

Univerza v Ljubljani
Fakulteta za elektrotehniko

Polde Benedik

Komunikacija USB med centrifugo in osebnim računalnikom

Magistrsko delo

Mentor: prof. dr. Andrej Žemva

Ljubljana, 2015

Univerza v Ljubljani
Fakulteta za elektrotehniko

Polde Benedik

Komunikacija USB med centrifugo in osebnim računalnikom

Magistrsko delo

Mentor: prof. dr. Andrej Žemva

Ljubljana, 2015

Zahvala

Zahvaljujem se Andreju Žemvi, ki me je vodil in usmerjal pri izdelavi diplomske naloge.
Zahvalil bi pa se tudi staršem in sodelavcem, ki so me spodbujali pri izdelavi dela.

Povzetek

V pričujočem delu je predstavljena realizacija komunikacije med napravo centrifuga in osebnim računalnikom. Pri tem je bilo treba pravilno napisati program za oddajanje podatkov za mikrokrmilnik, ki je vključeval branje in nastavljanje vrednosti. Na računalniku je bilo treba napisati program za sprejemanje podatkov. Za vmesnik med mikrokrmilnikom in računalnikom se je izdelalo posebno elektronsko vezje.

Ključne besede: centrifuga, USB, mikrokrmilnik STM8S207, Java, UART

Abstract

The main task of this diploma thesis is design and implementation of communication between the device centrifuge and the computer. It was necessary to correctly write a functions for transmitting and receiving data to the microcontroller. Second path of the program was on the computer. Second part includes graphic interface on the computer and interface for communication with centrifuge. Because microcontroller doesn't include USB module, it was necessary to build interface circuit.

Keywords: centrifuge, USB, microcontroller STM8S207, Java, UART

Kazalo vsebine

Kazalo vsebine.....	I
Seznam slik.....	III
Seznam tabel.....	V
Seznam uporabljenih simbolov	VII
1. Uvod	1
2. Komunikacija USB.....	3
2.1. Opis.....	3
2.2. Topologija vodila.....	3
3. Centrifuga	13
4. Vezje.....	15
4.1. Centrifuga vezja.....	15
5. Elementi.....	19
5.1. Čip FT232.....	19
5.2. Mikrokrmilnik STM8S207C	21
5.3. Programiranje mikroprocesorja	25
5.4. ST visual studio	25
5.5. Prevajalnik Cosmic.....	25
5.6. Krmiljenje uporabniškega vmesnika	27
5.7. Komunikacija z računalnikom.....	28
6. Vmesnik v operacijskem sistemu Windows.....	33
7. Aplikacija	35
7.1. Programski jezik Java.....	35

7.2. Knjižnice	35
7.3. Niti	37
7.4. Opis delovanja.....	40
7.5. Opis vmesnika.....	42
8. Uporabniški vmesnik	53
8.1. Nastavitev parametrov Centrifuge	53
8.2. Graf	59
9. Sklepne ugotovitve.....	63
10. Literatura.....	65

Seznam slik

Slika 1.1: Osnovni koncept komunikacije	2
Slika 2.1: Osnovni koncept povezovanja USB naprav.....	4
Slika 2.2:Koncept žic, ki so v kablu USB	6
Slika 2.3: Različni tipi priključkov USB.	6
Slika 2.4: Model paketa žeton	9
Slika 2.5: Model podatkovnega paketa	9
Slika 2.6: Model statusnega paketa	10
Slika 2.7: Preprosta povezava med gostiteljem in napravo USB	10
Slika 2.8: Povezava gostitelj in naprava USB podrobneje	12
Slika 3.1: Centrifuga Centric350	13
Slika 4.1: Močnostni del elektronskega vezja centrifuge	16
Slika 4.2: Prikazovalni del elektronskega vezja centrifuge	16
Slika 4.3: Tiskano vezje vmesnika USB UART	17
Slika 5.1: Shema FT232	20
Slika 5.2: Mikrokrmilnik STM8S207.....	21
Slika 5.3: Programator Rlink za programiranje STM8 in STM32 mikrokrmilnikov [5]	24
Slika 5.4: Razdelitev pomnilniškega prostora pri mikrokrmilniku STM8S207.....	26
Slika 6.1: Programski model gonilnika USB.....	33
Slika 7.1: Prazen JFrame.....	43
Slika 7.2: JFrame, ki vsebuje JPanel.....	44
Slika 7.3: JFrame, ki vsebuje JTable.....	45
Slika 7.4: JFrame, ki vsebuje gradnik JSpinner.	45
Slika 7.5: JFrame, ki vsebuje JButton.	46
Slika 7.6: JFrame z gradnikom JLabel	46
Slika 7.7: JFrame, ki vsebuje gradnik JTabbedPane.....	47
Slika 7.8: JFrame, ki vsebuje JComboBox.....	47
Slika 7.9: Meni datoteka.....	48
Slika 7.10: Tabela z vnesenimi podatki za cikler	50
Slika 7.11: Gumbi za urejanje podatkov za cikler	51

<u>Seznam slik</u>	IV
Slika 8.1: Jpanel za nastavitev hitrosti vrtenja centrifuge.....	53
Slika 8.2: Jpanel za nastavitev centrifugalnega pospeška.....	54
Slika 8.3: Jpanel za nastavitev nivoja pospeševanja	55
Slika 8.4: Jpanel za nastavitev nivoja zaviranja.....	55
Slika 8.5: Jpanel za nastavitev časa delovanja centrifuge.....	55
Slika 8.6: Jpanel za nastavitev zagona centrifuge, odprtje vrat in zaustavitev centrifuge.	56
Slika 8.7: Jpanel za nastavitev intervala branja podatkov.	56
Slika 8.8: Izpis trenutnih podatkov, ki se osvežuje vsako sekundo	57
Slika 8.9: Izpis podatkov v tabeli.....	57
Slika 8.10: Graf hitrosti za prikazovanje hitrosti centrifuge v realnem času.	58
Slika 8.11: Trije jpaneli za urejanje podatkov	61

Seznam tabel

Tabela 5.1: Nastavljanje mikrokrmilnika	26
Tabela 5.2: Ukazi za nastavljanje parametrov naprave	29
Tabela 5.3: Ukazi za branje parametrov naprave	30
Tabela 5.4: Ukazi za urejanje prikazovalnika in vklop ali izklop naprave.....	31
Tabela 5.5: Napake komunikacije	32

Seznam uporabljenih simbolov

USB	Universal serial bus
SYNC	Synchronize signal
PID	Packet identifier signal
ADDR	Address signal
CRC	Cyclic redundancy check
ENDP	Endpoint signal
UART	Universal asynchronous receiver/transmitter,
EEPROM	Electrically Erasable Programmable Read-Only Memory
FIFO	First in first out
HSE	High-speed external clock
ASCII	American Standard Code for Information Interchange
SWIM	Single wire interface module
SWD	Serial wire debugging
JDK	Java development kit
pdf	Portable Document Format
dll	Dynamic-link library
jpeg	Joint Photographic Experts Group
png	Portable Network Graphics

xls Excel Binary File Format

API Application programming interface

1. Uvod

Razvoj gre v smeri komunikacij, zato je treba tudi preprostejše naprave posodobiti in jih opremiti s komunikacijsko tehnologijo, ker mora uporabnik beležiti in spremljati podatke na komunikacijskih napravah. Cenovno ugodni mikrokrmilniki ne ponujajo vgrajenega celotnega komunikacijskega modula, zato je treba dodati zunanji vmesnik. Ti so lahko tudi prednost, ker se za spremembo načina komunikacije spremeni samo zunanji vmesnik in ni potrebne nove spremembe na krmilnem vezju naprave.

Prvotni obris komunikacije je bila komunikacija preko serijskih vrat. Vendar so serijska vrata že sorazmerno zastarela in se uporabljajo večinoma samo še v industriji. Serijska vrata so ena najenostavnejših serijskih komunikacij za uporabo. Vendar ta način komunikacije izginja.

Osnovna komunikacija je preko modula UART na mikrokrmilniku, ki komunicira preko vmesnika z računalnikom. Prvi prototip je deloval, vendar je projekt potreboval preskok v tehnologiji. Zato sem se odločil za naprednejšo tehnologijo. Naslednja stopnja je bila USB tehnologija, ki je izpodrinila tehnologijo serijskih vrat. Tehnologija USB je veliko bolj kompleksna za uporabo, vendar obstajajo podobni vmesniki za komunikacijo kot za serijsko vrata.

Vmesnik je zelo pomemben, ker olajša kompleksnost programiranja in omogoča preprosto komunikacijo med modulom UART in računalnikom. Pri tem se poenostavi programiranje na obeh koncih. Pri mikrokrmilniku ni treba nastavljanja protokola USB. Na računalniku so nastavljeni gonilniki, ki imitirajo delovanje serijskih vrat.

Vmesnik na računalniku imitira serijska vrata. Zato je bilo treba narediti aplikacijo za branje in pisanje podatkov serijskih vrat. Aplikacija deluje v operacijskem sistemu Windows. Za izdelavo aplikacij obstaja veliko različnih okolij, za njen razvoj pa plačljiva in brezplačna orodja. Pri orodjih za razvoj aplikacij je zelo pomemben programski jezik in kakšna je podpora s primeri in vnaprej napisanimi knjižnicami.

Knjižnice so zelo pomembne za lažje programiranje. Vsebujejo funkcije, s katerim se skrajša dolžina kode, ki jo je treba napisati za programiranje. Najpomembnejša je knjižnica za

grafični vmesnik, ki se ga oblikuje z računalniško miško, pomembna pa je še knjižnica za komunikacijo – vsebuje funkcije za vmesnik pri komunikaciji z napravo in funkcijo za detekcijo naprav ter za pošiljanje in sprejemanje podatkov. Za aplikacijo je bilo treba uporabiti še dve knjižnici. To sta knjižnici za ustvarjanje datotek formata pdf in xls.



Slika 1.1: Osnovni koncept komunikacije

2. Komunikacija USB

Vmesnik USB za komunikacijo je bil razvit za tri namene [12]. Prvi namen je bila komunikacija med telefonom in računalnikom. Drugi namen komunikacije USB je enostavna uporaba, da jo uporabnik samo priklopi in takoj začne uporabljati. Tretji namen pa je razširljivost vhodov.

2.1. Opis

Sistem USB je razdeljen na tri dele:

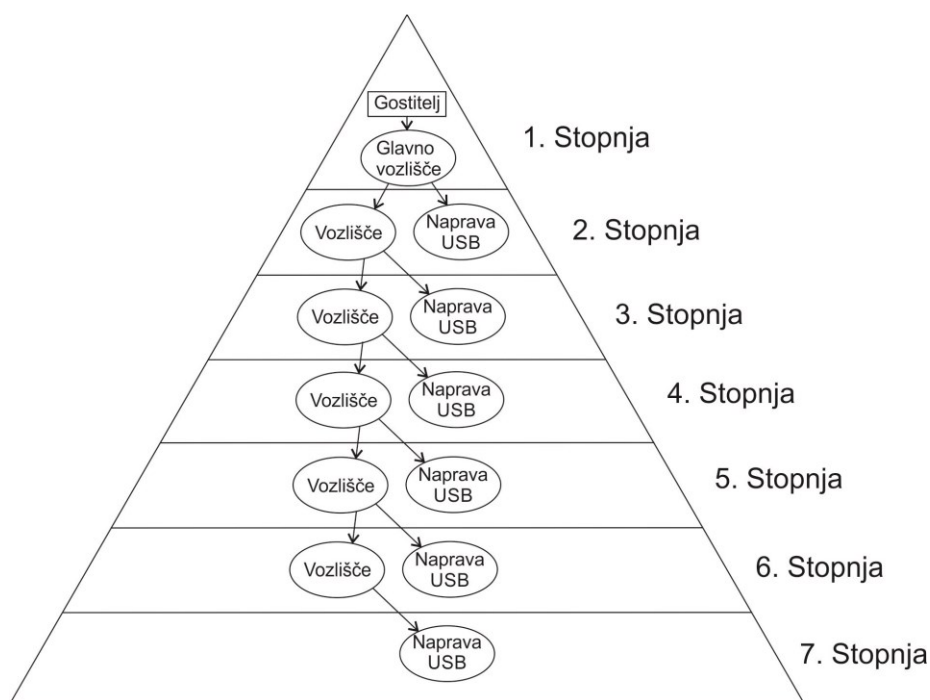
- povezava USB (USB interconnect)
- naprave USB (USB devices)
- gostitelj USB (USB host)

Povezava USB je način, kako so naprave USB med seboj povezane in kako gostitelj USB komunicira z njimi.

2.2. Topologija vodila

Komunikacija USB poveže naprave USB z gostiteljem USB [15]. USB je povezan v zvezno topologijo. Vozlišče (Hub) je center vsake zvezde. Vsaka povezava je od gostitelja do vozlišča ali funkcije oziroma drugega vozlišča, kar se vidi na Sliki 2.1.

S Slike 2.1 so razvidne tudi povezane naprave USB. Osnovni koncept povezovanja naprav USB je zvezna topologija, kjer je v središču gostitelj, ki je povezan na vozlišče. Na vozlišče



Slika 2.1: Osnovni koncept povezovanja USB naprav.

pa so povezane tudi naprave USB in vozlišča. Zaradi časovnih zakasnitev je možno povezati do sedem zaporednih vozlišč.

2.2.1. Gostitelj USB

Sistem USB ima samo enega gostitelja ali gospodarja. Vmesnik USB se za gostiteljski računalnik imenuje gostitelj USB. Gostitelj je odgovoren za najbolj pomembne naloge komunikacije USB.

To so:

- dodajanje in odstranjevanje naprav USB,
- upravljanje prenosa podatkov med gostiteljem USB in napravo USB,
- pregledovanje statusa in vodenje statistike,
- zagotavljanje napajanja naprave USB.

2.2.2. Naprave USB

Naprave USB delimo na dva dela:

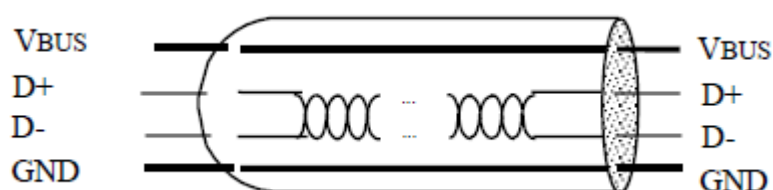
- vozlišča, ki dodajo dodatne točke priklopa naprav USB,
- funkcije, ki dodajo dodatne funkcije sistemu.

2.2.3. Fizikalne karakteristike USB

Električne lastnosti

Za prenos podatkov preko komunikacije USB skrbi štirižilni kabel (njegova skica je na Sliki 2.2 [13]), pri katerem sta dve namenjeni podatkom in dve napajanju.

Napajanje je sestavljeno iz dveh povezav GND in Vbus. Vbus je napajalna napetost 5 V, GND pa je masa. Komunikacija USB omogoča dolžine kablov različnih velikosti – do nekaj metrov – pri čemer je dolžina kabla odvisna od hitrosti povezave.



Slika 2.2: Koncept žic, ki so v kablu USB

Obstajajo štiri hitrosti podatkov:

- nizka hitrost (Low speed) USB 1.0 omogoča hitrosti do 1,5 megabita na sekundo,
- polna hitrost (Full speed) USB 1.1 omogoča hitrosti do 12,5 megabita na sekundo,
- visoka hitrost (High speed) USB 2.0 omogoča hitrosti do 480 megabitov na sekundo,
- super hitrost (Super Speed) USB 3.0 omogoča hitrosti do 5 gigabitov na sekundo.

Priključki USB

Priključki USB se delijo po velikosti priključka do tipa priključka. Tipa priključka sta dva, tip A in tip B. Po velikosti pa se delijo v osnovi na tri, vendar je še veliko drugih priključkov. Osnovni trije priključki so standardni USB, mini USB in mikro USB. Na Sliki 2.3 vidimo šest različnih priključkov USB. Pri komunikaciji s centrifugo se uporabljata prvi in drugi priključek z desne. Prvi oziroma priključek USB B se priklopi na napravo, drugi oziroma priključek USB A pa se priključi na računalnik.



Slika 2.3: Različni tipi priključkov USB.

2.2.4. Protokol komunikacije USB

Komunikacija USB deluje po načinu gospodar suženj, kjer gostitelj USB predstavlja gospodarja in naprava USB predstavlja sužnja. Gospodar vedno začne komunikacijo in suženj mu vedno samo odgovarja.

Večina prenosov preko vodila USB vključuje prenos do treh paketov. Prenos se začne, ko gostitelj načrtno pošilja paket USB, ki opisuje tip in smer prenosa do naslova naprave USB oziroma končne točke. Ta paket imenujemo žeton (token packet). Izbrana naprava USB se izbere z dekodiranjem paketa, ki ga je prejela. V tem prenosu podatkov so podatki lahko preneseni od gostitelja k napravi USB ali od naprave USB h gostitelju. Smer podatkov je specifičirana v paketu žeton. Nato pa vir prenosa podatkov pošlje podatke oziroma obvestilo, da nima ničesar za pošiljanje. Prejemnik sporoči s statusnim (handshake) paketom, da je bil prenos uspešen.

Obstajajo pa tudi prenosi podatkov s štirimi paketi [16]. To so prenosi, kjer je potrebno pretvarjati med gostiteljem in napravo, ki deluje z nizko hitrostjo.

Signali

Signal SYNC začenja vsak prenos, pri čemer le-ta določa maksimalno gostoto podatkov in je pri nizki hitrosti dolžina 8 bitov, pri visoki hitrosti pa je dolžina 32 bitov.

Signal PID vedno sledi signalu SYNC. Signal PID je osem-biten in vsebuje podatke o tipu paketa. Signal PID sestavlja osem bitov, pri čemer so 4 za podatke in 4 za preverjanje pravilnosti podatkov. Biti za preverjanje pravilnosti signala so negirani biti. V primeru, da naprava sprejme paket z napako, ga naprava ignorira. V primeru, da je naprava dobila paket, na katerega ne more odgovoriti, se ne odziva.

Signal PID je določen v štiri različne skupine paketov, to so žeton, podatki, status in specialni paket.

Signal ADDR nastavi končni naslov komunikacije USB. Končni naslov je lahko sprejemnik ali oddajnik, odvisno od signala PID. Signal ADDR je dolg sedem bitov, zato lahko protokol USB naslovi 127 naprav USB, pri čemer je naslov 0 rezerviran za privzeti naslov in ga ne more imeti nobena naprava.

Signal ENDP je sestavljen iz štirih bitov. Uporablja se za bolj fleksibilno nastavljanje naslovnikov, še posebej v primeru, da je treba nastaviti več naprav USB. Biti so specifični za funkcijo, razen če so v stanju 0. Možne so štiri funkcije, funkcije komunikacije so lahko sprejem, oddaja podatkov, nastavitve in dosegljivost naprave. Vse funkcije morajo podpirati nadzorno cev s funkcijo 0, pri čemer naprave z nizkimi hitrostmi podpirajo funkcije s tremi cevmi. Naprave s polno in visoko hitrostjo imajo lahko do 16 cevi.

Signal DATA ima dolžino do 1024 bajtov, ki pa je odvisna od dolžine podatka in se spreminja. Vendar pa mora biti dolžina signala DATA celo število.

Signal CRC se uporablja za zaščito vseh signalov razen signala PID v podatkovnih paketih in v paketih z žetonom. Signal CRC generira oddajnik pred dodajanjem glave paketa. Branje sporočila je izvedeno po odstranitvi glave paketa. Pred branjem pride do preverjanja kode CRC. V primeru, da pride do napake pri preverjanju kode CRC, sprejemnik zavrže celoten signal, v večini primerov pa celoten paket.

Žeton (token packet)

Paket se uporablja za vzpostavitev povezave med gospodarjem in napravo USB. Na sliki 2.4 vidimo sestavo paketa žeton. Paket je sestavljen iz petih signalov, to so: SYNC, PID, ADDR, ENDP in pet-bitna koda CRC. Osnovni model paketa je viden na Sliki 2.4.

*Slika 2.4: Model paketa žeton****Paket podatkov (data packet)***

Podatkovne pakete pošlje naprava USB ali gospodar, odvisno, kdo želi pošiljati podatke. Sestava podatkovnega paketa je na Sliki 2.5. Podatkovni paket sestavljajo signali SYNC, DATA1, DATA in šestnajst biten CRC.

*Slika 2.5: Model podatkovnega paketa****Statusni paket (handshake)***

Statusni paket se uporablja za končanje prenosa podatkov USB in se vedno pošlje po končani komunikaciji. Sestava statusnega paketa je prikazana na Sliki 2.6. Paket je sestavljen iz dveh signalov SYNC in PID. Signal PID vsebuje sporočilo, ali je naprava USB podatke sprejela in prebrala uspešno, ali pa je prišlo med pošiljanjem podatkov do napake.



Slika 2.6: Model statusnega paketa

2.2.5. Programski model

Komunikacija USB je sestavljena iz gostitelja in naprave USB. Za razumevanje je treba komunikacijo USB razdeliti na več delov. Možni sta dve razdelitvi: model naprave in model komunikacije.



Slika 2.7: Preprosta povezava med gostiteljem in napravo USB

Model naprave USB ima štiri dele:

- Naprava USB: to je neka strojna oprema z uporabno funkcijo, ki je povezana z gostiteljem preko povezave USB.
- Programski vmesnik: program, ki se opravlja na gostitelju in komunicira z napravo USB.
- Sistemski programski vmesnik USB: program, ki podpira komunikacijo USB na izbranem operacijskem sistemu. Običajno jo zagotavlja operacijski sistem.
- Gostitelj USB (USB host controller): strojna in programska oprema, ki dovoljuje, da

je naprava USB priključena na gostitelja.

Model komunikacije USB ima štiri dele:

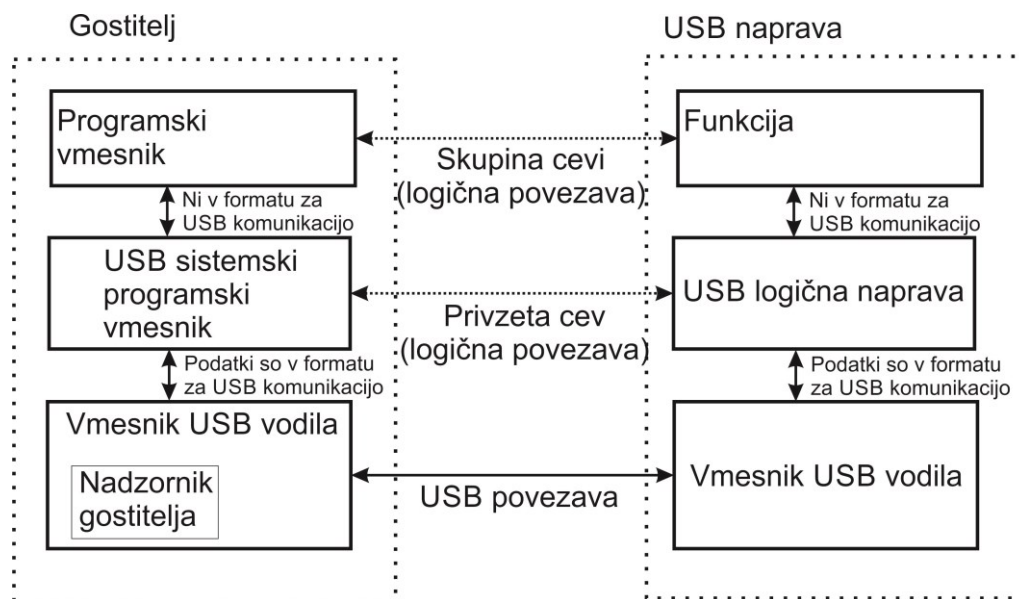
- Topologija vodila: predstavlja primarno logično in fizično komponento USB in kako sta povezani.
- Model pretoka podatkov: opisuje, kako podatki tečejo med gostiteljem in napravo preko USB in definira štiri USB načine prenosa.
- Upravljanje dostopa do vodila: opisuje upravljanje dostopa do vodila znotraj gostitelja, da je pretok podatkov naprav USB čim večji.
- Posebna obravnava za izohrone prenose: opisuje posebne primere naprav, ki potrebujejo izohroni prenos podatkov. Pri tem je vnaprej dogovorjena pasovna širina in ni preverjanja napak. To pa se uporablja pri prenosu video in avdio signala.

USB pretok podatkov

Pretok podatkov komunikacije USB zagotavlja komunikacijo med aplikacijo na gostitelju in funkcijo USB. Vsaka naprava potrebuje drugačen način prenosa podatkov, ker je vsak prenos podatkov med napravo USB in gostiteljem USB drugačen.

Logična naprava USB se gostitelju prikaže kot skupina končnih točk. Končne točke se združujejo v nize končnih točk, ki skupaj tvorijo vmesnik USB. Vmesnik je viden kot funkcija.

Aplikacija gostitelja USB nadzoruje sistem preko privzete nadzorne cevi. Aplikacija odjemalca nadzoruje vmesnik preko skupine cevi (pipe bundles). Odjemalec zahteva prenos podatkov za prenos le-teh iz pomnilnika gostitelja do USB naprave. Nadzornik gostitelja pošlje podatkovne pakete preko vodila USB. Slednji tudi nadzoruje, kdaj je uporabljeno vodilo za prenos podatkov.



Slika 2.8: Povezava gostitelj in naprava USB podrobneje

Programska cev komunikacije USB

Cev je povezava med napravo USB in gostiteljem, ter je izključno programski izraz. Zagotavlja prenos med pomnilnikom na gostitelju in pomnilnikom na napravi USB.

Vrste prenosa podatkov

- Kontrolni prenos podatkov (Control transfers): kontrolni prenos se uporablja za konfiguracijo, kontroliranje in preverjanje statusa naprave USB.
- Prekinitveni prenos podatkov (Interrupt Transfer): prekinitveni prenos podatkov se uporablja ob neperiodičnem oziroma prenosu podatkov samo ob določenem času.
- Izohroni prenos podatkov (Isochronous transfers): izohroni prenosi se v okoljih izven USB komunikacije uporabljajo za konstantne prenose brez preverjanja, ali se je zgodila napaka.
- Masovni prenos podatkov (bulk transfer): masovni prenos podatkov se uporablja za prenos velikih količin podatkov.

3. Centrifuga

Centrifuga je laboratorijska naprava, kjer se predmet vrtil okrog osi in s tem naredi zelo veliko centrifugalno silo [18]. S tem se gostejše snovi pomaknejo izven centra centrifuge. Redkejša snovi pa se pomaknejo bližje centru centrifuge.



Slika 3.1: Centrifuga Centric350

Na motor centrifuge je povezan rotor. Uporabnik vstavi vanj epruvete. Te pa so različnih velikosti in tudi iz različnih materialov. Centrifuga se vrtil pri velikih hitrostih, zato morajo biti epruvete zelo vzdržljive, da ne pride do poškodb in s tem uničenjem vzorca. Zato so epruvete narejene iz ojačene plastike in stekla. Pri tem pa je zelo pomembna enakomerna razporeditev mase. To je najbolj pomembno zaradi varnosti. Ob neenakomerni porazdelitvi mase pri zelo velikih vrtiljajih pride do velikih sil in lahko odtrga rotor od naprave ter pride do poškodb uporabnika.

4. Vezje

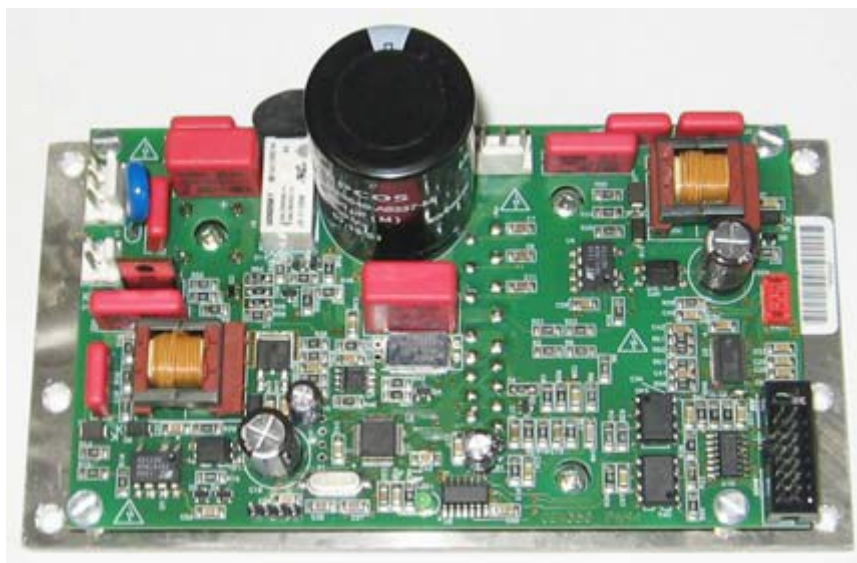
Vezje centrifuge je sestavljeno iz močnostnega dela za krmiljenje motorja in uporabniškega vmesnika, ki skrbi za nastavljanje in prikaz. Za povezavo z računalnikom in krmilnim delom skrbi vmesno vezje.

4.1. Centrifuga vezja

Vezje centrifuge je sestavljeno iz dveh delov močnostnega in prikazovalnega dela, pri čemer močnostni del skrbi za napajanje in pravilno krmiljenje motorja s frekvenčno regulacijo. Prikazovalni del skrbi za vmesnik med centrifugo in uporabnikom, kjer se nastavljajo parametri naprave.

4.1.1. Močnostni del

Močnostni del je sestavljen iz usmernika in frekvenčnega regulatorja. Usmernik pretvori omrežni izmenični signal v enosmernega. Frekvenčni regulator pa krmili motor. Vsebuje tudi napajanje za uporabniški vmesnik, ki je galvansko ločen zaradi varnosti. Galvanska ločitev je realizirana s stikalnim flyback pretvornikom. Povezava med močnostnim delom in uporabniškim vmesnikom je realizirana s komunikacijo UART. Komunikacija je galvansko ločena z uporabo dveh inverterjev za vsak signal. Prvi inverter je na močnostnem delu, drugi je na prikazovalniku.



Slika 4.1: Močnostni del elektronskega vezja centrifuge

4.1.2. Prikazovalnik

Prikazovalnik je sestavljen iz osmih sedem-segmentnih led prikazovalnikov, sedmih led diod, piskača, dveh enkoderjev s tipkami in dveh tipk. Za krmiljenje skrbi osem-bitni mikrokrmilnik STM8S207C6 [3].



Slika 4.2: Prikazovalni del elektronskega vezja centrifuge

Osem led prikazovalnikov je razdeljenih v dva dela, kjer prvi del vsebuje pet in drugi tri vmesnike. Prvi del se uporablja za prikaz hitrosti in centrifugalne sile, drugi del pa za prikaz

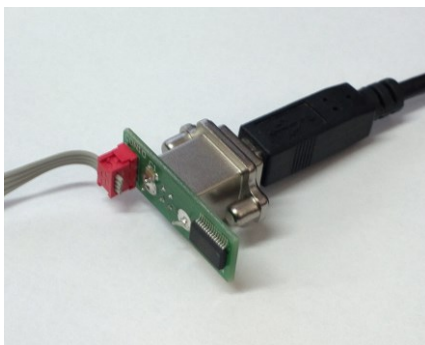
časa, nivo pospeška in nivo zaviranja.

Za vnos podatkov sta na vezju dva enkoderja s tipkama. Enkoder na levi strani skrbi za nastavljanje hitrosti in centrifugalne sile. Ko nastavljamo veličino, to izberemo s tipko na enkoderju. Enkoder na desni strani skrbi za nastavljanje časa delovanja, nivo pospeševanja in nivo zaviranja. Za preklap med veličinami skrbi tipka na njem. Na prikazovalniku sta tudi dve tipki, ki se uporabljata za vklop in izklop naprave in odprtje vrat. Za prikaz delovanja centrifuge so nameščene štiri led diode, ki se prižigajo v posebnem zaporedju, če naprava deluje. Če pa je naprava ugasnjena, so tudi led diode ugasnjene. Dve led diodi se uporabljata za indikacijo ali uporabnik nastavlja hitrost ali centrifugalno silo.

Za pomembna opozorila je nameščen piskač, ki se vklopi za posebna opozorila, kot je napaka pri delovanju ali pa konec delovanja centrifuge.

4.1.3. Vmesnik vezje

Vmesnik je sestavljen iz treh komponent: konektorja, kondenzatorja in čipa. Konektor USB-B je pritrjen za zunanje ohišje. Čip FT232R je vmesnik med komunikacijo UART in komunikacijo USB. Kondenzator je za stabilno napajanje. Vezje se napaja iz računalnika.



Slika 4.3: Tiskano vezje vmesnika USB UART

Na Sliki 4.3 je tiskano vezje vmesnika. Vmesnik pretvarja UART signal, ki ga dobi iz mikrokrmilnika STM8S207, v signal USB, ki ga pošlje naprej v računalnik.

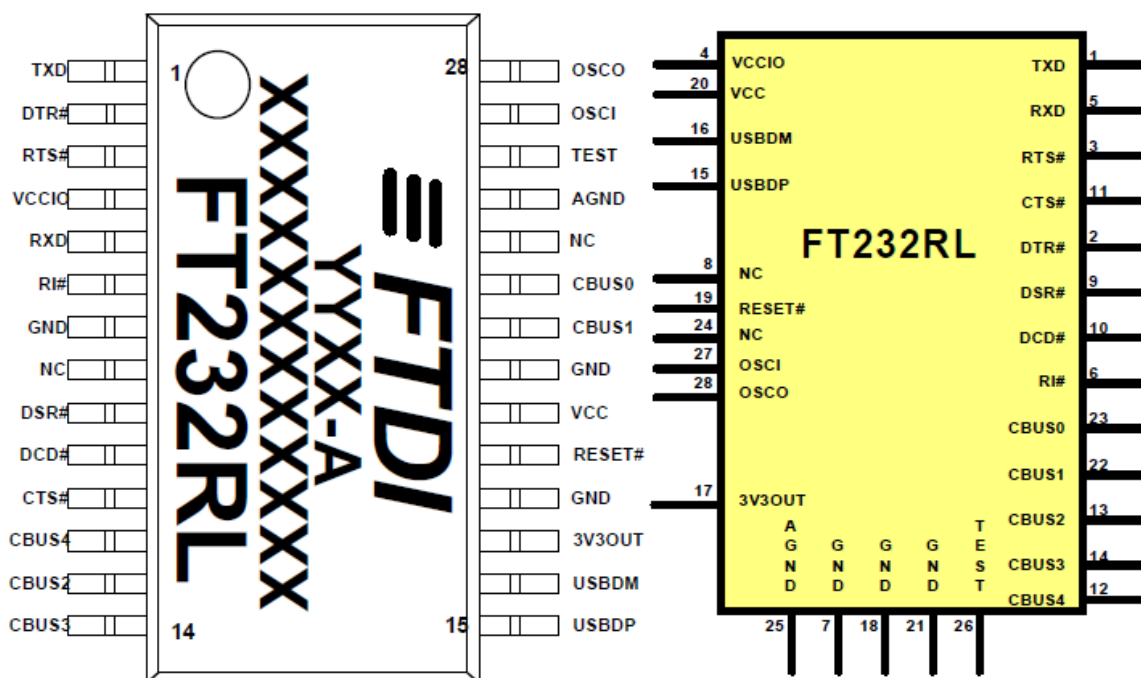
5. Elementi

Pomembni elementi na tiskanih vezjih za komunikacijo so mikrokrmilnik STM8S207 in vmesnik FT232, kjer mikrokrmilnik skrbi za pravilno delovanje naprave in pošiljanje podatkov. Vmesnik FT232 skrbi za pretvarjanje signala UART v signal USB.

5.1. Čip FT232

Vmesnik FT232 omogoča komuniciranje med različnimi komunikacijami, kot so USB, UART, I2C, SPI in za paralelno komunikacijo [2]. Za pretvorbo med komunikacijami je potrebno povezati signale in FT232 pretvori signal med vhodnim signalom in signalom USB. Pri komunikaciji med računalnikom in centrifugo je bil uporabljen UART vmesnik. Iz tega vmesnika je treba pripeljati dva signala na FT232. To sta signala TX in RX, pri čemer TX oddaja signal in čip FT232 ga sprejema, RX pa sprejema signal in čip FT232 ga oddaja. Pri povezavi z računalnikom je treba povezati dva signala: D+, D- in napajanje.

FT232 vsebuje notranji EEPROM, ki ga je mogoče tudi programirati. Vendar programiranje EEPROMA ni mogoče preko virtualnih serijskih vrat, ampak je treba namestiti posebne gonilnike. To je posebna knjižnica, ki jo je vnaprej napisal že proizvajalec. Ta način se imenuje D2XX [1]. Pri tem načinu je možno komunicirati z napravo in tudi pisati ter brati v EEPROM-u naprave. Vendar je že proizvajalec naredil osnovni program za pisanje vrednosti EEPROM.



Slika 5.1: Shema FT232

Pomnilnik FIFO RX

Velikost pomnilnika za pošiljanje je 128 bajtov. Podatki, ki so prejeti preko komunikacije USB, se shranijo v pomnilnik FIFO RX. Iz pomnilnika so odstranjeni v register UART za pošiljanje. Register za pošiljanje je pod nadzorom nadzornika UART FIFO.

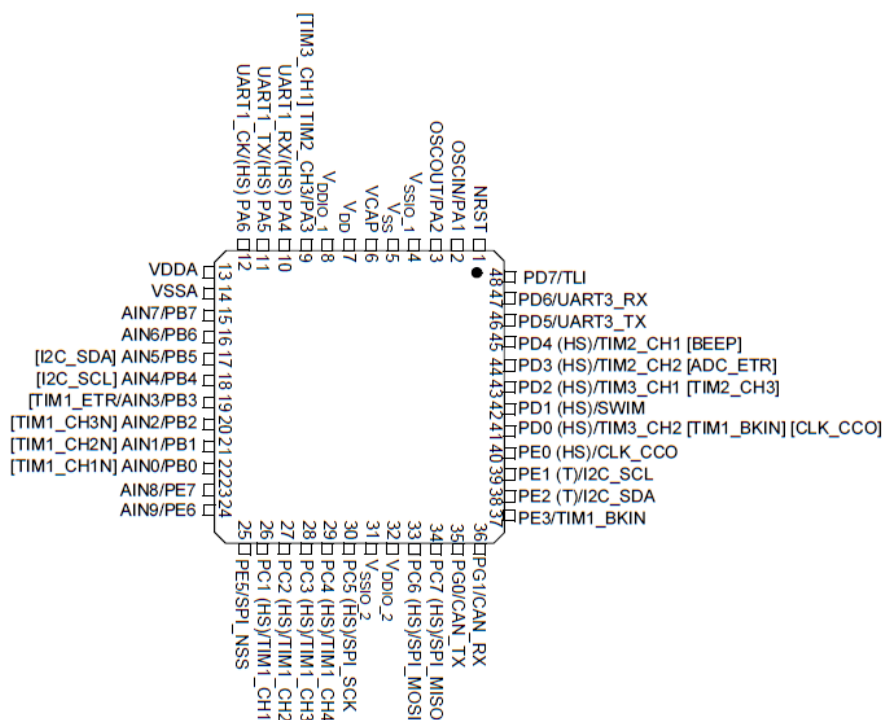
Pomnilnik FIFO TX

Podatki, ki so bili sprejeti preko vhoda TX, so shranjeni v pomnilnik TX. Nadzornik USB odstrani podatke iz pomnilnika TX FIFO in jih pošlje na zahtevo USB za podatke na računalniku. Velikost pomnilnika FIFO TX je 256 bajtov.

5.2. Mikrokrmilnik STM8S207C

Mikrokrmilnik je osembitni, ima 38 vhodno izhodnih pinov [3] ter 64 kilobajtov pomnilniškega prostora. Vključuje tudi analogno digitalni pretvornik, dva modula UART, tri šestnajst-bitne časovnike (timer) in enega osem-bitnega.

Procesor je možno nastaviti, da deluje na notranjo ali zunanjo uro. Za zunanjo uro je potrebno priklopiti kvarc na vhoda PA1 in PA2. Procesor deluje na zunanji uri 16 MHz, ki jo je treba nastaviti programsko, ima pa tudi notranji vir ure HSE.



Slika 5.2: Mikrokrmilnik STM8S207

5.2.1. Časovniki (timer)

Timer1 je šestnajst-bitni časovnik z dodatnimi funkcijami. Uporablja se za meritev obratov, ti pa se merijo z bipolarnim Hallovim senzorjem, ki vklopi tranzistor ob magnetnem polu sever in izklopi tranzistor ob magnetnem polu jug. Magneta sta nameščena na rotorju. Iz tega dobimo pravokotne impulze, kjer je frekvenca signala enaka frekvenci vrtenja motorja.

Timer2 in timer3 sta navadna šestnajst-bitna časovnika z dvema izhodoma. Timer2 se uporablja za krmiljenje koračnega motorja, timer3 pa za krmiljenje piskača s frekvenco 4 kHz.

Timer4 je osnovni osem-bitni časovnik. Uporablja se za časovni korak programa. Korak se izvede vsake štiri milisekunde. To pomeni, da se mora prekinitev časovno zgoditi vsake štiri milisekunde oziroma vsaj z deliteljem štirih sekund. To pa se doseže s pravilnim delilnikom ure za mikrokrmilnik in pravilno izbiro periode.

$$T_{\text{korak}} = \frac{1}{f_{\text{KVARC}} * N} \quad (1)$$

$$T_{\text{periode}} = T_{\text{korak}} * N_{\text{periode}} \quad (2)$$

T_{korak} je čas enega koraka časovnika.

N je delilnik urinega signala za timer4.

N_{periode} je število delcev, da dosežem želen čas.

T_{periode} je čas periode časovnika med dvema prekoračenima(overflow) prekinitvama časovnika.

5.2.2. Modul UART

Modul UART1 se uporablja za komunikacijo med močnostnim delom in prikazovalnikom.

Modul UART2 se uporablja za komunikacijo med računalnikom in centrifugo.

Inicializacija modula UART

Inicializacija UART vmesnika se naredi s funkcijo `UART1_config`, kjer je treba določiti parametre komunikacije. Pri meni sem nastavljal 38400 baudov, pri dolžini besede 8 bitov, en stop bit in brez preverjanja parnosti in sinhronizacije. Na koncu je treba omogočiti še modul UART. To storim s funkcijo `UART_Cmd` [21]. Po inicializaciji je treba še nastaviti sprejem podatkov. To je najbolje narediti s prekinitvijo, kjer se prekinitev izvede ob sprejemu podatkov. To se naredi s funkcijo `UART1_ITConfig`. Na koncu pa je potrebno še omogočiti prekinitve s funkcijo `enableInterrupts`. Inicializacija komunikacije UART:

```
void UART1_Config(void)
{
    UART1_DeInit();
    UART1_Init((u32)38400, UART1_WORDLENGTH_8D, UART1_STOPBITS_1, UART1_PARITY_NO,
    UART1_SYNCMODE_CLOCK_DISABLE, UART1_MODE_TXRX_ENABLE);
    UART1_Cmd(ENABLE);
    UART1_ITConfig(UART1_IT_RXNE, ENABLE);
    UART1_ITConfig(UART1_IT_TXE, ENABLE);
    enableInterrupts();
}
```

Pošiljanje podatkov z modulom UART1

Pošiljanje podatkov je realizirano s pomočjo niza znakov. Zato je treba vsako število pretvoriti v niz znakov. Za to pretvorbo se uporablja funkcija `sprintf`, ki pretvori število v niz ASCII znakov [17]. Funkcija `sprintf` vsebuje tri različne parametre. Prvi parameter je niz

znakov, v katerega funkcija shrani vrednost. Drugi parameter je model zapisa v niz znakov. Naslednji parametri so spremenljivke, ki jih bo funkcija zapisala v znakovni niz. Niz znakov pošlje naprej UART modul, ki pošlje naprej vsak znak posebej.

Prejem podatkov z modulom UART1

Sprejem podatkov je v pomnilnik, ki ga program prebere vsak cikel. Cikel se izvede vsake štiri milisekunde. Pri branju pomnilnika mikrokrmilnik najprej preveri, ali je bil sprejet celoten niz. Konec niza določa znak za presledek iz kode ASCII. Ob sprejetju celotnega niza mikrokrmilnik preveri, ali je bil prejet znan ukaz ali ne. Če je bil prejet neznan ukaz, pošlje nazaj, da se je zgodila napaka. V primeru pravilnega ukaza pa dejanje izvede.

5.2.3. Programator RLINK



Slika 5.3: Programator Rlink za programiranje STM8 in STM32 mikrokrmilnikov [5]

Rlink je programator za programiranje mikrokrmilnikov STM8 in STM32 [6]. Mikrokrmilniki STM8 so osem-bitni. Mikrokrmilniki STM32 so 32-bitni. Obe družini imata različna vmesnika za programiranje mikrokrmilnikov. Za osem-bitne se uporablja vmesnik SWIM, za 32-bitne pa se uporablja vmesnik SWD.

5.3. Programiranje mikroprocesorja

Programiranje mikrokrmilnika je razdeljeno v dva dela. Prvi del je pisanje kode. To se dela v programu ST Visual Studio [4]. Vendar proizvajalec STM ne nudi prevajalnika(compiler) za programiranje v programskem jeziku C, zato je treba uporabiti prevajalnik zunanjega proizvajalca.

5.4. ST visual studio

ST visual studio je razvojno orodje za razvoj aplikacij za procesorje tipa STM8 in STM32. Program omogoča programiranje v zbirniku(asembler) in v programskem jeziku C. Vsebuje preprost uporabniški vmesnik za programiranje brez naprednih funkcij.

5.5. Prevajalnik Cosmic

Prevajalnik Cosmic je posebej narejen za mikrokrmilnike družine STM8 [10]. Prevajalnik prevede kodo in jo s povezovalnikom(linker) združi v strojno kodo tipa intel hex. Pri tem so zelo pomembne nastavitve glede velikosti procesorja. Prevajalnik Cosmic omogoča dve nastanitvi pomnilniškega prostora. Prva nastavitev je za mikrokrmilnike, ki imajo vgrajeni pomnilnik flash z manj kot 64 kilobajti, drugi pa je za velikosti večje od 64 kilobajtov.

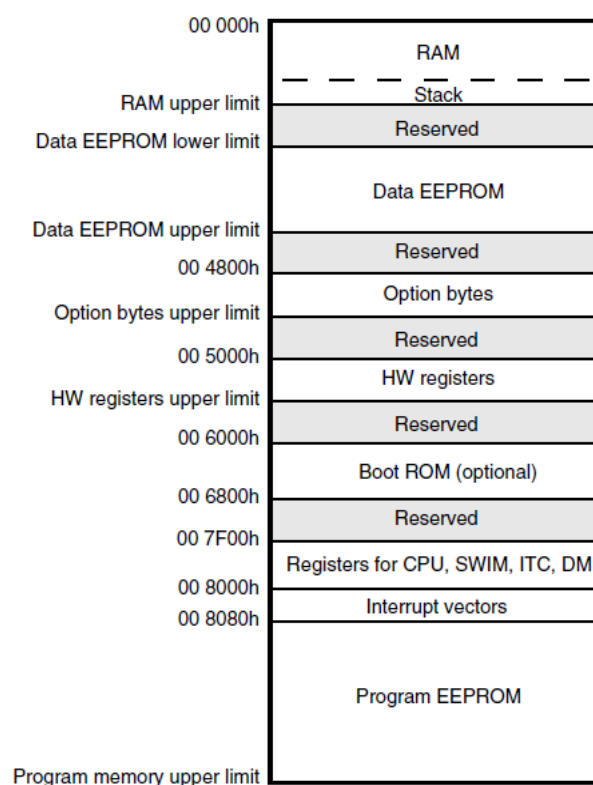
5.5.1. Razporeditev pomnilniškega prostora

Razporeditev pomnilniškega prostora vidimo na Sliki 5.4, kjer je razvidno, da je prostor za programsko kodo na koncu pomnilniškega prostora. Pri tem je zelo pomembno, da v privzetem pomnilniškem modelu prevajalnik Cosmic uporablja 16-bitno naslavljanje.

Šestnajst-bitno naslavljanje se nastavi z nizom `@near` pred funkcijo. Pri šestnajst-bitnem naslavljanju je možno nasloviti samo 65536 bajtov. Vendar je treba odšteti polovico naslovov za naslavljanje pomnilnika RAM in vhodno-izhodnih enot. Iz tega dobimo, da ostane za programsko kodo in podatke samo 32639 bajtov. Za uporabo mikrokrmilnika, ki ima 64 kilobajtov flash ROM-a, je potrebno uporabiti 24-bitno naslavljanje. S tem povečamo naslovni prostor na 16 megabajtov. 24-bitno naslavljanje se nastavi z uporabo niza `@far` pred funkcijo, ki bi imela 24-biten naslov.

Niz	Dolžina naslova	Število naslovov
<code>@near</code>	16-biten	32 kilobajtov
<code>@far</code>	24-biten	16 megabajtov

Tabela 5.1: Nastavljanje mikrokrmilnika



Slika 5.4: Razdelitev pomnilniškega prostora pri mikrokrmilniku STM8S207

5.5.2. Program mikrokrmilnika

Osnovni koncept programa mikrokrmilnika je delovanje v neskončni while zanki. Omejitev zanke je izvajanje vsake 4 milisekunde. Funkciji z zanko je bil prejet določen ukaz iz UART komunikacije z računalnikom in switch zanko, ki skrbi za delovanje uporabniškega vmesnika.

5.6. Krmiljenje uporabniškega vmesnika

Krmiljenje uporabniškega vmesnika je realizirano s switch stavkom, ki ima 36 stavkov case. Vendar so za osnovno delovanje naprave pomembni samo naslednji:

- M_PWRON v tem stavku počakamo 1 sekundo in preverimo, ali uporabnik drži obe tipki enkoderja. Če uporabnik drži obe tipki enkoderja, vstopimo v servisni način, če ne, pa v običajni način. Na prikazovalniku ta čas svetijo vse led diode in vsi led prikazovalniki prikazujejo številko osem.
- M_INIT_PWRON v tem case stavku počakamo 2 sekundi. Na prikazovalniku se izpiše verzija naprave in verzija programa.
- M_INIT v tem case stavku preveri, ali je povezava z močnostnim delom. In če ni, počaka, da se povezava vzpostavi. Na prikazovalniku se prikazujejo sami minusi.
- M_RDY_SERVICE je servisni način, kjer preverimo delovanje Hallovih senzorjev in vstop v serijski način. Serijski način se uporablja za nastavitev parametrov naprave, do katerega imajo dostop samo v proizvodnem procesu in serviserji.
- M_RDY je način, kjer nastavlja uporabnik parametre delovanja. To so čas delovanja, nivo pospeševanja, nivo zaviranja, centrifugalno silo in hitrost vrtenja motorja.
- M_RDY_OFF je način varčevanja z energijo, kjer po dveh urah naprava ugasne prikazovalnik zaradi varčevanja z energijo.

- M_ACEL_ROTTYPE detekcija tipa rotorja, ki je v centrifugi.
- M_ACCEL_RUN je način, kjer centrifuga pospešuje do nastavljene vrednosti. Na prikazovalniku pa prikazuje merjeno hitrost.
- M_RUNNING je način, kjer naprava deluje in prikazuje nastavljeno hitrost, centrifugalno silo, nivo pospeška in nivo zaviranja. Prikazuje tudi čas do konca delovanja. Če konec delovanja ni nastavljen, prikazuje napis hold. Ob preteku časa ali ustavitvi delovanja, pa naprava preklopi v način zaviranja.
- M_DECEL je način, kjer naprava zavira. Hitrost in centrifugalno silo centrifuga prikazuje merjeno. Časa ne prikazuje, ker prikazuje nivo zaviranja.
- M_DECEL_QUICK je način hitrega zaviranja, ki se uporablja v primeru napake med delovanjem centrifuge.

5.7. Komunikacija z računalnikom

Komunikacija z računalnikom je realizirana z if else stavki, ki so med seboj povezani. If stavek preveri, ali je prejeti ukaz enak. In če je, se izvede in zaključi s funkcijo ter preveri prejete podatke ob naslednjem ciklu. Ukazi so sestavljeni iz znakov kode ASCII, kjer dva ali trije znaki povejo ukaz, nato pa lahko sledi parameter, če je dan ukaz za nastavljanje le-teh. Na koncu vseh ukazov je znak kode ASCII za presledek, ki pove napravi, da je ukaza konec.

Ukazi so razdeljeni v dve glavni skupini in več manjših. Prva skupina je branje parametrov naprave, druga pa nastavljanje parametrov naprave.

Ukazi za branje parametrov naprave

Ukazi za nastavljanje so sestavljeni iz dveh delov. Prvi del sta dva znaka, ki pomenita, kaj želi računalnik nastaviti. Drugi del je sestavljen iz parametra, ki ga želi računalnik nastaviti na centrifugi. Ta pa je odvisen od parametra, ki ga želimo nastaviti, ker ima vsak parameter drugačen obseg vrednosti. Obstaja pa še poseben ukaz, ki skrbi za nastavitev časovno neomejenega delovanja. Ta pa je sestavljen samo iz dveh znakov QT.

Ukaz	Odgovor	Opis
STxxxxx		Nastavitev časa
SPxxx		Nastavitev nivoja pospeška
SDxxx		Nastavitev nivoja zaviranja
SSxxxxx		Nastavitev hitrosti
SAxxxxx		Nastavitev centrifugalne sile
QT		Nastavitev delovanja hold

Tabela 5.2: Ukazi za nastavljanje parametrov naprave

Ukazi za branje parametrov naprave

Ukazi za branje parametrov naprave so uporabljeni za branje le-teh, kjer računalnik pošlje zahtevo in naprava odgovori z ukazom ter vrednostjo, ki jo računalnik zahteva. Razdeljeni so v nekaj skupin. Prva skupina je branje merjenih parametrov naprave, kot so čas delovanja naprave, merjena hitrost in centrifugalna sila. Druga skupina ukazov pa so nastavljeni parametri naprave. Ti se razdelijo še v več skupin – to so nastavljeni parametri, ki jih nastavlja uporabnik: nastavljena hitrost, centrifugalna sila, čas delovanja, nivo pospeševanja in nivo zaviranja. Nastavljeni parametri, ki jih parametri ne morejo nastavljati, so omejitve za nastavljanje hitrosti, centrifugalne sile, časa delovanja, nivoja pospeševanja in nivoja zaviranja. Pri tem so še trije posebni ukazi: prvi ukaz je identifikacija naprave, drugi ukaz pošlje stanje delovanja, tretji ukaz pa pošlje stanje napake oziroma, ali se je zgodila napaka ali

ne.

Ukaz	Odgovor	Opis
CD	CDXXXXXCRLF	Nastavljen nivo zaviranja
CP	CPXXXXXCRLF	Nastavljen nivo pospeševanja
ID	CICentXCRLF	Ime naprave in tip
HA	HAXXXXXXCRLF	Maksimalen centrifugalni pospešek
CA	CAXXXXXXCRLF	Nastavljena centrifugalna sila
RA	RAXXXXXXCRLF	Merjena centrifugalna sila
LS	LSXXXXXCRLF	Najmanjša nastavljena hitrost
HS	HSXXXXXCRLF	Največja nastavljena hitrost
CS	CSXXXXXCRLF	Nastavljena hitrost
RS	RSXXXXXCRLF	Merjena hitrost
HT	HTXXXXXCRLF	Največji nastavljen čas
LT	LTXXXXXCRLF	Najmanjši nastavljen čas
CT	CTXXXXXCRLF	Nastavljen čas
RT	RTXXXXXCRLF	Merjen čas
CO		Pošlje stanje naprave
EC		Pošlje, ali se je zgodila napaka

Tabela 5.3: Ukazi za branje parametrov naprave

Ukazi za spremembo delovanja naprave

Ukazi za spremembo delovanja naprave so, kjer pride do spremembe delovanja, ne pa spremembe parametra. To pa razdelimo v dve skupini. Prva skupina obsega tri ukaze, to so vklop naprave, izklop naprave in odprtje vrat na napravi. Druga skupina je sestavljena iz ukazov, ki spreminjajo prikaz veličin na prikazovalniku. Te pa lahko razdelimo v dve skupini. Prva skupina je spreminjanje prikazov v prikazovalniku s tremi led prikazovalniki. Tukaj so ukazi za prikaz časa, nivoja pospeševanja in nivoja zaviranja. Druga skupina je prikaz na prikazovalniku, ki je sestavljen iz petih led prikazovalnikov. Tukaj pa sta dva ukaza, ki

spreminjata prikaz hitrosti in prikaz centrifugalne sile.

Ukaz	Odgovor	Opis
DT		Prikaži čas na prikazovalniku
DA		Prikaži nivo pospeška na prikazovalniku
DD		Prikaži nivo zaviranja na prikazovalniku
DS		Prikaži hitrost na prikazovalniku
DP		Prikaži centrifugalno silo na prikazovalniku
ON		Naprava zažene motor na nastavljene obrate
OF		Naprava ustavi motor
DOR		Naprava odpre vrata

Tabela 5.4: Ukazi za urejanje prikazovalnika in vklop ali izklop naprave

Napake

Napake delovanja naprave se razdelijo na dva dela. Prvi del so napake komunikacije, drugi del so napake naprave.

Napake komunikacije

Napake komunikacije so napake, ki so posledica napačnih ukazov komunikacije med računalnikom in centrifugo. Napaka nepoznan ukaz se zgodi, če ukaz ni poznan napravi, v tem primeru naprava pošlje računalniku ER01. Pri nastavljanju naprave pride ob branju parametra do prevelike ali premajhne vrednosti in naprava pošlje računalniku niz ER03.

Napaka	Opis napake
ER01	Nepoznan ukaz
ER02	Napačni parametri
ER03	Ukaza se ne da izvesti

Tabela 5.5: Napake komunikacije

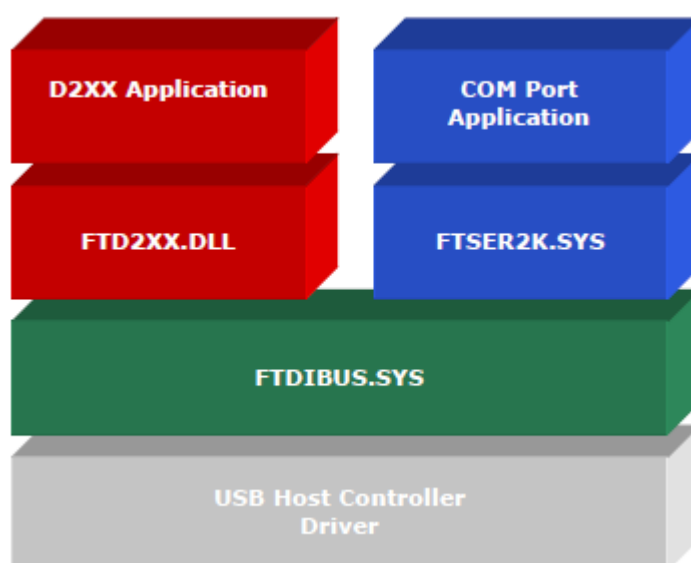
Napake naprave

Napake naprave se zgodijo, ko pride do nepravilnega delovanja naprave. Napake naprave je možno preveriti z ukazom EC, kjer centrifuga vrne niz ERXX. V tem nizu pomeni ER, da je napaka in XX je številka le-te. Ob napaki naprava prekine delovanje, izbris napake pa je možen s pritiskom tipke start stop na napravi.

6. Vmesnik v operacijskem sistemu Windows

FT232 je vmesnik med USB in serijsko UART komunikacijo z nekaj dodatnimi funkcijami.

FTDI omogoča dva načina vmesnika komuniciranja z napravami. To sta UART-USB in USB-FIFO Ics [1]. UART USB omogoča komunikacijo preko virtualnih COM vrat (virtual COM port), ki se pojavi med sistemskimi nastavitvami. Z USB-FIFO IC komunicira preko posebne datoteke DLL, vendar omogoča več dodatnih funkcij, kot jih običajna komunikacija s serijskimi vrati. Omogoča nastavljanje načinov delovanja naprave in branje ter pisanje iz EEPROMA.



Slika 6.1: Programski model gonilnika USB

6.1.1. Gonilnik VCP

Z uporabo gonilnika VCP se naprava USB v operacijskem sistemu Windows pokaže kot serijska vrata [11]. Za komunikacijo je treba uporabiti program za komunikacijo z serijskimi vrati ali napisati program s knjižnico za uporabo serijskih vrat.

6.1.2. Gonilnik DX22

Gonilnik D2XX omogoča neposredno komunikacijo preko vhoda USB preko datoteke DLL. Treba je napisati vmesni program, ki kliče funkcije iz datoteke DLL.

Funkcije iz datotek DLL omogočajo komunikacijo z napravo ter tudi branje in pisanje podatkov iz EEPROM-a, ki se nahaja v čipu FT232R. Pri pisanju v EEPROM je pomembno, ker se lahko spremeni ime naprave, ki je prikazana v računalniku in s tem postane komunikacija bolj domača uporabniku. Vsebuje pa tudi unikatno število ID, s katero prepozna čip in s tem napravo, s katero se komunicira. To je zelo pomembno pri sodobnih standardih, ker je pomembna sledljivost izdelkov.

7. Aplikacija

Aplikacija je napisana v programu Netbeans 8.02 z uporabo programskega jezika Java [8].

Program je bil preveden iz izvorne kode v izvršljivo kodo s programom JDK 1.8.

7.1. Programski jezik Java

Java je programski jezik, ki teče preko posebnega vmesnika, ki se imenuje Java virtual mashine. Posebnost programskega jezika Java je v tem, da lahko teče na različni strojni opremi in različnih operacijskih sistemih enako.

Za razvoj aplikacij v programskem jeziku Java je potrebno namestiti Java development kit, ki služi kot orodje za prevajanje kode v aplikacijo Java. Za zagon aplikacije pa je potrebno namestiti program Java runtime environment, ki je vmesnik med operacijskim sistemom in Java aplikacijo. Program Java runtime environment lahko teče na različnih operacijskih sistemih. Slabost programskega jezika Java je, da je potrebno narediti poseben vmesnik v programskem jeziku C, ki se poveže s programom, napisanim v programskem jeziku Java.

7.2. Knjižnice

7.2.1. Knjižnica Serial_lib

Knjižnica serial_lib je sestavljena iz dveh delov. Prvi del je napisan v programskem jeziku C, ki je preveden v knjižnico DLL. Drugi del je napisan v programskem jeziku Java.

Za knjižnico je treba generirati deklaracije za funkcije, ki jih kliče Java. Za deklaracijo funkcije je treba napisati posebno funkcijo v programskem jeziku C. Knjižnica DLL vsebuje funkcije za pet pomembnih funkcij za odprtje serijskih vrat, detekcijo serijskih vrat,

povezavo, pošiljanje podatkov in sprejemanje podatkov.

Drugi del knjižnice je napisan v programskem jeziku Java. Ta del kliče funkcije iz knjižnice DLL.

Vmesnik Java native lib

JNI je vmesnik med programskim jezikom Java in lokalnimi programi, ki tečejo neposredno v operacijskem sistemu [9]. JNI programi so lahko napisani v programskem jeziku C ali zbirniku [7]. JNI se uporablja za funkcije, ki se jih v programskem jeziku Java ne more napisati. V aplikaciji je v programskem jeziku C napisan vmesnik za komunikacijo s serijskimi vrati, kjer je del kode napisan v programskem jeziku Java in del kode v programskem jeziku C.

Za uporabo vmesnika JNI je potrebno najprej napisati programsko kodo v jeziku Java in s programom javah.exe narediti povezave med programskim jezikom Java in programskim jezikom C.

Inicializacija serijskih vrat

Za inicializacijo je bila potrebna knjižnica serial_lib.jar. V tej knjižnici je inicializacija serijskih vrat, kjer uporabnik izbere prava serijska vrata, na katera je priključena naprava. Ostali parametri komunikacije so isti pri vseh ostalih napravah. Ob vzpostavitvi povezave aplikacija najprej identificira napravo, s katero je povezana in izpiše stanje v gradnik jlabel, ki je v spodnjem desnem robu.

7.2.2. Knjižnica jfreechart

jFreechart je knjižnica za programski jezik Java, ki podpira risanje vseh vrst grafov [19]. Grafe izriše kot sliko in omogoča tudi shranjevanje le-teh v slike formata jpeg in formata png. V aplikaciji je bilo treba narisati kartezični graf, kjer je bil na abscisni osi merjen čas in na ordinatni osi veličina, ki pa je bila odvisna od izbire uporabnika.

Aplikacija uporablja tri različne klice funkcije za izris grafa. To so za izris grafa, prikaz veličine v realnem času in izris grafa za izpis podatkov v format pdf. Pri vseh načinih je koncept izrisa grafa podoben. Graf s pomočjo knjižnice izriše v sliko, ki jo nato aplikacija prikaže v različnem načinu. Edino pri prikazu v realnem času je potrebno še osveževanje slike, ki se zgodi vsako sekundo.

7.3. Niti

Nit je tip podprograma, ki se izvaja vzporedno z glavnim programom. Program vsebuje dve funkciji, ki se dogajata vzporedno s prikazom glavnega okna. To sta sprejem in pošiljanje podatkov.

7.3.1. Pošiljanje podatkov

Pošiljanje podatkov je realizirano s časovnikom, ki se izvede vsakih 100 milisekund. Ker je časovnik v programskem jeziku Java zelo nenatančen, ima program razdeljeno sekundo za pošiljanje podatkov. Razdeljena je na dve območji. Prvo območje je od 0 do 100 milisekund. To območje se uporablja za pošiljanje zahtev za trenutne podatke na centrifugi. Drugo

območje je od 600 do 800 milisekund. V tem območju program pošlje nastavitve, ki so se spremenile in v primeru nepravilnega sprejema podatka pošlje še enkrat zahtevo za trenutni podatek centrifuge.

Identifikacija naprave

Identifikacija naprave je potrebna zaradi tega, da program pošlje pravilne ukaze napravi. Za nastavitve naprave je treba poslati ukaz identifikacije naprave. Ukaz za identifikacijo pa se pošilja toliko časa v intervalu pol sekunde, dokler ne prejme imena naprave. Ob prejemu imena naprave le-tega nastavi na napravo, s katero je povezan in nato pošlje ukaze za doseg naprave. Ker mora program podpirati več tipov centrifug in ima vsaka drugačne nastavitve hitrosti, je treba poznati maksimalno in minimalno nastavitve veličine. Najbolj pomembno je poznati parametre hitrosti zaradi nastavljanja nastavitvev preko programa na računalniku, da se ne nastavi prevelike ali pa premajhne vrednosti.

Intervalno pošiljanje

Intervalno pošiljanje se zgodi vsako sekundo in se dogaja intervalno. Ker mora biti pošiljanje ponavljajoče se, je treba uporabiti časovnik.

Intervalni časovnik

Pri inicializaciji časovnika je treba nastaviti začetni zamik izvajanja časovnika in periodo časovnika. Pri pošiljanju je začetni zamik 0 sekund in perioda 500 ms. Perioda je nizka zaradi zelo velike napake. Za zmanjševanje napake se števec dvakrat izvede v eni sekundi.

Pri inicializaciji pa je treba vedeti, da časovnik ne deluje v realnem času in zato se program

ravna po času operacijskega sistema Windows. Pošiljanje podatkov se izvaja v dveh delih. V prvem delu se pošljejo zahteve za branje podatkov, v drugem delu pa se preveri, ali je prejel vse zahtevane podatke in če jih ni, le-te ponovno pošlje skupaj z novimi nastavitvami, če se razlikujejo od trenutnih nastavljenih vrednosti.

Tukaj se pojavi problem, ker naprava ne more slediti računalniku, ki pošilja podatke prehitro, zato je potreben zamik 7 ms za pošiljanjem ukaza. Za zamik poskrbi funkcija `Theed.sleep`, ki izvaja prazno zanko določen čas, da lahko naprava sprejme podatek in računalnik spet pošlje naslednji znak.

Program bere podatke v določenem vrstnem redu. Prvi poslani podatek mora biti tudi prvi prebran podatek. To pa je zaradi ukaza, ki je enak začetni identifikaciji pri branju ukaza. Zato lahko prepozna ukaz kot identifikacijo in vrne napako. S tem je podatek izgubljen. Program bere podatke v vrstnem redu, pri čemer prebere najprej merjene podatke, nato pa še nastavljene podatke, zato je treba najprej poslati merjen čas na napravi in zatem merjeno hitrost na napravi, nato pa v istem zaporedju, vendar ne merjene veličine, ampak nastavljene.

7.3.2. Sprejem podatkov

Sprejemanje podatkov je s pomočjo niti `serial_listner`, ki se izvede ob sprejemu podatka na računalniku. Ob sprejemu podatkov najprej preveri, ali je prejel cel niz ali pa samo del. Ob sprejemu dela podatkov počaka toliko časa, da sprejme cel niz podatkov. Ob sprejemu celotnega niza podatkov preveri, kateri podatek je prejel. Ob sprejemu neznanega podatka, program izpiše napako neznan podatek.

Ob sprejemu prebere podatek in izbriše vse drugo, tako da ostane samo število. Število shrani v pomnilnik, in označi, kateri podatek je prejel. Ob prejemu vseh potrebnih podatkov od centrifuge za zapis v tabelo preveri, s katero napravo je povezan in jih zapiše v tabelo ter označi, da lahko sprejme nov niz podatkov.

Pred zapisom v tabelo pa je treba izrezati vse, razen vrednosti, ki jo je poslala naprava v računalnik. Izrez se stori s ključno besedo, ki je odvisna od podatka in znaka za konec. Podatek pa je med ključno besedo in nizom za konec prejetega niza.

Ob prebranjem podatku je treba urediti število, ki ga je prebral, ker so za lažje pošiljanje podatkov vsi podatki pomnoženi z deset in jim je potrebno zapisati decimalno vejico.

Napaka pri branju pa je možna, ker vsak prejeti znak centrifuga pošlje nazaj. Zahteva za pošiljanje je enaka ukazu za identifikacijo niza. Zato je treba preveriti, ali program bere zahtevo za podatek ali pa ukaz za podatek. Pred branjem vrednosti je treba izbrisati zahtevo, ki jo je posredovala naprava nazaj in šele nato se prebere podatek.

Računanje časa

Za izračun časa je treba odčitati čas prvega prejetega podatka ter čas trenutnega podatka in nato izračunati razliko. Vendar je treba imeti nekaj varoval, da ne pride do napake. Razlika časa ne sme biti ista in tudi ne manjša od prejšnjega prejetega podatka.

7.4. Opis delovanja

7.4.1. Excel

Zapis podatkov v Excel se zgodi s pomočjo knjižnice Java Excel API. Za zapis v Excel mora najprej uporabnik izbrati ime datoteke in lokacijo, kamor bo program shranil podatke iz tabele. Zato se najprej odpre okno „Shrani kot“ in v tem oknu mora uporabnik izbrati lokacijo in ime datoteke, kamor bo aplikacija shranila preglednico. Po izbrani lokaciji je treba ustvariti Excel datoteko – to storimo s pomočjo funkcije iz knjižnice. Nato je treba prepisati vsako

polje posebej v preglednico.

7.4.2. Shranjevanje nastavitev

Shranjevanje nastavitev se zgodi vedno ob zaprtju programa, pri tem se nastavitve zapišejo v datoteko. Pri shranjevanju nastavitev je treba prebrati podatke iz nastavitev ter jih zapisati v datoteko. Pri zapisu podatkov v datoteko je treba zapisati tudi ključno besedo, da se bo ob branju vedelo, kateri podatek je bil zapisan v datoteko. Pri centrifugi se v datoteko zapisuje šest glavnih podatkov, to je dolžina intervala, čas delovanja, hitrost delovanja, centrifugalna sila, nivo pospeševanja in nivo zaviranja. V datoteko je treba zapisati tudi ime naprave, da se ve, od katere naprave so zapisane nastavitve.

7.4.3. Knjižnica Java Excel API

Java Excel API je brezplačna knjižnica za pisanje v format xls [20]. Knjižnica je napisana v javi. Omogoča pa branje in pisanje v preglednico (spreadsheets).

Aplikacija izpiše v preglednico samo osnovne podatke. Podatke prepíše iz tabele, kamor se zapisujejo podatki iz serijske komunikacije. Podatke zapiše v preglednico kot besedilo. Zato je treba podatke pretvoriti v število za računanje.

7.4.4. Odpiranje nastavitev

Branje podatkov se zgodi ob vzpostavitvi povezave s centrifugo. Pri odpiranju podatkov je treba odpreti datoteko in iz nje prebrati podatke. Po branju je treba tudi odčitati nastavitve iz prebranih podatkov. Branje podatkov se zgodi, da se preveri ali je ključna beseda v vrstici. Za branje števila je treba izbrisati ključno besedo iz niza in niz pretvoriti v število. Nato pa je še potrebno zapisati podatek v polje.

7.4.5. Odpiranje pomoči

Odpiranje pomoči je možno s klikom na gumb pomoč, ki je v meniju v levem zgornjem robu. S klikom na gumb „pomoč“ se izvede terminalni ukaz in odpre se datoteka „Instruction.pdf“. Datoteka uporaba programa mora biti na lokaciji, kjer je nameščen program. V primeru, da se ne nahaja na lokaciji, kjer je program, se zgodi napaka.

7.5. Opis vmesnika

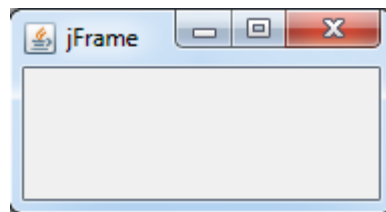
Vmesnik je izdelan grafično v okolju IDE v programu Netbeans 8.02. To okolje je namenjeno programiranje v programskem jeziku Java.

7.5.1. Opis gradnikov

Uporabniški vmesnik je sestavljen iz vnaprej pripravljenih gradnikov. Gradniki so del integriranega razvojnega okolja Netbeans. Okno v programskem jeziku Java je sestavljeno z gradnika JFrame. Ta vsebuje druge gradnike za prikaz in nastavljanje parametrov.

Gradnik JFrame

Jframe je osnovni gradnik grafičnega vmesnika v programskem jeziku Java [22]. Jframe je glavno okno grafičnega vmesnika, ki vsebuje vse grafične gradnike. Jframe ima tudi nekaj pomembnih dogodkov, ki se zgodijo ob zagonu in ob zaključku programa.



Slika 7.1: Prazen jframe.

Na Sliki 7.1 vidimo prazen jframe, v katerega je treba dodati komponente in napisati odziv komponent.

Gradnik Jpanel

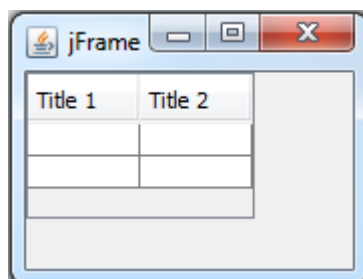


Slika 7.2: JFrame, ki vsebuje jpanel.

Jpanel je gradnik, ki ga vstavimo v jframe in s katerim razporedimo elemente v skupine za boljšo preglednost in ureditev. Nanj se da nastaviti tudi slika, kar je koristno za risanje grafa. Pri tem pa je zelo pomembno osveževanje gradnika jpanel, ker se ob spremembi velikosti okna mora povečati ali zmanjšati tudi velikost jpanela. Na Sliki 7.2 vidimo jframe, ki vsebuje gradnik jpanel. Jpanel je gradnik, ki je obarvan zeleno. Če bi bil enake barve kot jframe, se ga ne bi opazilo.

Gradnik jtable

Jtable razred se upodablja za prikaz podatkov v dveh dimenzijah [14]. Jtable lahko omogoča urejanje podatkov, vendar jtable ne vsebuje podatkov, je le pregledovalnik. Običajno je jtable gradnik znotraj jscrollpane. Jscrollpane se uporablja za ogled podatkov, da uporabnik ni omejen z velikostjo okna. Za vsako tabelo je potreben podatkovni model.

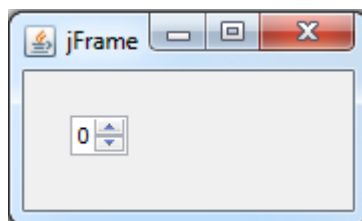


Slika 7.3: JFrame, ki vsebuje jtable

Na Sliki 7.3 vidimo JFrame, ki vsebuje osnovni gradnik JTable. Njegov podatkovni model vsebuje dva stolpca in dve vrstici.

Gradnik JSpinner

JSpinner je gradnik za nastavljanje vrednosti, kjer je možno nastaviti vrednost v obliki navadnega števila, datuma ali časa in modela vrednosti.

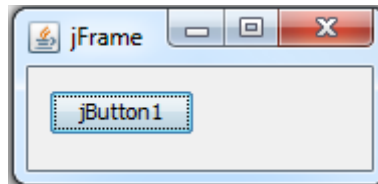


Slika 7.4: JFrame, ki vsebuje gradnik JSpinner.

Na Sliki 7.4 vidimo gradnik JFrame, ki vsebuje gradnik JSpinner, v katerem je možno nastaviti število. Število se lahko vnese s tipkovnico, s klikom na puščici gor in dol pa lahko spremenimo vrednost.

Gradnik Jbutton

Jbutton je gumb, ki se uporablja za nastavitvev.

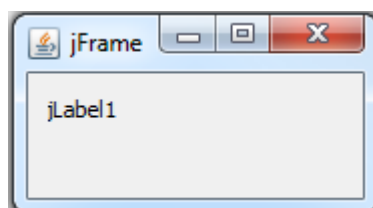


Slika 7.5: JFrame, ki vsebuje jbutton.

Na Sliki 7.5 je JFrame, ki vsebuje gradnik Jbutton. S tem gumbom se potrjuje odpiranje oziroma, kar je napisano v dogodku, ki se izvede v dogodku ob kliku na gumb.

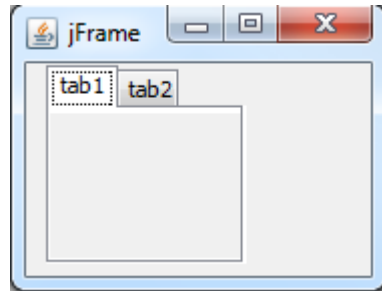
Gradnik JLabel

JLabel je gradnik, ki se uporablja za napis. V JLabel se napiše napis ali pa vstavi slika. Pri JLabel je veliko dogodkov, vendar v programu nisem uporabil nobenega. Na Sliki 7.6 vidimo JFrame, ki vsebuje gradnik JLabel z napisom »JLabel1«.



Slika 7.6: JFrame z gradnikom JLabel

Gradnik Jtabbetpanel

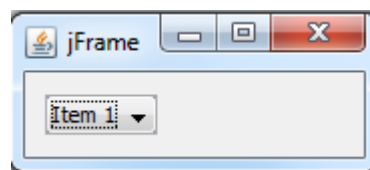


Slika 7.7: JFrame, ki vsebuje gradnik jtabetpanel.

Jtabetpanel se uporablja za bolj pregleden prikaz podatkov, kjer uporabnik izbira, kateri jpanel se bo prikazal ob kliku na zgornji zavitek. Na Sliki 7.7 sta dva zavitka tab1 in tab2, kjer vsak zavitek lahko prikazuje ločen jpanel, vendar lahko uporabnik vidi samo en jpanel naenkrat.

Gradnik JcomboBox

Jcombobox se uporablja za izbiro nastavitve, kjer so nastavitve že vnaprej nastavljene v gradniku. V programu se gradnik jcombobox uporablja za izbiro serijskih vrat in izbira, kateri podatek se bo izrisal na grafu.



Slika 7.8: JFrame, ki vsebuje jcombobox

Gradnik joptionPane

Joptionpane je razred, ki se uporablja za prikaz obvestil uporabniki. Večinoma se ta prikaz

uporablja za obvestila ob napaki in ob shranjevanju podatkov. Program uporablja dve funkciji razreda `JOptionPane`.

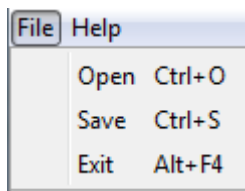
Funkcija `showmessagedialog`

Funkcija `showmessagedialog` pokaže okno z enim gumbom za potrditev in opozorilom. S klikom na gumb se okno zapre.

Funkcija `showoptiondialog`

Funkcija `showoptiondialog` pokaže okno z dvema ali tremi gumbi, pri čemer uporabnik izbere odgovor. Primer uporabe funkcije `showoptiondialog`, je ob brisanju podatkov iz tabele. Takrat uporabnika program vpraša, ali uporabnik želi podatke še shraniti ali ne.

7.5.2. Meni datoteka



Slika 7.9: Meni datoteka

Meni datoteka je v zgornjem levem kotu in s klikom na meni datoteka se odpre meni, v katerem so trije gumbi: odpri, shrani in zapri.

Odpri

S klikom na gumb Odpri, se odpre okno Odpri. Pri odpiranju podatkov bo izbrisalo katere koli druge podatke, ki so zapisani v tabeli. V oknu Odpri izberemo datoteko, kje so shranjeni

podatki od prejšnjih prejemanj podatkov. Izbrati je možno samo tiste datoteke, ki so pravilnega formata. Datoteka napačnega formata se ne prikaže v oknu. Vsaka naprava ima drugačen format. Za izbiro podatkov od centrifuge je potrebno imeti izbrano napravo centrifuga. Po izbiri podatkov je potrebno še klikniti na gumb Odpri.

Shrani

S klikom na gumb Shrani, se odpre okno Shrani, kjer je treba izbrati lokacijo, kamor bo program shranil datoteko. Po izbiri lokacije je treba vnesti še ime datoteke. Po izbiri lokacije in imena datoteke je treba klikniti na gumb Save.

Izhod

S klikom na gumb Izhod zapremo aplikacijo.

Zapis v datoteko

Lokacijo in ime datoteke izbere s pomočjo okna „Shrani kot“. V tem oknu je treba izbrati lokacijo, kamor bo aplikacija shranila datoteko in vnesti ime datoteke, v katero bo aplikacija shranila podatke.

Vrednosti se prepišejo iz tabele, kjer pred vsakim podatkom iz vrstice izpiše ime vrstice za lažjo identifikacijo. To se naredi za celotno tabelo.

Filter

Filter se uporablja za omejitev prikaza datotek. Prikaz se omejuje po tipu datotek. Filter filtrira imena datotek in prikaže samo tiste datoteke, ki imajo pravilno končnico (file extensions). Za vsako napravo je potreben svoj filter, ker je različna končnica pri poimenovanju datoteke, kjer so shranjeni podatki. Končnica datoteke je sestavljena iz prvih štiri črke imena naprave in mora v polju prikazati tip datoteke, ime naprave in končnico.

Branje iz datoteke

Branje iz datoteke se izvede, ko najprej preveri, s katero napravo je aplikacija povezana, nato pa mora uporabnik izbrati datoteko, iz katere želi prebrati podatke. To stori s pomočjo okna Odpri. V oknu Odpri uporabnik izbere datoteko, iz katere bo aplikacija prebrala podatke, ki jih bo zapisala v tabelo iz datoteke. Po izbrani lokaciji je treba prebrati podatke iz datoteke. To se stori s pomočjo ključne besede. Ključna beseda je ime stolpca. V vsaki vrstici se prebere vsak znak posebej in v pravilnem zaporedju znakov se izrežejo naslednji znaki za ključno besedo. To se ponavlja za vse vrstice do konca datoteke.

7.5.3. Cikler

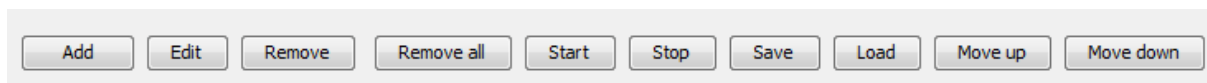
Cikler je funkcija za oddaljeno krmiljenje centrifuge. Pri tem uporabnik nastavi hitrost ali centrifugalno silo, čas delovanja, nivo pospeševanja in nivo zaviranja centrifuge. Nastavitve naprave se zapišejo v tabelo na Sliki 7.10. Nastavitve lahko uporabnik ureja in ponovno razvršča, shrani v datoteko in naloži iz datoteke.

Cycle	Begin	End	Duration	Speed	Select speed	Centrifugal force	Select centrifugal force	Acceleration level	Brake level
1	09:06:43 08.04.2015	09:16:43 08.04.2015	00:10:00	200	<input checked="" type="checkbox"/>	5	<input type="checkbox"/>	0	0
2	09:16:43 08.04.2015	09:26:43 08.04.2015	00:10:00	565	<input type="checkbox"/>	40	<input checked="" type="checkbox"/>	0	0
3	09:26:43 08.04.2015	09:36:43 08.04.2015	00:10:00	200	<input checked="" type="checkbox"/>	5	<input type="checkbox"/>	0	0

Slika 7.10: Tabela z vnesenimi podatki za cikler

Na Sliki 7.10 vidimo tabelo nastavitv ciklerja. Tabela vsebuje 10 stolpcev, kjer je prvi številka nastavitve, naslednje dva pa sta začetek in konec nastavljenega intervala. Ta dva časa se izračunata ob vklopu ciklerja. Četrty stolpec je čas intervala nastavitve. Peti in šesti sta nastavitvi hitrosti, kjer je peti vrednost hitrosti. Šesti stolpec je izbira uporabnika, ali je nastavljen hitrost ali centrifugalno silo. Ker je pri centrifugi možno nastaviti hitrost in centrifugalno silo, je pri ciklerju omogočeno nastaviti samo eno od njiju. Sedmi in osmi stolpec sta za nastavitvi centrifugalne sile, kjer je sedmi stolpec nastavitvi vrednosti

centrifuge in osmi stolpec izbira uporabnika, ali ima nastavljeno hitrost ali centrifugalno silo. Deveti stolpec je za nastavitve nivoja pospeševanja centrifuge. Deseti stolpec je za nastavitve nivoja zaviranja centrifuge.



Slika 7.11: Gumbi za urejanje podatkov za cikler

Cikler vsebuje deset gumbov za nastavitve.

- Gumb Add doda nastavitve ciklerja. Pri tem se odpre posebno okno, kjer uporabnik nastavi: čas intervala, hitrost ali centrifugalno silo, nivo pospeševanja centrifuge in nivo zaviranja centrifuge.
- Gumb Edit je gumb za urejanje izbranega vnosa v tabeli. Če je izbranih več vnosov v tabeli, ob kliku na gumb Edit, program prebere prvega izbranega za urejanje.
- Gumb Remove izbriše prvi izbrani podatek.
- Gumb Remove all izbriše vse podatke v tabeli.
- Gumb Start vklopi cikler in izračuna vse čase, kdaj se morajo nastavitve vklopiti in izklopiti.
- Gumb Stop izklopi cikler in nastavi program na normalno delovanje.
- Gumb Save shrani trenutne nastavitve centrifuge v datoteko, ki jo uporabnik sam izbere.
- Gumb Load odpre shranjene nastavitve, kjer mora uporabnik sam izbrati datoteko.
- Gumb Move up pomakne izbrani podatek za enega navzgor.

- Gumb Move down pomakne izbrani podatek za enega navzdol.

8. Uporabniški vmesnik

Parametri se nastavljaajo v uporabniškem vmesniku, kjer uporabnik nastavi parameter in pošlje napravi s klikom na gumb.

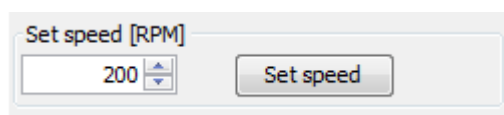
8.1. Nastavitev parametrov Centrifuge

Krmiljenje Centrifuge

Centrifuga se krmili s petimi parametri. Za nastavljanje parametra so potrebni trije gradniki jpanel, jspinner in jbutton. Jpanel se uporablja za pregledno razporeditev nastavitev, ki vsebuje jbutton in jspinner. Jspinner je gradnik za nastavitev vrednosti, ki jo hoče uporabnik nastaviti na napravi. Jbutton se uporablja za potrditev nastavitve, ki jo program pošlje napravi.

Nastavitev hitrosti

Hitrost se nastavlja v obratih na minuto. Vrednost se nastavi v gradniku jspinner. Gradnik jspinner ima dogodek, ki se zgodi ob spremembi nastavitve ali vrednosti. V tem dogodku program preveri vrednost ter jo primerja z maksimalno možno nastavitvijo in minimalno možno nastavitvijo. V primeru, da je vrednost prevelika ali premajhna, prikaže opozorilo in vrednost vrne na prejšnjo. Minimalna in maksimalna vrednost se prebereta iz naprave, takoj ko se vzpostavi povezava z napravo. Minimalna in maksimalna hitrost je zelo pomembna, ker v primeru, da bi bila nastavljena vrednost prevelika ali premajhna, naprava ne nastavi ničesar, ampak pošlje v računalnik, da se je zgodila napaka.

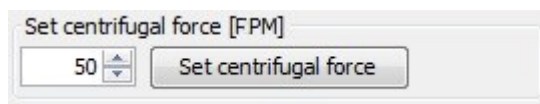


Slika 8.1: Jpanel za nastavitev hitrosti vrtenja centrifuge

Nastavitev centrifugalne sile

Centrifugalna sila je vezana na hitrost, ker je le-ta posledica vrtenja centrifuge. Vendar pri nastavljanju centrifugalne sile računalnik pošlje vrednost centrifugalne sile centrifuge. Centrifugalno silo naprava preračuna v hitrost in nastavi primerno hitrost za centrifugalno silo. Izračun hitrosti iz centrifugalne sile se naredi z enačbo (3), kjer je r radij rotorja. F_{cent} je nastavljena centrifugalna sila.

$$f = \sqrt{\frac{F_{cent}}{r} * 1,118} \quad (3)$$

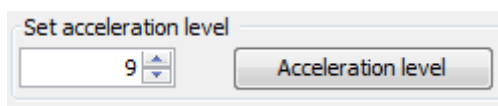


Slika 8.2: Jpanel za nastavitev centrifugalnega pospeška

Nastavitev centrifugalne sile je omejena z ločenimi omejitvami kot nastavitev hitrosti. Centrifugalna sila se ne meri, ampak se izračuna iz hitrosti. Zato se omejitve centrifugalne sile tudi preračunajo iz omejitev hitrosti. Zato mora računalnik prebrati omejitve centrifugalne sile, da ne bo prišlo do napačnih nastavitev centrifugalne sile.

Nastavitev nivoja pospeševanja

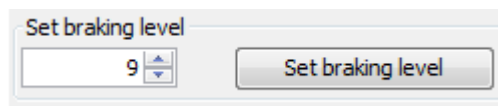
Nivo pospeševanja je parameter, ki nadzoruje strmino pospeševanja. Pospeševanje je realizirano z linearno funkcijo, kjer je večji nivo pospeševanja, naprava hitreje pospešuje do nastavljene hitrosti. Če je nastavljen nižji nivo pospeševanja, centrifuga počasneje pospešuje do nastavljene hitrosti.



Slika 8.3: Jpanel za nastavitev nivoja pospeševanja

Nastavitev nivoja zaviranja

Nivo zaviranja je parameter, ki določa strmino zaviranja centrifuge. Zaviranje je realizirano z linearno funkcijo. Kjer večji nivo zaviranja povzroči, da centrifuga hitreje zavira in manjši nivo zaviranja povzroči počasnejše zaviranje centrifuge.



Slika 8.4: Jpanel za nastavitev nivoja zaviranja

Nastavitev časa delovanja

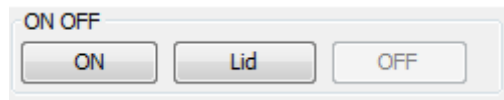
Nastavitev časa delovanja je možna na dva načina. Prvi način je, da uporabnik klikne na gumb Hold in naprava deluje, dokler je uporabnik ne prekine. Drugi način je, da uporabnik nastavi čas v treh jspinnerjih, pri čemer je možno nastaviti dan, ure, minute. S klikom na gumb Set time pa program pošlje izbrano nastavitev centrifugi.



Slika 8.5: Jpanel za nastavitev časa delovanja centrifuge

Vklop in izklop naprave in odpiranje vrat

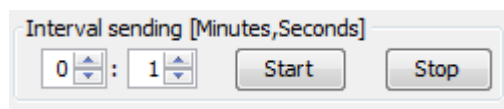
Za te funkcije so potrebni trije gumbi, ki pošljejo ukaz napravi. Pri tem program onemogoči gumb za vklop naprave, ko centrifuga deluje in onemogoči gumb za izklop, ko je centrifuga ugasnjena.



Slika 8.6: Jpanel za nastavitev zagona centrifuge, odprtje vrat in zaustavitev centrifuge.

Nastavitev intervala pošiljanja


Interval pošiljanja je zelo pomemben zaradi izpisa podatkov v tabelo. Možen razpon intervala je od 1 sekunde do 10 minut. Za vklop intervalnega pošiljanja je treba klikniti na gumb Start. Ob kliku na gumb Start program vklopi izpis v tabelo in vklopi prikaz grafov. Ob kliku na gumb Stop, program onemogoči prikaz podatkov v tabeli in izklopi prikaz podatkov.



Slika 8.7: Jpanel za nastavitev intervala branja podatkov.

Izpis podatkov

Izpis podatkov se zgodi ob nastavitvi intervala in deluje, dokler uporabnik ne prekliče pošiljanja podatkov. Izpisuje se ob vsaki sekundi neglede kolikšen interval je nastavljen za izpis podatkov v tabelo. Pri izpisu podatkov je še indikator delovanja naprave. Indikator je realiziran z gradnikom jlabel. Gradnik jlabel vsebuje dve sliki, ki jih menja, odvisno od stanja naprave. Ob mirovanju naprave prikazuje rdečo sliko. Ob delovanju naprave pa prikazuje zeleno sliko.

Time 13:53:44 30.03.2015	Mesured Time 00:12:45	Set time 01:40:00	Measured Centrifugal Force [RCF] 00000	Set Centrifugal Force[RCF] 00006	Mesured speed [RPM] 00000	Set speed [RPM] 00200	
-----------------------------	--------------------------	----------------------	-------------------------------------------	-------------------------------------	------------------------------	--------------------------	-------------------------------------------------------------------------------------

Slika 8.8: Izpis trenutnih podatkov, ki se osvežuje vsako sekundo

Izpis v tabelo

Tabela ima 10 stolpcev, ki so določeni z modelom tabele. Za izpis tabele je treba ustvariti novo vrstico, v kateri so podatki. Nova vrstica se ustvari ob sprejemu novih podatkov iz naprave. Ob tem mora biti izpolnjen pogoj, da je pretekli čas med prejšnjim zapisom in novim zapisom enak dolžini nastavljenega intervala pošiljanja podatkov.

Id	Time	Date	Error	Set time	Current time	Set speed	Measured speed	Set Centrifugal Force	Measured Centrifugal Force	Acceleration	Deceleration
1593	00:26:38 01	13:25:36 02.04.2015	ER00	09:36:27	01:40:00	01099	00000	00200	00000	00000	00009
1594	00:26:39 01	13:25:37 02.04.2015	ER00	09:36:28	01:40:00	01099	00000	00200	00000	00000	00009
1595	00:26:40 01	13:25:38 02.04.2015	ER00	09:36:29	01:40:00	01099	00000	00200	00000	00000	00009
1596	00:26:41 01	13:25:39 02.04.2015	ER00	09:36:30	01:40:00	01099	00000	00200	00000	00000	00009
1597	00:26:42 01	13:25:40 02.04.2015	ER00	09:36:31	01:40:00	01099	00000	00200	00000	00000	00009

Slika 8.9: Izpis podatkov v tabeli.

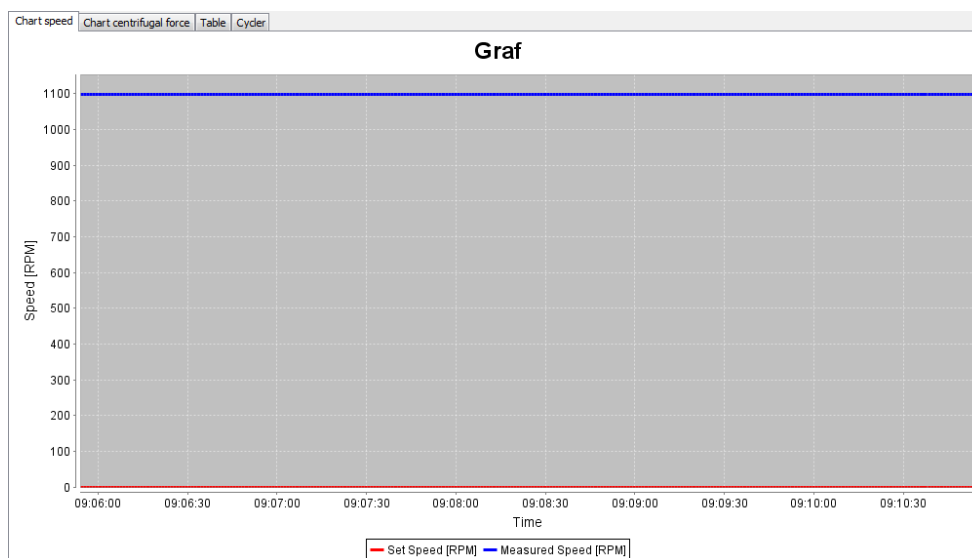
Izpis na graf

Na grafu se izpisujejo podatki v realnem času. Dolžina intervala, ki ga prikazuje graf, je odvisna od nastavljenega intervala. Ta pa je tristokrat daljša od nastavljenega intervala. To pomeni, če je nastavljen interval 1 sekunde, na grafu prikazuje vrednosti centrifuge zadnjih petih minut.

Za izpis na grafu skrbi posebna nit, ki osvežuje podatke vsako sekundo. Nit prebere trenutne podatke in jih izriše na graf.

Graf je sestavljen iz gradnika jpanel, ki ima nastavljeno za ozadje sliko. Graf program izriše s pomočjo knjižnice jfreechart. Knjižnica ustvari sliko grafa, ki jo pripne na jpanel. Jpanel potrebuje še dogodek componentchange, ki prebere dimenzijo gradnika jpanela in priredi velikost grafa. To je nujno, če ne, je graf vedno iste velikosti.

Podatki na grafu



Slika 8.10: Graf hitrosti za prikazovanje hitrosti centrifuge v realnem času.

Za podatke na grafu je potrebno najprej prebrati vrednosti gradnikov jspinner začetek in jspinner konec, v katerih je zapisan prvi podatek in zadnji podatek na grafu. Ker pa je v njih zapisan čas, je treba prebrati celotno tabelo in ugotoviti, kateri podatek je vnesen v gradnik jspinner. Po prebranem začetku in koncu je treba za risanje grafa nastaviti podatkovno zbirko. Za X os je potrebno nastaviti časovno serijo podatkov, ker je treba imeti na X osi čas. To storimo s funkcijo `TimeSeries`. V to zbirko podatkov prepisemo podatke s tabele in s pomočjo funkcije `Second` v to funkcijo vnesemo čas. Čas vnesemo po ključu sekunde, minute, ure, dneva v mesecu, mesec, leto. Za dan, mesec in leto je treba vzeti privzete vrednosti.

8.2. Graf

Risanje grafa

Graf se nariše s pomočjo knjižnice Jfreechart, ki je knjižnica za risanje grafov, kjer je možno narisati razne vrste grafov. Ker tukaj potrebujem kartezični koordinatni sistem brez barvil in samo krivuljo, ki bo pokazala prikaz podatkov, je bila izbira „xychart“. Xychart je preprost graf, kjer na os X postavimo čas in na os Y postavimo temperaturo.

Okno grafa

Okno grafa je razdeljeno na dva dela. V zgornjem delu je graf, v spodnjem delu pa sta dve polji, v katerih so vrednosti, ki se uporabniku odčitajo iz grafa. Ob kliku na graf se izpiše najbližja točka, ki je bila narisana na grafu.

To se stori s pomočjo funkcije, ki prebere točko, kamor je uporabnik kliknil in pretvori v podatek, ki je narisana na grafu, s pomočjo časovnika, ki osvežuje podatke, kakor je uporabnik kliknil vsakih 100 ms.

Označevanje

Označevanje se uporablja za risanje grafa. Če bi pri velikem številu podatkov narisali graf, to lahko upočasni delovanje računalnika. Zato pri risanju grafa nariše samo tiste podatke, ki so označeni v tabeli.

Označevanje lahko spremeni uporabnik sam ali pa spremeni vrednost v gradniku jspinner. Ob označitvi podatkov v tabeli se zgodi dogodek(event), ki zapiše prvi izbran čas v gradniku jspinner začetek in zadnji izbran podatek v gradniku jspinner konec. Ob spremembi podatka v gradniku jspinner z ročnim vnosom ali pa s klikom na stranski gumb se zgodi dogodek, ki prebere celotno tabelo in poišče v tabeli prvi vneseni datum in nato še zadnjega, ter označi

izbrane podatke.

Urejanje podatkov

Urejanje podatkov je sestavljeno iz brisanja tabele, izvoza v različne programe in prikaza podatkov na grafu.

Za brisanje podatkov sta dva načina. Prvi način je izbris samo ene vrstice. Ta način izbriše izbrano vrstico. Če pa je izbranih več vrstic, izbriše prvo izbrano vrstico. Drugi način je izbris vseh vrstic. To pa je zelo občutljivo, zato program uporabnika vpraša, ali želi izbrisati celotno vsebino. Vpraša z razredom `JOptionPane` s funkcijo `showOptionDialog`. Ta vpraša uporabnika, ali želi podatke shraniti, ne shraniti, ali želi preklicati dejanje.

Za prikaz podatkov na grafu je najprej treba izbrati podatke in nato klikniti na gumb `Chart`. Izbor podatkov je možen na tri načine. Prvi način je izbor z gumbom `Select all`. S klikom na gumb program izbere vse podatke v tabeli. Drugi način je z dvema gradnikoma `JSpinner`, katerih model so vsi podatki v tabeli. V gradnik `JSpinner` je možen vnos s tipkovnico ali s puščicami gor in dol. Tretji način je klasičen, kjer uporabnik z miško ali tipkovnico izbere podatke.

Izvoz podatkov je mogoč v dveh formatih. Za izvoz v format `pdf` program uporablja knjižnico `Itext`. S to knjižnico program ustvari dokument `pdf`, kjer so podatki v tabeli.

Za izvoz podatkov v program `Excel` skrbi knjižnica `Java Excel API`. S knjižnico program ustvari dokument `xls` na disku. Nato program z ukazom odpre datoteko.

The image shows a software interface with three distinct panels. The top panel, titled 'Delete', contains two buttons: 'Delete row' and 'Delete all'. The middle panel, titled 'Chart', includes a 'Select chart value' dropdown menu currently set to 'Speed'. Below this are 'Begin' and 'End' time selection fields, each with a numeric input and a small up/down arrow icon. To the right of these fields are 'Select all' and 'Chart' buttons. The bottom panel, titled 'Export', contains two buttons: 'Export in pdf' and 'Export in Excel'.

Slika 8.11: Trije jpaneli za urejanje podatkov

Na Sliki 8.11 so trije jpaneli. Zgornji vsebuje dva gumba za izbris podatkov. Levi izbriše samo en podatek, desni pa celotno tabelo. Srednji panel vsebuje orodje za izris grafa iz izbranih podatkov, pri čemer je treba izbrati veličino, ki jo bo program izrisal na grafu. V spodnjem panelu sta dva gumba za izvoz podatkov v Excel in v format pdf.

9. Sklepne ugotovitve

Pri projektu sem uporabil znanje, ki sem ga pridobil na Fakulteti za elektrotehniko in s pomočjo praktičnih izkušenj uspel razviti projekt in usposobiti komunikacijo za pravilno delovanje.

Projekt je bil kompleksen, zato je bilo treba napisati veliko programske kode. Pri tem je bilo veliko problemov za pravilno delovanje komunikacije, ker pri nezanesljivem delovanju le-te pride do velikih napak in s tem do izgube podatkov. Pri tem je bilo treba izbrati pravilne knjižnice za izpis v preglednico in pdf. To je bil dolgotrajen proces, ker je bilo treba poiskati brezplačno knjižnico in preveriti, ali knjižnica podpira vse zahtevane funkcije. Za izpis podatkov na grafu je bilo tudi potrebno uporabiti knjižnico, ker je risanje grafa kompleksna zadeva.

Programiranje mikrokrmilnika je bilo uspešno, vendar sem imel največ težav z naslavljanjem pravilnega modela naslavljanja naslovnega prostora. Pri tem je bilo potrebno spremeniti naslovni prostor iz 16-bitnega naslavljanja v 24-bitno naslavljanje.

Vmesno vezje je bilo preprosto, zato pri njem nisem imel veliko težav. Vendar je treba gledati v prihodnost in naslednja verzija bo z tehnologijo Bluetooth, ki je brezžična in s tem bolj preprosta za uporabo.

S tem lahko zaključim, da je projekt uspel, vendar ga bo treba nadgraditi, ker se komunikacijska tehnologija zelo hitro razvija.

10. Literatura

- [1] Uporabniški vodnik za komunikacijo z čipom FT232 z vmesnikom D2XX
http://www.ftdichip.com/Support/Documents/ProgramGuides/D2XX_Programmer%27s_Guide%28FT_000071%29.pdf
[Dostopno 18.3.2015]
- [2] Datasheet za FT232RL
http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT232R.pdf
[Dostopno 19.3.2015]
- [3] Data sheet za STM8S207C6
<http://www.st.com/st-web-ui/static/active/en/resource/technical/document/datasheet/CD00197787.pdf>
[Dostopno 19.3.2015]
- [4] STM studio
<http://www.st.com/web/en/catalog/tools/PF251373>
[Dostopno 25.3.2015]
- [5] Programator RLINK
<http://www.wvshare.com/product/STX-RLINK-STM32.htm>
[Dostopno 25.3.2015]
- [6] Opis programatorja Rlink
http://www.raisonance.com/RLink-STD__product__ride7.html
[Dostopno 25.3.2015]
- [7] Java native interface
http://en.wikipedia.org/wiki/Java_Native_Interface
[Dostopno 27.3.2015]
- [8] Java programski jezik
http://en.wikipedia.org/wiki/Java_%28programming_language%29
[Dostopno 27.3.2015]
- [9] Dokumentacija Javah
<http://docs.oracle.com/javase/8/docs/technotes/tools/windows/javah.html>
[Dostopno 30.3.2015]

- [10] Prevajalnik Cosmic
<http://www.cosmic-software.com/stm8.php>
[Dostopno 30.3.2015]
- [11] Gonilnik VCP
<http://www.ftdichip.com/Drivers/VCP.htm>
[Dostopno 30.3.2015]
- [12] Specifikacije USB
http://www.usb.org/developers/docs/usb20_docs/usb_20_031815.zip
[Dostopno 1.4.2015]
- [13] Protokol USB wikipedia
<http://en.wikipedia.org/wiki/USB>
[Dostopno 1.4.2015]
- [14] Jtable help
<http://docs.oracle.com/javase/7/docs/api/javax/swing/JTable.html>
[Dostopno 1.4.2015]
- [15] Opis protokola USB
<http://www.geoffknagge.com/uni/elec101/essay.shtml>
[Dostopno 3.4.2015]
- [16] USB protokol opis
<http://www.beyondlogic.org/usbnutshell/usb3.shtml>
[Dostopno 3.4.2015]
- [17] Opis funkcije sprintf
<http://www.cplusplus.com/reference/cstdio/sprintf/>
[Dostopno 1.5.2015]
- [18] Laboratorijska centrifuga
http://en.wikipedia.org/wiki/Laboratory_centrifuge
[Dostopno 1.5.2015]
- [19] Spletna stran knjižnica jFreechart
<http://www.jfree.org/jfreechart/>
[Dostopno 1.5.2015]

- [20] Spletna stran knjižnica java api Excel
<http://jexcelapi.sourceforge.net/>
[Dostopno 1.5.2015]
- [21] Standardna knjižnica za mikrokontroler STM8
<http://www.st.com/web/en/catalog/tools/PF258009#>
[Dostopno 1.5.2015]
- [22] Knjiga z osnovami programskega jezika java
Java™ All-in-One Desk Reference For Dummies®