```
'CR1000 Series Datalogger

'date: 05-18-2012
'Program author: William Cable
'this program was completely reworked from the old version "Box program
v2.cr1" that was
'converted from a CR10X program written by G. Winston.  The only thing that
is really the
'same are some of the variable names.
'date: 06-22-2012
'added controls for a multiplexer and moved all the thermistors to the
multiplexer.
'added CS616's for each chamber
'waldrop changed the settling time on the thermistors to 10000 from 250;
and scan interval from 200 to 400; 7/24/12

'There are some constants that can be adjusted to change the timing of
flushing, measurements,
'and data output.

'This program controls and acquires data from 8 autochambers to be measured
and sequentially and is written
'for the Licor 820 configured to output 2.5V at full scale for CO2
(2000ppm) and pressure (1150mbar)
'The conversion equations for CO2,pressure,and flow are:
'CO2(ppmv)=mV*2000/2500
'pressure(mbar)=mV*1150/2500
'Flow(ml/min)=mV*2000/5000

'**** Wiring ****
'CR1000
'Single Ended Channels
'SE1 - jumper to multiplexer (H1 COMM) - reference resistor to ground
'SE2 - jumper to multiplexer (H2 COMM)
'SE3 -
'SE4 -
'SE5 -
'SE6 -
'SE7 -
'SE8 -
'SE9 - to LI-820 input 9
'SE10 - to LI-820 input 10
'SE11 - to LI-820 input 7
'SE12 - to LI-820 input 8
'SE13 - to flow meter (pink)
'SE14 - to flow meter (red)
```

```
'SE15 - compressor shunt
'SE16 - compressor shunt
'Voltage Excitation
'VX1 - jumper to multiplexer (L1 COMM)
'VX2 -
'VX3 -
'Control Ports
'C1 - SDM - to AC/DC controller
'C2 - SDM - to AC/DC controller
'C3 - SDM - to AC/DC controller
'C4 - Jumper to Multiplexer (L2 COMM)
'C5 - Multiplexer Reset
'C6 - Multiplexer Clock
'C7 -
'C8 - to compressor relay

'Multiplexer AM16/32 in 4x16 mode
'COMM (wires connecting multiplexer and datalogger)
'H1 - Jumper to SE1
'L1 - Jumper to VX1
'H2 - Jumper to SE2
'L2 - Jumper to C4
'Res - Jumper to C5
'Clk - Jumper to C6
'Measurement Channels 1-8
'H1 - thermistor air
'L1 - thermistor air
'H2 - CS616 (green) signal
'L2 - CS616 (orange) enable
'Measurement Channels 9-16
'H1 - thermistor soil
'L1 - thermistor soil
'H2 - not used
'L2 - not used

'Declare Public Variables
'Public' means anybody can see with keypad
Public flushing, measuring, shrouded, dataout As Boolean
Public chambnum, chambtime
Public CO2, pressure, flow, chambT, soilT, T_ratio, T_res, batt, i, VWC
Public therm(16)
Public CS616s(8), per
Public relay(16) As Boolean
Public comp_load, comp_on_time
Public compressor As Boolean
```

```
Units CO2=ppmv
Units pressure=mbar
Units flow=ml/min
Units chambT = C
Units soilT = C
Units per = uS
Units comp_load=A/hr

'A timer will be started when the program switches to a new chamber
'This timer will count until the next chamber is started
'Timing of Flushing, Measuring and DataOutput can be set below

Const FlushTime = 8      'minutes to flush
Const MeasTime = 4       'minutes to measure for
Const DataOutTime = 1    'how long before end of the flushing time to start
outputting data
Const CompressorMax = 30 'max time in minutes to allow the compressor to be
on
Const Rf=200000  'ohms, fixed resistor for chamber thermistors
'Steinhart-Hart Fit Constants
Const C1 = 0.001203799
Const C2 = 0.0001815529
Const C3 = 0.0000002511254

'Define Data Tables
DataTable (CO2fluxC,dataout,-1)
  CardOut (0,-1)  'Send data to CF card, when card is full rewrite data (0;
1 if want to stop recording when full)
  DataInterval (0,2,Sec,10)  'data tables records data every 2 seconds
  Average (1,chambnum,FP2,False)
  Average (1,batt,FP2,False)
  Average (1,CO2,FP2,False)
  Average (1,pressure,FP2,False)
  Average (1,flow,FP2,False)
  Average (1,chambT,FP2,False)
  Average (1,soilT,FP2,False)
  Average (1,VWC,FP2,False)
  Average (1,per,FP2,False)
  Sample (1,compressor,Boolean)
  Sample (1,flushing,Boolean)
  Sample (1,measuring,Boolean)
  Sample (1,shrouded,Boolean)
EndTable

BeginProg
  'initial conditions
```

```
  chambnum = 1  'chamber number to start with
  flushing = False  'not flushing
  measuring = False  'not measuring
  dataout = False 'don't output data
  shrouded = False 'shroud not on
  compressor = True 'compressor is on
  relay()=False
  Timer(2,Sec,2)  'reset and start timer for compressor

  Scan(400,mSec,3,0)
    'Start chamber control logic
    chambtime=Timer(1,Sec,4)/60  'read the chamber timer
    If flushing=False AND measuring=False AND relay(chambnum)=False Then
'start flushing
      Timer (1,Sec,2)  'reset and start
      flushing=True
      relay(chambnum+8)=True  'connect measurement stream
    ElseIf flushing=True AND chambtime>=(FlushTime-DataOutTime) AND
dataout=False Then 'start outputting data
      dataout=True
    ElseIf flushing=True AND chambtime>=FlushTime Then 'close lid and start
measurments
      flushing=False
      measuring=True
      relay(chambnum)=True  'close lid
    ElseIf measuring=True AND chambtime>=FlushTime+MeasTime Then
'measurement period over
      measuring=False
      dataout=False
      relay(chambnum)=False  'open lid
      relay(chambnum+8)=False  'disconnect measurement stream
      chambnum=chambnum+1
      If chambnum>8 Then chambnum=1  'reset chamber counter when all 8 have
been measured
    EndIf
    SDMCD16AC(relay(),1,0)  'send relay states to relay driver

    'measurements
    VoltDiff(CO2,1,mV2500,5,True,0,250,0.8,0)  'measure [CO2] from LI-820
    VoltDiff(pressure,1,mV2500,6,True,0,250,0.46,0)  'measure pressure from
LI-820
    VoltDiff(flow,1,mV2500,7,True,0,250,0.4,0)  'Measure Flow, channel 7 is
flow
    Battery(batt)  'Measure Battery Voltage

    'measure the thermistors and CS616's on the multiplexer
```

```
    If TimeIntoInterval(0,10,Sec) Then
      PortSet(5,1)  'multiplexer on
      i = 1  'reset counter
      SubScan (0,mSec,16)
        PulsePort(6,10000)
        BrHalf(T_ratio,1,mV2500,1,1,1,2500,False,10000,250,1,0)
        T_res = Rf*((1-T_ratio)/T_ratio)  'calculate resistance
        therm(i) = 1/(C1+C2*LN(T_res)+C3*LN(T_res)^3) - 273.15  'calculate
temperature
        'measure the CS616's on 1st half of multiplexer
        If i < 9 Then
          CS616 (CS616s(i),1,2,4,1,1.0,0)
        EndIf
        i=i+1
      NextSubScan
      PortSet(5,0)  'multiplexer off
      'store just the current chambers temperatures and CS616
      chambT = therm(chambnum)
      soilT = therm(chambnum+8)
      VWC = -0.0663+CS616s(chambnum)*(-0.0063+CS616s(chambnum)*0.0007)
      per = CS616s(chambnum)
    EndIf

    CallTable CO2FluxC
  NextScan

  'Power Management
  'This will measure the power consupmtion of the compressor with the
current shunt
  SlowSequence
  Scan(10,Sec,2,0)
    VoltDiff (comp_load,1,mV250,8,True ,0,250,1.0,0)  'this probably needs
a multiplier to be Amp hours
    'increment a counter if the compressor is ON, reset to zero if it is
OFF
    If comp_load>1 Then
      comp_on_time=Timer(2,Sec,4)/60
    Else
      Timer(2,Sec,2)  'reset and start
      comp_on_time=Timer(2,Sec,4)/60
    EndIf
    'turn off compressor if CompressorMax is exceeded
    If comp_on_time > CompressorMax Then compressor=False  'turn off
compressor if max time is exceeded
    If compressor=False AND TimeIntoInterval(0,1,day) Then compressor=True
'once per day turn the compressor back on if it was turned off because it
```

```
was on too long
    PortSet(8,compressor)  'turn compressor on or off based on the above if
statements
  NextScan

EndProg
```