

# Taster

## sa

## dve brzine



Autor ima digitalni časovnik sa tasterima za podešavanje vremena - pritisne jednom, recimo, taster za podešavanje minuta i displej se uveća za 1 minut. Ako se taster duže drži pritisnut, minuti se odjednom počnu brzo pomerati, pa se s ovim "dvo brzinskim" tasterom vreme lako namešta.

**Kako u PBASIC-u napraviti ovu funkciju tema je ovog članka.**

**K**omandu BUTTON kod Parallax-ovog BASIC STAMP-a korisnici često zaobilaze, uglavnom zbog nesnaljenja oko njene obimne i zbujujuće sintakse:

**BUTTON pin, downstate, delay, rate, bytevariable, targetstate, address**

Namenjena je za očitavanje stanja tastera, eliminaciju treperenja kontakata (debounce), auto-repeat funkciju (kao kod računarske tastature: kad pritisnete neki taster, na ekranu se pojavi karakter - ako taster i dalje drite pritisnutim, posle nekog vremena (Delay) isti karakteri će se ispisivati velikom brzinom.

Parametri ove naredbe su:

- **pin** (varijabla ili konstanta koja opisuje koji pin se koristi, vrednost je od 0 do 7)
- **downstate** (varijabla ili konstanta koja opisuje stanje koje treba očekivati kada taster bude pritisnut, vrednost je 0 ili 1)
- **delay** (varijabla ili konstanta koja određuje koliko dugo taster mora biti pritisnut da bi proradila auto-repeat funkcija, ima vrednost 0 do 255. Za 0, ne radi ni debounce ni auto-repeat, za 255, ne radi samo auto-repeat.
- **rate** (varijabla ili konstanta koja određuje trajanje auto-repeat funkcije, i ima vrednost 0 do 255)
- **bytevariable** je radna varijabla za naredbu button i mora se postaviti na 0 pre prvog korišćenja naredbe

Button. Kako je ova naredba predviđena za korišćenje unutar programskih petlji, ovo brisanje mora se obaviti pre ulaska u petlju

- **targetstate** je varijabla ili konstanta koja određuje u kom stanju taster treba da bude da bi se iz naredbe "skakalo" na labelu **address** i moe imati vrednost 1, što znači da je taster pritisnut ili 0, što znači da taster nije pritisnut
- **address** je labela (oznaka) koja upućuje tok izvršenja programa za slučaj kada je taster u eljenom stanju (targetstate)

Kao što je već napomenuto, naredbu Button treba koristiti unutar programske petlje: pri svakom prolasku kroz petlju, proverava se stanje definisanog ulaza (pin) - kada se prvi put pojavi zadati downstate, aktivira se debounce. Po isteku tog intervala, gleda se parametar targetstate - tako se bira skok na labelu ako je taster pritisnut, ili skok ako nije pritisnut.

Primer koji će oigledno ilustrovati gornje izlaganje dat je u Listingu. Pin 0 Basic Stampa treba vezati na LED preko rednog otpornika 1k prema masi, a Pin 1 preko otpornika 4k7 na +Vcc i tasterom prema masi.

Opis primenjene naredbe Button:

na Pinu 1 očekujemo downstate 0 - kad se ono desi, izvršice se delay u trajanju od 20 ciklusa (za sve to vreme taster mora biti u downstate stanju). Ako po isteku delay perioda taster bude i dalje u downstate stanju, obavljaće se auto-repeat sa periodom od 3 ciklusa, radna varijabla je Tmp0, a ako taster nije pritisnut (targetstate = 0), program "skaće" na labelu Loop.

Programski to izgleda ovako: Dok taster

NIJE protisnut, naredba Button se grana na labelu Loop, zato što downstate 0 nije dostignut, a pošto se grananje vrši kada taster NIJE pritisnut dolazi do grananja na labelu Loop. LED ne svetli, jer se izvršava samo naredba Debug koja na ekranu prikazuje samo stanje varijable Tmp0, koje je jednako 0.

U momentu kada se pritisne taster, downstate je postignut, pa počinje da teče Delay, što se vidi po tome da Tmp0 dobija vrednost 20. Ako drimo pritisnut taster za kratko, videćemo da se vrednost Tmp0 smanjuje - teče nam vreme predviđeno za Delay.

Ako se taster otpusti pre nego Tmp0 padne na 1, on se opet postavi na 0 i jedina reakcija da je naredba Button očitala stanje tastera je LED koja je jednom kratko zasvetlela (izvršila se naredba Pulsout).

Ako, naprotiv, taster drimo stalno pritisnut, posle odbrojavanja Tmp0 od 20 do 1 videćemo se da Tmp0 počinje da odbrojava od 3 do 1 (počeo je auto-repeat) i LED učestalo treperi. Eto, dakle, mogućnosti za "sporo" i "brzo" nameštanje parametara u vašoj aplikaciji, a sve to samo sa jednim tasterom.

### Listing

```

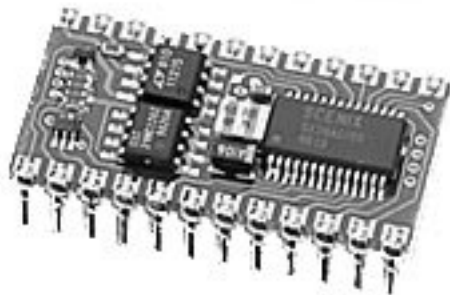
'BUTTON.bas
'Kratka ilustracija naredbe
'*****
Symbol Tmp0 = B0
Symbol LED = 0
Loop:
Debug Tmp0
Button 1,0,20,3,Tmp0,0,Loop
Pulsout LED,500
Goto Loop

```

# BASIC STAMP II

Novost u ponudi Parallax-a:

**BASIC 2 modul sa 16KB memorijskog prostora i taktom na 50MHz!**



**Parallax**

## BASIC Stamp IIsx

Itaoci koji prate proizvođače firme Parallax bazirane na njihovom PBASIC-u i BASIC STAMP modulima se čaju se da je posle pojave BS2-IC modula i unapređenje koje je doneo u odnosu na BS1-IC: više memorije, brže izvršavanje, veći broj I/O linija ... usledila podu a pauza bez značajnijih novosti, sve dok nije objavljeno da Parallax sa firmom Scenix Semiconductors saraduje u realizaciji SX mikrokontrolera koji je

oslonjen na arhitekturu popularne Microchipove serije PIC16C5x. Osim najavljenog visokog procesorskog takta i internih doterivanja arhitekture, obezbeđena je i puna podrška kroz razvojnu opremu, pa i kroz PBASIC.

Šta donosi BASIC Stamp IIsx najbolje se vidi iz tabele ispod. Specifičnost novog rešenja je u tome što BS2-SX ima 16KB

EEPROM-a, kojeg koristi u segmentima od po 2kB, tako da praktično možete napisati 8 programa od po 2kB dužine svaki.

Ti programi dele isti RAM, tako da mogu međusobno komunicirati i razmenjivati podatke, ali samo jedan

- konstanta od 0 do 7) kojom se zadaje koji od 8 programa će se startovati. Pri resetu ili startu, izvršenje kreće od programa 0 (Master program)

■ **PUT location, value** (location od 0 do 63, pokriva sve lokacije u Scratch Pad RAM-u) služi za odlaganje podataka u RAM.

■ **GET location, variable** (location od 0 do 63, pokriva sve lokacije u Scratch Pad RAM-u) služi za čitanje sadržaja zadate lokacije iz Scratch Pad RAM-a

program može biti aktivan u datom trenutku.

Tri nove naredbe su:

■ **RUN program** (gde je program varijabla

Treba napomenuti da su BS2 i BS2sx kompatibilni, samo treba obratiti pažnju na instrukcije koje sadrže tajming, zbog razlika u brzini kontrolera.

	BS1 - IC (Rev.D)	BS2 - IC	BS2 - SX
Kontroler	Microchip PIC16C56	Microchip PIC16C57	Scenix SX28AC
Brzina izvršenja programa	2000 u sec.	4000 u sec.	10 000 u sec
Brzina procesora	4 MHz	20 MHz	50 MHz
Memorija za program	256 Bytes	2kB	16kB (u segmentima po 2kB)
RAM	16 Bytes	32 Bytes	32
Scratch Pad RAM	Ne	Ne	64
I/O linija	8	16	16
Potrošnja pri 5V	1,4mA / 40uA Sleep	8mA / 100uA Sleep	60mA / 200uA Sleep
Strujno opterećenje po pinu*	20mA / 25mA	20mA / 25mA	30mA / 30mA
Ukupno strujno opterećenje*	40mA / 50mA	40mA / 50mA	60mA
Pakovanje modula	14pin SIP	24pin Dip	24pin Dip
Broj komandi	32	36	39
Veza prema PC	Paralelni port	Ser. port (9600Baud)	Serijski port (9600Baud)
Editor / Programiranje	Stamp.exe (14K)	Stamp2.exe (15K)	Stamp2sx.exe (15K)

\* Source/Sink

Za korisnike BS2 pojavila se i verzija za rad pod Win98, pa ko voli ...

[www.parallaxinc.com](http://www.parallaxinc.com)