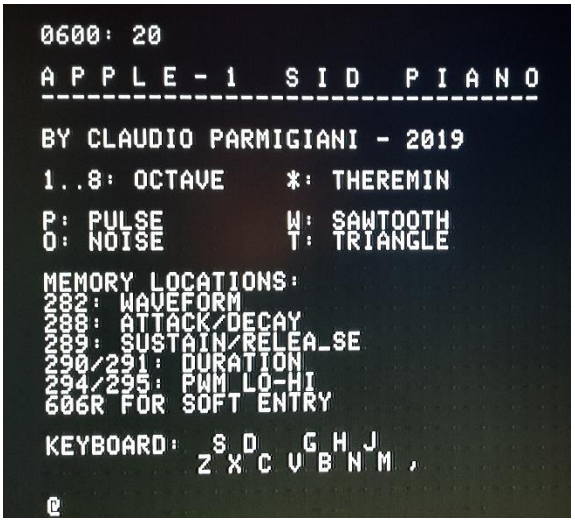
APPLE-1 SID INTERFACE ADDENDUM

by Claudio Parmigiani

ADDENDUM: APPLE-1 SID KEYBOARD PROGRAM

A simple program has been prepared, in order to use/play at least the basic functionality of the SID.



The musical notes will be played (one voice only and one octave and at the time) after choosing one of the 8 octaves available and the desired waveform (Pulse / Noise / Triangle / Sawtooth).

Attack / Decay / Sustain / Release (ADSR) parameters, note length, waveform, PWM parameters can also be modified changing (after a RESET) the content of the memory location shown in the picture.

In this case, in order to preserve the new values, a *soft entry* to the program must be performed.

An experimental *Theremin* can be played connecting two potentiometers, accordingly with the 6581's datasheet. In this case the output waveform is inherited from the previous settings and cannot be changed. *Theremin* sound will span all over the eight octaves [NOT TESTED YET].

Press RESET to exit from program and stop the sound.

SOURCE CODE (Player, starts at $0600)

```
define VOICE1   $C800       ;base address voice 1
define FREQLO1 $C800        ;frequency - LSB
define FREQHI1 $C801        ;frequency - MSB
define CR1      $C804       ;Waveform and Gate
define AD1      $C805       ;Attack/Decay
define SR1      $C806       ;Sustain/Release
define PWM1     $C802       ;PWM LO byte
define PWM2     $C803       ;PWM HO (4 bit)

JSR DEFAULT
JSR READ_1
JSR INIT
JSR LOOP

PARSER:
       JSR $FFEF     ;print the char
       CMP #$AA      ;* key, Theremin
       BEQ THEREMIN
       CMP #$DA      ;Z key, C note (do)
       BEQ CNOTE
       CMP #$D8      ;X key, D note (re)
       BEQ DNOTE
       CMP #$C3      ;C key, E note (mi)
       BEQ ENOTE
       CMP #$D6      ;V key, F note (fa)
       BEQ FNOTE
       CMP #$C2      ;B key, G note (sol)
       BEQ GNOTE
       CMP #$CE      ;N key, A note (la)
       BEQ ANOTE
       CMP #$CD      ;M key, B note (si)
       BEQ BNOTE
       CMP #$AC      ;, key, C note of next octave
       BEQ CNEXT


       CMP #$D3      ;S key, C# note (do#)
       BEQ CSNOTE
       CMP #$C4      ;D key, D# note (re#)
       BEQ DSNOTE
       CMP #$C7      ;G key, F# note (fa#)
       BEQ FSNOTE
       CMP #$C8      ;H key, G# note (sol#)
       BEQ GSNOTE
       CMP #$CA      ;J key, A# note (la#)
       BEQ ASNOTE

       CMP #$CF      ;O key, noise waveform
       BEQ NOI
       CMP #$D0      ;P key, pulse waveform
       BEQ PUL
       CMP #$D4      ;T key, triangle waveform
       BEQ TRI
       CMP #$D7      ;W key, sawtooth waveform
       BEQ SAW
```

```
       CMP #$B9     ;numeral
       BMI NUMERAL
       RTS

NUMERAL:
       SBC #$AF
       STA $0283
       JSR SET_OCTAVE
       RTS

THEREMIN:
       JSR THEREMIN_2
       RTS
NOI:   JSR NOI_2
       RTS
PUL:   JSR PUL_2
       RTS
TRI:   JSR TRI_2
       RTS
SAW:   JSR SAW_2
       RTS
CNOTE: JSR CNOTE_2
       RTS
DNOTE: JSR DNOTE_2
       RTS
ENOTE: JSR ENOTE_2
       RTS
FNOTE: JSR FNOTE_2
       RTS
GNOTE: JSR GNOTE_2
       RTS
ANOTE: JSR ANOTE_2
       RTS
BNOTE: LDA #$01     ;set exception bit
       STA $0292
       JSR BNOTE_2
       RTS
CSNOTE:      JSR CSNOTE_2
       RTS
DSNOTE:      JSR DSNOTE_2
       RTS
FSNOTE:      JSR FSNOTE_2
       RTS
GSNOTE:      JSR GSNOTE_2
       RTS
ASNOTE:      JSR ASNOTE_2
       RTS
CNEXT: JSR CNEXT_2
       RTS


NOI_2:
       LDA #$81
       STA $0282
       RTS

PUL_2:
       LDA #$41
```

```
       STA $0282
       RTS

TRI_2:
       LDA #$11
       STA $0282
       RTS

SAW_2:
       LDA #$21
       STA $0282
       RTS

CNOTE_2:                ;Z key, C note (do)
       LDA #$89
       STA $0284
       LDA #$2B
       STA $0285
       RTS

CNEXT_2:                ;, key, C note (do) next octave
       LDA #$01
       STA $0293        ;set next octave bit
       LDA #$89
       STA $0284
       LDA #$2B
       STA $0285
       RTS

DNOTE_2:                ;X key, D note (re)
       LDA #$99
       STA $0284
       LDA #$F7
       STA $0285
       RTS

ENOTE_2:                ;C key, E note (mi)
       LDA #$AC
       STA $0284
       LDA #$D2
       STA $0285
       RTS

FNOTE_2:                ;V key, F note (fa)
       LDA #$B7
       STA $0284
       LDA #$19
       STA $0285
       RTS

GNOTE_2:                ;B key, G note (sol)
       LDA #$CD
       STA $0284
       LDA #$85
       STA $0285
       RTS

ANOTE_2:                ;N key, A note (la)
```

```
       LDA #$E6
       STA $0284
       LDA #$B0
       STA $0285
       RTS

BNOTE_2:                  ;M key, B note (si)
       LDA #$02
       STA $0284
       LDA #$F0
       STA $0285
       RTS

CSNOTE_2:                 ;S key, C# note (do#)
       LDA #$91
       STA $0284
       LDA #$53
       STA $0285
       RTS
DSNOTE_2:                 ;D key, D# note (re#)
       LDA #$A3
       STA $0284
       LDA #$1F
       STA $0285
       RTS
FSNOTE_2:                 ;G key, F# note (fa#)
       LDA #$C1
       STA $0284
       LDA #$FC
       STA $0285
       RTS
GSNOTE_2:                 ;H key, G# note (sol#)
       LDA #$D9
       STA $0284
       LDA #$BD
       STA $0285
       RTS
ASNOTE_2:                 ;J key, A# note (la#)
       LDA #$F4
       STA $0284
       LDA #$67
       STA $0285
       RTS

KBDIN:              ;read key from keyboard
       LDA $d011
       BPL KBDIN
       LDA $d010
       STA $0286
       RTS

DELAY:         PHA
               LDA  $0290  ;outer loop
               STA  $0280
L1:           LDA  $0291   ;inner loop
               STA  $0281
L2:           DEC  $0281
               BNE  L2
```

```
              DEC  $0280
              BNE  L1
              PLA
              RTS

KILLER:
       LDA #$00
       STA CR1      ;kills the note
       RTS

SET_OCTAVE:
       LDY $0283
       JSR CHECK_NEXT_OCT
       CPY #$01
       BNE DIV_LOOP
       RTS

DIV_LOOP:
       JSR DIV_BY_2
       CPY #$1
       BNE DIV_LOOP
       RTS

DIV_BY_2:
       LSR $0292    ;exception
       ROR $0284       ;Shift the MSB
       ROR $0285       ;Rotate the LSB
       DEY
       RTS

CHECK_NEXT_OCT:
       LDA $0293
       CMP #$00
       BEQ DO_NO_CHANGE_OCT
       DEY
       RTS

DO_NO_CHANGE_OCT:
       RTS

DEFAULT:
       LDA #$50
       STA $0290    ;delay, outer loop
       LDA #$AE
       STA $0291    ;delay, inner loop
       LDA #$08
       STA $0288    ;Attack/Decay
       LDA #$44
       STA $0289    ;Sustain/Release
       LDA #$FF
       STA $0294    ;PWM1
       LDA #$08
       STA $0295    ;PWM2
       RTS

INIT:
       LDA $0294
       STA PWM1     ;store PWM1 on SID
```

```
       LDA $0295
       STA PWM2      ;store PWM2 on SID
       LDA $0288
       STA AD1             ;Attack/Decay
       LDA $0289
       STA SR1             ;Sustain/Relase
       LDA #$0F
       STA $C818     ;max volume
       RTS

LOOP:
       JSR KBDIN
       JSR PARSER
       JSR SET_OCTAVE
       LDA $0284
       STA FREQHI1   ;store MSB freq. on SID
       LDA $0285
       STA FREQLO1   ;store LSB freq. on SID
       LDA $0282
       STA CR1             ;play the note
       JSR DELAY
       JSR KILLER
       LDA #$00      ;clear the exception bit
       STA $0292
       LDA #$00      ;clear the next octave bit
       STA $0293
       JMP LOOP

THEREMIN_2:
       LDA $0282
       STA CR1             ;play the note (non stop)
       JSR THERLOOP

THERLOOP:
       LDA $C819     ;read paddle X
       STA FREQHI1   ;store MSB freq. in SID
       LDA $C81A     ;read paddle Y
       STA FREQLO1   ;store LSB freq. in SID
       JSR THERLOOP

READ_1:
       LDA $0850,X
       JSR $FFEF
       INX
       CPX #$ff      ;read first 255 chars
       BNE READ_1
READ_2:
       LDA $094F,X
       JSR $FFEF
       INX
       CPX #$6f      ;read remaining 111
       BNE READ_2
       RTS
```

Here follows the Welcome Screen hexdump as shown in the picture above:

```
850:8D 8D C1 A0 D0 A0 D0 A0 CC A0 C5 A0 AD A0 B1 A0
```

```
860:A0 A0 D3 A0 C9 A0 C4 A0 A0 A0 D0 A0 C9 A0 C1 A0
870:CE A0 CF 8D AD AD AD AD AD AD AD AD AD AD AD AD
880:AD AD AD AD AD AD AD AD AD AD AD AD AD AD AD AD
890:AD AD AD AD AD 8D 8D C2 D9 A0 C3 CC C1 D5 C4 C9
8a0:CF A0 D0 C1 D2 CD C9 C7 C9 C1 CE C9 A0 AD A0 B2
8b0:B0 B1 B9 8D 8D B1 AE AE B8 BA A0 CF C3 D4 C1 D6
8c0:C5 A0 A0 A0 A0 AA BA A0 D4 C8 C5 D2 C5 CD C9 CE
8d0:8D 8D D0 BA A0 D0 D5 CC D3 C5 A0 A0 A0 A0 A0 A0
8e0:A0 A0 D7 BA A0 D3 C1 D7 D4 CF CF D4 C8 8D CF BA
8f0:A0 CE CF C9 D3 C5 A0 A0 A0 A0 A0 A0 A0 A0 D4 BA
900:A0 D4 D2 C9 C1 CE C7 CC C5 8D 8D CD C5 CD CF D2
910:D9 A0 CC CF C3 C1 D4 C9 CF CE D3 BA 8D B2 B8 B2
920:BA A0 D7 C1 D6 C5 C6 CF D2 CD 8D B2 B8 B8 BA A0
930:C1 D4 D4 C1 C3 CB AF C4 C5 C3 C1 D9 8D B2 B8 B9
940:BA A0 D3 D5 D3 D4 C1 C9 CE AF D2 C5 CC C5 C1 D3
950:C5 8D B2 B9 B0 AF B2 B9 B1 BA A0 C4 D5 D2 C1 D4
960:C9 CF CE 8D B2 B9 B4 AF B2 B9 B5 BA A0 D0 D7 CD
970:A0 CC CF AD C8 C9 8D B6 B0 B6 D2 A0 C6 CF D2 A0
980:D3 CF C6 D4 A0 C5 CE D4 D2 D9 8D 8D CB C5 D9 C2
990:CF C1 D2 C4 BA A0 A0 D3 A0 C4 A0 A0 A0 C7 A0 C8
9a0:A0 CA 8D A0 A0 A0 A0 A0 A0 A0 A0 A0 A0 DA A0 D8
9b0:A0 C3 A0 D6 A0 C2 A0 CE A0 CD A0 AC 8D 8D
```