

Project: Arduino – On Board Blink LED

Boards: Arduino IDE devices that are WiFi capable. This project used the NodeMCU and WeMo boards which have the ESP8266 chip that provides the WiFi capability.

Pins: None used

Project Summary

This project will use the on board LED. The NodeMCU and WeMo boards was used for this project. However, you can apply the sketch to other Arduino boards that.

Serial Monitor

The following is the sample output from the Arduino IDE Serial Monitor tool. Per the code, it will first connect to your WiFi network, display the IP Address of the device.

Sketch Parameters

Your sketch will require some parameters to properly communicate with OnSiteMonitor. The sample sketch will reference the following:

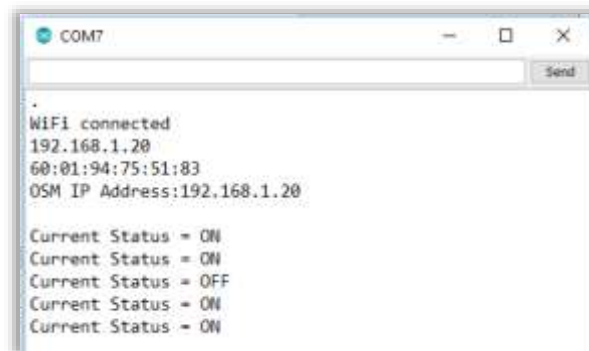
- Account Id
- Device Type
- Device ID
- Command Class

These parameters can be found by logging into your web account and selecting your DIY device.

Visit: <http://www.OnSiteMonitor.com/diy.aspx>

OnSiteMonitor

Adding the device to your OnSiteMonitor Console, you will be able to start sending device status and receiving action commands.



```
COM7
-
WiFi connected
192.168.1.20
60:01:94:75:51:83
OSM IP Address:192.168.1.20
Current Status = ON
Current Status = ON
Current Status = OFF
Current Status = ON
Current Status = ON
```

Sketch File:



OSM_OnBoardLed.i
no

Source Code:

```
//*****  
// On Board Blinking LED  
// This sketch will turn the device's LED light on & off  
//  
// Developed to work with OnSiteMonitor  
//*****  
  
#include <ESP8266WiFi.h>  
#include <WiFiClient.h>  
#include <ESP8266WebServer.h>  
#include <ESP8266HTTPClient.h>  
  
#include <Ticker.h>  
Ticker tickerOSWatch;  
  
#define OSWATCH_RESET_TIME 30  
  
static unsigned long last_loop;  
  
void ICACHE_RAM_ATTR osWatch(void)  
{  
    unsigned long t = millis();  
    unsigned long last_run = abs(t - last_loop);  
    if(last_run >= (OSWATCH_RESET_TIME * 1000))  
    {  
        // save the hit here to eeprom or to rtc memory if needed  
        ESP.restart(); // normal reboot  
        //ESP.reset(); // hard reset  
    }  
}  
  
// Replace with your network credentials  
//SSID is case sensitive  
const char* ssid = "updateme";  
const char* password = "updateme";  
  
//Required static values for device  
String account_id = "updateme";  
String device_type = "updateme";  
String device_id = "updateme";  
String command_class = "1";  
  
//Initial values  
String ip_address = "";  
String osm_ip_address = "";  
String current_status = "";  
int httpCode;  
  
// Status Function  
uint8_t pin_status( uint8_t level, int pin)  
{  
    uint8_t state;  
    digitalWrite(pin, 1);  
    pinMode(pin, INPUT);  
    state = digitalRead(pin);  
    pinMode(pin, OUTPUT);  
    digitalWrite(pin, level);  
    return state;  
}  
  
ESP8266WebServer server(80); //instantiate server at port 80 (http port)  
  
String page = "";  
void setup(void)  
{  
    last_loop = millis();  
    tickerOSWatch.attach_ms(((OSWATCH_RESET_TIME / 3) * 1000), osWatch);  
  
    //make the LED pin output and initially turned off  
    pinMode(LED_BUILTIN, OUTPUT);
```

```

digitalWrite(LED_BUILTIN, HIGH);
current_status = "OFF";

delay(1000);
Serial.begin(115200);
WiFi.begin(ssid, password); //begin WiFi connection
Serial.println("");

// Wait for connection
while (WiFi.status() != WL_CONNECTED)
{
    delay(500);
    Serial.print(".");
}

ip_address = WiFi.localIP().toString();

Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");

//Serial.println(WiFi.localIP());
Serial.println(ip_address);

//Get the IP Address for OnSiteMonitor
HTTPClient http; //Declare an object of class HTTPClient
http.begin("http://www.onsitemonitor.com/osm_ip.aspx?aid=" + account_id);
httpCode = http.GET();
if (httpCode > 0) { //Check the returning code
    osm_ip_address = http.getString(); //Get the request response payload
    osm_ip_address.replace("\n","");
    osm_ip_address.replace("\r","");
}

Serial.println("OSM IP Address:" + osm_ip_address);
http.begin("http://www.onsitemonitor.com/update_diy_ip_address.aspx?dev=" + account_id + "&dt=" + device_type
+ "&did=" + device_id + "&ip=" + ip_address);
httpCode = http.GET(); //Send the request
Serial.println(httpCode);
http.end();

server.on("/CMD=STATUS", []()
{
    //Report the current status of the device
    server.send(200, "text/plain", current_status);
    Serial.println("Current Status = " + current_status);
});

server.on("/CMD=ON", []()
{
    //Process the command to turn the On Board LED ON
    server.send(200, "text/plain", "SUCCESS");
    digitalWrite(LED_BUILTIN, LOW);
    current_status = "ON";
    Serial.println("ON");
});

server.on("/CMD=OFF", []()
{
    //Process the command to turn the On Board LED OFF
    server.send(200, "text/plain", "SUCCESS");
    digitalWrite(LED_BUILTIN, HIGH);
    current_status = "OFF";
    Serial.println("OFF");
});

server.begin();
Serial.println("Web server started!");
}

void loop(void)
{

```

```
server.handleClient();  
last_loop = millis();  
delay(1000);  
}
```