

Miniaturen USB vmesnik za nadzor PTT in CW

Avtor: Marko Pavlin, S54MTB
E-pošta: marko@pavlin.si

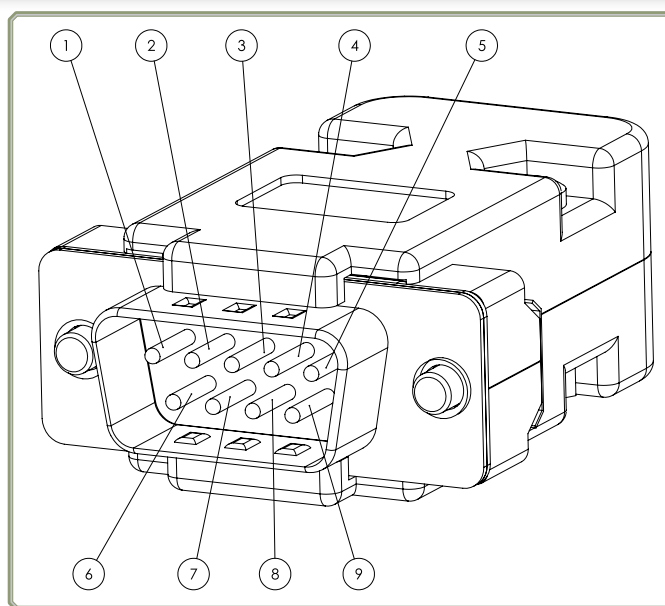
V prispevku bom najprej predstavil nekaj malega o tehničnem ozadju nadzora signala za oddajo (PTT) in signala in ostalih signalov (npr. CW ali vklop linearca). V drugem delu bo na kratko opisan potek razvoja in končni izdelek.

Novejše radijske postaje imajo možnost direktnega priklopa na USB vmesnik na računalniku. Pri tem se postaja »predstavi« kot virtualni serijski vmesnik (Virtual COM port) in zvočna kartica. Tako postajo je moč krmiliti z večino starejših in novejših programov za delo v foniji, telegrafiji ali digitalnih načinih brez uporabe dodatnih vmesnikov.

Seveda pa imamo na mizah, v delavnicah, gorskih postojankah, mobilnih lokacijah in drugod obilico postaj starejšega datuma, ki takih naprednih povezav nimajo. Po drugi strani se računalniška oprema in programi za delo v kontestih, za digitalne načine dela ali enostavno telegrafijo s pomočjo računalnika stalno posodabljujejo. Žal sočasno s posodabljanjem računalnikov počasi tudi izginja večina vmesnikov, s pomočjo katerih se je v preteklosti upravljalo postaje. S tem imam v mislih predvsem serijski vmesnik RS232 ali pogovorno »COM port«. Že davno nazaj pa smo »izgubili« paralelni tiskalniški vmesnik, ki ga pogovorno poznamo kot »LPT port«. Večino teh vmesnikov je nadomestil USB vmesnik ali morda kar kakšna od brezžičnih povezav. Seveda pa starejše postaje s tem niso za na odpad, le ustrezni vmesnik potrebujejo, ki bo povezal starejšo tehniko s sodobnimi povezovalnimi tehnologijami.

Priključni konektorji standardnega serijskega vmesnika

Za začetek si na kratko osvežimo spomin, kako deluje enostavni preklop oddaje »PTT« s pomočjo RS232 vmesnika. Če pogledamo konektor na računalniku imamo dve možnosti: ali ima 9 ali pa 25 priključkov. Na začetku je imel RS232 konektor 25 priključkov. Malo manj znano dejstvo je, da sta bila v razporedu priključkov 25 pinskega »D« konektorja dejansko dva serijska vmesnika. V praksi se to sicer ni najbolj prijelo in le redki računalniki so bili dejansko z dvema serijskima vmesnikoma na enem DB25 konektorju. Ideja sicer ni slaba, saj je bil osnovni vmesnik z vsemi nadzornimi signali namenjen komunikaciji



Slika 1: Konektor DB9 z oznakami pinov

z modemom, drug serijski vmesnik pa se je uporabljal za terminalski dostop in upravljanje takega modema brez prekinjanja podatkovne povezave. Kasneje se je 25 piskemu pridružil še 9 pinski konektor, ki je bolj praktičen in se tudi največ uporablja. Na računalniški strani je vedno moški tip konektorja. Če je slučajno na zadnji strani ženski 25 pinski konektor gre po vsej verjetnosti za tiskalniški »LPT« vmesnik.

DB9	DB25	Oznaka	Funkcija priključka	Smer pri računalniku
1	8	DCD	Data carrier detect	Vhod
2	3	RxD	Receive data	Vhod
3	2	TxD	Transmit data	Izhod
4	20	DTR	Data terminal ready	Izhod
5	7	GND	Signal ground	-
6	6	DSR	Data set ready	Vhod
7	4	RTS	Request to send	Izhod
8	5	CTS	Clear to send	Vhod
9	22	RI	Ring indicator	Vhod

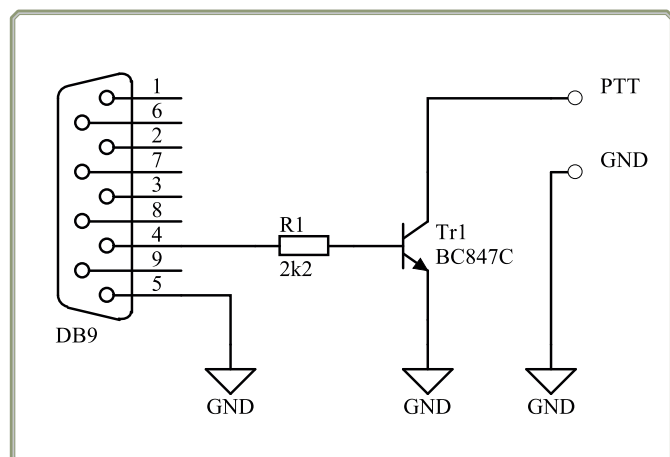
Tabela 1 – razpored priključkov na konektorju serijskega vmesnika

V tabeli 1 je prikazan razpored priključkov za 9 in za 25 pinski konektor serijskega vmesnika na računalniku. Enak 9 pinski konektor srečamo tudi na vseh »USB/RS232« pretvornikih. Imena signalov so zapisana z angleškimi nazivi, ker se ti tudi največ uporabljajo in tudi v vseh programih za radioamatersko delo bomo srečali take nazive ali njihove kratice. V zadnji koloni je označena smer signala v primeru, ko je serijski vmesnik pri računalniku. Ustrezno poimenovanje in vlogo naprav bom podrobneje opisal v nadaljevanju.

Standardni napetostni nivoji serijskega vmesnika Logične nivoje 1 in 0 lahko v elektronskih vezjih ponazorimo z različnimi napetostnimi nivoji. V 5V TTL logiki je stvar dokaj »logična«: ničla je 0V, enica pa 5V. Seveda v resnici ni čisto tako, saj je npr. enica lahko vsaka napetost v določenem razponu (npr. od 2,7V do 5V) in ničla vse, kar je pod npr. 0,5V.

Pri serijskem vmesniku RS232 pa so nivoji malce drugačni, predvsem je značilno to, da so napetosti tudi negativne. Enica na signalnih linijah ima lahko napetostni nivo od -3V do -25V, ničla pa ima nivo od +3V do +25V. Na kontrolnih linijah pa je ravno obratno: aktiven signal predstavlja pozitivna napetost na liniji med +3V in +25V, neaktiven signal pa predstavlja nivo od -3V do -25V.

Notranjost računalnika in njegovi vmesniki delujejo večinoma s standardnimi logičnimi nivoji, nekoč 5V, potem 3,3V danes pa kar na 1,8 ali celo 1,2V. Logične enice in ničle s temi nivoji je potrebno pretvoriti v nivoje, ki jih zahteva RS232 s pomočjo namenskih integriranih vezij. Ta vsebujejo stikalni napetostni pretvornik in nekaj tranzistorjev, ki pretvorijo voden logični nivo v pozitivne in negativne napetostne nivoje RS232. Te napetostne nivoje pa potrebujemo le v primeru, ko je na postaji standarden RS232 vmesnik, ki to zahteva. Za preklon releja za oddajo to seveda ni potrebno.



Slika 2: Shema najpreprostejšega »PTT« preklopnika na DB 9 konektorju

»Zloraba« kontrolnih linij za krmiljenje postaje

Krmilne linije RTS, CTS in DTR se pri običajni uporabi RS232 lahko uporabljajo za nadzor pretoka podatkov. S signali RTS, CTS in DTR se napravi na obeh koncih kableske povezave dogovorita ali sta pripravljena na sprejem in/ali oddajo podatkov. Poleg uporabe teh treh linij sicer obstaja še veliko drugih možnosti, vendar je tak način (t.i. hardverski nadzor / hardware handshake) najpogosteje uporabljen. Kratice in funkcija posameznih signalov so podane v tabeli 2.

Oznaka signala	Funkcija	Lokacija
RTS	Request to Send	DTE
CTS	Clear to Send	DCE
DTR	Data Terminal Ready	DTE

Tabela 2

Pri tem je »lokacija« DTE ali DCE, kar sta kratici za Data Terminal Equipment in Data Communications Equipment. Tipičen primer za DTE in DCE sta PC s terminalskim programom (DTE) in modem (DCE). Za lažje razumevanje besedila v nadaljevanju bom uporabil kar Terminal in Modem namesto DTE in DCE, čeprav to ni vezano le na take vrste naprav.

Na začetku prenosa terminal vklopi linijo RTS. Modem potem odgovori z aktiviranjem linije CTS in terminal odgovori z aktiviranjem DTR linije, ki ostane aktivna ves čas prenosa. Po koncu prenosa bo terminal preklonil liniji DTR in RTS v neaktivno stanje in modem linijo CTS prav tako v neaktivno stanje. S tem obe napravi sporočita druga drugi, kdaj sta pripravljena za prenos.

Če pogledamo izven konteksta prenosa podatkov lahko opazimo, da imamo na strani računalnika dve liniji, ki ju lahko nadzorujemo. To sta torej dva izhoda z dvema napetostnima nivojema: RTS in DTR. Če pogledamo nazaj v poglavje »Napetostni nivoji«, lahko vidimo, da za kontrolne linije pomeni aktiven signal pozitivno napetost, neaktiven pa negativno. To nam na enostaven način omogoča uporabo obeh linij za krmiljenje npr. NPN tranzistorja, ki deluje kot stikalo. Radijske postaje imajo ponavadi preklon za oddajo izveden tako, da moramo »PTT« signal povezati na GND, da gre postaja na oddajo. Izvedba takega vezja je torej zelo trivialna (Shema na sliki 2)

Kaj pa če naš računalnik nima »DB9« konektorja?

V uvodu sem omenil, da določeni vmesniki izginjajo iz sodobnih računalnikov. Prvi tak je bil »LPT«, oz. tiskalniški vmesnik. Sledil mu je

serijski vmesnik »COM port«, verjetno pa se bo računalniška industrija počasi znebila še kakega nebodigatreba vmesnika. K sreči zaenkrat še dobro kaže USB vmesniku, kar pomeni, da lahko na njegovi osnovi naredimo vezje, ki bo služilo za preklapljanje naše stare postaje še mnoga leta tudi po tem, ko bomo računalniško »ropotijo« že sedemkrat zamenjali.

Za izvedbo imamo več možnosti, pri tem pa je potrebno pogledati slabe in dobre lastnosti z različnih zornih kotov. USB vmesnik je zelo uporaben, saj poleg podatkovne povezave omogoča tudi napajanje naprav, ki so nanj priključene. Za skromno porabo našega vmesnika bo tistega pol ampera toka, ki ga lahko standardno dobimo iz USB več kot dovolj. Kakor je serijski vmesnik relativno preprost, tako s stališča povezovanja, konektorjev, nezahtevnih kablov, kot programiranja, pa so stvari pri USB malo bolj zapletene. Po priklopu neke naprave se najprej izvede izmenjava podatkov med USB »host« napravo (v našem primeru PC) in priključeno napravo. Ta začetna izmenjava se imenuje enumeracija in omogoča, da se novo priključena USB naprava predstavi sistemu, ta pa potem iz teh podatkov razbere, kakšna je ta naprava, kdo je proizvajalec in za kateri tip naprave tega proizvajalca gre. Oba podatka sta zapisana v obliki 16-bitne celoštevilске vrednosti. Proizvajalčeva oznaka je zapisana kot Vendor ID (VID), oznaka izdelka tega proizvajalca pa je zapisana v ProductID (PID). Iz para VID/PID potem sistem izbere ustrezen gonilnik (če je ta nameščen), ali pa ga gre iskati na določeno mesto (npr. Windows update ali izbrano mesto na disku). Gonilnik poleg vsega ostalega vsebuje kos programske kode, ki komunicira z dano napravo po vnaprej dogovorjenem protokolu. Poleg namenskih gonilnikov za specifične naprave so v uporabi tudi že vnaprej nameščeni gonilniki za komunikacijo preko USB vmesnika s pogosto uporabljenimi. Taki gonilniki, ki se namestijo ob nameščanju operacijskega sistema se uporabljajo za bolj vsakdanje, pogosto uporabljene naprave, npr. tipkovnico, miško, zunanji disk, kamero in podobno. Poleg VID/PID oznak naprava sistemu postreže tudi s podatkom o klasifikaciji naprave, t.i. device class. Obstaja nekaj standardiziranih vrst naprav, ki so določene s standardi USB, npr. HID (Human Interface Device), MSC (Mass Storage Class), CDC (Communication Device Class) in podobno. Tipična »HID« naprava je miška, igralna palica ali tipkovnica. Tipična MSC naprava je »USB ključek«, kamor shranjujemo datoteke. Za nas je najbolj zanimiva specifikacija za CDC naprave. Implementacija take naprave nam omogoča izdelavo t.i. virtualnega COM port oz. naprave, ki se priklopi na USB, v sistemu pa deluje kot serijski vmesnik.

Virtualni COM port

Vmesnik, ki se priklopi na USB in na »drugi« strani deluje kot serijski vmesnik imenujemo virtualni COM port. Seveda pa ni nujno da je tak virtualni COM port zaključen z DB9 konektorjem, na katerega priključimo napravo s serijskim vmesnikom, ampak je lahko tudi npr. modem, ki se priklopi na USB. V tem primeru, se »COM port« zaključi znotraj elektronike tega modema in sploh nimamo fizičnega dostopa do linij serijskega vmesnika.

Za nas je seveda zanimiva rešitev, ki ima priključke do vseh linij serijskega vmesnika in se seveda vklopi v USB. Za izdelavo imamo več možnosti: lahko ga kupimo (poceni kitajski ponaredek ali malo dražji »original«), lahko ga izdelamo s pomočjo namenskega vezja ali uporabimo mikrokontroler z USB vmesnikom in poleg elektronike izdelamo tudi programski del vmesnika. Pregled nekaterih rešitev z namenski vezji je prikazan v tabeli 3.

Oznaka	Proizvajalec	Velikost	~ Cena*	Opombe
PL2303SA	Prolific	SO8	~2€	Samo Rx in Tx, nedobavljiv
PL2303HXD	Prolific	QFN32	~5€	8xGPIO, 12MBps, nedobavljiv
CP2102	Silicon Labs	QFN28	2,3€	Pogosto v uporabi
MCP2221A	Microchip	SO14	1,4€	Ima dodatne GPIO
FT232xx	FTDI	SSOP28	2,6€	Najpogosteje v uporabi, poceni ponaredki nezanesljivi
USB-2SERA11	NXP	QFN24	2,7€	Manj znan in težje dobavljiv
TUSB3410	TI	QFN32	4,1€	+I2C, +IrDA, Ponuja več kot rabimo
CY7C65213	Cypress	QFN32	1,9€	Manj znan in težje dobavljiv

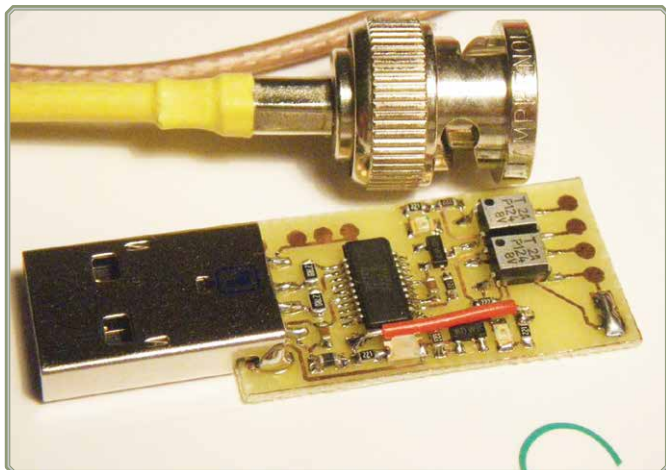
Tabela 3: Pregled namenskih krmilnikov za virtualni COM port. * za 100 kos, na octopart.com

Če v tabeli 3 vržemo bežen pogled na kolono s cenami hitro ugotovimo, da nekaj ni ravno prav. Kako je lahko namensko vezje, ki »samo« pretvori USB vmesnik z njegovim protokolom v asinhroni serijski protokol dražje od enega mikrokontrolerja, ki lahko počne isto stvar, pa še kaj drugega poleg. Točen odgovor meni ni poznan, slutim pa, da se skriva v več vzrokih. Eno je gotovo razvoj in vzdrževanje gonilnikov. Ko priklopimo npr. FT232 na USB, je gonilnik na voljo preko vseh standardnih »kanalov« distribucije gonilnikov in inštalacija se izvede povsem avtomatično, praktično brez »pritiska na gumb« in to v vseh znanih operacijskih sistemih za »100 let« nazaj. Druga stvar je tudi to, da gre za namensko vezja, ki se vendarle ne proizvajajo tako množično kot npr. en popularen mikrokontroler. Razvoj pa stane in skozi manjšo količino prodanih kosov to seveda pomeni višjo ceno na kos.

Potem pa seveda pride poševnooki Azijec in zmeša štrene s prodajo ponaredkov. Ti pa ali delajo ali pa ne. V boju proti ponaredek se proizvajalci borijo na različne načine. Znan je nedavna afera proizvajalca FTDI, ki je priredil gonilnike tako, da je v primeru, ko je zaznal ponaredek, vpisal v ProductID (PID) vrednost nič in s tem tak ponaredek uničil. Vse lepo in prav, vendar je bila težava, da uporabniki niso niti vedeli, da imajo ponaredke. Ko so se Windows sistemi preko noči posodobili, naslednji dan naprave na USB niso več delovale in v končni fazi je bilo več škode kot koristi.

Kaj potem izbrati za naš modulček za preklapljanje signalov?

Če lahko preživimo z »ročno« namestitvijo gonilnika in nekaj lastnoročnega spajkanja, bi bila gotovo idealna rešitev uporaba enega od sodobnih mikrokontrolerjev. Objavim shemo, enostransko tiskano vezje, napišem kodo za to napravo, priložim gonilnik in stvar deluje. To bi bilo seveda v idealnem svetu. Žal pa v resnici ni tako. Pri podrobnem preizkušanju rešitev za ta preprost vmesnik sem naletel na nerešljiv problem. Ker pri svojem delu redno uporabljam družino mikrokontrolerjev STM32, ki ima odlično podporo in veliko izbiro, je bila seveda to tudi prva izbira za preizkušanje. Ker sem v preteklosti naredil že nekaj lastnih aplikacij z virtualnim COM portom (ena je bila opisana tudi v CQ ZRS pod naslovom »Miniaturni CW keyer«), sem korajžno narisal malo tiskano vezje in ga dal v izdelavo brez posebnega predhodnega preizkušanja. Ko so ploščice prišle, sem vezje sestavil (Slika 3b) in spisal nekaj vrstic kode za test. Preizkušanje z različnimi programi za terminale in za nadzor radijskih postaj ni in ni delovalo, kot bi moralo. Prenos znakov po virtualnem COM portu ni delal težav, zataknilo se je pa pri nadzoru kontrolnih linij DTR in RTS. Šele natančen vpogled v surove



Slika 3a: Prvi prototip z FT232RL

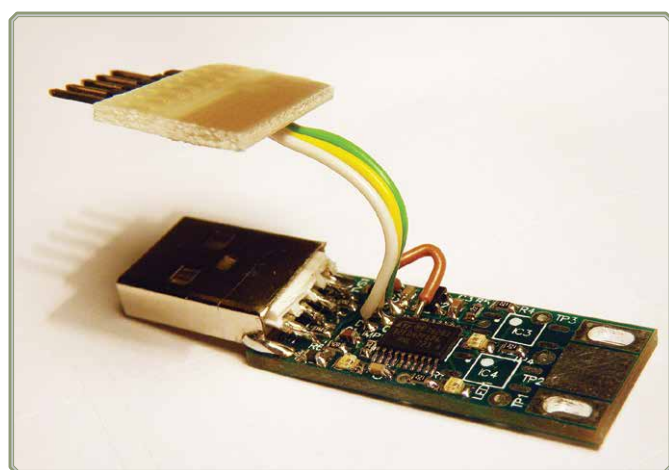
pakete podatkov na USB vmesniku je razkril, da se paketi za nastavljanje DTR in RTS ne razlikujejo med seboj. Težava je torej v gonilniku `usbser.sys`, za katerega sem kasneje ugotovil, da ima kar nekaj hroščev, ki so bolj ali manj dokumentirani. Rešitev bi bila pisanje lastnega sistemskega gonilnika, kar je obilen zalogaj, saj mora podpirati vse možne različice oken, linuxa in ostalih sistemov, ki jih radioamaterji uporabljamo, ali pa izbira komercialnega gonilnika, ki bi nadomestil hroščasti `usbser.sys`. Ne ena ne druga rešitev mi nekako ni dišala. Pisanje gonilnikov ni ravno preprosta naloga, komercialni pa niso prav poceni. Zaradi te nerešljive zagate sem se odločil, da izberem kar preverjeno rešitev z enim od uveljavljenih in preizkušenih namenskih čipov.

Ker sem se odločil, da lastnega sistemskega gonilnika ne bom delal, ampak bom raje še malo jedkal in spajkal, je na tem mestu torej končno čas za prvi prototip.

Prvi prototip

Prvi prototip sem izdelal na enostranski ploščici, ki je bila zjedkana doma. Malo sem pobrskal po predalih in predalčkih s komponentami in odkril, da sem nekoč nabavil eno peščico FT232RL v TSSOP ohišju in pozabil nanje. Kot naročeno za prvi poskus. Shema je enaka, kot v dokumentaciji za FT232, dodal sem še USB vtič, kot ga ima večina »USB ključkov« in vse skupaj preizkusil. Najlažje je bilo kar z terminalskim programom RealTerm, ki omogoča ročno preklapljanje kontrolnih linij [<https://realterm.sourceforge.io/>]

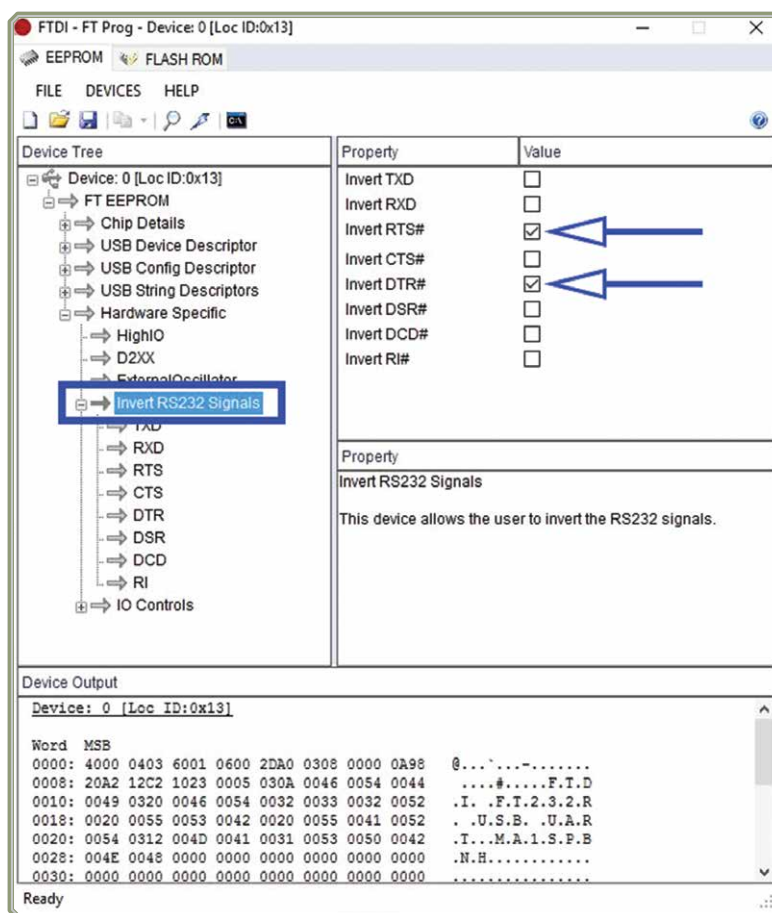
Preden sem se lotil dokončnega načrtovanja vezja, sem modul zanesel še do S52AS, ki ima veliko izkušenj s starimi postajami in vedno priskoči na pomoč s koristnimi povratnimi informacijami. Še isto popoldne me je »razveselil« z novico, da je sicer fino, ampak ne dela (prav). Seveda, krmilni izhodi na FT232RL so ravno invertirani (glej razdelek tega prispevka



Slika 3b: Prototip z STM32F042

pod naslovom »Zloraba kontrolnih linij za krmiljenje postaje«). To pa pomeni, da moram dodati še po en tranzistor, ki bo izhod negiral ali pa preprogramiranje FT232RL s pomočjo programa FTPROG. Ta programski pripomoček omogoča spremembo konfiguracije in invertiranje kontrolnih signalov, kar je elegantna rešitev in ne zahteva dodatnih komponent v vezju. Postopek je opisan v priročniku za FTPROG, v poglavju 5.5. Dokument pa se nahaja na naslovu http://www.ftdichip.com/Support/Documents/AppNotes/AN_124_User_Guide_For_FT_PROG.pdf

Zamenjava polaritete je preprosta. Najprej vezje z vmesnikom FT232xx vtaknemo v prost USB priključek in poženemo program FTPROG. V seznamu na levi strani se pojavi priključen vmesnik. V seznamu poiščemo opcijo FT EEPROM in sekcijo Invert RS232 signals. Na desni strani obkrijemo Invert RTS# in Invert DTR#. Kliknemo še na ikono v obliki strele, da se nastavitve vpišejo in s tem je delo končano. Okno programa je prikazano na sliki 4.



Slika 4: Okno programa FTPROG, s katerim invertiramo signala DTR in RTS.

Dokončna verzija

Kaj pa radioamaterji še potrebujemo poleg trivialnega »PTT« signala? Če je to edini vhod na postaji, potem praktično nič drugega. Če pa poleg postaje vklapljamo še npr. linearca ali pa izklopjamo predojačevalc na sprejemu, je pa tu že potrebno imeti predpisano sekvenco preklopov. Prav tako lahko z enim od obeh signalov krmilimo npr. krmilni signal za telegrafijo. V tem primeru so možnosti z »direktnim« pretvornikom iz USB na serijski port nekoliko omejene, a še vedno ponujajo dovolj možnosti za koristno uporabo. Seveda to drži ob ustrezni podpori s strani programa, ki upravlja te preklope. Zaradi tega sem tudi dal na vezje dva izhoda, ki sta oba galvansko ločena in lahko krmilita PTT in še drug signal, npr. vklop linearca ali pa morda vhod za taster (CW).

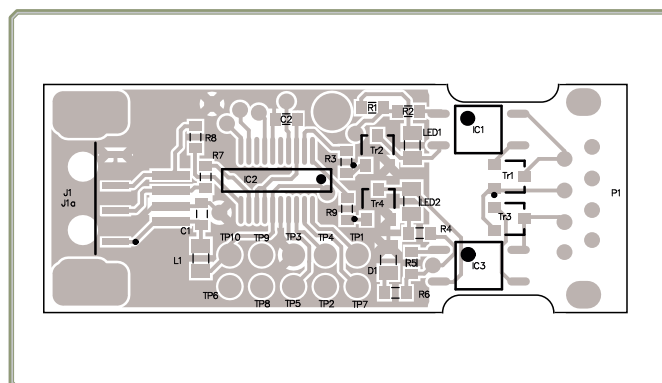
Električna shema

Električna shema je sestavljena iz treh delov:

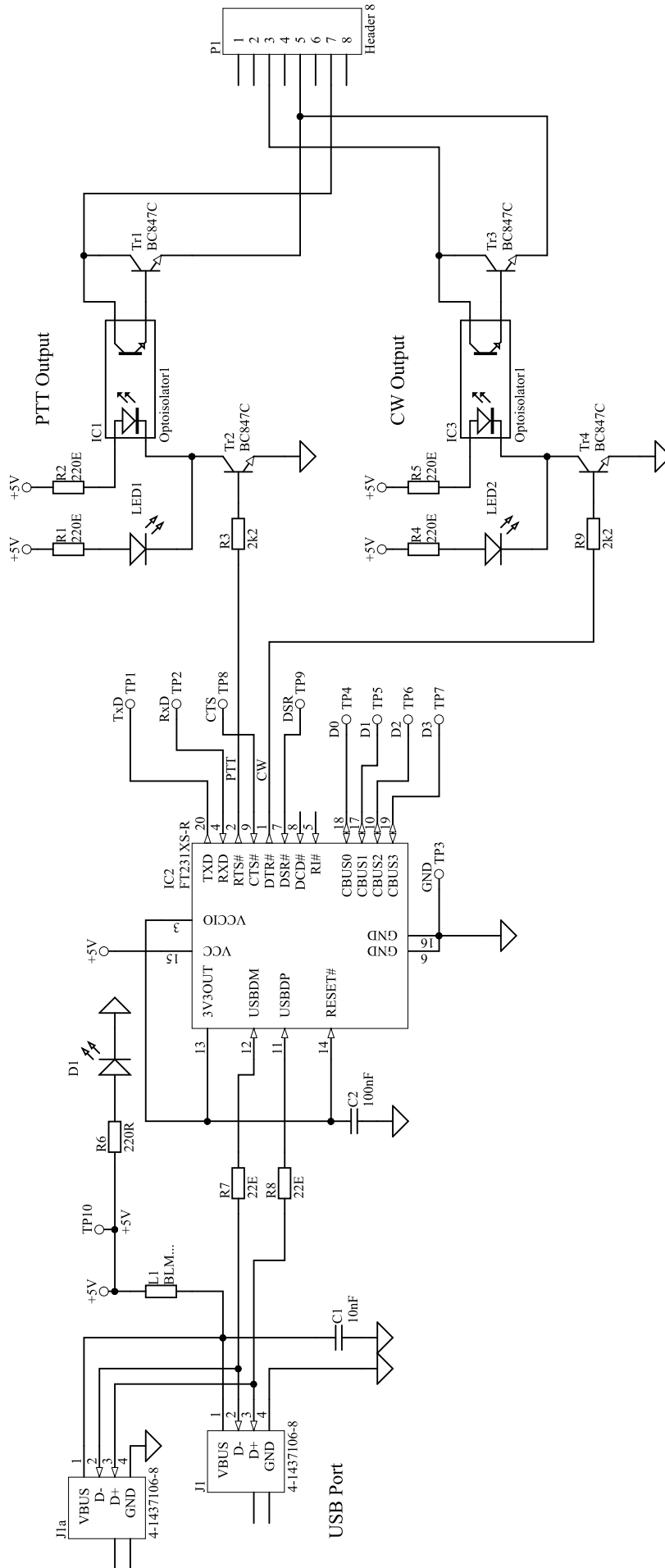
- USB priključek z zaščito proti elektrostatiki,
- Pretvornik iz USB v UART (FT232RL)
- Krmiljenje izhodnih signalov z galvansko ločitvijo

Na shemi sta dva USB konektorja J1 in J1a. J1 je konektor tip B, s pomočjo katerega povežemo vmesnik na USB kabel. Konektor J1a pa je tip A, s pomočjo katerega modul direktno vtaknemo v USB vtičnico. Naspajka se seveda samo enega. Konektorju sledi ESD zaščita. To je majhna

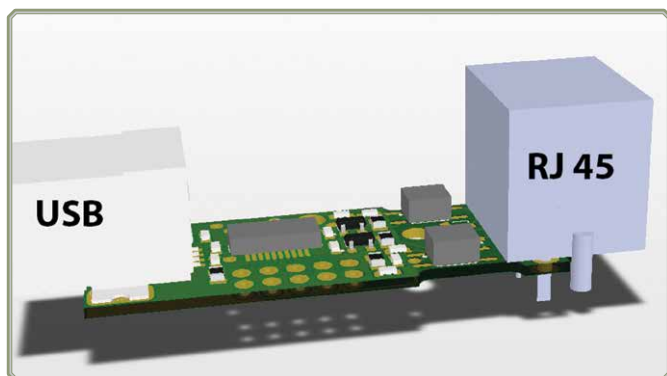
komponenta s petimi supresorskimi diodami, ki zadržijo razelektritvene udare, ki nastanejo ob dotiku vezja z naelektrenim uporabnikom. Seveda pa ne ta zaščita ne optična ločitev na izhodu ne bo prestregla strele, ki bi direktno udarila v anteno in preko postaje preskočila še v računalnik. Za to bo potrebno poskrbeti med anteno in postajo. Sam pretvornik USB na UART ni nič posebnega. V shemi sem sledil proizvajalčevim priporočilom, predvsem za kondenzatorje in priklop napajanja. Izhoda DTR in RTS krmilita dva NPN tranzistorja, ki vklapljata indikatorski svetleči diodi in obenem tudi diodi optičnih ločilnikov. Na galvansko ločenih



Slika 5: Montažna risba vmesnika



Slika 6: Shema



Slika 7: Končni izdelek za montažo na kabel postaje preko RJ45 in USB konektorjem tipa B

izhodiha sta dodana še dva NPN tranzistorja, ki poskrbita, da notranja v optičnem ločilniku nista preobremenjena. Montažna risba je na sliki 5.

Izhod je priključen na standardni RJ45 konektor (Slika 7). To poenostavi izdelavo kabla proti postaji, saj lahko vzamemo poljuben mrežni kabel, ga prerežemo na pol in dobimo dva na pol narejena kabla. Dodati je potrebno le še konektorje za postajo. Seveda pa ni nuno, da uporabimo tak konektor in kabel preprosto prispajkamo. V tem primeru pridejo prav zajede ob robu tiskanega vezja, kjer lahko pripnemo kabelsko vezico, ki drži priključni kabelček čvrsto ob tiskano vezje (Slika 8).



Slika 8: Končni izdelek z direktnim spajkanjem kabla in USB konektorjem tipa A

Neuporabljeni signali so na voljo na robu tiskanega vezja na blazinicah z razmakom 2,54mm, kar omogoča preproste razširitve (npr. še CAT ali vhod za taster in podobno).

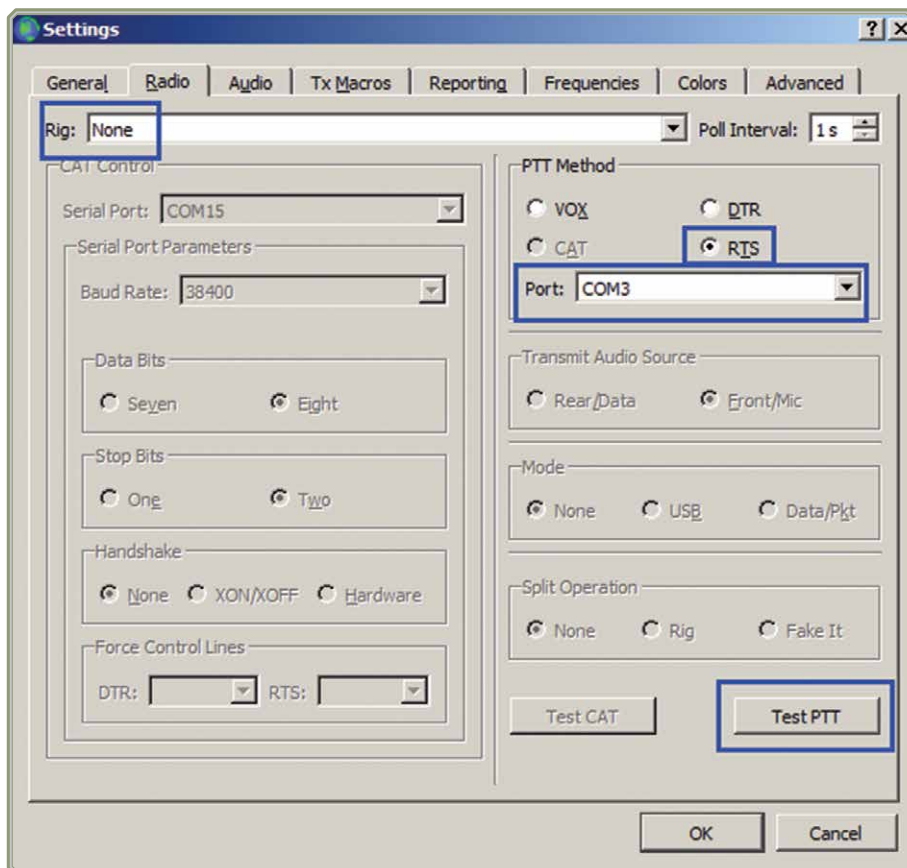
Tiskano vezje je dvostransko in je na voljo pri avtorju prispevka. Datoteke (gerber in drill) so na voljo na avtorjevi spletni strani.

Uporaba v praksi

Vmesnik je bil praktično preizkušen s postajo FT817, Racal in FT950 s programom WSJT-X za delo v digitalnih načinih. Z njim je bilo do sedaj opravljenih preko 1000 QSO brez posebnih težav. Za delo je potrebno v nastavitvah WSJT-X izbrati ustrezen COM port, ki se pojavi po priklopu vmesnika, za postaje izberemo »None«, za kontrolo PTT pa DTR ali RTS, odvisno od tega, kateri izhod smo uporabili za nadzor PTT (Slika 9).

Kako naprej?

V delu je že vmesnik, s pomočjo katerega se bo dalo zajemati in generirati zvočne signale in bo imel tudi galvansko ločen »CAT«, oz. RS232 vmesnik. Avdio in digitalne linije bodo v celoti galvansko ločene, celoten vmesnik pa se bo v sistemu pojavil kot virtualni COM port in USB zvočna kartica. S tem vmesnikom bo mogoča popolna digitalizacija vseh starih radijskih postaj preko enega samega USB kabla.



Slika 9: Nastavitev WSJT-x za delo z opisanim vmesnikom