Northwest Irrigation and Soils Research Laboratory
3793 N. 3600 E. • Kimberly, ID 83341-5076 • (208) 423-5582 Station Note 05-01

Sequential Tension AutoSampler – Operating Program

R.D. Lentz
USDA-Agricultural Research Service; 3793 N 3600 E, Kimberly, ID 83341 PH:(208)423-6531, FAX (208)423-6555

SUMMARY:
This note describes the Campbell Scientific, Inc. (Logan, Utah) CR10X datalogger program used to control the Sequential Tension Autosampler instrument. The instrument automatically collects a series of water volumes from a sampling device which extracts water under continuous tension. The use of the program is discussed and a listing of the program is provided.

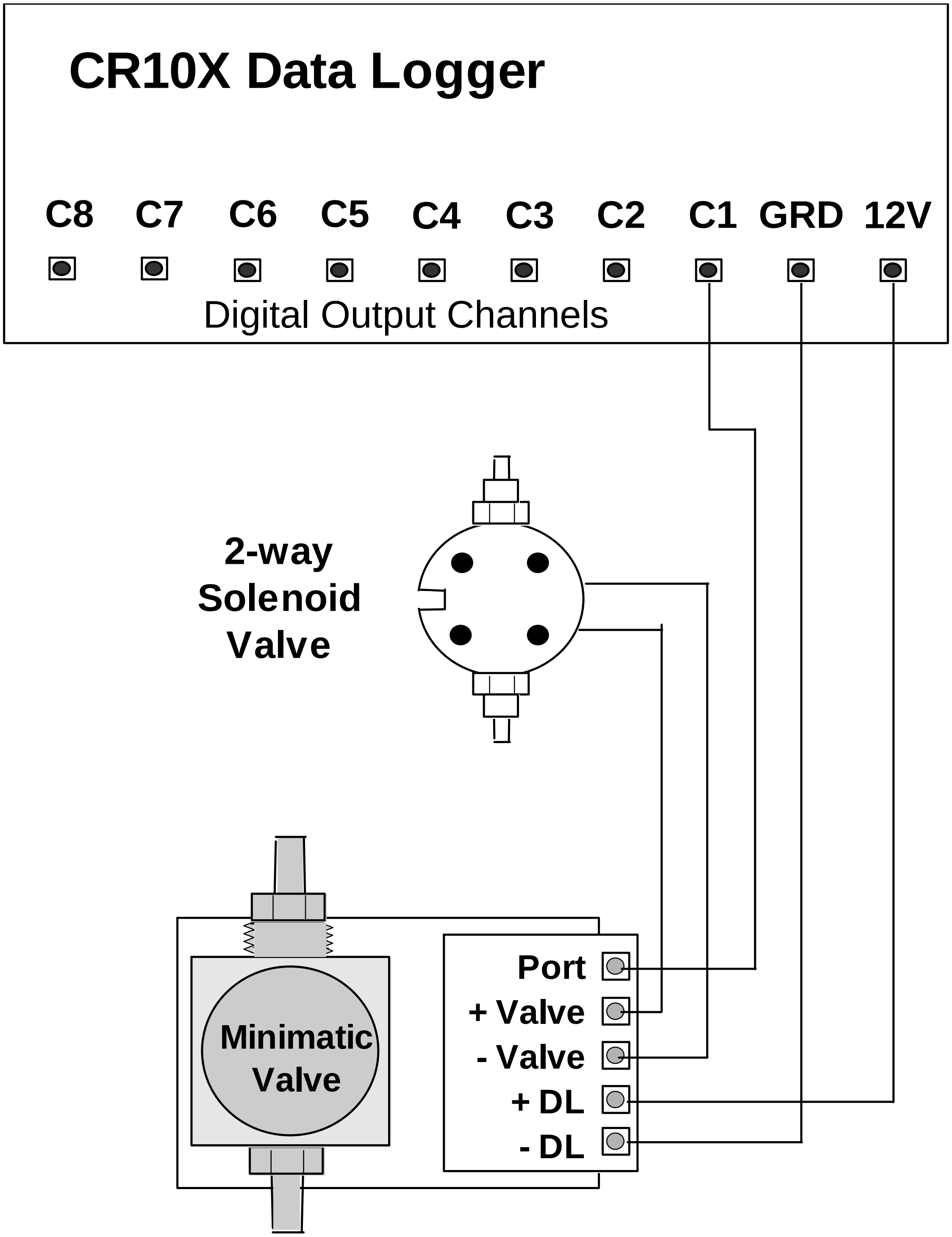
INTRODUCTION:
The sequential tension autosampler (STAS) instrument automatically collects a series of discrete water samples from a soil water percolation sampler, or similar device, that withdraws water from an unsaturated porous media under continuous vacuum. A detailed description of the STAS instrument is given by Lentz (2005). The STAS can extract a sequence of one to seven, 100-mL water samples over a period of minutes or days, depending on the sampling frequency programmed. Larger sample volumes can be collected by increasing sample bottle size. A Campbell Scientific, Inc. CR10X data logger operates air and water flow valves that perform the sampling functions. This note provides a listing of the CR10X program, describes its use, and illustrates the electrical cabling used to connect the datalogger to STAS controllers and valves.

DATALOGGER:
Campbells Scientific, Inc. CR10X.

CR10X WIRING CONNECTIONS:
The eight digital output, 12 VDC, and ground terminals on the CR10X wiring panel are used to operate the STAS. Digital output channel one is wired to the valve controller that operates the teflon solenoid fluid- and Minimatic air-flow valves for sample bottle one (Fig. 1). The valve controllers for sample bottles two through seven are connected in an identical manner to digital output channels two through seven. Digital output channel eight on the CR10X controls the 3-way teflon solenoid valve and is connected to the 3-way valve controller as per Fig. 2.

CR10X PROGRAM:
The CR10X data logger program (VacCntL3c), which controls the STAS instrument functions, is listed in the appendix in uncompiled format (VacCntL3c.CSI). A compiled version of the program that may be downloaded directly to the data logger(VacCntL3c.dld) is available on the web at "http://www.nwisrl.ars.usda.gov/lentz/vaccntl3c.dld"

PROGRAMMING THE CR10X FOR SAMPLING:
To begin a STAS sampling sequence, the user assigns values to control parameters, which are processed in the VacCntL3c's initialization routine. The program is then recompiled; this installs the new control parameters, and initiates the sample collection sequence. Table 1 lists the control parameters, identifies the program instruction used to initialize each parameter, and describes the stroke sequence used to key new parameter values into the data logger (via the keyboard/display).



Valves & Controller for Sample Bottle #1

Fig. 1 Wiring connections to controller and valves of sample bottle #1.

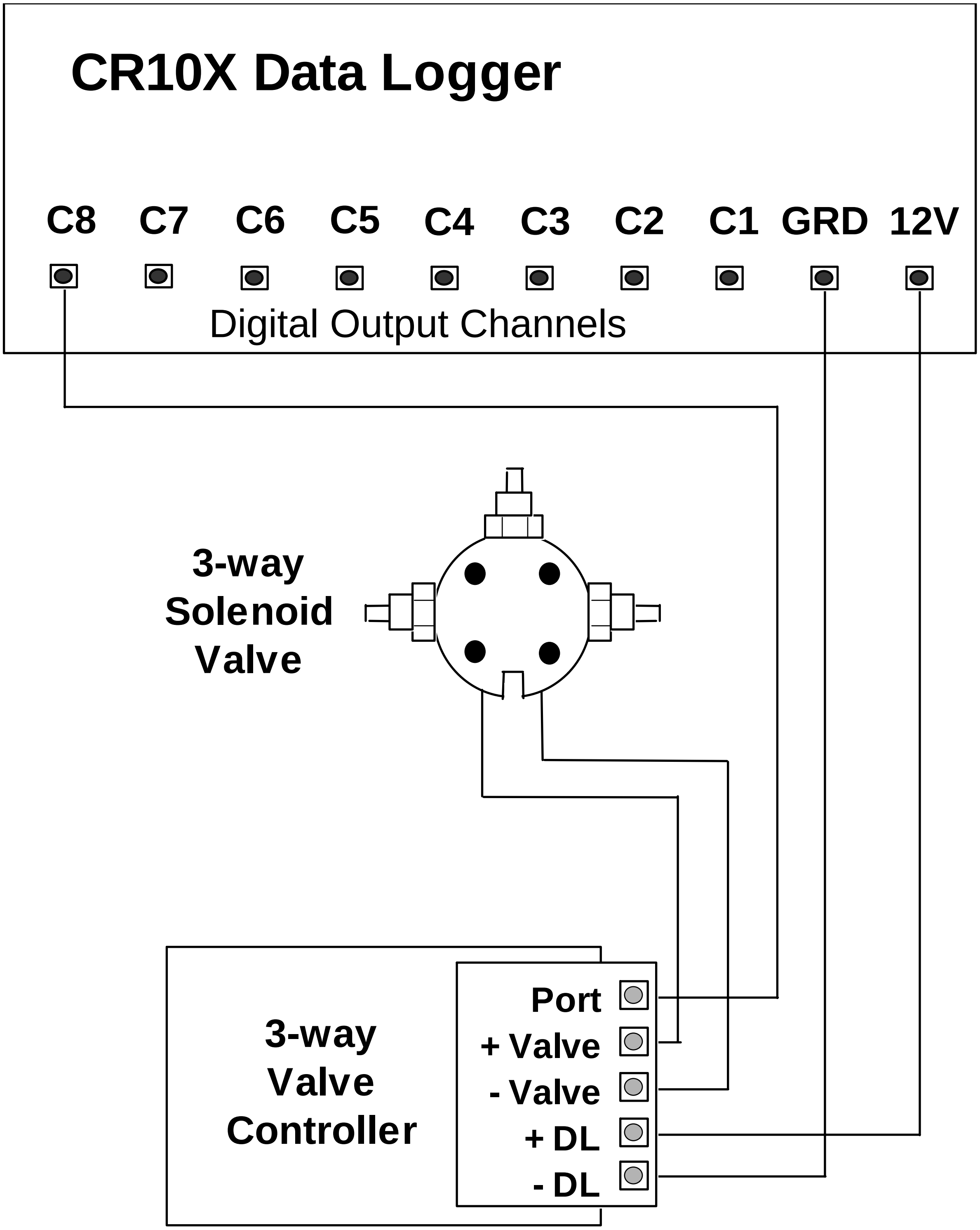


Fig. 2 Wiring connections to 3-way valve controller and valve.

Table 1. Listing of CR10X control parameters that guide the STAS sampling sequence, including the program Table-1 instruction, the instruction parameter number, and CR10X keying sequence used to reset each control parameter. After resetting parameters, key in '*,0' to recompile.

Control Parameter to Be Defined	Program Table 1 Instruction #	Instruction Parameter #	Key Sequence Pressed To Define New Parameter
Interval between sampling start times (1-2879 min)	2	1	*, 1, 2, A, A, min, A
Sample collection period (1-2779 min)	3	1	*, 1, 3, A, A, min, A
Number of samples to be collected (1 to 7)	4	1	*, 1, 4, A, A, min, A
Time to start collecting first sample			
Day of year (1-365)	6	1	*, 1, 6, A, A, DOY, A
Hour (0-23)	7	1	*, 1, 7, A, A, hour, A
Min (0-59)	8	1	*, 1, 8, A, A, min, A
Time to stop sampling sequence			
Day of year (1-365)	13	1	*, 1, 13, A, A, DOY, A
Hour (0-23)	14	1	*, 1, 14, A, A, hour, A
Min (0-59)	15	1	*, 1, 15, A, A, min, A

CURRENT PROGRAM LIMITATIONS:
The sampling program must be reset at the start of each new year. The minimum sampling interval must be greater or equal to the sampling period. The maximum sampling interval permissible is 2879 min.

PROGRAM OUTPUT:
Each time a new sampling sequence is initiated, the program outputs the current control parameter values to the data logger's final storage area (Table 2). This information record is identified as array identification (I.D.) #100. The actual start and stop times used in the collection of each sample are also output. The start time record is identified as array I.D. #101 and the stop time record is identified as array I.D. #102 (Table 2).

REFERENCES:
Lentz, R.D. 2005. Automated system for collecting multiple, sequential samples from soil water samplers under continuous vacuum. Communications in Plant and Soil Science (submitted).

Table 2. Information records output to the data logger final storage area.

Array ID	Loc. 1	Loc. 2	Loc. 3	Loc. 4	Loc. 5	Loc. 6	Loc. 7	Loc. 8	Loc. 9	Loc. 10	Loc. 11
100	year	day	hr/min	time interval	sampling period	start DOY	start hour	start min	stop DOY	stop hour	stop min
101	year	day	hr/min	sample #	hour collection started	min collection started	battery voltage	--	--	--	--
102	year	day	hr/min	sample #	hour collection stopped	min collection stopped	battery voltage	--	--	--	--

APPENDIX:

```
;{CR10X}
;{CR10X} VacCntrL3c [cr10x version] 04-28-04 [Latest version used]
; Program sequentially opens valves for automatic sequential sampling
;  of solutiion extracted from a percolation sampler
;
;  Control Ports #1-7 controls fluid & vacuum valves used to divert sample flow  collection flasks;
;  Control Port #8 - Controls fluid & vacuum valves for diverting sample flow from waste storage to collect
flasks;
;  Flag #0 high if output to storage area, low if not
;  Flag #1 high if program parameters have been initialized, low if not
;  Flag #2 high if sampling sequence has been initiated, low if not
;  Flag #3 high if sample now being diverted to collection flask, low if not
;  Flag #4 high if sampling sequence has been completed, low if not
;
;
; PARAMETERS INPUT REQUIRED
;  TimeInt[1]:  minutes between initiation of sample diversion to collection flask and start time of next
sampling
;  SampPrd[18]:  sampling period, time (minutes) during which sample is diverted to collection flask
;  NumSamp[21]:  number of samples to take [1-7]
;  StartDOY[12]:  day of year sampling sequence is to begin ie. first sampling initiated
;  StartTime[6]: time (military time-no colon) sampling sequence is to begin, ie. first sampling initiated
;  StopDOY[3]:  day of year all sample diversions to collection flasks are to end
;  StopTime[8]: time (military time- no colon) all sample diversion to collection flasks are to end
;
;
; OTHER VARIABLES
;  StartHour[9]: Hour first sample is taken
;  StartMin[7]: Min first sample is taken
;  StopHour[4]: Hour sample sequence is completed (fail safe)
;  StopMin[5]:  Min sample sequence is completed (fail safe)
;  _batVolt_[2]: battery voltage
;  CrntDOY[23]:  todays DOY
;  CrntHr[24]: hours into current day
;  CrntMin[25]:  minutes into the current hour
;  CrntSec[26]: seconds into the current min
```

```
;  NextDOY[11]: DOY of next sampling
;  NextMin[13]:  Min into Day of next sampling
;  NxtStpDay[20]:  DOY when this sample collection stops
;  NxtStpMin[19]:  Min into day when this sample collection stops
;  CntMinDay[15]:  minutes into current day
;  SmplCnt[16]: sample counter
;  Sumtime[12]: intermediate calculation
;  SampStrt[14]: min into day that sample diversion to collection flask was started


*Table 1 Program
  01: 1        Execution Interval (seconds)

; Initialize program parameters, if not already done
1:  If Flag/Port (P91)
 1: 21       Do if Flag 1 is Low
 2: 30       Then Do

;  Set period (minutes from last sampling
;   initiation) between samplings
     2:  Z=F x 10^n (P30)
      1: 5        F
      2: 00       n, Exponent of 10
      3: 1        Z Loc [ TimeInt   ]

;  Set sampling period (minutes)
     3:  Z=F x 10^n (P30)
      1: 2        F
      2: 00       n, Exponent of 10
      3: 16       Z Loc [ SampPrd   ]

;  Set Number of samples to b e taken (1-7)
     4:  Z=F x 10^n (P30)
      1: 2        F
      2: 0        n, Exponent of 10
      3: 19       Z Loc [ NumSamp   ]

;  Initialize sample counter to zero
     5:  Z=F x 10^n (P30)
      1: 0        F
      2: 0        n, Exponent of 10
      3: 14       Z Loc [ SmplCnt   ]

;  Set day of year on which sampling sequence is to begin
     6:  Z=F x 10^n (P30)
      1: 113      F
      2: 0        n, Exponent of 10
      3: 8        Z Loc [ StartDOY  ]

;  Set military hour into day sampling sequence is to begin
     7:  Z=F x 10^n (P30)
      1: 23       F
      2: 0        n, Exponent of 10
      3: 7        Z Loc [ StartHour ]

;  Set min into hour sampling sequence is to begin
```

```
  8:  Z=F x 10^n (P30)
   1: 59       F
   2: 0        n, Exponent of 10
   3: 6        Z Loc [ StartMin  ]

;  Initialize DOY of next sample to collect to no collect value
  9:  Z=F x 10^n (P30)
   1: 0        F
   2: 0        n, Exponent of 10
   3: 9        Z Loc [ NextDOY   ]

;  Initialize Minute into Day of next sample to collect to no collect value
  10:  Z=F x 10^n (P30)
   1: 1441     F
   2: 0        n, Exponent of 10
   3: 11       Z Loc [ NextMin   ]

;  Set DOY for stopping current sample collection
  11:  Z=F x 10^n (P30)
   1: 0        F
   2: 0        n, Exponent of 10
   3: 18       Z Loc [ NxtStpDOY ]

;  Set Minute into Day for stopping current sample collection
  12:  Z=F x 10^n (P30)
   1: 1        F
   2: 0        n, Exponent of 10
   3: 17       Z Loc [ NxtStpMin ]

;  Set day of year on which sampling sequence is to end
  13:  Z=F x 10^n (P30)
   1: 113      F
   2: 0        n, Exponent of 10
   3: 3        Z Loc [ StopDOY   ]

;  Set military hour into day  sampling sequence is to stop
  14:  Z=F x 10^n (P30)
   1: 1        F
   2: 0        n, Exponent of 10
   3: 4        Z Loc [ StopHour  ]

;  Set minute into hour sampling sequence is to stop
  15:  Z=F x 10^n (P30)
   1: 0        F
   2: 0        n, Exponent of 10
   3: 5        Z Loc [ StopMin   ]

  16:  Set Port(s) (P20)
   1: 7777     C8..C5 = output/output/output/output
   2: 7777     C4..C1 = output/output/output/output

;  Call Subroutine 1: Convert Start/StopTime to hour and min values

  17:  Do (P86)
   1: 1        Call Subroutine 1
```

```
18:  Do (P86)
 1: 11       Set Flag 1 High

19:  Do (P86)
 1: 22       Set Flag 2 Low

20:  Do (P86)
 1: 23       Set Flag 3 Low

21:  Do (P86)
 1: 24       Set Flag 4 Low

22:  Do (P86)
 1: 10       Set Output Flag High

23:  Set Active Storage Area (P80)^8303
 1: 1        Final Storage
 2: 100      Array ID or Loc [ _________ ]

24:  Real Time (P77)^20042
 1: 1110     Year,Day,Hour/Minute

25:  Sample (P70)^8735
 1: 1        Reps
 2: 1        Loc [ TimeInt   ]

26:  Sample (P70)^27868
 1: 1        Reps
 2: 16       Loc [ SampPrd   ]

27:  Sample (P70)^13021
 1: 1        Reps
 2: 19       Loc [ NumSamp   ]

28:  Sample (P70)^12717
 1: 1        Reps
 2: 8        Loc [ StartDOY  ]

29:  Sample (P70)^26670
 1: 1        Reps
 2: 7        Loc [ StartHour ]

30:  Sample (P70)^27681
 1: 1        Reps
 2: 6        Loc [ StartMin  ]

31:  Sample (P70)^78
 1: 1        Reps
 2: 3        Loc [ StopDOY   ]

;   Output stophour and stopmin
32:  Sample (P70)^6364
 1: 2        Reps
 2: 4        Loc [ StopHour  ]

33:  Serial Out (P96)
```

```
    1: 71      Storage Module

   34:  Do (P86)
    1: 20      Set Output Flag Low

35:  End (P95)


36:  Do (P86)
 1: 6       Call Subroutine 6


37:  If time is (P92)
 1: 0       Minutes into a
 2: 1       Minute Interval
 3: 30       Then Do

   38:  Batt Voltage (P10)
    1: 2       Loc [ _batVolt_ ]

;  Get current DOY, Hour, and Min
   39:  Do (P86)
    1: 2       Call Subroutine 2

40:  Do (P86)
 1: 6       Call Subroutine 6


;  If sampling sequence is not finished, proceed with program
   41:  If Flag/Input (P91)
    1: 24      Do if Flag 4 is Low
    2: 30      Then Do

;  If sample not being diverted to collection flask now, start doing so if time is right
      42:  If Flag/Input (P91)
       1: 23      Do if Flag 3 is Low
       2: 3       Call Subroutine 3

43:  Do (P86)
 1: 6       Call Subroutine 6

;  If sample is being diverted to collection flask now, end it if time is right
      44:  If Flag/Input (P91)
       1: 13      Do if Flag 3 is High
       2: 4       Call Subroutine 4

45:  Do (P86)
 1: 6       Call Subroutine 6

;  Check if Stoptime reached, then turn all sample diversion off
      46:  If (X<=>Y) (P88)
       1: 3       X Loc [ StopDOY   ]
       2: 1       =
       3: 21      Y Loc [ CrntDOY   ]
       4: 30      Then Do
```

```
        47:  If (X<=>Y) (P88)
         1: 4       X Loc [ StopHour  ]
         2: 1       =
         3: 22       Y Loc [ CrntHr    ]
         4: 30       Then Do

            48:  If (X<=>Y) (P88)
             1: 5       X Loc [ StopMin   ]
             2: 1       =
             3: 23       Y Loc [ CrntMin   ]
             4: 5       Call Subroutine 5

        49:  End (P95)

     50:  End (P95)

  51:  End (P95)

;  END OF MAIN PROGRAM
52:  End (P95)

;*******************************************************************
;*******************************************************************

*Table 2 Program
  02: 0.0000    Execution Interval (seconds)

;*******************************************************************
;*******************************************************************


*Table 3 Subroutines

;-------------------------------------------------
;#1: Converts Start & StopTime and to __Hour, __Min
;    Sets NextDOY and NextMin for next sample time

1:  Beginning of Subroutine (P85)
 1: 1       Subroutine 1

   2:  Z=X (P31)
    1: 8       X Loc [ StartDOY  ]
    2: 9       Z Loc [ NextDOY   ]


   3:  Z=X*F (P37)
    1: 7       X Loc [ StartHour ]
    2: 60       F
    3: 10       Z Loc [ Sumtime   ]

   4:  Z=X+Y (P33)
    1: 10       X Loc [ Sumtime   ]
    2: 6       Y Loc [ StartMin  ]
    3: 11       Z Loc [ NextMin   ]

;  Assign NextMin value to SampStrt variable
```

```
    5:  Z=X (P31)
     1: 11       X Loc [ NextMin   ]
     2: 12       Z Loc [ SampStrt  ]

6:  End (P95)
; End Subroutine #1
;-------------------------------------------------

;-------------------------------------------------
;#2:  Gets Current DOY, Hour, Min

7:  Beginning of Subroutine (P85)
 1: 2        Subroutine 2

    8:  Time (P18)
     1: 3        Store Year,Day,Hr,Min,Sec in 5 consecutive locations
     2: 0000     Mod/By
     3: 20       Loc [ CrntYr    ]

    9:  Z=X*F (P37)
     1: 22       X Loc [ CrntHr    ]
     2: 60       F
     3: 13       Z Loc [ CntMinDay ]

    10:  Z=X+Y (P33)
     1: 23       X Loc [ CrntMin   ]
     2: 13       Y Loc [ CntMinDay ]
     3: 13       Z Loc [ CntMinDay ]

11:  End (P95)
; End Subroutine #2
;-------------------------------------------------

;-------------------------------------------------
;#3:  Starts sampling collection if time is right

12:  Beginning of Subroutine (P85)
 1: 3        Subroutine 3

    13:  If (X<=>Y) (P88)
     1: 9        X Loc [ NextDOY   ]
     2: 1        =
     3: 21       Y Loc [ CrntDOY   ]
     4: 30       Then Do

        14:  If (X<=>Y) (P88)
         1: 11       X Loc [ NextMin   ]
         2: 1        =
         3: 13       Y Loc [ CntMinDay ]
         4: 30       Then Do

;  Increment Sample Number counter
            15:  Z=Z+1 (P32)
             1: 14       Z Loc [ SmplCnt   ]

;  Assign NextMin to SampStrt
```

```
        16:  Z=X (P31)
         1: 11       X Loc [ NextMin   ]
         2: 12       Z Loc [ SampStrt  ]

;  Open appropriate solenoid valve
        17:  CASE (P93)
         1: 14       Case Loc [ SmplCnt   ]

            18:  If Case Location < F (P83)
             1: 2        F
             2: 41       Set Port 1 High

            19:  If Case Location < F (P83)
             1: 3        F
             2: 42       Set Port 2 High

            20:  If Case Location < F (P83)
             1: 4        F
             2: 43       Set Port 3 High

            21:  If Case Location < F (P83)
             1: 5        F
             2: 44       Set Port 4 High

            22:  If Case Location < F (P83)
             1: 6        F
             2: 45       Set Port 5 High

            23:  If Case Location < F (P83)
             1: 7        F
             2: 46       Set Port 6 High

            24:  If Case Location < F (P83)
             1: 8        F
             2: 47       Set Port 7 High

        25:  End (P95)

;  Divert cup flow to the sample collection system from main or waste container.
        26:  Do (P86)
         1: 48       Set Port 8 High

;  Compute current sample collection stop times

        27:  Z=X+Y (P33)
         1: 12       X Loc [ SampStrt  ]
         2: 16       Y Loc [ SampPrd   ]
         3: 17       Z Loc [ NxtStpMin ]

        28:  Z=X (P31)
         1: 21       X Loc [ CrntDOY   ]
         2: 18       Z Loc [ NxtStpDOY ]

        29:  If (X<=>F) (P89)
         1: 17       X Loc [ NxtStpMin ]
         2: 3        >=
```

```
 3: 1440     F
 4: 30       Then Do

    30:  Z=X+F (P34)
     1: 17       X Loc [ NxtStpMin ]
     2: -1440    F
     3: 17       Z Loc [ NxtStpMin ]

    31:  Z=Z+1 (P32)
     1: 18       Z Loc [ NxtStpDOY ]

32:  End (P95)
```

; Set indicator flags 2 and 3 high

```
33:  Do (P86)
 1: 12       Set Flag 2 High

34:  Do (P86)
 1: 13       Set Flag 3 High
```

; Output sample count and collection start time

```
35:  Do (P86)
 1: 10       Set Output Flag High

36:  Set Active Storage Area (P80)^5607
 1: 1        Final Storage
 2: 101      Array ID or Loc [ _________ ]
```

; Output current sample number

```
37:  Sample (P70)^3743
 1: 1        Reps
 2: 14       Loc [ SmplCnt   ]

38:  Real Time (P77)^29963
 1: 1110     Year,Day,Hour/Minute (midnight = 0000)
```

; Output current hour into day, min into hour

```
39:  Sample (P70)^1184
 1: 1        Reps
 2: 22       Loc [ CrntHr    ]

40:  Sample (P70)^29904
 1: 1        Reps
 2: 23       Loc [ CrntMin   ]

41:  Sample (P70)^420
 1: 1        Reps
 2: 2        Loc [ _batVolt_ ]

42:  Do (P86)
 1: 20       Set Output Flag Low

43:  Serial Out (P96)
 1: 71       Storage Module
```

```
        44:  End (P95)

    45:  End (P95)

46:  End (P95)
; End Subroutine #3
;---------------------------------------------------


;--------------------------------------------------
;#4:  Ends current sampling collection if time is right
;     and computes next sampling time

47:  Beginning of Subroutine (P85)
 1: 4        Subroutine 4

     48:  If (X<=>Y) (P88)
      1: 18       X Loc [ NxtStpDOY ]
      2: 1        =
      3: 21       Y Loc [ CrntDOY   ]
      4: 30       Then Do

          49:  If (X<=>Y) (P88)
           1: 17       X Loc [ NxtStpMin ]
           2: 1        =
           3: 13       Y Loc [ CntMinDay ]
           4: 30       Then Do

;  3-way valve off:  diverts cup flow to main or waste container
;    from sample collection system
               50:  Do (P86)
                1: 58       Set Port 8 Low

;  Close appropriate sample solenoid valve
               51:  CASE (P93)
                1: 14       Case Loc [ SmplCnt   ]

                    52:  If Case Location < F (P83)
                     1: 2        F
                     2: 51       Set Port 1 Low

                    53:  If Case Location < F (P83)
                     1: 3        F
                     2: 52       Set Port 2 Low

                    54:  If Case Location < F (P83)
                     1: 4        F
                     2: 53       Set Port 3 Low

                    55:  If Case Location < F (P83)
                     1: 5        F
                     2: 54       Set Port 4 Low

                    56:  If Case Location < F (P83)
                     1: 6        F
```

```
          2: 55       Set Port 5 Low

         57:  If Case Location < F (P83)
          1: 7        F
          2: 56       Set Port 6 Low

         58:  If Case Location < F (P83)
          1: 8        F
          2: 57       Set Port 7 Low

     59:  End (P95)

;  Compute next sample collection start times

     60:  Z=X+Y (P33)
      1: 1        X Loc [ TimeInt   ]
      2: 11       Y Loc [ NextMin   ]
      3: 11       Z Loc [ NextMin   ]


     61:  If (X<=>F) (P89)
      1: 11       X Loc [ NextMin   ]
      2: 3        >=
      3: 1440     F
      4: 30       Then Do

         62:  Z=X+F (P34)
          1: 11       X Loc [ NextMin   ]
          2: -1440    F
          3: 11       Z Loc [ NextMin   ]

         63:  Z=Z+1 (P32)
          1: 9        Z Loc [ NextDOY   ]

     64:  End (P95)

;  Set indicator flag 3 low
     65:  Do (P86)
      1: 23       Set Flag 3 Low

;  Set indicator flag 4 high (indicating sampling sequence completed)
     66:  If (X<=>Y) (P88)
      1: 19       X Loc [ NumSamp   ]
      2: 1        =
      3: 14       Y Loc [ SmplCnt   ]
      4: 14       Set Flag 4 High

;  Output sample count and collection stop time
     67:  Do (P86)
      1: 10       Set Output Flag High

     68:  Set Active Storage Area (P80)^25437
      1: 1        Final Storage
      2: 102      Array ID or Loc [ _________ ]

;  Output current sample number
```

```
          69:  Sample (P70)^287
           1: 1        Reps
           2: 14       Loc [ SmplCnt   ]

          70:  Real Time (P77)^6123
           1: 1110     Year,Day,Hour/Minute (midnight = 0000)

;  Output current hour into day, min into hour
          71:  Sample (P70)^1184
           1: 1        Reps
           2: 22       Loc [ CrntHr    ]

          72:  Sample (P70)^2568
           1: 1        Reps
           2: 23       Loc [ CrntMin   ]

          73:  Sample (P70)^23058
           1: 1        Reps
           2: 2        Loc [ _batVolt_ ]

          74:  Serial Out (P96)
           1: 71       Storage Module

          75:  Do (P86)
           1: 20       Set Output Flag Low

       76:  End (P95)

    77:  End (P95)

78:  End (P95)
; End Subroutine #4
;---------------------------------------------------


;---------------------------------------------------
;#5:  routine turns off all
;     sample diversion (controlports #1-#7) and turns off port #8
;     so that ceramic cup flow goes to waste container

79:  Beginning of Subroutine (P85)
 1: 5        Subroutine 5

    80:  Do (P86)
     1: 51       Set Port 1 Low

    81:  Do (P86)
     1: 52       Set Port 2 Low

    82:  Do (P86)
     1: 53       Set Port 3 Low

    83:  Do (P86)
     1: 54       Set Port 4 Low

    84:  Do (P86)
```

```
    1: 55      Set Port 5 Low

   85:  Do (P86)
    1: 56      Set Port 6 Low

   86:  Do (P86)
    1: 57      Set Port 7 Low

   87:  Do (P86)
    1: 58      Set Port 8 Low

88:  End (P95)
; End Subroutine #5
;-----------------------------------------------------


;---------------------------------------------------
;#6:  routine outputs nextday

89:  Beginning of Subroutine (P85)
 1: 6       Subroutine 6

;  Output time and nextday
          90:  Do (P86)
           1: 10      Set Output Flag High

          91:  Set Active Storage Area (P80)^32399
           1: 1       Final Storage
           2: 111     Array ID or Loc [ _________ ]

;  Output current sample number
          92:  Sample (P70)^287
           1: 1       Reps
           2: 14      Loc [ SmplCnt   ]

          93:  Real Time (P77)^6123
           1: 1110    Year,Day,Hour/Minute (midnight = 0000)

;  Output nexthour
          94:  Sample (P70)^1184
           1: 1       Reps
           2: 9       Loc [ NextDOY   ]

          95:  Serial Out (P96)
           1: 71      Storage Module

          96:  Do (P86)
           1: 20      Set Output Flag Low

97:  End (P95)
```