(1) Simple datalogger program in CRBasic for Week 3 Demo

Rosie Howard
Tim Chui
Demo: Dataloggers and A/D conversion

```
'CR1000 Series Datalogger
'Program for practicing programming for ATSC303

'Measure air temperature using Campbell Sci HC-S3-XT temperature probe
'date: 9 December 2010
'program author: Rosie Howard
edited: 15 January 2015 by Rosie Howard, 17 January 2020 by Tim Chui
```

Program details
Author/date
Version
Revisions

```
'Declare Public Variables
Public Ptemp
Public Batt_Volt
Public Air_temp_CS

Units PTemp=degC
Units Batt_volt=Volts
Units Air_temp_CS=degC
```

Declare variables and units

```
'Define Data Tables
DataTable (AirTemp,1,-1)
        DataInterval (0,20,Sec,10)
        Minimum (1,batt_volt,FP2,0,False)
        Sample (1,PTemp,FP2)
        Sample (1,Air_temp_CS,FP2)
        Average (1,Air_temp_CS,FP2,False)
EndTable
```

Define data tables
Sampling
Averaging

```
Main Program
BeginProg
        Scan (2,Sec,0,0)
        PanelTemp (PTemp,250)
        Battery (Batt_volt)

        'Enter other measurement instructions
        Generic Differential Voltage measurements for CS temperature sensor:
        VoltDiff(Air_Temp_CS,1,mV2500,1,True,0,_60Hz,0.1,-50)
                                -50 for our XT model
        'Call Output Tables
        CallTable AirTemp
        NextScan
EndProg
```

Main program
Scan rate
Measurement instructions

(2) What datalogger program is this?

```
'CR1000 Series Datalogger
'date: 5 January 2015, edited: 17 January 2020
'program author: Rosie Howard, edited by Tim Chui


'Declare Public Variables and Units
Public Batt_Volt
Units Batt_Volt=Volts

'Variables for the 2 thermocouples, 3 groups on each cr1000
Public temp_1
Public temp_2
Public temp_3
Public temp_4
Public temp_5
Public temp_6
Public volt_1
Public volt_2
Public volt_3
Public volt_4
Public volt_5
Public volt_6

'Units
Units volt_1=microVolts
Units volt_2=microVolts
Units volt_3=microVolts
Units volt_4=microVolts
Units volt_5=microVolts
Units volt_6=microVolts
Units temp_1=degC
Units temp_2=degC
Units temp_3=degC
Units temp_4=degC
Units temp_5=degC
Units temp_6=degC

'Variable for CR1000 panel temperature
Public PTemp
Units PTemp=degC

'Define Data Tables
DataTable (lab1_TCs,True,-1)    'temperature calibration
   DataInterval (0,10,Sec,0)
   Minimum (1,batt_volt,FP2,0,False)

   Average (1,temp_1,FP2,False)
   Average (1,temp_2,FP2,False)
   Average (1,temp_3,FP2,False)
   Average (1,temp_4,FP2,False)

   Average (1,temp_5,FP2,False)

   Average (1,temp_6,FP2,False)
   Average (1,volt_1,FP2,False)
   Average (1,volt_2,FP2,False)
```

```
  Average (1,volt_3,FP2,False)
  Average (1,volt_4,FP2,False)
  Average (1,volt_5,FP2,False)
  Average (1,volt_6,FP2,False)

EndTable

'Main Program
BeginProg
  Scan (2,Sec,0,0)
   'Measure panel temperature of datalogger for reference for TC
    PanelTemp (PTemp,250)
    'Default Datalogger Battery Voltage measurement Batt_Volt:
    Battery (Batt_volt)

    'Thermocouple measurements Type T
      TCDiff (temp_1,1,mV2_5C,1,TypeT,PTemp,True,0,250,1.0,0)
      TCDiff (temp_3,1,mV2_5C,3,TypeT,PTemp,True,0,250,1.0,0)
      TCDiff (temp_5,1,mV2_5C,5,TypeT,PTemp,True,0,250,1.0,0)

      'Thermocouple measurements Type E
      TCDiff (temp_2,1,mV2_5C,2,TypeE,PTemp,True,0,250,1.0,0)
      TCDiff (temp_4,1,mV2_5C,4,TypeE,PTemp,True,0,250,1.0,0)
      TCDiff (temp_6,1,mV2_5C,6,TypeE,PTemp,True,0,250,1.0,0)

    'Voltage readings
    VoltDiff (volt_1,1,mV2_5C,1,True ,0,250,1000,0) 'times by 1000 to get it in microvolts
    VoltDiff (volt_2,1,mV2_5C,2,True ,0,250,1000,0) 'times by 1000 to get it in microvolts
    VoltDiff (volt_3,1,mV2_5C,3,True ,0,250,1000,0) 'times by 1000 to get it in microvolts
    VoltDiff (volt_4,1,mV2_5C,4,True ,0,250,1000,0) 'times by 1000 to get it in microvolts
    VoltDiff (volt_5,1,mV2_5C,5,True ,0,250,1000,0) 'times by 1000 to get it in microvolts
    VoltDiff (volt_6,1,mV2_5C,6,True ,0,250,1000,0) 'times by 1000 to get it in microvolts

    'Call Data Tables and Store Data
    CallTable(lab1_TCs)

  NextScan
EndProg
```

(3) Other LoggerNet functions for different inputs:

a) Therm109

b) SDI12 Recorder
Serial digital Interface (1200 baud rate)
Function to output distance to target for sonic snow depth sensor
Datalogger requests data via serial connection from intelligent sensor with unique address

c) BrHalf4W
Function to make a 4-wire half bridge measurement

d) PRT
Function to take result of bridge measurement and calculate temperature in degC
Needs input of X = Rs/Rf (from bridge circuit)
Used to measure temperature of net radiometer body (PTD)

Use these two functions together

e) What can you do if you don't know what a function does?

(4) Analog-to-digital converter: wind vane

a) Analog input with limited range:

b) Reference quantity

c) Output expressed as digital signal with discret quanta or subdivisions

(5) Counter input: anemometer

a)
b)
c)

(6) More complex program example using above functions:

```
'CR3000 Series Datalogger
'Measurements for instruments that will be suspended over ski run:
                    'HC-S3-XT air temperature and relative humidity
                    'IRR-P IR thermometer
                    'SR50 snow depth sensor
                    'CNR1 net radiometer
                    '61205V barometer.

'date: January 23, 2008
'program author: Rosie Howard
```

```
'Declare variables for air temperature and RH measurements (from CS sensor)
Public Batt_Volt
Public Air_Temp
Public RH
Units Batt_Volt=Volts
Units Air_Temp=degC
Units RH=%

'Declare variables for IR thermometer measurement
Public PTemp
Public SBTemp           'surface body temperature of IR therm
Public TmV              'voltage output of thermistor
Public TargTemp         'target temperature; calculated by adding the sensor body temperature
                        'raised to the fourth power and the mV output multiplied by the slope (m)
                        'and the y-intercept (b), then the fourth root of this sum
Dim Tsqr1, Tsqr2, SBTempK, m, b, TargTempK
Units PTemp=degC
Units SBTemp=degC
Units TmV=mV
Units TargTemp=degC

'Declare Constants. These values are unique to individual sensors. Values here
‘are obtained from CRBasic IRR-P 1242 Custom Coefficients document received with the sensor.
'Calibration date:20-Dec-2007.
Const mC2=16433.1
Const mC1=11153600
Const mC0=1471420000
Const bC2=70373.1
Const bC1=-2577420
Const bC0=-3878910

'Declare variables for snow depth measurement
Public DT               'Distance to target
Public TCDT             'Temperature corrected distance to target
Public DBTCDT           'Difference between base (ground) and TCDT - Snow Depth

Units Batt_Volt=Volts
Units DT=Meters
Units TCDT=Meters
Units DBTCDT=Meters

'Declare variables for net radiometer
'Up sensors measure downwelling, Dn sensors measure upwelling
Public CM3Up            'SW radiation (pyranometer)
Public CM3Dn            'SW radiation
Public CG3Up            'LW radiation (pyrgeometer)
Public CG3Dn            'LW radiation
Public CNR1TC           'temperature degrees C (from pt100 in CNR1 body)
Public CNR1TK           'temperature degrees K (from pt100 in CNR1 body)
Public NetRs            'Net SW
Public NetRl            'Net LW
Public Albedo
Public UpTot
Public DnTot
Public NetTot           'Total net radiation
Public CG3UpCo          'temperature corrected LW up
Public CG3DnCo          'temperature corrected LW down
Public Sens             'sensitivity of CNR1 in microVolts/(W/metre^2)

Units CM3Up=W/metre^2
Units CM3Dn=W/metre^2
Units CG3Up=W/metre^2
Units CG3Dn=W/metre^2
Units CNR1TC=DegC
Units CNR1TK=K
Units NetRs=W/metre^2
Units NetRl=W/metre^2
Units Albedo=W/metre^2
Units UpTot=W/metre^2
Units DnTot=W/metre^2
Units NetTot=W/metre^2
Units CG3UpCo=W/metre^2
Units CG3DnCo=W/metre^2
Units Sens=microVolts/(W/metre^2)

'Declare variables for barometer
Public Air_PRESShPa
Public Air_PRESSkPa

Units Air_PRESShPa=hPa
Units Air_PRESSkPa=kPa
```

```
'Define Data Tables for air temperature and RH measurements
DataTable (AT_RH,True,-1)
        DataInterval (0,10,Sec,0)
        Average (1,Air_Temp,FP2,False)
        Sample (1,RH,FP2)
EndTable

DataTable (Batt,True,-1)
        DataInterval (0,1440,Min,10)
        Minimum (1,Batt_Volt,FP2,False,False)
EndTable

'Define Data Tables for IR thermometer measurement
DataTable (IR_Temp,1,-1)
        DataInterval (0,1,Sec,10)
        Minimum (1,batt_volt,FP2,0,False)
        Sample (1,PTemp,FP2)
        Sample (1,SBTemp,FP2)
        Sample (1,TmV,FP2)
        Sample (1,TargTemp,FP2)
```

```
End Table

'Define Data Tables for snow depth measurement
DataTable (Sn_depth,True,-1)
        DataInterval (0,5,Sec,10)
        Sample (1,DT,FP2)
        Average (1,DT,FP2,False)
        Sample (1,TCDT,FP2)
        Average (1,TCDT,FP2,False)
EndTable

'Define Data Tables for net radiometer
DataTable (Net_rad,True,-1)
        DataInterval (0,1,Sec,10)
        Average (1,CM3Up,FP2,False)
        Average (1,CM3Dn,FP2,False)
        Average (1,CG3Up,FP2,False)
        Average (1,CG3Dn,FP2,False)
        Average (1,CNR1TC,FP2,False)
        Average (1,CNR1TK,FP2,False)
        Average (1,NetRs,FP2,False)
        Average (1,NetRl,FP2,False)
        Average (1,Albedo,FP2,False)
        Average (1,UpTot,FP2,False)
        Average (1,DnTot,FP2,False)
        Average (1,NetTot,FP2,False)
        Average (1,CG3UpCo,FP2,False)
        Average (1,CG3DnCo,FP2,False)
EndTable

'Define data table for barometer
DataTable (Baro,True,-1)
        DataInterval (0,15,Min,10)
        Average (1,Air_PRESSkPa,FP2,False)
EndTable
```

```
'Main Program
BeginProg
        Scan (1,Sec,0,0)
        'Default Datalogger Battery Voltage measurement Batt_Volt:
                Battery (Batt_volt)
        'Default datalogger panel temperature measurement PTemp:
                PanelTemp (PTemp,_60Hz)

        'Instructions for air temp and RH measurements:
                'Generic Differential Voltage measurements Air_Temp:
                VoltDiff (Air_Temp,1,mV1000,10,True,0,_60Hz,0.1,-50)
                                '-50 for our XT model
                'Generic Differential Voltage measurements RH:
                VoltDiff (RH,1,mV1000,11,True,0,_60Hz,0.1,0)

        'Instructions for IR thermometer measurement
                'Instruction to measure the sensor body temperature
                        Therm109 (SBTemp,1,17,Vx1,0,_60Hz,1.0,0)
                'Instruction to measure the mV output of the thermopile
                        VoltDiff (TmV,1,AutoRange,8,True,0,_60Hz,1.0,0)
                'Calculation of m (slope) and b (intercept) coefficients for target temperature calculation
                        m = mC2*SBTemp*SBTemp + mC1*SBTemp + mC0
                        b = bC2*SBTemp*SBTemp + bC1*SBTemp + bC0
                'Target temperature calculation based on m and b coefficients
                        SBTempK = SBTemp + 273.15
                        Tsqr1 = SBTempK*SBTempK*SBTempK*SBTempK + m*TmV + b
                        Tsqr2 = SQR(Tsqr1)
                        TargTempK = SQR(Tsqr2)
                        TargTemp = TargTempK - 273.15

        'Instructions for snow depth measurement:
                SR50 Sonic Ranging Sensor (SDI-12 Output) measurements DT, TCDT, and
                ‘DBTCDT (must make temperature measurement before this since
                ‘calculation of snow depth depends on temperature:
                        SDI12Recorder (DT,5,"0","M!",1.0,0)

                'Correct distance to target (speed of sound varies with temperature)
                        TCDT=DT*SQR(((Air_Temp+273.15)/273.15))

        'Instructions for net radiometer measurements:
                'CNR1 Net Radiometer measurements CM3Up, CM3Dn, CG3Up, CG3Dn, CNR1TC,
                'CNR1TK, NetRs, NetRl, Albedo, UpTot, DnTot, NetTot, CG3UpCo, CG3DnCo:
                '1000/Sens: '1000' converts mV to microns, dividing by Sens converts microns to W/metre^2
                'Sensitivity of CNR1 for calibration, from Kipp&Zonen
                        Sens = 6.16            'microVolts/(W/metre^2)
                        VoltDiff (CM3Up,1,mV20,1,True,0,_60Hz,1000.0/Sens,0)
                                               'wire colours: high red, low blue, jump
                        VoltDiff (CM3Dn,1,mV20,2,True,0,_60Hz,1000.0/Sens,0)
                                               'wire colours: high thin white, low thin black, jump
                        VoltDiff (CG3Up,1,mV20,3,True,0,_60Hz,1000.0/Sens,0)
                                               'wire colours: high grey, low yellow, jump
                        VoltDiff (CG3Dn,1,mV20,4,True,0,_60Hz,1000.0/Sens,0)
                                               'wire colours: high brown, low green, ground thick black
                        BrHalf4W (CNR1TC,1,mV50,mV50,5,Vx2,1,4200,True,True,0,250,1.0,0)
                                'used to make a 4 wire half bridge measurement
                                '4WPB100 connected to excitation channel
                                'two channels needed for 4WPB100 and PT100
                                'wire colours: 4WPB100 in ch 1: low red, ground blue.
                                wire colours: PT100 panel temperature in chl 2: high
                                'yellow, low green, ground thick black

                        PRT (CNR1TC,1,CNR1TC,1,0) ' Calculates temperature from the resistance
                                'of an RTD (thermistor)
                        CNR1TK = CNR1TC+273.15 'body temperature of CNR1 in K

                        NetRs=CM3Up-CM3Dn        'positive for net downwelling
                        NetRl=CG3Up-CG3Dn        'positive for net downwelling
                        Albedo=CM3Dn/CM3Up
                        UpTot=CM3Up+CG3Up        'total downwelling
                        DnTot=CM3Dn+CG3Dn        'total upwelling
                        NetTot=UpTot-DnTot       'total downwelling minus total upwelling,
                                                 'positive for net downwelling
                        CG3UpCo=CG3Up+5.67*10^-8*CNR1TK^4   'calibrated downwelling LW
                        CG3DnCo=CG3Dn+5.67*10^-8*CNR1TK^4   'calibrated upwelling LW

        'Instructions for barometer measurement
                        VoltDiff (Air_PRESShPa,1,mV5000,7,True,0,_60Hz,0.2,600)
                'Convert air pressure from hPa to kPa
```

```
            Air_PRESSkPa = Air_PRESShPa*0.1

    'Call Data Tables and Store Data
        CallTable AT_RH
        CallTable Batt
        CallTable IR_Temp
        CallTable Sn_depth
        CallTable Net_rad
        CallTable Baro
    NextScan
EndProg
```