

Razvoj komunikacijskega sklada za Nextion zaslone

Mitja Nemeč, Lucija Župevc, Danjel Vončina

Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška 25, 1000 Ljubljana, Slovenija
E-pošta: mitja.nemec@fe.uni-lj.si

Development of communication stack for Nextion displays

Abstract. The article presents how communication stack for Nextion touch screen LCD was developed. The communication stack was developed for communication and control of the power converter. Architecture of the communication stack was kept similar to the previously developed communication stack used for communication with personal computer. Even though the Nextion communication protocol is full of quirks and inconsistencies the stack was successfully developed.

1 Uvod

Na Fakulteti za elektrotehniko v Laboratoriju za regulacijsko tehniko in močnostno elektroniko (LRTME) se primarno ukvarjamo z razvojem močnostne elektronike. V zadnjem desetletju se je večina krmilne elektronike, ki skrbi za regulacijo močnostnega dela, digitalizirala. Tako lahko v glavni nadzorni enoti, ki proži močnostna stikala, po potrebi spreminjamo program, predvsem pa lahko implementiramo napredne in zelo fleksibilne pristope na področju regulacije močnostne elektronike [1], [2].

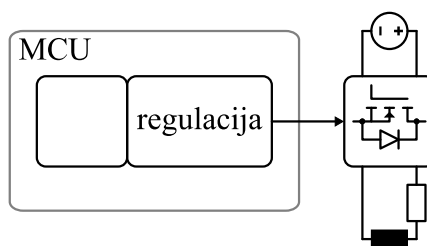
Za potrebe nadzora močnostnih pretvornikov smo v preteklosti razvili platformo, ki nam omogoča nadzor preko osebnega računalnika in jo brez večjih težav prilagajamo in vključujemo v različne projekte. Platforma sestoji iz programskih modulov, ki se vključijo v krmilno-regulacijski program, ki se izvaja v samem pretvorniku in samostojen program ki teče na osebnem računalniku. Največ dela pri razvoju te platforme je bilo namenjenega razvoju stabilnega komunikacijskega sklada [3].

V določenih aplikacijah, pa je nadzor preko osebnega računalnika nekoliko neroden (procesni viri, napajalniki, ...), direktni nadzor preko tipk, stikal in potenciometrov pa ne omogoča nastavljanja vseh parametrov. Za lokalni nadzor naprave, ki bi omogočal nekoliko bolj detajlno upravljanje in zahteva možnost zgoščenega prikaza, kot tudi fleksibilnega upravljanja smo se odločili razviti možnost upravljanja preko zaslona na dotik.

2 Opis tipičnega sistema

V LRTME se pogosto uporablja aplikacija, ki vključuje močnostni pretvornik za regulacijo toka in posredno tudi drugih veličin, kot so navor, hitrost, napetost in moč. Možno je uporabiti en sam pretvornik ali pa jih povezati več, bodisi zaporedno ali vzporedno.

Tako napravo tipično sestavlja močnostna elektronika, ki skrbi za prenos moči ter signalna elektronika s pripadajočim mikrokrmilnikom, ki preko pomerjenih signalov regulira stikala v močnostni elektroniki (slika 1).



Slika 1: Tipična aplikacija na področju močnostne elektronike

Razvoj programske opreme za takšen pretvornik običajno poteka po stopnjah. Začne se z izdelavo prvih verzij programske opreme, kjer se preizkušajo osnovne funkcionalnosti strojne opreme, kot so zajem merjenih veličin dela ali proženje močnostnih stikal. Nato se dodajajo potrebne transformacije merjenih veličin, kot so 3-2 transformacija, $\alpha\beta$ -dq transformacija, in sledi implementacija notranje regulacijske zanke, ki običajno vključuje regulacijo toka. Proti koncu razvoja dodamo tudi možnost normalnega vklopa in izklopa močnostne stopnje, kot tudi možnost izrednega izklopa v primeru zaznanega napačnega delovanja. Nato se po potrebi dodajo nadrejene regulacijske zanke, kot so regulacija napetosti, hitrosti, položaja itd.

Med razvojem lahko za nadzor in spreminjanje parametrov delovanja brez težav uporabljamo osebni računalnik in že razvito aplikacijo. Ko pa je naprava narejena in je regulacija uglašena in preizkušena je pri nekaterih napravah (napajalniki, ...) zaželeno možnost samostojnega delovanja. V kolikor gre za zelo preprosto napravo (e.g. napajalnik) se lahko nadzor vrši preko tipk, potenciometrov in analognih oz digitalnih prikazovalnikov. Če pa gre za napravo, ki je nekoliko bolj kompleksna (napajalnik, ki ima možnost emulacije baterije in/ali PV modula [4]) pa je nadzor take naprave nekoliko bolj kompleksen. Preklop med načini obratovanja in posebni parametri v posameznem načinu obratovanja pa je bistveno lažje nastavljanje preko zaslona na dotik (HMI). Osnovne parametre (tokovna limita, osvetljenost, napetost, ...) pa se lahko še vedno nastavlja preko tipk in/ali potenciometrov. V nekaterih primerih pa je nadzor naprave enostaven, zaslon pa bi potrebovali samo za bolj jasn prikaz delovanja naprave (prikaz izhodne napetosti pri serijskem močnostnem filtru[5]).

2.1 Zahteve

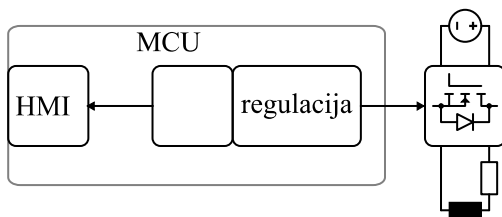
Že pri razvoju nadzorne aplikacije, ki teče na osebnem računalniku smo spoznali bistvene zahteve in omejitve ki v večji meri veljajo tudi pri razvoju nadzora preko zaslona na dotik.

Ker v laboratoriju nimamo veliko kapacitet za razvoj programske opreme mora biti programiranje takega zaslona relativno enostavno.

Ker je vmesnik preko zaslona primarno grafične narave je smiselno če se lahko grafični vmesni načrtuje v grafičnem okolju in ne preko kode.

Ker je družina mikrokontrolerov, ki jo uporabljamo za regulacijo močnostne elektronike zelo ozko usmerjena v svoje področje ni za pričakovati, da bi zaslon lahko nadziral oz. upravljal kar obstoječi mikrokontroler.

Ker bo koda za grafični vmesnik preko zaslona na dotik tekla na dodatnem mikrokontrolerju je zaželeno, če komunikacija med njima poteka po enakem komunikacijskem kanalu kot obstoječa komunikacija z osebnim računalnikom, saj bi v tem primeru lahko pri razvoju uporabili enako oz. podobno arhitekturo in nižjenivojske module. Blokovna shema iskane rešitve je prikazana na sliki 2.



Slika 2: Shema predlagane rešitve

Na srečo je za predlagano arhitekturo na voljo kar nekaj industrijskih rešitev. Daleč najbolj znana rešitev so zasloni podjetja Nextion za katere smo se odločili tudi mi, obstajajo pa še zelo zelo podobne rešitve podjetji:

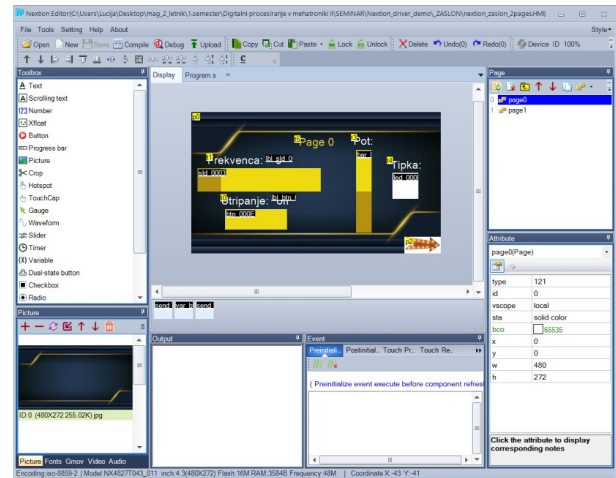
- 4D <https://4dsystems.com.au/>,
- Topway <https://www.topwaydisplay.com/>,
- Proculus <https://proclustech.en.alibaba.com/>

3 Opis sistema

Zasloni podjetja Nextion so postali popularni predvsem zaradi podpore, ki jo imajo v okolju Arduino. Komunikacija z zasloni poteka preko asinhronskega serijskega vodila – UART, ki vsebuje samo dva signala. Prvega za komunikacijo proti zaslonu in drugega v obratni smeri. Proizvajalec nudi zaslone od diagonale 2.4 palca do 10 palcev, pri čemer se jih programira in z njimi komunicira na enak način. Poleg osnovne funkcionalnosti prikaza in upravljanja z gradniki preko dotika določene družine zaslonov nudijo tudi razširjen nabor funkcionalnosti kot so:

- upravljanje z koledarjem in uro (RTC)
- možnost branja in shranjevanja na SD kartico
- možnost generiranja preprostih signalov
- možnost branja/pisanja v FLASH spomin

Grafični vmesnik se načrtuje v grafičnem okolju (slika 3), ki ga zagotavlja proizvajalec. Ko je grafični vmesnik dokončan se ga prenese (naloži) na zaslon preko UART vodila.

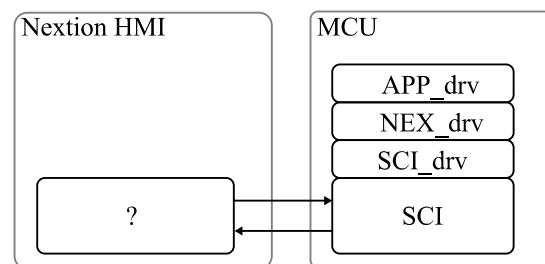


Slika 3: Grafično okolje za programiranje zaslona

4 Razvoj komunikacijskega sklada

Največji del aktivnosti je moral biti namenjen razvoju komunikacijskega sklada na mikrokontrolerju, ki sicer izvaja regulacijo. Ker za komunikacijo z zaslonom uporabljamo enako vodilo (UART) kot za komunikacijo z osebnim računalnikom smo grobo arhitekturo komunikacijskega sklada povzeli že narejenem komunikacijskem skladu [3].

Sama arhitektura programske opreme sestoji iz treh slojev (slika 4). Na strani mikrokontrolerja za generiranje in osnovno obdelavo UART signalov skrbi SCI (Serial Communication Interface) periferna enota. Z SCI enoto upravlja *SCI_drv* gonilnik, ki skrbi da se SCI enoti ustrezno posreduje paket, ki ga želimo poslati in javi, ko je prispel paket. Vmesni sloj *NEX_drv* skrbi za pravilno sestavljanje paketov, ki jih želimo poslati ter tudi tolmači prejete pakete. Najvišji sloj *APP_drv* pa implementira funkcionalnosti na nivoju naprava (sprememba amplitude, vklop/izklop, sporočanje stanja, ...)



Slika 4: Groba arhitektura komunikacijskega sklada

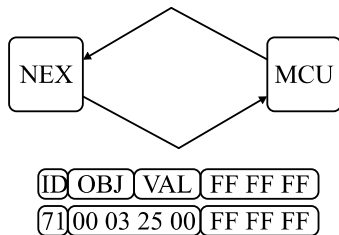
Pri razvoju komunikacijskega smo naleteli na več ovir. Prva in nezanemarljiva ovira je pomanjkljiva dokumentacija ko obsega samo in samo eno spletno stran [6]. Na spletu sicer lahko najdemo kar precej primero na podlagi katerih si lahko ustvarimo nekoliko jasnejšo sliko a vendar bi si želeli bistveno boljše dokumentacije. Edina »olajševalna« okolščina okoli dokumentacije je ta, da tudi ostali ponudniki

primerljivih rešitev nimajo bistveno boljše dokumentacije.

Druga ovira je popolnoma drugačna shema sestavljanja paketov za prejemanje oziroma pošiljanje (slika 5). Zaslon proti MCU-ju pošilja pakete, ki so v osnovi skupek bajtov pri čemer ima posamezen bajt določen pomen. Prvi bajt je sporoči tip paketa (ali se pošilja logično stanje ali pa se pošilja vrednost). Nato sledijo bajti, ki signalizirajo za kateri gradnik gre (ID gradnika – e.g. tipka, drsnik, ...) in nato še bajti z samo vsebino (vrednost drsnika, ...). Paket se zaključuje s tremi bajti vsebine 0xFF. Proti zaslonu pa so paketi pravzaprav ASCII znakovni nizi (string), kjer se navede ime gradnikov, parameter in vrednost na katero želimo nastaviti ta parameter. Tudi v tem primeru se paket zaključuje s tremi bajti vsebine 0xFF. Ta nesimetrija v popolnosti onemogoča da bi lahko vsaj manjši del kode uporabili tako za prejemanje, kot tudi za pošiljanje. Dodatno mora program v MCU ju vsebovati tako seznam imen gradnikov kot tudi njihovih identifikatorjev.

bar_0006.val=86

62 61 72 5F 30 30 30 36 2E 76 61 6C 3D 38 36 FF FF FF



Slika 5: Oblika paketov od in proti zaslonu

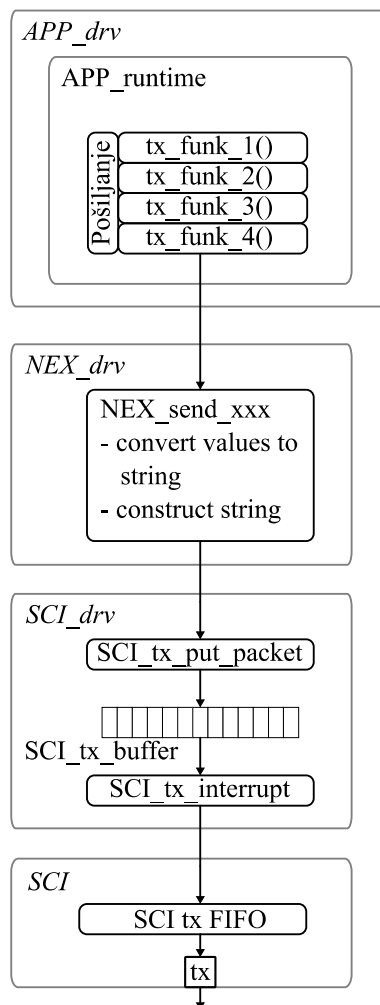
Tretja ovira je, da privzeto gradniki zaslona nimajo enovitih identifikatorjev. Tako imata lahko dva gradnika, ki se sicer nahajata vsak na drugi zaslonski sliki, enak identifikator. Rešitvi za ta problem sta dve. Ena možnost je, da program v MCU-ju sledi na kateri zaslonski sliki se trenutno nahaja zaslon. Da programu na MCU-ju ni treba slediti stanju zaslona, pa smo se mi raje odločili za drugo rešitev, pri kateri se omejimo na unikatno označevanje gradnikov. Tako morajo imeti gradniki identifikator, ki je sestavljen iz dveh števil in označujeta številko zaslonske slike na kateri se nahaja gradnik in številko gradnika. Pomanjkljivost tega pristopa pa je, da nihče ne preverja ali smo pravilno označili gradnike.

Četrta ovira pa je ne-enotna paradigma pošiljanja in prejemanja podatkov posameznih gradnikov. Za večino gradnikov prejmemo oziroma pošljemo stanje v enem samem paketu. Pri nekaterih gradnikih, ki bi zaradi svoje kompleksnosti zahtevali prejemanje/pošiljanje večih paketov, pa je treba preklopiti v popolnoma drugačen način pošiljanja, v tem načinu poslati en sam paket, ki je popolnoma drugačne oblike in nato nazaj preklopiti v standarden način.

Vse to, poleg ostalih navedenih posebnosti, povzroči, da komunikacijski sklad vsebuje veliko specialnih primerov ter mora slediti več stanjem zaslona, kar je dodatno otežilo razvoj in pisanje samega sklada.

4.1 Detajlna arhitektura komunikacijskega sklada

Pošiljanje podatkov proti zaslonu (slika 6) je relativno enostavno. MCU s klicem funkcij, ki se nahajajo v *APP_drv* pripravi podatke, ki jih želi poslati zaslonu za prikaz (naprava deluje, napetost je x V, ...). Ta podatek se skupaj z identifikatorjem gradnika posreduje *NEX_drv*. Le ta poišče kateremu gradniku ta identifikator pripada in sestavi celoten paket ter ga preda *SCI_drv*. Ta pa naloži vse bajte paketa, ki jih lahko v *SCI FIFO* spomin, preostale podatke pa pripravi, da jih bo v *SCI FIFO* spomin naložila prekinitvena funkcija, ko se bo *FIFO* spomin dovolj sprostil.



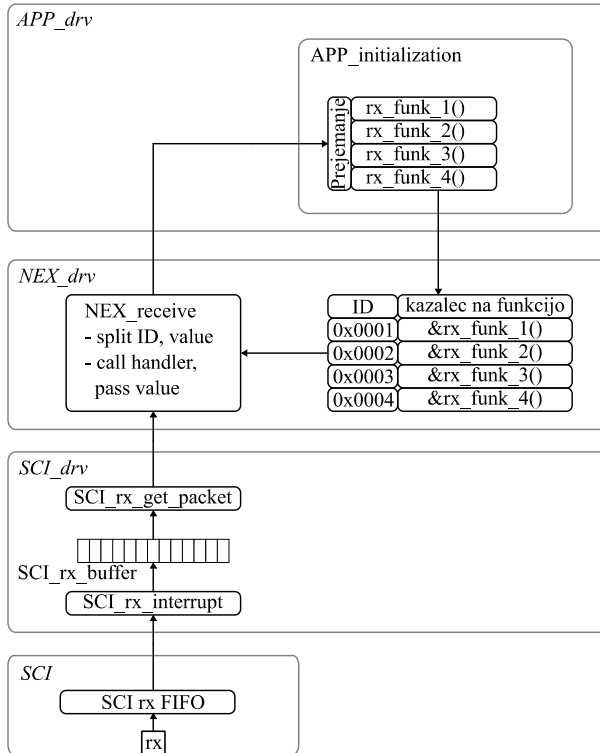
Slika 6: Pošiljanje paketov

Prejemanje podatkov pa je nekoliko bolj zapleteno. Ob inicializaciji, koda v *APP_drv* registrira ustrezne funkcije, ki naj se izvedejo kot povratni klic, ko zaslon pošlje paket na pobudo gradnika z določenim identifikatorjem. Te funkcije si skupaj s pripadajočimi identifikatorji v tabelo shrani *NEX_drv*. Ko zaslon pošlje paket, *SCI_drv* zazna konec paketa, ki je označen s tremi bajti z vsebino 0xFF. Periodično *NEX_drv* preveri, ali je že kakšen paket na voljo, in če je, ga prevzame, razčleni in glede na njegovo identifikator pokliče ustrezno funkcijo.

4.2 Kronološki potek razvoja

Kronološko je razvoj potekal v naslednjem vrstnem redu. V začetku smo z Nextion zaslonom komunicirali samo preko osebnega računalnika z terminalskim programom saj je tako bistveno hitreje in lažje preveriti kako oblikovane pakete pošilja zaslon. Prav tako lažje oblikujemo različne pakete, ki jih pošiljamo na zaslon.

Ko je bilo delovanje *SCI_drv* gonilnika potrjeno smo implementirali sestavljanje in razčlenjevanje paketov znotraj *NEX_drv*.



Slika 7: Prejemanje paketov

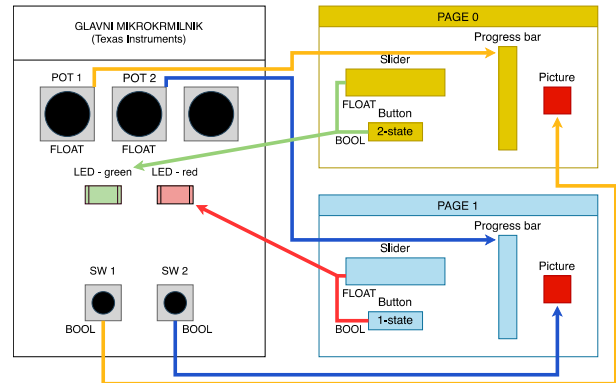
4.3 Implementirana funkcionalnost

S projektom smo pričeli leta 2021 in do sedaj smo uspešno implementirali najbolj osnovne funkcionalnosti kot so prejemanje in pošiljanje logičnih stanj (funkcionalnost tipke in signalne LED diode) ter prejemanje in pošiljanje števnih stanj (funkcionalnost potenciometra in analognega oz. digitalnega prikazovalnika).

5 Zaključek

Pri razvoju sklada smo se precej naslanjali na predhodno razvit podoben komunikacijski sklad. To nam je vsekakor olajšalo delo.

Največ težav je pri razvoju predstavljala pomanjkljiva dokumentacija ter zelo nedosledno zastavljen komunikacijski protokol Nextion zaslona.



Slika 8: Implementirana funkcionalnost

Sodeč po izkušnjah iz razvoja komunikacijskega sklada za komunikacijo z osebnim računalnikom bo v nekaj letih predstavljeni komunikacijski sklad vseboval večino pričakovanih funkcionalnosti ter bo tudi zelo dobro preizkušen.

Zahvala

Delo je bilo sofinancirano iz programa ARRS »Pretvorniki električne energije in regulirani pogoni« P2-0258 (B).

Literatura

- [1] A. Rihar, P. Zajec, M. Nemeč, M. Petkovšek, and D. Vončina, "Načrtovanje, zasnova in izdelava večfaznega razsmernika z MOSFET stikali," Zbornik Elektrotehniške in računalniške konference ERK. Portorož, Slovenija: IEEE Region 8, Slovenska sekcija IEEE, 2018.
- [2] A. Rihar, M. Nemeč, K. Drobnič, and D. Vončina, "Zagon in preliminarno preizkušanje vodenja 2x3-faznega pogona," Zbornik Elektrotehniške in računalniške konference ERK. Portorož, Slovenija: IEEE Region 8, Slovenska sekcija IEEE, 2018.
- [3] M. Nemeč, "Komunikacijski sklad in nadzorna aplikacija za mikrokrmilnike," Zbornik Elektrotehniške in računalniške konference ERK. Portorož, Slovenija: IEEE Region 8, Slovenska sekcija IEEE, 2022.
- [4] G. Šajn, "Načrtovanje in izdelava enofaznega usmernika z dvosmernim pretokom energije in galvansko ločitvijo: diplomsko delo univerzitetnega študija," Univerza v Ljubljani, Fakulteta za elektrotehniko, Ljubljana, 2014.
- [5] M. Rožej, "Enofazni serijski aktivni močnostni filter," Diplomsko delo univerzitetnega študija, Univerza v Ljubljani, Fakulteta za elektrotehniko, Ljubljana, 2017.
- [6] Patrick, "Instruction Set," *Nextion*. <https://nextion.tech/instruction-set/> (dosegljivo jul. 11, 2023).