



Orb AML Specification

Version 2.0

1 Introduction

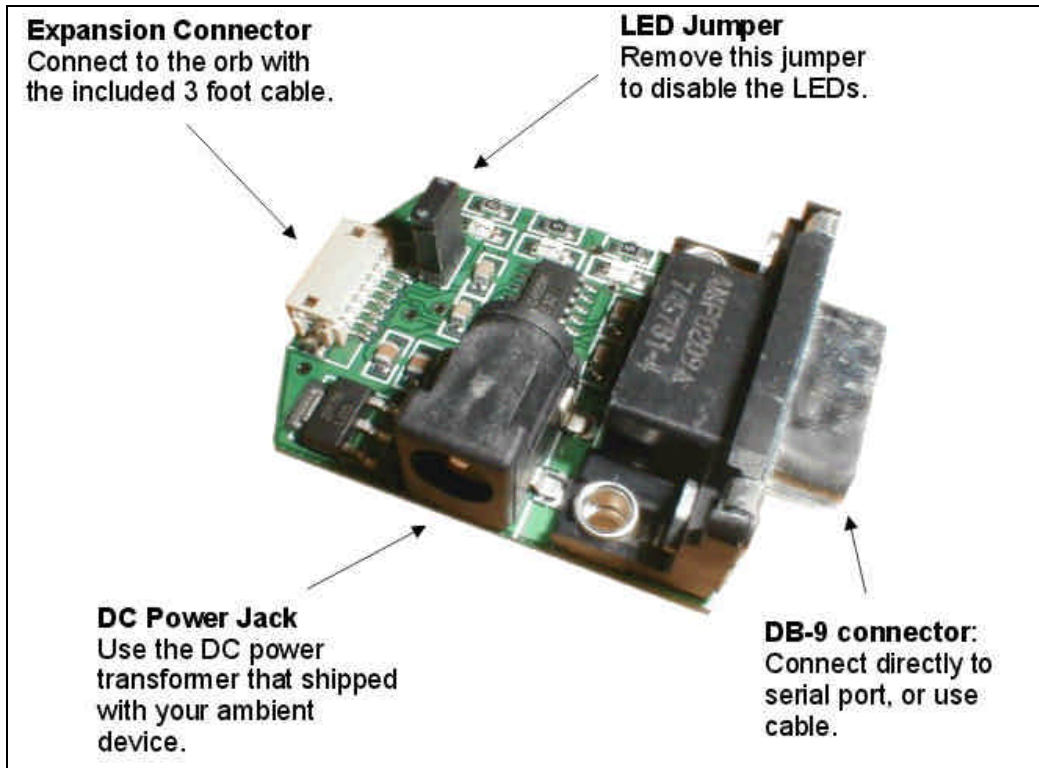
The Ambient Orb is normally controlled via wireless pager signals sent on the 929.6625 MHz band. But the orb can also be fully controlled by a serial port connection via the 7-pin expansion connector on the underside of the board.

It is important to note the orb only receives information on color and pulse rate. The orb has no knowledge of where this information came from. The Ambient web server collects content and uses the web based user configuration rules to determine the color of the orb. Only that information is sent to the orb. Therefore, the data sent to the orb is very compact and simple.

The format of this compact data is Ambient Markup Language (AML). AML is highly efficient at describing the state of ambient devices. This allows a large number of wireless signals to be sent at very low cost.

Please email us at engineering@ambientdevices.com with questions or comments. Let us know what you build. You can also visit our unofficial fan site at <http://www.ambient411.com/>

2 Serial Developer Board



Attach your ambient device to a serial port using the serial developer board as shown. Note that DC power can be supplied to EITHER the DC power jack on the developer board, OR the DC power jack on the orb. Do not attach power to both.

This board can be ordered from our website at:

<http://www.ambientdevices.com/developer/orbhdkorder.html>

This board is essentially a TTL to line level converter, as well as the physical connection between the orb and the serial port expansion port. For more information about the expansion port or building your own developer board, please see

<http://www.ambientdevices.com/developer/DIYSerialDeveloperBoard.html>

3 Serial Communication

All communication between the orb and the serial port is according to the following parameters. These parameters are fixed and cannot be changed.

- 19200 baud.
- 8 data bits
- no parity

- 1 stop bit
- No flow control

All the tilde commands described below can be entered via a terminal emulation program such as Hyperterm on the PC. It is recommended, however, to use the sample java software available on the developer website.

Serial input spaced more than 750 milliseconds apart will cause the orb to timeout and reset input. This is to prevent lockups from spurious or incorrect serial input. This timeout should not be a problem for machine control, but might present a challenge for human keyboard input through Hyperterm or similar. Unless you're a very fast typist, compose text in Notepad and then paste from the clipboard into Hyperterm.

4 Tilda ('~') Commands

All commands to the orb are preceded by a tilda (~) character (ASCII 126). In general, the orb response is to echo the command, followed by a '+' for successful operation, or a '-' indicating some type of error. No additional error information is supplied.

When the orb is first powered on, it writes 'ON' to the serial port (ASCII 79, ASCII 78)






































4.1 'A' - Basic AML Packet

This is the most compact mode of sending the orb information. Two bytes of data are sent according to the following formula:

$$\begin{aligned} \text{First Byte: } & (\text{Color} + (37 * \text{anim})) / 94 + 32 \\ \text{Second Byte: } & (\text{Color} + (37 * \text{anim})) \% 94 + 32 \end{aligned}$$

Where color and anim are selected from the following table:

COLOR:

0	1	2	3	4	5	6	7	8	9	10	11	12	13
													
14	15	16	17	18	19	20	21	22	23	24	25	26	27
													
28	29	30	31	32	33	34	35	36					
													

ANIMATION

0	None - Solid color
1	Very Slow Pulse (pulsing is just barely perceptible)
2	Slow Pulse
3	Medium Slow Pulse
4	Medium Pulse
5	Medium Fast Pulse
6	Fast Pulse
7	Very Fast Pulse (pulsing is annoyingly fast -- no longer an ambient signal)
8	Crescendo Pulse
9	Heartbeat Pulse

Example:

Description	color	anim	transmit (no quotes)
Solid Red	0	0	"~A " (two spaces)
Solid Blue	24	0	"~A 8"
Slow pulsing red	0	2	"~A j"
Medium Pulsing Yellow	6	4	"~A!\"

When transitioning from one AML packet to another, the orb firmware takes care of creating a smooth transition between colors and animations. The user will not see an abrupt transition between colors.

4.2 'D' - Direct RGB triplet

Direct RGB values are streamed to the orb. The orb will abruptly switch to the new color. If smooth transitions are desired, the controller must provide continuous illumination control.

The RGB values are binary values between 0 and 176. Therefore, many values are not represented by ASCII equivalents and cannot be entered from a text based terminal device.

Also note these values control the electrical current to each of the red, green, and blue LEDs. Because each color of LED has a different voltage to brightness curve, color mixing results may not be as expected. For instance, full red and full green is much closer to green than red. For a true yellow, green must be reduced. AML packet mode (~A) described above is calibrated for these variances, but direct mode is uncalibrated.

Please contact Ambient Devices for more information about calibration.

Example:

Description	red	green	blue	transmit
Solid Red	176	0	0	~D (177)(0)(0)
Pink	176	80	80	~D(176)(80)(80)

4.3 'G' - Ignore pager data

If orb data is being provided by an external connectivity module, it is important the orb cease responding to over the air pager data. Be sure to disable pager data when controlling the orb. This value is reset on power cycle. External modules must detect orb power cycles and re-disable the orb each startup.

Example:

Description	Transmit
Ignore pager data	~GT
Don't ignore pager data	~GF

4.4 'I' - Get Info

Returns info about the orb:

Byte Offset	Description	Notes
0	!	(exclamation point -- ASCII 33)
1	I	(uppercase 'I' -- ASCII 73)
2,3	product ID	Currently 4266
4,5	version ID	Currently 4116
6-14	serial number	9 digit serial number.
15-21	internal data	internal
22	Current brightness	0 - dim, 1 - medium, 2 - bright
23	num pages received (zero)	not implemented
24	ignore pager data	'T' for true, 'F' for false
25	premium capcode private	'T' for true, 'F' for false
26 - 27	internal data	internal data
28 ...	Copywrite string	(c)2003 Amb Dev, Orb 2.0b

Note: The copywrite string is variable length, and terminated by a zero byte.

4.5 'L' - Set Local Button Control

Allows an external device to take over control of the buttons. This allows an external device to respond to process user button presses activity. For instance, a quick press can toggle brightness, while a sustained press indicates a drilldown request.

Example:

Description	Transmit
Disable buttons	~LE
Enable buttons	~LI

Buttons are automatically enabled on orb power cycle.

When either the brightness or reset button is pressed, the following binary value is output from the orb to the serialport. These values are written to serial port regardless of whether buttons are enabled or disabled.

Description	Binary Value Transmitted
Brightness button pressed	202
Brightness button released	203
Reset button pressed	200
Reset button released	201

The orb brightness can be controlled by an external device. Note that while reset can be disabled, a reset can only be performed from the orb button. This is to prevent accidental resets.

4.6 'P' - Ping

Returns standard result code (P+). Lets module know orb is powered and operational

4.7 'R' - Set Brightness

Allows the brightness to be set by external connectivity module. Use in conjunction with local button control to add additional functionality for the brightness button.

Brightness is always set to maximum on orb power cycle.

Example:

Description	Transmit	Comments
Set to dim	~R0	Third digit is zero (ASCII 48)
Set to medium	~R1	
Set to bright	~R2	