**Общие сведения:**

Разработка системы управления на основе Arduino с использованием набора датчиков, Ethrenet Shield и жидкокристаллического экрана для обеспечения мониторинга данных с заданного набора датчиков через сеть Internet. Система должна выступать в роли интернет-устройства сбора данных с удалённым доступом по протоколу HTTP (Arduino-библиотека Ethernet или Ethernet2 в зависимости от версии платы, проект WebServer). Должен поддерживаться запрос к устройству через Интернет-браузер.

Данные с датчиков должны также указанным интервалом отображаться на подключённом жидкокристаллическом экране.

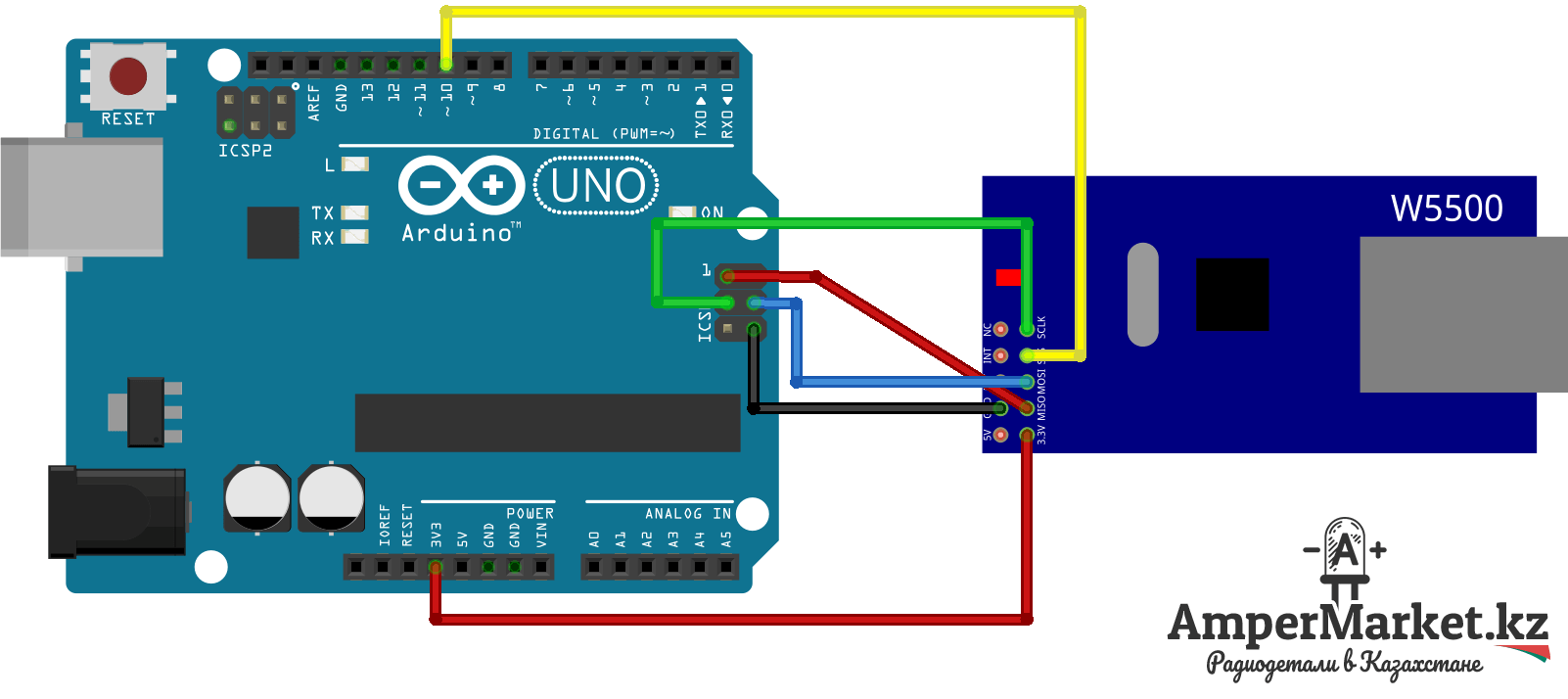
Общий алгоритм работы:

Контроллер регулярно опрашивает датчики и выводит информацию на ЖК-экран в читаемом виде.

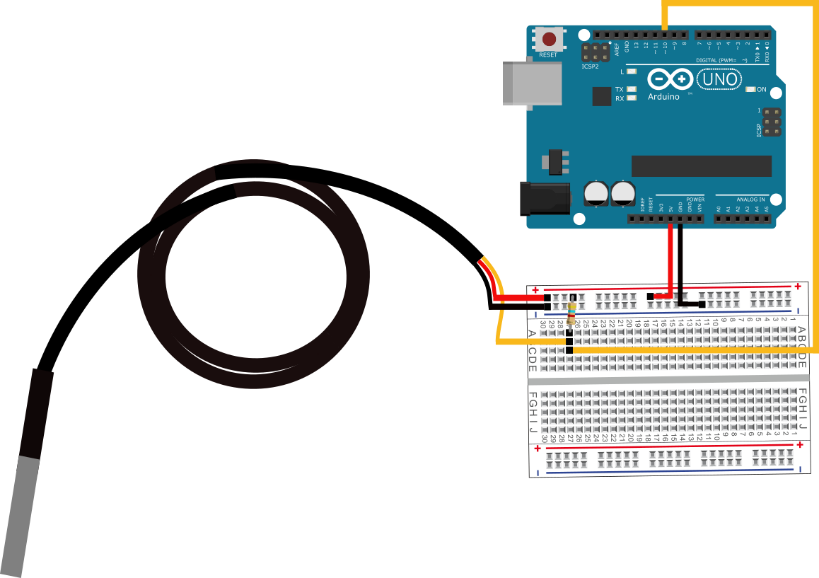
После поступления запроса от браузера включается светодиод на 1 секунду (не блокируя основную логику программы) и отправляется ответ в виде «Идентификатор устройства \n название датчика=значение» в формате html страницы.

**Описание ходов работы:**

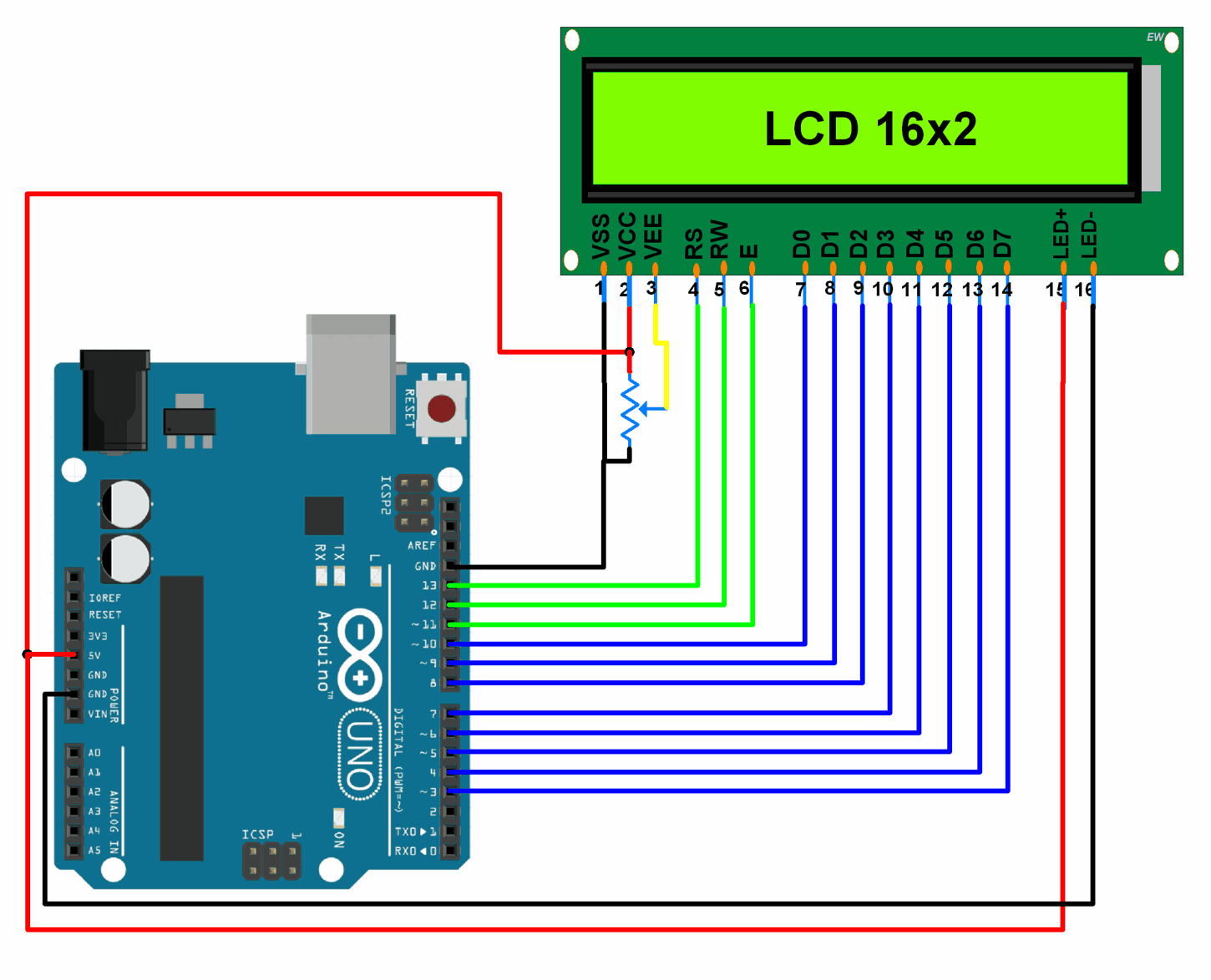
1. Для начала я подключил модуль etherner, Oled дисплей, и датчик влажности и температуры к ардуино.
2. Ethernet модуль подключен к пинам D10-D13, D2, GND, 5V, REST. Рисунок номер 1 показывает, как ethernet модуль подключается к ардуино.

 рис. 1

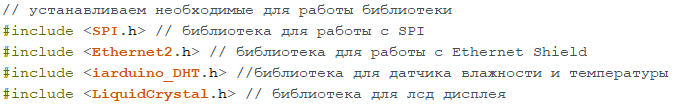
1. Датчик влажности и температуры к пинам D4, V5, GND. Рисунок номер 2 показывает, как датчик температуры и влажности подключается к ардуино.

 рис. 2

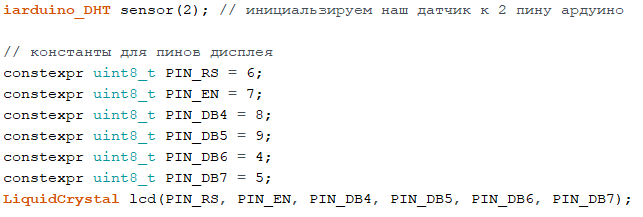
1. Oled дисплей к пинам D8, D9, GND, V5. Рисунок номер 4 показывает, как дисплей подключается к ардуино.

рис. 3

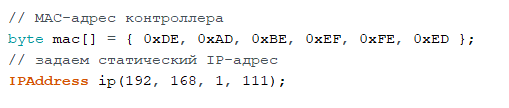
1. устанавливаем все необходимые библиотека для работы



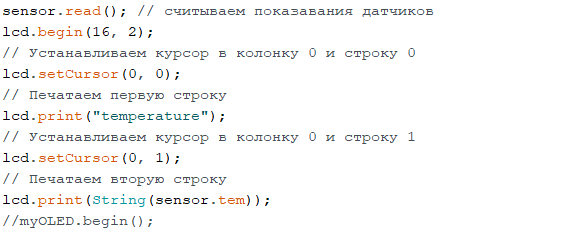
1. инициализируем все необходимые для работы пины



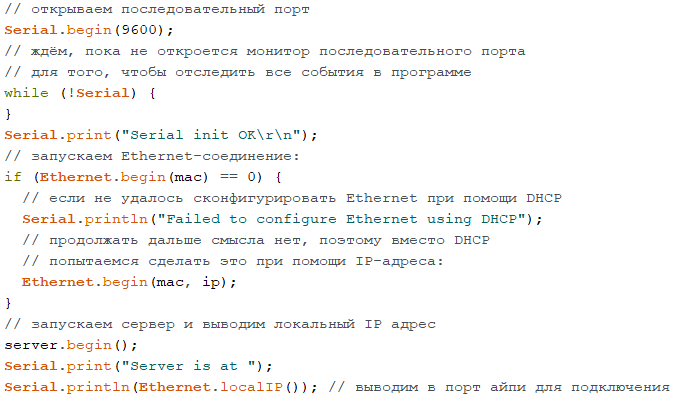
1. указываем ip устройства и mac address



1. считываем показания датчика и выводим их на лсд дисплее



1. открываем последовательный порт и в случае неверного указания мак адреса выводим ошибку в порт, а также выводим айпи устройства и подключаем сервер



1. запускаем наш веб сервер и выводим всю необходимую информацию



11)Вывод на web-станицу данные датчика. Рис. 4.

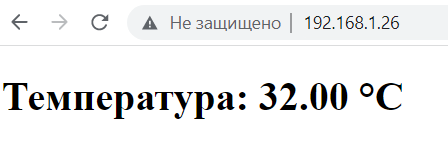
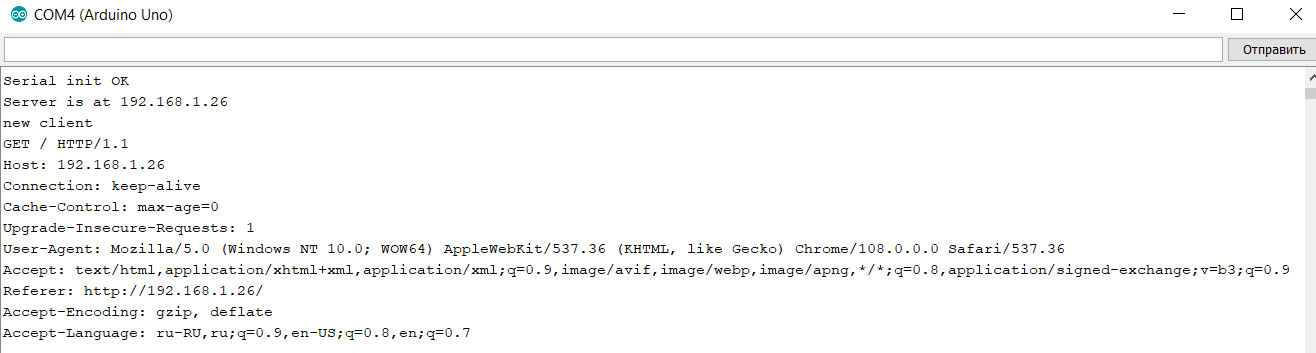


рис. 4

12)Смотрим данные из ком порта, рис. 5.

 рис. 5

13)Полная схема подключения изображена на рис. 7

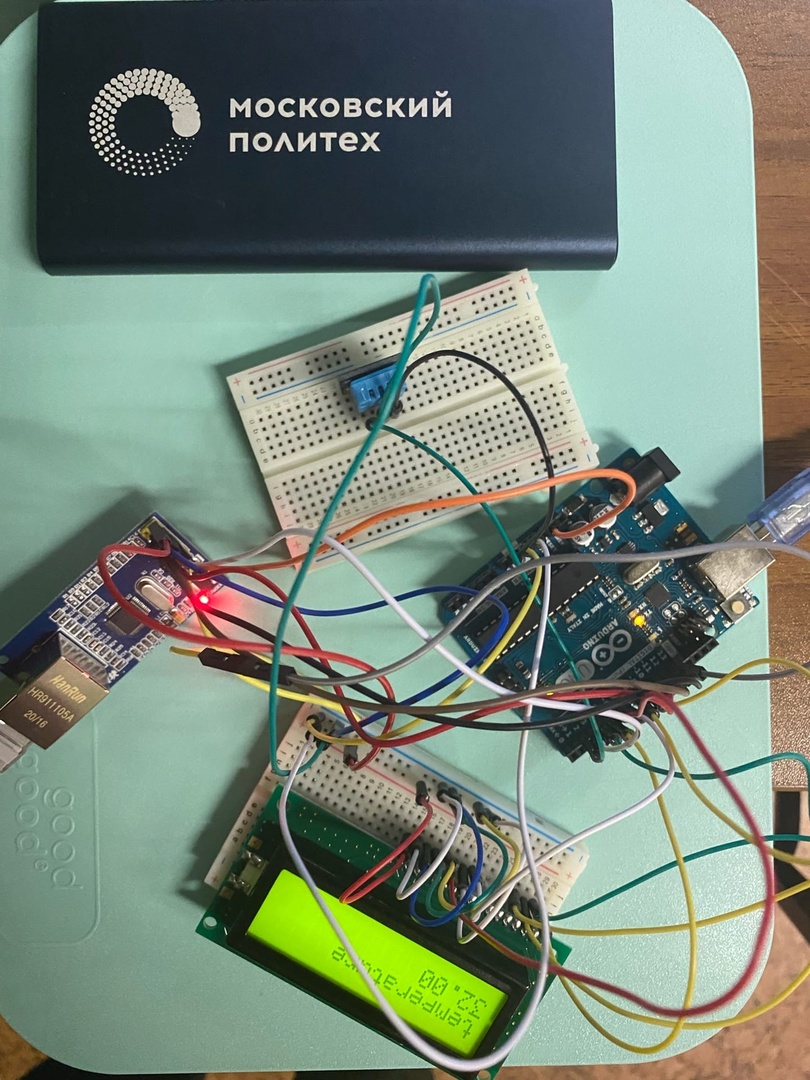


рис. 7

**ПриложениеA**

Листинг А.1 – Код программы обработки данных

// устанавливаем необходимые для работы библиотеки

#include <SPI.h> // библиотека для работы с SPI

#include <Ethernet2.h> // библиотека для работы с Ethernet Shield

#include <iarduino\_DHT.h> //библиотека для датчика влажности и температуры

#include <LiquidCrystal.h> // библиотека для лсд дисплея

iarduino\_DHT sensor(2); // инициальзируем наш датчик к 2 пину ардуино

// константы для пинов дисплея

constexpr uint8\_t PIN\_RS = 6;

constexpr uint8\_t PIN\_EN = 7;

constexpr uint8\_t PIN\_DB4 = 8;

constexpr uint8\_t PIN\_DB5 = 9;

constexpr uint8\_t PIN\_DB6 = 4;

constexpr uint8\_t PIN\_DB7 = 5;

LiquidCrystal lcd(PIN\_RS, PIN\_EN, PIN\_DB4, PIN\_DB5, PIN\_DB6, PIN\_DB7);

// MAC-адрес контроллера

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

// задаем статический IP-адрес

IPAddress ip(192, 168, 1, 111);

// инициализируем библиотеку Ethernet Server

// с необходимыми вам IP-адресом и портом

// порт 80 используется для HTTP по уzмолчанию

EthernetServer server(80);

void setup()

{

sensor.read(); // считываем показавания датчиков

lcd.begin(16, 2);

// Устанавливаем курсор в колонку 0 и строку 0

lcd.setCursor(0, 0);

// Печатаем первую строку

lcd.print("temperature");

// Устанавливаем курсор в колонку 0 и строку 1

lcd.setCursor(0, 1);

// Печатаем вторую строку

lcd.print(String(sensor.tem));

//myOLED.begin();

// открываем последовательный порт

Serial.begin(9600);

// ждём, пока не откроется монитор последовательного порта

// для того, чтобы отследить все события в программе

while (!Serial) {

}

Serial.print("Serial init OK\r\n");

// запускаем Ethernet-соединение:

if (Ethernet.begin(mac) == 0) {

// если не удалось сконфигурировать Ethernet при помощи DHCP

Serial.println("Failed to configure Ethernet using DHCP");

// продолжать дальше смысла нет, поэтому вместо DHCP

// попытаемся сделать это при помощи IP-адреса:

Ethernet.begin(mac, ip);

}

// запускаем сервер и выводим локальный IP адрес

server.begin();

Serial.print("Server is at ");

Serial.println(Ethernet.localIP()); // выводим в порт айпи для подключения

}

void loop()

{

EthernetClient client = server.available();

if (client) {

// выводим сообщение о новом клиенте

Serial.println("new client");

// HTTP-запрос заканчивается пустой линией

boolean ok = true;

while (client.connected()) {

if (client.available()) {

char c = client.read();

Serial.write(c);

if (c == '\n' && ok) {

client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");

client.println("Connection: close");

client.println("Refresh: 5"); // время обновления страницы

client.println();

client.println("<!DOCTYPE HTML>");

client.println("<html><meta charset='UTF-8'>");

client.println("<h1>Температура: "); // выводим надпись на веб страницу

client.println(String(sensor.tem));// выводим данные с датчика на веб стрнацу

client.println(" &#176;C</h1>");

client.println("</html>");

break;

}

if (c == '\n'){ok = true;}else if(c != '\r'){ok = false;}

}

}

// даем браузеру время, чтобы получить данные

delay(1);

// закрываем соединение

client.stop();

// клиент отключился

Serial.println("Client disconnected");

}

}