



# **GRAFIK 6000<sup>®</sup>**

**LIGHTING CONTROL SYSTEM**

## **RS-232 Protocol**

**Command Set to Communicate With  
the Central Processor Panel Via RS-232**

**LUTRON<sup>®</sup>**



# Table of Contents

<b>General Information .....</b>	<b>3</b>	<b>Timeclock Commands .....</b>	<b>10</b>
Communication Settings .....	3	DISABLE UNTIL ENABLE ISSUED .....	10
Preface Every Command With .....	3	DISABLE UNTIL END OF DAY OR ENABLE ISSUED .....	10
System Area Numbers (SYS AREA) .....	4	RE-ENABLE WITH COMPLETE CATCHUP ...	10
System Scene Numbers (SYS SCENE) .....	4	RE-ENABLE AND DO PREVIOUS EVENT .....	11
System Zone Numbers (SYS ZONE) .....	4	STANDARD RE-ENABLE .....	11
Intensity (INT) .....	4	SET TIMECLOCK .....	12
Cycle (CYCLE) .....	4		
Command Structure .....	4	<b>Wallstation Commands.....</b>	<b>13</b>
System Responses to Commands .....	5	ENABLE WALLSTATION .....	13
Response Strings .....	5	DISABLE WALLSTATION .....	13
End of Response .....	5	SIMULATE WALLSTATION SWITCH PRESS .....	13
<b>Zone Commands .....</b>	<b>6</b>	SIMULATE WALLSTATION SWITCH RELEASE .....	14
HALT .....	6	RUN SCRIPT(S) .....	14
TOGGLE .....	6	SET SYSTEM VARIABLE .....	14
FADE TO LEVEL (preset) .....	6	TOGGLE SYSTEM VARIABLE .....	14
RAMP UP .....	6		
RAMP DOWN .....	7	<b>System Monitor Commands .....</b>	<b>15</b>
POMMEL or PULSE .....	7	SYSTEM RESPONSES TO STATUS REQUESTS .....	15
MULTI LEVEL (FLASH or WARN or CHASE) .....	7	AREA STATUS .....	16
RAMP SCENE ZONES UP .....	8	TIMECLOCK STATUS .....	16
RAMP SCENE ZONES DOWN .....	8	WALLSTATION STATUS .....	17
STOP SCENE RAMP UP .....	8	SWITCH STATUS .....	17
STOP SCENE RAMP DOWN .....	8	ZONE INTENSITY .....	17
		TIME NOW .....	18
<b>Scene Commands .....</b>	<b>9</b>	ASTRO_TIMES .....	18
SELECT SYSTEM SCENE(S) .....	9	DATE .....	18
SELECT SYSTEM SCENE(S) USING (DEFINED) DEFAULT TIMES .....	9	CODE REV LEVEL .....	18
OUTPUT TEMP SCENE .....	9	BOOT CODE REV LEVEL .....	19
SELECT SCENE IN AREA .....	9	SERIAL NUMBER .....	19
		SYSTEM VARIABLE VALUE .....	19



## General Information

This document describes the commands available to monitor and operate the GRAFIK 6000 system through either the GRAFIK 6000 panel's local RS-232 port or OMX-RS232 interface wallstation(s).

### Communication Settings

To configure your device to talk to the GRAFIK 6000 processor panel, use the data conventions listed below. The GRAFIK 6000 panel follows the **DCE pinout convention**:

	<b>Pinout GRAFIK 6000</b>	<b>Pinout OMX-RS232</b>
<b>9600 BAUD</b>	9-pin "D" connector	pin 1 Common
<b>8 DATA</b>	pin 2 TX Data from 6000	pin 2 RS-232 Data In
<b>1 STOP</b>	pin 3 RX Data to 6000	pin 3 RS-232 Data Out
<b>NO PARITY</b>	pin 5 Common	

For the OMX-RS232 wallstation, BAUD rates of 2400, 19.2K, and 38.4K, in addition to 9600, are available. Use the DIP switches on the wallstation to select the appropriate rate as shown below. Refer to the OMX-RS232 wallstation's instruction sheet.

<b>BAUD</b>	<b>SW1</b>	<b>SW2</b>
2400	ON	ON
9600	ON	OFF
19.2K	OFF	ON
38.4K	OFF	OFF

Use a 9-pin cable for connection to the RS-232 jack inside the GRAFIK 6000 panel. For connection to the OMX-RS232 wallstation use the 9-pin cable supplied.

If you wish to send these commands from a PC to the GRAFIK 6000 processor, run the Microsoft Windows® Terminal program or an equivalent program. Then, select Local Echo and Line Feed on Carriage Return inbound and outbound. This allows you to see the characters that you are typing as well as keep the responses from overwriting typed characters. NOTE: Refer to the OMX-RS232 hardware instruction sheet for power requirements.

### Preface Every Command With

ALL COMMANDS BELOW ARE PRECEDED WITH THE FIVE-CHARACTER COMMAND STRING PREFIX: "**~11h**"

The HEX equivalent of the "**~11h**" string is:

```

0x7E(~)
0x31(1)
0x31(1)
0x68(h)
0x20(space)

```

The "~" clears the buffer in the GRAFIK 6000 processor and prepares it to receive commands. The "11" is a constant. The "h" indicates that all commands and data following this command string will be in the hexadecimal format. Optionally, a "d" could be used for the commands and data to be entered in the decimal format. The decimal option is only available for communications via the GRAFIK 6000 panel's local RS-232 port.



### System Area Numbers (SYS AREA)

A standard system has up to 32 areas, starting with area zero and ending with area 31 (\$1F). Some systems may have more or fewer areas as needed on a custom basis. (See job-specific database drawings for specific area data.)

### System Scene Numbers (SYS SCENE)

The system scene numbers start at zero and increment in number. The first area's first scene will be scene zero and the scene numbers increment in number. If there are unused scenes in areas, you will still have to account for them when calculating the system scene number you want to assign.

### System Zone Numbers (SYS ZONE)

The system zone numbers start at zero and continue up to 511 (\$1FF) (512 Zones maximum). The first area's first zone will be zone zero and the zone numbers continue upwards. If there are unused zones in areas, you will still have to account for them when calculating the System Zone Number you want to affect.

### Intensity (INT)

Intensity information can range from \$0 to \$7F. For all load types, a \$0 turns it OFF and opens the air gap relay. A value of \$1 closes the relay and keeps a minimum light level. A value of \$7F turns it ON to MAX.

### Cycle (CYCLS)

A Cycle is defined as 0.1 seconds. Cycles are important when working with Fade times and Delay times. Ten seconds would be equivalent to a value of 100 (\$64) cycles.

### Command Structure

All data values are 16 bits (0-\$FFFF) or (0-65,535) and are separated by spaces (\$20). **Note:** Leading zeros do not have to be entered.

Each command has the structure:

**~11h (command # in HEX) (device list or data in HEX) (<cr>)**

The word (device list or data in HEX) is a list of one or more items; either Zones, Scenes, or System Areas that are to receive this command. The execution of the command is stopped when an item encountered is higher than the maximum item defined in the system.

More than one command can be specified in an input string by using a separator (\$FFFF)

*Example:*   **~11h (CMD) (device list or data in HEX) \$FFFF (CMD) (device list or data in HEX) (<cr>)**

**Note:** FOR THE OMX-RS232 WALLSTATION, THE COMMAND STRING MAY CONTAIN AT MOST 30 CHARACTERS.





## Zone Commands

### HALT

HALT permanently stops any Zone dynamics. It freezes the Zone's output level until another command affects the Zone.

(CMD #) (ZONE(S)) <cr>

hex: (\$01)

decimal: ( 1)

Example: To freeze system Zones 7, 8, and 11, the command string is: ~11h 1 7 8 B <cr>

### TOGGLE

If the Zone Output Level is ON (at any Level 1-\$7F) it turns the Zone OFF (\$00). If the Zone Output Level is OFF, it turns the Zone Level ON to MAX (\$7F). After a TOGGLE, the Zone is in steady state. The transition will take place over a period of "Fade" cycles.

(CMD #) (Fade) (ZONE(S)) <cr>

hex: (\$06) (cycls)

decimal: ( 6) (cycls)

Example: To toggle system Zones 7, 8, and 10 with a 2 second (20 cycles) fade time, the command string is:  
~11h 6 14 7 8 A <cr>

### FADE TO LEVEL (preset)

This is a basic command of the GRAFIK 6000 processor's preset capability. It allows individual Zones to be controlled without having to create scenes. All the Zones in the Zone list will fade from their current intensity to the new level in "Fade" cycles. The transition starts after "Delay" cycles.

(CMD #) (Level) (Fade) (Delay) (ZONE(S)) <cr>

hex: (\$07) (0-7F) (cycls) (cycls) (Sys Zone list)

decimal: ( 07)

Example: To send system Zones 10, 11 to \$7F intensity with a fade of 5 seconds (50 cycles), the command string is:  
~11h 7 7F 32 0 A B <cr>

### RAMP UP

The InitialRate is used for the first five cycles then the repeat rate is used. The actual Step Sizes are divided by \$100 then added to the Zone's intensity value every 0.1 second. For example, a Step Size of \$200 causes an intensity change of about 6% a second.

(CMD #) (InitialRate) (Repeat Rate) (ZONE(S)) <cr>

hex: (\$0B) (Step Size) (Step Size)

decimal: ( 11)

Example: To ramp up system Zones 0 and 1 at about 15% a second, the command string is:  
~11h B 200 200 0 1 <cr>



## Zone Commands

### RAMP DOWN

The InitialRate is used for the first five cycles then the repeat rate is used. The actual Step Sizes are divided by \$100 then added to the Zone's intensity value every 0.1 second. For example, a Step Size of \$500 causes an intensity change of about 10% a second.

(CMD #) (InitialRate) (Repeat Rate) (ZONE(S)) <cr>  
*hex:* (\$0C) (Step Size) (Step Size)  
*decimal:* ( 12)

*Example:* To ramp down system Zones 255, 256, and 257 at about 40% a second, the command string is:  
~11h C 500 500 FF 100 101 <cr>

### POMMEL or PULSE

For this command, the Zone(s) waits Delay1 cycles then fades to Int1 in Fade1 cycles. It remains at Int1 for Delay2 cycles then fades down to Int2 (or the previous level if \$FF is used for Int2) using Fade2 cycles.

(CMD #) (Delay1) (Int1) (Fade1) (Delay2) (Int2) (Fade2) (ZONE(S)) <cr>  
*hex:* (\$09) (cycls) (0-7F) (cycls) (cycls) (0-7F) (cycls)  
or  
\$FF

*decimal:* ( 9)

*Example:* To take system Zones 7 and 8 immediately up to full for one second and then return them to their previous levels using a 1 second fade, the command string is: ~11h 9 0 7F 0 A FF A 7 8 <cr>

### MULTI LEVEL (FLASH or WARN or CHASE)

This command waits for DLY cycles then fades to INT1 in FADE1 cycles. It remains at INT1 for DLY1 cycles then fades to INT2 in FADE2 cycles. It remains at INT2 for DLY2 cycles. If REPEAT is zero, it fades to the original level for this Zone in FADE1 cycles. If REPEAT is \$FF, it fades back to INT1 using FADE1 and repeats until stopped with a new Zone command. Any value other than zero or \$FF causes it to repeat the INT1, INT2 sequence the number of times indicated by that value before it returns to the original level using FADE1.

The run-one-time (DLY1) is useful in starting CHASE effects.

Upon completion, this command **always** returns to the Pre-Command Intensity.

(CMD #) (DLY) (INT1) (FADE1) (DLY1) (INT2) (FADE2) (DLY2) (REPEAT) (ZONE(S)) <cr>  
*hex:* (\$0D) (cycls) (0-\$7F) (cycls) (cycls) (0-\$7F) (cycls) (cycls) (0-\$FF)  
*decimal:* (13)

*Example:* To flash system Zones 16 and 17 five times between low end and full at a 1 second frequency, and then return to their initial settings, the command string is: ~11h D 0 7F 0 5 0 0 5 4 10 11 <cr>



## Zone Commands

### RAMP SCENE ZONES UP

This command goes to the list of system areas and determines what current system scene is selected in each area. It then scans the preset database for Lighting Zones that are involved with this system scene. If the preset intensity is not equal to the unaffected\_inten, the Zone is commanded to start a RAMP UP. To stop this RAMP UP, you should use the STOP SCENE RAMP UP command (\$22). If an invalid scene is currently ON in the area, this command does not affect it.

See RAMP UP command (\$0B) above for details on the Step Size values.

The Preset Profile will **not** be preserved. (Zones will flatten out at the top of the intensity range.)

	(CMD #)	(unaffected_inten)	(InitialRate)	(Repeat Rate)	(system_area_list)	<cr>
<b>hex:</b>	(\$20)	(0-\$80)	(Step Size)	(Step Size)	(LIST OF SYSTEM AREAS)	
<b>decimal:</b>	( 32)	(80 = All Zones in scene will ramp)				

**Example:** To ramp up all Zones in the current scene that are ON in system area 7 and system area 10 except those that are OFF, the command string is: ~11h 20 0 200 200 7 A <cr>

### RAMP SCENE ZONES DOWN

This command scans the list of system areas and determines what current system scene is selected in each area. It then scans the preset database for Lighting Zones that are involved with this system scene. If the preset intensity is not equal to the unaffected\_inten, the Zone is commanded to start a RAMP DOWN. To stop this RAMP DOWN, you should use the STOP SCENE RAMP DOWN command (\$23). If an invalid scene is currently ON in the area, this command does not affect it.

See RAMP UP command (\$0B) above for information on the Step Size parameters.

The Preset Profile will **not** be preserved. (Zones will flatten out at the bottom of the intensity range.)

	(CMD #)	(unaffected_inten)	(InitialRate)	(Repeat Rate)	(system_area_list)
<b>hex:</b>	(\$21)	(0-\$80)	(Step Size)	(Step Size)	(LIST OF SYSTEM AREAS)
<b>decimal:</b>	( 33)	(80 = All Zones in scene will ramp)			

**Example:** To ramp down all Zones in the current scene that are ON in system area 0, the command string is: ~11h 21 80 200 200 0 <cr>

### STOP SCENE RAMP UP

This command goes to the list of system areas and determines what system scene is currently ON. It then scans the preset database for Zones involved with this system scene. It then HALTs the Zones of this scene that are currently in RAMP UP mode.

	(CMD #)	(sys_area list)	<cr>
<b>hex:</b>	(\$22)		
<b>decimal:</b>	( 34)		

**Example:** To stop scene ramping in system areas 0 and 16, the command string is: ~11h 22 0 10 <cr>

### STOP SCENE RAMP DOWN

This command goes to the list of system areas and determines what system scene is currently ON. It then scans the preset database for Zones involved with this system scene. It then HALTs the Zones of this scene that are currently in RAMP DOWN mode.

	(CMD #)	(sys_area list)	<cr>
<b>hex:</b>	(\$23)		
<b>decimal:</b>	( 35)		

**Example:** To stop scene ramping in system areas 0 and 16, the command string is: ~11h 23 0 10 <cr>





## SELECT SYSTEM SCENE(S)

This command selects a lighting scene using the previously stored set of intensities and Fade and Delay times.

**Note:** Attempted selection of scenes in locked areas or areas being programmed is ignored.

(CMD #) (SYSTEM SCENE(S)) <cr>  
*hex:* (\$12)  
*decimal:* ( 18)

*Example:* To select system scenes 1, 17, and 64, the command string is: ~11h 12 1 11 40 <cr>

## SELECT SYSTEM SCENE(S) USING OVERRIDE TIMES

This command selects a lighting scene using the previously stored set of intensities. It overrides the stored Fade and Delay times for each Zone involved and uses the Fade and Delay times from the command string.

**Note:** Attempted selection of scenes in locked areas or areas being programmed is ignored.

(CMD #) (FADE) (DELAY) (SYSTEM SCENE(S)) <cr>  
*hex:* (\$13) (cycls) (cycls)  
*decimal:* ( 19)

*Example:* To select system scenes 1, 17, and 64 using a 1 second fade time and a 2.5 second delay time, the command string is: ~11h 13 A 19 1 11 40 <cr>

## OUTPUT TEMP SCENE

Zone(s) preset information consists of four values: Zone Number, Intensity, Fade, and Delay.

This command causes the defined temporary preset information to be setup and run in the Zone processor. This does not change any scene LEDs that are currently ON. It only changes the Zone intensities being sent to the dimmers.

(CMD #) (ZONE) (INTENSITY) (FADE) (DELAY) (ADDITIONAL ZONEs PRESET INFO) <cr>  
*hex:* (\$1C)  
*decimal:* ( 28)

*Example:* To send system Zone 1 to full using a 5 second fade when it reaches full, send Zone 2 to OFF with a 1 second fade, the command string is: ~11h 1C 1 7F 32 0 2 0 A 32 <cr>

## SELECT SCENE IN AREA

This command selects the scene number in the given area. This is so that the actual system scene numbers do not have to be known. You can accomplish scene selection in multiple areas by adding the area/scene pairs before terminating the command (\$FFFF).

(CMD #) (SYS AREA) (AREA SCENE) . . . other area/scene pairs . . . <cr>  
*hex:* (\$1E)  
*decimal:* ( 30)

*Example:* To select scene 8 of system areas 0 and 1, the command string is: ~11h 1E 0 7 1 7 <cr>



## Timeclock Commands

---

### DISABLE UNTIL ENABLE ISSUED

This command stops any further events from occurring for the area timeclock(s) listed until an enable timeclock command is received.

(CMD #) (AREA TIMECLOCK(S)) <cr>

**hex:** (\$201)

**decimal:** ( 513)

*Example:* To disable timeclocks in system areas 0→3 and leave them disabled until the re-enable command is issued, the command string is: ~11h 201 0 1 2 3 <cr>

### DISABLE UNTIL END OF DAY OR ENABLE ISSUED

This command stops any further events in the area timeclock(s) listed until they are re-enabled or until midnight of the next day, whichever occurs first.

(CMD #) (AREA TIMECLOCK(S)) <cr>

**hex:** (\$202)

**decimal:** ( 514)

*Example:* To disable timeclocks in system areas 9, 10, and 11 for the rest of the day, the command string is: ~11h 202 9 A B <cr>

### RE-ENABLE WITH COMPLETE CATCHUP

This command re-enables the disabled area timeclock(s) listed. The schedule assigned for today is scanned and all “catchup” type events missed from the time when the timeclock(s) were disabled until the current time are executed. The timeclock resumes operating the assigned schedule. This command does not affect a listed timeclock that is not disabled.

(CMD #) (AREA TIMECLOCK(S)) <cr>

**hex:** (\$203)

**decimal:** ( 515)

*Example:* To enable timeclocks in system areas 21 and 22 and run all “catchup” type events in the current schedule that were passed over while they were disabled, the command string is: ~11h 203 15 16 <cr>



### RE-ENABLE AND DO PREVIOUS EVENT

This command re-enables the disabled area timeclock(s) listed. The assigned schedule is scanned for the last event that would have occurred. All events scheduled for that event's time (astronomic and/or real time) are executed immediately (this may be more than one). The timeclock resumes operating the assigned schedule. This command does not affect a listed timeclock that is not disabled.

**(CMD #) (AREA TIMECLOCK(S)) <cr>**

**hex:** (\$204)

**decimal:** ( 516)

*Example:* To enable timeclock for system area 7 and execute the last event that was to be done while the timeclock was disabled, the command string is: ~11h 204 7 <cr>

### STANDARD RE-ENABLE

The standard re-enable command re-enables the disabled area timeclock(s) listed. The next timeclock event is the next scheduled event of the schedule currently running. This command does not affect a listed timeclock that is not disabled.

**(CMD #) (AREA TIMECLOCK(S)) <cr>**

**hex:** (\$205)

**decimal:** ( 517)

*Example:* To re-enable timeclock for system areas 0 and 17, the command string is: ~11h 205 0 11 <cr>



## Timeclock Commands

---

### SET CLOCK

This command loads the present time into the timeclock using zero for the seconds and starts it running using the updated time.

	(CMD #)	(HOUR)	(MIN)	(MONTH)	(DATE)	(YEAR)	(DAY)	<cr>
<i>hex:</i>	(\$207)	0-\$17	0-\$3B	1-\$C	1-\$1F	0-\$63	1-\$7	
<i>decimal:</i>	( 519)	0-23	0-59	1-12	1-31	0-99	1-7	

HOURS ARE IN 24-HOUR FORMAT

DAY 1 = SUNDAY

DAY 2 = MONDAY

.

.

.

DAY 7 = SATURDAY

*Example:* To set timeclock to 5:00 P.M. Saturday, January 25, 1997, use the decimal in the command string:  
~11h 207 11 00 1 19 61 6 <cr>



## Wallstation Commands

### ENABLE WALLSTATION

This command enables all switches on the given wallstation(s) in the list.

(CMD #) (STATION ID(S)) <cr>

**hex:** (\$300)

**decimal:** ( 768)

(STATION ID(S)) = **hex:** link number (0→\$B) \* \$100 + Station on link (0→\$1F)

= **decimal:** link number (0→11) \* 256 + Station on link (0→31)

*Example:* To enable wallstation 21, link 1, the command string is: ~11h 300 14 <cr>

### DISABLE WALLSTATION

This command disables all switches on the given wallstation(s) in the list.

(CMD #) (STATION ID(S)) <cr>

**hex:** (\$301)

**decimal:** ( 769)

(STATION ID(S)) = **hex:** link number (0→\$B) \* \$100 + Station on link (0→\$1F)

= **decimal:** link number (0→11) \* 256 + Station on link (0→31)

*Example:* To disable wallstation 8, link 3, the command string is: ~11h 301 207 <cr>

### SIMULATE WALLSTATION SWITCH PRESS

This command simulates a switch press command to the system from any possible system wallstation.

The system then runs any programmed script actions for the “pressed” switch.

(CMD #) (SWITCH ID(S)) <cr>

**hex:** (\$500-\$50B)

**decimal:** (1280-1291)

Note: The CMD# is dependent upon the link on which the wallstations resides. For example, the command for a wallstation on Link 4 would be \$504 (hex) or 1024 (decimal).

**SWITCH ID = hex:** Station of link (0→\$1F) \* \$100 + Switch on station (0→\$1F)

= **decimal:** Station of link (0→31) \* 256 + Switch on station (0→31)

One or more switch IDs may be placed before the terminator.

*Example:* To simulate a switch press on wallstation 16, link 2, switch 5, the command string is:  
~11h 501 F04 <cr>



## Wallstation Commands

### SIMULATE WALLSTATION SWITCH RELEASE

This command simulates a switch release command to the system from any possible system wallstation. The system runs any script actions programmed for the "released" switch.

(CMD #) (SWITCH ID(S)) <cr>  
*hex:* (\$600-\$60B)  
*decimal:* (1536-1547)

Note: The CMD# is dependent upon the link on which the wallstation resides. For example, the command for a wallstation on Link 4 would be \$504 (hex) or 1024 (decimal).

**SWITCH ID = hex:** Station of link (0→\$1F) \* \$100 + Switch on station (0→\$1F)  
**= decimal:** Station of link (0→31) \* 256 + Switch on station (0→31)  
One or more switch IDs may be placed before the terminator.

*Example:* To simulate a switch release on wallstation 4, link 3, switch 1, the command string is:  
~11h 602 300 <cr>

### RUN SCRIPT(S)

This command runs the listed scripts in the order given. Any script over Script\_Max will cause termination of the command.

(CMD #) (SCRIPT NUMBER LIST) <cr>  
*hex:* (\$40C)  
*decimal:* ( 1036)

*Example:* To run system scripts 0, 256, the command string is: ~11h 40C 0 100 <cr>

### SET SYSTEM VARIABLE

This command sets all the variables in the list to the NEW\_VALUE.

(CMD #) (NEW\_VALUE) (...Variable list....) <cr>  
*hex:* (\$40D) (0-\$FF)  
*decimal:* ( 1037)

*Example:* To set system variables 1 and 7 to the value of 0, the command string is: ~11h 40D 0 0 6 <cr>

### TOGGLE SYSTEM VARIABLE

This command toggles the state of all system variables listed. A variable with a value of zero changes its value to \$FF. All non-zero values become zero.

(CMD #) (Variable list....)  
*hex:* (\$40E)  
*decimal:* ( 1038)

*Example:* To toggle system variables 1 through 7, the command string is: ~11h 40E 0 1 2 3 4 5 6 <cr>



System monitor commands provide a method for the RS-232 device to obtain system status information.

## SYSTEM RESPONSES TO STATUS REQUESTS

The system responds to the status request string by using a response string containing the status information in the same order that it was requested. Also, the type of information that matches the command is sent for further error checking by the host computer.

After each command line is entered, the GRAFIK 6000 processor transmits a response in the following form:

hex	\$FF	character (all bits set)
hex	\$7E	(~)

Followed by the response substrings (see examples below).

The last response substring is terminated with "N OK" where N is the number of commands executed followed by:

hex	\$0D	CR
hex	\$0A	LF

The substrings have the form:

hex	\$3A	(:)	start of substring
hex	number		nn low byte of request cmd (e.g., AREA STATUS = 00)
hex	\$20		space
hex	number 1		(1 to 4 digits)
hex	\$20		space
hex	number 2		(1 to 4 digits) if required
hex	\$20		space
followed by other hex numbers if required			
hex	\$31		(1) command executed
hex	\$4F		(0)
hex	\$4B		(K)
hex	\$0D		<CR>
hex	\$0A		<LF>



## AREA STATUS

This command requests information about the system area, such as whether a current scene is presently ON or whether a timed sequence is or is not running.

(CMD #) (SYSTEM AREA) <cr>

*hex:* (\$801)

*decimal:* ( 2049)

*Response:* :01 (area#) (system scene\_on) (seq running)  
\$FFFF = no sequence running

*Example:* To get area status of system area 0, the command string is: ~11h 801 0 <cr>

*Response:* :01 0 7 \$FFFF (The system area 0 is in system scene 8 with no sequence running.)

## TIMECLOCK STATUS

This command requests the status of the given system timeclock which includes schedule running, type of next event time, next event time, and next event script.

(CMD #) (TIMECLOCK NUMBER) <cr>

*hex:* (\$802)

*decimal:* ( 2050)

Next Event Type

- N = No more events in schedule
- D = Timeclock is disabled
- A = Astronomic event
- R = Real time event

**Note:** If bit 15 is set in the next event time, the event is a catchup event.

*Response:* :02 (tc#) (schedule#) (type) (next event time) (next script)

*Example:* To get the timeclock status of system area 7, the command string is: ~11h 802 7 <cr>

*Response:* :02 7 48 R 3FC 23B

(The timeclock is in system area 7, schedule #73 is a real-time event and will occur at 1020 minutes past midnight [5:00 p.m.] and will run system script 572)





## WALLSTATION STATUS

This command requests the priority and enable/disable status of all switches on the specified wallstation.

(CMD #) (STATION\_ID) <cr>

hex: (\$803)

decimal: ( 2051)

(STATION\_ID) = hex: link\_number (0→\$0B) \* \$100 + station of link (0→\$1F)

= decimal: link\_number (0→11) \* 256 + station of link (0→31)

**Response:** :03 (WS ID) (all switches on station and their priority)

Priority value in lower four bits (0–F) is enabled at that priority if disabled bit seven is set in priority value (80→8F).

*Example:* To check the status of wallstation 8 on link 3, the command string is: ~11h 803 207 <cr>

*Response:* :03 7 5 5 5 5 5 5 (wallstation 8 on link 3, all switches are priority 5 [Enabled])

## SWITCH STATUS

This command requests the priority of the given wallstation switch.

(CMD #) (STATION\_ID) (SWITCH)

hex: (\$804)

decimal: ( 2052)

(STATION\_ID) = hex: link\_number (0→\$0B) \* \$100 + station of link (0→\$1F)

= decimal: link\_number (0→11) \* 256 + station of link (0→31)

**Response:** :04 (SWITCH) (priority)

Priority value in lower four bits (0–F) is enabled at that priority if disabled bit seven is set in priority value (80→8F).

*Example:* To check switch status of wallstation 8 on link 3, switch 1, the command string is:

~11h 804 207 0 <cr>

*Response:* :04 207 0 (wallstation 8, link 3, switch 1, has a status of 0 [Enabled])

## ZONE INTENSITY

This command requests the current intensity of the given Zone.

(CMD #) (ZONE #) <cr>

hex: (\$805)

decimal: ( 2053)

**Response:** :05 (intensity)

(0–\$7F)

*Example:* To request current intensity of zone 257, the command string is: ~11h 805 101 <cr>

*Response:* :05 5F (the intensity of system zone 257 is 95 [75%])



## TIME NOW

This command requests the current system time given as the number of minutes past midnight.

(CMD #) <cr>

hex: (\$808)

decimal: ( 2056)

**Response:** :08 (minutes past from midnight)

*Example:* To request current time, the command string is: ~11h 808 <cr>

*Response:* :08 398 (the current time is 920 minutes past midnight [3:20 p.m.])

## ASTRO\_TIMES

This command requests the sunrise/sunset times for the current date given as the number of minutes past midnight.

(CMD #) <cr>

hex: (\$809)

decimal: ( 2057)

**Response:** :09 (sunrise minutes past from midnight) (sunset minutes past from midnight)

*Example:* To request today's time for sunrise and sunset, the command string is: ~11h 809 <cr>

*Response:* :09 18B 47D (sunrise is 395 minutes past midnight [6:35 a.m.], sunset is 1149 minutes past midnight [7:09 p.m.])

## DATE

This command requests the current date including the day of the week.

(CMD #) <cr>

hex: (\$80A)

decimal: ( 2058)

**Response:** :0A (month) (date) (year) (day of week)  
1-\$C 1-1F 0-63 1-7 (1 = SUNDAY)

*Example:* To request today's date, the command string is: ~11h 80A <cr>

*Response:* :0A 9 11 2 3 (the month is 9 [Sept.], the date is 17, the year is 2 [2002], and the day is 3 [Tuesday])

## CODE REV LEVEL

This command requests the software revision level of the GRAFIK 6000 embedded operating software. It is used for diagnostic purposes.

(CMD #) <cr>

hex: (\$811)

decimal: ( 2056)

**Response:** :11 revision level

*Example:* To get the revision level of the current operating software, the command string is: ~11h 811 <cr>

*Response:* :11 300 (the current operating software is revision 300)



## BOOT CODE REV LEVEL

This command requests the revision level of the embedded system's boot software. It is used for diagnostic purposes.

(CMD #)            <cr>  
*hex:*            (\$812)  
*decimal:*        ( 2066)

**Response:**     :12 boot revision level

*Example:* To get the revision level of the current boot software, the command string is: ~11h 812 <cr>  
*Response:*    :12    114 (the current boot software is revision 114)

## SERIAL NUMBER

This command requests the serial number of the GRAFIK 6000 processor board. It is used for diagnostic purposes.

(CMD #)            <cr>  
*hex:*            (\$813)  
*decimal:*        ( 2067)

**Response:**     :13 serial number

*Example:* To get the serial number of the GRAFIK 6000 processor board, the command string is: ~11h 813 <cr>  
*Response:*    :13    030A (the serial number of the GRAFIK 6000 processor board is 778)

## SYSTEM VARIABLE VALUE

This command requests the value of a specific system variable number.

(CMD #) (VARIABLE NUMBER)            <cr>  
*hex:*            (\$815)  
*decimal:*        ( 2069)

**Response:**     :15 variable value

*Example:* To get the value of system variable 1, the command string is: ~11h 815 0 <cr>  
*Response:*    :15    FF (the value of system variable 1 is 255 [closed])

## **Technical and Sales Assistance**

If you need assistance, call the toll-free  
***Lutron Technical Assistance Hotline:***

(800) 523-9466 (U.S.A., Canada, and the Caribbean),  
other countries call: (610) 282-3800  
Fax: (610) 282-3090

Lutron and GRAFIK 6000 are registered trademarks of Lutron Electronics Co., Inc.  
Windows is a registered trademark of Microsoft Corporation.  
© 2002 Lutron Electronics Co., Inc.

---

Lutron Electronics Co., Inc.  
7200 Suter Road  
Coopersburg, PA 18036-1299 U.S.A.  
Made and printed in U.S.A. 10/02 P/N 040-134 Rev. B

**LUTRON®**