



Univerza v Ljubljani
Fakulteta za elektrotehniko



Elektronska vezja

Projektna naloga

(DAQ kartica za LabView)

MENTOR:

doc. dr. Marko Jankovec

AVTOR:

Urban Gostenčnik

vpisna št.:64040052

Kotlje, 1. 2. 2012

Elektronska vezja.....	1
1. Uvod	3
2. Specifikacije	4
3. Časovni in finančni plan	5
4. Zasnova naprave	6
4.1 Opis delovanja DAQ kartice.....	6
4.2 Komunikacija.....	7
5. Načrti	8
5.1 Bločna shema.....	8
5.2 Vezja.....	8
5.2.1 Mikrokrmilnik NXP LPC2138 in PoUSB12	8
5.4 Program	10
5.4.1 Program za mikrokrmilnik.....	10
5.4.2 Gonilnik.....	13
6. Kosovnica	16
7. Priložene datoteke.....	17

1. Uvod

Zajem podatkov (ang. »*data acquisition*«) je proces vzorčenja signalov, ki so rezultat merjenja realnih fizikalnih veličin, pretvorba vrednosti vzorčenih signalov v digitalne numerične vrednosti s katerimi je možna kasnejša manipulacija z računalnikom. Aplikacije za zajem podatkov so navadno nadzorovane s pomočjo programske opreme, kot je *Visual C++*, *Visual Basic*, *Matlab*, *LabVIEW* in s podobnimi. (Vir: http://en.wikipedia.org/wiki/Data_acquisition)

V tej projektni nalogi je opisano in predstavljeno vezje za zajem analognih veličin (DAQ kartica). Neposredno lahko merimo električno napetost, posredno prek različnih senzorjev pa merimo še druge fizikalne veličine, kot so temperatura, vlaga, zračni tlak, frekvenca in podobno. Pogoj za merjenje slednjih veličin je, da s senzorji in prilagodilnimi vezji pretvorimo fizikalne parametre v električne signale primernih vrednosti. DAQ kartica temelji na mikrokontrolerju NXP LPC2138, ki s svojim vgrajenim analogno digitalnim pretvornikom skrbi za zajem analognih vrednosti, ki jih s procesira in preko PoUSB12 (RS-232 ↔ USB) pretvornika pošlje računalniku v nadaljnjo obravnavo. Za sprejem in prikaz analognih veličin na računalniku sem napisal gonilnik za programski paket *LabVIEW*.

2. Specifikacije

Število analognih/digitalnih kanalov:	16/16
Tip ADC pretvornika:	SAR
Resolucija ADC pretvornika:	10 bit
Maksimalna frekvenca vzorčenja:	500S/s
Merilno območje vhodnih signalov:	0V do 3.3V
Vhodni sklop:	DC
Maksimalna vhodna napetost:	Merilno območje +10%
Prenos podatkov:	USB
Absolutna napaka:	$E_T = \pm 4 \text{ LSB}$
Napaka ojačenja (gain error):	$E_G = \pm 0.5 \text{ LSB}$
Napaka odmika (offset error):	$E_O = \pm 3 \text{ LSB}$

Pogoji: $V_{DDA} = 2.5\text{V}$ do 3.6V ; $V_{SSA} = 0\text{V}$; $T_{amb} = -40^\circ\text{C}$ do $+85^\circ\text{C}$; frekvenca ADC 4.5MHz .

3. Časovni in finančni plan

Eden izmed ciljev tega projekta je, da bi bil končni izdelek najcenejši na trgu in primeren za vsak žep. Zato sem postavil zgornjo mejo, katero še lahko odštejemo za material 50€. Največji del dovoljenega zneska predstavlja mikrokrmilnik (16€) in PoUSB12 (15€). Nekaj stroškov odpade še na ohišje, izdelavo tiskanine in ostali potrošni material, vendar ne več kot 15€. Seveda se cene posameznih komponent ob serijski proizvodnji izdelka v večjih količinah (100+) razpolovijo ali celo razčetrirajo. Zato ocenjujem, da projekt ustreza vsem finančnim ciljem in kriterijem.

Pri razvoju elektronskih naprav, s katerimi še nimamo izkušenj, je zelo težko oceniti, koliko delovnih ur bo potrebnih za razvoj izdelka. Stvari, ki bodo šle narobe je nemogoče napovedati, ker če bi že v startu vedeli, kje nas čakajo težave, bi se jim že vnaprej izognili. Za lažjo oceno potrebnega časa, sem celoten projekt razdelil na več sklopov:

- okvirna zasnova projekta,
- raziskava trga in konkurence,
- iskanje tehnoloških rešitev,
- sestava vezja in ohišja,
- programiranje,
- testiranje izdelka in
- dokumentacija.

Na tej podlagi sem ocenil, da je za izvedbo od začetka do konca projekta potreben mesec dni časa. Žal se je kasneje izkazalo, da sem za projekt potreboval enkrat več od prvotno predvidenega časa.

4. Zasnova naprave

DAQ kartica je sama po sebi neuporaben kos elektronike, če na drugi strani ni osebnega računalnika, ki bi znal raztolmačiti rezultate meritev in jih na jasn način predstaviti uporabniku. Za pravilno delovanje sem napisal gonilnik s programskim orodjem *LabVIEW*, ki skrbi za komunikacijo in na enostaven način prikaže rezultate meritev z uporabniškim vmesnikom.

4.1 Opis delovanja DAQ kartice

Princip, delovanja DAQ kartice je precej enostaven. Mikrokrmilnik z že vgrajenim 16 kanalnim, 10 bitnim AD pretvornikom najprej čaka na ukaz od osebnega računalnika za začetek meritve. Tako moramo preko uporabniškega vmesnika v programskem paketu *LabVIEW* najprej podati zahtevo po začetku izvajanja meritev, hkrati pa mu moramo posredovati število kanalov, ki naj bodo aktivni. Po začetni inicializaciji mikrokrmilnik izmeri analogno vrednost napetosti na prvem kanalu AD pretvornika in jo pretvori v digitalno numerično vrednost, ki je veliko bolj prijazna za naknadno obdelavo in pošiljanje osebnemu računalniku. Ko imamo 10 bitni podatek o izmerjeni vrednosti, moramo le tega posredovati nazaj do računalnika. Da ne bi prihajalo do zmede, katera meritev pripada kateremu kanalu je treba informacijo o številki kanala pripeti samemu rezultatu meritve. Na razpolago imamo 16 kanalov, zato lahko informacijo o kanalu podamo s štirimi biti. Temu sledi pošiljanje podatkov do osebnega računalnika in meritve se tako ciklično ponavljajo.

1010XX01 10101010
ŠT. KANALA REZULTAT

Slika 1: Informacija za pošiljanje

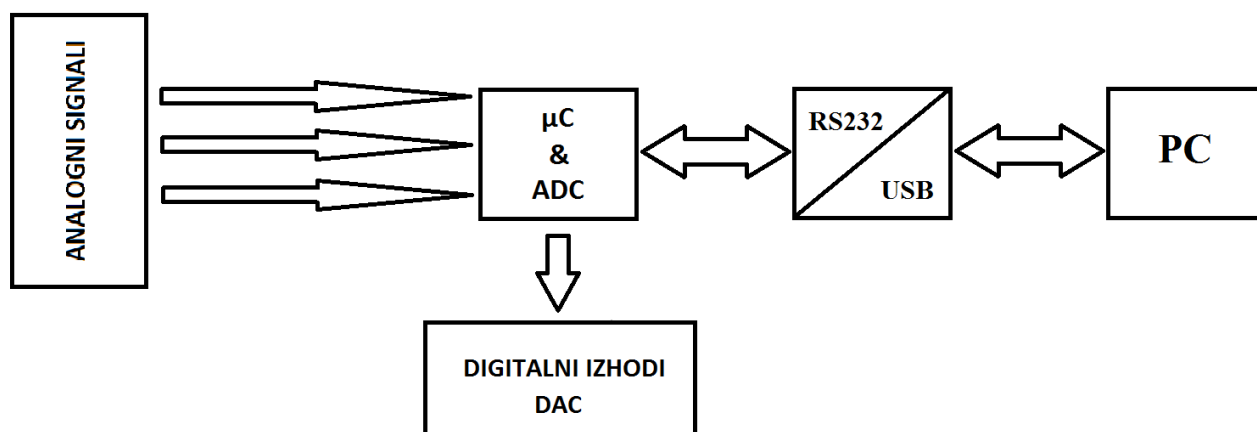
4.2 Komunikacija

Nujen pogoj za pravilno delovanje naprave je, da mikrokrmilnik in osebni računalnik med seboj komunicirata in da pri tem ne prihaja do napak. V našem primeru vsa komunikacija poteka na nivoju RS-232 protokola. Ta protokol sem izbral predvsem zaradi njegove enostavnosti in ker hitrosti prenosa podatkov povsem zadoščajo našim potrebam. V sodobnem računalniku pa ga zaradi svoje univerzalnosti in hitrosti prenosa močno izpodriva USB. Le redko še najdemo na policah trgovin stacionarne in prenosne računalnike, ki bi še podpirali ta protokol, zato je prehod na USB nujen. Za ta namen sem uporabil PoUSB12 vmesnik, ki poskrbi za ta prehod. Vmesnik temelji na čipu CP2102 od proizvajalca Silicon Labs, tukaj pa je nekaj njegovih lastnosti:

- Združljivost: USB 2.0
- Hitrost prenosa podatkov: od 300bps do 1Mbps
- UART podpora 5 do 8 podatkovnih bitov, 1 do 2 stop bita, pariteta soda/liha ali brez.
- Podpora za operacijske sisteme Windows (2000 in višje), MAC (OSX-9 in višje), Linux (2.4 in višje)
- Temperaturno območje delovanja: od -40 do +85 °C

5. Načrti

5.1 Bločna shema



Slika 2: Bločna shema DAQ pretvornika

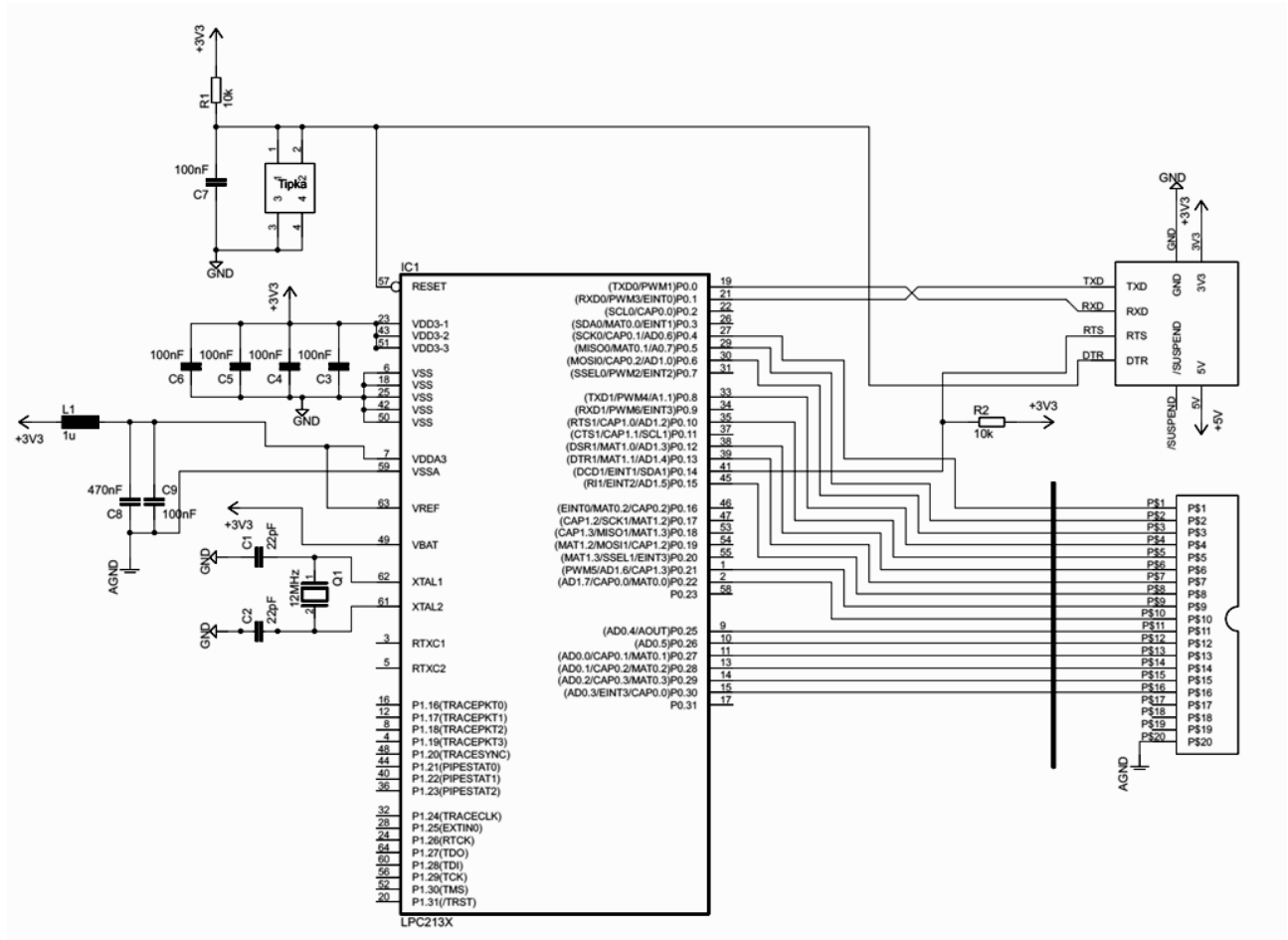
5.2 Vezja

5.2.1 Mikrokrmilnik NXP LPC2138 in PoUSB12

Srce vezja DAQ kartice predstavlja NXP-jev 32-bitni mikrokrmilnik LPC2138 z vgrajenim 16 kanalnim, 10-bitnim AD pretvornikom. Samo vezje je precej enostavno, saj vsebuje osnovne elemente, ki so potrebni za delovanje mikrokrmilnika:

- 12MHz kvarčni oscilator,
- reset tipka,
- napajanje,
- blokirni kondenzatorji,
- referenčna napetost za AD pretvornik.

Poleg mikrokrmilnika imamo še en zelo pomemben element in sicer pretvornik med komunikacijskima protokoloma USB in RS232, to je PoUSB12. Le ta pa ne skrbi samo za komunikacijo med osebnim računalnikom in mikrokrmilnikom ampak tudi zagotavlja stabilizirano 3.3V napajalno napetost. PoUSB12 ima osem pinov. Trije so napetostni izhodi (+5V, +3.3V in GND), štirje pa so komunikacijski (Tx, Rx, RTS, CTS). Pin /SUSPEND v našem primeru ni povezan nikamor. Pina Tx in Rx iz PoUSB12 vežemo križno na Tx in Rx na mikrokrmilniku (UART 0) in služita za dvosmerno komunikacijo. Pina RTS in DTR pa koristno uporabimo pri postopku programiranja mikrokrmilnika. Iz konektorja peljemo analogne kanale od 1 do 16 direktno na AD pretvornik.



Slika 3: Mikrokrmilnik NXP LPC2138 in PoUSB12

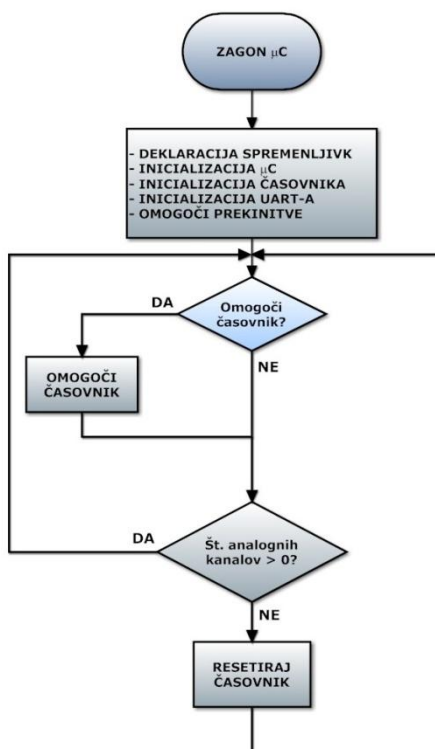
5.4 Program

5.4.1 Program za mikrokrmilnik

Program za mikrokrmilnik sem napisal v programskem jeziku C#, s pomočjo orodja winIDEA 2010. Delovanje in strukturo programa najlažje opišemo s pomočjo diagramov poteka, kateri so prikazani na slikah 4,5 in 6.

- **Zagon mikrokrmilnika:**

Ob priključitvi mikrokrmilnika na napajanje se najprej izvede majhen del kode imenovan »bootloader«, ki poskrbi za nastavitve najosnovnejših parametrov, ki so potrebni za delovanje mikrokrmilnika. Temu opravilu sledi deklaracija vseh globalnih spremenljivk in konstant, ter inicializacija pri čemer določimo frekvenco delovanja procesorja, frekvenco perifernega vodila, smer I/O pinov ipd. Za tem določimo še prekinitvene rutine, ter inicializiramo serijski port uart0 in časovnik timer0. Ob vstopu v neskončno zanko se najprej vprašamo, ali so izpolnjeni pogoji za omogočanje časovnika. Ob pritrdilnem odgovoru, le tega omogočimo z nastavitvijo ustreznih vrednosti v registru (T0TCR – timer0 ang. »*Timer Control Register*«). Na to se vprašamo še po številu aktivnih analognih kanalov in če je aktiven vsaj eden ali več analognih kanalov, ne storimo ničesar. V nasprotnem primeru pa resetiramo časovnik, da nam le ta ne proži prekinitvev. Ob prvem zagonu mikrokrmilnika ni izpolnjen nobeden od teh dveh pogojev in čakamo, da se zgodi prekinitvev iz serijskega porta uart0.



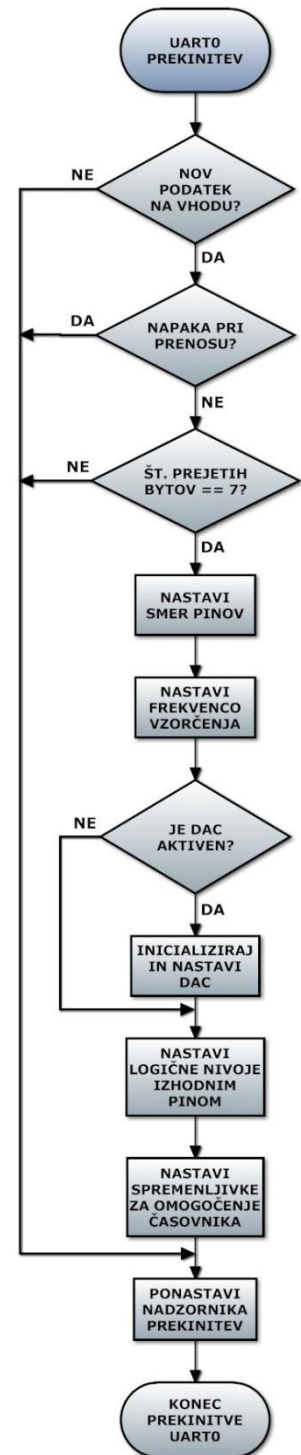
Slika 4: Zagon mikrokrmilnika

- **UART0 prekinitev:**

Ob zahtevi za prekinitev iz strani serijskega porta uart0, se prekine izvajanje neskončne zanke in se začne izvajati funkcija, ki je določena za to prekinitev. Najprej pogledamo, če je zahteva po prekinitvi sprožena zaradi novega podatka na vhodu in ali je prišlo do napake pri prenosu. V primeru, da smo prejeli podatek brez napak, si le tega shranimo v pomnilnik, dokler ne prejmemo vnaprej dogovorjenih sedem bytov. Sedaj smo prejeli dovolj informacij za nastavitev smeri I/O pinom, frekvence vzorčenja, digitalno analognega pretvornika DAC in logičnih nivojev na digitalnih izhodnih pinih. Na koncu še umaknemo zahtevo po prekinitvi z vpisom poljubne vrednosti v njegov register. Prekinitev se zaključi z vrnitvijo v neskončno zanko, katero smo spoznali pri zagonu mikrokrmilnika.

UART (ang. »*Universal Asynchronous Receiver/Transmitter*«; sl. univerzalni asinhroni sprejemnik/oddajnik) je del strojne opreme mikrokrmilnika, ki skrbi za pretvarjanje med paralelnimi in serijskimi oblikami podatkov. UART se običajno uporablja v povezavi z enim od komunikacijskih standardov, kot so RS-232, RS-422 ali RS-485. V našem primeru se za prenos podatkov poslužujemo standarda RS-232.

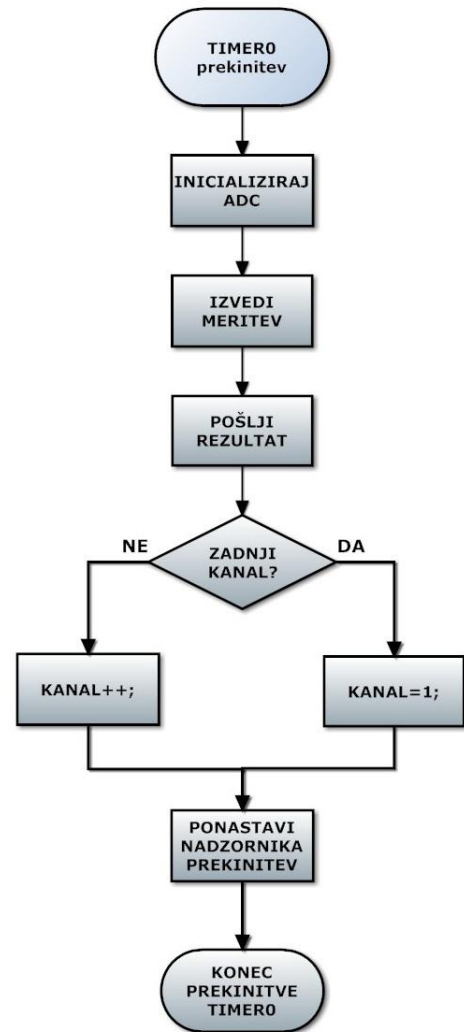
(Vir:http://en.wikipedia.org/wiki/Universal_asynchronous_receiver_transmitter)



Slika 5: UART0 prekinitev

- **TIMER0 prekinitev:**

To prekinitev proži časovnik timer0 v natančno določenih časovnih intervalih. Prekinitev se sproži takrat, ko števec timer0 prešteje do vrednosti, ki je zapisana v primerjalnem registru TOMR0. Vrednost TOMR0 je določena s kvocientom frekvence urinega signala procesnega jedra in želeno frekvenco vzorčenja. Ob vstopu v prekinitveno rutino najprej inicializiramo analogni digitalni pretvornik (ADC) na kanal, ki ga želimo izmeriti. Temu sledi zahteva za pretvorbo analogne vrednosti v digitalno. Ko je le ta zaključena, rezultatu pretvorbe pripnemo še informacijo o merjenemu kanalu in oboje skupaj pošljemo preko serijskega porta uart0. V kolikor je številka izmerjenega kanala enaka številu vseh aktivnih analognih kanalov, izvedemo naslednjo meritev na prvem kanalu. V nasprotnem primeru pa številko merjenega kanala povečamo za ena. Nato le še ponastavimo nadzornika prekinitev z vpisom poljubne vrednosti v njegov register. Prekinitev se zaključi z vrnitvijo v neskončno zanko. Ob programiranju te prekinitvene rutine sem težil k temu, da bi bila koda čim krajša, saj se mora končati pred začetkom nove prekinitve.

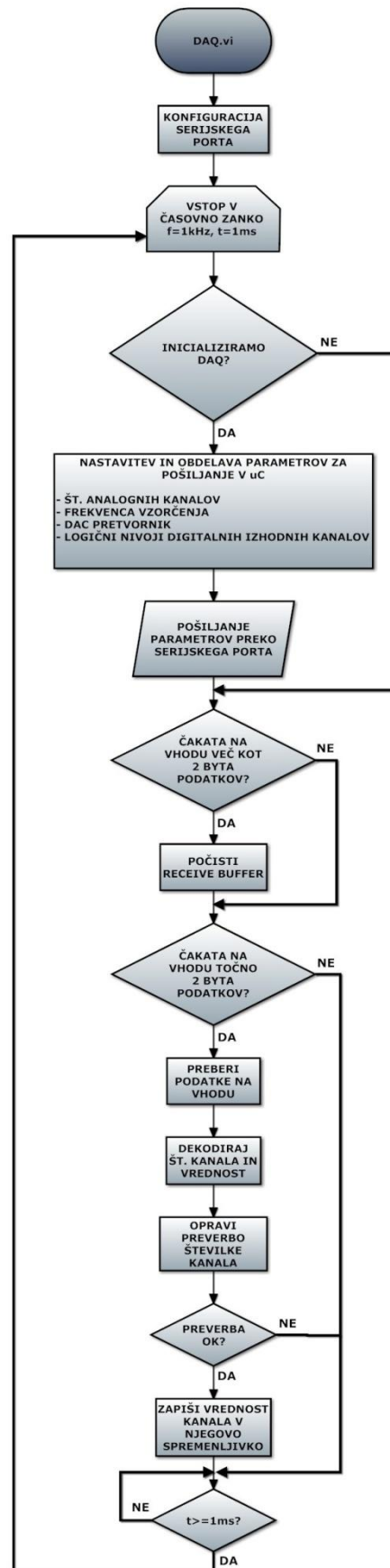


Slika 6: TIMER0 prekinitev

5.4.2 Gonilnik

Gonilnik za DAQ kartico sem oblikoval v grafičnem programskem okolju LabVIEW (kratica za Laboratory Virtual Instrumentation Engineering Workbench). Grafično programiranje je metoda, ki za zapis programa uporablja grafične objekte namesto tekstovnega opisa. Te grafične objekte v skladu s programskimi pravili nato ustrezno medsebojno spojimo. Slika grafičnega programa je podobna zgradbi diagrama poteka, s katerim v tekstovnih programskih jeziki opisujemo izvajanje programa. Program deluje v dveh oknih: okno čelne plošče in okno blok diagrama. Zaradi lažje predstavitve nepoznavalcem grafičnega programiranja pa bom okno blok diagrama predstavil s pomočjo klasičnega diagrama poteka. (Vir: Žarko Gorup: Uvod v LabView)

Okno blok diagrama: Ob zagonu programa najprej konfiguriramo serijski port tako, da mu določimo parametre kot so številka porta, hitrost prenosa, število podatkovnih bitov, kontrola paritete in število stop bitov. Parametri morajo biti enaki tistim, ki so bili nastavljeni pri inicializaciji uarta. Temu sledi vstop v časovno zanko. Časovna zanka je nadgradnja While zanke s časovnimi funkcijami in se ponavlja z nastavljivo periodo, ki znaša v našem primeru dvakratnik maksimalne frekvence vzorčenja in sicer $f = 1\text{kHz}$. Z tem dosežemo enak učinek, kot pri prekinitvi časovnika timer0 v mikrokrmilniku. Ob vstopu v zanko se najprej vprašamo ali želimo inicializirati DAQ kartico. Če je pogoj izpolnjen zberemo parametre (število analognih kanalov, frekvenca vzorčenja, nastavev DAC

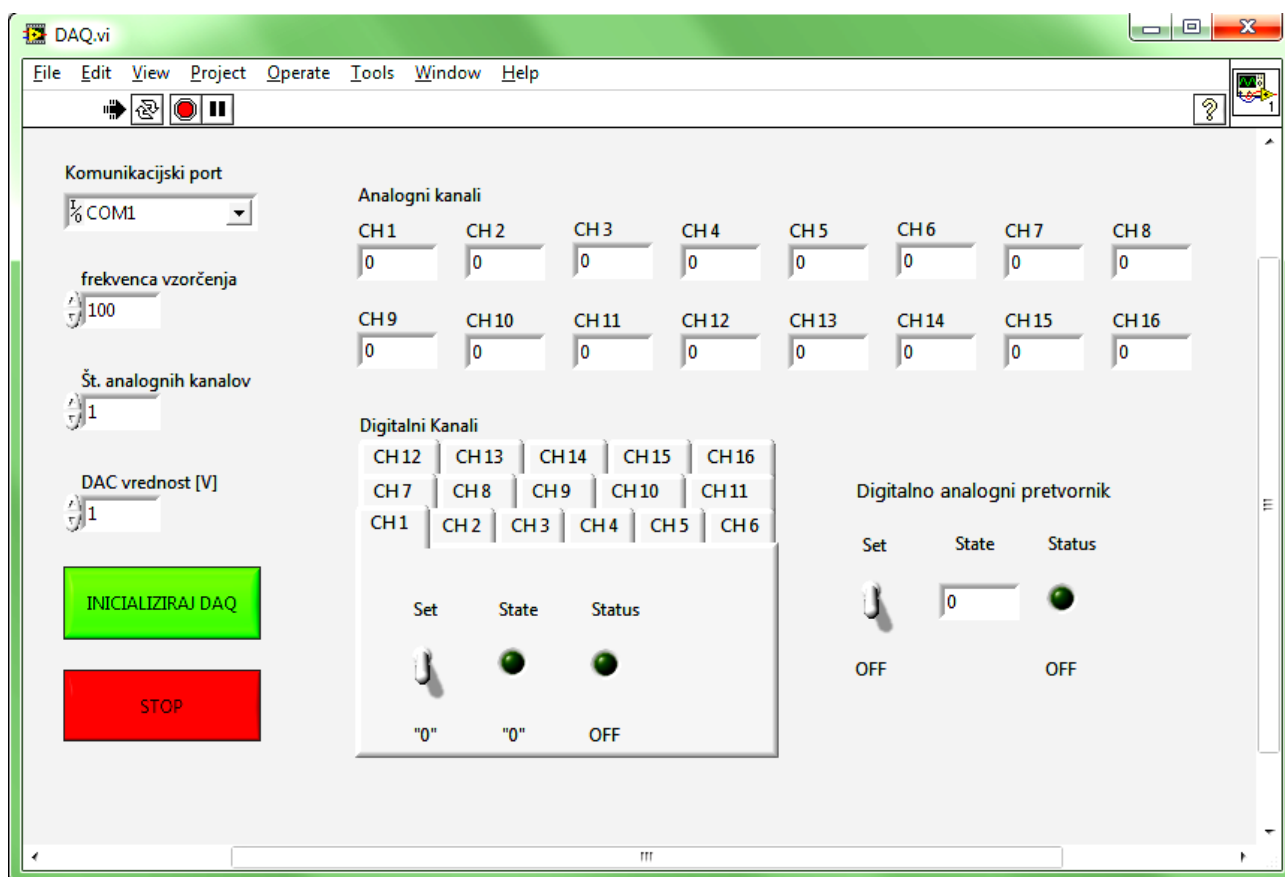


Slika 7: DAQ.vi

pretvornika in logični nivoji digitalnih izhodnih kanalov) iz okna čelne plošče in jih zložimo v niz sedmih bytov.

Nato te pošljemo preko serijskega porta v DAQ kartico. V primeru, da inicializacije DAQ nismo želeli, ta korak preskočimo. Za tem gledamo ali nas že čaka kakšen podatek s strani DAQ. V primeru, da sta na vhodu več kot dva byta podatkov, smo nekje v preteklosti zamudili z izvajanjem časovne zanke, zato zavržemo vse podatke, ki so se v tem času nakopičili. Če pa nas na vhodu čakata točno dva byta podatkov, ju preberemo in dekodiramo na informacijo o prejetem kanalu, ter dejansko izmerjeno vrednost. Za tem opravimo preverbo ali informacija o kanalu pripada naslednjemu pričakovanemu kanalu. V primeru, da je preverba uspešno opravljena zapišemo vrednost kanala v njegovo pripadajoči spremenljivko. Nato še počakamo, da se izteče čas predviden za izvajanje časovne zanke in ob preteku tega časa opravimo novo iteracijo.

- **Okno čelne plošče:**



Slika 8: Okno čelne plošče

Čelna plošča je programsko okno, ki predstavlja vhodno-izhodni vmesnik za uporabnika. Sestavljena je iz kontrol in indikatorjev. Kontrole so gradniki čelne plošče, ki predstavljajo vhode. Vhodi na čelni plošči so elementi, ki omogočajo kakršenkoli vnos podatkov, parametrov ali ukazov, na katere se program odziva. Vhodi na naši čelni plošči so numerični: frekvenca vzorčenja, število analognih kanalov, DAC vrednost, referenčna koda za I/O procese: komunikacijski port, logični: preklopno stikalo Set in gumba INICIALIZIRAJ DAQ, ter STOP. Indikatorji so elementi na čelni plošči, ki služijo za prikaz rezultatov, odzivov ali stanj. Programsko gledano so izhodi, ki uporabniku posredujejo zahtevano informacijo. Izhodi na naši čelni plošči so numerični: Analogni kanali od CH1 do CH16 in logični: led lučki za vrednosti State in Status.

6. Kosovnica

- Mikrokrmilnik NXP LPC2138
- PoUSB12
- Kristalni oscilator 12MHz
- PoTQFP PCB
- Ohišje

7. Priložene datoteke

- Poročilo v PDF formatu
- Program za mikrokrmilnik
- Program za LabVIEW
- Uporabniški priročnik za NXP LPC2138
- Datasheet PoUSB12