PROGRAM LIGHTING.8xp for the TI-84 Family of Calculators

ClrHome	Clear the screen and the CBL2 or LabPro
Send({0})	
Send({2001,0})	
Float	Format number display
2000üD	Set delay counter, D
1.5üG	Set power of bulb, G
{1999,2000,80,800,40}üáSOUNA	Define 2 sounds
{1999,1000,390}üáSOUNB	
DelVar áZ	Initialize list of 20 zeros
20üdim(áZ)	
Output(1,2,"** LIGHTING **")	Welcome Screen
Output(3,1,"CONTROL 4 LIGHTS")	
Output(7,1,"TI84 CBL2/LABPRO")	
Output(8,1,"<ENTER> TO START")	
Pause	
ClrHome	
Lbl ME	**Main Menu**
Menu("LIGHTING","CHECK FULL",CF,"INPUT LEVELS",IL,"AUTOMATIC",AU,"GRAPH",GR,"CLEAR DATA",CD,"QUIT",QU)	
Lbl CF	**Check Full Lighting Values**
ClrHome	Display CF Instructions
Output(1,1,"CHECK FULL")	
Output(3,1,"USE UP AND DOWN")	
Output(4,1,"ARROW KEYS TO")	
Output(5,1,"TURN ON OR OFF")	
Output(7,1,"E =")	
Output(8,1,"<ENTER> FOR MENU")	
0üK	Initialize constants, K and O
KüO	
Send({0})	Set CBL2 output to zero and start data collection on analog port 1.
Send({102,ú1})	
Send({2011,0})	
Send({1,1,1})	
Send({3,1,ú1,0})	
While Kø105	Loop until <ENTER>
If K=25	If UP arrow was pressed, turn on the lights
Send({2011,15})	
If K=34	If DOWN arrow was pressed, turn off the light.
Send({2011,0})	
Get(áE1)	Get and display data from analog port 1
Output(7,5," ")	
Output(7,5,round(áE1(1),4)	
getKeyüK	Check for key press
End	End While loop
Send({2011,0})	Zero output and return to main menu
Goto ME	

Lbl IL	**INPUT LIGHTING LEVELS**
ClrHome	Input number of lights, L, and force it to an integer from 0 to 4.
Output(6,1,"INPUT INTEGER")	
Output(7,1,"FROM 0 TO 4")	
Send(áSOUNA)	
Input "NUMBER OF LIGHTS",L	
max(0,min(4,int(L)))üL	
ClrHome	Input power level, P.
Output(6,1,"INPUT PERCENT")	
Output(7,1,"FROM 0 TO 100")	
Send(áSOUNA)	
Input "PERCENT POWER ? ",P	
max(0,min(20,int(P/5+.5)))üJ	Find J, power in “twentieths.”
ClrHome	Set up data display screen.
Output(1,3,"INPUT LEVELS")	
Output(3,3,"LIGHTS")	
Output(3,1,L)	
Output(4,1,J*5)	
Output(4,4,"% POWER")	
Output(5,1,"E =")	
If L=1	Display number of lights, L, and set S to provide the binary code which will display that number of lights.
Then	
Output(3,8," ")	
1üS	
End	
If L=4	
15üS	
If L=3	
7üS	
If L=2	
5üS	
20üdim(áH)	Initialize ₗH and ₗI as 20-element lists of the value, S.
Fill(S,áH)	
áHüáI	
If J=0	If power level is zero, set output list, ₗH, to contain 20 zeros.
Then	
áZüáH	
Else	If power level is not zero, put J elements with value S into list ₗH and fill the rest of the 20 elements with zeros. Intermingle the ON and OFF values to avoid flicker.
Jüdim(áH)	
augment(áH,áZ)üáI	
20üdim(áH)	
{1,3,5,7,9,11,13,15,17,19,2,4,6,8,10,12,14, 16,18,20}üáSORT	
SortA(áSORT,áH)	
End	
augment({1,31,20},áH)üáH	Add CBL2 instructions
Send({102,ú1})	Set constant power on for CBL2
Send({1,1,1})	Activate analog port 1 and digital port.
Send(áH)	
Send({3,.0002,ú1,0})	Start the lights and data collection
Output(7,1,"<ENTER> FOR MENU")	Display stop instructions.
Output(8,1,"OTHER TO REPEAT")	

0üK	While no key is pressed
While K=0	
Get(áE1)	Get and display data from analog port 1.
Output(5,5," ")	
Output(5,5,round(áE1(1),4)	
getKeyüK	
End	
Send({0})	Stop and reset the CBL2
Send({2001,0})	
If K=105	If <ENTER> was pressed, go to the main menu.
Goto ME	
Goto IL	Restart “Input Levels”
Lbl AU	**AUTOMATIC CONTROL**
ClrHome	Clear screen
Send({1,1,1})	Start data collection on analog port 1, autoidentifying the probe.
Send({3,.001,10,0})	
Get(áE1)	Get the data value from port 1.
mean(áE1)üE	
Send({8,1,0})	Get the probe identity and store it as Q.
Get(áE1)	
áE1(1)üQ	
Send({0})	Start continuous data collection with the lights off.
Send({102,ú1})	
Send({1,1,Q})	
Send({1,31,1,0})	
Send({3,.0002,ú1,0})	
Output(5,1,"E =")	Display data from analog port 1.
Output(5,5,round(E,4))	
Input "DESIRED E ? ",B	Input target level and tolerance.
Input "TOLERANCE ? ",C	
ClrHome	Setup display screen for data collection.
Output(1,3,"AUTO PROGRAM")	
Output(2,1,"B= 0000 C= 0000")	
Output(2,3,B)	
If Cù.001	
Then	
Output(2,11,C)	
Else	
round(C,4)üC	
Output(2,11,".")	
Output(2,15,int(C*10000))	
End	
Output(3,1,"E =")	
Output(4,3,"LIGHTS")	
Output(5,4,"% POWER")	
Output(6,1,"CYCLES")	
Output(7,1,"T = SECONDS")	
Output(8,2,"<ENTER> TO STOP")	
0üA	Initialize counters and data values.
0üK	
2üN	
0üL	
0üP	

{E}üáE	Store initial data values as the first element in data lists.
{0}üáP	
{0}üáL	
{0}üáT	
Send({0})	Start data continuous data collection with the lights off.
Send({102,ú2})	
Send({1,1,Q})	
Send({1,31,1,0})	
Send({3,.0002,ú1,0})	
startTmrüU	Initialize the calculator's timer (only for TI-84 family calculators).
While Kø105	While <ENTER> is not pressed
For(I,1,D)	Delay loop (D is set at the start of the program)
End	
Get(áE1)	Get data from analog port 1
áE1(1)üE	
checkTmr(U)üT	Read elapsed time as T
Output(3,5," ")	Display E and T
Output(3,5,round(E,4)	
Output(7,5,T)	
If abs(E-B)<C and A<5	If light level is in tolerance AND data was recently recorded
Then	Increment in-tolerance counter.
A+1üA	
Else	
EüáE(N)	Store E and T to lists
TüáT(N)	
prgmLIGHTSUB	Get updates from LIGHTSUB
max(0,min(20,int(P/5+.5)))üJ	Find J as power in twentieths and round P to nearest 5%.
5*JüP	
max(0,min(int(L+.5),4))üL	Force L to be 0, 1, 2, 3 or 4
PüáP(N)	Store new P and L to lists
LüáL(N)	
Output(6,8,N)	Display new cycle count
0üA	Reset in-tolerance counter
If PøáP(N-1) or LøáL(N-1)	If power level, P, has changed
Then	
Output(5,2," ")	Display the new values for power and number of lights.
Output(5,1,P)	
Output(4,1,L)	
If J=0	If new power level is zero, set signal list to 20 zeros
Then	
áZüáH	
Else	If new power level is >0
If L=1	Set the correct binary output pattern
1üS	
If L=2	
5üS	
If L=3	
7üS	
If L=4	
15üS	

20üdim(áH)	Create an intermingled list of output signals with J out of 20 elements ON with signal S.
Fill(S,áH)	
áHüáI	
Jüdim(áH)	
augment(áH,áZ)üáI	
20üdim(áH)	
{1,3,5,7,9,11,13,15,17,19,2,4,6,8,10,12,14,16,18,20}üáSORT	
SortA(áSORT,áH)	
End	
augment({1,31,20},áH)üáH	Implement the new signal list
Send(áH)	
Send({2001,0})	
Send(áSOUNB)	Sound to indicate lights changed
Send({3,.0002,ú1,0})	Restart the lights
End	End change lights procedure
N+1üN	Increment data counter
End	End record data procedure
getKeyüK	Check for key press
End	End the main while loop
Get(áE1)	Get a final data point
Send({0})	Turn off the CBL2 and lights.
Send({2001,0})	
áE1(1)üáE(N)	Record final data points
PüáP(N)	
LüáL(N)	
checkTmr(U)üáT(N)	Get time at shut-down
¾List(áT)üáDT	Calculate W, the total energy consumed.
augment({0},áDT)üáDT	
L*sum(áP*áDT)/100*GüW	
ClrHome	Display energy, power and time
Output(1,1,"ENERGY USED =")	
Output(2,3,round(W,3))	
Output(2,14,"J")	
Output(4,1,"TIME = sec")	
Output(4,7,T)	
Output(5,1,"AVERAGE POWER =")	
Output(6,3,round(W/T,3))	
Output(6,14,"W")	
Output(8,1,"<ENTER> TO CONT.")	
Pause	
Goto ME	Return to main menu

Lbl GR	**DISPLAY GRAPH**
max(áT)üT	Plot illumination (E) vs time (T) and show the upper and lower tolerances.
Plot1(xyLine,áT,áE,Ò)	
áTüáH	
Fill(B+C,áH)	
Plot2(xyLine,áT,áH,Ò)	
áH-2*CüáI	
Plot3(xyLine,áT,áI,Ò)	
0üXmin	Set the display window for the graph.
0üYmin	
TüXmax	
max(max(áE),2*C+B)üYmax	
10üXscl	
.1üYscl	
PlotsOn	Display the graph in trace mode
Trace	
Goto ME	Return to main menu
Lbl CD	**CLEAR DATA LISTS**
DelVar áT	Delete the data and other lists used to produce the graph.
DelVar áE	
DelVar áL	
DelVar áP	
DelVar áH	
DelVar áI	
ClrHome	Return to main menu when <ENTER> is pressed.
Output(2,1,"DATA DELATED")	
Output(8,1,"<ENTER> TO CONT.")	
Pause	
Goto ME	
Lbl QU	**QUIT PROGRAM**
ClrHome	Clear Screen
0üA	Zero all constants used. (This is probably not needed, but it helps keep track of what constants have been changed by the program.)
0üB	
0üC	
0üD	
0üE	
0üG	
0üJ	
0üK	
0üL	
0üM	
0üN	
0üP	
0üQ	
0üS	
0üT	
0üU	
0üW	
Send({2001,0})	Make sure the lights are off.

DelVar áE1	Delete the lists used EXCEPT those required for graphs or for further data analysis.
DelVar áDT	
DelVar áZ	
DelVar áSOUNA	
DelVar áSOUNB	
DelVar áSORT	
Send({0})	Shut down the CBL2.

Constants used in the program above:

A Counter in automatic mode to allow data records while within tolerance.

B Illumination target level

C Tolerance for illumination target. (The range above or below "B" within which no adjustments are needed.)

D Delay counter (Set near the start of the program, D can be decreased for more rapid updates, but this also means more data will be recorded, resulting in larger data lists. D should be increased for very long recording sessions.)

E Most recent data from analog port 1. This is normally luminance from a light probe, but it can be data from any auto-identifiable probe. E is recorded to list $_L$E when P or L changes or (less often) when E is within tolerance.

G Is the power (in watts) for each light bulb. It is set initially as 1.5 near the start of the program, corresponding to a 6-volt, 250 mA bulb. The value can be modified for different equipment. It is used only to calculate the total energy consumed.

J Power level in "twentieths." J is calculated from P and J/20 is the duty cycle for current to the bulbs. Electricity is delivered J/20 of the time, turning on and off too rapidly to produce a visible change in the light intensity.

K Last key pressed.

L Number of light bulbs to be illuminated. The number can have any real value when set from the key pad or the subprogram LIGHTSUB, but the main program always converts L to the nearest integer, 0, 1, 2, 3 or 4.

M A counter. It can also be used for internal counting within LIGHTSUB, but the main program will set M equal to D+1.

N Index to the data stored in lists. E(N) is the lasted light measure, P(N) is the latest power setting, L(N) is the latest number of bulbs illuminated, and T(N) is the time of the last recording.

P Power setting as a percent. For example, "50" (with NO % symbol) means 50%. P can have any real value when set from the keypad or the subprogram LIGHTSUB, but the main program always converts P to an integer from 0 to 100 in step of 5. (For example, 0, 5, 10, 15 … 100.)

Q Auto-identification code for the probe in analog port 1. It is 12 for an old-fashioned light probe.

S Decimal expression of the binary signal to turn on the lights. "15," for example, is binary 1111 and turns on all 4 lights. "5" is binary 0101 and turns on 2 of the 4 bulbs.

T Elapsed time in seconds from the calculator's timer. T is recorded in list $_L$T only when there is a change in P or L or when E has remained within tolerance for a while.

U Start time for the calculator's timer. It doesn't mean much, but be careful not to change it or T (elapsed time) will be incorrect.

W Total energy consumed during automatic control, in joules. It is calculated for each time interval using ΔT x P/100 x L x G (time interval x fractional power x number of bulbs x wattage of each bulb). The calculation is done at the end of the data collection so W is not available for use in LIGHTSUB.

Constants NOT Used in the Program LIGHTING.8xp:

Constants on the TI-83 and TI-84 family of calculators are very limited, since all constants

(a) can have only 1 letter ("ET," for example, means *E times T*, not a new constant), and

(b) act as universal, not local, variables (Changing the value of E in subprogram LIGHTSUB, for example, also changes its value in LIGHTING or for any other use.)

The constants F, H, I, O, R, V, X, Y and Z are NOT used by the LIGHTING program, which means they CAN be used freely in your subprogram LIGHTSUB without fear of any conflict.

Lists Used in the Program LIGHTING.8xp:

The most recent version of each of the 6 lists below is kept in memory after you quit the LIGHTING program **unless** you select the "CLEAR DATA" option before quitting. This makes it possible to continue working with the lists on your calculator or to transfer them to a computer for further analysis.

$_L$T(N) Time in seconds at which each change in the setting for P or L occurred. A limited number of additional data records are made while the illumination is within tolerance even if P or L does not change.

$_L$E(N) Illumination level recorded on analog port 1 just BEFORE the value of L or P was changed.

$_L$P(N) Power level as a percentage that WILL BE implemented as a result of the calculation in LIGHTSUB. {$_L$P(N-1) is the previous power level which resulted in illumination level $_L$E(N)}

$_L$L(N) Number of bulbs that WILL BE illuminated as a result of any calculations made in LIGHTSUB. {$_L$L(N-1) is the previous number of bulbs which produced illumination level $_L$E(N)}

$_L$H(N) List with all elements equal to the *lower* tolerance limit, for use in producing the graph.

$_L$I(N) List with all elements equal to the *upper* tolerance limit, for use in producing the graph.

Six other lists ($_L$E1, $_L$DT, $_L$Z, $_L$SOUNA, $_L$SOUNB and $_L$SORT) are used internally by the LIGHTING program and are always deleted when you quit the program in order to save memory.