ARDULOGIC Главная Блог О проекте



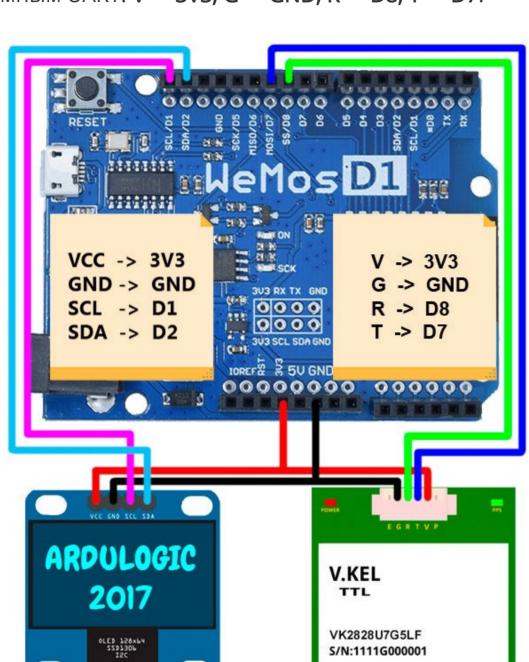
В одной из прошлых частей мы собрали простой спидометр с возможностью замера разгона до сотни, сегодня мы расширим его функционал. На борту платы Wemos D1, благодаря чипу ESP8266, есть полноценный WiFi, так давайте же задействуем его, создадим web сервер и будем передавать результаты замеров на телефон.

Подключение

Подключаем все модули абсолютно также, как в прошлый раз:

1. Соединим дисплей с шиной I2C: VCC -> 3V3, GND -> GND, SCL -> D1, SDA -> D2.

Соединим дисплеи с шинои I2C: VCC -> 3V3, GND -> GND, SCL -> D1, SDA -> D
 A GPS модуль с программным UART: V -> 3V3, G -> GND, R -> D8, T -> D7.



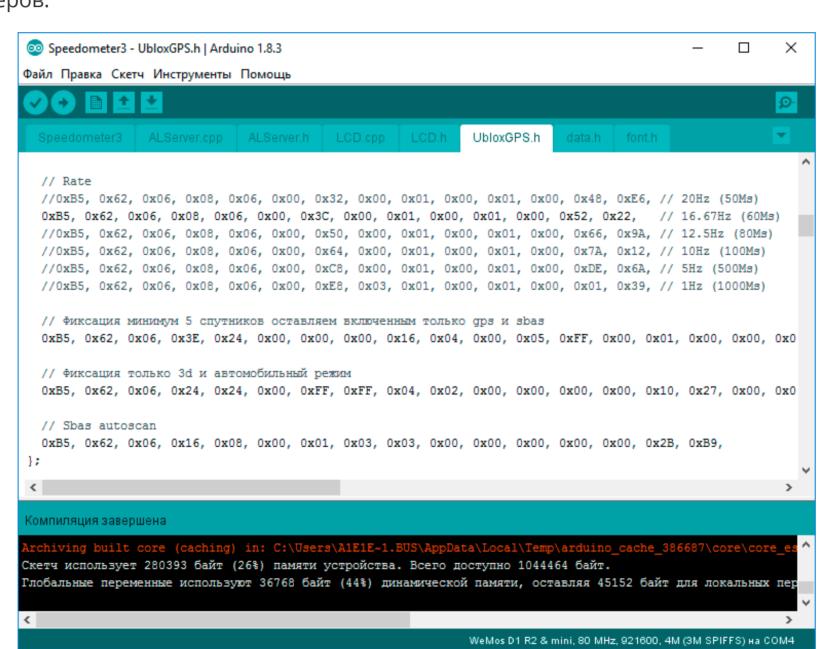
Замер разгона автомобиля Подключение Web сервер на ESP8266 Дисплей

```
ESP8266 MPU6050 NEO-7M

OLED 128x64 Wemos D1
```

Web сервер на ESP8266

В этот раз мы выжмем максимум из NEO-7, в настройках зададим рейт обновления 16 раз в секунду, минимум 5 спутников для фиксации и автомобильный режим. Эти настройки позволяют повысить точность замеров.

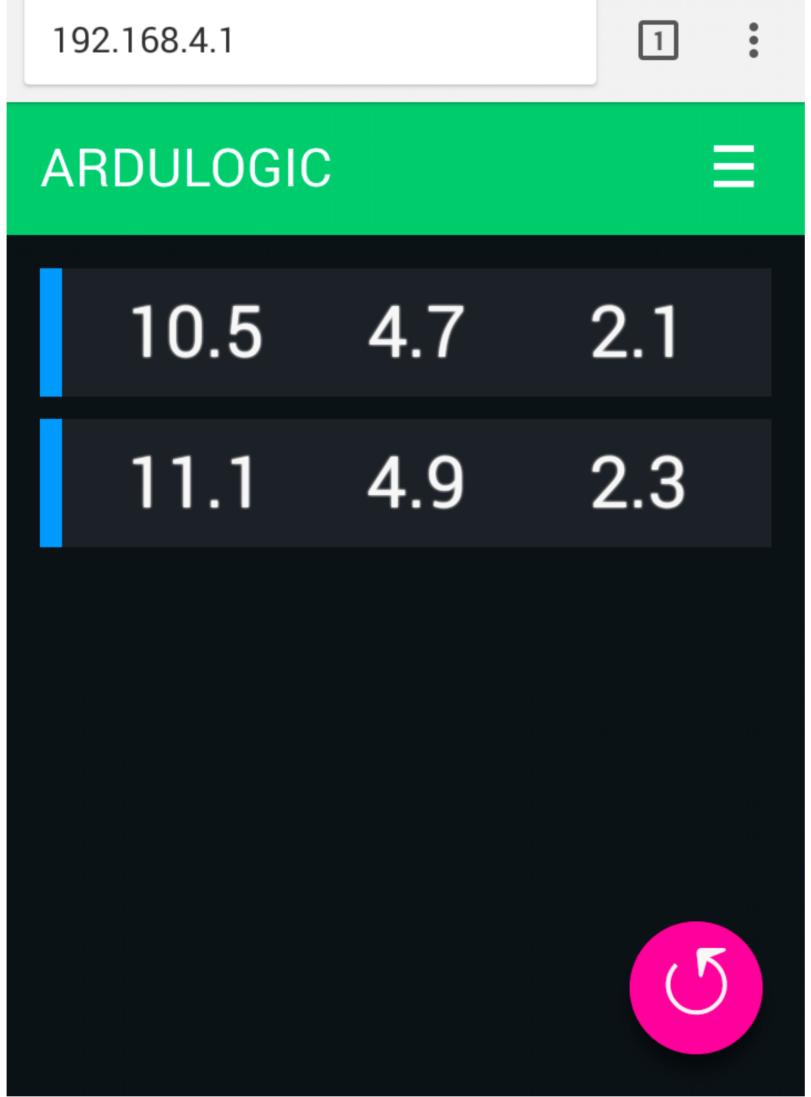


Создать web-сервер на ESP8266 проще простого. Необходимо указать на каком порту он будет ждать подключения, для http это традиционно 80й порт, и указать, по каким запросам, какую функцию выполнять. Я подготовил небольшую html страничку, которая будет запрашивать последние 10 результатов замеров и отображать их на экране. Для максимального быстродействия результаты будут отправляться в формате json.

```
const char *ssid = "ARDULOGIC"; // Название сети WiFi
    const char *password = "1234567890"; // Пароль для подключения
    ESP8266WebServer server(80);
 6 void ALServer::serverHandle()
        server.handleClient(); // Ждём подключения
11 void ALServer::createAP()
12 - {
13
        WiFi.softAP(ssid, password); // Создаём точку WiFi
14
        server.on("/", [this]() {
15 -
           handleRoot();
17
18 -
        server.on("/style.css", [this]() {
           handleStyle();
20
        server.on("/results", [this]() {
21 -
22
           handleResults();
23
        server.begin();
25 }
28 void ALServer::handleRoot()
```

Скачать архив со скетчем: Speedometer3.zip

Теперь если подключиться к устройству по WiFi, после замера можно увидеть такую картину.



ДИСПЛЕЙВ предыдущем варианте дисплей отображал только скорость и результат последнего замера. Давайте

добавим в него больше информации. Для удобства отладки будем выводить на экран количество спутников, точность позиционирования, время и координаты положения. Добавим в код новую структуру для хранения этой информации и немного изменим вывод на экран.

1 struct FullData

```
2 = {
           unsigned int numSV; // Спутники
           float hAcc;
           char gpsTime[9];
           char bufLatitude[10];
           char bufLongitude[10];
           unsigned int gSpeedKm = 0; // Скорость
   9 };
  void LCD::updateScreen(FullData* fullData, Metering* metering)
  12 - {
  13
           display->clear();
          display->drawVerticalLine(7, 2, 8);
  15
           display->drawVerticalLine(6, 2, 8);
  17
          display->drawVerticalLine(4, 5, 5);
          display->drawVerticalLine(3, 5, 5);
  18
  19
           display->drawVerticalLine(1, 8, 2);
          display->drawVerticalLine(0, 8, 2);
  20
  21
           display->drawHorizontalLine(0, 11, 128);
  22
           display->drawHorizontalLine(0, 54, 128);
  23
          display->setTextAlignment(TEXT_ALIGN_LEFT);
  25
           display->setFont(ArialMT_Plain_10);
           display->drawString(11, 0, (String)fullData->numSV);
  26
           display->setTextAlignment(TEXT_ALIGN_LEFT);
Вот что у нас получится в итоге.
```

