

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE**

**Fakulta chemickej a potravinárskej technológie**

Evidenčné číslo: FCHPT-5414-44211

# **Vzdialené laboratórium pre systémy s rýchlou dynamikou**

**Diplomová práca**

**SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE**

**Fakulta chemickej a potravinárskej technológie**

Evidenčné číslo: FCHPT-5414-44211

## **Vzdialené laboratórium pre systémy s rýchlou dynamikou**

**Diplomová práca**

Študijný program: automatizácia a informatizácia v chémii a potravinárstve

Číslo študijného odboru: 2621

Názov študijného odboru: 5.2.14. automatizácia

Školiace pracovisko: Ústav informatizácie, automatizácie a matematiky

Vedúci záverečnej práce: Ing. Martin Kalúz, PhD.

**Bratislava 2015**

**Bc. Daniel Janočko**

## **Zadanie práce**



## ZADANIE DIPLOMOVEJ PRÁCE

Študent: **Bc. Daniel Janočko**

ID študenta: 44211

Študijný program: automatizácia a informatizácia v chémii a potravinárstve

Študijný odbor: 5.2.14. automatizácia

Vedúci práce: Ing. Martin Kalúz, PhD.

Názov práce: **Vzdialené laboratórium pre systémy s rýchlou dynamikou**

Špecifikácia zadania:

V rámci projektu je uvažovaný návrh a realizácia vzdialeného laboratória, ktorého súčasťou bude laboratórny proces magnetickej levitácie CE152. Takýto systém je charakteristický veľmi rýchlou dynamikou a preto je nutné voliť iné návrhové postupy pre mechanizmus toku dát v rámci vzdialeného laboratória, ako je tomu u menej rýchlych procesov. Uvažuje sa použitie komunikačných protokolov pre nezávislý obojsmerný tok dát, tak ako aj kombinácia dávkovania dát.

Úlohy:

1. oboznámiť sa s vývojovou platformou Arduino Yún,
2. naštudovať možné spôsoby komunikácie,
3. programovo realizovať vhodné komunikačné rozhranie,
4. naštudovať základy elektronických obvodov a ich využitie pre signálové rozhrania,
5. pripojiť laboratórny proces k platforme prostredníctvom elektrických signálov,
6. overiť funkčnosť ovládania na diaľku a zberu dát,
7. napísať technickú dokumentáciu.

Rozsah práce: 50

Zoznam odbornej literatúry:

1. Wescott, T. *Applied Control Theory for Embedded Systems*. Oxford: Elsevier Inc., 2006. 303 s. ISBN 978-0-7506-7839-1.
2. Crockford, D. *JavaScript: The Good Parts*. USA: O'Reilly Media, 2008. ISBN 978-0-596-51774-8.

Riešenie zadania práce od: 16. 02. 2015

Dátum odovzdania práce: 24. 05. 2015

**L. S.**

**Bc. Daniel Janočko**

študent

**prof. Ing. Miroslav Fikar, DrSc.**

vedúci pracoviska

**prof. Ing. Miroslav Fikar, DrSc.**

garant študijného programu

## Čestné prehlásenie

Čestne prehlasujem, že som túto diplomovú prácu vypracoval samostatne s použitím uvedených zdrojov.

V Bratislave, dňa 23. mája 2015

.....

## Podakovanie

Týmto by som sa rád podakoval môjmu školiteľovi, Ing. Martinovi Kalúzovi PhD., za jeho odborné rady, pripomienky a ochotu pri vedení písania mojej diplomovej práce.

## Abstrakt

Práca sa zaoberá využitím otvorenej vývojovej platformy Arduino, konkrétne využitiu mikrokontroléra Arduino YÚN pre riadenie laboratórneho procesu magnetickej levitácie a následným spracovaním dát pre užívateľa. Úvodná kapitola rozoberá problematiku práce, a teda sleduje proces magnetickej levitácie po fyzikálnej a matematickej stránke. V jednotlivých kapitolách sa rozoberajú už aplikované systémy vzdialených laboratórií. Následné sa práca venuje rozboru mikrokontroléra Arduino, a to po stránke technologickej i aplikačnej. Nakoniec sa modeluje systém magnetickej levitácie a aplikuje sa vhodný regulátor. Praktická časť práce je rozdelená na dve kapitoly. Prvá rozoberá aplikáciu riadenia v reálnom čase pomocou mikrokontroléra a druhá rozoberá technológiu WebSocket pre zdieľanie nameraných údajov užívateľovi. Posledná sekcia práce sa podrobne venuje funkcionalite klientskej časti. Výsledkom je poukázanie na presné riadenie v reálnom čase magnetickej levitácie využitím nízkonákladového mikrokontroléra Arduino, ktoré je možné sledovať a ovládať bez nutnosti fyzickej prítomnosti užívateľa pri procese.

**Kľúčové slová:** magnetická levitácia CE152, Aruino YÚN, riadenie v reálnom čase, WEB



# Abstract

This thesis studies the usage of Arduino development platform (more specifically the Arduino YÚN micro-controller) to control magnetic levitation in laboratory environment and to process and show output data for user. First chapter describes the system of magnetic levitation from both mathematical and physical perspective. The following chapters are dedicated to applied systems of remote laboratories. After that, we describe the technological principles of Arduino micro-controller. Finally, we propose a model of magnetic levitation system while applying the appropriate controller. Practical part is divided into two chapters. The first chapter deals with the real time implementation of control scenario using the Arduino micro-controller. The second chapter describes the communication transport layer based on WebSocket technology, which is used to deliver and show data to user. The very last part of this thesis is dedicated to description of the client-side application. We have shown that magnetic levitation can be controlled remotely in real time mode by using low cost micro-controller.

**Keywords:** magnetic levitation CE152, Aruino YÚN, real-time controlling, WEB



# Zoznam obrázkov

2.1	Schéma uzavretého regulačného obvodu . . . . .	20
2.2	Štandardné laboratórium . . . . .	22
2.3	Schéma vzdialeného laboratória . . . . .	22
2.4	Schéma platformy WebLabDeusto . . . . .	25
2.5	Systém ISES . . . . .	26
2.6	Bloková schéma Magnetickej levitácie CE152 . . . . .	27
2.7	Zariadenie Magnetickej levitácie CE152 . . . . .	27
2.8	Arduino YÚN . . . . .	31
2.9	Porty a čipy zariadenia Arduino YÚN . . . . .	32
2.10	Schéma zariadenia Arduino YÚN . . . . .	33
2.11	Vývojové prostredie Arduinio IDE . . . . .	34
2.12	Linino . . . . .	35
2.13	Ideálna forma PID regulátora . . . . .	37
3.1	Priebeh riadiacej veličiny . . . . .	45
3.2	Priebeh riadenej veličiny . . . . .	45
4.1	Meranie rýchlosti čítania zo sériovej zbernice . . . . .	48
4.2	Meranie rýchlosti zapisovania dát do sériovej zbernice . . . . .	48
4.3	Úvodná obrazovka . . . . .	55
4.4	Ovládacie tlačidlá . . . . .	56
4.5	Vykresľovanie dát do grafu . . . . .	57
4.6	Zmena žiadanej veličiny . . . . .	59
4.7	Zmena parametrov PID regulátora . . . . .	60
4.8	Stiahnutie nameraných údajov . . . . .	61



# Obsah

<b>1</b>	<b>Úvod</b>	<b>15</b>
<b>2</b>	<b>Úvod do problematiky</b>	<b>17</b>
2.1	Fyzikálna podstata . . . . .	17
2.1.1	Magnetizmus . . . . .	17
2.1.2	Elektromagnetizmus . . . . .	19
2.1.3	Magnetická levitácia . . . . .	19
2.2	Vzdialené laboratória . . . . .	22
2.2.1	História vzdialených laboratórií . . . . .	22
2.2.2	Súčasný stav vývoja vzdialených laboratórií . . . . .	23
2.3	Technická realizácia . . . . .	26
2.3.1	Magnetická levitácia CE152 . . . . .	26
2.3.2	Matematický model magnetickej levitácie . . . . .	28
2.3.3	Arduino . . . . .	30
2.3.4	Linino . . . . .	35
2.3.5	Protokol WebSocket . . . . .	36
2.4	Riadenie systému . . . . .	37
2.4.1	PID regulátor . . . . .	37
2.4.2	Riadenie v reálnom čase . . . . .	40
<b>3</b>	<b>Implementácia riadenia</b>	<b>43</b>
3.1	Syntéza regulátora . . . . .	43
3.2	Ladenie regulátora . . . . .	44
3.3	Riadiaci algoritmus . . . . .	46

<b>4</b>	<b>Vzdialené riadenie a sledovanie</b>	<b>47</b>
4.1	Meranie rýchlosti sériovej linky . . . . .	47
4.1.1	Meranie čítania dát zo zbernice . . . . .	47
4.1.2	Meranie zápisu dát do sériovej zbernice . . . . .	47
4.2	WebSocket server . . . . .	49
4.3	Klientská časť . . . . .	54
4.3.1	Pripojenie . . . . .	54
4.3.2	Úvodná obrazovka . . . . .	55
4.3.3	Ovládacie tlačidlá . . . . .	56
4.3.4	Sledovanie pozície guľôčky . . . . .	57
4.3.5	Zmena hodnoty žiadanej veličiny . . . . .	58
4.3.6	Zmena hodnôt PID regulátora . . . . .	59
4.3.7	Stiahnutie nameraných údajov . . . . .	60
<b>5</b>	<b>Diskusia a záver</b>	<b>63</b>
	<b>Literatúra</b>	<b>64</b>
	<b>Prílohy</b>	<b>67</b>

# Kapitola 1

## Úvod

Vo svete priemyselných a komunikačných technológiách zohrávajú v dnešnej dobe veľkú rolu priemyselné riadiace automaty. Tieto priemyselné počítače riadia mnoho výrobných procesov v rôznych výrobných podnikoch. Táto používaná technológia však naráža na problémy ako zložitosť programovania či cena zariadenia. Odhliadnuc od vizualizácie je možné riadiť procesy aj iným zariadením, ako napríklad mikrokontrolérom. Tieto zariadenia sú pre programátora prístupnejšie, nakoľko zdrojový kód sa píše v jednom zo základných programovacích jazykoch. Cena takéhoto mikrokontroléra je taktiež zanedbateľná oproti ostatným riešeniam.

Nevýhodou mikrokontrolérov je ich obmedzená konektivita, riadenie iba na základe napätových signálov 0-5 V, ich obmedzená pamäť, ale aj absencia priemyselnej certifikácie.

Táto práca sa snaží poukázať na ich veľkú výhodu pri riadení nelineárneho procesu magnetickej levitácie mikrokontrolérom Arduino YÚN. Zariadenie riadi tento veľmi rýchly a nestabilný proces veľmi presne a rýchlo, pričom v práci je implementované riadenie v reálnom čase.

Hlavným cieľom práce je vytvoriť vzdialené laboratórium len s pomocou nízkonákladového mikrokontroléra. V práci sa bude implementovať vhodný a jednoduchý regulátor, ktorý bude riadiť proces magnetickej levitácie. Bude potrebné navrhnuť riešenie pre efektívne a jednoduché vykonávanie riadiacich algoritmov v reálnom čase.

Po aplikovaní riadenia sa bude v práci zabezpečovať spracovanie nameraných dát z procesu používateľom. Bude sa hľadať vhodná technológia, ktorá dokáže veľmi rýchlo prenášať údaje medzi mikrokontrolérom a užívateľskou aplikáciou, ktorou v práci bude webové rozhranie.

Jedným z cieľov práce je vytvoriť používateľovi funkčné a jednoduché webové rozhranie vzdialeného laboratória, pomocou ktorého by užívateľ mohol sledovať pozíciu guľôčky v magnetickom poli, a to v každej perióde vzorkovania. Užívateľ by taktiež mal mať možnosť prenos dát zastaviť, zmeniť hodnotu žiadanej veličiny či ladiť parametre regulátora. To všetko má zabezpečovať nízkonákladový hardvér.



# Kapitola 2

## Úvod do problematiky

### 2.1 Fyzikálna podstata

Aplikovaný proces magnetickej levitácie funguje na báze využitia premenlivého magnetického poľa, ktoré pôsobí fyzikálnou silou na iné magnetické polia, alebo feromagnetické objekty. Zadefinovanie fyzikálnej podstaty tohto je kľúčovým prvkom k pochopeniu deja, ktorý v procese prebieha.

#### 2.1.1 Magnetizmus

Magnetizmus je jav, kde sa aplikuje účinok magnetického poľa, ktoré pôsobí na akúkoľvek látku s magnetickými vlastnosťami.

K magnetizmu dochádza v samotnej štruktúre kovových, alebo iných magneticky vodivých telies, nakoľko dipóly nemagnetických predmetov sú orientované protismerne a tým sa ich magnetický efekt vyruší. Magnetické predmety majú dipóly orientované v jednom smere a vytvárajú vo svojom okolí magnetické pole.

V praxi poznáme dva druhy magnetov a to tzv.:

- Permanentné magnety
- Elektromagnety

Každá látka, zložené s atómov a teda s elektrónov, protónov a neutrónov, má základnú vlastnosť: kvantový mechanický spin. Spin dáva každej tejto častici určité magnetické pole.

V každom atóme a molekule je spin týchto častíc prísne usporiadaný podľa Pauliho vylučovacieho princípu, podľa ktorého sa v častici nemôžu nachádzať elektróny s rovnakým spinorbitálom. Tento princíp usporiadania spinu ale neplatí na veľkú vzdialenosť medzi atómami a molekulami. Bez tohto vzdialeného usporiadania tu nevzniká sieťové magnetické pole, pretože magnetický moment každej z častíc je vyrušený momentom ostatných častíc. ([14, sekcia 2.2.1])

Permanentné magnety sú zvláštne v tom, že u nich vzdialené usporiadanie existuje. Najvyšší stupeň usporiadania existuje v magnetických doménach, čo sú jednotlivé časti štruktúry. Čím vyššia je usporiadanosť v doméne, tým silnejšie je výsledné pole. ([14, sekcia 5.3])

Permanentné magnety na generovanie magnetického poľa nepotrebujú žiadnu vonkajšiu energiu. Magnetické pole v ich okolí je stále a nemenné. Tieto magnety sa bežne vyskytujú v prírode, napríklad v neraste magnetovec. Z pohľadu riadenia sú však pre prácu nezaujímavé, nakoľko nie je možné ich nijako ovplyvniť.

Naproti tomu elektromagnety indukujú magnetické pole iba v prítomnosti dodanej vonkajšej energie, konkrétne elektrickej energie. Nakoľko je to téma obsiahnejšia a hovorí o deji, ktorý sa využíva v práci, je mu venovaná sekcia (2.1.2).

Aby sa dostavil efekt samotného magnetického javu, je nutné umiestniť do magnetického poľa predmet, ktorý taktiež vykazuje magnetické vlastnosti.

Látky, ktoré vykazujú magnetické vlastnosti sa delia na:

- paramagnetické
- diamagnetické
- feromagnetické

Paramagnetické látky sú látky do poľa vťahované, diamagnetické látky sú od poľa odpudzované. Tento jav súvisí s párovaním elektrónov po vložení do magnetického poľa. Pohybujúce sa častice sa párujú a indukujú okolo častíc magnetické pole. To pri interakcii s vonkajším poľom do seba naráža, čo má za následok odpudzovanie, a teda vznik diamagnetizmu. Z tohto odvodu je jasné, že každá látka je vo svojej podstate diamagnetická. To, či bude vykazovať ale jav diamagnetickej, alebo paramagnetickej látky, závisí od zvyšných nespárených častíc. Tieto častice, nakoľko neindukujú priamo magnetické pole, sú priamo priťahované vonkajším poľom. Ak je sila týchto častíc väčšia ako sila indukované spárovanými časticami, je možné hovoriť o paramagnetickej látke.

O feromagnetickú látku hovoríme vtedy, ak sa jedná o paramagnetickú látku, no po odstránení vonkajšieho magnetického poľa sa všetky magnetické vlastnosti látky ihneď vyrušia.

### 2.1.2 Elektromagnetizmus

Medzi elektrickým prúdom a magnetizmom je priamoúmerná veličina magnetická intenzita  $H$ .

$$H = \frac{NI}{l} [Am^{-1}] \quad (2.1)$$

Intenzita magnetického poľa je definovaná ako magnetická sila vyvolaná na jednotku dĺžky pri pôsobení elektrickým prúdom.

Elektromagnet je vo svojej podstate husto navinutý drôt okolo feromagnetického jadra. Takýto elektromagnet sa nazýva *cievka*, prípadne *solenoid*. [1]

Cievka je súčiastka tvorená z feromagnetického jadra, najčastejšie železnej tyče, na ktorom je nahusto navinutý medený drôt. Pôsobením elektrického prúdu sa okolo cievky vytvára magnetické pole.

Ako je možné vidieť z rovnice (2.1), veľkosť magnetickej intenzity je priamoúmerná množstvu závitov drôtu na cievke.

V praxi je jav využívaný v značnej miere, ako napríklad v ohýbaní ocele, v kolajovej doprave, na generovanie elektrickej energie v elektrických automobiloch a iných procesoch.

V práci je magnetická cievka použitá na vybudenie magnetického poľa a vytvorenie javu magnetickej levitácie.

### 2.1.3 Magnetická levitácia

Magnetická levitácia je jav, kedy magnetická sila prekonáva gravitačnú silu.

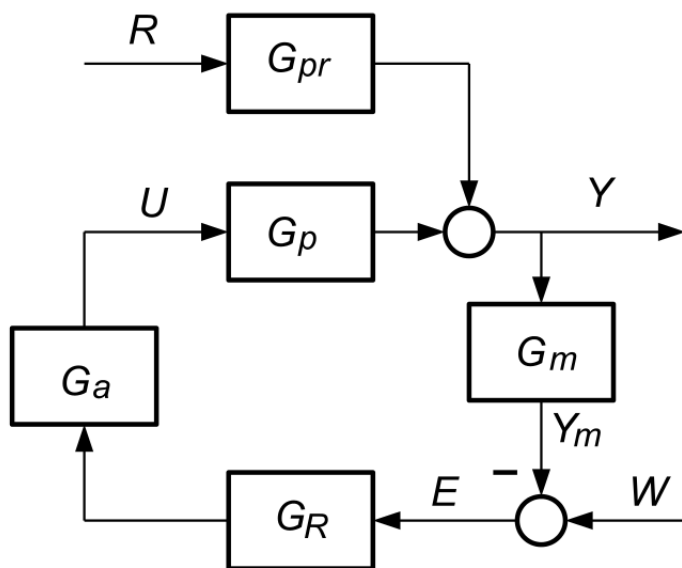
V roku 1842 Samuel Earnshaw prišiel s teóriou, že statické, magnetické alebo elektrické sústavy nábojov sú nestabilné, či už samé o sebe, alebo pod vplyvom gravitácie.

Táto teória hovorí, že žiadna látka, ktorá má levitovať v okolí magnetu, nebude stabilná a bez pohybu sa vznášať, ale vplyv zemskej gravitácie ju prinúti sa pohybovať, minimálne rotáciou okolo svojej osi. [11]

Táto teória je založená pri aplikovaní permanentných magnetov, ktoré sú v stave takzvanej *superkonduktivity*, teda v stave, že je magnet ochladený na teplotu, pri ktorej sa stráca akýkoľvek odpor a stáva sa supermagnetom. Na tejto báze fungujú známe vlaky MagLev v Japonsku.

V práci sa využíva levitácia typu *elektromagnet + feromagnetická látka*. Ide o najjednoduchší spôsob, ako nechať levitovať objekt umiestnený pod magnetickým závesom, teda vo vygenerovanom magnetickom poli elektromagneta. Táto sila presne vyruší gravitačnú silu a tým pádom obchádza Earnshawovu vetu využitím spätnej väzby. Výsledkom je silne nestabilný proces. [13]

Na obrázku 2.1 je možné vidieť jednoduchý uzavretý regulačný obvod, ktorým je možné riadiť takýto systém.



Obr. 2.1: Schéma uzavretého regulačného obvodu

Opis schémy:

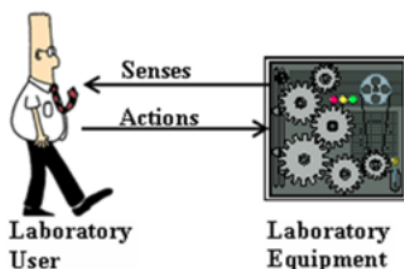
1. užívateľ zadá hodnotu pozície guľôčky, ( $W$ )
2. vypočíta sa regulačná odchýlka ako rozdiel medzi žiadanou hodnotou ( $W$ ) a aktuálnou pozíciou guľôčky ( $Y_m$ ) ( $E$ ),
3. regulátor vypočíta hodnotu akčného zásahu, v procese napätie na cievke, ktorý pošle elektromagnetu,
4. elektromagnet na vstupnú hodnotu zareaguje zmenou intenzity poľa, na ktorý guľôčka zareaguje zmenou pozície v elektromagnetickom poli,

5. prípadný vplyv porúch,  $(R)$ , sa pripočítava k výstupnej veličine  $(Y)$ ,
6. znova sa odmeria aktuálna pozícia guľôčky a proces sa opakuje

Podrobnejšie sa problematike riadenia venuje sekcia 3.3.

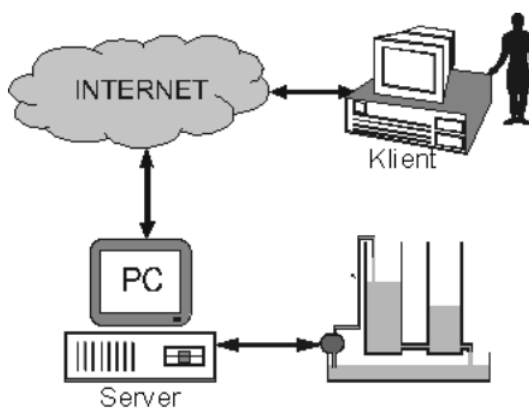
## 2.2 Vzdialené laboratória

Štandardné laboratória fungujú na báze *užívateľ - proces*, pričom užívateľ sa fyzicky nachádza pri procese (obr. 2.2).



Obr. 2.2: Štandardné laboratórium

Užívateľ v laboratóriu sleduje správanie sa systému, generuje akčné zásahy a priamo ich posiela do systému. Naproti tomu vzdialené laboratória fungujú na báze *užívateľ - server - proces*, viď obrázok 2.3.



Obr. 2.3: Schéma vzdialeného laboratória

### 2.2.1 História vzdialených laboratórií

Prvé pokusy s diaľkovým riadením sa datujú do 50. rokov minulého storočia. Raymond Goertz predviedol manipulátor riadený spôsobom master-slave. Mechanizmy pracujúce

na diaľku boli a sú žiadané najmä pre prácu v zamorenom prostredí, v podmorských, alebo vesmírnych výskumoch. [9]

V roku 1964, R. S. Mosher vyvinul pre General Electric systém dvoch robotických ramien s kamerami, ktorý našiel uplatnenie najmä v lekárskej diagnostike, výrobe a manipulácii s mikrosúčiastkami. [7]

### 2.2.2 Súčasný stav vývoja vzdialených laboratórií

Od prvých pokusoch a začiatkoch práce so vzdialenými laboratóriami z druhej polovice 20. storočia urobila technika a ľudstvo obrovský krok vpred.

V súčasnosti sa vzdialené experimenty využívajú na popredných univerzitách a výskumných spoločnostiach kvôli svojim výhodám, ktorými sú napr.:

- **Plná časová flexibilita:** študent môže svoj experiment opakovať či začať večer a nie je obmedzovaný časom;
- **Geografická nezávislosť:** študent alebo pracovník môže svoje experimenty vykonávať z akéhokoľvek miesta s pripojením na Internet;
- **Ekonomické požiadavky:** laboratórium môže byť umiestnené aj v malej miestnosti bez nutnosti platenia veľkých prevádzkových nákladov;
- **Efektivita:** študent môže v jednej chvíli ovládať viacero zariadení;
- **Merané dáta:** študent si vie svoje merané dáta jednoducho stiahnuť a hneď použiť.

Pri spomínaní výhod vzdialených laboratórií netreba zabúdať aj na nevýhody, ktorými sú napr.:

- **Bezpečnosť:** nutné ošetrovanie a plná podpora havarijných situácií;
- **Chybovosť zariadenia:** Absencia manuálneho zásahu pri nečakaných poruchách.

Ukážkovo sú vybraté príklady zo slovenskej a zahraničnej univerzity, ktoré vedú projekty vzdialených laboratórií.

### **Fakulta chemickej a potravinárskej technológie STU v Bratislave**

Ústav informatizácie, automatizácie a matematiky v posledných rokoch aplikoval vzdialené laboratóriá v snahe zefektívniť študentom prácu pri meraní a riadení laboratórnych procesov. Vzdialené laboratóriá sú postavené na priemyselnom aj nízko-nákladovom hardvéri a otvorenom softvéri, a sú prístupné prostredníctvom systému pre správu vzdialených laboratórií WebLab-Deusto, ktorý bol vyvinutý na španielskej univerzite Universidad de Deusto (UD) v Bilbao. Pomocou vzdialených laboratórií je možné vzdialene merať a ovládať experimenty umiestnené na STU aj UD. Prostredníctvom systému sú dostupné nasledovné experimenty:

- tepelno-optická sústava;
- sústava zásobníkov kvapaliny;
- DC motor;
- vzduchový výmenník tepla;
- experiment VISIR (konfigurovateľná elektronika);
- pohyb robota.

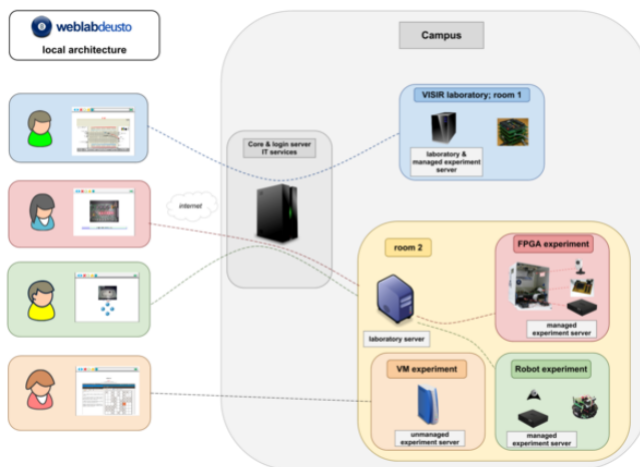
Tieto laboratóriá sú dostupné na stránke <http://weblab.chtf.stuba.sk>;

Systém WebLab-Deusto umožňuje plnú správu užívateľov, čo robí správu laboratória prehľadnou. Každý študent si po registrácii môže rezervovať zariadenie a vykonávať merania. Niektoré experimenty sú dostupné aj neregistrovaným používateľom. Architektúra systému WebLab-Deusto (obr. 2.4) umožňuje prepájať a zdieľať vzdialené laboratóriá, ktoré sú prístupné na rôznych univerzitách vo svete.

Klientská časť WebLab-Deusto je vyvíjaná pomocou projektu Google Web Toolkit, ašak je deklarovaná aj plná podpora vývoja klienta v jazyku Java a pomocou Java Flash.

Serverová časť (jadro, prihlasovací server, server pre laboratória) sú vyvíjané pomocou skriptovacieho jazyka Python. Sú taktiež dostupné aj iné API pre vývoj v iných programovacích jazykoch [2].





Obr. 2.4: Schéma platformy WebLabDeusto

### Matematicko-fyzikálna fakulta Karlovej univerzity v Prahe

Matematicko-fyzikálna fakulta Karlovej univerzity v Prahe prišla s nápadom aplikovať vzdialené laboratória pre výskum rádioaktívnych látok.

Cieľom projektu je priblíženie radiácie študentom, deťom alebo iným dospelým bez strachu z ožiarenia [10].

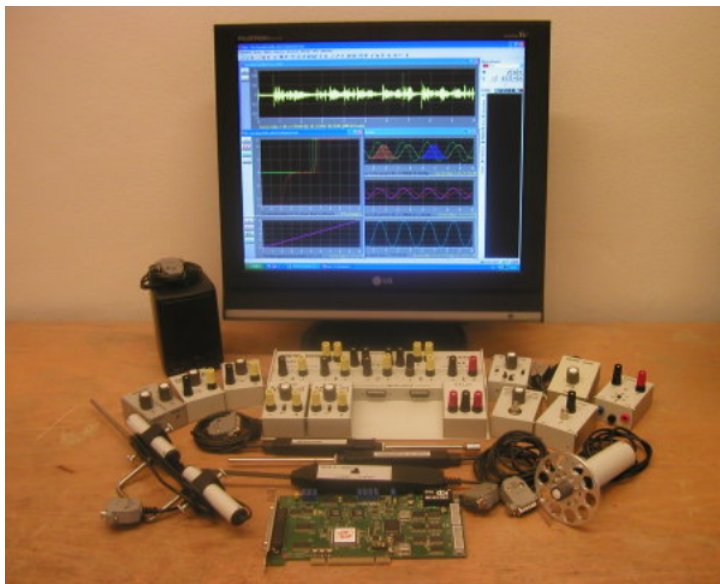
Virtuálne laboratórium ponúka nasledujúce merania

- Monitorovanie prírodného radiačného pozadia;
- Závislosť rádioaktivity na vzdialenosti od žiarica;
- Závislosť rádioaktivity na druhu a hrúbke vrstvy tieniaceho materiálu.

Experiment je založený na nástroji ISES, čo znamená **I**nternetové **Š**kolní **E**xperimentální **S**tudio.

ISES je komplexný nástroj pre zber, spracovanie, zobrazovanie a riadenie experimentov v reálnom čase. ISES je otvorený systém, zložený z fyzického hardvéru a informatického systému. Fyzický hardware je zložený z ISES panela a príslušných modulov, obr. 2.5. ISES panel povoľuje 10 rozličných kanálov (6 analógových a 2 binárne) a je možné ho programovať pomocou 2 programovacích výstupov.

Informatický systém je zložený so vstupnej karty s A/D a D/A prevodníkmi. Po softvérovej stránke je ISIS vybavený programom ISEWIN, pre vzdialené laboratória je



Obr. 2.5: Systém ISES

prístupný software *ISES WEB control* [4].

## 2.3 Technická realizácia

Táto časť práce sa venuje opisu zariadení a služieb využitých pri aplikovaní riadenia fyzikálneho procesu magnetickej levitácie pomocou vzdialeného laboratória.

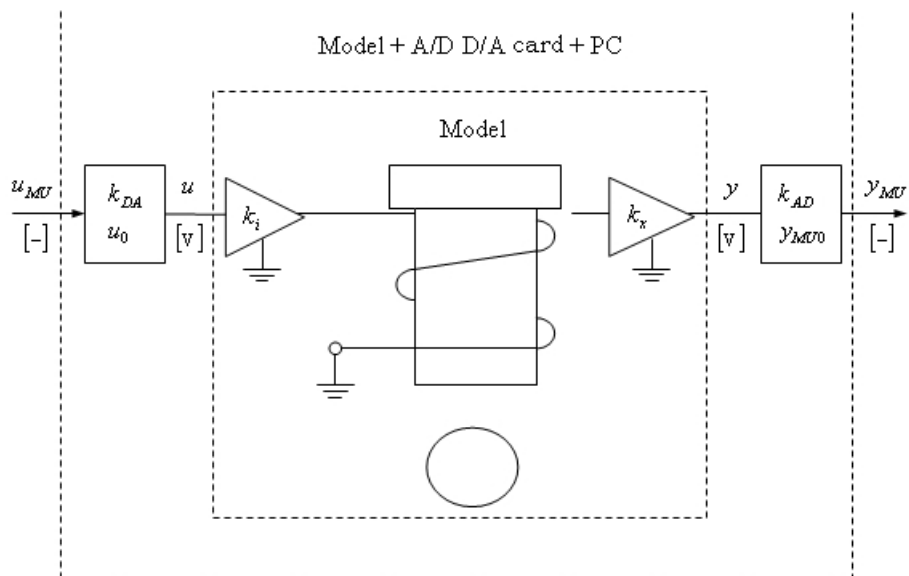
### 2.3.1 Magnetická levitácia CE152

Laboratórna pomôcka Magnetická levitácia CE152 je silne nelineárny a nestabilný proces, určený pre simuláciu riadenia pomocou magnetického poľa.

Na obrázku 2.3.1 je vidieť, že systém sa skladá z cievky, ktorá vybudením magnetického poľa spôsobí levitovanie ocelevej guľôčky v magnetickom poli.

Pozícia ocelevej guľôčky je snímaná lineárnym indukčným snímačom pripojeným k A/D prevodníku. Cievka je riadená výkonovým zosilňovačom pripojeným k D/A prevodníku. Základnou ovládacou úlohou je ovládať pozíciu guľôčky, ktorá sa voľne vznáša v magnetickom poli cievky. Systém magnetickej levitácie je nelineárny dynamický systém s jedným napäťovým vstupom a jedným napäťovým výstupom.

Na obrázku 2.7 je zobrazené reálne zariadenie levitácie.



Obr. 2.6: Bloková schéma Magnetickej levitácie CE152



Obr. 2.7: Zariadenie Magnetickej levitácie CE152

### 2.3.2 Matematický model magnetickej levitácie

Najjednoduchší dynamický opis magnetickej levitácie je možné vyjadriť pomocou bilancie síl

$$F_a = F_g + F_m, \quad (2.2)$$

kde  $F_a$  je výsledná sila pôsobiaca na guľôčku,  $F_g$  je gravitačná sila a  $F_m$  je sila pôsobenia magnetického poľa.

Gravitačná sila sa dá vyjadriť z Newtonovej pohybovej rovnice ako

$$F_g = m_b g, \quad (2.3)$$

kde  $m_b$  je hmotnosť guľôčky a  $g$  je konštanta gravitačného zrýchlenia.

Výsledná sila  $F_a$ , ktorá udeľuje guľôčke zrýchlenie, je vyjadrená v ako

$$F_a = m_b \frac{d^2 x}{dt^2}, \quad (2.4)$$

kde  $x$  je vzdialenosť stredu guľôčky od pozičného senzora.

Sila magnetického poľa vzhľadom na prúd je vyjadrená v tvare

$$F_m = K_m \frac{I_c^2}{x^2}, \quad (2.5)$$

kde  $K_m$  je magnetická konštanta a  $I_c$  je budiaci prúd magnetického poľa.

Pre reálne implementovanie riadenia je potrebné zjednotiť vstupnú a výstupnú veličinu v jednotkách napätia.

Využitím Ohmovho zákona

$$R_c = \frac{U_c}{I_c} \quad (2.6)$$

$$I_c = \frac{U_c}{R_c} \quad (2.7)$$

$$I_c^2 = \frac{U_c^2}{R_c^2} \quad (2.8)$$

vieme vyjadriť budiaci prúd v napäťových jednotkách. Pozícia guľôčky nech je aproximovaná rovnicou priamky

$$U_{pos} = a_{pos}x + b_{pos}, \quad (2.9)$$

potom môžeme z rovnice 2.5 vyjadriť  $x$  ako

$$x = \frac{U_{pos} - b_{pos}}{a_{pos}} \quad (2.10)$$

$$x^2 = \frac{U_{pos}^2 - 2b_{pos}U_{pos} + b_{pos}^2}{a_{pos}^2}. \quad (2.11)$$

Po úpravách získavame vzťah

$$F_m = \frac{K_m}{R_c^2} \frac{U_c^2 a_{pos}^2}{U_{pos}^2 - 2b_{pos}U_{pos} + b_{pos}^2}. \quad (2.12)$$

Zavedením konštanty prúdového zosilňovača

$$K_a = \frac{K_m}{R_c^2} \quad (2.13)$$

získame výsledný vzťah

$$F_m = K_a \frac{U_c^2 a_{pos}^2}{U_{pos}^2 - 2b_{pos}U_{pos} + b_{pos}^2}. \quad (2.14)$$

Dosadenie vyjadrených rovníc síl do rovnice 2.2 dostávame dynamický matematický model

$$\frac{m_b}{a_{pos}} \frac{d^2 U_{pos}}{dt^2} = m_b g - K_a \frac{U_c^2 a_{pos}^2}{U_{pos}^2 - 2b_{pos}U_{pos} + b_{pos}^2} \quad (2.15)$$

kde vstupná veličina  $U_c$  a výstupná veličina  $U_{pos}$  vystupujú v nelineárnych členoch. Na odstránenie nelinearit sa využila aproximácia Taylorovým rozvojom do prvého radu.

Všeobecný predpis rozvoja pre dve nelineárne premenné je

$$f(x, y) \approx f(x^s, y^s) + \frac{\partial f(x, y)}{\partial x} \Big|_{x=x^s} + \frac{\partial f(x, y)}{\partial y} \Big|_{y=y^s}. \quad (2.16)$$

Keďže proces magnetickej levitácie je nestabilný systém, nie je možné ho bez stabilizujúceho regulátora dostať do ustáleného stavu, ktorý by slúžil ako referenčný stav pre linearizáciu. Z toho dôvodu je nutné navrhnúť jednoduchý stabilizujúci regulátor.

Po linearizácii matematického modelu v okolí referenčného stavu ( $U_{pos}^s, U_c^s$ ) a zavedení

odchýlkových veličín získame

$$\frac{d^2 y}{dt^2} = k_1 y + k_2 u, \quad (2.17)$$

kde  $y = U_{pos} - U_{pos}^2$  a  $u = U_c - U_c^2$  sú definované odchýlkové veličiny vzhľadom na referenčný stav. Konštanty  $k_1$  a  $k_2$

$$k_1 = 2K_a \frac{a_{pos}^3 U_c^{2s}}{m_b (U_c^s - b_{pos})^3} \quad (2.18)$$

$$k_2 = -2K_a \frac{a_{pos}^3 U_c^s}{m_b (U_c^s - b_{pos})^3} \quad (2.19)$$

Laplaceovou transformáciou rovnice 2.17 získame výsledný prenos v tvare

$$G(s) = \frac{k_2}{s^2 - k_1} \quad (2.20)$$

Tabuľka 2.1: Zoznam použitých konštánt

Konštanta	Hodnota	Jednotka
$g$	9.81	$ms^{-2}$
$m_b$	$8.4 \cdot 10^{-3}$	kg
$a_{pos}$	$-1.1132 \cdot 10^3$	$Vm^{-1}$
$b_{pos}$	11.107	V
$U_{pos}^s$	1.4	V
$U_c^s$	2.29	V
$K_a$	$1.1899 \cdot 10^{-6}$	$JA^{-1}$

Aplikovaním použitých konštánt z tabuľky 2.1 bol odvodený výsledný prenos v spojitom čase

$$G(s) = \frac{9538}{s^2 - 2250}. \quad (2.21)$$

Systém po diskretizácii pre zvolenú periódu vzorkovania  $T_s = 0.00108$  je

$$G(z^{-1}) = \frac{0.00477z^{-1} + 0.00477z^{-2}}{1 - 2.002z^{-1} + z^{-2}}. \quad (2.22)$$

### 2.3.3 Arduino

Riadenie procesu magnetickej levitácie zabezpečuje mikrokontrolér Arduino YÚN.

Arduino je otvorená platforma založená na mikrokontroléroch ATMega od firmy Atmel a grafickom vývojovom prostredí, ktoré vychádzajú z prostredia Wiring a Processing. Arduino môže byť použité k vytvoreniu samostatných interaktívnych zapojení, alebo môže byť pripojené k softvéru v počítači.

Pre pripojenie externých zariadení, snímačov a akčných členov má Arduino k dispozícii

- 12 analógových vstupov s rozlíšením 10 bitov
- 20 digitálnych vstupov/výstupov, z toho 7 disponujúcich impulzovou šírkovou moduláciou s rozlíšením 8 bitov

Výhodou Arduina je jeho vysoká konektivita (A/D rozhranie, sériové rozhranie až do 115200 Baud Rate, USB serial), plnohodnotná technická dokumentácia, široká užívateľská komunita, nízka spotreba energie (napájanie 5V) a nízka cena.

Arduino vychádza v rôznych verziách ako napríklad UNO, YÚN, MEGA, LEONARDO. Rozdiely medzi verziami sú v množstve pinov pre pripojenie ovládaných zariadení, type mikrokontroléra, alebo v množstve sériových zberníc. V tejto práci bola využitá verzia Arduino YÚN. (obrázok 2.8)



Obr. 2.8: Arduino YÚN

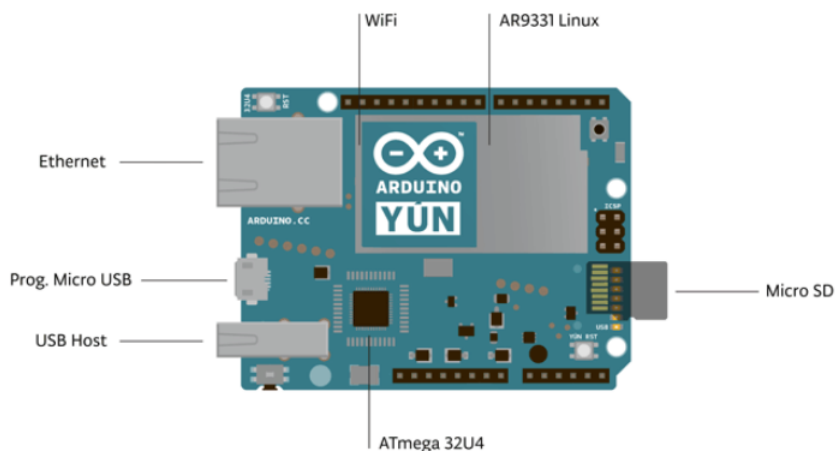
Arduino YÚN disponuje mikrokontrolérom ATmega32u4, ktorý pracuje s frekvenciou až 20MHz. Výpočtový výkon ATmega32u4 je 1MIPS na MHz, teda dokáže vykonať až 20 miliónov inštrukcií za sekundu.[5]

Platforma Arduino YÚN navyše v sebe zahŕňa aj chip Atheros AR9331, na ktorom je prevádzkovaná linuxová distribúcia Linino. Jej základom je systém Linux OpenWRT.

Arduino YÚN teda spája výhody mikrokontroléra a počítača. Komunikáciu medzi čipmi ATmega32u4 a Atheros AR9331 zabezpečuje vstavaná sériová linka. Presný popis komunikácie v zariadení Arduino YÚN je popísaný v kapitole 2.3.3.

Ako je možné vidieť na obrázku 2.9, Arduino YÚN okrem štandardov ako USB obsahuje aj ďalšie periférie, ktoré sú prevádzkované linuxovou distribúciou Linino. Kompletný zoznam periférií zariadenia je:

- Sieťový port RJ-45 s možnosťou napájania technológiou PoE
- Wifi hot-spot,
- USB-A port pre pripojenie periférnych zariadení,
- programovací a napájací port microUSB,
- slot pre pamäťovú kartu micro SD,
- tri resetovacie tlačidlá,
- 16 MHz kryštálový oscilátor.



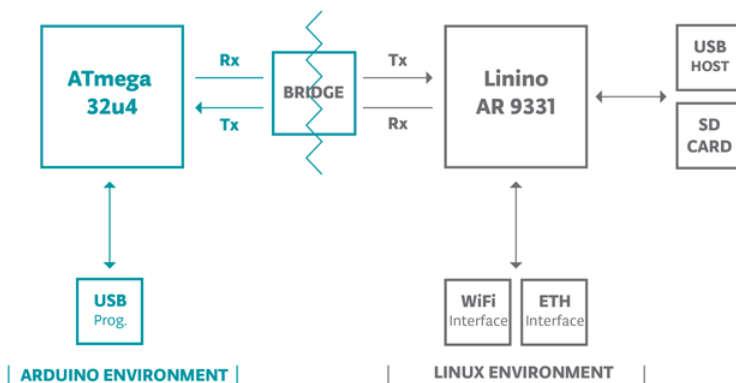
Obr. 2.9: Porty a čipy zariadenia Arduino YÚN



Čip ATmega32u4 disponuje pamäťou 32 kB, ktorá je vyhradená na nahrávanie programov do zariadenia. Čip Atheros AR9331 disponuje iba procesorom, pričom operačná pamäť RAM a interná pamäť sú napojené ako separátne obvody. Arduino YÚN disponuje operačnou pamäťou RAM typu DDR2 o veľkosti 64 MB. Veľkosť ukladacieho priestoru je 16 MB, pričom operačný systém potrebuje 9 MB, takže na vlastné programy je k dispozícii 7 MB pamäte.

### Komunikácia v Arduino YÚN

Schematická komunikácia zariadenia Arduina YÚN je zobrazená na obrázku 2.10.



Obr. 2.10: Schéma zariadenia Arduino YÚN

Sériová komunikácia alebo sériový prenos je proces prenosu dát postupne po jednotlivých bitoch pomocou komunikačného kanála alebo zbernice. Sériová komunikácia sa používa v diaľkových linkách a vo väčšine počítačových sietí. Na krátke vzdialenosti sa sériová komunikácia v počítačových zberniciach v poslednej dobe presadzuje čoraz viac. [12]

Jednotkou sériovej komunikácie je Baud Rate. Baud je v telekomunikáciách a informatike, jednotka "signálnej rýchlosti", čo predstavuje počet zmien stavu prenosového média za sekundu v modulovanom signále. Baud rate vyjadruje množstvo prenesených bitov za sekundu. Vývojová platforma Arduino plnohodnotne podporuje sériovú komunikáciu. Sériová komunikácia medzi užívateľským počítačom a Arduino je zabezpečovaná pomocou rozhrania USB, alebo taktiež pomocou pinov RX (pin 0) a TX (pin 1). Tieto piny sa pri využívaní sériovej komunikácie ináč využívať nedajú.

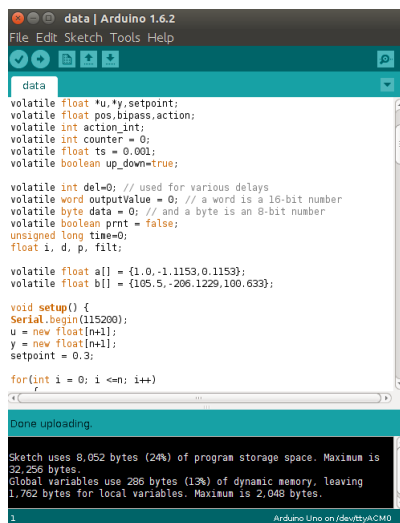
Pre komunikáciu pomocou sériovej linky medzi čipmi ATmega32u4 a Atheros AR9331 je vstavaný tzv. mostík (angl. bridge). Komunikácia pomocou mostíka zabezpečuje nasledovné funkcie.

- Na vyžiadanie zo strany ATmega32u4 vykonáva GNU/Linux príkazy,
- sprostredkováva výmenu údajov medzi mikrokontrolérom a užívateľom
- umožňuje programovanie Arduina pomocou Wifi pripojenia.

Arduino YÚN taktiež disponuje dvoma sériovými zbernicami. Jedna zbernica umožňuje komunikáciu medzi čipom ATmega32u4 a počítačom používateľa. Druhá zbernica, v zdrojovom kóde menovaná ako *Serial1*, je určená na sériovú komunikáciu medzi čipom ATmega32u4 a Atheros AR9331. [6]

## Arduino IDE

Arduino IDE je vývojové prostredie, ktoré je určené na programovanie platformy Arduino. Prichádza taktiež s otvoreným zdrojovým kódom, s podporou všetkým dostupných platforiem (MS Windows, Linux, Mac OS).



Obr. 2.11: Vývojové prostredie Ardunio IDE

Arduino IDE (obr. 2.11) je postavené na programovacom jazyku Java. Svojím prostredím sa snaží čo najviac prispôbiť užívateľovi, obsahuje tlačidlo *Verify*, pre skontrolovanie napísaného kódu, aj tlačidlo *Upload*, pre nahranie programu priamo do mikrokontroléra.

Program pre Arduino, tzv. *sketch*, je nutné písať v jazyku C alebo C++. Nakoľko Arduino IDE využíva framework Wiring, obsahuje priamo funkcie, ktoré dokážu pracovať s pinmi pre zápis a zber údajov, pracovať s dátami v sériovej zbernici, atď.

Samotný program potrebuje pre svoju plnú funkčnosť iba dve základné funkcie typu void, a to *setup()* a *loop()*. V práci sa ešte rozširuje program o funkciu v čase prerušenia.

Funkcia *setup()* sa vykonáva iba raz, a to pri spustení programu. V tejto funkcii je potrebné inicializovať sériovú zbernicu, pravidelné prerušenia pre zabezpečenie presne vykonávaného časovania a všetky dáta, ktoré sú potrebné pre spustenie samotnej realizácie.

Funkcia *loop()* sa dá interpretovať ako nekonečná slučka. Tvorí časť programu, ktorej úlohou je zbieranie dát z procesu, zapisovanie ich do zbernice a rôzne iné funkcie potrebné pre každú slučku.

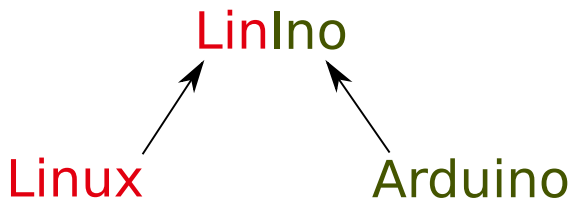
Pre nahranie programu do mikrokontroléra je potrebné zadať používaný port a typ dosky. Postup je nasledovný:

- **Doska:** Tools → Board
- **Port:** Tools → Port.

Pre prácu s Arduino YÚN je potrebné mať nainštalovanú verziu Arduino IDE 1.6.0, ktorá túto dosku už podporuje.

### 2.3.4 Linino

Ako už bolo spomenuté v sekcii 2.3.3, na čipe Atheros AR9331, je implementovaná linuxová distribúcia Linino, založená na operačnom systéme OpenWRT.



Obr. 2.12: Linino

Názov Linino je odvodený zo slov Linux a Arduino, ako je možné vidieť na obrázku 2.12.

OpenWRT je vysoko rozšírená GNU/Linux distribúcia pre vnorené systémy. Na rozdiel od mnohých iných distribúcií, OpenWRT je postavený od základu tak, aby bol plne

funkčný, ľahko upraviteľný operačný systém. V praxi to znamená, že je postavený na linuxovom jadre špeciálne upravený pre priemyselné využitie.[8]

### 2.3.5 Protokol WebSocket

Protokol WebSocket je komunikačný protokol, ktorý zásadne zlepšuje výmenu dát medzi serverom a cieľovým užívateľským rozhraním, v tomto prípade s Webovou technológiou JavaScript. Klasická výmena dát typu požiadavka – odpoveď funguje na jednoduchom princípe, kde server dostane celú požiadavku, spracuje ju a odošle odpoveď. Následne sa uzavrie komunikačný kanál a otvorí sa znova až pri ďalšej požiadavke zo strany užívateľa.

Naproti tomu je protokol WebSocket univerzálnejší, štandardne komunikuje portom 80, takže nie sú problémy s Firewallom. WebSocket je súčasťou HTTP 1.1. Request WebSocketu, nakoľko bol vytvorený ako rozšírenie pre webové prehliadače a servery. Základ komunikácie spočíva v otvorenom spojení aj po odoslaní odpovede na požiadavku užívateľa. Port sa otvára len raz a počas celého spojenia sa len vymieňajú dáta vo forme textových správ (v práci bola na prenos dát použitá štruktúra JSON). Port sa zatvorí sa až pri úplnom ukončení spojenia.

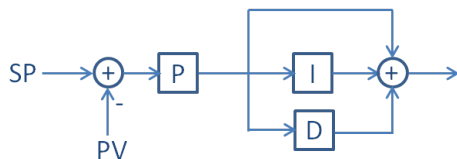
## 2.4 Riadenie systému

### 2.4.1 PID regulátor

Regulátor je reálny, alebo abstraktný (matematicky) objekt, ktorý na základe veľkosti regulačnej odchýlky vypočíta akčný zásah (vstup) do systému tak, aby zabezpečil stabilitu uzavretého regulačného obvodu a dosiahnutie žiadanej hodnoty riadenej veličiny.

Pri riadení magnetickej levitácie bol použitý PID regulátor. PID regulátor je jednoduchý typ lineárneho regulátora. Ako z názvu vyplýva, skladá sa z troch zložiek a to:

- P: Proporcionálna zložka
- I: Integračná zložka
- D: Derivačná zložka



Obr. 2.13: Ideálna forma PID regulátora

Ideálna forma PID regulátora je zobrazená na obrázku 2.13.

Regulátor ako vstup vyžaduje regulačnú odchýlku, čo je rozdiel medzi žiadanou veličinou (angl. Setpoint) a výstupnou veličinou (ang. Process Variable).

Všeobecný zákon riadenia je daný vzťahom

$$u(t) = K_p \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (2.23)$$

po aplikovaní Laplaceovej transformácie

$$U(s) = K_p \left( E(s) + \frac{E(s)}{T_i s} + T_d(s) E(s) s \right) \quad (2.24)$$

$$G_r(s) = \frac{U(s)}{E(s)} = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right) \quad (2.25)$$

$$G_r(s) = K_p + \frac{K_p}{T_i s} + K_p T_d s \quad (2.26)$$

zavedením substitúcie

$$\begin{aligned} P &= K_p \\ I &= \frac{K_p}{T_i} \\ D &= K_p T_d \end{aligned}$$

sa odvádza výsledný tvar pre PID regulátor

$$G_r(s) = P + \frac{I}{s} + Ds, \quad (2.27)$$

kde po úprave

$$G_r(s) = \frac{D s^2 + P s + I}{s} \quad (2.28)$$

je odvodený výsledný prenos regulátora.

Z prenosu je vidieť, že nie je splnená podmienka fyzikálnej realizovateľnosti prenosu, nakoľko rád čitateľa je vyšší ako rád menovateľa, a teda takýto regulátor nie je možné aplikovať. Preto sa pridáva to regulátora filter D zložky  $N$ .

Aplikovaním  $N$  zložky do rovnice 2.27 získame vzťah

$$G_r(s) = P + \frac{I}{s} + D \frac{N}{1 + N \frac{1}{s}}, \quad (2.29)$$

kde po úprave

$$G_r(s) = \frac{(P + DN)s^2 + (I + PN)s + NI}{s^2 + Ns}, \quad (2.30)$$

definujeme výsledný vzťah PID regulátora v spojitej oblasti, ktorý už spĺňa podmienku fyzikálnej realizovateľnosti.

Takýto regulátor je možné aplikovať pri simuláciách, prípadne v rôznych spojitých pneumatických regulátoroch.

Vo svete moderného riadenia sú však všetky regulátory aplikované v diskkrétnej oblasti. V diskkrétnej oblasti je pomocou  $\mathcal{Z}$  - transformácie definovaný ako

$$\mathcal{Z} \left( \frac{1}{s} \right) = \frac{1}{z - 1}. \quad (2.31)$$

Substitúciou integrátora v rovnici 2.29 získavame vzťah

$$G_r(z) = P + I T_s \frac{1}{z-1} + D \frac{N}{1 + N T_s \frac{1}{z-1}}, \quad (2.32)$$

kde po úpravách získame výsledný diskrétny prenos

$$G_r(z) = \frac{b_2 z^2 + b_1 z + b_0}{z^2 + a_1 z + a_0}, \quad (2.33)$$

kde aplikujeme konštanty v nasledujúcom tvare

$$\begin{aligned} b_2 &= P + D N, \\ b_1 &= P N T_s - 2 P + I T_s - 2 D N, \\ b_0 &= P - P N T_s - I T_s + I N T_s^2 + D N, \\ a_1 &= N T_s - 2, \\ a_0 &= 1 - N T_s. \end{aligned}$$

Prenos  $G_r(z)$ , ktorý definovaný ako  $\frac{E(z)}{U(z)}$ , je možné reprezentovať diferenčnou rovnicou

$$u(k) = -a_1 u(k-1) - a_2 u(k-2) + b_2 e(k) + b_1 e(k-1) + b_0 e(k-2,) \quad (2.34)$$

ktorá po dosadení číselných hodnôt bude implementovaná v mikrokontroléri Arduino pre riadenie procesu magnetickej levitácie.

### 2.4.2 Riadenie v reálnom čase

Pre riadenie magnetickej levitácie je potrebné zabezpečiť veľmi rýchle výpočty akčných zásahov podľa pozície guľôčky. Perióda vzorkovania musí byť pod dve až jednu milisekundu. Počas tohto času je potrebné prečítať aktuálnu pozíciu guľôčky v magnetickom poli, vypočítať potrebný akčný zásah a realizovať ho. Nakoľko základné nastavenia Arduina sa nedostanú pod tento čas, je potrebné nakonfigurovať presne časované výpočty v procesore.

Arduino má hlavný procesor, ktorý vykonáva všetky funkcie. Tento procesor je jednojadrový a jednovláknový. Znamená to, že sa vykonávajú všetky jeho procesy, ako čítanie, zapisovanie, výpočty, komunikácia v takom poradí, v akom sú zapísané v programe. To je však v tomto prípade nevyhovujúce, nakoľko pre riadenie, ktoré má byť vykonávané v reálnom čase, je potrebné v presne stanovených intervaloch vypočítavať akčný zásah. Nato slúžia hodiny a časovače.

Hodiny sú samostatný oscilačný obvod. Frekvencia hodín pre Arduino je nastavená na 16 MHz. Arduino poskytuje štyri časovače označené *timer0*, *timer1*, *timer3* a *timer4*. Tieto časovače sú počítadla, ktoré v základnom nastavení zvýšia svoju hodnotu každým tikom hodín o jednotku [3].

Hodnota v časovači sa porovnáva s tzv. CTC časovačom prerušenia. Túto hodnotu si volíme my. Ak je splnená porovnávacia podmienka, časovač inicializuje tzv. prerušenie. Vtedy sa prerušia všetky programy bežiace v hlavnom procesore a vykoná sa program uložený v pamäti, ktorý sa spúšťa len počas prerušenia. Po vykonaní programu sa časovač premaže a počíta sa odznova.

Je možné vidieť, že pri rýchlosti 16 miliónov tikov za sekundu je počítadlo o pamäti reprezentujúce  $2^{16}$  bitov naplnené okamžite a pri riadení hrozí pretečenie.

Preto sa medzi hodiny a časovač definuje tzv. delič. Ten je možné nastaviť presne pomocou tabuliek. Hodnoty deličov môžu byť 8, 64, 256, alebo 1024.

Deličom priamo nastavujeme, ako rýchlo sa bude naplňovať časovač. V práci je využitý delič, ktorý inicializuje zvýšenie hodnoty v časovači až po každom 1 024 tiku hodín. Pre výpočet použitia správneho časovača:

Výpočet rýchlosti zápisov do časovača

$$Frekvencia\ zapisov = \frac{Frekvencia\ hodin}{Delič} = \frac{16 \times 10^3}{1024} = 15,625\ kHz \quad (2.35)$$



Pre žiadanú periódu vzorkovania  $T_s = 0.00108 \text{ s}$

$$\text{Hodnota registra} = \frac{\text{Rychlost hodin}}{\text{Delic} \times \text{Pozadovane prerusenie}} - 1 \quad (2.36)$$

$$\text{Hodnota registra} = \frac{16 \times 10^6}{1024 \times 10^3} = 14,625 \approx 15 \text{ bit} \quad (2.37)$$

Nakoľko hodnota 15 je nižšia ako pre 8-bitový timer0, je možné použiť aj tieto časovače. V práci bol implementovaný timer1, nakoľko viac bitov znamená väčšiu presnosť pri riadení.



# Kapitola 3

## Implementácia riadenia

Kapitola sa venuje implementácii riadenia v reálnom čase pomocou mikrokontroléra Arduino YÚN. Zapisuje sa riadiaci algoritmus a aplikujú sa presne definované prerušenia pre zabezpečenie riadenia v reálnom čase.

### 3.1 Syntéza regulátora

Vhodný regulátor sa hľadal metódou umiestnenia pólov, z ktorého sa odvodili parametre PID regulátora.

Majme systém

$$G_s(s) = \frac{9538}{s^2 - 2250} \quad (3.1)$$

a regulátor v tvare

$$G_r(s) = \frac{(P + DN)s^2 + (PN + I)s + IN}{s^2 + Ns}. \quad (3.2)$$

Dosadením do charakteristickej rovnice uzavretého regulačného obvodu (URO)

$$1 + G_r(s)G_s(s) = 0 \quad (3.3)$$

získame tvar

$$1 + \frac{9538}{s^2 - 2250} \frac{(P + DN)s^2 + (PN + I)s + IN}{s^2 + Ns} = 0. \quad (3.4)$$

Z rovnice vyplýva že URO je 4. rádu, teda aj referenčný polynóm musí byť 4. rádu. Pre referenčný polynóm bol zvolený štvornásobný pól  $s_{1,2,3,4} = -60$ .

Aplikovaním umiestnenia pólov dostávame rovnicu v tvare

$$1 + \frac{9538}{s^2 - 2250} \frac{(P + DN)s^2 + (PN + I)s + IN}{s^2 + Ns} = (s - 60)^4. \quad (3.5)$$

Následnými úpravami získavame hodnoty PID regulátora.

$$P = 4.8$$

$$I = 3.27$$

$$D = 0.066$$

$$N = 200$$

Tieto hodnoty boli vzaté ako štartovací bod pre získanie výsledného PID regulátora.

## 3.2 Ladenie regulátora

Manuálnym ladením regulátora sa odvodil jeho výsledný tvar.

$$P = 8$$

$$I = 5$$

$$D = 0.035$$

$$N = 2000$$

Aplikovaním konštánt do rovnice 2.29, získame výslednú prenosovú funkciu regulátora v tvare

$$G_r(s) = \frac{78s^2 + 16005s + 10000}{s^2 + 2000s}. \quad (3.6)$$

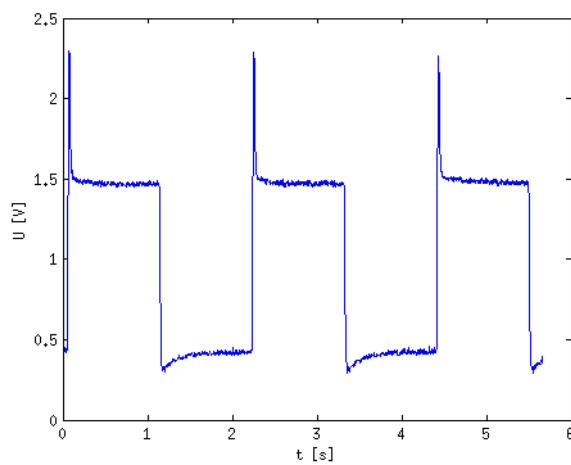
Prenos v diskretnom tvare pri perióde vzorkovania  $T_s = 0.00108$

$$G_r(z^{-1}) = \frac{78 - 148.9z^{-1} + 70.92z^{-2}}{1 - 1.115z^{-1} + 0.1153z^{-1}}, \quad (3.7)$$

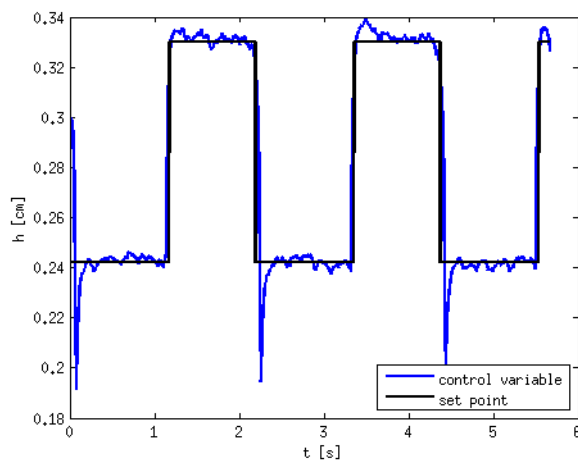
a korešpondujúca diferenčná rovnica v tvare

$$u_k = 1.115u_{k-1} - 0.1153u_{k-2} + 78e_k - 148.9e_{k-1} + 70.92e_{k-2}. \quad (3.8)$$

Osciloskopom boli namerané hodnoty riadiacej a riadenej veličiny. Tieto hodnoty sú vykreslené na grafoch 3.1 a 3.2.



Obr. 3.1: Pribeh riadiacej veličiny



Obr. 3.2: Pribeh riadenej veličiny

### 3.3 Riadiaci algoritmus

Ako už bolo spomenuté v kapitole 2.3.3, Arduino plne zabezpečuje riadenie magnetickej levitácie. Toto riadenie beží v pravidelných intervaloch bez nutnosti zásahov užívateľa. Užívateľské prostredie je vytvorené k zberu a spracovaniu údajov, a nastavovaniu nových parametrov.

V sekcii sa rozoberie implementovaný riadiaci algoritmus, ktorý samotné meranie zabezpečuje. Samotný riadiaci algoritmus

```
pos = 0.54*analogRead(A0)/950.0;
u[n] = setpoint-pos;
y[n] = 0.0;
for(int i = 1; i <= n; i++)
{
    y[n] += -a[i]/a[0]*y[n-i];
}
for(int j = 0; j <= n; j++)
{
    y[n] += b[j]/a[0]*u[n-j];
}
bypass = 1.0-0.94*pos;
action = y[n]*bypass;
if(action < 0.0) action = 0.0;
if(action > 1.0) action = 1.0;

action_int = floor(4095.0*(action));
dac(action_int);

for(int j=1;j<=n;j++)
{
    y[j-1]=y[j];
    u[j-1]=u[j];
}
```

zabezpečuje riadenie diskretným PID regulátorom v každej perióde vzorkovania.

# Kapitola 4

## Vzdialené riadenie a sledovanie

### 4.1 Meranie rýchlosti sériovej linky

Pri implementovaní nútených prerušení pomocou oscilátora je potrebné zistiť, či voľný čas bude stačiť na komunikáciu pomocou sériovej zbernice.

Samotné meranie pozostáva z merania rýchlosti zápisu a čítania bytov z/do linky. Všetky údaje v sériovej zbernici sú reprezentované ako refazec, preto sa vykonáva čítanie/zápis refazcov.

#### 4.1.1 Meranie čítania dát zo zbernice

Pri meraní rýchlosti čítania dát zo zbernice boli vykonané tri merania a ich následný priebeh je zobrazený na grafe. Postupne sa čítalo 1,2,7 znakov, pričom výsledný priebeh, obr. 4.1 je možné považovať za lineárnu závislosť.

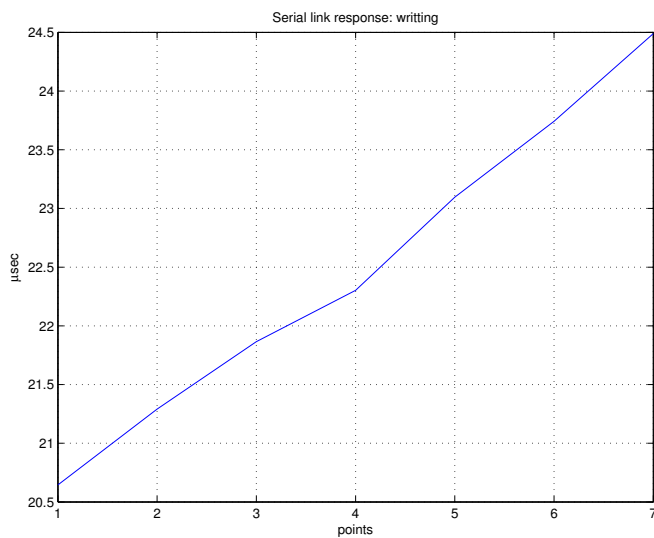
Odpovedajúca rovnica priamky

$$y = 6,09x + 15,24. \quad (4.1)$$

Z rovnice 4.1 možno vidieť, že čítanie jedného znaku trvá 6,09  $\mu sec$ , samotná inicializácia 15.24  $\mu sec$ .

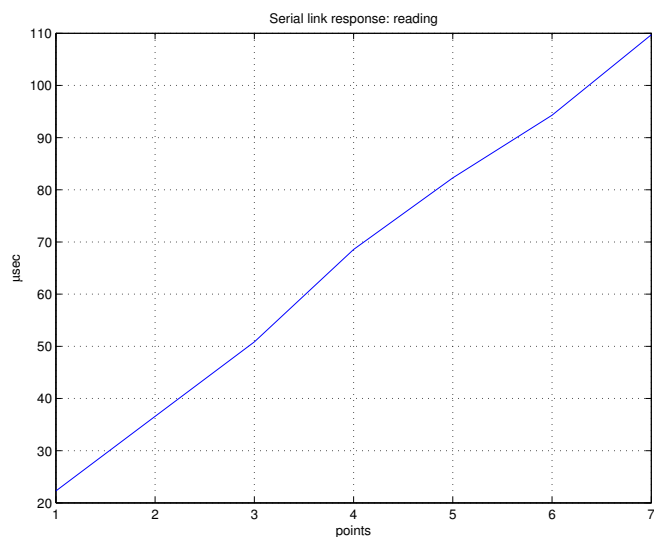
#### 4.1.2 Meranie zápisu dát do sériovej zbernice

Pri meraní rýchlosti zapisovania dát zo zbernice boli vykonané tri merania a ich následný priebeh je zobrazený na grafe. Postupne sa zapisovalo 1,2,7 znakov, pričom výsledný



Obr. 4.1: Meranie rýchlosti čítania zo sérieovej zbernice

priebeh, obr. 4.2 je možné považovať za lineárnu závislosť.



Obr. 4.2: Meranie rýchlosti zapisovania dát do sérieovej zbernice

Odpovedajúca rovnica priamky je:



$$y = 4.60x + 27.08. \quad (4.2)$$

Z výsledku možno vidieť, že zápis jedného znaku do zbernice trvá  $4,60 \mu\text{sec}$ , samotná inicializácia  $27,08 \mu\text{sec}$ .

V práci sa predpokladá neustály zápis dát do zbernice, pričom maximálna hodnota časovej stopy je jedna hodina, teda 3 600 000 milisekúnd, čo tvorí 7 znakov, oddeľovací znak a pozícia guľôčky s desatinnou čiarkou tvorí 4 znaky. Spolu je to teda 11 znakov. Samotný zápis tvorí teda  $50,6 \mu\text{sec}$ , čo je pri riadení za 1 milisekundu vyhovujúce.

## 4.2 WebSocket server

Ako bolo popísané v sekcii 2.3.3, Arduino YÚN neslúži iba ako mikrokontrolér, ale obsahuje aj čip Atheros AR9331 s operačným systémom Linino.

Nakoľko je implementovaný server napísaný v jazyku Python, bolo potrebné nainštalovať podporu programovania v jazyku Python a taktiež knižnice pre sériovú komunikáciu a Webový server. Príkazy sú nasledujúce:

```
apt-get install python2.7 python-serial python-openssl pip
pip install pyserial
pip install tornado
```

Požiadavky na tvorbu WebSocket servera boli nasledovné

- **Rýchlosť:** program musí byť natoľko rýchly, aby nestrácal žiadne merané vzorky dát
- **Stabilita:** program musí mať ošetrené všetky prípadné chyby
- **Bez interakcie:** WebSocket server musí bežať ako proces na pozadí operačného systému bez nutnosti zásahu užívateľa

Kód WebSocket servera

```
import sys, os, serial, time
import tornado.ioloop
import tornado.web
import tornado.websocket
from tornado.options import define, options, parse_command_line
```

```
i=0
print "Start program"
baud=115200
global f

from tornado.options import define, options, parse_command_line
define("port", default=8282, help="", type=int)

def ParsingToJson(array_string):
    d = {}
    json_string = ''
    time_array = []
    pos_array = []
    json_string = '{"time":[ '
    for row in array_string:
        d = row.split(',')
        time_array.append(d[0])
        pos_array.append(d[1])

    for i, elem in enumerate(time_array):
        lenght = len(time_array)
        json_string += elem
        if not (i == lenght - 1):
            json_string += ','
        else:
            json_string += '], '

    json_string += '"position":[ '
    for i, elem in enumerate(pos_array):
        json_string += elem
        lenght = len(pos_array)
        if not (i == lenght - 1):
            json_string += ','
        else:
            json_string += ']]'
```

```
        return json_string

def connectDevice():
    global ser
    devices = []
    devices.append('/dev/ttyATH1')
    devices.append('/dev/ttyATH0')
    for dev in devices:
        try:
            print "Trying connect to",dev
            ser = serial.Serial(dev, baud)
            print "Connection on ",dev," success"
            break
        except:
            print 'Connection on ',dev," failed"

def deleteFile():
    dataFile = 'data.csv'
    if os.path.getsize(dataFile) > 600000:
        os.remove(dataFile)
        f=open("data.csv","w+b")

class WebSocketHandler(tornado.websocket.WebSocketHandler):

    halt = False

    def check_origin(self, origin):
        return True

    def open(self,*args):
        connectDevice()
        print 'Connected'
        f=open("data.csv","w+b")
        self.halt = False
```

```
def on_message(self, message):

    data = str(message)
    print data
    if data == "Start ":
        self.halt = False
        secs = int(time.strftime("%S", time.gmtime()))

    if data == "Get_data ":
        if self.halt:
            return
        buffer = []
        counter=0
        while counter < 100:
            line=ser.readline()
            with open("data.csv", "a+b") as f:
                f.write(line)
            line=line.strip()
            if "," in line:
                buffer.append(line)
            counter += 1
        buffer.pop(0)
        self.write_message(ParsingToJson(buffer))
        deleteFile()

    if data == "Halt ":
        print 'Message for halt '
        f = None
        self.halt = True
        self.write_message('Halted ')

    if "." in data:
        print data
        ser.write(data)
```

```
def on_close(self):
    print "Disconnected"

print 'WebSocketServer initialized'

app = tornado.web.Application([(r'/w/', WebSocketHandler),])

if __name__ == '__main__':
    parse_command_line()
    app.listen(options.port)
    tornado.ioloop.IOLoop.instance().start()
```

Tornado WebSocket server funguje na báze správ medzi užívateľom a serverom. Samotné spustenie programu automaticky inicializuje WebSocket server, ktorý čaká na požiadavku od klienta.

Program po otvorení spojenia nadviaže komunikáciu so sériovou zbernicou Arduina. Je nastavené dávkové posielanie dát, nakoľko posielanie správ každú milisekundu má za nezvratný následok "nestíhanie"čítania dát zo zbernice a tým stratu dát. Dáta sú automaticky generované vo formáte CSV, čo znamená Comma Separated Value, voľným prekladom povedané Hodnoty oddelené čiarkou. Tieto dáta sú svojou univerzálnosťou a jednoduchosťou vhodné pre využitie v rôznych výpočtových programoch, akými sú GNU Octave, príp. MATLAB. Preto sú tieto dáta automaticky ukladané do súboru, aby si ich mohol užívateľ pohodlne stiahnuť a pracovať s nimi.

Pri správe *Get\_Data* sa prečíta 100 vzoriek dát zo zbernice, tieto údaje sa zapíšu do súboru a vykoná sa funkcia *createJson*, ktorá nazbierané dáta prepíše do formátu JSON. Ten sa následne posieľa používateľovi, kde ho jeho klientská časť spracuje a využije na vykreslenie grafu.

Pri komunikácii klient - mikrokontrolér slúži server ako komunikačný mostík. Jeho jedinou úlohou je preposielanie poslaných správ do mikrokontroléra, ktorý ich už spracuje a využije na nastavenie žiadanej pozície guľôčky, alebo zmenu parametrov regulátora. Nutné ošetrovanie je ošetrovanie veľkosti súboru dát. Arduino YÚN disponuje iba 7.5 MB priestoru, takže vytváranie veľkých súborov nie je vhodné, nakoľko hrozí zaplnenie pamäte a znefunkčnenie programu. Preto bola aplikovaná funkcia *deleteFile()*, ktorá má za úlohu sledovať veľkosť súboru a pri dosiahnutí hranice 600 kB súbor zmaže. 600 kB stačí na 7 minútový zápis, čo pri tak rýchlom systéme ako je proces magnetickej levitácie je úplne postačujúce.

## 4.3 Klientská časť

V práci bola klientská časť vytvorená pomocou webového rozhrania a teda využitia technológií HTML, CSS, JavaScript.

Pri tvorbe webového rozhrania sa brali do úvahy nasledovné

- **Intuitívnosť:** celé rozhranie musí byť jasné a prehľadné
- **Rýchlosť:** program na strane klienta musí stíhať čítať posielané vzorky a vykreslovať graf
- **Vizualizácia:** prostredie musí byť z grafickej stránky pre užívateľa príťažlivé
- **Funkcionalita:** užívateľ musí mať kontrolu nad začiatkom merania, jeho ukončením, stiahnutím dát, nastavovaním nových parametrov PID regulátora a žiadanej veličiny.

### 4.3.1 Pripojenie

Aby klient mohol pristupovať k serveru, potrebuje poznať jeho IP. V práci sa neuvažovalo o vyčlenení statickej vonkajšej IP adresy pre prístup zvonka, no uvažovalo sa o pripojení k Arduinu buď pomocou lokálneho pripojenia pomocou WiFi, ktorú Arduino vytvára, káblového pripojenia, alebo využitím vnútornej siete školy.

Linino je už od výroby nastavené na vytváranie webového servera, preto konfigurácia je obsiahnutá v nasledujúcich bodoch:

- pripojenie sa k Arduinu pomocou kábla alebo WiFi,
- nakopírovanie potrebných súborov do Arduina prostredníctvom vzdialeného pripojenia,

```
$ scp main.py root@arduino.local:/usr
$ scp -r web_client/ root@arduino.local:/www
```

- úprava konfiguračného súboru rc.local

```
$ ssh root@arduino.local
$ echo "python /usr/main.py" >> /etc/rc.local
$ exit
```

- reštartovanie zariadenia.

Po reštarte je zabezpečené, že sa na pozadí operačného systému automaticky spúšťa WebSocket server. Užívateľ sa do užívateľského prostredia dostane otvorením svojho webového prehliadača a to nasledovne:

- lokálnym pripojením

`http://arduino.local/web_client/`

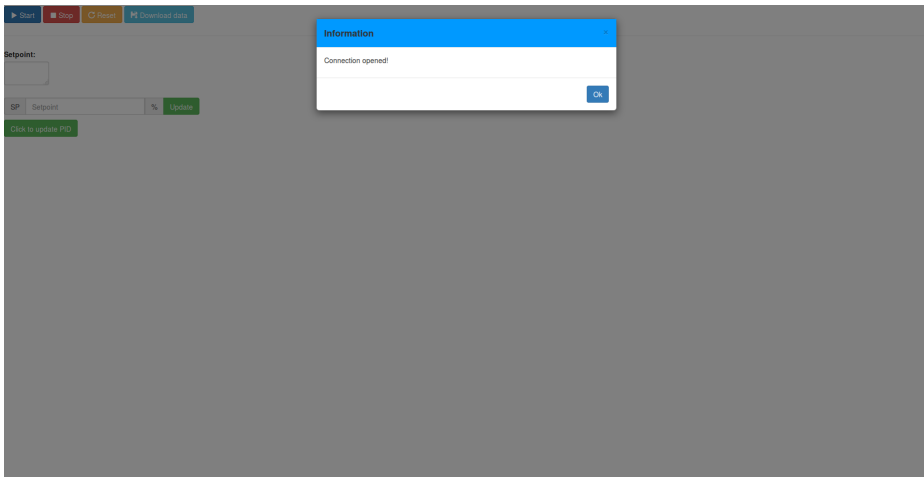
- pripojením v rámci školskej siete

1. zapojenie internetového kábla do Arduina,
2. zistenie pridelenej IP adresy,
3. zadanie adresy do prehliadača v tvare:

`http://$(IP)/web_client/`

### 4.3.2 Úvodná obrazovka

Po pripojení sa užívateľ dostáva do prostredia pre ovládanie a sledovanie systému magnetickej levitácie.

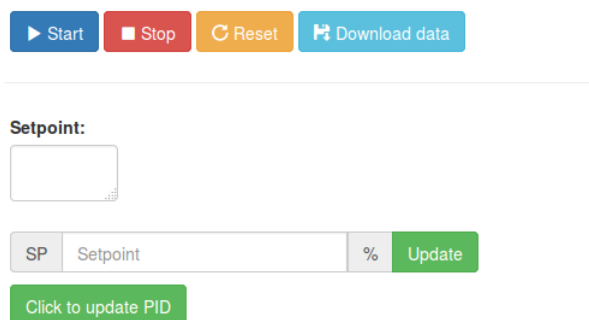


Obr. 4.3: Úvodná obrazovka

Na obrázku 4.3 je možné vidieť modálny dialóg, ktorý informuje užívateľa o úspešnom pripojení. Ak sa tento dialóg nezobrazí, tak to indukuje poškodenie WebSocket servera, napríklad jeho nežiadané zastavenie.

### 4.3.3 Ovládacie tlačidlá

Základné prostredie, obrázok 4.4, zobrazuje jednoduché užívateľské prostredie.



Obr. 4.4: Ovládacie tlačidlá

Užívateľské prostredie bolo navrhnuté pre ovládanie pomocou tlačidiel. Funkcionalita ovládacích tlačidiel bola písaná v jazyku JavaScript.

Klientska časť obsahuje nasledujúce tlačidlá:

- **Start:** spustenie vykresľovania grafu,
- **Stop:** zastavenie vykresľovania grafu,
- **Reset:** resetovanie časovej osi,
- **Download data:** stiahnutie nameraných dát,
- **Click to update PID:** zmena parametrov PID regulátora,
- **Update:** zmena hodnotu žiadanej veličiny v %.

Nasledujúce podkapitoly sa bližšie venujú jednotlivým tlačidlovým funkcionalitám.



### 4.3.4 Sledovanie pozície guľôčky

Po nadviazaní spojenia je nutné spustiť samotné vykresľovanie grafu, obrázok 4.5. Po stlačení tlačidla *Štart* sa odošle serveru správa o spustení prenosu údajov a taktiež sa nastaví interval na pravidelné vyžiadanie nových dát, ktoré sa vykreslia do grafu.



Obr. 4.5: Vykresľovanie dát do grafu

Dáta sa prekresľujú kontinuálne. Jemný šum je spôsobený poruchami pôsobiacimi na merací člen a rozlíšením A/D prevodníka pre čítanie signálu z proximitného senzora. Maximálna hodnota žiadanej veličiny môže byť nastavená na hodnotu 0.50 cm, nakoľko šum spôsobí narážanie guľôčky do senzora. Aktuálna hodnota sledovanej veličiny sa okrem grafu zobrazuje aj v textovom okne **Setpoint**. Os X reprezentuje čas v sekundách, os Y pozíciu v centimetroch.

Vykresľovanie grafu je možné kedykoľvek zastaviť tlačidlom *Stop*.

### 4.3.5 Zmena hodnoty žiadanej veličiny

Žiadanou veličinou pri riadení magnetickej levitácie je pozícia guľôčky v magnetickom poli. Odmeraná vzdialenosť medzi cievkou a pozičným snímačom činí 0,54 cm. Nakoľko riadenie pomocou PID regulátora vykazuje jemný šum, boli stanovené hodnoty minimálnej a maximálnej hodnoty pozície

$$w_{min}=0.01$$

$$w_{max}=0.5$$

V práci bolo možné implementovať

- absolútne zadávanie pozície,
- percentuálne zadávanie pozície.

Absolútne zadávanie však narážalo na dve nižšie uvedené bezpečnostné obmedzenia.

1. Používateľ musí poznať horný a dolný limit.
2. Desatinnú čiarku musí reprezentovať bodka.

Z týchto dôvodov bolo zvolené percentuálne zadávanie pozície guľôčky.

Výpočet pozície guľôčky prebieha podľa vzorca

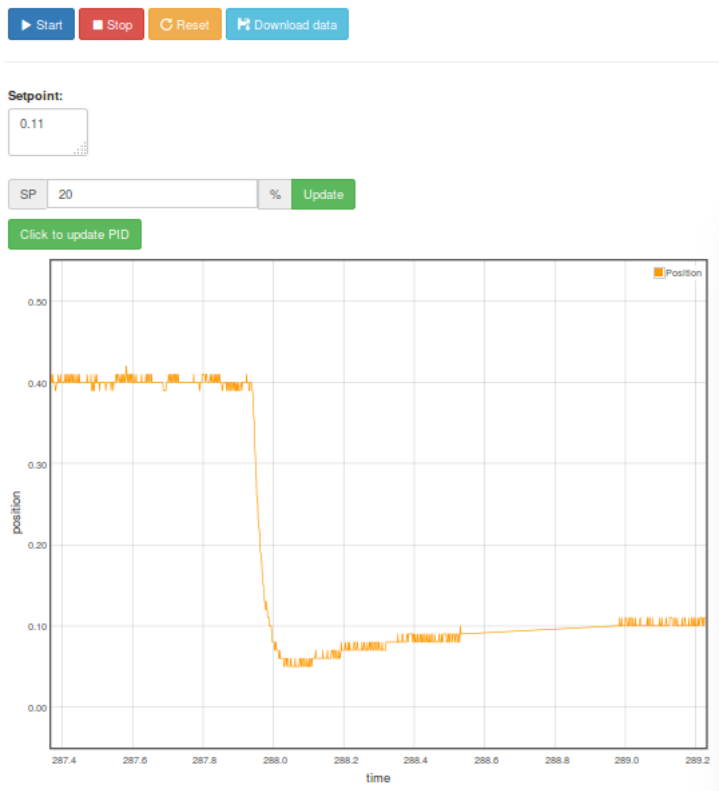
$$min=0,01 \tag{4.3}$$

$$max=0,5 \tag{4.4}$$

$$poz=\frac{max-min}{100}W+min \tag{4.5}$$

kde  $W$  je hodnota v percentách zadaná užívateľom, ktorá sa následne posiela cez WebSocket server mikrokontroléru.

Reakciu na zmenu žiadanej veličiny je možné sledovať na obrázku 4.6.



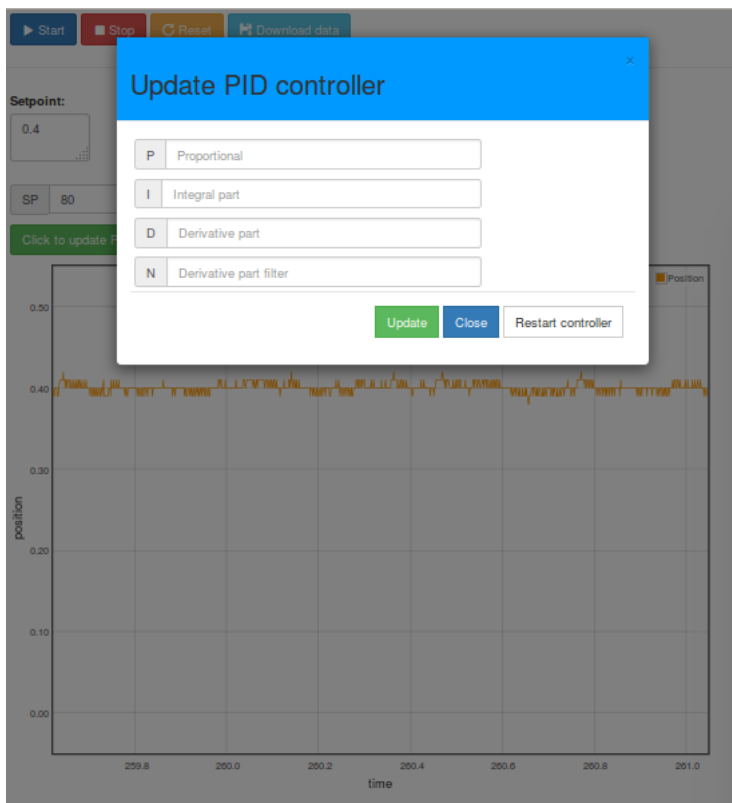
Obr. 4.6: Zmena žiadanej veličiny

#### 4.3.6 Zmena hodnôt PID regulátora

Magnetická levitácia je proces, ktorý nemá stabilný ustálený stav. Z toho vyplýva, že používateľ pri manuálnom doladení regulátora potrebuje už nainplementovaný stabilizujúci regulátor. Ten sa inicializuje už pri zapojení Arduina do obvodu bez akéhokoľvek vonkajšieho vplyvu.

Ak používateľ pri sledovaní odozvy systému usúdi, že regulátor je potrebné doladiť, stačí kliknúť na tlačidlo *Click to update PID*. Následne sa užívateľovi zobrazí modálny dialóg, obrázok 4.7, kde môže zadať požadované hodnoty.

V práci sa zadávajú iba všetky hodnoty regulátora naraz. Ak užívateľ jednu hodnotu nezadá, tak sa zobrazí výstražné okno o nutnosti zadania všetkých parametrov. Po zadaní požadovaných hodnôt sa pošlú tieto údaje do Arduina, ktoré ich roztriedi a nastaví. Ak užívateľ zadal zlú hodnotu a chce obnoviť nastavenie pôvodného regulátora, tak v



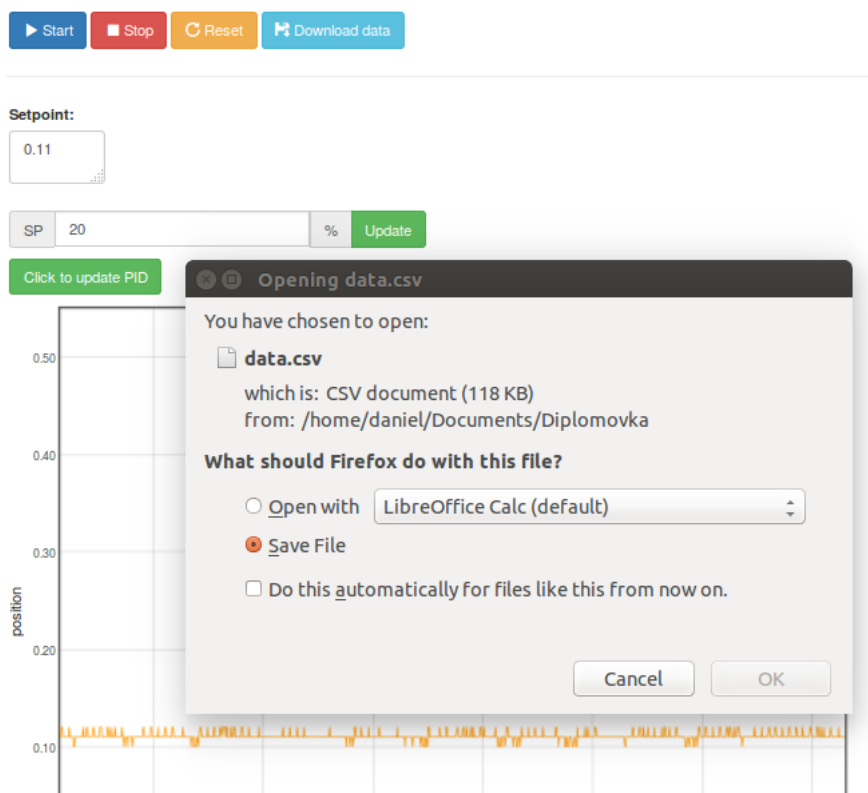
Obr. 4.7: Zmena parametrov PID regulátora

modálnom dialógu klikne na tlačidlo *Restart controller*.

### 4.3.7 Stiahnutie nameraných údajov

Každé meranie má zmysel vtedy, ak si užívateľ dokáže stiahnuť namerané údaje a tie potom spracovať. Ukladanie dát, ako bolo popísané v sekcii 4.2, má na starosti server. Po stlačení tlačidla *Download data* sa zobrazí okno, obrázok 4.8, ktoré umožňuje užívateľovi stiahnuť dáta do svojho počítača.

Stahovanie dát je citlivé na použitie internetového prehliadača. Pri používaní prehliadača *Mozilla Firefox*, sa po vyžiadaní stiahnutia dát automaticky uzavrie WebSocket spojenie. Jedná sa o chybu prehliadača, ktorá v čase písania práce ešte nie je odstránená. Pri využívaní prehliadača *Google Chrome*, prípadne *chromium*, sa spojenie neuzatvára.



Obr. 4.8: Stiahnutie nameraných údajov



## Kapitola 5

# Diskusia a záver

Práca bola zameraná na vytvorenie vzdialeného laboratória pre procesy s rýchlou dynamikou. Ako proces bol zvolený systém magnetickej levitácia, ako serverové a ovládacie zariadenie Arduino YÚN a ako klientska časť webové rozhranie vytvorené na technológiách HTML, CSS a JavaScript.

Riadenie magnetickej levitácie pomocou mikrokontroléra Arduino bolo realizované úspešne, pričom systém dokázal vo zvolenej perióde vzorkovania zbierať aktuálnu informáciu o výstupnej veličine, a teda pozície guľôčky v magnetickom poli, a vypočítať adekvátny akčný zásah. Zvolené riadenie pomocou PID regulátora s filtrom D zložky dokázalo veľmi presne regulovať systém na žiadanú veličinu. Využitie mikrokontroléra pre potreby riadenia sa z pohľadu využitia vo vzdelávaní ukázalo ako vhodná a efektívna alternatíva k priemyselne používaným PLC systémom, nakoľko im dokázal konkurovať v rýchlosti a obstarávacej cene, ktorá bola do 100 eur.

Výmena údajov medzi zariadením a serverom prebiehala pomocou sériovej linky. Arduino po každom prerušení posielal údaje serveru, ktorý ich následovne spracovával. Čas spracovávania údajov bol vyhovujúci pri používaní výkonného zariadenia. Ak bol implementovaný server na výkonnom počítači, všetky dáta tiekli bez obmedzenia. Pri implementovaní servera na čip Atheros AR9331, ktorý je súčasťou mikrokontroléra, bola absencia pamäte RAM citelná. Dátový tok prebiehal vo veľmi obmedzenej miere. Problém nastával pri triedení dát, ktoré zabrali strojový čas natoľko, že nasledovné vzorky dát boli často poškodené. Taktiež rýchlosť čítania čipu nebola postačujúca. Preto aplikovanie celého laboratória do jedného mikročipu bolo aplikovateľné, no v obmedzenej miere. Tento problém je možné v budúcnosti vyriešiť znížením počtu posielaných vzoriek smerom k uží-

vateľovi, teda zriedením dát, pri súčasnom zachovaní hlavných fyzikálnych charakteristík signálu, ktoré reprezentujú.

Klientska časť pracovala presne podľa žiadanej funkcionality, sledovanie pozície guľôčky a zmena parametrov regulátora, iba pri aplikovaní servera na externé zariadenie. Pri aplikovaní servera na mikrokontrolér sa vždy poslala aspoň jedna vzorka, ktorá mala chybnú hodnotu a tým celé prekresľovanie grafu zlyhalo. Problémom bolo aj využitie internetového prehliadača. Nakoľko prehliadač Mozilla Firefox nepodporuje sťahovanie súborov pri použití služby WebSocket, je nutné využívať alternatívne prehliadače chromium alebo Google Chrome. Tento problém bude je možné vyriešiť externou službou, ktorá by namerané dáta ukladala a poskytovala užívateľovi, alebo zjednotením natívnej implementácie technológie WebSocket, ku ktorej s veľkou pravdepodobnosťou v budúcnosti príde.



# Literatúra

- [1] T. R. A KOLEKTÍV, *Physics for Scientists and Engineers*, no. ISBN:1-4292-0410-9, 2007.
- [2] W.-L. DEUSTO, *Weblab-deusto 5.0 documentation, technologies*. online, 2014. <https://weblabdeusto.readthedocs.org/en/latest/technical.html>.
- [3] ENGBLAZE, *Microcontroller tutorial series: Avr and arduino timer interrupts*. online, 2015. <http://www.engblaze.com/microcontroller-tutorial-avr-and-arduino-timer-interrupts/>.
- [4] S. F. L. F. KRBEČEK, M., *Easy Remote ISES – Development Environment Remote Experiments*, no. ISBN 978-0-9818868-4-8, USA : Potomac, 2013.
- [5] A. LLC, *Arduino yún*. online, 2015. <http://www.arduino.cc/en/Main/ArduinoBoardYun?from=Products.ArduinoYUN>.
- [6] ———, *Bridge library for arduino yÚn*. online, 2015. <http://www.arduino.cc/en/Reference/YunBridgeLibrary>.
- [7] R. S. MOSHER, *Scientific American*, 1964.
- [8] OPENWRT, *Openwrt wiki*. online, 2015. <http://wiki.openwrt.org/>.
- [9] R. T. RAYMOND GOERTZ, *Nuocleonics*, 1954.
- [10] M.-F. F. UNIVERZITA KARLOVA, *Vzdálená školní laboratoř pro měření radioaktivity*. online, 2014. <http://kdt-38.karlov.mff.cuni.cz/>.
- [11] K. G. N. K. V. BLAŤÁK, V. ŠMÍD, *Levitace je legrace*. online, 2011. <http://fyzsem.fjfi.cvut.cz/2010-2011/Zima10/proc/levitace.pdf>.

- [12] I. WIKIPEDIA FOUNDATION, *Serial communication*. online, 2015. [http://en.wikipedia.org/wiki/Serial\\_communication](http://en.wikipedia.org/wiki/Serial_communication).
- [13] L. WILLIAMS, *Electromagnetic levitation thesis*, tech. rep., University of Cape Town, 2005. [http://www.rrsg.ee.uct.ac.za/theses/ug\\\_projects/williams\\\_ugthesis.pdf](http://www.rrsg.ee.uct.ac.za/theses/ug\_projects/williams\_ugthesis.pdf).
- [14] J. ŠIMA A KOLEKTÍV, *Anorganická chémia*, no. ISBN:978-80-227-4068-5, 2009.

# Prílohy

Príloha A: CD médium – práca v elektronickej podobe