

Program: Lys_V1_2e_example.CR1X

```
1  'CR1000X Series Datalogger
2  ' for two example lysimeters
3  '
4  'Ver 1.2 modified by Brad Lyles 20211229
5  ' Added code to read initial conditions from CPU:Example_current.txt on
6  ' reboot.
7  ' Changed cum_drainage code to update volumes greater than 5ml rather
8  ' than any positive value, to reduce the number of comments in
9  ' the EventLog.
10 '
11 'Ver 1_1f modified by Brad Lyles 20211118
12 ' made changes to delay stage reset for 20 seconds, to give time for
13 ' water to drain from discharge line back into stilling well.
14 ' added manual stage datum reset.
15 ' changed conversion for ET flux = mass flux + Perc flux
16 ' added code to not allow pump to turn on when Batt_Volt < 12
17 ' increase scan buffer from 3 to 10
18 '
19 ' Ver 1_1e modified by brad Lyles 20211109
20 ' added new calibration multipliers
21 '
22 ' V1_1d modified by Brad Lyles 20211107
23 ' Changed range code from 200mv to autorange
24 ' Changed the CS655 soil sensors to read into 2 dimensional array
25 '
26 ' V1_1c modified by Brad Lyles 20211104
27 ' added comments
28 '
29 'V1.1b modified by Brad Lyles 20211005
30 ' moved transducer for lysimeter 2 to C3
31 ' changed transducer ID to 0 (factory default)
32 '
33 ' V1_1 modified by Brad Lyles 20210915
34 ' added table EventLog
35 ' added Event messages
36 ' added code for drainage flux
37 ' added pump control code
38 '
39 ' V1_0 modified by Brad Lyles 20210826
40 ' This program was modified from CR3000 program
41 '
42 ' Datalogger Channels used
43 '1H - differential load cell in lysimeter 1
44 '1L - differential load cell in lysimeter 1
45 'VX1 - load cell in lysimeter 1
46 '2H - differential load cell in lysimeter 2
47 '2L - differential load cell in lysimeter 2
48 'VX4 - load cell in lysimeter 1
49 '6H - differential thermocouple in lysimeter 1
50 '6L - differential thermocouple in lysimeter 1
51 '7H - differential thermocouple in lysimeter 2
52 '7L - differential thermocouple in lysimeter 2
```

Program: Lys_V1_2e_example.CR1X

```
53  '8H  - differential thermocouple in discharge area
54  '8L  - differential thermocouple in discharge area
55  '
56  'C1  - CS451 pressure transducers 1
57  'C2
58  'C3  - CS451 pressure transducer 2
59  'C4
60  'C5  - CS655 soil moisture sensors 1 & 2 in Lysimeter 1
61  'C6
62  'C7  - CS655 soil moisture sensors 3 & 4 in Lysimeter 2
63  'C8
64  '
65
66  Const PI = 3.141592654
67
68  Public PTemp, Batt_volt
69  Public Tare1 As Boolean
70  Public Tare2 As Boolean
71  Public Calib As Boolean
72  Public reset_Stage1 As Boolean
73  Public reset_Stage2 As Boolean
74  Public manual_reset_Stage1 As Boolean
75  Public manual_reset_Stage2 As Boolean
76
77  Public SiteID As String
78
79  Public File_Current As Long, CloseStat
80  Public LastUpdate As String * 35, Readfile_result, Writefile_result
81  Dim time(9)
82  Public Update_time As String * 35, Update_Current_manual As Boolean
83
84  Public i, i_sensor
85  Public mV1_avg, known1_gm, mV1_SD, LC1_mult
86  Public mV2_avg, known2_gm, mV2_SD, LC2_mult
87
88  Public Lys1_Mass_gm, Tare1_gm, Lys1_Area_cm2, Lys1_ET_Flux_mm, Lys1_Flux_m
89  Public Lys2_Mass_gm, Tare2_gm, Lys2_Area_cm2, Lys2_ET_Flux_mm, Lys2_Flux_m
90
91  Public TC_LoadCell1, TC_LoadCell2, TC_3
92
93  'CS451 pressure transducer
94  Public CS450_1(2), CS450_2(2)
95  Alias CS450_1(1)=Lv11_cm
96  Alias CS450_1(2)=Temp1_C
97  Alias CS450_2(1)=Lv12_cm
98  Alias CS450_2(2)=Temp2_C
99  Units Lv11_cm=cm
100 Units Temp1_C=deg C
101 Units Lv12_cm=cm
102 Units Temp2_C=deg C
103 Public Trans1_mult, Trans2_mult
104
```

```
105 Public Stage1_cm, Stage2_cm
106 Public stage1_datum, stage2_datum, Stage1_trigger_cm, Stage2_trigger_cm
107 Public old_Drain1_ml, old_Drain2_ml, delta_Drain1_ml, delta_Drain2_ml
108 Public Drain1_ml, cum_Drain1_ml, Lys1_Drain_Flux_mm
109 Public Drain2_ml, cum_Drain2_ml, Lys2_Drain_Flux_mm
110 Public Pump1_time_sec, Pump2_time_sec
111 Public Pump1_status As Boolean, Pump2_status As Boolean
112 Public Pump1_manual As Boolean, Pump2_manual As Boolean
113 Public Pump1_man_status As Boolean, Pump2_man_status As Boolean
114 Public Pump_Ready As Boolean
115 Public stage1_delay_sec, stage2_delay_sec
116
117
118 'CS655 sensors
119 Public CS655(4,6)
120 Public Soil_VWC(4), Soil_EC(4), Soil_T(4), Soil_P(4), Soil_PA(4), Soil_VR(
121
122 Units Soil_VWC=m^3/m^3
123 Units Soil_EC=dS/m
124 Units Soil_T=Deg C
125 Units Soil_P=unitless
126 Units Soil_PA=nSec
127 Units Soil_VR=unitless
128
129 Public Message As String * 100
130
131 Public mV1(100), mV2(100)
132
133 'Define Data Tables.
134 DataTable (Daily,1,-1)
135     DataInterval (0,1,Day,10)
136     CardOut (0,365)
137     Sample (1,SiteID,String)
138     Average (1,Lys1_Flux_mm,IEEE4,False)
139     Average (1,Lys1_ET_Flux_mm,IEEE4,False)
140     Average (1,Lys1_Drain_Flux_mm,IEEE4,False)
141     Average (1,Soil_VWC(1),FP2,False)
142     Average (1,Soil_VWC(2),FP2,False)
143     Average (1,Lys2_Flux_mm,IEEE4,False)
144     Average (1,Lys2_ET_Flux_mm,IEEE4,False)
145     Average (1,Lys2_Drain_Flux_mm,IEEE4,False)
146     Average (1,Soil_VWC(3),FP2,False)
147     Average (1,Soil_VWC(4),FP2,False)
148 EndTable
149
150 DataTable (FifteenMin,1,-1) 'Set table size to # of records, or -1 to auto
151     DataInterval (0,15,Min,10)
152     CardOut (0,-1)
153     Sample (1,SiteID,String)
154     Average (1,Lys1_Mass_gm,IEEE4,False)
155     Average (1,Lys2_Mass_gm,IEEE4,False)
156     Average (1,Lys1_ET_Flux_mm,IEEE4,False)
```

```

157     Average (1,Lys2_ET_Flux_mm,IEEE4,False)
158     Average (1,Lys1_Drain_Flux_mm,IEEE4,False)
159     Average (1,Lys2_Drain_Flux_mm,IEEE4,False)
160     Average (1,Lys1_Flux_mm,IEEE4,False)
161     Average (1,Lys2_Flux_mm,IEEE4,False)
162     Average (1,TC_LoadCell1,IEEE4,False)
163     Average (1,TC_LoadCell2,IEEE4,False)
164     Average (1,TC_3,IEEE4,False)
165     Average (1,mV1_avg,IEEE4,False)
166     Average (1,mV2_avg,IEEE4,False)
167     Average (1,Lvl1_cm,FP2,False)
168     Average (1,Lvl2_cm,FP2,False)
169     Average (1,Temp1_C,FP2,False)
170     Average (1,Temp2_C,FP2,False)
171     Average (1,Drain1_ml,FP2,False)
172     Average (1,cum_Drain1_ml,FP2,False)
173     Average (1,Drain2_ml,FP2,False)
174     Average (1,cum_Drain2_ml,FP2,False)
175     Average (1,PTemp,FP2,False)
176     Minimum (1,Batt_volt,FP2,False,False)
177     Average (4,Soil_VWC(),FP2,False)
178     Average (4,Soil_EC(),FP2,False)
179     Average (4,Soil_T(),FP2,False)
180     Average (4,Soil_P(),FP2,False)
181     Average (4,Soil_PA(),FP2,false)
182     Average (4,Soil_VR(),FP2,false)
183 EndTable
184
185 DataTable (TenSec,Calib,-1) 'Set table size to # of records, or -1 to auto
186     DataInterval (0,10,Sec,10)
187     Sample (1,SiteID,String)
188     Average (1,mV1_avg,IEEE4,False)
189     Average (1,mV1_SD,FP2,False)
190     Average (1,known1_gm,FP2,False)
191     Sample (1,TC_LoadCell1,IEEE4)
192     Average (1,mV2_avg,IEEE4,False)
193     Average (1,mV2_SD,FP2,False)
194     Average (1,known2_gm,FP2,False)
195     Sample (1,TC_LoadCell2,IEEE4)
196 EndTable
197
198 DataTable (Event_Log,1,240)
199     CardOut (0,2400)
200     Sample (1,Message,String)
201 EndTable
202
203 Sub SendLog (Msg As String *100)
204     Message = Msg
205     CallTable (Event_Log)
206     Message = " "
207 EndSub
208

```

```

209 Function float2string(val) As String * 20
210     Public nchar
211     float2string = val+CHR(32)
212     nchar = Len(float2string)
213 EndFunction
214
215 'Function string2float(val2) As Float
216 '     string2float = val2
217 'EndFunction
218
219 Sub Update_CurrentFile() ' Write current conditions to Example_current.txt
220     File_Current = FileOpen("CPU:Example_current.txt","a",0)
221     RealTime (time(1))
222     Update_time = time(1)+"-"+time(2)+"-"+time(3)+"_"+time(4)+":"+time(5)+":
223     Writefile_result = FileWrite(File_Current,Update_time,18)
224     Writefile_result = FileWrite(File_Current,float2string(LC1_mult),nchar)
225     Writefile_result = FileWrite(File_Current,float2string(LC2_mult),nchar)
226     Writefile_result = FileWrite(File_Current,float2string(Trans1_mult),ncha
227     Writefile_result = FileWrite(File_Current,float2string(Trans2_mult),ncha
228     Writefile_result = FileWrite(File_Current,float2string(Lys1_Area_cm2),nc
229     Writefile_result = FileWrite(File_Current,float2string(Lys2_Area_cm2),nc
230     Writefile_result = FileWrite(File_Current,float2string(Tare1_gm),nchar)
231     Writefile_result = FileWrite(File_Current,float2string(Tare2_gm),nchar)
232     Writefile_result = FileWrite(File_Current,float2string(stage1_datum),nch
233     Writefile_result = FileWrite(File_Current,float2string(stage2_datum),nch
234     Writefile_result = FileWrite(File_Current,float2string(cum_Drain1_ml),nc
235     Writefile_result = FileWrite(File_Current,float2string(cum_Drain2_ml),nc
236     Writefile_result = FileWrite(File_Current,float2string(Stage1_trigger_cm
237     Writefile_result = FileWrite(File_Current,float2string(Stage2_trigger_cm
238     Writefile_result = FileWrite(File_Current,float2string(Pump1_time_sec),n
239     Writefile_result = FileWrite(File_Current,float2string(Pump2_time_sec),n
240     Call SendLog ("Current conditions where updated at CPU:Example_current.t
241     CloseStat = FileClose(File_Current)
242     Update_Current_manual = False
243 EndSub
244
245 Public cf_array(17) As String *20, current_chr As String *2000
246 Public FirstPass As Boolean
247
248 'Main Program
249 BeginProg
250     SiteID = "Example"
251
252     Call SendLog (SiteID)
253     Call SendLog ("Program Restarted")
254
255     File_Current = FileOpen("CPU:Example_current.txt","r",0) ' Read initial c
256     Readfile_result = FileRead(File_Current,current_chr,2000)
257     SplitStr (cf_array(),current_chr," ",17,5)
258     LastUpdate = cf_array(1)
259
260     If (Readfile_result < 1) Then Call SendLog ("Read error from CPU:Example

```

```

261
262     If (Readfile_result > 0) Then
263         LC1_mult = cf_array(2)
264         LC2_mult = cf_array(3)
265         Trans1_mult = cf_array(4)
266         Trans2_mult = cf_array(5)
267         Lys1_Area_cm2 = cf_array(6)
268         Lys2_Area_cm2 = cf_array(7)
269         Tare1_gm = cf_array(8)
270         Tare2_gm = cf_array(9)
271         stage1_datum = cf_array(10)
272         stage2_datum = cf_array(11)
273         cum_Drain1_ml = cf_array(12)
274         cum_Drain2_ml = cf_array(13)
275         Stage1_trigger_cm = cf_array(14)
276         Stage2_trigger_cm = cf_array(15)
277         Pump1_time_sec = cf_array(16)
278         Pump2_time_sec = cf_array(17)
279         Call SendLog ("Initial conditions from CPU:Example_current.txt")
280     Else
281         ' If all else fails, read initial conditions from the program
282         ' Load Cell
283         LC1_mult = 74.806
284         LC2_mult = 63.62
285         ' Stage transducer
286         Trans1_mult = 79.075
287         Trans2_mult = 79.075
288
289         Lys1_Area_cm2 = ((18*2.54)/2)^2 * PI
290         Lys2_Area_cm2 = ((18*2.54)/2)^2 * PI
291
292         Tare1_gm = -38145.14 'values as of 20211118
293         Tare2_gm = -44442.67 'values as of 20211118
294
295         stage1_datum = -1.015824 'values as of 20220103
296         stage2_datum = -1.249384 'values as of 20220103
297
298         cum_Drain1_ml = 2559.415 'values as of 20220103
299         cum_Drain2_ml = 2919.315 'values as of 20220103
300
301         Stage1_trigger_cm = 20
302         Stage2_trigger_cm = 20
303
304         Pump1_time_sec = -10
305         Pump2_time_sec = -10
306
307         Call SendLog ("Initial conditions from program")
308         Call Update_CurrentFile()
309
310     EndIf
311     FileClose(File_Current)
312

```

Program: Lys_V1_2e_example.CR1X

```

313 Call SendLog ("Tare1_gm =" + FormatFloat(Tare1_gm,"%8.1f"))
314 Call SendLog ("Tare2_gm =" + FormatFloat(Tare2_gm,"%8.1f"))
315 Call SendLog ("Stage1 Datum =" + FormatFloat(stage1_datum,"%5.2f"))
316 Call SendLog ("Stage2 Datum =" + FormatFloat(stage2_datum,"%5.2f"))
317
318 FirstPass = True
319
320 Scan (10,Sec,10,0)
321
322 PanelTemp (PTemp,60)
323
324 If (Update_Current_manual) Then Call Update_CurrentFile()
325
326 If (TimeIntoInterval (0,15,Min)) Then Pump_Ready = True
327 If (Batt_volt < 12) Then Pump_Ready = False 'Don't run on pump when b
328
329 stage1_delay_sec = Timer(1,Sec,4)
330 stage2_delay_sec = Timer(2,Sec,4)
331
332 'Measure load cells
333 For i=1 To 100
334 BrFull (mV1(i),1,AutoRange,1,Vx1,1,4000,True ,True ,0,250,1000.,0.0)
335 BrFull (mV2(i),1,AutoRange,2,Vx4,1,4000,True ,True ,0,250,1000.,0.0)
336 Next i
337
338 'Compute load cell spatial statistics
339 AvgSpa (mV1_avg,100,mV1())
340 StdDevSpa (mV1_SD,100,mV1())
341 AvgSpa (mV2_avg,100,mV2())
342 StdDevSpa (mV2_SD,100,mV2())
343
344 'Measure lysimeter drainage
345 'CS451 SDI-12 pressure transducer
346 SDI12Recorder(Lvl1_cm,C1,"0","M1!",1,0,-1)
347 Lvl1_cm=Lvl1_cm*70.307
348 SDI12Recorder(Lvl2_cm,C3,"0","M1!",1,0,-1)
349 Lvl2_cm=Lvl2_cm*70.307
350
351 Stage1_cm = Lvl1_cm - stage1_datum ' sn 20012329
352 Stage2_cm = Lvl2_cm - stage2_datum ' sn
353
354 'Drainage Calcs
355 ' compute volume of fluid in drainage stilling well, with the pump
356 ' and transducer installed.
357 Drain1_ml = Stage1_cm * Trans1_mult
358 Drain2_ml = Stage2_cm * Trans2_mult
359
360 If (FirstPass) Then
361 old_Drain1_ml = Drain1_ml
362 old_Drain2_ml = Drain2_ml
363 Call SendLog ("Initial Drain1_ml =" + FormatFloat(Drain1_ml,"%5.2f"))
364 Call SendLog ("Initial Drain2_ml =" + FormatFloat(Drain2_ml,"%5.2f"))

```

```

365     'reset_Stage1 = True
366     'reset_Stage2 = True
367 EndIf
368
369     ' Reset stage datum and drainage volume datum
370 If (reset_Stage1 = True AND stage1_delay_sec > 20 OR manual_reset_Stag
371     reset_Stage1 = False
372     stage1_datum = Lvl1_cm
373     Call SendLog ("Stage 1 datum was reset =" + FormatFloat(stage1_datum
374     Stage1_cm = Lvl1_cm - stage1_datum
375     Drain1_ml = Stage1_cm * Trans1_mult
376     old_Drain1_ml = Drain1_ml
377     Call SendLog ("old Drain 1(ml) was reset =" + FormatFloat(Drain1_ml,
378     Timer(1,Sec,1)
379     Call Update_CurrentFile()
380 EndIf
381 If (reset_Stage2 = True AND stage2_delay_sec > 20 OR manual_reset_Stag
382     reset_Stage2 = False
383     stage2_datum = Lvl2_cm
384     Call SendLog ("Stage 2 datum was reset =" + FormatFloat(stage2_datum
385     Stage2_cm = Lvl2_cm - stage2_datum
386     Drain2_ml = Stage2_cm * Trans2_mult
387     old_Drain2_ml = Drain2_ml
388     Call SendLog ("old Drain 2 (ml) was reset =" + FormatFloat(Drain2_ml
389     Timer(2,Sec,1)
390     Call Update_CurrentFile()
391 EndIf
392
393     ' Add new drainage volume to previously accumulated drainage.
394 delta_Drain1_ml = Drain1_ml - old_Drain1_ml
395 If (delta_Drain1_ml > 5) Then
396     cum_Drain1_ml = cum_Drain1_ml + delta_Drain1_ml
397     old_Drain1_ml = Drain1_ml
398     Call SendLog ("Cum Drain 1 =" + FormatFloat(cum_Drain1_ml,"%5.2f"))
399     Call Update_CurrentFile()
400 EndIf
401 delta_Drain2_ml = Drain2_ml - old_Drain2_ml
402 If (delta_Drain2_ml > 5) Then
403     cum_Drain2_ml = cum_Drain2_ml + delta_Drain2_ml
404     old_Drain2_ml = Drain2_ml
405     Call SendLog ("Cum Drain 2 =" + FormatFloat(cum_Drain2_ml,"%5.2f"))
406     Call Update_CurrentFile()
407 EndIf
408
409     ' Compute drainage flux
410 Lys1_Drain_Flux_mm = cum_Drain1_ml / Lys1_Area_cm2 * 10 'gm * 1 cc/gm
411 Lys2_Drain_Flux_mm = cum_Drain2_ml / Lys2_Area_cm2 * 10
412
413     ' Pump control
414     ' turn on Pump#1 if Stage1 > Stage1_trigger_cm
415     ' check every 15 minutes
416 If (Pump_Ready = True AND Stage1_cm > Stage1_trigger_cm AND Pump1_stat

```



```

417     SW12 (SW12_1,1 )
418     Pump1_status = True
419     Call SendLog ("Pump 1 turned ON because stage was greater than stage
420     Call SendLog ("Stage 1 (cm) =" + FormatFloat(Stage1_cm,"%5.2f"))
421     Call SendLog ("Stage 1 trigger (cm) =" + FormatFloat(Stage1_trigger_
422 EndIf
423 If (Pump1_manual = True AND Pump1_man_status = False) Then
424     SW12 (SW12_1,1 )
425     Pump1_man_status = True
426     Call SendLog ("Pump 1 manually turned ON")
427 End If
428 If (Pump1_manual = False AND Pump1_man_status = True) Then
429     SW12 (SW12_1,0 )
430     Pump1_man_status = False
431     Call SendLog ("Pump 1 manually turned OFF")
432 End If
433
434 If (Pump1_status = True) Then
435     Pump1_time_sec = Pump1_time_sec + 10
436     If(Pump1_time_sec > 20 OR Stage1_cm < 5) Then
437         SW12 (SW12_1,0)
438         Call SendLog ("Pump 1 turned Off")
439         Call SendLog ("Pump 1 time(sec) =" + FormatFloat(Pump1_time_sec,"%
440         Call SendLog ("Stage 1 (cm) =" + FormatFloat(Stage1_cm,"%5.2f"))
441         Pump1_status = False
442         Pump1_time_sec = -10
443         reset_Stage1 = True
444         Timer(1,Sec,2)
445     EndIf
446 EndIf
447 ' turn on Pump#2 if Stage2 > Stage2_trigger_cm
448 ' check every 15 minutes
449
450 If (Pump_Ready = True AND Stage2_cm > Stage2_trigger_cm AND Pump2_stat
451     SW12 (SW12_2,1 )
452     Pump2_status = True
453     Call SendLog ("Pump 2 turned ON because stage was greater than stage
454     Call SendLog ("Stage 2 (cm) =" + FormatFloat(Stage2_cm,"%5.2f"))
455     Call SendLog ("Stage 2 trigger (cm) =" + FormatFloat(Stage2_trigger_
456 EndIf
457 If (Pump2_manual = True AND Pump2_man_status = False) Then
458     SW12 (SW12_2,1 )
459     Pump2_man_status = True
460     Call SendLog ("Pump 2 manually turned ON")
461 End If
462 If (Pump2_manual = False AND Pump2_man_status = True) Then
463     SW12 (SW12_2,0 )
464     Pump2_man_status = False
465     Call SendLog ("Pump 2 manually turned OFF")
466 End If
467 If (Pump2_status = True) Then
468     Pump2_time_sec = Pump2_time_sec + 10

```

```

469     If (Pump2_time_sec > 20 OR Stage2_cm < 5) Then
470         SW12 (SW12_2,0)
471         Call SendLog ("Pump 2 turned Off")
472         Call SendLog ("Pump 2 time (sec) =" + FormatFloat(Pump2_time_sec,"
473         Call SendLog ("Stage 2 (cm) =" + FormatFloat(Stage2_cm,"%5.2f"))
474         Pump2_status = False
475         Pump2_time_sec = -10
476         reset_Stage2 = True
477         Timer(2,Sec,2)
478     EndIf
479 EndIf
480
481
482 'Measure lysimeter soil moisture content
483 'CS655 sensors
484 If TimeIntoInterval(0,15,Min) Then
485     SDI12Recorder(CS655(1,1),C5,"1","M4!",1,0,-1) 'Sand lower
486     SDI12Recorder(CS655(2,1),C5,"2","M4!",1,0,-1) 'Sand upper
487     SDI12Recorder(CS655(3,1),C7,"3","M4!",1,0,-1) 'Soil lower
488     SDI12Recorder(CS655(4,1),C7,"4","M4!",1,0,-1) 'Soil upper
489 EndIf
490 For i_sensor = 1 To 4
491     Soil_VWC(i_sensor) = CS655(i_sensor,1)
492     Soil_EC(i_sensor) = CS655(i_sensor,2)
493     Soil_T(i_sensor) = CS655(i_sensor,3)
494     Soil_P(i_sensor) = CS655(i_sensor,4)
495     Soil_PA(i_sensor) = CS655(i_sensor,5)
496     Soil_VR(i_sensor) = CS655(i_sensor,6)
497 Next i_sensor
498
499 'Measure temperatures
500 TCDiff (TC_LoadCell1,1,mV200C,6,TypeT,PTemp,True ,0,250,1.0,0)
501 TCDiff (TC_LoadCell2,1,mV200C,7,TypeT,PTemp,True ,0,250,1.0,0)
502 TCDiff (TC_3,1,mV200C,8,TypeT,PTemp,True ,0,250,1.0,0)
503
504 'Tare lysimeters
505 Lys1_Mass_gm = mV1_avg * LC1_mult - Tare1_gm
506 If (Tare1 = TRUE) Then
507     Tare1_gm = mV1_avg * LC1_mult
508     Tare1 = False
509     Call SendLog ("Tare Lys 1 was reset")
510     Call SendLog ("Tare 1 (gm) =" + FormatFloat(Tare1_gm,"%8.1f"))
511     Call Update_CurrentFile()
512 EndIf
513
514 Lys2_Mass_gm = mV2_avg * LC2_mult - Tare2_gm
515 If (Tare2 = TRUE) Then
516     Tare2_gm = mV2_avg * LC2_mult
517     Tare2 = False
518     Call SendLog ("Tare Lys 2 was reset")
519     Call SendLog ("Tare 2 (gm) =" + FormatFloat(Tare2_gm,"%8.1f"))
520     Call Update_CurrentFile()

```

Program: Lys_V1_2e_example.CR1X

```
521     EndIf
522
523     'Compute lysimeter mass flux
524     Lys1_Flux_mm = Lys1_Mass_gm / Lys1_Area_cm2 * 10 'gm * 1 cc/gm / cm^2
525     Lys2_Flux_mm = Lys2_Mass_gm / Lys2_Area_cm2 * 10
526
527     'Compute lysimeter ET flux
528     Lys1_ET_Flux_mm = Lys1_Flux_mm + Lys1_Drain_Flux_mm
529     Lys2_ET_Flux_mm = Lys2_Flux_mm + Lys2_Drain_Flux_mm
530
531     If (FirstPass) Then
532
533         Call SendLog ("Initial Cum_Drain1_ml =" + FormatFloat(cum_Drain1_ml,
534         Call SendLog ("Initial Cum_Drain2_ml =" + FormatFloat(cum_Drain2_ml,
535     EndIf
536
537     Battery (Batt_volt)
538     FirstPass = False
539
540     CallTable TenSec 'only written during calibration
541     CallTable FifteenMin
542     CallTable Daily
543
544     Pump_Ready = False
545
546     NextScan
547 EndProg
```