Application 2: Waveform Generator

This application allows the user to output 4 different waveforms (sine, square, triangle and sawtooth) from the digital to analog convertor. The desired waveform can be selected by moving DIP switches 6 and 7 to one of 4 possible combinations. The frequency of the waveforms can be changed by moving DIP switches 0 through 5.

```
timerhi: equ     15h      ; the timer mode and MSB of count length
timerlo: equ     14h      ; the LSB of count length
dip:     equ     12h      ; DIP switch port
dacout:  equ     13h      ; Digital to analog output port
cmdreg:  equ     10h      ; 8155 control register.

         org     0ff01h
getime:  in      dip              ;get value of DIP switches
         add     a                ;shift left padding zeros
         add     a                ;shift left padding zeros
         out     timerlo          ;set the low count
         mvi     a,11000000b
         out     timerhi          ;single pulse w/auto reload
         mvi     a,0cdh
         out     cmdreg           ;enable timer

         in      dip              ;read DIP again
         ani     11000000b        ;Mask all DIP bits except 6 and 7
         cpi     0
         jz      sinewv           ;if upper bits are 0, output sine wave
         cpi     01000000b
         jz      sqrwav           ;if upper 2 bits are 01, output square wave
         cpi     10000000b
         jz      triang           ;if upper 2 bits are 10, output triangle wave

         ; If none of the above, upper 2 bits are 11, so output a .......
         ; sawtooth wave

sawwav:  mvi     c,0              ; invert the pattern
         mvi     d,3fh            ; starting value to output
         jmp     trian2

         ; triangle wave
triang:  mvi     c,1
         mvi     d,0              ; upward slope 0 to 3e
trian1:  mov     a,d
         call    dactim           ; output the pattern to DAC and wait
         inr     d
         mvi     a,3fh            ; if D = 3F then slope down
         cmp     d
         jnz     trian1

trian2:  mov     a,d              ; downward slope 3f to 1
         call    dactim           ; output the pattern to DAC and wait
         dcr     d
         jnz     trian2
         jmp     getime           ; check DIP switch

         ; square wave
sqrwav:  mvi     c,1              ; non-inverted output
sqrwv2:  mvi     d,32             ; output 32 times for each half of period
sqrwv3:  xra     a
         call    dactim           ; output the pattern to DAC and wait
         dcr     d
         jnz     sqrwv3           ; jump if not output 32 times already
         dcr     c                ; change to inverted output mode
         jz      sqrwv2           ; if c=0 then sqrwv2
```

```
         jmp      getime            ; c=FF so check DIP switch

         ; sine wave
sinewv:  lxi      h,sintbl          ; point to sine table
quadst:  mvi      c,1               ; C=1 = 1st 2 quadrants, C=0 2nd two
quadrants
quad1:   inx      h                 ; skip the 0
qud1lp:  inx      h
         mov      a,m               ; A is value from table
         ora      a                 ; set Z flag if A = 0
         jz       quad2             ; if A = 0 then read the table backwards
         call     dactim            ; output the pattern to DAC and wait
         jmp      qud1lp

quad2:   dcx      h                 ; skip the 0
qud2lp:  dcx      h
         mov      a,m               ; A is value from table
         ora      a                 ; set Z flag if A = 0
         jz       quad3             ; if A=0 then invert the output pattern
         call     dactim            ; output the pattern to DAC and wait
         jmp      qud2lp

quad3:   dcr      c                 ; change invert flag
         jz       quad1             ; if C=0 start over but invert data
         jmp      getime            ; if C=FF then check DIP switch

         ; DACTIM:  This subroutine examines the C register and if C=0
         ; it will invert the data in the A register otherwise if C=1 it
         ; will not.  The A register is then output to the D to A convertor.
         ; After this, the RST 7.5 interrupt flag will be polled until a pulse
         ; is sent from the 8155 timer. This causes the program to pause after
         ; each output from the D to A convertor according the the length
         ; of the timer count.
dactim:  inr      c                 ; see what C is .... (0 or 1)
         dcr      c                 ; ...without changing it
         jnz      dactim1           ; jump if C = 1 and don't invert data
         mov      b,a               ; invert the data
         mvi      a,3fh             ; by subtracting it from this value
         sub      b

dactim1: out      dacout            ; output the data
polltmr: rim                        ; loop until rst 7.5 flag is high
         ani      01000000b         ; mask all but rst 7.5 flag
         jz       polltmr           ; check it again if not set
         mvi      a,10h
         sim                        ; clear the interrupt flag
         ret

         ; This is 1 quadrant of the sine wave pattern with zeros marking
         ; the start and the end.
sintbl:  defb     0, 1Fh,21h,23h,25h, 27h,29h,2Bh,2Dh, 2Eh,30h,32h,34h, 35h
         defb     36h,38h,39h,3Ah, 3Bh,3Ch,3Dh,3Dh, 3Eh,3Eh,3Fh,3Fh, 3Fh, 0

         end
```

ADDRESS	DATA	DESCRIPTION
FF01	DB	IN 12
FF02	12	
FF03	87	ADD A
FF04	87	ADD A
FF05	D3	OUT 14
FF06	14	
FF07	3E	MVI A,C0
FF08	C0	
FF09	D3	OUT 15
FF0A	15	
FF0B	3E	MVI A,CD
FF0C	CD	
FF0D	D3	OUT 10
FF0E	10	
FF0F	DB	IN 12
FF10	12	
FF11	E6	ANI C0
FF12	C0	
FF13	FE	CPI 00
FF14	00	
FF15	CA	JZ FF56
FF16	56	
FF17	FF	
FF18	FE	CPI 40
FF19	40	
FF1A	CA	JZ FF43
FF1B	43	
FF1C	FF	
FF1D	FE	CPI 80
FF1E	80	
FF1F	CA	JZ FF29
FF20	29	
FF21	FF	
FF22	0E	MVI C,00
FF23	00	
FF24	16	MVI D,3F
FF25	3F	
FF26	C3	JMP FF38
FF27	38	
FF28	FF	
FF29	0E	MVI C,01
FF2A	01	
FF2B	16	MVI D,00
FF2C	00	
FF2D	7A	MOV A,D
FF2E	CD	CALL FF7C
FF2F	7C	
FF30	FF	
FF31	14	INR D
FF32	3E	MVI A,3F
FF33	3F	
FF34	BA	CMP D
FF35	C2	JNZ FF2D
FF36	2D	
FF37	FF	
FF38	7A	MOV A,D
FF39	CD	CALL FF7C
FF3A	7C	
FF3B	FF	
FF3C	15	DCR D
FF3D	C2	JNZ FF38
FF3E	38	
FF3F	FF	
FF40	C3	JMP FF01
FF41	01	
FF42	FF	
FF43	0E	MVI C,01
FF44	01	
FF45	16	MVI D,20
FF46	20	
FF47	AF	XRA A
FF48	CD	CALL FF7C
FF49	7C	
FF4A	FF	
FF4B	15	DCR D
FF4C	C2	JNZ FF47
FF4D	47	
FF4E	FF	
FF4F	0D	DCR C
FF50	CA	JZ FF45
FF51	45	
FF52	FF	
FF53	C3	JMP FF01
FF54	01	
FF55	FF	
FF56	21	LXI H,FF91
FF57	91	
FF58	FF	
FF59	0E	MVI C,01
FF5A	01	
FF5B	23	INX H
FF5C	23	INX H
FF5D	7E	MOV A,M
FF5E	B7	ORA A
FF5F	CA	JZ FF68
FF60	68	
FF61	FF	
FF62	CD	CALL FF7C
FF63	7C	
FF64	FF	
FF65	C3	JMP FF5C
FF66	5C	
FF67	FF	
FF68	2B	DCX H
FF69	2B	DCX H
FF6A	7E	MOV A,M
FF6B	B7	ORA A
FF6C	CA	JZ FF75
FF6D	75	
FF6E	FF	
FF6F	CD	CALL FF7C
FF70	7C	
FF71	FF	
FF72	C3	JMP FF69
FF73	69	
FF74	FF	
FF75	0D	DCR C
FF76	CA	JZ FF5B

continued on next page...

ADDRESS	DATA	DESCRIPTION
FF77	5B	
FF78	FF	
FF79	C3	JMP FF01
FF7A	01	
FF7B	FF	
FF7C	0C	INR C
FF7D	0D	DCR C
FF7E	C2	JNZ FF85
FF7F	85	
FF80	FF	
FF81	47	MOV B,A
FF82	3E	MVI A,3F
FF83	3F	
FF84	90	SUB B
FF85	D3	OUT 13
FF86	13	
FF87	20	RIM
FF88	E6	ANI 40
FF89	40	
FF8A	CA	JZ FF87
FF8B	87	
FF8C	FF	
FF8D	3E	MVI A,10
FF8E	10	
FF8F	30	SIM
FF90	C9	RET
FF91	00	From here down is sine wave data
FF92	1F	
FF93	21	
FF94	23	
FF95	25	
FF96	27	
FF97	29	
FF98	2B	
FF99	2D	
FF9A	2E	
FF9B	30	
FF9C	32	
FF9D	34	
FF9E	35	
FF9F	36	
FFA0	38	
FFA1	39	
FFA2	3A	
FFA3	3B	
FFA4	3C	
FFA5	3D	
FFA6	3D	
FFA7	3E	
FFA8	3E	
FFA9	3F	
FFAA	3F	
FFAB	3F	
FFAC	00	