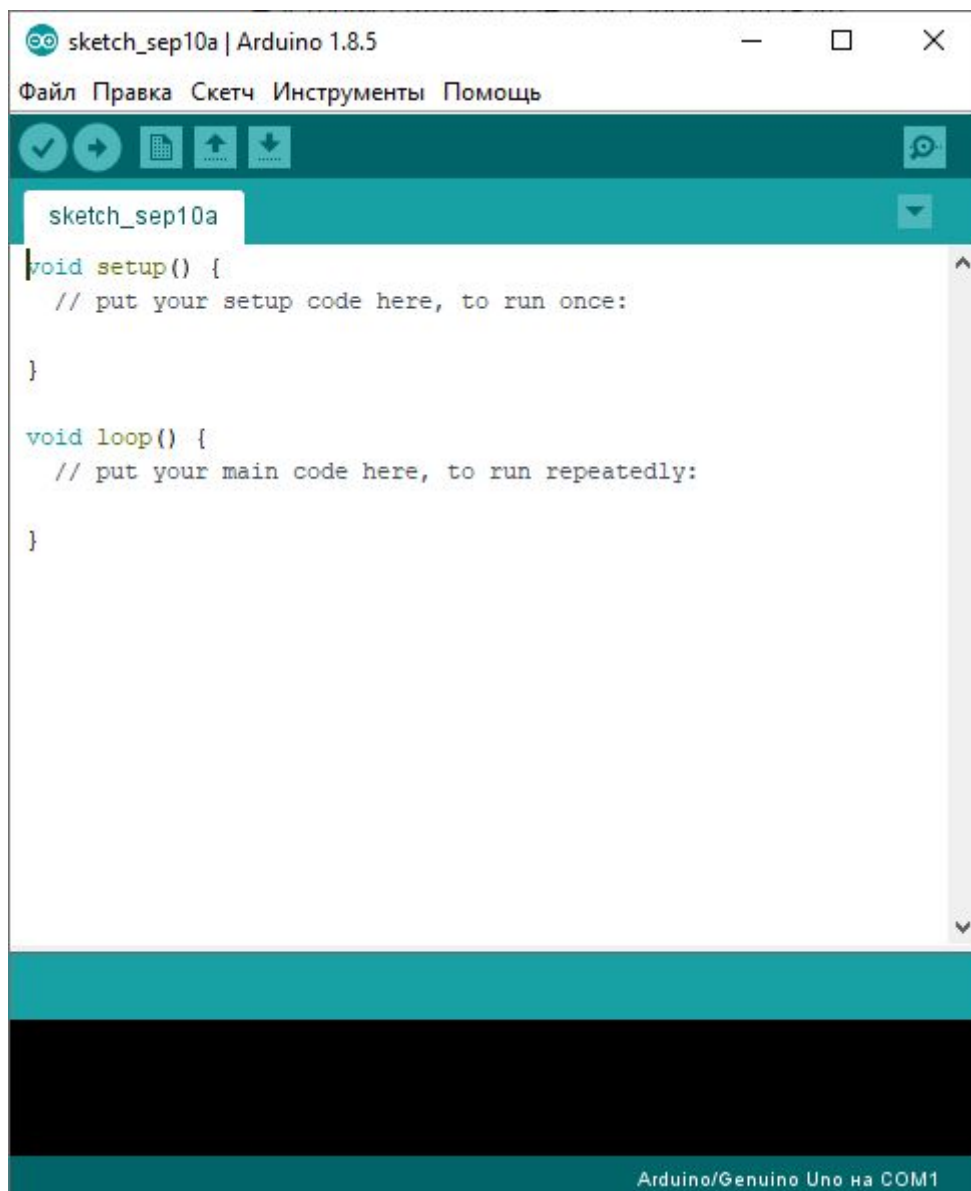
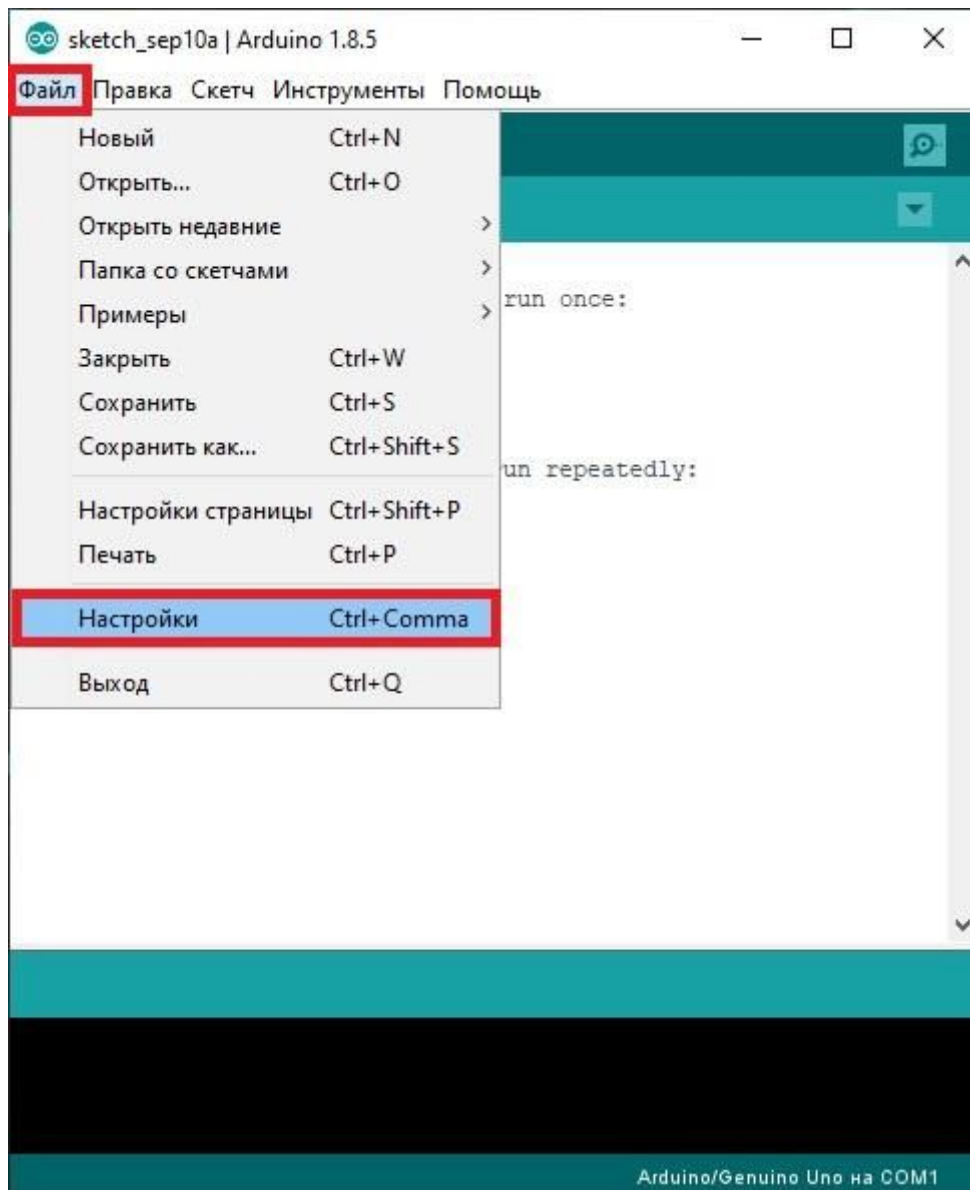


## 1) Первый запуск



## 2) Идем в настройки



3) Настраиваем как на скриншоте и указываем адрес

[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)

Настройки

Настройки Сеть

Размещение папки скетчей

C:\Arduino\sketchbook Обзор

Язык редактора: По умолчанию (нужен перезапуск Arduino IDE)

Размер шрифта: 12

Масштаб интерфейса: ☒ Автоматика 100% (нужен перезапуск Arduino IDE)

Показать подробный вывод: ☐ Компиляция ☐ Загрузка

Сообщения компилятора: Ничего

☐ Показать номера строк

☐ Включить сворачивание кода

☒ Проверять код после загрузки

☐ Использовать внешний редактор

☒ Агрессивное кэширование скомпилированного ядра

☒ Проверять обновления при запуске

☒ Конвертировать файлы скетчей в новый формат (.pde -> .ino)

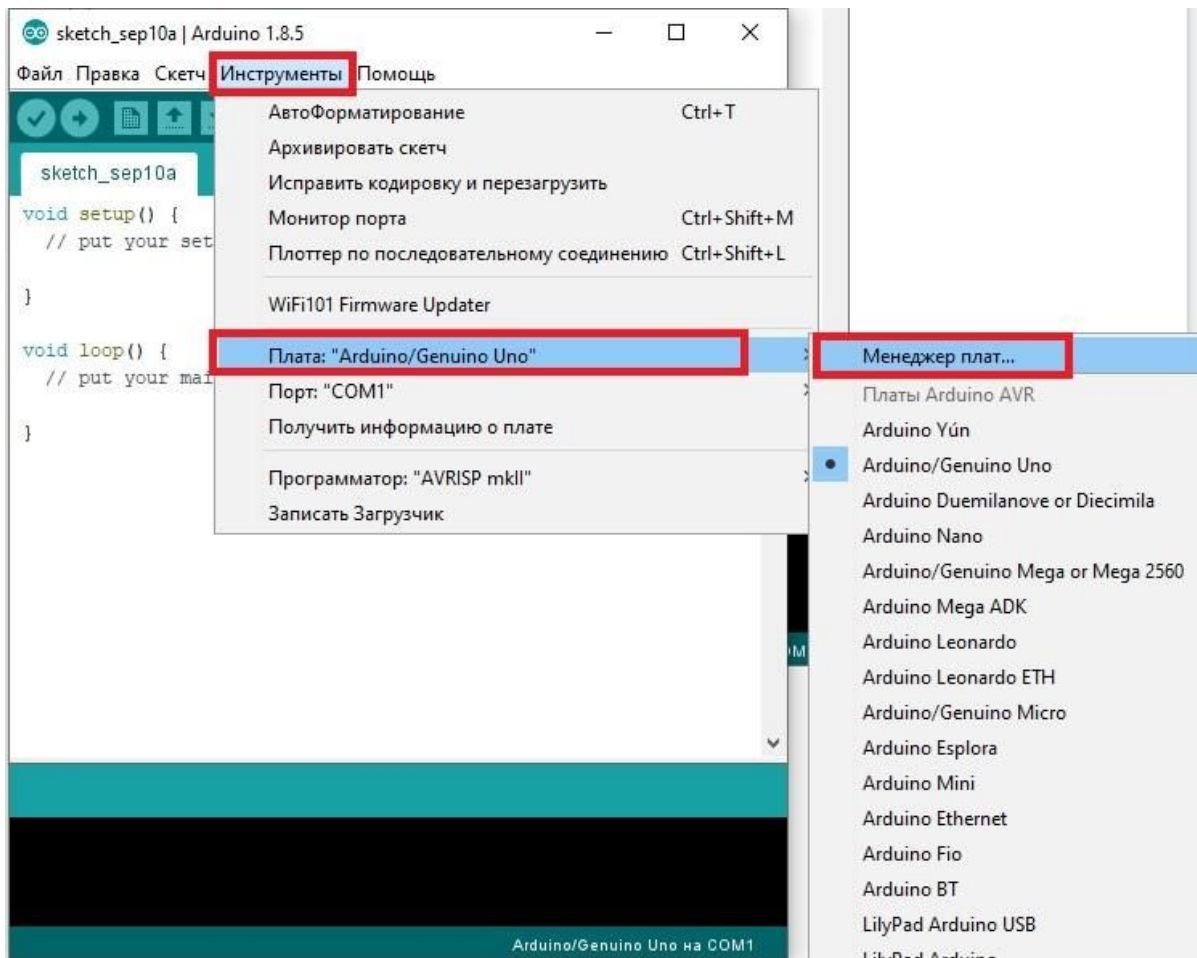
☒ Сохранять скетч при проверке или компиляции

Дополнительные ссылки для Менеджера плат: [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)

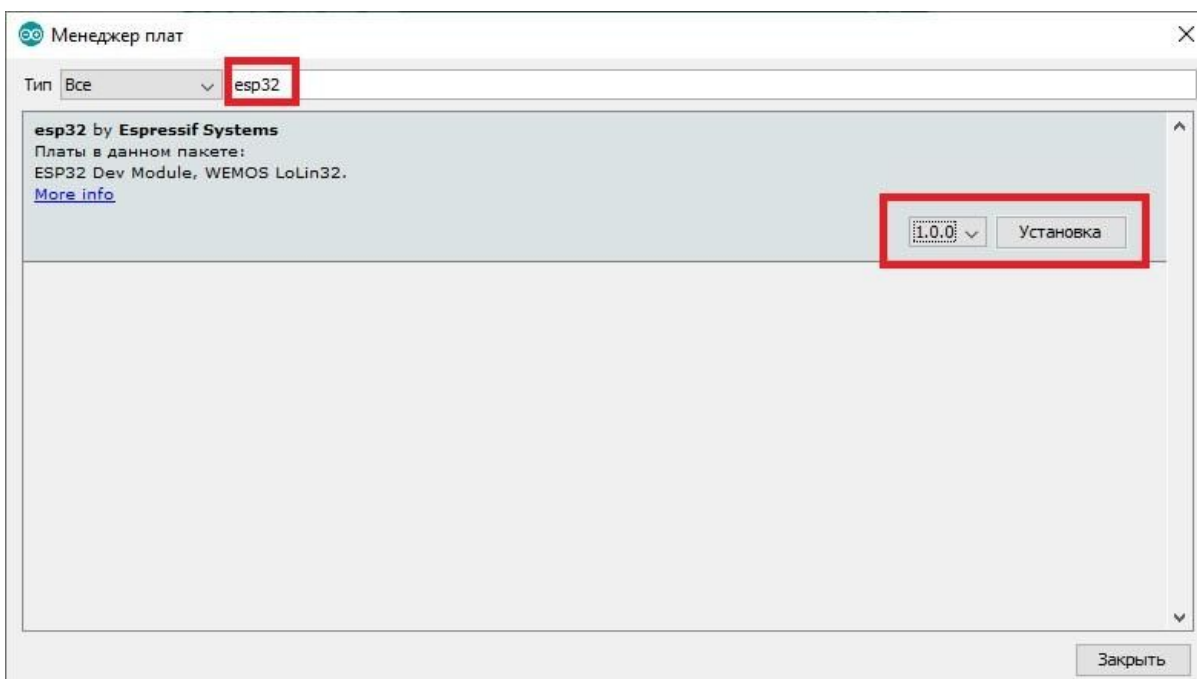
Другие настройки можно редактировать непосредственно в файле:  
C:\Users\...\AppData\Local\Arduino15\preferences.txt  
(только когда Arduino IDE не запущена)

OK Отмена

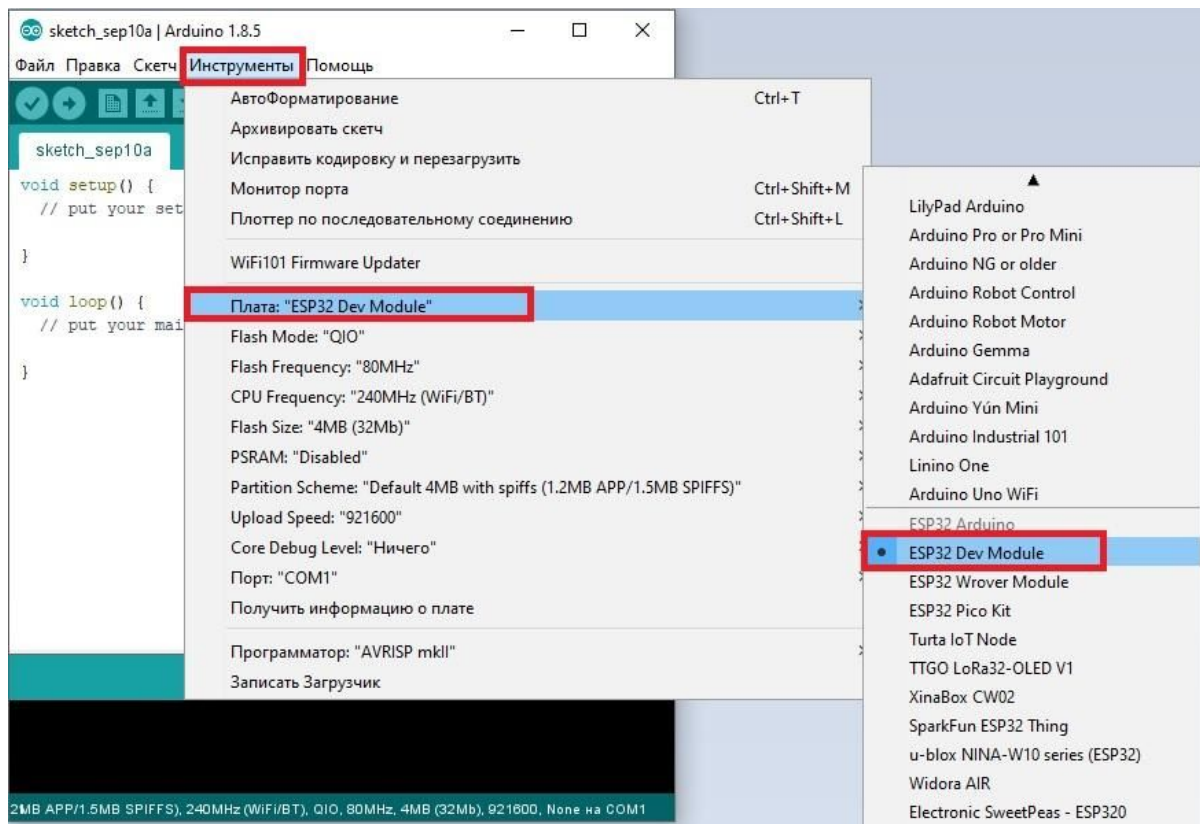
#### 4) Устанавливаем плату через Менеджер плат



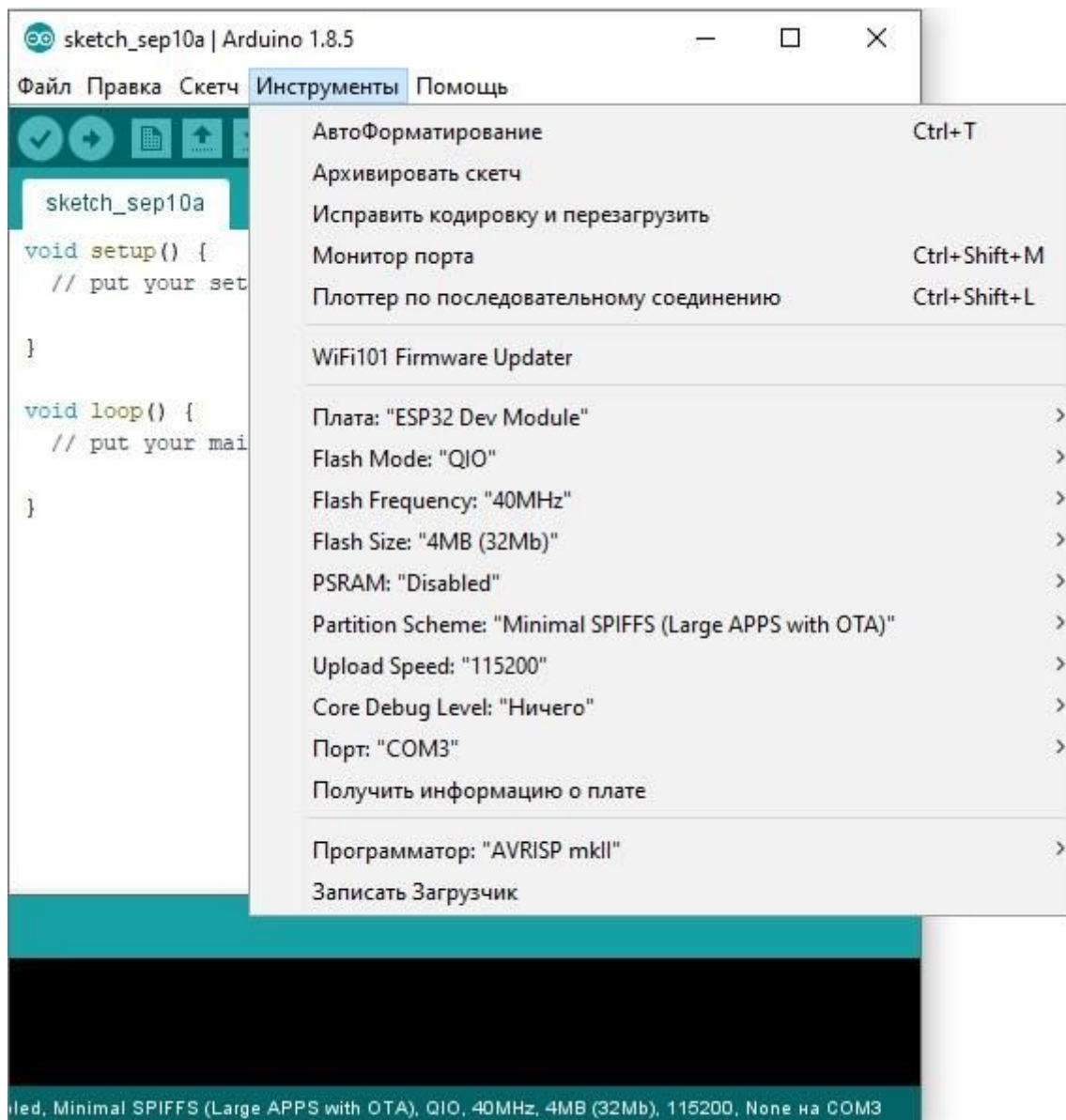
5) В строке пишем esp32 и устанавливаем строго версию [1.0.0](#), если установите версию выше, то установка скетча пройдет не успешно



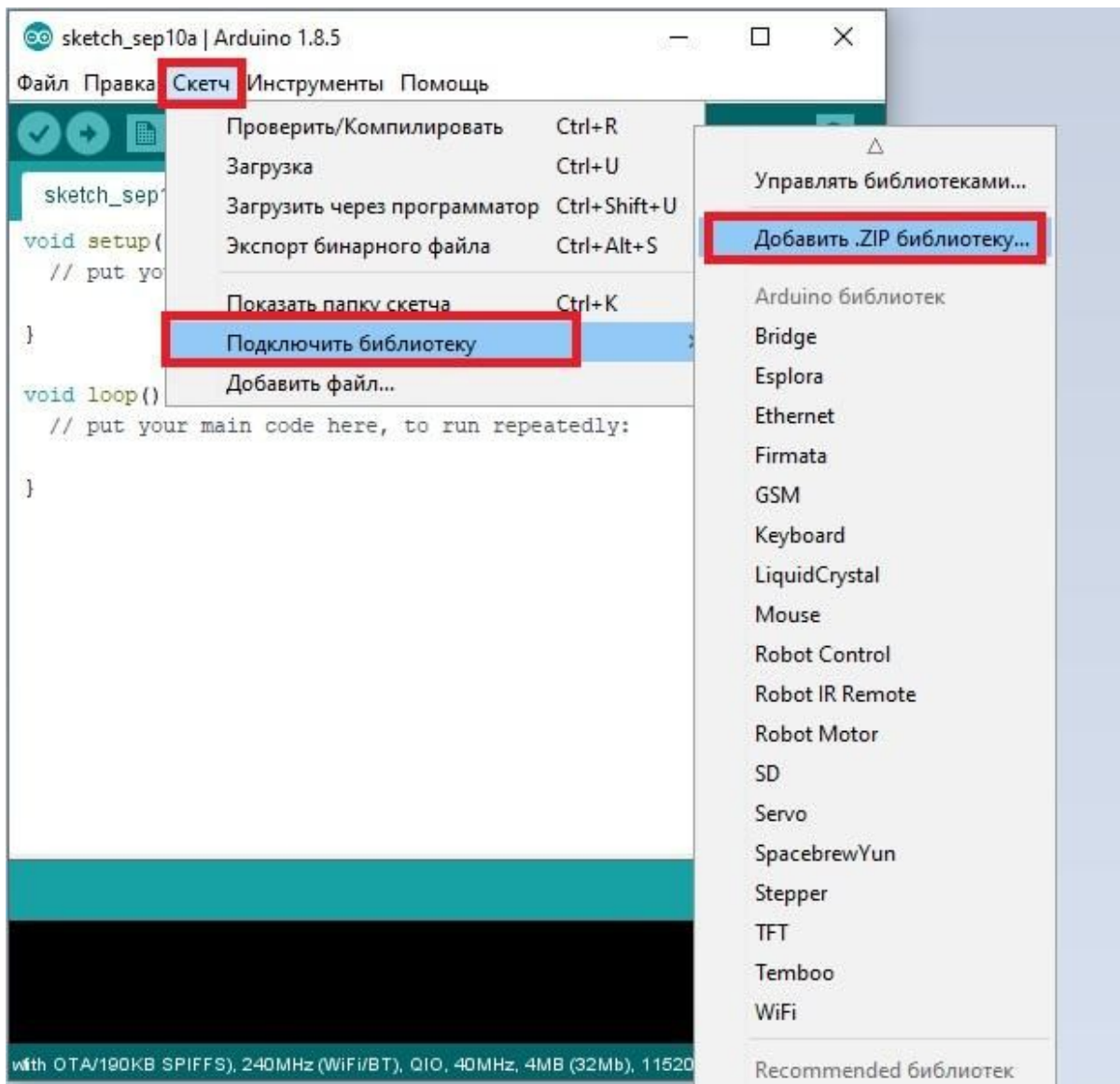
## 6) Выбираем плату ESP32 Dev Module



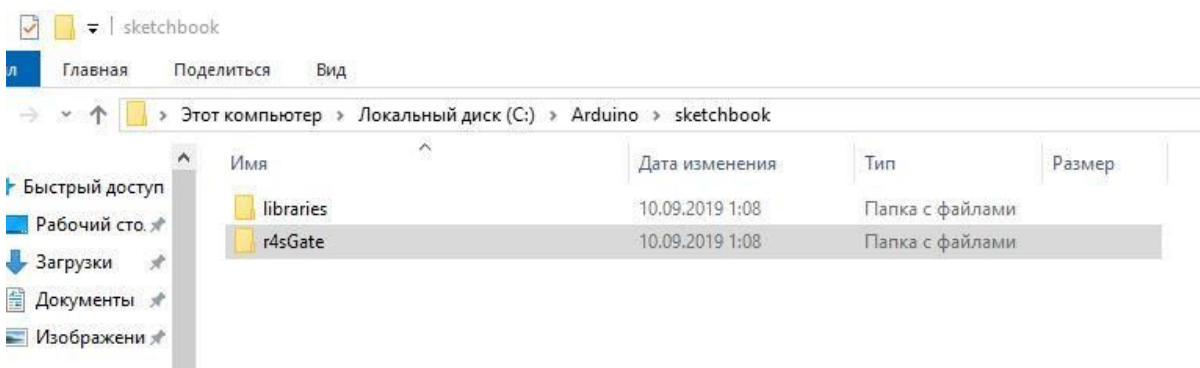
## 7) Мои настройки платы ESP32 Dev Module



8) Добавляем библиотеки: ESP32\_BLE\_Arduino-master, WebServer-esp32-master, PubSubClient

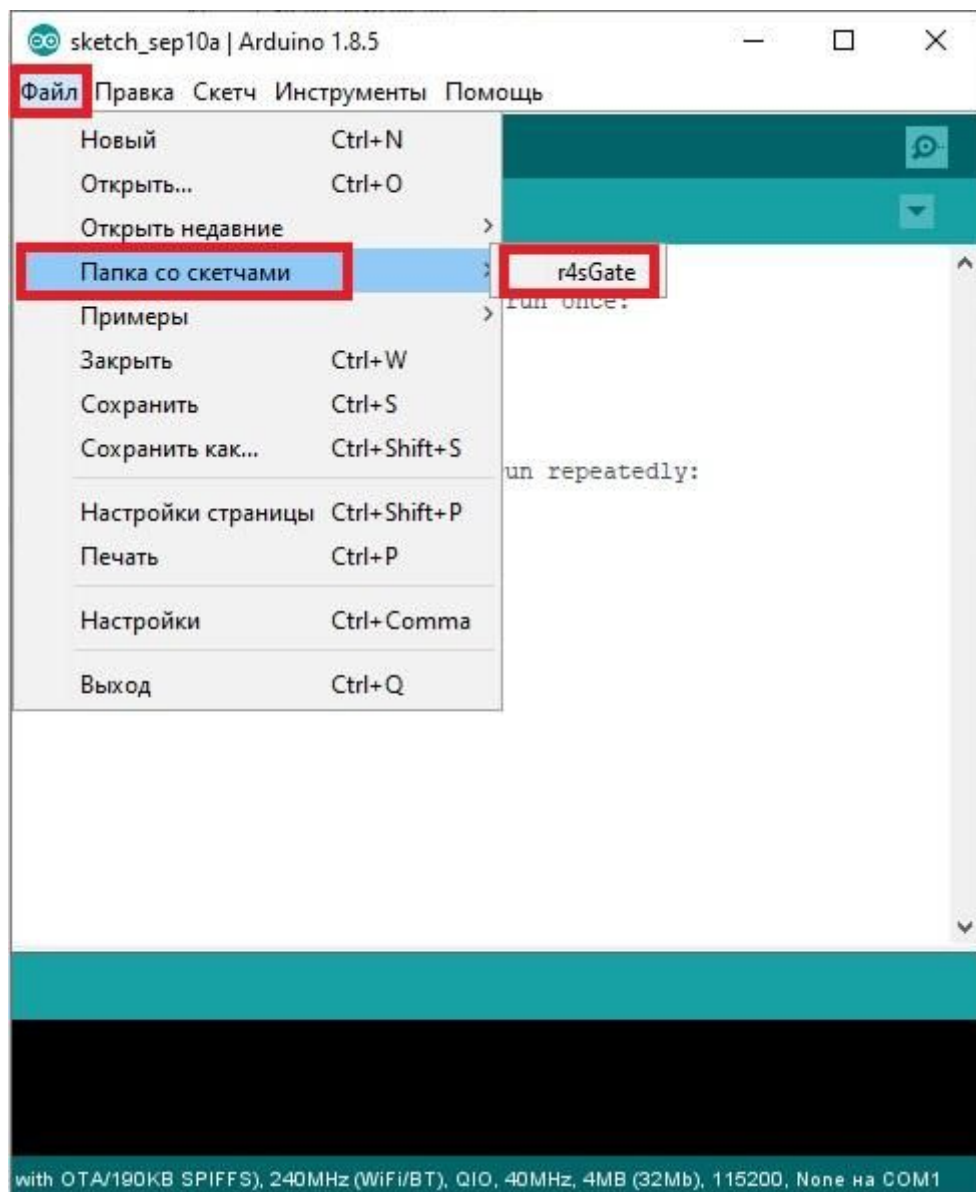


9) Копируем в папку sketchbook скетч r4sGate



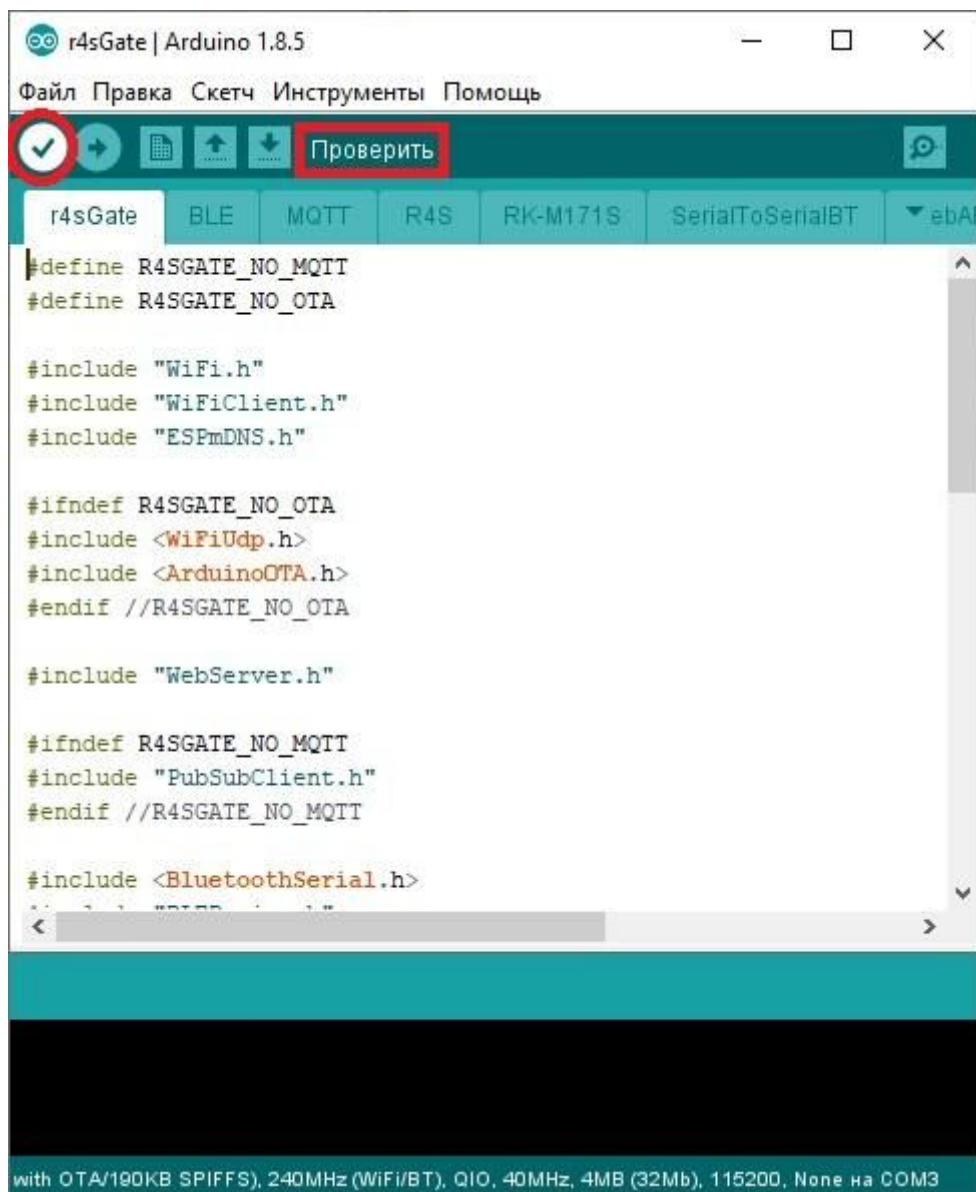


10) Перезапускаем Arduino и после этого мы увидим скетч r4sGate

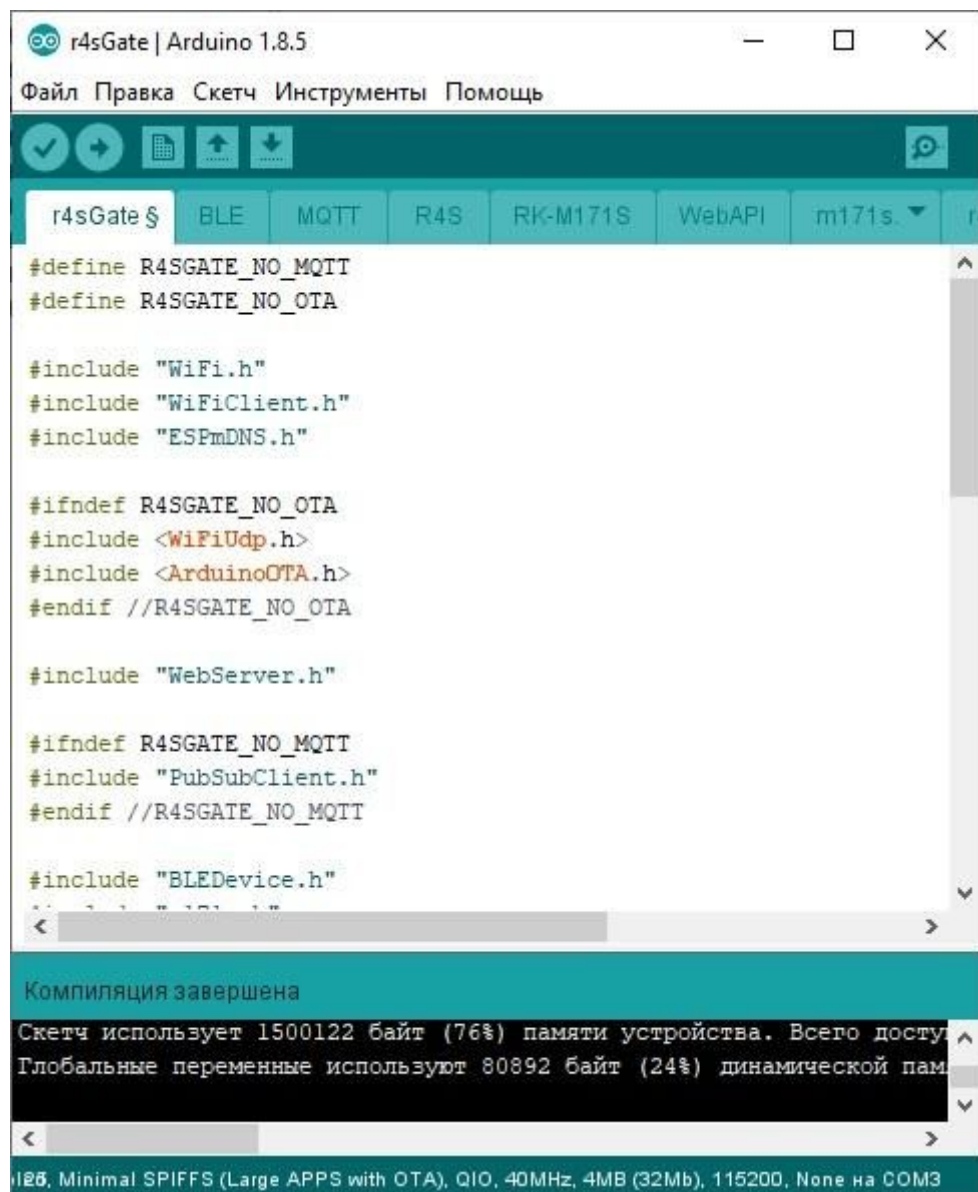




11) Перед загрузкой скетча r4sGate, его нужно проверить, нажимаем проверить скетч r4sGate



12) Проверка скетча r4sGate прошла успешно



The screenshot shows the Arduino IDE interface with the 'r4sGate' sketch open. The menu bar includes 'Файл', 'Правка', 'Скетч', 'Инструменты', and 'Помощь'. The toolbar contains icons for saving, running, and uploading. The 'Sketch' menu is open, showing options like 'BLE', 'MQTT', 'R4S', 'RK-M171S', 'WebAPI', and 'm171s'. The main editor displays the sketch code, which includes preprocessor directives and library includes. The status bar at the bottom indicates that the compilation is complete and provides memory usage statistics.

```
#define R4SGATE_NO_MQTT
#define R4SGATE_NO_OTA

#include "WiFi.h"
#include "WiFiClient.h"
#include "ESPmDNS.h"

#ifndef R4SGATE_NO_OTA
#include <WiFiUdp.h>
#include <ArduinoOTA.h>
#endif //R4SGATE_NO_OTA

#include "WebServer.h"

#ifndef R4SGATE_NO_MQTT
#include "PubSubClient.h"
#endif //R4SGATE_NO_MQTT

#include "BLEDevice.h"
#include "BLESerial.h"
```

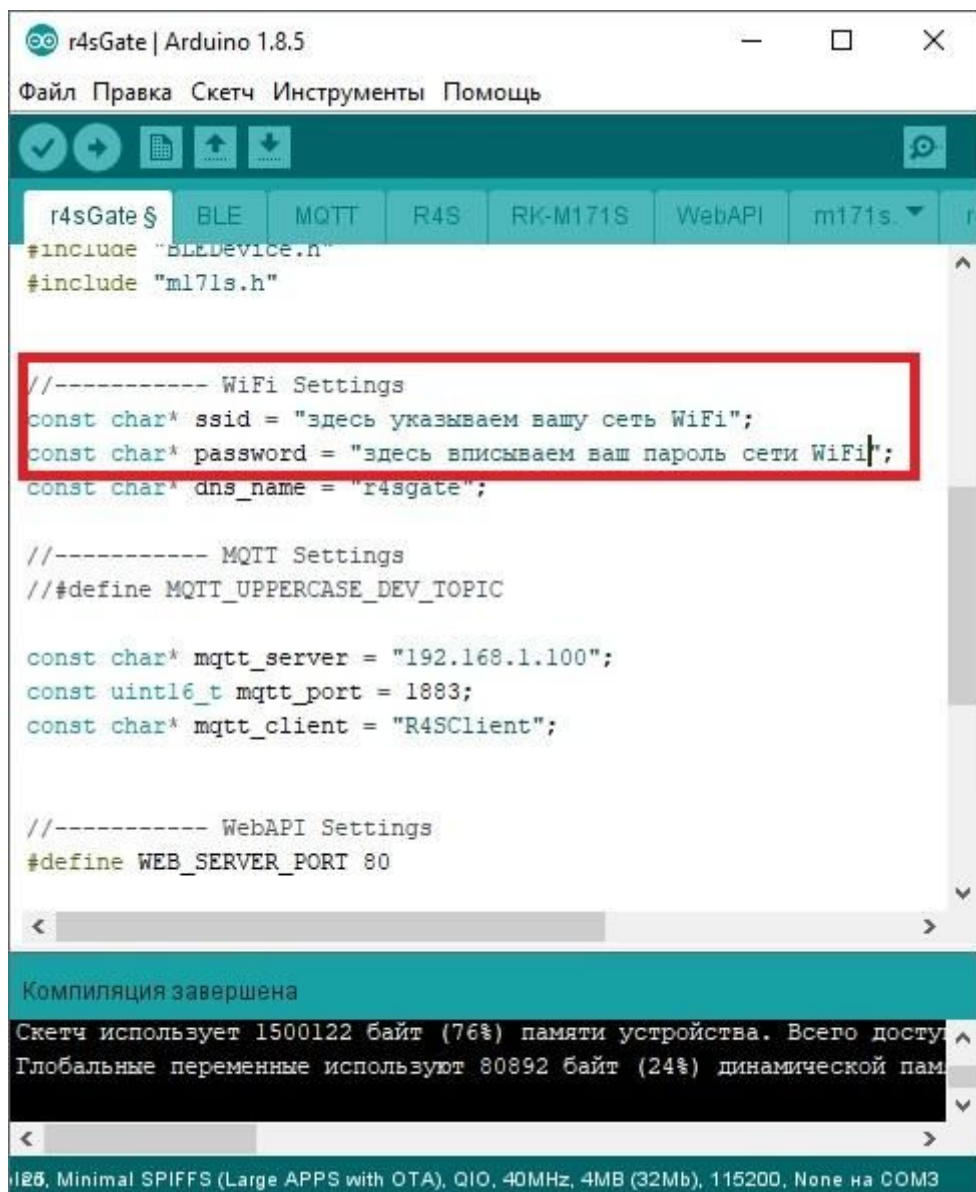
Компиляция завершена

Скетч использует 1500122 байт (76%) памяти устройства. Всего доступно 1967360 байт.

Глобальные переменные используют 80892 байт (24%) динамической памяти. Доступно еще 268108 байт.

ATmega328P, Minimal SPIFFS (Large APPS with OTA), QIO, 40MHz, 4MB (32Mb), 115200, None на COM3

13) Здесь мы вписываем настройки WiFi, имя сети и пароль



```
r4sGate | Arduino 1.8.5
Файл Правка Скетч Инструменты Помощь

r4sGate $ BLE MQTT R4S RK-M171S WebAPI m171s. r4
#include "BLEDevice.h"
#include "m171s.h"

//----- WiFi Settings
const char* ssid = "здесь указываем вашу сеть WiFi";
const char* password = "здесь вписываем ваш пароль сети WiFi";
const char* dns_name = "r4sgate";

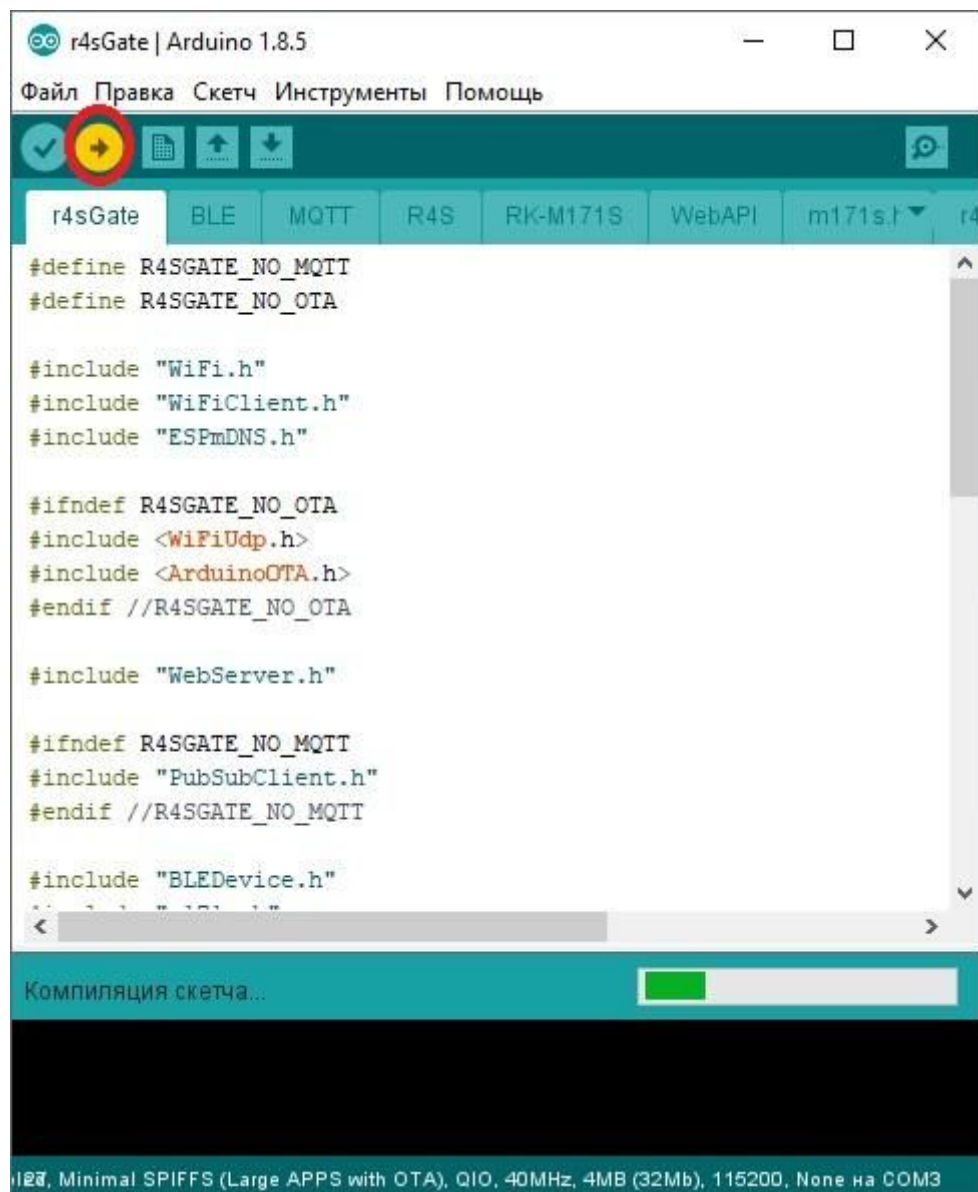
//----- MQTT Settings
//#define MQTT_UPPERCASE_DEV_TOPIC

const char* mqtt_server = "192.168.1.100";
const uint16_t mqtt_port = 1883;
const char* mqtt_client = "R4SClient";

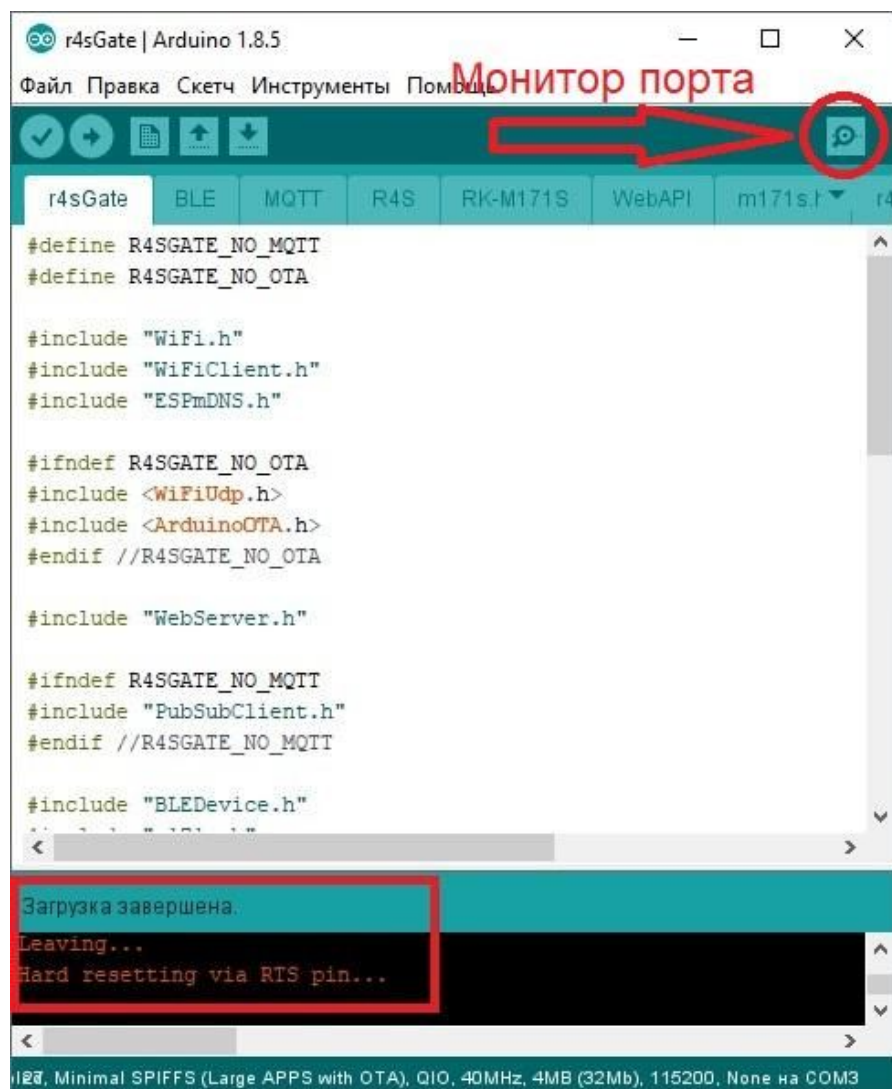
//----- WebAPI Settings
#define WEB_SERVER_PORT 80

Компиляция завершена
Скетч использует 1500122 байт (76%) памяти устройства. Всего досту
Глобальные переменные используют 80892 байт (24%) динамической пам
128, Minimal SPIFFS (Large APPS with OTA), QIO, 40MHz, 4MB (32Mb), 115200, None на COM3
```

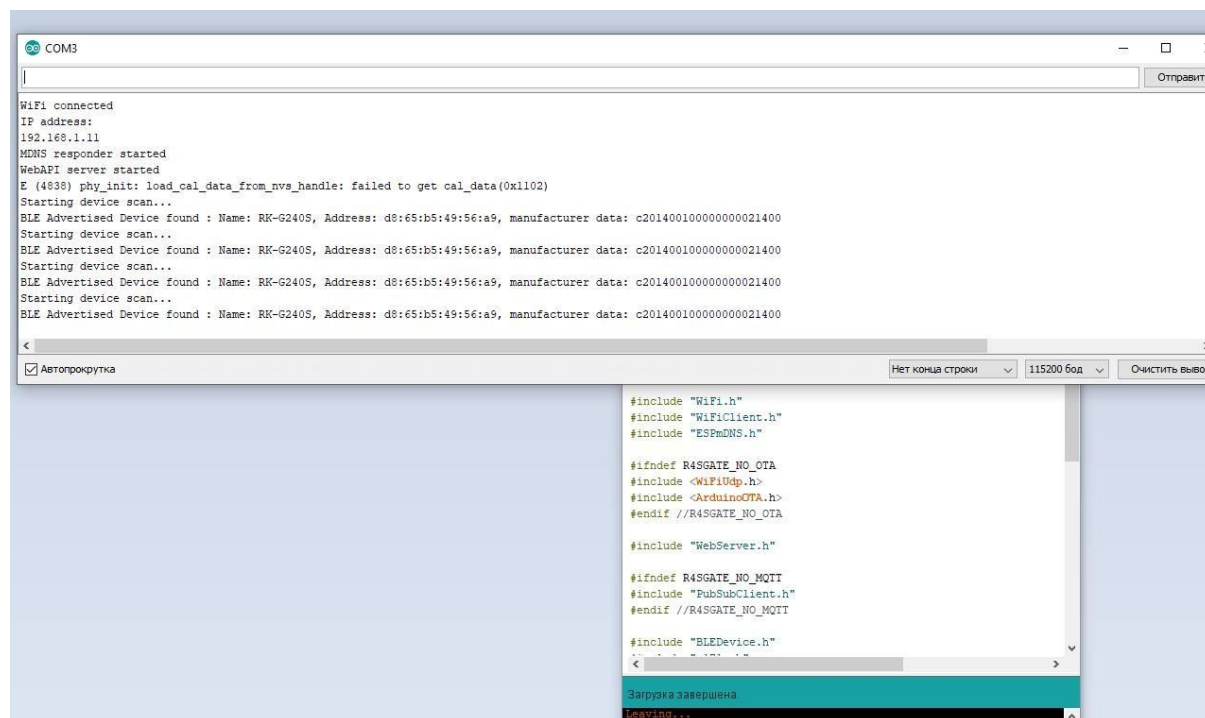
#### 14) Загружаем скетч r4sGate



15) Загрузка скетча r4sGate прошла успешно, о чем нам сообщают ниже, нажимаем на Мониторинг порта и нажимаем на плате на кнопку EN



16) Если все прошло нормально, то на мониторе порта вы увидите это



The screenshot shows a serial port monitor window titled 'COM3'. The top section displays a log of system messages, including WiFi connection status, IP address (192.168.1.11), MDNS responder start, WebAPI server start, and a phy\_init error. It also shows four successful BLE device scans for an RK-G2405. The bottom section is a code editor with a light blue background, displaying C++ preprocessor directives for WiFi, MQTT, and BLE. The status bar at the bottom indicates 'Загрузка завершена' (Loading completed) and 'Leaving...'. The window includes standard controls like 'Отправить' (Send), 'Автопрокрутка' (Autoscroll), and 'Очистить вывод' (Clear output).

```
COM3
WiFi connected
IP address:
192.168.1.11
MDNS responder started
WebAPI server started
E (4838) phy_init: load_cal_data_from_nvs_handle: failed to get cal_data(0x1102)
Starting device scan...
BLE Advertised Device found : Name: RK-G2405, Address: d8:65:b5:49:56:a9, manufacturer data: c201400100000000021400
Starting device scan...
BLE Advertised Device found : Name: RK-G2405, Address: d8:65:b5:49:56:a9, manufacturer data: c201400100000000021400
Starting device scan...
BLE Advertised Device found : Name: RK-G2405, Address: d8:65:b5:49:56:a9, manufacturer data: c201400100000000021400
Starting device scan...
BLE Advertised Device found : Name: RK-G2405, Address: d8:65:b5:49:56:a9, manufacturer data: c201400100000000021400

#include "WiFi.h"
#include "WiFiClient.h"
#include "ESPmDNS.h"

#ifdef R4SGATE_NO_OTA
#include <WiFiUdp.h>
#include <ArduinoOTA.h>
#endif //R4SGATE_NO_OTA

#include "WebServer.h"

#ifdef R4SGATE_NO_MQTT
#include "PubSubClient.h"
#endif //R4SGATE_NO_MQTT

#include "BLEDevice.h"

Загрузка завершена
Leaving...
```