

Муниципальное бюджетное общеобразовательное учреждение
«Средняя общеобразовательная школа №9»

**Тема: «Школьная цифровая метеолaborатория на основе
микроконтроллера Arduino»**

Выполнил:

Мальцев Л.А.,
обучающийся 11 класса МБОУ «СОШ №9»

Руководитель :

Литвиненко Р.И.,
учитель физики и информатики МБОУ «СОШ №9»

Таштагол 2021



Оглавление

1.	Введение	2
1.1	Общая характеристика цифровой метеолaborатории	2
1.2	Цель	2
1.3	Описание и назначение цифровой метеолaborатории	3
2.	Устройство	3
2.1.	Комплекующие ЦМЛ.....	3
2.2.	Схемы подключения	7
2.3.	Скетчи.....	8
2.4.	Интерфейс	10
3.	Заключение	12
4.	Список информационных источников.....	12



1. Введение

1.1 Общая характеристика цифровой метеолaborатории

Жизнь каждого человека неразрывно связана с окружающей средой. Зависит от времен года и погодных условий. Смена времен года предсказуема, а погодные условия весьма изменчивы. Температуру воздуха, влажность, атмосферное давление можно узнать по радио, телевидению, Интернету. Но из средств массовой информации мы узнаем готовые результаты прогноза погоды, и для данной местности они не всегда бывают точными и достоверными. Создание метеоплощадки на территории школы поможет понять, как составляются прогнозы погоды, позволит осуществлять научно-исследовательскую и экспериментальную деятельность школьников по географии, физики, экологии.

Разработанная цифровая метеостанция на основе микроконтроллера Arduino является электронным приложением и дополнением к Школьной учебной метеостанции. Она предназначена для сбора и анализа метеорологических данных, полученных с датчиков. Состоит из датчиков температуры и давления, измерения высоты, детектора качества воздуха. Датчики, установленные на окне кабинета географии или в любом другом месте здания школы, измеряют параметры погодных условий и передают все полученные данные на базовую станцию по радиоканалу на частоте 2,4 ГГц. Размеры и вес ячейки с датчиками позволяют проводить сбор данных на любой высоте, т.е. есть возможность использовать зонд для измерения метео данных в более высоких слоях атмосферы. Показания датчиков отображаются на сенсорном дисплее. Базовая станция находится в кабинете географии. Измерение параметров состояния атмосферы работает круглосуточно. Имеется возможность для составления графиков различных показателей погоды и наличия вредных примесей в атмосфере (температура воздуха, давление, концентрации CO, NH₃, NO₂) за определенный промежуток времени, что позволяет осуществлять мониторинг.



1.2 Цель

Создать электронное приложение к Школьной учебной метеостанции - устройство, способное получать данные о внешней среде (температура, давление, наличие вредных примесей в атмосфере) и систематизировать их для более удобного последующего анализа.



1.3 Описание и назначение цифровой метеолaborатории

Корпус цифровой метеостанции представляет собой алюминиевый чемоданчик, внутри которого расположены:

- экран для вывода информации;
- ячейка с микроконтроллерами - “мозгом” системы;
- съёмная ячейка с датчиками для направленного анализа;
- отсек для 2-х антенн, подключающихся к ячейкам для увеличения дальности передачи данных;
- батареек питания, дающих ячейкам независимость от проводов и общей электросети.

2. Устройство

2.1. Комплектующие ЦМЛ

- **Микроконтроллер Arduino UNO**



Рис. 1. Arduino Uno

Является сердцем ресивера, посылает данные на экран.

- Макс. потребляемый ток: 1А
- Питание: 6 - 20 В
- Входы/выходы: 14 пинов (6 ШИМ)
- Аналоговые входы: 6 пинов
- Память: 32 Кб

- **Микроконтроллер ArduinoNano**

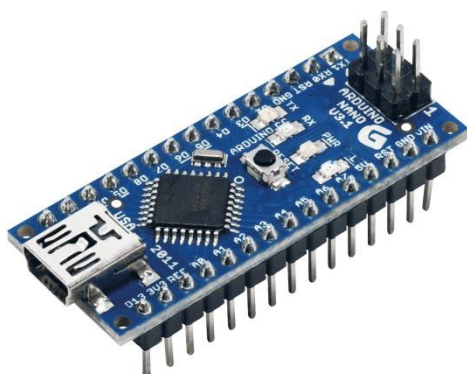


Рис. 2. ArduinoNano

Является сердцем трансммиттера, собирает данные с датчиков.

- Макс. потребляемый ток: 20 мА
- Питание: 6 - 20 В
- Входы/выходы: 14 пинов (6 ШИМ)
- Аналоговые входы: 8 пинов
- Память: 16 Кб



- **НМІ-дисплей Nextion (NX4832T035_011R)**



Рис. 3. Nextion

Выводит данные, полученные с датчиков. Подключён к ресиверу.

- Макс. потребляемый ток: 500 мА
- Питание: 5В
- Разрешение: 320 X 480 пикс
- Глубина цвета: 16 бит
- Количество цветов: 65536
- Память: 16 Мб

- **Радиомодуль NRF24L01+ с антенной (2 шт.)**



Рис. 4. NRF24L01+ с антенной

Обеспечивает передачу данных на расстоянии (с Нано – на Уно).

- Макс. потребляемый ток: 115 мА
- Питание: 3.3В
- Частота: 2.4 ГГц
- Скорость передачи: 0.25 – 2 Мбит/с
- Дальность связи: до 1000 м

- **Адаптер для NRF24L01 (2 шт.)**

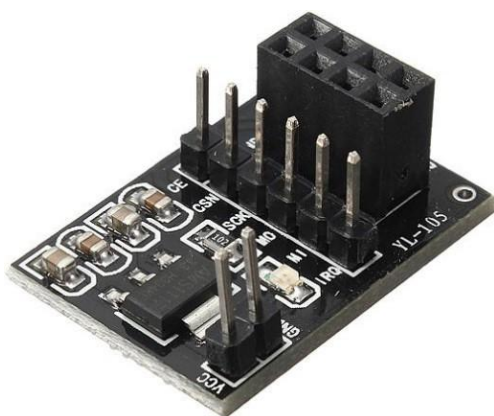


Рис. 5. Адаптер для NRF24L01

Стабилизирует напряжение для радиомодулей.

- Макс. потребляемый ток: 800 мА
- Питание: 12В
- Выходное напряжение: 3.3 В



- Датчик часов реального времени RTC-DS1302



Рис. 6. RTC DS1302

Датчик часов реального времени. Выводит текущее время. Установлен в ячейке ресивера.

- Макс. потребляемый ток: 800 мА
- Питание (от платы): 5В

- Клеммы для батарейки крона (2 шт.)



Рис. 7. Клеммы для кроны

Делает лабораторию автономной, посредством подключения батареек.

- Входное напряжение: крона, 9В

- Датчик газа CJMCU-6814

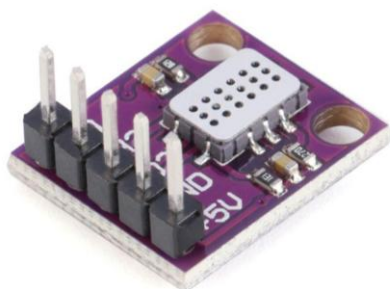


Рис. 8. CJMCU-6814

Определяет качество воздуха. Находит количество миллионных долей (мд) у Оксид азота (NO₂), Угарного газа (CO) и Аммиака (NH₃). Установлен в ячейке трансмиттера.

- Питание: 5В
- Макс. значение CO: 1000 мд
- Макс. значение NO₂: 10 мд
- Макс. значение NH₃: 300 мд



- Датчик температуры и давления BME/BMP280

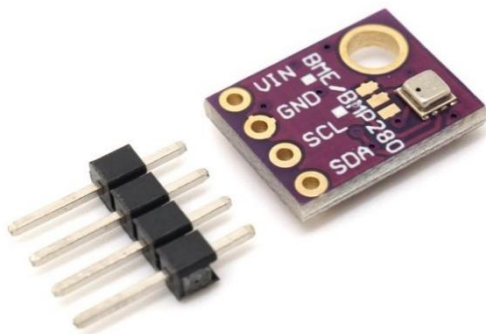


Рис. 9. BME/BMP280

Датчик атмосферного давления и температуры. Определяет температуру воздуха (°C), давление (гПа) и высоту над уровнем моря (м). Установлен в ячейке трансммиттера.

- Макс. потребляемый ток: 1 мА
- Питание: 5В
- Диап. изм. давления: 300-1100 гПа
- Диап. изм. темп-ры: -40 до +85 °C
- Точность изм. давления: ±1,0 гПа
- Точность изм. темп-ры: ±0,5 °C

- Корпус



Рис. 10. Чемодан

В форме чемоданчика. Мобильный, прекрасно подходит для переносок на расстояния.

- Материал: алюминий, картон, пластик, поролон
- Габариты: 22 X 25 X 5 см

- Провода



Рис. 11. Провода

Различные провода: мама-мама, папа-папа, папа-мама. Также готовые шлейфы.



2.2. Схемы подключения

- **Ресивер - приёмник**

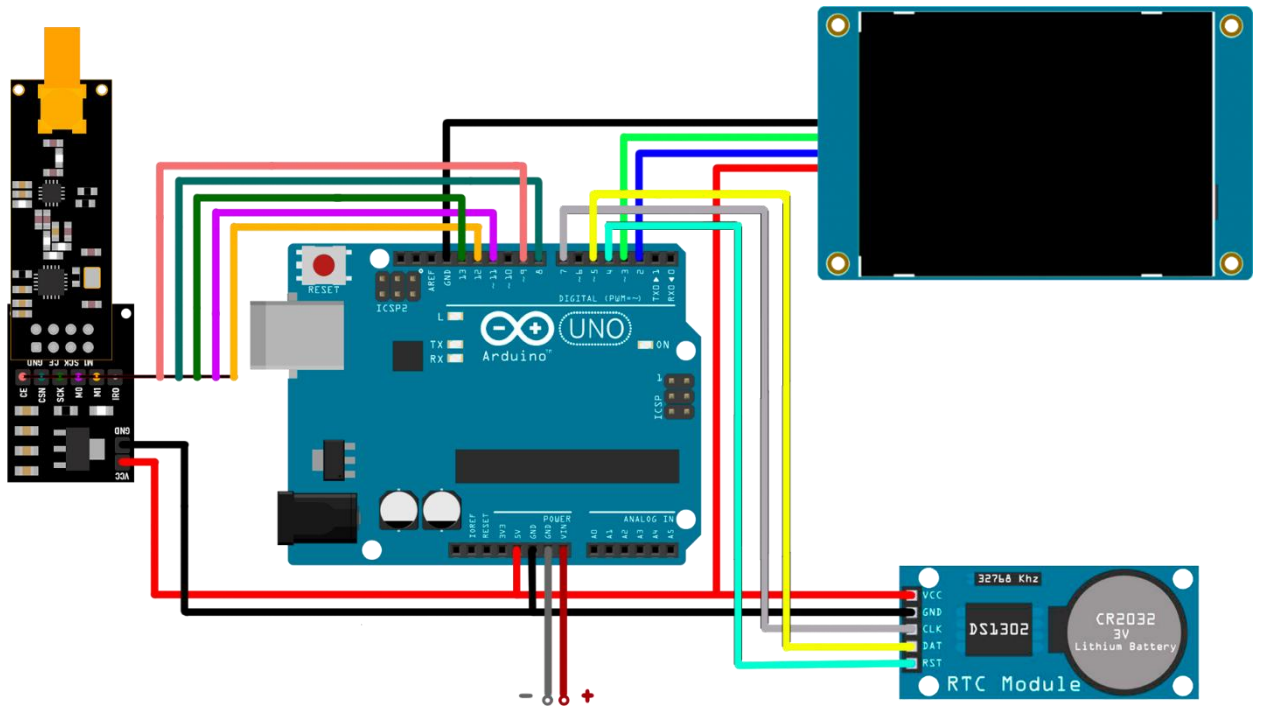


Рис. 12. Схема подключения компонентов в ресивере

- **Трансмиттер - передатчик**

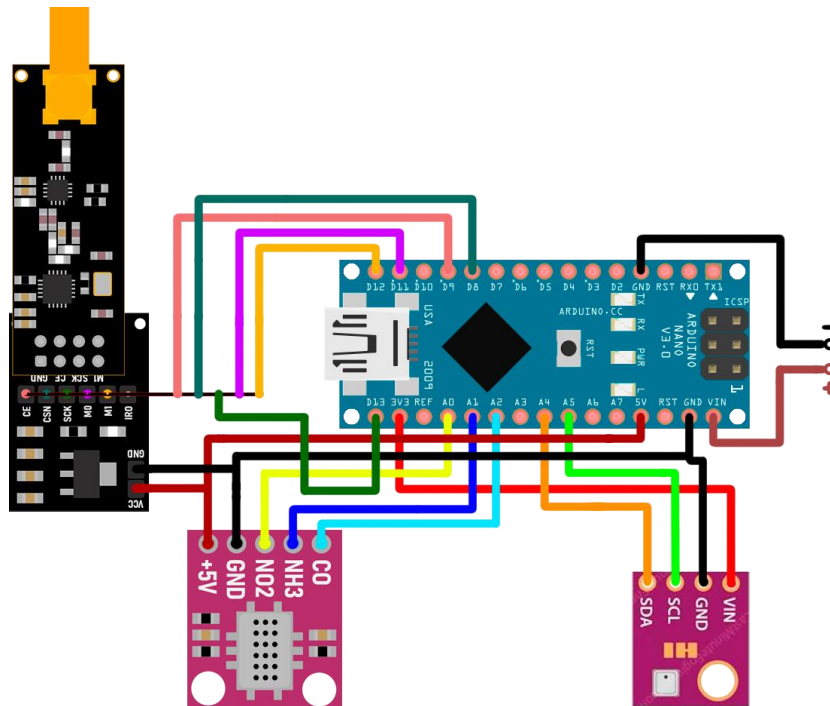


Рис. 13. Схема подключения компонентов в трансмиттере



2.3. Скетчи

- **Ресивер - приёмник**

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <SoftwareSerial.h>
#include <Nextion.h>
#include <iarduino_RTC.h>

iarduino_RTCTime(RTC_DS1302,4,7,5);

RF24 nrf(9, 8); // CE, CSN

SoftwareSerial port1(2, 3);
Nextion disp(port1, 9600);

int d;
int m;
int y;
int h;
int mi;
int s;
int j;
int n;
const byte greenLED = 10;
float data[6];

void setup()
{
  Serial.begin(9600);
  Serial.println("Starting");
  pinMode(greenLED, OUTPUT);
  nrf.begin();
  nrf.setChannel(115);
  nrf.setDataRate(RF24_1MBPS);
  nrf.setPALevel(RF24_PA_HIGH);
  nrf.openReadingPipe(0, 0x7878787878LL);
  nrf.startListening();

  time.begin();
  //time.setTime(45,18,3,3,10,20,6);
  disp.init();
}

void loop()
{
  if (disp.getComponentValue("sens") == 1)
  {
    d = disp.getComponentValue("nh0");
    m = disp.getComponentValue("nh1");
```

```
    y = disp.getComponentValue("nh2");
    h = disp.getComponentValue("nh3");
    mi = disp.getComponentValue("nh4");
    s = disp.getComponentValue("nh5");
    n = disp.getComponentValue("dn");
    time.setTime(s,mi,h,d,m,y,n);
  }

  digitalWrite(greenLED, HIGH);
  delay(50);
  digitalWrite(greenLED, LOW);
  delay(50);

  if (nrf.available())
  {
    nrf.read(&data, sizeof(data));
    Serial.print(data[0]); Serial.print("hPa ");
    Serial.print(data[1]);
    Serial.write(0xC2); //send degree symbol
    Serial.write(0xB0); //send degree symbol
    Serial.print("C ");
    Serial.print(data[2]); Serial.print("m ");
    Serial.print(data[3]); Serial.print("ppm (NO2) ");
    Serial.print(data[4]); Serial.print("ppm (NH3) ");
    Serial.print(data[5]); Serial.println("ppm (CO)");
  }
  Serial.println(time.getTime("d-m-Y, H:i:s, D"));
  disp.setComponentValue("n0", data[0]);
  disp.setComponentValue("n1", data[1]);
  disp.setComponentValue("n2", data[2]);
  disp.setComponentValue("n3", data[3]*100);
  disp.setComponentValue("n4", data[4]*100);
  disp.setComponentValue("n5", data[5]);
  disp.setComponentText("t0", String(data[0]));
  disp.setComponentText("t1", String(data[1]));
  disp.setComponentText("t2", String(data[2]));
  disp.setComponentText("t3", String(data[3]));
  disp.setComponentText("t4", String(data[4]));
  disp.setComponentText("t5", String(data[5]));
  disp.setComponentText("date0",
String(time.getTime("H:i:s")));
  disp.setComponentText("date1",
String(time.getTime("d.m.Y, D")));
  delay(100);
}
```



- Трансмиситтер - передатчик

```
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BME280.h>
#include <MiCS6814-I2C.h>

RF24 nrf(9, 8); // CE, CSN
Adafruit_BME280 bme;

float QNH = 1029.46; //www.wunderground.com
const int BME_address = 0x76;

int sensorPin0 = A0; // NO2
int sensorPin1 = A1; // NH3
int sensorPin2 = A2; // CO
float co = 0.0;
float nh3 = 0.0;
float no2 = 0.0;

float pressure;
float temperature;
float altimeter;

float data[6];
char charVal[17];

void setup()
{
  Serial.begin(9600);

  Serial.println("BME280 initialization...");
  bme.begin(BME_address);

  Serial.println("MiCS6814 initialization...");
  pinMode(sensorPin0, INPUT);
  pinMode(sensorPin1, INPUT);
  pinMode(sensorPin2, INPUT);

  Serial.println("NRF24L01 linking...");
  nrf.begin();
  nrf.setChannel(115);
  nrf.setDataRate(RF24_1MBPS);
  nrf.setPALevel(RF24_PA_HIGH);
  nrf.openWritingPipe(0x7878787878LL);
  nrf.stopListening();
}
```

```
void loop()
{
  pressure = bme.readPressure()/100; //Pa ->hPa
  temperature = bme.readTemperature();
  altimeter = bme.readAltitude (QNH); //QNH -
  давлениенавысотенадуровнем моря
  data[0] = pressure;
  data[1] = temperature;
  data[2] = altimeter;

  no2 = analogRead(sensorPin0);
  nh3 = analogRead(sensorPin1);
  co = analogRead(sensorPin2);
  data[3] = no2/100; //(no2/2)*0.5217
  data[4] = (nh3*1.4093)/100;
  data[5] = co*0.8571;

  Serial.print(data[0]); Serial.print("hPa ");
  Serial.print(data[1]);
  Serial.write(0xC2); //send degree symbol
  Serial.write(0xB0); //send degree symbol
  Serial.print("C ");
  Serial.print(data[2]); Serial.print("m ");
  Serial.print(data[3]); Serial.print("ppm (NO2) ");
  Serial.print(data[4]); Serial.print("ppm (NH3) ");
  Serial.print(data[5]); Serial.println("ppm (CO)");

  nrf.write(data, sizeof(data));

  delay(100);
}
```



2.4. Интерфейс



Рис. 14. Главная страница



—переход в меню графов



— переход в настройки



— показания датчиков



— текущее время



— текущая дата



Рис. 15. Страница с графами (1)



Рис. 16. Страница с графами (2)

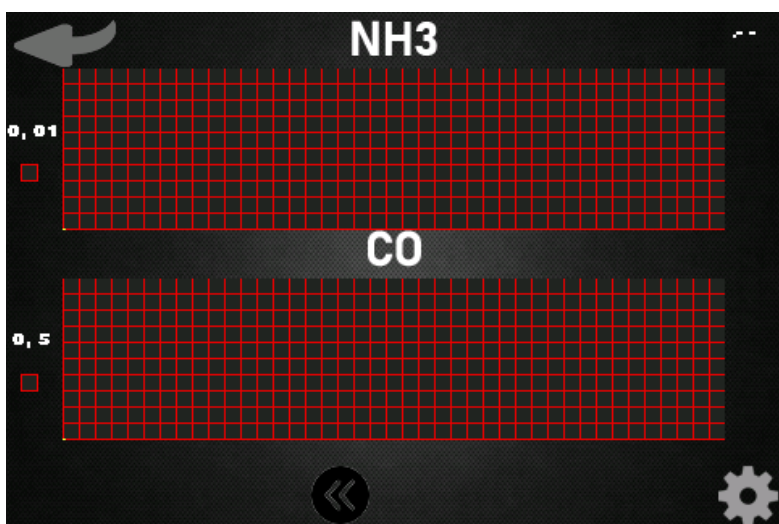
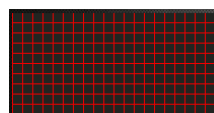


Рис. 17. Страница с графами (3)



— поле графика



— единичный отрезок графа



— предыдущая страница с графами



— следующая страница с графами



— вернуться на главную страницу



3. Заключение

Преимущества разработанной мной цифровой метеостанции в том, что она позволяет объективно оценивать получаемые данные и приближает школьные исследовательские работы к современному стандарту научной работы. Метеостанция имеет свои отличительные положительные особенности в сравнении с существующими аналогами:

- инновационное оборудование является дорогостоящим, и не все образовательные учреждения могут позволить себе такие комплексы. Я предлагаю создать проект подобной метеостанции на базе микропроцессора Arduino. Основные преимущества платформы Arduino: низкая стоимость – платы Arduino относительно дешевы по сравнению с другими платформами, их можно приобрести за 600 рублей, датчики так же имеют низкую стоимость;
- кроссплатформенность – программное обеспечение Arduino работает под ОС Windows, MacintoshOSX и Linux. Большинство микроконтроллеров ограничивается ОС Windows;
- простая и понятная среда программирования – среда ArduinoIDE подходит как для начинающих пользователей, так и для опытных;
- возможность усовершенствования: можно добавить датчики и увеличить количество измеряемых параметров

4. Список информационных источников

- <https://lesson.iarduino.ru>
- <https://wiki.iarduino.ru>
- <https://iarduino.ru/file/>
- <https://wiki.iarduino.ru/page/ustanovka-nastroyka-programmnoy-obolochki-arduino-ide-dlya-windows/>
- <https://robotclass.ru/tutorials/arduino-radio-433mhz/>