Ambient Lights - RGB LED controller with Bluetooth and Android client application



Ambient Lights

- Specifications
- Usage via twist-and-push-button
- Usage via Android client application
- Serial communication protocol
- Downloads

The *Ambient Lights* project is a RGB LED controller based on Arduino with optional Bluetooth connection and an Android client software.

Colors can be changed by using the single twist-and-push-controller or by sending commands to the controller's serial interface. It also holds connection pins for directly plugging in the Bluetooth module BTM-112.

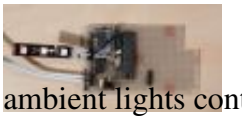
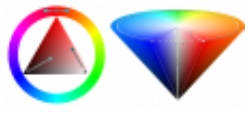
In order to update the controller's firmware, it can be connected to a compatible Arduino board.

The Android client software lets you control the *Ambient Lights* from your Android device via Bluetooth.

Unfortunately you are not able to watch the video entitled "Ambient Lights in action" without Javascript.

Video showing *Ambient Lights* in action

See some pictures of the controller followed by some spec and further details:

-  ambient lights controller with bt112
-  ambient lights controller without bt112
-  ambient light controller connected to arduino board
-  ambient lights button
- ambient lights controller schematic
-  HSV triangle and cone
-  ambient lights android client



- ambient lights photo

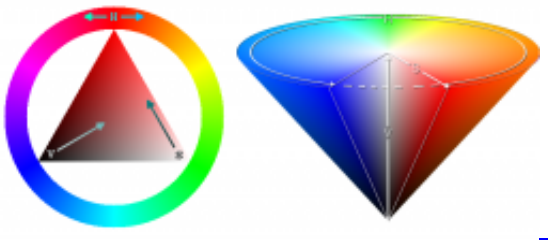
Back to top

Specifications

General Usage Specifications:	
Input Voltage:	12VDC
Supported RGB LED Strips:	12V common VCC, separate GND
Max. output power:	500mA per color, i.e. 18W max. total RGB power
Color adjustment via:	• single twist-and-push button • serial interface • Bluetooth SPP (via BTM-112)
Used components:	
Microcontroller:	ATMEGA 168 @ 16MHz with Arduino bootloader
Driver module:	ULN2004AN
DC/DC converter:	ST LD1117V33
twist-and-push-button:	Encoder PANASONIC EVEQDBRL416B
Bluetooth Module:	BTM-112

Back to top

Usage via twist-and-push-button



HSV triangle and cone, source: wikipedia

Attached to the *Ambient Lights* controller is a button which can be turned left or right and pressed. With these movements, the user can set any color available in the HSV color space.

The controller's firmware implements some kind of "HSV menu". Pressing the button switches between H-, S- and V-mode where you can adjust the values for Hue, Saturation and Brightness (=Value). Turning the button adjusts the currently selected value - turning right increases it, turning left decreases it. Holding the button down while turning it makes the value increase or decrease faster.

When switching between H-, S- and V-mode by pressing the button, a short color flash indicates the mode you're in:

red color flash:	H-mode	[adjust Hue]
green color flash:	S-mode	[adjust Saturation]
blue color flash:	V-mode	[adjust Brightness (Value)]

Long-pressing the button toggles the auto-mode. In auto-mode the current S- and V-values will stay the same but the color will change by steadily moving to the right on the color-wheel (Hue will be increased). Just long-press the button again to switch back to manual mode.

Back to top

Usage via Android client application



Android client application

The video shows the *Ambient Lights* being controlled by the Android client application running on a HTC Desire. The app requires Android 2.1+ and uses the device's Bluetooth connection.

With the current version it is possible to select the wanted color using a color wheel for Hue and Saturation and a color bar for brightness. When a color is selected, the command is directly sent to the *Ambient Lights* controller. It is also possible to send raw commands using the edit field.

Unfortunately you are not able to watch the video entitled "Ambient Lights controlled by Android App" without Javascript.

Video showing *Ambient Lights* controlled by Android client-app

Back to top

Serial communication protocol

The Android client application just implements the serial communication protocol supported by the *Ambient Lights* controller. The following table shows the commands available. Commands are sent without any line breaks etc. Serial speed is 19200bauds.

Command	**Description**	**Example**
cRRRGGGBBB	Sets color using RGB-values. RRR: red value as 3-digit integer, 0..255 GGG: green value as 3-digit integer, 0..255 BBB: blue value as 3-digit integer, 0..255	c255128096
dHHHSSSVVV	Sets color using HSV-values. HHH: Hue value as 3-digit integer, 0..360 SSS: Saturation value as 3-digit integer, 0..255 VVV: Brightness value as 3-digit integer, 0..255	d310255255
a	Activates auto-mode with automatic Hue-alteration. Saturation and Brightness will	a

	stay the same.	
h	Deactivates auto-mode.	h
sSSSDDD	Sets speed for auto-mode. SSS: amount of degrees Hue will be increased by every DDD ms DDD: delay in ms between color changes	s005025
r	Makes the controller send the current controller state. Answer's format: "status:STATE.H.S.V.AS.AD\\n" STATE: controller's current state id* H: current Hue value S: current Saturation value V: current Brightness value AS: auto mode's step size in degrees AD: auto mode's delay time in ms All values are sent as string-formatted integer values. * state id definitions: #define STATE_MANUAL_HUE 0 #define STATE_MANUAL_SATURATION 1 #define STATE_MANUAL_VALUE 2 #define STATE_AUTO 3	r
vVVV	Activates or deactivates verbose mode. VVV: "000" turns verbose mode off, "001" activates it In verbose mode informations about controller's state and errors while receiving commands will be sent via the serial connection.	v001

Back to top

Downloads

The zip-file contains schematics, the controller's firmware (Arduino sketch) and a simple python script to set the colors via serial interface from your PC (wxPython needed).

UPDATE:

The file "Ambient_Lights_schematic_firmware_and_test_client.zip" contains the content described above. It's the first revision of the hard- and software and works just as described in the article above.

I now added a new revision of it all. The file "Ambient_Lights_schematic_and_firmware_rev2.zip" contains schematics and firmware source code (Arduino) for this revision 2. Since I changed the serial protocol, the python client from rev1 will not work. I'll add the documentation for it soon. In the meantime, please use the APK which is also provided now. The Android App is not the best piece of software you may have had on your Droid - but it kind of works ;-). **The AmbientLights.apk only works with the rev2 firmware!**

AmbientLights.apk (674.3 kB)

Ambient_Lights_schematic_and_firmware_rev2.zip (150.2 kB)

Ambient_Lights_schematic_firmware_and_test_client.zip (100.1 kB)

Back to top

Leave a comment

Comment by Nguyen | 2011-04-05

Hi Bro!
you can send one list device use connect android phone and lights. software develop and device use connect. Thank pro

Comment by Robert | 2011-04-05

Hi Nguyen,
as described in the article, I use the BTM-112 module. It's easy as pie: The BTM is connected to the serial rx/tx pins of the microcontroller.
To connect from your Android device, refer to the Android API reference.
Here's an example application using BT:
http://developer.android.com/resources/samples/BluetoothChat/index.html

Comment by Delirium | 2011-11-22

Tell me please, what did you use for UUID to connect BTM-112 with Android?

Comment by Robert | 2011-11-22

Hi!
As written in the Android API reference for the method "public BluetoothSocket createInsecureRfcommSocketToServiceRecord (UUID uuid)":

"Hint: If you are connecting to a Bluetooth serial board then try using the well-known SPP UUID 00001101-0000-1000-8000-00805F9B34FB."

So you might try this UUID. If that does not work, try the way I implemented it (since that was the only way that worked with my HTC Desire):

```
Method m = device.getClass().getMethod("createRfcommSocket", new Class
[] { int.class });

socket = (BluetoothSocket) m.invoke(device, 1);


mAdapter.cancelDiscovery();

socket.connect();
```

```
mOutputStream = socket.getOutputStream();

mInputStream = socket.getInputStream();
```

device is of type BluetoothDevice.

Comment by Delirium | 2011-11-22

Thank you very much!) The second method is the only that works with my HTC Wildfire too.

Comment by Yvan Quehec | 2012-03-20

Your project is exactly what I'm looking for (from a long time) to make a small light for my job (documentary operator) how is it possible to find the android application ? Sincerely
Yvan

Comment by Robert | 2012-03-20

Hi Yvan,
since the Android application only works with this controller, it'll be useless without it.
Are you planning on building one? I'll try to find the time to upload the whole package containing schematics, Arduino source code and the Android sources.

Cheers!
Robert

Comment by yvan | 2012-04-21

I'm good hand working but not good enough in electronic I would like to make a small light with 3 common RGB 10w led and 3 common daylight 10w. The idea is to be able to programm (with an android phone)a color storable (or if not too complicate, possibilities to have 2 or 3 presets stored)on the unit then have a general dimmable on the unit.
Your project is so close to what I'm looking for.

Comment by Robert | 2012-04-21

Hi Yvan,
sorry, it took a while. You can find a link to a zip-file containing schematics, sources for the firmware and a python script for controlling it via serial interface at the end of this article. The Android app is not in a distributable state at the moment ;-)

Salut!

Comment by Sandor | 2012-04-22

Dear project author!Thank You for this great device.This is the best project that I found in the internet.I have long wanted to build RGB controller and control it via Bluetooth interface.I have a question - where i can download Android(only android?) client software which is used by you in this project.If it's possible send me please to my email address. Would be very thankful!Thank you!

Comment by Rowan | 2012-08-13

Hiya, been looking for something just like this. Is your android app in good enough condition to distribute yet? Looks brilliant - haven't found any other apps that are this good looking and functional!

Comment by yvan | 2013-01-25

Hello Robert, your work is so close to my dream, I can't resit to try to build one, I wrote a small "call for tribute" on my website because I dream of some enhancements but I'm not good enough in programming...
http://elemac.fr/spip.php?article172

I wish you an happy new year and thanks for your job.
Sincerely

Comment by Robert | 2013-01-25

Hi Yvan,
thank you. A happy new year to you as well!
I read your article. Seems like a good project you're on! Especially the additional 6000K white LED is a good idea.
Just go ahead and build it! Your programming skills are going to increase during practice ;-)

Best of luck!

Comment by yvan | 2013-01-25

For the moment, I try to find information and hope to make some prototype one day.
It's very important to be an open project, I don't want to make any money on this.
For the moment it's too big for me because I have to learn many things...
It should be nice to modify your apk to enhance in grbw and also have 3 rgb sliders on the side of the color wheel and add some presets...
Still thinking and searching...

Comment by John Doe | 2013-02-23

Hi, please could You tell me where to get ASKModulator.h ? Your program references to it but nothing found on google and inside v2 zip.
Thank You for answer. John

Comment by Rob Murphy | 2013-05-03

This looks great, by far the most professional project of its type. However I can't find ASKModulator.h either, it's referenced in one place as local and another as part of a library?

Comment by Robert | 2013-05-13

Hi Rob,
sorry for the issues! I removed the references to ASKModulator and updated the rev2.zip file. Just download it again and it should work ;-)
Cheers,
Robert

Comment by Ledlight | 2022-04-20

Don’t purchase an LED Light or disco light and don’t waste your money install this app in your android device and solve your disco with music, flashlight & strobe light problem.

Comment by flash light | 2022-06-02

Don’t purchase an flash Light or disco light and don’t waste your money install this app in your android device and solve your disco with music, flashlight & strobe light problem.