Wifi communications  
- Connect Arduino to internet.  
- Control LED on/off from web/mobile phone.  
- Display temperature value on web/mobile phone.

**Hardware** Required

* Arduino or Genuino Board
* ESP8266 Wifi module
* Breadboard
* LED
* Wires
* Pushbutton
* Resistor

Theory

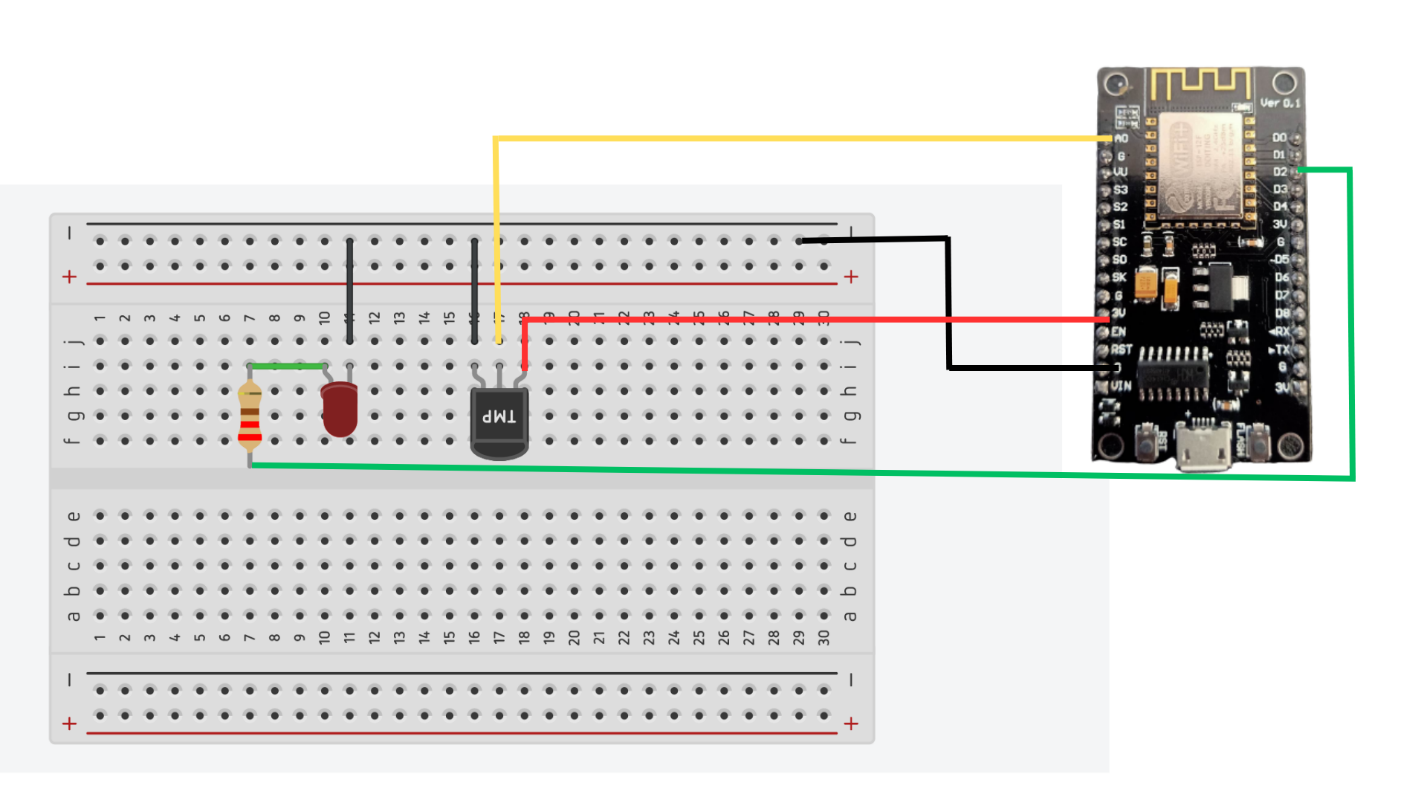
The Arduino Uno WiFi is an Arduino Uno with an integrated WiFi module. The board is based on the ATmega328P with an ESP8266WiFi Module integrated. The ESP8266WiFi Module is a self contained SoC with integrated TCP/IP protocol stack that can give access to your WiFi network (or the device can act as an access point). One useful feature of Uno WiFi is support for OTA (over-the-air) programming, either for transfer of Arduino sketches or WiFi firmware.

The Arduino Uno WiFi is programmed using the Arduino Software (IDE), our Integrated Development Environment common to all our boards and running both online and offline.

In this project, we are going to connect our device to the internet in order to upload data from a temperature sensor and control the LED light. With the support of Blynk, an app that specially designed for IoT, which provides cloud data storage and others useful tools.

Circuit

*(Block diagram and Circuit diagram shows the wire connection).*



|  |  |  |  |
| --- | --- | --- | --- |
| ESP8266 | Breadboard | LED1(trái) | TMP |
| TX |  |  |  |
| RX |  |  |  |
| 3V |  |  | V |
| G(phải) |  | cathode |  |
| D2 |  | anode |  |
| G(trái) | GND(trêni) |  | GND |
| VIN |  |  |  |
| A0 |  |  | AnalogOutput |

Code

**Board NodeMCU 1.0(ESP-12E Module)**

1. #define BLYNK\_PRINT Serial
2. #include <ESP8266WiFi.h>
3. #include <BlynkSimpleEsp8266.h>
4. #include <SoftwareSerial.h>
5. #include <SimpleTimer.h>
7. #define BLYNK\_TEMPLATE\_ID "..."
8. #define BLYNK\_TEMPLATE\_NAME "..."
10. #define led D2
11. char auth[] = "Blynk auth code";
12. char ssid[] = "WiFi Name"
13. char pass[] = "WiFi Password";
15. SimpleTimer timer;
16. //Truyền dữ liệu sensor
17. void myTimerEvent()
18. {
19. Blynk.virtualWrite(V1, millis() / 1000);
21. int analogValue = analogRead(A0);
22. float millivolts = (analogValue/ 1024.0) \*3300;
23. float celsius = millivolts/10; //Tính toán dữ liệu
24. Blynk.virtualWrite(V2, celsius);
25. }
26. //Bật tắt LED
27. BLYNK\_WRITE(V10){
28. int pinvalue = param.asInt();
29. if (pinvalue == 0) {
30. digitalWrite(led, LOW);
31. } else {
32. digitalWrite(led, HIGH);}
33. }
35. void setup()
36. {
37. Serial.begin(9600);
38. Blynk.begin(auth, ssid, pass);
39. pinMode(led, OUTPUT);
40. timer.setInterval(1000L, myTimerEvent);
41. }
43. void loop()
44. {
45. if (Serial.available() == 0 )
46. {
47. Blynk.run();
48. timer.run(); // Initiates BlynkTimer
49. }
50. }

Demonstrations

*(Demonstrations with photos of experiments)*

Before coding the project, we need to be able to connect 2 devices to Arduino IDE in order to upload codes on the devices.

1. We need to [install CH340 driver](https://www.youtube.com/watch?v=MM9Fj6bwHLk) in order to connect to ESP8266. Then *download ESP8266 board* (Tools -> Board -> Boards manager -> Search for “esp8266” by ESP8266 community)
2. We use Blynk to support our project. So that we have to *install Blynk library* (Sketch -> Include library -> Manage library -> Search for “Blynk” by Volodymyr Shymanskyy) .
3. Adding crucial library:

* [SimpleTimer.h](https://github.com/jfturcot/SimpleTimer)(Sketch -> Include library -> ADD .ZIP library...)
* ESP8266WiFi.h (File -> Preferences -> Additional Boards Manager URLs: -> add [this link](http://arduino.esp8266.com/stable/package_esp8266com_index.json))

1. Create Blynk account
2. Create device
3. Get these code by going to **Device Info**
4. Start coding
5. Wiring

References

<https://docs.arduino.cc/retired/getting-started-guides/ArduinoUnoWiFi>

<https://www.youtube.com/watch?v=lp93f4tFqh4&t=924s>

<https://www.youtube.com/watch?v=MM9Fj6bwHLk>

<https://www.youtube.com/watch?v=vDm1zhqS-wE>