

Primitivni osciloskop baziran na ARM mikrokontroleru

Toni Gualtieri, Predrag Teodorović, MSc *Fakultet tehničkih nauka, Novi Sad*

Sadržaj — U ovom radu prikazana je realizacija jednog primitivnog dvokanalnog osciloskopa baziranog na ARM mikrokontroleru, prikazivanje podataka na LCD displeju i snimanje vrednosti, koji se dalje mogu obraditi uz pomoć MATLAB softverskog paketa. Podaci se snimaju na NFS server, a posle završetka merenja iniciranjem prenosa fajlovi se preimenuju i šalju na FTP server.

Gljučne reči — A/D konverzija, ARM mikrokontroler, Linux operativni sistem, LPC2468, Osciloskop

I. UVOD

Osciloskop je neophodan instrument za ispitivanje i nalaženje greške u nekom elektronskom sklopu.

Osciloskop je instrument, koji prikazuje promene posmatranog signala u funkciji vremena. Može se meriti amplituda signala, frekvencija, čak i vreme porasta i opadanja istog. Stariji osciloskopi sa katodnim cevima su imali mogućnost da lepo prikazuju samo periodične signale do određene frekvencije, dok na slučajnim signalima bi izgubili sinhronizaciju i slika bi počela „trčati“ po ekranu. Stvar se promenila pojavom DSO – Digital Storage Oscilloscope (digitalnih osciloskopa) koji imaju mogućnost snimanja merenih podataka u memoriju.

Sastavni blokovi osciloskopa su :

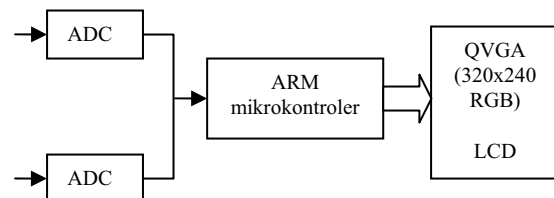
- Ekran za prikaz signala (Katodna cev ili LCD)
- Vertikalna kontrola
 - Podešavanje amplitude prikazanog signala (Volt/podeok)
 - Izdvajanje jednosmerne komponente
 - Podešavanje vertikalnog položaja za prikaz signala
- Horizontalna kontrola
 - Podešavanje vremenske baze za prikaz signala (Sekunda/podeok)
 - Mogućnost da jedan kanal bude prikazan na Y a drugi kanal na X osi

Toni Gualtieri, Fakultet tehničkih nauka u Novom Sadu, Trg Dositeja Obradovića 6, 21000 Novi Sad, Srbija (telefon 381-64-6124835, e-mail: toni.gualtieri@gmail.com)

Predrag Teodorović, Fakultet tehničkih nauka u Novom Sadu, Trg Dositeja Obradovića 6, 21000 Novi Sad, Srbija (e-mail: t_pedja@uns.ac.rs)

- Podešavanje horizontalnog položaja za prikaz signala
- Triggerovanje
 - Kontroliše mlazom za iscrtavanje signala
 - Može biti u automatskom režimu rada, ili da bude priključen na eksterni izvor signala.

Veoma važnu ulogu u merenju osciloskopom igra i sonda, koja može uneti greške u merenje.



Sl. 1. Blok šema uređaja

Na **Sl. 1** je prikazana principijelna blok šema uređaja. Projekat je rađen na razvojnom sistemu LPC2468 OEM firme Embedded Artists. Centralni procesor razvojnog okruženja je LPC2468 od proizvođača NXP. LPC2468 je mikrokontroler baziran na ARM7TDMI-S jezgru. Radni takt LPC2468 mikrokontrolera je 72MHz. Na ploči se nalazi NAND fleš memorija kapaciteta 128MB, 10/100Mbit Ethernet kontroler za mrežnu komunikaciju [1] i obilje drugih perifernih jedinica. Između ostalog, sistem poseduje 10bitni osmokanalni A/D konvertor na koji je povezan izlaz potencijometra koji se nalazi na razvojnoj ploči, i koji daje napon u opsegu od 0 do 3.3V. Vreme trajanja konverzije je 2.44μs. Ovaj ulazni signal je korišćen kao izvor signala za naš primitivni osciloskop. Za prikaz podataka i interakciju sa osciloskopom se koristi LCD displej rezolucije 240x320 piksela koji je osetljiv na dodir, pa se može koristiti i kao ulazni interfejs. Sistem se napaja jednosmernim naponom od +3.3V.

Ulazni signali su, kao što je već spomenuto, simulirani uz pomoć dva potencijometra, koji se nalaze na razvojnom sistemu i koji su povezani na ulaze A/D konvertora. Opseg ulaznih signala se kreće od 0V do 3.3V. Platforma za razvoj aplikacije je Linux operativni sistem, koji je portovan na LPC2468 mikrokontroler. Kreirana aplikacija primitivnog osciloskopa se zasniva na korišćenju drajvera za A/D konvertor, LCD displej i mrežni kontroler, a implementira obradu i prikaz talasnih oblika signala

dobijenih A/D konverzijom. Aplikacija za LPC2468 mikrokontroler je razvijena na radnoj stanici koja je povezana sa razvojnim okruženjem. Sa druge strane, na radnoj stanici je aktiviran NFS (Network File System) server, koji omogućava da se na razvojnom okruženju pokrene aplikacija, koja se nalazi na radnoj stanici. Ovim je omogućeno, da razvojno okruženje aplikaciju vidi kao da se nalazi u okviru lokalnog fajl sistema. Ovakav pristup omogućava skraćanje faze testiranja aplikacije jer korisnik ima mogućnost, da tek nakon što u potpunosti ispita funkcionalnost aplikacije, prebaci aplikaciju na lokalnu fleš memoriju na razvojnoj ploči kako bi u budućnosti mogla da se pokreće bez prisustva radne stanice.

II. ANALIZA PROBLEMA

Prvobitna ideja za realizaciju je bila, da se nakon svakog merenja signal prikaže na polju za grafik. Na taj način bismo približno u realnom vremenu uočili promenu signala. Međutim, usled ograničenih resursa vezanih za memoriju i brzinu odziva Linux operativnog sistema, nije primenjena ova verzija. Polje za prikaz bi se, u tom slučaju, osvežavalo 240 puta u sekundi (ScopeFieldX_max), što bi dovelo do smanjenja performansi sistema u celini.

Druga ideja za prikazivanje je da se znatno zakasni prikaz, tako što se prvo napuni bafer sa podacima, prikaže se i stoji na LCD displeju dok se bafer ponovo ne napuni i zatraži se zahtev za ponovnim crtanjem. Ovo znatno smanjuje broj osvežavanja displeja, međutim stvara neugodan efekat za prikaz, pojavljuje se efekat seckanja slike, uz gubitak prikaza u realnom vremenu.

Treća ideja, koja je i implementirana, je kombinacija prve dve. Konstantom SAMPLES se određuje koliko uzoraka treba da se uzme sa A/D konvertora, da bi se signal prikazao. Finim podešavanjem ove konstante se može postići da sistem u približno realnom vremenu prikazuje merene veličine, a da se time kontinualno iscrtava grafik na displeju i što je najbitnije, da se smanji opterećenje procesora.

III. PRORAČUNI

U ovom poglavlju su dati svi izrazi za proračun parametara koji su neophodni za ispravan rad uređaja. Parametri ScopeFieldX_max i ScopeFieldY_max su jedini ograničeni hardverom na kojem se koristi aplikacija. Maksimalna dozvoljena vrednost ScopeFieldX_max određena je brojem piksela po vertikalnoj osi displeja, dok za ScopeFieldY_max treba uzeti vrednost koja je 62.5% od broja piksela po horizontalnoj osi displeja. U slučaju prekoračenja ScopeFieldY_max došlo bi do preklapanja polja za iscrtavanje signala sa kontrolnim interfejsom.

Pošto polje za iscrtavanje signala ima 240 piksela (ScopeFieldX_max) broj osvežavanja sadržaja se računa na sledeći način:

$$\text{Broj_osvežavanj } a = \frac{\text{ScopeField X_max}}{\text{SAMPLES}} \quad (1)$$

ScopeFieldX_max – Dužina polja za iscrtavanje signala.

SAMPLES – Broj koji govori koliko treba odabrati A/D konverzije uzeti, da bi se osvežilo polje za iscrtavanje signala. Što je veći broj, to će se ređe osvežavati sadržaj

Da bi dobili što tačniji odraz merenog signala i da bi bilo lakše podešavanje sistema, neophodno je proračunati vrednosti za konstante.

$$\text{stepX} = \frac{\text{ScopeField X_max}}{\text{gridX}} \quad (2)$$

stepX – Širina kvadratića u mreži

gridX – Broj kvadratića po X osi

$$\text{stepY} = \frac{\text{ScopeField Y_max}}{\text{gridY}} \quad (3)$$

ScopeFieldY_max – Visina polja za iscrtavanje signala

stepX – Širina kvadratića u mreži

gridX – Broj kvadratića po X osi

Izrazi (2) i (3) na osnovu zadatih vrednosti izračunavaju dimenzije kvadrata koji će se prikazivati u pozadini kako bi bilo lakše pratiti vrednosti na X i Y osi.

Raspodela kvadratića u mreži po X i Y osi utiče na njihovu širinu i visinu. Da bi dobili pravilne kvadrate, treba da se sačuva odnos širine i visine. To se računa uz pomoć izraza (4).

$$\text{gridYnum} = \text{gridXnum} \cdot \frac{\text{ScopeField Y_max}}{\text{ScopeField X_max}} \quad (4)$$

gridYnum – Broj kvadratića u mreži po Y osi

gridXnum – Broj kvadratića u mreži po X osi

$$\text{ScopeY_FS} = \frac{\text{ADC_max}}{\text{gridYnum}} \quad (5)$$

ADC_max – Maksimalna vrednost koju prosleđuje drajver za A/D konvertor

$$Y_scale = \text{ScopeY_FS} \cdot \text{gridYnum} \quad (6)$$

Y_scale – Konstanta za skaliranje vrednosti

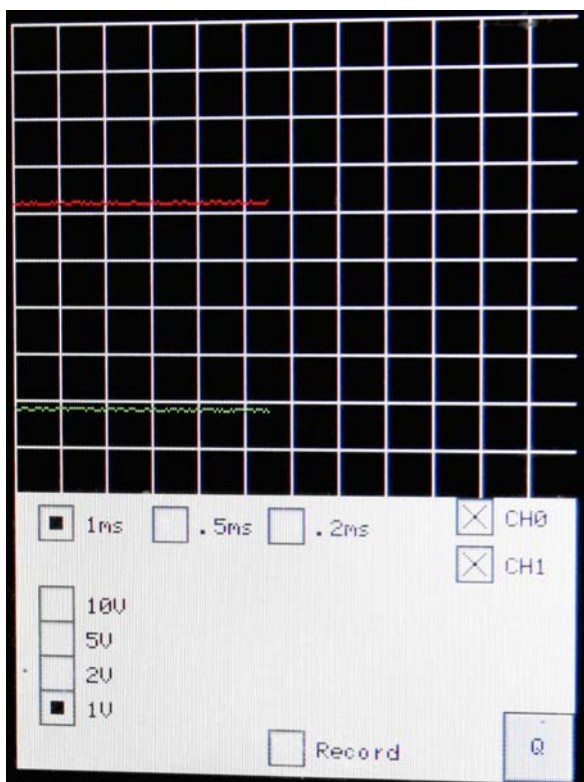
ScopeY_FS – Konstanta za prikaz pune skale

Drajver za očitavanje A/D konvertora vraća vrednosti u opsegu od 0 do 1023 (pri tome maksimalna vrednost 1023 odgovara maksimalnom ulaznom naponu od 3.3V). Usled toga, moramo proračunati konstante, da bi ceo opseg stao u polje za iscrtavanje.

$$\text{adc_new}[ch] = \frac{\text{mtk_read_adc}(ch)}{\frac{Y_scale}{\text{volts_div}}} \quad (7)$$

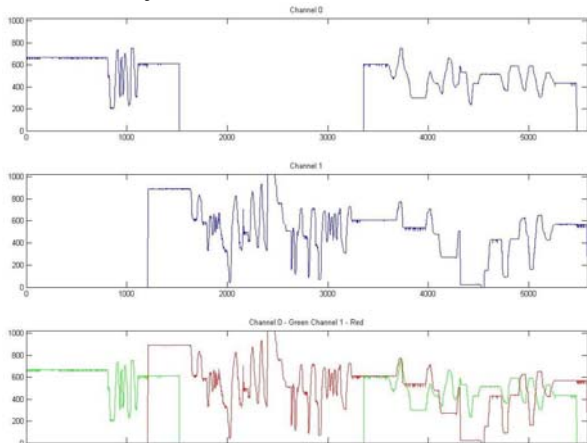
Adc_old[ch] je isto što i adc_new[ch], stoga formulu nećemo ponovo navesti.

Adc_old[ch] je stara vrednost sa A/D konvertora a Adc_new[ch] je nova vrednost, i oni se koriste u funkciji za iscrtavanje linije grafika.



Sl. 2. Izgled aplikacije u radu na razvojnom okruženju

Na Sl. 2 se može videti izgled aplikacije neposredno posle startovanja.



Sl. 3. Snimljeni signali prikazani u MATLAB softverskom paketu

Na Sl. 3 su prikazani talasni oblici napona na ulazima A/D konvertora uz pomoć MATLAB softverskog paketa. Pri tome su trenutne vrednosti ulaznih signala menjane korišćenjem potencijometra na razvojnoj ploči. Vrednosti na X-osi odgovaraju broju snimljenih uzoraka tj. odbiraka, dok se na Y-osi može videti stvarna vrednost signala, koja je očitana sa A/D konvertora (vrednost konverzije koju treba pomnožiti sa faktorom konverzije da bi se dobila vrednost u osnovnim mernim jedinicama-V).

IV. PRENOS I OBRADA PODATAKA

Mereni podaci se uz pomoć NFS (mrežnog fajl sistema) direktno snimaju na sekundarnu memoriju radne stanice, odakle se uz pomoć FTP protokola prenose na udaljenu lokaciju. FTP (File Transfer Protocol) je protokol, za

prenos fajlova korišćenjem TCP/IP protokola.

Linux skripta, pod nazivom transfer, je testirana uz pomoć portabilnog Xlight[3] FTP servera za Windows operativni sistem. Pre početka prenosa treba podesiti IP adresu na kojoj se nalazi FTP server, kao i korisničko ime i lozinku. Posle prenosa, snimljeni podaci sa talasnim oblicima snimljenih signala se automatski brišu sa radne stanice. Razvojni sistem sa LPC2468 mikrokontrolerom je preko LAN (Local Area Network) mreže povezan sa radnom stanicom, dok se FTP server, na koji će se slati fajlovi sa snimljenim signalima, može nalaziti na bilo kojoj udaljenoj lokaciji.

Dalja obrada i analiza signala se može vršiti u MATLAB softverskom paketu, kao na Sl. 3. Radi lakšeg importovanja merenih podataka u MATLAB, svaki kanal A/D konverzije se snima u poseban tekstualni fajl.

V. ZAKLJUČAK

Uređaj je uspešno realizovan, testiran i veće softverske greške su otklonjene.

Da bi ovaj sistem mogao služiti kao pravi osciloskop, potrebno je dodati još prateću elektroniku, za zaštitu ulaza A/D konvertora kao i stepena za prilagođavanje opsega ulaznog signala. Takođe, postoji mogućnost opcionog snimanja podataka na SD karticu prisutnu na razvojnom okruženju, umesto da se podaci šalju radnoj stanici.

Određivanje maksimalne frekvencije ulaznog signala, moralo bi se izvršiti eksperimentalno, jer ona ne zavisi samo od ulaznih karakteristika A/D konvertora već i od vremenskog odziva operativnog sistema na kome je implementirana aplikacija.

ZAHVALNICA

Izradi ovog rada je jako puno doprineo rad kolege Daniela Keslera, koji je razvio MTK simulator, radi lakšeg razvijanja i testiranja aplikacija za LPC2468 razvojni sistem.

LITERATURA

- [1] Embedded Artists, LPC2468 Datasheet <http://www.standards.nxp.com/products/lpc2000/datasheet/lpc2468.pdf>
- [2] Wikipedia, O osciloskopu <http://en.wikipedia.org/wiki/Oscilloscope>
- [3] Xlight Portable FTP server v3.5.5 http://dl3.afterdawn.com/download/0148bd628bcb65c575928aef67efe2ff/4c386f6b/xlight_v3.5.5.zip

ABSTRACT

This paper describes the implementation of a primitive two-channel oscilloscope based on the ARM microcontroller, displaying data on LCD and recording waveforms, which could be further processed using MATLAB software package. Sampled data is stored on a remote workstation by NFS protocol. Files are then renamed and sent to FTP server for further analysis.

PRIMITIVE OSCILLOSCOPE BASED ON THE ARM MICROCONTROLLER

Toni Gualtieri, Predrag Teodorović, MSc Faculty of technical sciences, Novi Sad