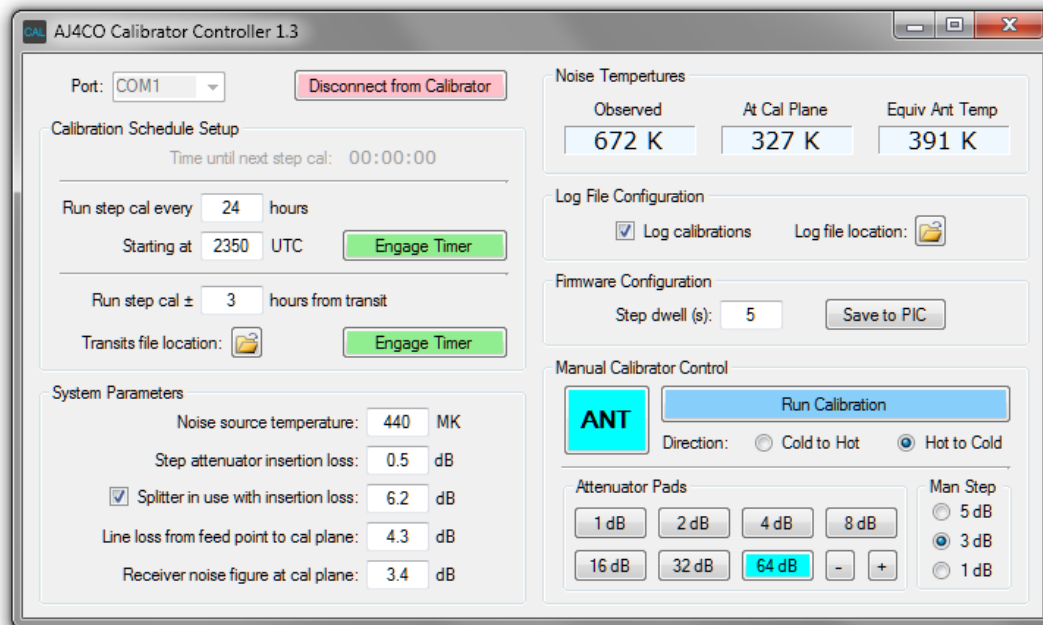


AJ4CO Automatic Calibrator

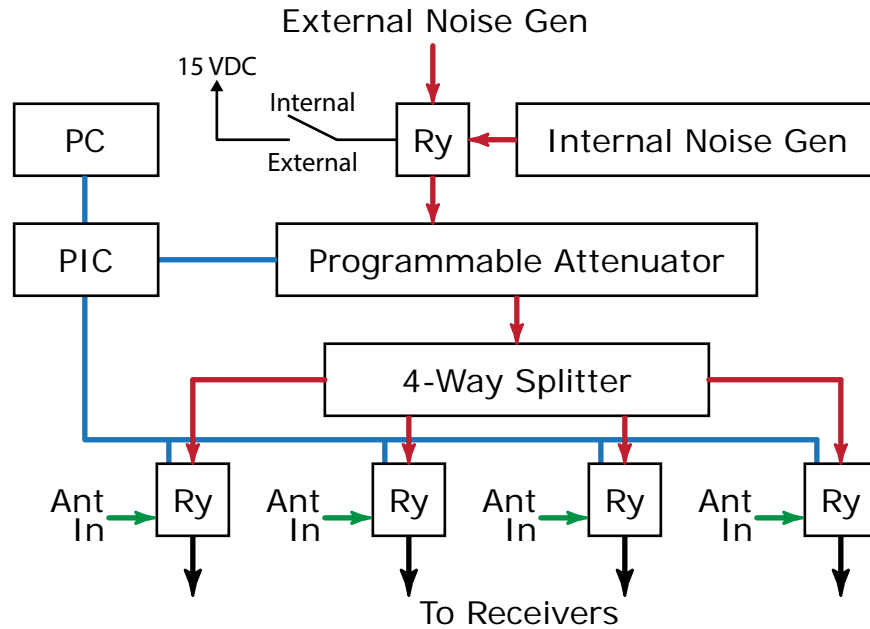
17 Jun 2017



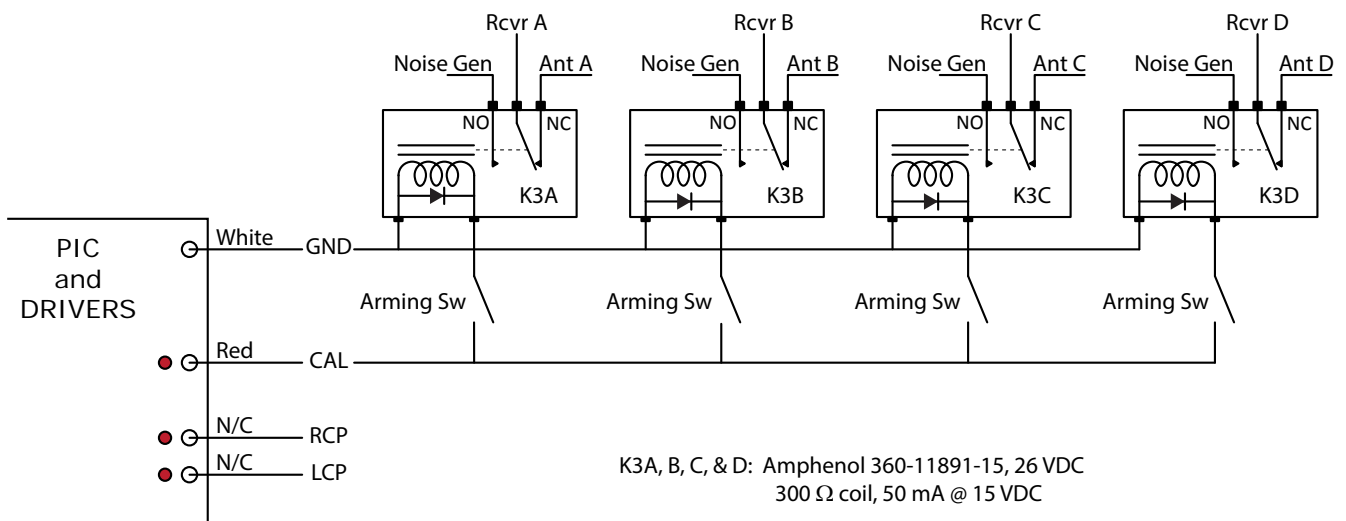
Dave Typinski

HARDWARE

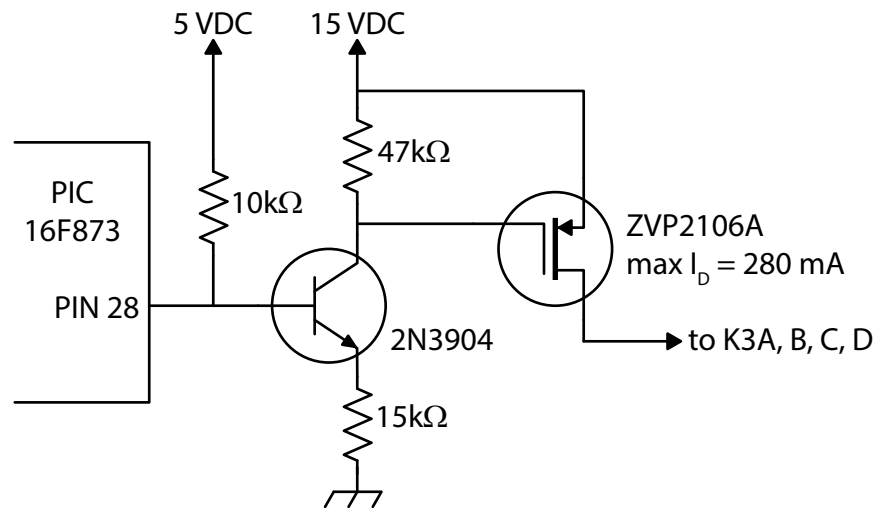
The automatic calibrator hardware is described fully in the FS-200B Instruction Manual on pages 52 through 58 (see pages attached to this document). In brief, software running on a PC controls the firmware running on a PIC, which in turn operates the programmable attenuator and system relays.



The figure on page 53 of the FS-200B Instruction Manual is modified as follows.



The figure on page 57 of the FS-200B Instruction Manual is modified by the addition of the following driver circuit for the K3 line to the CAL/ANT relays.



PHASE UNIFORMITY

The lengths of the coax jumpers between the inputs from the antennas, the feed relays, and the outputs to the receiving systems are identical in order to maintain the relative phase relationship between the four input signals. The four-way splitter on the noise source, however, is connected to the four feed relays with unequal lengths of coax. With the noise source switch set to “external,” this unit may be used as a generic 4-way splitter when no calibration is in progress; however, the phases of the four resulting output signals will not be identical.

FIRMWARE

The firmware in the PIC provides serial communication with the PC and controls the states of the relays and attenuator pads. It contains built-in routines to step from cold to hot or hot to cold in 3 dB steps, 0 dB attenuator setting to 48 dB attenuator setting. It will also accept commands from the PC to set the state of any relay and any attenuator cell. The firmware allows the user to save the step dwell time setting, in seconds from 1 to 255, and saves this setting to non-volatile RAM. The current firmware version is AJ4COCALv02.asm. The assembler source code is available in Appendix A.

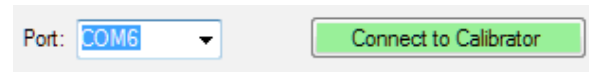
SOFTWARE

The Calibrator Controller (CC) software was developed using Visual Basic 2010 and .NET 4.0 on a Win7 x64 machine and a WinXP SP3 machine. Source code is available in Appendix B.

The system works using a PC’s native serial port. USB-to-serial adapters are unreliable and are not recommended. If the host PC does not have a serial port, then addition of a serial port PCI card is recommended.

1. Communications Setup

Select the appropriate COM port and press the green Connect to Calibrator button. The software detects all available COM ports; it is up to the user to select the port that is connected to the calibrator hardware. Once the Connect to Calibrator button is pressed, the button turns red and the COM port selection box is disabled.



2. Calibration Schedule Setup

While CC allows step calibrations to be performed manually (see below), CC has the ability to run the automatic step calibration routine on two different user-specified schedules.

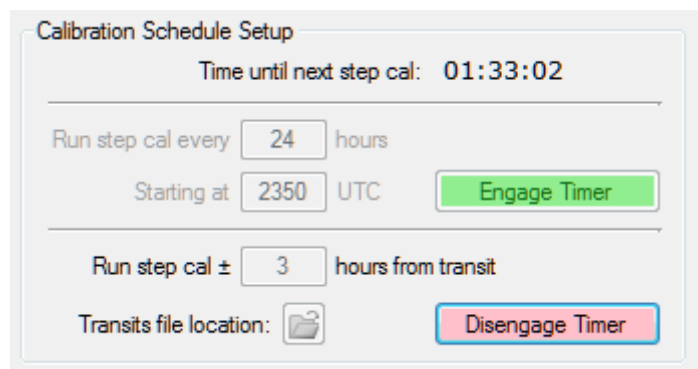
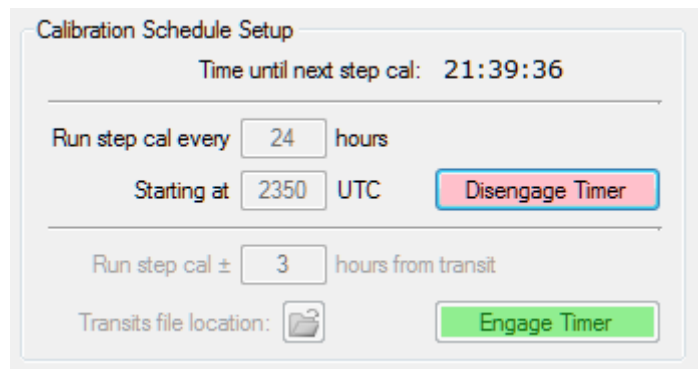
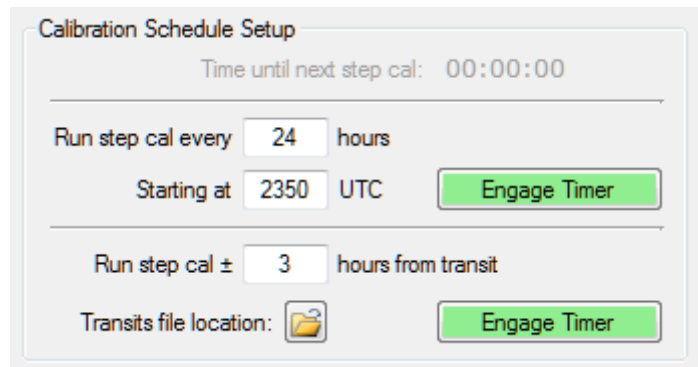
To repeat a step cal every fixed number of hours, enter the time to wait between step calibrations and the time to start the first calibration, then press the associated Engage Timer button.

Once the associated Engage Timer button is pressed, a countdown timer becomes active showing the time until the next step calibration, the timer button turns red, and the time entry fields are disabled. When the countdown reaches zero, the automatic calibration will start and the counter will reset for the next step cal. This will continue until the timer is manually disengaged.

To run an automatic step cal a fixed number of hours before and after transit, enter the number of hours and select the location of the transits file, then press the associated Engage Timer button.

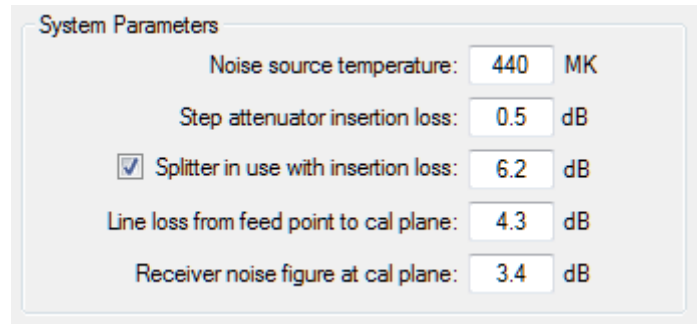
The current transit file includes Jupiter transit times for AJ4CO Observatory from 2017 through 2060. Transits are stored in a text file and may be modified as long as the date and time formats are not altered. The format is YYYY-MM-DD HH:MM, one transit per line. A table of transit times may be generated using JPL Horizons. The transits file must be organized in ascending order; i.e., earliest transit first.

Once the associated Engage Timer button is pressed, the timer acts as described above. This will continue until the timer is manually disengaged or the transit file runs out.



3. System Parameters

Noise source temperature: The temperature at the output of the internal noise generator at the point where it connects to the calibrator's attenuator (440 MK for the internal noise generator) or, if used, the temperature of an external noise source where it connects to the automatic calibrator's front panel.



Step attenuator insertions loss: Enter the insertion loss of the step attenuator (0.5 dB for the internal Kay 4450 step attenuator). Additional corrections between 0 and 0.09 dB are hard-coded into the PC software to account for the difference between the attenuations measured with a VNA and the average 0.5 dB insertion loss of the internal attenuator. See page 9 for a table listing the measured attenuation at each step.

Splitter in use with insertion loss: For dual polarization systems, the output of the calibrator must be split to feed both halves of the receiver system. If using such a splitter, check the box and enter the total loss of the splitter (6.2 dB for the internal Mini-Circuits ZSC-4-1).

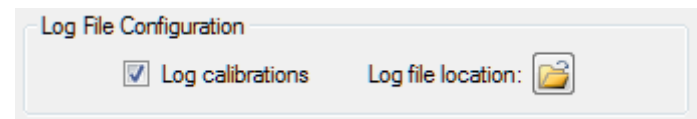
Line loss from feed point to cal plane: The total loss between the antenna feed point(s) and the calibration plane (4.3 dB for the TFD array at AJ4CO Observatory). The calibration plane is defined by the location of the CAL relay(s).

Receiver noise figure at cal plane: The receiver's noise figure referenced to the calibration plane (3.4 dB for the Dual Polarization Spectrograph at AJ4CO Observatory). Note: the DPS itself has a higher noise figure at its input connectors, about 6 dB. However, there is a wide band hybrid ring with a loss of 0.3 dB followed by multicouplers with a gain of 13 dB between the calibration plane and the DPS. These components act to reduce the noise figure at the calibration plane to about 3.4 dB. This highlights how the position of the cal plane within a radio telescope can make a big difference in the noise figure present at the cal plane.

4. Log File Configuration

CC has the ability to write a log file, recording the states of the relays and attenuator.

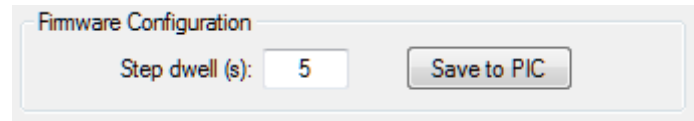
To use this feature, place a check in Log checkbox and use the folder icon button to



select the destination folder. The system parameters are also written to the log file, along with the three temperatures shown on the user interface (see below). A new line is added to the log file every time a relay or the attenuator changes state. Log files are limited to one month of operation to keep the file size manageable. New files are created as needed. The log files are in CSV format. The file name format is Cal_Log_YYYY_MM.csv. See page 10 for an example of the log file.

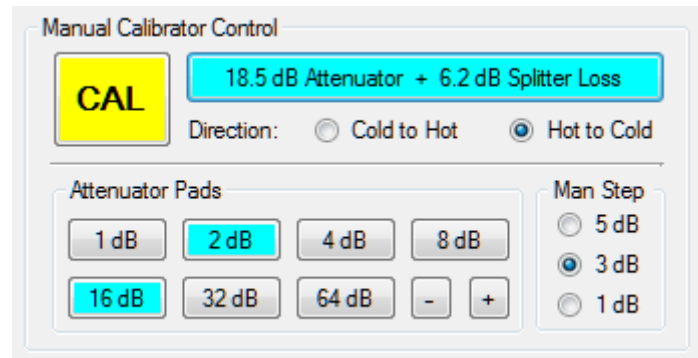
5. Firmware Configuration

Step dwell (s): Indicates the length of time the automatic calibration routine dwells at each step. The default is five seconds. Valid dwell is any integer between 1 and 255 seconds, inclusive. To change the dwell time, simply enter the desired dwell and click the Save to PIC button.



6. Manual Calibrator Control

Run Calibration: Pressing this button manually triggers the automatic calibration routine to step through a range of 0 to 48 dB attenuation in 3 dB steps. Such a manually triggered calibration will not affect the cal schedule; the timer will continue to run as if the manually triggered routine did not happen. During the cal routine, the Run Calibration button text changes to show the current losses in terms of dB between the noise source and the calibration plane. The relay button colors and the step attenuator pad button colors also change to reflect the current status of the relays and pads.



Direction: This setting determines the attenuation step direction, 0 to 48 dB, or 48 to 0 dB.

ANT/CAL: The ANT/CAL relay buttons, in addition to being a state indicator, also acts to toggle the state of the ANT/CAL relays by simply clicking on the button. The button is active any time CC is connected to the calibrator. Toggling the ANT/CAL relay to the CAL state switches the Run Calibration button text to display the losses as described in the paragraph above.

Attenuator Pads: Like the ANT/CAL relay button, the Attenuator Pads buttons allow the user to manually change the state of any pad. They are active any time CC is connected to the calibrator. The plus (+) and minus (-) buttons allow the user to manually increment or decrement the current attenuator by the number of dB selected in the Man Step selection. The up-arrow and down-arrow keyboard keys are shortcuts for the + and - buttons, respectively.

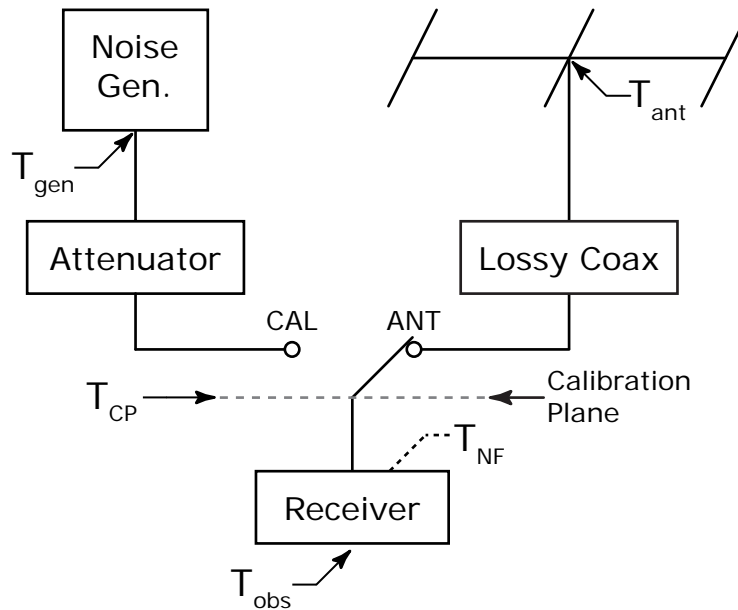
Man Step: This selection controls the step size of the + and - buttons only, it does not change the automatic step cal routine, which is hard-coded with 3 dB steps.

7. Temperature Displays

CC displays three temperatures any time it is connected to the calibrator hardware. Temperatures are shown with three significant figures and the appropriate units designator. In reality, the accuracy of the displayed temperatures is probably somewhere between two and three significant figures.

Noise Temperatures		
Observed	At Cal Plane	Equiv Ant Temp
24.0 kK	23.7 kK	63.3 kK

The conceptual diagram below shows the relevant noise temperatures and their locations within the radio telescope.



At Cal Plane: This is the actual noise temperature at the calibration plane. It is the noise generator output, less the summed losses of the attenuator insertion loss, the attenuator setting, and the splitter insertion loss, plus the noise input from the attenuator and the splitter. The equation used is:

$$T_{CP} = \frac{T_{gen}}{L_{att}} + T_{att} \left(1 - \frac{1}{L_{att}} \right)$$

where

L_{att} is loss factor = $10^{\left(\frac{dB}{10}\right)}$ where +dB is the attenuation

T_{gen} is the noise generator temp

T_{att} is the physical temperature of the attenuator, assumed to be 290 K

T_{CP} is the temperature at the calibration plane

Observed: This is the temperature one would see on the front panel of a hypothetical receiver having the noise figure stated in the System Parameters. This shows the effect of the receiver's internal noise. This effect is generally very small and is only evident when the calibration plane is at relatively low temperatures, less than a few kK for a well designed HF receiver. The effect of receiver noise can become significant for receivers with a higher noise figure and antennas with low efficiency and low directivity. The equation used is:

$$T_{Obs} = T_{CP} + T_{rcvr} (F_{rcvr} - 1)$$

where

F_{rcvr} is receiver noise factor = $10^{\left(\frac{NF}{10}\right)}$ where NF is the receiver's noise figure

T_{CP} is the temperature at the calibration plane

T_{rcvr} is the physical temperature of the receiver, assumed to be 290 K

T_{Obs} is the temperature indicated by the receiver

Equip Ant Temp: This is the noise temperature that would have to exist at the antenna terminals in order to produce the same noise temperature at the calibration plane. This accounts for the antenna feed system losses and the physical temperature of the feed system. This is the most important number produced by the software, for this is the temperature used to calibrate the radio telescope's data. In terms of the noise source temperature, the equivalent antenna temperature is found using the following equation:

$$T_{Ant} = L_{coax} \left[\frac{T_{gen}}{L_{att}} + T_{att} \left(1 - \frac{1}{L_{att}} \right) - T_{coax} \left(1 - \frac{1}{L_{coax}} \right) \right]$$

where

L_{att} is the attenuator loss factor = $10^{\left(\frac{dB}{10}\right)}$ where +dB is the attenuation

L_{coax} is the feed line loss factor = $10^{\left(\frac{dB}{10}\right)}$ where +dB is the attenuation

T_{gen} is the noise generator temp

T_{att} is the physical temperature of the attenuator, assumed to be 290 K

T_{Ant} is the equivalent antenna temperature

Automatic Calibrator Temperatures

T ₀ (K)	290	
Noise Source Temperature (MK)	440	
Splitter Loss @ 20 MHz (dB)	6.2	
Effective Noise Source Temp (MK)	106	(after splitter)
Antenna Feed Loss @ 20 MHz (dB)	4.3	
DPS Noise Figure @ 20 MHz (dB)	3.4	= 344 K @ HYBRID INPUTS

Calibration Plane: CAL relays between antenna feed panel outputs and hybrid ring inputs.

Nom. Att. (dB)	Meas. Att. (dB)	Observed Temp. (K)	Equivalent Antenna Temp. (K)	Nom. Att. (dB)	Meas. Att. (dB)	Observed Temp. (K)	Equivalent Antenna Temp. (K)
0	0.56	92.8 MK	250 MK	0	0.56	92.8 MK	250 MK
1	1.52	74.4 MK	200 MK	3	3.43	47.9 MK	129 MK
2	2.56	58.5 MK	158 MK	6	6.47	23.8 MK	64.0 MK
4	4.57	36.8 MK	99.2 MK	9	9.45	12.0 MK	32.2 MK
8	8.55	14.7 MK	39.7 MK	12	12.58	5.83 MK	15.7 MK
16	16.58	2.32 MK	6.24 MK	15	15.48	2.99 MK	8.04 MK
32	32.50	60.0 kK	160 kK	18	18.55	1.47 MK	3.97 MK
64	64.65	671 K	388 K	21	21.50	748 kK	2.01 MK
				24	24.55	371 kK	997 kK
				27	27.51	188 kK	504 kK
				30	30.58	93.0 kK	249 kK
				33	33.49	47.9 kK	128 kK
				36	36.54	24.0 kK	63.3 kK
				39	39.55	12.3 kK	31.8 kK
				42	42.55	6.50 kK	16.1 kK
				45	45.59	3.55 kK	8.14 kK
				48	48.55	2.11 kK	4.26 kK

Excerpt from Log File – Cal Log 2017 06.csv

Date	Timestamp	Cal Ry State	Att Setting (dB)	T_ant	T_calplane	T_observed	T_gen (MK)	Att Ins Loss (dB)	Splitter Ins Loss (dB)	Attn Total (dB)	Feed Line Loss (dB)	Noise Figure (dB)
17 Jun 2017	04:10:00.095	CAL	64.15	387 K	326 K	671 K	440	0.5	6.2	70.85	4.3	3.4
17 Jun 2017	04:10:00.125	CAL	64.15	387 K	326 K	671 K	440	0.5	6.2	70.85	4.3	3.4
17 Jun 2017	04:10:05.142	CAL	0.06	250 MK	92.8 MK	92.8 MK	440	0.5	6.2	6.76	4.3	3.4
17 Jun 2017	04:10:10.166	CAL	2.93	129 MK	47.9 MK	47.9 MK	440	0.5	6.2	9.63	4.3	3.4
17 Jun 2017	04:10:15.188	CAL	5.97	64.0 MK	23.8 MK	23.8 MK	440	0.5	6.2	12.67	4.3	3.4
17 Jun 2017	04:10:20.210	CAL	8.95	32.2 MK	12.0 MK	12.0 MK	440	0.5	6.2	15.65	4.3	3.4
17 Jun 2017	04:10:25.234	CAL	12.08	15.7 MK	5.83 MK	5.83 MK	440	0.5	6.2	18.78	4.3	3.4
17 Jun 2017	04:10:30.256	CAL	14.98	8.04 MK	2.99 MK	2.99 MK	440	0.5	6.2	21.68	4.3	3.4
17 Jun 2017	04:10:35.279	CAL	18.05	3.97 MK	1.47 MK	1.47 MK	440	0.5	6.2	24.75	4.3	3.4
17 Jun 2017	04:10:40.302	CAL	21.00	2.01 MK	748 KK	748 KK	440	0.5	6.2	27.7	4.3	3.4
17 Jun 2017	04:10:45.325	CAL	24.05	997 KK	371 KK	371 KK	440	0.5	6.2	30.75	4.3	3.4
17 Jun 2017	04:10:50.348	CAL	27.01	504 KK	188 KK	188 KK	440	0.5	6.2	33.71	4.3	3.4
17 Jun 2017	04:10:55.370	CAL	30.08	249 KK	92.6 KK	93.0 KK	440	0.5	6.2	36.78	4.3	3.4
17 Jun 2017	04:11:00.393	CAL	32.99	127 KK	47.5 KK	47.9 KK	440	0.5	6.2	39.69	4.3	3.4
17 Jun 2017	04:11:05.415	CAL	36.04	63.3 KK	23.7 KK	24.0 KK	440	0.5	6.2	42.74	4.3	3.4
17 Jun 2017	04:11:10.438	CAL	39.05	31.8 KK	12.0 KK	12.3 KK	440	0.5	6.2	45.75	4.3	3.4
17 Jun 2017	04:11:15.462	CAL	42.05	16.1 KK	6.16 KK	6.5 KK	440	0.5	6.2	48.75	4.3	3.4
17 Jun 2017	04:11:20.483	CAL	45.09	8.13 KK	3.2 KK	3.55 KK	440	0.5	6.2	51.79	4.3	3.4
17 Jun 2017	04:11:25.507	CAL	48.05	4.26 KK	1.76 KK	2.11 KK	440	0.5	6.2	54.75	4.3	3.4
17 Jun 2017	04:11:30.529	CAL	64.15	387 K	326 K	671 K	440	0.5	6.2	70.85	4.3	3.4
17 Jun 2017	04:11:35.527	ANT	64.15	387 K	326 K	671 K	440	0.5	6.2	70.85	4.3	3.4

Software / Hardware Change Log

Version 1.0.0

Original beta test program, limited functionality, basically a proof-of-concept.

Version 1.1.0

Fully functional application to drive the unmodified UFRO automatic calibrator hardware with PIC firmware version CALIB5.asm (note: not CALIB05.asm).

Version 1.2.0

Works with the modified hardware and firmware version AJ4COCALv02.asm.

Calibrator hardware changes:

- Installed internal 5 and 12 volt regulators powered from single 15 VDC supply.
- Installed power switch and LED power indicator.
- Changed the CAL relay driver circuit to use a power MOSFET.
- Removed the RCP/LCP/CAL relay module.
- Installed two SPDT coaxial relays driven in parallel by the CAL relay control line.
- Installed splitter between the programmable attenuator and the two CAL relays.

PIC firmware changes:

- New delay routines, step delays possible from 1 sec to 255 sec.
- Added more steps, now runs 0 to 48 dB in 3 dB steps.
- Step cal runs cold to hot or hot to cold.

PC software changes:

- Added 1, 2, and 5 dB step radio buttons for the + and – increment buttons.
- Removed the RCP and LCP relay buttons. The relays aren't used in the modified version of the hardware. Their output states are now used as a confirmation that the PIC is talking to the application software.
- Added "cold to hot" and "hot to cold" radio buttons.

Version 1.2.1

Makes the log CSV file slightly easier to read.

PC software changes:

- Added a comma between the date and time portions of the log file date stamps.

Version 1.3.0 (latest version)

Works with the upgraded 4-port noise generator and adds transit cal timing option.

Calibrator hardware changes:

- Installed internal 440 MK noise generator.
- Installed 4-way power splitter, associated relays, and arming switches.
- Installed internal/external noise source relay and switches.
- Mounted all within a 3U 19" rack chassis.

PC software changes:

- Added transit cal timing option.

AUTOMATIC CALIBRATOR

POWER



NOISE SOURCE

INTERNAL

EXTERNAL



EXT NOISE IN



ARM CAL

DISABLE



FEED A



ARM CAL

DISABLE



FEED B



ARM CAL

DISABLE



FEED C

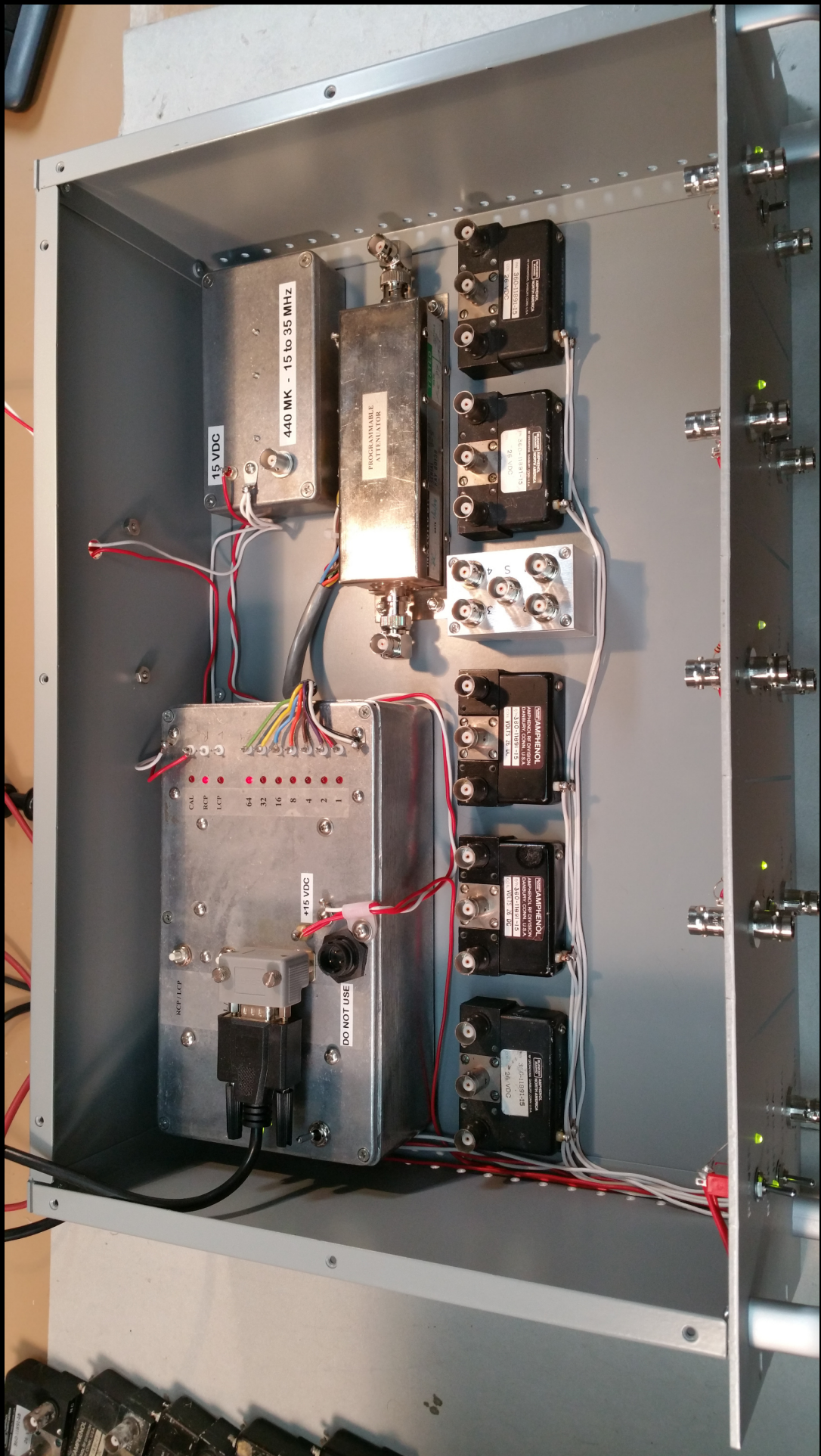


ARM CAL

DISABLE



FEED D



15 VDC

440 MK - 15 to 35 MHz

PROGRAMMABLE ATTENUATOR

AMPHENOL
DO NOT USE

AMPHENOL
DO NOT USE

AMPHENOL
DO NOT USE

AMPHENOL
DO NOT USE

AMPHENOL
DO NOT USE

CAL

RCP

LCP

64

32

16

8

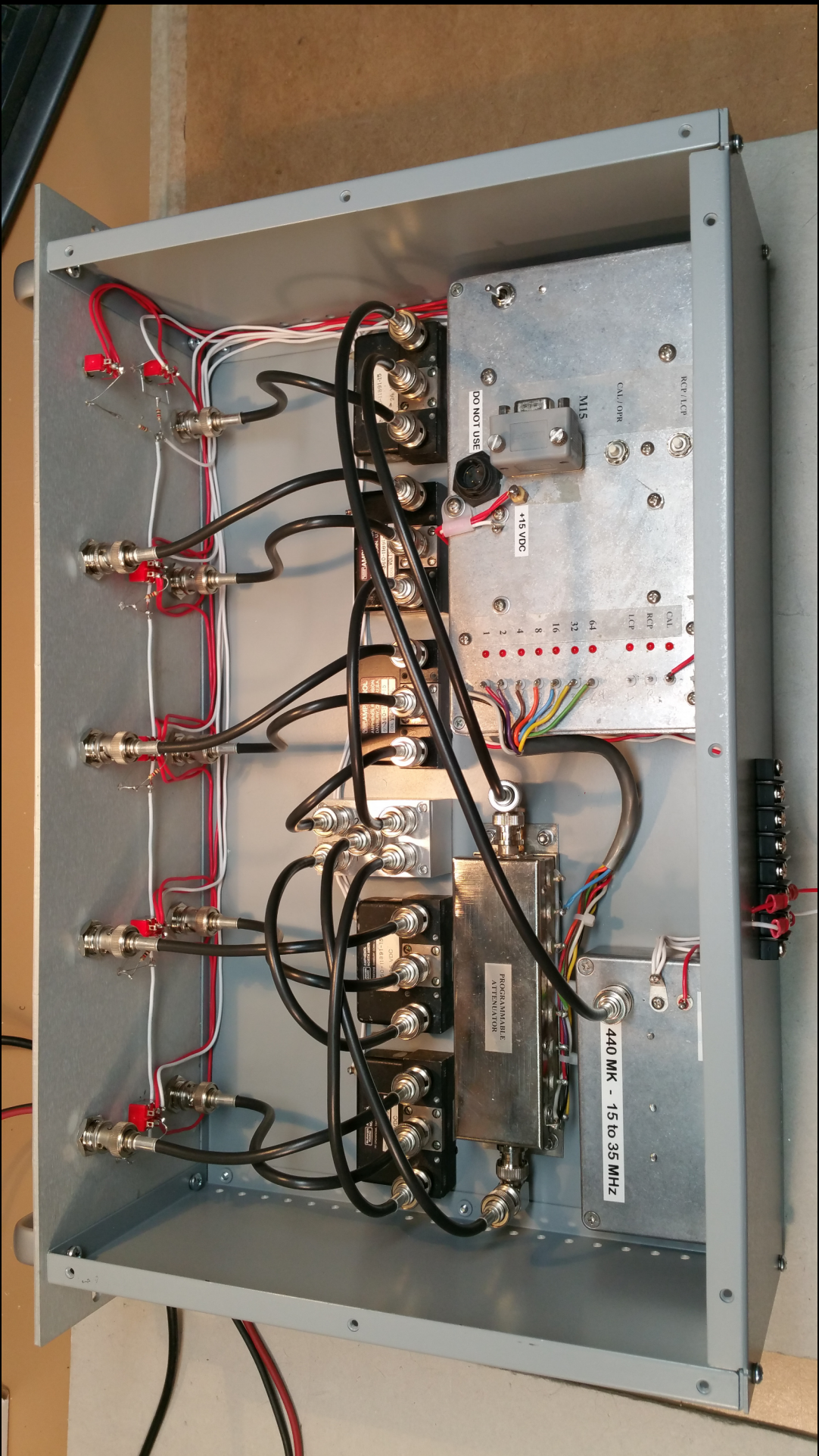
4

2

1

*15 VDC

DO NOT USE



DO NOT USE

+15 VDC

M15

CAL/OHR

RCP/LCP

64

32

16

8

4

2

1

OAL

RCP

LCP

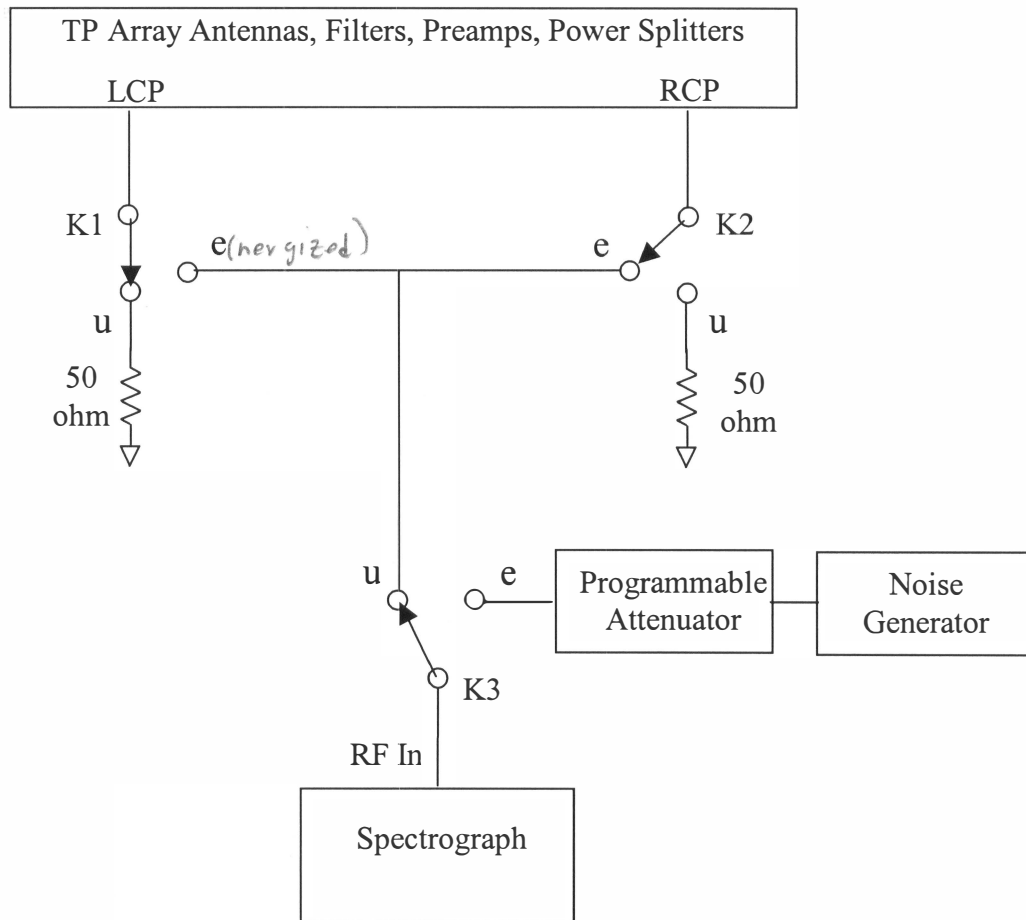
PROGRAMMABLE
ATTENUATOR

440 MK - 15 to 35 MHz

Mode Switch/Calibrator

A mode switch (M16) with 3 coaxial relays (K1, K2, and K3) selects the RF source for the spectrograph (for the Florida installation this is either RCP, LCP or a calibrator).

The calibrator comprises a high temperature RF noise source followed by a commercial programmable attenuator.



K1 is LCP select relay (energize to select LCP)

K2 is RCP select relay (energize to select RCP)

K3 is CAL/OPR relay (energize for CAL)

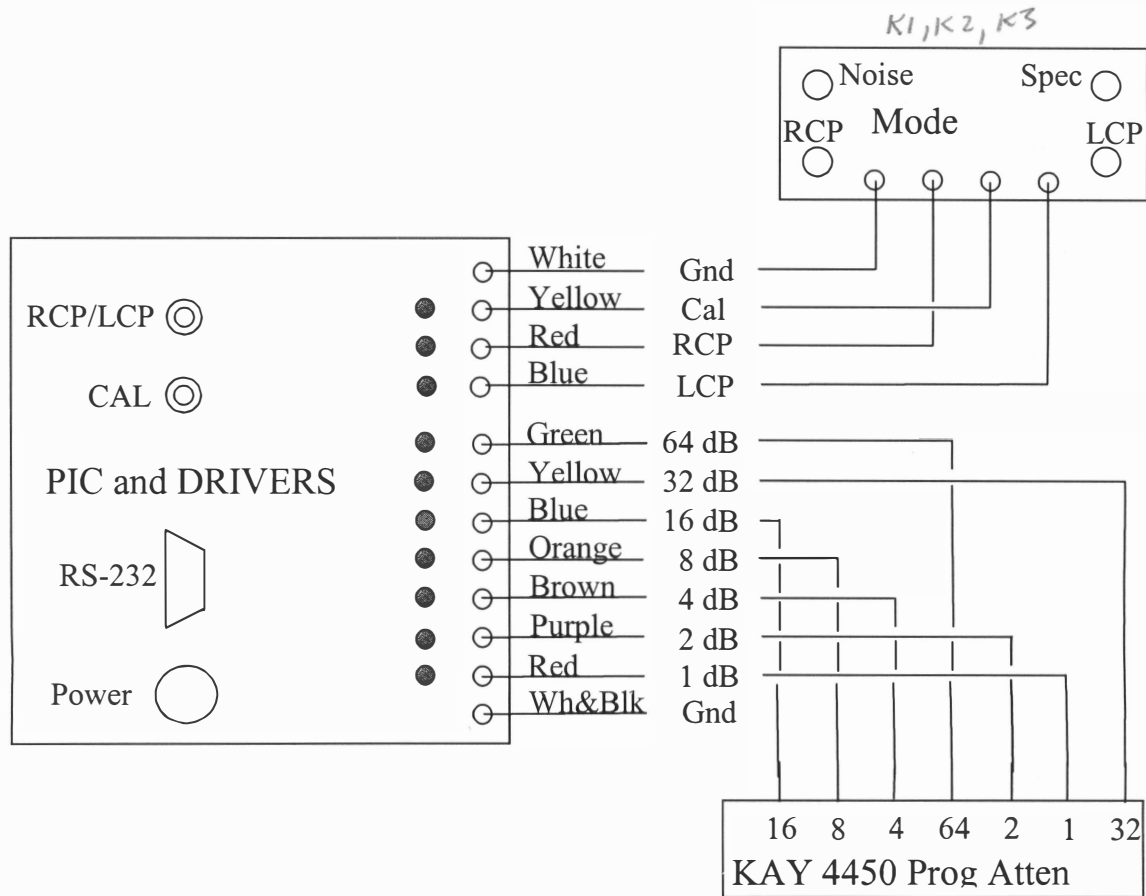
During CAL both K1 and K2 are un-energized, terminating both antenna ports in 50 ohms

Both the mode switch and the step attenuator are controlled by a microcontroller (PIC), which communicates with the a300y computer via an RS232 serial link.

The mode switch/calibrator components are mounted on a 5.25" by 19" wide rack panel.

M15 – A PIC microcontroller and driver circuit generate command signals to control a commercial step attenuator and the mode switch. Three control lines operate the LCP, RCP, Cal/Opr Mode switch and seven control lines drive the individual 1, 2, 4, 8, 16, 32, and 64 dB cells of the step attenuator. The PIC communicates with the a300y computer via an RS232 line.

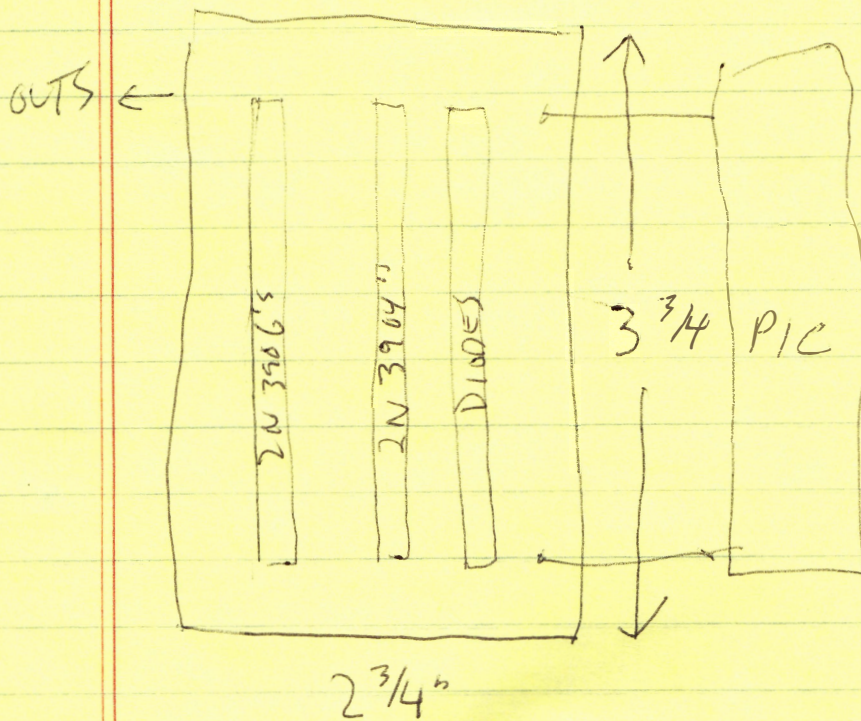
The physical layout of the modules and their interconnecting signal cables is seen below.



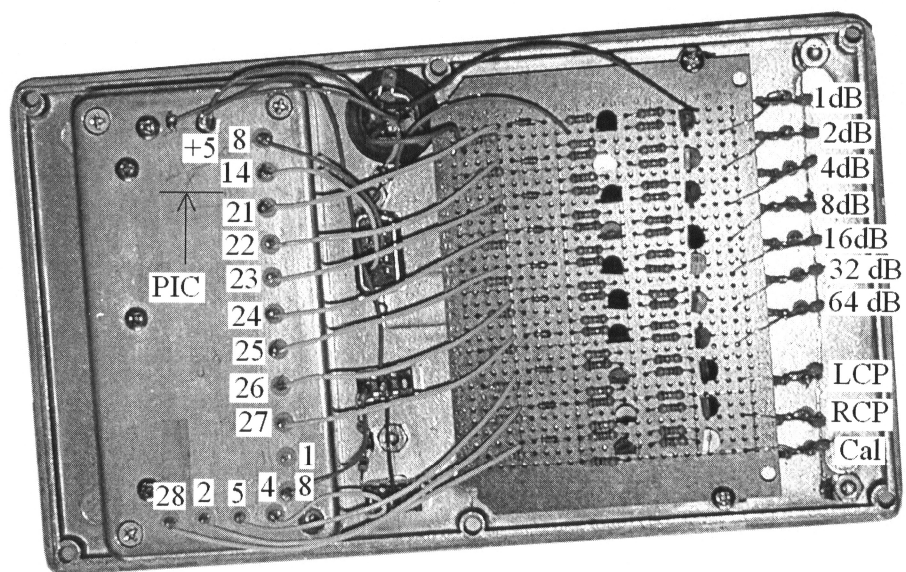
The PIC controller module is connected to the Elenco Power Supply by a 3-wire cable supplying +5, and +12 volts.

Pushbuttons on M15 can be used to test the RCP/LCP polarization select relays as well as the calibrate function. One push of the CAL button initiates the CAL sequence while pushing the RCP/LCP button toggles the spectrograph input between the RCP and LCP antennas.

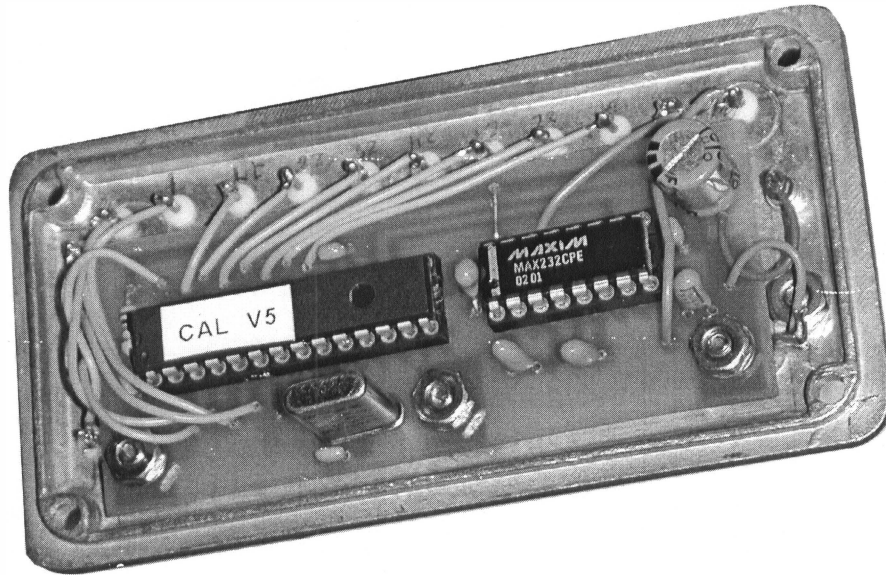
CAL PC BOARD



12/30/07 Replaced 2N3904 + 2N3906
on "4" cal bit.



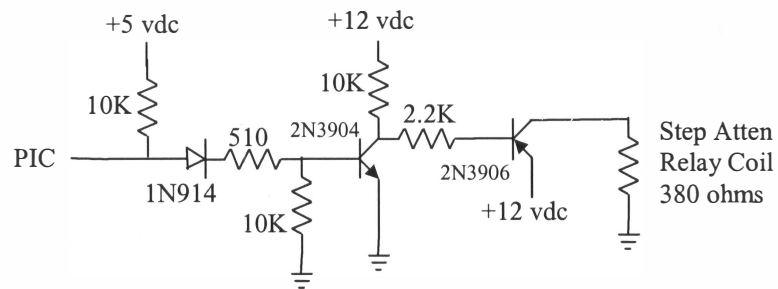
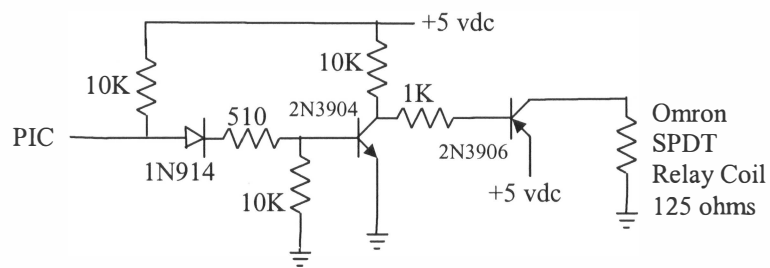
PIC Microcontroller and Driver Module (M15)
 Numbers on PIC controller box indicate PIC and MAX-232 pin numbers.



The PIC16F873 microcontroller communicates with the main a300y computer via a MAX232 RS-232 chip. Serial communications rate is 56 K bits, which is derived from an 18.432 MHz crystal. The 16F873 contains the custom control code for the mode switch and the calibrator programmable attenuator. The PIC and MAX232 are mounted on a PC board inside of a small diecast box. Connections to the driver board and RS232 9 pin D-connector are via Teflon insulated feedthroughs. The PIC box is mounted inside a larger diecast box, which also contains the relay driver board.

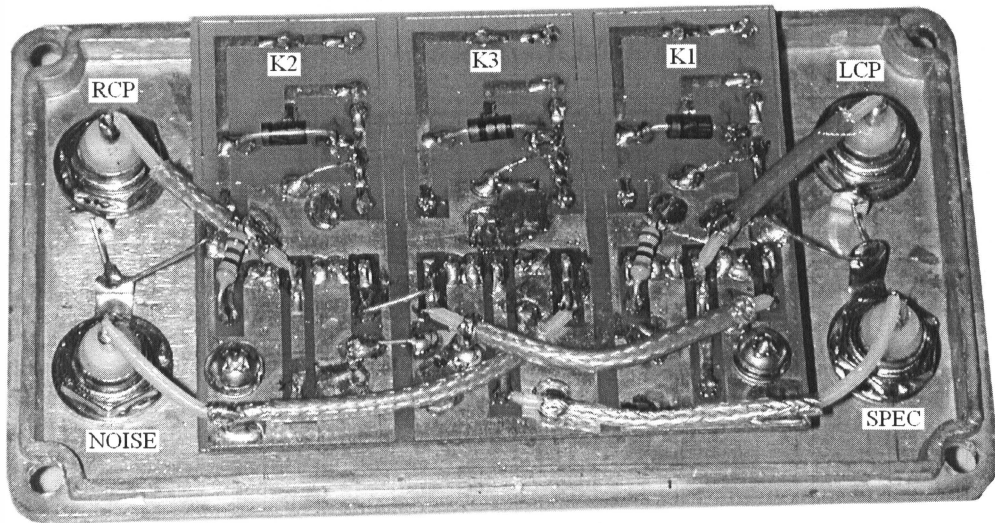
Driver circuits.

Two different driver circuits are used – one to control the 5 volt OMRON RF relays in the mode switch (M16) and the other, a 12 volt circuit used to drive the solenoids in the Kay programmable step attenuator. Each relay solenoid requires a dedicated driver circuit. The drivers turn on their solenoid when a +5v signal is applied by the PIC.



M16 Mode Switch

The mode switch contains three, 5 volt, SPDT Omron RF relays. These relays allow selection of either RCP or LCP signals and also control selection between calibrate and operate modes. The relays are controlled by signals generated by the PIC and amplified in the driver circuits located in module M15.



```
1 ; AJ4COCALv02.ASM
2 ; Written by Dave Typinski, August, 2015
3 ; This is a modified version of the CALIB5.asm code
4 ; written by Jim Sky in July, 2003
5 ;
6 Title "Calibrator Test"
7 ;
8 list P = 16F873
9 ;
10 #include "p16F873.inc"
11 ;
12 ; PIC16F873 Configuration Bit Settings
13 ; CONFIG
14 ; External crystal oscillator
15 ; Watchdog timer is OFF
16 ; Power-up timer is OFF
17 ; Program code protection is OFF
18 ; Brown-out RESET Enable is ON
19 ; Low voltage programming is OFF (otherwise RB3 cannot be an
    output port)
20 ; Data EEPROM code protection is OFF
21 ; FLASH program memory write enable is ON
22 ; __config 0xFF79
23 __CONFIG _FOSC_XT & _WDTE_OFF & _PWRTE_OFF & _CP_OFF & _BOREN_ON &
    _LVP_OFF & _CPD_OFF & _WRT_ON
24 ;
25 ; CRYSTAL = 18.432 MHz
26 ;
27 ;
28 CAL1DB equ 0x00 ; control pins
29 CAL2DB equ 0x01
30 CAL4DB equ 0x02
31 CAL8DB equ 0x03
32 CAL16DB equ 0x04
33 CAL32DB equ 0x05
34 CAL64DB equ 0x06
35 KLCP equ 0x03
36 KRCP equ 0x00
37 KCAL equ 0x07
38 CALBTN equ 0x01
39 RLBTN equ 0x02
40 ;
41 ; -----
42 ; RAM DEFINITIONS
43 ; -----
44 ;
```

```
45     CBLOCK 0x20
46     RLSTAT
47     DelayL    ; low byte for 25 ms delay counter
48     DelayH    ; high byte for 25 ms delay counter
49     CalDLY    ; step cal 1-second delay counter
50     DelayStore ; number of 1-second delays to perform per cal step
51     Temp
52     Temp2
53     RxHold
54     MessPt
55     BCount
56     d1      ; 1-sec delay counter 1
57     d2      ; 1-sec delay counter 2
58     d3      ; 1-sec delay counter 3
59
60     ENDC
61
62     ;
63     ; -----
64     ; CODE START AND INITIALISE PORTS
65     ; -----
66     ;
67
68     org 0000h    ; start address = 0000h
69
70     goto Start
71     ;
72
73     Start movlw 0x00    ; PORTA = all low
74     movwf PORTA
75     movlw 0x00    ; PORTB = all low
76     movwf PORTB
77     bsf STATUS,RP0    ; RAM Page 1
78     ; SET ADCON1 so that all RA pins are digital
79     movlw b'00000110'
80     movwf ADCON1
81
82     movlw b'00000110' ; PORTA = all outputs except for RA1 and RA2
83     movwf TRISA
84     movlw b'00000000' ; PORTB = all outputs
85     movwf TRISB
86
87     ; -----
88     ; PROVIDE A SETTling TIME FOR START UP
89     ; -----
90     ;
```



```
91 ; clrf Temp
92 ;settle decfsz Temp,F
93 ; goto settle
94 ; call Delay25 ; 25mS delay
95
96
97 bcf STATUS,RP0 ; RAM Page 0
98
99
100 ; -----
101 ; SET BAUD RATE
102 ; -----
103 ; Boot Baud Rate = 57600 and 1 Stop Bit
104 ;
105 bsf STATUS,RP0 ; RAM Page 1
106 movlw d'4' ; 57600 baud @ 18.432MHz
107 movwf SPBRG
108 movlw b'00100000' ; brgh = low
109 movwf TXSTA ; enable Async Transmission
110 bcf STATUS,RP0 ; RAM Page 0
111 movlw b'10010000' ; enable UART and
112 movwf RCSTA ; 8 bit reception
113
114 ; -----
115 ; SET BAUD RATE
116 ; -----
117 ; Boot Baud Rate = 19200 and 1 Stop Bit
118 ;
119 ; bsf STATUS,RP0 ; RAM Page 1
120 ; movlw d'14' ; 19200 baud @ 18.432 MHz
121 ; movwf SPBRG
122 ; movlw b'00100000' ; brgh = low
123 ; movwf TXSTA ; enable Async Transmission
124 ; bcf STATUS,RP0 ; RAM Page 0
125 ; movlw b'10010000' ; enable UART and
126 ; movwf RCSTA ; 8 bit reception
127
128
129
130 movlw 0x00 ; PORTA = all low
131 movwf PORTA
132 movlw 0x00 ; PORTB = all low
133 movwf PORTB
134 call Delay25
135 call Delay25
136 call CalRelayOff ; make extra sure cal relay is off
```

```
137     call GetSettings  ; make sure RLSTAT and DelayStore hold the
      values they last had
138     call RecallCP    ; set K1 or K2 accordingly
139
140
141 MainLoop call RcvNoWait    ; anything coming from the PC?
142     call ProcRcv
143     call PToggle
144     call ManCal
145     goto MainLoop
146
147 ;
148 ; Process the Received byte
149 ;
150
151
152 ProcRcv movf RxHold,W
153     xorlw d'10'
154     btfsc STATUS,Z
155     bsf PORTB, d'0'    ; turn on Port B bit 0
156
157     movf RxHold,W
158     xorlw d'20'
159     btfsc STATUS,Z
160     bcf PORTB, d'0'    ; turn off Port B bit 0
161
162     movf RxHold,W
163     xorlw d'11'
164     btfsc STATUS,Z
165     bsf PORTB, d'1'    ; turn on Port B bit 1
166
167     movf RxHold,W
168     xorlw d'21'
169     btfsc STATUS,Z
170     bcf PORTB, d'1'    ; turn off Port B bit 1
171
172     movf RxHold,W
173     xorlw d'12'
174     btfsc STATUS,Z
175     bsf PORTB, d'2'    ; turn on Port B bit 2
176
177     movf RxHold,W
178     xorlw d'22'
179     btfsc STATUS,Z
180     bcf PORTB, d'2'    ; turn off Port B bit 2
181
```

```
182     movf RxHold,W
183     xorlw d'13'
184     btfsc STATUS,Z
185     bsf PORTB, d'3'      ; turn on Port B bit 3
186
187     movf RxHold,W
188     xorlw d'23'
189     btfsc STATUS,Z
190     bcf PORTB, d'3'      ; turn off Port B bit 3
191
192     movf RxHold,W
193     xorlw d'14'
194     btfsc STATUS,Z
195     bsf PORTB, d'4'      ; turn on Port B bit 4
196
197     movf RxHold,W
198     xorlw d'24'
199     btfsc STATUS,Z
200     bcf PORTB, d'4'      ; turn off Port B bit 4
201
202     movf RxHold,W
203     xorlw d'15'
204     btfsc STATUS,Z
205     bsf PORTB, d'5'      ; turn on Port B bit 5
206
207     movf RxHold,W
208     xorlw d'25'
209     btfsc STATUS,Z
210     bcf PORTB, d'5'      ; turn off Port B bit 5
211
212
213     movf RxHold,W
214     xorlw d'16'
215     btfsc STATUS,Z
216     bsf PORTB, d'6'      ; turn on Port B bit 6
217
218     movf RxHold,W
219     xorlw d'26'
220     btfsc STATUS,Z
221     bcf PORTB, d'6'      ; turn off Port B bit 6
222
223     movf RxHold,W
224     xorlw d'17'
225     btfsc STATUS,Z
226     bsf PORTB, d'7'      ; turn on Port B bit 7
227
```

```
228     movf RxHold,W
229     xorlw d'27'
230     btfsc STATUS,Z
231     bcf PORTB, d'7'    ; turn off Port B bit 7
232
233     movf RxHold,W
234     xorlw 'S'
235     btfsc STATUS,Z
236     call SendStatus    ; Get Status
237
238     movf RxHold,W
239     xorlw d'130'
240     btfsc STATUS,Z
241     call SetLCP    ; switch to LCP
242
243     movf RxHold,W
244     xorlw d'131'
245     btfsc STATUS,Z
246     call SetRCP    ; switch to RCP
247
248     movf RxHold,W
249     xorlw d'132'
250     btfsc STATUS,Z
251     call CalRelayOn    ; Turn on the Cal Relay
252
253     movf RxHold,W
254     xorlw d'133'
255     btfsc STATUS,Z
256     call CalRelayOff    ; Turn off cal relay
257
258     movf RxHold,W
259     xorlw d'140'
260     btfsc STATUS,Z
261     call Set0DB    ; zero dB
262
263     movf RxHold,W
264     xorlw d'141'
265     btfsc STATUS,Z
266     call Set3DB    ; 3 dB
267
268     movf RxHold,W
269     xorlw d'142'
270     btfsc STATUS,Z
271     call Set6DB    ; 6 dB
272
273     movf RxHold,W
```

```
274     xorlw d'143'
275     btfsc STATUS,Z
276     call Set9DB      ; 9 dB
277
278     movf RxHold,W
279     xorlw d'144'
280     btfsc STATUS,Z
281     call Set12DB     ; 12 dB
282
283     movf RxHold,W
284     xorlw d'145'
285     btfsc STATUS,Z
286     call Set15DB     ; 15 dB
287
288     movf RxHold,W
289     xorlw d'146'
290     btfsc STATUS,Z
291     call Set18DB     ; 18 dB
292
293     movf RxHold,W
294     xorlw d'147'
295     btfsc STATUS,Z
296     call Set21DB     ; 21 dB
297
298     movf RxHold,W
299     xorlw d'148'
300     btfsc STATUS,Z
301     call Set24DB     ; 24 dB
302
303     movf RxHold,W
304     xorlw d'149'
305     btfsc STATUS,Z
306     call Set27DB     ; 27 dB
307
308
309     movf RxHold,W
310     xorlw d'150'
311     btfsc STATUS,Z
312     call Set64DB     ; 64 dB
313
314     movf RxHold,W
315     xorlw d'160'
316     btfsc STATUS,Z
317     call DoC2HCal    ; do the cold to hot Sequence
318
319     movf RxHold,W
```

```
320     xorlw d'161'
321     btfsc STATUS,Z
322     call DoH2CCal    ; do the hot to cold Sequence
323
324
325     movf RxHold,W
326     xorlw d'176'
327     btfsc STATUS,Z
328     call SaveSettings    ; Save Current Settings DACL and DACH
329
330     movf RxHold,W
331     xorlw d'172'
332     btfsc STATUS,Z
333     call RcvDelays    ; receive CalDelay setting
334
335
336     btfsc RCSTA,OERR
337     call ORUNerror    ; overrun error
338
339     return
340
341
342     ;
343     ; Send Status to PC - Port A the Port B
344     ;
345
346 SendStatus movf PORTA,W    ; put port A value in W
347     movwf TXREG    ; send it
348     call TransWt    ; wait for send to finish
349     movf PORTB,W    ; PORT B value
350     movwf TXREG    ; send it
351     call TransWt    ; wait for send to finish
352     movlw 0xFF    ; end of status marker
353     movwf TXREG    ; send it
354     call TransWt    ; wait for send to finish
355     return
356
357     ;
358     ; Test R/LCP switch
359     ;
360 PToggle btfsc PORTB, KCAL ; is there a Cal in progress?
361     return    ; yes so don't mess with the antenna relays
362     btfss PORTA, RLBTN ; return if button isn't pushed
363     return
364     call Delay25
365     btfss PORTA, RLBTN ; it must be pushed for at least this amount
```

```
    of time
366     return      ; it wasn't so return
367 PXWait call Delay25 ; now wait for it to be released
368     call SendStatus
369     btfsc PORTA, RLBTN ; released yet?
370     goto PXWait ; no continue to wait
371     call Delay25
372     btfsc PORTA, RLBTN ; is it still released or was this just a
    bounce?
373     goto PXWait ; it was a bounce start waiting again
374     call SendStatus
375     btfsc PORTA, KLCP ; finally, is klcp high?
376     goto DoRCP ; yep lets do rcp
377     call SetLCP ; no klcp is low so lets make it high
378     return
379 DoRCP call SetRCP
380     return
381
382
383
384
385 ;
386 ; Test Manual Cal switch
387 ;
388 ManCal btfsc PORTB, KCAL ; is there a Cal in progress?
389     return      ; yes so don't try it now
390     btfss PORTA, CALBTN ; return if button isn't pushed
391     return
392     call Delay25
393     btfss PORTA, CALBTN ; it must be pushed for at least this amount
    of time
394     return      ; it wasn't so return
395 CXWait call Delay25 ; now wait for it to be released
396     call SendStatus
397     btfsc PORTA, CALBTN ; released yet?
398     goto CXWait ; no continue to wait
399     call Delay25
400     btfsc PORTA, CALBTN ; is it still released or was this just a
    bounce?
401     goto CXWait ; it was a bounce start waiting again
402     call SendStatus
403     Call DoH2CCal ; done bouncing so lets do a cal
404     return
405
406
407 ;
```

```
408 ; Set Attenuator to xdB
409 ;
410
411 Set0DB bcf PORTA, KRCP
412 bcf PORTA, KLCP
413 call Delay25
414 movlw d'128'
415 movwf PORTB
416 call SendStatus
417 return
418
419 Set3DB bcf PORTA, KRCP
420 bcf PORTA, KLCP
421 call Delay25
422 movlw d'131'
423 movwf PORTB
424 call SendStatus
425 return
426
427 Set6DB bcf PORTA, KRCP
428 bcf PORTA, KLCP
429 call Delay25
430 movlw d'134'
431 movwf PORTB
432 call SendStatus
433 return
434
435 Set9DB bcf PORTA, KRCP
436 bcf PORTA, KLCP
437 call Delay25
438 movlw d'137'
439 movwf PORTB
440 call SendStatus
441 return
442
443 Set12DB bcf PORTA, KRCP
444 bcf PORTA, KLCP
445 call Delay25
446 movlw d'140'
447 movwf PORTB
448 call SendStatus
449 return
450
451 Set15DB bcf PORTA, KRCP
452 bcf PORTA, KLCP
453 call Delay25
```



```
454     movlw d'143'
455     movwf PORTB
456     call SendStatus
457     return
458
459 Set18DB bcf PORTA, KRCP
460     bcf PORTA, KLCP
461     call Delay25
462     movlw d'146'
463     movwf PORTB
464     call SendStatus
465     return
466
467 Set21DB bcf PORTA, KRCP
468     bcf PORTA, KLCP
469     call Delay25
470     movlw d'149'
471     movwf PORTB
472     call SendStatus
473     return
474
475 Set24DB bcf PORTA, KRCP
476     bcf PORTA, KLCP
477     call Delay25
478     movlw d'152'
479     movwf PORTB
480     call SendStatus
481     return
482
483 Set27DB bcf PORTA, KRCP
484     bcf PORTA, KLCP
485     call Delay25
486     movlw d'155'
487     movwf PORTB
488     call SendStatus
489     return
490
491 Set30DB bcf PORTA, KRCP
492     bcf PORTA, KLCP
493     call Delay25
494     movlw d'158'
495     movwf PORTB
496     call SendStatus
497     return
498
499 Set33DB bcf PORTA, KRCP
```

```
500     bcf PORTA, KLCP
501     call Delay25
502     movlw d'161'
503     movwf PORTB
504     call SendStatus
505     return
506
507 Set36DB bcf PORTA, KRCP
508     bcf PORTA, KLCP
509     call Delay25
510     movlw d'164'
511     movwf PORTB
512     call SendStatus
513     return
514
515 Set39DB bcf PORTA, KRCP
516     bcf PORTA, KLCP
517     call Delay25
518     movlw d'167'
519     movwf PORTB
520     call SendStatus
521     return
522
523 Set42DB bcf PORTA, KRCP
524     bcf PORTA, KLCP
525     call Delay25
526     movlw d'170'
527     movwf PORTB
528     call SendStatus
529     return
530
531 Set45DB bcf PORTA, KRCP
532     bcf PORTA, KLCP
533     call Delay25
534     movlw d'173'
535     movwf PORTB
536     call SendStatus
537     return
538
539 Set48DB bcf PORTA, KRCP
540     bcf PORTA, KLCP
541     call Delay25
542     movlw d'176'
543     movwf PORTB
544     call SendStatus
545     return
```

```
546
547 Set51DB bcf PORTA, KRCP
548     bcf PORTA, KLCP
549     call Delay25
550     movlw d'179'
551     movwf PORTB
552     call SendStatus
553     return
554
555 Set64DB bcf PORTA, KRCP
556     bcf PORTA, KLCP
557     call Delay25
558     movlw d'192'
559     movwf PORTB
560     call SendStatus
561     return
562
563
564 ;
565 ; Go To RCP
566 ;
567
568 SetRCP  bcf PORTA, KLCP    ;turn off K1
569     call Delay25
570     bsf PORTA, KRCP    ;turn on K2
571     movlw 0x01    ; save status in RAM 1 = RCP
572     movwf RLSTAT    ; RLSTAT = 1 when RCP is used
573     call SaveSettings
574     call SendStatus
575     return
576
577 ;
578 ; Go To LCP
579 ;
580
581 SetLCP  bcf PORTA, KRCP    ;turn off K2
582     call Delay25
583     bsf PORTA, KLCP    ;turn on K1
584     movlw 0x00    ; save status in RAM 0 = LCP
585     movwf RLSTAT    ; RLSTAT = 0 when LCP is used
586     call SaveSettings
587     call SendStatus
588     return
589
590 ;
591 ; Resets the LCP RCP Relays based on value in RLSTAT
```

```
592 ;
593
594 RecallCP bcf PORTA, KRCP ;turn off K2 - we want them both off
temporarily
595     bcf PORTA, KLCP ;turn off K1
596     call Delay25 ;give relays a chance to disengage
597 ; movf RLSTAT, W ;get the value in that relay status register
598     btfsc RLSTAT,0 ;
599     goto ITRSCP ; it wasn't zero so goto RCP code
600     call SetLCP ; it was zero so do LCP
601     call Delay25
602     return
603 ITRSCP call SetRCP
604     call Delay25
605     return
606 ;
607 ; Turn on/off Cal Relay
608 ;
609
610 CalRelayOn bsf PORTB, KCAL
611     call SendStatus
612     return
613
614 CalRelayOff bcf PORTB, KCAL
615     call SendStatus
616     return
617
618
619 ;
620 ; Cold to Hot Cal
621 ;
622 DoC2HCal call Set64DB
623     call Delay25
624     call CalRelayOn
625     call CalDelay
626     call Set48DB
627     call CalDelay
628     call Set45DB
629     call CalDelay
630     call Set42DB
631     call CalDelay
632     call Set39DB
633     call CalDelay
634     call Set36DB
635     call CalDelay
636     call Set33DB
```

```
637     call CalDelay
638     call Set30DB
639     call CalDelay
640     call Set27DB
641     call CalDelay
642     call Set24DB
643     call CalDelay
644     call Set21DB
645     call CalDelay
646     call Set18DB
647     call CalDelay
648     call Set15DB
649     call CalDelay
650     call Set12DB
651     call CalDelay
652     call Set9DB
653     call CalDelay
654     call Set6DB
655     call CalDelay
656     call Set3DB
657     call CalDelay
658     call Set0DB
659     call CalDelay
660     call Set64DB
661     call CalDelay
662     call CalRelayOff      ; turn off the relay
663     call Delay25         ; settling time
664     call RecallCP       ; restore RCP/LCP relays to former state
665     return
666
667 ;
668 ; Hot to Cold Cal
669 ;
670 DoH2CCal  call Set64DB
671           call Delay25
672           call CalRelayOn
673           call CalDelay
674           call Set0DB
675           call CalDelay
676           call Set3DB
677           call CalDelay
678           call Set6DB
679           call CalDelay
680           call Set9DB
681           call CalDelay
682           call Set12DB
```

```
683     call CalDelay
684     call Set15DB
685     call CalDelay
686     call Set18DB
687     call CalDelay
688     call Set21DB
689     call CalDelay
690     call Set24DB
691     call CalDelay
692     call Set27DB
693     call CalDelay
694     call Set30DB
695     call CalDelay
696     call Set33DB
697     call CalDelay
698     call Set36DB
699     call CalDelay
700     call Set39DB
701     call CalDelay
702     call Set42DB
703     call CalDelay
704     call Set45DB
705     call CalDelay
706     call Set48DB
707     call CalDelay
708     call Set64DB
709     call CalDelay
710     call CalRelayOff      ; turn off cal relay
711     call Delay25         ; settling time
712     call RecallCP       ; restore RCP/LCP relays to former state
713     return
714
715
716
717
718
719     ;
720     ; Wait for Delay Store Value to be received
721     ;
722
723 RcvDelays call Receive      ; wait for byte from COM
724     movf RxHold,W          ; DelayStore in W
725     movwf DelayStore      ; put it in DelayStore
726     clrf RxHold           ; clear the RxHold reg so it doesn't cause false cmd
727     call SaveSettings
728     return
```

```
729
730
731 ;
732 ; -----
733 ; 25mS DELAY @ 18.432 MHz
734 ; -----
735 ;
736 Delay25      ;115193 cycles
737     movlw 0xFE
738     movwf DelayH
739     movlw 0x5A
740     movwf DelayL
741 Delay25_0
742     decfsz DelayH, f
743     goto  $+2
744     decfsz DelayL, f
745     goto  Delay25_0
746     goto  $+1 ;3 cycles
747     nop
748     return   ;4 cycles (including call)
749
750
751 ;
752 ; -----
753 ; 1 sec DELAY @ 18.432 MHz
754 ; -----
755 ;
756 Delay1sec    ;4607993 cycles
757     movlw 0x6C
758     movwf d1
759     movlw 0x0C
760     movwf d2
761     movlw 0x0B
762     movwf d3
763 Delay1sec_0
764     decfsz d1, f
765     goto  $+2
766     decfsz d2, f
767     goto  $+2
768     decfsz d3, f
769     goto  Delay1sec_0
770     goto  $+1 ;3 cycles
771     nop
772     return   ;4 cycles (including call)
773
774
```

```
775 ;
776 ; -----
777 ; Calibration step delay
778 ; -----
779 ;
780 CalDelay movf DelayStore,W ; get the value in the delay
781     btfsc STATUS,Z      ; if DelayStore was zero then use the default
of 5
782     movlw d'5'         ; for approx 5 sec delay for cal steps
783     movwf CalDLY
784 MoreCAL call Delay1sec
785     decfsz CalDLY,F
786     goto MoreCAL
787     return
788
789
790 ;
791 ; OVERRUN ERROR OCCURRED
792 ;
793 ORUNerror bcf RCSTA,CREN ; disable receiver - clears OERR
794     movf RCREG,W      ; flush receive buffer
795     movf RCREG,W
796     movf RCREG,W
797     bsf RCSTA,CREN ; re-enable receiver
798     return
799
800
801 ;
802 ; -----
803 ; RECEIVE BYTE
804 ; -----
805 ; This routine does not return until a byte is received.
806
807 Receive btfss PIR1,RCIF ; check for received data
808     goto Receive
809
810     movf RCREG,W      ; also clears RCIF
811     movwf RxHold     ; save data
812     return
813 ;
814 ; Receive Data if it is there but don't wait for it
815 ;
816
817 RcvNoWait clrf RxHold ; clear the rcv holding buf
818     btfss PIR1,RCIF ; is there a byte?
819     return          ; No so just return
```



```
820     movf RCREG,W      ; Yes so get it also clears RCIF
821     movwf RxHold     ; save data /no action taken if it was a 0.
822         return
823
824
825     ;
826     ; -----
827     ; WAIT UNTIL UART IS FINISHED SENDING
828     ; -----
829     ;
830 TransWt bsf STATUS,RP0
831 WtHere  btfss TXSTA,TRMT ; transmission is complete if hi
832         goto WtHere
833
834     bcf STATUS,RP0     ; RAM Page 0
835     return
836
837
838
839     ;
840     ; SUBROUTINE: READ FROM 16F873 EEPROM
841     ; On entry, W = address to read from
842     ; On exit, W = data that was read
843     ;
844 EepRead bcf STATUS,RP0 ; RAM Page 2
845         bsf STATUS,RP1
846         movwf EEADR
847         bsf STATUS,RP0 ; RAM Page 3
848         bcf EECON1,EEPGD ; data EEPROM
849         bsf EECON1,RD ; read the data
850         bcf STATUS,RP0 ; RAM Page 2
851         movf EEDATA,W
852         bcf STATUS,RP1 ; RAM Page 0
853         return
854     ;
855     ; SUBROUTINE: WRITE TO 16F873 EEPROM
856     ; On entry, W = data to be written
857     ; and EEADR is already set
858     ;
859 EepWrite bsf STATUS,RP1 ; RAM Page 2
860         bcf STATUS,RP0
861         movwf EEDATA
862         bsf STATUS,RP0 ; RAM Page 3
863         bcf EECON1,EEPGD ; data EEPROM
864         bsf EECON1,WREN ; enable EEPROM writes
865         movlw 0x55 ; begin writing
```

```
866     movwf EECON2
867     movlw 0xAA
868     movwf EECON2
869     bsf EECON1,WR
870 WriteWait  btfsc EECON1,WR    ; wait for write completion
871     goto WriteWait
872     bcf STATUS,RP1    ; RAM Page 0
873     bcf STATUS,RP0
874     return
875 ;
876 ; get Last Saved Settings from EEPROM
877 ;
878 GetSettings clrw        ; set start address
879     call EepRead
880     movwf RLSTAT
881     movlw 0x01
882     call EepRead
883     movwf DelayStore
884     return
885
886 ;
887 ; Save Settings RLSTAT, and DelayStore to EEPROM
888 ;
889
890 SaveSettings  bsf STATUS,RP1    ; RAM Page 2
891     clrf EEADR    ; set for first address
892     bcf STATUS,RP1    ; RAM Page 0
893     bcf STATUS,RP0
894     movf RLSTAT,W
895     call EepWrite
896     bsf STATUS,RP1    ; RAM Page 2
897     incf EEADR    ; set for next address
898     bcf STATUS,RP1    ; RAM Page 0
899     bcf STATUS,RP0
900     movf DelayStore,W
901     call EepWrite
902     return
903
904 ;
905 ; -----
906 ; NULL TERMINATED STRINGS LIST
907 ; -----
908 ;
909 MessList  movf MessPt,W
910     addwf PCL,F
911 MessNum  DT "Message #", 0x00
```

```
912 MessErr DT "Oops!!", 0x00
913
914     end
915
916
917
918
919
920
921
```

```
1 ' FS-200B Calibrator Controller v1.3.0
2 ' Dave Typinski, June 2017
3 ' works with modified UFRO automatic calibrator and PIC firmware version
  AJ4COCALv02.asm
4
5 Imports System
6 Imports System.ComponentModel
7 Imports System.Threading
8 Imports System.IO
9 Imports System.Text
10
11 Public Class Form1
12
13     Dim myPort As Array 'COM Ports detected on the system are stored here
14
15     Private TargetDT As DateTime 'The next scheduled firing of the step cal
  routine (full UTC date & time)
16     Private Shared CountdownFrom As TimeSpan 'Countdown from this date/time
17     Private Shared Hrs As Integer 'hours component of schedule start
18     Private Shared Mins As Integer 'minutes component of schedule start
19     Private Shared RepeatEveryHrs As Double 'number of hours to wait between
  start of each calibration run
20     Private Shared TransitHrs As Double 'number of hours before and after transit
  to trigger a calibration run
21     Private Shared TransitTiming As Boolean 'True if using transit timing,
  otherwise we are using repeat hours timing
22     Private Shared T0 As Double = 290 'T_nought
23     Private Shared Tgen As Double 'Noise generator output temperature
24     Private Shared Tant As Double 'Equivalent antenna temperature
25     Private Shared Latt As Double 'Attenuator loss in dB
26     Private Shared Lattinsloss As Double 'Attenuator insertion loss in dB
27     Private Shared Lcoax As Double 'Feed line loss in dB
28     Private Shared NF As Double 'Receiver noise figure in dB, referenced to
  calibration plane
29     Private Shared Tobs As Double 'Observed noise temperature (i.e., on front
  panel of receiver)
30     Private Shared Tcalplane As Double 'Noise temperature at the calibration plane
31     Private Shared Lsplitterinsloss As Double 'Splitter insertion loss
32     Private Shared attValue As Byte = 0 'current status of the digital step att
  cells
33     Private Shared attMeasured As Double = 0 'measured attenuation less insertion
  loss (which is added back in)
34     Private Shared UseLogFile As Boolean 'indicates whether the log file should be
  updated or not
35     Private Shared LogFilePath As String 'the path to the log file
36     Private Shared LogFileName 'the log file name
37     Private Shared TransitsFilePath As String 'the path to the transits file
  INCLUDING the file name
38     Private Shared LogTime As DateTime 'the timestamp for the log file entries
39     Private Shared LogOutList(11) As String 'the array of strings to be written
  to the log file
40     Private Shared LogHeaderLine As String = "Date,Timestamp,Cal Ry State,Att
  Setting (dB),T_ant,T_calplane,T_observed,Tgen (MK),Att Ins Loss (dB),Splitter Ins
  Loss (dB),Attn Total (dB),Feed Line Loss (dB),Noise Figure (dB)"
```

```
41
42     Private Const K As String = " K"
43     Private Const kK As String = " kK"
44     Private Const MK As String = " MK"
45     Private Const GK As String = " GK"
46
47     Private Sub Form1_Load(sender As System.Object, e As System.EventArgs) Handles
MyBase.Load
48
49         ' line to stop trhreading errors when trying to update color of buttons
from the serial port read data thread
50         System.Windows.Forms.Control.CheckForIllegalCrossThreadCalls = False
51
52         'Get all com ports available
53         myPort = IO.Ports.SerialPort.GetPortNames()
54         For i = 0 To UBound(myPort)
55             cmbPort.Items.Add(myPort(i))
56         Next
57         'Set cmbPort text to the last COM port detected
58         cmbPort.Text = cmbPort.Items.Item(UBound(myPort))
59
60         'disable most of the UI controls until the claibrator is "connected" (by
opening the COM port)
61         lblTimeUntilLabel.Enabled = False
62         lblTimeUntilNextCal.Enabled = False
63         grpboxAuto.Enabled = False
64         grpboxManual.Enabled = False
65         grpboxTemps.Enabled = False
66         grpboxFirmware.Enabled = False
67
68         'fill in the user stored settings
69         txtNoiseGenTemp.Text = CStr(My.Settings.UserNoiseGenTemp)
70         txtSplitterInsertionLoss.Text = CStr(My.Settings.UserSplitterInsertionLoss)
71         txtNoiseFigure.Text = CStr(My.Settings.UserNoiseFigure)
72         txtFeedLineLoss.Text = CStr(My.Settings.UserFeedLoss)
73         chkboxSplitter.Checked = My.Settings.UserUseSplitter
74         txtStepAttInsLoss.Text = CStr(My.Settings.UserStepAttInsLoss)
75         txtStepDwellTime.Text = My.Settings.UserStepDwell
76         UseLogFile = My.Settings.UserUseLogFile
77         chkboxUseLogFile.Checked = UseLogFile
78         btnLogFilePath.Enabled = UseLogFile
79         LogFilePath = My.Settings.UserLogFilePath
80         cmbPort.Text = My.Settings.UserComPortName
81         TransitsFilePath = My.Settings.UserTransitsFilePath
82         txtRepeatHours.Text = My.Settings.UserRepeatHours
83         txtCalTime.Text = My.Settings.UserCalTime
84         txtTransitHours.Text = My.Settings.UserTransitHours
85         txtStepDwellTime.Text = My.Settings.UserDwellTime
86
87     End Sub
88
89     'Detect up-arrow and down-arrow key strokes and process as shortcuts for the
+ and - buttons
90     Protected Overrides Function ProcessCmdKey(ByRef msg As Message, ByVal keyData
```

```
    As Keys) As Boolean
91     If keyData = Keys.Up Then
92         btnAttPlus1.PerformClick()
93         Return True
94     ElseIf keyData = Keys.Down Then
95         btnAttMinus1.PerformClick()
96         Return True
97     End If
98     Return MyBase.ProcessCmdKey(msg, keyData)
99 End Function
100
101     'NOTE: the following works fine to turn "normal" keyboard keys into
shortcuts, but does NOT work for the arrow keys
102     'This routine needs "Me.KeyPreview = True" in the form's Load event handler
103     'Private Sub Form1_KeyDown(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles Me.KeyDown
104     '     If e.KeyCode = Keys.Up Then
105     '         btnAttPlus1.PerformClick()
106     '         e.Handled = True
107     '     ElseIf e.KeyCode = Keys.Down Then
108     '         btnAttMinus1.PerformClick()
109     '         e.Handled = True
110     '     End If
111     '
112     '     'Reference code for handling Alt+key combinations
113     '     'If e.Modifiers = Keys.Alt AndAlso e.KeyCode = Keys.X Then
114     '         btnAttPlus1.PerformClick()
115     '     'End If
116     'End Sub
117
118     'Save all the user's settings to the app's XML settings file ("user.config"
in the user's local appdata folder tree)
119     Private Sub Form1_Closing(sender As Object, e As
System.ComponentModel.CancelEventArgs) Handles MyBase.Closing
120         My.Settings.UserNoiseGenTemp = CDb1(txtNoiseGenTemp.Text)
121         My.Settings.UserSplitterInsertionLoss = CDb1(txtSplitterInsertionLoss.Text)
122         My.Settings.UserNoiseFigure = CDb1(txtNoiseFigure.Text)
123         My.Settings.UserFeedLoss = CDb1(txtFeedLineLoss.Text)
124         My.Settings.UserUseSplitter = chkboxSplitter.Checked
125         My.Settings.UserStepAttInsLoss = CDb1(txtStepAttInsLoss.Text)
126         My.Settings.UserStepDwell = txtStepDwellTime.Text
127         My.Settings.UserUseLogFile = UseLogFile
128         My.Settings.UserLogFilePath = LogFilePath
129         My.Settings.UserComPortName = cmbPort.Text
130         My.Settings.UserTransitsFilePath = TransitsFilePath
131         My.Settings.UserRepeatHours = txtRepeatHours.Text
132         My.Settings.UserCalTime = txtCalTime.Text
133         My.Settings.UserTransitHours = txtTransitHours.Text
134         My.Settings.UserDwellTime = txtStepDwellTime.Text
135
136     End Sub
137
138     Private Sub btnConnect_Click(sender As System.Object, e As System.EventArgs)
Handles btnConnect.Click
```

```
139
140     If btnConnect.BackColor = Color.LightGreen Then 'if green, then we're not
connected
141
142     SerialPort1.PortName = cmbPort.Text 'Set SerialPort1 to the selected
COM port
143     SerialPort1.BaudRate = 57600 'Set Serial Port Properties
144     SerialPort1.Parity = IO.Ports.Parity.None
145     SerialPort1.StopBits = IO.Ports.StopBits.One
146     SerialPort1.DataBits = 8
147     SerialPort1.Open() 'Open the serial port
148
149     Dim status() As Byte = {83} 'command to make the PIC return its
ports' status
150     SerialPort1.Write(status, 0, 1)
151
152     Threading.Thread.Sleep(100) 'pause for 100 millisecc to allow PIC to
send status word
153     If btnCalRy.BackColor = SystemColors.Control Then
154         SerialPort1.Close()
155         MsgBox("Cannot communicate with automatic calibrator hardware." &
vbCrLf & "Please check the serial port number and try again.", vbExclamation +
vbOKOnly, "No Hardware Found")
156         Exit Sub
157     End If
158
159     'change the Connect button text and color
160     btnConnect.BackColor = Color.Pink
161     btnConnect.Text = "Disconnect from Calibrator"
162
163     'disable port selection combo box
164     cmbPort.Enabled = False
165
166     'enable to UI controls now that the calibrator is "connected"
167     grpboxAuto.Enabled = True
168     grpboxManual.Enabled = True
169     grpboxTemps.Enabled = True
170     grpboxFirmware.Enabled = True
171
172     Else
173
174     SerialPort1.Close() 'Close the Serial Port
175
176     btnConnect.BackColor = Color.LightGreen
177     btnConnect.Text = "Connect to Calibrator"
178
179     cmbPort.Enabled = True 're-enable the COM port selection combo box
180
181     'disable the timer, reset the countdown display, and re-enable the
timing entry text boxes
182     Timer1.Enabled = False
183     lblTimeUntilNextCal.Text = "00:00:00"
184     btnTimer.BackColor = Color.LightGreen
185     btnTimer.Text = "Start Timer"
```

```
186         lblTimeUntilLabel.Enabled = False
187         lblTimeUntilNextCal.Enabled = False
188         txtRepeatHours.Enabled = True
189         txtCalTime.Enabled = True
190
191         'set all the cal status button colors back to the OFF state
192         btnCalRy.BackColor = SystemColors.Control
193         btnCalRy.UseVisualStyleBackColor = True
194         btn1dBAtt.BackColor = SystemColors.Control
195         btn1dBAtt.UseVisualStyleBackColor = True
196         btn2dBAtt.BackColor = SystemColors.Control
197         btn2dBAtt.UseVisualStyleBackColor = True
198         btn4dBAtt.BackColor = SystemColors.Control
199         btn4dBAtt.UseVisualStyleBackColor = True
200         btn8dBAtt.BackColor = SystemColors.Control
201         btn8dBAtt.UseVisualStyleBackColor = True
202         btn16dBAtt.BackColor = SystemColors.Control
203         btn16dBAtt.UseVisualStyleBackColor = True
204         btn32dBAtt.BackColor = SystemColors.Control
205         btn32dBAtt.UseVisualStyleBackColor = True
206         btn64dBAtt.BackColor = SystemColors.Control
207         btn64dBAtt.UseVisualStyleBackColor = True
208
209         'finally, disable the calibrator's controls and tempertaure displays
210         grpbxAuto.Enabled = False
211         grpbxManual.Enabled = False
212         grpbxTemps.Enabled = False
213         lblTempatCalPlane.Text = "-----"
214         lblObserved.Text = "-----"
215         lblAntTemp.Text = "-----"
216         btnCalRy.Text = "----"
217
218     End If
219
220 End Sub
221
222 Private Sub SerialPort1_DataReceived(ByVal sender As Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
223
224     Threading.Thread.Sleep(5) 'pause thread for 5 millisec to ensure buffer
is full before processing (still doesn't always work)
225
226     Dim rx As Integer
227     rx = SerialPort1.BytesToRead
228     Dim comBuff As Byte() = New Byte(rx - 1) {}
229     SerialPort1.Read(comBuff, 0, rx)
230
231     ReceivedBytes(comBuff, rx) 'Call ReceivedBytes() every time data is
received at the serialPort
232
233 End Sub
234
235 Private Sub ReceivedBytes(ByVal inBytes As Byte(), rx As Integer)
236
```



```
237         If rx = 3 Then 'check the number of bytes first; sometimes the
DataReceived event triggers with less than 3 bytes received whereupon the next
check (below) will throw an index out of range error
238
239         If CInt(inBytes(2)) = 255 Then 'should always receive three status
bytes with last byte 0xFF; if not, ignore the mal-formed status word and wait for
next status msg to be received.
240
241         LogTime = DateTime.UtcNow
242
243         Dim portAStatus As Integer = CInt(inBytes(0))
244         Dim portBStatus As Integer = CInt(inBytes(1))
245         attValue = 0
246
247         If (portAStatus And 1) Or (portAStatus And 8) Then
248             btnCalRy.BackColor = Color.Cyan
249             btnCalRy.Text = "ANT"
250         Else
251             btnCalRy.BackColor = SystemColors.Control
252             btnCalRy.UseVisualStyleBackColor = True
253             btnCalRy.Text = "-----"
254         End If
255
256         If portBStatus And 1 Then
257             btn1dBAtt.BackColor = Color.Cyan
258             attValue += 1
259         Else
260             btn1dBAtt.BackColor = SystemColors.Control
261             btn1dBAtt.UseVisualStyleBackColor = True
262         End If
263
264         If portBStatus And 2 Then
265             btn2dBAtt.BackColor = Color.Cyan
266             attValue += 2
267         Else
268             btn2dBAtt.BackColor = SystemColors.Control
269             btn2dBAtt.UseVisualStyleBackColor = True
270         End If
271
272         If portBStatus And 4 Then
273             btn4dBAtt.BackColor = Color.Cyan
274             attValue += 4
275         Else
276             btn4dBAtt.BackColor = SystemColors.Control
277             btn4dBAtt.UseVisualStyleBackColor = True
278         End If
279
280         If portBStatus And 8 Then
281             btn8dBAtt.BackColor = Color.Cyan
282             attValue += 8
283         Else
284             btn8dBAtt.BackColor = SystemColors.Control
285             btn8dBAtt.UseVisualStyleBackColor = True
286         End If
```

```
287
288     If portBStatus And 16 Then
289         btn16dBAtt.BackColor = Color.Cyan
290         attValue += 16
291     Else
292         btn16dBAtt.BackColor = SystemColors.Control
293         btn16dBAtt.UseVisualStyleBackColor = True
294     End If
295
296     If portBStatus And 32 Then
297         btn32dBAtt.BackColor = Color.Cyan
298         attValue += 32
299     Else
300         btn32dBAtt.BackColor = SystemColors.Control
301         btn32dBAtt.UseVisualStyleBackColor = True
302     End If
303
304     If portBStatus And 64 Then
305         btn64dBAtt.BackColor = Color.Cyan
306         attValue += 64
307     Else
308         btn64dBAtt.BackColor = SystemColors.Control
309         btn64dBAtt.UseVisualStyleBackColor = True
310     End If
311
312     If portBStatus And 128 Then
313         btnCalRy.BackColor = Color.Yellow
314         btnCalRy.Text = "CAL" '& vbCr & CStr(attValue)
315         btnManualRun.BackColor = Color.Cyan
316         Try
317             btnManualRun.Text = CStr(attValue + CDb1(
txtStepAttInsLoss.Text)) & " dB Attenuation"
318         Catch ex As Exception
319             MsgBox("Attenuator insertion loss must be a number >= 0.",
vbExclamation + vbOKOnly, "Invalid Attenuator Insertion Loss")
320         End Try
321         If chkbxSplitter.Checked Then
322             Try
323                 Lsplitterinsloss = CDb1(txtSplitterInsertionLoss.Text)
324                 btnManualRun.Text = CStr(attValue + CDb1(
txtStepAttInsLoss.Text)) & " dB Attenuator + " & CStr(Lsplitterinsloss) & " dB
Splitter Loss"
325             Catch ex As Exception
326                 MsgBox("Splitter insertion loss must be a number >=
0.", vbExclamation + vbOKOnly, "Invalid Splitter Insertion Loss")
327             End Try
328         End If
329     Else
330         'btnCalRy.BackColor = SystemColors.Control
331         'btnCalRy.UseVisualStyleBackColor = True
332         btnManualRun.BackColor = Color.LightSkyBlue
333         btnManualRun.Text = "Run Calibration"
334     End If
335
```

```
336         Try
337             Tgen = CDb1(txtNoiseGenTemp.Text) * 1000000
338         Catch ex As Exception
339             MsgBox("Noise source temp must be a number >= 0.",
vbExclamation + vbOKOnly, "Invalid Noise Source Temperture")
340         End Try
341     Try
342         Lattinsloss = CDb1(txtStepAttInsLoss.Text)
343     Catch ex As Exception
344         MsgBox("Attenuator insertion loss must be a number >= 0.",
vbExclamation + vbOKOnly, "Invalid Attenuator Insertion Loss")
345     End Try
346     If chkbxSplitter.Checked Then
347         Try
348             Lsplitterinsloss = CDb1(txtSplitterInsertionLoss.Text)
349         Catch ex As Exception
350             MsgBox("Splitter insertion loss must be a number >= 0.",
vbExclamation + vbOKOnly, "Invalid Splitter Insertion Loss")
351         End Try
352     Else
353         Lsplitterinsloss = 0
354     End If
355     Try
356         Lcoax = CDb1(txtFeedLineLoss.Text)
357     Catch ex As Exception
358         MsgBox("Feed line loss must be a number >= 0.", vbExclamation
+ vbOKOnly, "Invalid Feed Line Loss")
359     End Try
360     Try
361         NF = CDb1(txtNoiseFigure.Text)
362     Catch ex As Exception
363         MsgBox("Noise figure must be a number >= 0.", vbExclamation +
vbOKOnly, "Invalid Noise Figure")
364     End Try
365
366     ' correction for measured attenuator losses less 0.5 dB insertion
loss
367
368     Dim corr As Double = 0
369     Select Case attValue
370     Case 0
371         corr += 0.06
372     Case 3
373         corr -= 0.07
374     Case 6
375         corr -= 0.03
376     Case 9
377         corr -= 0.05
378     Case 12
379         corr += 0.08
380     Case 15
381         corr -= 0.02
382     Case 18
383         corr += 0.05
```

```
384         Case 21
385             corr += 0.0
386         Case 24
387             corr += 0.05
388         Case 27
389             corr += 0.01
390         Case 30
391             corr += 0.08
392         Case 33
393             corr -= 0.01
394         Case 36
395             corr += 0.04
396         Case 39
397             corr += 0.05
398         Case 42
399             corr += 0.05
400         Case 45
401             corr += 0.09
402         Case 48
403             corr += 0.05
404         Case 64
405             corr += 0.15
406     End Select
407
408     attMeasured = attValue + corr
409
410     Latt = attMeasured + Lattinsloss + Lsplitterinsloss
411     Tcalplane = Tgen * 10 ^ (-Latt / 10) + T0 * (1 - 10 ^ (-Latt / 10))
412     lblTempatCalPlane.Text = FormattedTemp(Tcalplane)
413     Tobs = Tcalplane + T0 * (10 ^ (NF / 10) - 1)
414     lblObserved.Text = FormattedTemp(Tobs)
415     Tant = 10 ^ (Lcoax / 10) * (Tgen * (10 ^ (-Latt / 10)) + T0 * (1 -
10 ^ (-Latt / 10)) - T0 * (1 - 10 ^ (-Lcoax / 10)))
416     lblAntTemp.Text = FormattedTemp(Tant)
417
418     If UseLogFile Then
419         UpdateLogFile(LogTime)
420     End If
421
422     Else
423         'The following is useful for debugging, but not much point for
release version. If status message from PIC is corrupt, just wait for the next
one.
424         'MsgBox("Incomplete data received from calibrator." & vbCrLf &
CStr(rx) & " bytes received, last byte = " & CStr(inBytes(rx - 1)), vbInformation)
425
426     End If
427
428 End If
429
430 End Sub
431
432 Private Sub cmbPort_SelectedIndexChanged(sender As System.Object, e As
System.EventArgs) Handles cmbPort.SelectedIndexChanged
```

```
433
434     If SerialPort1.IsOpen = False Then
435         SerialPort1.PortName = cmbPort.Text
436     Else
437         'pop a message box to user if trying to change ports w/o
disconnecting first
438         MsgBox("CHanging the port number is Valid only if the port is closed",
vbCritical + vbOKOnly, "Prohibited Operation")
439     End If
440
441 End Sub
442
443 Private Sub btnSetStepDelay_Click(sender As System.Object, e As
System.EventArgs) Handles btnSetStepDelay.Click
444     Try
445         If CInt(txtStepDwellTime.Text) < 1 Or CInt(txtStepDwellTime.Text) >
255 Then
446             MsgBox("Cal step dwell time must be an integer between 1 and 255
seconds inclusive.", vbExclamation + vbOKOnly, "Invalid Dwell Time Value")
447         Else
448             Dim delayValue As Byte = CByte(CInt(txtStepDwellTime.Text))
449             Dim delay() As Byte = {172, delayValue} 'command to set step
dwell time (1st byte), number of 1 sec delays (2nd byte)
450             SerialPort1.Write(delay, 0, 2)
451             Dim status() As Byte = {83} 'command to make the PIC return its
ports' status
452             SerialPort1.Write(status, 0, 1)
453         End If
454     Catch ex As Exception
455         MsgBox("Cal step dwell time must be an intgeer between 1 and 255
seconds inclusive.", vbExclamation + vbOKOnly, "Invalid Delay Value")
456     End Try
457
458 End Sub
459
460 Private Sub btnManualRun_Click(sender As System.Object, e As System.EventArgs)
Handles btnManualRun.Click
461     If btnManualRun.BackColor = Color.Cyan Then
462         MsgBox("Calibration already in progress.", vbInformation + vbOKOnly,
"Calibration in Progress")
463     Else
464         btnManualRun.BackColor = Color.Cyan
465         If rbbtnCtoH.Checked Then
466             Dim start() As Byte = {160} 'command to make the PIC start
running the firmware-coded cold to hot step calibration (48 to 0 dB att in 3 dB
steps, 17 steps total)
467             SerialPort1.Write(start, 0, 1)
468         Else
469             Dim start() As Byte = {161} 'command to make the PIC start
running the firmware-coded hot to cold step calibration (0 to 48 dB att in 3 dB
steps, 17 steps total)
470             SerialPort1.Write(start, 0, 1)
471         End If
472     End If
```

```
473         'note: the PIC's cal routine sends status messages from PIC to PC for
every step automatically, no need to call for status
474
475     End Sub
476
477     'Private Sub btnRCPRy_Click(sender As System.Object, e As System.EventArgs)
478     '     If btnRCPRy.BackColor = Color.Cyan Then
479     'Dim lcp() As Byte = {130} 'command to switch to LCP
480     '     SerialPort1.Write(lcp, 0, 1)
481     '     Else
482     'Dim rcp() As Byte = {131} 'command to switch to RCP
483     '     SerialPort1.Write(rcp, 0, 1)
484     '     End If
485     'note: the PIC's relay change state routines send status messages from PIC
to PC automatically, no need to call for status
486     '
487     '     End Sub
488
489     'Private Sub btnLCPRy_Click(sender As System.Object, e As System.EventArgs)
490     '     If btnLCPRy.BackColor = Color.Cyan Then
491     'Dim rcp() As Byte = {131} 'command to switch to RCP
492     '     SerialPort1.Write(rcp, 0, 1)
493     '     Else
494     'Dim lcp() As Byte = {130} 'command to switch to LCP
495     '     SerialPort1.Write(lcp, 0, 1)
496     '     End If
497     'note: the PIC's relay change state routines send status messages from PIC
to PC automatically, no need to call for status
498     '
499     '     End Sub
500
501     Private Sub btnCalRy_Click(sender As System.Object, e As System.EventArgs)
Handles btnCalRy.Click
502         If btnCalRy.BackColor = Color.Yellow Then
503             Dim caloff() As Byte = {133} 'command to turn off Cal relay
504             SerialPort1.Write(caloff, 0, 1)
505         Else
506             Dim calon() As Byte = {132} 'command to turn on Cal relay
507             SerialPort1.Write(calon, 0, 1)
508         End If
509         'note: the PIC's relay change state routines send status messages from
PIC to PC automatically, no need to call for status
510
511     End Sub
512
513     Private Sub btnldBAtt_Click(sender As System.Object, e As System.EventArgs)
Handles btnldBAtt.Click
514         If btnldBAtt.BackColor = Color.Cyan Then
515             Dim padoff() As Byte = {20} 'command to turn off 1 dB Att relay
516             SerialPort1.Write(padoff, 0, 1)
517         Else
518             Dim padon() As Byte = {10} 'command to turn on 1 dB Att relay
519             SerialPort1.Write(padon, 0, 1)
520         End If
```

```
521     Dim status() As Byte = {83} 'command to make the PIC return its ports'
522     status
523
524     SerialPort1.Write(status, 0, 1)
525
526     End Sub
527
528     Private Sub btn2dBAtt_Click(sender As System.Object, e As System.EventArgs)
529     Handles btn2dBAtt.Click
530     If btn2dBAtt.BackColor = Color.Cyan Then
531         Dim padoff() As Byte = {21} 'command to turn off 2 dB Att relay
532         SerialPort1.Write(padoff, 0, 1)
533     Else
534         Dim padon() As Byte = {11} 'command to turn on 2 dB Att relay
535         SerialPort1.Write(padon, 0, 1)
536     End If
537     Dim status() As Byte = {83} 'command to make the PIC return its ports'
538     status
539     SerialPort1.Write(status, 0, 1)
540
541     End Sub
542
543     Private Sub btn4dBAtt_Click(sender As System.Object, e As System.EventArgs)
544     Handles btn4dBAtt.Click
545     If btn4dBAtt.BackColor = Color.Cyan Then
546         Dim padoff() As Byte = {22} 'command to turn off 4 dB Att relay
547         SerialPort1.Write(padoff, 0, 1)
548     Else
549         Dim padon() As Byte = {12} 'command to turn on 4 dB Att relay
550         SerialPort1.Write(padon, 0, 1)
551     End If
552     Dim status() As Byte = {83} 'command to make the PIC return its ports'
553     status
554     SerialPort1.Write(status, 0, 1)
555
556     End Sub
557
558     Private Sub btn8dBAtt_Click(sender As System.Object, e As System.EventArgs)
559     Handles btn8dBAtt.Click
560     If btn8dBAtt.BackColor = Color.Cyan Then
561         Dim padoff() As Byte = {23} 'command to turn off 8 dB Att relay
562         SerialPort1.Write(padoff, 0, 1)
563     Else
564         Dim padon() As Byte = {13} 'command to turn on 8 dB Att relay
565         SerialPort1.Write(padon, 0, 1)
566     End If
567     Dim status() As Byte = {83} 'command to make the PIC return its ports'
568     status
569     SerialPort1.Write(status, 0, 1)
570
571     End Sub
572
573     Private Sub btn16dBAtt_Click(sender As System.Object, e As System.EventArgs)
574     Handles btn16dBAtt.Click
575     If btn16dBAtt.BackColor = Color.Cyan Then
```

```
567         Dim padoff() As Byte = {24} 'command to turn off 16 dB Att relay
568         SerialPort1.Write(padoff, 0, 1)
569     Else
570         Dim padon() As Byte = {14} 'command to turn on 16 dB Att relay
571         SerialPort1.Write(padon, 0, 1)
572     End If
573     Dim status() As Byte = {83} 'command to make the PIC return its ports'
status
574     SerialPort1.Write(status, 0, 1)
575
576     End Sub
577
578     Private Sub btn32dBAtt_Click(sender As System.Object, e As System.EventArgs)
Handles btn32dBAtt.Click
579         If btn32dBAtt.BackColor = Color.Cyan Then
580             Dim padoff() As Byte = {25} 'command to turn off 32 dB Att relay
581             SerialPort1.Write(padoff, 0, 1)
582         Else
583             Dim padon() As Byte = {15} 'command to turn on 32 dB Att relay
584             SerialPort1.Write(padon, 0, 1)
585         End If
586         Dim status() As Byte = {83} 'command to make the PIC return its ports'
status
587         SerialPort1.Write(status, 0, 1)
588
589     End Sub
590
591     Private Sub btn64dBAtt_Click(sender As System.Object, e As System.EventArgs)
Handles btn64dBAtt.Click
592         If btn64dBAtt.BackColor = Color.Cyan Then
593             Dim padoff() As Byte = {26} 'command to turn off 64 dB Att relay
594             SerialPort1.Write(padoff, 0, 1)
595         Else
596             Dim padon() As Byte = {16} 'command to turn on 64 dB Att relay
597             SerialPort1.Write(padon, 0, 1)
598         End If
599         Dim status() As Byte = {83} 'command to make the PIC return its ports'
status
600         SerialPort1.Write(status, 0, 1)
601
602     End Sub
603
604     Private Sub btnTimer_Click(sender As System.Object, e As System.EventArgs)
Handles btnTimer.Click
605
606         If btnTimer.BackColor = Color.Pink Then 'if its pink, then the timer is
running and we want to turn it off
607
608             Timer1.Enabled = False
609             lblTimeUntilNextCal.Text = "00:00:00"
610             btnTimer.BackColor = Color.LightGreen
611             btnTimer.Text = "Engage Timer"
612             lblTimeUntilLabel.Enabled = False
613             lblTimeUntilNextCal.Enabled = False
```



```
614         txtRepeatHours.Enabled = True
615         txtCalTime.Enabled = True
616         Label20.Enabled = True
617         txtTransitHours.Enabled = True
618         Label21.Enabled = True
619         Label22.Enabled = True
620         btnTransitsFile.Enabled = True
621         btnTimerTransit.Enabled = True
622
623     Else 'we want to turn the timer on
624
625         'Validate the user input fields
626         If txtCalTime.Text.Length <> 4 Then
627             MsgBox("Cal start time must be four numeric digits of the form
HHMM", vbExclamation + vbOKOnly, "Invalid Start Time")
628             Exit Sub
629         End If
630         Try
631             Hrs = CInt(txtCalTime.Text.Substring(0, 2))
632             Mins = CInt(txtCalTime.Text.Substring(2, 2))
633         Catch ex As Exception
634             MsgBox("Cal start time must be four numeric digits of the form
HHMM", vbExclamation + vbOKOnly, "Invalid Start Time")
635             Exit Sub
636         End Try
637         If Hrs < 0 Or Hrs > 23 Then
638             MsgBox("Cal start time hours must be in the range 00 to 23",
vbExclamation + vbOKOnly, "Invalid Start Time")
639             Exit Sub
640         End If
641         If Mins < 0 Or Mins > 59 Then
642             MsgBox("Cal start time minutes must be in the range 00 to 59",
vbExclamation + vbOKOnly, "Invalid Start Time")
643             Exit Sub
644         End If
645         Try
646             'use Round method and CDec casting to handle imprecision of
binary floating point numbers (e.g., repeat of 0.1 hours)
647             If Math.Round(24 Mod CDec(txtRepeatHours.Text), 10) > 0 Then
648                 MsgBox("Cal repeat hours must be a number evenly divisible
into 24", vbExclamation + vbOKOnly, "Invalid Repeat Period")
649                 Exit Sub
650             Else
651                 RepeatEveryHrs = CDb1(CDec(txtRepeatHours.Text))
652             End If
653         Catch ex As Exception
654             MsgBox("Cal repeat hours must be a number evenly divisible into
24", vbExclamation + vbOKOnly, "Invalid Repeat Period")
655             Exit Sub
656         End Try
657
658         'Set the date/time of the next scheduled step cal
659         CountdownFrom = TimeSpan.FromHours(CDb1(Hrs)) + TimeSpan.FromMinutes(
CDb1(Mins))
```

```
660         TargetDT = DateTime.UtcNow.Date.Add(CountDownFrom)
661
662         'If the "start at" time has already passed for this day, increment
the date by one day
663         If DateTime.Compare(TargetDT, DateTime.UtcNow) < 0 Then
664             TargetDT = TargetDT.Add(TimeSpan.FromDays(1))
665         End If
666
667         'set up and run the timer component and configure the UI
668         Timer1.Interval = 100
669         Timer1.Enabled = True
670         btnTimer.BackColor = Color.Pink
671         btnTimer.Text = "Disengage Timer"
672         lblTimeUntilLabel.Enabled = True
673         lblTimeUntilNextCal.Enabled = True
674         txtRepeatHours.Enabled = False
675         txtCalTime.Enabled = False
676         Label20.Enabled = False
677         txtTransitHours.Enabled = False
678         Label21.Enabled = False
679         Label22.Enabled = False
680         btnTransitsFile.Enabled = False
681         btnTimerTransit.Enabled = False
682
683     End If
684
685 End Sub
686
687 Private Sub Timer1_Tick(sender As System.Object, e As System.EventArgs)
Handles Timer1.Tick
688     'find out how much time is left between now and the next step cal
689     Dim ts As TimeSpan = TargetDT.Subtract(DateTime.UtcNow)
690     If ts.TotalMilliseconds > 0 Then 'keep on counting down
691         lblTimeUntilNextCal.Text = ts.ToString("hh:mm:ss")
692     Else 'set the countdown timer target date/time to the time of the next
step cal after this one and fire the step cal routine
693         lblTimeUntilNextCal.Text = "00:00:00"
694
695         If rbtnCtoH.Checked Then
696             Dim start() As Byte = {160} 'command to make the PIC start
running the firmware-coded cold to hot step calibration (48 to 0 dB att in 3 dB
steps, 17 steps total)
697             SerialPort1.Write(start, 0, 1)
698         Else
699             Dim start() As Byte = {161} 'command to make the PIC start
running the firmware-coded hot to cold step calibration (0 to 48 dB att in 3 dB
steps, 17 steps total)
700             SerialPort1.Write(start, 0, 1)
701         End If
702
703     If TransitTiming = True Then
704         Try
705             Dim sr As StreamReader = New StreamReader(TransitsFilePath)
706             Dim line As String = ""
```

```

707         Dim TransitDT As DateTime
708         Do
709             line = sr.ReadLine()
710             TransitDT = DateTime.Parse(line)
711             TargetDT = TransitDT.Subtract(TimeSpan.FromHours(
TransitHrs))
712             If DateTime.Compare(TargetDT, DateTime.UtcNow) < 0 Then
713                 TargetDT = TargetDT.Add(TimeSpan.FromHours(TransitHrs
* 2))
714             End If
715             Loop Until DateTime.Compare(TargetDT, DateTime.UtcNow) > 0
716             sr.Close()
717         Catch ex As Exception
718             MessageBox.Show("Error in timer reset subroutine while
reading transits file. Original error: " & ex.Message)
719         End Try
720     Else
721         TargetDT = TargetDT.Add(TimeSpan.FromHours(RepeatEveryHrs))
722     End If
723
724 End If
725
726 End Sub
727
728 Private Sub chkboxSplitter_CheckedChanged(sender As System.Object, e As
System.EventArgs) Handles chkboxSplitter.CheckedChanged
729     If chkboxSplitter.Checked Then
730         txtSplitterInsertionLoss.Enabled = True
731     Else
732         txtSplitterInsertionLoss.Enabled = False
733     End If
734     'get new status to update the temperature displays
735     If SerialPort1.IsOpen Then
736         Dim status() As Byte = {83} 'command to make the PIC return its
ports' status
737         SerialPort1.Write(status, 0, 1)
738     End If
739
740 End Sub
741
742 'Function rounds the input value to three significant figures and adds the
appropriate SI prefix to the units abbreviation
743 Function FormattedTemp(ByVal Temp As Double) As String
744     Dim Out As String
745     Dim Neg As Boolean = False
746     If Temp < 0 Then
747         Neg = True
748         Temp = -1 * Temp
749     End If
750     If Temp >= 10 ^ 9 Then
751         Out = CStr(Math.Round(Temp / 10 ^ 9, 2)) & GK
752     ElseIf Temp < 10 ^ 9 And Temp >= 10 ^ 8 Then
753         Out = CStr(Math.Round(Temp / 10 ^ 6, 0)) & MK
754     ElseIf Temp < 10 ^ 8 And Temp >= 10 ^ 7 Then

```

```
755         Out = CStr(Math.Round(Temp / 10 ^ 6, 1)) & MK
756     ElseIf Temp < 10 ^ 7 And Temp >= 10 ^ 6 Then
757         Out = CStr(Math.Round(Temp / 10 ^ 6, 2)) & MK
758     ElseIf Temp < 10 ^ 6 And Temp >= 10 ^ 5 Then
759         Out = CStr(Math.Round(Temp / 10 ^ 3, 0)) & kK
760     ElseIf Temp < 10 ^ 5 And Temp >= 10 ^ 4 Then
761         Out = CStr(Math.Round(Temp / 10 ^ 3, 1)) & kK
762     ElseIf Temp < 10 ^ 4 And Temp >= 10 ^ 3 Then
763         Out = CStr(Math.Round(Temp / 10 ^ 3, 2)) & kK
764     ElseIf Temp < 10 ^ 3 And Temp >= 10 ^ 2 Then
765         Out = CStr(Math.Round(Temp, 0)) & K
766     ElseIf Temp < 10 ^ 2 And Temp >= 10 ^ 1 Then
767         Out = CStr(Math.Round(Temp, 1)) & K
768     Else
769         Out = CStr(Math.Round(Temp, 2)) & K
770     End If
771     If InStr(Out, " ") = 2 Then
772         If InStr(Out, ".") < 1 Then
773             Out = Out.Insert(Out.IndexOf(" "), ".00")
774         End If
775     ElseIf InStr(Out, " ") = 3 Then
776         If InStr(Out, ".") < 1 Then
777             Out = Out.Insert(Out.IndexOf(" "), ".0")
778         End If
779     End If
780     If Neg Then
781         Out = "-" & Out
782     End If
783     Return Out
784 End Function
785
786 Private Sub btnAttMinus1_Click(sender As System.Object, e As System.EventArgs)
Handles btnAttMinus1.Click
787     If attValue = 0 Then
788         Exit Sub
789     Else
790         Dim incr As Byte
791         If rbtn5dB.Checked Then
792             incr = 5
793         ElseIf rbtn3dB.Checked Then
794             incr = 3
795         Else
796             incr = 1
797         End If
798         SetAttenuator(attValue - incr)
799     End If
800 End Sub
801
802 Private Sub btnAttPlus1_Click(sender As System.Object, e As System.EventArgs)
Handles btnAttPlus1.Click
803     If attValue = 127 Then
804         Exit Sub
805     Else
806         Dim incr As Byte
```

```
807         If rbtn5dB.Checked Then
808             incr = 5
809         ElseIf rbtn3dB.Checked Then
810             incr = 3
811         Else
812             incr = 1
813         End If
814         SetAttenuator(attValue + incr)
815     End If
816 End Sub
817
818 Private Sub SetAttenuator(ByVal attSetting As Byte)
819     If attSetting And 1 Then
820         Dim padon() As Byte = {10} 'command to turn on 1 dB Att relay
821         SerialPort1.Write(padon, 0, 1)
822     Else
823         Dim padoff() As Byte = {20} 'command to turn off 1 dB Att relay
824         SerialPort1.Write(padoff, 0, 1)
825     End If
826     If attSetting And 2 Then
827         Dim padon() As Byte = {11} 'command to turn on 2 dB Att relay
828         SerialPort1.Write(padon, 0, 1)
829     Else
830         Dim padoff() As Byte = {21} 'command to turn off 2 dB Att relay
831         SerialPort1.Write(padoff, 0, 1)
832     End If
833     If attSetting And 4 Then
834         Dim padon() As Byte = {12} 'command to turn on 4 dB Att relay
835         SerialPort1.Write(padon, 0, 1)
836     Else
837         Dim padoff() As Byte = {22} 'command to turn off 4 dB Att relay
838         SerialPort1.Write(padoff, 0, 1)
839     End If
840     If attSetting And 8 Then
841         Dim padon() As Byte = {13} 'command to turn on 8 dB Att relay
842         SerialPort1.Write(padon, 0, 1)
843     Else
844         Dim padoff() As Byte = {23} 'command to turn off 8 dB Att relay
845         SerialPort1.Write(padoff, 0, 1)
846     End If
847     If attSetting And 16 Then
848         Dim padon() As Byte = {14} 'command to turn on 16 dB Att relay
849         SerialPort1.Write(padon, 0, 1)
850     Else
851         Dim padoff() As Byte = {24} 'command to turn off 16 dB Att relay
852         SerialPort1.Write(padoff, 0, 1)
853     End If
854     If attSetting And 32 Then
855         Dim padon() As Byte = {15} 'command to turn on 32 dB Att relay
856         SerialPort1.Write(padon, 0, 1)
857     Else
858         Dim padoff() As Byte = {25} 'command to turn off 32 dB Att relay
859         SerialPort1.Write(padoff, 0, 1)
860     End If
```

```
861     If attSetting And 64 Then
862         Dim padon() As Byte = {16} 'command to turn on 64 dB Att relay
863         SerialPort1.Write(padon, 0, 1)
864     Else
865         Dim padoff() As Byte = {26} 'command to turn off 64 dB Att relay
866         SerialPort1.Write(padoff, 0, 1)
867     End If
868
869     Dim status() As Byte = {83} 'command to make the PIC return its ports'
status
870     SerialPort1.Write(status, 0, 1)
871
872     End Sub
873
874     Private Sub chkboxUseLogFile_CheckedChanged(sender As System.Object, e As
System.EventArgs) Handles chkboxUseLogFile.CheckedChanged
875         UseLogFile = chkboxUseLogFile.Checked
876         btnLogFilePath.Enabled = UseLogFile
877     End Sub
878
879     Private Sub btnLogFilePath_Click(sender As System.Object, e As
System.EventArgs) Handles btnLogFilePath.Click
880         If LogFilePath = "x" Then
881             FolderBrowserDialog1.RootFolder = Environment.SpecialFolder.Desktop
882             FolderBrowserDialog1.ShowNewFolderButton = True
883             FolderBrowserDialog1.Description = "Select Log Files Storage Folder"
884             If FolderBrowserDialog1.ShowDialog() = Windows.Forms.DialogResult.OK
Then
885                 LogFilePath = FolderBrowserDialog1.SelectedPath
886             End If
887         Else
888             FolderBrowserDialog1.SelectedPath = LogFilePath
889             FolderBrowserDialog1.ShowNewFolderButton = True
890             FolderBrowserDialog1.Description = "Select Log Files Storage Folder"
891             If FolderBrowserDialog1.ShowDialog() = Windows.Forms.DialogResult.OK
Then
892                 LogFilePath = FolderBrowserDialog1.SelectedPath
893             End If
894         End If
895     End Sub
896
897     Private Sub UpdateLogFile(TimeStamp As DateTime)
898         LogFileName = "\Cal Log " & TimeStamp.ToString("yyyy MM") & ".csv"
899         Dim FilePath As String = LogFilePath & LogFileName
900
901         LogOutList(0) = TimeStamp.ToString("dd MMM yyyy, HH:mm:ss.fff")
902         LogOutList(1) = IIf(btnCalRy.BackColor = Color.Yellow, "CAL", "ANT")
903         LogOutList(2) = CStr(attMeasured)
904         LogOutList(3) = lblAntTemp.Text
905         LogOutList(4) = lblTempatCalPlane.Text
906         LogOutList(5) = lblObserved.Text
907         LogOutList(6) = txtNoiseGenTemp.Text
908         LogOutList(7) = txtStepAttInsLoss.Text
```

```
910     LogOutList(8) = txtSplitterInsertionLoss.Text
911     LogOutList(9) = CStr(Latt)
912     LogOutList(10) = txtFeedLineLoss.Text
913     LogOutList(11) = txtNoiseFigure.Text
914
915     Dim LogOutLine As String = ""
916     For x As Integer = 0 To 11
917         LogOutLine += LogOutList(x) & ", "
918     Next
919
920     Dim sw As StreamWriter
921     If File.Exists(FilePath) Then 'add a line to the file
922         sw = File.AppendText(FilePath)
923         sw.WriteLine(LogOutLine)
924         sw.Flush()
925         sw.Close()
926     Else ' Create a new log file, write the header line, and write the
current log line
927         sw = File.CreateText(FilePath)
928         sw.WriteLine(LogHeaderLine)
929         sw.WriteLine(LogOutLine)
930         sw.Flush()
931         sw.Close()
932     End If
933
934 End Sub
935
936 Private Sub btnTransitsFile_Click(sender As System.Object, e As
System.EventArgs) Handles btnTransitsFile.Click
937
938     Dim openFileDialog1 As New OpenFileDialog()
939
940     openFileDialog1.InitialDirectory = TransitsFilePath
941     openFileDialog1.Filter = "txt files (*.txt)|*.txt|All files (*.*)|*.*"
942     openFileDialog1.FilterIndex = 1
943     openFileDialog1.RestoreDirectory = False
944
945     If openFileDialog1.ShowDialog() = System.Windows.Forms.DialogResult.OK Then
946         TransitsFilePath = openFileDialog1.FileName
947     End If
948
949 End Sub
950
951 Private Sub btnTimerTransit_Click(sender As System.Object, e As
System.EventArgs) Handles btnTimerTransit.Click
952
953     Dim myStream As Stream = Nothing
954
955     If btnTimerTransit.BackColor = Color.Pink Then 'if its pink, then the
timer is running and we want to turn it off
956
957         TransitTiming = False
958         Timer1.Enabled = False
959         lblTimeUntilNextCal.Text = "00:00:00"
```

```
960         btnTimerTransit.BackColor = Color.LightGreen
961         btnTimerTransit.Text = "Engage Timer"
962         lblTimeUntilLabel.Enabled = False
963         lblTimeUntilNextCal.Enabled = False
964         txtRepeatHours.Enabled = True
965         txtCalTime.Enabled = True
966         Label3.Enabled = True
967         Label4.Enabled = True
968         Label5.Enabled = True
969         Label6.Enabled = True
970         txtTransitHours.Enabled = True
971         btnTransitsFile.Enabled = True
972         btnTimer.Enabled = True
973
974     Else 'we want to turn the timer on
975
976         'Validate the user input fields
977         Try
978             If CDb1(CDec(txtTransitHours.Text)) < 0.1 Or CDb1(CDec(
979 txtTransitHours.Text)) > 11.5 Then
980                 MsgBox("Hours from transit must be a number >= 0.1 and <=
981 11.5", vbExclamation + vbOKOnly, "Invalid Transit Offset")
982                 Exit Sub
983             Else
984                 TransitHrs = CDb1(CDec(txtTransitHours.Text))
985             End If
986         Catch ex As Exception
987             MsgBox("Hours from transit must be a number >= 0.1 and <= 11.5",
988 vbExclamation + vbOKOnly, "Invalid Transit Offset")
989             Exit Sub
990         End Try
991
992         Try
993             Dim sr As StreamReader = New StreamReader(TransitsFilePath)
994             Dim line As String = ""
995             Dim TransitDT As DateTime
996
997             Do
998                 line = sr.ReadLine()
999                 TransitDT = DateTime.Parse(line)
1000                 TargetDT = TransitDT.Subtract(TimeSpan.FromHours(TransitHrs))
1001                 If DateTime.Compare(TargetDT, DateTime.UtcNow) < 0 Then
1002                     TargetDT = TargetDT.Add(TimeSpan.FromHours(TransitHrs * 2))
1003                 End If
1004             Loop Until DateTime.Compare(TargetDT, DateTime.UtcNow) > 0
1005
1006             'MessageBox.Show(TransitDT)
1007             'MessageBox.Show(TargetDT)
1008
1009             sr.Close()
1010
1011         Catch ex As Exception
1012             MessageBox.Show("Error in transits file subroutine. Original
1013 error: " & ex.Message)
```



```
1010         End Try
1011
1012         ''set up and run the timer component and configure the UI
1013         TransitTiming = True
1014         Timer1.Interval = 100
1015         Timer1.Enabled = True
1016         btnTimerTransit.BackColor = Color.Pink
1017         btnTimerTransit.Text = "Disengage Timer"
1018         lblTimeUntilLabel.Enabled = True
1019         lblTimeUntilNextCal.Enabled = True
1020         txtRepeatHours.Enabled = False
1021         txtCalTime.Enabled = False
1022         Label13.Enabled = False
1023         Label14.Enabled = False
1024         Label15.Enabled = False
1025         Label16.Enabled = False
1026         txtTransitHours.Enabled = False
1027         btnTransitsFile.Enabled = False
1028         btnTimer.Enabled = False
1029
1030     End If
1031
1032 End Sub
1033
1034 Private Sub Label14_Click(sender As System.Object, e As System.EventArgs)
Handles Label14.Click
1035
1036 End Sub
1037 End Class
1038
```