Univerza v Ljubljani
Fakulteta za elektrotehniko

Ana Marija Turšič

SODOBEN FREKVENČNI ŠTEVEC ZA OBMOČJE RADIJSKIH FREKVENC

Magistrsko delo

Mentor: prof. dr. Matjaž Vidmar

Ljubljana, 2014

Zahvala

Zahvaljujem se prof. dr. Matjažu Vidmarju za mentorstvo pri izdelavi magistrskega dela, nasvete, usmeritve, vzpodbude in njegovo prilagodljivost ter potrpežljivost. Hvala as. dr. Leonu Pavloviču, Simonu Staniču in ostalim članom Laboratorija za sevanje in optiko za pomoč v laboratoriju ter za nesebično deljenje dragocenih nasvetov in izkušenj. Hvala Sandiju, Igorju ter drugim prijateljem, ki so mi pomagali tako po strokovni, kakor tudi po prijateljski plati. Za vzpodbudo in predvsem prilagodljivost se zahvaljujem sodelavcem iz sektorja za nadzor radiofrekvenčnega spektra Agencije za komunikacijska omrežja in storitve; Matiji, Dušanu, Alešu, Miranu in Niku. Nenazadnje gre posebna zahvala moji družini, Mojci in Andreju, ter ostalim sorodnikom, ki so me tekom izdelave magistrske naloge podpirali, vzpodbujali in z menoj čutili.

Kazalo vsebine

Kazalo slik

Kazalo tabel

Seznam uporabljenih simbolov

V pričujoči magistrski nalogi so uporabljene naslednje veličine in simboli:

Veličina / oznaka		**Enota**	
IME	*SIMBOL*	IME	SIMBOL
frekvenca	f	hertz	Hz ali s^{-1}
čas	t	sekunda	s
jakost signala	P	decibel ali decibel-miliwatt	dB ali dBm
napetost	U	volt	V
pozitivna napajalna napetost (ECL)	V_{CC}	volt	V
negativna napajalna napetost (ECL)	V_{EE}	volt	V
upornost	R	ohm	Ω
kapacitivnost	C	farad	F
ločljivost	L	hertz	Hz
modulo deljenja	N	-	-
število impulzov	n	-	-

Natančnejši pomen simbolov in njihovih oznak je razviden iz spremljajočega besedila ali ustreznih slik.

Povzetek

V pričujočem delu smo iskali rešitev za merilni inštrument, ki je utonil v pozabo. Marsikje so ga nadomestili so ga ceneni spektralni analizatorji, ki imajo funkciji štetja frekvence ter merjenja jakosti signala že vgrajeni. Vendar nam je pri delu pogosto odveč velika in težka škatla, kot je spektralni analizator, in potrebujemo le priročen merilnik frekvence. Poleg tega je zastarela zasnova visokofrekvenčnih števcev s počasnim preddelilnikom, neprimerno vhodno občutljivostjo, neprimerno vhodno impedanco in neprimernimi sondami naredila merilnik nepraktičen. To magistrsko delo opisuje sodoben merilnik frekvence in jakosti signala. Njegove glavne odlike so visoka vhodna občutljivost, širok frekvenčni razpon ter visoka hitrost merjenja tako frekvence kakor tudi jakosti signala.

V uvodu so opisane izvedba ter osnove delovanja števcev frekvence. Predmet tega dela, sodoben števec frekvence, prinaša kar nekaj izboljšav in modernejših rešitev v primerjavi z običajnimi števci. Deluje brez preddelilnika, ne potrebuje ponastavljanja (reseta) vrat, ima stabilen zunaji oscilator, upravljamo ga lahko preko zunanjega računalnika, če naštejemo nekaj najznačilnejših razlik od običajnih oziroma starejših števcev. V jedru pričujočega dela poleg delovanja vhodnega dela, delilnikov in vrat beremo tudi o „motorju“ merilnika, mikrokrmilniku z 32-bitnim procesorjem ARM7. Glavna naloga, ki jo opravlja, je štetje frekvence. Za dosego kar najvišje algoritemske učinkovitosti ga programiramo v zbirniškem jeziku.

Uporaba mikrokrmilnika nam daje tudi to možnost, da lahko programsko popravimo nelinearen odziv čipov. To smo storili pri meritvi jakosti signala in s tem še izboljšali natančnost merilnika. Željen frekvenčni razpon merilnika je bil od 1 MHz do 1 GHz. Zadovoljivo natančnost smo dosegli v razponu od 2 MHz do 800 MHz. Omejitve predstavljajo predvsem čipi, ki smo jih imeli na voljo, nekatere od njih ženemo precej preko njihovih zagotovljenih zmogljivosti, zato smo lahko z rezultatom več kot zadovoljni.

Ključne besede: števec frekvence, delovanje števcev frekvence, merilnik jakosti signala, mikrokrmilnik LPC2138/01

Abstract

In this thesis we tried to develop a measurement equipment which has gone to oblivion. In many places it was substituted by cheap spectrum analysers which have integrated both frequency counter and amplitude meter. Being big and heavy box, spectrum analyser is not always practical as we often only need lightweight and efficient instrument for measuring frequency. Besides, old-fashioned design of high-frequency counters with slow prescaler, unsuitable input sensitivity, unsuitable input impedance and unsuitable probes results in impractical instrument. This thesis describes the state-of-the-art frequency counter and amplitude meter. It's main characteristics are high input sensitivity, broad frequency range and high speed measurement of frequency as well as amplitude.

In the first part, the design of frequency counters and basic operation is presented. Further, the main topic of this thesis, a state-of-the-art frequency counter, is described. It delivers quite a few improvements and modern solutions compared to regular counters. Some most typical characteristics are: operation without a prescaler, reset of the main gate is not needed, stable external oscillator and possibility of manipulation via external computer (PC). The core section of this thesis includes operation of the input part, the dividers, the gate and operation of the microcontroller with 32-bit processor ARM7. Its main task is to count frequency. For maximum algorithmic efficiency it is programmed in assembly language.

Use of a microcontroller also gives us the ability to correct the nonlinear response of chips by software. We did that with amplitude measurement and thereby improved instrument's accuracy. Our goal for the frequency range was from 1 MHz to 1 GHz. We reached satisfactory accuracy in the frequency range from 2 MHz to 800 MHz. Upper and lower limits are set by the capabilities of the chips we used. Given the fact that we drive some of the chips way over their guaranteed capabilities, the results are more than satisfactory.

Key words: frequency counter, operation of frequency counters, amplitude meter, microcontroller LPC2138/01

1 Uvod

Tako kot na marsikaterem drugem področju, je tudi k razvoju frekvenčnih števcev v veliki meri pripomogla vojska. V času hladne vojne so znanstveniki za razvoj atomske bombe potrebovali napravo, ki bi štela impulze (delce) ionizirajočega sevanja razpadajočih atomov. To je sprožilo razvoj frekvenčnih števcev [1].

Začetki števcev frekvence segajo v začetek petdesetih let 20. stoletja, ko je Philips razvil dekadno števno cev z imenom E1T. Znala je šteti električne impulze in prikazati njihovo vsoto. Delovala je na osnovi katodne cevi z elektrostatičnim odklonom. Glede na vsoto preštetih električnih impulzov se je žarek odmikal v vodoravni ravnini in se na fluorescentnem zaslonu s skalo od 0 do 9 prikazal ob ustreznem številu. Več števnih cevi, povezanih v kaskado, lahko prikazuje večmestna števila, pri čemer prva cev prikazuje enice, druga desetice, tretja stotice, in tako dalje [2], [3].

Ko je Hewlett-Packard leta 1952 predstavil svoj prvi digitalni elektronski števec, HP 524A, je bila to prelomnica na področju elektronske inštrumentacije. Merjenje frekvence do 10 MHz oziroma štetje zaporednih dogodkov v časovnem razmaku do 100 ns je postalo izvedljivo.

Odtlej so elektronski števci z nadaljnjim razvojem postajali vedno bolj uporaben in vsestranski pripomoček, ki je našel svoj prostor v laboratorijih, v proizvodnih linijah in servisnih centrih za telekomunikacije, elektroniko, v vojski, računalništvu, izobraževalnih ustanovah in v drugih industrijah. K širjenju nabora izdelkov in njihovih zmogljivosti na trgu števcev so med drugim prispevali: vzpon integriranih vezij, (višja hitrost delovanja in višja stopnja integracije MOS LSI vezij), ter kasneje mikroprocesor.

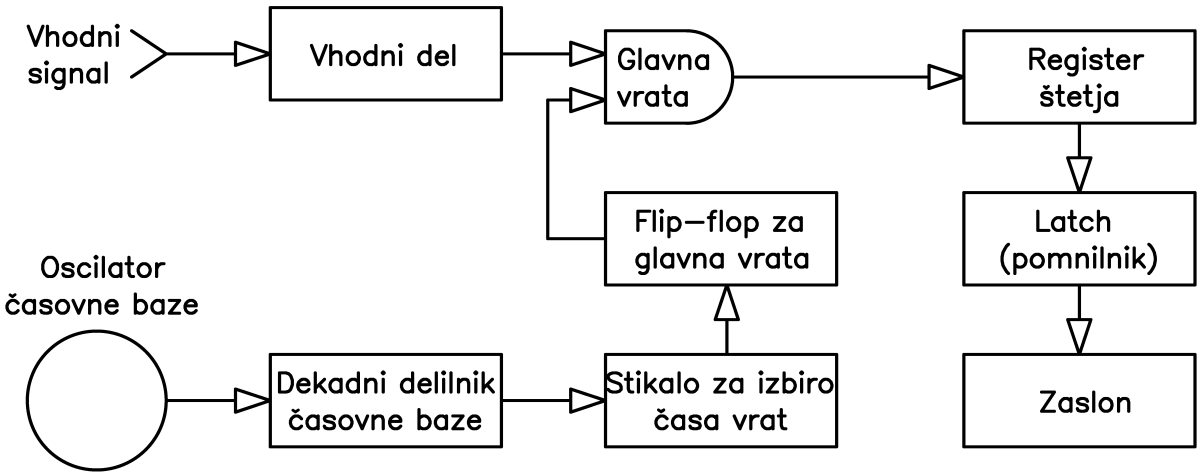
1.1 Osnova delovanja števcev frekvence

Frekvenco, *f*, ponavljajočega se signala lahko definiramo kot število impulzov signala v časovni enoti. To je predstavljeno z enačbo (1.1):

$$f = \frac{n}{t} \tag{1.1}$$

kjer je *n* število impulzov signala, ki se pojavi v časovnem intervalu, *t*. Če za *t* izberemo 1 sekundo, potem je frekvenca izražena v impulzih na sekundo oziroma v Hertzih (*Hz*).

Kot nakazuje enačba (1.1), števci frekvence merijo frekvenco tako, da preštejejo število impulzov, *n*, (naraščajoče ali padajoče fronte signala) v točno določenem časovnem intervalu, *t*. Osnovni blokovni načrt preprostega števca frekvence je narisan na Sliki 1.



Slika 1: Osnovni blokovni načrt števca frekvence

Vhodni signal se v vhodnem delu preoblikuje do oblike, ki je združljiva z notranjim vezjem števca, torej z zahtevami uporabljene družine logičnih vezij. Vhodni

del mora zagotoviti primerno vhodno občutljivost in vhodno impedanco, lahko je sklopljen izmenično ali enosmerno. Nenazadnje, vhodni del lahko vsebuje histerezo za čiščenje nizkofrekvenčnih signalov.

Preoblikovani signal, ki pride na vhod glavnih vrat, je niz pulzov, kjer vsak pulz predstavlja en dogodek vhodnega signala. Ko so glavna vrata odprta, pulzi potujejo skoznje in se seštejejo v registru štetja. Od časovne baze je odvisno, koliko časa bodo glavna vrata odprta, kar je določeno s časovnim intervalom, *t*. Iz enačbe (1.1) je razvidno, da je natančnost meritve frekvence odvisna od natančnosti, s katero je določena časovna baza oziroma časovni interval. Zato večina števcev za osnovo časovne baze uporablja stabilne kristalne oscilatorje s frekvenco 1, 5 ali 10 MHz (TCXO, OCXO).

Ločljivost, *L*, frekvenčnega števca je neposredno povezana s časom vrat, *t*, natančneje, ločljivost je recipročna vrednost časa vrat in se meri v Hz oz v s^{-1}, kar prikazuje enačba (1.2).

$$L = \frac{1}{t} \tag{1.2}$$

Za ločljivost 1 Hz je torej potreben čas vrat 1 s, za ločljivost 10 Hz desetkrat manj, torej 0.1 s in tako dalje. Ker je rezultat meritve celo število, bo vedno opletal med najmanj dvema sosednjima vrednostima [4].

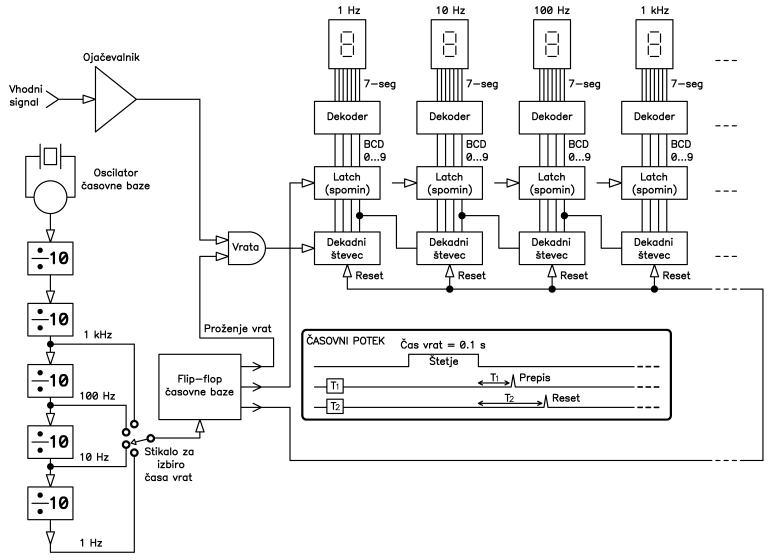
Delilnik časovne baze preoblikuje signal iz oscilatorja v niz pulzov, katerih frekvenca se izbira v dekadnih stopnjah s stikalom. Frekvenco časovne baze krmili flip-flop glavnih vrat. Časovni interval, *t*, oziroma čas vrat, je določen s periodo izbranega niza pulzov, ki izhajajo iz delilnika časovne baze [5].

Klasični števci frekvence so desetiški. Vsaka desetiška številka ima svoj desetiški števec, svoj vmesni pomnilnik (ang. latch) in svoj dekoder, ki številko iz BCD (Binary Coded Decimal) pretvori v obliko, primerno prikazovalniku: posamezne lučke, nixie cev oziroma sedem-segmentni prikazovalnik. Na vsak desetiški števec je napeljan reset. Vmesne pomnilnike proži signal za prepis.

Vezje s flip-flopom glavnih vrat proizvaja vse tri krmilne signale: vrata, prepis in reset. Zaporedje dogodkov pri meritvi je sledeče: v času, ko so vrata odprta, se preštejejo cikli vhodnega signala in se zapišejo v dekadni števec. Ko se vrata zaprejo, sledi prepis v spomin, nato se v dekoderju preoblikuje v primerno obliko za izpis na zaslon. Končno se sproži ukaz reset, ki ponastavi dekadne števce, da so pripravljeni na novo meritev.

Izmerjena frekvenca je nato numerično prikazana na zaslonu. Na primer, če je število impulzov, ki jih prešteje register štetja, enako 100.000, in je čas vrat enak eni sekundi, to pomeni, da je frekvenca vhodnega signala enaka 100.000 Hz.

Poenostavljen načrt s časovnim potekom signalov iz časovne baze prikazuje Slika 2.



Slika 2: Načrt števca in časovni potek signalov iz časovne baze

Omejitev vseh števcev frekvence je najvišja dopustna frekvenca štetja. Za merjenje še višjih frekvenc v radijskem in mikrovalovnem vezju potrebujemo dodatna vezja na vhodu. Sinusne periodične signale zelo visokih frekvenc lahko s pomočjo lokalnega oscilatorja znane frekvence mešamo na primerno nižjo medfrekvenco, obvladljivo s strani frekvenčnega števca. Proizvajalci merilne opreme so vhodni del števcev najprej opremili z ročno nastavljivim lokalnim oscilatorjem in kasneje še s samodejno nastavljivim. Nekateri mikrovalovni števci frekvence so znali svoja vrata sinhronizirati celo na frekvenco ponavljanja radarskih impulzov.

Frekvenčnemu števcu lahko dodamo tudi zunanji digitalni preddelilnik. Slaba lastnost preddelilnika je, da upočasnjuje meritev oziroma poslabša ločljivost meritve za svoj faktor deljenja frekvence (modulo). Preddelilnik je izdelan v hitri logični družini, običajno ECL, in je opremljen z lastnim vhodnim delom s predojačevalnikom. Preddelilnik ima običajno en sam izhod zadnje stopnje deljenja in nima nobenega krmilnega vhoda za reset. Ker notranje stanje večstopenjskega preddelilnika ni dostopno in nanj ne moremo vplivati, lahko kvečjemu upočasnimo meritev za faktor deljenja preddelilnika, oziroma se zadovoljimo z nižjo ločljivostjo meritve [6].

Ločljivost, L, je pri števcih frekvence s preddelilnikom poleg časa vrat, t, povezana tudi s številom stopenj deljenja, torej z modulom N, kar prikazuje enačba (1.3).

$$L = \frac{N}{t} \tag{1.3}$$

Iz enačbe (1.3) sledi, da je za isto ločljivost (npr. 10 Hz) pri manjšem modulu oziroma odsotnosti (N=1) preddelilnika, potreben čas vrat manjši kot pri večjem modulu. Z drugimi besedami, bolj kot delimo vhodno frekvenco, počasneje dobimo rezultat meritve, saj je čas vrat daljši, ob dejstvu, da obdržimo enako ločljivost [7].

Osnovni frekvenčni števec v tehnologiji TTL običajno dosega frekvenčno mejo okoli 50 MHz, v sodobnih CMOS tehnologijah nekaj sto MHz. Preddelilniki dosegajo frekvence od 1 GHz do preko 20 GHz. Modulo deljenja preddelilnika je lahko desetiški za preprosto uporabo oziroma dvojiški za najvišjo možno frekvenco štetja. Najvišja možna frekvenca štetja preprečuje, da bi vhodni ojačevalnik preddelilnika vseboval histerezo. Spodnja frekvenčna meja preddelilnika je zato omejena na 30 MHz ali več [8].

Pri marsikaterem števcu frekvence predstavljala težavo tudi slaba izvedba vhodnega dela, ki ima neprimerno vhodno impedanco in je opremljen z neustrezno sondo.

Pomankljivost večine frekvenčnih števcev je tudi to, da merilnik ne preverja niti uporabniku ne sporoča, ali je jakost vhodnega signala znotraj meja, v katerih vhodni del lahko deluje in števec zanesljivo šteje. Večina števcev se pri tem zanaša na opazovanje uporabnika, ki ocenjuje zanesljivost meritve iz opletanja rezultata, kar ne more biti objektivna meritev.

Klasični števci frekvence imajo to nezaželeno značilnost, da kljub konstantnemu vhodnemu signalu, torej ko spreminjanja frekvence ni, preštejejo in izpišejo en impulz, kar ni točno. Razlog za to se skriva v izvedbi vrat. Vhoda v vrata sta dva, prvi je neznana frekvenca, drugi pa signal iz časovne baze, ki določa interval, v katerem so vrata odprta. V primeru, ko je vhodni signal konstanten, brez impulzov in ga na primer predstavlja logična enica, „1“, potem vsakokrat v času, ko so vrata odprta, register štetja našteje en impulz (1 Hz). Čeprav tega v resnici ni, se izpiše na zaslonu, kar je napaka.

2 Sodoben frekvenčni števec za območje radijskih frekvenc

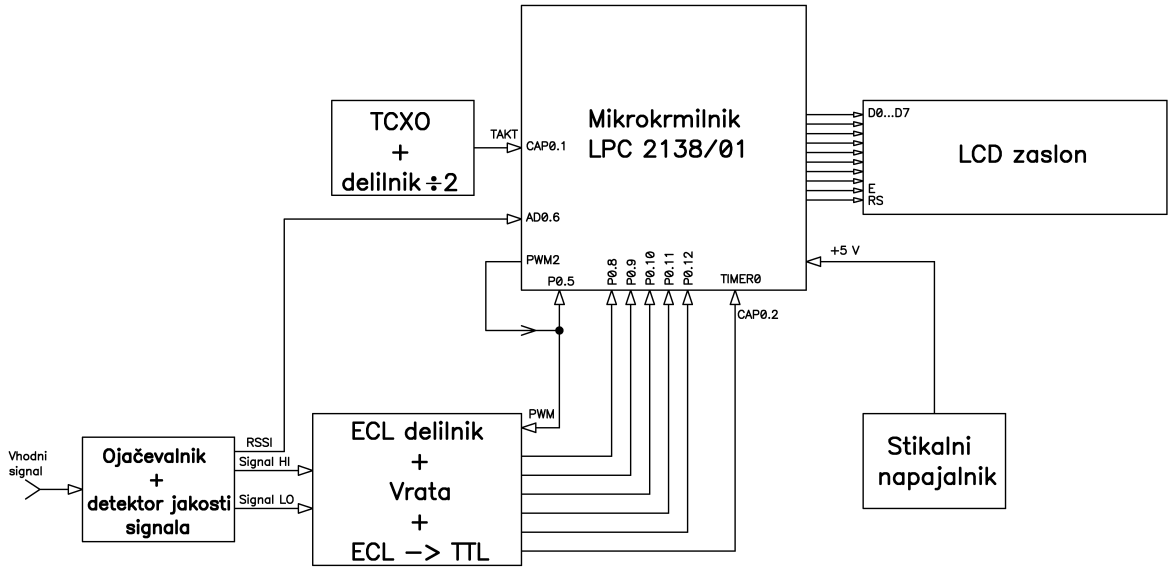
V tej nalogi smo iskali izvedbo frekvenčnega števca, ki bi dosegel čim višjo vhodno frekvenco v neposrednem štetju, torej brez uporabe preddelilnikov. Hkrati naj merilnik preverja in prikazuje tudi jakost vhodnega signala kot potrditev pravilnosti meritve.

2.1 Zasnova sodobnega števca

Sodoben frekvenčni števec, ki je predmet te naloge, deluje brez preddelilnika, vendar vlogo deljenja vhodne frekvence prav tako igrajo hitra ECL vezja. Za pomembno razliko od običajnih števcev z ECL preddelilnikom, ta v prvi stopnji poleg delilnika vsebuje tudi vrata. Vse ostale naloge frekvenčnega števca opravlja mikrokrmilnik z 32-bitnim procesorjem ARM7 in bogatim naborom vhodno/izhodnih enot.

Mikroprocesor omogoča, da števca ni potrebno ponastavljati za vsako meritev. Rezultat štetja je preprosto razlika med starim in novim stanjem števca, ki jo izračuna mikroprocesor. Obdelava v mikroprocesorju poteka v dvojiški obliki, tudi vsi števci so dvojiški, kar se odraža v hitrejšem delovanju inštrumenta in višji gornji frekvenčni meji. Rezultat meritve se v desetiško obliko pretvori šele tik pred izpisom na zaslon.

Osnovni blokovni načrt sodobnega frekvenčnega števca je prikazana na Sliki 3.



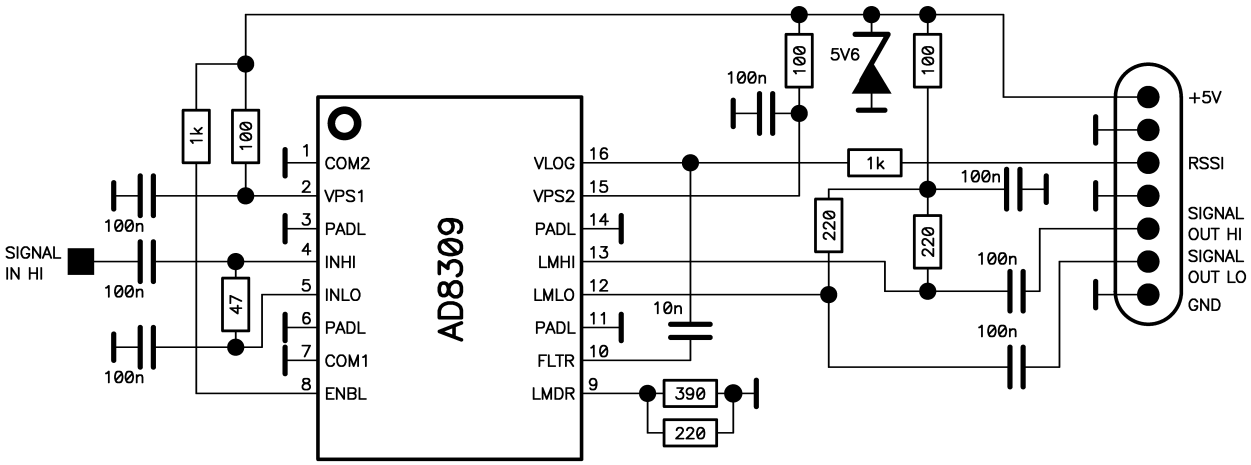
Slika 3: Osnovna blokovna shema sodobnega frekvenčnega števca

2.2 Vhodni del števca

Prvi člen frekvenčnega števca je čip AD8309, ki vsebuje ojačevalnik-omejevalnik in logaritemski detektor jakosti vhodnega signala. Ojačevalnik-omejevalnik krmili digitalni del števca z oblikovanim signalom znane jakosti. Logaritemski detektor meri jakost vhodnega signala v razponu 100 dB. Delovanje ojačevalnika-omejevalnika in logaritemskega detektorja je zagotovljeno v razponu frekvenc od 5 MHz do 500 MHz. S popravki v programski opremi mikrokrmilnika smo uspeli doseči zadovoljivo delovanje vezja AD8309 v razponu od 2MHz do 800MHz [9].

Neznani signal iz sonde z zanko oziroma drugega nizkoimpedančnega vira pripeljemo preko SMA vhoda na ploščico s čipom AD8309. Kondenzatorja 100 nF poskrbita za izmenični sklop in pravilno delovno točko AD8309. Upor 47 Ω poskrbi za vhodno zaključitev približno 50 Ω za izmenične signale. Inštrument je namenjen

meritvam (približno) sinusnih signalov, zato izmenični sklop zadostuje in enosmernega ne potrebujemo. Podroben načrt vezja je predstavljen na Sliki 4.



Slika 4: Podroben načrt vezja čipa AD8309

Nato se v čipu AD8309 signal ojača ter se mu hkrati izmeri logaritemska vrednost. Velikost ojačanja se nastavlja z uporom na pinu LMDR (pin št. 9). Po poskusih se je za optimalno vrednost pokazala upornost 140 Ω (390 Ω || 220 Ω). Zener dioda 5.6 V ima vlogo prenapetostne varovalke. Pri nastavljanju velikosti ojačanja je potrebno paziti, da ne nastavimo previsoke vrednosti, saj se potem (preveč ojačan) izhodni signal sklopi nazaj na vhod in vezje lahko prične samooscilirati.

2.3 Delilniki ECL in vrata

Ojačan merjeni signal pripeljemo na ECL vezja. Pred deljenjem signal zopet izmenično sklopimo preko 100 nF kondenzatorjev ter ga ponovno ojačamo z ECL čipom MC10EL16 [10]. Ponovno ojačanje in omejevanje je potrebno, ker signala s čipom AD8309 ne moremo ojačati do ravni, potrebne za krmiljenje nadaljnjih ECL flip-flopov. Ti namreč zavoljo hitrejšega delovanja ne vsebujejo ojačevalnika, za razliko od nekaterih drugih delilnikov.

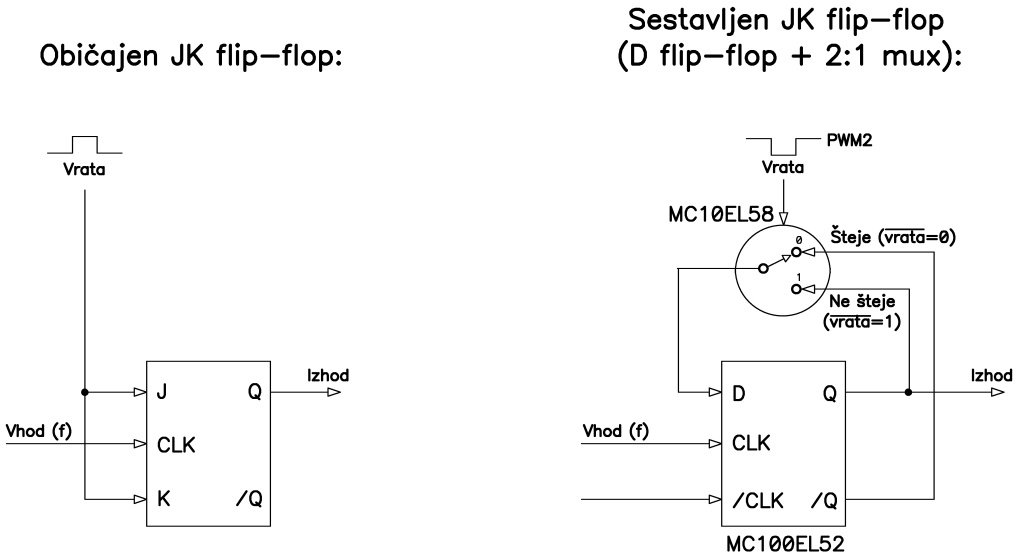
Merjeni signal nato pripeljemo na čip MC100EL52 (D flip-flop), ki skupaj s čipom MC10EL58 (2:1 multiplekser) tvori navidezni JK flip-flop. Vhod D čipa MC100EL52 je vezan na izhod multiplekserja. Izhoda Q oziroma $\overline{Q}$ D flip-flopa krmilita vhoda multiplekserja in ostale flip-flope v števni verigi. Merjeno frekvenco pripeljemo na vhoda CLK in $\overline{CLK}$ D flip-flopa, ker diferencialno (dvofazno) krmiljenje omogoča višjo frekvenco delovanja.

Signal vrat frekvencmetra proizvaja notranji časovnik mikrokrmilnika na izhodu PWM2. Vrata peljemo na vhod Select multiplekserja, ki zaustavlja oziroma sprošča štetje celotne verige, torej se obnaša kot (negirana) vzporedna vezava vhodov J in K. Ko je PWM2 oziroma Select visok (logična enica), D flip-flop ob vsakem taktu prepisuje staro stanje. D flip-flop tedaj ne šteje, je zaustavljen in posledično tudi celotna veriga flip-flopov za njim. Ko je PWM2 oziroma Select nizek (logična ničla), D flip-flop ob vsakem taktu menja stanje. D flip-flop tedaj šteje in deli frekvenco z dva, kar potem šteje naslednji flip-flop v verigi.

Opisana rešitev omogoča točnejše štetje od običajnih IN vrat na vhodu števca. V primeru, da je vhodni takt zaustavljen v stanju logične enice ali logične ničle (vseeno), JK flip-flop z zaustavljenim taktom nikoli ne šteje, ne glede na impulze na vhodih JK oziroma na vhodu Select opisane navidezne izvedbe JK flip-flopa. Posledica tega je, da rezultat meritve s takšnim števcem bolj točen in manj opleta.

Navidezni JK flip-flop, sestavljen iz dveh ločenih ECL čipov, ima tudi eno slabo lastnost. Zakasnitve posameznih ECL čipov se seštevajo. Gornja frekvenčna meja MC100EL52 naj bi bila po podatkih proizvajalca 2.8 GHz. Sestavljeni navidezni JK flip-flop doseže najvišjo frekvenco štetja le 1.2 GHz, kar pa je še vedno boljše od vrednosti, ki jo izračunamo iz navedenih zakasnitev v podatkih proizvajalca [11], [12].

Načrt vezave čipov MC100EL52 in MC10EL58 ter primerjava med običajnim JK flip-flopom ter sestavljenim navideznim sta predstavljena na Sliki 5.

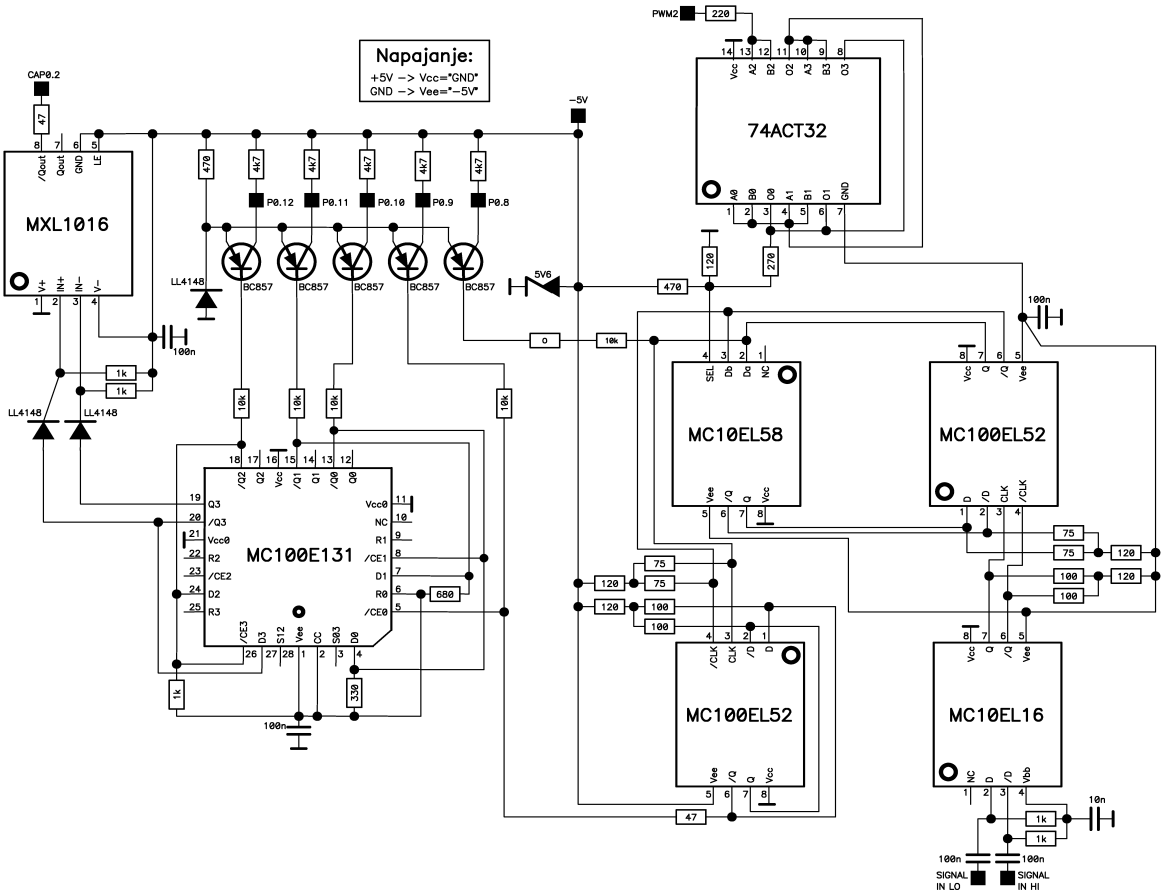


Vrata	J	K	TAKT	Qn
Ne šteje (0)	0	0	↑	Qn−1
/	0	1	↑	0
/	1	0	↑	1
Šteje (1)	1	1	↑	$\overline{Qn-1}$

$\overline{Vrata}$	TAKT	Qn
Ne šteje (1)	↑	Qn−1
Šteje (0)	↑	$\overline{Qn-1}$

Slika 5: Primerjava izvedbe običajnega in sestavljenega navideznega JK flip-flopa ter načrt vezave slednjega

Signal PWM2, ki proži vrata, je potrebno zaradi različnih napetostnih nivojev delovanja ECL vezij in mikrokrmilnika prej ustrezno prilagoditi. To nalogo opravi čip 74ACT32, ki lahko sprejme različne vhodne napetosti, izhodi iz njega pa so točno definirane napetosti. Tako sprejme signal iz procesorja na približno 3.3 V in ga ojača na 5 V, kolikor znaša njegova napetost napajanja [13]. Nato napetost signala s pomočjo uporov (270 Ω, 120 Ω, 470 Ω) znižamo na napetost 4.2 V (točno 4.25 V), kar predstavlja enico za (P)ECL vezja, medtem ko nivo napetosti okrog 3 V zanje pomeni ničlo. Na ta način opravimo pretvorbo iz TTL nivoja na ECL nivo. Slika 6 prikazuje podroben načrt celotnega vezja ECL delilnikov.



Slika 6: Načrt vezja ECL delilnikov

Prvemu flip-flopu z MC100EL52 in MC10EL58 sledi asinhrona (ang. ripple clocking) veriga števcev: pet ECL flip-flopov: še en MC100EL52 in štirje flip-flopi vezja MC100E131 [14] ter števec TIMER0 v mikrokrmilniku LPC2138/01. Skupno vsebuje števec šest ECL flip-flopov, ki delijo vhodno frekvenco 1 GHz s 64, kar daje približno 15 MHz oziroma ravno toliko, kolikor je TIMER0 v mikrokrmilniku še sposoben obdelati.

TIMER0 krmilimo preko vhoda CAP0.2, ki zahteva TTL ravni signalov. Pretvorbo iz ECL na TTL opravi ultra-hiter primerjalnik MXL1016 s frekvenčno mejo 100 MHz [15]. Ker ECL flip-flopov ne uporabljamo kot preddelilnik, temveč kot števec, moramo mikrokrmilniku zagotoviti vpogled v vsebino vseh ECL stopenj, ne samo zadnje. Na srečo mikrokrmilnik opazuje vse stopnje razen zadnje v zaustavljenem stanju, zato je dostop do njih lahko razmeroma počasen.

Izhodne signale prvih petih ECL flip-flopov: obeh MC100EL52 in prvih treh flip-flopov iz MC100E131 na TTL ravni pretvori pet razmeroma počasnih ojačevalnikov s PNP tranzistorji BC857. Mikrokrmilnik odčita te signale na vhodih P0.8, P0.9, P0.10, P0.11 in P0.12. Mikroprocesor omogoča izračun frekvence kot razliko dveh odčitkov števca, torej ponastavljanje števca za novo meritev ni potrebno, kakor tudi ni potreben reset ECL flip-flopov.

2.3.1 Vroča masa, vhodne logične ravni in zaključitve izhodov ECL vezij

ECL vezja delujejo na nestandardnih napetostnih nivojih, običajno v negativnem načinu, tako da je pozitivni pol napajalne napetosti vezan na maso, za razliko od večine ostalih vezij, kjer je na maso vezan negativni pol napajalne napetosti. Razlog za takšno vezavo tiči v tem, da se na tak način zmanjša vpliv nihanja napetosti na logičnih nivojih, saj so ECL vezja bolj občutljiva na šum na vhodu V_{CC} in so pretežno imuna na šum na vhodu V_{EE}. Običajno je pri vezjih masa tista, ki zagotavlja najnižje motnje, zato ECL čipi delujejo s pozitivno (vročo) maso.

Delilniki ECL so vgrajeni na dvostransko tiskano vezje, kjer ena stran ni jedkana in predstavlja ravnino mase za ECL vezja, povezano na +5 V za vse ostale gradnike frekvenčnega števca. ECL logično enico predstavlja raven -0.8 V pod maso V_{CC} oziroma +4.2 V nad V_{EE}, logično ničlo pa -2 V pod maso V_{CC} oziroma +3 V nad V_{EE} [16], [17]. Pri uporabi ECL vezij moramo paziti predvsem na pravilno napetost logične enice, ker previsoka vhodna napetost pošlje tranzistorje v nasičenje, kar zelo upočasni delovanje.

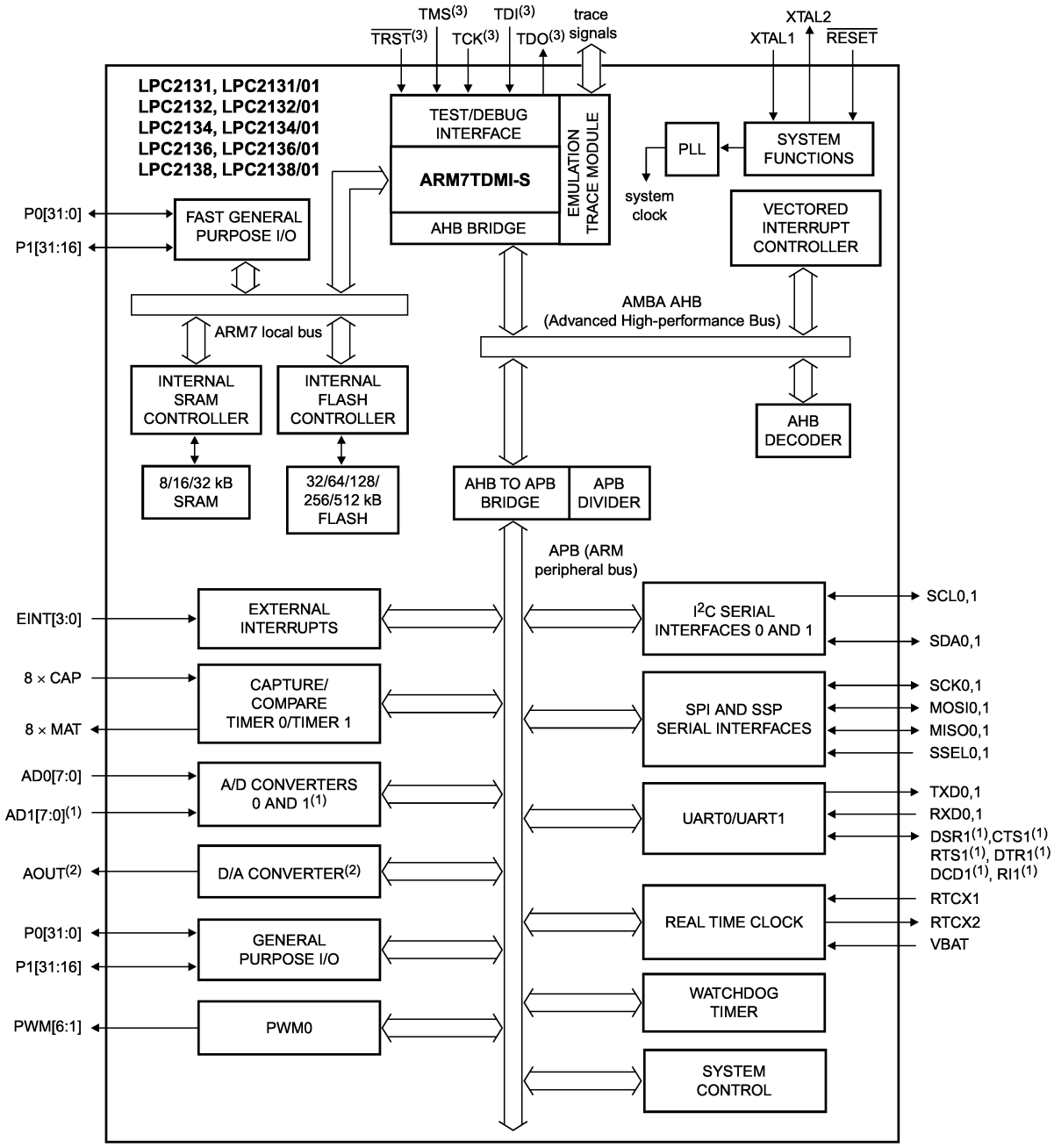
Izhodi ECL vezij so odprti emitorji NPN tranzistorjev, ki zahtevajo zunanje zaključitvene upore. Najzahtevnejše so zaključitve pri najvišjih frekvencah, zato imajo vezja MC10EL16, MC100EL52 in MC10EL58 zaključena oba izhoda z upori zelo nizkih vrednosti. Flip-flopi iz MC100E131 imajo zaključene samo uporabljene izhode in to na uporih čedalje višjih vrednosti, skladno z zniževanjem frekvence delovanja. Neuporabljeni izhodi MC100E131 so nezaključeni.

2.4 Mikrokrmilnik LPC2138/01

Srce frekvenčnega števca predstavlja 32-bitni mikrokrmilnik LPC2138/01 s procesorjem ARM7TDMI-S, ki izvršuje nabor ukazov ARMv4T [18]. Sodobni mikrokrmilniki, kot je ta, imajo na voljo mnogo več funkcij, kot jih sploh lahko uporabimo pri isti nalogi. Tudi če damo mikrokrmilniku zelo obširno nalogo in se potrudimo izkoristiti karseda veliko število funkcij, jih še vedno vsaj polovica ostane neizrabljenih.

Mikrokrmilnik LPC2138/01 poleg procesorja vsebuje statični RAM, pomnilnik Flash (EEPROM), dva analogno-digitalna pretvornika (ADC), od katerih ima vsak do 16 analognih vhodov, en digitalno-analogni pretvornik (DAC), ki priskrbi analogne izhode, dva 32-bitna števca zunanjih dogodkov (TIMER0, TIMER1), PWM enoto, 32 kHz uro za notranji takt, serijske vmesnike, med njimi dva vrste UART, enoto za zunanje prekinitve, vektorski krmilnik prekinitev in vektorske naslove, do sedeminštirideset vhodno-izhodnih enot GPIO (General-Purpose In/Out) do napetosti 5 V, CPU uro s programirljivo fazno sklenjeno zanko (PLL) do največ 60 MHz, oscilator z zunanjim kristalom v razponu med 1 MHz in 30 MHz. Procesor deluje pri napetosti 3.3 V oziroma v razponu med 3.0 V in 3.6 V.

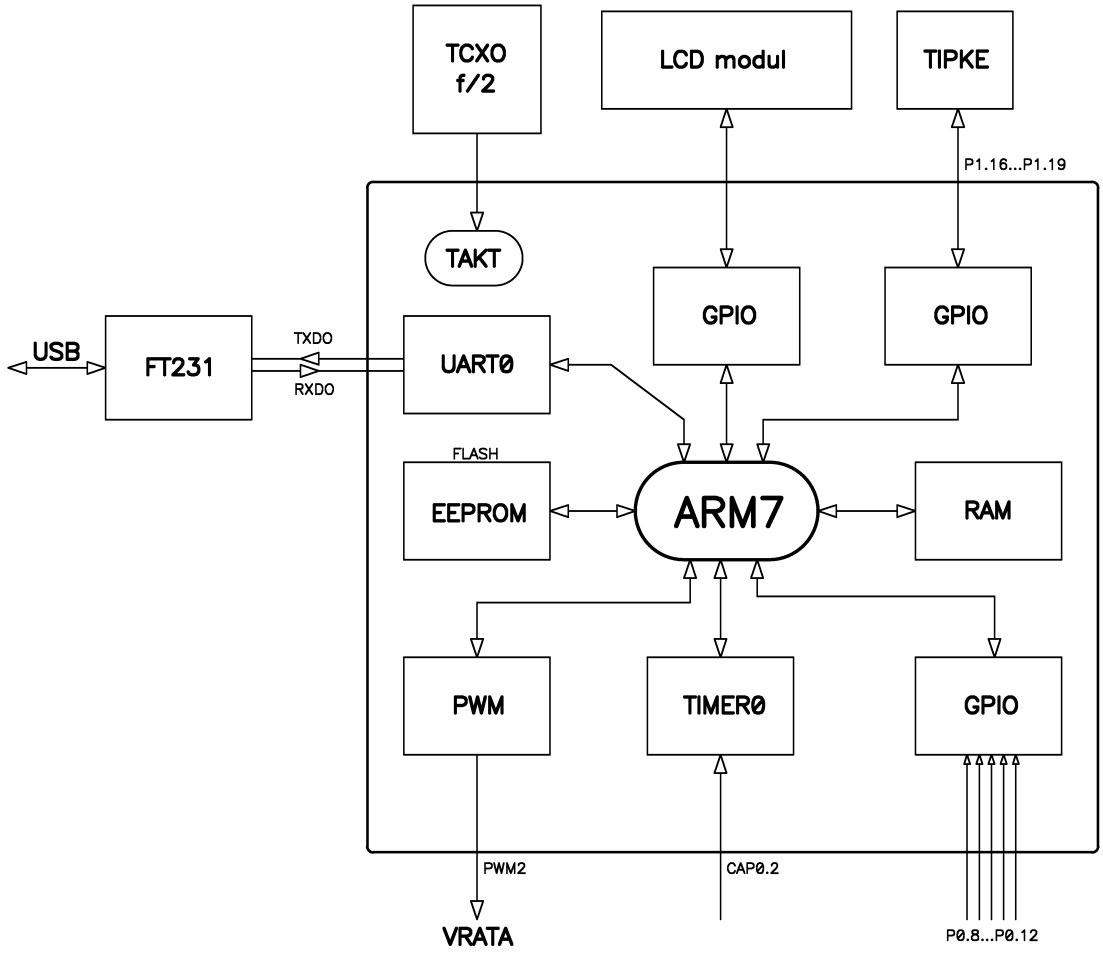
Blokovni načrt mikrokrmilnika z vsemi opisanimi lastnostmi vidimo na Sliki 7.



Slika 7: Blokovni načrt mikrokrmilnika LPC2138/01 [19]

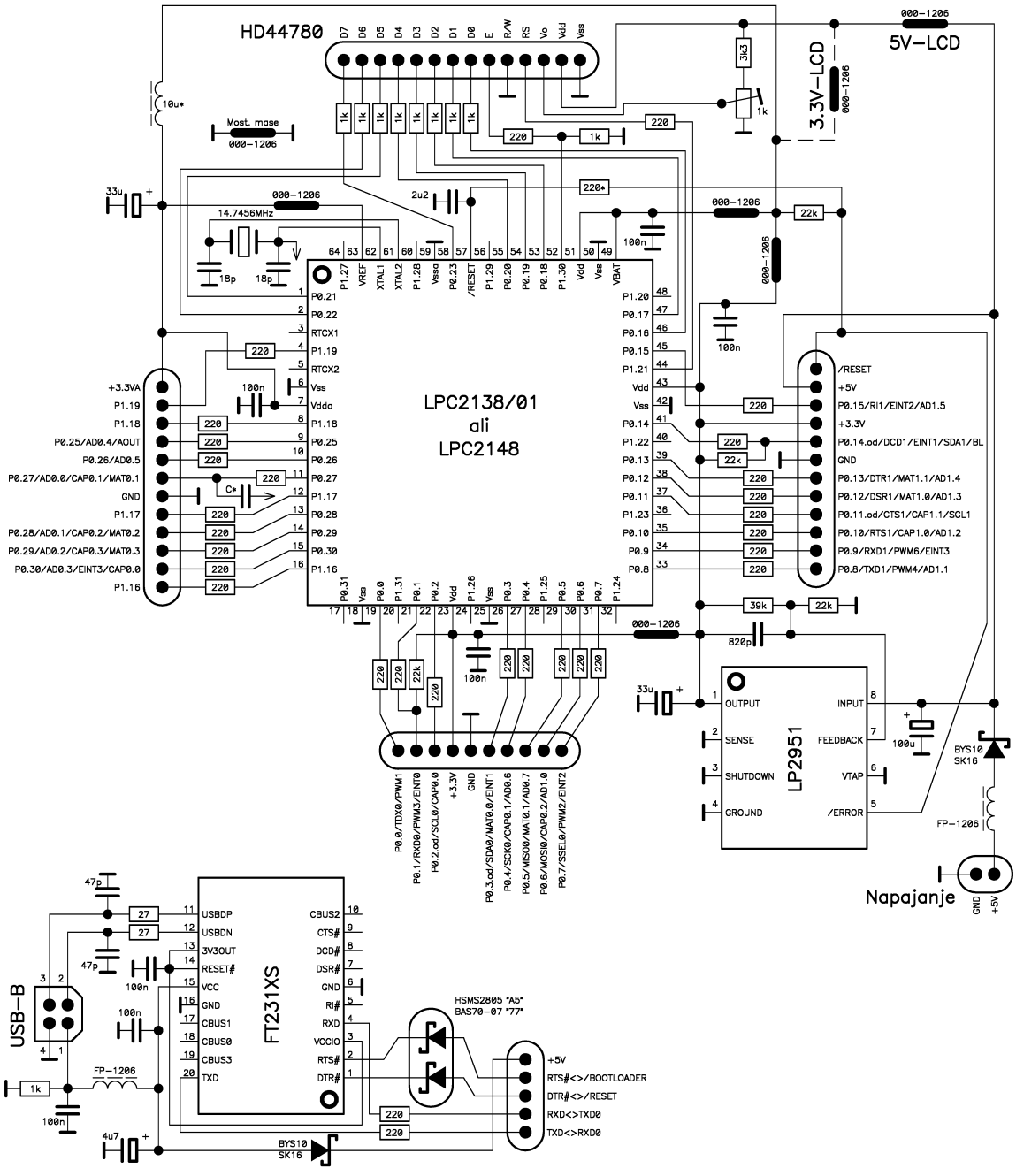
Za števec frekvence uporabimo le del nabora funkcij, ki jih ponuja mikrokrmilnik LPC2138/01. Mednje sodijo procesor ARM7TDMI-S, statični RAM, pomnilnik EEPROM, serijski vmesnik UART0, 16 vhodno-izhodnih enote GPIO, en 32-bitni števec zunanjih dogodkov TIMER0 in PWM enota.

Poleg naštetih funkcij smo za potrebe frekvenčnega števca dodali še nekaj zunanjih enot. Te so zunanji temperaturno kompenzirani kristalni oscilator (TCXO) s frekvenco 38.88 MHz, zaradi boljše stabilnosti, USB vmesnik za komunikacijo z računalnikom, LCD krmilnik z zaslonom ter štiri tipke za vhodne ukaze. Vse uporabljene enote mikrokrmilnika ter zunanje enote so prikazane na Sliki 8.



Slika 8: Blokovni načrt uporabljenih enot mikrokrmilnika LPC2138/01 in zunanje enote

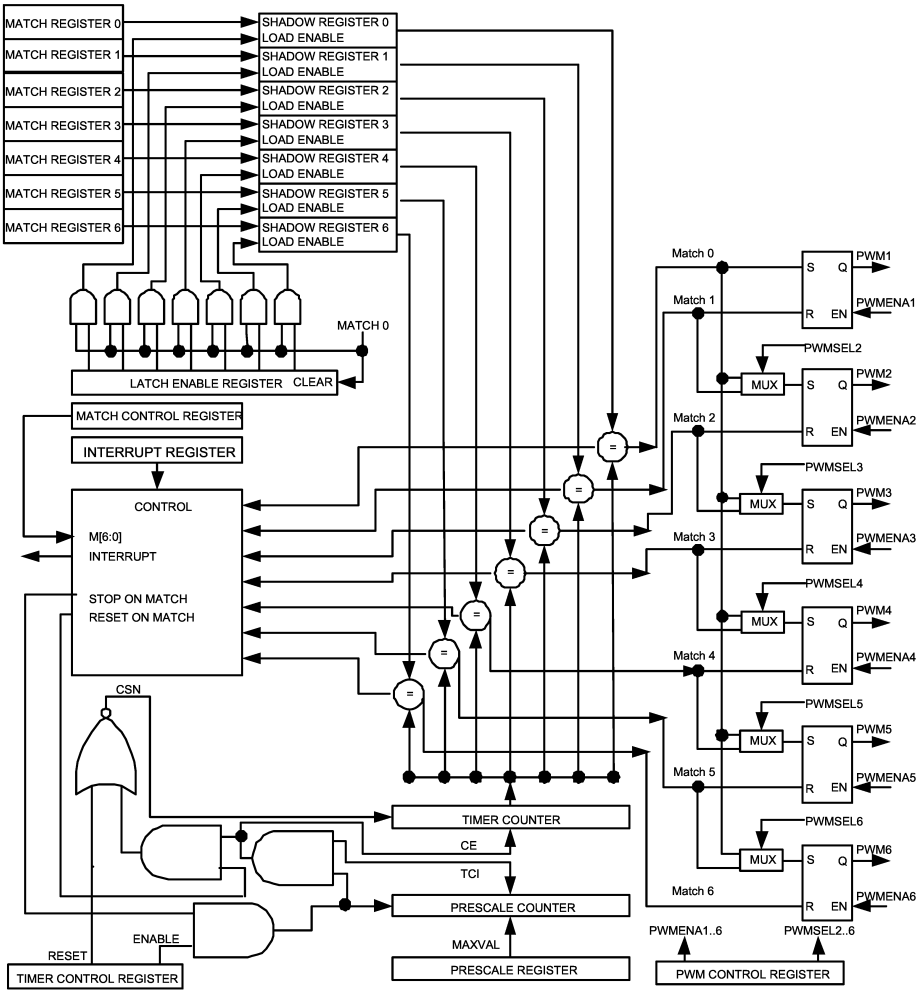
Slika 9 prikazuje podroben načrt vezave mikrokrmilnika LPC2138/01, z LCD krmilnikom HD44780, USB vmesnikom FT231XS in regulatorjem napetosti LP2951.



Slika 9: Načrt vezave mikrokrmilnika LPC2138/01 [18]

2.4.1 PWM – signal vrat

Za signal vrat števca frekvence smo uporabili signal PWM iz mikrokrmilnika. PWM (Pulse Width Modulator) temelji na števcu (ang. Timer), ki šteje cikle zunanje ure (ang. Peripheral Clock-PCLK) in proizvaja prekinitve oziroma druge dogodke, ko se vrednost števca ujema z enim od sedmih primerjalnih registrov (ang. Match Register 0...6) [20]. Vsak primerjalni register ima še svojo senco (ang. Shadow Register), da lahko vse primerjalne registre nastavimo istočasno na novo vrednost. Na Sliki 10 je prikazan načrt PWM enote.



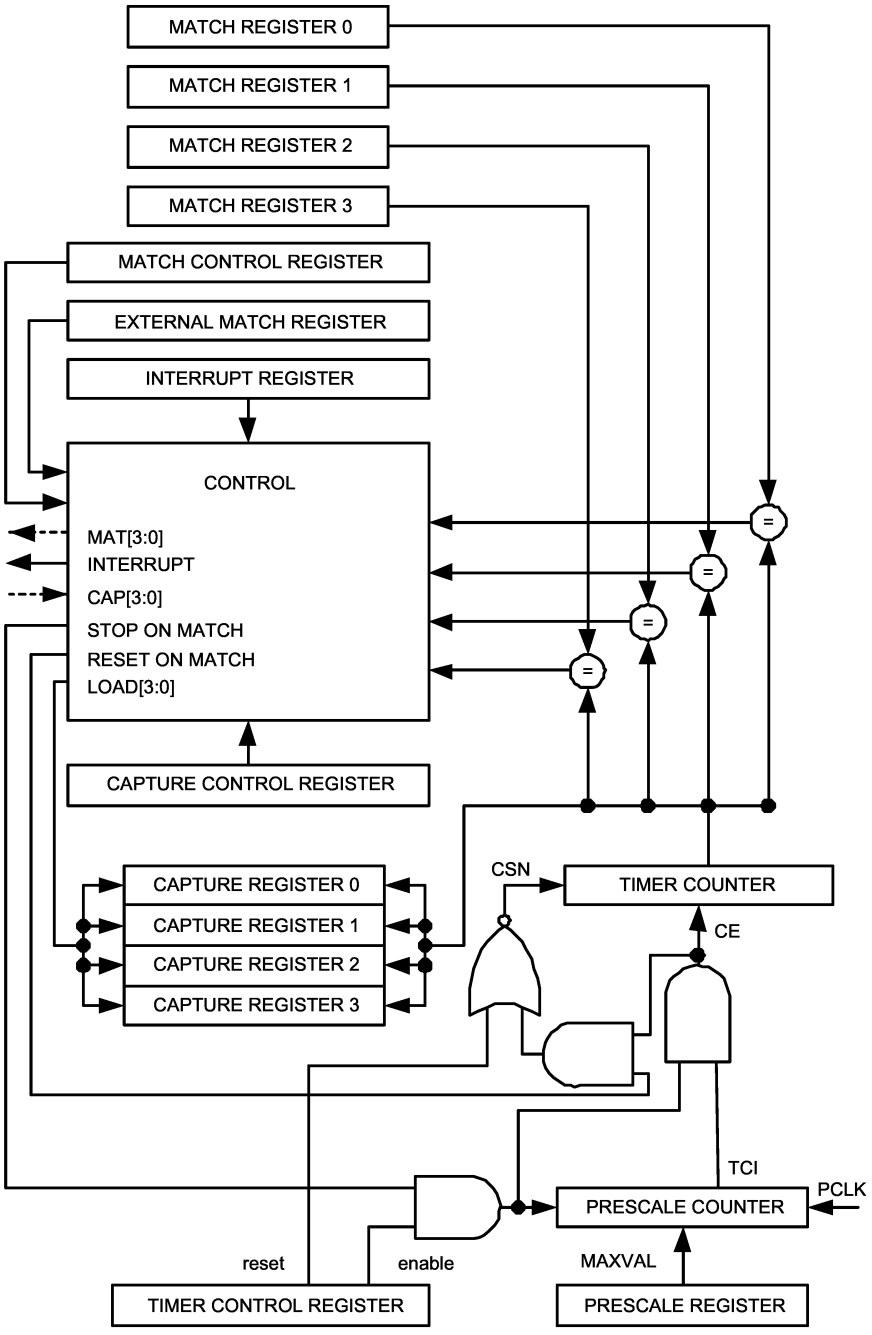
Slika 10: Načrt PWM enote [19]

Na ujemanju s primerjalnimi registri deluje tudi PWM. Ta omogoča upravljanje nad naraščajočimi in padajočimi frontami izhodov PWM, kar pomeni široko uporabnost. Dve primerjavi sta potrebni za izhod PWM signala, ki ima nadzor nad eno od front, naraščajočo ali padajočo. En register (MR0) nadzira štetje in ponastavi števec ob ujemanju. Drugi upravlja s položajem ene od front izhoda PWM.

Za izhod PWM signala, ki ima nadzorovani obe fronti, tako naraščajočo kot tudi padajočo, so potrebne tri primerjave. Za nadzor štetja in ponastavljanje števca skrbi MR0. Druga dva registra (MR1 in MR2) določata položaj front PWM signala, kjer en od registrov proži naraščajoče fronte, drugi pa padajoče. Prav slednjo možnost uporabljamo v opisani izvedbi frekvenčnega števca, saj omogoča večjo svobodo pri izvedbi programske opreme.

2.4.2 Števec TIMER0

Neznano frekvenco štejemo s funkcijo Timer. To lahko nastavimo, da šteje takte notranje ure PCLK ali takte zunanje ure oziroma zunanjega signala. Slednjo možnost uporabljamo v našem frekvenčnem števcu. Vhodne vrednosti za meritev frekvence so prvih 5 bitov iz posameznih stopenj ECL delilnikov, zadnji, šesti bit pa pripeljemo na vhod TIMER0. Ta je sestavljen iz 32-bitnega števca (ang. Timer Counter) in 32-bitnega registra preddelilnika (ang. Prescaler Register). Blokovni načrt števca Timer je na Sliki 11.



Slika 11: Blokovni načrt števca Timer [19]

Preddelilnik je neuporabljen: premoščen je tako, da deli z 1. Šesti bit ECL delilnika štejemo na obeh frontah z nastavitvijo Counter Control Register (CTCR). V

Counter Control Register bita 1:0 (Counter/Timer Mode) nastavimo na vrednost 11, tako določimo štetje na obeh frontah. Z bitoma 3:2 (Count Input Select), ki ju nastavimo na 10, določimo vhod CAP0.2. Match registrov pri štetju frekvence ne potrebujemo.

Funkcija Timer ima svoje omejitve. Ena od njih je največja vrednost vhodne frekvence zunanjega signala. Ta mora biti manjša od ene četrtine ure PCLK. Za meritev enga takta vhodne frekvence na vhodu CAP sta namreč potrebna dva zaporedna takta notranje ure PCLK. Frekvenco ure PCLK programsko nastavimo. Osnovni takt zagotavlja zunanji kristalni oscilator s frekvenco 38.88 MHz, ki jo delimo z dva. Na mikrokrmilnik pripeljemo takt 19.44 MHz, za frekvenco PCLK ga množimo s tri in dobimo frekvenco notranje ure 58.32 MHz. Ker je zahteva za največjo vrednost vhodne frekvence največ ena četrtina frekvence PCLK, je naša omejitev za vhodno frekvenco 14.58 MHz. Najvišja vhodna frekvenca, ki jo lahko mikrokrmilnik obdela, je torej 933 MHz (14.58 MHz · 64) [19], [21].

Program prebere 5 bitov iz ECL delilnika in 32bitov iz Timer0. Vsebino Timer0 zamakne za pet dvojiških mest v levo in zavrže gornjih 5 bitov. Na spodnjem koncu program doda 5 bitov iz ECL delilnika za 32-bitno meritev. Program ne ponastavlja števcev, pač pa rezultat štetja izračuna tako, da od nove vrednosti štetja odšteje staro vrednost.

2.4.3 Meritev jakosti signala

Poleg frekvence naš inštrument meri tudi jakost signala. To je dober pokazatelj, kdaj je jakost merjenega signala v okvirih, kjer inštrument pravilno deluje.

Jakost signala izmeri čip AD8309, ki je hkrati vhodni ojačevalnik ter omejevalnik in je podrobneje opisan v poglavju *2.2. Vhodni del števca*. Izhod RSSI (Received Signal Strength Indicator) čipa AD8309, kjer se nahaja izmerjena jakost signala, izražena v napetosti 0 V do 3.3 V [9], peljemo na vhod AD0.6

mikrokrmilnika. Vhod AD0.6 je del analogno-digitalnega pretvornika (ang. ADC – Analog-to-Digital Converter), kateremu osnovni takt zagotavlja PCLK. V pretvornik je vključen programirljiv delilnik, ki nastavi takt (uro) na največ 4.5 MHz, kolikor zahteva postopek pretvorbe [19]. Za popolnoma točno pretvorbo je potrebnih enajst taktov. V našem primeru osnovni takt nastavimo na 4 MHz.

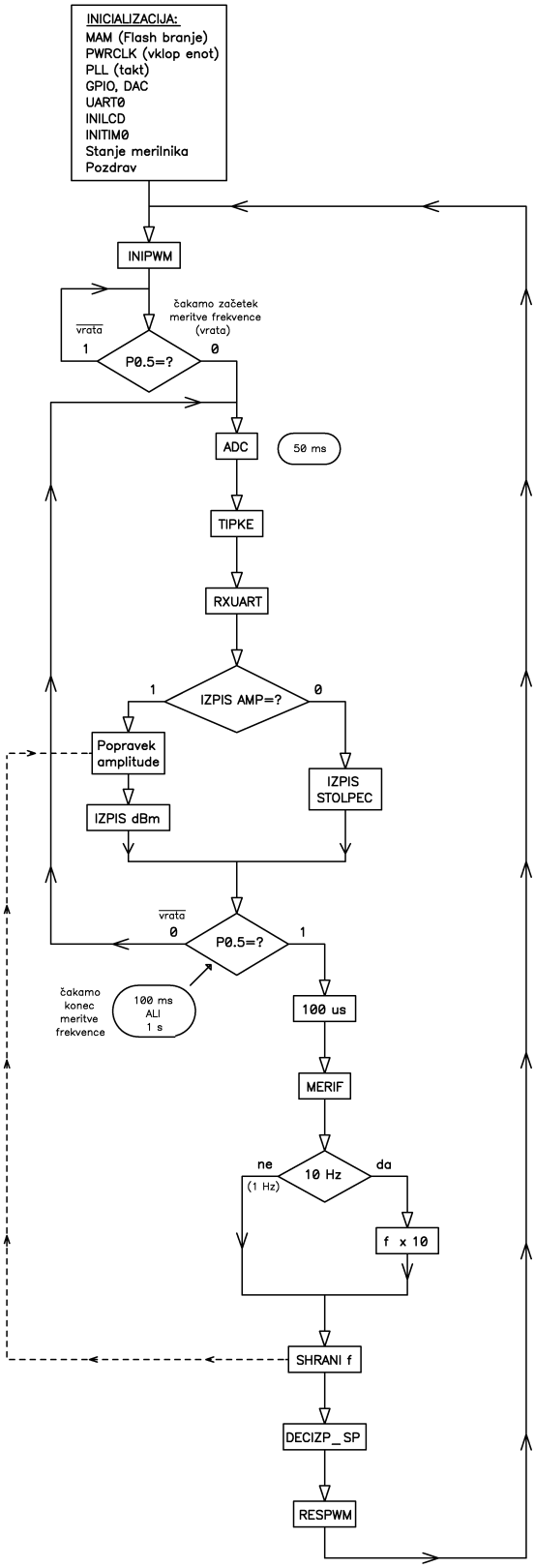
Logaritemski detektor čipa AD8309 meri jakost vhodnega signala v razponu 100 dB [9]. V našem primeru ga uporabljamo v razponu 75 dB, od -60 dBm do +15 dBm. Pri tem jakosti -60 dBm ustreza napetost 0.7 V in jakosti +15 dBm ustreza napetost 2.2 V.

Za bolj točno meritev jakosti signala naredimo povprečenje 2^{14}=16384 meritev, kar traja približno 50 ms. Povprečenje se izvrši med potekom meritve frekvence, ki traja bodisi 100 ms bodisi 1 s, odvisno od izbranega časa vrat.

2.4.4 Programska koda

Program, ki teče v procesorju in opravlja nalogo merjenja frekvence in jakosti signala, je napisan v zbirnem programskem jeziku zbirniku. Ta je nizkonivojski jezik druge ravni (najnižja raven je strojna koda). Sestavljen je iz ukaznih kod – mnemonikov, ki so v procesorju definirani po ISA (Instruction Set Architecture) arhitekturi. Vsak mnemonik predstavlja en ukaz ali operacijo, s katerimi dostopamo do posameznih registrov procesorja oziroma vplivamo na podatke v njih. Programsko kodo tako pišemo na ravni operacij med registri, s čimer lahko ob dobrem znanju programiranja dosežemo visoko algoritemsko učinkovitost. Za razliko od višjih programskih jezikov, kot je na primer jezik C++, ki potrebujejo prevajalnik, zbirniški jezik prevajalnika ne potrebuje. V strojno kodo ga pretvori pomožni program Zbirnik (ang. Assembler) [22]. Poleg pretvarjanja v strojno kodo Zbirnik opravi tudi nekaj preprostih, a duhamornih opravil: preračuna razdalje do label, preračuna razdalje do 32-bitnih konstant v LTORG, vstavlja makroje, omogoča zapis pogosto uporabljanih konstant z labelami in podobno.

Na začetku programske kode določimo konstante mikrokrmilnika, kot so frekvenca oscilatorja TCXO (*KVARC*), konstante za PWM (*PWMR0, PWMR1, PWMR1A*), takt za analogno-digitalni pretvornik (*ADC_DIV*) in druge. Nato dodelimo pomnilnik različnim spremeljivkam (*SKLAD, JAKOST, STANJE, ...*). Zatem inicializiramo različne enote mikrokrmilnika: Flash branje MAM, vklop enot PWRCLK, nastavitev takta vhodno/izhodnih enot APBDIV, fazno sklenjeno zanko za frekvenco jedra PLL, vhodno-izhodne enote GPIO, digitalno-analogni pretvornik DAC, vmesnik UART0, LCD krmilnik HD44780 in časovnilk TIMER0. Določimo začetno stanje inštrumenta ter pozdravni napis. Pred tem definiramo rutini ZNAK in UKAZ za izpis na LCD zaslon ter rutine za različne zakasnitve: 1 µs, 100 µs, 4 ms, 100 ms, 1 s. Diagram poteka programske kode je predstavljen na Sliki 12.



Slika 12: Diagram poteka programske kode za frekvenčni števec

Naloga glavne zanke programa je to, da se več različnih opravil izvaja istočasno. Ta opravila so meritev amplitude, meritev frekvence ter vnos ukazov s tipk oziroma UART0. Različna opravila se sicer izvajajo znotraj vhodno/izhodnih enot PWM, TIMER0, AD0, GPIO, UART0, a zahtevajo različno hitro ukrepanje procesorja. Hitra opravila, kot so merjenje jakosti signala ter ukazi s tipk in UART0 zahtevajo hitro ukrepanje procesorja v notranji zanki 50 ms. Počasenjša opravila, kot je meritev frekvence, ki traja 100 ms ali 1 s, se lahko izvedejo v zunanji zanki za meritev frekvence.

V zunanji zanki programa najprej inicializiramo enoto PWM (*INIPWM*). V inicializaciji PWM preverimo spremenljivko *STANJE* in ugotovimo, katera časovna baza je izbrana, ali 1 Hz ali 10 Hz. V zunanji zanki preverimo, ali se je meritev začela oziroma če so vrata odprta. Če so, preskočimo v notranjo zanko za merjenje jakosti signala in vnos ukazov. Tu nadaljujemo z meritvijo jakosti signala preko analogno-digitalnega pretvornika (ADC). Vsaka meritev jakosti signala pomeni povprečenje 16384 zaporednih A/D pretvorb, kar traja približno 50 ms. Nato preberemo morebitne ukaze iz zunanjega računalnika preko vmesnika UART0 oziroma iz tipk.

Jakost signala lahko na zaslon izpišemo bodisi s številsko vrednostjo v enoti dBm bodisi grafično z rastočo vrstico, kar izbiramo s tipkami ali ukazi iz računalnika. V primeru, da je izbrani način izpisa v številski obliki z enoto, vrednost jakosti signala preračunamo s pomočjo izmerjene frekvence, da dosežemo večjo natančnost. Podrobnejša razlaga se nahaja v poglavju *2.4.8. Popravek frekvenčnega odziva AD8309.* Za številskim rezultatom izpišemo enoto *dBm*. Meritev je natančna le v omejenem področju jakosti signala. To območje je za opisani merilnik med -60 dBm in +15 dBm. Zato v številskem načinu izpisa izpišemo *LOW*, ko je jakost merjenega signala nižja od -60 dBm, ter *HIGH*, ko jakost merjenega signala preseže zgornjo mejo +15 dBm. Kadar jakost signala izpisujemo grafično, ne izpišemo enote, niti ne preračunavamo vrednosti s pomočjo frekvence, saj nas pri grafičnem izpisu zanima le približna ocena, kje na skali se nahajamo.

Po izpisu jakosti signala preverimo, ali se je meritev frekvence zaključila. V tem primeru skočimo iz hitre notranje zanke v počasnejšo zunanjo zanko. Meritev frekvence traja 100 ms ali 1 s, odvisno od izbranega časa vrat. Ko je meritev frekvence zaključena, preverimo, kateri čas vrat je izbran. Če je to 100 ms (10 Hz), potem rezultat pomnožimo z deset, da dobimo pravilno vrednost. Rezultat nato shranimo v spremenljivko *FREKVNC.* To spremenljivko uporabimo pri preračunavanju vrednosti jakosti signala, kakor je opisano v prejšnjem odstavku.

Rezultat meritve frekvence pretvorimo iz dvojiške v desetiško obliko in izpišemo na zaslon. Rezultate meritev pošiljamo na LCD zaslon. Krmilniku Hitachi HD44780, katerega del je LCD zaslon, pošljemo znake, ki jih želimo izpisati na zaslonu. Ti znaki se zapišejo v pomnilnik (RAM) krmilnika. Z naslednjim ukazom iz procesorja se znaki iz pomnilnika HD44780 izpišejo na zaslon.

Število pretvorimo v desetiško tako, da ga delimo z deset, ostanek shranimo na sklad in količnik delimo naprej z deset. To storimo tolikokrat, na kolikor desetiških mest preračunavamo rezultat. Rezultat izpisujemo na deset mest, zato se zanka desetkrat ponovi. V opisanem postopku dobimo številke rezultata v obratnem vrstnem redu, najprej najnižjo številko. Zato jih zapišemo na sklad, da jih v običajnem vrstnem redu, torej najprej najvišjo številko, preberemo in izpišemo na LCD ter pošljemo na UART0. Tako lahko preko USB povezave meritve spremljamo in beležimo tudi na zunanjem računalniku.

Zaradi preglednosti pri izpisu frekvence brišemo vodilne ničle. To storimo tako, da številko, enako nič, nadomestimo s presledkom. Nato preverimo, če smo na mestu enic, pred decimalno piko, kjer ničle nikoli ne brišemo. V primeru, da je tam v resnici ničla, te ne nadomestimo s presledkom, ampak na tisto mesto vpišemo nič. V decimalnem delu ničel ne brišemo. Za rezultatom zapišemo na LCD še mersko enoto *MHz*.

Univerzalni asinhroni sprejemnik in oddajnik UART0 (Universal Asynchronous Receiver/Transmitter) ima na zunanjem delu dva pina, RXD0, ki je vhod oziroma sprejemnik ter TXD0, ki je izhod oziroma oddajnik. Pri pošiljanju rezultatov na zunanji računalnik uporabimo oddajnik TXD0. V register U0THR (Transmit Holding Register) prepišemo rezultat meritve in ga po metodi FIFO (First-In First-Out) preko COM vmesnika pošljemo na USB, povezan z zunanjim računalnikom.

Na oddajnik UART0 ne pošiljamo merske enote, temveč dva znaka za skok v novo vrstico: <CR> in <LF>, da je izpis rezultatov preglednejši. Oddajnik UART0 sicer vsebuje medpomnilnik FIFO za 16 znakov, kar pri hitrosti 9600bps, start bit, 8 podatkovnih bitov in stop bit oddamo v 16.7 ms. Ker meritev frekvence oddamo z manj kot 16 znaki in novega nezultata ne pričakujemo prej kot v 100 ms, je strah, da bi prekoračili zmogljivost medpomnilnika FIFO, odveč.

Zunanjo zanko zaključimo tako, da resetiramo PWM (*RESPWM*) in se vrnemo na začetek glavne zanke programa [18], [23]. Celotna programska koda se nahaja v *Prilogi 1*.

2.4.5 Upravljanje števca

Merilnik upravljamo s štirimi tipkami. Tipke so bile prvotno mišljene za upravljanje z menuji: gor, dol, levo in desno. Naš program je zaenkrat tako preprost, da lahko vsaki tipki dodelimo kar svojo nalogo. Z dvema izbiramo čas vrat, drugi dve pa služita za izbiro načina izpisa jakosti vhodnega signala.

Poleg tipk lahko merilnik upravljamo tudi preko zunanjega računalnika. Za povezavo uporabimo USB kabel do USB/COM pretvornika FT231 [24] na ploščici mikrokrmilnika, kot vmesnik med računalnikom in procesorjem služi enota mikrokrmilnika UART0. Preko nje lahko merilnik upravljamo ter iz njega zajemamo meritve. Za to na računalniku potrebujemo terminalski program, na primer Termite [25].

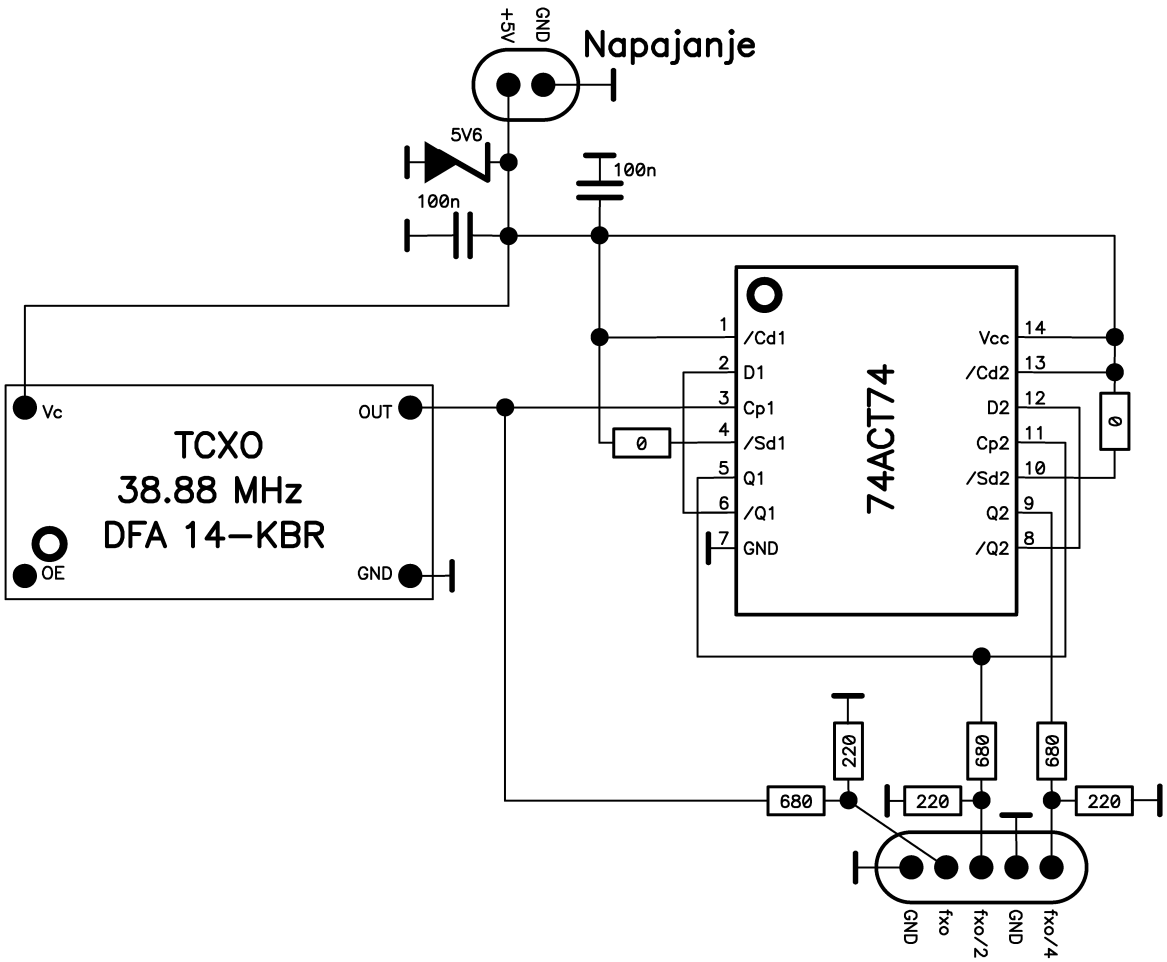
V nastavitvah programa izberemo pripadajoča COM vrata (odvisno od računalnika) in ustrezen baud rate (9600 bps).

Preko računalnika števec upravljamo tako, da v terminalski program vpišemo določeno črko, ki igra enako vlogo, kot če bi pritisnili tipko z enako funkcijo. Za izbiro časa vrat tako vpišemo „p“ ali „P“ za počasno meritev, ko je čas vrat 1 s, za hitro meritev s časom vrat 100 ms pa vpišemo „h“ ali „H“. Med načinoma izpisa jakosti signala izbiramo s črko „s“ ali „S“ za stolpec oziroma rastočo vrstico ter s črko „d“ ali „D“ za številski izpis v enoti dBm.

2.4.6 Zunanji kristalni oscilator TCXO

Mikrokrmilnik LPC2138/01 ima vgrajen lasten kristalni oscilator, ki popolnoma zadostuje za večino nalog, ki jih mikrokrmilnik opravlja. Za karseda natančen števec frekvence potrebujemo stabilnejši oscilator. Zato izbran zunanji oscilator pritrdimo na posebno ploščico z lastnim napajanjem in ga povežemo na mikrokrmilnik.

Kristalni oscilator DFA14-KBR, ki smo ga uporabili, je frekvence 38.88 MHz, tipa TCXO (Temperature Compensated X-tal Oscillator) [26]. Zaradi večje svobode pri uporabi in pisanju programske kode za fazno sklenjeno zanko PLL (Phase Locked Loop), smo na ploščico z oscilatorjem vgradili čip 74ACT74, ki vsebuje dva D flip-flopa [27]. Z njim frekvenco oscilatorja dvakrat delimo. Tako pripravimo tri različne frekvence na izhodu, ki jih lahko peljemo na mikrokrmilnik. En izhod predstavlja neposredna frekvenca TCXO, torej 38.88 MHz (f_0). Drug izhod je osnovna frekvenca, deljena z dva, torej 19.44 MHz ($f_0/2$), tretji izhod je enak drugemu izhodu, deljenemu z dva, torej 9.72 MHz ($f_0/4$). Načrt ploščice s kristalnim oscilatorjem in delilnikom 74ACT74 je na Sliki 13.



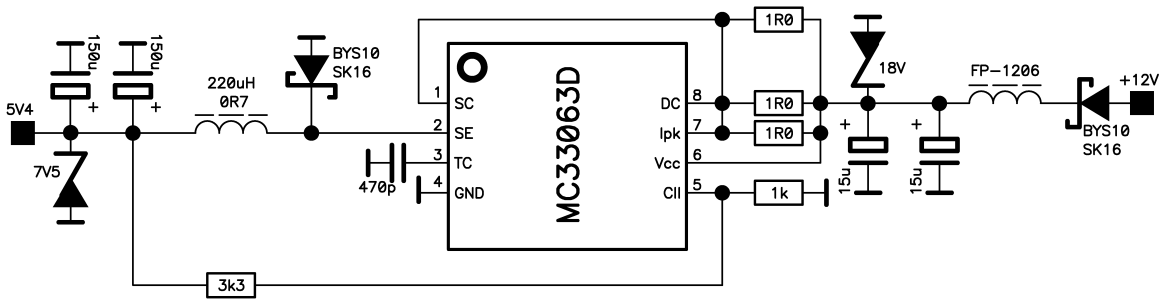
Slika 13: Načrt ploščice s kristalnim oscilatorjem in delilnikom 74ACT74

Zener dioda 5.6 V ima vlogo prenapetostne varovalke, da previsoka napetost ne poškoduje vezja. Kondenzatorja 100 nF skrbita, da je impedanca napajanja nizka za visoke frekvence. Na vseh treh izhodih so uporovni delilniki (220 Ω in 680 Ω), ki poskrbijo za ustrezno napetost na vhod takta mikrokrmilnika LPC2138/01. Ta napetost ne sme presegati 1.5 V vrh-vrh, saj vhod za takt v svoji notranjosti vsebuje CMOS vezje, napajano na 1.8 V.

2.4.7 Napajalnik

Za napajanje mikrokrmilnika ter ostalih vezij, ki so napajana preko njega, želimo stabilno napajalno napetost, tako 3.3 V za mikrokrmilnik LPC2138/01 kot tudi 5 V za vhodno-izhodne enote mikrokrmilnika ter za ostala vezja inštrumenta. (TCXO, ECL delilniki, vhodni ojačevalnik AD8309). Kristalni oscilator TCXO zahteva stabilno napetost 5 V s toleranco ± 5 %. Z manj stabilnim napajanjem preko USB procesor sicer živi, vendar TCXO ni točen.

Za točno meritev torej potrebujemo zunanje napajanje. Zunanje napajanje 12 V pripeljemo na stikalni napajalnik, ki napetost stabilizira in zniža na 5.4 V. Načrt vezja stikalnega napajalnika prikazuje Slika 14.

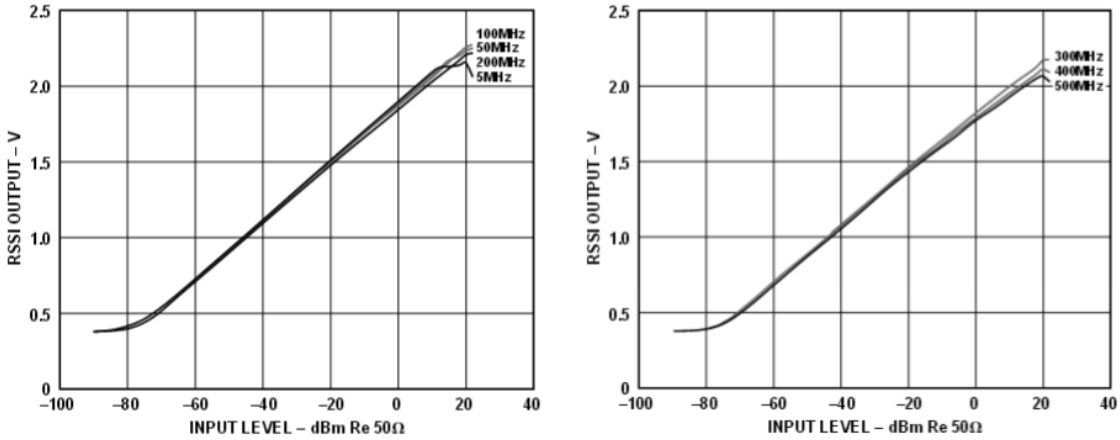


Slika 14: Načrt vezja stikalnega napajalnika

Izhodno napetost iz stikalnega napajalnika nastavljamo z uporoma 3.3 kΩ in 1 kΩ. Izhodno napetost 5.4 V peljemo skozi Schottky diodo SK16, na kateri je padec napetosti približno 400 mV, na regulator napetosti LP2951 [28] (glej Sliko 9: *Načrt vezave mikrokrmilnika LPC2138/01)* . Ta zniža napetost iz 5 V (5.4 V – 0.4 V) na zahtevanih 3.3 V. Napetost 3.3 V nato za napajalno napetost peljemo na mikrokrmilnik LPC2138/01. Del napetosti peljemo skozi Schottky diodo mimo čipa LP2951 na enega od priključkov (5 V), od koder se napajajo preostala vezja inštrumenta.

2.4.8 Popravek frekvenčnega odziva AD8309

Vhodni ojačevalnik in omejevalnik ter logaritemski detektor AD8309 za naš frekvenčni števec ženemo preko njegovih zagotovljenih frekvenčnih zmogljivosti. Že brez tega se razmerje med vhodnim in izhodnim RSSI signalom na spodnji in predvsem na zgornji meji razlikuje za različne frekvence. Razlog je upadanje ojačanja posameznih stopenj AD8309 na spodnji in predvsem na zgornji frekvenčni meji. To je prikazano na Sliki 15 a) in b).



a) *b)*

Slika 15 a) in b): Frekvenčni odziv čipa AD8309 - razmerje med vhodnim signalom in izhodnim RSSI signalom pri različnih frekvencah [9]

Pri frekvencah, višjih od zagotovljene 500 MHz, se ta razlika le še poveča. Tabela 1 podaja frekvenčni odziv čipa AD8309 pri osmih različnih frekvencah (3 MHz, 10 MHz, 30 MHz, 100 MHz, 300 MHz, 500 MHz, 700 MHz in 1 GHz) v območju od -60 dBm do +20 dBm, brez prilagoditve decibelske skale, torej brez kompenzacije.

Jakost \ f	3 MHz	10 MHz	30 MHz	100 MHz	300 MHz	500 MHz	700 MHz	1 GHz
+20 dBm	+17.7	+18.6	+21.1	+22.9	+16.6	+12.3	+7.5	+5.6
+10 dBm	+14.4	+14.3	+14.0	+13.0	+8.6	+4.7	+0.2	-14.2
+0 dBm	+3.5	+3.4	+3.2	+2.3	-0.8	-3.5	-7.4	-17.6
-10 dBm	-7.4	-7.4	-7.6	-8.3	-10.9	-13.2	-15.3	-25.9
-20 dBm	-18.1	-18.1	-18.2	-18.8	-20.6	-22.1	-23.9	-35.7
-30 dBm	-28.7	-28.8	-28.8	-29.3	-31.0	-32.1	-43.5	-48.4
-40 dBm	-39.3	-39.4	-39.4	-39.8	-41.8	-42.1	-45.8	-61.6
-50 dBm	-49.8	-49.9	-50.0	-50.9	-53.7	-51.6	-57.5	-71.4
-60 dBm	-59.8	-60.4	-60.5	-61.9	-70.5	-59.9	-70.0	-72.1

Tabela 1: Frekvenčni odziv čipa AD8309 brez kompenzacije

Rezultati v Tabeli 1 natančno potrjujejo Sliko 15 a) in b). Odstopanje je največje pri nizkih in visokih frekvencah ter pri nižjih in višjih jakostih vhodnega signala. Povprečno odstopanje jakosti signala znaša približno pet decibelov. Najboljše rezultate dobimo na sredini skale, pri frekvencah med 30 MHz in 300 MHz ter pri jakostih vhodnega signala med -50 dBm in 0 dBm.

Mikrokrmilnik nam daje manevrski prostor, da odstopanje izmerjenih vrednosti jakosti signala od želenih programsko kompenziramo z linearizacijo krivulje frekvenčnega odziva. Na podlagi meritev z nekompenzirano decibelsko skalo (Tabela 1) smo določili, kolikšen mora biti popravek amplitude. Ta znaša 17% pri frekvenci 500 MHz. Linearizacijo izvedemo tako, da s pomočjo izmerjene vrednosti frekvence preračunavamo amplitudo. In sicer tako, da shranimo izmerjeno vrednost frekvence (*f*) in jo delimo s popravkom ($3 \cdot 10^6$=500.000/17%). Nato prištejemo začetno vrednost $1 \cdot 4096$ ($1 \cdot 2^{12}$), kakor kaže enačba (2.1).

$$fs * 4096 = 4096 + \frac{f * 4096}{\frac{500\,MHz}{0.17}} \qquad (2.1)$$

Z dobljenim številom (***fs·4096***) izračunamo novo decibelsko skalo SKALADB', tako da nepopravljeno skalo (zapisano med konstantami mikrokrmilnika: SKALADB) delimo s tem številom. Tako dobimo popravljeno skalo, s pomočjo katere izračunamo popravek konstante MINUS60 (zapisana med konstantami mikrokrmilnika), ki jo potrebujemo za preračun odmika skale. V izogib računanju z negativnimi števili namreč jakost računamo v območju od 0 dBm do +75 dBm, tik pred izpisom pa odštejemo odmik, da dobimo pravilno območje od -60 dBm do +15 dBm. Popravek konstante MINUS60 izračunamo tako, da jo delimo s pravkar izračunano novo skalo SKALADB'. Prilagoditev decibelske skale prikazujeta enačbi (2.2) in (2.3).

$$SKALADB' = \frac{SKALADB * 4096}{fs * 4096} \tag{2.2}$$

$$MINUSDB = \frac{MINUS60}{SKALADB'} \tag{2.3}$$

Na ta način prilagodimo decibelsko skalo za izpis jakosti signala za točno tisto frekvenco, ki jo trenutno merimo. Spremenljivki SKALADB' in MINUSDB uporabimo pri izpisu jakosti v decibelih. Spremenljivko JAKOST, v kateri se nahaja rezultat meritve jakosti signala, delimo s popravljeno skalo SKALADB', nato količniku odštejemo popravljeni odmik MINUSDB. To prikazuje enačba (2.4).

$$Izpis\ dBm[-60...+15] = \frac{JAKOST}{SKALADB'} - MINUSDB \tag{2.4}$$

S tem, ko smo decibelsko skalo uspešno prilagodili, je meritev jakosti signala točnejša, povprečno odstopanje je le še okrog enega do dveh decibelov. Rezultati meritev jakosti signala s kompenzacijo pri osmih različnih frekvencah so predstavljeni v Tabeli 2.

Jakost \ f	3 MHz	10 MHz	30 MHz	100 MHz	300 MHz	500 MHz	700 MHz	1 GHz
+20 dBm	+12.4	+13.4	+16.2	+19.6	+18.2	+18.2	+17.2	+20.9
+10 dBm	+9.3	+9.4	+9.6	+10.1	+10.0	+9.9	8.9	-3.5
+0 dBm	-0.8	-0.8	-0.6	-0.2	+0.5	+1.2	+0.1	-7.6
-10 dBm	-10.9	-10.9	-10.8	-10.3	-9.7	-9.3	-8.9	-17.9
-20 dBm	-20.9	-20.8	-20.7	-20.4	-19.7	-18.9	-18.7	-29.9
-30 dBm	-30.8	-30.7	-30.7	-30.5	-30.3	-29.8	-30.8	-45.6
-40 dBm	-40.6	-40.6	-40.6	-40.6	-41.2	-40.6	-43.7	-61.9
-50 dBm	-50.4	-50.5	-50.5	-50.9	-53.5	-50.9	-57.0	-71.6
-60 dBm	-59.7	-60.2	-60.4	-61.8	-70.6	-59.8	-70.9	-72.2

Tabela 2: Frekvenčni odziv čipa AD8309 s kompenzacijo

3 Zaključek

Magistrsko delo opisuje inovativno zasnovo frekvenčnega števca, ki omogoča sočasno merjenje frekvence in amplitude – jakosti signala. Takšnih merilnikov, ki bi bili enostavni in majhnih dimenzij, na trenutnem tržišču kljub bogati ponudbi ne najdemo.

Predstavljeni merilnik je lahko osnova za nadaljnji razvoj in razširitev nabora funkcionalnosti frekvenčnega števca. Nedvomno bi bil dobrodošel dodatni nizkofrekvenčni vhod z enosmernim sklopom za štetje zelo nizkih frekvenc. Za merjenje frekvenc, višjih od 1 GHz, bi veljalo dodati preddelilnik, saj sodobni gradniki omogočajo deljenje frekvence vse do 25 GHz in več.

Frekvenčni števec, predstavljen v tem delu, ima preprost uporabniški vmesnik. Ta zajema tipkovnico, LCD zaslon ter USB/COM vrata za komunikacijo z računalnikom. Funkcionalnost merilnika lahko povečamo z razširitvami uporabniškega vmesnika; dodamo več različnih časov vrat, predvsem daljša (10 sekund) za natančnejšo meritev, omogočimo meritev recipročne vrednosti frekvence, torej periodo signala, dodamo različne načine izpisa rezultata, uporabimo zunanje frekvenčne normale, programsko umerjamo merilnik, itd.

Čeprav ima nekaj pomanjkljivosti in še veliko možnosti za razširiteve, se je inštrument v praksi že izkazal za uporabnega pri meritvah v področju radijskih frekvenc ter za dokaj natančnega pri meritvah jakosti signalov. Opisani merilnik torej nadomešča dva običajna laboratorijska merilnika, kjer dodatno prednost prinaša samodejni popravek pogreška amplitude iz izmerjene frekvence.

4 Priloge

Priloga 1: Programska koda

```
;*** Frekvencmeter - Ancka 06.09.2014 ***

;*** Konstante mikrokrmilnika LPC2138/01 ***
KVARC EQU   19439906                ;frekvenca TCXO (Hz)
CLK   EQU   KVARC*3                 ;frekvenca jedra ARM (PLL)
PCLK  EQU   CLK                     ;frekvenca vhodno/izhodnih enot
                                     (APBDIV)
BAUD  EQU   PCLK/153600             ;modulo deljenja UART za 9600bps*16
N1US  EQU   CLK/3000000             ;konstanta cakanja T1US za HD44780

PWMMR0      EQU   CLK*33            ;konstante za PWM
PWMMR1      EQU   CLK+1000
PWMMR1A     EQU   CLK/10+1000
PWMMR2      EQU   1000
ADC_DIV     EQU   PCLK/4000000      ;ADC clock 4 MHz
MINUS60     EQU   3550000           ;zacetek uporabne skale -60dBm (0.7V,
                                     24bit)
SKALA80     EQU   10400             ;skala stolpca 80dBm (768 korakov do
                                     2.3V, 24bit)
SKALADB     EQU   SKALA80*768/800   ;skala dBm (80dBm = 800 korakov)
POPRAVA     EQU   3000000           ;popravek amplitudnega odziva AD8309

;*** Vhodno/izhodni prikljucki LPC2138/01 ***
;/RESET - vhod vezan na /ERROR.LP2951 in na /DTR.FT231XS
;P0.0/TXD0/PWM1 - izhod vezan na RXD.FT231XS
;P0.1/RXD0/PWM3/EINT0 - vhod vezan na TXD.FT231XS
;P0.2.od/SCL0/CAP0.0 - nepovezan
;P0.3.od/SDA0/MAT0.0/EINT1 - nepovezan
;P0.4/SCK0/CAP0.1/AD0.6 - vhod AD0.6 za merjenje jakosti signala
;P0.5/MISO0/MAT0.1/AD0.7 – vhod P0.5 ki opazuje PWM2 za preprostprogram
;P0.6/MOSI0/CAP0.2/AD1.0 - vhod CAP0.2 za vhod stevca TIMER0
;P0.7/SSEL0/PWM2/EINT2 - izhod PWM2 za vrata stevca ECL 74ACT32
;P0.8/TXD1/PWM4/AD1.1 - ECL stevec vhod #1
;P0.9/RXD1/PWM6/EINT3 - ECL stevec vhod #2
;P0.10/RTS1/CAP1.0/AD1.2 - ECL stevec vhod #3
;P0.11.od/CTS1/CAP1.1/SCL1 - ECL stevec vhod #
```

```
;P0.12/DSR1/MAT1.0/AD1.3 - ECL stevec vhod #5
;P0.13/DTR1/MAT1.1/AD1.4 - nepovezan
;P0.14.od/DCD1/EINT1/SDA1/BOOTLOADER(!) - vhod vezan na /RTS.FT231XS
;P0.15/RI1/EINT2/AD1.5 - nepovezan
;P0.16-23 - digitalni izhodi za LCD D0-7.HD44780
;P0.24 - ne obstaja ***
;P1.16 - vhod z vgrajenim PULL-UP, modra tipka levo na maso
;P1.17 - vhod z vgrajenim PULL-UP, rumena tipka gor na maso
;P1.18 - vhod z vgrajenim PULL-UP, rdeca tipka dol na maso
;P1.19 - vhod z vgrajenim PULL-UP, zelena tipka desno na maso
;P1.20(!) - nepovezan ***
;P1.21 - digitalni izhod za LCD RS.HD44780
;P1.22-25 - nepovezani ***
;P1.26(!) - nepovezan ***
;P1.27-29 - nepovezani ***
;P1.30 - digitalni izhod za LCD E.HD44780
;P1.31 - nepovezan ***

;*** Dodelitev pomnilnika RAM ***
SKLAD       EQU   0x40000100  ;sklad raste navzdol -256byte
JAKOST      EQU   0x40000400  ;dvojiski rezultat meritve (32bit)
DELTAF      EQU   0x40000404  ;razlika frekvence (32bit)
FREKVNC     EQU   0x40000408  ;izmerjena frekvenca za preracunavanje
                               amplitude (32bit)
STANJE      EQU   0x40000500  ;bit0=stolpec/dbm,bit1=10Hz/1Hz,bit2=
                               meritev_veljavna (8bit)
STANJE1     EQU   0x40000504  ;enkrat zakasnjeno STANJE
STANJE2     EQU   0x40000508  ;dvakrat zakasnjeno STANJE

;*** Ukazi za delovanje zbirnika ARM ***
      AREA  RESET, CODE, READONLY, ALIGN=9
      ENTRY
      CODE32

;*** Tabela izjem na zacetnem naslovu 0 ***
IZJEME
      B     ZACNI       ;RESET
      B     IZJEME      ;Undef
      B     IZJEME      ;SWI
      B     IZJEME      ;PAbt
```

```
      B     IZJEME      ;DAbt
      NOP               ;Reserved Vector
      B     IZJEME      ;IRQ
      B     IZJEME      ;FIQ

;*** Onemogoci zascito vsebine FLASH na naslovu 0x000001FC ***
      ALIGN 512

;*** Inicializiraj MAM *** (R0-1)
MAM
      LDR   R0,=0xE01FC000    ;MAMCR=0
      MOV   R1,#0
      STRB  R1,[R0]
      LDR   R0,=0xE01FC004    ;MAMTIM=3
      MOV   R1,#3
      STRB  R1,[R0]
      LDR   R0,=0xE01FC000    ;MAMCR=2
      MOV   R1,#2
      STRB  R1,[R0]
      MOV   PC,LR
      LTORG

;*** Nastavi PCONP in APBDIV *** (R0-1)
PWRCLK
      LDR   R0,=0xE01FC0C4    ;PCONP
      LDR   R1,=0x0000102A    ;TIM0,UART0,PWM,AD0
      STR   R1,[R0]
      LDR   R0,=0xE01FC100    ;APBDIV=1
      MOV   R1,#1
      STRB  R1,[R0]
      MOV   PC,LR
      LTORG

;*** Inicializiraj PLL *** (R0-2)
PLL
      LDR   R0,=0xE01FC084    ;PLLCFG=0x22 P=2,M-1=2
      MOV   R1,#0x22
      STRB  R1,[R0]
      LDR   R0,=0xE01FC080    ;PLLCON=0x01 PLL-enable
```

```
      MOV   R1,#0x01
      STRB  R1,[R0]
      LDR   R0,=0xE01FC08C    ;PLLFEED
      MOV   R1,#0xAA
      MOV   R2,#0x55
      STRB  R1,[R0]
      STRB  R2,[R0]

PLL1
      LDR   R0,=0xE01FC088    ;PLLSTAT cakanje uklenitve
      LDR   R1,[R0]
      ANDS  R1,R1,#0x400
      BEQ   PLL1
      LDR   R0,=0xE01FC080    ;PLLCON=0x03 PLL-enable&connect
      MOV   R1,#0x03
      STRB  R1,[R0]
      LDR   R0,=0xE01FC08C    ;PLLFEED
      MOV   R1,#0xAA
      MOV   R2,#0x55
      STRB  R1,[R0]
      STRB  R2,[R0]
      MOV   PC,LR
      LTORG

;*** Inicializiraj GPIO in DAC *** (R0-1)
GPIO
      LDR   R0,=0xE002C000    ;PINSEL0
      LDR   R1,=0x0000A305    ;P0.0-1=UART0,P0.4=AD0.6,P0.6=CAP0.2,
                               P0.7=PWM2
      STR   R1,[R0]
      LDR   R0,=0xE002C004    ;PINSEL1 P0.25=AOUT
      LDR   R1,=0x00080000
      STR   R1,[R0]
      LDR   R0,=0xE002C014    ;PINSEL2 izklop JTAG&TRACE
      LDR   R1,[R0]
      BIC   R1,R1,#0x0C
      STR   R1,[R0]
      LDR   R0,=0xE01FC1A0    ;SCS=0x03 high-speed GPIO 0&1
      MOV   R1,#0x03
```

```
      STR   R1,[R0]
      LDR   R0,=0x3FFFC014    ;FIO0PIN vsi 1
      MOV   R1,#-1
      STR   R1,[R0]
      LDR   R0,=0x3FFFC034    ;FIO1PIN vsi 0
      MOV   R1,#0
      STR   R1,[R0]
      LDR   R0,=0x3FFFC000    ;FIO0DIR izhodi 0.13-31, vhodi 0.4-12,
                               izhodi0.0-3
      LDR   R1,=0xFFFFE00F
      STR   R1,[R0]
      LDR   R0,=0x3FFFC020    ;FIO1DIR izhodi 1.20-1.31
      LDR   R1,=0xFFF00000
      STR   R1,[R0]
      LDR   R0,=0xE006C000    ;DACR sredina skale, visok tok
      LDR   R1,=0x00008000
      STR   R1,[R0]
      MOV   PC,LR
      LTORG

;*** Inicializiraj UART0 *** (R0-1)
UART0
      LDR   R0,=0xE000C00C    ;U0LCR=0x83 DLAB=1,no-parity,1stop,8bit
      MOV   R1,#0x83
      STRB  R1,[R0]
      LDR   R0,=0xE000C000    ;U0DLL
      LDR   R1,=BAUD
      STRB  R1,[R0]
      LDR   R0,=0xE000C004    ;U0DLM
      MOV   R1,R1,LSR#8
      STRB  R1,[R0]
      LDR   R0,=0xE000C008    ;U0FCR=0xC1 RXtrig=14bit,FIFOenable
      MOV   R1,#0xC1
      STRB  R1,[R0]
      LDR   R0,=0xE000C00C    ;U0LCR=0x03 DLAB=0,no-parity,1stop,8bit
      MOV   R1,#0x03
      STRB  R1,[R0]
      MOV   PC,LR
      LTORG
```

```
;*** Inicializiraj HD44780 5X7,2vrstici,CGRAM *** (R0-3)
INILCD
      STR   LR,[SP,#-4]!      ;resi link
      BL    T100MS            ;zakasnitev 100ms
      MOV   R0,#0x38          ;function set
      BL    UKAZ
      BL    T4MS              ;zakasnitev 4ms
      MOV   R0,#0x38          ;function set
      BL    UKAZ
      MOV   R0,#0x38          ;function set
      BL    UKAZ
      MOV   R0,#0x08          ;display off
      BL    UKAZ
      MOV   R0,#0x01          ;display clear
      BL    UKAZ
      BL    T4MS              ;zakasnitev 4ms
      MOV   R0,#0x06          ;entry mode set
      BL    UKAZ
      MOV   R0,#0x0C          ;display on
      BL    UKAZ
      MOV   R0,#0x40          ;set CGRAM address
      BL    UKAZ
      MOV   R2,#64            ;prepisi 8 znake = 64 bajtov
      LDR   R3,=INILCD2
INILCD1
      LDRB  R0,[R3],#1
      BL    ZNAK
      SUBS  R2,R2,#1
      BNE   INILCD1
      MOV   R0,#0x80          ;set DDRAM address zacetek prve vrstice
      BL    UKAZ
      LDR   PC,[SP],#4        ;povratek na link
      LTORG
INILCD2
      DCB   0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x04  ;0=praznina
      DCB   0x10,0x10,0x10,0x10,0x10,0x10,0x00,0x04  ;1=enacrtica
      DCB   0x14,0x14,0x14,0x14,0x14,0x14,0x00,0x04  ;2=dvecrtici
      DCB   0x15,0x15,0x15,0x15,0x15,0x15,0x00,0x04  ;3=tricrtice
      DCB   0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x1F  ;0=praznina podcrtana
```

```
      DCB   0x10,0x10,0x10,0x10,0x10,0x10,0x00,0x1F  ;1=enacrtica podcrtana
      DCB   0x14,0x14,0x14,0x14,0x14,0x14,0x00,0x1F  ;2=dvecrtici podcrtana
      DCB   0x15,0x15,0x15,0x15,0x15,0x15,0x00,0x1F  ;3=tricrtice podcrtana
      ALIGN

;*** Podcrtan stolpec v desno v drugi vrstici iz R3 *** (R0-3)
STOLP STR   LR,[SP,#-4]!      ;resi link
      MOV   R0,#0xC0          ;set DDRAM address druga vrstica
      BL    UKAZ
      MOV   R2,#0
STOLP1      MOV   R0,#0       ;praznina za velike korake po 16
      CMP   R3,#16            ;ena crtica?
      MOVCS R0,#1
      CMP   R3,#32            ;dve crtici?
      MOVCS R0,#2
      CMP   R3,#48            ;tri crtice?
      MOVCS R0,#3
      AND   R1,R3,#15         ;podcrta pripadajoci znak za male korake po 1
      CMP   R1,R2
      ADDEQ R0,R0,#4
      BL    ZNAK              ;izpisi crtice
      SUBS  R3,R3,#48         ;odstej 3 crtice, negativno ustavi?
      ANDMI R3,R3,#15
      ADD   R2,R2,#1          ;zanka 16 znakov
      CMP   R2,#16
      BMI   STOLP1
      LDR   PC,[SP],#4        ;povratek na link
      LTORG

;*** Znak iz R0 v HD44780 *** (R0-1)
ZNAK
      STR   LR,[SP,#-4]!      ;resi link
      LDR   R1,=0x3FFFC016    ;FIO0PIN2 podatek
      STRB  R0,[R1]
      LDR   R1,=0x3FFFC03A    ;FIO1SET2
      MOV   R0,#0x20          ;RS=1
      STRB  R0,[R1]
      BL    T1US              ;zakasnitev 1us
      LDR   R1,=0x3FFFC03B    ;FIO1SET3
```

```
      MOV   R0,#0x40          ;vklopi E
      STRB  R0,[R1]
      BL    T1US              ;zakasnitev 1us
      LDR   R1,=0x3FFFC03F    ;FIO1CLR3
      MOV   R0,#0x40          ;izklopi E
      STRB  R0,[R1]
      BL    T100US            ;zakasnitev 100us
      LDR   PC,[SP],#4        ;povratek na link
      LTORG

;*** Ukaz iz R0 v HD44780 *** (R0-1)
UKAZ
      STR   LR,[SP,#-4]!      ;resi link
      LDR   R1,=0x3FFFC016    ;FIO0PIN2 podatek
      STRB  R0,[R1]
      LDR   R1,=0x3FFFC03E    ;FIO1CLR2
      MOV   R0,#0x20          ;RS=0
      STRB  R0,[R1]
      BL    T1US              ;zakasnitev 1us
      LDR   R1,=0x3FFFC03B    ;FIO1SET3
      MOV   R0,#0x40          ;vklopi E
      STRB  R0,[R1]
      BL    T1US              ;zakasnitev 1us
      LDR   R1,=0x3FFFC03F    ;FIO1CLR3
      MOV   R0,#0x40          ;izklopi E
      STRB  R0,[R1]
      BL    T100US            ;zakasnitev 100us
      LDR   PC,[SP],#4        ;povratek na link

;*** Zakasnitev 1us *** (R0)
      ALIGN 16
T1US  MOV   R0,#N1US
T1US1 SUBS  R0,R0,#1
      BPL   T1US1
      MOV   PC,LR

;*** Zakasnitev 100us *** (R0-1)
T100US
      STR   LR,[SP,#-4]!      ;resi link
```

```
      MOV   R1,#100
T100US1
      BL    T1US
      SUBS  R1,R1,#1
      BPL   T100US1
      LDR   PC,[SP],#4        ;povratek na link
;*** Zakasnitev 4ms *** (R0-2)
T4MS
      STR   LR,[SP,#-4]!      ;resi link
      MOV   R2,#40
T4MS1
      BL    T100US
      SUBS  R2,R2,#1
      BPL   T4MS1
      LDR   PC,[SP],#4        ;povratek na link

;*** Zakasnitev 100ms *** (R0-3)
T100MS
      STR   LR,[SP,#-4]!      ;resi link
      MOV   R3,#25
T100MS1
      BL    T4MS
      SUBS  R3,R3,#1
      BPL   T100MS1
      LDR   PC,[SP],#4        ;povratek na link

;*** Zakasnitev 1s *** (R0-3)
T1S
      STR   LR,[SP,#-4]!      ;resi link
      MOV   R3,#250
T1S1
      BL    T4MS
      SUBS  R3,R3,#1
      BPL   T1S1
      LDR   PC,[SP],#4        ;povratek na link

;*** Zacetek programa po resetu ***
ZACNI LDR   SP,=SKLAD         ;inicializiraj SP
      BL    MAM               ;inicializiraj MAM
```

```
      BL    PWRCLK          ;nastavi PCONP in APBDIV
      BL    PLL             ;inicializiraj PLL
      BL    GPIO            ;inicializiraj GPIO in DAC
      BL    UART0           ;inicializiraj UART0
      BL    INILCD          ;inicializiraj HD44780 5X7,2vrstici,CGRAM
      BL    INITIM0         ;inicializiraj TIMER0
      MOV   R1,#5           ;zacetno stanje instrumenta 05,01,01
      LDR   R0,=STANJE
      STRB  R1,[R0]
      MOV   R1,#1
      LDR   R0,=STANJE1
      STRB  R1,[R0]
      MOV   R1,#1
      LDR   R0,=STANJE2
      STRB  R1,[R0]
      MOV   R0,#0x80        ;prva vrstica besedila
      BL    UKAZ
      LDR   R2,=ZACNI5

ZACNI1
      LDRB  R0,[R2],#1
      MOVS  R0,R0
      BEQ   ZACNI2
      BL    ZNAK
      B     ZACNI1

ZACNI2
      MOV   R0,#0xC0        ;druga vrstica besedila
      BL    UKAZ
      LDR   R2,=ZACNI6

ZACNI3
      LDRB  R0,[R2],#1
      MOVS  R0,R0
      BEQ   ZACNI4
      BL    ZNAK
      B     ZACNI3
```

```
ZACNI4
      B     ZANKA             ;glavna zanka programa
      LTORG
ZACNI5      DCB   "Frekv.meter 1GHz",0
ZACNI6      DCB   "ANCKA 06.09.2014",0
      ALIGN 4

;*** Glavna zanka programa ***
ZANKA
      BL    INIPWM            ;inicializiraj PWM
ZANKA2
      LDR   R0,=0x3FFFC014    ;FIO0PIN0 meritev zacne?
      LDRB  R1,[R0]
      TST   R1,#0x20          ;vhod P0.5?
      BNE   ZANKA2
ZANKA1
      BL    ADC               ;meritev amplitude preko ADC
      BL    TIPKE             ;ukaz s tipk
      BL    RXUART            ;ukaz z UART0

;Izbira izpisa dBm ali stolpec
      LDR   R2,=STANJE2       ;meritev veljavna?
      LDRB  R0,[R2]
      TST   R0,#0x04
      BEQ   NEVELJAVNA
      LDR   R2,=STANJE        ;izpis na LCD v dBm?
      LDRB  R0,[R2]
      TST   R0,#1
      BLNE  IZPIS_dBm
      LDR   R2,=STANJE        ;izpis stolpca na LCD?
      LDRB  R0,[R2]
      TST   R0,#1
      BLEQ  IZPIS_STOLPCEV

NEVELJAVNA
      LDR   R0,=0x3FFFC014    ;FIO0PIN0 meritev zakljucena?
      LDRB  R1,[R0]
      TST   R1,#0x20          ;vhod P0.5?
```

```
      BEQ   ZANKA1
      BL    T100US            ;pocakam 100us
      BL    MERIF             ;meritev frekvence
      LDR   R0,=STANJE2       ;1Hz ali 10Hz?
      LDRB  R2,[R0]
      TST   R2,#0x02
      ADDNE R3,R3,R3,LSL#2    ;pomnozimo z 10 v nacinu 10Hz
      MOVNE R3,R3,LSL#1
      LDR   R0,=FREKVNC       ;shranimo izmerjeno vrednost frekvence za
                               preracunavanje amplitude
      STR   R3,[R0]
      LDR   R0,=STANJE1       ;zakasnitev za 2 cikla/meritvi
      LDR   R1,=STANJE2
      LDRB  R2,[R0]
      STRB  R2,[R1]
      LDR   R1,=STANJE
      LDRB  R2,[R1]
      STRB  R2,[R0]
      LDR   R0,=STANJE2       ;meritev veljavna?
      LDRB  R2,[R0]
      TST   R2,#0x04
      BLNE  DECIZP_SP         ;desetiski izpis z brisanjem vodilnih nicel
      BL    RESPWM            ;reset PWM
      B     ZANKA
      LTORG

;*** Desetiski izpis z brisanjem vodilnih nicel *** (R0-5, SKLAD-48)
;uporaba sklada: en polozaj (4byte) za LR in 10 polozajev (40byte) za 10
cifer


DECIZP_SP
      STR   LR,[SP,#-4]!
      MOV   R4,#9             ;zacetek zanke
DECIZP1_SP
      BL    DELI10            ;izpisi desetiska mesta na sklad
      STR   R0,[SP,#-4]!      ;ostanek (cifro) shrani na sklad
      SUBS  R4,R4,#1
      BPL   DECIZP1_SP        ;cifre s sklada posiljamo na zaslon
      LDR   R0,=0x80          ;nastavi polozaj
      BL    UKAZ
```

```
      MOV   R4,#9             ;stevec cifer
      MOV   R5,#' '           ;podremo zastavico za cifro razlicno od 0 z
                               ASCII kodo za presledek
SKLAD_BERI
      CMP   R4,#6             ;ali smo na mestu enic, ker nicle enic nikoli
                               ne brisemo?
      MOVEQ R5,#'0'           ;postavimo ASCII znak za '0' v zastavico, da
                               nicel ne brisemo vec
      CMP   R4,#5             ;ali smo na mestu za dec. piko
      MOV   R0,#'.'
      LDR   R1,=0xE000C000    ;U0THR izpis pike na UART0
      STRBEQ R0,[R1]
      BLEQ  ZNAK              ;izpisemo piko, klic podprograma pogazi
                               zastavico EQ!
      LDR   R0,[SP],#4        ;precitamo cifro iz sklada
      CMP   R0,#0             ;ali je cifra enaka nic?
      MOVEQ R0,R5             ;niclo zamenjamo z vrednostjo zastavice
                               (presledek ali nicla ASCII)
      ADDNE R0,R0,#'0'        ;ne-niclo pretvorimo v ASCII znak tako, da
                               pristejemo kodo za niclo
      MOVNE R5,#'0'           ;postavimo ASCII znak za '0' v zastavico, da
                               nicel ne brisemo vec
      LDR   R1,=0xE000C000    ;U0THR izpis znaka na UART0
      STRB  R0,[R1]
      BL    ZNAK              ;izpisemo presledek ali cifro kot ASCII znak
      SUBS  R4,R4,#1          ;zmanjsamo stevec cifer
      BPL   SKLAD_BERI
      LDR   R1,=0xE000C000    ;U0THR izpis nove vrste na UART0
      MOV   R0,#13            ;<CR>
      STRB  R0,[R1]
      MOV   R0,#10            ;<LF>
      STRB  R0,[R1]
      MOV   R0,#' '           ;izpisemo mersko enoto
      BL    ZNAK
      MOV   R0,#'M'
      BL    ZNAK
      MOV   R0,#'H'
      BL    ZNAK
      MOV   R0,#'z'
      BL    ZNAK
      MOV   R0,#' '           ;izpisemo, ce je 1Hz
      LDR   R2,=STANJE        ;1Hz ali 10Hz
```

```
      LDRB  R1,[R2]
      TST   R1,#0x02          ;preverimo bit1
      MOVNE R0,#'*'           ;izpisemo, ce je 10Hz
      BL    ZNAK
      LDR   PC,[SP],#4
      LTORG

;*** Deli vsebino R3 z 10, ostanek v R0 in kvocient v R3 *** (R0-4)
DELI10
      MOV   R0,R3             ;deljenec v R0
      MOV   R1,#0xA0000000    ;odstevam od deljenca 10*2^N
      MOV   R2,#0x10000000    ;pristevam k kvocientu  1*2^N
      MOV   R3,#0             ;kvocient v R3
DELI101
      SUBS  R0,R0,R1          ;odstejem R0-R1
      ADDCS R3,R3,R2          ;ni izposoje >>> pristejem kvocientu R3+R2
      ADDCC R0,R0,R1          ;je izposoja >>> pristejem prevec odsteto
                               R0+R1
      MOV   R1,R1,LSR#1       ;pomaknem R1 v desno
      MOVS  R2,R2,LSR#1       ;pomaknem R2 v desno, preverim konec zanke!
      BNE   DELI101
      MOV   PC,R14

;*** Nepredznaceno deljenje R3/R0=R3, ostanek v R0 *** (R0-3)
NDELI
      MOVS  R1,R0             ;prepreci deljenje z 0?
      MOVEQ PC,LR
      MOV   R2,#1
      BMI   NDELI2
NDELI1
      MOV   R2,R2,LSL#1       ;enica delitelja v MSB
      MOVS  R1,R1,LSL#1
      BPL   NDELI1
NDELI2
      MOV   R0,R3             ;ostanek v R0
      MOV   R3,#0             ;kvocient v R3
NDELI3
      CMP   R0,R1
      SUBCS R0,R0,R1
```

```
      ADDCS R3,R3,R2
      MOV   R1,R1,LSR#1
      MOVS  R2,R2,LSR#1
      BNE   NDELI3
      MOV   PC,LR

;*** Inicializacija TIMER0 *** (R0-1)
INITIM0
      LDR   R0,=0xE0004070    ;T0CTCR
      MOV   R1,#0x0B          ;CAP0.2,BOTH
      STRB  R1,[R0]
      LDR   R0,=0xE0004004    ;T0TCR
      MOV   R1,#0x01
      STRB  R1,[R0]
      MOV   PC,LR
      LTORG

;*** Inicializacija PWM *** (R0-1)
INIPWM
      LDR   R0,=0xE0014014    ;PWMMCR
      LDR   R1,=0x00000002    ;PWMMR0R
      STR   R1,[R0]
      LDR   R0,=0xE001404C    ;PWMPCR
      LDR   R1,=0x00000404    ;PWMSEL2,PWMENA2
      STR   R1,[R0]
      LDR   R0,=0xE0014050    ;PWMLER
      MOV   R1,#0x07          ;Enable PWM latch 0,1,2
      STRB  R1,[R0]
      LDR   R0,=0xE0014018    ;PWMMR0
      LDR   R1,=PWMMR0
      STR   R1,[R0]
      LDR   R0,=STANJE        ;stanje 1Hz ali 10Hz?
      LDRB  R2,[R0]
      TST   R2,#0x02
      LDR   R0,=0xE001401C    ;PWMMR1
      LDREQ R1,=PWMMR1  ;1Hz
      LDRNE R1,=PWMMR1A ;10Hz
      STR   R1,[R0]
      LDR   R0,=0xE0014020    ;PWMMR2
```

```
      LDR   R1,=PWMMR2
      STR   R1,[R0]
      LDR   R0,=0xE0014004    ;PWMTCR CounterEnable & PWMEnable
      MOV   R1,#0x09
      STRB  R1,[R0]
      MOV   PC,LR
      LTORG

;*** Reset PWM *** (R0-1)
RESPWM
      LDR   R0,=0xE0014018    ;PWMMR0
      LDR   R1,[R0]
      LDR   R0,=0xE0014008    ;PWMTC
      STR   R1,[R0]
      MOV   PC,LR
      LTORG

;*** Ukaz s tipk *** (R0-2)
TIPKE
      STR   LR,[SP,#-4]!      ;resi link
      LDR   R0,=0x3FFFC036    ;FIO1PIN2 branje tipk
      LDRB  R1,[R0]
      LDR   R2,=STANJE
      LDRB  R0,[R2]
      TST   R1,#0x01          ;modra tipka levo?
      BICEQ R0,R0,#0x01       ;izpis stolpec
      TST   R1,#0x02          ;rumena tipka gor?
      ORREQ R0,R0,#0x01       ;izpis dBm
      TST   R1,#0x04          ;rdeca tipka dol?
      BICEQ R0,R0,#0x02       ;meritev 1Hz
      TST   R1,#0x08          ;zelena tipka desno?
      ORREQ R0,R0,#0x02       ;meritev 10Hz
      STRB  R0,[R2]
      LDR   PC,[SP],#4        ;povratek na link


;*** Ukaz z UART0 *** (R0-2)
RXUART
      LDR   R0,=0xE000C014    ;U0LSR ali je kaj prislo v sprejemnik?
```

```
      LDRB  R1,[R0]
      TST   R1,#0x01
      MOVEQ PC,LR
      LDR   R0,=0xE000C000    ;U0RBR precitamo sprejeti bajt
      LDRB  R1,[R0]
      AND   R1,R1,#0x7F       ;brisemo pariteto za vsak slucaj
      CMP   R1,#0x60          ;male crke v velike?
      SUBCS R1,R1,#0x20
      LDR   R2,=STANJE
      LDRB  R0,[R2]
      CMP   R1,#"S"           ;ukaz s ali S (stolpec za jakost)?
      BICEQ R0,R0,#0x01       ;izpis stolpec
      CMP   R1,#"D"           ;ukaz d ali D (dBm za jakost)?
      ORREQ R0,R0,#0x01       ;izpis dBm
      CMP   R1,#"P"           ;ukaz p ali P?
      BICEQ R0,R0,#0x02       ;pocasna meritev frekvence 1Hz
      CMP   R1,#"H"           ;ukaz h ali H?
      ORREQ R0,R0,#0x02       ;hitra meritev frekvence 10Hz
      STRB  R0,[R2]
      MOV   PC,LR
      LTORG

;*** Meritev amplitude preko ADC s povprecenjem *** (R0-3)
ADC   MOV   R2,#16384         ;16384=2**14 povprecenj! cca 50ms
      MOV   R3,#0             ;rezultat povprecenja
ADC_INIT                      ;inicializacija ADC in zacetek konverzije
      MOV   R0,#ADC_DIV       ;izracunaj deljenje takta ADC0

      LDR   R1,=0x01200040    ;START=001,PDN=1,CLKS=11,CLKDIV,SEL=6
      ORR   R1,R1,R0,LSL#8
      LDR   R0,=0xE0034000    ;AD0CR
      STR   R1,[R0]
ADC_MERITEV
      LDR   R0,=0xE0034004    ;AD0GDR cakaj konec pretvorbe-zanka
      LDR   R1,[R0]
      TST   R1,#0x80000000
      BEQ   ADC_MERITEV
      MOV   R1,R1,LSL#16      ;pocisti register in postavi bite na zacetek
      MOV   R1,R1,LSR#22
```

```
      ADD   R3,R3,R1          ;pristejemo povprecenju v R3
      SUBS  R2,R2,#1          ;zanka povprecenja
      BNE   ADC_INIT
      LDR   R0,=JAKOST        ;shrani 24-bitni rezultat v JAKOST(32bit)
      STR   R3,[R0]
      MOV   PC,LR
      LTORG

;*** Izracun in izpis ADC stolpcev *** (R0-3)
IZPIS_STOLPCEV
      STR   LR,[SP,#-4]!
      LDR   R0,=JAKOST
      LDR   R3,[R0]
      LDR   R0,=MINUS60       ;odstejemo offset za 0.7 V (-60 dBm)
      SUBS  R3,R3,R0
      MOVMI R3,#0
      LDR   R0,=SKALA80       ;ADC vrednost za 2.2 V (+15 dBm) zmanjsana za
                               0.7 V, ki smo jih odsteli
      BL    NDELI
      BL    STOLP
      LDR   PC,[SP],#4
      LTORG
;*** Izpis na LCD v dBm *** (R0-5)
IZPIS_dBm
      STR   LR,[SP,#-4]!
      MOV   R0,#0xC0          ;set DDRAM address druga vrstica
      BL    UKAZ
      MOV   R0,#' '           ;presledki na zacetku za poravnavo merske
                               enote v prvi (MHz) in drugi (dBm) vrstici
      BL    ZNAK
      MOV   R0,#' '
      BL    ZNAK
      MOV   R0,#' '
      BL    ZNAK
      MOV   R0,#' '
      BL    ZNAK
      MOV   R0,#' '
      BL    ZNAK
      MOV   R0,#' '
      BL    ZNAK
```

```
;Izpis ADC // 0 V = 0(dec), 3.3 V = 1023(dec) / 0.7 V = 217(dec) = -60 dBm,
2.2 V = 682(dec) = 15 dBm
      LDR   R0,=FREKVNC       ;izracunam popravek frekvence fs*4096 v R0
      LDR   R3,[R0]
      LDR   R0,=POPRAVA
      BL    NDELI
      ADD   R0,R3,#4096
      LDR   R3,=SKALADB       ;izracunam popravek SKALADB v R0 in na sklad
      MOV   R3,R3,LSL#12      ;pomnozim s 4096
      BL    NDELI
      MOV   R0,R3
      MOV   R4,R3
      LDR   R1,=JAKOST        ;ADC vrednost zadnje meritve v R3
      LDR   R3,[R1]
      BL    NDELI
      MOV   R0,R4             ;izracunam popravljeni MINUSDB v R0
      MOV   R4,R3
      LDR   R3,=MINUS60
      BL    NDELI
      MOV   R0,R3             ;odstejem MINUSDB
      MOV   R3,R4
      SUBS  R3,R3,R0
      BMI   DBM_LOW           ;manjse od -60dBm
      LDR   R1,=750           ;razpon 75 dB
      CMP   R3, R1
      BPL   DBM_HIGH          ;vecje od +15dBm

      LDR   R1,=600           ;0=-60dBm, 600=0dBm, 750=+15dBm
      SUBS  R3,R3,R1
      BPL   DBM_PLUS

      MOV   R0,#'-'           ;izpisemo -dBm
      BL    ZNAK              ;izpisi -
      RSB   R3,R3,#0
      B     IZPIS_dBm_CIFRE

DBM_PLUS                      ;izpisemo +dBm
      MOV   R0,#'+'
      BL    ZNAK              ;izpisi +
```

```
IZPIS_dBm_CIFRE
      MOV   R4,#3             ;zacetek zanke, 3 cifre
dBm_CIFRA
      BL    DELI10
      STR   R0,[SP,#-4]!      ;ostanek (cifro) shrani na sklad
      SUBS  R4,R4,#1
      BNE   dBm_CIFRA
      MOV   R4,#3             ;stevec cifer
      MOV    R5,#' '          ;podremo zastavico za cifro razlicno od 0 z
                              ASCII kodo za presledek
dBm_PISI
      CMP   R4,#2             ;ali smo na mestu enic, ker nicle enic nikoli
                              ne brisemo?
      MOVEQ R5,#'0'           ;postavimo ASCII znak za '0' v zastavico, da
                              nicel ne brisemo vec
      CMP   R4,#1             ;ali smo na mestu za dec. piko
      MOV   R0,#'.'
      BLEQ  ZNAK              ;izpisemo piko, klic podprograma pogazi
                              zastavico EQ!
      LDR   R0,[SP],#4        ;nalozimo cifro iz sklada
      CMP   R0,#0             ;ali je cifra enaka nic?
      MOVEQ R0,R5             ;niclo zamenjamo z vrednostjo zastavice
                              (presledek ali nicla ASCII)
      ADDNE R0,R0,#'0'        ;ne-niclo pretvorimo v ASCII znak tako, da
                              pristejemo kodo za niclo
      MOVNE R5,#'0'           ;postavimo ASCII znak za '0' v zastavico, da
                              nicel ne brisemo vec
      BL    ZNAK              ;izpisemo presledek ali cifro kot ASCII znak
      SUBS  R4,R4,#1          ;zmanjsamo stevec cifer
      BNE   dBm_PISI
      MOV   R0,#' '           ;izpisemo mersko enoto
      BL    ZNAK
      MOV   R0,#'d'
      BL    ZNAK
      MOV   R0,#'B'
      BL    ZNAK
      MOV   R0,#'m'
      BL    ZNAK
      MOV   R0,#' '
      BL    ZNAK
      LDR   PC,[SP],#4
      LTORG
```

```
DBM_LOW
      MOV   R0,#' '
      BL    ZNAK
      MOV   R0,#' '
      BL    ZNAK
      MOV   R0,#' '
      BL    ZNAK
      MOV   R0,#' '
      BL    ZNAK
      MOV   R0,#' '
      BL    ZNAK
      MOV   R0,#' '
      BL    ZNAK
      MOV   R0,#' '
      BL    ZNAK
      MOV   R0,#'L'
      BL    ZNAK
      MOV   R0,#'O'
      BL    ZNAK
      MOV   R0,#'W'
      BL    ZNAK
      MOV   R0,#' '
      BL    ZNAK
      LDR   PC,[SP],#4
      LTORG

DBM_HIGH
      MOV   R0,#' '
      BL    ZNAK
      MOV   R0,#' '
      BL    ZNAK
      MOV   R0,#' '
      BL    ZNAK
      MOV   R0,#' '
      BL    ZNAK
      MOV   R0,#' '
      BL    ZNAK
      MOV   R0,#'H'
      BL    ZNAK
      MOV   R0,#'I'
      BL    ZNAK
```

```
      MOV   R0,#'G'
      BL    ZNAK
      MOV   R0,#'H'
      BL    ZNAK
      MOV   R0,#' '
      BL    ZNAK
      LDR   PC,[SP],#4
      LTORG

;*** Meritev frekvence *** (R0-3)
MERIF
      LDR   R0,=0xE0004008    ;T0TC izpis razlike Timer Counter 0
      LDR   R3,[R0]
      MOV   R3,R3,LSL#5
      LDR   R0,=0x3FFFC015    ;FIO0PIN1
      LDRB  R1,[R0]
      AND   R1,R1,#0x1F       ;5 bitov ECL
      ORR   R3,R3,R1
      MOV   R1,R3
      LDR   R0,=DELTAF        ;razlika do stare meritve
      LDR   R2,[R0]
      SUB   R3,R3,R2
      STR   R1,[R0]
      MOV   PC,LR
      LTORG

;*** Konec delovanja zbirnika ARM ***
      END
```

5 Literatura

[1] Early Electronic Frequency Meter & Counter. *HP Memory Project* [Online]. Dosegljivo: http://www.hpmemory.org/wa_pages/wall_a_page_11.htm. [Dostopano: 06. 08. 2014]

[2] R. Dekker. A E1T Decade Scaler Tube raised from the dead. [Online]. Dosegljivo: http://www.dos4ever.com/trochotron/TROCH.html. [Dostopano: 13. 08. 2014]

[3] Dieter. E1T (Philips Miniwatt) Decade counting tube. *Dieter's Nixie World* [Online]. Dosegljivo: http://www.tube-tester.com/sites/nixie/different/e1t-tubes/E1T_philips/e1t-phil.htm. Dostopano: [13. 08. 2014]

[4] I. Poole. Frequency counter accuracy and resolution. *Radio-electronics.com* [Online]. Dosegljivo:http://www.radio-electronics.com/info/t_and_m/frequency_counter/measurement-resolution-accuracy-errors.php. Dostopano: [03. 06. 2014]

[5] I. Poole. Counter Timer / Interval Timer. *Radio-electronics.com* [Online]. Dosegljivo: http://www.radio-electronics.com/info/t_and_m/frequency_counter/frequency-counter-timer-interval.php. Dostopano: [03. 06. 2014]

[6] M. Vidmar. Marker frekvencmeter za spektralni analizator. (1999, 2000, 2001). *Beacon 99, RTV klub Murska Sobota* [Online]. str. 161-168. Dosegljivo: http://lea.hamradio.si/~s53mv/beacon.pdf. [Dostopano: 12. 05. 2014]

[7] Fundamentals of Microwave Frequency Counters. (1997). *HP Memory Project* [Online]. Dosegljivo: www.hpmemory.org/an/pdf/an_200-1.pdf. [Dostopano: 11. 07. 2014]

[8] M. Vidmar. Simple RF/Microwave Frequency Counter. *Matjaž Vidmar S53MV* [Online] Dosegljivo: http://lea.hamradio.si/~s53mv/counter/counter.html. [Dostopano: 06. 05. 2014]

[9] Analog devices. 5 Mhz-500 MHz 100 dB Demodulating Logarithmic Amlifier with Limiter Output AD8309. (1999). *Analog Devices, Inc.* [Online] Dosegljivo:http://www.analog.com/static/imported-files/data_sheets/AD8309.pdf. [Dostopano: 26. 05. 2014]

[10] ON Semiconductor. MC10EL16, MC100EL16 5.0 V ECL Differential Receiver. *http://onsemi.com* [Online]. Dosegljivo: http://www.onsemi.com/pub_link/Collateral/MC10EL16-D.PDF. [Dostopano: 18. 06. 2014]

[11] ON Semiconductor. MC10EL52, MC100EL52 5V ECL Differential Data and Clock D Flip-Flop. *http://onsemi.com* [Online]. Dosegljivo: http://www.onsemi.com/pub_link/Collateral/MC10EL52-D.PDF. [Dostopano: 18. 06. 2014]

[12] ON Semiconductor. MC10EL58, MC100EL58 5.0 V ECL 2:1 Multiplexer. *http://onsemi.com* [Online]. Dosegljivo: http://www.onsemi.com/pub_link/Collateral/MC10EL58-D.PDF. [Dostopano: 18. 06. 2014]

[13] Fairchild Semiconductor. 74AC32, 74ACT32 Quad 2-Input OR Gate. (2008). *www.fairchildsemi.com* [Online]. Dosegljivo: https://www.fairchildsemi.com/datasheets/74/74AC32.pdf. [Dostopano: 20. 06. 2014]

[14] ON Semiconductor. MC10E131, MC100E131 5VECL 4-Bit D Flip-Flop. *http://onsemi.com* [Online]. Dosegljivo: http://www.onsemi.com/pub_link/Collateral/MC10E131-D.PDF. [Dostopano: 18. 06. 2014]

[15] Maxim. Ultra-Fast Precision TTL Comparator MXL1016. (2003). *Maxim Integrated Products* [Online]. Dosegljivo: http://datasheets.maximintegrated.com/en/ds/MXL1016.pdf. [Dostopano: 27. 06. 2014]

[16] J. F. Wakerly. ECL: Emitter-Coupled Logic. (2006) *Digital Design Principles and Practices, Fourth Edition, Supplementary sections* [Online]. Dosegljivo: http://www.ddpp.com/DDPP4student/Supplementary_sections/ECL.pdf. [Dostopano: 06. 06. 2014]

[17] P. Lee. Interfacing Between LVDS and ECL. *http://onsemi.com* [Online]. Dosegljivo: http://www.onsemi.com/pub_link/Collateral/AN1568-D.PDF. [Dostopano: 09. 06. 2014]

[18] M. Vidmar. Mikrokrmilnik. *www.antena.fe.uni-lj.si*. [Online]. Dosegljivo:http://antena.fe.uni-lj.si/literatura/VajeOK/USB2138/prirocnik/Mikrokrmilnik.pdf. [Dostopano: 02. 05. 2014]

[19] UM10120 LPC2131/2/4/6/8 User manual. (2012). *NXP* [Online]. Dosegljivo:http://www.nxp.com/documents/user_manual/UM10120.pdf. [Dostopano: 24. 05. 2014]

[20] Peripheral simulation. (2013) *KEIL Tools by ARM* [Online]. Dosegljivo: http://www.keil.com/dd/vtr/3880/9789.htm. [Dostopano: 03. 07. 2014]

[21] LPC2131/32/34/36/38 Product Data Sheet. (2011). *NXP* [Online]. Dosegljivo: http://www.nxp.com/documents/data_sheet/LPC2131_32_34_36_38.pdf. [Dostopano: 02. 07. 2014]

[22] „Assembly Language – Wikipedia, the free encyclopedia“ [Online]. Dosegljivo: http://en.wikipedia.org/wiki/Assembly_language. [Dostopano: 22. 07. 2014]

[23] P. Cockerell. ARM Assembly Language Programming [Online].
Dosegljivo: http://www.peter-cockerell.net/aalp/html/frames.html.
[Dostopano: 14. 07. 2014]

[24] Future Technology Devices International Ltd. FT231X – Full Speed USB to Full Handshake UART. (2014). *FTDI Chip* [Online].
Dosegljivo: http://www.ftdichip.com/Products/ICs/FT231X.html.
[Dostopano: 10. 08. 2014]

[25] T. Riemersma. Termite: a simple RS232 terminal. (2014). *CompuPhase* [Online].
Dosegljivo:http://www.compuphase.com/software_termite.htm.
[Dostopano: 29. 08. 2014]

[26] Fordahl-frequency control products. DIL 14 PACKAGE TCXO DFA 14-K (5 V) & DFA 14-L (3.3 V). (2014). *Datasheet Archive* [Online].
Dosegljivo:http://www.datasheetarchive.com/dl/Datasheets-IS68/DSAH00173888.pdf
[Dostopano: 26. 07. 2014]

[27] Fairchild Semiconductor. 74AC74, 74ACT74 Dual D-Type Positive Edge-Triggered Flip-Flop. (2008). *www.fairchildsemi.com* [Online].
Dosegljivo: https://www.fairchildsemi.com/datasheets/74/74AC74.pdf.
[Dostopano: 20. 07. 2014]

[28] Texas Instruments. Adjustable Micropower Voltage Regulators with Shutdown. (2012). *www.ti.com* [Online].
Dosegljivo: http://www.ti.com/lit/ds/symlink/lp2950-33.pdf.
[Dostopano: 22. 07. 2014]