

;SND1.SRC

;APPLE.SOUND

;P.P.

SPEAKER EQU \$C030

◇ ◇ ◇ ◇ ◇ ◇ ◇ NOTE.THAT.THE.ONLY.ZERO.PAGE.THAT.IS.NOT.TEMPORARY
AT.PRESENT.IS.Z1COUNT.....

PUBLIC .begin

.begin

JMP XSOUNDGO

PUBLIC TSUBTBL

PUBLIC XSOUNDGO ;INTERFACE.TO.SOUND

PUBLIC PLYPIANO ;SET.THIS.FOR.PIANO.INTERFACE

Z EQU 0 ;TRY.PUTTING.SOUND.ZP.RAM.AT.0000

NOTECOUNT EQU Z+15 ;DS 1

Z1COMP EQU Z+0 ;DS 2

Z2COMP EQU Z+2 ;DS 2

Z1COUNT EQU Z+18 ;DS 2

Z1WAVE EQU Z+4 ;DS 2 ;OCT.ADJ.

;PULSE.WIDTH

Z2WAVE EQU Z+6 ;DS 2

TIMEFACT EQU Z+12 ;DS 1

OCTUPUS EQU Z+08 ;DS 4

TEMPERARY EQU Z+13 ;DS 2

TEMP EQU Z+16 ;DS 1

TIME1 FCB 30

TIME3 FCB 33

TIME5 FCB 35

TIME7 FCB 44

SAVECOUNT fcb 1

LENADJ fcb 1

APPLEGO

NXTNOTE

GONOTE

LDA NOTECOUNT

;GET.NO

AND #\$F

ASL A

TAX

LDA NOTETBL,X

STA Z1COMP ;>>>Z1COMP<<<FREQUENCY

LDA NOTETBL+1,X

STA Z1COMP+1

LDA SLIDFLAG ;IF.SLIDE.NO.PULSE.VARIATION

BEQ DOWIDTH

LDA #0

STA Z1WAVE

STA Z1WAVE+1

STA FIRSTSLID ;CAN.SET.FIRST.FLAG.HERE.ALSO

BEQ DIDPWVAL

DOWIDTH

LDA PWVALUE

STA Z1WAVE

LDA PWVALUE+1

STA Z1WAVE+1

DIDPWVAL

LDA TIMETBL,X

STA OCTUPUS

LDA TIMETBL+1,X

STA OCTUPUS+1 ;>>>OCTUPUS<<<LENGTH.OF.TIME

LDA QUIK

BEQ OKQUIKS

LSR OCTUPUS+1

sent 9-3 as final Sound effect
maker

12
11
10
9
8
7
6
5
4
3
2


```

ROR OCTUPUS
OKQUIKS
LDA #0
STA LENADJ
STA OCTUPUS+2
STA OCTUPUS+3
LDA #35
STA TIMEFACT
LDA NOTECOUNT
LSR A
LSR A
LSR A
LSR A
ORA #F0
CLC
ADC #11
TAX
DEX
STX SAVECOUNT
DEC SAVECOUNT
OCTLOOP ;>>>>>>>>ROTATE.LENGTH.LEFT.AND.FREQ.RIGHT
; >>>>>>>>TO.GET.OCTAVE
LSR Z1COMP+1
ROR Z1COMP
LSR Z2COMP+1
ROR Z2COMP
LSR Z1WAVE+1 ;>>>>>PULSE.WIDTH.ALSO!!
ROR Z1WAVE
DEX
BMI OUTOCTLOOP
CPX SAVECOUNT
BEQ OCTLOOP
ASL OCTUPUS
ROL OCTUPUS+1
ROL OCTUPUS+2
ROL OCTUPUS+3
LDA LENADJ
CLC
ADC #22 ;>>>TRY.TO.ADJUST.FOR.CPU.TIME.IN.DRIVER
STA LENADJ
JMP OCTLOOP
OUTOCTLOOP
LDA OCTUPUS
BEQ DOOCT1ER
SEC
SBC LENADJ
BCS OKLEN1
DOOCT1ER
LDA OCTUPUS+1
PHP
DEC OCTUPUS+1
PLP
BEQ OKLEN2
BNE OKLEN1
OKLEN2
LDA OCTUPUS+2
PHP
DEC OCTUPUS+2
PLP
BNE OKLEN1
OKLEN3
LDA OCTUPUS+3
BNE OKOCT3
LDA #0

```



```

LDA    PUSSY+3
STA    OCTUPUS+3
LDA    SAMT
ORA    SAMT+1          ; CHECK. FOR. ZEROES
BNE    OKAMTSNZ
STA    AMTWORK+1
LDX    #1
LDA    AMTSIGN
BPL    STRAMT
LDX    #$FF
STX    AMTWORK+1
STX    AMTWORK
JMP    DIDAMT
OKAMTSNZ LDA    SAMT+1
ORA    AMTSIGN          ; <<<<<-----<<<<<-----<<<<<<<<<
AND    AMTANDOUT        ; PRECISION. OF. SIGN. DONE. HERE!!!!
STA    AMTWORK+1
LDA    SAMT
STA    AMTWORK
DIDAMT  LDA    OLDCOMP
CLC
ADC    AMTWORK
STA    Z1COMP
LDA    OLDCOMP+1
ADC    AMTWORK+1
STA    Z1COMP+1
BNE    NOSLUNDER        ; GO. CHECK. IF. OVER. LIMIT
LDA    Z1COMP            ; LOWER. LIMIT
CMP    #$20
BCS    LIMALRIGHT
LDA    AMTSIGN
BPL    SETSLUP
LDA    #$FF
STA    Z1COMP
LDA    #$07
STA    Z1COMP+1          ; SET. TO. UPPER. LIM
BNE    LIMALRIGHT
NOSLUNDER CMP    #$08
BCC    LIMALRIGHT
LDA    AMTSIGN
BMI    SETSLDOWN
LDA    #0
STA    Z1COMP+1
LDA    #$1F
STA    Z1COMP
LIMALRIGHT LDA    OLDCOMP
STA    SLPUTE
LDA    OLDCOMP+1
STA    SLPUTE+1
JSR    GETSLINDEX
STX    SVOLDINDEX
LDA    Z1COMP
STA    SLPUTE
LDA    Z1COMP+1
STA    SLPUTE+1
JSR    GETSLINDEX        ; X=NEW. INDEX(= OCTAVE)
TXA
SEC
SBC    SVOLDINDEX
STA    SLPUTE            ; JS. NEW - OLD INDEX
TAX
LDA    SLPUTE
BPL    PUTEOK

```


OKDOZW2 STA Z2COMP

LDA Z2COMP+1

CLC

ADC Z1WAVE+1

STA Z2COMP+1

END.DO.PULSE.WIDTH

ENDPWTW EQU *

INC Z1COMP

INC Z2COMP

INC Z1COMP+1

INC Z2COMP+1

DOZSLIDE

DO.THE.SLIDE

LDA SLIDFLAG

BEQ TONOSLIDE

LDA FIRSTSLID

BEQ TOSAVSLIDE

;FIRST.TIME.JUST.SAVE.VARS

AMONGSLIDE

LDX #0

LDA \$C000

CMP #\$9B

BNE NOTESCSL

LDA \$C010

RTS JMP NEXTNOTE

NOTESCSL

LDA \$C062

;BLACK.BUTTON.ABORTS

BPL DOAMONGS

RTS JMP NEXTNOTE

DOAMONGS

DEC SNUM

LDA SNUM

CMP #\$FF

BNE OKSLNUM

DEC SNUM+1

LDA SNUM+1

CMP #\$FF

BNE OKSLNUM

JMP NEXTNOTE

;FINISHED.SLIDING

TONOSLIDE

JMP NOZSLIDE

TOSAVSLIDE

DEC FIRSTSLID

STA OCTMINUS1

LDX #2

;FIX.HIGH.FREQ.LENGTH.BUG???

ADJOCTLP

LDA OCTUPUS+1,X

CMP #1

BCS OKSSOCT

INC OCTUPUS+1,X

OKSSOCT

DEX

BPL ADJOCTLP

;END.OF.FIX.BUG.ATTEMPT

JMP SAVSLIDE

SLPUTE

FCB 0,0,0

;SLIDE.COMPUTATION

SVOLDINDEX

FCB 0

;TO.COMPARE.THE.2.AND.ACT

OCTMINUS1

FCB 0

FIRSTSLID

FCB 0

OLDCOMP

FCB 0,0

FCB 0

;PART.OF.STIMEFACT(NEXT.BYTS)

STIMEFACT

FCB 0,0

;MUST.BE.A.FCB.0.AT.PREV.LINE

AMTSIGN

FCB 0

;PRECISION.OF.SIGN(11.BIT.)

AMTANDOUT

FCB 0

;POSITIVE.AND.OUT.NEGATIVES

AMTWORK

FCB 0,0

OKSLNUM

LDA PUSSY

STA OCTUPUS

LDA PUSSY+1

STA OCTUPUS+1

LDA PUSSY+2

STA OCTUPUS+2


```

LDA PUSSY+3
STA OCTUPUS+3
LDA SAMT
ORA SAMT+1 ; CHECK FOR ZEROES
BNE OKAMTSNZ
STA AMTWORK+1
LDX #1
LDA AMTSIGN
BPL STRAMT
LDX #$FF
STX AMTWORK+1
STRAMT STX AMTWORK
JMP DIDAMT
OKAMTSNZ LDA SAMT+1
ORA AMTSIGN ; <<<<<-----<<<<<-----<<<<<<<<<
AND AMTANDOUT ; PRECISION OF SIGN DONE HERE!!!!
STA AMTWORK+1
LDA SAMT
STA AMTWORK
DIDAMT LDA OLDCOMP
CLC
ADC AMTWORK
STA Z1COMP
LDA OLDCOMP+1
ADC AMTWORK+1
STA Z1COMP+1
BNE NOSLUNDER ; GO CHECK IF OVER LIMIT
LDA Z1COMP ; LOWER LIMIT
CMP #$20
BCS LIMALRIGHT
LDA AMTSIGN
BPL SETSLUP
SETSLDOWN LDA #$FF
STA Z1COMP
LDA #$07
STA Z1COMP+1 ; SET TO UPPER LIM
BNE LIMALRIGHT
NOSLUNDER CMP #$05
BCC LIMALRIGHT
LDA AMTSIGN
BMI SETSLDOWN
SETSLUP LDA #0
STA Z1COMP+1
LDA #$1F
STA Z1COMP
LIMALRIGHT LDA OLDCOMP
STA SLPUTE
LDA OLDCOMP+1
STA SLPUTE+1
JSR GETSLINDEX
STX SVOLDINDEX
LDA Z1COMP
STA SLPUTE
LDA Z1COMP+1
STA SLPUTE+1
JSR GETSLINDEX ; X=NEW INDEX(= OCTAVE)
TXA
SEC
SBC SVOLDINDEX
STA SLPUTE ; JS.NEW - OLD INDEX
TAX
LDA SLPUTE
BPL PUTEOK

```


EDR #\$FF
CLC
ADC #1
TAX

! #. OF. TIMES. TO. ADJUST. OCTUPUS

PUTEOK LDA SLPUTE
 BMI SLLEFT ;DO. LEFT. ROTATES (NEW. LESS. BY. X. AMOUNT)
 BEQ SAVSLXX

RSLROT DEX
 BMI SAVSLXX

 LSR OCTUPUS+3
 ROR OCTUPUS+2
 ROR OCTUPUS+1
 ROR OCTUPUS
 ROR OCTMINUS1

; ASL OCTMINUS1
; ROL OCTUPUS
; ROL OCTUPUS+1
; ROL OCTUPUS+2
; ROL OCTUPUS+3

; LSR STIMEFACT-1
; ROR STIMEFACT
; ROR STIMEFACT+1

; ASL SAMT-1
; ROL SAMT
; ROL SAMT+1
; ROL SAMT+2

 JMP RSLROT
SLLEFT BEQ SAVSLXX
XLLEFT DEX
 BMI SAVSLXX

; LSR OCTUPUS+3
; ROR OCTUPUS+2
; ROR OCTUPUS+1
; ROR OCTUPUS
; ROR OCTUPUS
; ROR OCTMINUS1

 ASL OCTMINUS1
 ROL OCTUPUS
 ROL OCTUPUS+1
 ROL OCTUPUS+2
 ROL OCTUPUS+3

; ASL STIMEFACT+1
; ROL STIMEFACT
; ROL STIMEFACT-1

; LSR SAMT+2
; ROR SAMT+1
; ROR SAMT
; ROR SAMT-1

SAVSLXX JMP XLLEFT
 LDA OCTUPUS
 STA PUSSY

 LDA OCTUPUS+1
 STA PUSSY+1
 LDA OCTUPUS+2
 STA PUSSY+2
 LDA OCTUPUS+3
 STA PUSSY+3

 LDA OCTUPUS+1
 BNE OKSOK1
 INC OCTUPUS+1
OKSOK1 LDA OCTUPUS+2
 BNE OKSOK2

12
11
10
9
8
7
6
5
4
3


```

INC      OCTUPUS+2
OKSOK2   LDA      OCTUPUS+3
        BNE      OKSOK3
        INC      OCTUPUS+3

```

OKSOK3

```

LDA      STIMEFACT
STA      TIMEFACT

```

```

JMP      SVDPUSS
SAVSLIDE LDA      SAMT+1
        AND      #$80

```

```

BEQ      POSIGN
LDA      #$FC      ;PRECISION.OF.11.BITS
STA      AMTSIGN   ;SAVE.SIGN.OF.AMOUNT
LDA      #$FF
STA      AMTANDOUT
BNE      DONESIGN

```

```

POSIGN   STA      AMTSIGN      ;POSITIVELY
        LDA      #$03      ;DELICIOUS
        STA      AMTANDOUT

```

```

DONESIGN LDA      TIMEFACT
        STA      STIMEFACT

```

```

LDA      #0
STA      STIMEFACT-1
STA      STIMEFACT+1
LDA      OCTUPUS
STA      PUSSY
LDA      OCTUPUS+1

```

```

STA      PUSSY+1
LDA      OCTUPUS+2
STA      PUSSY+2
LDA      OCTUPUS+3
STA      PUSSY+3
DEC      PUSSY+1

```

```

DEC      PUSSY+2
DEC      PUSSY+3
SVDPUSS LDY      #1

```

```

LDA      Z1COMP
STA      OLDCOMP
BEQ      WHOOPSV

```

```

WHOOPSV  STA      Z2COMP
        BNE      NXTSVDPUSS
        STY      Z2COMP
        STY      Z1COMP

```

```

NXTSVDPUSS LDA      Z1COMP+1
        STA      OLDCOMP+1

```

```

BEQ      NXZSAVER
STA      Z2COMP+1
BNE      NOZSLIDE

```

```

NXZSAVER STY      Z1COMP+1
        STY      Z2COMP+1

```

NOZSLIDE

```

LDX      RASTER ;<<<<<<-----<-----<-----<-----<

```

```

LDY      #1
LDA      Z1COMP
; ; ZERO. NOT. OK. HERE!
; BNE      OK1COM11 ;GOTTA.BE.A.1.
; STY      Z1COMP
OK1COM11 ;LDA      Z1COMP+1
; BNE      NEXTWAVE
; STY      Z1COMP+1

```

```

JMP      NEXTWAVE ;<<<<<<-----<-----<-----<-----<

```

```

ORGF1    - ;EQU  >*

```

12
11
10
9
8
7
6
5
4
3
2


```

;
GITOUTHERE      ;DS  256--ORGP1
NEXTWAVE        JMP  NEXTNOTE

```

```

    LDA  Z1COMP
    STA  Z1WAVE
    LDA  Z2COMP
    STA  Z2WAVE
    LDA  Z1COMP+1
    STA  Z1WAVE+1

```

18

```

    NOP      ;<<<<<
    NOP      ;<<<<<<<<WHAT.WAS
                LDA  $C062      ;TIME.FOR.BLACK.BUTTON?(1.XTRA.CC?)
                BMI  GITOUTHERE
    LDA  $C02F,X  ;SPEAKER

```

24.CCS
WL1

```

    DEC  Z1WAVE
    BNE  WL1
    DEC  Z1WAVE+1
    BEQ  OUTWL1
    DEC  Z1WAVE
    DEC  Z1WAVE
    BNE  WL1

```

7.OUT
OUTWL1

```

    LDA  Z2COMP+1
    STA  Z2WAVE+1
    LDA  OCTUPUS
    CLC
    SBC  TIMEFACT
    STA  OCTUPUS
    BCS  WAVEOK

```

```

    LDA  $C000      ;CHECK.ESCAPE.KEY.TO.GIT.OUT
    BMI  CHKWOBE

```

NOWSTROBE

```

    DEC  OCTUPUS+1
    BNE  WAVEOK
    DEC  OCTUPUS+2
    BEQ  EXITWAVE

```

WAVEOK

```

    LDA  $C02F,X

```

31.CCS
WL2

```

    DEC  Z2WAVE
    BNE  WL2
    DEC  Z2WAVE+1
    BEQ  NEXTWAVE
    DEC  Z2WAVE
    DEC  Z2WAVE
    BNE  WL2

```

7.OUT
CHKWOBE

```

    CMP  #$9B      ;ESC+$80
    BNE  NOWSTROBE ;IF.ESC.FALL.THROUGH
    RTS

```

WAVEOUT
EXITWAVE

```

    LDA  SLIDFLAG
    BEQ  NONOSLIDE
    JMP  AMONGSLIDE

```

NONOSLIDE

```

    JMP  NEXTNOTE

```

12
11
10
9
8
7
6
5
4
3

OCTAVE fcb 1,2,3,4,5
fcb 6,7,8,9,10,11,12,13

fcb 14

PTBL fcb \$80

fcb \$81

fcb \$82

fcb \$83

fcb \$84

fcb \$85

fcb \$86

fcb \$87

fcb \$88

fcb \$89

fcb \$8A

fcb \$8B

fcb \$8C

fcb \$8D

NOTETBL fdb 64814

fdb 61176

fdb 57743

fdb 54502

fdb 51443

fdb 48556

fdb 45830

fdb 43258

fdb 40830

fdb 38539

fdb 36376

fdb 34384

fdb 32407

TIMETBL

fdb 21*2

fdb 22*2

fdb 23*2

fdb 24*2

fdb 25*2

fdb 26*2

fdb 27*2

fdb 28*2

fdb 29*2

fdb 30*2

fdb 31*2

fdb 32*2

fdb 33*2

fdb 34*2

???????

SLCOMPARE fdb \$20,\$40,\$80,\$100,\$200,\$400,\$800,\$1000,\$2000

FCB 10,0,\$20,0,\$40,0,\$80,0,0,\$01,0,\$02,0,\$04,0,\$08,0,\$10,0,\$20,0,\$40,
\$>>>USED.TO.GET.OCTAVE.OF.NOTE.FOR.SLIDE.ADJUSTING.OCTUPUS(LENGTH)

GETSLINDEX LDX #11

LDY #22

GETSLXL LDA SLPUTE+1

CMP SLCOMPARE+1,Y

BEQ CHKLOWSL

BCS GOTSLXP

BCC NXTSLXP

CHKLOWSL LDA SLPUTE

CMP SLCOMPARE,Y

BCS GOTSLXP

NXTSLXP DEX

12
11
10
9
8
7
6
5
4
3
2

U.T.C. CORP. 6008

```

RSTER      RMB      1
RTIME      RMB      1
QUIK      FCB      0

```

TIME13	RMB	1
TIME17	RMB	1

\$OUNDORG		RMB	2
PASS	FCB	0	

\$NUM	FCB	0,0
PUSSY	FCB	0,0,0,0

	FCB	0	; >>> THIS.BYTE.GOES.WITH.SAMT. (NEXT)
SAMT	FCB	0,0,0	

```
*.FIRST.2.BYTES.OF.EACH.SOUND.=
*.SUBTBL.ADDRESS...
```

```

;*.CALL.XSOUNDGO.WITH:
;*.A=HIGH.ADDRESS.BYTE

```

```
*.MYPIANO(LOCAL) IS SET FROM PLYPIANO(EXTERN), WHICH SAYS  
*.TO CONTINUE.PIANO.OR.ABORT. IF.PLYPIANO.GOES.TO.ZERO
```


```

* CURRENTLY AT $2000
*****

```

```
*****
* ZICOUNT IS WHERE ABSOLUTE
*****
```

```
LDA    #0
STA    PWVALUE
STA    PWVALUE+1
STA    PWDELTA
```



```

        STA PWDELTA+1
        STA QUIK
        STA SLIDFLAG
        STA SLOOPCNT
        DEC SLOOPCNT
NEXTNOTE: EQU *
        LDY #0

        LDA #1
        STA RSTER
        INC TIME1
        LDA TIME3
        clc
        adc #31
        sta TIME3
        BNE OINKT3
        INC TIME3
OINKT3:  LDA TIME5
        ADC #5
        STA TIME5

OINKT5:  EQU *
        LDA TIME7
        ADC #7
        STA TIME7

OINKT7:  EQU *
        LDA TIME11
        ADC #11
        STA TIME11
        LDA TIME13
        ADC #51
        STA TIME13
        LDA TIME17
        SBC #17
        STA TIME17

OINKT11: EQU *
        LDA RTIME
        LSR A
        SBC TIME1
        ADC TIME3
        SBC TIME5
        ADC TIME7
        SBC TIME11
        ADC TIME13
        SBC TIME17
        STA RTIME
        LDA $C000
        BMI WAITGOX
        LDA NOIFLAG
        BEQ OKTINY
        JMP DONOISER

OKTINY:  EQU *
        LDA (Z1COUNT),Y
        BNE NORETS
        LDA MYPIANO
        BEQ NOTENDPNO
        LDA #0
        STA MYPIANO
        STA PLYPIANO
        SBC #1
NOTENDPNO:
        rts

```

12
11
10
9
8
7
6
5
4
3
2


```

WAITGO
WAITGOX
PLYPIANO      FCB      0      ;HOW. HE. TELLS. ME. TO. PIANO
MYPIANO       FCB      0      ;HOW. I. TELL. MYSELF. TO. CONTINUE
                                ;PLAYING. PIANO

```

```

XSOUNDGO
OKKEY1
DOKEY

```

```

      PHA
      LDA      MYPIANO
      BEQ      NOTPIANO
      LDA      PLYPIANO
      BEQ      NOTPIANO      ;ABORT. PIANO!
      PLA

```

```

      JMP      DOINGPIANO
NOTPIANO LDA      PLYPIANO      ;INTERFACE

```

```

      STA      MYPIANO
      ;NOW. DO. THE. SOUNDS
      PLA
      STX      Z1COUNT

```

```

      STA      Z1COUNT+1

```

```

      LDA      SOUNDS,X
      STA      Z1COUNT
      LDA      SOUNDS+1,X
      STA      Z1COUNT+1

```

```

      LDY      #0
      LDA      (Z1COUNT),Y
      INY
      CLC

```

```

      ADC      TSUBTBL
      STA      SUBTBL
      LDA      (Z1COUNT),Y
      ADC      TSUBTBL+1
      STA      SUBTBL+1

```

```

      LDA      Z1COUNT
      CLC
      ADC      #2
      BCC      OKZ1AOK
      INC      Z1COUNT+1
OKZ1AOK STA      Z1COUNT

```

```

      LDA      Z1COUNT
      STA      SOUNDORG
      LDA      Z1COUNT+1
      STA      SOUNDORG+1

```

```

;soundorg=abs.start.of.first.note
;z1count=abs.pointer.to.curr.note
;subtbl=abs.pointer.to.subtbl.start
      JMP      GOTUNEX

```

```

;*****
;ALSO.NEED.RELOCATE.SOUNDS
;AND.SUBTBL
;*****

```

```

      ;RTS
NORETS      BMI      DONOTES
NEXTBYT      EQU      *
      INC      Z1COUNT

```

12
11
10
9
8
7
6
5
4
3

BNE OKBYT
INC Z1COUNT+1

OKBYT EQU *

STA PASS
CMP ##30
BCS NEXTGOON
CMP ##20
BCS CHKEMA
LDA #11
BNE DOEMA

CHKEMA EQU *

CMP #12
BCS NEXTNGO

DOEMA EQU *

ASL A
TAX
LDA SUBS,X
STA GODOIT+1
LDA SUBS+1,X
STA GODOIT+2

GODOIT JSR NEXTNGO
NEXTGOON EQU *

JMP NEXTNOTE

SUBS FDB

0
FDB SLOOP
FDB SSETPW
FDB SSETPWD
FDB SBOTHPW
FDB SCLEARPW
FDB SDOPWD
FDB SREST
FDB SROTS
FDB SRETS
FDB STOGLQUIK
FDB SWHITE
FDB SSLIDE

JMP NEXTNGO

SREST LDA #0

STA RSTER
PLA
PLA
LDA ##C0
STA NOTECOUNT
BNE OKB2

DONOTES EQU *

STA NOTECOUNT
INC Z1COUNT
BNE OKB2
INC Z1COUNT+1

OKB2 LDA (Z1COUNT),Y

STA TIMEFACT
INC Z1COUNT
BNE OKB3
INC Z1COUNT+1

OKB3 EQU *

LDA MYPIANO
BEQ GOGGOS
LDA RTIME
and ##33
cmp oldpart
bne newrtm
eor RTIME

12
11
10
9
8
7
6
5
4
3
2


```
newrtm      lda      RTIME
            and      ##33
            sta      oldpart
            AND      ##3
```

```
TAX
            lda      RTIME
            ROR A
            ROR A
            ror A
            AND      ##6
```

```
RTS
```

```
oldpart fcb 0
GOGGIES
DOINGPIANO
```

```
JMP GONOTE ;GOES TO PLAY IT
```

```
ie
SLOOPCNT    FCB      $FF
SLOOP        ;EQU    *
```

```
LDA SLOOPCNT
BMI STARTLOOP
DEC SLOOPCNT
BMI ENDLOOP
```

```
BOMYLOOP    ;EQU    *
            INC      Z1COUNT
            BNE      OKZ11
            INC      Z1COUNT+1
```

```
OKZ11        ;EQU    *
            LDA      (Z1COUNT),Y
```

```
PHA
            INY
```

```
OKZ12        LDA      (Z1COUNT),Y
            STA      Z1COUNT+1
            PLA
            STA      Z1COUNT
```

```
*****
;*.USE.SOUNDRG.TO.COMPUTE
;*.ABSOLUTE.ADDRESS.OF.LOOP.FROM
;*.RELATIVE.ADDRESS.
*****
```

```
LDA SOUNDRG
CLC
ADC Z1COUNT
STA Z1COUNT
```

```
OKZ1NL      LDA SOUNDRG+1
            ADC      Z1COUNT+1
            STA      Z1COUNT+1
```

```
RTS
```

```
STARTLOOP   LDA      (Z1COUNT),Y
            STA      SLOOPCNT
            JMP      BOMYLOOP
```

```
ENDLOOP     LDA      Z1COUNT
            CLC
```

```
ADC #3
BCC OKENDL
INC Z1COUNT+1
```

```
OKENDL      STA      Z1COUNT
            RTS
```

```
SSETPW      ;EQU    *
```

```
LDA (Z1COUNT),Y
STA PWVALUE
INY
LDA (Z1COUNT),Y
STA PWVALUE+1
```

12
11
10
9
8
7
6
5
4
3

LDA Z1COUNT
CLC
ADC #2
BCC OKZ13
INC Z1COUNT+1

OKZ13 STA Z1COUNT
DEY
RTS

SSETPWD :EQU *

LDA (Z1COUNT),Y
STA PWDELTA
INY
LDA (Z1COUNT),Y
STA PWDELTA+1
LDA Z1COUNT
CLC

ADC #2
BCC OKZ14
INC Z1COUNT+1

OKZ14 STA Z1COUNT
DEY
RTS

SDOPWD :EQU *

LDA PWVALUE+1
STA TEMPERARY+1
LDA PWVALUE
CLC
ADC PWDELTA

STA TEMPERARY
LDA PWDELTA+1
BMI CHKPONS
BCC OK1PWDO
LDA TEMPERARY+1
ADC #0

BCS NEGPWD
STA TEMPERARY+1
BCC OK1PWDO

CHKPONS BCS OK1PWDO

LDA TEMPERARY+1
SBC #0
BCC NEGPWD
STA TEMPERARY+1

OK1PWDO :EQU *

LDA TEMPERARY+1
CLC
ADC PWDELTA+1

STA TEMPERARY+1
LDA PWDELTA+1
BMI DOCHDOWN
BCS NEGPWD

NONEGPWD :EQU *

LDA TEMPERARY
STA PWVALUE
LDA TEMPERARY+1
STA PWVALUE+1
RTS

DOCHDOWN BCS NONEGPWD

NEGPWD :EQU *

LDA PWDELTA+1
EOR #\$FF
STA PWDELTA+1
LDA PWDELTA
EOR #\$FF

12
11
10
9
8
7
6
5
4
3
2

STA PWDELTA
INC PWDELTA
BNE DUNGNeg
INC PWDELTA+1

DUNGNeg ;EQU *

SCLEARPW

RTS
LDA #0
STA PWDELTA
STA PWDELTA+1
STA PWVALUE
STA PWVALUE+1
RTS

SBOTHWP JSR SSETPW

JMP SSETPWD

HOLDROT RMB 2
SROTS ;EQU *

LDA SUBTBL
STA TEMPERARY
LDA SUBTBL+1
STA TEMPERARY+1
LDA (Z1COUNT),Y
PHA
INC Z1COUNT
BNE OKIZS
INC Z1COUNT+1

OKIZS LDA Z1COUNT

STA HOLDROT
LDA Z1COUNT+1
STA HOLDROT+1
PLA
AND RTIME

CLC
ADC (Z1COUNT),Y
ASL A
BCC PGSB1
INC TEMPERARY+1

PGSB1 ;EQU *

TAY
LDA (TEMPERARY),Y
STA Z1COUNT
INY
LDA (TEMPERARY),Y
STA Z1COUNT+1

*.USE.SOUNDORG.AGAIN.FOR
*.SUBROUTINE.TABLE

LDA SOUNDORG
CLC
ADC Z1COUNT
STA Z1COUNT
LDA SOUNDORG+1
ADC Z1COUNT+1
STA Z1COUNT+1
LDY #0

SRETS ;EQU *

RTS
LDA HOLDROT
STA Z1COUNT
LDA HOLDROT+1

12
11
10
9
8
7
6
5
4
3

STA Z1COUNT+1
INC Z1COUNT
BNE HROK1
INC Z1COUNT+1

HROK1

RTS
STOGLQUIK LDA QUIK
EDR #\$FF
STA QUIK
RTS

EXITGOER

LDA #0
STA NOIFLAG
RTS
;JMP NEXTNOTE

OTHRNOSE

CKNSTOP

DEC NOITIMER
BEQ EXITGOER ;TRY.FOR.RAND.PITCH.HIGHER
LDA SPEAKER

NEWNOISE

LDA #31
ADC TIME3
STA TIME3

LDA #17
SBC TIME17
STA TIME17

LDA #51
ADC TIME13
STA TIME13

LDA \$C000
CMP #\$9B
BEQ EXITGOER
LDA RTIME

DONOISER

NXTNOISE

LSR A
LSR A
LSR A
BNE OKRTY
LDA #1

OKRTY

TAY
LDA PASS
LDX SPEAKER

DRIVE

;EQU *

TAX

DODEX

DEX

BPL DODEX
DEY
BNE DRIVE
INC TIME1
LDA TIME1
AND #7
BEQ OTHRNOSE

LDA RTIME
ADC TIME3
SBC TIME13
ADC TIME17
STA RTIME
JMP DONOISER

WHITE

LDA (Z1COUNT),Y
INC Z1COUNT
BNE ZWHZ
INC Z1COUNT+1

12
11
10
9
8
7
6
5
4
3
2

ZWHZ

STA NOITIMER
LDA #\$FF
STA NOIFLAG

; PLA
; PLA

LDA PASS
AND #\$F
ASL A
ASL A
STA PASS

CMP #\$8
BCC OKDNOISE
LDA NOITIMER
LSR A

BNE STTMRN
LDA #1
STA NOITIMER
JMP NXTNOISE

STTMRN
OKDNOISE

SSLIDE LDA #\$FF
STA SLIDFLAG
LDA (Z1COUNT),Y
STA SNUM

INY
LDA (Z1COUNT),Y
STA SNUM+1

INY
LDA (Z1COUNT),Y
STA SAMT

INY
LDA (Z1COUNT),Y
STA SAMT+1

LDA Z1COUNT

CLC
ADC #4
STA Z1COUNT
BCC OKZ1SL1
INC Z1COUNT+1

OKZ1SL1 RTS

12
11
10
9
8
7
6
5
4
3

AS12 222
356

CTL : 16 FLA 8

108 exp.

136

139

ZICANT : zero p
Temperature (L)

405, 419, 420