



MICROCONTROLLORE AT89C2051

Nello Alessandrini

Un piccolo-grande processore con
un economico sistema di sviluppo.

5^a parte

Premessa

In questo numero verranno presentate alcune istruzioni del microcontrollore 2051 e tre programmi di esempio per l'uso del TIMER 0.

Inoltre come già iniziato nel numero precedente si allegano le istruzioni del set 51, ma questa volta separate secondo le loro funzioni (vedi tabelle di figura 1, 2, 3).

Programma n.1

Questo programma consente di contare gli impulsi del tasto T0 visualizzandoli sul video collegato alla seriale. Dopo la premuta di T0 (ricordarsi di tenere il ponticello strip verso T0) per poter avere un successivo impulso è necessario premere il tastino P0 (corrispondente al PORT P3.5). Pertanto sul video se si vorrà vedere incrementare il conteggio occorrerà premere alternativamente T0 e P0.

```

;Programma IMPULSO.SRC      Conteggio degli impulsi T0-P3.4 tramite
;                               la funzione Counter del timer/counter 0
;                               e visualizzazione sul monitor
;
;
ADDR      EQU      1000h      ;Inizio codice in EPROM.
;
;
;           org      ADDR+0000h      ;Vettore di reset.
;           ljmp     START           ;Salto all'inizio del codice.
;
;           org      ADDR+0035h      ;Inizio del codice.
;
;
START:    clr      TR1           ;Timer/counter T1 in off.
          clr      TR0           ;Timer/counter T0 in off.
          mov      SCON,#01010010b
          mov      TMOD,#00100101b
          mov      PCON,#00000000b
          mov      TL1,#0FDh
          mov      TH1,#0FDh
          mov      TL0,#00h      ;Si resettano i registri di T0
    
```



```
        mov     TH0,#00h          ;
        setb   TR1                ;Abilitazione alla comunicazione
        setb   TR0                ;Abilitazione al Counter
;
        mov     DPTR,#MES1
        acall  VISMES              ;Visualizzazione del messaggio.
MAIN1:  mov     DPTR,#MES2
        acall  VISMES              ;Visualizzazione del messaggio.
        mov     A,TL0              ;Visualizzazione sul monitor
        mov     R1,A
        mov     B,#100             ;Determinazione delle centinaia.
        div    AB
        mov     R2,A
        add    A,#48               ;Trasmissione delle centinaia.
        acall  TXBYTE
        mov     B,R2
        mov     A,#100
        mul    AB
        mov     R2,A
        mov     A,R1
        clr    C
        subb   A,R2
        mov     R1,A
        mov     B,#10             ;Determinazione decine e unita`.
        div    AB
        mov     R2,A
        add    A,#48               ;Trasmissione delle decine.
        acall  TXBYTE
        mov     B,R2
        mov     A,#10
        mul    AB
        mov     R2,A
        mov     A,R1
        clr    C
        subb   A,R2
        add    A,#48               ;Trasmissione delle unita`.
        acall  TXBYTE
        mov     A,#13             ;Trasmissione di un CR.
        acall  TXBYTE
        sjmp   MAIN1
;
;
;*****
;** TXBYTE: Routine di trasmissione di un byte sulla seriale.
;** IN: ACC contenente il byte da trasmettere.
;** OUT: Nessuno.
;*****
TXBYTE:  jnb    TI,TXBYTE          ;Attesa per trasmettere.
        clr    TI                 ;Reset del flag TI.
        mov    SBUF,A             ;Byte nel buffer di trasmissione.
        ret
;
;
;*****
;** RXBYTE: Routine di ricezione di un byte dalla seriale.
;** IN: Nessuno.
;**OUT: ACC contenente il byte ricevuto.
;*****
RXBYTE:  jnb    RI,RXBYTE          ;Si attende fino a che non si
        mov    A,SBUF             ;riceve il byte.
        mov    A,SBUF             ;Si legge il byte dal buffer
        clr    RI                 ;della seriale.
        ret                       ;Reset del flag RI.
;
```



```

;*****
;** VISMES: Routine che visualizza un messaggio.
;** IN: DPTR contenente l' indirizzo del messaggio.
;** OUT: Nessuno.
;*****
VISMES:   push   ACC           ;Istruzioni che visualizzano i
VISMES1:  mov    A,#0          ;caratteri di un messaggio da DPTR
          movc   A,@A+DPTR     ;fino a che non è estratto "$".
    
```

Mnemonic	Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS			
ADD	A,Rn	Add register to Accumulator	1 12
ADD	A,direct	Add direct byte to Accumulator	2 12
ADD	A,@Ri	Add indirect RAM to Accumulator	1 12
ADD	A,#data	Add immediate data to Accumulator	2 12
ADDC	A,Rn	Add register to Accumulator with Carry	1 12
ADDC	A,direct	Add direct byte to Accumulator with Carry	2 12
ADDC	A,@Ri	Add indirect RAM to Accumulator with Carry	1 12
ADDC	A,#data	Add immediate data to Acc with Carry	2 12
SUBB	A,Rn	Subtract Register from Acc with borrow	1 12
SUBB	A,direct	Subtract direct byte from Acc with borrow	2 12
SUBB	A,@Ri	Subtract indirect RAM from ACC with borrow	1 12
SUBB	A,#data	Subtract immediate data from Acc with borrow	2 12
INC	A	Increment Accumulator	1 12
INC	Rn	Increment register	1 12
INC	direct	Increment direct byte	2 12
INC	@Ri	Increment direct RAM	1 12
DEC	A	Decrement Accumulator	1 12
DEC	Rn	Decrement Register	1 12
DEC	direct	Decrement direct byte	2 12
DEC	@Ri	Decrement indirect RAM	1 12

Mnemonic	Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS (continued)			
INC	DPTR	Increment Data Pointer	1 24
MUL	AB	Multiply A & B	1 48
DIV	AB	Divide A by B	1 48
DA	A	Decimal Adjust Accumulator	1 12
LOGICAL OPERATIONS			
ANL	A,Rn	AND Register to Accumulator	1 12
ANL	A,direct	AND direct byte to Accumulator	2 12
ANL	A,@Ri	AND indirect RAM to Accumulator	1 12
ANL	A,#data	AND immediate data to Accumulator	2 12
ANL	direct,A	AND Accumulator to direct byte	2 12
ANL	direct,#data	AND immediate data to direct byte	3 24
ORL	A,Rn	OR register to Accumulator	1 12
ORL	A,direct	OR direct byte to Accumulator	2 12
ORL	A,@Ri	OR indirect RAM to Accumulator	1 12
ORL	A,#data	OR immediate data to Accumulator	2 12
ORL	direct,A	OR Accumulator to direct byte	2 12
ORL	direct,#data	OR immediate data to direct byte	3 24
XRL	A,Rn	Exclusive-OR register to Accumulator	1 12
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2 12
XRL	A,@Ri	Exclusive-OR indirect RAM to Accumulator	1 12
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2 12

figura 1 - Tabella 1: Sommario del set istruzioni dell'AT89.



```
        cjne    A,#36,VISMES2
        sjmp    VISMESF
VISMES2: lcall   TXBYTE
        inc    DPTR
        sjmp    VISMES1
VISMESF: pop    ACC
        ret

;
;
;
MES1:   DB      12
        DB      '** CONTEGGIO DI IMPULSI TRAMITE COUNTER *****'
        DB      13,10,10
        DB      'Conteggio impulsi pervenuti su una linea,13,10
        DB      'di I/O (P3.5/T0) con T0 e visualizzo',13,10
        DB      'il relativo valore (0...255) sul monitor.'
        DB      10,10,13,'$'

;
MES2:   DB      'N. di impulsi = ','$'
;
;
        end
```

Programma n.2

Questo programma realizza un orologio di tipo

software visualizzando secondi, minuti e ore sul terminale video.

```
;Programma OROLOGIO.SRC Uso del TIMER 0 per generare un orologio
; software per gestire ORE:MINUTI:SECONDI.
;
;*****
;
;
ADDR    EQU    1000h                ;Inizio codice in EPROM.
C10MSH  EQU    0DBh                ;Tempo di interrupt T0=10ms.
C10MSL  EQU    0FFh

;
;
        org    ADDR+0000h          ;Vettore di reset.
        ljmp   START              ;Salto all'inizio del codice.
;
        org    ADDR+000Bh          ;Vettore di interrupt per.
        ljmp   TMCOO              ;Salto alla routine di risposta
;
interrupt T0
        org    ADDR+0035h          ;Inizio del codice.
;
;
START:   clr    TR1                ;Timer/counter T1 in off.
        clr    TR0                ;Timer/counter T0 in off.
        mov    SCON,#01010010b
        mov    TMOD,#00100001b
        mov    PCON,#00000000b
        mov    TL1,#0FDh          ;Si caricano i registri di T1 per
        mov    TH1,#0FDh          ;un baud rate di 9600 Baud
        mov    TH0,#C10MSH        ;Si Carica il timer 10ms.
        mov    TL0,#C10MSL
        setb   TR1                ;Abilitazione alla comunicazione
        mov    R4,#0              ;Reset contatore per ottenere
                                   ;il secondo
        mov    R5,#0              ;Reset contatore delle ORE.
        mov    R6,#0              ;Reset contatore dei MINUTI.
```



```

        mov     R7,#0                ;Reset contatore dei SECONDI.
        setb   ET0                   ;Abilitazione interrupt T0
        setb   EA                     ;Abilitazione degli interrupts
        setb   TR0                   ;T0 in on, abilitazione al Timer
;
        mov     DPTR,#MES1
MAIN1:  acall   VISMES                 ;Visualizzazione del messaggio.
        mov     A,#' \                ;Trasmissione di 5 SPACE.
        acall   TXBYTE
        mov     A,#' \
        acall   TXBYTE
        mov     A,#' \
        acall   TXBYTE
        mov     A,#' \
        acall   TXBYTE
        mov     A,R5                  ;Lettura contatore delle ORE.
        acall   V2DIGIT
        mov     A,#': '              ;Trasmissione di un ":".
        acall   TXBYTE
        mov     A,R6                  ;Lettura contatore dei MINUTI.
        acall   V2DIGIT
        mov     A,#': '              ;Trasmissione di un ":".
        acall   TXBYTE
        mov     A,R7                  ;Lettura contatore dei SECONDI.
        acall   V2DIGIT
        mov     A,#13                 ;Trasmissione di un CR.
        acall   TXBYTE
        sjmp   MAIN1
;
;*****
; ** TXBYTE: Routine di trasmissione di un byte sulla seriale.
; ** IN: ACC contenente il byte da trasmettere.
; ** OUT: Nessuno.
;*****
TXBYTE: jnb    TI,TXBYTE              ;Abilitazione a trasmettere.
        clr    TI                     ;Reset del flag TI.
        mov    SBUF,A                 ;Byte nel buffer di trasmissione.
        ret
;
;
;*****
; ** RXBYTE: Routine di ricezione di un byte dalla seriale.
; ** IN: Nessuno.
; ** OUT: ACC contenente il byte ricevuto.
;*****
RXBYTE: jnb    RI,RXBYTE              ;Attesa del byte.
        mov    A,SBUF                 ;Si legge il byte dal buffer
        ;della seriale
        clr    RI                     ;Reset del flag RI.
        ret
;
;
;*****
; ** VISMES: Routine che visualizza un messaggio.
; ** IN: DPTR contenente l' indirizzo del messaggio.
; ** OUT: Nessuno.

```



```
;*****  
VISMES:   push   ACC           ;Istruzioni di visualizzazione  
VISMES1:  mov    A,#0           ;di caratteri puntati da DPTR  
          movc   A,@A+DPTR      ;fino al carattere "$".  
          cjne  A,#36,VISMES2  
          sjmp  VISMESF  
VISMES2:  lcall  TXBYTE  
          inc   DPTR  
          sjmp  VISMES1  
VISMESF:  pop    ACC  
          ret  
  
;  
;  
;*****  
;** V2DIGIT: Vvisualizza sul monitor un valore di 2 digit  
;** IN: ACC contenente il valore  
;** OUT: Nessuno.  
;*****  
V2DIGIT:  mov    R1,A  
          mov    B,#10           ;determinazione delle decine e  
          div   AB              ;e delle unità  
          mov    R2,A  
          add   A,#48           ;Trasmissione delle decine.  
          acall TXBYTE  
          mov    B,R2  
          mov    A,#10  
          mul   AB  
          mov    R2,A  
          mov    A,R1  
          clr   C  
          subb  A,R2  
          add   A,#48           ;Trasmissione delle unita'.  
          acall TXBYTE  
          ret  
  
;  
;*****  
;** TMCO0: Routine di risposta all' interrupt T0.  
;** IN: Nessuno.  
;** OUT: Nessuno.  
;*****  
TMCO0:    push   ACC           ;Salvataggio dell' accumulatore  
          push   PSW           ;nello stack  
          push   PSW           ;Salvataggio della status word  
          push   PSW           ;nello stack  
          clr   TR0           ;Stop al T0.  
          mov   TH0,#C10MSH    ;Si ricarica il timer 10ms.  
          mov   TL0,#C10MSL  
          setb  TR0           ;Start al T0.  
  
;  
          mov   A,R4  
          cjne  A,#100,TMCO01   ;E' trascorso 1 secondo ?  
          mov   R4,#0          ;Reset del contatore per generare  
          mov   R4,#0          ;il secondo  
  
          mov   A,R7  
          cjne  A,#59,TMCO02   ;Reset del contatore dei SECONDI.  
          mov   R7,#0  
          mov   A,R6  
          cjne  A,#59,TMCO03
```



```

mov    R6,#0           ;Reset del contatore dei MINUTI.
mov    A,R5
cjne   A,#23,TMCO04
mov    R5,#0           ;Reset del contatore delle ORE.
sjmp   TMCO0F
TMCO04: inc    R5       ;Incremento delle ORE.
sjmp   TMCO0F
TMCO03: inc    R6       ;Incremento dei MINUTI.
sjmp   TMCO0F
    
```

Mnemonic	Description	Byte	Oscillator Period
LOGICAL OPERATIONS (continued)			
XRL	direct,A	2	12
XRL	direct,#data	3	24
CLR	A	1	12
CPL	A	1	12
RL	A	1	12
RLC	A	1	12
RR	A	1	12
RRC	A	1	12
SWAP	A	1	12
DATA TRANSFER			
MOV	A,Rn	1	12
MOV	A,direct	2	12
MOV	A,@Ri	1	12
MOV	A,#data	2	12
MOV	Rn,A	1	12
MOV	Rn,direct	2	24
MOV	Rn,#data	2	12
MOV	direct,A	2	12
MOV	direct,Rn	2	24
MOV	direct,direct	3	24

Mnemonic	Description	Byte	Oscillator Period
DATA TRANSFER (continued)			
MOV	direct,@Ri	2	24
MOV	direct,#data	3	24
MOV	@Ri,A	1	12
MOV	@Ri,direct	2	24
MOV	@Ri,#data	2	12
MOV	DPTR,#data16	3	24
MOVC	A,@A+DPTR	1	24
MOVC	A,@A+PC	1	24
MOVX	A,@Ri	1	24
MOVX	A,DPTR	1	24
MOVX	@Ri,A	1	24
MOVX	@DPTR,A	1	24
PUSH	direct	2	24
POP	direct	2	24
XCH	A,Rn	1	12
XCH	A,direct	2	12
XCH	A,@Ri	1	12
XCHD	A,@Ri	1	12

figura 2 - Continua il sommario del set istruzioni dell'AT89.



```
TMCO02:  inc    R7                ;Incremento dei SECONDI.
          sjmp   TMCO0F
TMCO01:  inc    R4                ;Incremento base temporale
          ;per il secondo

TMCO0F:  pop    PSW
          pop    ACC
          reti

;
;
MES1:    DB     12
          DB     '** GENERAZIONE DI UN OROLOGIO SOFTWARE**'
          DB     13,10,10
          DB     'Utilizza di T0 per generare un orologio',13,10
          DB     'software per ORE, MINUTI e SECONDI.',13,10,10
          DB     'ORE:MINUTI:SECONDI',10,10,13,'$'

;
;
          end
```

Programma n.3

Questo programma utilizza T0 per generare un PWM sul P3.4. Se si ponticella lo strip in posizione buzzer si sentirà una nota, altrimenti si vedrà l'illu-

minazione del LED T0. Lasciamo al lettore-programmatore il compito di modificare i tempi per avere un lampeggio del LED T0 facilmente riconoscibile.

```
;Programma BUZZER.SRC          Utilizzo di T0 per generare sul pin P3.4
;                               un 1 KHz con un duty_cicle del 75%.
;
;
;
ADDR      EQU      1000h        ;Inizio codice in EPROM.
C075MSH   EQU      0FDh        ;Tempo di interrupt T0=0.75ms.
C075MSL   EQU      04Ch
C025MSH   EQU      0FFh        ;Tempo di interrupt T0=0.25ms.
C025MSL   EQU      018h

;
;
          org      ADDR+0000h    ;Vettore di reset.
          ljmp     START         ;Salto all' inizio del codice.
;
          org      ADDR+000Bh    ;Interrupt per timer/counter T0.
          ljmp     TMCO0         ;Salto alla risposta interrupt T0.
;
          org      ADDR+0035h    ;Inizio del codice.
;
;
START:    clr      TR1          ;Timer/counter T1 in off.
          clr      TR0          ;Timer/counter T0 in off.
          mov      SCON,#01010010b
          mov      TMOD,#00100001b
          mov      PCON,#00000000b
          mov      TL1,#0FDh
          mov      TH1,#0FDh
          mov      TH0,#C075MSH    ;Si Carica il timer 0.75ms
          mov      TL0,#C075MSL
          setb     TR1            ;Abilitazione alla comunicazione
          setb     ET0           ;Abilitazione interrupt T0
          setb     EA            ;Abilitazione generale interrupts
          setb     P3.4          ;Si inizia con il segnale alto
```




```

setb    TR0                ;T0 in on, abilitazione al Timer
;
mov     DPTR,#MES1
acall  VISMES              ;Visualizzazione del messaggio
sjmp   $                  ;Loop infinito.
;
;
;*****
;** TXBYTE: Routine di trasmissione di un byte sulla seriale.
;** IN:    ACC contenente il byte da trasmettere.
;** OUT:   Nessuno.
    
```

Mnemonic	Description	Byte	Oscillator Period
BOOLEAN VARIABLE MANIPULATION			
CLR	C	Clear Carry	1 12
CLR	bit	Clear direct bit	2 12
SETB	C	Set Carry	1 12
SETB	bit	Set direct bit	2 12
CPL	C	Complement Carry	1 12
CPL	bit	Complement direct bit	2 12
ANL	C,bit	AND direct bit to CARRY	2 24
ANL	C,/bit	AND complement of direct bit to Carry	2 24
ORL	C,bit	OR direct bit to Carry	2 24
ORL	C,/bit	OR complement of direct bit to Carry	2 24
MOV	C,bit	Move direct bit to Carry	2 12
MOV	bit,C	Move Carry to direct bit	2 24
JC	rel	Jump if Carry is set	2 24
JNC	rel	Jump if Carry not set	2 24
JB	bit,rel	Jump if direct Bit is set	3 24
JNB	bit,rel	Jump if direct Bit is Not set	3 24
JBC	bit,rel	Jump if direct Bit is set & clear bit	3 24
PROGRAM BRANCHING			
ACALL	addr11	Absolute Subroutine Call	2 24
LCALL	addr16	Long Subroutine Call	3 24
RET		Return from Subroutine	1 24
RETI		Return from interrupt	1 24
AJMP	addr11	Absolute Jump	2 24
LJMP	addr16	Long Jump	3 24
SJMP	rel	Short Jump (relative addr)	2 24

Mnemonic	Description	Byte	Oscillator Period
PROGRAM BRANCHING (continued)			
JMP	@A+DPTR	Jump indirect relative to the DPTR	1 24
JZ	rel	Jump if Accumulator is Zero	2 24
JNZ	rel	Jump if Accumulator is Not Zero	2 24
CJNE	A,direct,rel	Compare direct byte to Acc and Jump if Not Equal	3 24
CJNE	A,#data,rel	Compare immediate to Acc and Jump if Not Equal	3 24
CJNE	Rn,#data,rel	Compare immediate to register and Jump if Not Equal	3 24
CJNE	@Ri,#data,rel	Compare immediate to indirect and Jump if Not Equal	3 24
DJNZ	Rn,rel	Decrement register and Jump if Not Zero	2 24
DJNZ	direct,rel	Decrement direct byte and Jump if Not Zero	3 24
NOP		No Operation	1 12

figura 3 - Continua il sommario del set istruzioni dell'AT89.



```
;*****  
TXBYTE:   jnb    TI,TXBYTE           ;Attesa per trasmettere.  
          clr    TI                   ;Reset del flag TI.  
          mov    SBUF,A               ;Byte nel buffer di trasmissione.  
          ret  
  
;  
;  
;*****  
;** RXBYTE: Routine di ricezione di un byte dalla seriale.  
;** IN: Nessuno.  
;** OUT: ACC contenente il byte ricevuto.  
;*****  
RXBYTE:   jnb    RI,RXBYTE           ;Attesa del byte.  
          mov    A,SBUF               ;Si legge il byte dal buffer  
                                     ;della seriale  
          clr    RI                   ;Reset del flag RI.  
          ret  
  
;  
;  
;*****  
;** VISMES: Routine che visualizza un messaggio.  
;** IN: DPTR contenente l' indirizzo del messaggio.  
;** OUT: Nessuno.  
;*****  
VISMES:   push   ACC                  ;Queste istruzioni visualizzano  
VISMES1:  mov    A,#0                  ;i caratteri puntati da DPTR  
          movc   A,@A+DPTR            ;fino a che al carattere "$".  
          cjne  A,#36,VISMES2  
          sjmp  VISMESF  
VISMES2:  lcall  TXBYTE  
          inc   DPTR  
          sjmp  VISMES1  
VISMESF:  pop    ACC  
          ret  
  
;  
;  
;*****  
;** TMC00: Routine di risposta all' interrupt T0.  
;** IN: Nessuno.  
;** OUT: Nessuno.  
;*****  
TMC00:   push   PSW                  ;Salvataggio della status word  
          ;                               nello stack  
          clr    TR0                  ;Stop al T0.  
          jb    P3.4,TMC001  
          mov   TH0,#C075MSH          ;Si ricarica il timer 0.75ms.  
          mov   TL0,#C075MSL  
          sjmp  TMC002  
TMC001:  mov   TH0,#C025MSH          ;Si ricarica il timer 0.25ms.  
          mov   TL0,#C025MSL  
          sjmp  TMC002  
TMC002:  setb   TR0                  ;Start al T0.  
  
          ;                               ;Settaggio della linea.  
          cpl   P3.4  
  
          pop   PSW  
          reti  
  
;
```



```
;  
MES1:    DB    12  
         DB    '***** DEMO PER LA GENERAZIONE DI UN PWM *****'  
         DB    13,10,10  
         DB    'Utilizza di T0 per generare sul pin P3.4',13,10  
         DB    ' un 1 KHz con duty_cicle del 75%.',13,10,10,'$'  
;  
;  
         end
```

Reperibilità e costi

KIT completo di microcontrollore	£ 75.000
Programmatore-Emulatore SIM2051	£ 400.000
Software ASM-51	£ 240.000
Chip 89C2051	£ 15.000
CD ROM manuale del 2051	£ 145.000

spese di spedizione.

Indirizzare richieste e informazioni a:

**Nello Alessandrini - via Timavo, 10
40131 Bologna - tel. e fax 051/649.10.80**

Nelle richieste sia telefoniche che fax ricordarsi di lasciare anche un recapito telefonico. _____

