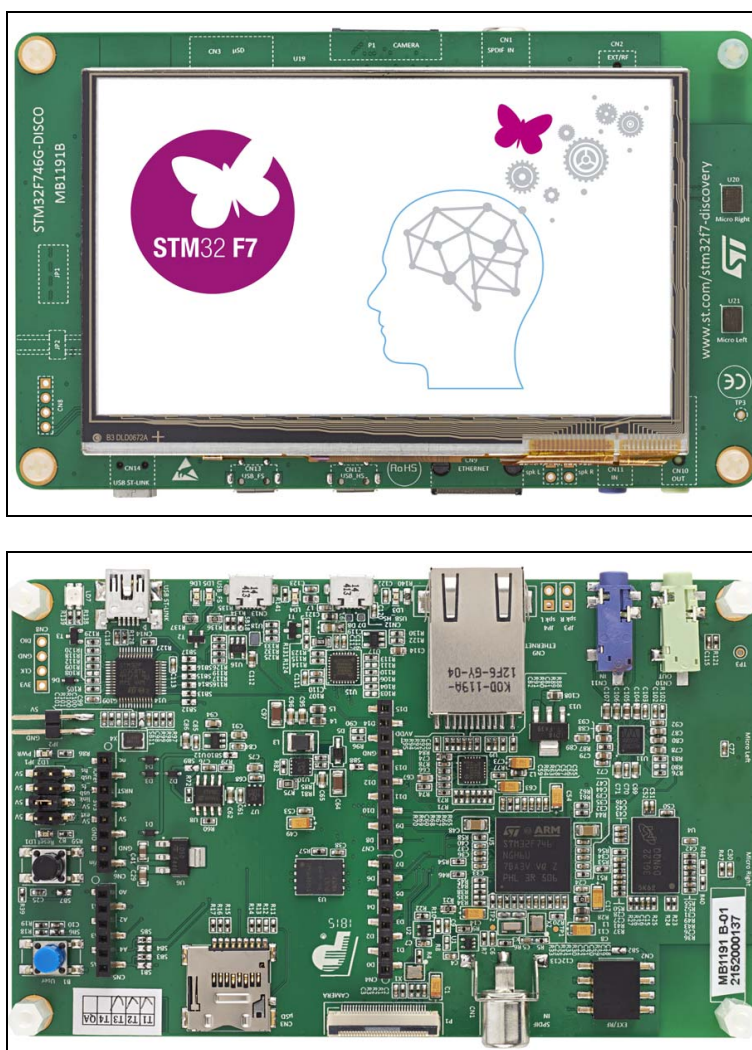
Navrhnutý systém

Pre riešenie tohto zadania bol navrhnutý systém znázornený na obr. 2.1.

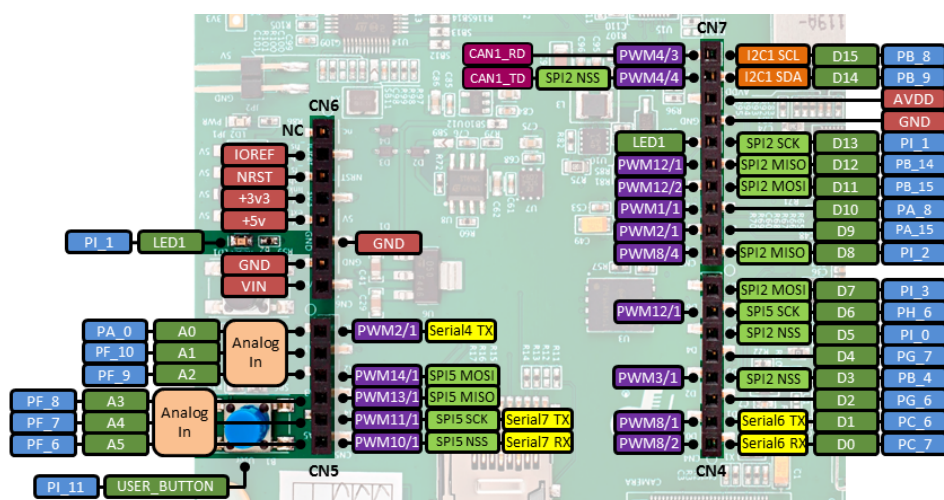


Obrázok 2.1: Diagram navrhnutého monitorovacieho systému [1]

- 128 Mbit (64 Mbit adresovateľných)
- Programátor ST-LINK
 - Zabudovaný na doske
 - Verzia 2.1
 - Podporujúci USB rozhranie (cez mini-USB konektor na doske)
- Arduino Uno V3 konektor
 - Použitý na pripojenie prídavnej vstupne/výstupnej dosky
 - Zapojenie vid' obr. 2.3
- Kamerový konektor



Obrázok 2.2: Vývojový kit STM32F746NG-DISCOVERY [1]

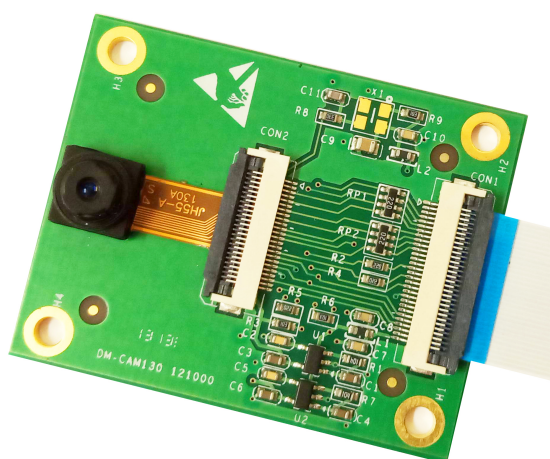


Obrázok 2.3: Zapojenie Arduino konektora vývojového kitu STM [2]

Základom vývojového kitu je 32-bitový mikrokontrolér STM32F746NGH6, založený na architektúre ARM Cortex-M7 (ďalej tiež *aplikačný mikrokontrolér, mikrokontrolér*). Ten disponuje 1 Mbit FLASH pamäte, 340 KB RAM pamäte a maximálnou hodinovou frekvenciou 216 MHz. Obsahuje viacero vstupne/výstupných periférií, medzi tie použité v tejto aplikácii patria: SPI, USART, I2C, DCMI, DMA, ADC.

2.2 Kamera

K vývojovému kitu je pomocou kamerového konektora pripojený prídavný kit STM32F4DIS-CAM s kamerou OV9655. Táto disponuje rozlíšením 640 x 480 a farebným formátom RGB-565. Komunikuje s aplikačným mikrokontrolérom cez perifériu DCMI (prenos snímok) a zbernicu I2C (konfigurácia kamery).

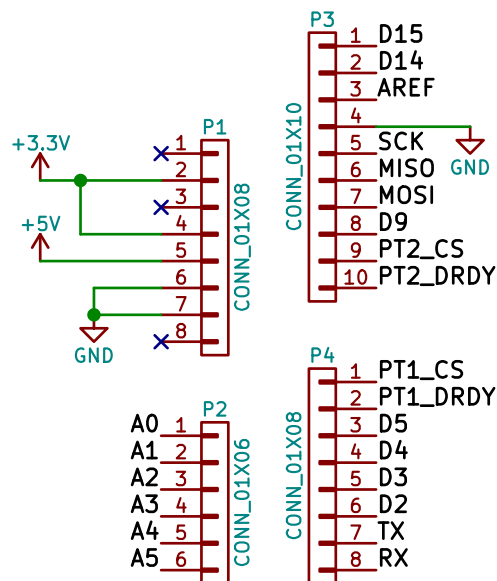


Obrázok 2.4: Prídavný kit STM32F4DIS-CAM s kamerou

Kapitola 3

Rozširujúca doska

Použitý vývojový kit poskytuje základné funkcie monitorovacieho systému, ako pripojenie k internetovej sieti, zobrazovanie informácií na displeji apod. Nepodporuje však pripojenie všetkých potrebných senzorov či galvanicky oddelené vstupy/výstupy. Za týmto účelom bola navrhnutá prídavná vstupne/výstupná doska plošného spoja, obsahujúca potrebné konektory, prídavné obvody pre senzory, napájanie atď. V ďalšom texte budú bližšie popísané všetky časti rozširujúcej dosky a podporných obvodov pre jednotlivé vstupy a výstupy systému.



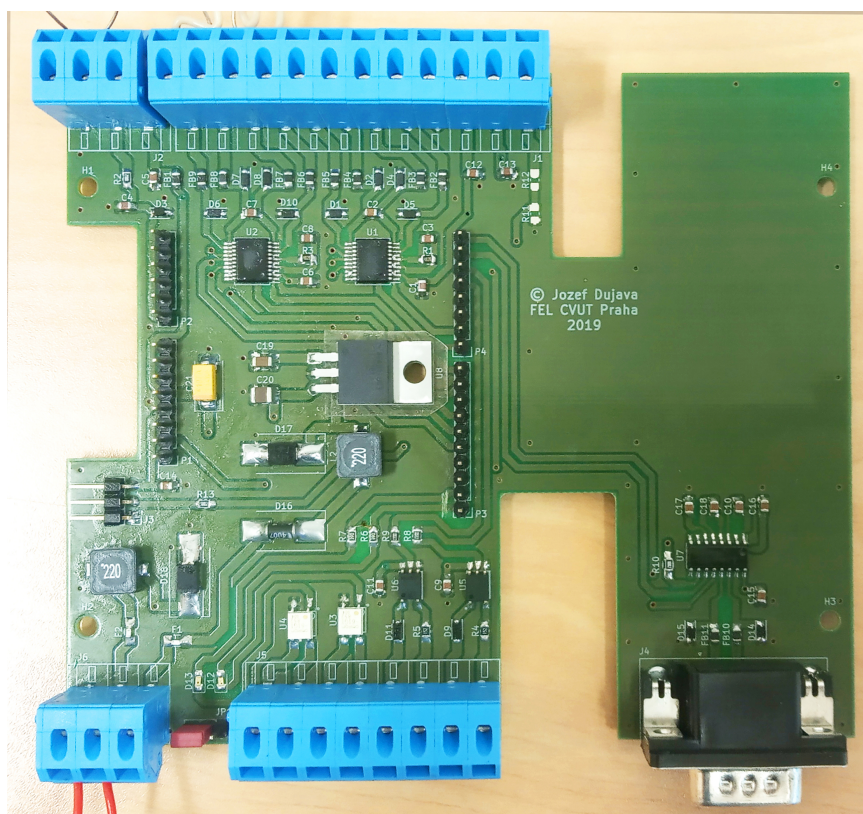
Obrázok 3.1: Zapojenie Arduino konektora rozširujúcej dosky

Fyzická forma rozširujúcej DPS bola navrhnutá tak, aby ju bolo možné priamo nasunúť na vývojový kit, s jediným elektrickým prepojením pomocou Arduino konektora (viď obr. 2.3, 3.1). Fyzicky je s ňou potom vývojový kit spojený pomocou 4 skrutiek a dištančných stĺpikov v miestach pôvodných montážnych dier kitu. Takýto spôsob prichytenia teda stanovil minimálne rozmery rozširujúcej dosky (podľa rozteče montážnych dier kitu).

Rozširujúca doska má vo výsledku rovnakú šírku ako doska vývojového kitu, a výšku rozšírenú o priestor pre svorkovnice/konektory na pripojenie napájania a vstupov/výstupov po obvode dosky.

Samotné elektronické komponenty sú následne umiestnené na vnútornej časti DPS, kde sú neprístupné po montáži vývojového kitu nad rozširujúcu dosku. Z dôvodu zabránenia fyzickej interferencie Ethernet a RCA konektorov, konfiguračných jumperov a tlačidiel vývojového kitu s rozširujúcou doskou, boli realizované výrezy v inak obdĺžnikovej DPS. Takéto rozloženie teda poskytuje prístup k všetkým potrebným prvkom na obvode DPS (konektory vývojového kitu a sériovej linky, svorkovnice pre vstupy/výstupy, jumper napájania, indikačné LED, atď.) pri zachovaní kompaktných rozmerov.

Rozširujúca DPS je 2-vrstvová, s rozliatou zemou na oboch vrstvách. Prekopy sú použité vo veľkom počte na prepojenie zemí po obvode dosky a v blízkosti signálových dráh. Tento postup znižuje úbytky napätia na zemnej ceste, čo minimalizuje vyžarované elektromagnetické rušenie dosky [3, s. 120]. Vodivé dráhy a plôšky boli navrhnuté čo najširšie, za účelom zníženia ich el. odporu a zvýšenia ich odolnosti pri spájkovaní. Na hornej strane dosky boli následne ručne osadené SMD a THT komponenty, viď obr. 3.2.



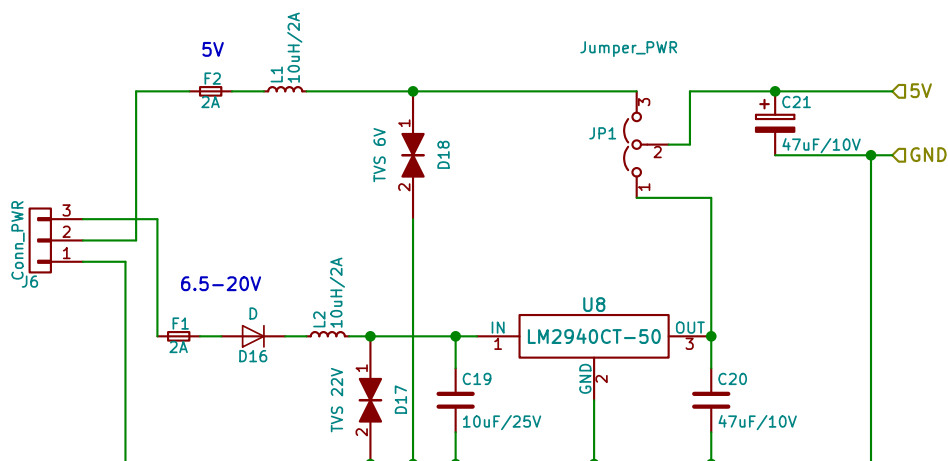
Obrázok 3.2: Osadená rozširujúca doska

3.1 Napájanie

Prídavná doska obsahuje vlastné napájacie obvody, ktorými následne napája aj vývojový kit, schéma vid' obr. 3.3. Pri takomto externom napájaní je potrebné správne nastaviť jumper na kite na polohu **5V_ext** (narozdiel od **5V_link** použitej pri programovaní kitu). Prídavná doska podporuje dva varianty napájacích zdrojov:

- +5 V: Stredná poloha na svorkovnici napájania
- +6,5 V až +20 V: Krajne pravá poloha na svorkovnici napájania

Zem (GND) sa nachádza v krajne ľavej polohe svorkovnice napájania. Výber použitého zdroja sa vykonáva osadením napájacieho jumpera JP1. Pri napájaní systému zo zdroja +5 V je vstup chránený proti skratom poistkou, filtrovaný tlmivkou a chránený proti prepätiu pomocou TVS diódy. Nakoniec je pripojený na krajne ľavú polohu JP1.



Obrázok 3.3: Schéma napájania prídavnej dosky

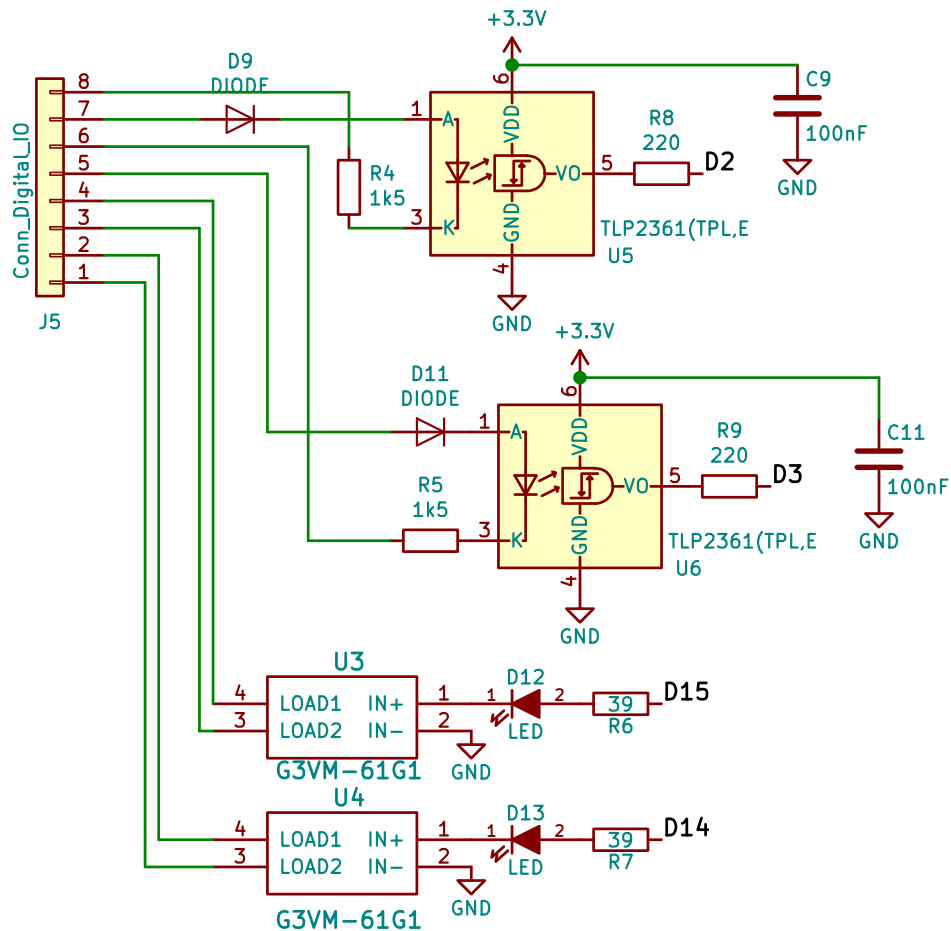
Pri napájaní zo zdroja +6,5 V až +20 V sa využíva integrovaný lineárny stabilizátor 5 V typu LM2940CT-50. Na vstupe a výstupe regulátora sú umiestnené požadované stabilizačné kondenzátory a vstup je taktiež chránený obdobne ako v predošlom prípade. Jeho výstup je pripojený na krajne pravú polohu JP1.

Výstup napájacieho jumperu JP1 sa nachádza v strednej polohe, a je pripojený na 5V vetvu napájania prídavnej dosky a vývojového kitu. V blízkosti 5V pinu na Arduino konektore je taktiež umiestnený $47\mu\text{F}$ tantalový blokovací kondenzátor, ktorý pomáha stabilizovať napájacie napätie systému pri odberových špičkách.

3.2 Digitálne vstupy a výstupy

Na prídavnej doske sú realizované 2 digitálne vstupy a 2 výstupy. Všetky sú galvanicky oddelené optickou cestou a merané/ovládané pomocou digitálnych pinov aplikačného mikrokontroléra. Celková schéma zapojenia digitálnych vstupov a výstupov je znázornená na obr. 3.4.

Pre digitálne výstupy boli použité optomofety (*SSR – Solid State Relay*) typu G3VM-61G1. Tieto sú ovládané digitálnymi výstupmi vývojového kitu, pričom v sérii s nimi sú zapojené indikačné LED diódy a rezistor obmedzujúci prúd. Ak je na výstupnom pine mikrokontroléra log. 1 (3,3 V), rozsvieti sa indikačná LED a výstupom môže tiecť prúd (max. $R_{ON} = 2 \Omega$). V prípade log. 0 výstupom prúd netečie (max. $I_{LEAK} = 1 \mu A$).



Obrázok 3.4: Schéma zapojenia digitálnych vstupov/výstupov

Pre digitálne vstupy sa používa optočlen TLP-2361, ktorý disponuje 3,3V logickými výstupmi s inverznou logikou. Vďaka tejto forme výstupu je jednoduché merať stav tohto digitálneho vstupu pomocou aplikačného mikrokontroléra. Na vstupe optočlena je umiestnená ochranná usmerňovacia dióda a rezistor obmedzujúci prúd optočlenom. V pokojovom stave (ak vstupom netečie prúd) je na výstupe optočlena log. 1. Pri pripojení napätia 3 až 15 V správnej polarity na vstup prejde výstup do hodnoty log. 0. Tento interval prípustných napätí je daný použitými ochrannými prvkami, ktoré sú dimenzované tak, aby optočlen pracoval v medziach povolených výrobcom.

3.3 Senzory teploty

Na meranie teploty sa v tomto monitorovacom systéme používajú odporové senzory teploty (angl. *RTD - Resistance Temperature Detector*). Tie pracujú na princípe teplotnej závislosti elektrického odporu kovového vodiča. Meria sa zmena odporu, z ktorej je následne možné výpočtom určiť zmenu teploty.

Najpoužívanejším materiálom pre takéto senzory je platina, vďaka svojej linearite, dlhodobej stabilite a odolnosti [4, s. 461]. Štandardne sa pre každý takýto senzor určí nominálna hodnota odporu R_0 , ktorú senzor vykazuje pri teplote 0 °C. Typické hodnoty (a označenia) sú 100 Ω (PT100) alebo 1000 Ω (PT1000). Z relatívnej zmeny tohto odporu je možné vypočítať teplotu senzoru v rozsahu 0 °C až 630 °C podľa [4, s. 463]:

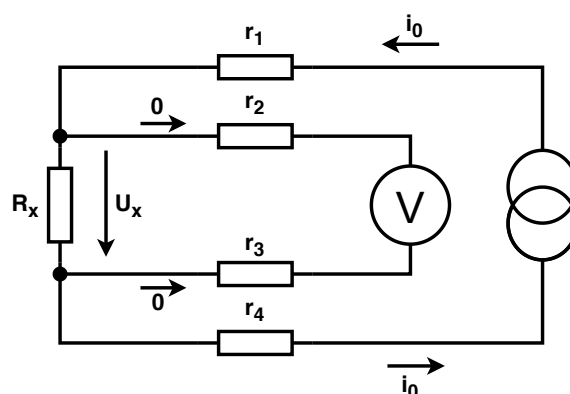
$$R_t = R_0 \cdot (1 + \alpha t + \beta t^2). \quad (3.1)$$

kde t je teplota v °C, a α, β sú konštanty použitého materiálu. Pre platinu sú typické hodnoty (α podľa DIN/IEC 751):

$$\alpha = 3,850 \cdot 10^{-3} \text{ } ^\circ\text{C}^{-1} \quad (3.2)$$

$$\beta = -5,775 \cdot 10^{-7} \text{ } ^\circ\text{C}^{-2}. \quad (3.3)$$

Pri meraní odporu senzoru jednoduchou 2-svorkovou metódou dochádza k nepresnostiam spôsobeným úbytkami napätia na prívodoch, ktorými tečie napájací prúd senzoru. Na zamedzenie tohto efektu sa preto používa 4-svorkové zapojenie, viď obr. 3.5. Pri takomto zapojení je senzor napájaný jedným párom prívodov, ktorými tečie prúd a úbytok napätia na senzore sa meria druhým, samostatným párom prívodov, cez ktoré netečie takmer žiaden prúd (sú pripojené na vysoko-impedančný vstup voltmetra alebo A/D prevodníka).

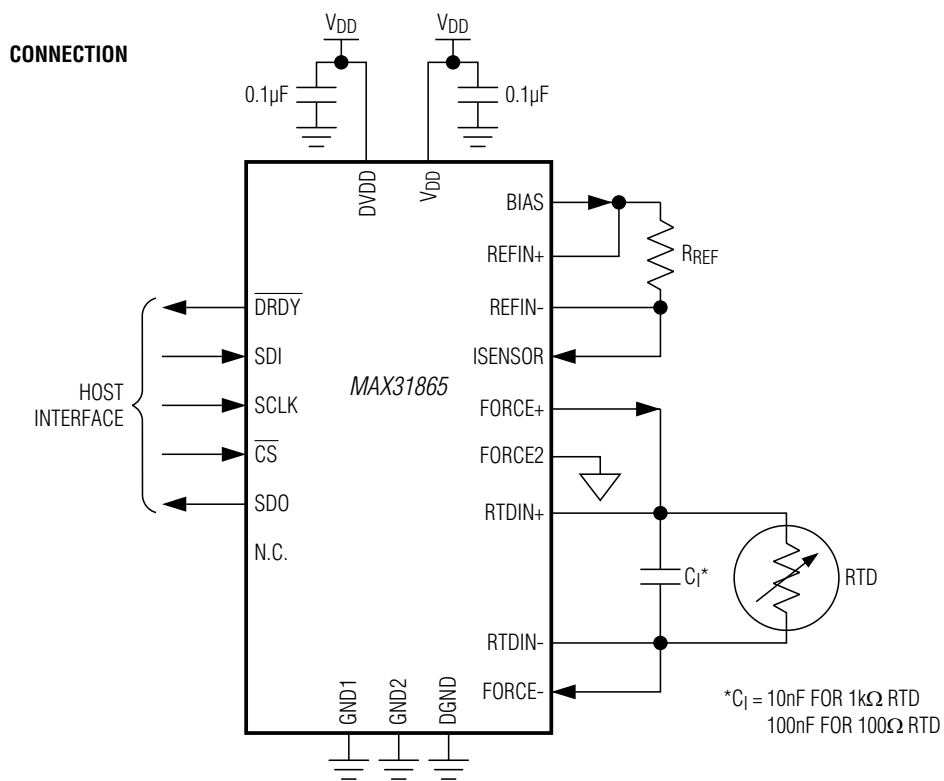


Obrázok 3.5: Schéma 4-svorkového zapojenia odporového senzoru

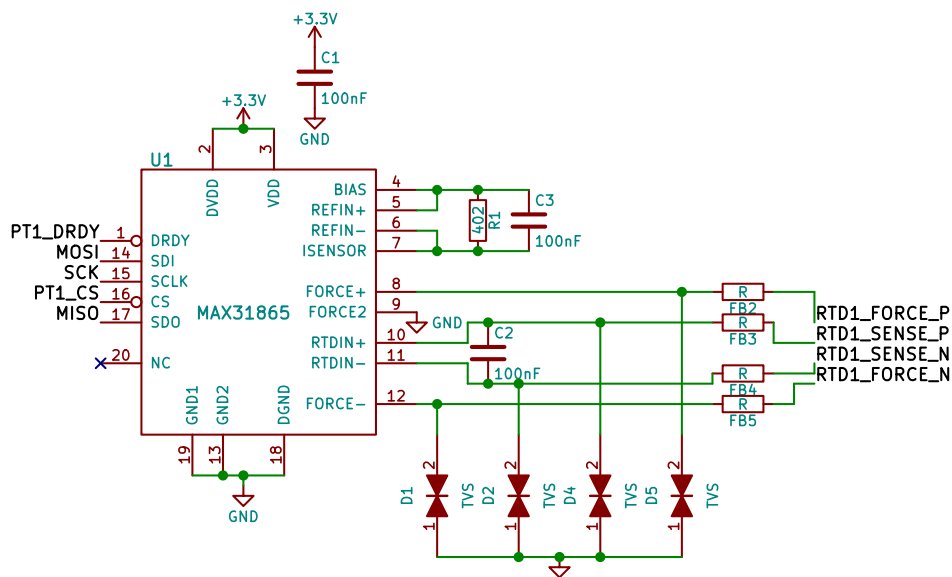
Na samotné meranie odporu senzoru sa využívajú dedikované prevodníky RTD-To-Digital typu MAX31865 [5]. Typická schéma zapojenia je znázornená na obr. 3.6. Princíp ich funkcie je nasledovný:

- Medzi piny **REFIN** (spojený s **BIAS**) a **REFIN-** (spojený s **ISENSOR**) je pripojený referenčný rezistor, ktorý má v ideálnom prípade odpor rovný štvornásobku nominálneho odporu teplotného senzora [5, s. 10].
- Na pin **BIAS** je privedené kladné napätie (typ. +2 V) a referenčným rezistorom začne tiecť prúd.
- Ten istý prúd následne tečie cez piny **ISENSOR**, **FORCE+** do RTD senzoru a nakoniec do pinu **FORCE-**.
- Medzi pinmi **REFIN+**, **REFIN-** (na referenčnom rezistore) vzniká úbytok napätia. Ten je použitý ako referenčné napätie interného 15-bitového sigma-delta AD prevodníka.
- Medzi pinmi **RTDIN+**, **RTDIN-** sa týmto AD prevodníkom meria úbytok napätia na samotnom RTD senzore.
- Na výstupe A/D prevodníka je teda vo výsledku pomer úbytkov napätia na senzore a referenčnom rezistore, $q = \frac{U_{RTD}}{U_{REF}}$. Pretože ale oboma tečie rovnaký prúd sa podľa Ohmovho zákona tento pomer zároveň rovná pomeru ich odporov, $q = \frac{R_{RTD}}{R_{REF}}$.
- Tento pomer odporov q je následne po skončení merania uložený binárne v interných registroch prevodníka. Odtiaľ môže byť čítaný pomocou rozhrania SPI, viď podkapitola 4.4.2.

V tomto systéme sa používajú senzory typu PT100 a preto boli použité precízne referenčné rezistory s odporom 402Ω . Tie majú malú teplotnú závislosť, typicky $5 \text{ ppm}/^\circ\text{C}$, a sú časovo stabilné. Vstupy prevodníkov sú opatrené feritovými jadrami pre obmedzenie rušenia a TVS diódami voči zemi pre ochranu pred prepätím.



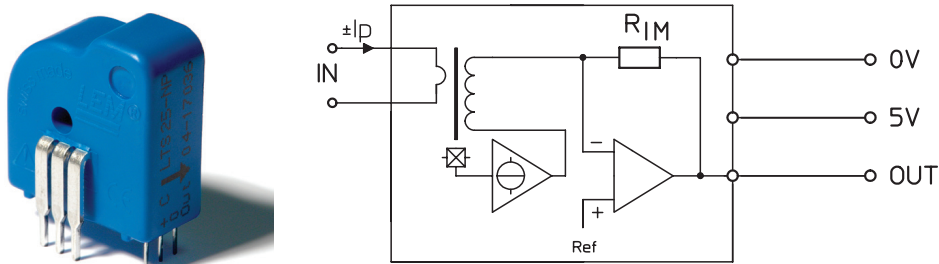
Obrázok 3.6: Typická schéma zapojenia prevodníka MAX31865 [5, s. 24]



Obrázok 3.7: Schéma zapojenia odporového senzoru teploty

3.4 Senzor prúdu

Na meranie prúdu existuje mnoho možností, Každá z týchto možností má špecifické vlastnosti, výhody a nevýhody pre danú aplikáciu. V tomto systéme bolo požadované galvanicky oddelené meranie prúdov, čo možnosti zúžilo. Nakoniec bol vybraný senzor LTS 15-NP (LEM)[6], fungujúci na princípe kompenzovaného Hallovhho senzoru.

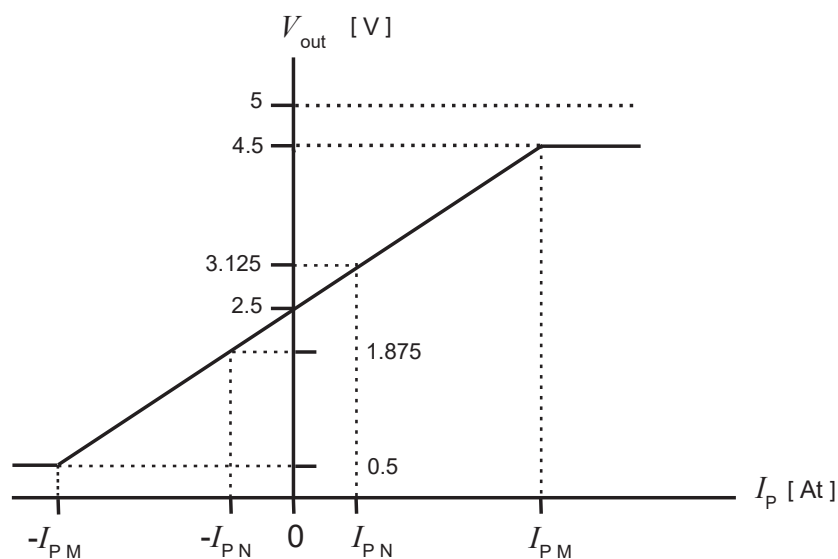


Obrázok 3.8: Senzor prúdu LTS 15-NP a jeho bloková schéma [6, s. 1, 4]

Tento typ senzoru umožňuje merať jednosmerné aj striedavé prúdy v max. rozsahu ± 48 A (nominálne však ± 15 A). Senzor je napájaný z +5 V a jeho výstupom je napätie, pre ktoré platí

$$U_{out} = U_0 + I_{in} \cdot G \approx 2,5 \text{ V} + I_{in} \cdot 0,0416 \text{ V/A}. \quad (3.4)$$

Pri nulovom prúde je teda na výstupe $U_0 = 2,5$ V (polovica napájacieho napätia), a pri toku prúdu sa toto napätie mení s citlivosťou $G = 41,6$ mV/A. Táto prevodná charakteristika senzoru je znázornená graficky na obr. 3.9.

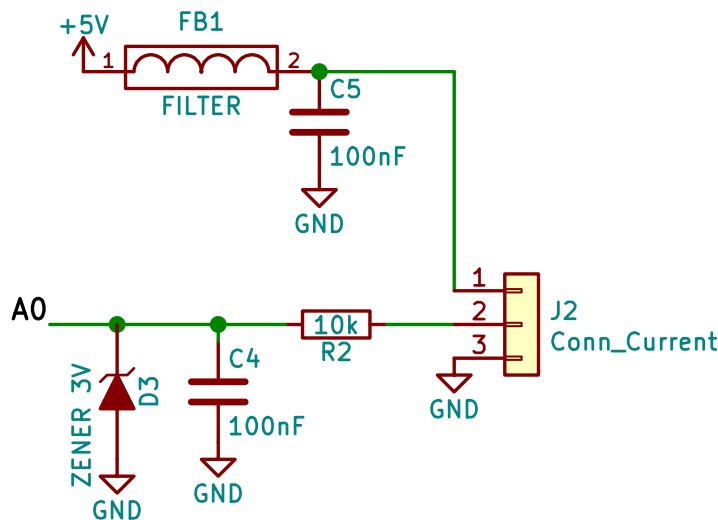


Obrázok 3.9: Prevodná charakteristika senzoru prúdu LTS 15-NP [6, s. 4]

Napätie na výstupe senzoru je merané interným A/D prevodníkom mikrokontroléra. Ten dokáže merať max. napätie 3,3 V (napájacie napätie), a preto sa senzorom meria vždy len prúd tej polarity, ktorá výstupné napätie znižuje od 2,5 V smerom k 0 V (proti smeru šípky na senzore). Z rovnakého dôvodu je na vstupe A/D prevodníka umiestnená ochranná 3V Zenerova dióda, ktorá prípadné vyššie napätie obmedzí max. na 3 V (napr. pri prepólovaní prúdu senzorom). Taktiež je na vstupe A/D prevodníka umiestnený pasívny RC filter typu dolná prepust' s časovou konštantou

$$\tau = R_2 \cdot C_4 = 10 \text{ k}\Omega \cdot 100 \text{ nF} = 1 \text{ ms},$$

ktorým sa filtruje striedavý šum na výstupe senzoru. Rezistor tohto filtru zároveň obmedzuje prúd do A/D prevodníka.

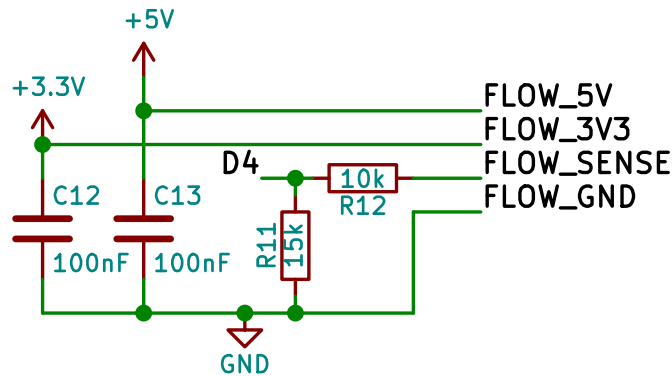


Obrázok 3.10: Schéma zapojenia senzoru prúdu

3.5 Senzor prietoku

Pri návrhu prídavnej dosky ešte nebol známy presný typ senzoru prietoku, ktorý by sa v systéme použil. Z tohto dôvodu bolo na svorkovnicu tohto senzoru vyvedené napájanie 3,3 V aj 5 V. Taktiež bol navrhnutý jednoduchý odporový delič výstupného napätia senzoru, pre konverziu z 5V na 3V logiku, s ktorou môže pracovať aplikačný mikrokontrolér.

Predpokladal sa výstup senzoru vo forme obdĺžnikových pulzov napätia o amplitúde rovnej napájaciemu napätiu, a frekvencii, ktorá sa mení v závislosti na meranom prúde. Takýto výstup senzoru sa následne jednoducho vyhodnocuje digitálnym vstupom mikrokontroléra, napr. za použitia čítača. Schéma zapojenia senzoru je znázornená na obr. 3.11.

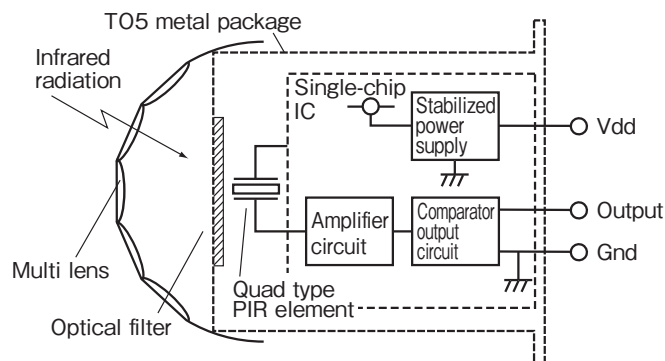


Obrázok 3.11: Schéma zapojenia senzoru prietoku

3.6 Senzor pohybu

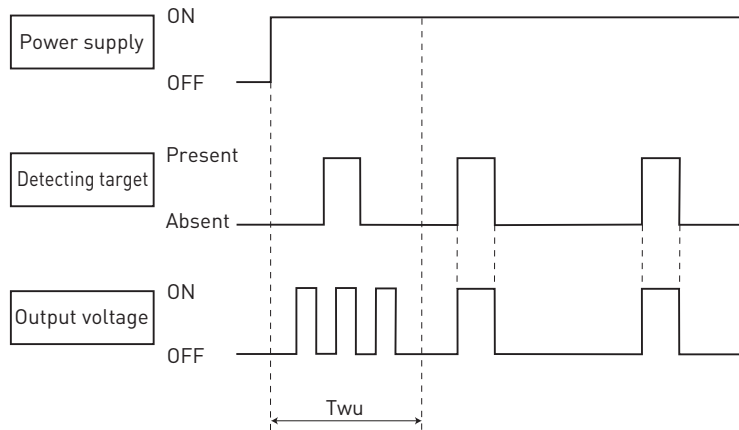
Na detekciu pohybu v sledovanom priestore používa monitorovací systém PIR (*Passive Infra-Red*) senzor. Tento typ senzoru pracuje na princípe pyroelektrického efektu, pri ktorom sa dopadom infračerveného žiarenia na pyroelektrický materiál senzoru vytvorí elektrický náboj. Zjednodušene povedané sa jedná o piezoelektrický efekt materiálu, ktorý sa dopadom infračerveného žiarenia zahrieva a tepelne rozťahuje. Tento náboj sa následne prejaví ako úbytok napätia medzi elektródami na opačných koncoch senzoru, ktorý môže byť nakoniec meraný [4, s. 245].

Pretože sa ale podobný náboj môže vytvoriť aj z dôvodu fyzickej deformácie materiálu a ovplyvňovať tak merania, sú v praxi používané senzory s dvoma symetrickými detekčnými elementami v spoločnom púzdre. Tie sú následne zapojené s opačnými fázami, čím sa odstráni vplyv rušivých vplyvov pôsobiacich na oba senzory rovnako, a zostane len užitočný diferenčný signál, ktorý je meraný [4, s. 246].



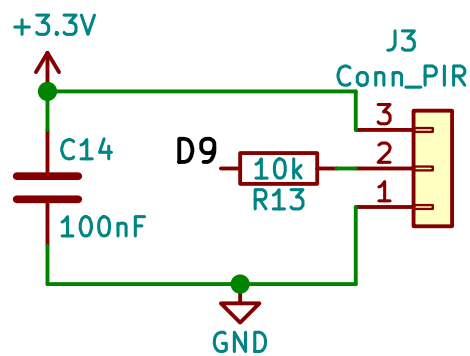
Obrázok 3.12: Bloková schéma digitálneho PIR senzoru [7, s. 12]

V tejto aplikácii sa používa integrovaný digitálny PIR senzor typu EKMC1693112 (Panasonic) [7], viď obr. 3.12. Tento senzor má digitálny výstup s pokojovou hodnotou v log. 0 a pulzami log. 1 pri detekcii objektu v sledovanej oblasti, viď časový priebeh na obr. 3.13.



Obrázok 3.13: Časový diagram výstupu PIR senzoru EKMC1693112 [7, s. 7]

Výstup tohto senzoru sa teda veľmi jednoducho meria pomocou digitálnych vstupov mikrokontroléra, viď schéma na obr. 3.14. Pre ochranu pri prípadnom chybnom naprogramovaní pripojeného pinu mikrokontroléra ako výstupu, je výstup senzoru pripojený cez rezistor obmedzujúci prúd.



Obrázok 3.14: Schéma zapojenia PIR senzoru

Kapitola 4

Firmvér serveru

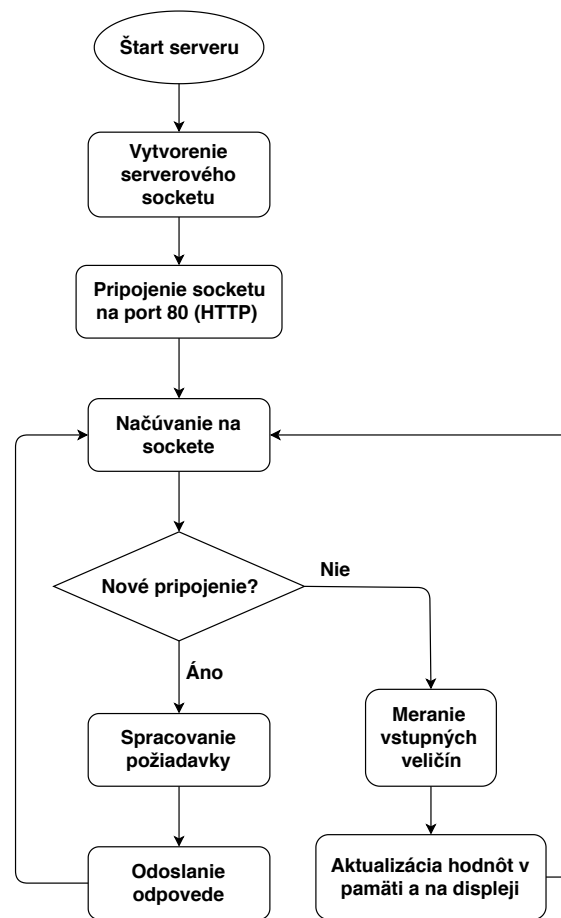
4.1 Použité softvérové prostriedky

Kód firmvéru serveru je napísaný výhradne v programovacom jazyku C. Ten bol zvolený pre svoju jednoduchosť a možnosti prístupu k hardvéru. Zároveň je dobre podporovaný kompilátormi pre väčšinu platforiem embedded systémov. Hardvér mikrokontroléra je možné konfigurovať a ovládať priamo v samotnom jazyku C, pomocou pamäťovo adresovaného prístupu ku konfiguračným registrom procesora. Jednotlivé registre, ich funkcie, adresy a možné hodnoty je možné nájsť v referenčnom manuáli mikrokontroléra, viď [9].

Takýto prístup je ale zdĺhavý, náročný, a typicky nevhodný pre vývoj rozsiahlejších aplikácií. Taktiež je náchylný na chyby programátora. Z týchto dôvodov sú namiesto toho použité knižnice HAL (*Hardware Abstraction Layer*), ktoré sú vyvíjané firmou STMicroelectronics a poskytované k mikrokontrolérom ich výroby. Tieto knižnice poskytujú jednoduchý prístup k všetkému hardvéru mikrokontroléra (vstupy/výstupy, periférie, rozhrania) pomocou štruktúr a funkcií v jazyku C.

Na knižniciach HAL je založená taktiež použitá knižnica BSP (*Board Support Package*). Knižnice tohto typu sú poskytované k väčšine vývojových kitov STM a obsahujú funkcie podporujúce vývoj na nich. Jedná sa napr. o ovládanie LCD displeja, kamery, komunikáciu s externými pamäťami atď. Taktiež obsahujú už pripravené definície jednotlivých prepojení na vývojom kíte (napr. piny, ku ktorým je pripojený displej), čo značne zjednodušuje prácu programátora. Vlastná aplikácia túto knižnicu využíva hlavne na ovládanie LCD obrazovky a pripojenej kamery.

Vlastný firmvér serveru je sčasti založený na voľne dostupných ukázkových projektoch pre použitý vývojový kit. Tieto projekty používajú knižnice HAL a BSP na obsluhu hardvéru, a prípadne aj knižnice tretích strán pre špecifickú funkcionálnosť. Mimo iné zahŕňajú komunikáciu zbernicami USART, SPI a I2C, ukázkový HTTP server a pod. Prevzaté časti kódu tvoria však len malú časť výsledného firmvéru, pričom boli značne upravené pre špecifické potreby aplikácie (hlavne HTTP server).



Obrázok 4.1: Diagram funkcie serveru

4.3 HTTP server

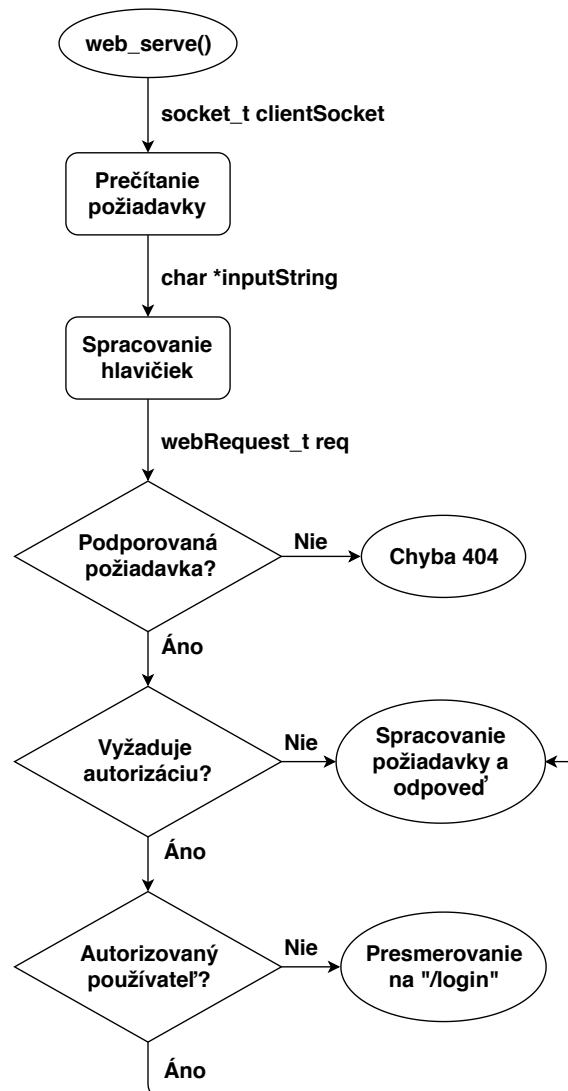
4.3.1 Použitá technológia

Implementovaný HTTP server bol v základe inšpirovaný príslušným ukázkovým projektom, no značne upravený a rozšírený. Vo vlastnej aplikácii aj v ukázkovom projekte sa využíva knižnica LwIP [11] (*Lightweight IP stack*). Táto poskytuje implementáciu potrebných sieťových protokolov od fyzickej až po transportnú vrstvu. Jedná sa najmä o protokoly TCP, IP, Ethernet a taktiež prístup k sieťovému hardvéru fyzickej vrstvy.

Táto knižnica má dva hlavné varianty rozhraní, `netconn` [12] (použitý v pôvodnom ukázkovom projekte) a `socket` [13] (použitý vo vlastnej aplikácii). Dôvodom prechodu na rozhranie typu `socket` bola jeho flexibilita a zároveň skutočnosť, že je dobre zdokumentované a štandardizované z viacerých otvorených platforiem (POSIX, BSD). To významne uľahčilo hľadanie ukázkových kódov a riešenie prípadných chýb. Zároveň toto rozhranie poskytuje možnosť tzv. neblokovaného socketu, ktorý nezastavuje chod programu pri čakaní na

URL	Metóda	Aut.	Popis
/, /index.html	GET	0	Hlavná stránka aplikácie
/main.css	GET	0	Hlavný CSS štýl
/small.css	GET	0	CSS štýl pre malé obrazovky
/login	GET, POST	0	Prihlasovacia stránka
/logout	GET	1	Odhlasovacia stránka
/getIO	GET	1	Vyžiadanie hodnôt senzorov
/setIO	POST	1	Nastavenie výstupov
/getCam	GET	1	Vyžiadanie snímky z kamery

Tabuľka 4.1: Podporované požiadavky HTTP serveru



Obrázok 4.2: Diagram spracovania požiadavky HTTP serverom

4.4 Komunikácia so vstupmi/výstupmi

Rozhranie (funkcie a štruktúry) použité na komunikáciu so senzormi je nazývané IOP (podľa *Input, Output, Peripheral*). V hlavičkovom súbore rozhrania, `iop.h`, sa nastavujú všetky konfiguračné parametre periférií (SPI, USART, ADC...) a použité hardvérové prostriedky (GPIO piny a ich funkcie). Hlavičkové súbory pre jednotlivé senzory sa doň následne vkladajú pomocou `#include "iop/<nazov_senzoru>.h"`. Týmto spôsobom (prítomnosťou alebo absenciou tohto príkazu) je možné vypínať a zapínať sledovanie jednotlivých senzorov (napr. ak sú odpojené alebo nefunkčné).

Pre každý typ použitého senzoru sú definované univerzálne makrá v ich príslušných hlavičkových súboroch:

- `IOP_<NAZOV_SENZORU>_COUNT`: Určuje, koľko kusov tohto typu senzoru je použitých v aplikácii (≥ 1)
- `IOP_<NAZOV_SENZORU>_NAME`: Názov daného senzoru v ľudske čitateľnej forme (napr. 'Prúd')
- `IOP_<NAZOV_SENZORU>_UNIT`: Jednotka meranej hodnoty senzoru, ako textový reťazec (napr. 'A' pre senzor prúdu)

Každý vstup/výstup má vlastný dátový typ pre svoje hodnoty, nazývaný `iop_<NAZOV_SENZORU>val_t`. Ten je pre analógové vstupy/výstupy spravidla definovaný ako `double`, a pre digitálne ako `GPIO_PinState`. Všetky tieto hodnoty senzorov sú po skončení jedného meracieho cyklu uložené do štruktúry typu `iop_data_t`. Tá obsahuje pole hodnôt príslušného typu pre každý typ použitého senzoru, s dĺžkou rovnou hodnote makra `COUNT` daného senzoru. Táto štruktúra je obsiahnutá v hlavnej štruktúre `iop_t`, ktorá okrem týchto najnovších hodnôt obsahuje aj handlers na jednotlivé periférie (USART, SPI, ADC), odkaz na načítací socket HTTP serveru a pod. Táto štruktúra sa vyskytuje len v jednej inštancii, globálnej premennej `iopGlobal`, s ktorou následne pracujú ostatné časti programu.

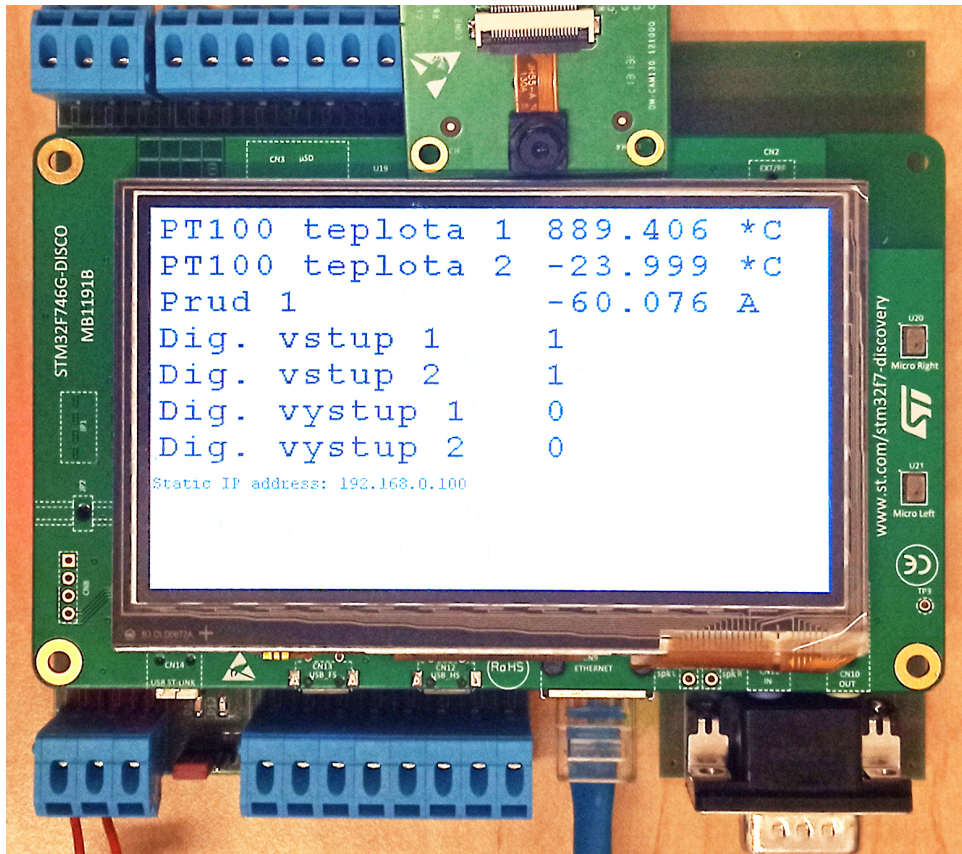
Pre všetky vstupy a výstupy zároveň existujú funkcie:

- `iop_<NAZOV_SENZORU>_init`: Inicializačná funkcia pre daný senzor, je volaná jednorázovo pri inicializácii IOP rozhrania, pred čítaním alebo zapisovaním hodnôt.
- `iop_<NAZOV_SENZORU>_getVals(valueArray)`: Prečíta hodnoty všetkých senzorov daného typu a zapíše ich do poľa 'valueArray'.

Pre výstupy (napr. PID regulátor) je taktiež implementovaná funkcia:

- `iop_<NAZOV_VYSTUPU>_setVal(index, value)`: Nastaví na výstupe s poradovým číslom 'index' (v rámci daného typu výstupu) požadovanú hodnotu 'value'.

Digitálne vstupy a výstupy taktiež implementujú funkcie na čítanie a zápis hodnôt, analogicky k predchádzajúcim funkciám. V ďalších častiach textu sú bližšie popísané spôsoby implementácie týchto funkcií a komunikácie so senzormi pre ich jednotlivé typy.

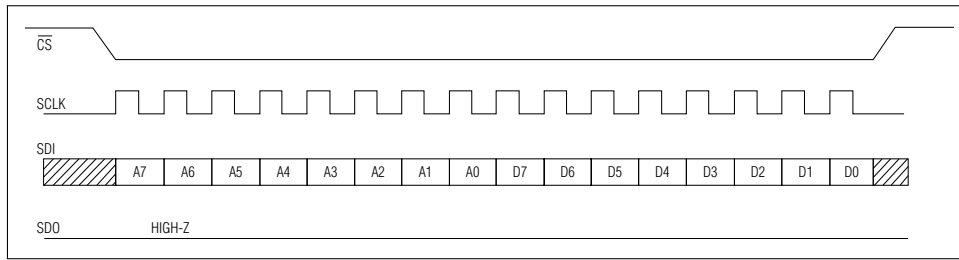


Obrázok 4.3: Zobrazenie hodnôt senzorov na displeji kitu

4.4.1 Digitálne vstupy/výstupy

Inicializačná funkcia `iop_DIGITAL_init()` najprv nastaví príslušné GPIO piny ako vstupy (s interným pull-up rezistorom) alebo výstupy. Definície použitých GPIO portov a čísiel pinov sú nastaviteľné v hlavičkovom súbore `iop/digital.h`.

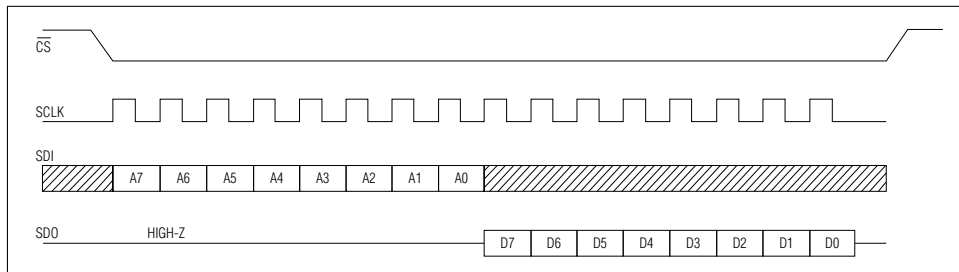
Pre čítanie stavu vstupov a nastavovanie výstupov sa jednoducho používajú funkcie knižnice HAL `HAL_GPIO_WritePin()` a `HAL_GPIO_ReadPin()`. Pretože optočlen TLP2361 použitý v obvodoch digitálnych vstupov poskytuje na výstupe inverznú logickú hodnotu, sú namerané hodnoty vstupov znova invertované v softvéri.



Obrázok 4.4: Časový priebeh zápisu do registra MAX31865 [5, s. 18]

Prevodníky sú po týchto úkonoch pripravené začať meranie odporu. To sa zahájí vždy zápisom do kontrolného registru s bitom `1-shot = 1`. Tým sa zahájí jedno meranie odporu prevodníkom, ktorý zmeria pomer meraného odporu voči odporu referenčnému. Hodnota sa ukladá ako 15 bitov (`q`) v dvoch 8-bitových registroch (`RTD MSBs` a `RTD LSBs`), pričom jej najnižší bit je uložený do druhého najnižšieho bitu LSB registra. Najnižší bit registra LSB indikuje prítomnosť akejkoľvek detekovanej chyby.

Tieto registre sú následne čítané aplikačným mikrokontrolérom pomocou rozhrania SPI ako 2 nadväzujúce bajty. To je možné vďaka skutočnosti, že zariadenie MAX31865 automaticky po prečítaní registra inkrementuje adresu čítania dovedy, pokiaľ je aktivované ($CS = 0$) a prichádzajú mu hodinové takty signálu SCK. Časový priebeh typického čítania jedného bajtu je znázornený na obr. 4.5.



Obrázok 4.5: Časový priebeh čítania z registra MAX31865 [5, s. 18]

Z hodnoty horných 15 bitov sa potom jednoducho vypočíta vlastný odpor teplotného senzoru podľa

$$R_{RTD} = R_{REF} \cdot \frac{q}{2^{15}}. \quad (4.1)$$

Z tejto hodnoty je možné výpočtom odvodiť meranú teplotu t [°C], a to s použitím vzorca pre odpor RTD v závislosti na teplote (3.1):

$$R_{RTD} = R_0 \cdot (1 + \alpha t + \beta t^2) \quad (4.2)$$

$$t = \frac{-\alpha + \sqrt{\alpha^2 + 4\beta \cdot \left(\frac{R_{RTD}}{R_0} - 1\right)}}{2\beta}. \quad (4.3)$$

V prípade vyvolania prerušenia sa teda nastaví ako hodnota tohto senzoru log. 1. V prípade, že medzi dvoma meracími cyklami systému nedošlo k tomuto prerušeniu, resetuje sa hodnota späť na log. 0. Klientska aplikácia reaguje na hodnotu log. 1 pri tomto senzore vydaním výstražného tónu a takýmto spôsobom upozorní používateľa na možné narušenie žihacieho procesu.

4.4.5 PID regulátor

Externý PID regulátor OM 402PID použitý na reguláciu teploty komunikuje so serverom pomocou sériovej linky RS-232. Na strane aplikácie sa používa periféria USART mikrokontroléru, pričom sa využíva metóda prenosu dát pomocou DMA. Vždy sa však čaká na dokončenie prenosu (príjem súčasnej hodnoty, alebo potvrdenie nastavenia požadovanej) pred pokračovaním chodu aplikácie. Inicializačná funkcia (`iop_PID_init()`) konfiguruje túto perifériu na parametre kompatibilné s regulátorom:

Parameter	Hodnota
Modulačná rýchlosť	9600 Bd
Dĺžka slova	8 bitov
Stop bity	1
Paritný bit	žiadnen

Tabuľka 4.5: Použité nastavenie USART pre PID regulátor

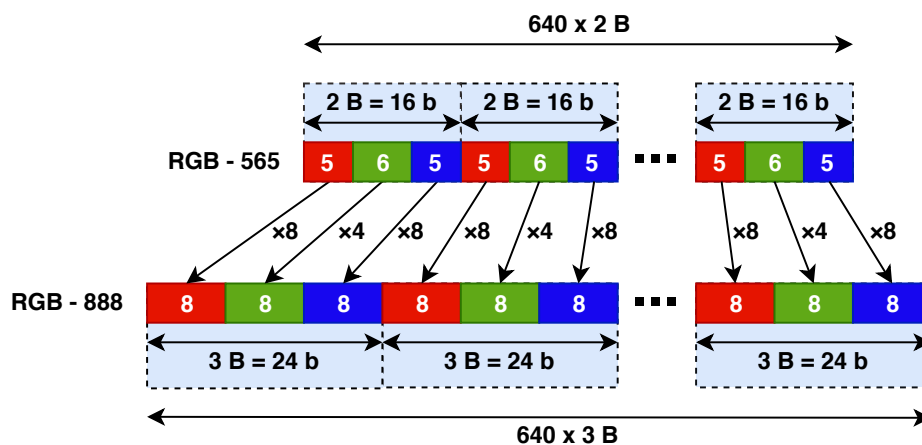
Následne sa s regulátorom komunikuje formou príkaz (od serveru) - odpoveď (od regulátora). Prehľad základných príkazov a ich formátu je uvedený v tabuľke 4.5. Polia **A** predstavujú adresu prístroja, ktorá môže byť v rozsahu 0 až 31. Východiskovou hodnotou adresy je 0, a je možné ju zmeniť v menu prístroja. Polia **D** predstavujú numerické dáta, teda znaky '0'..'9', ',', '-'. Tieto reprezentujú spravidla súčasnú nameranú hodnotu, alebo žiadanú hodnotu, podľa typu príkazu.

Akce	Přenášená dat										
Vyžádání dat (PC)	#	A	A	<CR>							
Vysílání dat (Přístroj)	>	R	<SP>	D	D	D	D	D	(D)	(D)	<CR>
Potvrzení příkazu (Přístroj) - OK	!	A	A	<CR>							
Potvrzení příkazu (Přístroj) - Bad	?	A	A	<CR>							
Identifikace přístroje	#	A	A	1Y	<CR>						
Identifikace HW	#	A	A	1Z	<CR>						
Dočasné nastavení žádané hodnoty	#	A	A	2h	D	D	D	D	D	D	<CR>
Start programu (1...14), 0=OFF	#	A	A	2N	D	D	<CR>				
Start regulace na pevnou hodnotu	#	A	A	3N	<CR>						

Tabuľka 4.6: Popis sériovej komunikácie OM 402PID [14]

chybné obrázky, ktoré nebolo možné prečítať. Po dôkladnej kontrole binárneho obsahu súborov bolo zistené, že obsahujú správnu JPEG hlavičku, avšak neobsahujú potrebné tabuľky. Jednalo sa teda buď o chybu použitia či konfigurácie knižnice, čomu po dôkladnej kontrole kódu nič nenasvedčovalo; alebo chybu v knižnici samotnej. Tento problém bol vyriešený dodatočným volaním funkcie `pass_startup()` zo súboru `jcmaster.c`, ktorá tieto tabuľky zapisuje do výstupu.

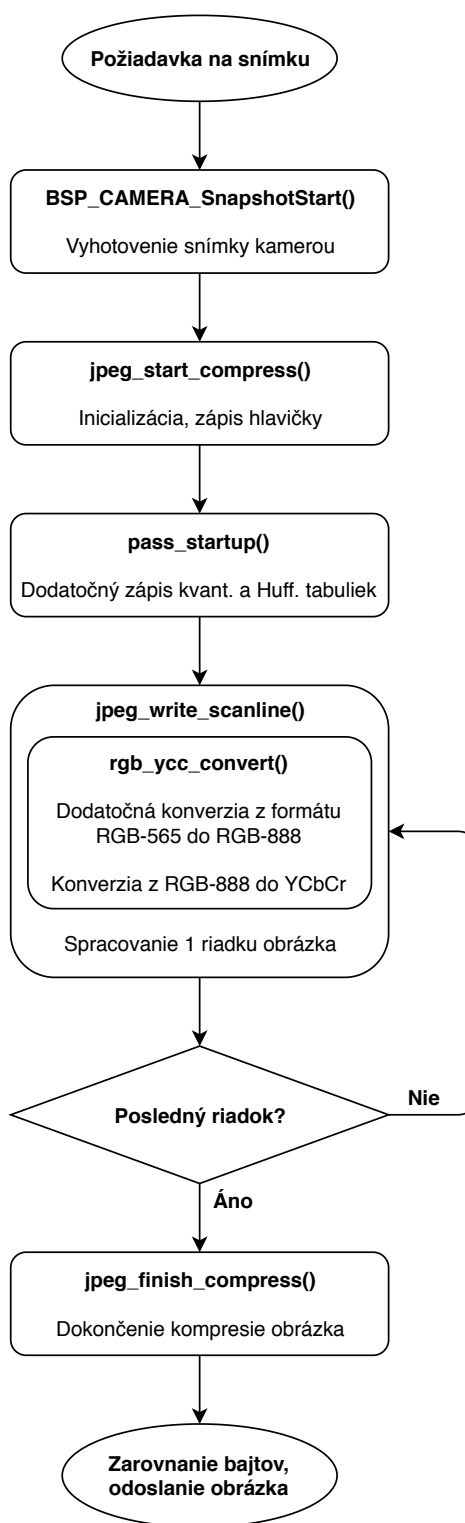
Spracovanie obrázka pri kompresii prebieha po riadkoch, na ktoré je postupne volaná prevodná funkcia `jpeg_write_scanlines()`. Táto funkcia riadok skomprimuje do formátu JPEG a pridá ho na koniec výstupu. V rámci nej je použitá funkcia `rgb_ycc_convert()` zabezpečujúca konverziu z farebného formátu RGB do YCbCr používaného vo formáte JPEG. Pretože ale obraz z kamery používa RGB-565 namiesto štandardného RGB-888, bola do tejto funkcie pridaná pomocná konverzia (viď obr. 4.6), ktorú je možné zapnúť/vypnúť makrom `JPEG_RGB565_INPUT`. To je potrebné nastaviť správne podľa formátu riadkov na vstupe, v opačnom prípade dôjde k nesprávnej konverzii obrázka, prípadne k pamäťovým chybám.



Obrázok 4.6: Konverzia riadku obrázka z formátu RGB-565 do RGB-888

Po pridaní všetkých riadkov vstupu je konverzia zakončená pomocou `jpeg_finish_compress()`. Následne je hotový JPEG obrázok pripravený v pamäti na odoslanie klientovi. Pri testovaní odosielenia obrázku pomocou LwIP však zjavne náhodne nastávali Hard Fault chyby mikrokontroléra. Po rozsiahlejšom testovaní bolo zistené, že chybu zapríčiňuje nezarovnaný prístup do externej pamäte SDRAM (t.j. čítanie počtu bajtov, ktorý nie je deliteľný 4). Na odstránenie tejto chyby sa so samotným obrázkom odosiela aj 0-3 bajty navyše tak, aby bolo zachované zarovnanie pamäte. Tento postup neovplyvňuje validitu obrázka, pretože tieto dodatočné bajty sa nachádzajú za jeho koncovou značkou a sú teda ignorované softvérom ktorý ho číta.

Celý tento algoritmus je graficky znázornený vo vývojovom diagrame na obrázku 4.7.



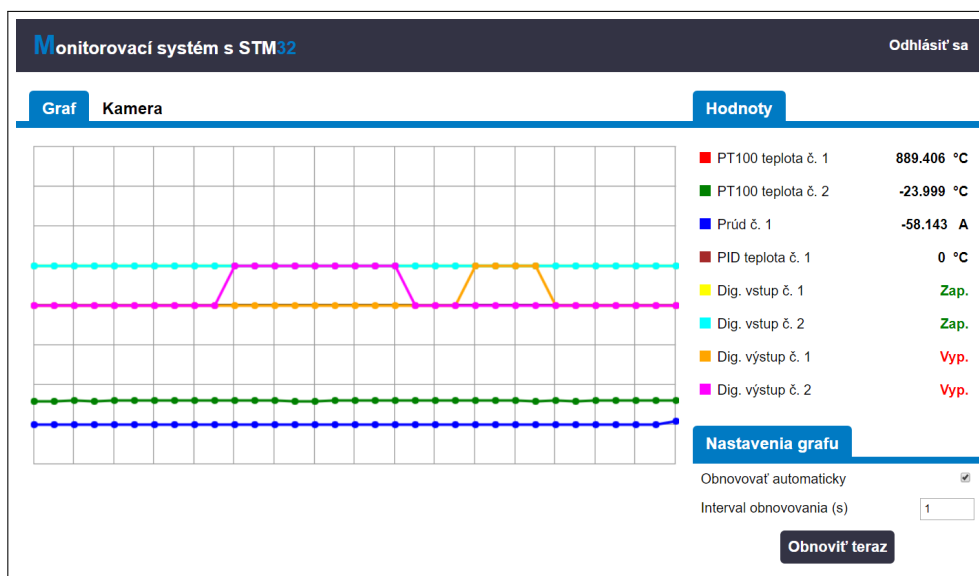
Obrázok 4.7: Diagram spracovania snímok z kamery

Kapitola 5

Klientska aplikácia

Monitorovacia klientska aplikácia je realizovaná formou web stránky. Tá je responzívna, čo znamená, že rozloženie jednotlivých prvkov sa dynamicky mení v závislosti na veľkosti obrazovky klientskeho zariadenia. Tento postup zlepšuje prehľadnosť a použiteľnosť rozhrania pre používateľov rôznych typov zariadení. V tejto aplikácii sa alternatívne rozloženie aktivuje pri rozlíšeníach so šírkou menšou než 950 bodov.

Statické časti stránky sú vytvorené klasicky za použitia jazykov HTML a CSS. Dynamické časti stránky (komunikácia so serverom, vykresľovanie grafu, atď.) sú naprogramované v jazyku Javascript. V celej klientskej časti aplikácie neboli použité žiadne dodatočné softvérové knižnice (ako napr. jQuery). Bolo rozhodnuté, že je možné implementovať požadovanú funkcionálnosť bez nich, a zároveň sa tak ušetrí objem prenášaných dát (zdrojového kódu) pri načítaní aplikácie a jej komplexnosť.



Obrázok 5.1: Klientska aplikácia (graf)



Obrázok 5.2: Klientska aplikácia (kamera)

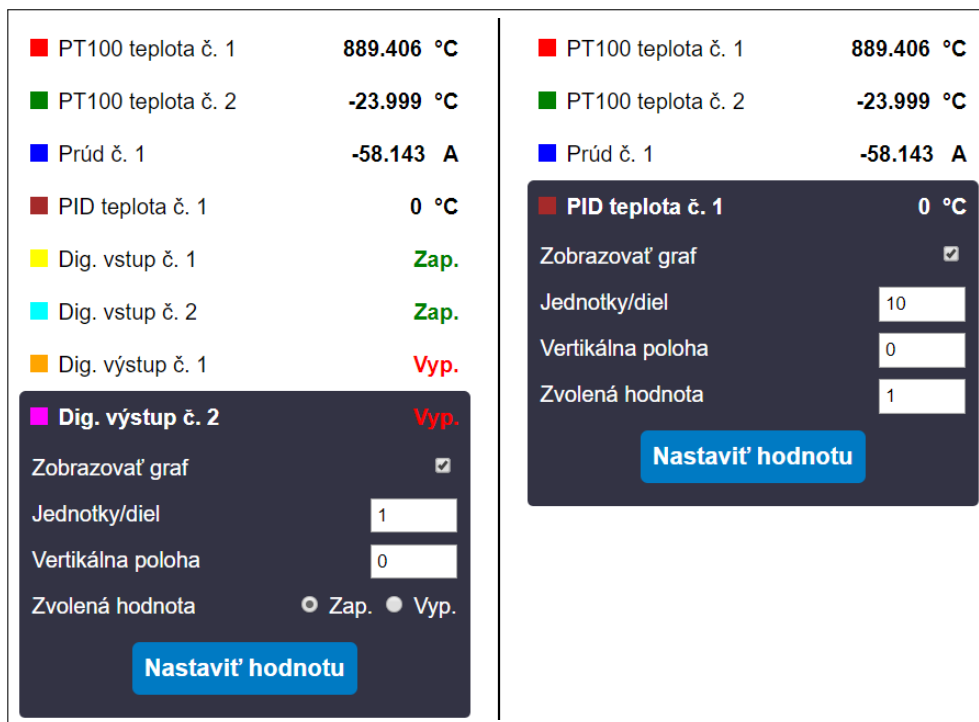
5.1 Používateľské rozhranie

Používateľské rozhranie je rozdelené na viacero častí: hlavička, hlavné okno a vedľajší stĺpec. V hlavičke sa nachádza nadpis aplikácie a v pravej časti odkaz na odhlásenie prihláseného používateľa zo systému. Nad hlavným oknom sa nachádzajú 2 záložky, kliknutím na ktoré sa prepína obsah hlavného okna:

- *Graf*: zobrazenie grafu časového priebehu nameraných dát (viď obr. 5.1)
- *Kamera*: zobrazenie obrazu z kamery (viď obr. 5.2)

Vo vedľajšom stĺpci sa nachádza zoznam aktívnych vstupov (senzorov) a výstupov. Pri každom z nich sú znázornené postupne: farba v grafe, názov, aktuálna hodnota s jednotkou. Po kliknutí na jednu z týchto položiek sa zobrazí menu príslušného vstupu/výstupu, viď. obr. 5.3. Toto menu obsahuje nastavenia jeho zobrazenia v grafe:

- *Zobrazovať graf*: zaškrtnuté políčko, ktorým sa zapína/vypína zobrazovanie hodnôt tohto senzoru v grafe.
- *Jednotky/diel*: číselný vstup, označujúci koľko jednotiek meranej veličiny sa má zobraziť na jeden vertikálny diel grafu.
- *Vertikálna poloha*: číselný vstup, nastavuje sa ním vertikálny posun hodnôt daného senzoru v grafe. Jednotkou je diel grafu, teda napr. hodnota 1 znamená posun o 1 diel grafu nahor, hodnota -0.5 potom posun o polovicu dielu nadol.



Obrázok 5.3: Menu vstupov/výstupov v klientskej aplikácii

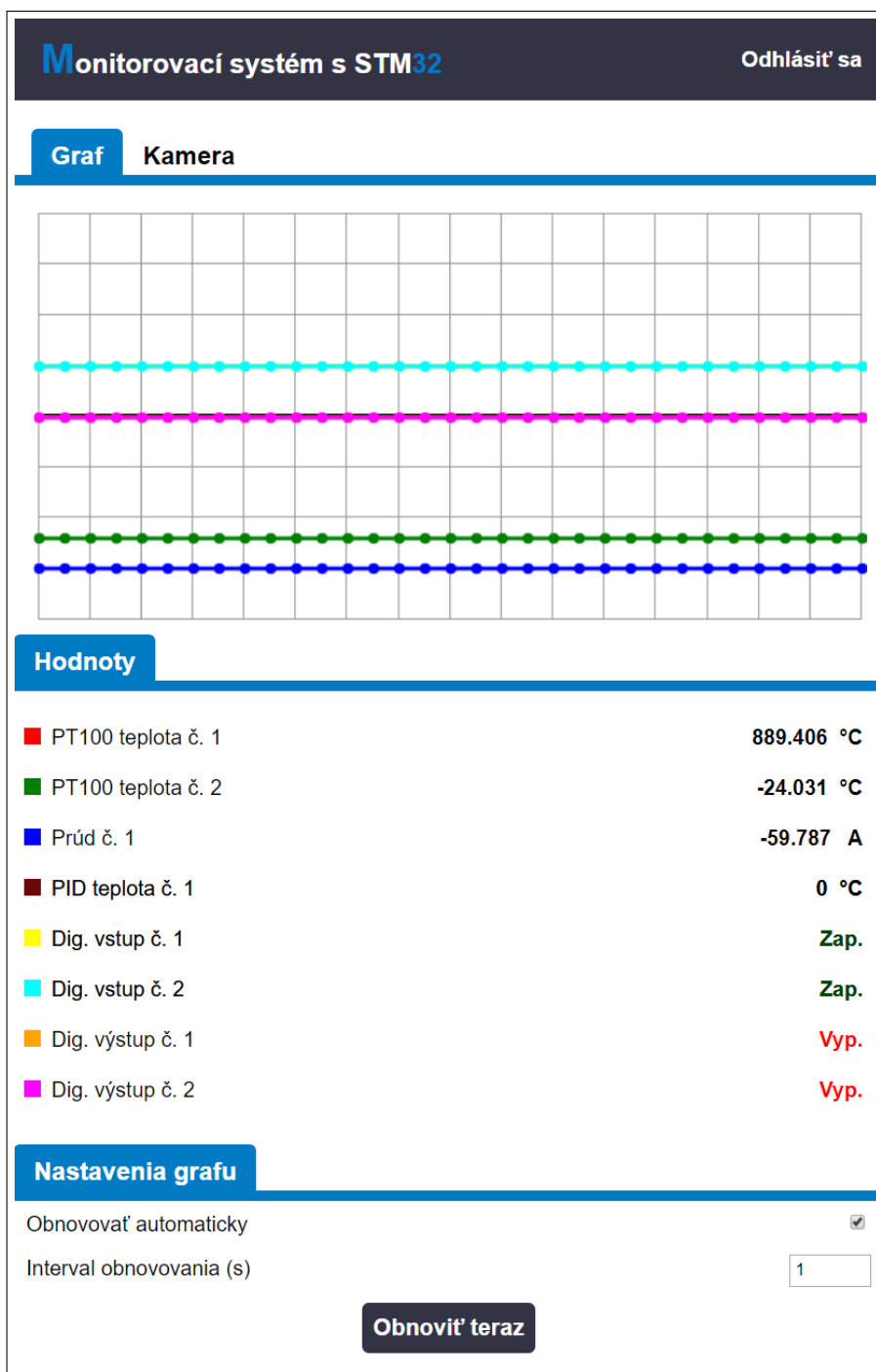
Pre výstupy je toto menu ešte rozšírené o možnosť nastavenia žiadanej hodnoty, a to prvkami:

- *Zvolená hodnota*: výberové prepínače Zap./Vyp. pre prípad digitálneho výstupu; inak číselný vstup pre analógový výstup, označujúce požadovanú hodnotu výstupu.
- *Nastaviť hodnotu*: tlačidlo potvrdzujúce nastavenie požadovanej hodnoty, po jeho stlačení sa odosiela požiadavka serveru.

Týmto spôsobom je teda možné v klientskej aplikácii monitorovať aktuálne hodnoty vstupov, upravovať ich zobrazenie v grafe, a nastavovať požadované hodnoty výstupov. Pod zoznamom vstupov/výstupov sa vo vedľajšom stĺpci nachádzajú taktiež nastavenia aktívnej záložky hlavného okna, teda grafu a kamery, ktoré sú:

- *Obnovovať automaticky*: zaškrtnuté políčko určujúce prítomnosť automatického obnovovania aktuálnych hodnôt sensorov, resp. obrazu z kamery.
- *Interval obnovovania (s)*: číselný vstup udávajúci interval obnovovania dát (sensorov, kamery), v sekundách.
- *Obnoviť teraz*: tlačidlo, ktorým sa vykonáva jednorázové okamžité obnovenie dát sensorov alebo snímku z kamery. V prípade, že v momente jeho stlačenia je aktivované automatické obnovovanie, vyšle sa požiadavka navyše a neovplyvní sa tým pravidelný cyklus obnovovania.

Celý tento vedľajší stĺpec rozhrania sa štandardne nachádza napravo od hlavného okna, avšak v responzívnom móde sa presúva pod toto okno, vid' obr. 5.4. Týmto spôsobom sa uvoľňuje väčší priestor pre zobrazenie grafu alebo obrazu z kamery a zároveň sa zachováva veľkosť textu a prehľadnosť rozhrania, napr. zoznamu aktuálnych hodnôt.



Obrázok 5.4: Klientska aplikácia (graf) v responzívnom móde

5.2 Prijímanie dát zo senzorov

Komunikácia klientskej aplikácie so serverom prebieha pomocou HTTP protokolu za využitia technológie AJAX. Jedná sa o asynchrónnu HTTP komunikáciu v jazyku Javascript. Prenos dát začína vždy odoslaním požiadavky na ne serveru, a to konkrétne požiadavkou na URL `/getIO` s metódou GET, teda ekvivalent načítania súboru z web serveru. Server následne odpovedá s najaktuálnejšími hodnotami senzorov. Dáta sa prenášajú v takomto jednoduchom textovom formáte:

```
NAZOV_SENZORU_1=HODNOTA_SENZORU_1
NAZOV_SENZORU_2=HODNOTA_SENZORU_2
...
NAZOV_SENZORU_N=HODNOTA_SENZORU_N
```

Klientska aplikácia následne túto odpoveď spracuje a obnoví hodnoty zobrazené v používateľskom rozhraní. Pri spracovaní sa rozloží odpoveď na riadky (jeden riadok na senzor). Následne sa pre každý z nich hľadá zhoda názvu senzoru v odpovedi serveru s názvami známych senzorov v zozname klientskej aplikácie. V prípade zhody sa nahradí uložená hodnota pre daný senzor novou hodnotou z tejto odpovede. Jednotky veličín meraných jednotlivými senzormi sa neprenášajú, ale sú uložené lokálne pre každý senzor v oboch častiach aplikácie (klient, server).

Spolu so zobrazením aktuálnych hodnôt senzorov je možné zobrazovať aj graf ich priebehu v čase. Spôsob jeho implementácie je vcelku jednoduchý, pričom sa nepoužívajú knižnice, ale naopak sa kreslí priamo do elementu typu `canvas`. Vždy sa ukladá niekoľko posledných hodnôt každého senzoru (počet je daný konštantou `IOGRAPH_DRAW_SAMPLES_COUNT`). Následne sa pri prekresľovaní grafu (napr. po obnovení hodnôt, či zmene veľkosti okna prehliadača) postupne vykreslia body označujúce tieto hodnoty pre každý senzor, interpolované úsečkami. Škálovanie a posun týchto hodnôt je riadené nastavením v menu daného senzoru (viď podkapitola 5.1). Farba bodov a úsečiek medzi nimi je potom daná statickým nastavením v zdrojovom kóde, špecifická pre daný senzor.

V prípade, že je aktivované automatické obnovovanie týchto hodnôt, sa po prijatí odpovede taktiež spustí časovač doby danej nastavenou obnovovacou periódou. Po jeho vypršaní sa opakuje odosielanie požiadavky na nové hodnoty senzorov a cyklus sa opakuje.

5.3 Nastavovanie výstupov

Pri nastavovaní výstupov sa serveru posiela POST požiadavka na URL `/setIO`, pričom v dátovej časti (tele HTTP požiadavky) sa nachádzajú dáta označujúce požadované hodnoty výstupov. Formát týchto dát je rovnaký, ako formát prijímaných dát zo senzorov, viď vyššie. Po prijatí tejto požiadavky ju server spracuje a nastaví požadované výstupy. Jeho odpoveďou sú následne najnovšie zmerané hodnoty, ktoré sú spracované rovnako ako v 5.2.

V prípade úspešného nastavenia výstupov budú tieto nové zmerané hodnoty odzrkadľovať práve nastavené požadované hodnoty. Pre digitálne výstupy sa teda budú priamo rovnať, a pre PID regulátor sa jeho aktuálna meraná hodnota začne približovať k požadovanej. Ak k tomuto nedôjde, zjavne sa vyskytla chyba v systéme, a je potrebné ju diagnostikovať.

5.4 Prijímanie obrazu z kamery

K obrazu z kamery sa prístupuje jednoducho zaslaním GET požiadavky na adresu `/getCam`. Odpoveďou serveru (po vyhotovení snímku a spracovaní) je priamo JPEG obrázok, ktorý sa následne vykresľuje v hlavnom okne aplikácie. Je ale samozrejme možné k tejto adrese prístupiť aj priamo a zobrazíť len samotný obrázok, napr. za účelom jeho stiahnutia či ďalšieho spracovania. Za účelom zabrániť cache-ovaniu obrázka, t.j. aby bol pri každom obnovení obrázok naozaj aktualizovaný, sa však v skutočnosti prístupuje k URL v tvare `/getCam?X`, kde X je náhodne generované číslo.

V prípade, že je aktivované automatické obnovovanie obrazu z kamery, sa po načítaní obrázka spustí časovač na opätovné načítanie snímku, analogicky k postupu v 5.2. Jedným rozdielom však je, že požiadavky na snímky z kamery sa neposielaajú v prípade, že nie je aktivovaná záložka kamery. Takýmto spôsobom sa obmedzuje objem prenášaných dát v čase, kedy by obraz z kamery aj tak nebol viditeľný.

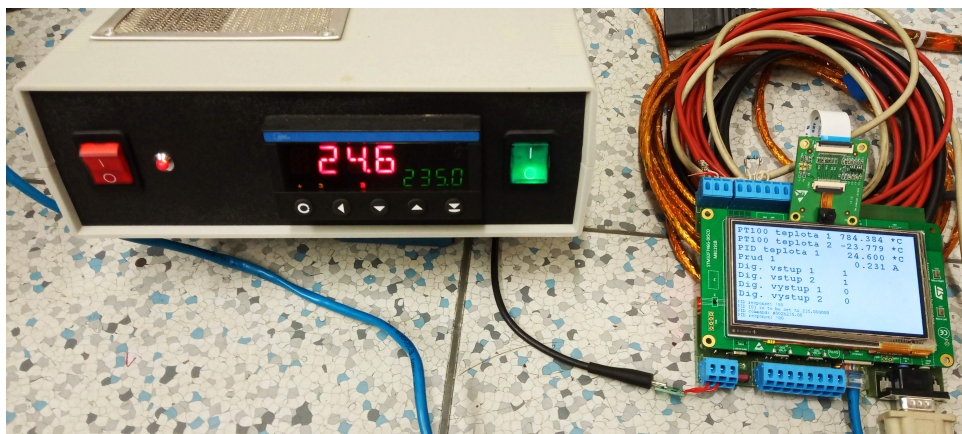
Kapitola 6

Testovanie systému

6.1 PID regulátor

Pre testovanie komunikácie s PID regulátorom bol k nemu najprv pripojený monitorovací systém pomocou sériovej linky RS-232 cez konektor DB-9 na rozširujúcej doske. Po spustení regulátora a následne monitorovacieho systému, bola na displeji kitu správne zobrazovaná súčasná teplota zmeraná regulátorom (24,6 °C, červenou).

Následne bolo odskúšané nastavenie požadovanej hodnoty (235 °C) z klient-skej aplikácie. Tento úkon bol taktiež úspešný, čo bolo indikované správnym nastavením regulovanej hodnoty na displeji regulátora (zelenou). Testovanie bolo opakované viackrát s rôznymi hodnotami a v každom prípade komunikácia fungovala správne. Testovanie je fotograficky zachytené na obr. 6.1.



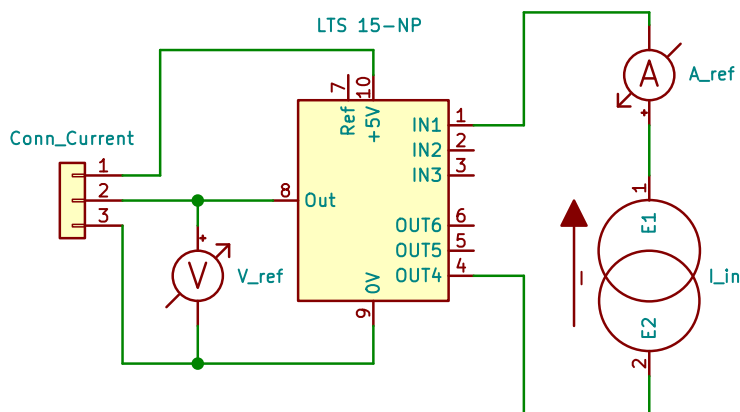
Obrázok 6.1: Test komunikácie s PID regulátorom

6.2 Senzor prúdu

Pre testovanie merania prúdu bolo realizované zapojenie podľa obr. 6.2. Boli testované prúdy o veľkosti od 0 A do 8 A s krokom približne 0,250 A (od 0 A do 3 A), respektíve 0,500 A (od 3 A do 8 A). Polarita prúdu bola zvolená

tak, aby napätie na výstupe senzoru pri rastúcej amplitúde prúdu klesalo od 2,5 V smerom nadol (záporný smer).

Ako referenčný ampérmeter bol použitý laboratórny multimeter Agilent 34401A s externým bočníkom Agilent 34330A s citlivosťou 1 mV/A. Taktiež bolo merané napätie na výstupe senzoru pomocou druhého multimetra rovnakého typu. Z displeja monitorovacieho systému boli následne zaznamenávané prepočítané hodnoty prúdu.



Obrázok 6.2: Testovací obvod merania prúdu

Obrázok 6.3 zaznamenáva nameranú prevodnú charakteristiku samotného senzora LTS 15-NP, t.j. vzťah výstupného napätia k vstupnému prúdu. Graf taktiež na vedľajšej zvislej osi znázorňuje odchýlku medzi ideálnou charakteristikou (podľa rovnice 3.4) a nameranou. Maximálna odchýlka napätia bola nameraná pri nulovom vstupnom prúde $I_{in} = 0$ A, a to

$$\Delta U_{max} = 9,6 \text{ mV},$$

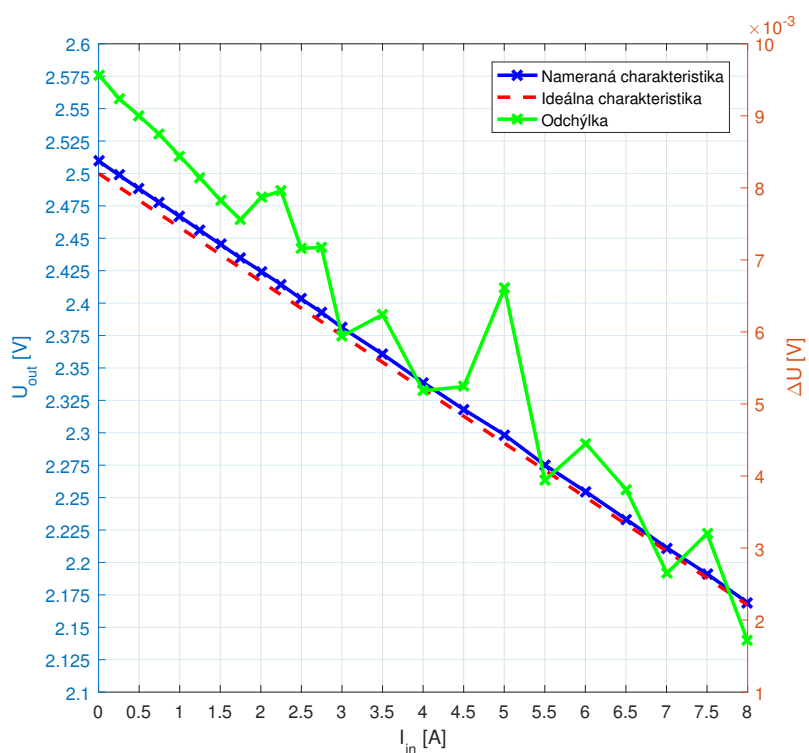
čo predstavuje chybu $\approx 0,24$ % z výstupného rozsahu senzoru (± 2 V). Z pohľadu na charakteristiku je potom vidieť prítomnosť istej chyby zosilnenia, ktorá sa prejavuje rôznymi smernicami ideálnej a nameranej charakteristiky.

Obrázok 6.4 znázorňuje celkovú prevodnú charakteristiku merania, t.j. vzťah medzi vstupným prúdom senzoru, a hodnotou prúdu zmeranou monitorovacím systémom (zobrazovanou na displeji a v klientskej aplikácii). Odchýlka je opäť znázornená na vedľajšej vertikálnej osi.

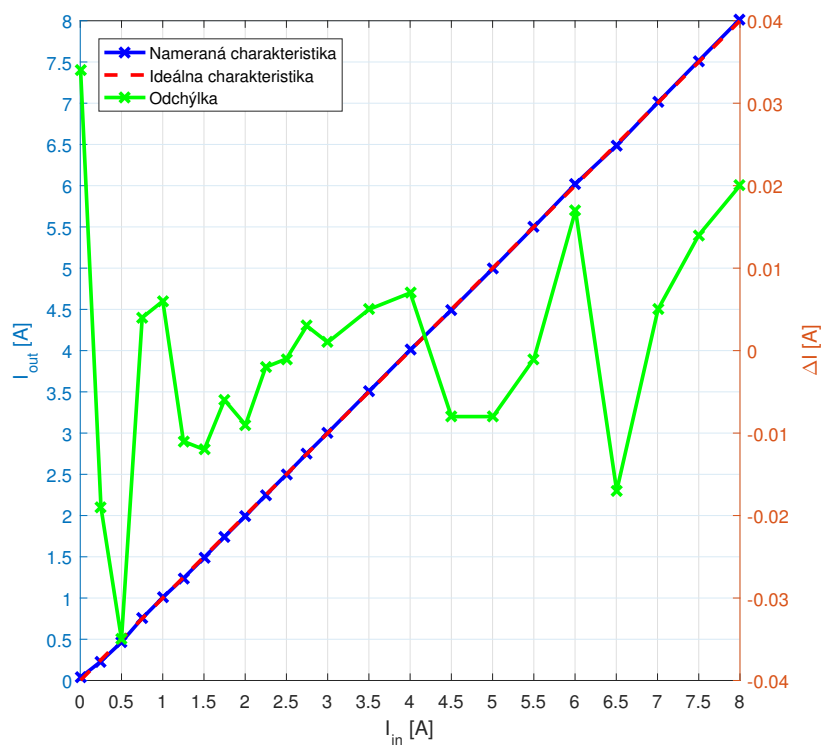
Maximálna nameraná odchýlka prúdu bola pri vstupnom prúde $I_{in} = 0,500$ A, a jej veľkosť bola

$$\Delta I_{max} = 35 \text{ mA},$$

čo činí chybu $\approx 0,11$ % z nominálneho vstupného rozsahu senzoru (± 15 A).



Obrázok 6.3: Nameraná prevodná charakteristika senzoru LTS 15-NP

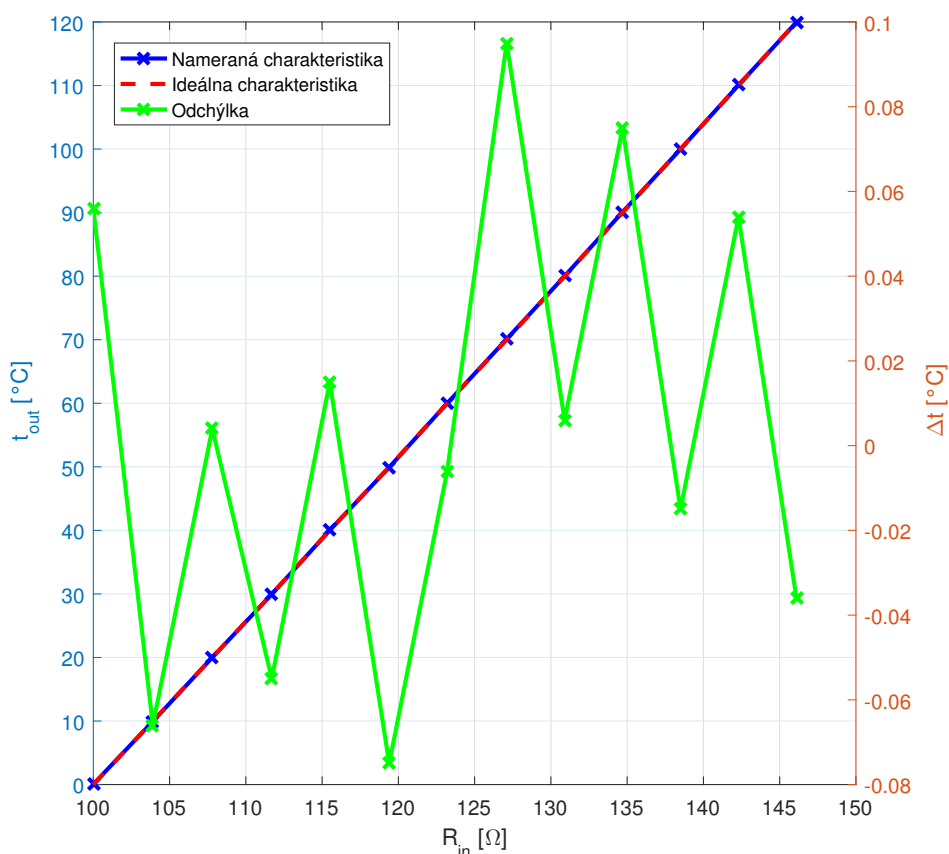


Obrázok 6.4: Nameraná charakteristika merania prúdu

6.3 Senzor teploty

Pre zjednodušenie bol meraný nastaviteľný odpor odporovej dekády namiesto merania teploty samotným odporovým sensorom. Presnosť odporovej dekády bola pred testovaním systému potvrdená meraním laboratórnym multimetrom Agilent 34401A. Následne boli postupne nastavované hodnoty odporu príslušné teplotám 0 až 120 °C, t.j. 100 až 146,1 Ω.

Použitý krok medzi meraniami bol približne 3,85 Ω, čo zodpovedá 10 °C. Pre každú hodnotu odporu bola zaznamenaná výstupná prepočítaná hodnota teploty. Táto nameraná prevodná charakteristika je znázornená v grafe na obr. 6.5, spolu s ideálnou charakteristikou a odchýlkou od nej (na vedľajšej zvislej osi). Všetky merania odporu prebiehali 4-svorkovou metódou za účelom zamedzenia vplyvu odporu prívodov.



Obrázok 6.5: Nameraná charakteristika merania teploty

Maximálna odchýlka bola nameraná pri $R_{in} = 127,1 \Omega$, a to

$$\Delta t_{max} = 0,1 \text{ } ^{\circ}C,$$

čo predstavuje približne 0,08% z meraného rozsahu teplôt.

Kapitola 7

Záver

Bol navrhnutý a vyrobený monitorovací systém vrátane rozširujúcej dosky na pripojenie vstupov/výstupov. Pre vývojový kit STM bol vyvinutý firmvér zabezpečujúci komunikáciu so vstupmi a výstupmi a zároveň funkcionality HTTP serveru. Na ovládanie systému bola vyvinutá klientska aplikácia vo forme webstránky, ktorá poskytuje všetky špecifikované funkcie vzdialeného prístupu k systému, t.j. čítanie hodnôt senzorov, záznam obrazu z kamery a nastavovanie výstupov. Aplikácia je navyše responzívna, čo poskytuje prehľadnejšie rozhranie na prenosných zariadeniach s menšou obrazovkou.

Funkčnosť spojenia rozširujúcej dosky s vývojovým kitom, ako aj správnosť funkcie softvéru boli overené testovaním. Presnosť senzorov teploty a senzoru prúdu boli overené meraniami, pri ktorých sa vyskytovali len pomerne malé chyby. Taktiež bola overená komunikácia monitorovacieho systému s externým PID regulátorom teploty.

Jedinou nedokončenou časťou systému zostal senzor prietoku, ktorý sa žiaľ nepodarilo vybrať a zaobstaráť do termínu spracovania tejto práce. Z tohto dôvodu je preň zatiaľ len pripravené pripojenie k rozširujúcej doske, pričom softvérová obsluha a testovanie senzoru budú doplnené dodatočne. Okrem tejto jednej výnimky bolo teda splnené zadanie práce.

Medzi možné budúce vylepšenia tohto systému patrí napríklad implementácia živého prenosu obrazu z kamery, ktorý by tak dovoľoval nepretržité monitorovanie žihacieho procesu s minimálnou latenciou. Takáto funkcionality by si však s najväčšou pravdepodobnosťou vyžadovala viacvláknové rozloženie aplikácie a značné úpravy klientskej časti.

Taktiež by zrejme bola užitočná možnosť exportovania dát zo senzorov mimo monitorovaciu aplikáciu, napr. vo formáte Excel. Rovnako sa naskytuje priestor rozširovať funkcionality grafu hodnôt senzorov, ako napr. zobrazovanie hodnoty pri prejdení kurzorom ponad bod v grafe, či pridanie meracích kurzorov. Alternatívnym postupom by bol prechod na už existujúcu knižnicu, ktorá poskytuje tieto funkcie, čo by si ale vyžadovalo rozsiahlejší zásah do aplikácie.



Literatúra

- [1] *32F746GDISCOVERY: Discovery kit with STM32F746NG MCU* [online katalógový list]. STMicroelectronics, ©2015 [cit. 12.10.2019]. Dostupné z: https://www.st.com/resource/en/data_brief/32f746gdiscovery.pdf
- [2] *DISCO-F746NG* [online]. Arm Limited, ©2018 [cit. 16.10.2019]. Dostupné z: <https://os.mbed.com/platforms/ST-Discovery-F746NG/>
- [3] MONTROSE, Mark I. *EMC and the printed circuit board: design, theory, and layout made simple*. New York: IEEE Press, ©1999. ISBN 0-7803-4703-X.
- [4] FRADEN, Jacob. *Handbook of modern sensors: physics, designs, and applications*. 3rd ed. New York: Springer, ©2004. ISBN 0-387-00750-4.
- [5] *MAX31865 RTD-to-Digital Converter* [online katalógový list]. Maxim Integrated Products, Inc., ©2015 [cit. 10.5.2019]. Dostupné z <https://datasheets.maximintegrated.com/en/ds/MAX31865.pdf>
- [6] *Current Transducer LTS 15-NP* [online katalógový list]. LEM, ©2017 [cit. 13.05.2019]. Dostupné z: https://www.lem.com/sites/default/files/products_datasheets/lts_15-np.pdf
- [7] *PIR Motion Sensor PaPIRs* [online katalógový list]. Panasonic Corporation, ©2016, [cit. 13.05.2019]. Dostupné z: https://www3.panasonic.biz/ac/cdn/e/control/sensor/human/catalog/bltn_eng_papiers.pdf
- [8] *OM 402PID - Univerzální PID regulátor* [online katalógový list]. ORBIT MERRET, spol. s r. o., ©2017 [cit. 10.5.2019]. Dostupné z: <http://www.merret.cz/sites/default/files/podpora/om402pid2017.1cs.pdf>
- [9] *RM0385 Reference Manual* [online]. STMicroelectronics, ©2018. Dostupné z: https://www.st.com/content/ccc/resource/technical/document/reference_manual/c5/cf/ef/52/c0/f1/4b/fa/DM00124865.pdf/files/DM00124865.pdf/jcr:content/translations/en.DM00124865.pdf

- [10] *FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions* [online]. Dostupné z: <https://www.freertos.org/>
- [11] *lwIP: Overview* [online]. Dostupné z: <http://www.nongnu.org/lwip/>
- [12] *lwIP: Netconn API* [online]. Dostupné z: http://www.nongnu.org/lwip/2_1_x/group__netconn.html
- [13] *lwIP: Socket API* [online]. Dostupné z: http://www.nongnu.org/lwip/2_1_x/group__socket.html
- [14] *OM 402PID - Návod k obsluze* [online]. ORBIT MERRET, spol. s r.o., ©2010 [cit. 10.5.2019]. Dostupné z: <http://www.merret.cz/sites/default/files/podpora/nom402pid20101v0czv.pdf>
- [15] *OM 402PID - Popis funkcí pro RS232/RS485 komunikaci* [online]. ORBIT MERRET, spol. s r.o. Dostupné z: http://www2.merret.cz/podpora/Rs/OM%20402PID_77-002.htm
- [16] *libjpeg* [online]. Independent JPEG Group. Dostupné z: <http://libjpeg.sourceforge.net/>