

Dynamic DNS Update Client based on M5 Stack ATOM Lite (ESP32 microcontroller)



Introduction

When you sign up with an Internet Service Provider, you will end up either with a static IP address or a dynamic IP address. What is the difference between the two? Dynamic means “constantly changing.” The prefix dyna means power; however, dynamic IP addresses are “not more powerful”, but internet provider can change them without any notice. Static means staying the same. Static. Stand. Stable. Yes, static IP addresses do not change.

For a home user there is no difference between the two plans until one do not need access to home devices or home computer from internet. However, if you have IP camera or other Internet of Things (IoT) device and want to control them from Internet, the situation becomes different.

For most home internet users a cost of Static IP address is too expensive. You can still use dynamic IP address but then some internal home system must track it changes. Then DDNS is

very helpful. For an example: <http://www.duckdns.org/>. Dynamic domain name service (DDNS) enables you remotely access your computer, control remote IP cameras and other Internet of Things devices on your home or small business broadband connection even when connected to the internet via a dynamic IP address. Some public organizations or companies provide this service free on their web sites.

Ok, but how the DDNS knows when your dynamic IP address is changed by Internet provider and how it upgrades it from your home? The answer: you need to install IP address update client.

So what is an DDNS Update Client?

An update client is a computer application or a feature in your router that keeps your hostname's IP address up-to-date. The update client periodically checks your network's IP address and, if it sees that your IP address has changed, it sends (updates) the new IP address to your hostname in your DDNS account.

I have programmed and installed DDNS Client into cheap and small ESP32 based device. It uses WIFI for internet and USB for +5V power. Ref.: <https://shop.m5stack.com/collections/m5-atom/products/atom-lite-esp32-development-kit>

How it works?

Every 4 hours it obtains your home Public IP address from some Internet server using Get my IP address request. You can program 3 servers, or use default ones. For an example: <https://ip.seeip.org/>

Note: The software uses secure "https:" requests (SSL TCP/IP) to obtain Public IP address and send it to DDNS. Public IP address Servers with non-secure "http:" requests are not supported.

If it detect your home IP address change, it updates your DDNS hostname to resolve to your remote IP address.

Then the device disconnects from WIFI and sleeps with a very low power consumption for 4 hours. Then it awakes, connects to WIFI, and checks Public IP address, sends it to DDNS if Internet provider in case of changes and goes to sleep again for 4 hours.

M5 Stack ATOM Lite programming and Installation with Arduino IDE (<https://www.arduino.cc/>).

Download and install ESP32 zip package from <https://github.com/espressif/arduino-esp32/>
Install ESP32 zip package from: Menu->Sketch->include Library->Add Zip Library

From Menu->Tools->Manage Libraries -> Download: ESP32 Lite Pack Library, ESP_HTTPS_Server, Adafruit NeoPixel, and others if missing.

Arduino IDE setup->File->Preferences->Additional Boards Manager URLs:

https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/arduino/package_m5stack_index.json
https://dl.espressif.com/dl/package_esp32_index.json

Arduino IDE setup->Board:"M5Stack-ATOM", Partition scheme "No OTA"

Compile the sources files with Arduino IDE (<https://www.arduino.cc/>).

Setting WIFI name, WIFI password, and other parameters of M5 Stack ATOM Lite device

Connect "M5Stack ATOM Lite" to computer USB. USB to COM port settings, standard, just select 115200 bits/second. Follow the terminal hints.

WIFI name, password, DDNS name https command and three public IP addresses can be set or modified through Serial port connection during first run time (30 seconds timeout). They becomes default (saved in Nonvolatile memory). For serial port programming, I recommend free software Terminate.exe (<https://termite.software.informer.com/3.2/>).

Then you must to set GET request string to DDNS. If you get a token, the request looks like browser command line, as an example:

"https://www.duckdns.org/update?domains=PUT_YOUR_NAME&token=e2403930-f58a-4237-a162-16f28444d8f7"). You can insert 'MY_PUBLIC_IP_ADDR' substring in https request. Then the substring automatically is replaced with your router real public IP address: xx:xx:xx:xx.

The request format is universal and lets you to send any https command to any DDNS server.

Do not forget to set ports forwarding of your home router for IoT controllers or/and cameras which you would like to control over Internet. See my example for M5Stack-Timer-Camera X:

https://github.com/mumris/M5Stack-Timer-Camera-X/tree/master/examples/M5%20Web_cam_timer%20X