

# Elementarno, dragi Watsone

Odbrojavanje unazad je stiglo do nule. Pred vama je projekat instrumenta koji objedinjuje celu hardversku laboratoriju

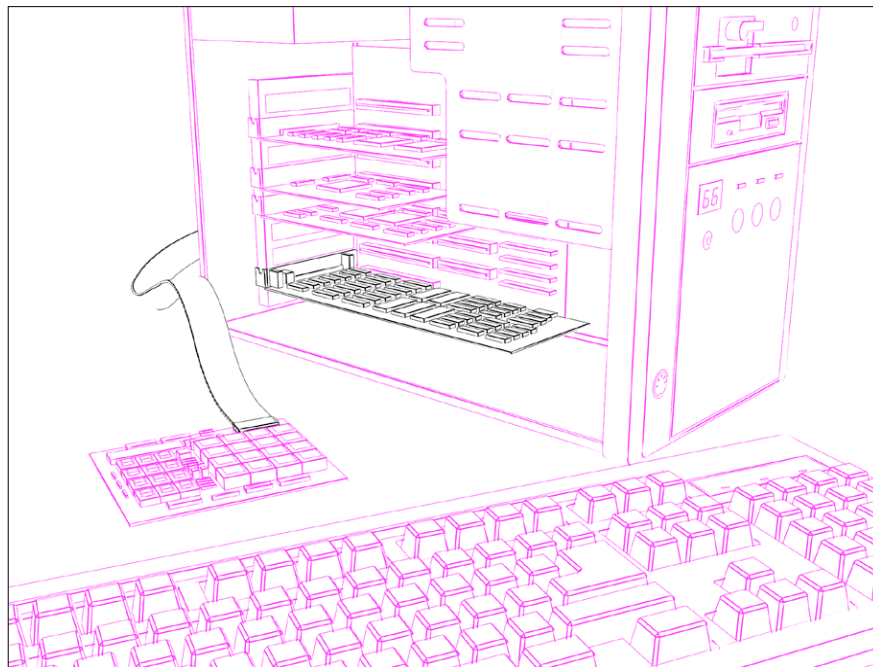
 **Voja Antić**

**U** ovom tekstu se nećemo baviti praktičnim problemima samogradnje razvojnog sistema, jer je taj aspekt obrađen u dodatku časopisa, nego pitanjima razvojnog procesa uopšte; kako vam ovaj uređaj može pomoći da ideju ili zahtev naručioa pretočite u prototip uređaja.

Gledano sa pozicije konstruktora, mikroprocesorski kontroleri imaju jednu zgodnu osobinu: bez obzira na namenu, hardverski su veoma slični. Svi imaju mikroprocesor, EPROM i RAM i sve se to vezuje uvek na isti način. Jedino ulazno/izlazni stepen donekle varira u zavisnosti od konkretne namene. Naravno, podrazumeva se da se program smešten u EPROM-u značajno razlikuje od slučaja do slučaja, mada vešti programeri grade svoje biblioteke u koje ugrađuju procedure koje kasnije koriste da bi skratili vreme razvoja i uštedeli trud.

Razvojni sistem koji vam ovde predstavljamo sadrži alatke koje su potrebne da bi se projekat mikroprocesorskog kontrolera brzo i lako doveo do kraja. Osim toga, neki delovi sistema (prvenstveno analizator) imaju i mnogo širu primenu: analiza rada čisto hardverskih sklopova, analiza protokola komunikacionih jedinica, lokalizovanje kvarova na pločama sa ili bez mikroprocesora, a u prošlom broju "PC"-ja već smo pomenuli da nas uskoro čeka samogradnja manjeg dodatka koji će ovaj razvojni sistem pretvoriti u analogni osciloskop sa funkcijom memorisanja talasnih oblika.

Gornja slika pokazuje tipičnu primenu razvojnog sistema *Sherlock*. Kartica je u slotu računara, a trakasti kabl povezuje konektor na njoj sa uređajem koji razvijamo (u daljem tekstu ćemo za taj uređaj koristiti naziv **cilj-**



**ni sistem**). Ako se radi o razvoju mikroprocesorskog uređaja, sonda na kraju trakastog kabla se ubacuje u podnožje za EPROM, jer ima isti raspored i oblik pinova. Posebnim kablom se dovede RESET izlaz, najbolje na elektrolitski kondenzator vremenske RC konstante kola za *reset*, a eksterni ulaz je univerzalni priključak i od konkretne namene zavisi gde će biti priključen. Uključujemo računar i, kada god treba prepisati izvršni fajl asembliranog programa u emulator, otkucamo EMUL i (samo prvi put) ime fajla, a ako nam zatreba kompletan razvojni sistem sa analizatorom i ostalim dodacima, pozivamo program SHERLOCK.

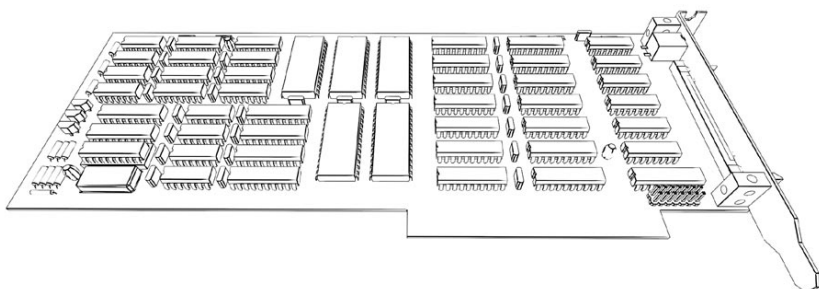
## Laboratorija

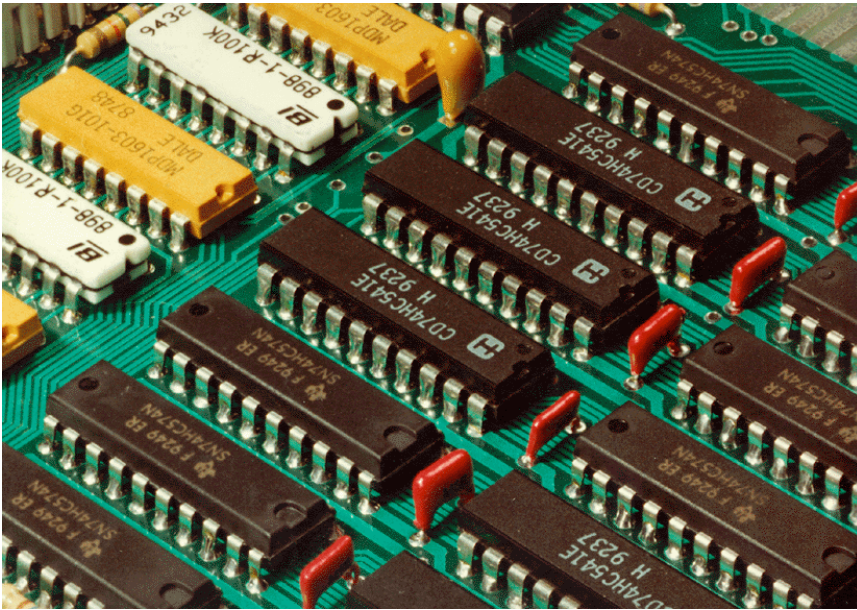
Ipak, priča mora da ima svoj početak. Pre nego što krenemo u opis delova koji čine razvojni sistem *Sherlock*, da kažemo šta će vam **osim** njega biti potrebno da biste radili na razvoju mikroprocesorske opreme. Potreban je, pre svega, kompjuter, IBM PC ili kompatibilan, bilo koje generacije (mada će sasvim

korektno raditi i najstariji XT, radi brzine grafike ipak je poželjno da bude neki od novijih modela) sa VGA karticom koja podržava rezoluciju 640\*480 u 16 boja, dok je za monitor poželjno (ali ne i neophodno) da bude u boji. Što se RAM-a tiče, nema posebnih zahteva: dovoljno je i minimalnih 640 K. Pošto je program raden u assembleru, na disku zauzima tek nekoliko stotina kilobajta, tako da ni tu nema posebnih prohteva. *Windows* okruženje nije neophodno, dovoljna je bilo koja verzija DOS-a (valjda niko više ne koristi DOS 2.11?).

Sledeća stvar koju treba da nabavite je assembler ili kompajler za procesor koji ste odabrali. Izbor ovakvih programa je prilično veliki, pa to nećemo smatrati preprekom. Poželjno je da assembler može da generiše izvršni kod u binarnom obliku, ali smo ipak pripremili i programe za prevodenje *Intel*-ovog .OBJ i standardnog .HEX formata u binarni.

Treća neophodna stvar je programator EPROM-a. Mnogi od nas koji se bave ovim poslom već imaju svoj programator (neki su ga sami napravili), a ako ne preterujemo u zahtevima i ne insistiramo na tome da programator mora da pokriva neke egzotične EPROM-e, možemo relativno jeftino da ga kupimo. Na kraju krajeva, pošto je programator ionako potreban samo po jednom za svaki projekat, i to na kraju razvojnog ciklusa, za prvo vreme možemo da se poslužimo i programatorom nekog prijatelja.





Toliko o tome šta *Sherlock* ne sadrži. A sad ćemo potanko objasniti svaku od funkcijskih grupa koju ovaj razvojni sistem sadrži.

### Emulator EPROM-a

Mada prilično jednostavan, ovo je najvažniji deo svakog razvojnog sistema. To je statički RAM kome se pristupa sa dve strane: PC može da upisuje u njega kao u RAM, a s druge strane ciljni sistem ga preko sonde i trakastog kabla "vidi" kao EPROM. Ne postoji mogućnost (a ni potreba) da se oba ova procesa obavljaju istovremeno: upis od strane računara traje tek neki deo sekunde, i za to vreme ciljni sistem nije u stanju da očitava sadržaj RAM-a. Za to vreme je najbolje držati mikroprocesor ciljnog sistema u RESET stanju, što se u našem razvojnom sistemu obavlja automatski svaki put kad se u emulator upisuje novi sadržaj, posebnim kablom koji se najčešće spaja sa RESET nožicom mikroprocesora.

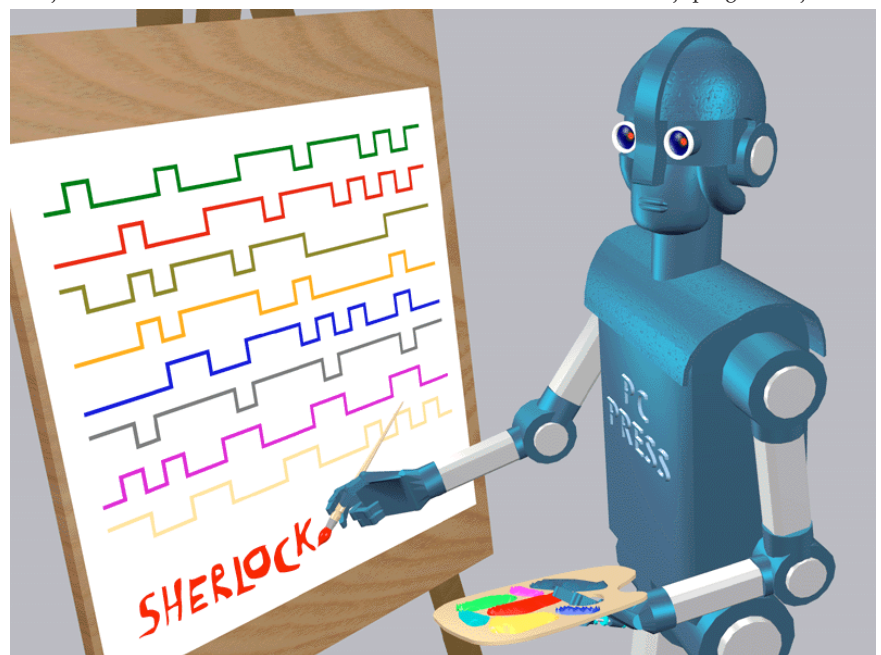
Evo kako izgleda primena emulatora u praksi: posle svake izmene u programu (kavkih u toku razvoja jednog kontrolera može da bude i na stotine) program se asemblira, otkuca se EMUL i pritisne ENTER. To je sve - posle delića sekunde kontroler koji razvijamo počinje da radi po novom programu.

### Analizator

Nema ničeg neobičnog u tome što je višekanalni digitalni analizator san svakog hardveraša, jer je to jedini instrument kojim se može videti šta se stvarno događa u digitalnom sklopu.

Univerzalni meri instrument je u svakom slučaju koristan, ali njime možemo je-

dino da izmerimo da li je napon za napajanje u dozvoljenim granicama i da li je neki od izlaza (koji se ne menja suviše brzo) na visokom ili niskom nivou. Osciloskop, koji je najvažniji instrument u analognoj tehnici, u laboratoriji za digitalni razvoj uglavnom skuplja prašinu ili, u najboljem slučaju, služi za to da svojim prisustvom fascinira slučajnog posetioca. Ovo nije preterivanje jer, ako ispitujemo neki mikroprocesorski sklop, jedina tačka na koju ima svrhe priključivati osciloskop je izlaz oscilatora, jer je to jedini ponovljiv signal u celom sklopu. Čak i *storage* osciloskop (koji može da "upamti" i na ekranu prikaže "zamrznut" talasni oblik) nije od velike koristi, jer nam jedan (pa još tako kratak) signal ne daje dovoljno informacija.

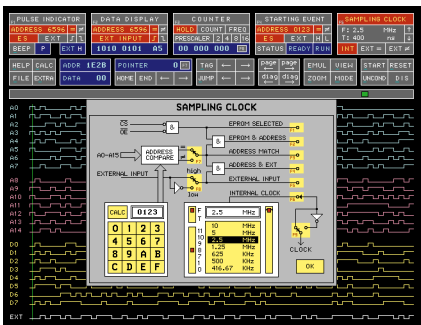


Zato je razvijena nova generacija merenih instrumenata, nazvanih analizatori, koji su prilagođeni digitalnim kolima. Oni prate veći broj tačaka istovremeno i mogu da memorišu signale u dužem vremenskom periodu. Postoje tajming (*timing*) i stejt (*state*) analizatori: prvi bacaju akcenat na precizno merenje vremena između dve pojave, a drugi (kakav je i ovaj naš) na snimanje niza logičkih stanja.

Evo kako to izgleda u praksi: analizator uzima uzorke logičkih stanja sa svojih ulaza i memoriše ih u određenim vremenskim intervalima, što znači da ovo uzorkovanje mora da se obavlja dovoljno često da se neki događaj ne bi "ispustio" između dva uzorka. Osim ovog, internog takta upisa, postoji i način da se takt dovodi spolja, preko eksternog ulaza ili da se definišu posebni uslovi pod kojima će se semplovanje obavljati.

*Sherlock* ima u svojoj ponudi obe mogućnosti, i to za interni takt počev od 1 Hz pa do 10 MHz, a za eksterni takt promenu logičkog nivoa, trenutak kad je emulator prozvan i kad je na adresnoj magistrali neko unapred definisano stanje.

Kompletan ciklus upisa se sastoji od 32768 semplovanih uzoraka po 24 bita, što u nekim slučajevima može da protekne prilično brzo, pa se postavlja pitanje kako ćemo "uloviti" baš onaj trenutak koji je od značaja za analizu. Ovaj problem rešava poseban sklop kojim se definiše uslov starta analizatora. Kao i kod eksternog takta, na raspolaganju su nam praktično sve kombinacije EXT, ES i ADDRESS ulaza, ali postoji i mogućnost bezuslovnog starta, koji je od koristi kad hoćemo da lociramo "mrtvu petlju" u koju je mikroprocesor uleteo. Tako smo ovu "noćnu moru" svih koji programiraju u ma-



šinskom jeziku pretvorili u problem koji je lako rešiv u jednom potezu.

Nemojte da se uplašite od navedenih izbora opcija, jer vam za njih neće biti potrebne hardverske izmene na ploči razvojnog sistema - sve se to radi mišem ili preko tastature, iz SHERLOCK softvera.

### Frekvencmetar

Frekvencmetar ćete sresti u mnogim elektronskim laboratorijama: to je uređaj koji meri i prikazuje učestanost nekog naizmeničnog signala. Osim toga, naš frekvencmetar može da se prekonfigurira i da radi kao brojač impulsa: jedina razlika je u tome što se brojač neće resetovati svake sekunde, nego će se stanje kumulativno uvećavati do ručnog reset-a.

Digitalni signal čiju učestanost treba izmeriti (ili čije impulse treba brojati) dovedi se na eksterni (EXT) ulaz. Tačnost merenja zavisi od preciznosti upotrebljenog kvarcnog oscilatora, a ona u tipičnom slučaju iznosi oko 10 ppm (*parts per milion* = delova na milion). Ugrađeni preskaler može da se upotrebi za proširenje mernog opsega, ali se onda smanjuje rezolucija očitavanja.

### Indikator impulsa

Nikad ne treba procenjivati korisnost nekog instrumenta na osnovu njegove složenosti: indikator impulsa je vrlo jednostavan dodatak analizatoru, ali je od velike koristi kod razvoja softvera. Njegov zadatak je da nam javi svaki put kad se steknu uslovi koje smo unapred definisali, recimo kad mikroprocesor očitava sadržaj (drugim rečima, kad izvršava instrukciju) na nekoj određenoj adresi u sistemu koji testiramo. To su najčešće instrukcije koje se nalaze neposredno posle uslovnih skokova u programu: recimo, kad je pritisnut neki taster na tastaturi kontrolera ili kad je stigao novi bajt na serijski port. Često je od presudne važnosti informacija da li je program uopšte registrovao promenu, a to će nam dojaviti indikator impulsa pojavom slova P (ekvivalent oznaci PULSE na logičkoj sondi) i zvučnim signalom, ako smo ga uključili.

### Pokazivač stanja memorije

Ovo je još jedan strogo namenski dodatak koji će nam tokom rada pomoći da vidimo što više parametara sistema koji razvijamo. Možemo da odaberemo jednu adresu, pa će, svaki put kada ciljani sistem prozove tu adresu i kada je na EXT ulazu zadata (rastuća ili opadajuća) ivica, biti registrovani i prikazano trenutno stanje na magistrali podataka. Ako EXT ulaz spojimo sa WR nožicom na eksternom RAM-u ciljnog sistema, imaćemo stalni uvid u sadržaj jedne memorijske lokacije, a ako EXT ulaz prebacimo na signal kojim se prozivaju izlazni portovi, imaćemo stanje jednog osmobitnog izlaznog porta.

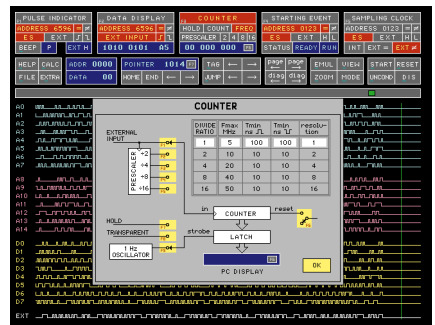
### Prateći softver

Uz komplet delova za gradnju razvojnog sistema *Sherlock* isporučuje se i disketa sa nekoliko izvršnih fajlova. Tu su programi EMUL, koji služi za brzi prepis izvršnog koda generisanog od strane asemblera u emulator, zatim programi OBJ2BIN i HEX2BIN kojima se vrši prevodenje objektnih i heksadecimalnih ASCII fajlova u binarne (koji jedino mogu da se prepisuju u emulator), i na kraju SHERLOCK, na čijem objašnjenju ćemo se malo duže zadržati.

Ovaj program sadrži kompletnu podršku svih delova razvojnog sistema. Neka ekrani iz ovog programa posluže kao ilustracija postavljanja parametara analizatora: pored toga što se parametri biraju u malim prozorima iznad dijagrama, udobnije je otvoriti velike prozore, tim pre što je na njima grafički predstavljena ekvivalentna šema sklopa. Nju možemo da modifikujemo mišem, tako što menjamo položaje preklopnika na ekranu. Za hardveraše ovo je lakši način izbora parametara od tabelarnog.

Pojedine tačke u dijagramu možemo da obeležimo i da im dajemo komentare, koji će se pojaviti svaki put kad postavimo kursor na neku od njih. Moguće je snimiti sve semplovane sekvence na disk, a za hakere je posebno interesantna opcija da se u radni prostor analizatora upiše segment memorije iz računara, ako analiziramo rad PC-ja.

Postoji i veliki broj podataka koje ćemo samo nabrojati: kalkulator za konverziju brojnih baza, VIEW koji omogućava brzi pregled ASCII fajlova sa udobnim višesmernim skrolovanjem pomoću miša, kao i SEARCH funkcija za pretraživanje semplovane sekvence. Imamo takođe mogućnost da zumiramo, skrolujemo dijagram horizontalno na nekoliko načina, premeravamo broj semplova između dve bilo koje tačke na dijagramu, da signale bojimo po želji i da im dajemo proizvoljna imena. Za sve funkcije je podržan rad mišem i tastaturom.



Za kraj ostaje da se zahvalimo svima koji nam ukazuju poverenje i kreću u ovu akciju. Časopis "PC" će ubuduće posvetiti dovoljno prostora konstruktorima, pre svega stalnom hardverskom rubrikom, a i novim projektima koji su već u pripremi. A što se tiče ovog projekta, za sve čitaoce kojima je potrebna pomoć prilikom gradnje ili puštanja uređaja u pogon organizovali smo otvorena vrata u redakciji časopisa, svakog ponedeljka i četvrtka od 12 do 18 časova (ili dokle bude potrebno). U ovo vreme biće prisutan i konstruktor uređaja, koji će vam rado ponuditi besplatnu pomoć i obuku u rukovanju *Sherlock*-om.

### Komponente

Šta je sve potrebno za gradnju razvojnog sistema? Prvo ćemo nabrojati stavke koje nisu standardne i koje ne možete da kupite u prodavnicama elektroničarskog materijala: to su štampana ploča, programirani PIC mikroprocesor i disketa sa PC softverom. Cena ovog osnovnog kompleta je 330 dinara.

Druga mogućnost je kupovina svih komponenti za gradnju - komplet, uz ploču, PIC i disketu, uključuje i sva integrirana kola, tranzistore, pasivne komponente, konektore i kablove. Cena ovog kompleta je 810 dinara.

Možete, naravno, da kupite i kompletan, sklopljen uređaj, što uključuje i šestomesečnu garanciju. Cena je u tom slučaju 1140 dinara. Sve navedene cene su bez poreza na promet, i podložne su promenama ukoliko bi došlo do pomeranja na deviznom tržištu.

Za sve ostale informacije i predračun obratite se redakciji "PC"-ja (Krušedolska 5-7, telefoni 011/451-263 i 436-855) svakog radnog dana od 9 do 19 časova. Obzirom da je u ovom trenutku teško prognozirati odziv samograditelja, obezbeđen je ograničen broj kompleta koji će biti isporučivani onim redosledom kako narudžbenice budu pristizale. U slučaju da odziv bitno prevaziđe planirani, maksimalni rok isporuke neće preći 30 dana.