AUSTRALIA TELESCOPE

AT TECHNICAL REPORT No. E1

CULGOORA WEATHER STATION

M.J. KESTEVEN

AUSTRALIA TELESCOPE

CSIRO DIVISION OF RADIOPHYSICS

CNR VIMIERA & PEMBROKE RDS, EPPING NSW

29/6/88

Table of Contents

Culgoora Weather Station

M. Kesteven

1 INTRODUCTION

An automated weather station has now been installed at Culgoora. A set of eleven weather statistics are recorded every hour:

a). wind speed - the mean, maximum, minimum and variance from the previous hour;

b). wind direction - hourly mean;

c). temperatures at ground level and at 10m level - five minute mean;

d). solar radiation - hourly mean;

e). barometric pressure - hourly mean;

f). rainfall - total accumulated during the previous hour;

g). humidity - hourly mean.

Additional software is provided to allow the archived data to be examined and listed.

This manual describes the operation of the unit, the hardware involved and the maintenance.

2 OPERATION OF THE STATION

The station is automatic - operator intervention is required only to start (or restart) the program, and to supervise the data archiving once a month. The data is written to a 5" floppy diskette every hour (on the hour). Although the diskette's capacity is adequate for several months operation, it would be wise to replace the diskette every month. The PC display maintains a tally of the amount of data stored on the diskette.

2.1 Startup

a). The sensors

The solar radiation monitor requires 240 v. There is a switch in the junction box at the base of the weather station tower; power to the tower derives from a switch board in the control building.

Two other sensors require power: the anemometer and the barometer. In both cases 24 v DC is needed, derived from the logger rack.

The remaining sensors are passive, and need no action to start operation.

b). The logger

The logger executes a scanning program: every 15 seconds it reads the sensors, scales the data to useful units and then sends the results to the PC. Switching on the logger will cause the program to be transferred from a cartridge tape; the program will then be started automatically. Recovery from a drastic power failure is equally automatic.

The logger has a key switch on the front panel. This needs to be set to the ON position. Two LEDs adjacent to the switch should be lit: one to indicate that the 24 v battery supply is on; the second LED indicates the operational state of the logger (ON or OFF).

Thus the operator's function on start-up is simple: ensure that the tape is in the reader, then switch the unit to ON.

c). The PC

The PC executes a simple task to accept the data from the logger, compile the statistics, and every hour, on the hour, archive the data.

The startup sequence:

1. Place the system in the upper drive, and the data disk in the lower drive.

2. Switch on the machine; after some diagnostics the computer will display its estimate of time and date. The operator should check that these are correct, and enter either a confirming RETURN or else the correct time/date. **NOTE:** the computer runs on **AEST**, and not **Summer time.**

3. Type:

```
GWBASIC
LOAD "ORION.BAS"
RUN
```

4. The screen should then be cleared, and the current weather values displayed, accumulating over a period of time.

2.2 Restart

It is possible that a failure may occur after extended power failures, or after violent storms. In this case the logger and/or the PC may need restarting.

a). Logger : switch off, then ON.

b). PC. : type RUN

This may produce the error message : "Communications buffer overflow". The response to this is to hit a number of RETURNS, then type: RUN

If all this fails, then switch off the computer and start again.

2.3 Backups

A backup logger program tape and a backup system disk for the PC are kept beside the units. Both are labelled as "Backup".

2.4 Data Archiving

The data is written to disk automatically, and needs no intervention. The PC display shows the time and date of the last record written; this should be checked periodically as a guide to the health of the PC operations.

Once a month the disk should be replaced with a fresh disk. Remove the disk from the lower drive and insert a fresh one (pre-initialised). This can be done at any time provided that the PC is not attempting to write to the disk - ie, check the time.

Spare (pre-initialised) disks are kept beside the machine.

Initialising the disks: use DISKCOPY on any other PC, with a pre-initialised disk as model.

The Archive disks should be sent to M. Kesteven.

2.5 Examining Archived Data

a). Find a spare PC.

b). Place the WEATHER system disk in the upper drive, and the archive data disk in the lower.

c). Type WEATHER to start the task.

d). Data filename: the standard filename is

B:WEATHER.DAT

e). Data can be listed and/or printed.

This task can be run on the current data disk: this can be removed from the logging PC provided that no data is to be written to the disk - ie. at times other than on the hour. Check the PC display.

3 HARDWARE OVERVIEW

The station consists of:

a). The sensors.

b). An "ORION" data logger which interrogates the sensors.

c). A P.C. computer which records the data and compiles some statistics.

The logger and the computer are battery powered so the weather station will continue to operate during power failures; however, the solar radiation monitor requires 240 volts power, so the solar radiation data may not be continuous.

4 THE SENSORS

4.1 The Anemometer

This is a WEATHERTRONICS device, model # 2030

It has three cups, 5 cm in diameter, on 8 cm arms. The rotating shaft carries a 30 slot optical chopper. An OPTRON optoelectronic chopper (model OPB 804) detects the rotation of the shaft.

The scaling adopted : V (km/hr) = F (hz) * 0.158 + 0.837
is derived from the WEATHERTRONICS manual.

The wiring diagram for the unit is shown in fig. 1

24 v DC is required; this is derived from the logger rack power suply.

4.2 The Wind Direction Monitor

This is a WEATHERTRONICS device, model #2020

The weather vane drives a 5kΩ variable resistor.

4.3 The Temperature Probes

These are all four-wire platinum resistor thermometers. Although NML calibration curves are available for all units, we have chosen to use the calibration factors built into the logger.

4.4 Solar Radiation (solarimeter)

This is a CASELLA device, model #W6500

The unit requires +10v and -8v power, derived from a 240v unit. It returns a DC analogue voltage, 1 mV/(W/m^2).

4.5 The Barometer

This is a WEATHERTRONICS device, model #7100

It requires 24v DC supply, provided directly from the logger rack. The unit is mounted on the logger rack, with a hose connected to the outside to provide the ambient pressure. It returns a DC analogue voltage proportional to the pressure.

The scaling : P (mb) = V * 100. + 600. comes from the manual.

4.6 The Rainfall Gauge

This is a WEATHERTRONICS device, model #P501 - I/PN 648001

The unit contains a small tipping bucket, each tip corresponding to 0.25 mm of rain. Each "tip" produces a pulse which is counted by the logger.

4.7 The Hygrometer

Not yet installed.

4.8 CONNECTING THE SENSORS TO THE LOGGER

All the sensors except for the barometer are located on the weather tower to the E of the control building. Four sensors are mounted at the tower top: the anemometer and wind direction; a temperature probe and the hygrometer. A second temperature probe and the solar radiation monitor are at ground level. The rain gauge is some 10m to the south of the tower.

The wiring diagram for the junction box at the tower top is shown in fig. 2; the wiring diagram for the ground level junction box is in fig. 3, and the connections at the logger rack are shown in fig. 4

The anemometer is connected to channel 5 in the Counter/Timer module of the logger, as shown in figs 5 and 6.

The rain gauge is connected to channel 21 in the Event/Status module, as shown in fig. 7

All the other sensors are connected to the Reed relay unit, as shown in figs 8, 9, 10, 11 and 12.

5 PROGRAMMING THE LOGGER

The scanning task in the logger is called "weather", and occupies slot #1 on the tape. The program contains:

5.1 Definition of the scanning task

In ORION terms:

TA1 OP ME	This is a measurement task
TA1 TR TI DE 5.0	Start after a 5 second delay
TA1 CO * RE IN IN 0:0:15.	Continuous scanning, looping every 15 s
TA1 CH1-5,21,61-65,67,68-70	These are the channels to scan
TA1 AT F	Take 10 readings per sensor
TA1 LO EV	Log every channel sampled
TA1 MA EV	And send a time stamp
TA1 FO CO	Using a compact format
TA1 TO S0	Send the data to the RS232 port S0

5.2 Definition of the channels to be interrogated

In ORION terms:

CH1-5 SE 902 CO1 Wind speed; frequency measurement, 1.0 sec gate; Conversion routine #1

CH21 SE 741 CO2	Rain gauge; count events
CH61 SE 410	Ground level T; 4-wire platinum resistor
CH63 SE 410	10m level T
CH65 SE 410	Room T
CH67 SE 254 CO3	Wind direction; resistance measurement
CH68 SE 114 CO4	Solar radiation; voltage measurement
CH69 SE 114 CO5	Barometeric pressure
CH70 SE 114 CO6	Relative Humidity

5.3 Definition of the calibration algorithms and factors to be applied

In ORION terms:

CO1 MO SC MC 0.158, 0.837 UN KMH Conversion routine #1, linear scaling (0.158*x + 0.837),units km/h

CO2 MO SC MC 0.25, 0. UN MM

CO3 MO SC MC 0.072, 0. UN DEG

CO4 MO SC MC 1000.,0. UN WM2

CO5 MO SC MC 100., 600. UN MB

CO6 MO SC MC 1.,0. UN %

5.4 CONVERSING WITH THE ORION

The ORION's RS232 output is currently set to: 300 baud, 7 bits, EVEN parity, 1 stop bit.

A terminal should be connected (in place of the PC).

Communication with the ORION is a bit complicated because of the need to distinguish the two competing output channels which share the same RS232 output port: the microprocessor port and the data port. A simple protocol allows the user to switch from one to the other:

!	to gain attention of the ORION's microprocessor
#	to release the ORION, and open the data path.

The procedure to modify a program, for example would be:

!	This should return	C:
HALT	To stop any current task	C:
TA1 OP ME ... etc		
#	If data is to flow to S0.	

Note: all communication to the ORION **must** be in upper case.

Further details on the commands available are described in the ORION manual.

6 THE PC PROGRAMS

"ORION", the PC logging task is listed in appendix A. The task runs under GWBASIC. (An as yet unresolved problem prevents a compiled version running)

"WEATHER", the task to examine the archival data is listed in appendix B. This task is compilable.

7 HISTORY AND ACKNOWLEDGEMENTS

The weather tower was designed by C.J. van der Meulen. The sensors were installed by E.Broderick, with W.Payten supervising. A.Spencer designed and built the pulse shaper/driver for the anemometer. B.Whitelaw drew the diagrams.

Appendix A

The Data Logging Program

```
10 '
20 '      ORION weather station logging task
30 '
40 '  mjk 11 - NOV - 87
50 '
60 '..........................................
70 '
80    DIM POINTS(8), V.DATA(8)
90    DIM A.NDX(100), B.NDX(100)
95    DIM ARR.DAT(6,10), INDX(100)
100 '
110    FALSE = 0:  TRUE = NOT FALSE
115    PI = 3.1415926#
117    TM.LOG$ = " (none yet)  "
118    KEY OFF
119    T.FACT = .9
120    BLANK$ = STRING$(8,32)
125 '
130  OPEN "COM1:" AS #1
140 '
150  SKIP = 5
160 '
165   GOSUB 2000        : ' task initioaliser
170   GOSUB 1250        : ' screen initialiser
180 '
190 '
200   FIRST.REC = TRUE
210 '  simple minded loop
220 '
230     MSG$ = " "
240 '
250  WHILE LOC(1)<>0
260    A$ = INPUT$(1, #1)
270    IF (A$ = CHR$(13)) GOTO 370
280    IF (A$ = CHR$(10)) GOTO 370
290     MSG$ = MSG$ + A$
300 '
310     GOSUB 560
320    GOTO 500
330 '
340 '
350 '
```

```
360 '
370 '              decide if there is anything worth printing
380 '
390  IF (NOT FIRST.REC) GOTO 460
420  MSG$ = " "
430  FIRST.REC = FALSE
440  GOTO 510
450 '
460      GOSUB 660
470      GOSUB 1100
480      MSG$ = " "
490 '
500 '
510   WEND
520 '
521   THIS.HR = VAL (MID$(TIME$,1,2))
522   IF (THIS.HR <> LAST.HR) THEN GOSUB 4000
523 '
524   A$ = INKEY$
525   IF ((A$="Q") OR (A$="q")) THEN GOTO 545
526 '
530 GOTO 210
540 '
542  GOSUB 4000  :'  write a last record
543 '
545  CLS
546   PRINT:  PRINT "   Weather logger stopped"
547   PRINT:
548   PRINT  "data file now has "; N.RECORDS;  "  records"
550 '
555  END
557 '..........................................................
560 '
570 '    SUBROUTINE TO WASTE TIME
580 '
590      J = 0
600     FOR I = 1 TO SKIP
610      J = J + 1
620     NEXT I
630 '
640 RETURN
645 '..........................................................
650 '
660 '    subroutine to parse an input line
670 '
680     PNT = LEN(MSG$)
690     N1 = 1
700     NTERMS = 0
710 '
```

```
720     FOR I = N1 TO PNT
730      IF (ASC(MID$(MSG$,I,1)) > 32) GOTO 770
740     NEXT I
750     GOTO 1080
760 '
770 '
780    N1 = I
790 '              is this time ?
800 '
810      IF (MID$(MSG$,N1,1) <> "S") GOTO 880
820 '
830       NTERMS = 1
840       POINTS(1) = -1
850        TM.STR$ = MID$(MSG$,N1+6,10)
851        TIME$ = TM.STR$
852        TM.STR$ = TIME$
860       GOTO 1080
870 '
880 '              ":" ?
890      IF (MID$(MSG$,N1,1) <> ":") GOTO 940
900       N1 = N1 + 19
910       IF (N1 < PNT) GOTO 940
920       GOTO 1080
930 '
940      IF (MID$(MSG$,N1,1) <> "C") GOTO 1030
950       NTERMS = NTERMS + 1
960       PT = VAL(MID$(MSG$,N1+2,3))
962       POINTS(NTERMS) = PT
964       IF ((PT>0) AND (PT<6)) THEN POINTS(NTERMS) = 1
970        V.DATA(NTERMS) = VAL(MID$(MSG$,N1+6,7))
980       N1 = N1 + 20
990       IF (N1 < PNT) GOTO 720
1000 '
1010        GOTO 1080
1020 '
1030     IF (MID$(MSG$,N1,1) = "D") GOTO 1080
1040      PRINT "possible trouble : "
1050      PRINT MSG$
1070 '
1080     RETURN
1085 '..........................................................
1090 '
1100 '  Subroutine to describe the values found
1110 '
1120    IF (NTERMS = 0) GOTO 1220
1130     FOR I = 1 TO NTERMS
1140      IF (POINTS(I) = -1) GOTO 1200
1150       A1 = A.NDX(POINTS(I)) :  B1 = B.NDX(POINTS(I))
1160       LOCATE A1,B1 :  PRINT BLANK$
```

```
1170     LOCATE A1,B1:  PRINT USING "######.#"; V.DATA(I)
1172     J = INDX(POINTS(I))
1175     GOSUB 7000          : ' update the data
1180     GOTO 1210
1190 '
1200     LOCATE 6,33:  PRINT TM.STR$
1210    NEXT I
1220 '
1230  RETURN
1235 '..............................................................
1240 '
1250 ' subroutine to initialise the screen
1260 '
1270  CLS
1280  LOCATE 2,10
1290  PRINT "Culgoora weather station and logger  -   jan6/88 version"
1300 '
1310  LOCATE 3, 60
1320  PRINT DATE$
1330 '
1340  LOCATE 6, 5
1350  PRINT "time of last data sample : "
1360 '
1370  LOCATE 8, 1 :  PRINT "wind speed : "
1383  LOCATE 8, 25:  PRINT "k/hr"
1385  A.NDX(1) = 8:  B.NDX(1) = 15
1390  A.NDX(2) = 8:  B.NDX(2) = 15
1400  LOCATE 8, 40:  PRINT "wind direction : "
1410   LOCATE 8,70:  PRINT "degs"
1420  A.NDX(67) = 8:  B.NDX(67) = 60
1430 '
1440  LOCATE 11,1
1450   PRINT "ground temp : "
1455  LOCATE 11, 25:  PRINT "C"
1460  A.NDX(61) = 11:  B.NDX(61) = 15
1470  LOCATE 11, 40
1480   PRINT "10m temp : "
1485  LOCATE 11, 70:  PRINT "C"
1490  A.NDX(63) = 11:  B.NDX(63) = 60
1500 '
1510  LOCATE 14,1
1520   PRINT "precipitation : "
1525    LOCATE 14,25:  PRINT "mm"
1530   A.NDX(21) = 14:  B.NDX(21) = 15
1540  LOCATE 14, 40
1550   PRINT "humidity : "
1555  LOCATE 14, 70 : PRINT "%"
1560  A.NDX(70) = 14:  B.NDX(70) = 60
1570 '
```

```
1580 LOCATE 17,1
1590  PRINT "pressure : "
1595  LOCATE 17,25 :  PRINT "mb"
1600  A.NDX(69) = 17:  B.NDX(69) = 15
1610  LOCATE 17, 40
1620  PRINT "sol. rad : "
1625  LOCATE 17, 70 : PRINT "W/sqm"
1630  A.NDX(68) = 17:  B.NDX(68) = 60
1640 '
1645  LOCATE 20,1 :  PRINT "room temp : "
1647  LOCATE 20, 25: PRINT "C"
1650  A.NDX(65) = 20:  B.NDX(65) = 15
1670  LOCATE 22,1
1680 PRINT "Last record at "; TM.LOG$;
1682 PRINT "   there are now "; N.RECORDS; " records on disk"
1685  PRINT " This task started on "; STRT.DATE$;
1687  PRINT "   at "; STRT.TIME$
1690 '
1692  LOCATE 25,1
1693  PRINT "Type Q to halt task";
1700  RETURN
1705 '..........................................................
1999 '
2000 '   Subroutine to initialise the various arrays
2010 '
2030 '
2040    STRT.DATE$ = DATE$
2045    STRT.TIME$ = TIME$
2050 '
2060    GOSUB 3000    : '  zeroes the data array
2070 '
2080    INDX(1) = 1   : ' wind speed
2090    INDX(2) = 1   : ' wind speed
2100    INDX(21) = 6  : ' precipitation
2110    INDX(61) = 3  : ' ground level temperature
2120    INDX(63) = 4  : ' 10 m temp
2130    INDX(65) = 5  : ' room temperature
2140    INDX(67) = 2  : ' wind direction
2145    INDX(68) = 9  : ' solar radiation
2150    INDX(69) = 7  : ' pressure
2160    INDX(70) = 8  : ' humidity
2170 '
2180    LAST.HR = VAL (MID$(TIME$,1,2))
2200 '
2210    OPEN "b:records.dat" FOR INPUT AS #3
2220      INPUT #3, N.RECORDS
2230   CLOSE #3
2240 '
2280 RETURN
```

```
2285 '..........................................................
2999 '
3000 ' Subroutine
3010 '                zeroes the data array
3020   FOR I = 1 TO 2
3030    FOR J = 1 TO 6
3040      ARR.DAT(J,I) = 0!
3045      IF (J=3) THEN ARR.DAT(J,I) = 1000000!
3050    NEXT J
3060   NEXT I
3070 '
3080   FOR I = 1 TO 9
3090    FOR J = 4 TO 6
3100      ARR.DAT(J,I) = 0!
3110    NEXT J
3120   NEXT I
3130 '
3135   ARR.DAT(5,3) = 1!
3136   ARR.DAT(5,4) = 1!
3140 '
3180 RETURN
3185 '..........................................................
3999 '
4000 ' Subroutine to log data to disk
4010 '
4020   OPEN "b:weather.dat" FOR APPEND AS #2
4030 '
4040   TM.LOG$ = TIME$
4050 '
4060   PRINT #2, TM.LOG$, DATE$
4070 '
4080 ' gather the statistics
4090 '                wind
4100   J = 1
4110   GOSUB 5000
4120 '
4130 '                wind direction
4140   J = 2
4150   GOSUB 6000
4160 '
4170   J = 3           :' ground temp
4180   GOSUB 5000
4190 '
4200   J = 4           :' 10 m temp
4210   GOSUB 5000
4220 '
4222   J = 7           :' pressure
4224   GOSUB 5000
4226 '
```

```
4230  J = 8              : ' humidity
4240  GOSUB 5000
4250 '
4251  J = 9           : ' INSOLATION
4252  GOSUB 5000
4253 '
4260   PRINT #2, ARR.DAT(4,1), ARR.DAT(5,1), ARR.DAT(2,1),
                  ARR.DAT(3,1)
4270   PRINT #2, ARR.DAT(2,2), ARR.DAT(4,3), ARR.DAT(4,4),
                  ARR.DAT(4,7)
4280   PRINT #2, ARR.DAT(1,6), ARR.DAT(4,8), ARR.DAT(4,9)
4290 '
4300  CLOSE (2)
4310 '
4315   GOSUB 9000    :'  update record number counter
4317 '
4320  GOSUB 3000 : '   zero the data arrays
4330 '
4340   GOSUB 1250 : '   refresh the screen
4350 '
4360   LAST.HR = THIS.HR
4370 '
4390 '
4400  RETURN
4405 '..............................................................
4999 '
5000 '  subroutine
5010 '           generate the statistics
5020 '
5030   WEIGHT = ARR.DAT(6,J)
5040  IF (WEIGHT = 0) GOTO 5150
5050 '
5060   MEAN = ARR.DAT(4,J) / WEIGHT
5070 '
5080   RMS  = ARR.DAT(5,J) / WEIGHT - MEAN*MEAN
5090  IF (RMS < 0) GOTO 5160
5100   RMS = SQR (RMS)
5110  GOTO 5200
5120 '
5150 '         problems
5160   RMS = 0!
5170 '
5200   ARR.DAT(4,J) = MEAN
5210   ARR.DAT(5,J) = RMS
5220 '
5230 RETURN
5235 '..............................................................
5999 '
6000 '   subroutine to deal with  the wind direction
```

```
6010 '
6020   IF (ARR.DAT(1,J) = 0 ) GOTO 6500
6030 '
6040   CC = ARR.DAT(4,J)
6050   SS = ARR.DAT(5,J)
6060 '
6070   GOSUB 8000   : ' get ATAN2D
6160 '
6170    ARR.DAT(2,J) = TT
6180 '
6500 RETURN
6505 '..............................................................
6999 '
7000 ' Subroutine to update the data arrays
7010 '
7015    DATUM = V.DATA(I)
7020    ARR.DAT(1,J) = DATUM
7030 '
7040   IF (J = 2) GOTO 7500  :'  wind direction
7042   IF (J = 3) GOTO 7700  :'  temperature
7044  IF (J = 4) GOTO 7700
7050 '
7060    TMP = ARR.DAT(1,J)
7070    ARR.DAT(4,J) = ARR.DAT(4,J) + TMP
7080    ARR.DAT(5,J) = ARR.DAT(5,J) + TMP*TMP
7090    ARR.DAT(6,J) = ARR.DAT(6,J) + 1
7100 '
7110    IF (ARR.DAT(2,J) < TMP) THEN ARR.DAT(2,J) = TMP
7120    IF (ARR.DAT(3,J) > TMP) THEN ARR.DAT(3,J) = TMP
7130 '
7140 '       check daily wind max/min
7150 '
7160   IF (J <> 1) GOTO 7800
7170 '
7180    IF (ARR.DAT(2,10) < TMP) THEN ARR.DAT(2,10) = TMP
7190    IF (ARR.DAT(3,10) > TMP) THEN ARR.DAT(3,10) = TMP
7200   GOTO 7800
7210 '
7500    T = ARR.DAT(1,2)
7510   T = T * PI / 180!
7520   SPEED = ARR.DAT(1,1)
7530 '
7540   C = COS(T) * SPEED
7550   S = SIN(T) * SPEED
7560 '
7570    ARR.DAT(4,2) = ARR.DAT(4,2) + C
7580    ARR.DAT(5,2) = ARR.DAT(5,2) + S
7590    ARR.DAT(6,2) = ARR.DAT(6,2) + 1
7600 '
```

```
7610  CC = ARR.DAT(4,2)
7620  SS = ARR.DAT(5,2)
7630  GOSUB 8000        : ' get atan2D
7640 '
7650   ARR.DAT(2,2) = TT
7660 '
7665  GOTO 7800
7667 '
7700 '            temperature : want 5 min. averaging
7705 '
7710   ARR.DAT(4,J) = ARR.DAT(4,J)*T.FACT + DATUM
7715   ARR.DAT(6,J) = ARR.DAT(6,J) + ARR.DAT(5,J)
7720   ARR.DAT(5,J) = ARR.DAT(5,J)*T.FACT
7725 '
7800 RETURN
7805 '..............................................................
7999 '
8000 '   subroutine to evaluate ATAN2D
8010 '
8060  IF (CC <> 0) GOTO 8110
8070   TT = PI / 2!
8080   IF (SS < 0) THEN TT = -PI/2!
8090   GOTO 8500
8100 '
8110   TT = SS / CC
8120    TT = ATN(TT)
8130   IF (CC < 0) THEN TT = TT + PI
8140 '
8150   TT = TT * 180! / PI
8160 '
8500 RETURN
8505 '.................................................................
8999 '
9000  'subroutine
9010 '       updates the record counter file
9020 '
9030   OPEN "b:records.dat" FOR INPUT AS #3
9040   INPUT #3, N.RECORDS
9050   CLOSE #3
9060 '
9070   N.RECORDS = N.RECORDS + 1
9080 '
9090   OPEN "b:records.dat" FOR OUTPUT AS #3
9100   PRINT #3, N.RECORDS
9110   CLOSE #3
9120 '
9130 RETURN
```

Appendix B

The Data Examination Program

```
20 '
30 '      Weather data analysis
40 '
50 '      mjk, 7/jan/88
60 '
70 '.....................................................
80 '
90        DIM ARR.DAT(11), WANT(11), L.TITLE$(11), L.UNIT$(11)
100 '
110       FALSE = 0 :  TRUE = NOT FALSE
120 '
130       KEY OFF
140 '
150       GOSUB 2290  :'        initialise titles
160 '
170       CLS :  LOCATE 5,5
180        INPUT " Enter data filename : ", IN.FIL$
190 '
200       OPEN IN.FIL$ FOR INPUT AS #1
210 '
220       LOCATE 7,5 :  PRINT " Patience while we read the file "
230 '
240 '
250 '                find the first record
260 '
270      N.LINE = 0
280      LINE INPUT #1, F.MSG$
290      N.LINE = N.LINE + 1
300      IF (N.LINE>10) GOTO 860          :' probable error
310      IF (MID$(F.MSG$,3,1)<>":") GOTO 280
320 '
330      GOSUB 960           :' read the 3 lines of the first record
340 '
350      N.REC = 1
360      WHILE (NOT EOF(1))
370        LINE INPUT #1, MSG$
380        IF (MID$(MSG$,3,1)<>":") GOTO 370     :' resynch if necessary
390        GOSUB 960           :' read 3 lines of data
400        N.REC = N.REC + 1
410      WEND
420 '
430      CLOSE #1
```

```
440 '
450 '
460      P.DATE$ = "01-01-1900"
470      OPEN IN.FIL$ FOR INPUT AS #1
480 '
485      SKIP = FALSE
490      MORE = TRUE
500      WHILE (MORE)
510 '
520      GOSUB 1360      : '  options setup
530 '
540      IF (NOT MORE) GOTO 800
550 '
560      FOUND = FALSE
570 '
580      WHILE (NOT EOF(1))
590 '
600      IF (NOT SKIP) THEN LINE INPUT #1, T.MSG$
605      SKIP = FALSE
610      IF (MID$(T.MSG$,3,1) <> ":") GOTO 600
620      IF (MID$(T.MSG$,15,10)<>L.DATE$) GOTO 680
630 '
640      GOSUB 1050          :' read and print the data
650      FOUND = TRUE
660      GOTO 710
670 '
680      IF (FOUND) GOTO 730      :' can quit
690 '
700      GOSUB 960           :' skip 3 lines
710      WEND
720 '
730 '
740      PRINT
750 '
760      IN.MSG$ = "Continue"
770      GOSUB 2160          :' asks for yes/no to continue
780      MORE = ANS
790 '
800      WEND
810 '
820      CLOSE #1
830 '
840   END
850 '.......................................................
860 '    error exit
870 '
880     CLOSE #1
890     CLS
900      PRINT
```

```
910       PRINT " No time stamp found"
920       PRINT " Is "; IN.FIL$; "  the correct filename?"
930      PRINT
940     STOP
950 '.....................................................
960 '      subroutine
970 '                reads 3 lines, no data resulting
980 '
990        LINE INPUT #1, C1$
1000       LINE INPUT #1, C2$
1010       LINE INPUT #1, C3$
1020 '
1030  RETURN
1040 '.....................................................
1050 '      subroutine
1060 '                reads 3 lines; extracts 10 items
1070 '
1080 '
1090     FOR I = 1 TO 11
1100       IF ((I=11) AND (L.DATE$<"01-08-1988")) GOTO 1120
1110       INPUT #1, ARR.DAT(I)
1120     NEXT I
1130 '
1140      GOSUB 1180       :' print
1150 '
1160 RETURN
1170 '.....................................................
1180 '     subroutine
1190 '                prints the required data
1200 '
1210      PRINT MID$(T.MSG$,1,8);
1220      IF (L.PRINT) THEN LPRINT MID$(T.MSG$,1,8);
1230 '
1240      FOR I = 1 TO 11
1250        IF(NOT WANT(I)) GOTO 1300
1260 '
1270         PRINT USING "#####.#"; ARR.DAT(I);
1280         IF (L.PRINT) THEN LPRINT USING "#####.#"; ARR.DAT(I);
1290 '
1300     NEXT I
1310      PRINT " "
1320      IF (L.PRINT) THEN LPRINT " "
1330 '
1340 RETURN
1350 '.....................................................
1360 '      subroutine
1370 '                options setup
1380 '
1390     CLS
```

```
1400     PRINT " There are "; N.REC; "  records
1410     PRINT
1420     PRINT " from "; F.MSG$
1430     PRINT " to   "; MSG$
1440 '
1450     F.DATE$ = MID$(F.MSG$,15,10)
1460     E.DATE$ = MID$(MSG$,15,10)
1470     IF (NOT FOUND) THEN P.DATE$ = "01-01-1900"
1480 '
1490     LOCATE 6,2
1500     INPUT "Enter date to list (mm-dd-yyyy) : ", L.DATE$
1510     IF ((L.DATE$>=F.DATE$)AND(L.DATE$<=E.DATE$)) GOTO 1560
1520       ERR.MSG$ = "Date not on file"
1530      GOSUB 2680
1540      GOTO 1490
1550 '
1560     LOCATE 8,2
1570      IN.MSG$ = "Do you want to change the options? "
1580     GOSUB 2160
1590     IF (NOT ANS) GOTO 1830
1600 '
1610      IN.MSG$ = "Do you want to list on the printer? "
1620     GOSUB 2160
1630      L.PRINT = ANS
1640 '
1650       PRINT "Options: Wind speed,rms,max,min"
1660      PRINT "          wind dir, temps (ground level and 10m)"
1670      PRINT "          pressure, precipitation, humidity"
1680      PRINT " numbered 1 to 11"
1690 '
1700      INPUT "Enter number of items to list (<9) : ", L.NUM
1710      IF (L.NUM>9) THEN L.NUM = 8
1720 '
1730     FOR I = 1 TO 11
1740       WANT(I) = FALSE
1750     NEXT I
1760 '
1770     FOR I = 1 TO L.NUM
1780        PRINT "Enter "; I; "th. item number ";
1790       INPUT ": ", J
1800       WANT(J) = TRUE
1810     NEXT I
1820 '
1830     SKIP = TRUE
1835     IF (L.DATE$>P.DATE$) GOTO 1870
1840       CLOSE #1
1850        OPEN IN.FIL$ FOR INPUT AS #1
1855     SKIP = FALSE
1860 '
```

```
1870     P.DATE$ = L.DATE$
1880 '
1890     PRINT
1900     PRINT L.DATE$ : PRINT
1910     IF (NOT L.PRINT) GOTO 1940
1920      LPRINT: LPRINT L.DATE$: LPRINT
1930 '
1940     PRINT "  time   "; : IF (L.PRINT) THEN LPRINT "  time   ";
1950     FOR I = 1 TO 11
1960       IF (NOT WANT(I)) GOTO 1990
1970        PRINT L.TITLE$(I);
1980        IF (L.PRINT) THEN LPRINT L.TITLE$(I);
1990     NEXT I
2000     PRINT " "
2010     IF (NOT L.PRINT) GOTO 2030
2020       LPRINT " "
2030 '
2040     PRINT "         "; : IF (L.PRINT) THEN LPRINT "         ";
2050     FOR I = 1 TO 11
2060       IF (NOT WANT(I)) GOTO 2090
2070        PRINT L.UNIT$(I);
2080        IF (L.PRINT) THEN LPRINT L.UNIT$(I);
2090     NEXT I
2100     PRINT " ":  PRINT
2110     IF (NOT L.PRINT) GOTO 2130
2120       LPRINT " ":  LPRINT " "
2130 '
2140  RETURN
2150 '.........................................................
2160 '     subroutine
2170 '               asks for a yes/no response
2180 '               returns ans (logical)
2190 '
2200     OUT.MSG$ = IN.MSG$ + " [y/n]  "
2210     PRINT OUT.MSG$;
2220     INPUT ": ", CH$
2230 '
2240     ANS = TRUE
2250     IF ((CH$="n") OR (CH$="N")) THEN ANS = FALSE
2260 '
2270  RETURN
2280 '.........................................................
2290 '     subroutine
2300 '               initialise the titles and options
2310 '
2320     L.TITLE$(1) = "  speed"
2330     L.TITLE$(2) = "  rms  "
2340     L.TITLE$(3) = "  max  "
2350     L.TITLE$(4) = "  min  "
```

```
2360    L.TITLE$(5) = "  dirtn"
2370    L.TITLE$(6) = "  T(gr)"
2380    L.TITLE$(7) = "  T(10)"
2390    L.TITLE$(8) = "  Press"
2400    L.TITLE$(9) = "  Rain "
2410    L.TITLE$(10) = "  Humid"
2420    L.TITLE$(11) = "  Sun  "
2430 '
2440    L.UNIT$(1) =  "  km/hr"
2450    L.UNIT$(2) =  "  km/hr"
2460    L.UNIT$(3) =  "  km/hr"
2470    L.UNIT$(4) =  "  km/hr"
2480    L.UNIT$(5) =  "  degs "
2490    L.UNIT$(6) =  "  C    "
2500    L.UNIT$(7) =  "  C    "
2510    L.UNIT$(8) =  "  mbars"
2520    L.UNIT$(9) =  "  mm   "
2530    L.UNIT$(10) =  "  %    "
2540    L.UNIT$(11) =  "  w/sqm"
2550 '
2560    L.PRINT = FALSE
2570    FOR I = 1 TO 11
2580      WANT(I) = FALSE
2590    NEXT I
2600 '
2610     WANT(1) = TRUE
2620     WANT(3) = TRUE
2630     WANT(4) = TRUE
2640     WANT(6) = TRUE
2650 '
2660 RETURN
2670 '..........................................................
2680 '        subroutine
2690 '                      error report
2700 '
2710    LOCATE 25,1
2720     PRINT ERR.MSG$
2730 '
2740      BEEP:  BEEP
2750 '
2760 RETURN
2770 '..........................................................
2780 '        subroutine
2790 '                        extract N.VAL items from the string C.STR$
2800 '
2810    L = LEN(C.STR$)
2820 '
2830    F1 = 0
2840    FOR I = 1 TO N.VAL
```

```
2850       F1 = F1 + 1
2860       IF (F1>L) GOTO 2990
2870        IF (MID$(C.STR$,F1,1) = " ") GOTO 2850
2880 '
2890       F2 = F1
2900       F2 = F2 + 1
2910       IF (F2>L) GOTO 2940
2920        IF (MID$(C.STR$,F2,1) <> " ") GOTO 2900
2930 '
2940 '
2950       F3 = F2 - F1
2960        TMP = VAL(MID$(C.STR$,F1,F3))
2970       GOTO 3010
2980 '
2990       TMP = -1
3000 '
3010        INDX = N.START + (I-1)
3020        ARR.DAT(INDX) = TMP
3030 '
3040    NEXT I
3050 '
3060  RETURN
```

WIND SPEED SENSOR

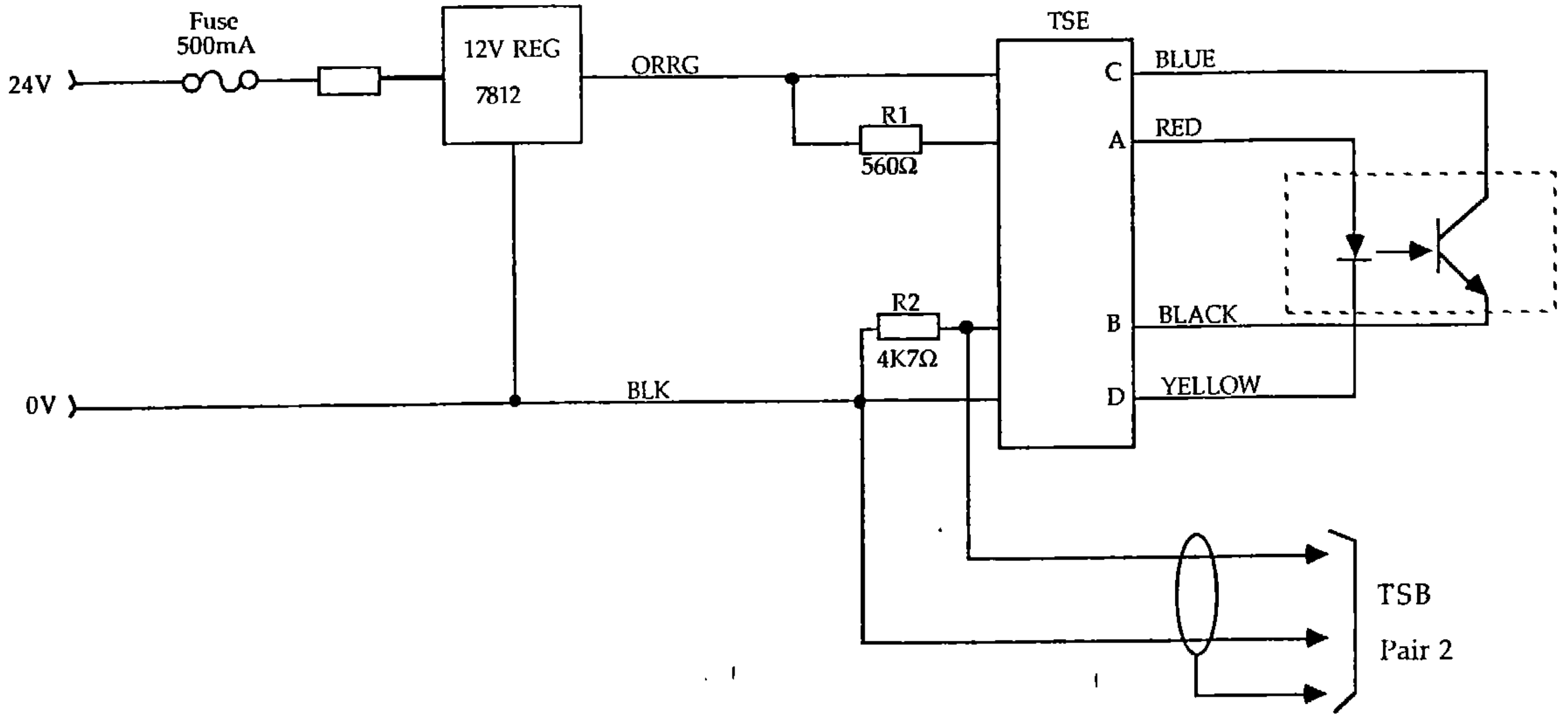


Fig. 1

Culgoora Weather Station

Top Junction Box Wiring (Revised)

(as Per Wind Direct Change 6/12/85)

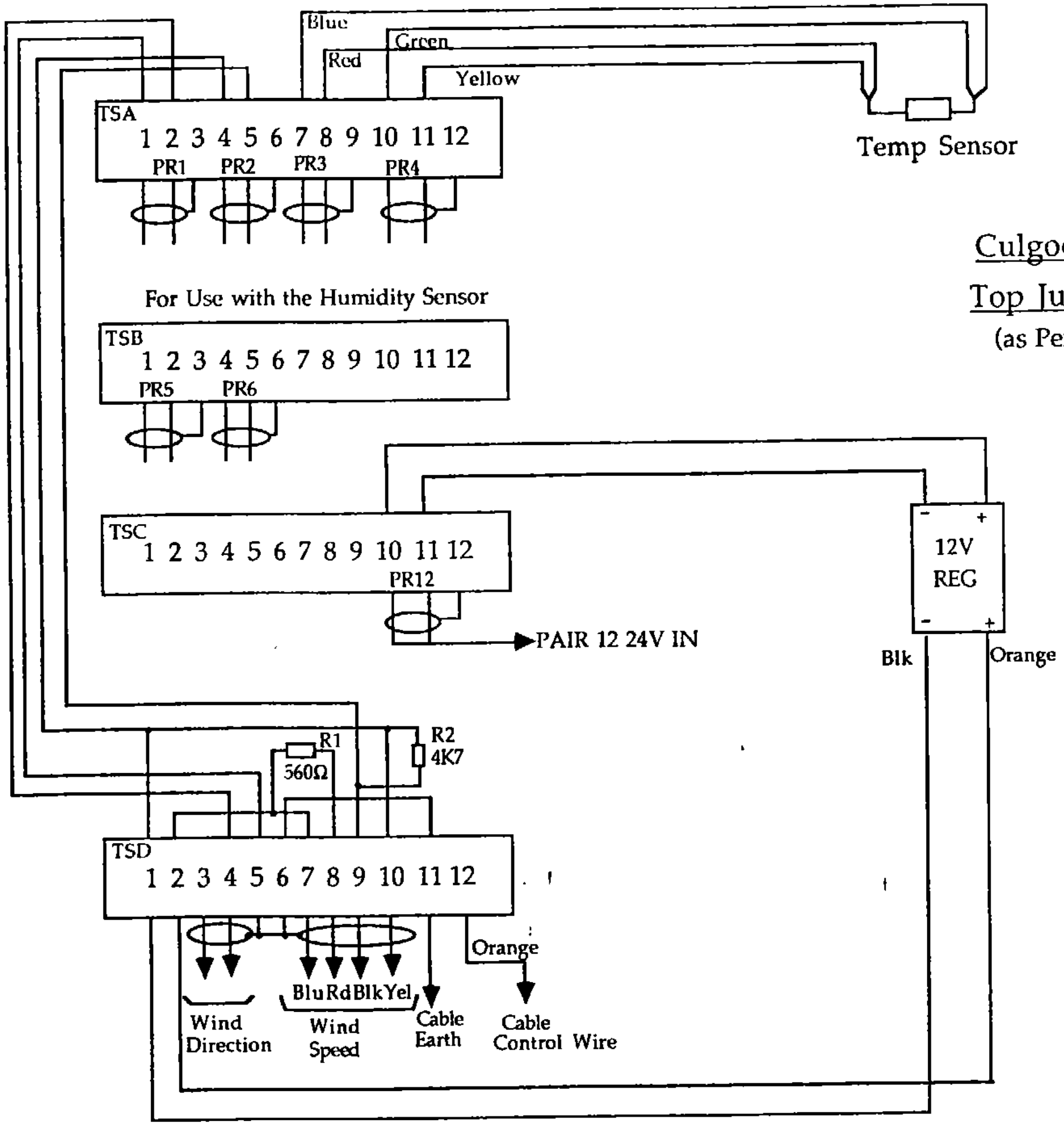
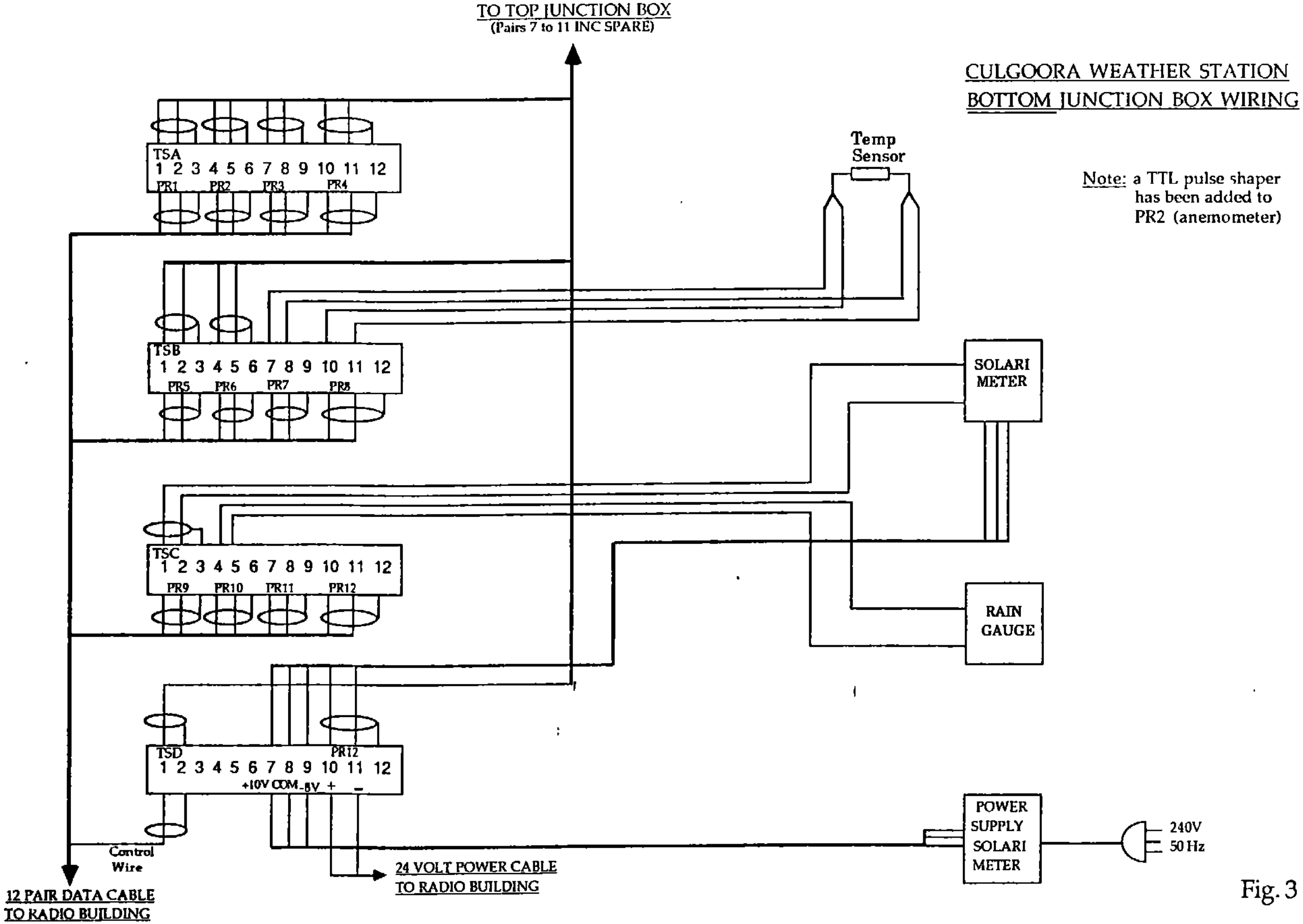


Fig. 2


CULGOORA WEATHER STATION
BOTTOM JUNCTION BOX WIRING
Note: a TTL pulse shaper has been added to PR2 (anemometer)
TO TOP JUNCTION BOX
(Pairs 7 to 11 INC SPARE)
Temp Sensor
TSA
1 2 3 4 5 6 7 8 9 10 11 12
PR1 PR2 PR3 PR4
TSB
1 2 3 4 5 6 7 8 9 10 11 12
PR5 PR6 PR7 PR8
TSC
1 2 3 4 5 6 7 8 9 10 11 12
PR9 PR10 PR11 PR12
TSD
1 2 3 4 5 6 7 8 9 10 11 12
PR12
+10V COM -8V + −
SOLARI METER
RAIN GAUGE
POWER SUPPLY SOLARI METER
240V
50 Hz
Control Wire
12 PAIR DATA CABLE TO RADIO BUILDING
24 VOLT POWER CABLE TO RADIO BUILDING
Fig. 3

CULGOORA WEATHER STATION

CABLE TERMINATION LOGGER RACK

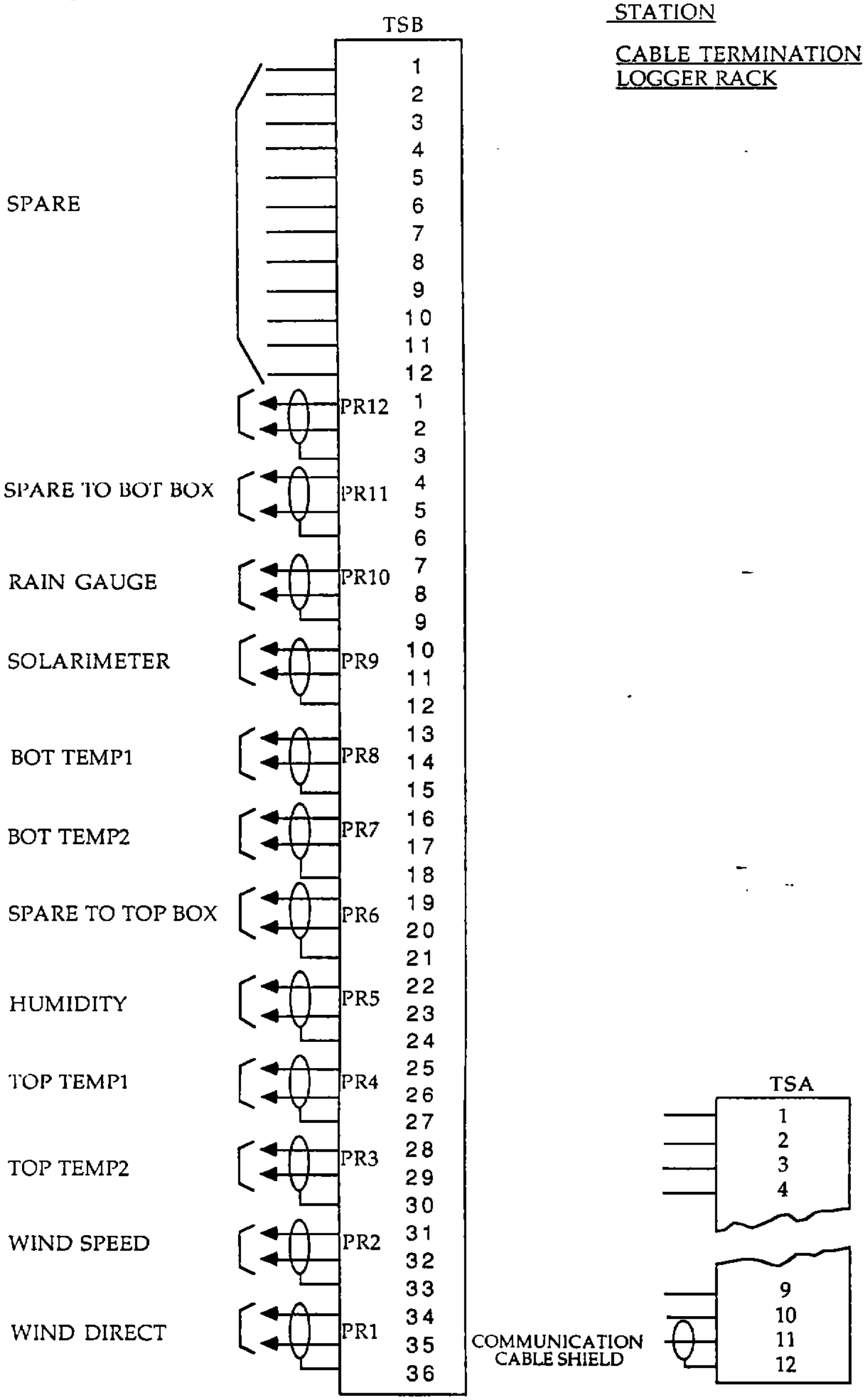


Fig. 4

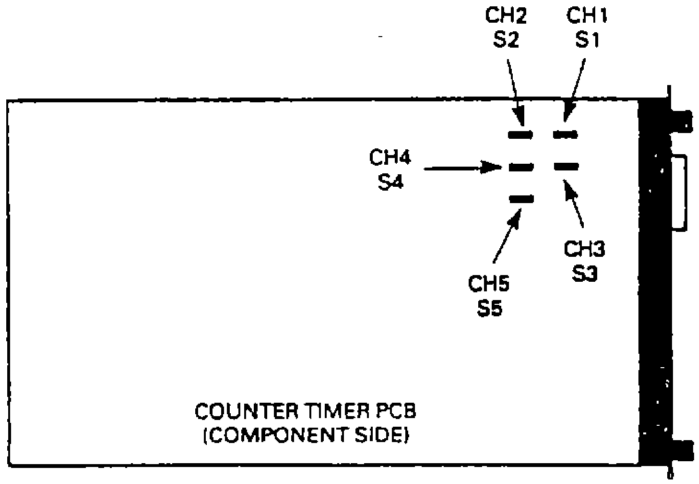
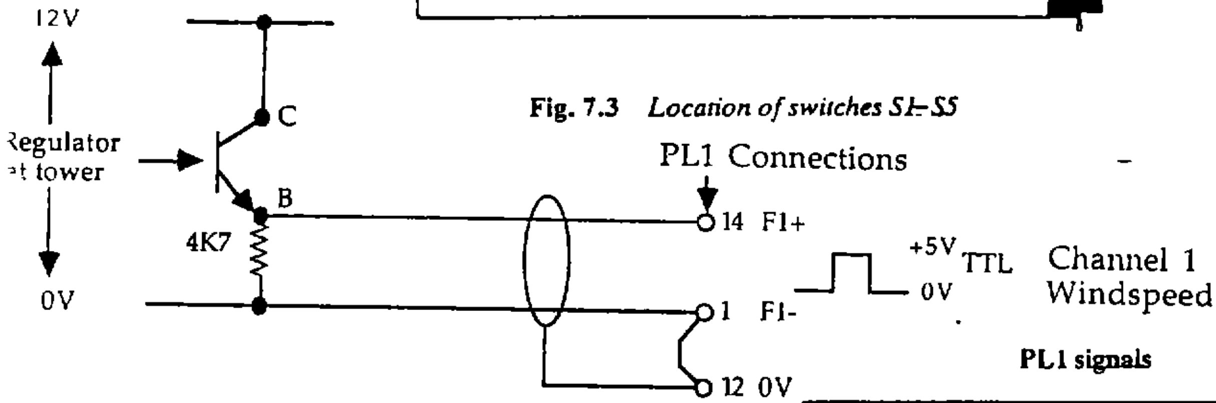
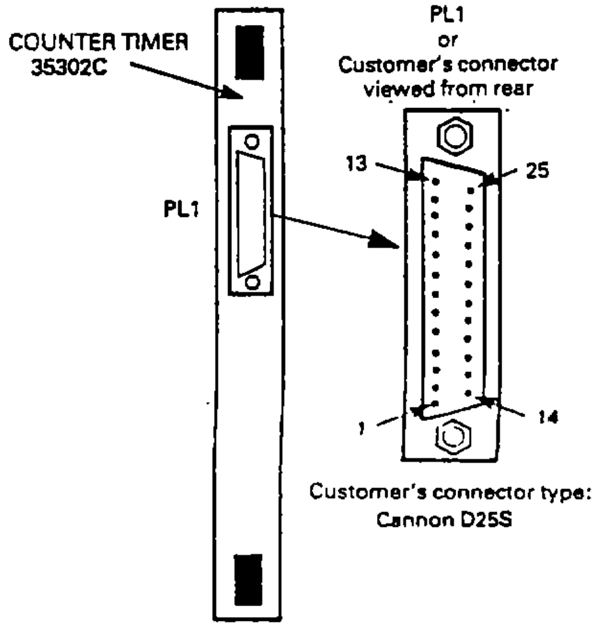


Fig. 7.3 *Location of switches S1–S5*





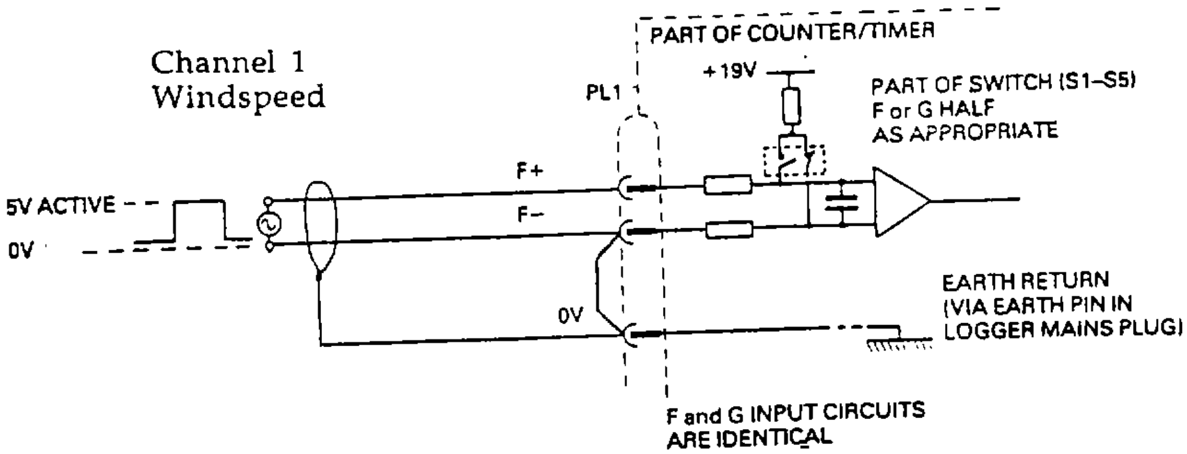
PL1 signals

PIN NO.	SIGNAL
1	F1−
2	G1−
3	F2−
4	G2−
5	F3−
6	G3−
7	F4−
8	G4−
9	F5−
10	G5−
11	+19V
12	0V
13	0V19
14	F1+
15	G1+
16	F2+
17	G2+
18	F3+
19	G3+
20	F4+
21	G4+
22	F5+
23	G5+
24	−30V } FOR TEST PURPOSES ONLY
25	+12V } FOR TEST PURPOSES ONLY

Fig. 5 PL1 Pin Numbering

TTL INPUTS: Active Hi

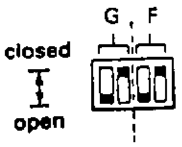
Input connections for a floating signal source



S1 to S5 Settings (for floating or earthed signal source)

Required Switch Settings
S1

S1 to S5 RELATE TO CHANNELS 1 to 5 RESPECTIVELY



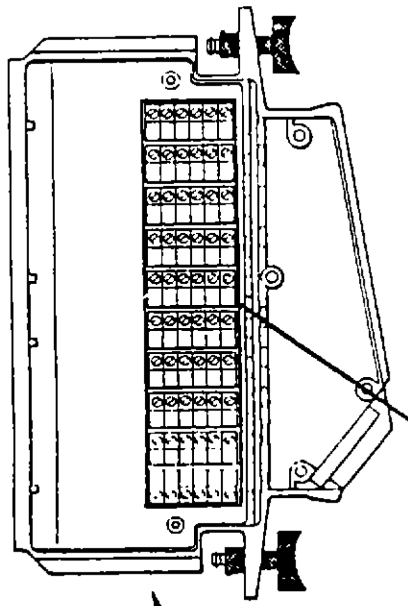
BOTH F and G SWITCHES ARE SHOWN SET FOR TTL ACTIVE HI INPUT.
F and G SWITCHES MAY BE SET INDEPENDENTLY FOR DIFFERENTIAL OR TTL AS REQUIRED.
(DIFFERENTIAL SETTINGS ARE SHOWN IN FIG. 7.1)

Fig. 6 TTL Inputs: Active Hi

Event/Status Module Slot 5 35302A

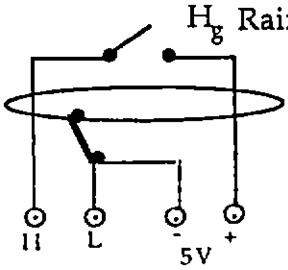
Rain Gauge Connections

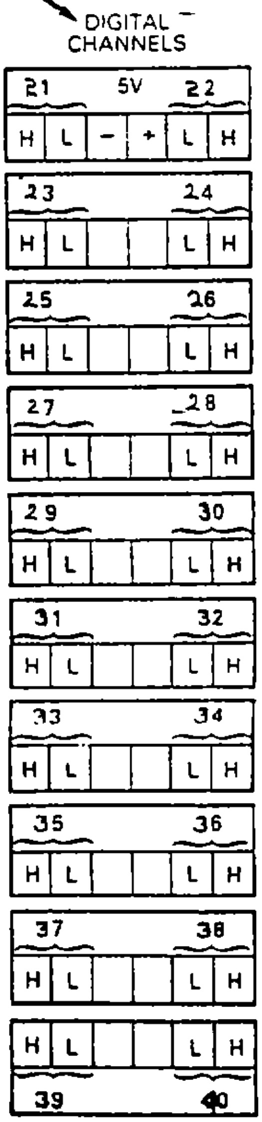
Event Recording



INPUT CONNECTOR ASSEMBLY
(COVER PLATE REMOVED)

H	Hi	see H & L TERMINALS below
L	Lo	
+ −	5V floating supply	





NUMBERING

Numbering for channels 1-20 only is shown; subsequent sets of 20 channels are similarly numbered, ie 21-40, 41 to 60, and so on, up to 181-200.
Channel numbering on any input connector is dependent on which slot it occupies in the logger mainframe.

H & L TERMINALS

For each input channel circuit to conduct, the H terminal must have a +ve potential with respect to the L terminal. When the input circuit conducts the logger gives a measurement of Logic 0. When the input circuit does not conduct the logger gives a measurement of Logic 1.

Fig 7. Input connector: Channel Numbering and Pin Assignment.

Reed Relay Unit. Slot 7. 35301A

PRT. Connections.	PRT. REcording
Wind Direction Connections	Resistance Recording
Solar Radiation Connections	D.C. Volts Recording
Barometric Pressure Connection	D.C. Volts Recording
Relative Humidity Connection	D C. Volts Recording

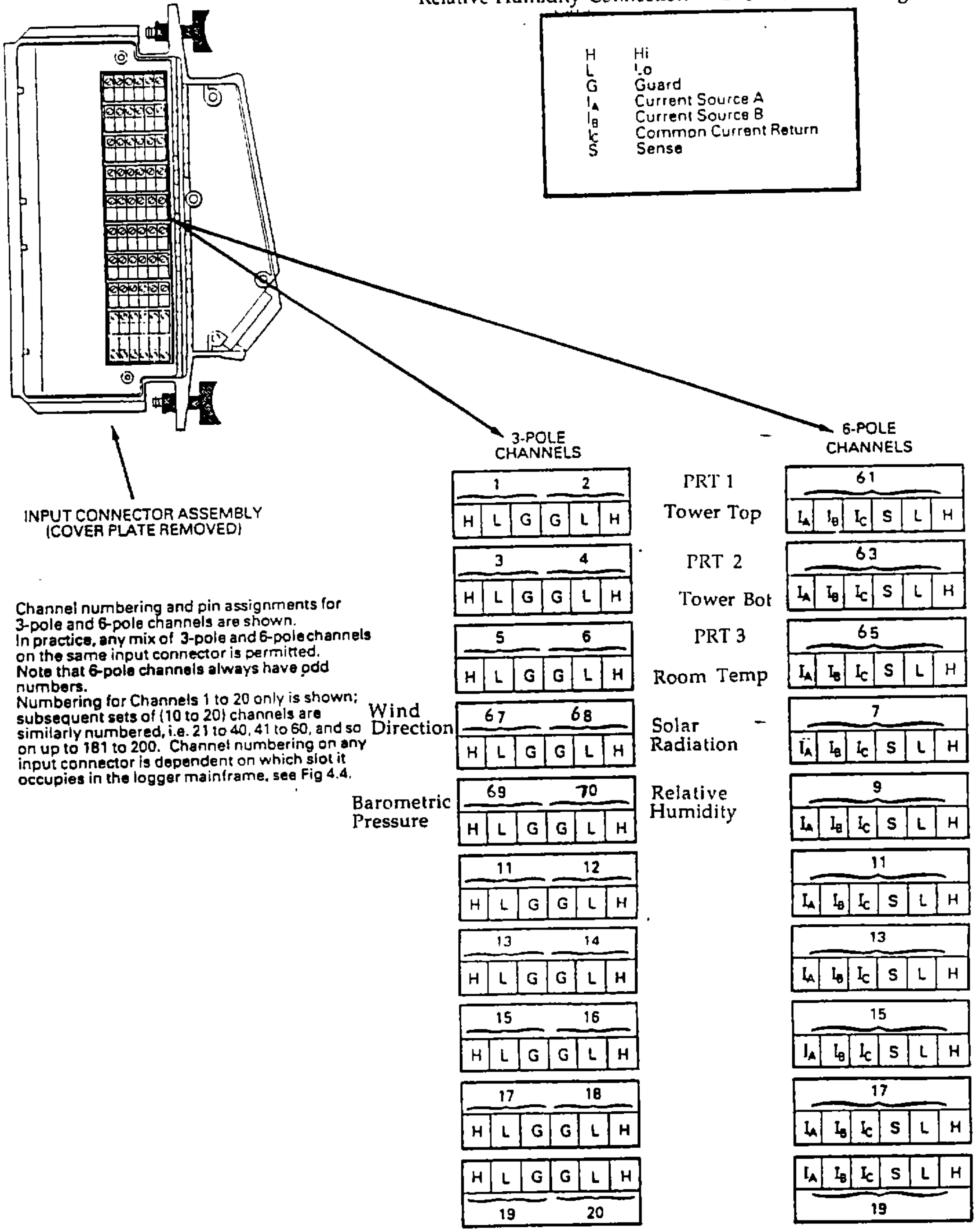
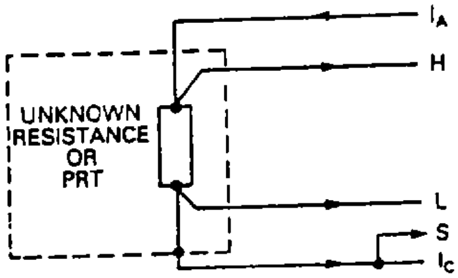


Fig. 8 Channel Numbering and Assignment of Terminals on Input Connectors

2.5 4-TERMINAL RESISTANCE OR PRT MEASUREMENT

2.5.1 **Theoretical Diagram**

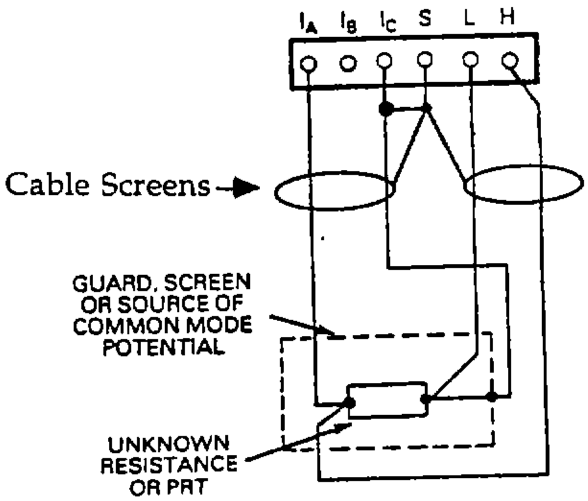


The energising current is generated and switched automatically, no additional linking or connections are required.

Slot 7

2.5.2 **Practical Connection Diagram**

Connection Diagram for PRTs

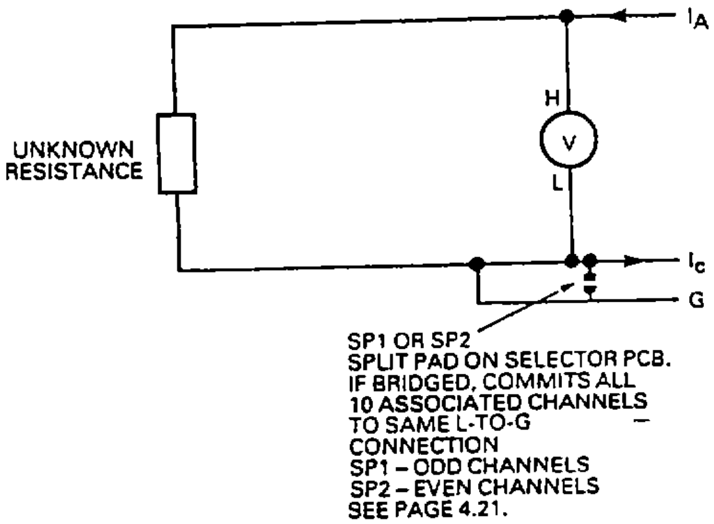


PRT 1. Tower top ch. 61
PRT 2. Tower bot ch. 63
PRT 3. Room Temp ch. 65

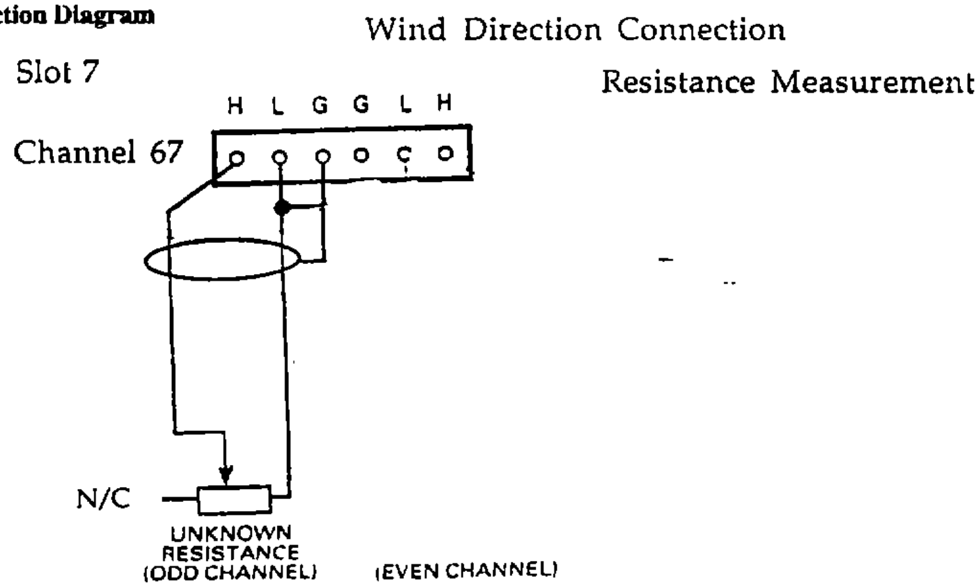
Fig. 9

2.4 2-TERMINAL RESISTANCE MEASUREMENT

2.4.1 Theoretical Diagram



2.4.2 Practical Connection Diagram

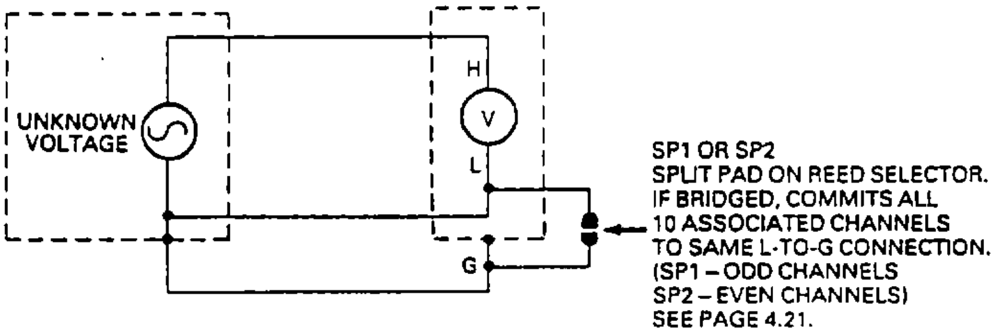


N.B. Use Reed Relay Selector 35301A only. 2-terminal resistance measurement is suitable for applications where less accuracy is required (e.g. measuring 10k to 1% accuracy). For greater accuracy, use 4-terminal resistance measurement, see page 4.24.

Fig. 10

2.2 VOLTAGE MEASUREMENT

2.2.1 Theoretical Diagram



2.2.2 Practical Connection Diagram

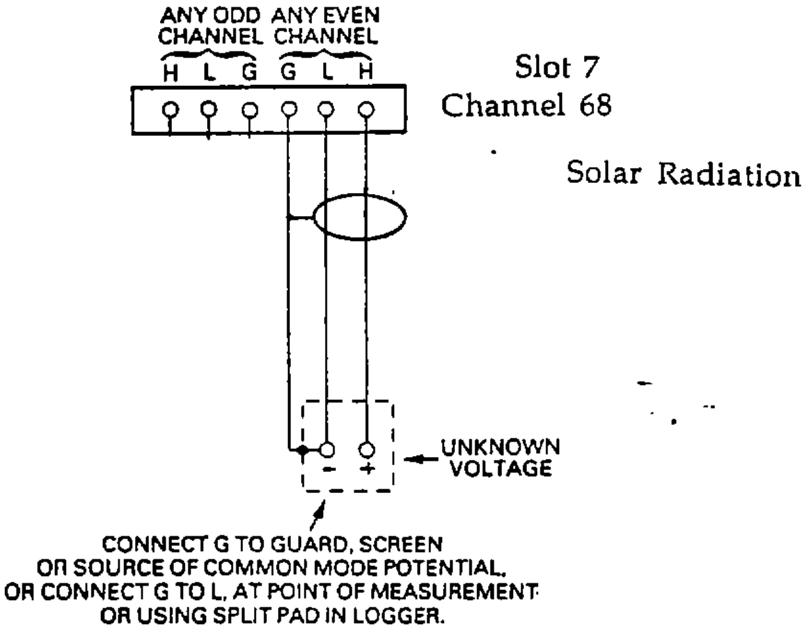
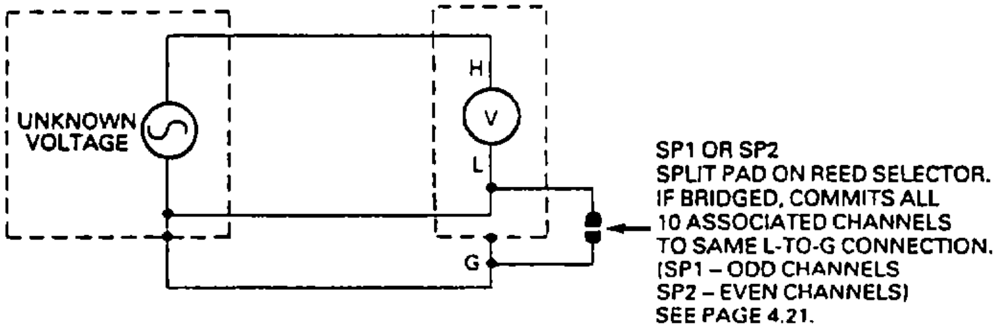


Fig. 11

2.2 VOLTAGE MEASUREMENT

2.2.1 Theoretical Diagram



2.2.2 Practical Connection Diagram

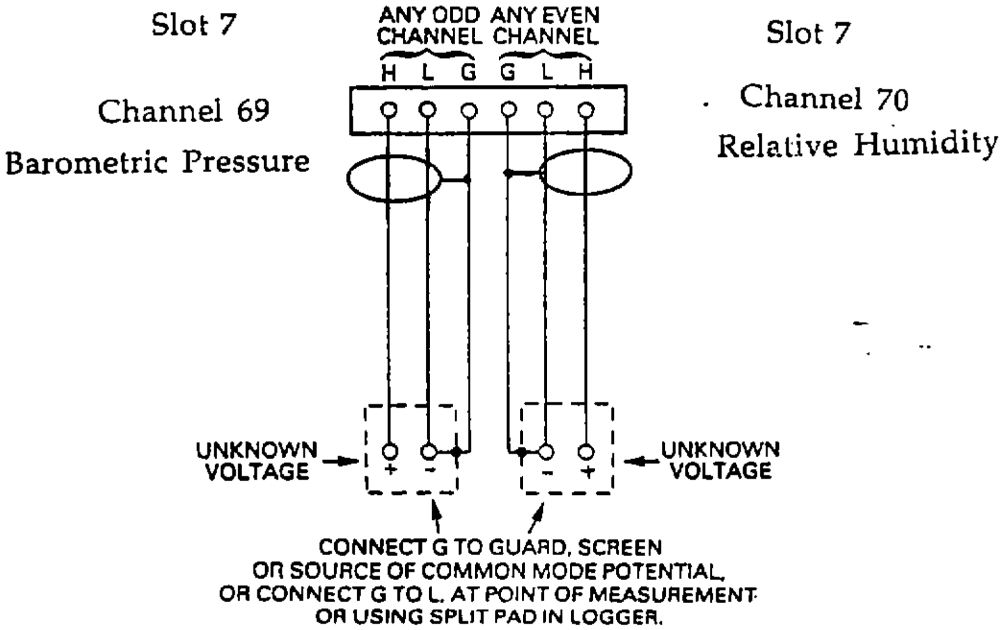


Fig. 12

INTERRUPTER ASSEMBLIES

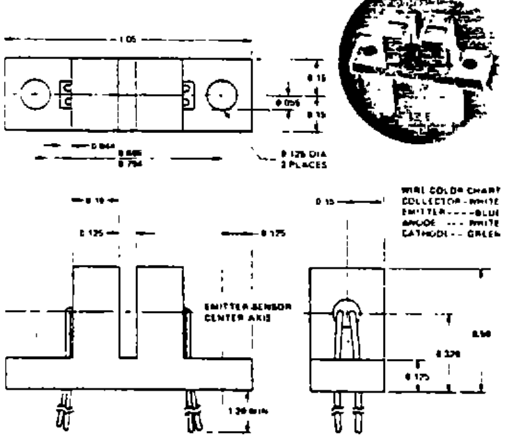
Non-contact switching is provided by OPTRON's optoelectronic interrupter assemblies which consist of a high efficiency gallium arsenide LED and silicon N-P-N phototransistor or photodarlington in a molded plastic housing. A gap in the housing provides a means of sensing motion of opaque objects. Design flexibility is provided by a variety of gap widths and mounting configurations. In addition, units are available with infrared filters to eliminate background illumination and with apertures for high resolution position sensing.

optoelectronic characteristics (25°C)

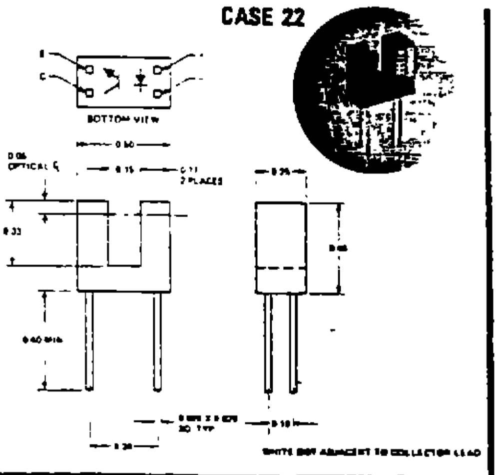
PARAMETER	ON-STATE COLLECTOR CURRENT		OFF-STATE COLLECTOR CURRENT	SATURATION VOLTAGE			LED FORWARD VOLTAGE		RESPONSE TIME	SLOT WIDTH	OUTPUT CONFIGURATION	CASE NUMBER
TEST CONDITION	V_{CE}=10 V, I_F as shown		V_{CE}=10 V, I_F=0	I_C and I_F as shown			I_F as shown		R_L=100 Ω, V_{CC}=5 V, I_F=40 mA			
SYMBOL	I_L		I_D	$V_{CE(SAT)}$			V_F		t_r, t_f	d		
UNIT	mA		nA	VOLTS			VOLTS		μsec	INCHES		
	MIN	I_F mA	MAX	MAX	I_C mA	I_F mA	MAX	I_F mA	TYP	NOM		
OPB 120	0.8	40 (1)	100	0.4	0.4	40	1.5	50	2.5 (3)	0.125	Transistor	21
OPB 242	1.6	40 (1)	100	0.4	0.4	40	1.5	50	2.5 (3)	0.125	Transistor	21
OPB 243	3.2	40 (1)	300	1.1	0.4	50	1.6	50		0.125	Darlington	21
OPB 804	0.5	20	100	0.4	0.25	20	1.7	20	5.0	0.150	Transistor	22
OPB 806	1.6	35 (4)	25 (5)	0.5	0.4	15	1.5	15	2.5	0.125	Transistor	23
OPB 810	2.0	30 (1)	100	0.6	1.8	40	1.8	40	5.0	0.200	Transistor	24
OPB 813	0.5	20	100	0.4	0.25	20	1.7	20	5.0	0.125	Transistor	25
OPB 813S (6)	0.5	20	100	0.4	0.25	20	1.7	20	5.0	0.125	Transistor	25
OPB 814	1.0	10 (1)	100	0.4	0.25	10	1.7	20	5.0	0.125	Transistor	25
OPB 815	1.8	20 (2)	100	0.6	1.8	20	1.7	20	5.0	0.125	Transistor	25

Notes
(1) V_{CE}=5 V
(2) V_{CE}=0.6 V
(3) Response time measured with I_C=0.8 mA
(4) V_{CE}=0.5 V
(5) V_{CE}=30 V
(6) Sensor apertured internally to 0.010 × 0.040 inches

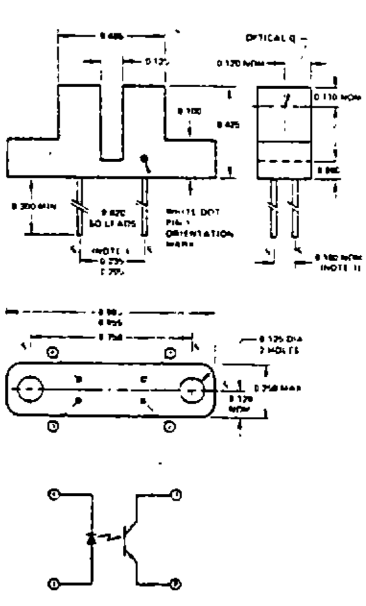
CASE 21



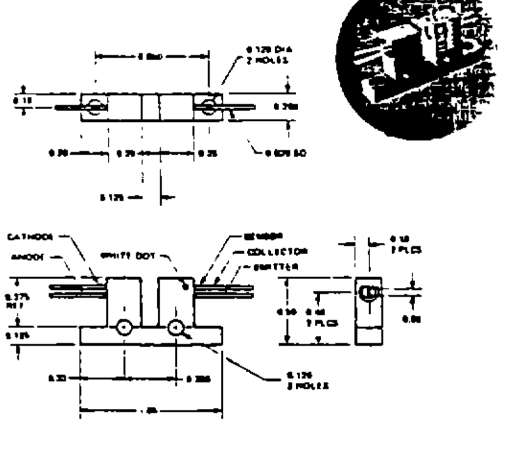
CASE 22



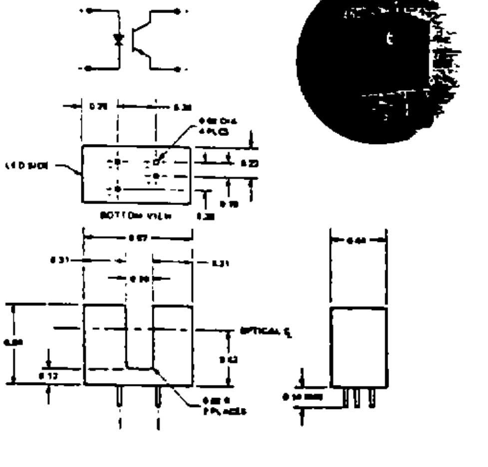
CASE 25



CASE 23



CASE 24



ALL DIMENSIONS ARE IN INCHES