

PLAY - MACHINE CODE TONE

```

DEPENDENCIES : PICDUR @ {Internal to play}
ROUTINES USED : BCDBIN @ 02E3
INPUT : PLAY = X reg is tone (8bit) decimal value is built up
           from bits 7:3 TONE (0-31)
           bits 2:0 DURATION (0-7)
: PLAYM = A[S&X] bits 7:3 TONE (0-31)
           2:0 DURATION (0-7)
OUTPUT : TONE AS INDICATED - IGNORES AUDIO ENABLE!

```

```

Y:
A
L
P
x OFF7 PLAY:: C=REG 3/X ; FOCAL ENTRY POINT
8 * ; GET TONE AND CONVERT
9 NC XQ BCDBIN ; TO BINARY IN REGISTER C[S&X]
A A=C ALL
B PLAYM: ST=0 ; MCODE ENTRY POINT
          ST<>T ; GET TONE REGISTER READY
          A=0 XS ; VALID TONES 0-255 INC.
          C=0 ALL
          A<>C S&X
          A=C S&X
          C=C+C S&X
          RCR 12 ; OFFSET INTO TABLE
          A=C M ; IN A[M]
          LDI
          CON 24
          C=A+C S&X
          RCR 2
          C=-C-1 MS
          A=C MS ; A[MS]= CYCLES PER NOTE
          *
          *
          GOSUB PICDUR ; JUMP OVER TABLE, SAVING ITS ADDR
          HEX 0FA ; SPACE ON THE STACK
          HEX 01E ; TONE 1 (33 CYCLES)
          HEX 020 ; 2 (32 " )
          HEX 020 ; 3 (21 " )
          HEX 022 ; 4 (30 " )
          HEX 022 ; 5 (29 " )
          HEX 024 ; 6 (28 " )
          HEX 026 ; 7 (27 " )
          HEX 026 ; 8 (26 " )
          HEX 028 ; 9 (25 " )
          HEX 02A ; 10 (24 " )
          HEX 02C ; 11 (23 " )
          HEX 02E ; 12 (22 " )
          HEX 030 ; 13 (21 " )
          HEX 032 ; 14 (20 " )
          HEX 034 ; 15 (19 " )

```

; NR. ENTRIES IN TABLE MUST BE EVEN AND ALL (DURATIONS * CYCLES) NEED TO BE ROUGHLY THE SAME.

HEX	038	:	16	(18 CYCLES)
HEX	03A	:	17	(17 ")
HEX	03E	:	18	(16 ")
HEX	042	:	19	(15 ")
HEX	048	:	20	(14 ")
HEX	04C	:	21	(13 ")
HEX	054	:	22	(12 ")
HEX	058	:	23	(11 ")
HEX	064	:	24	(10 ")
HEX	070	:	25	(9 ")
HEX	07E	:	26	(8 ")
HEX	08E	:	27	(7 ")
HEX	0A6	:	28	(6 ")
HEX	0C8	:	29	(5 ")
HEX	0FA	:	30	(4 ")
HEX	14E	:	31	(3 ")

x12D

PICDUR:

```

POP
C=C+A M
FETCH
C=0 M
RCR 11
A<>C S&X
A=C S&X
ST=C
CLRF 3
C<>ST
PT=0
A=A-C PT
A=0 M
MULTD: A=A+C M
C=C-1 PT
JNC-02 MULTD
FINMUL: A<>C MS
A=A+1 M
LDI
HEX 0FF
?FSET 3
JL+14 ODD
EVEN: ST=C
?A#0 S&X
JNC+0B GAP
C=0 @R
?A#C S&X
JNC+09 TONE30

```

```

; GET DURATION TABLE
; ADDRESS AND ADD OFFSET FOR TONE.
; GET DURATION WORD FROM TABLE.
; PUT DURATION INTO C[M].
; GET DURATION MULTIPLIER INTO C[S&X].
; ST = xxxx xDDD
; ST = xxxx 0DDD
; DURATION MULTIPLIER IN C[0]
; C[S&X] = xxxx xxxx 0DDD
; A[S&X] = 0000 TTTT T000
; FOR DDD DURATIONS ADD UP
; DURATION VALUE FROM TABLE TO
; GET TOTAL TONE DURATION COUNTER.
; PUT MASTER CYCLE COUNT IN C[MS].
; MAKE DURATION ODD SO TONE
; REGISTER ENDS UP CLEAR.
; TONE 'ON' VALUE
; CHECK FOR ODD TONE
; (ST = TTTT IDDD)
; PUT TONE 'ON' INTO ST.
; IF TONE IS 0 THEN PRODUCE
; A GAP INSTEAD.
; NOW C[S&X] = 0FF = TONE 30
; IF TONE = 30 THEN DO SPECIAL
; TONE 30 TIGHT LOOP

```

```

DELPE:  A=C MS           ; ELSE DO STANDARD EVEN CYCLE
        ST<>T           ; TONES( 6 CYCLES UPWARDS)
        A=A-1 MS
        JNC-01
        A=A-1 M
        JNC-05 DELPE
        RTN

```

```

GAP:    ST=0           ; GAP => MAKE NO SOUND !
TONE30: ST<>T           ; TONE 30 (& GAP) SPECIAL
        NOP            ; TIGHT LOOP THAT IS
        A=A-1 M        ; 4 CYCLES PER TONE
        JNC-03         ; TRANSITION.
        RTN

```

```

ODD:    ST=C           ; SEE IF ODD TONE IS SPECIAL
        LDI
        HEX 0E8
        ?A#C S&X
        JNC+0D TONE29
        LDI
        HEX 0F8
        ?A#C S&X
        JNC+0F TONE31

```

```

DELPO:  A=C MS           ; STANDARD ODD LOOP
        ST<>T           ; ( 7 CYCLES UPWARDS)
        A=A-1 MS
        JNC-01
        NOP
        A=A-1 M
        JNC-06 DELPO
        RTN

```

```

TONE29: ST<>T           ; SPECIAL TONE29 TIGHT LOOP
        NOP            ; ( 5 CYCLES PER TONE
        NOP            ; REGISTER TRANSITION )
        A=A-1 M
        JNC-04 TONE29
        RTN

```

```

x160   TONE31: ST<>T           ; SPECIAL TONE31 TIGHT LOOP
        A=A-1 M        ; ( 3 CYCLES PER TONE
        JNC-02 TONE31 ; REGISTER TRANSITION )
        RTN
x170

```

MCPLAY - MACHINE CODE TUNES

```

DEPENDENCIES  : PLAYM @
ROUTINES USED : NONE
INPUT        : MCODE OF FORM :
                                     GOSUB MCPLAY
                                     HEX   ~TONE~
                                     HEX   ~TONE~
                                     {
                                     }
                                     HEX   3FF

OUTPUT       : PLAYS TONES FOLLOWING CALL, ENDS
               WHEN FINDS 3FF - IGNORES AUDIO ENABLE!
    
```

```

x040 MCPLAY : POP
              FETCH
              C=C+1 M
              PUSH
              A=C SBX
              ?A#0 XS
              C RTN
              GOSUB PLAYM
              JNC- MCPLAY
    
```

BIRTHDAY - HAPPY BIRTHDAY TUNE

```

Y:
A
D
H
T
R
B
x0A2 BIRTHDAY :: *
              *
              GOSUB MCPLAY
                   072
                   070
                   083
                   073
                   09B
                   097
                   072
                   070
                   083
                   073
                   0AB
                   09F
                   072
                   070
                   0C3
                   0B3
                   ↘
                   09B
                   093
                   083
                   001
                   0BA
                   0B8
                   0B3
                   09B
                   0AB
                   09F
                   3FF
                   3E0
                   END-TONE
                   RTN
                   ↙
    
```