

brojačem velike brzine. Model CPU44 kontrolera CQM1 ima ugrađena 2 interfejsa za ulaze sa apsolutnih enkdera, odnosno uređaja koji na izlazu daje višebitnu digitalnu vrednost. Ovi portovi imaju kapacitet obrade 12-bitnih signala, a mogu se koristiti i za ulaz pulsnih signala u režimu generisanja interapta pomoću ugrađenog brzog brojača.

Memorija CQM1 kontrolera podeljena je u dva osnovna sektora, i to su korisnička memorija, u kojoj je smešten korisnički program, i I/O memorija. I/O memorija dalje je podeljena u veći broj sektora. Ona je dužine 16 bita, i lokacijama unutar nje pristupa se pozivanjem reči odnosno bitova. Izuzetak je radna zona, čijim podacima možemo pristupiti samo u obliku reči. Uloga I/O memorije je rad sa signalima iz spoljašnje okoline (ulaznim i Izlaznim), a koristi se i kao memorija podataka koji se koriste u programu. U njoj se takođe nalazi i sektor sa ugrađenim tajmerima i brojačima, kao i veliki broj bitova namenjenih pozivanju funkcija koje su sistemski podržane. Primeri ovih bitova su bit koji menja stanje na svakih 100 milisekundi, ili bit koji je aktivan kada je u toku prenos podataka serijskom vezom.

U samom CPU modulu ne postoji ugrađen sat realnog vremena, ali je njega opciono moguće ugraditi montiranjem memorijske kasete sa ugrađenim časovnikom. Jedan deo I/O memorije čuva sadržaj prilikom prestanka rada kontrolera, i u njega je potrebno skladištiti podatke koji treba da zadrže vrednost u tom slučaju.

Za detaljnije informacije obratite se firmi MIKROKONTROL iz Beograda koja je autorizovani distributer OMRON-ove opreme za industrijsku automatizaciju.

Kontakt adresa

MIKRO KONTROL

Computerized Industrial Automation
Yugoslavia, 11000 Belgrade, Dositejeva 7a
tel./fax: 381 11 3283732, 381 11 3283840

OMRON Authorized Distributor

Vizuelni alati

podrška embedid aplikacijama

Aleksandar Tošić

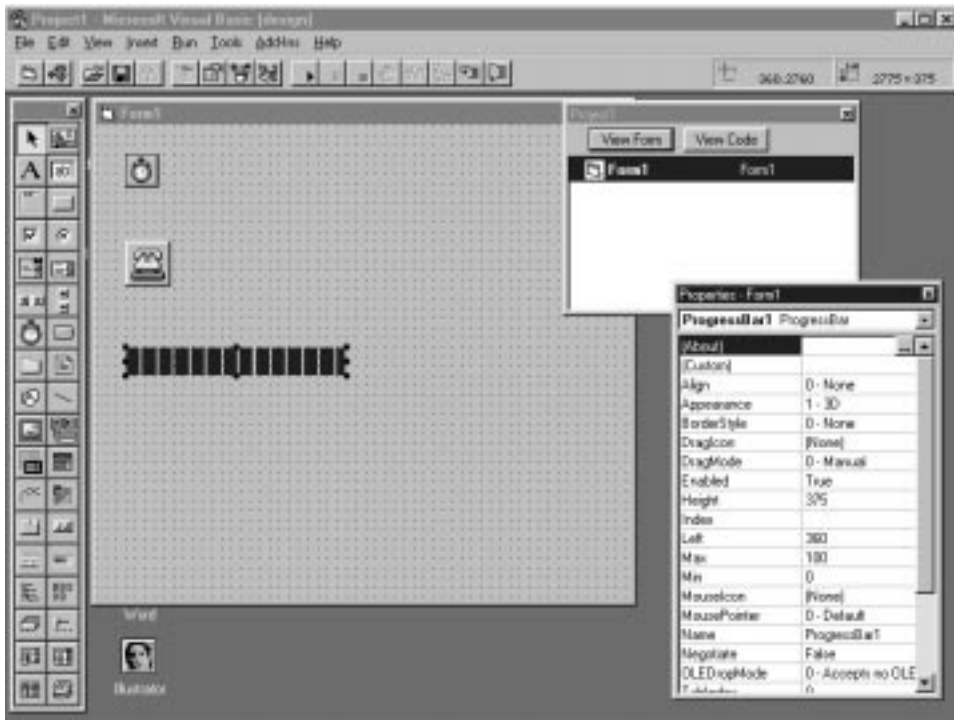
Svaka primena mikrokontrolera se odnosi na prikupljanje određenih podataka, obradi tih podataka čija je posledica izvršavanja određene aktivnosti. Obično se mere neke fizičke veličine (AD konverzija) i vrši upravljanje na osnovu izmerenih vrednosti. Periferali koji su deo embedid aplikacije obično imaju ograničene mogućnosti, a i sam mikrokontroler nije tako moćan kao PC koji može daleko brže i efikasnije da odradi neke poslove. Dakle, polazi se od činjenice da PC i mikrokontroler mogu da komuniciraju na neki način. Pitanje je samo kako?

Serijska komunikacija I PC i mikrokontroler poseduju hardverski deo koji je namenjen za asinhronu serijsku komunikaciju. Pošto cilj ovog članka nije detaljno objašnjavanje mehanizma rada serijske komunikacije to će detalji biti opisani na nekom drugom mestu. Ovde smo se opredelili za jednu stranu – programiranje na PC-ju, ali sa naglaskom na mogućnostima koje podržavaju rad embedid aplikacija. Odmah se može povući jasna crta između programiranja u DOS-u i WINDOWS-u. Iako je ovaj poslednji napravio pravu malu revoluciju u oblasti programiranja, što je kao posledicu imalo potrebu da se dobar deo programera morao edukovati za nove programske pakete, još uvek ima aplikacija koje rade pod DOS-om kao podrška embedid aplikacijama. Obično su programi u DOS-u pisani u Turbo Pascalu, C/C++ i assembleru kada je to potrebno. Kao globalna podrška programerima za izradu korisničkog interfejsa napravljena je biblioteka funkcija koja se zove Turbo Vizija. Na taj način je i DOS pružao određene mogućnosti za solidnu vizuelizaciju podataka. Pojavom Windows-a stvari su se mnogo promenile. Programski jezici kao

Embedid aplikacije mogu samostalno da obavljaju određene zadatke ali je vrlo poželjno da komuniciraju sa PC-jem omogućujući operateru da »vidi šta se dešava na terenu«.

što su **Borlandov Delphi, C++ i Cbuilder i Microsoftovi Visual Basic i Visual C++**, kao najpoznatiji predstavnici, uneli su mnoge novine u oblast programiranja. Programiranje se više ne svodi više samo na monotono ispisivanje linija programskog koda već postoje mnogi gotovi alati za izgradnju same aplikacije kao celine koja se sastoji iz korisničkog interfejsa koji mora odgovarati funkciji, odnosno onome šta aplikacija mora da odradi. Samo izgrađivanje aplikacije postalo je mnogo interesantnije i brže.

Podaci su stigli, šta dalje? Pretpostavimo da su izmerene vrednosti neke fizičke veličine (na primer temperature) »stigle« do PC-ja. Šta dalje uraditi sa njima? Postoji nekoliko stvari koje je poželjno obaviti, a što opet zavisi od potreba same aplikacije. Kao prvo dobro bi bilo da se prispeli podaci sačuvaju, jer će možda u budućnosti iskrsnuti potreba da se pregledaju podaci koji se trenutno prikupljaju, na primer statistička obrada itd. Programska podrška ovakvoj potrebi su baze podataka ili datoteke uopšte. Kao drugo, podaci se trebaju na neki način vizuelno predstaviti jer samo »viđenje« mnogo doprinosi razumevanju situacije »na terenu« i eventualno daljnjem povratnom uticaju samog operatera koji vrši nadgledanje i kontrolu nekog procesa. Programska podrška se bazira na **GDI** – Graphic Device Interface. Treća stvar koja će sigurno naći primenu su tajmeri. Svi gore nabrojani programski paketi imaju podršku za rad sa tajmerima. Postoji još obilje alata koji olakšavaju rad i čitav postupak čine efikasnijim. To su na primer funkcije za rad sa printerom, stablom direktorijuma itd. Drugim rečima gotovo sve što srećemo u svakodnevnom radu sa Windows-om kao operativnim sistemom možemo koristiti pri programiranju. Najniži nivo programiranja Windows-a su API



poslužiti za vizuelizaciju neke fizičke veličine u nekom opsegu.

Primenom odgovarajućih kontrola i pisanjem programskog koda mogu se realizovati korisne PC aplikacije za podršku embedid sistemima, čak i u VB. Možda je najveća njegova prednost da se relativno brzo uči. Takođe ne treba zaboraviti da je veliki adut VB njegova otvorenost. I na kraju, ipak je to Microsoftov mezimac, ma koliko to iritirajuće zvučalo.

Dakle, namera ove novouvedene rubrike je upoznavanje sa alatima iz te klase. Za početak – Visual Basic ■

(Application Peripheral Interface) funkcije. Mnoge ove funkcije se ne koriste direktno pri programiranju, već je u svaki programski paket ugrađena neka vrsta »omotača«. Često se kombinuju mogućnosti samog programskog jezika i API (Application Programming Inteface) funkcija.

Visual Basic – za i protiv Na kraju još jedna važna stvar – multithreading (paralelni rad sa više zadataka istovremeno). Podržavaju ga nove verzije gore navedenih programskih paketa osim Visual Basic-a. Još jedna mana Visual basica (VB) - ne podržava rad sa pokazivačima. Da li je zaključak da je zbog toga VB neprikladan za saradnju sa embedid sistemima. Nipošto! Ako ste bili u prilici da čitate članke u domaćim stručnim časopisima o akviziciji podataka, verovatno ste iznenađeni da nije redak bio slučaj primene VB. Naime, postoji »kontrola« u VB koja podržava asinhronu serijsku komunikaciju. Na Slici 1. Je prikazana radna površina VB 4.0 u fazi izrade aplikacije, na kojoj se nalaze tri kontrole: *Timer*, *Communications* i *ProgressBar* kontrole. Po imenu kontrola se može zaključiti kakva je njihova funkcija. Prva kontrola (*Timer*) je namenjena za merenje vremena (baza je 1ms), druga (i najvažnija) kontrola (*Communications*) je namenjena za rad sa asinhronom serijskom komunikacijom i treća kontrola (*ProgressBar*) može lepo

Deborah Kurata

Microsoft Visual Basic 4.0

Rad sa objektima

Knjiga za naprednije korisnike. Autor, Deborah Kurata, glavni konsultant i suosnivač firme InStep Technologies, na razumljiv način predstavlja objektnu mogućnosti Visual Basic-a 4.0 i primere kako ih koristiti u praksi. Uvod u objektno orijentisane koncepte je predstavljen u prvom delu knjige. Posebno treba istaći način na koji autor u drugom delu izlaže organizaciju timskog rada na projektovanju objektno orijentisanih aplikacija, zalazeći na taj način »iza scene« samog postupka programiranja dajući

jasne smernice na onome što prethodi sprovođenju ideja u programski kod. Dok su prva dva poglavlja prikladna za sve nivoe znanja programera koji koriste VB 4.0., za čitanje trećeg dela knjige se podrazumeva da poznajete VB 4.0., u kojem se praktično primenjuje OOP (Objektno Orijentisano Programiranje): pravljenje sopstvenih klasa, OLE servera, kostura aplikacija, »čarobnjaka«, baza podataka (DAO i ODBC), itd. Uz knjigu se dobija i disketa sa gotovim projektima koja čitaocu olakšava praćenje materije koja se izlaže. Gotovi projekti su dobra osnova za razvijanje vlastitih aplikacija. Razumevanjem materije u knjizi se dobija mnogo više, opšte saznanje da se napisani kod može ponovo iskoristiti (ako se napiše sa takvom namerom) što je karakteristično za OOP. Tako vas autor neprekidno »savetuje« da gledate unapred, razmišljajući da ono što sad radite bude korisno za pravljenje budućih aplikacija. (A.T.)

Izdavač:

Mikro knjiga

P.Fax 20-109 tel: 011 3540-544

11030 Beograd fax: 011 542-516

