WIDGET SERVO AND MOTHER BOARD TESTER

4/12/83 G.C.

Objective:

The objective of this procedure is to show someone the correct method for testing Widget servo and mother boards.

Equipment Required:

1. Widget servo, mother boards and HDAs to be tested.
2. Good motor speed control boards (one for each HDA).
3. Good, "standard" Widget HDA (Head Disk Assembly).
4. Test base plate to mount the HDA and the mother board.
5. +5V/+12v power supply (Profile power supply).
6. Apple][, //e or /// with integer basic available.
7. One disk drive and a monitor for the above.
8. Oscilloscope, Tektronix model 465B or equiv.
9. SVO writer interface card with its cables.
10. Software, "Apple to Servo" titled "SV0020283CMP".

Test Procedure:

1.) With the Apple power off, install the "SVO WRITER INTERFACE" in slot #4 of the Apple and connect the 40-pin ribbon cable from J7 of the interface card to the small PCB that has the 5-pin power connector and the 40-pin male and female connectors on it. Connect the Profile power supply (switched off) to the 5-pin connector on the small PCB. When connecting the above cables, be certain that pin one goes to pin one.

2.) Be very careful with the "flex" printed circuits that come out of the HDA. Mount the HDA and the mother board to the base plate and connect the cables.

3.) Boot the Apple with "Apple to Servo" and select item #1 from the first menu to run "SV0020283CMP".

4.) Turn on the Profile power supply and the spindle motor must start spinning. Hit the system reset button on the end of the cable that connects to the SVO SERVO WRITER INTERFACE card.

5.) See Dan Retzinger's (Dated Mar. 16, 1983) WIDGET OPTICS AND TACH ADJUSTMENT PROCEDURE for adjusting the optics. Follow the procedure but be sure to refer to the addendum to the above procedure.

6.) SERVO BOARD CHECK OUT

A. Do a RECAL (via "R").

B. Select ALTERNATE TRACK SEEKS (via "Z") and type in 80 60 0 0 84 60 0 0 N (Hit <RETURN> after each space). Connect channel one of the 'scope to TP9 and trigger off TP27. Adjust R1 (on the servo board) for a period of 35ms (from start to end). (Scope on 5V/div and time base of 5ms). Press ESCAPE to exit.

C. Do a RECALL (via "R").

D. Select INCREMENT (via "I"). REVERSE RANGE of 500, FORWARD RANGE of 0, INCREMENT oF 1 ,0 0 N .and let the drive do several complete cycles. Next, do increments of 2,3,4 and 5, also letting the drive do several cycles each time. Press ESCAPE to exit.

E. Do a RECALL (via "R").

F. Select RANDOM (via "X"). REVERSE RANGE of 500, FORWARD RANGE of 0 , 0 0 N . Let the drive run this test for 3-5 minutes. Press ESCAPE to exit.

7.) Once the boards have passed all the above tests, place a sticker on the side of the HDA and write on it the date, your name and the serial numbers of the servo and mother board.

Tear Down:

1.) Turn the Profile power supply off and unsnap the small PCB from the mother board. Also remove the servo card from the mother board.

2.) Carefully unscrew the HDA (unless it is the standard) and the mother board from the test base plate and screw the mother board to the HDA.

3.) Slide the servo board on the top rails of the HDA and plug it into the mother board. All done!

ADDENDUM TO THE WIDGET OPTICS AND TACH ADJUSTMENT PROCEDURE

4/13/83 G.C.

PURPOSE:

The purpose of this note is to add some helpful information when doing the the WIDGET OPTICS AND TACH ADJUSTMENT PROCEDURE by Dan Retzinger (Dated Mar. 16, 1983).

1.) PAGE 2; AFTER STEP 10.
If you still cannot get the circle sides adjusted within the range of RP6 middle pot, set the pot to the center and go back and adjust to the value of R34 in increments of 100K so that it can be adjusted within the range of RP6 middle pot. Don't worry that TP5 will no longer measure 3.6 volts; <u>as long as the final voltage is between 2.7 and 4.3.</u>

2.) PAGE 3; DO THIS FIRST THING.
Reset the system (via the reset button at the end of the cable) and ground TP19. Observe TP7 with the 'scope set on 200mV/div and adjust RP6 opposite Q2 for a null (zero volts). Remove the grounding wire on TP19.

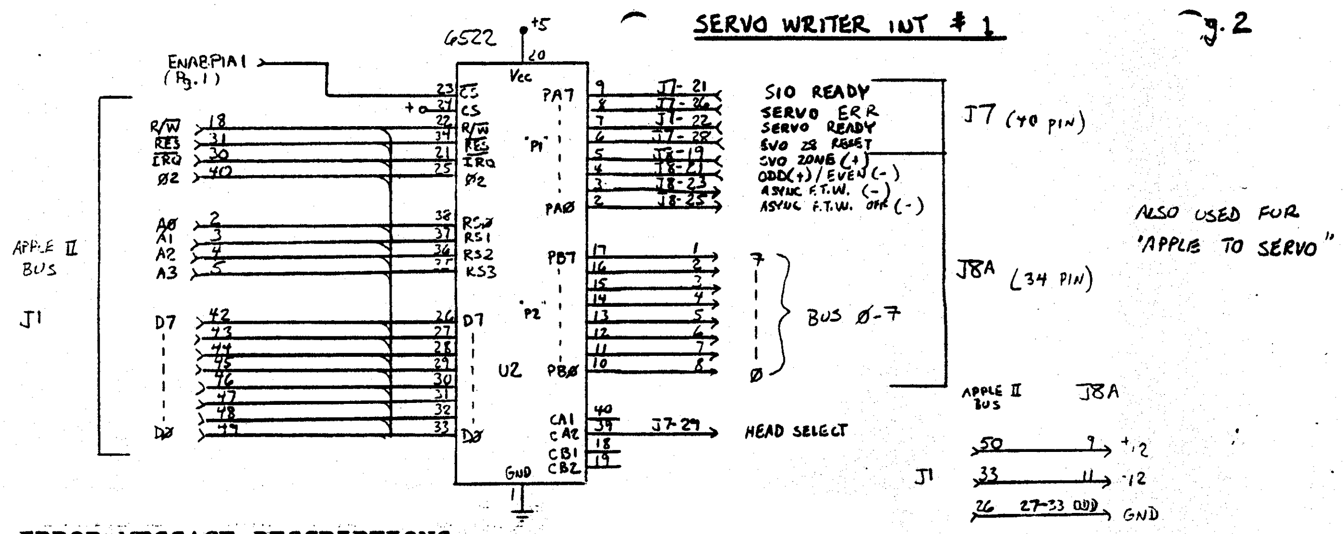
Dan Retzinger
March 28, 1983

APPLE TO SERVO ERROR DESCRIPTIONS

At the bottom of the APPLE TO SERVO menu "STAT= $XX" is displayed whenever a function such as RANDOM or ALTERNATE is not active. STAT gives a direct display of port #1 (IN HEX). The purpose is to give a visual display of the status of the Z8 Servo controller (bits 0 to 4 are displayed for fun). See the schematic below of Interface card. The bits correspond as follows:

BIT 7	SIO READY	
BIT 6	SERVO ERR	
BIT 5	SERVO READY	
BIT 4	SVO Z8 RESET	(not implemented)
BIT 3	SVO ZONE (+)	(not implemented)
BIT 2	ODD(+)/EVEN(-)	(not implemented)
BIT 1	ASYNC F.T.W. (-)	(not implemented)
BIT 0	ASYNC F.T.W. (-)	(not implemented)

SCHEMATIC OF THE INTERFACE PORT



ERROR MESSAGE DESCRIPTIONS

All error messages are generated by the levels on the the three interface lines or by testing for checksum errors (checksum tests the integrity of the five byte sequence).

"SERVO ~~NOT READY~~"	~~Servo must be reset~~
"SERVO ERROR"	Servo must be reset
"SIO NOT READY"	Communication path not ready, (Reset Servo)
"SIO STILL READY ERROR"	Communication path error (Reset Servo)
~~"SIOREADY WON'T COME ACTIVE"~~	~~Communication error~~
"TIMEOUT WAITING FOR STATUS"	Z8 won't respond
"CHECKSUM ERROR"	Checksum does not compute. (Bit lost)

Dan Retzinger
Mar. 16, 1983

WIDGET OPTICS AND TACH ADJUSTMENT PROCEEDURE

INTRODUCTION

The purpose of this note is to describe the procedure for adjusting five pots on the widget mother board used to control the amplitude of the optics signal (RP5, all four pots, and middle pot of RP6). Also, the Tach adjustment pot (Pot nearest Q2 of PR6,) and the Tach zero offset (Pot opposite Q2 of RP6) adjustments are described.

EQUIPMENT REQUIRED

An oscilloscope capable of operating in the X-Y mode of operation. A Tektronix model 465 works fine.

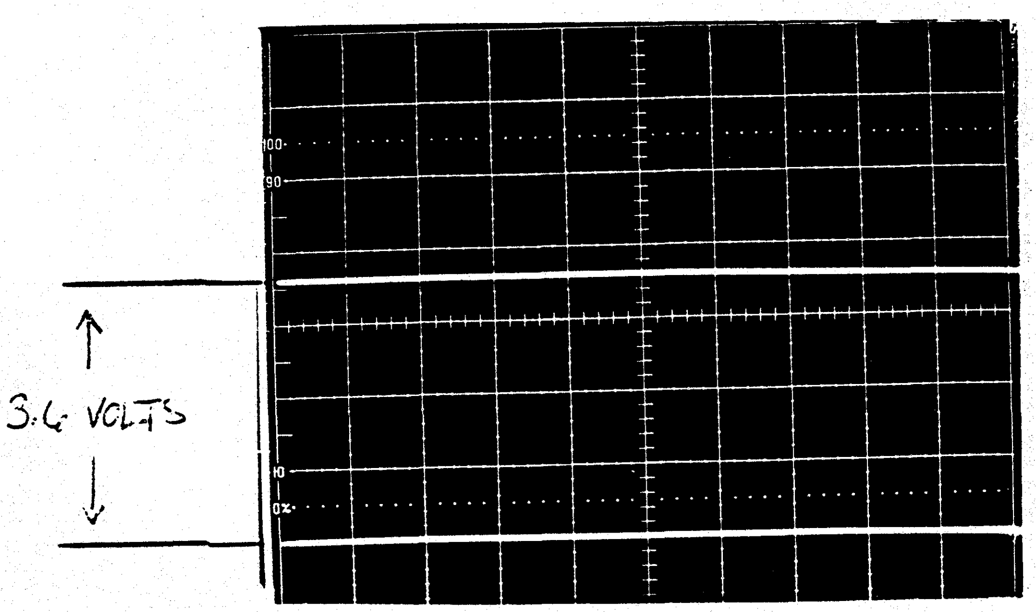
PROCEEDURE

OPTICS LED DRIVE ADJUSTMENT

1. Connect channel 1 of the oscilloscope to TP 5 on the Widget Mother Board.
2. Scope Vert. setting: 1 Volt/Div. Horizontal: Any sweep rate.
3. Adjust RP6, MIDDLE POT, so the voltage at TP5 is 3.6 volts +/- .2 volts.
   (Counter-clockwise, or more resistance=lower voltage)

NOTE: It may be necessary to change R34 in the mother bd. if the pot (RP6-Middle pot) does not allow 3.6 volts to be reached. R34 smaller= higher voltage. Increase or decrease R34 in 100k increments.

FIGURE 1: TP 5 AMPLITUDE



POSITION A AND POSITION B ADJUSTMENT

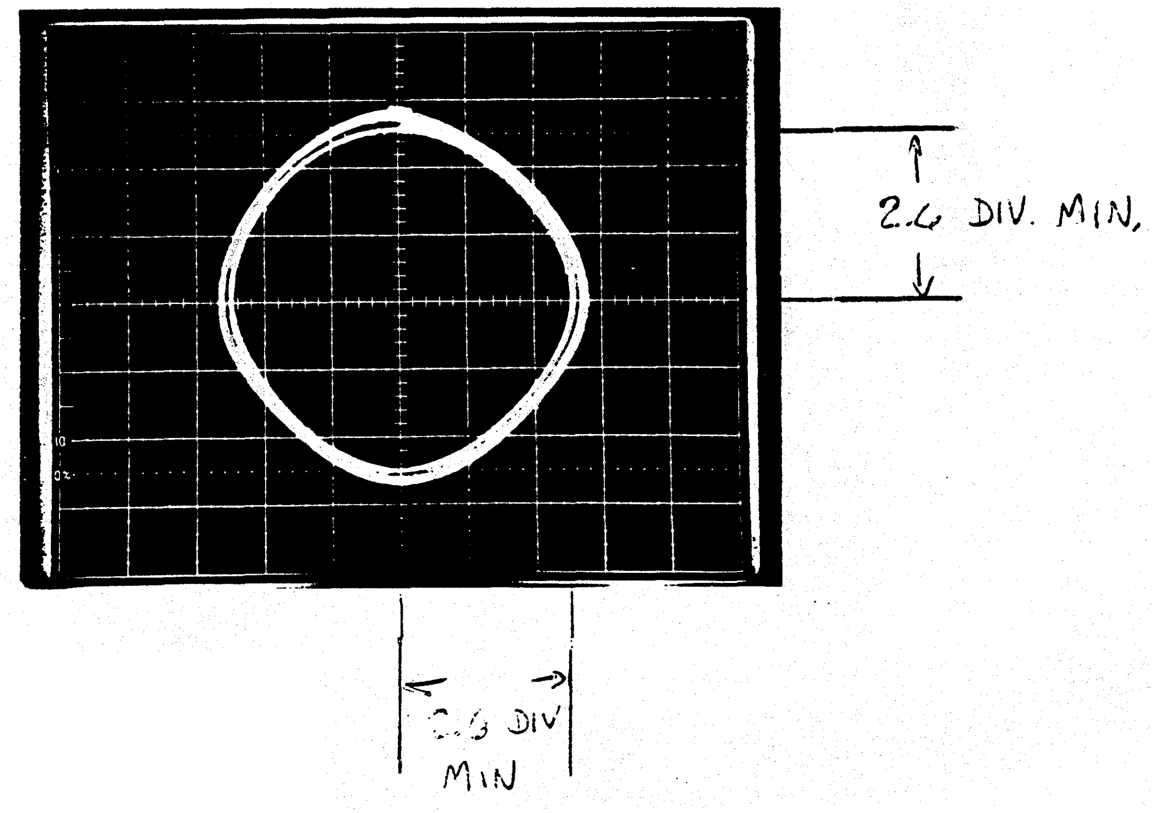
4. Put scope in X-Y mode, ground channels X and Y, move dot to center of screen.
5. Connect chan X to TP9, chan Y to TP8. (Both TP's are located near pin 1 of the Z8 microprocessor)
6. Scope vertical: Chan X and Y, 2 volts/Div.
7. At this point arm is to be moved. ** Move by hand at this time **
8. With arm in movement, a circular pattern should appear on the scope. Four pots, all part of RP5 are to be adjusted so that the top, bottom, right and left sides of the circle come at but no closer than a minimum of 2.6 scope divisions from the center of the screen.
9. Each pot adjusts the circle as follows:

Pot near TP1	Bottom side	clockwise or lower res=smaller circle
Pot near TP2	Top side	"
Pot near TP3	Left side	"
Pot near TP4	Right side	"

10. Figure 2 shows a properly adjusted optics signal.

NOTE: If the circle sides cannot be adjusted small enough, adjust pot RP6, middle pot (step 3, above) so that this can be achieved.

Figure 2: Position A and B



TACH ADJUSTMENT

NOTE: The tach pot, (RP6—Pot near Q2) is adjusted while seeking alternate 60 (hex) track seeks. If alternate 60 track seeks will not function, type "R" once, then adjust RP6—pot opposite Q2 while monitoring TP7 (adj for approx zero volts on the 200 milivolt/div scale.)

Also do the following while typing "R", Recal:

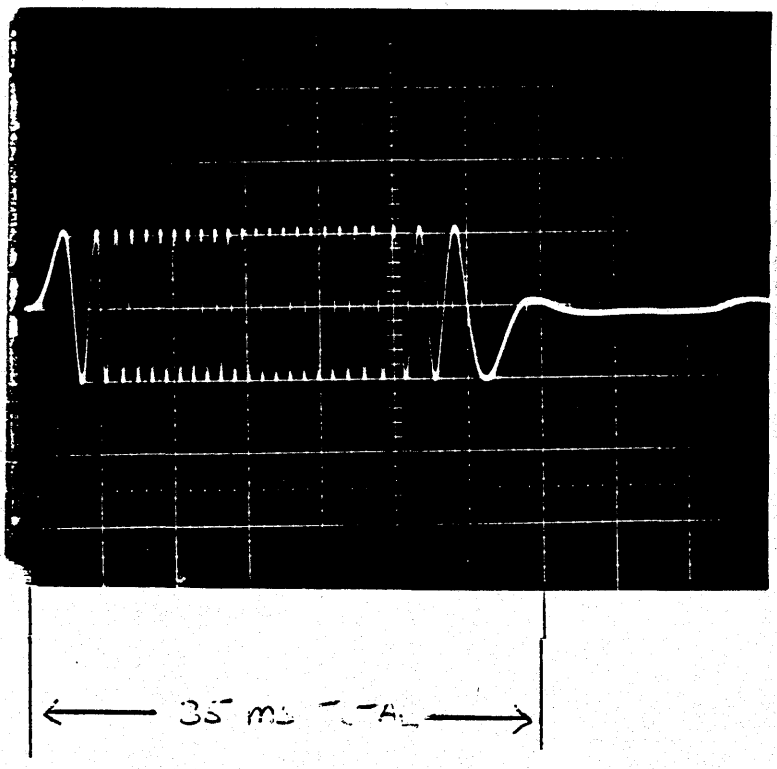
A. Externally trig off TP25
B. Scope channel 1 to TP8
C. Adjust tach pot for a recal period of approx. 120 ms.

11. With the "Apple to Servo" software, enter "Z" for alternate, then
    80 <Ret>, 60 <Ret>, 00 <Ret>, 00 <Ret>
    84 <Ret>, 60 <Ret>, 00 <Ret>, 00 <Ret>
    SEND READ STAT CMD? (Y/N)=N <RET>

12. Connect scope Chan 1 to TP9
    Scope Vert. 5 Volts/Div.
    Scope Horizontal 5ms./Div
    Ext. trig TP 27

13. Adj. Tach pot RP6—near Q2, for 35ms total seek time.

Figure 3: Tach Adj.

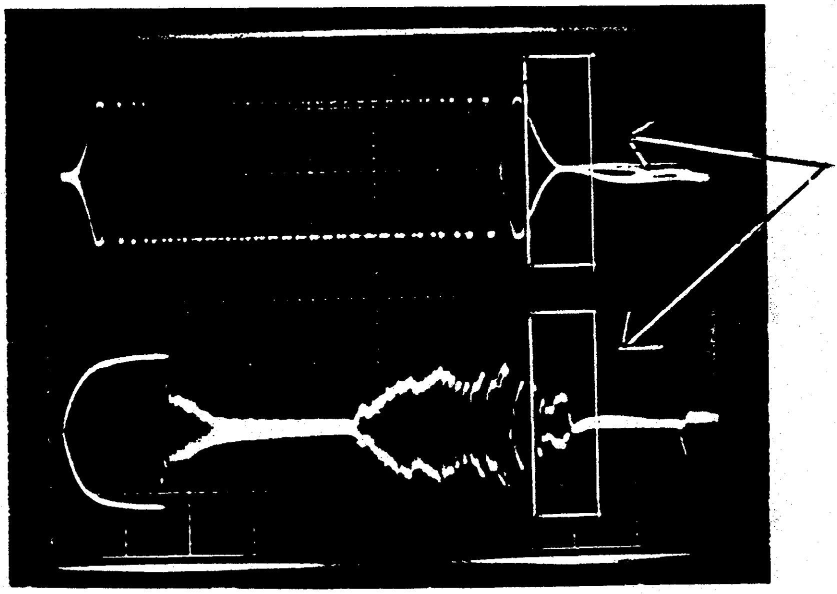


TACH ZERO OFFSET ADJUSTMENT

14. With drive performing 60 track alternate seeks set up the following:
    A. Scope chan 1 to TP9
    B. Scope chan 2 to TP19
    C. Both chanels vertical 5V/Div.
    D. Scope Horizontal 2 ms./Div.
    E. Trigger externally off of TP27.
    F. Adjust the variable horizontal delay so that alternate triggering appears as in the picture below (FIG. 4).

15. Adjust Zero offset pot, RP6—Pot opposite Q2, so that the ends of the waveforms are symetrical as shown below.

Figure 4: Zero Offset Adjustment



ADJUST FOR SYMETRY IN THESE AREAS.

PROCEEDURE SUMMARY

1. Adjust RP6—middle pot, so the voltage at TP5 (R37) is 3.6 Volts +/- .2 volts. (Absolute Max TP5 range is 2.7-4.3 volts)

2. Put scope in X-Y mode, chan 1 & 2 set to 2 volts/div. Adjust RP5 so that the sides of the circle (during minimum fluctuation) are each within 2.6 Divisions (-0, +.2 div) of the center. This corresponds to 5 Volts from the center to the top, bottom, or either side.

3. Adjust RP6—pot near Q2 for 35ms 60 track seek time.

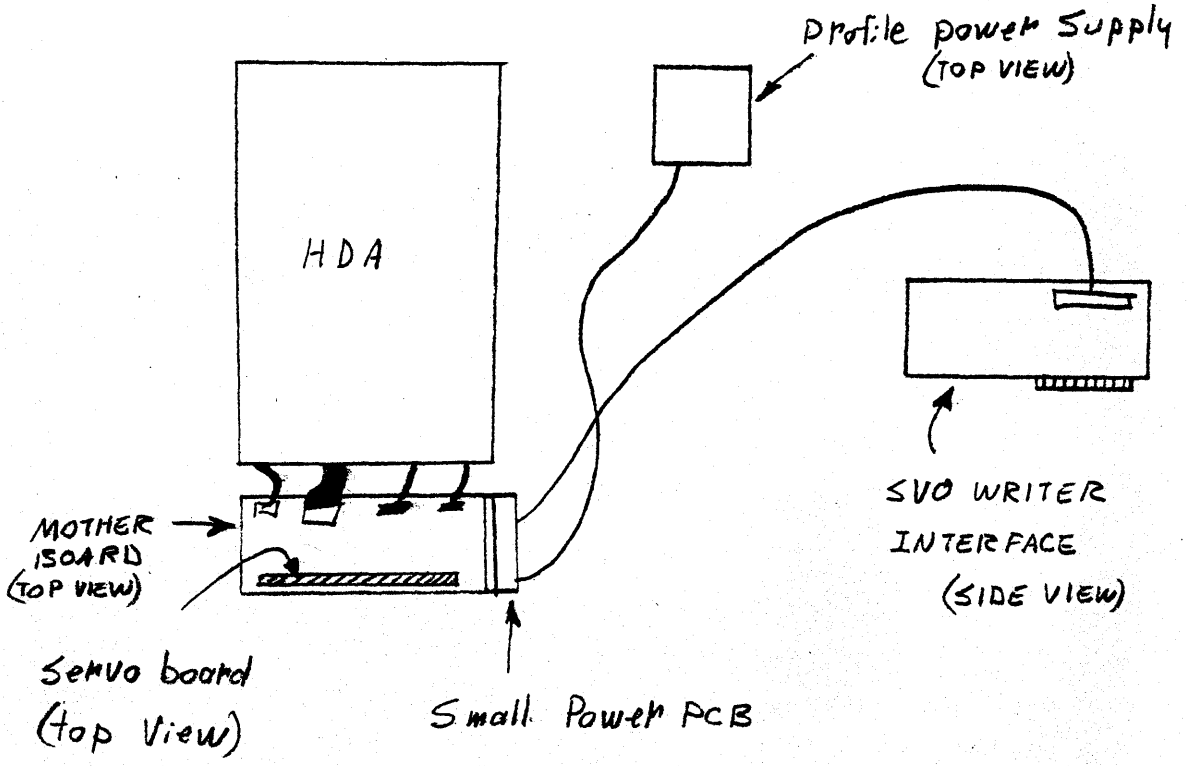
4. Adjust RP6—pot opposite Q2 for Zero Offset as per figure 4.

ADDITIONAL INFORMATION NEEDED FOR WALT WEBBER

To provide information to convert the resistor trimming process into a laser trimming process, Walt Webber needs the following information:

1. The actual final resistor value of R34 and RP6 (center) on a properly adjusted mother board. (LED current drive adj.)

2. The final resistor value of the resistor pairs for adjusting the sides of the circle: pairs RP1 100k SIPS and RP5 pots.



WIDGET SERVO AND MOTHER BOARD PARTS LOCATOR

apple computer inc.	SIZE A	DRAWING NUMBER BY GLEN CLOSSON
	SCALE: 4/15/83	SHEET 1 . OF 1

WIDGET SERVO AND MOTHER BOARD TESTER BILL OF MATERIALS

4/15/83 G.C.

APPLE PN	DESCRIPTION	QTY	REFERENCE DESIGNATION
337-0002	6551 ASYNC COMM ADAP	1	U1
337-0005	6522 MICROPOCESSOR (2MHz)	2	U2, U6
305-0003	74LS03 QUAD TWO-INPUT NAND (OPEN C)	1	U3
305-0030	74LS30 EIGHT-INPUT NAND	1	U4
305-0089	74LS86 QUAD TWO-INPUT EXCLUSIVE OR	1	U5
305-0138	74LS138 1-OF-8 DECODER/DEMUX	1	U7
	CRYSTAL(S)		
197-0015	1.8432 MHz CRYSTAL	1	Y1
	CONNECTORS		
609-3432M	34-PIN MALE STRAIGHT HEADER	1	J8
609-4032M	40-PIN MALE STRAIGHT HEADER	1	J7
609-4015M	40-POS FEMALE CARD EDGE CONN.	1	
609-4030	40-PIN FEMALE CABLE CONNECTOR	1	
609-4031	40-PIN STRAIN RELIEF FOR ABOVE	1	
	NOTE : The above part numbers are T&B Ansley.		
EZC20DRXN	20/20 PIN PC EDGE CONNECTOR	1*	
	NOTE : The above part number is SULLINS.		
09-50-3051	5 POS CRIMP TERMINAL HOUSING	1*	
22-05-2021	SQUARE 2-PIN RIGHT ANGLE WAFER	1	
	NOTE : The above part number are MOLEX.		
515-0150	HEADER, RT ANGLE, 5-PIN (POWER)	1*	
	NOTE : The above part number is APPLE.		
	RESISTOR(S)		
101-4103	10K OHM 1/4 WATT 5%	6	R1-R6
	SWITCHES		
8121	SPDT PUSH-BUTTON SWITCH	1	
8018	BLACK CAP FOR ABOVE	1	
	NOTE : The above part numbers are C & K.		
DLS-4	4-POS DIP SWITCH	1	S1
	NOTE : The above part number is ALCOSWITCH.		
	CAPACITORS		
126-5102	10uF 15V (OR MORE) ELECTROLYTIC	1	C6
135-9101	.1uF, +80%/-20%, 50V, BY-PASS	5	C1-C5

IC SOCKETS

511-1401	STANDARD 14 PIN DIP SOCKETS	3
511-1601	STANDARD 16 PIN DIP SOCKETS	1
511-2801	STANDARD 28 PIN DIP SOCKETS	1
511-4001	STANDARD 40 PIN DIP SOCKETS	2

MISC.

699-0059	PROFILE POWER SUPPLY	1
590-0003	APPLE AC POWER CORD (FOR ABOVE)	1
-*	PCB #611128, SMALL POWER PCB	1*
-*	PCB #611123, SVO WRITTER INTERFACE	1

NOTE : Parts listed that have a "*" mean that they are for the small power PCB.

Dan Retzinger
March 8, 1983

APPLE TO SERVO
HARDWARE/SOFTWARE DESCRIPTION (IN-HOUSE USE ONLY)

TABLE OF CONTENTS

INTRODUCTION

"Apple To Servo" is a program designed to exercise the arm positioning servo circuitry of Apple's Widget hard disk drive. The hardware/software program runs on an Apple II and offers commands ranging from format recal to random accesses. No read/write to disk surface functions are supported.

EQUIPMENT REQUIRED

The following items are required:

1. Diskette titled "Apple To Servo"
2. An Apple II with Integer Basic
3. Hardware interface card titled (erroneously) "SVO WRITER INTERFACE" (p/n 821123 from Hal Masamori) Install this PC board in slot 4 of the Apple II.
4. Various cables, power supply, Widget drive, etc.
5. Paper titled "Widget Servo Functional Objective" by Jim Reed containing servo command byte formats

OPERATION

START-UP

Run the HELLO program on the diskette and nine selections will be presented on the screen. Press 1 to select the most current version of Apple To Servo, "SVO020283CMP" in the disk catalog. This is a COMPILED version (for speed) of "SVO020283". After the program loads, ten selections for controlling the servo will be shown.

Turn on the Widget power supply, and press the RESET switch (connected to the interface card plugged into the Apple II).

USING THE COMMANDS

Selection A: ACCESS1 , allows four byte sequences, or commands to be sent to the Z8 Widget servo controller. This command is normally used to cause the servo to seek to some location. For example, entering 80<ret>, 10<ret>, 00<ret>, 00<ret>, will cause a ten (hex) track seek. See the "Widget Servo Functional Objective" (Jim Reed) for a description of the exact byte formats.

Note: Once "A" is pressed, all fo$\overset{u}{\wedge}$r bytes must be entered for the command to terminate.

```
Example:
          Value For Byte 1 $       (enter 80 <RETURN>)
          Value For Byte 2 $       (enter 40 <RETURN>)
          Value For Byte 3 $       (enter 00 <RETURN>)
          Value For Byte 4 $       (enter 00 <RETURN>)
```

This example will issue a 40 track seek.

Selection S: RETRY (A) , provides a simple means of re-issuing command A without re-typing the four bytes entered.

Selection C: ACCESS2 and selection D: RETRY (C) are simply duplicates of commands A and S so that two command sequences can be sent, or alternated between easily.

Selection R: SEND RECAL when pressed, immediately issues a recal command to the Z8 servo. This is usually done before using the following commands such as Alternate, Random, or Increment. If the servo does not respond to Recal at any time, press the RESET switch connected to the interface card (plugged into the Apple).

Selection Z: ALTERNATE requires eight hex entries be made, and then asks if the user would like to SEND READ STATUS COMMANDS in addition to the two alternated commands.

Example: Press Z then enter:

```
80, 60, 00, 00              (RETURN after each byte)
84, 60, 00, 00              (commas not entered)
SEND READ STAT CMD?(Y/N)    (If Y then enter 4 bytes)
00, 00, 00, 01
```

This causes alternate 60 (hex) track seeks and reports to the screen the Z8 status received between each command.

Selection X: RANDOM generates seek commands of random lengths. When selected, the following parameters must be entered:

Reverse Range	(maximum distance for reverse travel, relative to arm's present position)
Forward Range	(maximum distance for forward travel)
Offset Capability?(Y/N)?	(add auto or manual offset ability)
Value for byte 3?	(enter byte 3, sent with every command)
Value for byte 4?	(enter byte 4, sent with every command)
Send Read Stat Cmd?(Y/N)?	(Y if read stat to be added after cmds)

(Enter 4 bytes if Yes for Read Stat for Read Stat Cmd)

Selection I: INCREMENT generates incremental seeks. The parameters are the same as for RANDOM seeks except that after Forward Range, the Increment Step is entered.

Selection W: SERVO STAT requires four bytes to be entered, and responds with status information from the Z8 system. See "Widget Servo Functional Objective" for the types of status and their meanings.

OPERATION SUMMARY

1. Once any command is entered, all information must be entered. The ESCAPE key can be used for termination of ALTERNATE, RANDOM, or INCREMENT.
2. Press the RESET switch (on the interface board) if no response from Z8 servo, or the message "SERVO NOT READY" appears.
3. Jim Reed can answer any questions about using "Apple To Servo".

HARDWARE DESCRIPTION

Refer to schematic titled "SVO WRITER INTERFACE TO APPLE II". The same interface PC Board is used for both the servo writer interface and "Apple to Servo".

I.C. U3 (74LS04) and I.C. U4 (74LS30) decode the 5 upper-most address bits (A15-A11) while I.C. U5 (74LS86) and Dip Switch S1 decode address bits A10 through A8. These 8 bits form the high order address byte which fall between hex $C1XX and $C7XX. Slot 4 is used in "Apple to Servo" (C4XX). I.C. U7 (74LS138) decodes the low order address bits A0-A7 resulting in 8 unique combinations hex $C40X, $C42X, $C44X, $C46X, etc., through $C4EX.

Three major chips reside on the interface card. One is a 6551, an "Asynchronous Communications Adapter", and two 6522's, "Versatile Interface Adapters". The 6551 is allocated address space $C42X (X being a number between 0 and 15 for internal register addressing). One 6522 is allocated $C40X, and the second one $C44X.

The 6551 handles all serial communication between the Z8 (Servo processor) and the Apple II. The first 6522 (Port 1) monitors the status of three lines necessary for proper communication: these lines are SIO READY, SERVO ERR, and SERVO READY. The other 6522 is not used in "Apple to Servo".

SOFTWARE DESCRIPTION

The "Apple to Servo" program consists of two programs:

SV0020283CMP	(compiled integer basic)
Z8 COMM.OBJ0	(assembly code)

The actual program run is called "SV0020283CMP". This stands for "SERVO-Feb 2 1983 COMPILED". When run it automatically BLOADS "Z8 COMM.OBJ0". All screen functions and user interaction are handled by the Basic code while all high speed communications to the Z8 are handled in assy code.

The source codes for both of these are:

SV0020283	(Integer Basic)
Z8 COMM	(assembly source code)

BASIC CODE DESCRIPTION

To understand "Apple to Servo" we will describe the uncompiled version, "SV0020283". The code is organized so that functions such as RANDOM, INCREMENT, displaying the main menu, and user interactions fall into groups of lines such as 10000-10999.

Here is a breakdown of the functions, starting at the lowest line, 10, and following through to the end, line 31999.

LINES	DESCRIPTION
10-50	Title, Date, Author
60-62	Initialize variables, jump to start of code
100-110	Decimal to hex routine
120-130	Displays 4 bytes on the screen
210-320	Display Main Menu
330-430	Misc. inits & display stat.
440-530	Goto selected command
700-799	Input 4 hex bytes prompts
800-910	Convert one hex byte to decimal
1000-1199	Code for "A: ACCESS"
1500-1999	Code for "C: ACCESS2"
2000-2230	Code for "R: SEND RECAL" and "F: FORMAT RECAL"
3000-3100	Code for "Z: ALTERNATE"
4000-4910	Code for "I: INCREMENT"
5000-5199	Code for "S: RETRY (A)"
5500-5999	Code for "D: RETRY (C)"
6000-6990	Code for displaying ACIA status (not implemented)
8000-8992	Code for "X: RANDOM"
9000-9610	Code for "W: SERVO STAT"
10000-10040	Enter "OFFSET CAPABILITY?"
11000-11999	Code that handles "SEND READ STAT" and displaying offset amounts during INCREMENT, RANDOM, and ALTERNATE

17000-17070	Pokes 4 bytes to assy code for communication And check for SIO READY (See assy routine)
18000-18400	Main communication routine and command count
19000-19520	Error display messages for communication
22000-22010	Clear bottom of screen sub
25000-25310	Code for inversing the selection
30000-30999	Initialize all variables (done at start)

ASSEMBLY CODE DESCRIPTION

The Assembly code consists of seven "subroutines" that are "called" from Basic. They are easy to understand by looking at lines 2 through 8 in the listing. Each line defines an entry point that, when entered, carries out a particular function and then returns to Basic via an "RTS".

For example, when a 5 byte command sequence is sent to the Z8 servo, first, five bytes are poked into locations 4117 through 4121 (done from Basic). Then "COMM" is called from Basic-"CALL 4108". When "COMM" completes, control is returned to Basis by way of an "RTS".

Here are the seven routines and their functions:

JMP SVORDYT	(Servo ready test)
JMP SVOERRT	(SERVO ERROR test)
JMP SIORDYT	(SIO ready test with timeout)
JMP SIORDY	(SIO ready line ready?)
JMP COMM	(Communicate to servo)
JMP XMITCMD	(Transmit command)
JMP GETSTAT	(Get Z8 status)

COMPILING THE PROGRAM (Using the Integer Compiler)

The Integer compiler is used for the purpose of increasing the overall speed of "Apple to Servo". The disk used is titled "Integer Compiler". The file called "Instructions" on the disk should be run. This will explain its use.

Here are some other things to keep in mind:

1. Type "EXEC COMPILE" <RETURN> to start compiler with Integer program in memory.

2. Choose object start address Number "2" for hex $4000.

3. Give the program a name with date such as "SVO020283CMP" (Must be 12 characters or less)

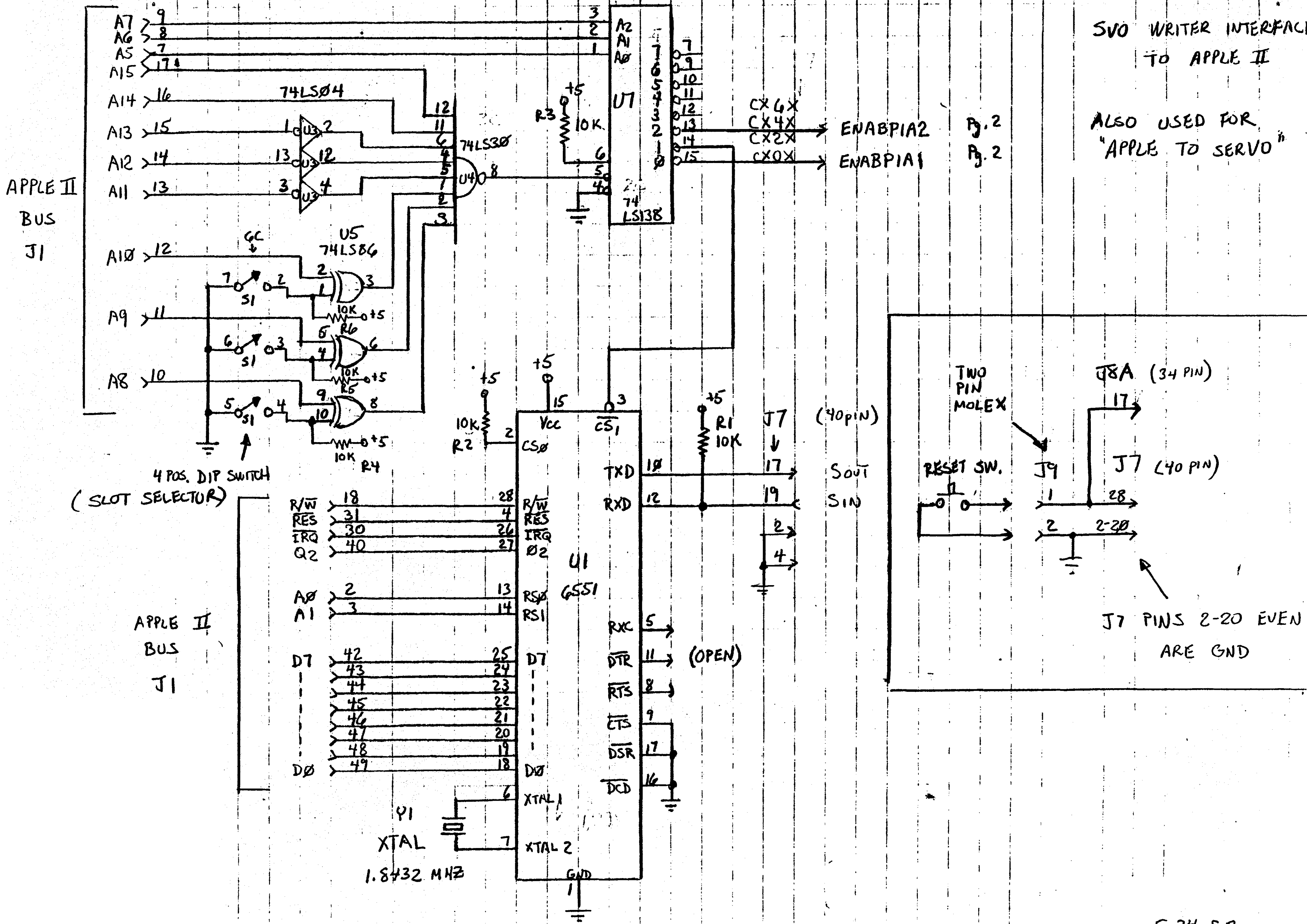
4. The program that loads and runs your compiled program should look like this:

```
10 D$=""         (Ctrl. D here)
20 PRINT D$; "BLOAD SVO020283CMP VAR."
25 PRINT D$; "BLOAD COM SUB"
30 PRINT D$; "BRUN SVO020283CMP OBJ."
40 END
```

Fig. 1

SVO WRITER INTERFACE TO APPLE II

ALSO USED FOR "APPLE TO SERVO"



5-24-82

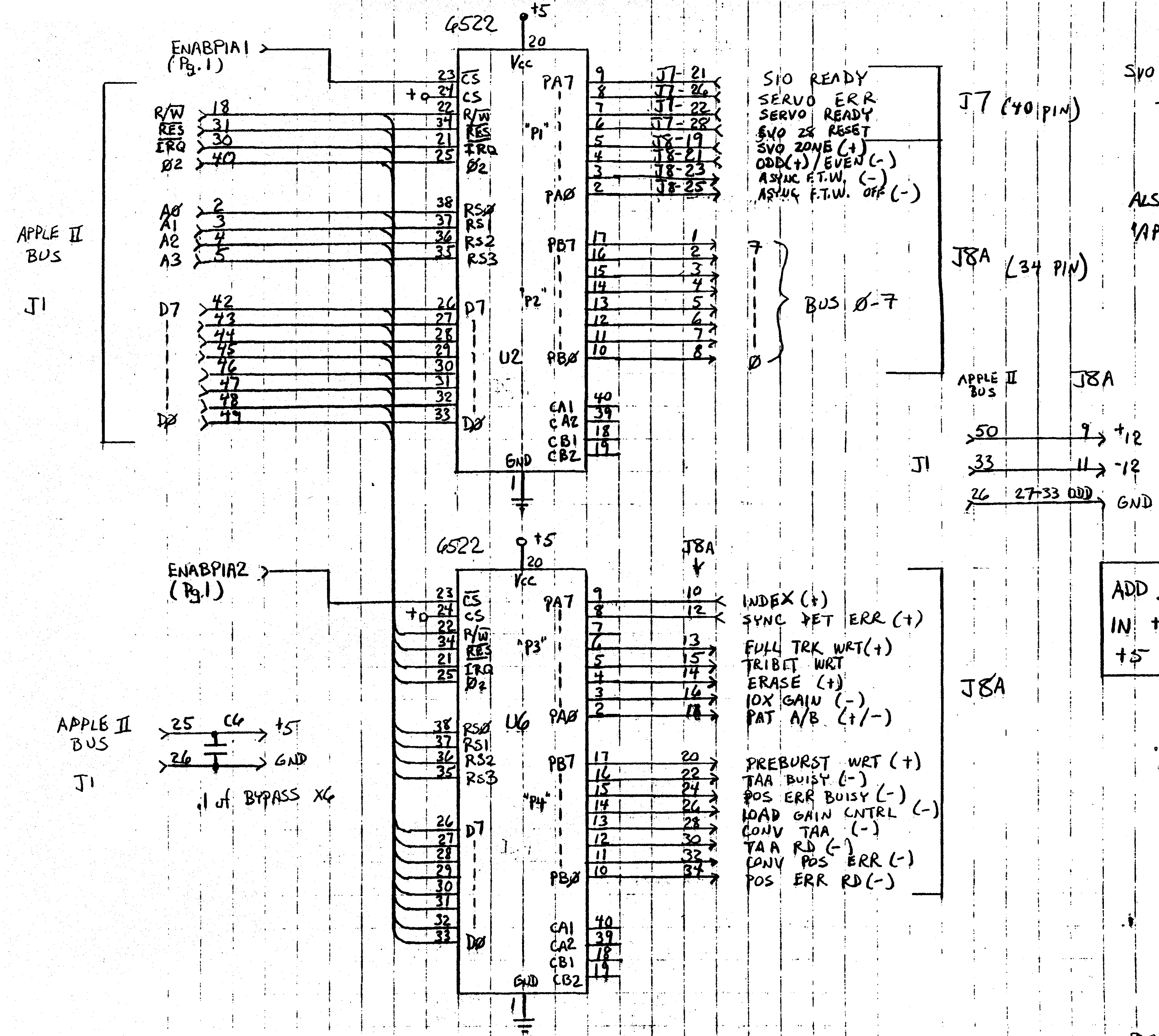
DAN RETZINGER

SVO WRITER INTERFAC

TO APPLE II

ALSO USED FOR

APPLE TO SERVO



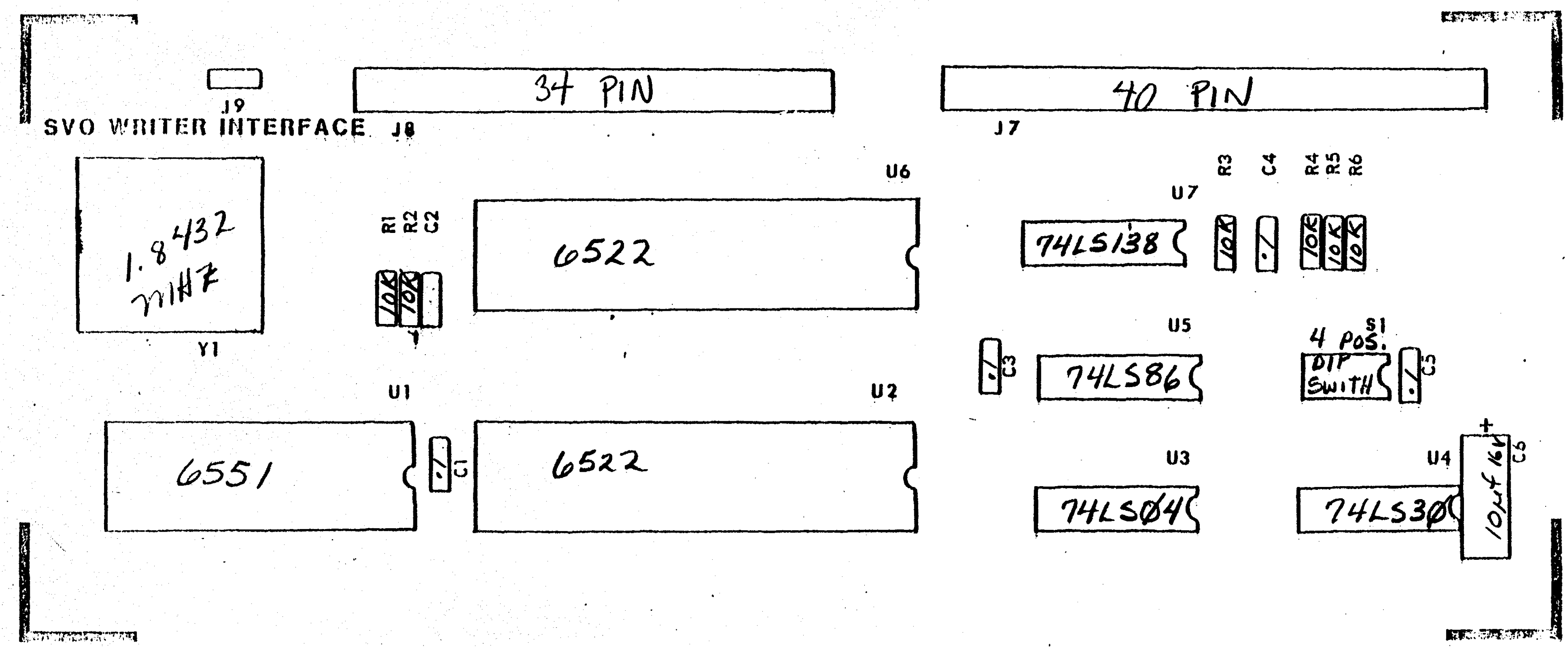
ADD 3 HOLES (.1" SP) IN +5 BUSS FOR +5 POWER IN.

5-24-82

DAN RETZINGER

2/28/83

D. Retzinger



1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26