



Synertek
Systems Corporation

**TECHNICAL
NOTES**

Leland Goertz

TECHNICAL NOTES

Copyright © by Synertek Systems Corporation

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written consent of Synertek Systems Corporation.

SSC Pub NP-1

First Printing: April, 1979



Synertek Systems Corporation

P.O. BOX 552 SANTA CLARA, CALIFORNIA 95052 TEL. (408) 988-5600 TWX: 910-338-0135

TABLE OF CONTENTS

TN 22	SYM-1 Input/Output Pins Utilization.....	1
TN 35	SYM-1 Updates for "VIM" Reference Manual.....	5
TN 49	SYM-1 Sample Programs.....	10
TN 50	SYM-1 Display Routine.....	17
TN 52	SYM-1 Time Delay Using 6532 Timer.....	20
TN 53	Trigonometric Functions for Synertek BASIC.....	27
TN 54	SYM-1 Power-Up to User ROM.....	33
TN 59	SYM-1 Modification for Inputting Lower Case Characters.....	36
TN 72	Cassette Data Reading Using SYM-1 High Speed Format.....	38



SynerTek

INCORPORATED

3001 STENDER WAY • SANTA CLARA, CA 95051
TELEPHONE (408) 988-5600 • TWX: 910-338-0135

TECHNICAL
NOTE

#22

JUNE 1978

SYM-I INPUT/OUTPUT PINS UTILIZATION

The SYM-I Microcomputer Systems has a large number of general purpose I/O lines available. In addition, it is possible to expand the number of lines by a variety of ways. The purpose of this note is to explain how this can be achieved.

OVERALL I/O STRUCTURE

There are three interface devices in the basic SYM-I system: one SY-6532 and two SY6522s. There is also an expansion socket for a third SY6522. Each interface device has 16 I/O lines. In addition, the SY-6522s have 4 control pins. The reader should consult the data sheets for these parts for a complete understanding of the operation of the I/O pins. Figure 1 summarizes how the pins for each device are utilized. In the basic SYM-I system (6522 #2 not installed) there are 25 general-purpose I/O lines and 2 CONTROL lines:

- 1) 6522 #1 has 15 I/O and no CONTROL.
- 2) 6522 #3 has 10 I/O (this includes 4 I/O available with or without BUFFERS) and 2 CONTROLS.

The four BUFFERED I/O pins deserve some comments. The circuit for each is shown on page 4-11 in the SYM Reference Manual. Connections for wiring options permit the BUFFERS to be utilized in a variety of ways, such as relay drivers, level converters, D/A Converters, and one-shots. Furthermore, it is possible to by-pass the buffers entirely, thus allowing the 4 lines to be used as general-purpose I/O lines.

The following sections describe how to expand the number of I/O lines further. Figure 2 aids in this description.

PIN DESIGNATION	PERIPHERAL INTERFACE DEVICE			
	6532 (U27)	6522#1 (U25)	OPTIONAL 6522#2 (U28)	6522#3 (U29)
PA0				
PA1				
PA2				
PA3				
PA4	&			
PA5	DISPLAY			
PA6				
PA7				
PB0				
PB1				
PB2				
PB3				
PB4				
PB5	CRT & TTY			
PB6		AUDIO		
PB7	INTERFACE	KIM I/O		
CA1		NOT USED		I/O CONTROL
CA2	NOT AVAILABLE	POWER-ON-RESET		SCOPE
CB1		NOT USED		I/O CONTROL
CB2		SPECIAL FUNC.		SCOPE

Figure 1 - SYM-1 I/O SUMMARY

SYSTEM CONFIGURATION	PERIPHERAL INTERFACE DEVICE				TOTAL
	6532	6522#1	6522#2	6522#3	
BASIC SYSTEM	0/0	15/0	---	10/2	25/2
WITH EXPANSION 6522	0/0	15/0	16/4	10/2	41/6
WITHOUT DEBUG, WP, FEATURE	0/0	15/0	16/4	16/2	47/6
WITHOUT KYPD, DISPLAY	11/0	15/0	16/4	16/2	58/6

NOTE: X/Y, where X is no. of I/O and Y is no. of CONTROL pins.

FIGURE 2 - Number of Pins Versus System Configuration

6522#2

By installing another SY6522 device into socket U28, and additional 16 I/O and 4 control lines are immediately made available on the Auxiliary Application Connector.

WRITE-PROTECT FEATURES

6522#3 used pins PA0 through PA3 for WRITE-PROTECT features, as follows (see Figure 3 for schematic):

- 1) Each pin has a place for a wire jumper to permit using that I/O pin to control the gating of R/W.
- 2) The gating is as follows:
 - PA0 controls writing into the 6532 RAM.
 - PA1 controls writing into address 400-7FF (HEX).
 - PA2 controls writing into address 800-BFF (HEX).
 - PA3 controls writing into address C00-FFF (HEX).
- 3) Software to perform the WRITE-PROTECT function is a part of the SUPERMON system, activated with the "WP" key. In addition, the user can achieve the same thing by programming the 6522 I/O bits. Note that the jumper must be in place and the bit must a logic "0" output to WRITE-PROTECT the memory block.

Thus, if WRITE-PROTECT is not needed, the jumpers should not be installed and PA0-PA3 may be used as general-purpose I/O. The jumpers are located near the crystal on the board and can easily be traced to the 6522#3.

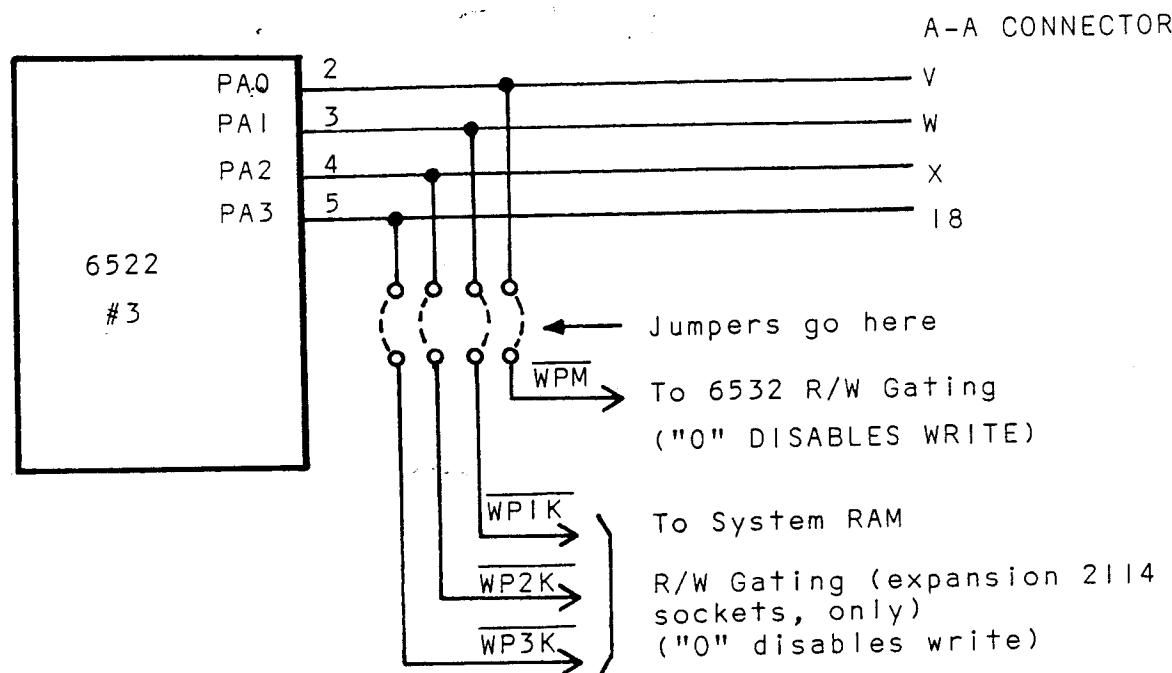


Figure 3 - WRITE PROTECT WIRING

DEBUG FEATURES

PA4 and PA5 on 6522#3 are used for DEBUG Control to permit program trace operation. These features are also selectable via wire jumpers. In this case, the jumper positions are located directly across the audible response speaker from the WRITE-PROTECT jumpers. With the jumpers in, the trace features can be used; with them out, the pins are general-purpose I/O.

KEYPAD AND DISPLAY

There are 12 lines on the 6532 used for keypad and HEX Display functions. Eleven of these are also routed to the Application Connector. Thus, if the Keypad and Display are not used, then these 11 pins can be utilized as general-purpose I/O. The schematic of this area of the board is found on page 4-13 of the SYM Reference Manual. Some applications not requiring the Keypad and Display are:

Applications using the RS-232 port for a CRT terminal.

Dedicated controllers using user-developed software (on ROM/PROM).

SUMMARY

The above-described I/O expansion schemes permit up to 58 general-purpose I/O and 6 Control pins, depending on the actual requirements of the system. It should be understood that all I/O expansion (except for the extra 652) requires sacrificing some SYM-1 feature, but under the proper circumstances, the trade-offs may be of great value.



Synertek
INCORPORATED

3001 STENDER WAY • SANTA CLARA, CA 95051
TELEPHONE (408) 988-5600 • TWX: 910-338-0135

TECHNICAL
NOTE

35

September '78

SYM-1 UPDATE'S FOR VIM REFERENCE MANUAL

(MAY, 1978 EDITION)

1. Remote Control Connections

See the instructions in Section 3-8 of the SYM Reference Manual, and use the corrected Figure 3-3 and Table 3-1, attached.

2. Adjusting Your Recorder

The audio signal appears on the T and A connectors in two forms: Aud Out (HI) and Aud Out (LO). The only difference between these signals is their magnitude. For most recorders, the best arrangement is to run Aud Out (LO) into the MIC input of the recorder. Some recorders also have an AUX input, which bypasses the MIC pre-amp, and may work better if Aud Out (HI) is wired into AUX.

Read Appendix F, and follow the procedure for creating a 'SYNC' tape. Rewind the tape and enter the LD command appropriate to the SYNC tape you created. Adjust the tone and volume controls, observing the S on the display. Leave the controls in the middle of the range where the S remains off. (If there are two ranges of volume which cause the S to turn off, the higher range should be used. If a sharp tap causes the S to relight and remain lit, you are in the wrong range.)

If your recorder has an automatic-recording-level defeat switch, it will probably work better in the engaged position.

Now write a short record to tape and read it back to verify correct operation. (Do not use the memory form \$F8 to \$FF, or the stack area (page 1), as these are used by the cassette software.)

Recommended Tape Equipment

Most moderate quality tape recorders should produce satisfactory results. (A tone control is recommended.) The following models have been used successfully at Synertek Systems:

Sanyo M2533A
Sony TC-205
Sony TC-62

GE IC #3-5002B
Superscope C-190
Realistic Ctr-40

Almost any tape will suffice, so long as it winds smoothly (does not produce a jittery tape motion). A very short tape will be more convenient. The following tapes have been used successfully at Synertek Systems:

TDK
AMPEX
MALLORY
REALISTIC

READING A (center tip voltage)			
	-6v to -8v	GND	+6v to +8v
READING B (shield voltage)	-6v to -8v	READING C GND Type VIII -8v Type V	
	GND	READING C GND Type VII -8v Type VI	READING C GND Type I +8v Type IV
	+6v to +8v	READING C GND Type II +8v Type III	

Reading C (shorted)

Table 3-1 Audio Cassette remote control type determination

SYM-1 Monitor Addenda

1. While tracing or single stepping, SUPERMON uses G01ENT (\$83FA) to return to the user program. G01ENT write protects System RAM. If you must trace a program that needs access to System RAM, use a user trace routine and go to G01ENT +3, or remove jumper MM-45 (enables System RAM protect).
2. The DEBUG-ON switch bounces, therefore it should not be used to interrupt user programs while using a user trace routine or while OUTVEC points to a user routine. (This will cause recursive interrupts.)
3. The audio cassette software will not read or write location \$FFFF. Use \$A67F (\$A600 through \$A67F is echoed at \$FF80 through \$FFFF.)

APPLICATION NOTE - Changing Automatic Logon 6/30/78

After power is applied to the SYM, SUPERMON waits for the keyboard or the device connected to PB7 on the 6532 (normally the RS232 device) to become active. PB6 (the current loop device) is ignored because a disconnected current loop always looks active.

If you expect always to logon to a current-loop device, the following jumper change will eliminate the necessity of entering (SHIFT) (JUMP) (1):

Change CC-32 and BB-31 to CC-31 and BB-32

Now the logon for your current loop device is simply a 'Q,' entered at the device. (Note that you cannot now logon automatically to the keyboard unless the current loop device is connected, and powered-up.)

SYM-1 REFERENCE MANUAL - ERRATA

Page 3-11, Figure 3-5

Ignore everything left of 'T' and 'A' connectors.

Page 4-18, Figure 4-10

E000-F7FF unused
F800-FFFF echo locations

Page E-3, Table E-3

INCHAR = 8A1B, Read = 1C6A

Page 4-7, Figure 4-2

See corrected pages, attached.

Page 3-7, Figure 3-3

See corrected page, attached.

Types I-IV, printed B(T-18) now reads B(T-16).

Types II and VI; inner/outer cable connections were reversed.

TABLE 4-2. (continued)POWER (P)

1	+5V	A	+VP (optional)
2	GND	B	GND
3	+5V	C	+5V
4	GND	D	GND
5	+5V	E	-VN (optional)
6	GND	F	GND

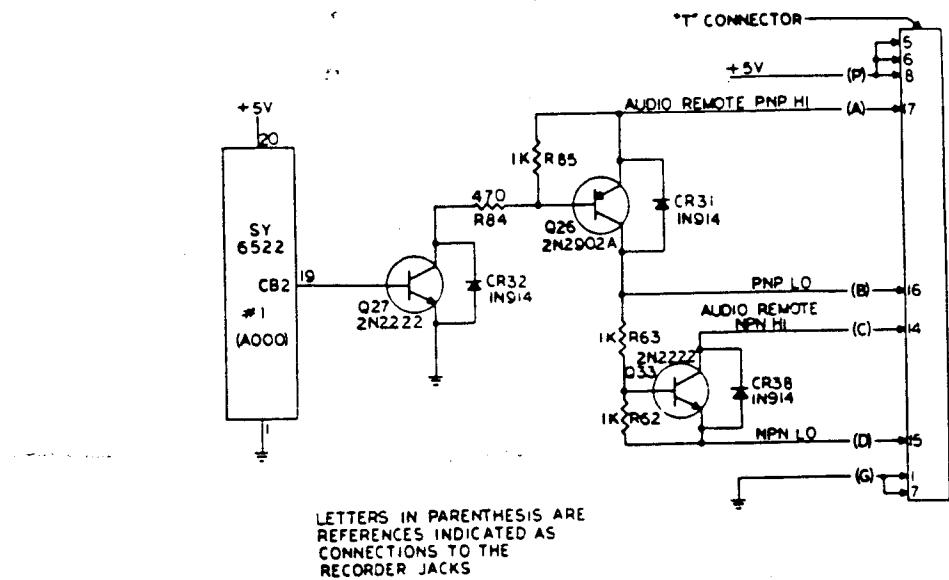
TERMINAL (T)

1	GND	13	N.C.
2	RS-232 IN	14	Audio Remote NPN HI
3	RS-232 OUT	15	Audio Remote NPN LO
4	N.C.	16	Audio Remote PNP LO
5	+5V	17	Audio Remote PNP HI
6	+5V	18	Audio IN
7	GND	19	Audio GND
8	+5V	20	N.C.
9	TTY Keyboard IN +	21	Audio Out (HI)
10	TTY Keyboard IN -	22	N.C.
11	TTY Printer OUT -	23	Audio Out (LO)
12	TTY Printer Out +	24	N.C.
		25	Audio GND

KEYBOARD (K)

1	+5V	8	-VN
2	+5V	9	GND
3	+5V	10	GND
4	+5V	11	GND
5	+VP	12	GND
6	+VP	13	RS-232 IN
7	-VN	14	RS-232 OUT

AUDIO CASSETTE SYM REMOTE CONTROL CONNECTION



AUDIO CASSETTE RECORDER JACKS REMOTE CONTROL CONNECTIONS

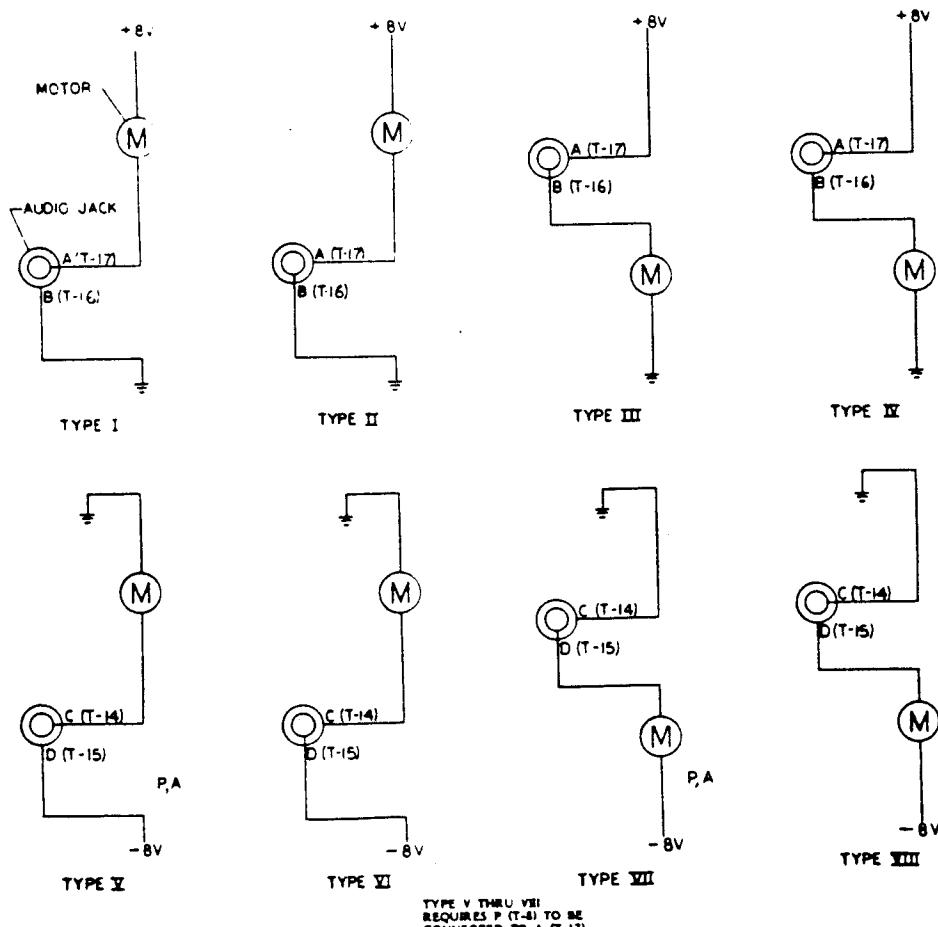


Figure 3-3. REMOTE CONTROL TYPES AND CONNECTIONS



Syntek
INCORPORATED

3001 STENDER WAY • SANTA CLARA, CA 95051
TELEPHONE (408) 988-5600 • TWX: 910-338-0135

TECHNICAL
NOTE

#49

DEC'78

SYM-1 SAMPLE PROGRAMS

PROGRAM #1 - BEEPER DEMO

This program demonstrates the use of the piezo-electric "beeper" on the SYM-1 board to generate programmable tones.

.....PAGE 0001

LINE #	LOC	CODE	LINE	CLASS2 DEMONSTRATION PROGRAM #2 THE "BEEPER"
0002	0000		;	;SUBRTN TO SET UP FOR BEEPER
0003	0000		;	;BEEPER DATA REGISTER
0004	0000		;	;DATA DIRECTION FOR BEEPER
0005	0000	CONFIG = \$89A5	;	;TONE STORAGE
0006	0000	FBDA = \$A402	;	;HOW LONG FOR EACH NOTE
0007	0000	FBDDR = \$A403	;	;PLACE FOR INTERRUPT EVECTOR
0008	0000	TONE = \$10	;	;REG TO ENABLE INTERRUPT FLAG
0009	0000	LENGTH = \$11	;	;INTERRUPT FLAG REGISTER
0010	0000	IRQVEC = \$A67E	;	;TIMER INTERRUPT SET UP REG.
0011	0000	IER1 = \$AC0E	;	;ADDRESS OF LOW TIMER
0012	0000	IFR1 = \$AC0D	;	;ADDRESS OF HIGH TIMER
0013	0000	ACR1 = \$AC0B	;	;RAM EENABLE SUBRTN
0014	0000	T1LL1 = \$AC06	;	;
0015	0000	T1CH1 = \$AC05	;	;
0016	0000	ACCESS = \$8E86	;	;
0017	0000		*	PROGRAM STARTING LOCATION
0018	0000		= \$0250	
0019	0250		;	;
0020	0250	78	START SEI	DISABLE ALL INTERRUPTS
0021	0251		;	;
0022	0251	20 86 8B	JSR ACCESS	ENABLE SYSTEM RAM
0023	0254		;	;
0024	0254	A9 50	LDA #>INTRPT	SET UP INTERRUPT ADDRESS
0025	0256	8D 7E A6	STA IRQVEC	AT THE LOCATION IN
0026	0259	A9 03	LDA #>INTRPT	MEMORY TO CAUSE THE INTERRUPT
0027	025B	8D 7F A6	STA IRQVEC+1	TO BRANCH TO OUR INTERRUPT
0028	025E		;	SUBROUTINE
0029	025E		;	;
0030	025E	A9 0F	LDA #\$0F	SET UP DATA DIRECTION REG.
0031	0260	8D 03 A4	STA FBDDR	;
0032	0263		;	;
0033	0263	A9 FF	LDA #\$FF	INITIALIZE THE TONE
0034	0265	85 10	STA TONE	;
0035	0267		;	;
0036	0267	A9 01	LDA #\$01	INITIAL THE TONE LENGTH
0037	0269	85 11	STA LENGTH	;
0038	026B		;	;
0039	026B	A9 40	LDA #\$40	SET UP INTERRUPT TIMER
0040	026D	8D 0B AC	STA ACR1	WITHOUT HAVING SQ. WAVES
0041	0270		;	;
0042	0270	A9 C0	LDA #\$C0	ENABLE THE INTERRUPT FLAG
0043	0272	8D 0E AC	STA IER1	;
0044	0275		;	;
0045	0275	A9 40	LDA #\$40	CLEAR ANY PENDING FLAGS
0046	0277	8D 0D AC	STA IFR1	IF THEY ARE THERE
0047	027A		;	;
0048	027A	A9 20	LDA #\$20	START THE INTERRUPT TIMERS
0049	027C	8D 06 AC	STA T1LL1	;
0050	027F	8D 05 AC	STA T1CH1	;
0051	0282		;	;
0052	0282	A9 0D	BEEPER LDA #\$0D	CONFIGURE FOR THE BEEPER
0053	0284	20 A5 89	JSR CONFIG	;
0054	0287		;	;
0055	0287		CLI	NOW ENABLE THE INTERRUPTS
0056	0288		;	START THE 'MUSIC'.

.....PAGE 0002

LINE #	LOC	CODE	LINE	
0057	0288	A9 08	BE1	LDA #\$08 ;TURN BEEPER ON
0058	028A	8D 02 A4		STA PBDA
0059	028D	20 9A 02		JSR BE2 ;WAIT AWHILE
0060	0290	A9 06		LDA #\$06 ;TURN BEEPER OFF
0061	0292	8D 02 A4		STA PBDA
0062	0295	20 9A 02		JSR BE2 ;NOW WAIT THE SAME WHILE.
0063--	0298	90 EE		BCC BE1 ;NOW DO IT ALL OVER AGAIN.
0064	029A		;	
0065	029A	A4 10	BE2	LDY TONE ;DELAY SUBROUTINE
0066	029C	88	BE3	DEY ;JUST COUNT DOWN
0067	029D	D0 FD		BNE BE3 ;DO IT AGAIN IN NOT ZERO.
0068	029F		;	
0069	029F	60		RTS ;RETURN FROM SUBROUTINE
0070	02A0		;	
0071	02A0			* = \$0350
0072	0350		;	
0073	0350	48	INTRPT	PHA ;SAVE THE ACCUMULATOR
0074	0351	98		TYA ;TRANSFER Y REG TO ACC.
0075	0352	48		PHA ;SAVE THAT TOO.
0076	0353		;	
0077	0353	A9 40		LDA #\$40 ;CLEAR THE PENDING INTERRUPT FLAG
0078	0355	8D 0D AC		STA IFR1
0079	0358		;	
0080	0358	C6 11		DEC LENGTH ;COUNT DOWN EACH NOTE LENGTH
0081	035A	D0 10		BNE RETURN ;IF NOT ZERO GO BACK.
0082	035C		;	
0083	035C	A9 01		LDA #\$01 ;RESTORE LENGTH REG.
0084	035E	85 11		STA LENGTH
0085	0360		;	
0086	0360	C6 10		DEC TONE ;MAKE A HIGHER NOTE.
0087	0362	A5 10		LDA TONE ;IS IT HIGH ENOUGH?
0088	0364	C9 10		CMP #\$10 ;10 SHOULD BE A LIMIT.
0089	0366	B0 04		BCS RETURN ;IF NOT 10, GO BACK.
0090	0368		;	
0091	0368	A9 FF		LDA #\$FF ;RESTORE TO LOWEST NOTE
0092	036A	85 10		STA TONE
0093	036C		;	
0094	036C	68	RETURN	PLA ;REPLACE EVERYTHING.
0095	036D	98		TYA ;Y REGISTER IS BACK.
0096	036E	68		PLA ;ACCUMULATOR IS BACK.
0097	036F		;	
0098	036F	40		RTI ;RETURN FROM INTERRUPT.
0099	0370		;	
0100	0370			.END

ERRORS = 0000 <0000>
END OF ASSEMBLY

PROGRAM #2 - ROTATING DISPLAY

This program demonstrates the use of the 6-digit HEX display to show a rotating message.

.....PAGE 0001

LINE #	LOC	CODE	LINE
0002	0000		;
0003	0000		;
0004	0000		FILE = \$0300
0005	0000		DISBUF = \$A640
0006	0000		SCANDS = \$8906
0007	0000		COUNT = \$11
0008	0000		ACCESS = \$8E86
0009	0000		;
0010	0000		;
0011	0000		;
0012	0000		00 = \$3F
0013	0000		01 = \$06
0014	0000		02 = \$5B
0015	0000		03 = \$4F
0016	0000		04 = \$66
0017	0000		05 = \$6D
0018	0000		07 = \$07
0019	0000		08 = \$7F
0020	0000		09 = \$67
0021	0000		OELANK = \$00
0022	0000		ODOT = \$80
0023	0000		OA = \$77
0024	0000		OB = \$7C
0025	0000		OC = \$39
0026	0000		OD = \$5E
0027	0000		OE = \$79
0028	0000		OF = \$71
0029	0000		OG = \$6F
0030	0000		OH = \$76
0031	0000		OI = \$06
0032	0000		OJ = \$1E
0033	0000		OK = \$74
0034	0000		OL = \$38
0035	0000		OM1 = \$33
0036	0000		OM2 = \$27
0037	0000		ON = \$54
0038	0000		OO = \$3F
0039	0000		OF = \$73
0040	0000		OQ = \$67
0041	0000		OR = \$50
0042	0000		OS = \$6D
0043	0000		OT = \$46
0044	0000		OU = \$3E
0045	0000		OV1 = \$64
0046	0000		OV2 = \$52
0047	0000		OW1 = \$3C
0048	0000		OW2 = \$1E
0049	0000		OX = \$00
0050	0000		OY = \$6E
0051	0000		OZ = \$00
0052	0000		;
0053	0000		* = \$300
0054	0300		;
0055	0300		.OPT GEN
0056	0300		;

.....PAGE 0.002

LINE #	LOC	CODE	LINE	
0057	0300		;	
0058	0300		;	
0059	0300		;	
0060	0300		;	
0061	0300		;	
0062	0300		;	
0063	0300		;	
0064	0300		;	
0065	0300	33	.BYTE OM1,OM2,0I,OK,OE,0BLANK,0BLANK	ALTHOUGH SCANDS SUBROUTINE USES ONLY THE LOCATIONS A640 THROUGH A645, WE ARE GOING TO FILE A646 WITH A BLANK SO WE CAN SHIFT A BLANK THROUGH TO SEPARATE OUR 6 CHARACTER WORD.
0065	0301	27		
0065	0302	06		
0065	0303	74		
0065	0304	79		
0065	0305	00		
0065	0306	00		
0066	0307		;	
0067	0307		* = \$200	
0068	0200		;	
0069	0200	20 86 8B	JSR ACCESS	;WRITE ENABLE SYSTEM RAM
0070	0203		;	
0071	0203	A0 06	LDY #\$06	;TAKE THE CHARACTERS FROM
0072	0205	B9 00 03	ONE LDA FILE,Y	;THE FILE WE HAVE ARBITRARILY
0073	0208	99 40 A6	STA DISBUF,Y	ESTABLISHED AND FILL
0074	0208	88	DEY	;THE DISPLAY BUFFER WITH THEM.
0075	020C	10 F7	BPL ONE	
0076	020E		;	
0077	020E	A9 FF	CYCLE LDA #\$FF	;SET THE NUMBER OF TIMES WE
0078	0210	85 11	STA COUNT	;FLASH THE LEDS WITH THE CONTENTS OF DISBUF.
0079	0212		;	
0080	0212	20 06 89	TWO JSR SCANDS	;NOW FLASH THE DISPLAYS
0081	0215	C6 11	DEC COUNT	;DEC THE COUNT FOR EACH FLASH
0082	0217	D0 F9	BNE TWO	;ARE WE DONE YET?
0083	0219		;	IF NOT, FLASH AGAIN
0084	0219	AD 40 A6	LDA DISBUF	;IF SO, THEN SAVE THE TOP NUMBER
0085	021C	48	PHA	
0086	021D	A0 00	LDY #\$00	;SHIFT UP THE REMAINING 6 REGISTERS
0087	021F		;	
0088	021F	B9 41 A6	THREE LDA DISBUF+1,Y	;BY LOOPING AND INCREMENTING
0089	0222	99 40 A6	STA DISBUF,Y	;THE Y REGISTER ON EACH LOOP.
0090	0225	CB	INY	
0091	0226	C0 06	CPY #\$06	;AND THEN CHECKING TO SEE
0092	0228	D0 F5	BNE THREE	;IF WE HAVE INCREMENTED THE RIGHT NUMBER OF TIMES.
0093	022A		;	
0094	022A	68	PLA	;NOW PLACE THE FIRST REGISTER IN
0095	022B	80 46 A6	STA DISBUF+6	;THE LAST MEMORY LOCATION
0096	022E		;	
0097	022E	4C 0E 02	JMP CYCLE	;NOW DO IT ALL OVER AGAIN.
0098	0231		;	
0099	0231		.END	

ERRORS = 0000 <0000>
END OF ASSEMBLY

PROGRAM # 3 - COUNT AND BEEP

This program uses the display to count from 00 to FF and then energizes the audio tone generator while displaying the word "beep".

.....PAGE 0001

LINE #	LOC	CODE	LINE
0002	0000		;DEMO PROGRAM WRITTEN BY D. SATTERFIELD
0003	0000		;11/27/78
0004	0000		;
0005	0000		BP = \$8972
0006	0000		SEGTAB = \$8C29
0007	0000		IRQVEC = \$A67E
0008	0000		FLAG = \$40
0009	0000		FLAG1 = \$42
0010	0000		T1LL = \$AC06
0011	0000		T1CH = \$AC05
0012	0000		NUM = \$41
0013	0000		SCAND = \$8906
0014	0000		D1 = \$A645
0015	0000		D2 = \$A644
0016	0000		D3 = \$A643
0017	0000		D4 = \$A642
0018	0000		D5 = \$A641
0019	0000		D6 = \$A640
0020	0000		ACCESS = \$8B86
0021	0000		ACR1 = \$AC08
0022	0000		IER1 = \$AC0E
0023	0000		IFR1 = \$AC0D
0024	0000		;
0025	0000		;INITIALIZE
0026	0000		;
0027	0000		x=\$300
0028	0300	20 86 88	JSR ACCESS ;UNWRITE PROTECT SYSTEM RAM
0029	0303	A9 20	LDA #\$20
0030	0305	B5 42	STA FLAG1
0031	0307	A9 55	LDA * <INTCNT
0032	0309	8D 7E A6	STA IRQVEC
0033	030C	A9 03	LDA * >INTCNT
0034	030E	8D 7F A6	STA IRQVEC+1
0035	0311	A9 40	LDA #\$40
0036	0313	8D 08 AC	STA ACR1 ;ENABLE T1 CONTINUOUS INTERRUPTS
0037	0316	A9 4E	LDA #\$4E ;LO BYTE OF T1 COUNTER
0038	0318	8D 06 AC	STA T1LL
0039	031E	A9 C0	LDA #\$C0 ;ENABLE T1
0040	031D	8D 0E AC	STA IER1 ;INTERRUPTS
0041	0320	A9 00	LDA #\$00
0042	0322	B5 40	STA FLAG ;ZERO FLAG
0043	0324	A9 20	LDA #\$20 ;T1 HI BYTE
0044	0326	8D 05 AC	STA T1CH ;START COUNTING
0045	0329	18	CLC ;CLEAR CARRY
0046	032A	58	CLI ;ENABLE INTERRUPTS
0047	032B		;
0048	032B		;DISPLAY ROUTINE
0049	032B		;
0050	032B	A9 00	DISFL LDA #\$00
0051	032D	8D 40 A6	STA D6
0052	0330	8D 41 A6	STA D5
0053	0333	8D 44 A6	STA D2
0054	0336	8D 45 A6	STA D1 ;BLANK ALL DISPLAYS EXCEPT D3 AND D4
0055	0339	A5 41	DISP LDA NUM
0056	033B	29 0F	AND #\$0F ;STRIP UPPER BITS

.....PAGE 0002

LINE #	LOC	CODE	LINE	
0057	033D	20 73 03	JSR CONV	; CONVERT TO 7 SEGMENT CODE
0058	0340	8D 43 A6	STA D3; STORE IT	
0059	0343	A5 41	DISP2 LDA NUM	
0060	0345	4A	LSR A	
0061	0346	4A	LSR A	
0062	0347	4A	LSR A	
0063	0348	4A	LSR A	; SHIFT RIGHT 4 TIMES
0064	0349	20 73 03	JSR CONV	; CONVERT IT
0065	034C	8D 42 A6	STA D4	; STORE IT
0066	034F	20 06 89	JSR SCAND	
0067	0352	4C 2B 03	JMP DISPL	
0068	0355	48	INTCNT FHA	; SAVE ALL REGISTERS
0069	0356	8A	TXA	
0070	0357	48	FHA	; SAVE X
0071	0358	98	TYA	
0072	0359	48	FHA	; SAVE Y
0073	035A	AD 00 AC	LDA IFR1	
0074	035D	8D 00 AC	STA IFR1	
0075	0360	E6 40	INC FLAG	; INCREMENT NUMBER OF INTERRUPTS COUNT
0076	0362	A5 40	LDA FLAG	
0077	0364	C9 05	CMP #\$5	; 5 INTERRUPTS YET?
0078	0366	F0 02	BEQ ADD	; YES,INCREMENT DISPLAY
0079	0368	50 66	BVC REST	; NO,GO BACK AND WAIT
0080	036A	A9 00	ADD LDA #\$00	
0081	036C	85 40	STA FLAG	
0082	036E	20 78 03	JSR COUNT	
0083	0371	50 50	BVC REST	
0084	0373	AA	CONV TAX	
0085	0374	8D 29 8C	LDA SEGTAB,X	
0086	0377	60	RTS	
0087	0378	18	COUNT CLC	; CLEAR CARRY
0088	0379	A5 41	LDA NUM	
0089	037B	69 01	ADC #\$01	
0090	037D	18	CLC	
0091	037E	B8	CLV	
0092	037F	85 41	STA NUM	
0093	0381	C9 FF	CMP #\$FF	
0094	0383	F0 01	BEQ BEEP	
0095	0385	60	RTS	
0096	0386	A9 7C	BEEP LDA #\$7C	; 7 SEG CODE FOR B
0097	0388	8D 41 A6	STA D5	
0098	038B	A9 79	LDA #\$79	; E
0099	038D	8D 42 A6	STA D4	
0100	0390	8D 43 A6	STA D3	
0101	0393	A9 73	LDA #\$73	; CODE FOR P
0102	0395	8D 44 A6	STA D2	
0103	0398	A9 00	LDA #\$00	
0104	039A	85 41	STA NUM	
0105	039C	20 72 89	DELAY JSR BP	
0106	039F	20 06 89	JSR SCAND	
0107	03A2	20 06 89	JSR SCAND	
0108	03A5	20 06 89	JSR SCAND	
0109	03A8	20 06 89	JSR SCAND	
0110	03AB	20 06 89	JSR SCAND	
0111	03AE	20 06 89	JSR SCAND	

.....PAGE 0003

LINE #	LOC	CODE	LINE
0112	03B1	C6 42	DEC FLAG1
0113	03B3	20 06 89	JSR SCAND
0114	03B6	20 06 89	JSR SCAND
0115	03B9	20 06 89	JSR SCAND
0116	03BC	20 06 89	JSR SCAND
0117	03BF	20 06 89	JSR SCAND
0118	03C2	20 06 89	JSR SCAND
0119	03C5	A5 42	LDA FLAG1
0120	03C7	C9 00	CMF #\$0
0121	03C9	D0 D1	BNE DELAY
0122	03CB	A9 20	LDA #\$20
0123	03CD	85 42	STA FLAG1
0124	03CF	60	RTS
0125	03D0	68	REST
0126	03D1	A8	PLA
0127	03D2	68	TAY
0128	03D3	AA	PLA
0129	03D4	68	TAX
0130	03D5	40	PLA
0131	03D6		RTI
			.END

ERRORS = 0000 <0000>
END OF ASSEMBLY



SynerTek SYSTEMS

3050 CORONADO DRIVE • SANTA CLARA, CA 95051
TELEPHONE (408) 984-8900 • TWX: 910-338-0135

(408) 988-5689

TECHNICAL
NOTE

No. 50

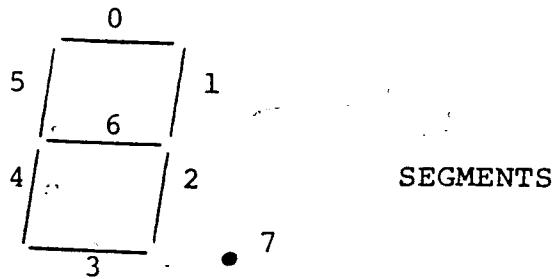
NOVEMBER 1978

SYM-1 DISPLAY ROUTINE

This routine allows the user to display his own message on the SYM-1 6-digit display.

To use the routine, the user must first load memory locations 0250 through 0255 with segment codes from Figure 1, using the M or D commands of SUPERMON. Note location 0255 will contain the segment code for the right-hand digit. Next the user must enter and execute the code shown in Figure 2. To stop the display and return to the monitor, press the RST and CR Keys.

This routine may be structured as a sub-routine and incorporated in user programs to display various values and results.



Byte in memory: b₇ b₆ b₅ b₄ b₃ b₂ b₁ b₀

b₀ = 1 lights segment 0

b₁ = 1 lights segment 1

etc.

Figure 1a.— Segment pattern

<u>CHARACTER</u>	<u>SEGMENT CODE</u>	<u>CHARACTER</u>	<u>SEGMENT CODE</u>
A	77	P	73
b	7C	r	50
C	39	S	6D
c	58	U	3E
d	5E	Y	6E
E	79	Z	5B
e	7B	0	3F
F	71	1	06
G	7D	2	5B
H	76	3	4F
h	74	4	66
I	06	5	6D
J	1E	6	7C
L	38	7	07
n	54	8	7F
O	3F	9	67
o	5C		

Figure 1 b. Segment Codes



**Syntek Systems
Corporation.**

Figure 2. Display Routine

ADR	B1	B2	B3	LABEL	NMemonic	OPERAND	COMMENTS
200	20	BC	8B		JSR	ACCESS	Un-write-protect system RAM
203	A2	05			LDX	# \$05	Transfer characters from temporary
205	BD	50	02	L1	LDA	\$250,X	locations at 250 to 255
208	9D	40	AC		STA	DISBUF,X	to DISBUF
20B	CA				DEX		
20C	10	F1			BPL	L1	
20E	20	06	B9	L2	JSR	SCAND	Light display (single scan)
211	4C	0E	02		JMP	L2	Repeat for continuous display



Synertek SYSTEMS CORPORATION

3050 CORONADO DRIVE • SANTA CLARA, CA 95051
TELEPHONE (408) 984-8900 • TWX: 910-338-0135

TECHNICAL
NOTE

NO. 52-SSC

January, 1979

SYM-1 TIME DELAY USING 6532 TIMER

The internal interval timer in device U27, SY6532, provides a method for generating time delays from one microsecond to 262,144 microseconds.

To use the timer, the user must load the desired count into the 8-bit counter and set the prescaler to the desired mode. These two operations are accomplished simultaneously by control of the address lines, Figure 1.

Since the IRQ line from U27 is not connected to the CPU chip, it is necessary to test bit 7 in the Interrupt Flag Register (IFR) to determine when timeout has occurred. The count remaining in the timer may be read at any time without affecting the count. Reading or writing to the timer after the interrupt is set will clear the interrupt.

An example of a subroutine to generate a one millisecond delay is given in Figure 2. Note that the total delay is counted from the JSR in the calling program to the return to the calling program as shown in the Timing Analysis, Figure 3, and Timing Diagram, Figure 4.

<u>ADDRESS</u>	<u>FUNCTION</u>
A404	Read Timer
A405	Read Interrupt flags
A41C	Load Timer; set prescaler to $\div 1$
A41D	Load Timer; set prescaler to $\div 8$
A41E	Load Timer; set prescaler to $\div 64$
A41F	Load Timer; set prescaler to $\div 1024$

Figure 1. Address table for 6532 timer, device U27.

Another example, a 60 second timer, is given in Figure 5. This routine shows how the delay routine might be used in a larger program. The delay routine, Figure 6, is nearly the same as that of Figure 2 except the timing has been adjusted to 993 microseconds to compensate for overhead time lost in the calling program.



PROGRAMMING SHEET

Program Title

1 ms delay : - Figure 2

Address	B1	B2	B3	Label	Nemonic	Operand	Comments
Instructions							
0300	48			*	PHA		SAVE Accumulator
0301	A9	79			LDA	# \$ 79	Load Accumulator for Count of 121 ₁₀
0303	8D	ID	A4		STA	\$ A41D	Store Count; set prescaler to ÷ 8
0306	2C	05	A4	D1	BIT	\$ A405	Test bit 1 of IFR
0309	10	F8			BPL	D1	Loop if not set
030B	25	FF			AND	\$ FF	Waste three cycles
030D	68				PLA		Restore Accumulator
030E	60				RTS		Return

JSR DELAY	6 cycles
PHA	3
LDA	2
STA	4

loop ... 138 times

BIT 4 cycles

*BPL 3 cycles

$$\underline{7 \text{ cycles} \times 138 = 996}$$

last loop

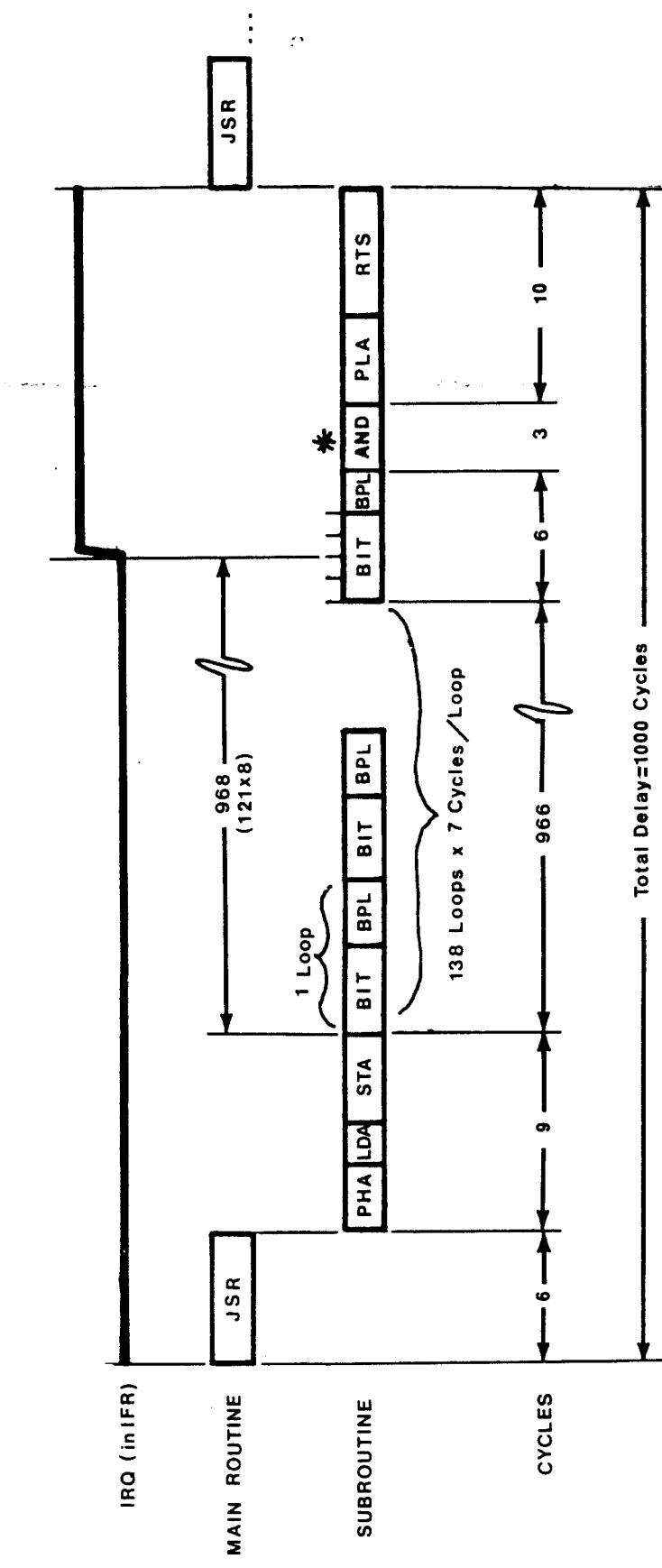
BIT	4
*BPL	2
AND	3
PLA	4
RTS	6

$$\underline{1000 \text{ cycles}}$$

$$1000 \text{ cycles} @ 1 \text{ microsecond/cycle} = 1 \text{ ms}$$

*NOTE: BPL takes 3 cycles to branch back to beginning of loop but only 2 cycles when the branch is not taken.

Figure 3. 1 ms Delay Timing Analysis



* Dummy operation to waste three cycles

FIG. 4 Timing Diagram for 1 MS Delay



PROGRAMMING SHEET

Program Title **FIGURE 5.** **CLOCK {**uses 993 μs **delay }**
60 SEC.

Programmer

Address	Instructions	B1	B2	B3	Label	Mnemonic	Operand	Comments
0200	A9 00					LDA	#00	* = \$200 Initialize counter
0202	85 00					STA	00	in page zero
0204	85 01					STA	01	
0206	20 72 89					JSR	BEEP	Sound beeper
0209	A9 EA					LOA	#\$EA	Load A for 234,0 passes
020B	20 00 03	L1				JSR	DELAY	Loop 1 : Call delay (993 μs)
020E	EE 00					INC	\$00	Subroutine 256 times
0210	Do F9					BNE	L1	
0212	EE 01					INC	\$01	REPEAT Loop 1
0214	C5 01					CMP	\$01	234,0 times (as set by
0216	Do F3					BNE	L1	value in A)
0218	A9 22					LOA	#\$22	Load A for 34,0 passes
021A	85 00					STA	\$00	thru loop 2
021C	20 00 03	L2				JSR	DELAY	Loop 2 : Call delay (993 μs)
021F	CC 00					DEC	\$00	Subroutine 34 times
0221	Do F9					BNE	L2	
0223	20 72 89					JSR	BEEP	Sound beeper
0226	00						BRK	



PROGRAMMING SHEET

Program Title
DE

^mTitle Decay - 993 ns - Figure

Date

26



Synertek
INCORPORATED

3001 STENDER WAY • SANTA CLARA, CA 95051
TELEPHONE (408) 988-5600 • TWX: 910-338-0135

TECHNICAL
NOTE

53-SSC

April 1979

Trigonometric Functions for Synertek BASIC

This note describes the incorporation of trig functions into Synertek BASIC, BAS-1. Using the procedures described allows trig functions to be loaded from cassette tape when needed and called by a simple function call.

FEATURES

- * SIN, COS, TAN, ATN
- * Accuracy to 10^{-7} (for arguments between minus two Pi and plus two Pi)
- * Calculates SIN in less than 28mS
- * Takes up only 313 bytes of RAM
- * May be located on any two consecutive pages in memory

GETTING TRIG ON YOUR SYSTEM

For a 4K RAM system, the listing of Figure 1 should be typed in as shown. This will locate the trig functions at the top of memory. If you have more (or less) memory, then you will need to relocate it at the top of your memory space. The first byte of the listing is 0B at location 0EC7. The last byte is 01 at location 0FFF. Type in the bytes as shown using the monitor Memory or Deposit modes. After you are done, do a Verify listing. The checksum value should be 9476 if you have not made any mistakes. See Figure 2.

Now save the bytes on cassette. You will probably want to save it as the first file on a tape which contains BASIC programs that require trig functions. The following monitor command will do this:

.S2 54,EC7,FFF

By using file number 54, this can be read back in BASIC as file T. Be sure this won't conflict with any BASIC programs named T on the same tape.

USING TRIG FUNCTIONS

After a .J 0 from monitor to get to BASIC, type in the memory size that will reserve enough room for trig functions (and machine language if necessary) at the top of memory. On a 4K RAM system this would be 3782 if no machine language space is reserved. When BASIC responds with OK, insert the cassette that contains the trig functions and type LOAD T. After it is loaded, you must type either NEW or LOAD x. Next type the following line to attach the trig functions to BASIC:

POKE 196,104 : POKE 197,15

Instead of typing this line each time you load BASIC, you may use this as the first line in any BASIC program that uses trig functions. See Figures 3 and 4.

In the case where it is desired to load the trig functions when a BASIC program already exists in RAM, exit BASIC to SUPERMON, load the trig functions and return to BASIC. Be sure to un-write-protect system RAM and to attach the trig functions to BASIC. See Figure 5.

F68

RELOCATING TRIG FUNCTIONS

Trig functions have been written so that they may reside on any two consecutive pages in RAM. However, the relative location on the page must stay as it is. In other words, the 0B at location 0EC7 must be at location XXC7, where XX is the page on which it is located, and the 01 at location 0FFF must be at location YYFF, where YY is one greater than XX. When attaching trig functions using the POKE statements, the number 15 must be replaced by the decimal equivalent of page YY.

CEC7 - CF03 - DATA
CF04 - CF34 - PROG
CF35 - CF67 - DATA
CF68 - CFFF - PROG.

29

Figure 1. Object Code Listing for Trig Functions

V EC7,FFF

OECD	0B	76	B3	83	BD	D3	79	1E,DE
OECF	F4	A6	F5	7B	83	FC	B0	10,27
OED7	7C	0C	1F	67	CA	7C	DE	53,AC
OEIF	C8	C1	7D	14	64	70	4C	70,66
OEE7	B7	EA	51	7A	7D	63	30	88,6A
OEEF	7E	7E	92	44	99	3A	7E	4C,D9
OEF7	CC	91	C7	7F	AA	AA	AA	13,8D
OEFF	81	00	00	00	00	45	B6	48,B1 (FO4)
OF07	10	03	20	36	DD	A5	B1	48,95
OF0F	C9	81	90	07	A9	72	A0	07,08
OF17	20	C5	D8	A9	C7	A4	C5	88,26
OF1F	20	C2	DD	68	C9	81	90	07,2E
OF27	A9	35	A4	C5	20	06	D6	68,D9
OF2F	10	03	4C	36	DD	60	81	49,75
OF37	0F	DA	A2	7F	00	00	00	00,7F
OF3F	05	84	E6	1A	2D	1B	86	28,FE
OF47	07	FB	F8	87	99	68	89	01,0A
OF4F	87	23	35	0F	E1	86	A5	5D,31
OF57	E7	28	83	49	0F	DA	A2	A1,38
OF5F	54	46	8F	13	8F	52	43	89,21
OF67	CD	C0	72	F0	4A	90	41	C0,EB
OF6F	76	F0	92	20	80	D9	A9	00,05
OF77	85	16	A5	C5	48	A9	85	48,C8
OF7F	A5	C5	48	A9	B5	48	60	A2,22
OF87	9E	A0	00	20	8A	D9	A9	A7,33
OF8F	A0	00	20	58	D9	A9	00	85,52
OF97	B6	A5	C5	48	A9	A7	48	A5,F7
OF9F	16	48	A5	C5	48	A9	E7	48,DF
OFA7	60	A9	9E	A0	00	4C	C5	D8,0F
OFAF	A9	35	A4	C5	20	1D	D6	20,89
OFB7	C2	D9	A9	59	A4	C5	A6	BE,F3
OFBF	20	BD	D8	20	C2	D9	20	82,05
OFC7	DA	A9	00	85	BF	20	09	D6,CB
OFCF	A9	3A	A4	C5	20	06	D6	A5,B8
OFD7	B6	48	10	0D	20	FF	D5	A5,6C
OFIG	B6	30	09	A5	16	49	FF	85,E3
OFE7	16	20	36	DD	A9	3A	A4	C5,78
OFEF	20	1D	D6	68	10	03	20	36,5C
OFF7	DD	A9	3F	A4	C5	4C	C2	DD,75
OFFF	01,76							

9476

Figure 2. Example of Loading and Verifying Trig Function Code

.M EC7
 OEC7,00,OB
 OEC8,00,76 ← Type in
 OEC9,00,B3 trig
 OECA,00,83 functions
 OECB,00,

* * *
 OFFB,00,C5
 OFFC,00,4C
 OFFD,00,C2
 OFFE,00,D0
 OFFF,00,01
 1000,10,
.V EC7,FFF ← Verify
 your work
 OEC7 OB 76 B3 83 BD D3 79 1E,DE
 OECF F4 A6 F5 7B 83 FC B0 10,27
 OED7 7C 0C 1F 67 CA 7C DE 53,AC
 OEDF CB C1 7D 14 64 70 4C

* * *
 OFD7 B6 48 10 0D 20 FF D5 A5,6C
 OFDF B6 30 09 A5 16 49 FF 85,E3
 OFE7 16 20 36 DD A9 3A A4 C5,78
 OFEF 20 1D D6 68 10 03 20 36,5C
 OFF7 DD A9 3F A4 C5 4C C2 DD,75
 OFFF 01,76
 9476 ← Checksum must
 be 9476

Save on one
 or more
 cassettes

Figure 3. Loading Trig Functions and a Program Using the Trig Functions

```

J 0
MEMORY SIZE? 3782 ← Save room for trig
WIDTH? __

3269 BYTES FREE

BASIC V1.1
COPYRIGHT 1978 SYNERTEK SYSTEMS CORP.

OK
LOAD T ← Load trig functions
LOADED
OK
LOADA ← Load rec/polar
LOADED program
OK
RUN
TO WHAT? P
X,Y? 3 , 4
MAG= 5 ANGLE= 53.1301024
TO WHAT? R
MAG,ANGLE? 5 , 53.1301024
X= 3 Y= 4
TO WHAT? __

OK

```

Figure 4. Coordinate Conversion Program Which Uses Trig Functions

Note Line 110

```

100 REM RECTANGULAR/POLAR COORDINATE CONVERSION
110 POKE 196,104 : POKE 197,15 : REM ATTACH TRIG FUNCTIONS
120 INPUT "TO WHAT? ";A$
130 IF A$="P" GOTO 210
140 IF A$="R" GOTO 160
150 PRINT "USE P OR R" : GOTO 120
160 INPUT "MAG,ANGLE? ";M,T : T=T*3.141592654/180 : REM CONVRT TO RADs
170 X=M*COS(T)
180 Y=M*SIN(T)
190 PRINT "X=";X,"Y=";Y
200 GOTO 120
210 INPUT "X,Y? ";X,Y
220 M=SQR(X*X+Y*Y)
230 T=ATN(Y/X)*180/3.141592654 : REM CONVRT RADs BACK TO DEGs
240 PRINT "MAG=";M,"ANGLE=";T
250 GOTO 120
999 END
OK

```

Figure 5. Loading Trig Functions when Another Program Already Exists in Memory

```

.J 0
MEMORY SIZE? 3782 ← Always
WIDTH? — save room for trig
3269 BYTES FREE

BASIC V1.1
COPYRIGHT 1978 SYNERTEK SYSTEMS CORP.

OK
100INPUT Y
200X=LOG(Y*5) ← Type in a program
300PRINT X
400Z=SIN(Y/3)
500PRINT Z
999END

RUN
? 4
2.99573227

?FC ERROR IN 400 ← Trig is needed
OK
Q=USR(&"8035",0) ← Go to monitor

CB6D,3
.L2 54 ← Load trig
.G 0 ← Go back to basic

OK ← Un-write protect monitor
Q=USR(&"8B86",0) RAM !!

OK
50POKE 196,104 : POKE 197,15
RUN ← Attach trig
? 4
2.99573227
.971937901 ← Now it's OK !

OK

```



SynerTek

SYSTEMS

3050 CORONADO DRIVE • SANTA CLARA, CA 95051
TELEPHONE (408) 984-8900 • TWX: 910-338-0135

(408) 988-5689

TECHNICAL
NOTE

No. 54-SSC

DECEMBER 1978

SYM-1 POWER-UP TO USER ROM

Many applications for the SYM-1 require that user-written code be executed upon power-on reset. To understand how this is done, we will describe the normal power-on reset sequence, show how to modify this sequence, and give an example.

In response to the RES signal from the power-on circuit, the SY6502 microprocessor attempts to fetch a reset vector from locations FFFC and FFFD. RES also sets CA2, pin 39, from device U25 (SY6522 #1) to a high state, generating POR. While POR is active, all select lines from decoder U10 and U11 are deselected, disabling all peripheral circuits. POR additionally enables ROM device U20 through a jumper from 19 to N.

ROM U20 generally contains the SUPERMON monitor program. Since this is the only device enabled by the POR signal, the reset vector is fetched from this chip at locations 8FFC and 8FFD, even though the microprocessor "thinks" it is fetching from FFFC and FFFD. The reset vector points to the location of the reset routine, in this case 8B4A, which must be in this same physical ROM.

Once the reset vector is fetched, the microprocessor begins execution of the reset routine (at location 8B4A for SUPERMON). Among other things, the reset routine initializes the stack pointer, initializes the status register, disables POR, initializes system RAM, and jumps to the monitor.

Note that the power-on reset signal, POR, is disabled by the program during the reset routine. However, at the time POR is disabled, the microprocessor is already fetching its instructions from locations in device U20 so the process proceeds smoothly.

To enable a different ROM at power-on-reset, jumpers 19 and 20 must be changed and a reset routine placed in the new ROM. As an example, consider the requirements for doing a power-on reset to

to a 2K byte user program located in device U23, address space D000 to D7FF. First, POR would have to be rerouted from device U20 to device U23 by changing jumpers per Figure 1. (These jumpers are located just above the socket for user supplied SY6422 VIA, device U28.)

1. Delete jumper from 19 to N.
2. Delete jumper from 20 to S.
3. Add jumper from 19 to S.
4. Add jumper from 20 to N.

FIGURE 1. JUMPER CHANGES

The final jumpers should be 19 to S and 20 to N, P, and R. Refer to Table 4-3 in the SYM Reference Manual for additional information. Next, a new reset vector must be located in the device at locations D7FC and D7FD. This vector must point to a location within this device, say D700.

Finally, install a new reset routine at location D700 with steps to initialize the stack pointer, disable POR, initialize the status register and system RAM (if used). Other features may be included here as required, such as initializing I/O ports, etc. As a last step in the reset routine, include a jump to the starting location of the user program. The listing in Figure 2 shows a sample reset routine. Also, refer to the listing of the SUPERMON reset routine (program location 8B4A) in the SYM Reference Manual.

NOTE: System RAM must be initialized if any SUPERMON subroutines are to be used.

Symtek Systems Corporation.

Figure 2. Sample Reset Routine

35

ADR	INSTRUCTIONS			LABEL	NMEMONIC	OPERAND	COMMENTS
	B1	B2	B3				
D100	A1	FF		RESET	LDX	# \$ FF	j Initialize Stack Pointer
D102	9A				TXS		
D103	A9	CC		POP	LDA	# \$ CC	
D105	8D	OC	A0		STA	PCR1	; Disable POR, tape off
D108	A9	04			LDA	# 4	
D10A	48				PHA		
D10B	28	8C	8B		PLP		
D10C	20				JSR	ACCESS	; Initialize Flags, disable IRQ
D10F	A2	5F		DFTXFR	LDX	# \$ 5F	; Unwrite Protect System RAM
D111	BD	A0	8F		LDA	DFTBLK, X	; Initialize System RAM
D114	9D	20	AC		STA	RAM, X	
D117	CA				DEX		
D118	10	F7			BPL	DFTXFR+2	
D11A	4C	X	Y		JMP	USER	j Jump to User Code
D1FC	00				*	\$ D7FC	
D1FD	D7				.BYT	\$ 00, \$ D7	



Synertek SYSTEMS CORPORATION
3050 CORONADO DRIVE • SANTA CLARA, CA 95051
TELEPHONE (408) 984-8900 • TWX: 910-338-0135

TECHNICAL
NOTE

No. 59-SSC

January 24, 1979

SYM-1 MODIFICATION FOR INPUTTING LOWER CASE CHARACTERS

The normal input routine for SYM-1 converts all incoming alpha data from a terminal to upper case. This is useful in those cases where a terminal normally sends lower case alpha characters unless the shift key is held down.

In other cases, however, it is desirable to be able to input lower case characters; for example, to enter text and character strings in Synertek BASIC (BAS-1).

The following short routine bypasses the upper-case conversion and allows lower case alpha characters to be input to the SYM-1 for further processing. After entering the routine, the vector INVEC (location \$A661) must be changed to point to the new routine. For example, for the routine given in the listing, change INVEC to point to \$0FF0:

.SD 0FF0,A661}

INVEC must be changed to point to the new location after each reset (RST key depressed). The program will not have to be re-entered, however, unless power is removed.

When using Synertek BASIC, be sure to allow space in memory for the new input routine. Since BASIC expects all values to be in decimal, 4K of RAM is actually 4096 bytes. Allowing 16 bytes for this routine (only 12 are actually used), we have available 4096 minus 16, or 4080 bytes of memory available. Therefore, when logging onto BASIC, answer the question "Memory size" with "4080."

Remember that after entering the new input routine and changing INVEC all commands to the SYM-1 and to BASIC will have to be entered in upper case letters only.



Synertek
INCORPORATED

3001 STENDER WAY • SANTA CLARA, CA 95051
TELEPHONE (408) 988-5600 • TWX: 910-338-0135

TECHNICAL
NOTE

No. 72-SSC

April, 1978

CASSETTE DATA READING USING SYM-1 HIGH SPEED FORMAT

In most cases, the best setting for cassette-player controls is found by the sync-tape procedure (SYM Reference Manual, Appendix F), or simply by a small amount of experimentation. This note discusses topics and techniques for consistant cassette reading by determining the characteristics of your recorder and SYM-1 board. These methods utilize the advantages of SYM-1 SUPERMON V 1.1 but are also applicable using SUPERMON V 1.0.

READ TIMING OPTIMIZATION

In some cassette players, peculiarities of the frequency response affect the read waveform enough to cause inconsistent data reading. Such players need not be of poor quality--indeed, the units with good high-frequency response are sometimes the worst offenders. (Good high-frequency response is sometimes achieved at the expense of severe phase distortion. The human ear is not sensitive to phase distortion, but computer data circuits are.)

An indication of this sort of problem is a very narrow or non-existent range of control settings which provide reliable reading. In such a case, it may be possible to widen the satisfactory range by adjusting timing parameters in the read program. The programs shown in Figures 1 and 2 will assist the adjustment. A test pattern is generated with the Pattern Generator program, and the resulting tape is read by the Timing Measurement program to determine data timing and margins.

BIT WAVEFORMS

Figure 3 shows idealized pictures of the waveforms for a "0" bit and a "1" bit. During reading the squared waveform continually alternates states. The direction of the transitions, and the levels between transitions (high or low), are not important. The information is contained entirely in the time between transitions (regardless of their direction).

If one inter-transition time is about 730 microseconds, a "0" has just been read. If two inter-transition times add up to about 730 microseconds, a "1" has just been read. (Actually, the 1/0 decision is based on the first of the two inter-transition times. If the bit is a "1," the second time is ignored.)

Figure 3 is idealized because it ignores the distortions which can occur in a real cassette player. Variations in amplitude and phase shift with frequency distort the waveform. The distortions cause variations in the zero-crossing times of the analog waveform, and hence in the timing of the squared waveform. (An additional cause of timing distortion is tape-speed variation. See later in this note.)

When reading, the system distinguishes a "1" from an "0" by comparing the inter-transition time with a boundary value. If the time is less than the boundary value, the bit is a "1"; if greater, a "0." Sufficiently large timing distortions will convert a "1" to an apparent "0" or vice versa. This can occur at different volume and tone settings on your cassette recorder.

The timing boundary is normally set to its default value, which is the best compromise for most cassette players. However, it may be changed by the user if desired.* (It is restored to the default value after every power-on or Reset.) The two programs discussed in the following paragraphs provide guidance for such changes.

PATTERN GENERATOR PROGRAM

The Pattern Generator Program places a known, fixed test pattern on a tape. This pattern is required by the Timing Measurement Program, and may also be used for other purposes if desired.

The pattern consists of a contiguous series of 256-byte blocks, each containing all possible 8-bit combinations in order: 00, 01, ..., FE, FF. The pattern is preceded by a string of SYNs and a * character.

The procedure for pattern generation is as follows:

1. Key in or read from tape the Pattern Generator Program (GEN).
2. Select the memory region to be used as temporary storage for the pattern. The region should have an integer number of 256-type pages, and should start on a page boundary (location xx00). It should not include pages 00, 01, or 02. It should be as long as possible, up to 15 or 20 pages, depending on available memory.
3. Using M commands, set the high byte of the regions start address into location \$01, and the number of 256-byte blocks (pages) into location \$02. (The low byte of the start address is always \$00, and need not be entered.)
4. Start the program with a G command (normally at \$2E2). The program will execute, filling the region with the pattern. If more than a few pages are filled, the pause due to execution will be perceptible.

5. Check two or three random locations in the region. The contents of the bytes should be equal to the low bytes of their addresses.
6. Write the test pattern to tape with a S2 command. Any file number may be used. The addresses will be the start and end of the memory region (xx00 to yyFF).
7. Rewind the tape. The pattern is now ready for use. You may wish to save the program on another tape.

- NOTE:
1. To minimize the data to be keyed in, tape control is manual. If you have automatic control, remove the Remote plug, and start the tape at the same time you press CR for the S2 command.
 2. If your recorder has capability to disable the automatic recording level circuit, you may have better results using a low-medium manual setting.

TIMING MEASUREMENT PROGRAM

The Timing Measurement Program reads the test-pattern tape, and calculates the following times:

MINT0--shortest "0" interval found in any byte of any block read

MAXT1A--longest "first part of a 1" found in any byte of any block read

MAXT1B--longest "second part of a 1" found in any byte of any block read

The times displayed are the number of 8-microsecond intervals in the time between transitions. For example, a value of 30 (hex) is 48 x 8 or 384 microseconds.

The optimum setting for the timing boundary (HSBDRY-- see monitor listing) is halfway between MINT0 and MAXT1A. If tape speed variations are suspected, this can be displaced somewhat toward MINT0, since most tape speed problems are slowdowns.* Conversion to microseconds is not required; HSBDRY is in the same units as MAXT1A and MINT0.

Procedure for using the Timing Measurement Program is as follows:

1. Key in or read from tape the Timing Measurement Program (TIMEAS).
2. Place the test-pattern tape in cassette player.
3. Select the number of 256-byte blocks to be analyzed. This must be equal to or less than the number of such blocks in the test-pattern record.

4. Using the M command, store the desired number of blocks (hexadecimal) into location \$26.
5. Start the program by a G200. The program will begin like an ordinary L2 read, with an S display. The S display will go out when the sync region is encountered. When the test pattern proper begins, a display consisting of only an underline in the first character will appear. Upon completion, this will disappear and the program will return to the monitor.
6. Display the results by an M command as follows:

\$20 - MINT0
 \$21 - MAXT1A
 \$22 - MAXT1B

7. If you wish to change HSBDRY for future reads, store the new value in \$A632.* This must be restored after every power-on or Reset.

NOTE: Tape control is manual. See note on Pattern Generator procedure.

IMPORTANT: Repeated use of the program with different volume and tone control settings will make evident the setting with the best margins (Largest MINT0-MAXT1A difference).

In extreme cases, it may not be possible to synchronize well enough to turn out the S and turn on the underline. The program is not useful when this happens. In less extreme cases, the data may be invalidated by loss of an entire transition (or transition pair), or MAXT1A may be greater than MINT0. Control adjustment may sometimes show the proper direction for improvement in this case.

TAPE SPEED VARIATION

If the read waveform is reasonably good, with good timing margins, most tape speed variations are within the timing tolerance of the system. However, tight read-timing margins or excessive tape-speed variation can cause tape speed to be a significant factor in read errors.

Tape speed variation has two major causes: binding in the cassette and contamination in the roller and capstan mechanism. Slight binding in the cassette, due perhaps to unevenness in the way the tape winds on the reel, can cause occasional data errors. The effect is made worse by a dirty capstand or pinch roller, which allows some slippage when the tension increases because of binding.

Most cassettes will have occasional slight binding, especially after they have been used for a time. The higher the mechanical quality, the less likely this is to be a problem. Cassettes designed specifically for data processing are reported to be better in this respect.

(Note that this need not have anything to do with the audio properties of the tape, which are what is usually thought of when cassette quality is mentioned.) Some improvement may be noticed when short tapes are used.

Cleaning of the capstan and pinch roller periodically will also help reduce speed variation.

The basic design and manufacture of the cassette player has some effect on speed variation, but this is usually a small factor. Wow and flutter which are large enough to be clearly audible are normally still small enough to have little effect on data read reliability.

TAPE DROPOUTS

Dirt, crinkling or creasing, and poor manufacturing control can cause momentary dropouts in a record. The dropout need not go all the way to zero amplitude to cause a data error--all it has to do is shift the apparent timing of one or more transitions. It may be inherent in the tape or temporary.

Again, tapes made for data processing will suffer less from this. Alternatively, it is possible to certify a tape before using it, by writing long records and assuring that they read back correctly. C

The probability of a dropout is lower after the first ten to twenty seconds of a tape. (For this reason, the Kansas City standard uses a 30 second leader interval.) If desired, the parameter TAPDEL may be altered to change the length of the pre-data SYN interval.* TAPDEL (location \$A630) is set to its default value of about 9 seconds at every power-on or Reset. It may be changed; each unit changes causes about 1.5 seconds additional delay.

MOTOR NOISE

Sometimes, the sync-tape process shows a wide range of control settings over which the reading is almost, but not quite, consistent. Typically, if the controls are set to the center of the good range, the S display flashes occasionally--once per second to once per minute, usually irregularly. This may be due to excessive motor-noise pulses from the player motor.

In order to reduce the effect of this motor-noise, you may wish to install a capacitor from the input of the read comparator (U26 pin 3) to ground. Depending on the noise the value should be between .05uf to .1uf. One convenient place to install it is shown in Figure 4. Installation of this capacitor may effect the largest MINT0-MAXT1A difference as determined earlier from the Timing Measurment Program.

If the problem persists, suspect an incorrect location for the capacitor, a poor solder joint, or a bad capacitor. If none of these is the cause, try a different brand of cassette player. A small percentage of players has noise so bad that the capacitor does not eliminate its effects. The trial replacement player should preferable be one which has worked on another SYM.

* Values for HSBDRY and TAPDEL may be changes only in SUPERMON V 1.1.

TWENTY IMPORTANT CASSETTE RECORDING GUIDELINES

1. Use high quality tape (Maxell UD or equivalent).
2. Use shortest tapes possible. You can shorten tapes to several minutes in length if you enjoy splicing.
3. Use shielded cable between your computer and the cassette recorder.
4. Keep heads and pinch rollers clean.
5. Keep heads aligned for tape interchangability.
6. Avoid recording too close to beginning of tape.
7. Make sure cassette is properly seated in recorder.
8. If you have trouble with a cassette try another. You can have a bad spot on tape or a warped cassette.
9. Highest setting of tone control is usually best.
10. A dirty recorder volume control can cause tape dropouts.
11. Make sure cassette connection plugs make good contact.
12. Rewind cassettes before removing them from recorder.
13. Store cassettes in dust-proof containers.
14. Avoid exposing cassettes to heat or magnetic fields.
15. Before recording, wind cassette to one end and fully rewind.
16. Cassette recorders will give you problems once in a while (They don't like certain cassettes, etc.). If one gives you problems most of the time replace it.
17. Make sure that MIKE plug is connected before recording. On most recorder the TAPE light will glow while recording.
18. You may have to record with the EAR plug out for some tape recorders.
19. Always use AC adaptor with recorder for best results.
20. When a tone control is available, adjust it to the highest possible setting (maximum treble).



PROGRAMMING SHEET

SSC

Program Title
TEST PATTERN GENERATION PROGRAM

Date
APRIL 21, 1979

45

Address	Instructions			Mnemonic	Operand	Comments	1 OF 1
	B1	B2	B3				
<pre>; FILL MEMORY REGION WITH REPETITIVE ; 00,01,.....,FF,FF PATTERN ; BEFORE STARTING, FILL "ADDRH" WITH HIGH ; BYTE OF START ADDRESS, AND "PAGES" WITH ; # OF 256 - BYTE BLOCKS TO FILL. DATA REGION ; MUST START ON A PAGE BOUNDARY. PROGRAM ; CAN BE RELOCATED ANYWHERE.</pre>							
2E2	A9	00	GEN	LDA	#0		
2E4	AA			TAX			
2E5	A8			TAY			
2E6	85	00	BYTE	STA	ADDRL		
2E8	91	00		STA	(ADDRL),Y		
2EA	E8			INX			
2EB	8A			TXA			
2EC	DO	F8		BNE	BYTE		
2EE	E6	01		INC	ADDRH		
2F0	C6	02		DEC	PAGES		
2F2	DO	F2		BNE	BYTE		
2F4	60			RTS			

FIGURE 1 TEST PATTERN GENERATION PROGRAM



PROGRAMMING SHEET

Program Title
CORPORATION

Programmer
SSC

Date
APRIL 21, 1979

Address	Instructions	B1	B2	B3	Label	Mnemonic	Operand	Comments	1 OF 6
j TAPE TIMING MEASUREMENT PROGRAM									
: START AT \$0200									
: RESULTS APPEAR IN PAGE 0 MEMORY:									
: \$20 = MINTO 8μs CTS IN MIN. "0"									
: \$21 = MAXTIA 8μs CTS IN MAX / 15									
: HALF OF "1"									
: \$22 = MAXTIB 8μs CTS IN MAX 2 ND									
: HALF OF "1"									
: TEST TAPE:									
: NORMAL S2 RECORDING CONTAINING									
: ONE OR MORE 00,01,...,FE,FF									
: BLOCKS, EACH 256 BYTES LONG.									
: BEFORE STARTING, "PASSES" (LOC. \$26)									
: MUST BE SET TO THE DESIRED #									
: OF BLOCKS TO BE RUN. THIS MUST									
: BE ≤ # OF BLOCKS IN TEST RECORD.									
: ANY VALUE OF "HSBDY" WHICH ALLOWS									
: THE READ ROUTINE TO CAPTURE SYNC									
: MAY BE USED.									

FIGURE 2 TAPE TIMING MEASUREMENT PROGRAM



PROGRAMMING SHEET

Program Title

systems
CORPORATION

PROGRAMMING SHEET Program Title S S C Programmer	Date APRIL 21, 19
---	-----------------------------

Programmer S5C

Date April 21, 1979

PROGRAMMING SHEET

Program Title



PROGRAMMING SHEET

Program Title
CORPORATION

Programmer
SSC

Date
April 21, 1979

Address	Instructions	B1	B2	B3	Label	Mnemonic	Operand	Comments
								BEFORE STARTING, BLOCK MOVE
								"LOADT" THRU "LT>H-1" TO
								217 - 27C (V1.0 MONITOR)
								217 - 27D (V1.1 MONITOR)
								THIS IS:
								.B 217-8C78-8CF4 (V1.0)
								.B 217-8C78-8CDD (V1.1)
								PROGRAM REQUIRES MOST OF
								PAGE 2 MEMORY
								DISPLAY OUTPUT
								FROM TAPE READ ROUTINE (V1.0)
								GETTR = \$8DCA (V1.1)
								MONITOR WRITE ACCESS
								MODE MARK FOR LOAD
								0/1 BOUNDARY TIME FOR GETTR
								MINO = \$20 COUNT MIN "0"
								MAXTA = \$21 COUNT MAX FIRST "1" HALF
								MAXTG = \$22 COUNT MAX SECOND "1" HALF



PROGRAMMING SHEET

Programmer

SSC

Program Title

APRIL 21, 1979

49

Address	Instructions			Label	Mnemonic	Operand	Comments	4 OF 6
	B1	B2	B3					
							; BLOCK MOVE "LOADT" THROUGH LTHW-1	
							TO HERE	
							; B 217-8C78-8CDE (VI.0 MONITOR)	
							; B 217-8C78-8CDE (VI.1 MONITOR)	
217	.RES	LTHW-LOADT			\$7D	IN VI.0 MONITOR		
							; AT END OF BLOCK MOVE, INSERT	
							; NOP'S (EA's) OR INSERT A	
							; JUMP TO NUBYT TO LOCATION 294	
							027D 4C 94 02 (VI.1)	
294	A2 08	NUBYT	LDX	#8		BIT COUNT & DISPLAY		
296	8E 00	A4	STX	DIG		DISPLAY UNDERLINE		
299	20 B3	8D	JSR	GETTR		PASS START BIT OF BYTE ("0")		
29C	46 23	BIT	LSR	TEMP		EXPECTED BIT		
29E	BO 0E		BCS	ONE		"1" OR "0"		
2A0	20 B3	8D ZERO	JSR	GETTR		TIME FOR EXPECTED 0		
2A3	38		SEC					
2A4	E5 F8		SBC	BDRY		A = 255 - COUNT + BDY		
2A6	C5 20		CMP	MINTO				
2A8	90 1C		BCC	NXTBIT		COUNT < OLD MIN?		



PROGRAMMING SHEET

Program Title

Programmer **SSC**
Date **APRIL 21, 1979**

Address	Instructions	B1	B2	B3	Label	Mnemonic	Operand	Comments	5 OF 6
2AA	85 20				STA	MINTO			
2AC	80 18				BCS	NXTBIT	; YES, STORE NEW MIN. COUNT		
2AE	20 B3	8D	01	1	JSR	GET1B	; TIME FOR EXPECTED FIRST HALF "1"		
2B1	38				SEC				
2B2	E5 F8				SBC	BDRY			
2B4	C5 21				CMP	MAXTIA	; A = 255 - COUNT		
2B6	80 02				BCS	ONE1	; COUNT > OLD MAX?		
2B8	85 21				STA	MAXTIA	; YES, STORE NEW MAX. COUNT		
2BA	20 B3	8D	01	1	JSR	GETTR	; REPEAT FOR SECOND HALF OF "1"		
2BD	38				SEC				
2BE	E5 F8				SBC	BDRY			
2C0	CS 22				CMP	MAXTIB			
2C2	80 02				BCS	NXTBIT			
2C4	85 22				STA	MAXTIB			
2C6	CA				NXTBIT	DEX			
2C7	DO D3				BNE	BIT			
2C9	E6 24				INC	BYTE	; NO, DO NEXT BIT		
2CB	A5 24				LDA	BYTE			
2CD	85 23				STA	TEMP			
2CF	DO C3				BNE	NUBYT	; REFRESH SHIFTING BYTE		
2D1	C6 25				DEC	PASS	; IF NOT END OF PASS, DO NEXT BYTE		
2D3	Da BF				BNE	NUBYT	; END OF PASS. LAST PASS?		



PROGRAMMING SHEET

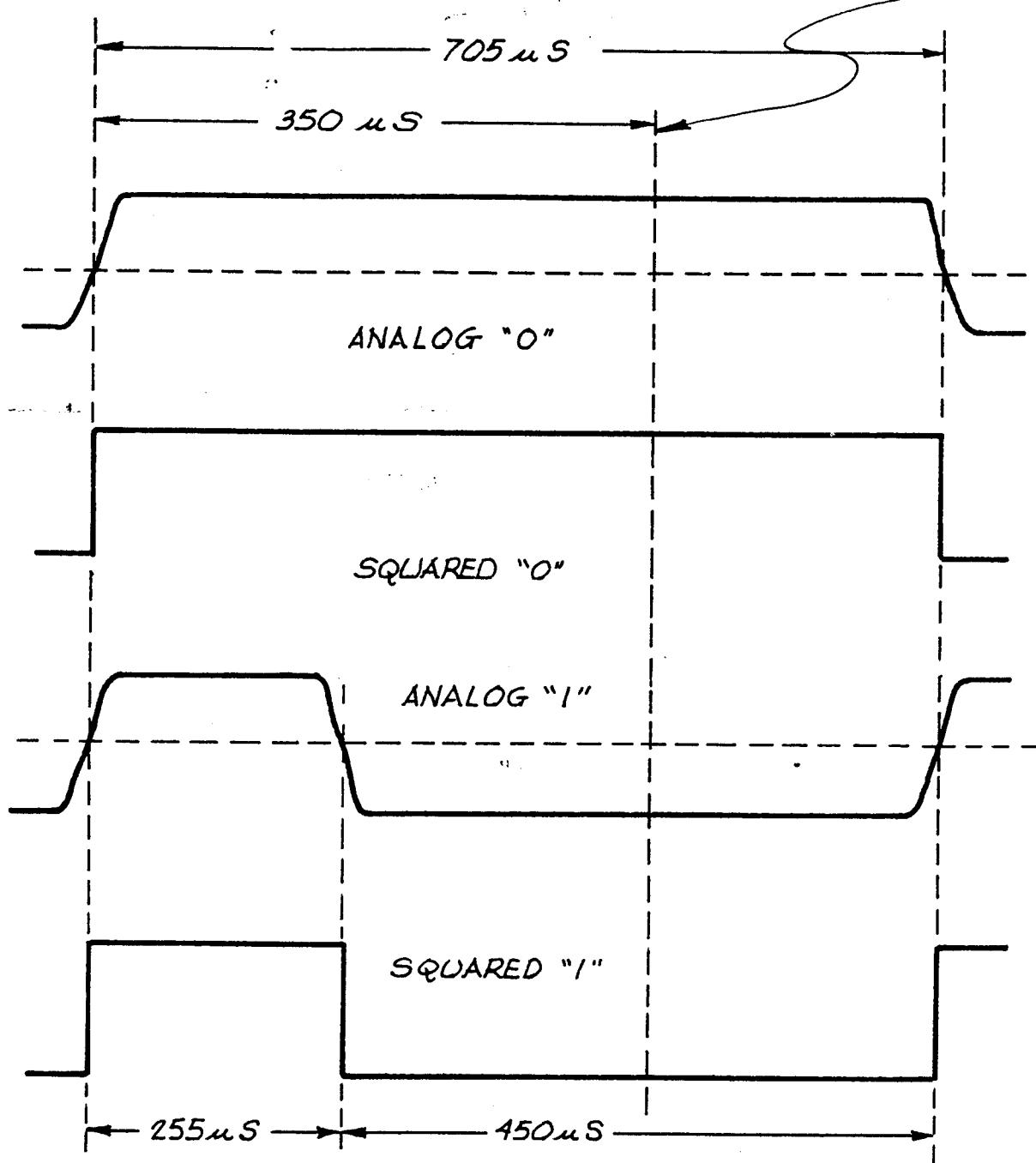
Programmer
SSC

Program Title

Date
APRIL 21, 1979

Address	Instructions			Label	Mnemonic	Operand	Comments
2D5	A2	03		LDX	#3		3 RESULTS
2D7	38			SEC			COUNT = 255 - (255 - COUNT)
2D8	A9	FF		LDA	#\$FF		
2DA	F5	1F		SBC	MINTO-1,X		
2DC	95	1F		STA	MINTU-1,X		
2DE	CA			DEX			
2DF	DO	F6		BNE	CALC		
2E1	60			RTS			
							MONITOR RETURN KILLS UNDERLINE
							DISPLAY

TYPICAL BOUNDARY VALUE (HSBDRY = \$46)
 LOCATION \$A632



<u>NAME</u>	<u>LOCATION</u>	<u>DEFAULT VALUE</u>	<u>DESCRIPTION</u>
HSBDRY	A632	\$46 (350 μs)	HS BOUNDARY
TAPET1	A635	\$33 (255 μs)	HS FIRST HALF "1" BIT
TAPET2	A63C	\$5A (450 μs)	HS SECOND HALF "1" L

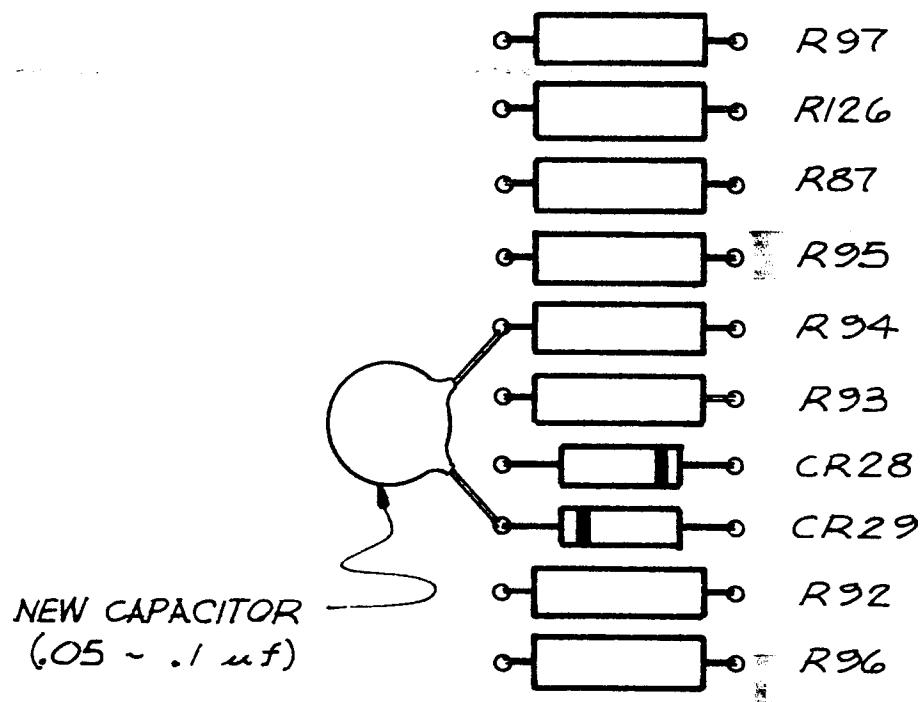


Figure 4. Motor Noise Capacitor Addition (Optional)



Synertek Systems Corporation

P.O. BOX 552 SANTA CLARA, CALIFORNIA 95052 TEL. (408) 988-5600 TWX: 910-338-0135