7

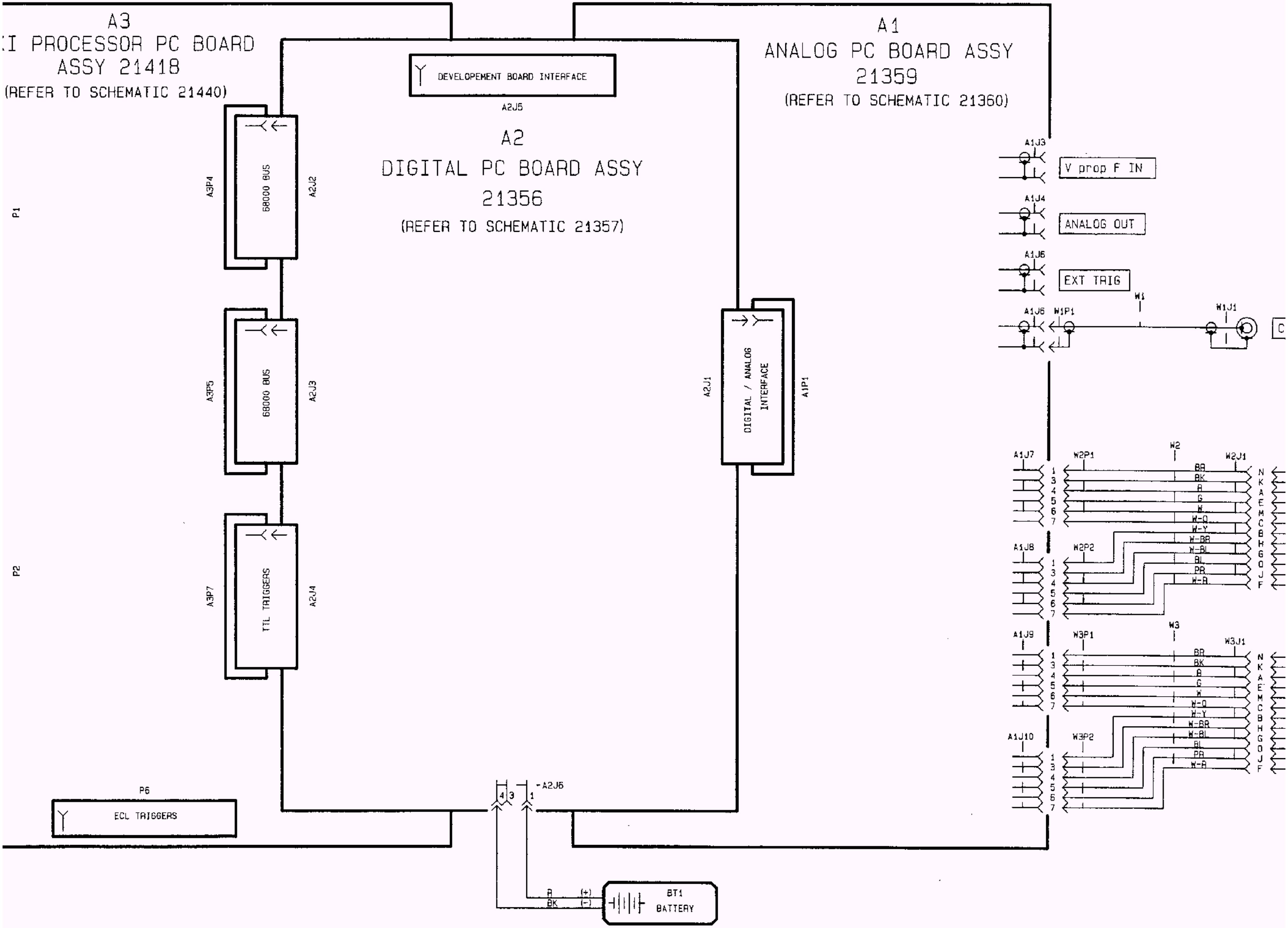
Diagrams

7.1 Introduction

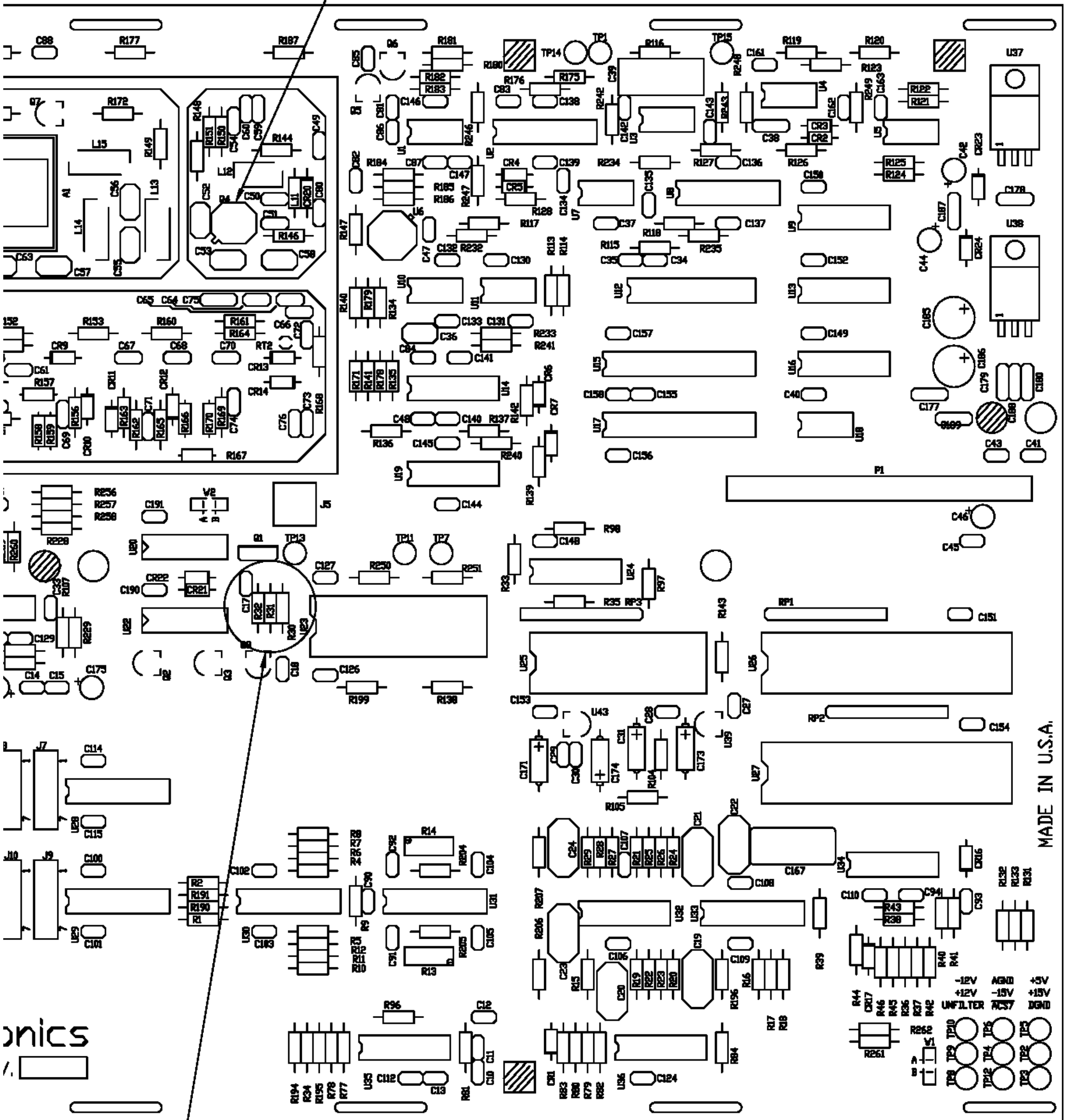
This chapter contains assembly drawings and circuit schematics for the 58542 VXIbus Universal Power Meter.

Parts lists for all assemblies are contained in Chapter 6.

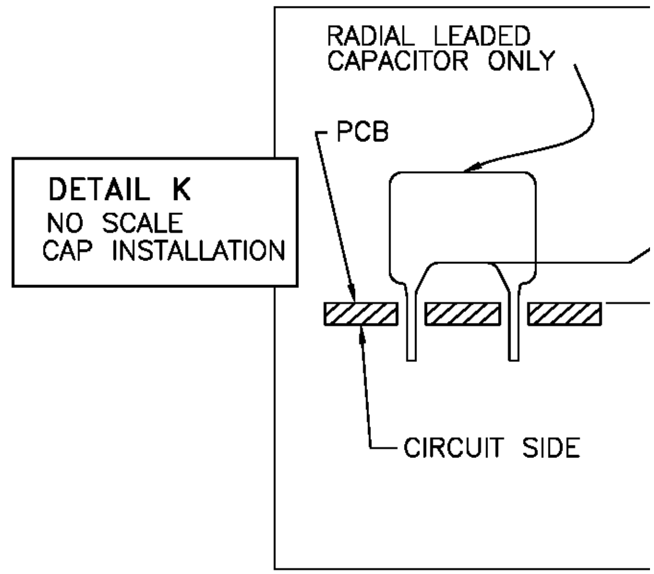
11" x 17" (Landscape) pages follow continuing Chapter 7 of the 58542 Publication

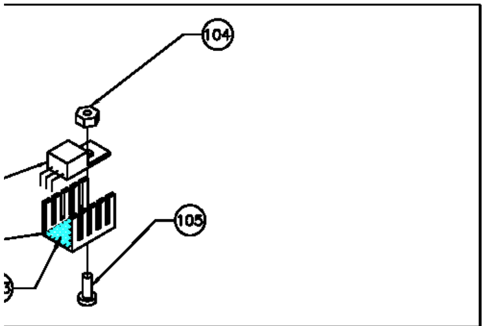

A3
I PROCESSOR PC BOARD
ASSY 21418
(REFER TO SCHEMATIC 21440)
A2
DIGITAL PC BOARD ASSY
21356
(REFER TO SCHEMATIC 21357)
A1
ANALOG PC BOARD ASSY
21359
(REFER TO SCHEMATIC 21360)
DEVELOPEMENT BOARD INTERFACE
A2J5
68000 BUS
68000 BUS
TTL TRIGGERS
A3P4
A3P5
A3P7
A2J2
A2J3
A2J4
P1
P2
P6
ECL TRIGGERS
DIGITAL / ANALOG INTERFACE
A2J1
A1P1
A1J3
V prop F IN
A1J4
ANALOG OUT
A1J5
EXT TRIG
A1J6
W1P1
W1
W1J1
A1J7
W2P1
W2
W2J1
A1J8
W2P2
A1J9
W3P1
W3
W3J1
A1J10
W3P2
BR
BK
R
G
W
W-O
W-Y
W-BR
W-BL
BL
PR
W-R
A2J6
BT1
BATTERY

SEE DETAIL H
SHEET 2 (Q4)

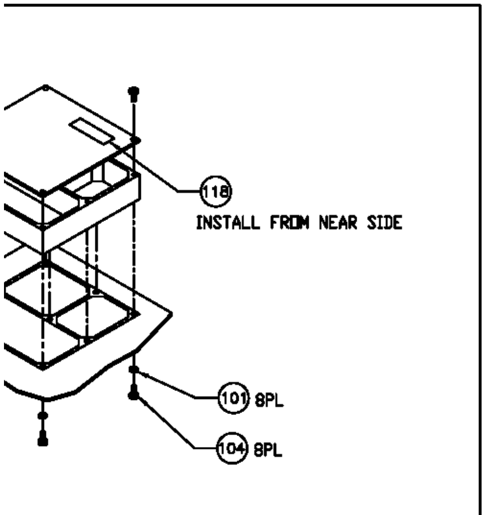


DETAIL G

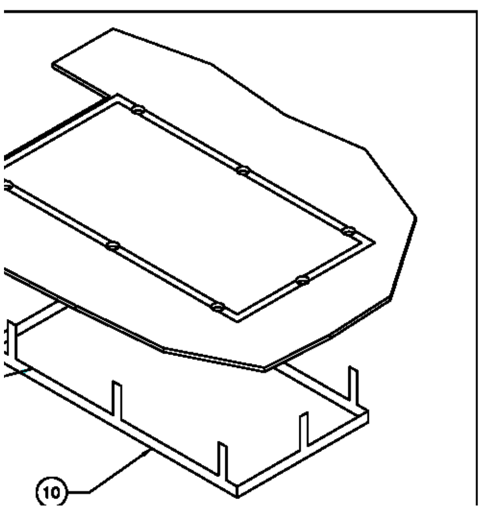


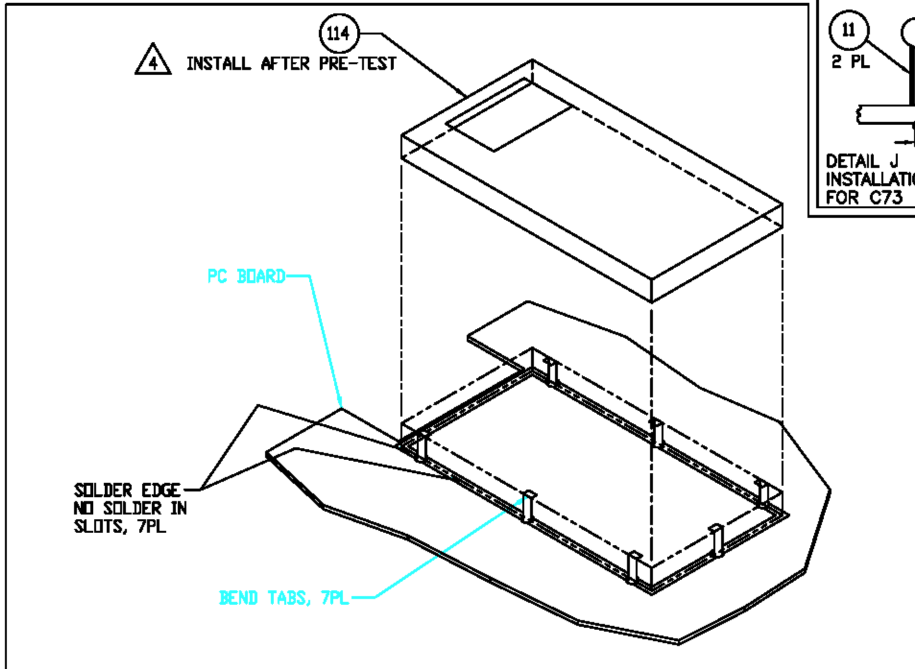


(2), U37, U38 - DETAIL A

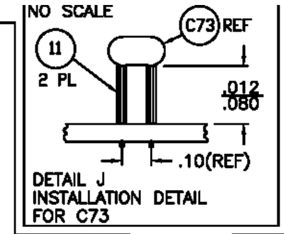


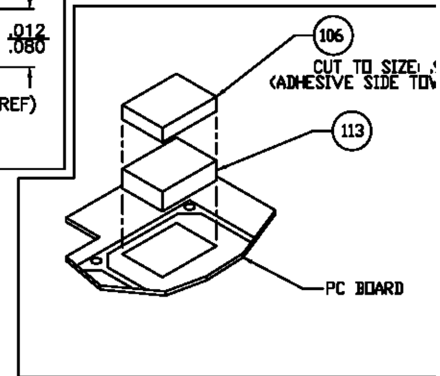
TOR ASSY - DETAIL B SCALE: 1/2



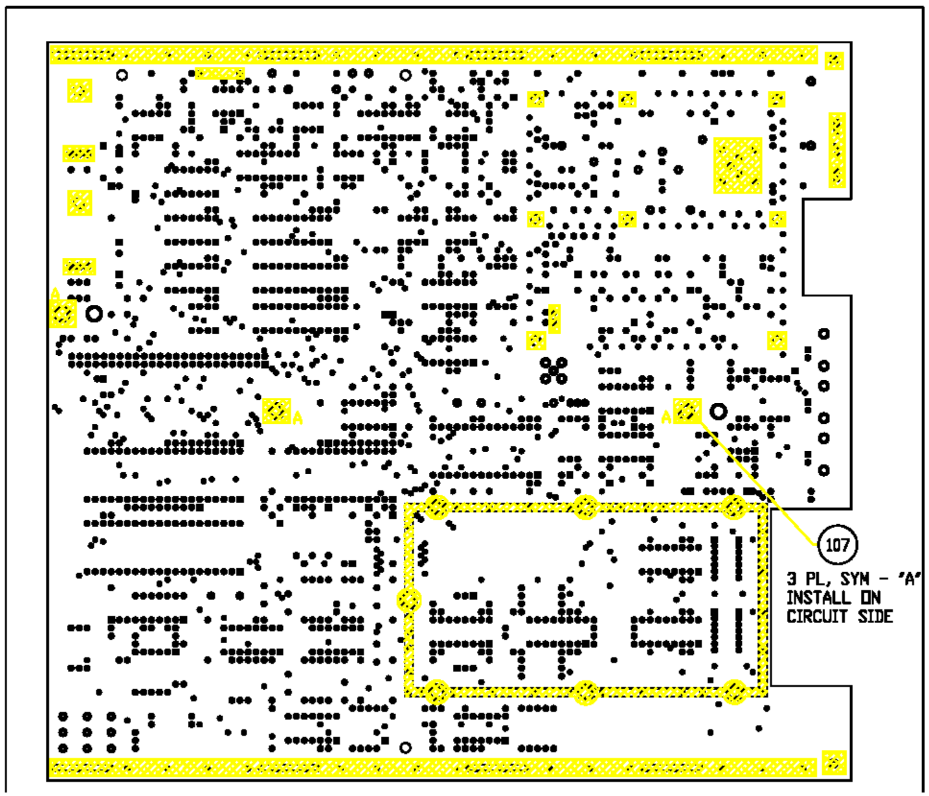


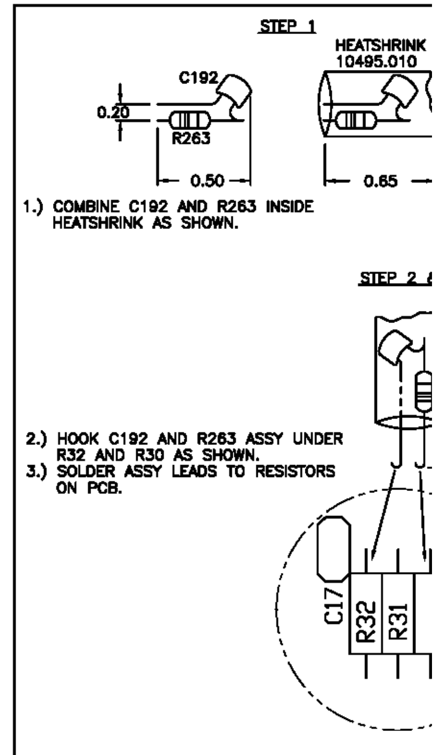
COVER, TOP, SENSOR INPUT - DETAIL D



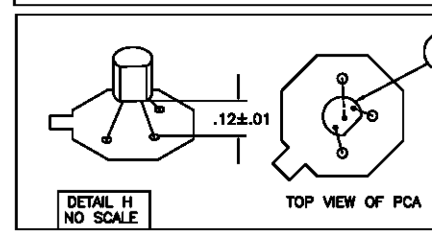


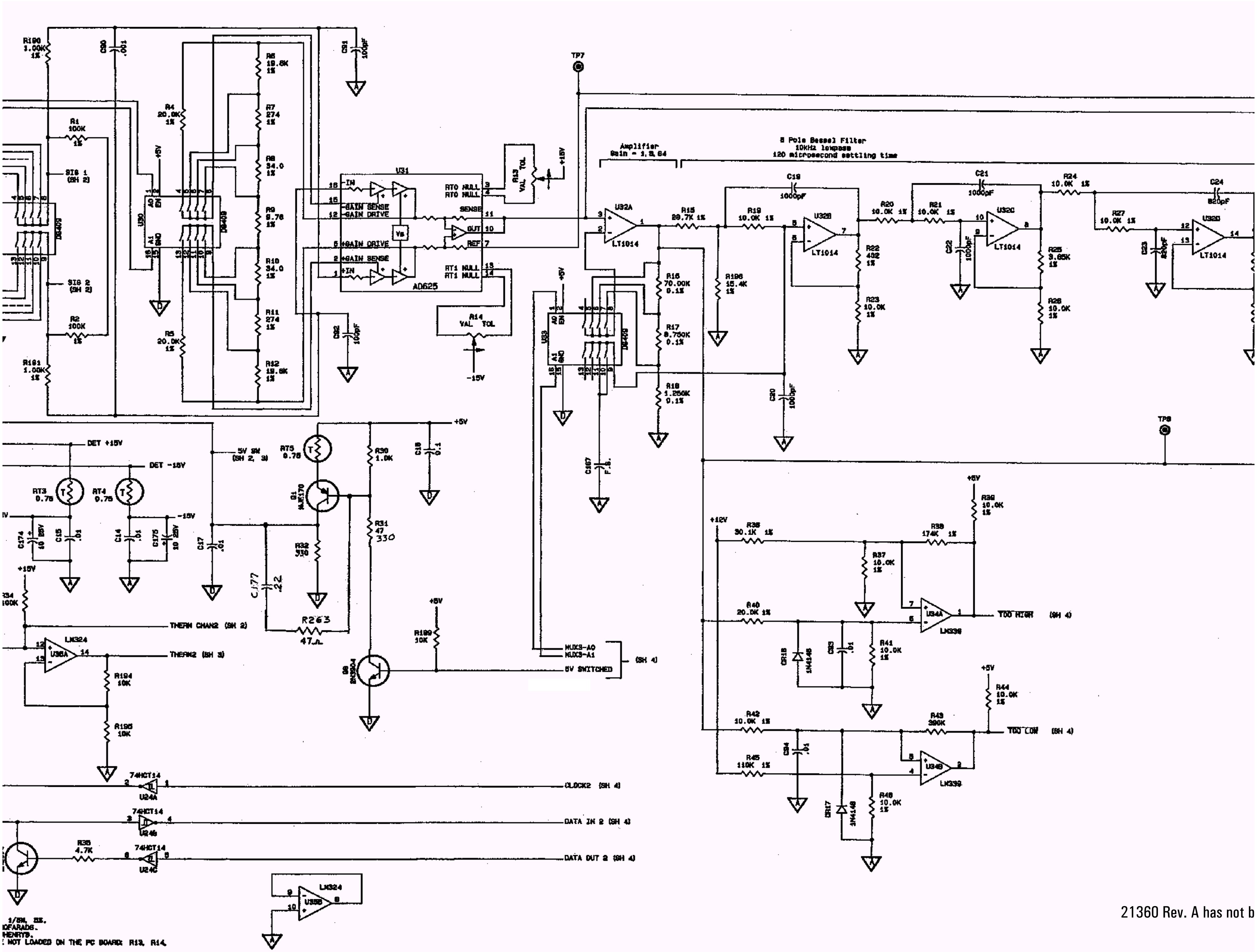
A1 COVER ASSY - DETAIL F



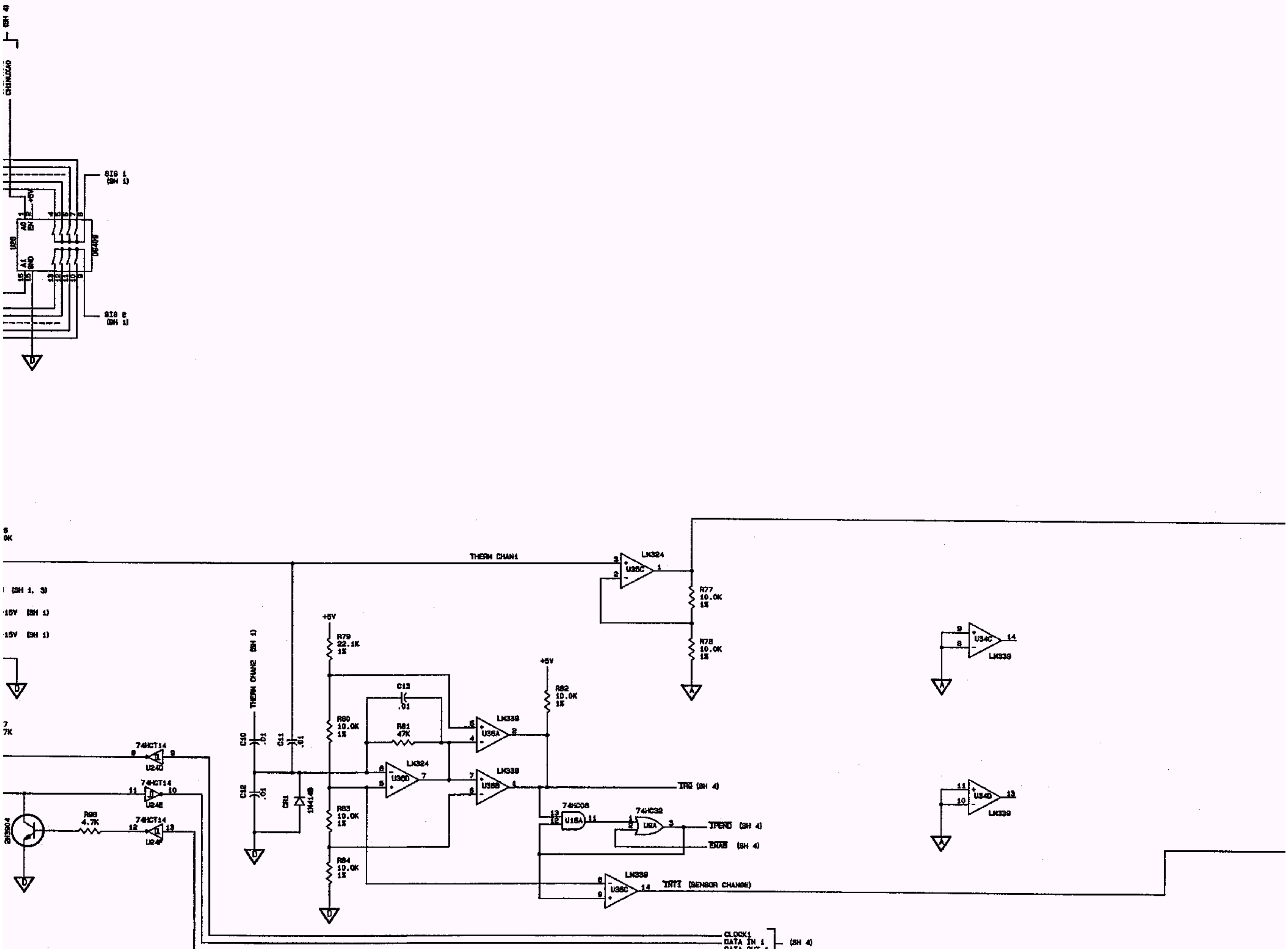


C192 AND R263 ASSY - DETAIL G

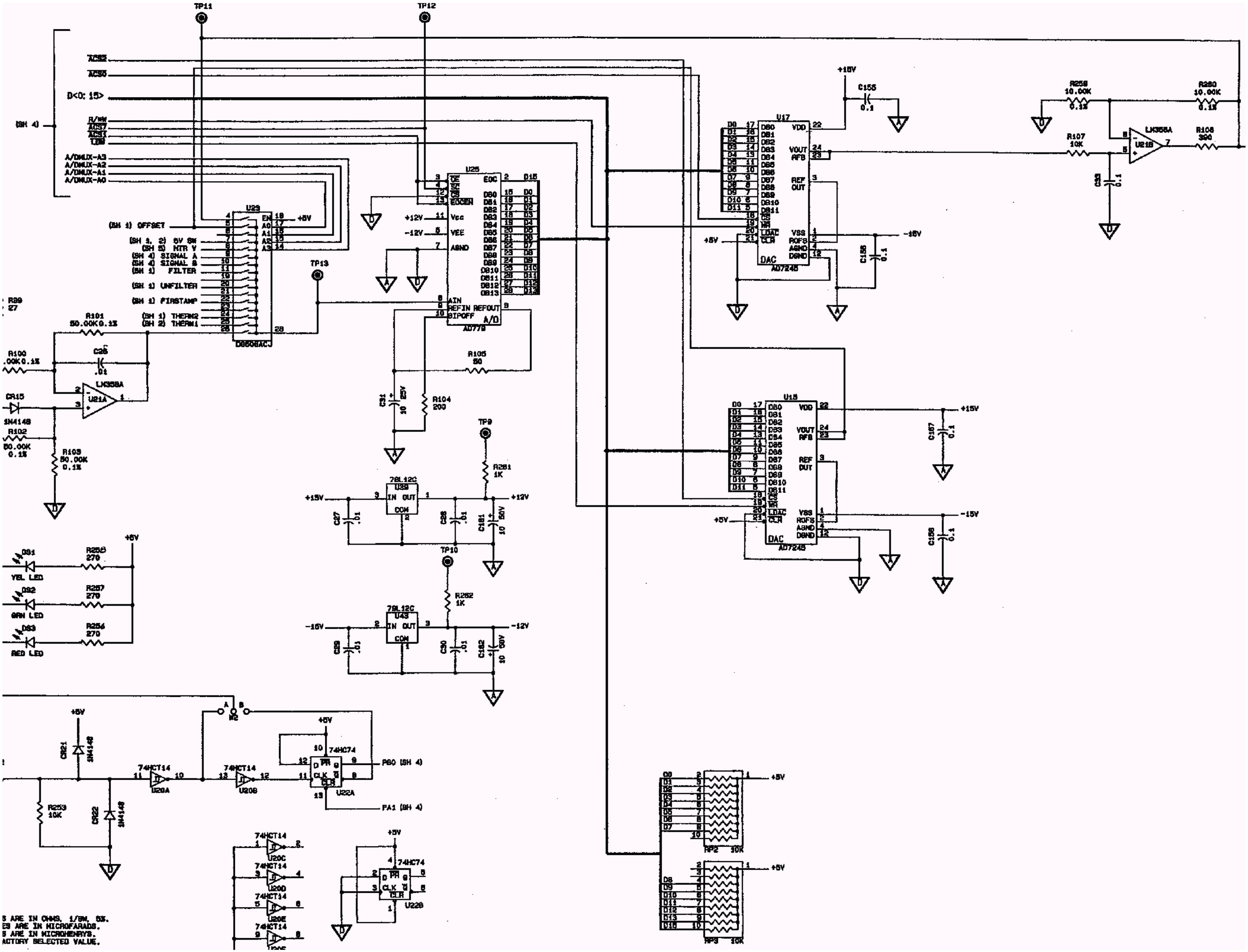




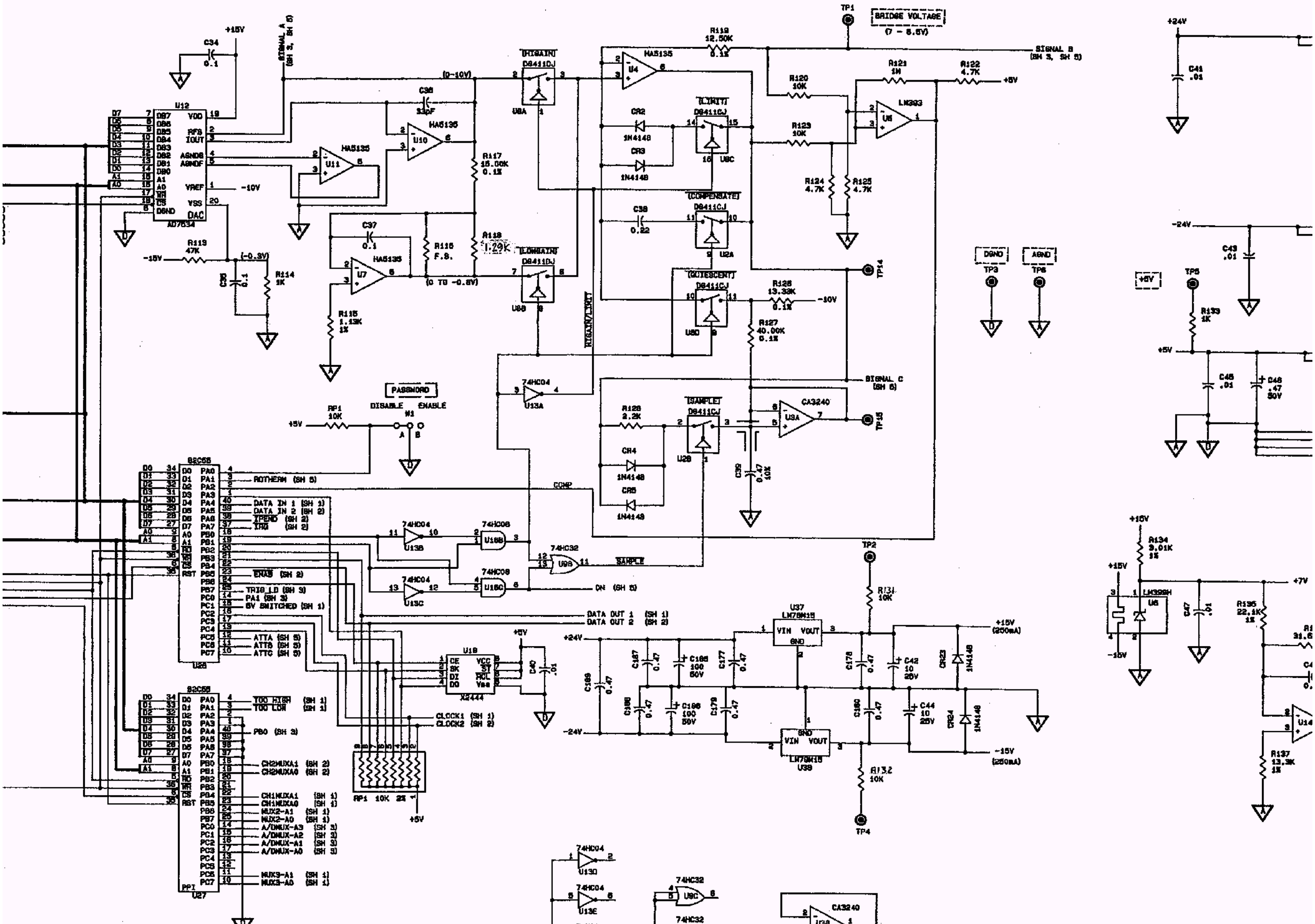
21360 Rev. A has not b



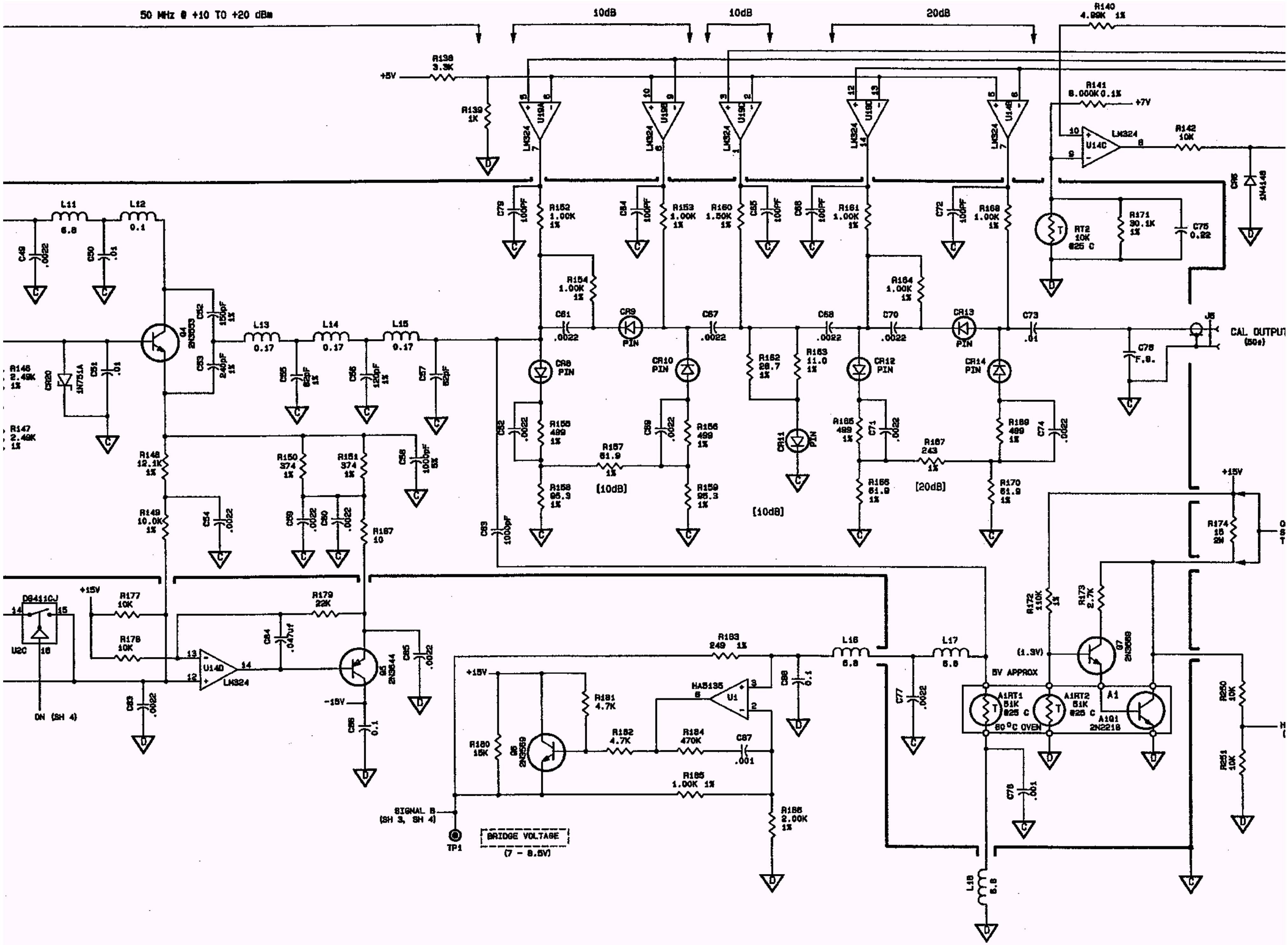
...ES ARE IN OHMS, 1/8W, 5%.
...UES ARE IN MICROFARADS.
...ES ARE IN MICROHENRYS.



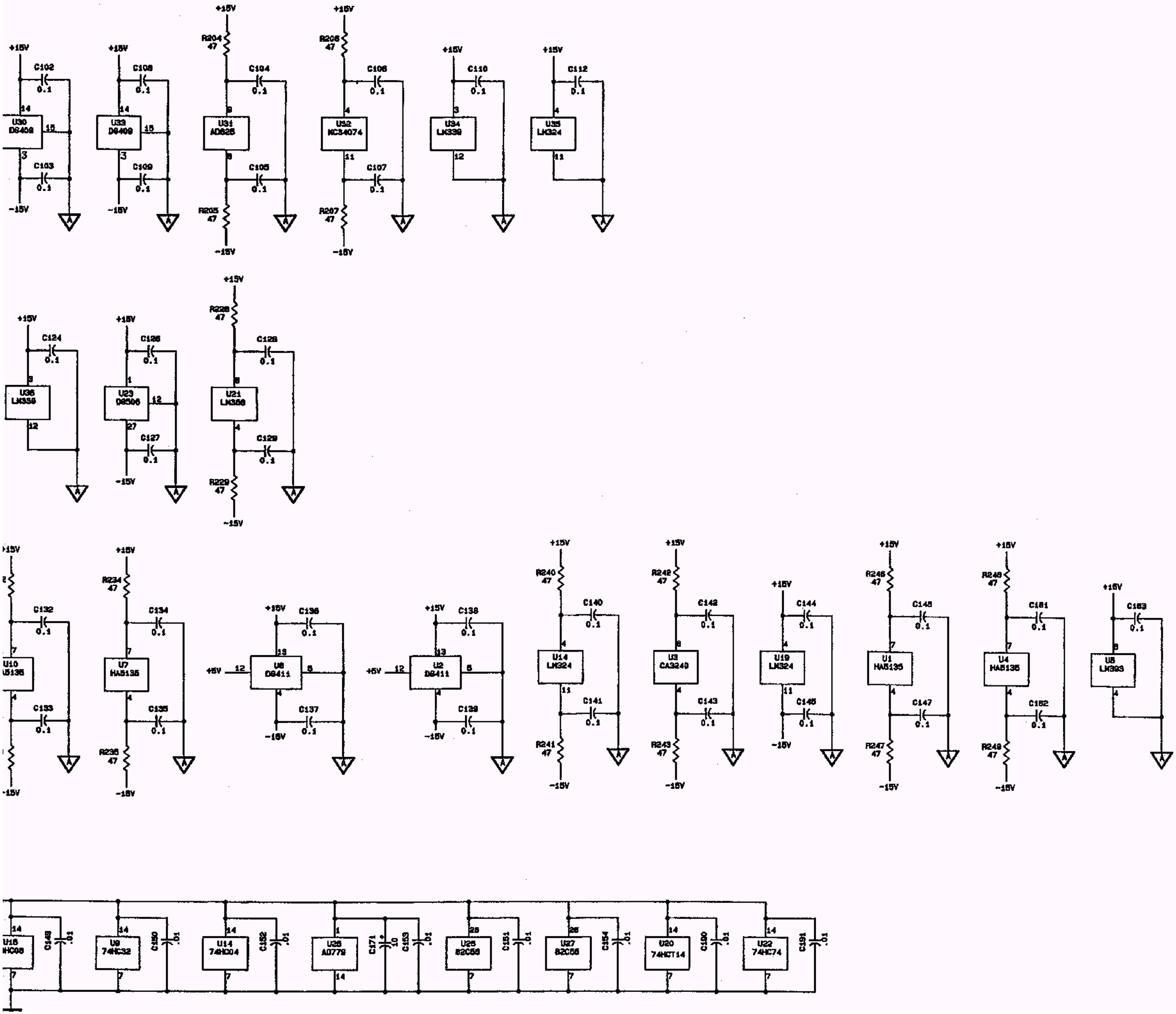
TP11
TP12
TP13
TP9
TP10
ACS2
ACS0
D<0:15>
(SH 4)
R/WR
ACS7
ACS1
LDM
A/DMUX-A3
A/DMUX-A2
A/DMUX-A1
A/DMUX-A0
U29
DG506ACJ
(SH 1) OFFSET
(SH 1, 2) 5V SW
(SH 5) HTR V
(SH 4) SIGNAL A
(SH 4) SIGNAL B
(SH 1) FILTER
(SH 1) UNFILTER
(SH 1) FIRSTAMP
(SH 1) THERM2
(SH 2) THERM1
U25
AD779
A/D
EOC
D15
AIN
REFIN
REFOUT
BIPOFF
AGND
Vcc
VEE
+12V
-12V
+5V
+15V
-15V
U17
U18
AD7245
DAC
VDD
VOUT
RFB
REF OUT
VSS
ROFS
AGND
DGND
CS
WR
LDAC
CLR
C155
0.1
C156
C157
C158
R258
10.00K
0.1%
R260
10.00K
0.1%
R107
10K
R108
390
C33
LM358A
U21B
U21A
R101
50.00K 0.1%
R100
C25
.01
CR15
1N4148
R102
50.00K
0.1%
R103
50.00K
0.1%
R39
27
R105
50
R104
200
C31
10
25V
R261
1K
R262
1K
78L12C
U39
79L12C
U43
IN OUT
COM
C27
C28
C161
C29
C30
C162
50V
DS1
YEL LED
R259
270
DS2
GRN LED
R257
270
DS3
RED LED
R256
270
CR21
CR22
1N4148
R253
10K
74HCT14
U20A
U20B
U20C
U20D
U20E
U20F
74HC74
U22A
U22B
PB0 (SH 4)
PA1 (SH 4)
W2
RP2
RP3
10K
S ARE IN OHMS, 1/8W, 5%.
ES ARE IN MICROFARADS.
S ARE IN MICROHENRYS.
ACTORY SELECTED VALUE.



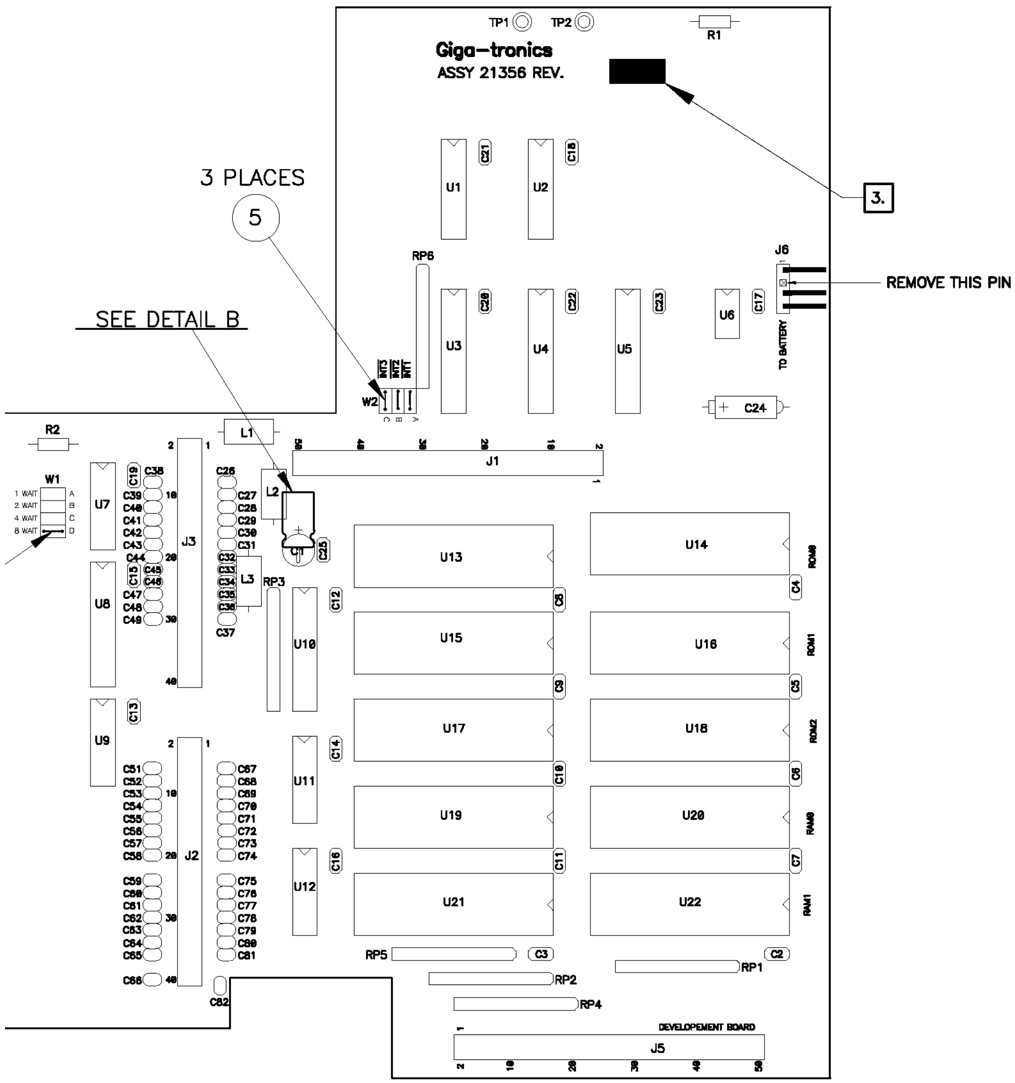
RE IN OHMS, 1/8W, 5%.
ARE IN MICROFARADS.
RE IN MICROHENRYS.



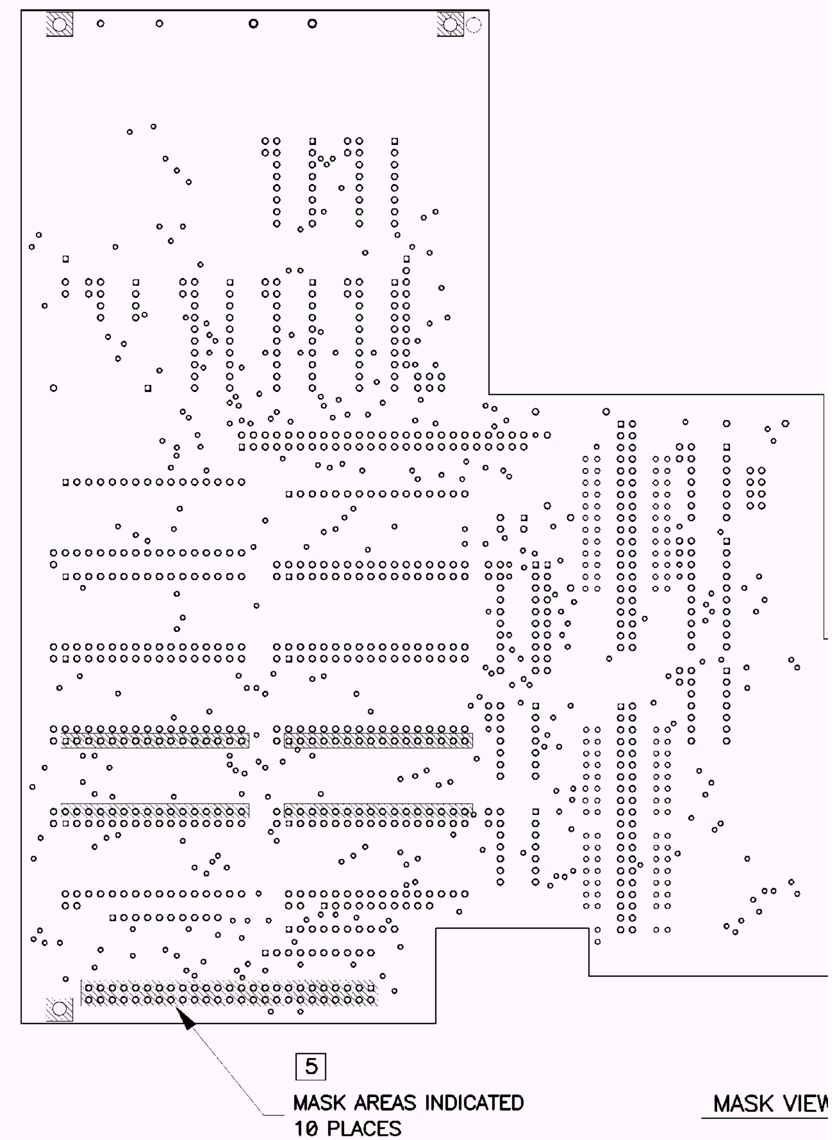
50 MHz @ +10 TO +20 dBm
10dB
10dB
20dB
R138 3.3K
+5V
R139 1K
U19A LM324
U19B LM324
U19C LM324
U19D LM324
U14B LM324
R140 4.99K 1%
R141 8.000K 0.1%
+7V
U14C LM324
R142 10K
CR6 1N4148
L11 6.8
L12 0.1
C49 .0022
C50 .01
Q4 2N3553
C52 150pF 1%
C53 240pF 1%
L13 0.17
L14 0.17
L15 0.17
C55 82pF 1%
C56 120pF 1%
C57 82pF
R146 2.49K 1%
R147 2.49K 1%
CR20 1N751A
C51 .01
C79 100PF
C84 100PF
C85 100PF
C86 100PF
C72 100PF
R152 1.00K 1%
R153 1.00K 1%
R160 1.50K 1%
R161 1.00K 1%
R168 1.00K 1%
RT2 10K @25 C
R171 30.1K 1%
C75 0.22
R154 1.00K 1%
R164 1.00K 1%
C61 .0022
CR9 PIN
C67 .0022
C68 .0022
C70 .0022
CR13 PIN
C73 .01
J5
CAL OUTPUT (50Ω)
C78 F.S.
CR8 PIN
CR10 PIN
R162 28.7 1%
R163 11.0 1%
CR12 PIN
CR14 PIN
CR11 PIN
C62 .0022
R155 499 1%
C69 .0022
R156 499 1%
R157 61.9 1%
[10dB]
R158 95.3 1%
R159 95.3 1%
[10dB]
R165 499 1%
C71 .0022
R167 243 1%
R169 499 1%
C74 .0022
R166 61.9 1%
[20dB]
R170 61.9 1%
R148 12.1K 1%
R149 10.0K 1%
C54 .0022
R150 374 1%
R151 374 1%
C58 1000pF 5%
C59 .0022
C60 .0022
R187 10
C63 1000pF
+15V
R174 15 2W
DG411CJ
U2C
ON (SH 4)
R177 10K
R178 10K
R179 22K
U14D LM324
C64 .047uf
Q5 2N3644
C65 .0022
-15V
C66 0.1
C83 .0022
R183 249 1%
L16 6.8
L17 6.8
5V APPROX
R172 110K 1%
R173 2.7K
(1.3V)
Q7 2N3569
HA5135 U1
C96 0.1
C77 .0022
A1RT1 51K @25 C
A1RT2 51K @25 C
80°C OVEN
A1
A1Q1 2N2218
R250 10K
R251 10K
R181 4.7K
R180 15K
Q6 2N3569
R182 4.7K
R184 470K
C87 .001
R185 1.00K 1%
R186 2.00K 1%
C76 .001
SIGNAL B (SH 3, SH 4)
TP1
BRIDGE VOLTAGE (7 - 8.5V)
L18 6.8



NOTES:
1. RESISTOR VALUES
2. CAPACITOR VALUES
3. INDUCTOR VALUES
4. F.S. DENOTES FA

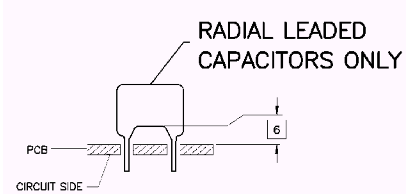


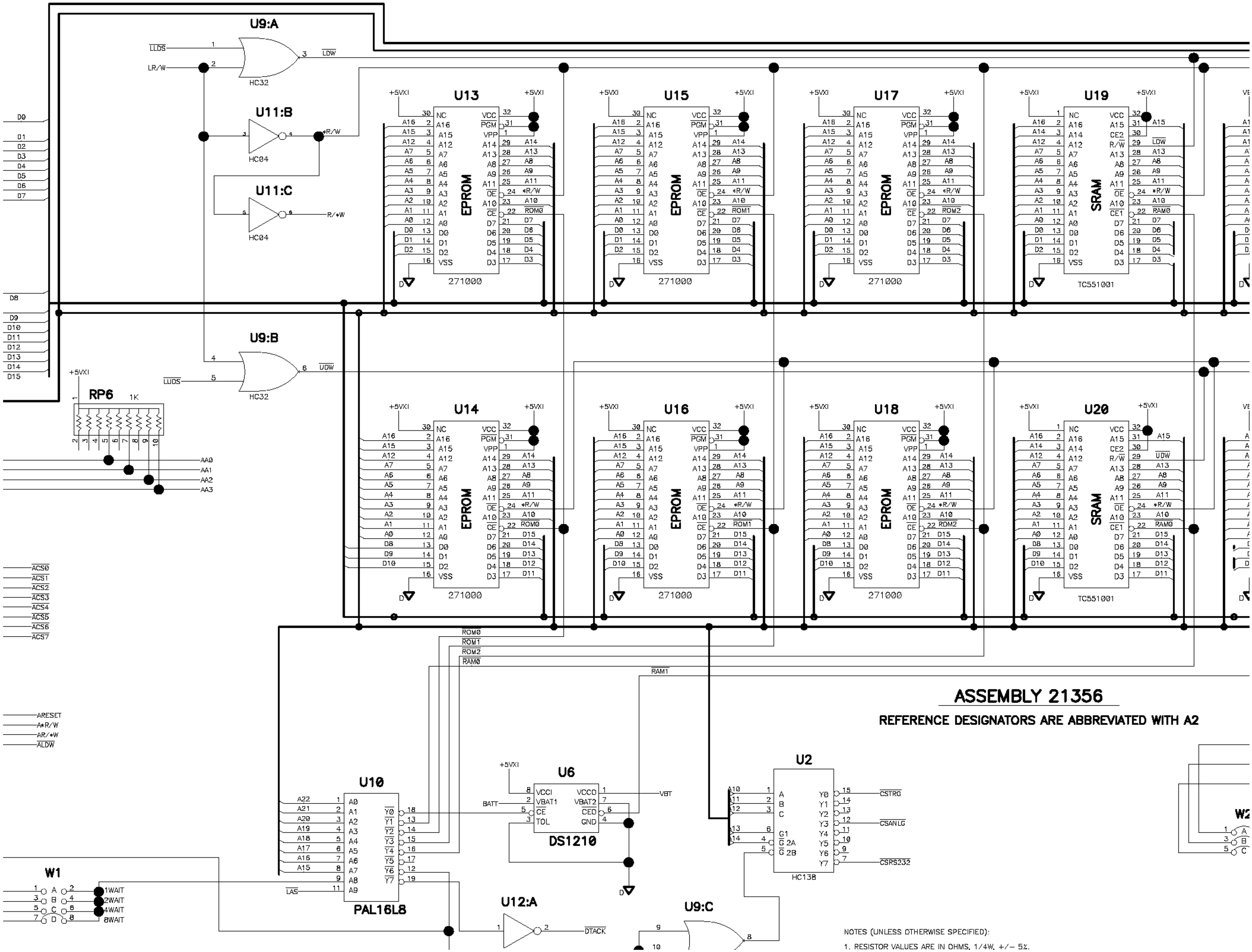
COMPONENT SIDE



MASK VIEW

BOTTOM SIDE OF PC



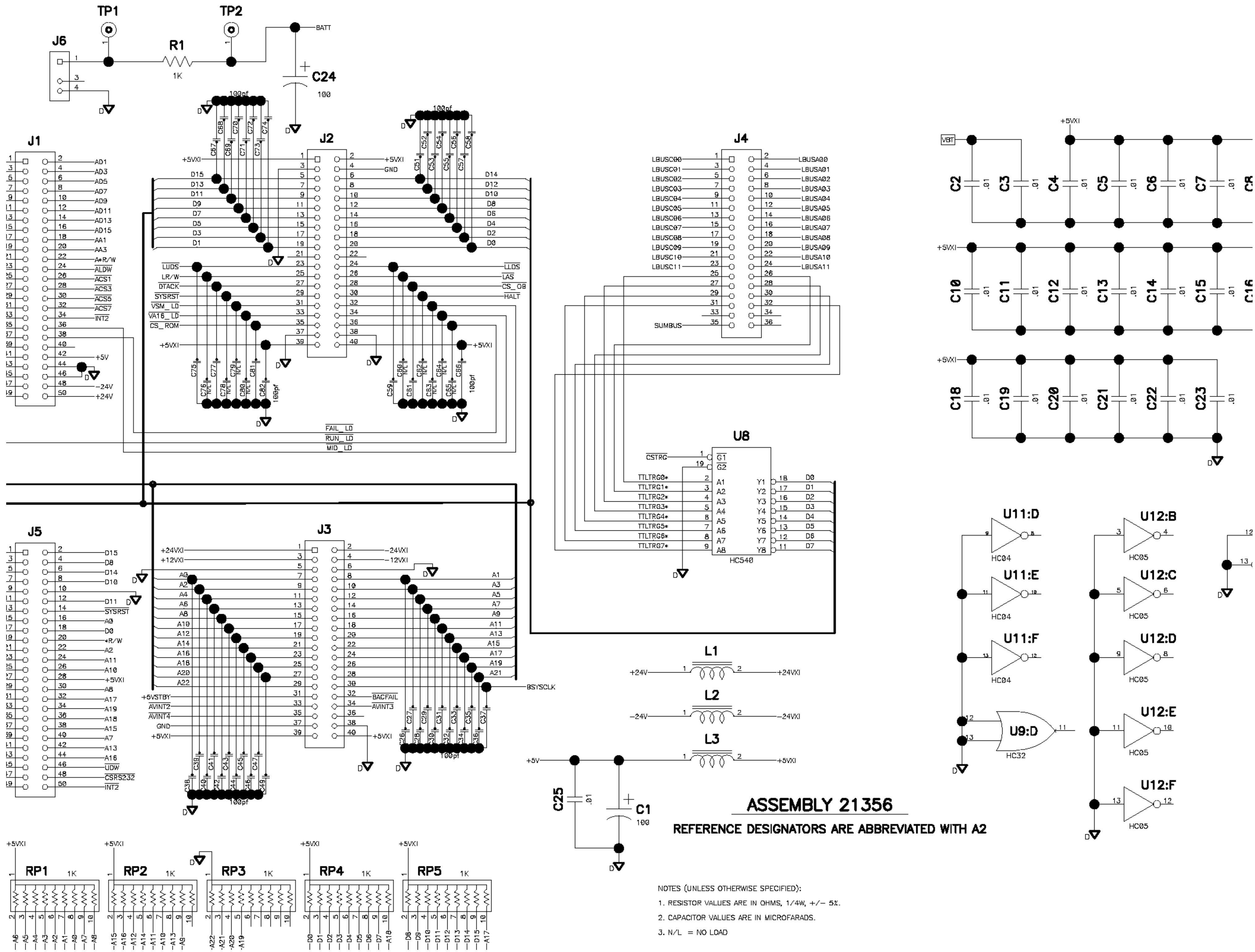


ASSEMBLY 21356

REFERENCE DESIGNATORS ARE ABBREVIATED WITH A2

NOTES (UNLESS OTHERWISE SPECIFIED):

1. RESISTOR VALUES ARE IN OHMS, 1/4W, +/- 5%.



ASSEMBLY 21356

REFERENCE DESIGNATORS ARE ABBREVIATED WITH A2

NOTES (UNLESS OTHERWISE SPECIFIED):

1. RESISTOR VALUES ARE IN OHMS, 1/4W, +/- 5%.
2. CAPACITOR VALUES ARE IN MICROFARADS.
3. N/L = NO LOAD

A

Program Examples

A.1 Introduction

This appendix contains examples of the various operating programs in the 58542 VXIbus Universal Power Meter.

The examples shown in this chapter are written in HTBasic™ format. Different languages will use different commands, but the string sent or received will always be the same. In HTBasic, the OUTPUT command sends a string to the GPIB bus. The number or variable after the word OUTPUT is the GPIB address of the instrument.

A.2 Sensor Calibration Examples

Sensors must be calibrated to the meter before performing measurements. Only one sensor at a time can be calibrated to a given 58542 VXI Power Meter input. That is, each time a sensor is calibrated to the meter using the power sweep calibration, previous calibrations for that meter input are voided automatically. Thus, assuring that the measurement system is always performing under a valid sensor calibration.

A.2.1 Sensor Calibration Example 1

The following example performs power sweep calibration of the power sensor connected to input number 2 and sends back a pass/fail flag upon completion using CAL2?

```
410   ASSIGN @Pwr_mtr to 70101
420   CLEAR @Pwr_mtr
430   WAIT 1
440        PRINT Connect the sensor to the power sensor cable.
450        INPUT Then connect the sensor to the calibrator and hit ENTER.,Dmy
460        PRINT Calibrating Sensor 2...
470        OUTPUT@Pwr_mtr,CAL2?
480        ENTER@Pwr_mtr,Pass_cal
490        IF Pass_cal=0 THEN PRINT Calibration Passed
500        IF Pass_cal=1 THEN
510        PRINT Calibration FAILED, Sensor 2 on calibrator?
560   ELSE
570        IF Pass_cal<>0 THEN PRINT Strange Response to CAL2?.Clear output Queue?
580   END IF
590   !
```

A.2.2 Sensor Calibration Example 2

The power meter measurement functions are operated to tight tolerances during sensor calibration. If the sensor passes calibration, it is a good health check of the 58542 VXI Power Meter. In general, the power meter and sensors are operating properly if they pass the calibration process. For instance, the calibration process will fail if the sensor detector elements have been damaged. Also see the self test functions, which are initiated using the *TST command or the DIAG: commands (See Table 5-1).

The following program uses the CAL? query format to perform power sweep calibration of a power sensor and sends back a pass/fail flag upon completion. Completion of the calibration function is monitored via service requests using the *OPC, operation complete command.

```
330  !
340  !##################################
350  !
350  !   CALIBRATE THE SENSORS
370  !
380  !##################################
390  ASSIGN@Pwr_mtr to 70101
400  CLEAR@Pwr_mtr
410  WAIT 1
420  OUTPUT@Pwr_mtr;*CLS          ! Clear status registers and error queue buffer
425  WAIT 0.3
430  OUTPUT@Pwr_mtr;*ESE 1        ! Standard Event Status Enable bit 0, Operation Complete
440  OUTPUT@Pwr_mtr;*SRE 32       ! Service Request Enable Status Byte bit 5, Standard
     ! Event Status Register
450  !
460  ON INTR 7 GOSUB Spoll_intr
470  ENABLE INTR 7;2
480  !
490  ! Calibration Routine
500  !
510  INPUT Connect the sensor to the calibrator. Then press ENTER.,Dmy
520  PRINT Calibrating Sensor 1...
525  Srq_flag=0
530  OUTPUT@Pwr_mtr;CAL1?;*OPC    ! For zeroing substitute CAL1:ZERO? for CAL1?
540  !
550  !
560  ! Now wait for srq
570  !
580  !
590  !
600  WHILE Srq_flag=0
610  END WHILE
620  !
630  ! Enter Pass/Fail Flag, This comes with query format, CAL1?, rather than command format, CAL1.
640  !
650  ENTER@Pwr_mtr;Pass_cal
660  IF Pass_cal=0 THEN PRINT Calibration Passed
670  IF Pass_cal=1 THEN
680  OUTPUT@Pwr_mtr;SYST:ERR?
690  ENTER@Pwr_mtr;Err_msg$       ! Reading an error message clears it.
700  PRINT The power meter reports the following error.,ERR_msg$
710  PRINT Calibration FAILED, sensor on calibrator?
720  ELSE
730  IF Pass_cal<>0 THEN PRINT Strange Response to CAL1?. Other Event Status Register Bits Too?
740  END IF
750  !
760  STOP
770  !
780  Spoll_intr: !
790  !
800  State=SPOLL(@Pwr_mtr)
810  PRINT SPOLL INTR:;State
```

```
820    Srq_flag=1
830    ENABLE INTR 7;2
840    !
850    RETURN
860    !
870    !
880    !    END OF SENSOR CAL
890    !###################################
900    !
910    END
```

A.2.3 Sensor Calibration Example 3

The following program uses the CAL format to perform power sweep calibration of a power sensor. Completion of the calibration function is monitored via service requests using the *OPC, operation complete command. The Standard Event Status Register is used to report pass/fail by asserting bit 3, value 8, the Device Dependent Error bit.

```
320  !
330  ! ###################################
340  !
330  !   CALIBRATE THE SENSORS
360  !
370  !      ###################################
380  !
390  ASSIGN@Pwr_mtr to 70101
400  CLEAR@PWR_mtr
410  WAIT 1
420  OUTPUT@Pwr_mtr;*CLS              ! Clear status registers and error queue buffer
425  WAIT 0.3
430  OUTPUT@Pwr_mtr;*ESE 1            ! Standard Event Status Enable bit 0, Operation  complete
440  OUTPUT@Pwr_mtr;*SRE 32           ! Service Request Enable Status Byte bit 5,
                                      ! Standard Event Status Register
450  !
460  ON INTR 7 GOSUB Spoll_intr
470  ENABLE INTR 7;2
480  !
490  ! Calibration Routine
500  !
510  INPUT Connect the sensor to the calibrator. Then press ENTER.,Dmy
520  PRINT Calibrating Sensor 1
525  Srq_flag=0
530  OUTPUT@Pwr_mtr;CAL1;*OPC         ! For zeroing substitute CAL1:ZERO for CAL1
540  OUTPUT@Pwr_mtr;*OPC              ! If not sent with previous line, send *OPC here
550  !
560  ! Now wait for srq
570  !
580  !
590  !
600  WHILE Srq_flag=0
610  END WHILE
620  !
630  ! Enter Error Reporting From Standard Event Status Register
640  !
650  OUTPUT@Pwr_mtr;*ESR?
660  ENTER@Pwr_mtr;Esr
670  IF Esr=0 THEN PRINT Operation was not completed. Also, no errors.
680  IF Esr=1 THEN
690  PRINT Operation Completed. No errors reported during CAL execution.
700  ELSE
710  IF Pass_cal<>0 THEN PRINT Operation completed. An error has occurred or power
     was just turned on.
720  OUTPUT@Pwr_mtr;SYST:ERR?
730  ENTER@Pwr_mtr;Err_msg$           ! Reading an error message clears it
```

```
740 PRINT The power meter reports the following error.,Err_msg$
750 PRINT Calibration FAILED, sensor on calibrator?
760 PRINT Esr
770 !
780 END IF
790 !
800 !
810 STOP
820 !
830 Spoll_intr:!
840 !
850 State=SPOLL(@Pwr_mtr)
860 PRINT SPOLL INTR:;State
870 Srq_flag=1
880 ENABLE INTR 7;2
890 !
900 RETURN
910 !
920 !
930 !  END OF SENSOR CAL
940 ! ############################
950 !
960 END
```

A.2.4 Sensor Calibration Example 4

The following program prompts the user to connect a return loss bridge for calibration and attach an open or short to the bridge test port. It then performs a power sweep calibration and sends back a pass/fail flag upon completion.

```
410 ASSIGN @Pwr_mtr to 70101
420 CLEAR @Pwr_mtr
430 WAIT 1
440 PRINT Connect the 80503 Precision Return Loss Bridge to the input #2 power sensor cable.
445 PRINT Connect the open or short calibration connector to the bridge test port.
450 INPUT Then use an adapter to connect the bridge input port to the calibrator and hit ENTER.,Dmy
460 PRINT Calibrating Sensor 2...
470 OUTPUT @Pwr_mtr;CAL2?
480 ENTER @Pwr_mtr;Pass_cal
490 IF Pass_cal=0 THEN PRINT Calibration Passed
500 IF Pass_cal=1 THEN
510 PRINT Calibration FAILED, Bridge connected to calibrator?
560 ELSE
570 IF Pass_cal<>0 THEN PRINT Strange Response to CAL2?. Clear output Queue?
580 END IF
590 !
```

A.2.5 Sensor Calibration Example 5

The following program prompts the user to remove the high power attenuator from a high power sensor main housing and connect the main housing to the power sweep calibrator port. It then performs a power sweep calibration and sends back a pass/fail flag upon completion.

```
410   ASSIGN @Pwr_mtr to 70101
420   CLEAR @Pwr_mtr
430   WAIT 1
440   PRINT Connect the 80325 50W Power Sensor to the input #2 power sensor cable.
445   PRINT Remove the high power attenuator or leave it attached to the measurement port on the DUT
450   INPUT Then connect the high power sensor main housing to the calibrator and hit ENTER.,Dmy
460   PRINT Calibrating Sensor 2...
470   OUTPUT @Pwr_mtr;CAL2?
480   ENTER @Pwr_mtr;Pass_cal
490   IF Pass_cal=0 THEN PRINT Calibration Passed. Re-attach the high power attenuator immediately!
500   IF Pass_cal=1 THEN
510   PRINT Calibration FAILED, Main sensor housing connected to calibrator?
560   ELSE
570   IF Pass_cal<>0 THEN PRINT Strange Response to CAL2?. Clear output Queue?
580   END IF
590   !
```

A.3 Sensor Zeroing Examples

A.3.1 Sensor Zeroing Example 1

The following example requests that the operator turn off the signal source or disconnect the sensor from the source prior to zeroing. It is preferable to disable the source and leave the sensor attached to the measurement port for proper zeroing. If the source cannot be disabled, use a switch or attach a connector to the grounded metal near the DUT measurement port. DO NOT attach the sensor to the power meter's calibrator port. The zeroing offset process will account for the DUT's ground plane noise and thermal emf, not the power meter's.

```
410   ASSIGN @Pwr_mtr to 70101
420   CLEAR @Pwr_mtr
430   WAIT 1
440   INPUT Disable the source's RF output or attach it to a grounded connector. Then hit ENTER.,Dmy
450   PRINT Zeroing Sensor 1
460   OUTPUT @Pwr_mtr;CAL1:ZERO?
470   ENTER @Pwr_mtr;Pass_zer
480   IF Pass_zer=0 THEN PRINT Zeroing Passed
490   IF Pass_zer=1 THEN
500   PRINT Zeroing FAILED, Source turned off?
550   ELSE
560   IF Pass_zer<>0 THEN PRINT Strange Response to CAL1:ZERO?. Clear output Queue?
570   END IF
580   !
```

A.3.2 Sensor Zeroing Example 2

The following program turns off the signal source, zeros the power sensor, reports a pass/fail indicator upon completion, and turns the source back on.

```
400   ASSIGN @Pwr_mtr to 70101
410   ASSIGN @Source to 70102
420   CLEAR @Pwr_mtr
430   WAIT 1
440   OUTPUT @Source RF0! Turns source output power off (<-90 dBm).
450   PRINT Zeroing Sensor 1
460   OUTPUT @Pwr_mtr;CAL1:ZERO?
470   ENTER @Pwr_mtr;Pass_zer
480   IF Pass_zer=0 THEN PRINT Zeroing Passed
490   IF Pass_zer=1 THEN
500   Ques$=Zeroing FAILED, Source turned off?
510   OUTPUT @Pwr_mtr;SYST:ERR?
520   ENTER @Pwr_mtr;Err_msg$! Reading an error message clears it.
530   PRINT Err_msg$
540   PRINT Ques$
550   ELSE
560   IF Pass_zer <>0 THEN PRINT Strange Response to CAL1:ZERO?. Clear output Queue?
570   END IF
580   !
590   OUTPUT @Source RF1! Turns source output power back on.
```

Zeroing is recommended whenever performing critical final power measurements in the bottom 15 to 20 dB of a power sensors dynamic range. Zeroing removes zero drift error from the measurement. At higher power levels, zero drift is typically insignificant when compared to the other sources of error such as source/sensor mismatch uncertainty.

A.3.3 Error Control Examples

Selected basic SCPI syntax and execution errors apply to these commands.

The following command will operate properly.

```
460    OUTPUT @Pwr_mtr;CAL1?              ! Calibrate channel 1
```

The following command is not a legal command. The 58542 only has two channels.

```
460    OUTPUT @Pwr_mtr;CAL3?              ! Calibrate channel 3
```

If the user asks for error information, the error reporting is as follows:

```
510    OUTPUT @Pwr_mtr;SYST:ERR?
520    ENTER @Pwr_mtr;Err_msg$            ! Reading an error message clears it.
530    PRINT Err_msg$
```

Program Output:

```
-7,Invalid Error Number
```

The following command is not a legal command. The command is mis-typed.

```
460    OUTPUT @Pwr_mtr;CALS?              ! Calibrate channel 3
```

If the user asks for error information, the error reporting is as follows.

```
510    OUTPUT @Pwr_mtr;SYST:ERR?
520    ENTER @Pwr_mtr;Err_msg$            ! Reading an error message clears it.
530    PRINT Err_msg$
```

Program Output:

```
-113,Undefined Header;CALS
Device-specific errors include the following.
-300,Device-specific error;No Sensor
```

The following examples demonstrate the conditions for various errors.

```
380    ASSIGN @Pwr_mtr to 70101
390    CLEAR @Pwr_mtr
400    WAIT 1
401    INPUT Disconnect Sensor From Cable for Error Demonstration. Then hit ENTER.,Dmy
410    PRINT Calibrating Sensor 1
420    OUTPUT @Pwr_mtr;CAL1?
430    ENTER @Pwr_mtr;Pass_cal
440    IF Pass_cal=0 THEN PRINT Calibration Passed
450    IF Pass_cal=1 THEN
460    Ques$=Calibration FAILED, sensor on calibrator?
470    OUTPUT @Pwr_mtr;SYST:ERR?
480    ENTER @Pwr_mtr;Err_msg$            ! Reading an error message clears it.
490    PRINT Err_msg$
500    PRINT Ques$
520    ELSE
521    IF Pass_cal<>0 THEN PRINT Strange Response to CAL1?. Clear output Queue?
530    END IF
531    !
532    !
533    STOP
```

Program Output:

```
Calibrating Sensor 1
```

-300, Device-specific error;No Sensor
Calibration FAILED, sensor on calibrator?

```
410   ASSIGN @Pwr_mtr to 70101
420   CLEAR @Pwr_mtr
440   WAIT 1
450   INPUT Disconnect the sensor from the cable 1 second after hitting ENTER. Hit ENTER.,Dmy
460   PRINT Calibrating Sensor 1
470   OUTPUT @Pwr_mtr;CAL1?
480   ENTER @Pwr_mtr;Pass_cal
490   IF Pass_cal=0 THEN PRINT Calibration Passed
500   IF Pass_cal=1 THEN
510   Ques$=Calibration FAILED, sensor on calibrator?
520   OUTPUT @Pwr_mtr;SYST:ERR?
530   ENTER @Pwr_mtr;Err_msg$            ! Reading an error message clears it.
540   PRINT Err_msg$
550   PRINT Ques$
560   ELSE
570   IF Pass_cal<>0 THEN PRINT Strange Response to CAL1?. Clear output Queue?
580   END IF
590   !
600   !
610   STOP
```

Program Output:
Calibrating Sensor 1
-300, Device-specific error;Sensor not connected to calibrator
Calibration FAILED, sensor on calibrator?

```
410   ASSIGN @Pwr_mtr to 70101
420   CLEAR @Pwr_mtr
440   WAIT 1
450   INPUT Do not connect the sensor to the calibrator. Hit ENTER.,Dmy
460   PRINT Calibrating Sensor 1
470   OUTPUT @Pwr_mtr;CAL1?
480   ENTER @Pwr_mtr;Pass_cal
490   IF Pass_cal=0 THEN PRINT Calibration Passed
500   IF Pass_cal=1 THEN
510   Ques$=Calibration FAILED, sensor on calibrator?
520   OUTPUT @Pwr_mtr;SYST:ERR?
530   ENTER @Pwr_mtr;Err_msg$            ! Reading an error message clears it.
540   PRINT Err_msg$
550   PRINT Ques$
560   ELSE
570   IF Pass_cal<>0 THEN PRINT Strange Response to CAL1?. Clear output Queue?
580   END IF
590   !
600   !
610   STOP
```

Program Output:
Calibrating Sensor 1
-300, Device-specific error;Sensor not connected to calibrator
Calibration FAILED, sensor on calibrator?

```
410   ASSIGN @Pwr_mtr to 70101
420   CLEAR @Pwr_mtr
440   WAIT 1
449   PRINT Connect the sensor to the calibrator.
450   INPUT Remove the sensor 10 seconds after hitting ENTER. Hit ENTER.,Dmy
460   PRINT Calibrating Sensor 1

470   OUTPUT @Pwr_mtr;CAL1?
480   ENTER @Pwr_mtr;Pass_cal
490   IF Pass_cal=0 THEN PRINT Calibration Passed
```

```
500   IF Pass_cal=1 THEN
510         Ques$=Calibration FAILED, sensor on calibrator?
520         OUTPUT @Pwr_mtr;SYST:ERR?
530         ENTER @Pwr_mtr;Err_msg$      ! Reading an error message clears it.
540         PRINT Err_msg$
550         PRINT Ques$
560   ELSE
570               IF Pass_cal<>0 THEN PRINT Strange Response to CAL1?. Clear output Queue?
580         END IF
590         !
600
610         STOP
```

Program Output:

```
      Calibrating Sensor 1
      -300,Device-specific error;Sensor calibration error
      Calibration FAILED, sensor on calibrator?
```

```
410   ASSIGN @Pwr_mtr to 70101
420   CLEAR @Pwr_mtr
430   WAIT 1
440   INPUT Connect the sensor to a source set to about 0dBm. Hit ENTER.,Dmy
450   PRINT Zeroing Sensor 1
460   OUTPUT @Pwr_mtr;CAL1:ZERO?
470   ENTER @Pwr_mtr;Pass_zer
480   IF Pass_zer=0 THEN PRINT Zeroing Passed
490   IF Pass_zer=1 THEN
500         Ques$=Zeroing FAILED, Source turned off?
510         OUTPUT @Pwr_mtr;SYST:ERR?
520         ENTER @Pwr_mtr;Err_msg$! Reading an error message clears it.
530         PRINT Err_msg$
540         PRINT Ques$
550   ELSE
560         IF Pass_zer<>0 THEN PRINT Strange Response to CAL1:ZERO?. Clear output Queue?
570   END IF
580   !
590   !
600   STOP
```

Program Output:

```
      Zeroing Sensor 1
            -300,Device-specific error;Sensor zeroing error
            Zeroing FAILED, Source turned off?
```

```
410   ASSIGN @Pwr_mtr to 70101
420   CLEAR @Pwr_mtr
430   WAIT 1
440   INPUT Disconnect the sensor from the sensor cable. Hit ENTER.,Dmy
450   PRINT Zeroing Sensor 1
460         OUTPUT @Pwr_mtr;CAL1:ZERO?
470         ENTER @Pwr_mtr;Pass_zer
480         IF Pass_zer=0 THEN PRINT Zeroing Passed
490         IF Pass_zer=1 THEN
500         Ques$=Zeroing FAILED, Source turned off?
510         OUTPUT @Pwr_mtr;SYST:ERR?
520         ENTER @Pwr_mtr;Err_msg$! Reading an error message clears it.
530         PRINT Err_msg$
540         PRINT Ques$
550   ELSE
560         IF Pass_zer<>0 THEN PRINT Strange Response to CAL1:ZERO?. Clear output Queue?
570   END IF
580   !
590   !
600   STOP
```

Program Output:

```
Zeroing Sensor 1
-300,Device-specific error;No Sensor
Zeroing FAILED, Source turned off?
```

A.4 Reading Power Measurement Examples

This section gives as quick start in performing measurements. Therefore, the easiest measurement commands are in the first example, MEAS#? in NORMal Mode. This may not be the optimum 58542 configuration for the user's application. There are three example groups, NORMal Mode, SWIFt Mode, and BURSt Mode. The use of the three 58542 measurement data commands, MEAS#?, READ#?, and FETCh#? are explained in detail under the NORMal Mode, which is the meter's power-on default mode.

SWIFt and BURSt Mode measurements use only FETCh#?, not READ#? or MEAS#?. In this getting started section, high speed SWIFt and BURSt Mode examples are few in number and brief in description; detailed information on high speed power measurements can be found in Section A.8 (High Speed Measurements).

A.4.1 Reading Power Measurements Example 1

The following program configures the power meter for single channel operation, and uses the MEAS#? command to perform limited Auto-Configuration and return power measurement data in NORMal Mode. Prior to executing the measurement cycle, the MEAS#? command performs low level configuration functions for averaging and trigger sequence arming automatically. Use MEAS#? to get started performing power measurements quickly. Typically, MEAS#? will satisfy most power measurement needs; however, more advanced power meter users of NORMal Mode may prefer lower level controls of the READ#? command and the higher measurement rates of the FETCh#? command. High speed measurements are performed using the SWIFt and BURSt Modes, not NORMal Mode.

Note that the MEAS#? command's measurement response speed is slower at very low power levels and faster at high power levels. This is due to a chopper stabilization system used in the NORMal Mode (also used in SWIFt Mode, but not BURSt mode) and Auto-Averaging when using SENS#:AVER:TCON REP. At the most sensitive gain ranges, chopper stabilization allows the very small voltages from the power sensor to be accurately measured. Also, at lower power meter gain ranges, Auto-Averaging automatically adds additional averaging to counter the effects of noise in the measurement signal. The extra time required to perform these Auto-Averaged measurements will not be apparent unless switching the default averaging control, SENS#:AVER:TCON MOV, to SENS#:AVER:TCON REP.

When using the MEAS#? command, Auto-Averaging is automatically set to ON. If another part of programming was using manual averaging, be sure to turn Auto-Averaging back off (SENS#:AVER:COUN:AUTO OFF) before exiting the section of the program code using MEAS? When operating in NORMal Mode the user can always use the SENS#:AVER:COUN? query to find out the current averaging value for a particular sensor, and the SENS#:AVER:COUN ### command to set a higher or lower averaging value.

```
260   ASSIGN @Pwr_mtr to 70101
270   CLEAR @Pwr_mtr
280         WAIT 1
290   OUTPUT @Pwr_mtr;CALC1:MODE NORM! Can ONLY send Configuration Commands in Normal Mode!!
300   OUTPUT @Pwr_mtr;CALC1:POW 1          ! Channel 1 configured to measure sensor 1 power
310   !
320   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
                                           ! Applies Cal Factor for 1.44 GHz to measurement data.
330   !
340   FOR I=1 TO 10
350         OUTPUT @Pwr_mtr;MEAS1?         ! Measures Power at sensor.
360         ENTER @Pwr_mtr;Rdg
370         PRINT Rdg
380   NEXT I
390   !
510   END
```

A.4.2 Reading Power Measurements Example 2

If sending the MEAS#? command, two configuration changes occur. These are listed as shown below. To perform measurements without changing the configuration, use the READ#? command.

Command/Configuration Item	MEAS#? Auto-Setting
SENS#:A VER:COUN:AUTO	ON
INIT:CONT	OFF

No other items than the above will change configuration when sending the MEAS#? command.

The following example illustrates that Auto-Averaging changes the active averaging number. Since Auto-Averaging was activated through use of the MEAS#? command, the averaging value currently in use when Auto-Averaging is set to OFF is taken as the new averaging number. This is likely to differ from the averaging number in effect before the MEAS#? command was sent. If this is an undesirable effect, remember to read the averaging number using SENS#:AVER:COUN? and store the value for post Auto-Averaging output at the conclusion of the MEAS#? power meter reading measurement routine.

```
260   ASSIGN @Pwr_mtr to 70101
270   CLEAR @Pwr_mtr
290   WAIT 1
300   OUTPUT @Pwr_mtr;CALC1:MODE NORM
310   OUTPUT @Pwr_mtr;SENS1:AVER:COUN:AUTO OFF
320   OUTPUT @Pwr_mtr;SENS1:AVER:COUN:AUTO?
330   ENTER @Pwr_mtr;A$
340   PRINT A$,1 is AUTO ON
350   OUTPUT @Pwr_mtr;SENS1:AVER:COUN 16
360   OUTPUT @Pwr_mtr;SENS1:AVER:COUN?
370   ENTER @Pwr_mtr;A$
380   PRINT A$,16 MEANS AVERAGING #=16 ACCEPTED.
390   OUTPUT @Pwr_mtr;INIT:CONT ON
400   OUTPUT @Pwr_mtr;INIT:CONT?
410   ENTER @Pwr_mtr;A$
420   PRINT A$,1 is INIT:CONT ON
430   OUTPUT @Pwr_mtr;CALC1:POW 1              ! Channel 1 configured to measure sensor 1 power
440   !
450   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
          ! Applies Cal Factor for 1.44 GHz.
460   !
470   FOR I=1 TO 10
480       OUTPUT @Pwr_mtr;MEAS1?               ! Measures Power at sensor.
490       ENTER @Pwr_mtr;Rdg
500       PRINT Rdg
510   NEXT I
520   !
530   OUTPUT @Pwr_mtr;SENS1:AVER:COUN:AUTO?
540   ENTER @Pwr_mtr;A$
550   PRINT A$,1 is AUTO ON
560   OUTPUT @Pwr_mtr;INIT:CONT?
570   ENTER @Pwr_mtr;A$
580   PRINT A$,1 is INIT:CONT ON
590   OUTPUT @Pwr_mtr;SENS1:AVER:COUN:AUTO OFF
600   OUTPUT @Pwr_mtr;SENS1:AVER:COUN:AUTO?
610   ENTER @Pwr_mtr;A$
620   PRINT A$,1 is AUTO ON
630       OUTPUT @Pwr_mtr;SENS1:AVER:COUN?
640       ENTER @Pwr_mtr;A$
650       PRINT A$,Value not equal to 16 means reset by Auto-Averaging during MEAS#?
760   !
770   END
```

A.4.3 Reading Power Measurements Example 3

The following program configures the power meter for dual channel operation and uses the MEAS#? command to return power measurement data in NORMal Mode. Please note that measurement speed per channel decreases slightly in NORMal mode when two channels, versus only a single channel, are connected and calibrated. In NORMal Mode, single channel measurement rates are slightly faster than two channel measurement rates. If using CW power sensors, also please note that power measurements are not performed at specifically the exact, simultaneous instant in time. In NORMal Mode there will be a short time delay of about 3 ms between the two channels' sample points, about two and a half orders of magnitude faster than traditional CW power meters. In BURSt and SWIFT mode this time is about 1 ms. If the user needs to guarantee that power sampling occurs on both channels within a smaller interval of time, this can be accomplished to within about 2% of the Sample Delay time plus 2 ns using the Series 80350A Peak Power Sensors and triggering both sensors at the same time. The example below operates with either Peak or CW power sensors.

```
260   ASSIGN @Pwr_mtr to 70101
270   CLEAR @Pwr_mtr
280   WAIT 1
290   OUTPUT @Pwr_mtr;CALC1:MODE NORM  ! Can ONLY send Configuration Commands in Normal Mode!!
300   OUTPUT @Pwr_mtr;CALC1:POW 1          ! Channel 1 configured to measure sensor 1 power
310   OUTPUT @Pwr_mtr;CALC1:UNIT W         ! Transmitter output power in Watts.
320   OUTPUT @Pwr_mtr;CALC2:RAT 1,2        ! Transmitter gain stays in default, dBm.
330   !
340   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
          ! Applies Cal Factor for 1.44 GHz to sensor 1 data.
350   OUTPUT @Pwr_mtr;SENS2:CORR:FREQ 0.96E9
          ! Applies Cal Factor for 960 MHz to sensor 2 data.
360   !
370   FOR I=1 TO 10
380       OUTPUT @Pwr_mtr;MEAS1?;MEAS2?    ! Measures Power at sensors.
390       ENTER @Pwr_mtr;Chn_pow 1
400       ENTER @Pwr_mtr;Chn_1rat2
410       PRINT Chn_pow1,Chn_1rat2
420   NEXT I
430   !
550   END
```

A.4.4 Reading Power Measurements Example 4

The following program configures the power meter for single channel operation and uses the READ#? command to return power measurement data in NORMal Mode. Using the READ#? command will not change power meter configuration items under any circumstances. Note that INIT:CONT ON is illegal when using READ#?. When continuous trigger arming is desired, use INIT:CONT ON with the FETCh#? command if desired. Also, use TRIG:SOUR IMM when using the READ#? command.

```
280   ASSIGN @Pwr_mtr to 70101
290   CLEAR @Pwr_mtr
300   WAIT 1
310   OUTPUT @Pwr_mtr;CALC1:MODE NORM! Can ONLY send Configuration Commands in Normal Mode!!
320   OUTPUT @Pwr_mtr;TRIG:SOUR IMM       ! TRIG:SOUR must be IMM when using READ#?
330   OUTPUT @Pwr_mtr;INIT:CONT OFF       ! INIT:CONT arming must be OFF when using READ#?
340   !
350   OUTPUT @Pwr_mtr;SENS2:AVER:COUN 1! Sets averaging to manual and one sample per reading.
360   OUTPUT @Pwr_mtr;CALC1:POW 2          ! Channel 1 configured to measure sensor 2 power
370   !
380   OUTPUT @Pwr_mtr;SENS2:CORR:FREQ 17.4E9
                                           ! Applies Cal Factor for 17.4 GHz to measurement data.
390   !
400   FOR I=1 TO 10
410   OUTPUT @Pwr_mtr;READ2?               ! Measures Power at sensor.
420       ENTER @Pwr_mtr;Rdg
430       PRINT Rdg
440   NEXT I
450   !
650   END
```

A.4.5 Reading Power Measurements Example 5

READ#? measurement response speed is slower at very low power levels and faster at high power levels. This is due to a chopper stabilization system used in the NORMal Mode which is also used in the SWIFt Mode, but not the BURSt mode. At the most sensitive gain ranges, chopper stabilization allows the very small voltages from the power sensor to be accurately measured.

The following program configures the power meter for dual channel operation and uses the READ#? command to return power measurement data in NORMal Mode.

```
260   ASSIGN @Pwr_mtr to 70101
270   CLEAR @Pwr_mtr
280   WAIT 1
290   OUTPUT @Pwr_mtr;CALC1:MODE NORM! Can ONLY send Configuration Commands in Normal Mode!!
300   OUTPUT @Pwr_mtr;CALC1:POW 1            ! Channel 1 configured to measure sensor 1 power
310   OUTPUT @Pwr_mtr;CALC1:UNIT W           ! Transmitter output power in Watts.
320   OUTPUT @Pwr_mtr;CALC2:RAT 1,2          ! Transmitter gain stays in default, dBm.
330   !
340   OUTPUT @Pwr_mtr;TRIG:SOUR IMM          ! TRIG:SOUR must be IMM when using READ#?
350   OUTPUT @Pwr_mtr;INIT:CONT OFF          ! INIT:CONT arming must be OFF when using READ#?
360   !
370   OUTPUT @Pwr_mtr;SENS1:AVER:COUN 1
                                             ! Sets averaging to manual and one sample per reading.
380   OUTPUT @Pwr_mtr;SENS2:AVER:COUN 16
                                             ! Sets averaging to manual and 16 samples per reading.
390   !
400   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
                                             ! Applies Cal Factor for 1.44 GHz to sensor 1 data.
410   OUTPUT @Pwr_mtr;SENS2:CORR:FREQ 0.96E9
                                             ! Applies Cal Factor for 960 MHz to sensor 2 data.
420   !
430   FOR I= 1 TO 10
440       OUTPUT @Pwr_mtr;READ1?;READ2?      ! Measures Power at sensors.
450       ENTER @Pwr_mtr;Chn_pow1
460       ENTER @Pwr_mtr;Chn_1rat2
470       PRINT Chn_pow1,Chn_1rat2
480   NEXT I
490   !
610   END
```

A.4.6 Reading Power Measurements Example 6

FETCh? allows finer control of the meter's measurement sequences. The low level control function of FETCh#? is to first, process the measurement channel information based upon sensor data and configuration settings and then, place the result in the meter data output buffer to be read by the slot 0 controller/resource manager.

The following program uses the INIT command to control acceptance of measurement values in conjunction with the FETCh? command. When using FETCh#?, both trigger sequence arming and triggering/data acquisition must be controlled in the program. This is juxtaposed with MEAS#? and READ#? which, being higher level commands, include these functions.

This program will allow the fastest measurement speed performance in NORMal mode.

```
280   ASSIGN @Pwr_mtr to 70101
290   CLEAR @Pwr_mtr
300   WAIT 1
310   OUTPUT @Pwr_mtr;CALC1:MODE NORM! Can ONLY send Configuration Commands in Normal Mode!!
320   OUTPUT @Pwr_mtr;TRIG:SOUR IMM          ! Power Meter controls triggering with TRIG or *TRG
330   OUTPUT @Pwr_mtr;INIT:CONT ON           ! Power Meter controls instrument trigger arming
340   !
350   OUTPUT @Pwr_mtr;SENS2:AVER:COUN 1      ! Sets averaging to manual and one sample per reading.
360   OUTPUT @Pwr_mtr;CALC1:POW 2            ! Channel 1 configured to measure sensor 2 power
370   !
380   OUTPUT @Pwr_mtr;SENS2:CORR:FREQ 17.4E9
                                             ! Applies Cal Factor for 17.4 GHz to data.
390   !
400   FOR I=1 TO 10
430       OUTPUT @Pwr_mtr;FETC1?             ! Measures Power at sensor.
440       ENTER @Pwr_mtr;Rdg
450       PRINT Rdg
460   NEXT I
470   !
670   END
```

A.4.7 Reading Power Measurements Example 7

The following example is similar to the example above; however, it now uses TRIG:SOUR BUS instead of IMMediate so that triggering is controlled by the TRIG command. EXT or TTLT triggering can not be used in NORMal Mode. Additionally, INIT:CONT is set to OFF, allowing the INIT (or INIT:IMM) command to control arming of the triggering cycle.

```
280   ASSIGN Pwr_mtr to 70101
290   CLEAR @Pwr_mtr
300   WAIT 1
310   OUTPUT @Pwr_mtr;CALC1:MODE NORM! Can ONLY send Configuration Commands in Normal Mode!!
320   OUTPUT @Pwr_mtr;TRIG:SOUR BUS          ! Program controls triggering with TRIG or *TRG
330   OUTPUT @Pwr_mtr;INIT:CONT OFF          ! Program controls instrument trigger arming
340   !
350   OUTPUT @Pwr_mtr;SENS2:AVER:COUN 1      ! Sets averaging to manual and one sample per reading.
360   OUTPUT @Pwr_mtr;CALC1:POW 2            ! Channel 1 configured to measure sensor 2 power
370   !
380   OUTPUT @Pwr_mtr;SENS2:CORR:FREQ 17.4E9
                                             ! Applies Cal Factor for 17.4 GHz to data.
390   !
400   FOR I=1 TO 10
410         OUTPUT @Pwr_mtr;INIT             ! INIT arms the triggering and measurement cycle
420         OUTPUT @Pwr_mtr;TRIG             ! BUS trigger
430         OUTPUT @Pwr_mtr;FETC1?           ! Measures Power at sensor.
440         ENTER @Pwr_mtr;Rdg
450         PRINT Rdg
460   NEXT I
470   !
670   END
```

A.4.8 Reading Power Measurements Example 8

The following program shows fast BUS triggering in the SWIFt Mode. TRIG (or *TRG) is used to acquire data, and FETCh#? processes and outputs the data to the slot 0 controller/resource manager. This program does not use the meter's data buffering capability.

```
170   ASSIGN @Pwr_mtr to 70101
180   CLEAR @Pwr_mtr
190   WAIT 1
200   OUTPUT @Pwr_mtr;*CLS                    ! Clears old messages from SYST:ERR buffer
205    WAIT 0.3
210   PRINT Running...
220   !
230   OUTPUT @Pwr_mtr;CALC1:MODE NORM         ! NORMal Mode to perform channel configuration
240   OUTPUT @Pwr_mtr;CALC1:POW 1             ! Channels 1 or 2 can be either POW 1 or POW 2
250                                           ! RAT and DIFF are illegal in SWIFt and BURSt Modes.
260   !
270   !###########################################
280   !
290   ! Entering SWIFt Mode
300   !
310   OUTPUT @Pwr_mtr;CALC1:MODE SWIF         ! Enters SWIFt Mode for fastest individual data point
                                              ! triggered measurements.
320   !
330   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
                                              ! Applies Cal Factor in SWIFt mode
340                                           ! Can be sent before or after CALC#:MODE SWIF
350   !
360   OUTPUT @Pwr_mtr;TRIG:SOUR BUS           ! BUS or EXT triggering is slower than IMM
370                                           ! Can be sent before or after CALC#:MODE SWIF
380   !
390   FOR I=1 TO 10
400       FOR K=1 TO 20
410       OUTPUT @Pwr_mtr;TRIG                ! TRIG is the SCPI Bus trigger. Can also use *TRG
420       OUTPUT @Pwr_mtr;FETC1?              ! FETC#? acquires data
430       ENTER @Pwr_mtr;Chan1sens_1(K)
440       !ENTER @Pwr_mtr;Chan1sens_1(K),Chan2sens_2(K)
                                              ! Use this line when two sensors are attached.
450   NEXT K
460   !
470       PRINT Chan1sens_1(*)
480   !PRINT Chan1sens_1(*),Chan2sens_2(*)     ! Use this line when two sensors are attached.
490       PRINT “”
500   NEXT I
510   !
520   END
```

A.4.9 Reading Power Measurements Example 9

The following program uses BURSt Mode for the fastest measurement rates possible. The maximum measurement speed is performed when TRIG:DEL is set to 0. TRIG:DEL controls the speed of sampling and data buffering. When TRIG:DEL is set to 0.004, samples will be taken every 4 milliseconds, on both channels if connected, and stored in internal data buffer. This speed does not control or account for the meter's internal data processing time after data acquisition or the speed of data transfer to the controller. This second component of time, the time to get data from the 58542, is proportional to the number of data points measured. Therefore, the example below uses only one channel and keeps the number of points buffered to a minimum.

Both channels' data will be taken at the same time during BURSt Mode. Power meter measurement speed does not change when two sensors are connected versus only one sensor. However, the meter's processing time and the time to transmit the data over the VXI and GPIB interfaces takes longer due to the additional sensor data. If two sensors are connected, calibrated, and their respective channels are set to ON, then must read both arrays of data. Only read one array of data when one sensor is attached, calibrated, and set to ON.

As shown in the examples, send the CALC#:MODE BURS command prior to sending the BURSt Mode configuration commands TRIG:SOUR, TRIG:MODE, TRIG:COUN, and TRIG:DEL.

```
10    ALPHA ON
20    CLEAR SCREEN
30    !
40    OPTION BASE 1
50    DIM Id$[50],Err_msg$[70]
60    DIM Ques$[200],A$[80],Chan1sens_1(500)
70    DIM Chan2sens_2(500)
80    !
90    !
100   !####################################
110   !     Instrument ADDRESS ALLOCATION
120   !
130   ASSIGN @Slot0 TO 70100           ! 70100 is Logical Address of the Slot 0 controller
140   ASSIGN @Pwr_mtr TO 70101         ! 70101 is Power Meter in 1st position right of slot 0
150   !####################################
160   !
170   ! Identify Attached Instruments
180   !
190   OUTPUT @Slot0;*Slot0;*IDN?
200   ENTER @Slot0;Id$
210   PRINT SLOT 0 is ;Id$
220   !
230   OUTPUT @Pwr_mtr;*IDN?
240   ENTER @Pwr_mtr;Id$
250   PRINT SLOT 1 is ;Id$
260   WAIT 1
270   !
280   !
290   CLEAR @Pwr_mtr
300   WAIT 1
310   OUTPUT @Pwr_mtr;*CLS             ! Clears old messages from SYST:ERR buffer
320   WAIT 0.3
330   OUTPUT @Pwr_mtr;CALC1:MODE NORM! NORMal Mode to perform channel configuration
340   OUTPUT @Pwr_mtr;CALC1:POW 1      ! Channels 1 or 2 can be either POW 1 or POW 2
350                                    ! RAT and DIFF are illegal in SWIFt and BURSt Modes.
360   OUTPUT @Pwr_mtr;TRIG:SOUR IMM            ! IMM set here to highlight conflict with BURSt operation.
370   !
380   !####################################
390   !
400   ! Entering BURSt Mode
410   !
```

```
420   OUTPUT @Pwr_mtr;CALC1:MODE BURS          ! Enters BURSt Mode for fastest measurement speeds.
430   !
440   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 2.44E9
          ! Applies Cal Factor in BURSt mode
450               ! Can be sent before or after CALC#:MODE BURS
460   !
470   OUTPUT @Pwr_mtr;TRIG:SOUR BUS! IMM triggering is illegal in BURSt Mode, Use BUS or EXT.
480               ! Can be sent before or after CALC#:MODE BURS
490   !
500   OUTPUT @Pwr_mtr;TRIG:MODE POST           ! Data acquired after trigger, not before as with PRE.
510               ! Send only after CALC#:MODE BURS
520   !
530   OUTPUT @Pwr_mtr;TRIG:COUN 500            ! 500 readings acquired and stored with each trigger.
540               ! Send only after CALC#:MODE BURS, following TRIG:MODE.
550               ! Be sure COUN# matches ENTER variable dimension.
560   !REDIM Chan1sens_1(500),Chan2sens_2(500)  ! REDIM to smaller array size only if necessary.
570   !
580       OUTPUT @Pwr_mtr;TRIG:DEL .000        ! 0 millisecond between rdgs setting is 5100 rdgs/sec.
590               ! Send only after CALC#:MODE BURS, following TRIG:COUN.
600   !
610   !
620   FOR I=1 TO 10
630       Time1=TIMEDATE
640   !
650       OUTPUT @Pwr_mtr;TRIG                 ! TRIG is the SCPI Bus trigger. Can also use *TRG
660       OUTPUT @Pwr_mtr;FETC1?! FETC#? acquires data
670       ENTER @Pwr_mtr;Chan1sens_1(*)
680       !ENTER @Pwr_mtr;Chan1sens_1(*),Chan2sens_2(*)
          ! Use this line when two sensors are attached.
690       !
700       Time2=TIMEDATE
710       Time=Time2-Time1
720       Speed=500/Time
730       PRINT Chan1sens_1(*)
740       !PRINT Chan1sens_1(*),Chan2sens_2(*)   ! Use this line when two sensors are attached.
750       PRINT Speed; readings per second, round trip.
760       PRINT “”
770   NEXT I
780   !
790   END
```

A.5 Instrument Triggering Examples

A.5.1 TRIGgering Example 1

The following two programs illustrate the operation of TRIG:SOUR HOLD with the MEAS#? and FETCh#? measurement data queries. This first program shows TRIG:SOUR HOLD used with MEAS#?. Since MEAS#? is a high level command containing it's own trigger sequence arming, triggering, and measurement data acquisition functions, the program returns valid measurement data. The output from the SYST:ERR query is -0, No error.

```
260   ASSIGN @Pwr_mtr to 70101
270   CLEAR @Pwr_mtr
275   WAIT 1
280   OUTPUT @Pwr_mtr;*CLS                ! Clears old messages from SYST:ERR buffer
290   WAIT 0.3
300   OUTPUT @Pwr_mtr;CALC1:MODE NORM! NORMal Mode.
310   OUTPUT @Pwr_mtr;CALC1:POW 1         ! Channel 1 configured to measure sensor 1 power
320   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
                                          ! Applies Cal Factor for 1.44 GHz to data.
330   !
340   OUTPUT @Pwr_mtr;TRIG:SOUR HOLD      ! Halts triggering when used with FETCh#?
350   !
360   FOR I=1 TO 10
370       OUTPUT @Pwr_mtr;MEAS1?          ! MEAS#? returns valid data with TRIG:SOUR set to HOLD.
380       ENTER @Pwr_mtr;Rdg
390       PRINT Rdg
400   NEXT I
410   !
420   OUTPUT @Pwr_mtr;SYST:ERR?           ! Query error buffer
430   ENTER @Pwr_mtr;A$
440   PRINT A$
530   END
```

A.5.2 TRIGgering Example 2

When FETCh#? is used while TRIG:SOUR is HOLD, invalid data, 9.e+40, is returned and the SYST:ERR? query returns -230, Data corrupt or stale.

```
260   ASSIGN @Pwr_mtr to 70101
270   CLEAR @Pwr_mtr
275   WAIT 1
280   OUTPUT @Pwr_mtr;*CLS                    ! Clears old messages from SYST:ERR buffer
290   WAIT 0.3
300   OUTPUT @Pwr_mtr;CALC1:MODE NORM         ! NORMal Mode.
310   OUTPUT @Pwr_mtr;CALC1:POW 1             ! Channel 1 configured to measure sensor 1 power
320   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
                                              ! Applies Cal Factor for 1.44 GHz to data.
330   !
340   OUTPUT @Pwr_mtr;TRIG:SOUR HOLD          ! Halts triggering when used with FETCh#?
350   !
360   FOR I=1 TO 10
370       OUTPUT @Pwr_mtr;FETC1?              ! FETC#? will not acquire valid data when TRIG:SOUR
                                              ! is set to HOLD.
380       ENTER @Pwr_mtr;Rdg
390       PRINT Rdg
400   NEXT I
410   !
420   OUTPUT @Pwr_mtr;SYST:ERR?               ! Query error buffer
430   ENTER @Pwr_mtr;A$
440   PRINT A$
530   END
```

A.5.3 TRIGgering Example 3

Please note, the READ#? measurement data query requires TRIG:SOUR IMM for proper operation. While TRIG:SOUR is HOLD, data output is also invalid, 9.e+40, but the SYST:ERR? query response is different, -214, Trigger deadlock.

```
260   ASSIGN @Pwr_mtr to 70101
270   CLEAR @Pwr_mtr
275   WAIT 1
280   OUTPUT @Pwr_mtr;*CLS                     ! Clears old messages from SYST:ERR buffer
290   WAIT 0.3
300   OUTPUT @Pwr_mtr;CALC1:MODE NORM          ! NORMal Mode.
310   OUTPUT @Pwr_mtr;CALC1:POW 1              ! Channel 1 configured to measure sensor 1 power
320   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
                                               ! Applies Cal Factor for 1.44 GHz to data.
330   !
340   OUTPUT @Pwr_mtr;TRIG:SOUR HOLD           ! Halts triggering when used with FETCh#?
350   !
360   FOR I=1 TO 10
370       OUTPUT @Pwr_mtr;READ1?               ! READ#? requires TRIG:SOUR IMM not, HOLD.
380       ENTER @Pwr_mtr;Rdg
390       PRINT Rdg
400   NEXT I
410   !
420   OUTPUT @Pwr_mtr;SYST:ERR?                ! Query error buffer
430   ENTER @Pwr_mtr;A$
440   PRINT A$
530   END
```

A.5.4 TRIGgering Example 4

The following examples show the use of BUS triggering with FETCh#? in NORMal and BURSt Modes.

```
270   ASSIGN @Pwr_mtr to 70101
280   CLEAR @Pwr_mtr
290   WAIT 1
300   OUTPUT @Pwr_mtr;CALC1:MODE NORM          ! NORMal Mode.
310   OUTPUT @Pwr_mtr;CALC1:POW 1              ! Channel 1 configured to measure sensor 1 power
320   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
                                               ! Applies Cal Factor for 1.44 GHz to data.
330   !
340   OUTPUT @Pwr_mtr;TRIG:SOUR BUS
350   !
360   FOR I=1 TO 10
362   OUTPUT @Pwr_mtr;TRIG                     ! TRIG is the SCPI Bus trigger. Can also use *TRG
370       OUTPUT @Pwr_mtr;FETC1?               ! FETC#? acquires data
380       ENTER @Pwr_mtr;Rdg
390       PRINT Rdg
400   NEXT I
410   !
530   END
```

A.5.5 TRIGgering Example 5

BURSt Mode BUS triggering with FETCh?

```
10    ALPHA ON
20    CLEAR SCREEN
30    !
40    OPTION BASE 1
50    DIM Id$[50],Err_msg$[70]
60    DIM Ques$[200],A$[80],Chan1sens_1(50)
70    DIM Chan2sens_2(50)
80    !
90    !
100   !######################################
110   ! Instrument ADDRESS ALLOCATION
120   !
130   ASSIGN @Slot0 TO 70100                  ! 70100 is Logical Address of the Slot 0 controller
140   ASSIGN @Pwr_mtr TO 70101                ! 70101 is Power Meter, in 1st position right of slot 0
150   !######################################
160   !
170   ! Identify Attached Instruments
180   !
190   OUTPUT @Slot0;*IDN?
200   ENTER @Slot0;Id$
210   PRINT SLOT 0 is ;Id$
220   !
230   OUTPUT @Pwr_mtr;*IDN?
240   ENTER @Pwr_mtr;Id$
250   PRINT SLOT 1 is ;Id$
260   WAIT 1
270   !
280   !
290   CLEAR @Pwr_mtr
300   WAIT 1
310   OUTPUT @Pwr_mtr; *CLS                   ! Clears old messages from SYST:ERR buffer
320   !
330   OUTPUT @Pwr_mtr;CALC1:MODE NORM         ! NORMal Mode to perform channel configuration
340   OUTPUT @Pwr_mtr;CALC1:POW 1             ! Channels 1 or 2 can be either POW 1 or POW 2
350                                           ! RAT and DIFF are illegal in SWIFt and BURSt Modes
360   OUTPUT @Pwr_mtr;TRIG:SOUR IMM           ! IMM set here to highlight conflict with BURSt operation
370   !
380   !#################################
390   !
400   !     Entering BURSt Mode
410   !
420   OUTPUT @Pwr_mtr;CALC1:MODE BURS         ! Enters BURSt Mode for fastest measurement speeds
430   !
440   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 2.44E9
                                              ! Applies Cal Factor in burst mode
450                                           ! Can be sent before or after CALC#:MODE BURS
460   !
470   OUTPUT @Pwr_mtr;TRIG:SOUR BUS           ! IMM triggering is illegal in BURSt Mode, Use BUS or EXT.
480                                           ! Can be sent before or after CALC#:MODE BURS
490   !
500   OUTPUT @Pwr_mtr;TRIG:MODE POST          ! Data acquired after trigger, not before as with PRE.
510                                           ! Send only after CALC#:MODE BURS
520   !
530   OUTPUT @Pwr_mtr;TRIG:COUN 50            ! 50 readings acquired and stored with each trigger
540                                           ! Send only after CALC#:MODE BURS
550                                           ! Be sure COUN# matches ENTER variable dimension.
560   !REDIM Chan1sens_1(50),Chan2sens_2(50)   ! REDIM to smaller array size only if necessary
570   !
580   OUTPUT @Pwr_mtr;TRIG:DEL .001           ! 1 millisecond between rdgs, 0 ms is 5100 rdgs/sec
590                                           ! Send only after CALC#:MODE BURS
600   !
```

```
610  !
620  FOR I=1 TO 10
630  WAIT .01                           ! For handshaking compensation. If necessary, use wait
640  !
650  !
660  OUTPUT @Pwr_mtr;TRIG               ! TRIG is the SCPI Bus trigger. Can also use *TRG
670  WAIT .01                           ! For handshaking compensation. If necessary, use wait
680  OUTPUT @Pwr_mtr;FETC1?             ! FETC#? acquires data
690  ENTER @Pwr_mtr;Chan1sens_1(*)
700  !ENTER @Pwr_mtr;Chan1sens_1(*),Chan2sens_2(*)
                                        ! Use this line when two sensors are attached.
710  PRINT Chan1sens_1(*)
720  !PRINT Chan1sens_1(*),Chan2sens_2(*)! Use this line when two sensors are attached.
730  PRINT ""
740  NEXT I
750  !
760  END
```

A.5.6 TRIGgering Example 6

Be careful when using BURSt Mode. For example, the user must use INIT:CONT ON; the user does not have manual control of trigger arming sequence using INIT, as the user do using the FETCh#? command with the NORMal Mode. This is of particular concern using TTL level triggering on either the front panel EXT connector or the VXI backplane TTLT trigger functions. The user must control the triggering through control of the trigger source. The user must use INIT:CONT ON in BURSt Mode; using INIT control arming of the trigger sequence is not allowed.

The following example uses EXTernal TTL level triggering using the external trigger input on the front panel of the 58542 VXI Universal Power Meter. Twenty readings are stored in the measurement buffer. Then FETCh#? is used during SWIFt Mode. Please note there is a TTL level hardware handshake capability using the ANALOG OUT BNC connector which is also on the front panel. The Analog BNC will output a high (5V) when the instrument is ready for triggering. After a trigger is received, the Analog BNC output goes low (0V).

TRIG:SOUR HOLD halts the SWIFt Mode triggering sequence.

```
10    ALPHA ON
20    CLEAR SCREEN
30    !
40    OPTION BASE 1
50    DIM Id$[50],Err_msg$[70]
60    DIM Ques$[200],A$[80],Chan1sens_1(20)
70    DIM Chan2sens_2(20)
80    !
90    !
100   !##########################################
110   !      Instrument ADDRESS ALLOCATION
120   !
130   ASSIGN @Slot0 TO 70100                    ! 70100 is Logical Address of the Slot 0 controller
140   ASSIGN @Pwr_mtr TO 70101                  ! 70101 is Power Meter, in 1st position right of slot 0
150   !###########################################
160   !
170   !
180   CLEAR @Pwr_mtr
190   WAIT 1
200   OUTPUT @Pwr_mtr;*CLS                      ! Clears old messages from SYST:ERR buffer
210   WAIT 0.3
220   OUTPUT @Pwr_mtr;CALC1:MODE NORM           ! NORMal Mode to perform channel configuration
230   OUTPUT @Pwr_mtr;CALC1:POW 1               ! Channels 1 or 2 can be either POW 1 or POW 2
240                                             ! RATio and DIFFerence configurations are illegal in
                                                ! SWIFt and BURSt Modes.
250   !
260   !############################################
270   !
280   ! Entering SWIFt Mode
290   !
300   OUTPUT @Pwr_mtr;CALC1:MODE SWIF           ! Enters SWIFt Mode for fastest individual data point
                                                ! triggered measurements.
310   !
320   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 2.22E8
                                                ! Applies Cal Factor in SWIFt mode
330                                             ! Can be sent before or after CALC#:MODE SWIF
340   !

350   OUTPUT @Pwr_mtr;TRIG:SOUR EXT             ! TTL level triggering, Each trigger acquires one data
                                                ! point in SWIFt Mode.
360                                             ! Can be sent before or after CALC#:MODE SWIF
370   !
380   OUTPUT @Pwr_mtr;TRIG:COUN 20              ! Stores 20 points for simultaneous output using
                                                ! FETCh#?.
390                                             ! Must be sent after CALC#:MODE SWIF
```

```
400    !
410    !
420    FOR I=1 TO 10
430    !WAIT 2                                  ! If necessary, use wait
440
450    OUTPUT @Pwr_mtr;FETC1?                   ! FETC#? acquires data
460    ENTER @Pwr_mtr;Chan1sens_1(*)
470    !ENTER @Pwr_mtr;Chan1sens_1(*),Chan2sens_2(*)
                                                ! Use this line when two sensors are attached.
480    !
490    PRINT Chan1sens_1(*)
500    !PRINT Chan1sens_1(*),Chan2sens_2(*)     ! Use this line when two sensors are attached.
510         PRINT “”
520    NEXT I
530    !
540    END
```

The fastest SWIFt Mode measurement speeds are achieved with TRIG:SOUR:IMM and TRIG:COUN set to values larger than about 25. See High Speed Measurements in Section A.8 for additional information and examples.

The fastest BURSt Mode measurement speeds are achieved with TRIG:COUN at about 80 and TRIG:DEL 0. See High Speed Measurements in Section for additional information and examples.

A.6 Channel Configuration Examples

A.6.1 Single Sensor Power Measurement

Default definition of software calculation channels 1 and 2 are for sensors 1 and 2, respectively. This configuration allows measurement of the power level incident upon sensor 1 on software configuration channel 1, and measurement of the power level incident upon sensor 2 on software configuration channel 2. Normal Mode and Swift Mode measurements are faster with only one sensor set to STATe ON. Since this is the default configuration, consider turning one channel off occasionally. This will only be necessary when two sensors are attached and both are calibrated. Channel configuration can be changed by sending the CALCulate:RATio or CALCulate:DIFFerence commands.

```
OUTPUT @Pwr_mtr;*RST                        ! Configure 58542 to Default Setup
OUTPUT @Pwr_mtr;CALC1?;CALC2?               ! Query channel configuration
ENTER @Pwr_mtr;Chn1_config,Chn2_config
PRINT Chn1_config,Chn2_config
     Program Output:                        POW 1    POW 2
```

The following program reverses the default sensor-to-channel assignments.

```
OUTPUT @Pwr_mtr;CALC2:POW 1                 ! Configures channel 2 to measure Sensor 1 power
OUTPUT @Pwr_mtr;CALC1:POW 2                 ! Configures channel 1 to measure Sensor 2 power
OUTPUT @Pwr_mtr;CALC1?;CALC2?               ! Query channel configuration
ENTER @Pwr_mtr;Chn1_config,Chn2_config
PRINT Chn1_config,Chn2_config
     Program Output:                        POW 2    POW 1
```

A.6.2 Ratio Measurement

The CALCulate:RATio command is used to automatically measure the ratio of the power levels incident on the two sensors. Permissible settings are Sensor 1/Sensor 2 & Sensor 2/Sensor 1.

```
OUTPUT @Pwr_mtr;CALC1:RAT 2,1               ! Configures channel 1 to measure Sensor 2 over
                                            ! Sensor 1 power
OUTPUT @Pwr_mtr;CALC2:RAT 1,2               ! Configures channel 2 to measure Sensor 1 over
                                            ! Sensor 2 power
OUTPUT @Pwr_mtr;CALC1?;CALC2?               ! Query channel configuration
ENTER @Pwr_mtr;Chn1_config,Chn2_config
PRINT Chn1_config,Chn2_config
     Program Output:                        RAT2,1    RAT1,2
```

A.6.3 Difference Measurement

The CALCulate:DIFFerence command is used to automatically measure the difference of the power levels incident upon the two sensors.

```
OUTPUT @Pwr_mtr;CALC1:DIFF 2,1              ! Configures channel 1 to measure Sensor 2 minus
                                            !  Sensor 1 power
OUTPUT @Pwr_mtr;CALC1?                      ! Query Channel 1 configuration
ENTER @Pwr_mtr;Chn1_config
PRINT Chn1_config
     Program Output:                        DIFF2,1
```

A.7 Cal Factor Examples

Entering a frequency causes the power meter to use frequency calibration factors which are stored in the power sensor's internal EEPROM. Generally, frequency calibration factors are stored in one gigahertz steps.

Two methods are available for frequency entry. Use SENSe:CORRection:FREQuency to enter a specific carrier frequency, or use SENSe:CORRection:VPROpf to enable the meter's voltage proportional to frequency input BNC (See front panel of the meter).

A.7.1 Cal Factor Example 1

The following program automatically applies the correct cal factor for an 8.23 GHz measurement frequency to the measured data value.

```
310   ASSIGN @Pwr_mtr to 70101
320   CLEAR @Pwr_mtr
330   WAIT 1
350   OUTPUT @Pwr_mtr;CALC1:MODE NORMAL
                                            ! Can ONLY send Cal Factor Correction Commands in
                                            ! Normal Mode!!
360   !
370   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 8.23E9
                                            ! Applies Cal Factor for 8.23 GHz to measurement data.
380   !
400   OUTPUT @Pwr_mtr;MEAS1?                ! Measures Power of 8.23 GHz signal.
410   ENTER @Pwr_mtr;Rdg
420   PRINT Rdg
460   !
470   END
```

A.7.2 Cal Factor Example 2

The following program steps from 1.8 GHz to 2.2 GHz in 10 MHz intervals. The measurement at each step is automatically corrected for cal factor.

```
340   ASSIGN @Pwr_mtr to 70101
350   CLEAR @Pwr_mtr
360   WAIT 1
370   REAL Rdg(41)
380   REAL Frq(41)
390   OUTPUT @Pwr_mtr;CALC1:MODE NORMAL     ! Can ONLY send Cal Factor Correction Commands in
                                            ! Normal Mode!!
400   !
410   FOR I=1 TO 41
420   !
430   Freq=1.8E+9+(I-1)*1.E+7
440   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ;Freq
                                            ! Applies Cal Factor to measurement data.
450   !
460   OUTPUT @Pwr_mtr;MEAS1?
470   ENTER @Pwr_mtr;Rdg(I)
480   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ?      ! Power meter outputs freq setting to confirm command
                                            ! reception.
490   ENTER @Pwr_mtr;Frq(I)
500   NEXT I
510   !
520   PRINT Rdg(*)
530   PRINT Frq(*)
540   !
550   END
```

A.7.3 Cal Factor Example 3

The following program also steps from 1.8 GHz to 2.2 GHz in 10 MHz intervals. The measurement at each step is automatically corrected for cal factor using the V_{PROP}F connector on the front panel. In the first part of the program the power meter's V_{PROP}F input is configured for compatibility with the Giga-tronics Source.

```
340  ASSIGN @Pwr_mtr to 70101
350  CLEAR @Pwr_mtr
360  WAIT 1
370  REAL Rdg(41)
380  OUTPUT @Pwr_mtr;CALC1:MODE NORMAL
                                          ! Can ONLY send VpropF Config. Commands in Normal
                                          ! Mode!!
390  !
400  ! Configure VpropF input for Cal Factor correction
410  !
420  OUTPUT @Pwr_mtr;MEM:SEL VPROPF1      ! Selects memory table for sensor on for editing
430  OUTPUT @Pwr_mtr;MEM:FREQ 0.0         ! Sets freq that corresponds to 0.0V out from source.
440  OUTPUT @Pwr_mtr;MEM:SLOP 0.5E-9      ! Sets the voltage to freq (must be linear) relationship of
                                          ! source's output (V/Hz).
450  !
460  OUTPUT @Pwr_mtr;SENS1:CORR:VPRO ON
                                          ! Turns VpropF on and all other sources of Cal Factor
                                          ! correction OFF
470  !
480  FOR I=1 TO 41
490  !
500  Freq=1.8E+9+(I-1)*1.E+7
510  OUTPUT @Source;CW;Freq;HZ
520  !
530  OUTPUT @Pwr_mtr;MEAS1?
540  ENTER @Pwr_mtr;Rdg(I)
550  NEXT I
560  !
570  PRINT Rdg(*)
580  !
590  END
```

A.7.4 Cal Factor Example 4

The following program shows how to input specific User Calibration Factor during measurement. This is not the same as the special Frequency Cal Factors that can be programmed into the EEPROM. (See Sensor EEPROM Commands in Section 2.5.32 for more information). This technique applies a specific known value during measurements. It is useful when having performed a sensor calibration with other devices attached to the sensor input, such as a power splitter or coupler, but do not want to change or reprogram the CAL factor information inside the power sensor's EEPROM. First the program sets the 58542 to 50 MHz where the Cal Factors of Giga-tronics power sensors are always 0.0 dBm.

```
310   ASSIGN @Pwr_mtr to 70101
320   CLEAR @Pwr_mtr
330   WAIT 1
340   OUTPUT @Pwr_mtr;CALC1:MODE NORMAL
                                               ! Can ONLY send Sensor Offset Commands in
                                               ! Normal Mode!!
350   !
360   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 5E7
                                               ! Cal Factors always 0.00dB at 50 MHz.
370   OUTPUT @Pwr_mtr;SENS1:CORR:OFFS:STAT ON
                                               ! Enables sensor offset control.
380   OUTPUT @Pwr_mtr;SENS1:CORR:OFFS 6.024
                                               ! Enters 6.024 dB as a Sensor Offset
390   !
400   OUTPUT @Pwr_mtr;MEAS1?                   ! Measures Power
410   ENTER @Pwr_mtr;Rdg
420   PRINT Rdg
430   !
440   END
```

A.7.5 Cal Factor Example 5

In Burst and Swift Mode, the meter's functionality is restricted to allow the microprocessor to devote most of its operation to performing measurement operations. If measuring a single frequency, this technique will not be need for Burst or Swift Mode data. In Burst or Swift Mode, measurement correction for Cal Factor is always performed with the SENS:CORR:FREQ ### command. By sending this command before entering the Burst or Swift Mode from the Normal mode, all subsequent Burst or Swift Mode measurements will be corrected for that frequency.

The above operation creates a problem if changing frequency during Burst or Swift Mode measurement. The following program is used to apply correct cal factors to swept or multi frequency measurements that have been performed during Burst or Swift Modes. In these modes, swept measurement correction functions are performed in the computer, thus increasing the measurement speed more than would otherwise be possible. First the Cal Factors are read from the sensor's EEPROM. Then the Cal Factors are put through an interpolation algorithm into a correction data array that matches the start/stop frequencies of the test source and the number of measurement points to be collected during measurement. Then measurement is performed, and the correction factor array is added to the measurement data array before being output to a file or swept onto a screen display.

```
10    ALPHA ON
20    CLEAR SCREEN
30    OPTION BASE 1
40    !
50    DIM Freqs(80),Clfcs(80),Corr_clf(5000),Rdgs(5000),Rdgs_corr(5000)
                                            ! These are in REDIMs later in program
60    DIM Id$[90],Err_msg$[70]
70    DIM Ques$[200],Calf$[200]
80    !
90    Tim_per_pnt = 5                       ! milliseconds per sample point. Set 1 to 999.
100   !
110   !##########################################
120   !       Instrument ADDRESS ALLOCATION
130   !
140   ASSIGN @Slot0 TO 70100                ! 70100 is Logical Address of the Slot 0
                                            ! controller (Resource Manager)
150   ASSIGN @Sweeper TO 720                ! Use normal address for non-VXI instruments
160   ASSIGN @Pwr_mtr TO 70101              ! Power Meter is next to the slot 0 Resource Manager
170   !##########################################
180   !
190   ! Identify Attached Instruments
200   !
210   OUTPUT @Slot0;*IDN?
220   ENTER @Slot0;Id$
230   PRINT SLOT 0 is ;Id$
240   !
250   OUTPUT @Pwr_mtr;*IDN?
260   ENTER @Pwr_mtr;Id$
270   PRINT SLOT 1 is ;Id$
280   WAIT 1
290   !
300   !#########################################
310   !
320   ! CALIBRATE THE SENSORS
330   !
340   !#########################################
350   !
360   CLEAR @Pwr_mtr
370   WAIT 1
380   OUTPUT @Pwr_mtr;CALC1:MODE NORMAL
                                            ! Can ONLY send Sensor Offset
```

```
                                           ! Commands in Normal Mode!!
390   !
400   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 5E7
                                           ! Cal Factors always 0.00dB at 50MHz
410   !
420   ! Find out the number of Cal Factors in EEPROM
430   ! Include Std Cal Factors at 1 GHz intervals and any special Cal Factors.
440   !
450   OUTPUT @Pwr_mtr;DIAG:SENS1:EEPROM:CALFR?
460   ENTER @Pwr_mtr;Frs_std_freq,Std_freq_step,No_std_freqs,No_spl_freqs
470   ENTER @Pwr_mtr;Frst_std_freq,Std_freq_step,No_std_freqs
480   No_spl_freqs=1                       ! Added to correct bug in DIAG:SENS#:EEPROM:CALFR?
490   PRINT Frst_std_freq,Std_freq_step,No_std_freqs,No_spl_freqs
500   !
510   !
520   ! When you query the 58542 for frequencies and Cal Factors there will be
530   ! No_std_freqs + No_spl_freqs = number of items you need to read
540   !
550   No_cal_pnts=No_std_freqs+No_spl_freqs
560   REDIM Freqs(No_cal_pnts),Clfcs(No_cal_pnt)
                                           ! Dimension according to number of Cal Factors to be read in.
570   PRINT There are ;No_cal_pnts; of Cal Factors in this sensor.
580   !
590   OUTPUT @Pwr_mtr;SENS1:CORR:EEPROM:FREQ?
                                           ! Asks the 58542 for the Frequency array from sensor 1
600   ENTER @Pwr_mtr;Freqs(*)
610   PRINT Freqs(*)
620   !
630   OUTPUT @Pwr_mtr;SENS1:CORR:EEPROM:CALF?
                                           ! Asks the 58542 for the Frequency array from sensor 1
640   ENTER @Pwr_mtr;Clfcs(*)
650   PRINT Clfcs(*)
660   !
670   !##################################################################################
680   ! Now that all the Cal Factors are loaded with their corresponding frequencies,
690   ! we need to create a table of interpolated Cal Factor points based upon the frequencies used
700   ! and number of measurement points in the test program.
710   !
720   ! First get a couple pieces of necessary information
730   !
740   INPUT Input sweep START frequency in GHz.,Strt_freq
750   INPUT Input sweep STOP frequency in GHz.,Stop_freq
760   INPUT Number of points per sweep. 100 to 400 suggested.,Swep_pnts
770   IF Strt_freq>Stop_freq THEN
780       PRINT Make STOP freq. > START freq.
790       WAIT 1
800       GOTO 740
810   ELSE
820       IF Strt_freq<.051 THEN GOTO 740
830       IF Swep_pnts<1 THEN GOTO 740! You can put additional requirements in this section.
840   END IF
850   Strt_freq=1.E+9*Strt_freq
860   Stop_freq=1.E+9*Stop_fr              ! Set units to Hz
870   REDIM Corr_clf(Swep_pnts),Rdgs(Swep_pnts),Rdgs_corr(Swep_pnts)
                                           ! Re-sized to match number of measuremen
                                           ! points in sweep
880   !
890   ! Interpolation routine creates Cal Factor Correction Table in Clf_corr(*)
900   ! Values are in dB!!!! NOT W linear units
910   !
920   FOR I=1 TO Swep_pnts                 ! For each point in Clf_corr(*)
930       Rdgs(I)=1
```

```
940         Corr_freq=Strt_freq+(I-1)*(Stop_freq-Strt_freq)/(Swep_pnts-1)
                                            ! Corr_freq is freq corresponding to Clf_corr(I)
950         FOR K=1 TO No_cal_pnts          ! Find next highest and lowest Cal Factor
                                            ! frequency from Corr_freq
960   IF Freqs(K)>=Corr_freq THEN
970         Next_higher_f=Freqs(K)
980         Next_higher_clf=Clfcs(K)
990         Next_lower_f=Freqs(K-1)
1000        Next_lower_clf=Clfcs(K-1)
1010        K=No_cal_pnts
1020  !
1030        END IF
1040        NEXT K
1050        F_delta_multpl=(Corr_freq-Next_lower_f)/(Next_higher_f-Next_lower_f)
                                            ! Multiplier for next formula
1060        Corr_clf(I)=Next_lower_clf+F_delta_multpl*(Next_higher_clf-Next_lower_clf)
                                            ! Interpolates the Cal Factor Value
1070        PRINT Corr_freq,Corr_clf(I)
1080  NEXT I
1090  !
1100  ! Set the source for swept operation
1110  !
1120  PRINT Configuring Gigatronics 7200 Microwave Sweeper.
1130  OUTPUT @Sweeper,HP                    ! Set Giga-tronics 7200 to HP emulation mode
1140  OUTPUT @Sweeper;OPFA;Strt_freq
1150  OUTPUT @Sweeper;OPFB;Stop_freq
1160  OUTPUT @Sweeper;FC;Tim_per_pnt*Swep_pnts;MS
1170  !
1180  ! Perform Measurement in Burst Mode or Swift Mode for Fast measurements.
1190  ! Shared Variables may include Sweep_pnts and Tim_per_pnt
1200  !
1210  GOSUB Burst_mode_swp                  ! Goes to the swept frequency, Burst/Swift Mode Data
                                            ! acquisition subroutine
1220  !
1230  ! Data comes back in from Burst Mode-or Swift Mode-as rdgs(*). Always in log units, dBm.
1240  !
1250  MAT Rdgs_corr=Rdgs+Corr_clf           ! Array math function that adds the measurement
                                            ! data to the cal factor array
1260  !
1270  GOSUB Output_data                     ! Output to screen graph, printout, file, etc.....
1280  !
1290  END
```

A.8 High Speed Measurement Examples

A.8.1 High Speed Measurement Example 1

The following program shows the fastest SWIFt Mode measurement speed possible. IMMediate triggering is used to allow the 58542 to trigger a measurement automatically. Be sure to only use one channel defined to one sensor for the fastest speeds.

```
170  ASSIGN @Pwr_mtr to 70101
180  CLEAR @Pwr_mtr
190  WAIT 1
200  OUTPUT @Pwr_mtr;*CLS                  ! Clears old messages from SYST:ERR buffer
210  WAIT 0.3
220  OUTPUT @Pwr_mtr;CALC1:MODE NORM       ! NORMal Mode to perform channel configuration
230  OUTPUT @Pwr_mtr;CALC1:POW 1           ! Channels 1 or 2 can be either POW 1 or POW 2
240                                        ! RAT and DIFF are illegal in SWIFt and BURSt Modes.
250  !
260  !#################################
270  !
280  ! Entering SWIFt Mode
290  !
300  OUTPUT @Pwr_mtr;CALC1:MODE SWIF       ! Enters SWIFt Mode for fastest continuous
                                           ! measurements with IMM.
310  !
320  OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 16.97E9
                                           ! Applies Cal Factor in SWIFt mode
330                                        ! Can be sent before or after CALC#:MODE SWIF
340  !
350  OUTPUT @Pwr_mtr;TRIG:SOUR IMM
360
370  !
380  !
390  Loopcount=100
400  Time1=TIMEDATE
410  FOR I=1 TO Loopcount
420  OUTPUT @Pwr_mtr;FETC1?                ! FETC#? acquires data
430  ENTER @Pwr_mtr;Chan1sens_1(I)
440  NEXT I
450  Time2=TIMEDATE
460  Time=Time2-Time1
470  Speed=Loopcount/Time                  ! Units are readings per second.
480  !
490  PRINT Chan1sens_1(*)
500  PRINT Speed; readings per second.
510  !
520  END
```

A.8.2 High Speed Measurement Example 2

Using one of the slowest system configurations available, external PC controller with GPIB slot 0 resource manager and programming through a very slow Basic program, the previous program achieved a speed of 24 readings per second. With faster systems using embedded PCs and faster software, measurement speeds have been recorded as high as 71 readings per second.

```
10    ALPHA ON
20    CLEAR SCREEN
30    !
40    OPTION BASE 1
50    DIM Id$[50],Err_msg$[70]
60    DIM Ques$[200],A$[80],Chan1sens_1(200)
70    DIM Chan2sens_2(200)
80    Counter=1
90    !
100   !#############################################
110   !          Instrument ADDRESS ALLOCATION
120   !
130   ASSIGN @Slot0 TO 70100                ! 70100 is Logical Address of the Slot 0 controller
140   ASSIGN @Pwr_mtr TO 70101              ! 70101 is Power Meter in 1st position right of slot 0
150   !#############################################
160   !
280   !
290   CLEAR @Pwr_mtr
300   WAIT 1
310   OUTPUT @Pwr_mtr;*CLS                  ! Clears old messages from SYST:ERR buffer
320   WAIT 0.3
330   OUTPUT @Pwr_mtr;CALC1:MODE NORM       ! NORMal Mode to perform channel configuration
340   OUTPUT @Pwr_mtr;CALC1:POW 1           ! Channels 1 or 2 can be either POW 1 or POW 2
350                                         ! RAT and DIFF are illegal in SWIFt and BURSt Modes.
370   !
380   !#############################################
390   !
400   ! Entering SWIFt Mode
410   !
420   OUTPUT @Pwr_mtr;CALC1:MODE SWIF       ! Enters SWIFt Mode for fastest continuous
                                            ! measurements with IMM.
430   !
440   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 16.97E9
                                            ! Applies Cal Factor in SWIFt mode
450                                         ! Can be sent before or after CALC#:MODE SWIF
460   !
470   OUTPUT @Pwr_mtr;TRIG:SOUR IMM! IMM triggering is illegal in SWIFt Mode, Use BUS or EXT.
480                                         ! Can be sent before or after CALC#:MODE SWIF
490   !
610   !
611   Loopcount=50
613   Time1=TIMEDATE
620   FOR I=1 TO Loopcount
670       !OUTPUT @Pwr_mtr;FETC1?           ! FETC#? acquires data
671   OUTPUT @Pwr_mtr;FETC1?;FETC2?         ! Use this line when two sensors are attached.
680       !ENTER @Pwr_mtr;Chan1sens_1(I)
681   ENTER @Pwr_mtr;Chan1sens_1(I),Chan2sens_2(I)
                                            ! Use this line when two sensors are attached.
730   NEXT I
731       Time2=TIMEDATE
733       Time=Time2-Time1
743       Speed=Loopcount/Time              ! Units are readings per second.
753       !
763       !PRINT Chan1sens_1(*)
773       PRINT Chan1sens_1(*),Chan2sens_2(*)  ! Use this line when two sensors are attached.
783       !PRINT Speed; readings per second.
784       PRINT Speed; readings per second per channel.
793       !
803    END
```

A.8.3 High Speed Measurement Example 3

Using the same system configuration, the previous program achieved a speed of 25 readings per second per channel. With faster systems using embedded PCs and faster software, measurement speeds have been recorded as high as 71 readings per second.

The following program shows fast BUS triggering in the SWIFt Mode. TRIG (or *TRG) is used to acquire data, and FETCh#? processes and outputs the data to the slot 0 controller/resource manager. This program does not use the meter's data buffering capability.

```
170  ASSIGN @Pwr_mtr to 70101
180  CLEAR @Pwr_mtr
190  WAIT 1
200  OUTPUT @Pwr_mtr;*CLS                    ! Clears old messages from SYST:ERR buffer
205  WAIT 0.3
210  PRINT Running...
220  !
230  OUTPUT @Pwr_mtr;CALC1:MODE NORM         ! NORMal Mode to perform channel configuration
240  OUTPUT @Pwr_mtr;CALC1:POW 1             ! Channels 1 or 2 can be either POW 1 or POW 2
250                                          ! RAT and DIFF are illegal in SWIFt and BURSt Modes.
260  !
270  !################################################
280  !
290  ! Entering SWIFt Mode
300  !
310  OUTPUT @Pwr_mtr;CALC1:MODE SWIF         ! Enters SWIFt Mode for fastest individual data point
                                             ! triggered measurements.
320  !
330  OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.44E9
                                             ! Applies Cal Factor in SWIFt mode
340                                          ! Can be sent before or after CALC#:MODE SWIF
350  !
360  OUTPUT @Pwr_mtr;TRIG:SOUR BUS           ! BUS or EXT triggering is slower than IMM
370                                          ! Can be sent before or after CALC#:MODE SWIF
380  !
390  FOR I=1 TO 10
400      FOR K=1 TO 20
410  OUTPUT @Pwr_mtr;TRIG                    ! TRIG is the SCPI Bus trigger. Can also use *TRG
420  OUTPUT @Pwr_mtr;FETC1?                  ! FETC#? acquires data
430  ENTER @Pwr_mtr;Chan1sens_1(K)
440  !ENTER @Pwr_mtr;Chan1sens_1(K),Chan2sens_2(K)
                                             ! Use this line when two sensors are attached.
450  NEXT K
460      !
470             PRINT Chan1sens_1(*)
480      !PRINT Chan1sens_1(*),Chan2sens_2(*) ! Use this line when two sensors are attached.
490             PRINT “”
500      NEXT I
510      !
520      END
```

A.8.4 High Speed Measurement Example 4

The following program shows SWIFt Mode measurements using EXT TTL triggering and Buffered data. INIT:CONT must be set to ON; The user can not use INIT with INIT:CONT set to OFF. This program buffers 30 measurements in the 58542 before group download to the controller using the TRIG:COUN command in line 390. In the SWIFT mode, this command must be sent after the TRIG:SOUR command.

```
170  ASSIGN @Pwr_mtr to 70101
180  CLEAR @Pwr_mtr
190  WAIT 1
200  OUTPUT @Pwr_mtr;*CLS                    ! Clears old messages from SYST:ERR buffer
210  WAIT 0.3
220  OUTPUT @Pwr_mtr;CALC1:MODE NORM         ! NORMal Mode to perform channel configuration
230  OUTPUT @Pwr_mtr;CALC1:POW 1             ! Channels 1 or 2 can be either POW 1 or POW 2
240                                          ! RATio and DIFFerence configurations are illegal in
                                             ! SWIFt and BURSt Modes.
250  !
260  !################################################
270  !
280  ! Entering SWIFt Mode
290  !
300  OUTPUT @Pwr_mtr;CALC1:MODE SWIF         ! Enters SWIFt Mode for fastest individual data point
                                             ! triggered measurements.
310  !
320  OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 2.22E8
                                             ! Applies Cal Factor in SWIFt mode
330                                          ! Can be sent before or after CALC#:MODE SWIF
340  !
350  OUTPUT @Pwr_mtr;TRIG:SOUR EXT           ! TTL level triggering, Each trigger acquires one
                                             ! data point in SWIFt Mode.
360                                          ! Can be sent before or after CALC#:MODE SWIF
370  !
380  Num_pts=30
390  OUTPUT @Pwr_mtr;TRIG:COUN;Num_pts       ! Stores 20 points for simultaneous output using FETCh#?.
400                                          ! Must be sent after CALC#:MODE SWIF
410  !
420  REDIM Chan1sens_1(Num_pts),Chan2sens_2(Num_pts)
430  !
440  Time1=TIMEDATE
450  WAIT 5                                  ! If necessary, use wait statements
460                                          ! For example, use it if TRIG:COUN # is high or TTL
                                             ! trigger source is slow.
470      OUTPUT @Pwr_mtr;FETC1?              ! FETC#? acquires data
480      !OUTPUT @Pwr_mtr;FETC1?;FETC2?
490      ENTER @Pwr_mtr;Chan1sens_1(Num_pts)
500  !ENTER @Pwr_mtr;Chan1sens_1(Num_pts),Chan2sens_2(Num_pts)
                                             ! Use this line when two sensors are attached.
510  Time2=TIMEDATE
520  Time=Time2-Time1
530  Speed=Num_pts/Time
540  !
550            PRINT Chan1sens_1(*)
560      !PRINT Chan1sens_1(*),Chan2sens_2(*)   ! Use this line when two sensors are attached.
570      PRINT Speed; readings per second.
580            PRINT “”
590      !
600      END
```

A.8.5 High Speed Measurement Example 5

The following program uses BURSt Mode for the fastest measurement rates possible. The maximum measurement speed is performed when TRIG:DEL is set to 0. TRIG:DEL controls the speed of sampling and data buffering. When TRIG:DEL is set to 0.004, samples will be taken every 4 milliseconds, on both channels if connected, and stored in the meter's internal data buffer. This speed does not control or account for the meter's internal data processing time after data acquisition or the speed of data transfer to the controller. This second component of time, the time to get data from the 58542, is proportional to the number of data points measured. Therefore, the example below uses only one channel and keeps the number of points buffered to a minimum.

Both channels' data will be taken at the same time during BURSt Mode. Power meter measurement speed does not change when two sensors are connected versus only one sensor. However, the 58542 processing time and the time to transmit the data over the VXI and GPIB interfaces takes longer due to the additional sensor data. If two sensors are connected, calibrated, and their respective channels are set to ON, then read both arrays of data. Only read one array of data when one sensor is attached, calibrated, and set to ON.

As shown in the examples, send the CALC#:MODE BURS command prior to sending the BURSt Mode configuration commands TRIG:SOUR, TRIG:MODE, TRIG:COUN, and TRIG:DEL.

```
10    ALPHA ON
20    CLEAR SCREEN
30    !
40    OPTION BASE 1
50    DIM Id$[50],Err_msg$[70]
60    DIM Ques$[200],A$[80],Chan1sens_1(500)
70    DIM Chan2sens_2(500)
80    !
90    !
100   !########################################
110   ! Instrument ADDRESS ALLOCATION
120   !
130   ASSIGN @Slot 0 TO 70100                 ! 70100 is Logical Address of the Slot 0 controller
140   ASSIGN @Pwr_mtr TO 70101                ! 70101 is Power Meter in 1st position of slot 0
150   !###########################################
160   !
170   ! Identify Attached Instruments
180   !
190   OUTPUT @Slot0;*IDN?
200   ENTER @Slot0;Id$
210   PRINT SLOT 0 is ;Id$
220   !
230   OUTPUT @Pwr_mtr;*IDN?
240   ENTER @Pwr_mtr;Id$
250   PRINT SLOT 1 is ;Id$
260   WAIT 1
270   !
280   !
290   CLEAR @Pwr_mtr
300   WAIT 1
310   OUTPUT @Pwr_mtr;*CLS                    ! Clears old messages from SYST:ERR buffer
320   WAIT 0.3
330   OUTPUT @Pwr_mtr;CALC1:MODE NORM         ! NORMal Mode to perform channel configuration
340   OUTPUT @Pwr_mtr;CALC1:POW 1             ! Channels 1 or 2 can be either POW 1 or POW 2
350                                           ! RAT and DIFF are illegal in SWIFt and BURSt Modes.
360   OUTPUT @Pwr_mtr;TRIG:SOUR IMM           ! IMM set here to highlight conflict with BURSt operation
370   !
380   !#########################################
390   !
```

```
400  ! Entering BURSt Mode
410  !
420  OUTPUT @Pwr_mtr;CALC1:MODE BURS           ! Enters BURSt Mode for fastest measurement speeds.
430  !
440  OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 2.44E9
                                               ! Applies Cal Factor in BURSt mode
450                                            ! Can be sent before or after CALC#:MODE BURS
460  !
470  OUTPUT @Pwr_mtr;TRIG:SOUR BUS             ! IMM triggering is illegal in BURSt Mode, Use BUS or
                                               ! EXT.
480                                            ! Can be sent before or after CALC#:MODE BURS
490  !
500  OUTPUT @Pwr_mtr;TRIG:MODE POST            ! Data acquired after trigger, not before as with PRE.
510                                            ! Send only after CALC#:MODE BURS
520  !
530  OUTPUT @Pwr_mtr;TRIG:COUN 500             ! 500 readings acquired and stored with each trigger.
540                                            ! Send only after CALC#:MODE BURS, following
                                               ! TRIG:MODE.
550                                            ! Be sure COUN# matches ENTER variable dimension.
560  !REDIM Chan1sens_1(500),Chan2sens_2(500)  ! REDIM to smaller array size only if necessary.
570  !
580  OUTPUT @Pwr_mtr;TRIG:DEL .000             ! 0 millisecond between rdgs setting is 5100 rdgs/sec.
590                                            ! Send only after CALC#:MODE BURS, following
                                               ! TRIG:COUN.
600  !
610  !
620  FOR I=1 TO 10
630  Time1=TIMEDATE
640  !
650  OUTPUT @Pwr_mtr;TRIG                      ! TRIG is the SCPI Bus trigger. Can also use *TRG
660  OUTPUT @Pwr_mtr;FETC1?                    !FETC#? acquires data
670  ENTER @Pwr_mtr;Chan1sens_1(*)
680  !ENTER @Pwr_mtr;Chan1sens_1(*),Chan2sens_2(*)
                                               ! Use this line when two sensors are attached.
690  !
700  Time2=TIMEDATE
710  Time=Time2-Time1
720  Speed=500/Time
730  PRINT Chan1sens_1(*)
740  !PRINT Chan1sens_1(*),Chan2sens_2(*)       ! Use this line when two sensors are attached.
750  PRINT Speed; readings per second, round trip.
760  PRINT “”
770  NEXT I
780  !
790  END
```

A.8.6 High Speed Measurement Example 6

The preceding program performed 500 measurements at a rate of 5100 per second then processed and output the data to the controller. Round trip speed was between 120 and 140 per second. Using 100 measurements per BURSt instead of 500, the round trip speed was 100 to 115 per second.

The following example is similar to the previous program except for the use of EXT triggering to initiate the BURSt measurement.

```
10    ALPHA ON
20    CLEAR SCREEN
30    !
40    OPTION BASE 1
50    DIM Id$[50],Err_msg$[70]
60    DIM Ques$[200],A$[80],Chan1sens_1(500)
70    DIM Chan2sens_2(500)
80    !
90    !
100   !##################################################
110   !        Instrument ADDRESS ALLOCATION
120   !
130   ASSIGN @Slot0 TO 70100                  ! 70100 is Logical Address of the Slot 0 controller
140   ASSIGN @Pwr_mtr TO 70101                ! 70101 is Power Meter in 1st position right of slot 0
150   !##################################################
160   !
170   ! Identify Attached Instruments
180   !
190   OUTPUT @Slot0;*IDN?
200   ENTER @Slot0;Id$
210   PRINT SLOT 0 is ;Id$
220   !
230   OUTPUT @Pwr_mtr;*IDN?
240   ENTER @Pwr_mtr;Id$
250   PRINT SLOT 1 is ;Id$
260   WAIT 1
270   !
280   !
290   CLEAR @Pwr_mtr
300   WAIT 1
310   OUTPUT @Pwr_mtr;*CLS                    ! Clears old messages from SYST:ERR buffer
320   WAIT 0.3
330   OUTPUT @Pwr_mtr;CALC1:MODE NORM         ! Normal Mode to perform channel configuration
340   OUTPUT @Pwr_mtr;CALC1:POW 1             ! Channels 1 or 2 can be either POW 1 or POW 2
350                                           ! RAT and DIFF are illegal in SWIFt and BURSt Modes.
360   !
370   !##############################################
380   !
390   ! Entering BURSt Mode
400   !
410   OUTPUT @Pwr_mtr;CALC1:MODE BURS         ! Enters BURSt Mode for fastest measurement speeds.
420   !
430   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 2.44E9
420                                           ! Applies Cal Factor in burst mode
440                                           ! Can be sent before or after CALC#:MODE BURS
450   !
460   OUTPUT @Pwr_mtr;TRIG:SOUR EXT           ! EXT TTL Level input begins BURSt Mode measure-
ment.
470                                           ! Can be sent before or after CALC#:MODE BURS
480   !
490   OUTPUT @Pwr_mtr;TRIG:MODE POST          ! Data acquired after trigger, not before as with PRE.
500                                           ! Send only after CALC#:MODE BURS
510   !
520   OUTPUT @Pwr_mtr;TRIG:COUN 500           ! 500 readings acquired and stored with each trigger.
530                                           ! Send only after CALC#:MODE BURS, following
```

```
TRIG:MODE
540                                              ! Be sure COUN# matches ENTER variable dimension.
550   !REDIM Chan1sens_1(500),Chan2sens_2(500)  ! REDIM to smaller array size only if necessary.
560   !
570   OUTPUT @Pwr_mtr;TRIG:DEL .000              ! 0 millisecond between rdgs setting is 5100 rdgs/sec.
580                                          ! Send only after CALC#:MODE BURS, following TRIG:COUN.
590   !
600   !
610   FOR I=1 TO 10
620   !WAIT .01                         ! Wait for trigger compensation. If necessary, use wait statements
630
640
650   Time1=TIMEDATE
660   !
670   OUTPUT @Pwr_mtr;FETC1?            ! FETC#? acquires data
680   WAIT 2                            ! for POST triggering
690   ENTER @Pwr_mtr;Chan1sens_1(*)
700   !ENTER @Pwr_mtr;Chan1sens_1(*),Chan2sens_2(*)
                                        ! Use this line when two sensors are attached.
710   !
720   Time2=TIMEDATE
730   Time=Time2-Time1
740   Speed=500/Time
750   PRINT Chan1sens_1(*)
760   !PRINT Chan1sens_1(*),Chan2sens_2(*)! Use this line when two sensors are attached.
770        PRINT Speed; readings per second, round trip.
780        PRINT “”
790   NEXT I
800   !
810   END
```

The preceding program performs measurement at approximately 5100 measurements per second. Assuming pretriggering during the PRINT statements, a round trip speed of 180 to 195 measurements per second was achieved.

A.9 Relative or Referenced Measurement Examples

A.9.1 Relative or Referenced Measurements Example 1

The following program automatically sets a power level reference when the computer's ENTER key is actuated. From that point forward the power level - relative to the power level at the time the ENTER key was actuated - is monitored and displayed with minimum and maximum values since the reference was set.

```
290  ASSIGN @Pwr_mtr to 70101
300  CLEAR @Pwr_mtr
310  WAIT 1
320  !
330  OUTPUT @Pwr_mtr;CALC1:MODE NORMAL! Can ONLY use REFerence commands in NORMal Mode!!
340  OUTPUT @Pwr_mtr;CALC2:POW 2            ! Channel 2 configured to measure sensor 2 power
350  !
360  OUTPUT @Pwr_mtr;TRIG:SOUR IMM          ! TRIG:SOUR must be IMM when using READ#?
370  OUTPUT @Pwr_mtr;INIT:CONT OFF          ! INITiate:CONTinuous arming must be OFF when using
          ! READ#?
380  !
390  OUTPUT @Pwr_mtr;SENS2:AVER:COUN 1      ! Sets averaging to manual and one sample per
                                            ! measurement reading.
400  !
410  OUTPUT @Pwr_mtr;SENS2:CORR:FREQ 17.4E9
                                            ! Applies Cal Factor for 17.4 GHz to measurement data.
420  !
430  !############################################
440  !
450  ! Relative measurement setup using CALC#:REF:COLL.
460  !
470  OUTPUT @Pwr_mtr;CALC2:REF:MAG 0.00     ! Resets REFerence level from value set with
CALC2:REF:COLL
480  OUTPUT @Pwr_mtr;CALC2:REF:STAT ON      ! Enables REFerence operation.
490  !
500  INPUT Press ENTER key to set to 0.0dB reference operation,Dmy
510  OUTPUT @Pwr_mtr;CALC2:REF:COLL         ! Takes current measurement and adds inverse to further
                                            ! measurements
520  !
530                                         ! Setup power meters' MIN and MAX monitors
540  !
550  OUTPUT @Pwr_mtr;CALC2:MIN:STAT ON
560  OUTPUT @Pwr_mtr;CALC2:MAX:STAT ON
570  !
580  PRINT Power Variation, Largest Min, Largest Max.
590  WHILE Cont_meas=1
600       OUTPUT @Pwr_mtr;MEAS2?            ! Measures Power at sensor.
610       ENTER @Pwr_mtr;Rdg
620       !
630       OUTPUT @Pwr_mtr;CALC2:MIN?        ! Grabs minimum since CALC2:MIN:STAT ON
640       ENTER @Pwr_mtr;Min
650       OUTPUT @Pwr_mtr;CALC2:MAX?        ! Grabs maximum since CALC2:MAX:STAT ON
660       ENTER @Pwr_mtr;Max
670       PRINT Rdg,"",Min,"",Max
680       END WHILE
690  !
890  END
```

A.9.2 Relative or Referenced Measurements Example 2

The following program finds the 1 dB compression of an amplifier/transmitter at a single frequency.

```
300   ASSIGN @Pwr_mtr to 70101
310   CLEAR @Pwr_mtr
320   WAIT 1
330   !
340   OUTPUT @Pwr_mtr;CALC1:MODE NORMAL
                                                    ! Can ONLY use REFerence commands in NORMal
Mode!!
350   OUTPUT @Pwr_mtr;CALC1:POW 1                   ! Channel 1 configured to measure sensor 1 power
360   OUTPUT @Pwr_mtr;CALC1:UNIT W                  ! Transmitter output power in Watts.
370   OUTPUT @Pwr_mtr;CALC2:RAT 1,2                 ! Transmitter gain stays in default, dBm.
380   OUTPUT @Pwr_mtr;CALC2:UNIT DBM                ! Transmitter output power in Watts. Units are channel,
                                                    ! not sensor, specific
390   !
400   !
410   OUTPUT @Pwr_mtr;TRIG:SOUR IMM  ! TRIG:SOUR must be IMM when using READ#?
420   OUTPUT @Pwr_mtr;INIT:CONT OFF  ! INITiate:CONTinuous arming must be OFF when using READ#?
430   !
440   OUTPUT @Pwr_mtr;SENS1:AVER:COUN 1! Sets averaging to manual and one sample per measurement
reading.
450   OUTPUT @Pwr_mtr;SENS2:AVER:COUN 4! Sets averaging to manual and 4 samples per measurement
reading.
460   !
470   OUTPUT @Pwr_mtr;SENS1:CORR:FREQ 1.94E9
                                             ! Applies Cal Factor for 1.94 GHz, Sensor 1.
480   OUTPUT @Pwr_mtr;SENS2:CORR:FREQ 7.0E7
                                             ! Applies Cal Factor for 70 MHz IF input, Sensor 2.
490   !
500   !####################################################
510   !
520   ! System Setup
530   !  Power Splitter attached to source.
540   !  DUT input and sensor 2 on the splitter outputs.
550   !  Sensor 1 at DUT output.
560   !  Check max power rating on sensor 1 and use an appropriate attenuator if necessary.
570   !  Use offset to account for any necessary attenuation when not using Giga-tronics high power sensors.
580   !
590   !####################################################
600   !
610   ! Relative measurement setup using CALC#:REF:COLL.
620   !
630   OUTPUT @Pwr_mtr;CALC2:REF:MAG 0.00            !  Resets REFerence level from value set with
CALC2:REF:COLL
640   OUTPUT @Pwr_mtr;CALC2:REF:STAT ON             ! Enables REFerence operation.
650   !
660   Smal_sig_pwr=-30
670   OUTPUT @Source;OPPL;Smal_sig_pwr              ! Set power to small signal region of amp, -30 dBm.
680   OUTPUT @Pwr_mtr;CALC2:REF:COLL                ! Takes current measurement and adds inverse to further
                                                    ! measurements
690   !
700   !  STEP POWER FROM SMALL SIGNAL REGION TO ABOVE 1 dB COMPRESSION LEVEL
710   !
720   FOR I=Smal_sig_pwr TO Smal_sig_pwr+40 STEP .05
730        OUTPUT @Source;OPPL;I             ! Increments power level into amplifier's input. Add wait statement
                                             ! at 735 if source settles too slow.
740   OUTPUT @Pwr_mtr;READ2?                 ! Measures Sensor 1 over sensor 2, dB relationship
750   ENTER @Pwr_mtr;Rdg
760   PRINT Rdg
770   !
780   Comp_lvl=-1
790   IF Rdg<Comp_lvl THEN                   ! Finds 1 dB compression level
800   OUTPUT @Pwr_mtr;READ1?                 ! Measures Output Power at sensor 1.
```

```
810    ENTER @Pwr_mtr;Rdg
820    PRINT 1 dB Compression level is ;Rdg; Watts, Output Power.
830    GOSUB End
840            END IF
850    NEXT I
860    !
1060   End: !
1070   END
```

A.10 SRQ Interrupt Examples

```
10   ASSIGN @Pwr_mtr to 70101            ! Set GPIB address
20   DIM Buf$[100]                       ! Define data buffer
30   ON INTR 7 GOSUB Srq_isp             ! Assign interrupt service function
40   ENABLE INTR 7;2                     ! enable interrupt
50   !
60   CLEAR @Pwr_mtr                      ! Reset instrument
70   WAIT 1
80   OUTPUT @Pwr_mtr;*CLS                ! Clear interface and output queue
90   WAIT 0.3
100  !
110  OUTPUT @Pwr_mtr;*ESE 1                     ! Enable OPC bit
120  OUTPUT @Pwr_mtr;STAT:OPER:ENAB 7712        ! Enable channel 1 & 2 limit and trigger wait mask
130  !
140  Srq_flag=0                                 ! Reset task done flag
150  OUTPUT @Pwr_mtr;CAL1:ZERO?;*OPC            ! Zero sensor 1
160  WHILE Srq_flag = 0                         ! Wait until it is done
170      END WHILE
180  ENTER @Pwr_mtr;Result                      ! Read zero result
190  IF Result = 0 THEN PRINT Zero is good
200      ELSE
210      PRINT Zero is no good
220      END IF
230  ENTER @Pwr_mtr;Esr_val                     ! Read ESR status
240  !
250  Srq_flag = 0                               ! Reset task done flag
260  Meas_flag = 0                              ! Reset measurement flag
270  OUTPUT @Pwr_mtr;MEAS1?;*OPC                ! Measure channel 1 data
280  WHILE Srq_flag = 0                         ! Wait until it is done
290      END WHILE
300  IF Meas_flag = 0 THEN                      ! If data is not read
310      ENTER @Pwr_mtr;Result                  ! Read measurement data
320      END IF
330      ENTER @Pwr_mtr;Esr_val                 ! Read ESR status
340  !
350  STOP
360  !
370  Srq_isp:!
380  State = SPOLL(@Pwr_mtr)                    ! Serial poll
390  !
400  IF BIT(State,3) THEN                       ! Error bit
410       OUTPUT @Pwr_mtr;SYST:ERR?             ! Query error message
420       ENTER @Pwr_mtr;Buf$                   ! Read in
430       PRINT Error:;Buf$                     ! Print out message
440       END IF
450  !
460  IF BIT(State,7) THEN                       ! OSB bit
470      !
480       IF BIT(State,4) THEN                  ! MAV bit
490   ENTER @Pwr_mtr;Result                     ! Query measurement data
500   Meas_flag = 1                             ! Set data read flag
510   END IF
520  !
530   OUTPUT @Pwr_mtr;STAT:OPER?                ! Query operation status register
540   ENTER @Pwr_mtr;Status
550   PRINT OSB;Status
560   IF BIT(Status,5) THEN PRINT Swift/Burst trigger ready
570   IF BIT(Status,8) THEN PRINT Channel 1 lower limit
580      IF BIT(Status,9) THEN PRINT Channel 1 upper limit
590      IF BIT(Status,10) THEN PRINT Channel 2 lower limit
600      IF BIT(Status,11) THEN PRINT Channel 2 upper limit
610      END IF
620  !
```

```
630   IF BIT(State,5) THEN                      ! ESR bit
640       OUTPUT @Pwr_mtr;*ESR?                 ! Reset event status register
650       END IF
660   !
670   Srq_flag = 1                              ! Set task done flag
680   ENABLE INTR 7;2
690   RETURN
860   END
```

A.11 Instrument & Sensor Identification Examples

A.11.1 Instrument Identification

The following example reads manufacturer identification, model number and software version number. Software version number is important for troubleshooting and factory technical support. Make sure user or users can identify the software version when requesting technical support.

```
10    ALPHA ON
20    DIM ID$[60],Mfgr$[12]
          ! FOR INSTRUMENT ID VIEWING AT START UP
30    !
40    CLEAR SCREEN
50    !
60    !####################################################
70        ! Instrument ADDRESS ALLOCATION
80    !
90    ASSIGN @Slot0 TO 70100          ! 70200 is Logical Address of the Slot 0 controller
100   ASSIGN @Pwr_mtr TO 70101        ! Default Power Meter Address of 255 allows VXI dynamic
110                                   ! address configuration
120   !
130   !####################################################
140   !
150   OUTPUT @Slot0;*IDN?
160   ENTER @Slot0;Id$
170   PRINT SLOT 0 is ;Id$
180   !
190   OUTPUT @Pwr_mtr;*IDN?
200   ENTER @Pwr_mtr;Mfgr$,Model,No,Software_ver
                                      ! DIM Mfgr$ to 12 characters
210   PRINT A;Mfgr$; model;Model;with software version;Software_ver;is installed in slot 1.
211   !
212       ! Main Measurement Loop Start.
```

Instrument identification can be read as a single string as is done in lines 150 to 170 above. Sometimes, however, it is more convenient to read these four items separately for use in comment statements or as identifiers for selecting special command libraries or subroutines.

A.11.2 Sensor Model Identification

A simple method of separating the manufacturer sub-string from the three following numeric items requires reading the manufacturer alphabetical letters into a string variable of limited size. Use a DIM or a REDIM statement to limit size to 12 characters. Thus Model and Software Version can easily be read into numeric variables. No is always 0; it is placed in the *IDN? command output as called for by the IEEE 488.2 Standard. The 58542 can not report it's serial number - which is printed on a label positioned on the side of the instrument - over the bus. If preferred, there is space on the 58542's front panel for the serial number label. The factory will place it there upon request.

The following example reads the model number and serial number of the power sensor currently in use. What does currently in use mean? It means that this was the last sensor connected to this meter input. It does not mean that the sensor is currently attached or that the sensor is calibrated. For in-program checking for sensor attachment or completed power sweep calibration, see the Error Control Examples in Section A.3.3 or the Reading Power Measurements in Section A.4. By operating in this manner, if preferred use a simple WHILE/UNTIL loop in the program to detect when the operator connects the correct power sensor.

Tracking the model number allows the user to be sure that a sensor is being used which is appropriate for measurement application. For example, some user measurement routines may require the use of Peak Power Sensors rather than CW Power Sensor or use of high power sensors rather than standard sensors. The user's program can automatically check for the appropriate sensor and prompt to Please connect Giga-tronics Sensor model #####, if necessary.

```
240   !
250   GOSUB Pwr_amp_test
260   !
270   GOSUB Store_results
280   !
290   END LOOP                       ! Return to beginning of main program.
300   !
310   Pwr_amp_test:                  ! Output is 30 to 40 Watts, TOO HIGH for Standard Sensors
320   !
330   Reqd_snsr1_mdl=80355           ! Actual 50W Peak Power Sensor model is 80355A.
340                                  ! The A is ignored in Gigatronics software to simplify programs like
                                     ! this one.
350   Reqd_snsr2_mdl=80350           ! Standard 80350A Peak Power Sensor on input
360   !
370   !
380
!##############################################
390   !
400   ! Read and compare sensors for correct model number
410   !
420   OUTPUT @Pwr_mtr;SENS1:CORR:EEPROM:TYPE?
                                     ! responds with model and serial numbers
430   ENTER @Pwr_mtr;New_snsr1_model,New_snsr1_serno
440   OUTPUT @Pwr_mtr;SENS1:CORR:EEPROM:TYPE?
                                     ! responds with model and serial numbers
450   ENTER @Pwr_mtr;New_snsr1_model,New_snsr1_serno
460   !
470   IF Reqd_snsr1_mdl=New_snsr1_model THEN
480   ELSE
490   PRINT The amplifier's high power output must be attenuated for this test.,
500   PRINT Connect a Gigatronics model;Reqd_snsr1_mdl;to the channel 1 power sensor cable.,
510   INPUT Then press ENTER,Dmy
520   PRINT""
530   GOSUB Pwr_swp_call
540   END IF
550   !
560   IF Reqd_snsr2_mdl=New_snsr2_model THEN
570   ELSE
```

```
580  PRINT Connect a Gigatronics model;Reqd_snsr2_mdl;to the channel 2 power sensor cable.,
590  INPUT Then press ENTER,Dmy
600  GOSUB Pwr_swp_cal2
610  END IF
620  !
630  GOSUB Compare_serno          ! Calibrates sensors, if necessary. SEE next example
640  !
650  GOSUB Gain_tst               ! Configures and performs amplifier gain test.
660  GOSUB Onedb_comp_tst         ! Configures and performs amplifier 1 dB compression test.
670  !
680  RETURN
690  !
700  Pwr_swp_call:                ! Power Sweep Calibration calibrates the sensor to the meter.
710  !
```

A.11.3 Sensor Serial Number Identification

Tracking sensor serial numbers is important. Measurements cannot be performed unless the sensor has been calibrated to the meter using the built-in power sweep calibrator: this ensures that measurements will never be performed with an uncalibrated sensor. The Giga-tronics 58542 tracks this requirement by reading the sensor's serial number. Thus, by reading in the sensor's serial number at the beginning of measurement subroutines, the user can automatically determine whether or not power sweep calibration is required. This is illustrated in the following example.

```
990         Compare_serno: !
995    !
1000   !##############################################
1010   !
1020        ! This program identifies sensors by their serial number
1030        ! and jumps to a calibration subroutine if it is a new sensor.
1040   !
1050   !##############################################
1060   !
1070        ! Sensor Identification routine at beginning of measurement sequence.
1080        ! New sensors will not operate unless they pass calibration first.
1090   !
1100   OUTPUT @Pwr_mtr;SENS1:CORR:EEPROM:TYPE?
1110   ENTER @Pwr_mtr;New_snsr1_model,New_snsr1_serno
1120   OUTPUT @Pwr_mtr;SENS2:CORR:EEPROM:TYPE?
1130   ENTER @Pwr_mtr;New_snsr2_model,New_snsr2_serno
1140   !
1150   IF Old_snsr1_serno=New_snsr1_serno THEN
1160   !
1170   ELSE
1180   GOSUB Pwr_swp_cal1                        ! New Sensor 1, Must Calibrate To the Meter
1190   END IF
1200   !
1210   IF Old_snsr2_serno=New_snsr2_serno THEN
1220   !
1230   ELSE
1240   GOSUB Pwr_swp_cal2                        ! New Sensor 2, Must Calibrate To the Meter
1250   END IF
1260   !
1270        ! Concludes Sensor Serial Number Comparison
1280        ! Use a similar routine when a particular sensor model is required.
1290        ! To do both: Nest an IF-THEN for model # ahead of GOSUB
                                                 ! Pwr_swp_cal#.
1300   !
1310   !#######################################################
1320   !
1330        ! Main loop
1340   !
1350   OUTPUT @Pwr_mtr;SENS1:AVER:TCON:MOV
            ! MOV, Average new sample with previous samples
1360   OUTPUT @Pwr_mtr;SENS2:AVER:TCON:MOV
            ! MOV, Average new sample with previous samples
1370   !
1380   LOOP
1390   OUTPUT @Pwr_mtr;MEAS1?
            ! TRIGGER AND READ CHANNEL 1
1400   ENTER @Pwr_mtr;MEAS1
1410   PRINT CHANNEL 1 IS ;Meas1;dBm.
1420   !
1430   OUTPUT @Pwr_mtr;MEAS2?         ! TRIGGER AND READ CHANNEL 2
1440   ENTER @Pwr_mtr;Meas2
1450   PRINT CHANNEL 2 IS ;Meas2;dBm.
1460   END LOOP
1470   !
1480   !######################################################
```

```
1490  !
1500        ! SUBROUTINES
1510  !
1520  !
1530  Pwr_swp_cal1:                   ! Calibrates Sensor 1 Power Sweep Calibration calibrates the
                                      ! sensor to the meter.
1540  !
1541 PRINT Calibrating.........
1550  OUTPUT @Pwr_mtr;CAL1?           ! Starts Calibration and delivers a pass/fail flag
1560  ENTER @Pwr_mtr;Pass_cal
1570  IF Pass_cal=0 THEN
1580  PRINT Calibration Passed
1590  Old_snsr1_model=New_snsr1_model
1600  Old_snsr1_serno=New_snsr1_serno
1610  ELSE
1620  PRINT Calibration FAILED, Sensor 1 Attached To Calibrator?
1630  WAIT 10                         ! Wait for Operator to attach sensor
1640  GOTO Pwr_swp_cal1
1650  END IF
1660  RETURN
1670  !
1680  Pwr_swp_cal2:! Calibrates Sensor 2
1690  !
1691 PRINT Calibrating........
1700  OUTPUT @Pwr_mtr;CAL2?           ! Starts Calibration and delivers a pass/fail flag
1710  ENTER @Pwr_mtr;Pass_cal
1720  IF Pass_cal=0 THEN
1730  PRINT Calibration Passed
1740  Old_snsr2_model=New_snsr2_model
1750  Old_snsr2_serno=New_snsr2_serno
1760  ELSE
1770  PRINT Calibration FAILED, Sensor 2 Attached To Calibrator?
1780  WAIT 10                         ! Wait for Operator to attach sensor
1790  GOTO Pwr_swp_cal2
1800  END IF
1810  RETURN
1940  END
```

B

Power Sensors

B.1 Introduction

This appendix contains the selection, specifications and calibration data for power sensors used with Giga-tronics power meters. This appendix is divided into the following major sections:

- Power Sensor Selection
- Power Sensor Calibration

B.2 Power Sensor Selection

Standard Series 803XXA Sensors measure CW signals from -70 to +20 dBm; and the Series 804XXA Sensors measure modulated or CW signals from -67 to +20 dBm; the 58542 VXIbus Universal Power Meter also use Peak Power Sensors for measuring radar and digital modulation signals.

Giga-tronics True RMS sensors are recommended for applications such as measuring quadrature modulated signals, multi-tone receiver intermodulation distortion power, noise power, or the compression power of an amplifier. These sensors include a pad to attenuate the signal to the RMS region of the diode's response. This corresponds to the -70 dBm to -20 dBm linear operating region of Standard CW Sensors. The pad improves the input VSWR to ≤1.15 at 18 GHz.

High Power (1, 5, 25 and 50 Watt) and Low VSWR sensors are also available for use with the power meter.

Table B-1 lists the Giga-tronics power sensors used with the power meters. Refer to applicable notes on page B-4. See Figures B-1 for modulation-induced measurement uncertainty.

B.2.1 Modulation Power Sensors

Table B-1: Power Sensor Selection Guide

Model	Freq. Range/ Power Range	Max. Power	Power Linearity[4] (Freq > 8 GHz)	RF Conn.	Length	Dia.	Wgt.	VSWR
Standard CW Sensors								
80301A	10 MHz to 18 GHz -70 to +20 dBm	+23 dBm (200 mW)	-70 to -20 dBm ±0.00 dB -20 to +20 dBm ±0.05 dB/10 dB	Type N(m) 50Ω	114.5 mm (4.5 in)	32 mm (1.25 in)	0.18 kg (0.4 lb)	1.12:0.01 - 2 GHz 1.22:2 - 12.4 GHz 1.29:12.4 - 18 GHz
80302A				APC-7 50Ω				
80303A	10 MHz to 26.5 GHz -70 to +20 dBm		-70 to +20 dBm ±0.00 dB -20 to +20 dBm ±0.1 dB/10 dB	Type K(m)1 50Ω				1.12:0.01 - 2 GHz 1.22:2 - 12.4 GHz 1.38:12.4 - 18 GHz 1.43:18 - 26.5 GHz 1.92:26.5 - 40 GHz
80304A	10 MHz to 40 GHz -70 to 0 dBm		-70 to -20 dBm ±0.00 dB -20 to 0 dBm ±0.2 dB/10 dB					
Low VSWR CW Sensors								
80310A	10 MHz to 18 GHz -64 to +26 dBm	+29 dBm (800 mW)	-64 to -14 dBm ±0.00 dB -14 to + 26 dBm ±0.05 dB/10 dB	Type K(m)[1] 50W	127mm (5.0 in)	32 mm (1.25 in)	0.23 kg (0.5lb)	1.13:0.01 - 2 GHz 1.15:2 - 12 GHz 1.23:12 - 18 GHz 1.29:18 - 26.5 GHz 1.50:26.5 - 40 GHz
80313A	10 MHz to 26.5 GHz -64 to +26 dBm		-64 to -14 dBm ±0.00 dB -14 to + 26 dBm ±0.1 dB/10 dB					
80314A	10 MHz to 40 GHz -64 to +6 dBm		-64 to -14 dBm ±0.00 dB -14 to + 6 dBm ±0.2 dB/10 dB					
1W CW Sensors								
80320A	10 MHz to 18 GHz -60 to +30 dBm	+30 dBm (1 W)	-60 to -10 dBm ±0.00 dB -10 to +30 dBm ±0.05 dB/10 dB	Type K(m)[1] 50Ω	127 mm (5.0 in)	32 mm (1.25 in)	0.23 kg (0.5 lb)	1.11:0.01 - 2 GHz 1.12:2 - 12 GHz 1.18:12 - 18 GHz 1.22:18 - 26.5 GHz 1.36:26.5 - 40 GHz
80323A	10 MHz to 26.5 GHz -60 to +30 dBm		-60 to -10 dBm ±0.00 dB -10 to +30 dBm ±0.1 dB/10 dB					
80324A	10 MHz to 40 GHz -60 to +10 dBm		-60 to -10 dBm ±0.00 dB -10 to +10 dBm ±0.2 dB/10 dB					

Table B-1: Power Sensor Selection Guide (Continued)

Model	Freq. Range/ Power Range	Max. Power	Power Linearity[4] (Freq > 8 GHz)	RF Conn.	Length	Dia.	Wgt.	VSWR
5W CW Sensor[2]								
80321A	10 MHz to 18 GHz -50 to +37 dBm	+37 dBm (5 W)	-50 to +0 dBm ±0.00 dB 0 to +37 dBm ±0.05 dB/10 dB	Type N(m) 50Ω	150 mm (5.9 in)	32 mm (1.25 in)	0.23 kg (0.5 lb)	1.20:0.01 - 2 GHz 1.25:6 - 12.4 GHz 1.35:12.4 - 18 GHz
25W CW Sensor[3]								
80322A	10 MHz to 18 GHz -40 to +44 dBm	+44 dBm (25 W)	-40 to +10 dBm ±0.00 dB +10 to +44 dBm ±0.05 dB/10 dB	Type N(m) 50Ω	230 mm (9.0 in)	104 mm (4.1 in)	0.3 kg (0.6 lb)	1.20:0.01 - 2 GHz 1.30:6 - 12.4 GHz 1.40:12.4 - 18 GHz
50W CW Sensor[3]								
80325A	10 MHz to 18 GHz -40 to +47 dBm	+47 dBm (50 W)	-40 to +10 dBm ±0.00 dB +10 to +47 dBm ±0.05 dB/10 dB	Type N(m) 50Ω	230mm (9.0 in)	104 mm (4.1 in)	0.3 kg (0.6 lb)	1.25:0.01 - 2 GHz 1.35:6 - 12.4 GHz 1.45:12.4 - 18 GHz
True RMS Sensors (-30 to +20 dBm)								
80330A 80333A 80334A	10 MHz to 18 GHz 10 MHZ to 26.5 GHz 10 MHz to 40 GHz	+33 dBm (2 W)	-30 to +20 dBm ±0.00 dB	Type K(m)[1] 50Ω	152.5 mm (6.0 in)	32 mm 1.25 in)	0.27 kg (0.6 lb)	1.12:0.01 - 12 GHz 1.15:12 - 18 GHz 1.18:18 - 26.5 GHz 1.29:26.5 - 40 GHz
80340 Series Peak Power Sensors (-30 to +20 dBm)								
80340A	50 MHz to 18 GHz	+23 dBm (200 mW)	-30 to -20 dBm ±0.13 dB 0 to +20 dBm	Type N(m)[1] 50Ω	146 mm (5.75 in)	37 mm (1.44 in)	0.3 kg (0.6lb)	1.12:0.01 - 2 GHz 1.22:2 - 12.4 GHz 1.37:12.4 - 18 GHz
80343A 80344A	50 MHz to 26.5 GHz 50 MHz to 40 GHz		±0.13 dB ±0.01 dB/dB	Type K(m)[1] 50Ω				1.50:18 - 26.5 GHz 1.92:26.5 - 40 GHz

Notes:

1. The K connector is electrically and mechanically compatible with the APC-3.5 and SMA connectors.
2. Power coefficient equals <0.01 dB/Watt.
3. Power coefficient equals <0.015 dB/Watt.
4. For frequencies above 8 GHz, add power linearity to system linearity.
5. Peak operating range above CW maximum range is limited to <10% duty cycle.
6. Includes uncertainty of reference standard and transfer uncertainty. Directly traceable to NIST.
7. Square root of sum of the individual uncertainties squared (RSS).
8. Cal Factor numbers allow for 3% repeatability when connecting attenuator to sensor, and 3% for attenuator measurement uncertainty and mismatch of sensor/pad combination. Attenuator frequency response is added to the Sensor Cal Factors which are stored in the sensor's EEPROM.

Table B-2: Power Sensor Cal Factor Uncertainties

Freq. (GHz)		Sum of Uncertainties (%)[6]						Probable Uncertainties (%)[7]					
Lower	Upper	80301A 80302A 80340 80401A 80402 80303 80304 80343 80344	80310A 80313A 80314A	80320A 80323A 80324A	80321A[8] 80322A[8] 80325A[8]	80330A 80333A 80334A	80301 80302 80340 80401A 80402 80303 80304 80343 80344	80310A 80313A 80314A	80320A 80323A 80324A	80321A[8] 80322A[8] 80325A[8]	80330A 80333A 80334A	80321A[8] 80322A[8] 80325A[8]	80330A[8] 80333A[8] 80334A[8]
0.1	1	1.61	3.06	2.98	2.96	7.61	2.95	1.04	1.64	1.58	1.58	4.54	1.58
1	2	1.95	3.51	3.58	3.57	7.95	3.55	1.20	1.73	1.73	1.73	4.67	1.73
2	4	2.44	4.42	4.33	4.29	8.44	4.27	1.33	1.93	1.91	1.91	4.89	1.90
4	6	2.67	4.74	4.67	4.63	8.67	4.60	1.41	2.03	2.02	2.01	5.01	2.01
6	8	2.86	4.94	4.87	4.82	8.86	4.80	1.52	2.08	2.07	2.06	5.12	2.06
8	12.4	3.59	6.04	5.95	5.90	9.59	5.87	1.92	2.55	2.54	2.53	5.56	2.53
12.4	18	4.09	6.86	6.76	6.69	10.09	6.64	2.11	2.83	2.80	2.79	5.89	2.78
18	26.5	––	9.27	9.43	9.28	––	9.21	––	3.63	3.68	3.62	––	3.59
26.5	40	––	15.19	14.20	13.86	––	13.66	––	6.05	5.54	5.39	––	5.30

Notes:

1. The K connector is electrically and mechanically compatible with the APC-3.5 and SMA connectors.
2. Power coefficient equals <0.01 dB/Watt.
3. Power coefficient equals <0.015 dB/Watt.
4. For frequencies above 8 GHz, add power linearity to system linearity.
5. Peak operating range above CW maximum range is limited to <10% duty cycle.
6. Includes uncertainty of reference standard and transfer uncertainty. Directly traceable to NIST.
7. Square root of sum of the individual uncertainties squared (RSS).
8. Cal Factor numbers allow for 3% repeatability when connecting attenuator to sensor, and 3% for attenuator measurement uncertainty and mismatch of sensor/pad combination. Attenuator frequency response is added to the Sensor Cal Factors which are stored in the sensor's EEPROM.

Modulation-Induced Measurement Uncertainty for the 80401A Sensor

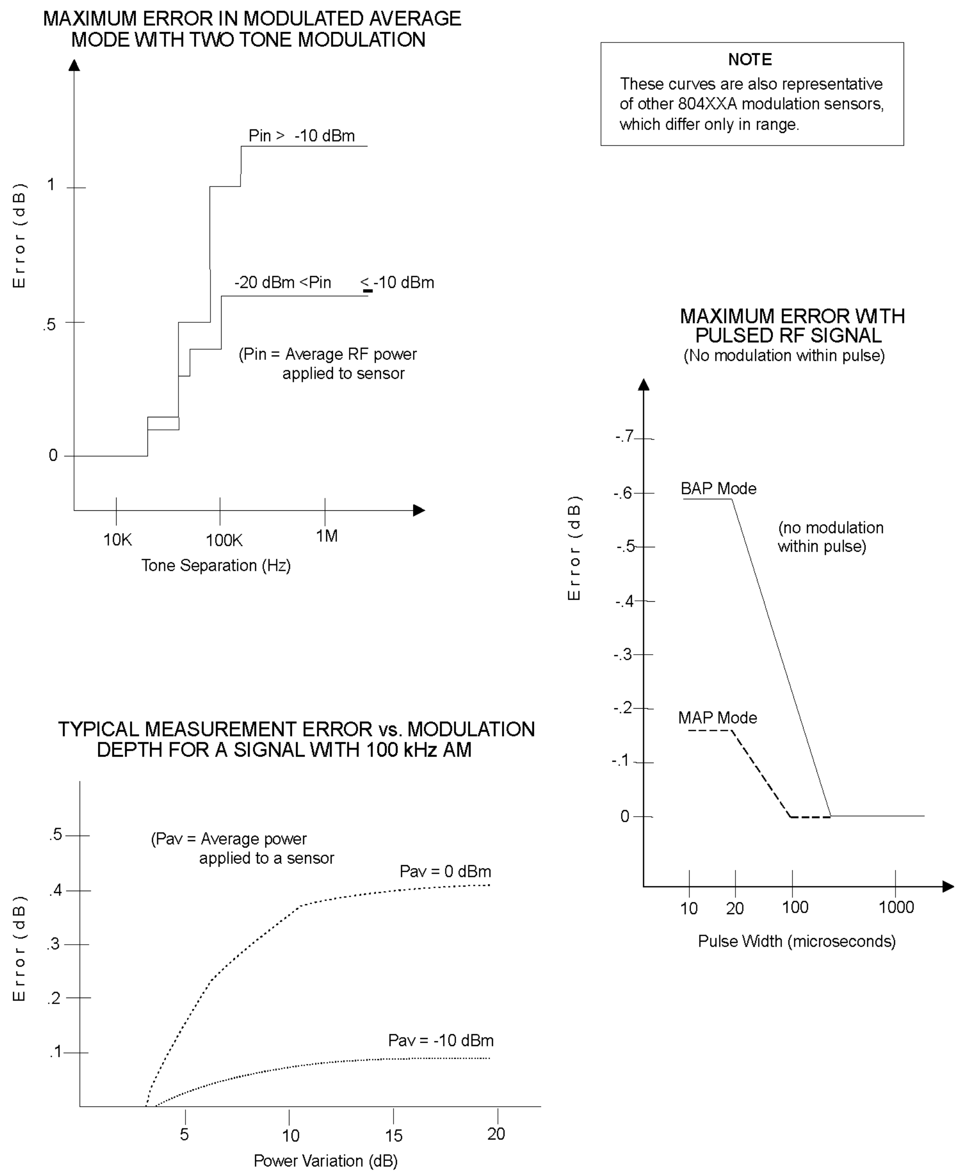


Figure B-1: 80401A Modulation-Related Uncertainty

B.2.2 BAP Mode Limitations

The minimum input level is -40 dBm (average); the minimum pulse repetition frequency is 20 Hz. If the input signal does not meet these minima, **BURST AVG** LED will flash to indicate that the input is not suitable for BAP measurement. The power meter will continue to read the input but the BAP measurement algorithms will not be able to synchronize to the modulation of the input; the input will be measured as if the power meter were in MAP mode. In addition, some measurement inaccuracy will result if the instantaneous power within the pulse falls below -43 dBm; however, this condition will not cause LED to flash.

B.2.3 Peak Power Sensors

Table B-3: 8035XA Peak Power Sensor Selection Guide

Peak Power Sensors								
Model	Freq. Range/ Power Range	Max. Power	Power Linearity[4]	RF Conn.	Dimensions		Wgt.	VSWR
					Length	Dia.		
Standard Peak Power Sensors								
80350A	45 MHz to 18 GHz -20 to +20 dBm, Peak -30 to +20 dBm, CW	+23 dBm (200 mW) CW or Peak	-30 to -20 dBm ±0.00 dB -20 to +20 dBm ±0.05 dB/10 dB	Type N(m) 50Ω	165 mm (6.5 in)	37 mm 1.25 in)	0.3 kg (0.7 lb)	1.12:0.045 - 2 GHz 1.22:2 - 12.4 GHz 1.37:12.4 -18 GHz
80353A	45 MHz to 26.5 GHz -20 to +20 dBm, Peak -30 to +20 dBm, CW		-30 to -20 dBm ±0.00 dB -20 to +20 dBm ±0.1 dB/10 dB	Type K(m)[1] 50Ω				1.12:0.045 - 2 GHz 1.22:2 - 12.4 GHz 1.37:12.4 -18 GHz 1.50:18 - 26.5 GHz
80354A	45 MHz to 40 GHz -20 to +0.0 dBm, Peak -30 to +0.0 dBm, CW		-30 to -20 dBm ±0.00 dB -20 to 0.0 dBm ±0.2 dB/10dB					1.12:0.045 - 2 GHz 1.22:2 - 12.4 GHz 1.37:12.4 -18 GHz 1.50:18 - 26.5 GHz 1.92:26.5 - 40 GHz
5W Peak Power Sensor [2,5]								
80351A	45 MHz to 18 GHz 0.0 to +40 dBm, Peak -10 to +37 dBm, CW	CW: +37 dBm (5 W Avg.) Peak: +43 dBm	-10 to +0 dBm ±0.00 dB +0 to +40 dBm ±0.05 dB/10 dB	Type N(m) 50Ω	200 mm (7.9 in)	37 mm (1.25 in)	0.3 kg (0.7 lb)	1.15:0.045 - 4 GHz 1.25:4 - 12.4 GHz 1.35:12.4 -18 GHz
25W Peak Power Sensor [3,5]								
80352A	45 MHz to 18 GHz +10 to +50 dBm, Peak 0.0 to +44 dBm, CW	CW: +44 dBm (25 W Avg.) Peak: +53 dBm	0.0 to +10 dBm ±0.00 dB +10 to +50 dBm ±0.05 dB/10 dB	Type N(m) 50Ω	280 mm (11.0 in)	104 mm (4.1 in)	0.3 kg (0.7 lb)	1.20:0.045 - 6 GHz 1.30:6 - 12.4 GHz 1.40:12.4 -18 GHz
50W Peak Power Sensor[3,5]								
80355A	45 MHz to 18 GHz +10 to +50 dBm, Peak 0.0 to +47 dBm, CW	CW: +47 dBm (50 W Avg.) Peak: +53 dBm	0.0 to +10 dBm ±0.00 dB +10 to +50 dBm ±0.05 dB/10 dB	Type N(m) 50Ω	280 mm (11.0 in)	104 mm (4.1 in)	0.3 kg (0.7 lb)	1.25:0.045 - 6 GHz 1.35:6 - 12.4 GHz 1.45:12.4 -18 GHz

Notes:

1. The K connector is electrically and mechanically compatible with the APC-3.5 and SMA connectors.
2. Power coefficient equals <0.01 dB/Watt (AVG).
3. Power coefficient equals <0.015 dB/Watt (AVG).
4. For frequencies above 8 GHz, add power linearity to system linearity.
5. Peak operating range above CW maximum range is limited to <10% duty cycle.

Table B-4: Peak Power Sensor Cal Factor Uncertainties

Freq. (GHz)		Sum of Uncertainties (%)[1]					Probable Uncertainties (%)[2]		
Lower	Upper	80350A	80353A 80354A	80351A[3]	80352A[3]	80355A[3]	80350A	80353A 80354A	80351A[3] 80352A[3] 80355A[3]
0.1	1	1.61	3.06	9.09	9.51	10.16	1.04	1.64	4.92
1	2	1.95	3.51	9.43	9.85	10.50	1.20	1.73	5.04
2	4	2.44	4.42	13.10	13.57	14.52	1.33	1.93	7.09
4	6	2.67	4.74	13.33	13.80	14.75	1.41	2.03	7.17
6	8	2.86	4.94	13.52	13.99	14.94	1.52	2.08	7.25
8	12.4	3.59	6.04	14.25	14.72	15.67	1.92	2.55	7.56
12.4	18	4.09	6.86	19.52	20.97	21.94	2.11	2.83	12.37
18	26.5	——	9.27	——	——	——	——	3.63	——
26.5	40	——	15.19	——	——	——	——	6.05	——

Notes:

1. Includes uncertainty of reference standard and transfer uncertainty. Directly traceable to NIST.
2. Square root of sum of the individual uncertainties squared (RSS).
3. Cal Factor numbers allow for 3% repeatability when connecting attenuator to sensor, and 3% for attenuator measurement uncertainty and mismatch of sensor/pad combination. Attenuator frequency response is added to the Sensor Cal Factors which are stored in the sensor's EEPROM.

For additional specifications, see the Series 80350A (P/N 21568) publication and data sheet.

B.2.4 Directional Bridges

The 80500 CW Directional Bridges are designed specifically for use with Giga-tronics power meters to measure the Return Loss/SWR of a test device. Each bridge includes an EEPROM which has been programmed with Identification Data for that bridge.

Table B-5: Directional Bridge Selection Guide

Precision CW Return Loss Bridges								
Model	**Freq. Range/ Power Range**	**Max. Power**	**Power Linearity[4]** (Frequency > 8 GHz)	**Input**	**Test Port**	**Directivity**	**Wgt.**	**VSWR**
80501	10 MHz to 18 GHz -35 to +20 dBm	+27 dBm (0.5W)	-35 to +10 dBm ±0.1 dB +10 to +20 dBm ±0.1 dB ±0.005 dB/dB	Type N(f) 50 Ω	Type N(f) 50 Ω	38 dB	0.340 kg	< 1.17:0.01 - 8 GHz < 1.27:8 - 18 GHz
80502	10 MHz to 18 GHz -35 to +20 dBm	+27 dBm (0.5 W)	-35 to +10 dBm ±0.1 dB +10 to +20 dBm ±0.1 dB ±0.005 dB/dB	Type N(f) 50 Ω	APC-7(f) 50 Ω	40 dB	0.340 kg	< 1.13:0.01 - 8 GHz < 1.22:8 - 18 GHz
80503	10 MHz to 26.5 GHz -35 to +20 dBm	+27 dBm (0.5 W)	-35 to +10 dBm ±0.1 dB +10 to +20 dBm ±0.1 dB ±0.005 dB/dB	SMA(f) 50 Ω	SMA(f) 50 Ω	35 dB	0.340 kg	< 1.22:0.01 - 18 GHz < 1.27:8 - 26.5 GHz
80504	10 MHz to 40 GHz -35 to +20 dBm	+27 dBm (0.5 W)	-35 to +10 dBm ±0.1 dB +10 to +20 dBm ±0.1 dB ±0.005 dB/dB	Type K(f) 50 Ω	Type K(f) 50 Ω	30 dB	0.198 kg	< 1.35:0.01 - 26.5 GHz < 1.44:26.5 - 40 GHz

The Selection Guide in Table B-5 shows primary specifications. Additional specifications are:

Bridge Frequency Response

Return loss measurements using the 8541/8542 power meter can be frequency compensated using the standard *Open/Short* supplied with the bridge.

Insertion Loss

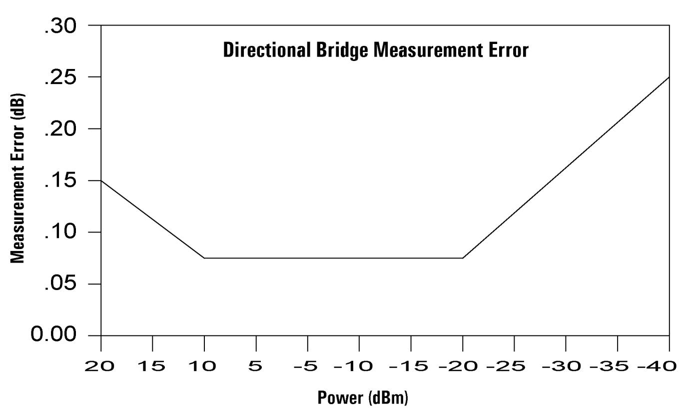
6.5 dB, nominal, from input port to test port

Maximum Input Power

+27 dBm (0.5 W)

Directional Bridge Linearity Plus Zero Set & Noise vs. Input Power (50 MHz, 25 °C ±5 °C)

See Table below



Dimensions

80501	76 x 50 x 28 mm (3 x 2 x 1-$^1/_8$ in.)
80502	76 x 50 x 28 mm (3 x 2 x 1-$^1/_8$ in.)
80503	19 x 38 x 29 mm ($^3/_4$ x 1-$^1/_2$ x 2-$^1/_8$ in.)
80504	19 x 38 x 29 mm ($^3/_4$ x 1-$^1/_2$ x 2-$^1/_8$ in.)

Weight

80501	340 g (12 oz.)
80502	340 g (12 oz.)
80503	198 (7 oz.)
80504	198 (7 oz.)

Directional Bridge Accessories

An Open/Short is included for establishing the 0 dB return loss reference during path calibration.

B.3 Power Sensor Calibration

This procedure is for calibrating a power sensor by remote control with a 58542 VXIbus Universal Power Meter over the IEEE 488 interface bus. This procedure writes the cal factors to the sensor EEPROM.

Power sensors have built-in EEPROM data that manage the cal factors by a set of frequencies entered during calibration of the sensor at the factory. The user can program additional cal factors with special data for user-specific frequencies. A cal factor expressed in dB is programmed for each factory-calibrated frequency. The calibration process compares the measurement to the frequency standard and applies the cal factor to offset frequency deviations.

B.3.1 Equipment Required

58542 VXIbus Universal Power Meter • GPIB Controller • Power Sensor

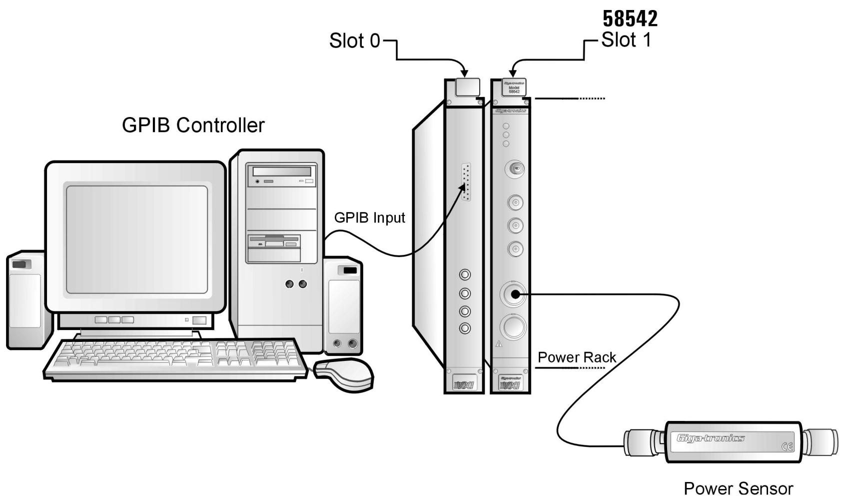


Figure B-1: Power Sensor Calibration Setup

B.3.2 Procedure

Connect the power sensor to Channel 1 or 2 on the 58542 front panel and perform the following steps. In this procedure, bold letters are commands; the query form of a command has a question mark (?) at the end of the command. This form returns the data in the EEPROM.

1. **DIAG:SENS1 (or 2):EEPROM:READ**

   Read sensor 1 (or 2) EEPROM data into the 58542 editor buffer.

   Example: OUTPUT@ PWR_MTR; DIAG:SENS1:EEPROM:READ

2. (Optional) **DIAG:SENS < 1 OR 2 > :EEPROM:CALFR?**

   a.) Query sensor 1 (or 2) standard cal factor start frequency, number of standard frequencies and number of special frequencies.

   b.) Read the standard cal from the input buffer and extract the start frequency and number of standard frequencies.

   c.) Calculate and set the frequencies of the cal factor table.

3. **DIAG:SENS < 1 or 2 > :EEPROM:CALFST?**

   a.) Query sensor 1 (or 2) standard cat factor table.

   b.) Read the standard cal from the input buffer and extract the standard cal factor; e.g., INPUT (GPIB address).

   c.) DIAG:SENS<1 OR 2>:EEPROM:CALFST*space*"0.02, 0.03,0.04,0.05,0.06,0.07,0.08,0.09,0.10,0.11,0.12,0.13,0.14,0.15,0.16,0.17,0.18" form to write to editor buffer.

   d.) Set the sensor standard cal factor table.

   e.) Make changes from the table and put them back into the table.

   f.) After all changes are made, put the table back into the input buffer.

4. **DIAG:SENS < 1 or 2 > :EEPROM:WRITE 0**

   a.) Write sensor 1 (or 2) EEPROM data into the 58542 buffer. If no password has been set up, use 0 for a password.

   b.) Restore the input buffer from step 3.d above to the EEPROM buffer (e.g., OUTPUT [GPIB] address, input buffer).

   c.) Write sensor 1 (or 2) editor buffer data into the EEPROM with the password number, e.g., OUTPUT (GPIB address, DIAG:SENS1 (OR 2):EEPROM:WRITE 0).

   d.) Editing the EEPROM routine is complete.

Sensor EEPROM Code Sequence Writing:

(The following program example is written in Microsoft® Visual Basic using National Instruments® VISA Instrument calls)

```
Private Sub cmdWriteNewData_Click()
   'Write new calibration data to sensor for form

   cmdWriteNewData.Caption = "Writing"

   wrt = "DIAG:SENS" + SensNum + ":EEPROM:TYPE " + space$(1) _
      + Chr$(34) + txtModelNum.Text + "," + txtSerNum.Text + "," + _
      txtCalLoc.Text + "," + MinVal + "," + HourVal + "," + DayVal + _
      "," + MonthVal + "," + YearVal + "," + PWDVal + Chr$(34)

   stat = viWrite(MVISAaddress, wrt, Len(wrt), retCnt)

   SensorWrite 'Call Write new type data

   wrt = "DIAG:SENS" + SensNum + ":EEPROM:CALFR" + space$(1) + _
      Chr$(34) + "2.000e9,1.000e9,17,1" + Chr$(34) 'txtCalFreq.Text

   stat = viWrite(MVISAaddress, wrt, Len(wrt), retCnt)

   SensorWrite 'Call Write new range data

   wrt = "DIAG:SENS" + SensNum + ":EEPROM:CALFST" + space$(1) _
      + Chr$(34) + Str(txtCF1) + "," + Str(txtCF2) + "," _
      + Str(txtCF3) + "," + Str(txtCF4) + "," + Str(txtCF5) + _
      "," + Str(txtCF6) + "," + Str(txtCF7) + "," + Str(txtCF8) _
      + "," + Str(txtCF9) + "," + Str(txtCF10) + "," + _
      Str(txtCF11) + "," + Str(txtCF12) + "," + Str(txtCF13) + _
      "," + Str(txtCF14) + "," + Str(txtCF15) + "," + Str(txtCF16) _
      + "," + Str(txtCF17) + Chr$(34)

   stat = viWrite(MVISAaddress, wrt, Len(wrt), retCnt)

   SensorWrite 'Call Write new cal factor data

   stat = viClear(MVISAaddress)
   rd = ""
   Beep

   cmdWriteNewData.Caption = "Write New Information"
End Sub

Public Sub SensorWrite()
   wrt = "DIAG:SENS" + SensNum + ":EEPROM:WRIT 0"

   stat = viWrite(MVISAaddress, wrt, Len(wrt), retCnt)

   Sleep (5000) 'Wait five seconds

End Sub
```

C

Options

C.1 Introduction

This appendix describes all options that are currently available for use with the 58542 VXIbus Universal Power Meter.

C.2 Option 02: 256K Buffer

Option 02 (P/N 21335) adds 256K Buffer for Burst Mode readings. The following parts will be installed at the factory when the 58542 is ordered with this option.

21335		EXTRA MEMORY, 128K, 8540/02, REV. A				
Item		**P/N**	**Qty**	**Cage**	**Mfr's P/N**	**Description**
U	29	21165	1	61802	TC551001BPL-85	TC551001BPL-10 1M RAM
U	30	21165	1	61802	TC551001BPL-85	TC551001BPL-10 1M RAM

Index

58542 VXIbus Universal Power Meter

Numerics

A

B

C

D

E

G

H

I

L

M

O

P

V

Z