Multi-channel signals with programmable inter-channel phase

This solution requires a C-size VXI cage, a control card, an E1420A counter and, if more than two E1440As are used, an E1366A switching unit.

Features and benefits

- Slave E1440A frequencies can be a multiple of master frequency.
- Channels can be independently modulated (AM or PM).
- 11-digit frequency resolution.
- Phase accuracy /-2 degree at 20 MHz (even better at lower frequencies due to counter's 200 ps time-interval resolution).
- Output waveforms need not be the same, and are selectable from:
  sine (up to 21 MHz), square or triangle (up to 11 MHz), and ramp (up to 11 kHz). Minimum frequency is 1 uHz.
- 5 ppm stability, if E1440As are daisy-chained. Can be improved by using the counter reference.
- Independent amplitude 1 mVpp to 10 Vpp into 50 ohm. Option 001 provides 4 mVpp to 40 Vpp into 500 ohm.

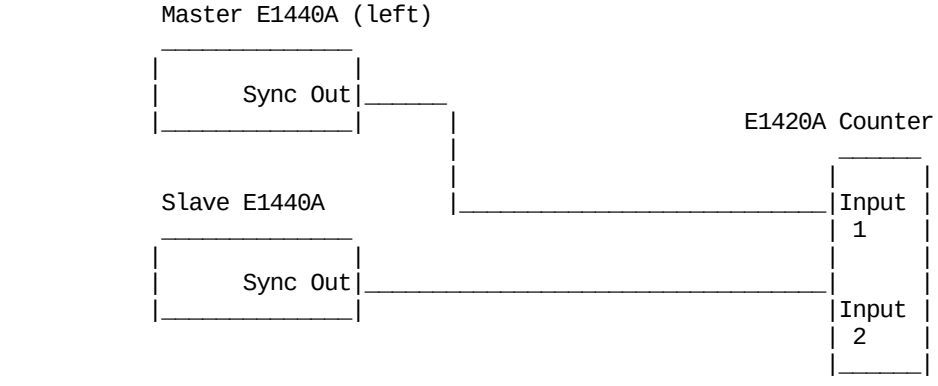
Application example: Navigation Systems

Test of vehicle receivers used in Decca-type location systems requires a number of synthesizers to simulate the signals from a number of fixed transmitters. The frequencies, which are multiples of the master transmitter frequency, must not only be phase-locked, but also of programmable phase so that receiver performance can be verified for any vehicle position.

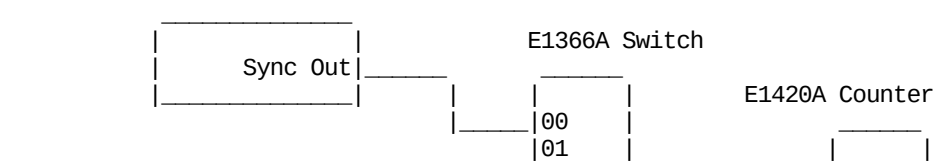
Setting up for the first time

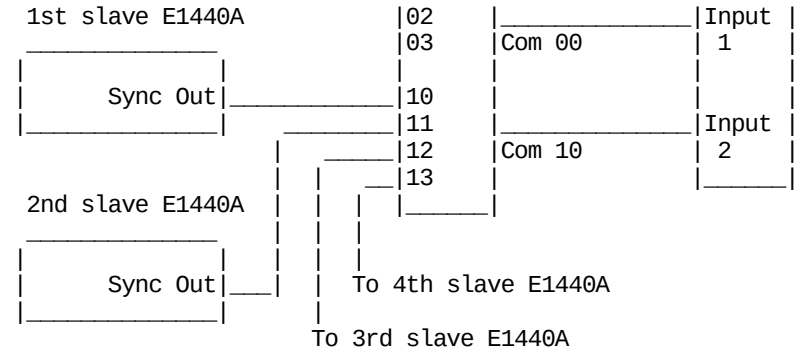
After inserting the modules in the VXI-cage, the front panel connections shown in the following diagram must be made. Additionally, the counter's 10 MHz reference can be "daisy-chained" to all E1440As; this however is not essential, the VXI-internal CLK10 line can be used instead.

a) Two channels



b) Three to five channels





Try to keep the Sync Out path lengths as equal as possible. The same applies to the main outputs. Perform a phase measurement at the device and note any offset needed. With propagation times equalized, the phase accuracy lies in the range /-1 to /- 7 degree depending on waveform and amplitude mix, and frequency.

From here on, correct phase relationship is simply a matter of setting each slave to the required phase. Subsequently, whenever frequency, waveform or amplitude are changed, a phase-cal should be performed. If more than one counter is detected, the one with the lowest slot number will be taken.

Also, the current settings of the counter and switch are stored. This is to allow the counter to be used for other measurements. After the phase calibration, the previous counter settings are automatically recalled.

The calibration is done by the subprogram Sc_calibrate, which first measures the time delay between the positive slopes of the sync signal from the master device (which is used as a reference) and the sync signals from the slaves. It then calculates the phase error between master and slave and adjusts each slave so that all channels are in phase with the master.

After calibration, the program sets the different E1440A's to the desired phase values. Then, as mentioned above, counter and switch are returned to their previous settings.

Automation

UNSUPPORTED programs are attached:

1. PHASECAL: provides a user interface and phase-calibrates automatically.
2. ERRORHANDLING: a utility for use in any application where self-test is to be implemented.

The programs are in HP Basic and stored as ASCII files (use the HP Basic command GET to load the file).

PHASECAL implements phase calibration for up to five HP E1440As. If a daisy-chained reference signal is used instead of VXI line CLK10, change line 109 from:

```
  109  OUTPUT Hpib E1440(I,2);":SOURCE:ROSC:SOURCE CLK10"
       ! CLK10 is ref
```

to:

```
  109  OUTPUT Hpib E1440(I,2);":SOURCE:ROSC:AUTO ON"
       ! Daisy chain ref
```

Initially, the routine detects the cards in the VXI-cage (SUB Sc_getdev) and assigns the E1440A with the lowest slot number as master, the one with the next higher slot number will be slave 1, and so-on. If more than one E1420A counter is detected, the one with the lowest slot number will be taken.

Also, the current settings of the counter and switch are stored.

This is to allow the counter to be used for other measurements.

The calibration is done by the subprogram Sc_calibrate, which first measures the time delay between the positive slopes of the sync signal from the master device (which is used as reference) and the sync signals from the slaves. It then calculates the phase error between master and slave and adjusts each slave so that all channels are in phase with the master.

After calibration, the program sets the different E1440's to the desired phase values, and returns the counter and switch to their previous settings.

```
1     !  PHASE CALIBRATION UTILITY (IN RMB) FOR A MASTER E1440A
2     !  WITH UP TO FOUR SLAVE E1440As.
3     !
4     !  E1420A COUNTER MUST BE INSTALLED IN THE SAME MAINFRAME.
5     !
6     !  IF THERE IS MORE THAN ONE SLAVE, A PROGRAMMABLE CO-AX
7     !  SWITCH WILL ALSO BE NEEDED. THIS PROGRAM USES THE
8     !  E1366A SWITCH
9     !
10    !
11    !
12  REAL Counter      ! HPIB-address of E1420 e.g. 71605
13  REAL Switch       ! HPIB-address of Switchbox
14  REAL Master       ! HPIB-address of master E1440
15  REAL Frequency    ! Frequency of master E1440
16  REAL Master_phase ! Phase of master E1440 after calibration
17  REAL Slave(1:4)   ! HPIB-addresses of slave E1440's, 0 for
                        unused slaves
18  INTEGER Slave_multi(1:4)
19  REAL Slave_phase(1:4)  ! Slave phase after cal
20  REAL Error        ! Return value: 0=cal ok, 1=timeout
21  INTEGER Pm_cal    ! 0: CALIBRATION WILL BE DONE FOR PM BY
22                    !    SWITCHING OFF/ON PM INPUTS
23              !    THIS MAY CAUSE FOR SQUAREWAVES A PHASE
24              !    SHIFT OF 180 DEGREES
25              !<>0: FOR SQUAREWAVES WITH PM, THE PROGRAM WILL
26              !     CALIBRATE WITH PM SWITCHED ON AND ASK
27              !     TO DISCONNECT/CONNECT THE CABLES TO THE
28              !     PM INPUTS. THIS PROTECTS FROM THE PHASE
29              !     SHIFT, BUT NEEDS MANUAL MODIFICATIONS OF
30              !     THE TEST CABLE CONNECTIONS
31    !
32  Frequency=1500
33  Slave_multi(1)=1
34    !
35  REAL Hpib ! incl sec addr. Used only for Sc_getdev subpgm
36  Hpib=71600  !     HAS TO BE CHANGED TO VALID ADDRESS
37  CALL Sc_getdev(Hpib,Counter,Switch,Master,Slave(*))
                !  assign dev addresses
38  !CALL
39 !!c_calibrate(Counter,Switch,Master,Frequency,
                  Master_phase,Slave(*),
40 !!Slave_multi(*),Slave_phase(*),Error,Pm_cal)
41  END
42    !
43    !=======================================================
44  SUB Sc_getdev(Hpib,Counter,Switch,Master,Slave(*))
       ! CHECK DEV ADDRESSES
45     !  this subprogram is only used to show how the
          configuration of the VXI
46     !  -mainframe can be detected automatically. It is
           not needed for calibration
47   INTEGER I,Devnumber,Count,E1440a,E1420a,Sbox
48 !!REAL Mfg,Dummy,La(1:13),E1440(1:5,1:2),Slot,Dev,
49 !!Switchbox(1:3,1:2),E1420(1:3,1:2)
50   DIM Dev$[20],Dummy$[200],Dlad$[50]
```

```
51   Counter=0
52   IF NOT Hpib THEN SUBEXIT
53   ON TIMEOUT INT(Hpib/10000),5 GOTO Timeout_error
54   Dev=Hpib
55   OUTPUT Hpib;":VXI:CONF:DNUM?"
56   ENTER Hpib;Devnumber
57   OUTPUT Hpib;":VXI:CONF:DLAD?"
58   ENTER Hpib;Dlad$
59   FOR I=1 TO Devnumber
60    La(I)=VAL(Dlad$)
61    Dlad$=Dlad$[POS(Dlad$,",") 1,LEN(Dlad$)]
62    OUTPUT Hpib;":VXI:CONF:DLIS? "&VAL$(La(I))
63    ENTER Hpib;Dummy,Dummy,Mfg,Dummy,Slot,Dummy$
64    IF Mfg=4095 THEN       !  manufacturer hp ?
65     La(I)=INT(La(I)/8)
66     Dev=Hpib La(I)
67     OUTPUT Dev;"*IDN?"
68     ENTER Dev;Dummy$
69     Dummy$=Dummy$[POS(Dummy$,",") 1,LEN(Dummy$)]
70     Dummy$=Dummy$[1,POS(Dummy$,",")-1]
71     SELECT Dummy$
72     CASE "E1440A"
73      E1440a=E1440a 1
74      Count=Count 1
75      IF Count<6 THEN
76       E1440(Count,1)=Slot
77       E1440(Count,2)=La(I)
78      END IF
79     CASE "E1420A"
80      E1420a=E1420a 1
81      E1420(E1420a,1)=Slot
82      E1420(E1420a,2)=La(I)
83     CASE "SWITCHBOX","E1472A"
84      Sbox=Sbox 1
85      Switchbox(Sbox,1)=Slot
86      Switchbox(Sbox,2)=La(I)
87     CASE ELSE
88     END SELECT
89    END IF
90   NEXT I
91   IF Count THEN
92    !!MAT SORT E1420(*,1)
93    !!MAT SORT Switchbox(*,1)
94    FOR I=3 TO 1 STEP -1
95     IF E1420(I,2) THEN
96      Counter=Hpib E1420(I,2)
97     END IF
98     IF Switchbox(I,2) THEN
99      Switch=Hpib Switchbox(I,2)
100    END IF
101   NEXT I
102   !!MAT SORT E1440(*,1)
103   Count=0
104   FOR I=1 TO 5
105    IF E1440(I,2) THEN
106     Count=Count 1
107     IF Count=1 THEN Master=Hpib E1440(I,2)
108     IF Count>1 THEN Slave(Count-1)=Hpib E1440(I,2)
109     OUTPUT Hpib E1440(I,2);":SOURCE:ROSC:SOURCE CLK10"
                  ! CLK10 is ref
110    END IF
111   NEXT I
112  END IF
113  IF E1420a=0 THEN PRINT "CONFIG ERROR: No E1420 cntr"
114  IF Sbox=0 AND E1440a>1 THEN
115   PRINT "CONFIG MESSAGE: No Switchbox"
116  END IF
117  IF E1440a<2 THEN PRINT "CONFIG ERROR: No E1440"
118  OFF TIMEOUT
119  SUBEXIT
```

```
120 Timeout_error:!
121  OFF TIMEOUT
122  !!PRINT "ERROR: No device "&VAL$(Dev)&
123 SUBEND !!=====================================
124 !!SUB Sc_calibrate(Counter,Switch,Master,Frequency,
                Master_phase,Slave(*),
125 !!INTEGER Slave_multi(*),REAL Slave_phase(*),Error,
                  INTEGER Pm_cal)
126  REAL Master_pm,Slave_pm,Test1,Test2,J,Slave_frequency,Fault
127  DIM Master_wave$[15],Slave_wave$[15]
128  INTEGER I
129  Error=1
130  IF NOT Master OR NOT Counter THEN SUBEXIT ! no master or
                                                 counter addr
131  IF Master<10000 OR Counter<10000 THEN SUBEXIT
                                       ! incomplete addr
132  IF Frequency>2.1E 7 THEN SUBEXIT    ! Frequency limit
133  ON TIMEOUT INT(Master/10000),7 GOTO No_count
                                       !timeout? ->abort
134  OUTPUT Master;":FREQ:FIX "&VAL$(Frequency)&"HZ;
                                           :SOURCE:PM:STATE?"
135     !  check master pm
136  ENTER Master;Master_pm
137  OUTPUT Master;":SOURCE:FUNCTION:SHAPE?"
                                     ! check master waveform
138  ENTER Master;Master_wave$
139  IF Master_wave$="DC" THEN SUBEXIT !  no calibration for dc
140  OUTPUT Counter;"*SAV 9"           !  save counter setup
141  !!OUTPUT Counter;":INP1:ROUTE SEP;:SENSE1:EVENT:SLOPE
142  !!POS;:SENSE2:EVENT:SLOPE POS;:INPUT1:COUP DC;:INPUT2
143  !!:COUP DC;:INP1:IMP 50;:INP2:IMP 50"
144  !!OUTPUT Counter;":SENSE1:EVENT:LEVEL:AUTO OFF;
           :SENSE1:EVENT:LEVEL
145  !!:ABS 1.5V;:SENSE2:EVENT:LEVEL:AUTO OFF;
           :SENSE2:EVENT:LEVEL:ABS 1.5V"
146  IF (Switch MOD 10000) THEN OUTPUT Switch;"*SAV 9;
           :ROUT:CLOS (@100)"
147  !!save mux settings, switch mux to connect master to
                                               counter input 1
148  ON TIMEOUT INT(Counter/10000),7 GOTO No_cal
149  IF (Master_wave$<>"SQUARE") OR Master_pm<>1 OR
                                                 Pm_cal=0 THEN
150   OUTPUT Master;":SOUR:PM:STAT OFF"
151   Test1=0 !  output has normal phase during calibration
152  ELSE !  select desired and necessary calibration
153   Test1=180!  output has inverted phase during calibration
154   PRINT "CALIBRATION MESSAGE: Please disconnect cable from
                                                      PM input"
155  !! "at master device ("&VAL$(Master)&") then press any key"
156   !!!ON KBD ALL GOTO !152
157   GOTO 151                          !  wait for user action
158   !!OFF KBD
159  END IF
160  IF Master_phase<>Test1 THEN
161   FOR J=Master_phase TO Test1 STEP (Test1-Master_phase)/15
162   !!    ! omit phase shift below 25 kHz
163    OUTPUT Master;":SOUR:PHAS "&VAL$(PROUND(J,-1))&" DEG"
164   NEXT J
165  END IF
166  FOR I=1 TO 4
167   IF Slave(I) AND ((I=1) OR (Switch MOD 100)) THEN
                                     !   slave on setup ?
168    !!IF (Switch MOD 10000) THEN OUTPUT Switch;":rout
                                     :clos (@11"&VAL$(I-1)&")
169   !! "Slave_frequency=Frequency*Slave_multi(I)
              !  calculate new slave frequency
170  OUTPUT Slave(I);":FREQ:FIX "&VAL$(Slave_frequency)&"HZ;
                                           :SOURCE:PM:STATE?"
171    ENTER Slave(I);Slave_pm        !  check slave for pm
172    OUTPUT Slave(I);":SOURCE:FUNCTION:SHAPE?"
```

```
173    ENTER Slave(I);Slave_wave$      !  check slave waveform
174    IF Slave_wave$<>"DC" AND Slave_frequency<2.1E 7 THEN
175    !! !  no calibration for dc or frequencies above 21MHz
176     IF (Slave_wave$<>"SQUARE") OR Slave_pm<>1 OR Pm_cal=0
                  THEN
177      OUTPUT Slave(I);":SOUR:PM:STAT OFF"
178      Test2=180
179     ELSE
180      Test2=0
181      PRINT "CALIBRATION MESSAGE: Please disconnect
                                       cable from PM input"
182     !! "at slave"&VAL$(I)&" device ("&VAL$(Slave(I))&")
                                          then press any key"
183      !!ON KBD ALL GOTO 176
184     !! GOTO 175             !  wait for user action
185      !!OFF KBD
186     END IF
187     IF Slave_phase(I)<>(180-Test2) THEN
188      FOR J=Slave_phase(I) TO 180-Test2 STEP
                                       (180-Slave_phase(I))/15
189    !!! omit phase shift below 25 kHz for phase changes
                                       > 25 deg (squarewave)
190       OUTPUT Slave(I);":SOUR:PHAS"&VAL$(PROUND(J,-1))&" DEG"
191      NEXT J
192     END IF
193       !
194     CALL Sc_measure(Counter,Master,Slave(I),Test2,
                                     Slave_frequency,Fault)
195       !
196     !!IF ABS(Fault-180)<179.5 THEN OUTPUT Slave(I);":SOUR:
                                                  PHAS "&VAL$
197     !! (Fault)&" DEG;:SOURCE:PHASE:REFERENCE"
198     IF Slave_phase(I) THEN
199      FOR J=0 TO Slave_phase(I) STEP (Slave_phase(I))/15
200     ! omit phase shift below 25 kHz
201       OUTPUT Slave(I);":SOUR:PHAS "&VAL$(PROUND(J,-1))&
                                                       " DEG"
202      NEXT J
203     END IF
204     OUTPUT Slave(I);":SOUR:PM:STAT "&VAL$(Slave_pm)
205     IF (Slave_wave$="SQUARE") AND Slave_pm=1 AND Pm_cal<>0
                                                        THEN
206     !! PRINT "CALIBRATION MESSAGE: Please connect cable to
                                                    PM input at
207     !!slave"&VAL$(I)&" device ("&VAL$(Slave(I))&") then
                                                  press any key"
208     !! ON KBD ALL GOTO 197
209      GOTO 196          !  wait for user action
210     !! OFF KBD
211     END IF
212    END IF
213   END IF
214  NEXT I
215  IF Test1<>Master_phase THEN
216   FOR J=Test1 TO Master_phase STEP (Master_phase-Test1)/15
217  ! omit phase shift below 25 kHz for changes > 25 deg (only
                                                  for squarewave)
218    OUTPUT Master;":SOUR:PHAS "&VAL$(PROUND(J,-1))&" DEG"
219   NEXT J
220  END IF
221  OUTPUT Master;":SOUR:PM:STAT "&VAL$(Master_pm)
222  IF (Master_wave$="SQUARE") AND Master_pm=1 AND Pm_cal<>0
                                                        THEN
223  !! PRINT "CALIBRATION MESSAGE: Please connect cable to
                                                    PM input at
224  !!master device ("&VAL$(Master)&"), then press any key"
225  !! ON KBD ALL GOTO 212
226   GOTO 211       !  wait for user action
227   !!OFF KBD
228  END IF
```

```
229  OUTPUT Counter;"*rcl 9"         ! re-store counter setup
230  IF (Switch MOD 10000) THEN OUTPUT Switch;"*rcl 9"
                                     ! re-store mux setup
231  OFF TIMEOUT
232  Error=0     ! no error occured
233  SUBEXIT
234 No_cal:!
235  ON TIMEOUT INT(Counter/10000),7 GOTO No_count
236  !  if device timeout then abort
237  OUTPUT Counter;"*rcl 9"         ! re-store counter setup
238  IF (Switch MOD 10000) THEN OUTPUT Switch;"*rcl 9"
                                      ! re-store mux setup
239 No_count:      !
240  OFF TIMEOUT     !  calibration not successful, timeout
241 SUBEND
242   !=============================================
243 SUB Sc_measure(Counter,Master,Slave,Test2,Frequency,Fault)
244  REAL Resolution,Measure(1:3),J,Tout
245  IF Frequency>2000 THEN
246  !  select measurement resolution of counter depending on
                             expected timeinterval
247   Resolution=1.E-10
248  ELSE
249   Resolution=1.E-9
250  END IF
251  Tout=3/Frequency 5
252  Sc_getdelay(Frequency,Resolution,360,Counter,Tout,
                                                  Measure(1))
253  Sc_getdelay(Frequency,Resolution,360,Counter,Tout,
                                                  Measure(2))
254  !!IF (ABS(Measure(2)-Measure(1))>1.5) AND ((ABS
                                        (Measure(1)-180)<179)
256  !! OR (ABS(Measure(2)-180)<179)) THEN
                                  ! both values in tolerance?
257   ON TIMEOUT INT(Counter/10000),5 GOTO End
258   FOR J=180-Test2 TO Test2 STEP (Test2/5-18)
259   ! omit phase shift below 25 KHz for phase changes >
                        25 deg on squarewave signals
260    OUTPUT Slave;":SOUR:PHAS "&VAL$(PROUND(J,-1))&" DEG"
261   NEXT J
262   WAIT .3
263   Sc_getdelay(Frequency,Resolution,540,Counter,Tout,
                                                 Measure(3))
264   !!IF (ABS(Measure(2)-Measure(3))>2) AND ((ABS
                                 (Measure(2)-180)<178.5) OR
265   !! (ABS(Measure(3)-180)<178.5)) THEN
                ! compare second with third measurement
266    Measure(3)=Measure(1)
267    Measure(1)=Measure(3)
268    Sc_getdelay(Frequency,Resolution,540,Counter,Tout,
                                                  Measure(3))
269    !!IF (ABS(Measure(1)-Measure(3))>3) AND ((ABS
                                  (Measure(1)-180)<178) OR
270    !!(ABS(Measure(3)-180)<178)) THEN 246
              ! last two measurements in tolerance?
271    Measure(2)=Measure(3)
272   ELSE
273    Measure(1)=Measure(3)
274   END IF
275   FOR J=Test2 TO 180-Test2 STEP (18-Test2/5)
                                          ! omit phase shift
276   ! below 25 KHz for phase changes > 25 deg on squarewave
                                                  signals
277    OUTPUT Slave;":SOUR:PHAS "&VAL$(PROUND(J,-1))&" DEG"
278   NEXT J
279  END IF
280  IF Measure(1)>355 AND Measure(2)<5 THEN Measure(1)=
                                               Measure(1)-360
281  IF Measure(2)>355 AND Measure(1)<5 THEN Measure(2)=
                                               Measure(2)-360
```

```
282  Fault=PROUND((Measure(1) Measure(2))/2,-1)
                                      ! thats the phase error
283  BEEP 200,.05      !  MEASUREMENT DONE
284 End:OFF TIMEOUT
285 SUBEND
286       !============
287 SUB Sc_getdelay(Frequency,Resolution,Degree,Counter,Tout,
                                                      Measure)
288  OFF TIMEOUT
289  IF Tout<32 THEN
290  ! if calibration needs longer than 32.767 seconds, the
291  ! timeout command can't be used to avoid a hang-up
292   ON TIMEOUT INT(Counter/10000),Tout GOTO T_out
293  ELSE
294  !! ON TIME Tout GOTO T_out
295  END IF
296  OUTPUT Counter;":MEASURE1:TINTERVAL? 1,"&VAL$(Resolution)
297  ENTER Counter;Measure
298  Measure=(PROUND(Measure*Frequency*360,-1) Degree) MOD 360
299 T_out:OFF TIMEOUT
300  !!OFF TIME
301  SUBEND
```

Part 4.

```
10    !E1440A Self-cal error-handling routine.
20    !Prints "self-cal OK" or, if there is a fault,  the
                                             self-cal results.
30    !A timeout recovers system control if the E1440A fails
                                                 to respond.
40    CLEAR SCREEN
50    ASSIGN @Fg TO 71611
60    ABORT 7
70    CLEAR 7
80    CLEAR @Fg
90    !
100   Cal$=" 9"
110   !
120   !
130   OUTPUT @Fg;"*RST;:STATUS:PRESET;*CLS"
140   OUTPUT @Fg;":VOLT 5V" ! This statement causes self-cal
150   ON TIMEOUT 7,3 GOTO Timeout
160   OUTPUT @Fg;":CAL?" ! Self-cal data requested
170   ENTER @Fg;Cal$ ! Self-cal data uploaded
180   IF Cal$=" 0" THEN
190       PRINT "Self-cal ok"
200   ELSE
210     PRINT "Self-cal error ";Cal$;"see Manual p 5-9"
220   END IF
230   GOTO End
240   !
250 Timeout:PRINT "Timeout, E1440A doesn't respond within normal self-cal time (
3 s)"
260   !
270 End:LOCAL @Fg
280   !
290   END
```