



```

        STAA    CYCLES           ;Initially CYCLES is zero
        LDX    $3300           ;Frequencies are stored at this memory location
        STX    LED             ;Display on LED
        LDX    #$3300         ;Need to store address of next frequency
at $3400
        INX
        INX
        STX    $3400
        LDAA   #$80           ;Write #1000 0000 to TMSK1 to enable
TOC1
        STAA   TMSK1          ;Store $80 in TMSK1 at address 1022
        CLI
        BEGIN  LDX    #N       ;Enable interrupts to occur
        LDY    #LKUP         ;Load the number of samples
Lookup table
        LDY    #LKUP         ;Y Points to the first SAMPLE in the
LOOP1   LDAA   0,Y           ;LOAD A with CURRENT sample from Lookup
table
        STAA   PORT1         ;OUTPUT DATA to PORT1
        INY
        INY                 ;INC index Y to be ready for NEXT
SAMPLE
        JSR    T1            ;Call DELAY subroutine
        DEX
        BEQ    BEGIN        ;DEC X
cycle starts
        JMP    LOOP1         ;IF X=0, ONE Cycle finishes and next
                                ;Otherwise, OUTPUT next SAMPLE

```

```

*-----
-----

```

\* DELAY SUBROUTINE:

```

T1      PSHY                 ;Record Y in a Stack so the register
can be used in this subroutine
        LDY    #$4100       ;Load Y with the starting address of
the tone lookup table
        LDAB   FREQ         ;Load ACCB with tone number
        ABY
of the tone lookup table
        LDAA   $0,Y         ;Add it to Y to get the current address
L1      DECA                 ;Get current tone repeat value
correct time delay
        BNE    L1           ;start time delay loop to get the
        PULY                 ;Recall Y from Stack before returning
to main program
        RTS                  ;Return from Subroutine

```

```

*-----
-----

```

\* Interrupt Service Routine

```

ISR    LDY    $3200         ;Initially #$3100 is stored in $3200. This is
updated and relevant memory accessed to get duration
        LDD    #$FFFF       ;It takes 0.016384 seconds to access
all memory locations
        ADDD   TOC1         ;add $FFFF to TOC1, so after 0.016384
sec TCNT=TOC1 causing interrupt

```

```

        STD      TOC1          ;Stored in TOC1 so as to check when
next whole cycle finishes
        LDAA     CYCLES       ;ACCA has the number of times whole of
memory (65535) has been accessed
        INCA     ;then increase it
        STAA     CYCLES       ;Updated value stored
        CMPA     0,Y         ;Compare if number in ACCA is equal to
duration given by (0,Y).Initially it will be (0,$3100)
        BNE     DONE         ;Checks if duration is reached or not
        INY     ;If equal next note and duration have to be
accessed
        STY     $3200        ;Value in Y stored so that next duration can be
accessed with the next note
        LDAA     FREQ        ;Load ACCA with the number of notes that
have been played (25 altogether)
        INCA

        PSHA     ;This section creates a slight delay of around
0.4s so that theres a slight pause between 2 notes
        LDAA     #25         ;Two loops are used. Outer loop is 0.016384
seconds ($FFFF loop) and the inner runs 25 times.
        LDAB     #$FFFF     ;therefore delay is 0.01634 * 25 = 0.4096
secs
DELAY   DECB     DECB
SM      DECA     DECA
        BNE     SM
        CMPB    #0
        BNE     DELAY
        PULA

        PSHY     ;This section displays the dec value of the
frequency being played, on the LED display
        LDX     $3400        ;Access frequency to display
        LDY     0,X
        STY     LED         ;Display to LED's
        INX
        INX
        STX     $3400        ;Store address of next frequency location
        PULY     ;Restore Y

        STAA     FREQ        ;Updated number of notes that have been
played, stored
        CMPA     #25         ;Checks if all 25 notes have been
played
        BNE     START      ;If not end of tones continue with next
tone
        LDY     #$3100       ;End of tones. Start loading durations
again from $3100
        STY     $3200        ;Point to beginning of duration-look-up table

        LDY     $3300        ;This bit of code initialises the LED's at the
end of one melody cycle
        STY     LED
        LDY     #$3302
        STY     $3400

```

```
        CLRA                ;End of playing 25 tones. Start from the
beginning
        STAA    FREQ        ;Updated number of tones stored at $3900
START   CLRA                ;End of 1 tone, but not end of tone-
look-up table
        STAA    CYCLES      ;Start CYCLES from zero
DONE    LDX     #$1000      ;Needed later for BCLR
        BCLR    TFLG1,x $7F ;This need to be cleared to enable
interrupt in the next cycle
        RTI                ;Return from ISR
        END
```

```
*-----END OF PART B-----
-----
```