

COPY A: BUTS.SRC COM1:
; BUTS.FILE
; BUTTON.ROUTINES
; MUED...MUSIC.EDITOR

Music maker

```
CUTBUF EQU $800 ;TO.$17FF
PUBLIC BUT0,ARROWP,DOARROW,DOFWORDS,DONWORDZ
PUBLIC BEDIT,BPLAY,BSTOP,BUTADR,MAXFLAG
PUBLIC BUT4,BQUIT,BCLR,BTEMPO,MOVRIGHT,MOVLEFT
PUBLIC BLOAD,BSAVE,BINIT,BDEL,BEDIT,BPLAY,BSTOP
PUBLIC BJFTT,WAITLONG,WAITSHORT,OURVBLANK,STICK1
PUBLIC THLLOOP,THLTBL,LRAOFF,LRARR,NOTEX,SCRPOS,NUMNOTES
PUBLIC BLARROW,WHITESPC,BLACKSPC,LETTER,HIOUT,CX,CY
PUBLIC SAVNOTE,HIIN,INV,HWLINE
PUBLIC QFLAG,REDRAW,REDR2,FLASHNOTE,NCHANNEL,DSPNOTE
PUBLIC GETINDEX,TO,T1,T8,BACKFLAG,OLDBFLAG
PUBLIC FRAME,SITARROW,KINDANOTE,MAXKINDAS
PUBLIC NLENGTHS,PLAY2NOTES,NRESTS,NTRANS
PUBLIC NDRIVE,NSCALE,GODRIVE,NDURS,NDURSM,NDURSF,TEMPO
PUBLIC PLAYING,PLAYCUT,CUTWHO,CLRVAR,CLRNOTES,CLRSNOTES,CLRSVARS
PUBLIC WARMSTART,CLBRD,RUSURE,PBOXMES,MCLEAR,MESSAGE
PUBLIC BCH1,BCH2,ML1POS,NTIEFLG
PUBLIC BINSERT,BDELETE,CKEYON,CKEYOFF
PUBLIC BINIT,FILEDELETE,BLOAD,BSAVE,BTIE
PUBLIC UPKINDA,B12FLAG ;FOR.BIG.STAFF.LEFT.JOY
PUBLIC CUTPTR ;POINTS.TO.SCRPOS.CUT.LOCATION
PUBLIC VBLANK,JOYS,OURVBLANK,BUTTON,ENDB6X
PUBLIC STOPCONT,MCONT,MSTOP,PMES,MES
```

INSTXT /B:SPED0.SRC/

;
;* BUTTON ADDRESSES

```
BRTS RTS
BUTADR FDB BUT0 ;FILE
FDB BINSERT ;INS
FDB BDELETE ;DEL
FDB BCLR ;CLR
FDB BQUIT ;QUIT
FDB BUT5 ;TOP
FDB BUT6 ;PLAY
FDB STOPBUT ;STOP
FDB BUT8 ;LFT.RIT.ARROWS
FDB BTEMPO ;TEMPO
FDB BCH1 ;CHANNEL.1
FDB BCH2 ;CHANNEL.2
FDB BUT12 ;INDICATOR.NOTE
```

;FILE.BOXES

```
FDB BLOAD ;13.load file
FDB BSAVE ;14.SAVE
FDB BINIT ;15.init disk...BINIT
FDB FILEDELETE ;16.DELETE.FILE
FDB BEDIT ;17:QUIT FILE.MODE
FDB BCLR ;18:CLEAR.SONG
FDB BUT6 ;19:PLAY.THE.SONG
FDB STOPBUT ;20:HALT.THE.SONG
```

;MUED.SO.FAR

;FILE NAME CHARACTERS

```
FDB BTIE ;21
FDB BRTS ;22
FDB BRTS ;23
FDB BRTS ;24
FDB BRTS ;25
FDB BRTS ;26
```

```

FDB      BRTS      ;27
BUTO LDA      #13
STA      ARROWP
JSR      DOARROW
JSR      DOFWORDS
JSR      DOARROW
RTS

BEDIT LDA      #0
STA      ARROWP
JSR      DOARROW
JSR      DOWWORDZ
JSR      DOARROW
RTS

BCLR LDX      #CMCLEAR-MESSAGE
LDA      #28
LDY      #4
JSR      PBOXMES
JSR      RUSURE
JSR      ENDB6X
JSR      CLRSVAR
JSR      CLRSNOTES      ;REDRAW, AND, CLEAR!!! (TRY, IT)
JMP      CLRBRD

BPLAY BSTOP RTS
CORKFLAG FCB      0
STOPFLAG FCB      0      ; TELLS, IF, IN, STOP, OR, CONT.
KILLPLAY FCB      0      ; TELLS, PLAY, TO, STOP
PLAYING FCB      0      ; TELLS, IF, IN, PLAY, MODE
PLAYCUT FCB      0      ; TELLS, IF, AND, WHICH, CHANNEL, LENGTH, IS, CUT
CUTWHO FCB      0,0,0,0 ; HOLDS, CUT, VALUE
CUTME FCB      0      ; TELL, WHO, TO, CUT (LOCAL)
CUTPTR FCB      0      ; DUMMY
CUTPITCH FCB      0,0   ; SAVE, PITCH, OF, WHO, IS, CUT
CUTLOC FCB      0,0     ; SAV, LOCATION, OF, WHO, CUT
MANIM FCB      0
LRBLINK FCB      0
HOLDER FCB      0
OUTB8 LDA      BJFTT
BNE      INB8
DEC      BJFTT
JMP      LRARRON      ; EXIT, WITH, LR, ARROWS, ON
BUT8 ; MOVE, NOTES, LEFT, OR, RIGHT
; JSR      WAITLONG
JSR      ENDB6X
; LDA      #0      ; ENDB6, DOES, THIS
; STA      PLAYCUT
JSR      OURVBLANK
JSR      LRARRBLINK
LDA      STICK1
AND      #$10
BEQ      OUTB8
LDA      #0
STA      BJFTT
INB8 LDA      STICK1
AND      #$4
BEQ      DOMRITE ; LEFT/RIGHT, JOY, ACTION, LOGIC, IS, INVERTED
LDA      STICK1
AND      #$8
BEQ      MOVEL
BNE      BUT8
MOVEL JSR      ML1POS
JMP      BUT8

```

12
11
10
9
8
7
6
5
4
3
2

```

DOMRITE JSR MR1POS
        JMP BUT8
LEFTFLAG FCB 0
MOVLEFT LDA #0
        BEQ DOALEFT
ML1POS LDA #FF
DOALEFT STA LEFTFLAG
        LDA SCRPOS
        CMP NUMNOTES
        BCC LMOK1
; BEQ LMOK1 ;FOR, ONE, MORE, THAN, NUMNOTES!
        LDA SCRPOS
        BEQ LMOK1
        RTS
LEN1 FCB 0
LEN2 FCB 0
LMOK1 CMP #F8 ;MAX, #, NOTES, FOR, NOW
        BCC LMOK2
        STA MAXFLAG
        RTS
LMOK2 ;CHECK, HOW, FAR, TO, SCROLL, 1, POS, PER, 32nd, NOTE
; JSR GETINDEX
; LDY #1
; LDA (T0),Y
; LDY #0
; STY LEN1
; STY LEN2
; CMP #FE ;EMPTY
; BCS GL1L
; LSR A ;COMPUTE, HOW, FAR, TO, SCROLL
; TAX
; LDA NLENGTHS,X
; STA LEN1
GL1L ;LDY #3
; LDA (T0),Y
; CMP #FE
; BCS TESTL1
; LSR A
; TAX
; LDA NLENGTHS,X
; STA LEN2
TESTL1 ;LDA LEN1
; ORA LEN2
; BEQ NODOL ;IF, BOTH, ZERO, DONT, DO, A, THING.
; LDA LEN1
; BEQ DOL21IZ
; LDA LEN2
; BEQ GTRLX ;IF, ONLY, ONE, IS, ZERO, BE, SURE, TO, DO, OTHER, ONE.
; LDA LEN1
; CMP LEN2
; BCC GTRLX
DOL21IZ ;LDA LEN2
; STRL1 ;STA LEN1
GTRLX ;LEN1, CONTAINS, HOW, MANY, PLACES, TO, GO
; <<-----,
; <<-----, CODE, INSERTED, TO, SCROLL, ONLY, ONE, POSITION
; <<-----,
        LDA #1 ;<<-----, JUST, THIS, LINE, ONLY
        CLC
        ADC SCRPOS ;TEST, OVER, MAX
; BCS NODOL
        CMP #F8
        BCS MNODOL ;GOES, IF, OVER, MAX#NOTE, POSITIONS

```

LMOK	STA	LEN2	DESTINATION, POSITION
	INC	SCRPOS	
	LDA	SCRPOS	
	CMP	LEN2	
	BCS	DONEL	
\$LDA	PLAYING		;FLAG, SAYS, IF, EDITING, OR, PLAYING
\$BNE	DOLRDR		; IF, PLAYING, DONT, TEST, FOR, NEW, NOTES
	JSR	GETINDEX	;LOOK, FOR, A, NOTE, TO, STOP, BEFORE, NEXT, ONE, ON
	LDA	NUMNOTES	; IF, PLAYING, NEED, TO, TEST, FOR, END!!!!
	CMP	SCRPOS	
	BCS	DOLRDR	
\$JSR	GETINDEX		
	LDY	#0	;DO, THIS, BECAUSE, WEARE, GOING, TO
	LDA	\$\$FE	;A, NEW, NOTE, POSITION, AND, NEW, NOTES
	STA	(TO),Y	;NEED, TO, BE, NOOPED, OUT, (FE=NOOP)
	INY		
	STA	(TO),Y	
	INY		
	STA	(TO),Y	
	INY		
	STA	(TO),Y	
DOLRDR	JMP	DSPNNEXT	
	LDA	(TO),Y	
	CMP	\$\$FE	
	BNE	DSPNNEXT	
	INY		
	INY		
	LDA	(TO),Y	
	CMP	\$\$FE	
	BNE	DSPNNEXT	
	JMP	DONEL	; IF, FOUND, STOP, HERE,.....
DSPNNEXT			
	LDA	PLAYING	
	BNE	NOSLOWGO	
	JSR	REDRAW	;DISPLAY, NEW, POSITION
NOSLOWGO			
	JMP	LMOK	;LOOP
DONEL	LDA	NTIEFLG	
	BEQ	OKREDRW	
	LDA	PLAYING	
	BNE	NOREDREW	
OKREDRW	JSR	REDRAW	
NOREDREW	LDA	LEFTFLAG	
	BEQ	NODOL	
	JSR	PLAY2NOTES	
NODOL			
	RTS		
MXNODOL	STA	MAXFLAG	
	RTS		
HX2	FCB	0	
LRARRBLINK	INC	LRBLINK	
	LDX	#<LRAOFF-THLTBL	
	LDA	LRBLINK	
	AND	#1	
	BNE	DOLRBX	
LRARRON	LDX	#<LRARR-THLTBL	
DOLRBX	STX	HX2	
	LDA	\$\$80	
	STA	ARROWP	
	JSR	DOARROW	
	LDX	HX2	
	JSR	THLLOOP	


```

LDA    #8
STA    ARROWP
INC     FRAME
JMP     SITARROW

```

```

MRFLAG FCB    0

```

```

MOVRIGHT LDA  #0

```

```

BEQ     DOMR1

```

```

MR1POS LDA  #$FF

```

```

DOMR1  STA  MRFLAG

```

```

LDA     #1

```

```

STA     MANIM

```

```

LDA     SCRPOS

```

```

BNE     RMOKX ; IF.ON.FIRST.NOTE.CANNOT.MOVE.RIGHT

```

```

RTS

```

```

RMOKX  DEC  SCRPOS

```

```

RMOKX  LDA  #26

```

```

SEC

```

```

SBC     MANIM

```

```

STA     NOTEX ; REDR2.ACCEPTS.DIFFERENT.VALUES.OF.NOTEX

```

```

STA     NOTEX+1 ; REDRAW.USES.SCRPOS.TO.COMPUTE.NOTEX

```

```

STA     QFLAG

```

```

JSR     REDR2 ; <----- <-----

```

```

DEC     MANIM

```

```

BNE     RMOK

```

```

JSR     REDRAW

```

```

LDA     MRFLAG ; IF.MOVRIGHT.CALLED.THEN.EXIT.NOW.DONT.CHECK

```

```

BEQ     MRDONE ; IF.GOT.TO.A.NOTE.OR.BEGINNING

```

```

LDA     SCRPOS

```

```

BEQ     NOMORR

```

```

JSR     GETINDEX ; CHECK.IF.GOT.TO.A.NOTE

```

```

LDY     #0

```

```

LDA     (T0),Y

```

```

CMP     #$FE

```

```

BNE     NOMORR

```

```

LDY     #2

```

```

LDA     (T0),Y

```

```

CMP     #$FE

```

```

BEQ     MR1POS ; IF.EMPTY.DO.ANOTHER.POSITION!!!!

```

```

NOMORR JSR  PLAY2NOTES

```

```

MRDONE

```

```

RTS

```

```

GETINDEX

```

```

LDA     #0

```

```

TAY     ; <<<<-----.Y<--0

```

```

STA     T1

```

```

LDA     SCRPOS

```

```

ASL     A

```

```

ROL     T1

```

```

ASL     A

```

```

ROL     T1 ; *4

```

```

CLC

```

```

ADC     #$20

```

```

BCC     OKT0

```

```

INC     T1

```

```

OKT0  STA  T0

```

```

LDA     T1

```

```

CLC

```

```

ADC     #$40

```

```

STA     T1

```

RTS

```

;
SAVONLOC      FCB      0
SAVINLOC      FCB      0      ; SAV. NLOCs. FOR. PIANO. KEY. COLORING
PLFLAG FCB      0
PLAY2NOTES    : <----FOR. TESTING....<----FOR. DOING!
LDX           ##FF
STX           SAVONLOC
STX           SAVINLOC
INX

```

```

;STX      NDRIVE+1
;STX      NDRIVE+2      ; ZERO. BOTH. PITCHES. FOR. NOW
STX      PLFLAG      ; IF. PLFLAG. ZERO. NO. NOTES. TO. PLAY
LDA      PLAYCUT
CMP      #1
BEQ      ONECUT      ; IF=1. THIS. CHANNEL. HAS. BEEN. CUT

```

```

LDA      #0
STA      NDRIVE+1      ; CHANL. 1. NOT. CUT
STA      CUTWHO

```

```

JSR      GETINDEX
LDA      (T0),Y
CMP      ##FE      ; IF. NOOP. OR. ENDBUF. NO. PLAY. THIS. CHANNEL.

```

```

BCS      NOPLC1
DEC      PLFLAG      ; SET. FLAG. SAYING. SOMEONE. NEEDS. TO. PLAY.
INX

```

```

LDA      (T0),Y      ; CHECK. FOR. RESTS
JSR      GETDUR

```

```

STA      CUTWHO      ; SAVE. LENGTH. TO. TEST. FOR. CUT
LDA      NRESTS,X
BNE      NOPLC1      ; TABLE. OF. FLAGS

```

```

DEY      ; IF. TRUE. IS. A. REST.
LDA      (T0),Y      ; BACK. TO. LOCATION
TAX

```

```

LDA      NSCALE,X
STA      NDRIVE+1      ; SET. PITCH. 1
STA      CUTPITCH

```

```

STX      CUTLOC
; <<<<<-----
; CODE. INSERTED. TO. COLOR. PIANO. KEY
TXA      ; NLOC
STA      SAVONLOC
LDX      #0      ; CHANNEL. #
JSR      CKEYON      ; COLOR. ON

```

```

; <<<<<-----
ONECUT
NOPLC1 JSR      GETINDEX      ; GET. BACK. TO. T1,Y.
LDA      PLAYCUT
CMP      #2
BEQ      TWOCUT      ; CHECK. IF. CHAN. 2. CUT
LDA      #0
STA      CUTWHO+1
STA      NDRIVE+2
LDY      #2
LDA      (T0),Y
CMP      ##FE
BCC      OKPL2
LDA      PLFLAG
BNE      OKTOPLAY

```

```

OKPL2 RTS      ; DONT. PLAY. 2. NONEXISTENT. NOTES
INX
LDA      (T0),Y
JSR      GETDUR
STA      CUTWHO+1      ; INTO. CHANNEL. 2. LENGTH

```

```

LDA      NRESTS,X      ;TABLE.OF.FLAGS
BNE      OKTOPLAY      ;IF.TRUE.IS.A.REST.
DEY
LDA      (TO),Y        ;BACK.TO.LOCATION

```

```

TAX
LDA      NSCALE,X
STA      NDRIVE+2      ;SECOND.PITCH
STA      CUTPITCH+1
STX      CUTLOC+1

```

; <<<<----CODE.TO.COLOR.KEY.ON.PIANO

```

TXA
STA      SAV1NLOC
LDX      #1
JSR      CKEYON        ;COLOR.2ND.PIANO

```

; <<<<----

```

OKTOPLAY
TWOUCUT      ;NOW.CHECK.IF.NEED.TO.CUT.SOMEONE!
LDA      CUTWHO      ;IF.A.LENGTH.IS.ZERO
BNE      CW10K        ;DONT.CUT,MEANS.THAT.ONE.CHANNEL
LDA      CUTWHO+1     ;MUSIC.IS.GOIN.ON.
BEQ      MRNOBODY     ;BOTH.ZERO.DO.NOTHING

```

```

DOOOP      STA      CUTWHO+2      ;SET.WHATEVER.IS.NONZERO.AS.LENGTH
BNE      ZCUP          ;GO.TO.RESET.CUT.FLAG.AND.PLAY.
CW10K      LDA      CUTWHO+1
BNE      TESTCUTS
LDA      CUTWHO
BNE      DOOOP

```

```

TESTCUTS
LDA      CUTWHO
CMP      CUTWHO+1
BNE      DOACUT
STA      CUTWHO+2

```

```

ZCUP
LDA      #0
STA      PLAYCUT
DOTHEPLAY LDA      CUTWHO+2      ;PUT.DURATION.INTO.TABLE

```

```

STA      NDRIVE
JSR      GODRIVE
LDA      SAVONLOC
CMP      #$FF
BEQ      NOBACKKO
LDX      #0

```

```

JSR      CKEYOFF
NOBACKKO LDA      SAV1NLOC
CMP      #$FF
BEQ      NOBAK1K
LDX      #1
JSR      CKEYOFF

```

```

NOBAK1K RTS
MRNOBODY LDA      #0
STA      PLAYCUT

```

```

RTS
DOACUT      BCS      CUTN1      ;UNEQUAL,NEED.TO.CUT.BIGGER.LENGTH
LDA      #2      ;BUT.NOT.IF.NOT.PLAYING
STA      CUTME      ;IDENTITY.OF.CUT(CH.2)
LDA      CUTWHO

```

```

STA      CUTWHO+2      ;LENGTH.TO.PLAY=SHORTER.ONE
LDA      CUTWHO+1      ;LARGER.ONE.FOR.COMPUTING
STA      CUTWHO+3
JMP      JEEZEE

```

```

CUTN1      STA      CUTWHO+3      ;LARGER.IS.#0

```

	LDA	#1	
	STA	CUTME	
	LDA	CUTWHO+1	; SMALLER, ONE, TO, PLAY
	STA	CUTWHO+2	
JEEZEE	;LDA	PLAYING	
	;BNE	DOCUT	
	;STA	PLAYCUT	; NO, CUT, IF, MOVING, JOYSTICK (NOT, PLAY)
	;JMP	DOTHEPLAY	; GO, PLAY, IT, NOW.
DOCUT	LDA	CUTWHO+3	
	SEC		
	SBC	CUTWHO+2	; SUB, SMALLER, FROM, LARGER
	LDY	CUTME	
	STY	PLAYCUT	; SAVE, WHO, IS, CUT
	DEY		
	STA	CUTWHO, Y	; SAVE, NEW, CUT, VALUE
	JMP	DOTHEPLAY	; WILL, BE, USED, NEXT, TIME
BUT5	;GO, TO, TOP, OF, MUSIC!		
	JSR	ENDB6X	
MUSICTOP			
	LDA	#0	
	STA	SCRPOS	
	JMP	REDRAW	
BUT4	JMP	BQUIT	
BUT1	;INS...INSERT		
BUT2	;DEL...DELETE		
BUT3	;CLR...CLEAR		
BUT7	;STOP..STOP		
BUT9	;TEMPO,TEMPO		
BUT10	;CHANNEL,1		
BUT11	;CHANNEL,2		
	RTS		
FCNT	FCB	0	
B12FLAG	FCB	0	
BUT12	;INDICATOR,NOTE		
	LDA	#0	
	STA	B12FLAG	; SAYS, IF, ENTERED, THROUGH, BIG, STAFF, LEFT, JOY
B12LOOP			
	INC	FCNT	
	LDA	FCNT	
	AND	##01	
	STA	FLASHNOTE	
	LDY	NCHANNEL	
	LDA	KINDANOTE	
	LDX	#1	; SAYS, INDICATOR
	JSR	DSPNOTE	
	INC	FRAME	
	JSR	OURVBLANK	
	LDA	B12FLAG	; IF, MOVED, BY, BIG, STAFF, LEFT, JOY
	BNE	ENDB12	
	LDA	STICK1	
	AND	##10	
	BEQ	HITB12	
	LDA	#0	
	STA	BJFTT	
	LDA	STICK1	
	LSR	A	
	BCC	UPKINDA	
	LSR	A	
	BCC	DNKINDA	
	LSR	A	

	BCC	DNCHAN	
	LSR	A	
	BCC	UPCHAN	
	JMP	B12LOOP	
HITB12	LDA	BJFTT	
	BNE	B12LOOP	
	LDA	#\$FF	
	STA	BJFTT	
ENDB12			
	LDA	#0	
	STA	FLASHNOTE	;NO.MORE.FLASH.INDICATOR
	LDA	#0	
	STA	OLDBFLAG	
	LDY	NCHANNEL	
	LDA	KINDANOTE	
	LDX	#1	
	JMP	DSPNOTE	;THROW.IT.UP.THERE!
	RTS		;EXIT.BUT.12
TRANSKINDA	FCB	0	
UPCHAN			
DNCHAN			
	LDA	NCHANNEL	
	EOR	#\$1	
	STA	NCHANNEL	
	JSR	WAITLONG	
TOB12	JSR	WAITLONG	
	JSR	WAITLONG	
	JSR	WAITLONG	
	JMP	B12LOOP	
UPKINDA			
	LDX	TRANSKINDA	
	INX		
	CPX	#15	
	BCS	TKOVER	
OKROLL	LDA	NTRANS,X	
	ASL	A	
OKKINDA	STA	KINDANOTE	
	STX	TRANSKINDA	
NOTROLL	JMP	TOB12	
TKOVER	LDA	B12FLAG	
	BEQ	NOTROLL	
	LDX	#0	
	BEQ	OKROLL	
DNKINDA			
	LDX	TRANSKINDA	
	LDA	KINDANOTE	
	BNE	DODNKINDA	
	LDA	B12FLAG	
	BEQ	NOTROLL	
	LDX	#15	
DODNKINDA	DEX		
	JMP	OKROLL	
BUT6	;PLAY		
	JSR	ENDB6X	
BUT6GIN	JSR	MUSICTOP	;GO.TO.MUSIC.TOP
PLAYNOTOP			
	LDA	#1	
	STA	PLAYING	
	LDA	#0	
	STA	PLAYCUT	
	JSR	PLAY2NOTES	;PLAY.FIRST.GUYS
BUT6LOP	LDA	NUMNOTES	
	CMP	SCRPOS	

	BEQ	ENDB6	
	JSR	ML1POS	; MOVE.LEFT.AND.PLAY
	LDA	%C000	; CHECK.ESCAPE.KEY.ON.APPLE
	CMP	##9B	
	BEQ	ENDB6	
	LDA	NTIEFLG	
	BNE	BUT&LOP	
	JSR	VBLANK	
	JSR	JOYS	
	JSR	OURVBLANK	
	JSR	BUTTON	
	LDA	KILLPLAY	
	BEQ	BUT&LOP	
ENDB6	JSR	REDRAW	
	LDA	#0	
	STA	PLAYING	
	STA	PLAYCUT	
	STA	KILLPLAY	
	RTS		
ENDBB	LDA	#0	
	STA	PLAYING	
	STA	PLAYCUT	
	STA	KILLPLAY	
OKRTSP	LDA	#0	
	STA	STOPFLAG	
	JSR	STOPCONT	
	JMP	STOPCONT	
	RTS		
ENDB6X	LDA	PLAYING	
	BEQ	ENDBB	
	STA	KILLPLAY	
	RTS		
STOPBUT	LDA	##FF	
	BNE	INSTOPCONT	
STOPCONT	LDA	#0	
INSTOPCONT	STA	CORKFLAG	
	LDX	##CMSTOP-MES	
	LDA	STOPFLAG	; IF.STOPFLAG=NZ.THEN.CONT.IS.DISPLAYED
	BNE	GOSTOPX	
	LDX	##CMCONT-MES	
	LDA	##FF	
	BNE	GOSTOPY	
GOSTOPX	LDA	#0	
GOSTOPY	STA	STOPFLAG	
GOSTOP	LDA	#35	
	LDY	#11	
	JSR	PMES	
	LDA	CORKFLAG	
	BNE	DOACORK	
	RTS		
DOACORK	LDA	#0	
	STA	CORKFLAG	
	LDA	STOPFLAG	
	BNE	ENDB6X	
	JMP	PLAYNOTOP	
GETDUR	LSR	A	
	TAX		
	LDA	TEMPO	
	BEQ	GTFAST	
	CMP	#1	
	BNE	GTSLOW	
	LDA	NDURSM,X	
	RTS		

GTSLOW	LDA	NDURS, X
	RTS	
GTFAST	LDA	NDURSF, X
	RTS	

1 File(s) copied

C>

12
11
10
9
8
7
6
5
4
3