PRECISION MEASUREMENT ENGINEERING, INC
2792 Loker Ave. West, Suite 105
Carlsbad, CA 92010-6615

TEL - (760)579-0300
FAX - (760)579-0301
www.pme.com

SOFTWARE SETUP: RS-232 T-CHAIN / DATALOGGER

REV: 01-APR-2010

INTRODUCTION

This document describes how to connect a PC / laptop and set up the software interface between a PME T-Chain and Campbell Scientific datalogger.

CONNECTING PC / LAPTOP TO DATALOGGER

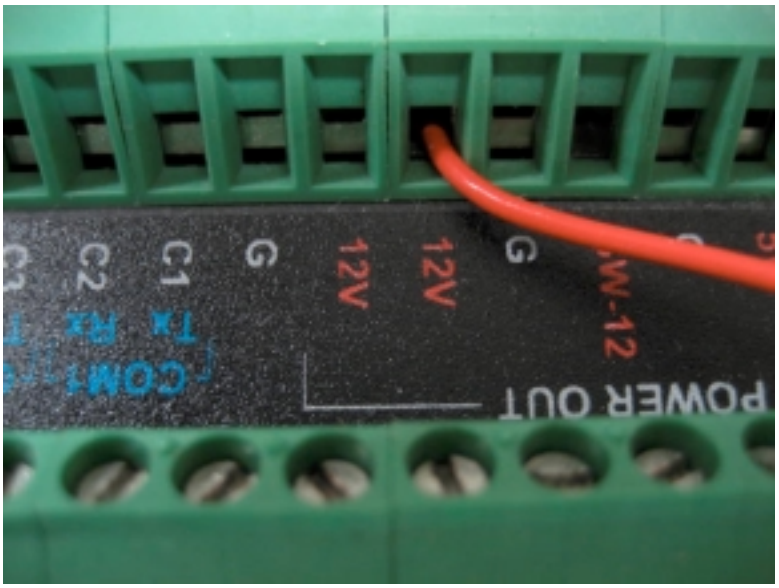
If using **DC only** to power the computer/laptop, then connect the datalogger to the computer through the datalogger's RS-232 port (Model CR3000 will take A/C or DC). Use a standard RS-232 serial cable.

If using **DC or A/C** to power the computer/laptop, then connect the datalogger to the computer through the datalogger's CS I/O port using the SC32B module (http://www.campbellsci.com/sc32b) and a 9-pin serial cable. If you don't have a serial port (USB only) then use a 17394 cable (http://www.campbellsci.com/17394-cable) along with the SC32B module. Another option is the SC-USB device. (http://www.campbellsci.com/documents/product-brochures/b_sc-usb.pdf)

MODEL	RS-232 PORT	CS I/0 PORT
800 / 850	DC powered PC/laptop only	DC/AC powered PC/laptop
1000	DC powered PC/laptop only	DC/AC powered PC/laptop
3000	DC/AC powered PC/laptop	DC/AC powered PC/laptop

USING THE 12V PORT FOR CONSTANT POWER

The T-Chain's power (red wire) can be connected to either 12V port. (photo below)
Do not exceed pulling more than 3Amps out of both ports.



During the first two seconds the receipt of any character causes the T-Chain to enter its monitor program.

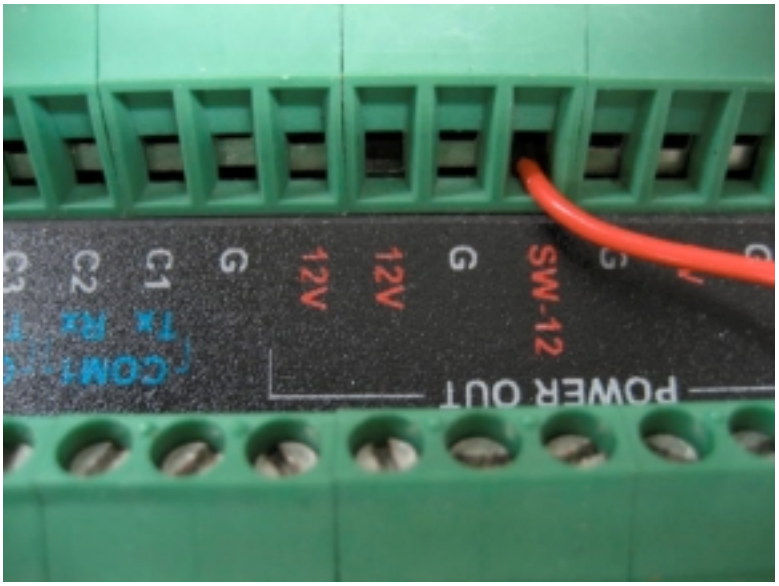
If no character is received during the first 2 seconds, then the T-Chain program begins. The banner prints and the program respond to characters with a temperature string.

If it enters the monitor program, then it remains there until the power is turned off or until a 'G' character is received.

If a 'G' character is received, then the monitor program begins the T-Chain program.

USING THE SW-12 (SWITCHED) PORT FOR USER-SPECIFICED POWER

The T-Chain's power (red wire) can be connected to the SW-12 port. (photo below)



The datalogger can be programmed to power up at a specified time or interval.

The program code below reflects a 10 to 11 second scan. If a 5 to 6 second scan rate is desired, then change the scan rate to one second. Note also the program uses the Sequentialmode instruction at the beginning, which is recommended when using the SW12() command.

Program code

```
SequentialMode

Const CR=&H0000000D
Const LF= &H0000000A
Const CRLF=&H00000D0A
Const G=&H00000047

'Declare variables
Public PTemp, batt_volt
Public Temp1, Temp2, Temp3, Temp4, Temp5, Temp6, Temp7, Temp8,
Public Temp9, Temp10,
Public SerialInput As String * 150
Public Split(10) As Long
Public NBytesReturned As Long
Dim porton
```

```
'Define Data Tables
DataTable (Test,1,1000)
DataInterval (0,60,Min,10)
Minimum (1,batt_volt,FP2,0,False)
Sample (1,PTemp,FP2)
Average (1,Temp1,FP2,False)
Average (1,Temp2,FP2,False)
Average (1,Temp3,FP2,False)
Average (1,Temp4,FP2,False)
EndTable

'Main Program
BeginProg

'Open serial port for communication with the Datalogger
SerialOpen(ComRS232,9600,0,0,10000) 'opening RS232 serial port to read T-Chain.

Scan (5,Sec,0,0)
PanelTemp (PTemp,250)
Battery (batt_volt)
'T-CHAIN measurements - RS 232 Serial, Com ports 1,2,3,4 will not work using this T-Chain
'RS-232 logic levels are +-12
SW12(1)
Delay (1,5,Sec)
SerialFlush (COMRS232)
SerialOut (COMRS232,G,"",0,0) 'sends "G" to start exit of sensor setup state
SerialOut (ComRS232,CR,"",0,0) 'sends a CR to complete exit of sensor setup state
Delay(1,1,Sec)
SerialIn(SerialInput, COMRS232, 0,CR,400)  'get serial string from tchain
SplitStr (Split(),SerialInput," ",10,0) 'split input using any non-numeric character as the delimiter
Temp1 = Split(1) / 1000 'Converts number into engineering unit
Temp2 = Split(2) / 1000 'Converts number into engineering unit
Temp3 = Split(3) / 1000 'Converts number into engineering unit
Temp4 = Split(4) / 1000 'Converts number into engineering unit
Temp5 = Split(5) / 1000 'Converts number into engineering unit
Temp6 = Split(6) / 1000 'Converts number into engineering unit
Temp7 = Split(7) / 1000 'Converts number into engineering unit
Temp8 = Split(8) / 1000 'Converts number into engineering unit
Temp9= Split(9) / 1000 'Converts number into engineering unit
Temp10 = Split(10) / 1000 'Converts number into engineering unit
sw12(0)
'Call Output Tables
'Example:
CallTable Test
NextScan
EndProg
```

USING THE RS-232 PORT TO CONNECT TO THE T-CHAIN

BEFORE CONNECTING OR DISCONNECTING THE T-CHAIN, THE POWER MUST BE OFF!!!

Configure RS-232 port to 9600 Baud.

Connect the T-Chain to the RS-232 port. Follow the instructions in the HARDWARE SETUP: RS-232 T-CHAIN / DATALOGGER document.

Load program onto datalogger. The scan rate should be 4 seconds or greater

Must use RS-232 port; cannot use Com ports 1, 2, 3, or 4 on datalogger.

Sample CR1000 Datalogger program for measuring T-Chain with 10 thermistors

```
'Declare constants for line feed and carriage return in hexidecimal
Const CR=&H0000000D
Const LF= &H0000000A
Const CRLF=&H00000D0A

'Declare variables
Public PTemp, batt_volt
Public Temp1, Temp2, Temp3, Temp4, Temp5, Temp6, Temp7, Temp8,
Public Temp9, Temp10,
Public SerialInput As String * 150
Public Split(10) As Long
Public NBytesReturned As Long '
Dim counter
```

'Define Data Tables

```
DataTable (Test,counter>1,1000) 'data is only stored after second scan to remove first scan error readings
  DataInterval (0,1,Min,10)
  Minimum (1,batt_volt,FP2,0,False)
  Sample (1,PTemp,FP2)
  Average (1,Temp1,FP2,False)
  Average (1,Temp2,FP2,False)
  Average (1,Temp3,FP2,False)
  Average (1,Temp4,FP2,False)
EndTable

'Main Program
BeginProg
  'Open serial port for communication with the Datalogger
  SerialOpen(ComRS232,9600,0,0,10000) 'opening RS232 serial port to read T-Chain.


  Scan (5,Sec,0,0)
```

```
   PanelTemp (PTemp,250)
   Battery (batt_volt)


 'T-CHAIN measurements - RS 232 Serial, Com ports 1,2,3,4 will not work using this T-Chain
   'RS-232 logic levels are +-12v
   SerialFlush(ComRS232)
   SerialOut (ComRS232,CR,"",0,0) 'sends a CR to wake up the t-chain
   SerialIn(SerialInput, COMRS232, 0,CR,400)  'get serial string from tchain
   SplitStr (Split(),SerialInput,"   ",10,0) 'split input using any non-numeric character as the
delimiter
   Temp1 = Split(1) / 1000  'Converts number into engineering unit
   Temp2 = Split(2) / 1000  'Converts number into engineering unit
   Temp3 = Split(3) / 1000  'Converts number into engineering unit
   Temp4 = Split(4) / 1000  'Converts number into engineering unit
   Temp5 = Split(5) / 1000  'Converts number into engineering unit
   Temp6 = Split(6) / 1000  'Converts number into engineering unit
   Temp7 = Split(7) / 1000  'Converts number into engineering unit
   Temp8 = Split(8) / 1000  'Converts number into engineering unit
   Temp9= Split(9) / 1000  'Converts number into engineering unit
   Temp10 = Split(10) / 1000  'Converts number into engineering unit
counter=counter+1
   'Call Output Tables
   'Example:
   CallTable Test
  NextScan
EndProg
```

USING THE SDM-SI01 MODULE TO CONNECT TO THE T-CHAIN

BEFORE CONNECTING OR DISCONNECTING THE T-CHAIN, THE POWER MUST BE OFF!!!

Connect the DB-9 connector in PME's I/O cable assembly to the male end of Campbell Scientific's SC110 cable. (http://www.campbellsci.com/sc110)

Wire SC110 cable into SDM-SI01 module (http://www.campbellsci.com/sdm-sio1) and datalogger according to the following wiring pattern:

SC110 white wire to SI01 RX-A terminal
SC110 brown wire to SI01 TX-Z terminal
SC110 yellow wire to datalogger ground (G)

T-Chain red wire to datalogger 12v terminal
T-Chain black wire to datalogger G terminal

SDM_SI01 C1 to datalogger C1
SDM_SI01 C2 to datalogger C2
SDM_SI01 C3 to datalogger C3
SDM_SI01 12v to datalogger 12v
SDM_SI01 G to datalogger G

Scan rate should be 2 seconds or greater.

Program example for using SDM-SI01 to measure T-Chain with 10 thermistors:

```
'CR800 Series Datalogger
'Program using SDM-SI01 (address 1) to measure PME T-chain

Const CR=&H0000000D
Const LF= &H0000000A
Const CRLF=&H00000D0A

'Declare variables
Public PTemp, batt_volt
Public Temp1, Temp2, Temp3, Temp4, Temp5, Temp6, Temp7, Temp8,
Public Temp9, Temp10,
Public SerialInput As String * 150
Public Split(10) As Long
Public NBytesReturned As Long
Dim  counter

'Define Data Tables
DataTable (Test,counter>1,1000) 'data is only stored after second scan to remove first scan error readings
  DataInterval (0,1,Min,10)
  Minimum (1,batt_volt,FP2,0,False)
  Sample (1,PTemp,FP2)
  Average (1,Temp1,FP2,False)
  Average (1,Temp2,FP2,False)
  Average (1,Temp3,FP2,False)
  Average (1,Temp4,FP2,False)
EndTable

'Main Program
BeginProg
  'Open serial port for communication with the Datalogger
  SerialOpen(33,9600,0,0,10000) '33 parameter reflects the SDM device comport code for SDM address 1

  Scan (5,Sec,0,0)
    PanelTemp (PTemp,250)
    Battery (batt_volt)


'T-CHAIN measurements -using SDM-SI01
   SerialFlush(33)
    SerialOut (33,CR,"",0,0) 'sends a CR to wake up the t-chain
    SerialIn(SerialInput, COMRS232, 0,CR,400)  'get serial string from tchain
    SplitStr (Split(),SerialInput,"   ",10,0) 'split input using any non-numeric character as the delimiter
    Temp1 = Split(1) / 1000  'Converts number into engineering unit
    Temp2 = Split(2) / 1000  'Converts number into engineering unit
    Temp3 = Split(3) / 1000  'Converts number into engineering unit
    Temp4 = Split(4) / 1000  'Converts number into engineering unit
```

```
    Temp5 = Split(5) / 1000  'Converts number into engineering unit
    Temp6 = Split(6) / 1000  'Converts number into engineering unit
    Temp7 = Split(7) / 1000  'Converts number into engineering unit
    Temp8 = Split(8) / 1000  'Converts number into engineering unit
    Temp9= Split(9) / 1000  'Converts number into engineering unit
    Temp10 = Split(10) / 1000  'Converts number into engineering unit
counter=counter+1

    'Call Output Tables
    'Example:
    CallTable Test
  NextScan
EndProg
```

CR1000 DATALOGGER