HOME LIGHTING CONTROL SYSTEM USING BLYNK

WHAT IS A LIGHTING CONTROL SYSTEM?

A **lighting control system** is an intelligent network based lighting control solution that incorporates communication between various system inputs and outputs related to lighting control with the use of one or more central computing devices. Lighting control systems are widely used on both indoor and outdoor lighting of commercial, industrial, and residential spaces. Lighting control systems serve to provide the right amount of light where and when it is needed.

Lighting control systems are employed to maximize the energy savings from the lighting system, satisfy building codes, or comply with green building and energy conservation programs. Lighting control systems are often referred to under the term Smart Lighting.

REQUIREMENTS

In this project, we are going to build a home light controlling system using the Internet. Using this project we are able to control our home light from anywhere in the world.

Hardware Requirements:

- Wi-Fi Node MCU ESP8266
- Standard USB cable to connect Node MCU
- 4 Channel Relay module (5V)
- Jumper wires female to female
- Home Light (Bulb)
- Bulb holder
- Wire
- AC 220v/120v home appliances or 9v Hi-Walt Battery

Software requirements:

- Arduino software
- Blynk app

WORKING PRINCIPLE

In this project, there are three main components used an Android Blynk app, Wi-Fi Node MCU and 4 Channel Relay module. The Android Blynk app sends the serial data to the Wi-Fi Node MCU by clicking ON button. The Wi-Fi Node read the input data and process it according to program uploaded inside it and generate the output to 4 Chanel Relay Module.



Steps







Widget Box				
	POTINIANE 11,200	+ Add		
	Button			
Ballon	Styled Button	Ē		
	Slider			
¢	Vertical Slider			
٢	Timer 1200			
\odot	Joystick			
	zeRGBa 1400			





←	Button Se	ettings		i
	Ę	area C		
AC	BULB			
output	IN	1		0
	PUSI) switc	н
OF	F		ON	
DESIGN FONT S	T T	TE	XT	
		🖄 De	lete	



ARDUINO IDE

Add the board type to your Arduino IDE if not added earlier before compiling. Go to File > Preferences > Settings and add http://arduin o.esp8266.com/st able/package_esp 8266com_index.js on in Additional **Boards Manager** URLs and click OK.

references			×
Settings Network			
Sketchbook location:			
C: \Users\Gigabit\Documents\A	rduino		Browse
Editor language:	System Default 👻	(requires restart of Arduino)	
Editor font size:	12		
Interface scale:	Automatic 100 +% (requires restart of Arduino))	
Theme:	Default theme 👻 (requires restart of Arduino)		
Show verbose output during:	compilation i upload		
Compiler warnings:	None 👻		
Display line numbers			
Enable Code Folding			
Verify code after upload			
Use external editor			
Aggressively cache compile	ed core		
Check for updates on start	tup		
Update sketch files to new	extension on save (.pde -> .ino)		
Save when verifying or up	loading		
Additional Boards Manager URL	s: http://arduino.esp8266.com/stable/package_esp826	i6com_index.json	
More preferences can be edite	d directly in the file		
C: \Users\Gigabit\AppData\Loca	al\Arduino15\preferences.txt		
(edit only when Arduino is not r	running)		



home_light_controll	ing_using_wifi Arduino 1.8.8		_	-		
File Edit Sketch Too	ls Help					Include
home_light_cc	Auto Format Archive Sketch Fix Encoding & Reload	Ctrl+T				the esp8266 board community befo
<pre>#include <esp #include="" <bly="" auth[]="</pre" char=""></esp></pre>	Manage Libraries Serial Monitor Serial Plotter	Ctrl+Shift+I Ctrl+Shift+M Ctrl+Shift+L				re compiling the above code otherwise, it
<pre>char ssid[] = char pass[] =</pre>	Blynk: Check for updates Blynk: Example Builder		Bo	oards Manager		generates an error. To include library
<pre>void setup() { // Debug cons</pre>	Blynk: Run USB script WiFi101 / WiFiNINA Firmware Updater		Ar Ar Ar	duino AVR Boards duino Yún duino/Genuino Uno		click Tools > Board > Board Manage
Serial.begin(Board: "NodeMCU 1.0 (ESP-12E Module)"		Ar	duino Duemilanove or	Diecimila	



Search for esp8266 and install it.

💿 Boards Manager	The second se	×
Type All	* esp8266	
esp8266 by ESP8; Boards included in Generic ESP8266 M XinaBox CW01, ES (ESP-12E Module), ESP-210, LOLIN(W Module), ThaiEasy Seeed Wio Link, E <u>Online help</u> More info	266 Community 1 this package: Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, Invent Or SPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.1, , Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SweetPea VEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 mini Pro, LOLIN(WEMOS) D1 mini Lite, WeMos D1 R1, ESPino (ESI /Elec's ESPino, WiFInfo, Arduino, 4D Systems gen4 IoD Range, Digistump Oak, WiFiduino, Amperka WiFi Sk ESPectro Core.	ne, 0 P-12 ot,
	Installi	ing
		•
	Downloading boards definitions. Downloaded 347kb of 20.241kb	Cancel



Download the latest Blynk library from https://github.com/blynkkk/blynk-library/releases/latest and add it to program

💿 Select a zip file	e or a folder co	ntaining the library you'd like to add		×
Look in:	🕕 control ho	ne light using wifi		
Recent Items Desktop My Documents Computer Network	Blynk_Rele blynk-libr	ease_v0.5.4 ary-0.5.4 nt_controlling_using_wifi ease_v0.5.4 ary-0.5.4		
	File <u>n</u> ame:	Blynk_Release_v0.5.4.zip		Open
	Files of <u>t</u> ype:	ZIP files or folders	-	Cancel

If the Blynk libraries are not able to be added in your program then, extract it and paste all the directories in Arduino libraries C:\Program Files (x86)\Arduino\libraries. Now, connect Node MCU device with your

personal computer using

Standard USB cable to

connect Node MCU and upload the program in it.

nodemcu_wifi_smart_home | Arduino 1.8.13 (Windows Store 1.8.42.0) File Edit Sketch Tools Help Verify nodemcu_wifi_smart_home #define BLYNK_PRINT Serial #include <ESP8266WiFi.h> #include <BlynkSimpleEsp8266.h> char auth[] = "yW5shlcZJJPhehws393GSRcpoOone2gC"; // the auth code that you got on your gmail and Blynk app char ssid[] = "M21"; // username or ssid of your WI-FI char pass[] = "password"; // password of your Wi-Fi void setup() / Debug console Serial.begin(9600); inMode(D2,OUTPUT); //extend these to D8 if you are using a 8 pin relay pinMode (D3, OUTPUT); //pinMode(D3,OUTPUT); //pinMode(D4,OUTPUT); igitalWrite(D2,HIGH); // Make it low if you want everything to go off digitalWrite(D3, HIGH); // in case of a power cut //digitalWrite(D3,HIGH); /digitalWrite (D4, HIGH); Blynk.begin(auth, ssid, pass); void loop() Blynk.run();

Now, select your board type, go to **Tools > Board: > NodeMCU 1.0**.

💿 home_light_controlling_using_wifi Arduino 1.8.8				
File Edit Sketch Tools Help				
	Auto Format Archive Sketch	Ctrl+T		
<pre>#define BLYNK #include <esp #include="" <bly<="" pre=""></esp></pre>	Fix Encoding & Reload Manage Libraries Serial Monitor Serial Plotter	Ctrl+Shift+I Ctrl+Shift+M Ctrl+Shift+L		
<pre>char auth[] = char ssid[] = char pass[] = void setup() {</pre>	Blynk: Check for updates Blynk: Example Builder Blynk: Run USB script WiFi101 / WiFiNINA Firmware Updater			
// Debug cons Serial.begin (pinMode (D1, OU pinMode (D2, OU pinMode (D3, OU pinMode (D4, OU	Board: "Arduino/Genuino Uno" Port: "COM4" Get Board Info Programmer: "AVRISP mkII" Burn Bootloader	1		

ESPresso Lite 2.0 Phoenix 1.0 Phoenix 2.0 NodeMCU 0.9 (ESP-12 Module) NodeMCU 1.0 (ESP-12E Module) Olimex MOD-WIFI-ESP8266(-DEV) SparkFun ESP8266 Thing SparkFun ESP8266 Thing Dev SweetPea ESP-210 LOLIN(WEMOS) D1 R2 & mini LOLIN(WEMOS) D1 mini Pro LOLIN(WEMOS) D1 mini Lite WeMos D1 R1 ESPino (ESP-12 Module) ThaiEasyElec's ESPino WifInfo Arduino 4D Systems gen4 IoD Range Digistump Oak WiFiduino Amperka WiFi Slot Seeed Wio Link ESPectro Core

DIGITAL CIRCUIT DIAGRAM

4 Channel Relay Module Node MCU

- GND -----> GND
 IN1 -----> D0
- IN2 -----> D1
 IN3 -----> D2
- IN4 -----> D3
- VCC -----> Vin

Connection between Relay Module, Bulb, and input power:

- Connect common-point (com) of Relay Module with home light.
- Connect normally-open (no) of Relay Module with power.
- Connect remaining one home light wire with the power source.
- This project has operated using both Wi-Fi as well as Mobile data.

Output

