ЗАДАНИЕ: обеспечить сбор информации с датчиков через Arduino, отправка их на ESP8266(Node MCU) c целью отправки информации на сервис “dweet.io”.

# СХЕМА СБОРКИ

Схема состоит из Arduino UNO, Node MCU, Датчик температуры и влажности DHT11, потенциометр, фоторезистор и сигнальный светодиод.

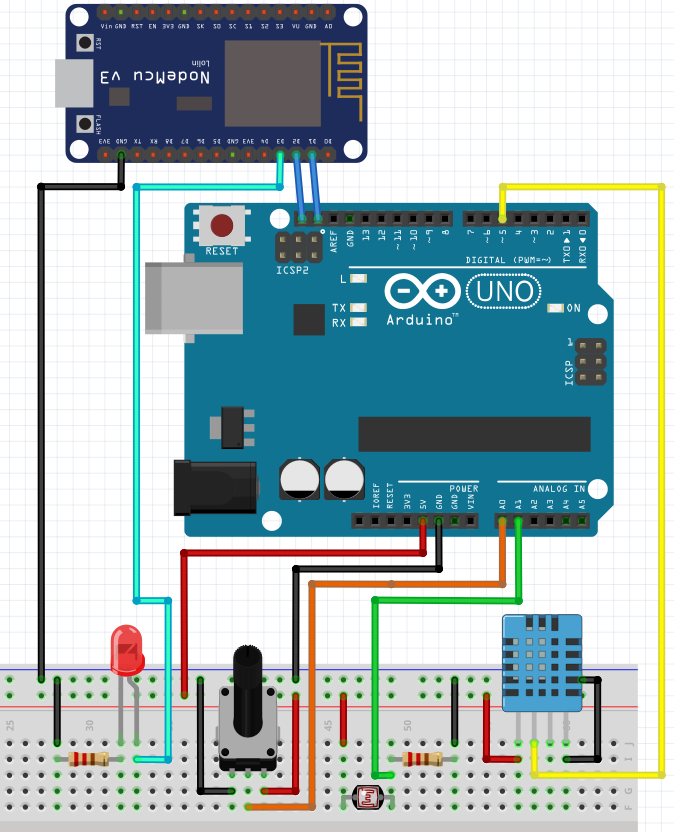


Рисунок 1.

# ПРОГРАММА ДЛЯ ARDUINO

Программа для Arduino состоит в том, что прочитать значение датчиков и отправить их на Node MCU. Отправка данных происходит интерфейсу I2C где Arduino является ведомым устройством с адресом 55.

Блок схема:



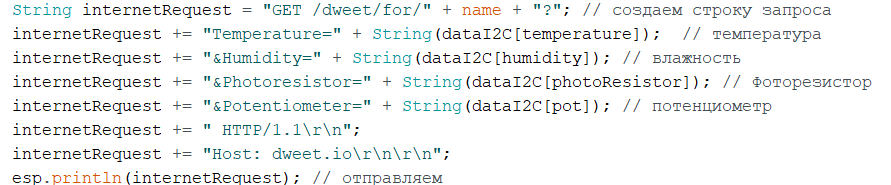
Программный код [приложение 1](#Приложение1).

# ПРОГРАММА ДЛЯ NODE MCU

Программа для Node MCU совершает весь основной алгоритм, получение информации с датчиков, создание запроса, произведение запроса на сервер, выявление и сигнализация ошибок.

Программа детектирует три вида ошибок: ошибка значения, ошибка соединения по I2C, ошибка соединения с сервером. При выявлении ошибки загорается “Сигнальная колона”, сигнальный светодиод.

Запрос на dweet.io представляет из себя строку, состоящую из параметров, значения и адрес сервера, параметры связываются между собой с помощью &.



Переменная name отвечает за индификатор на dweet.io, по которому мы будем отправлять и наблюдать наши данные.

Блок схема:



Программный код [приложение 2](#Приложение2).

# РЕЗУЛЬТАТ

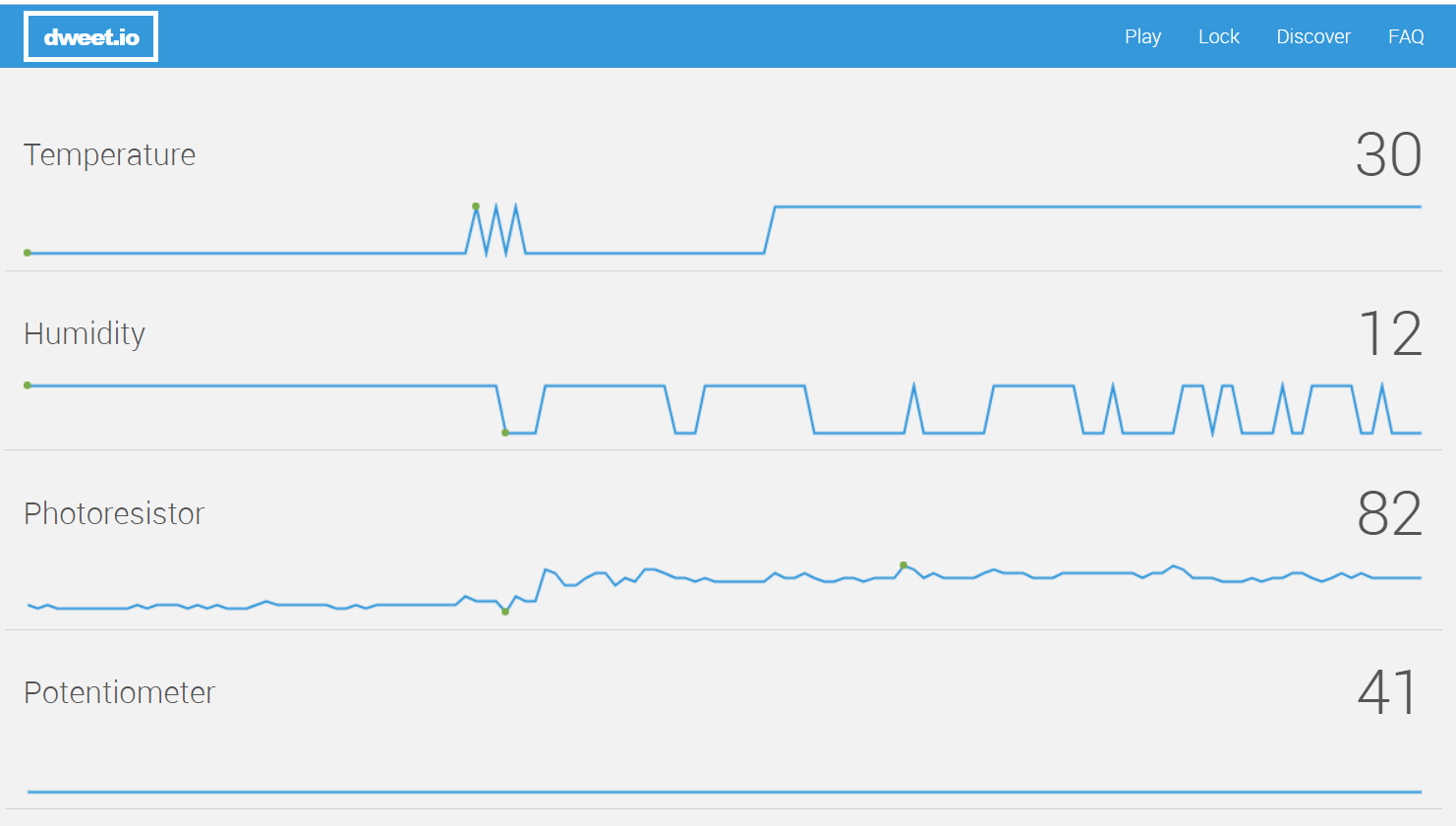


Рисунок 2 графики на сервисе “dweet.io”.

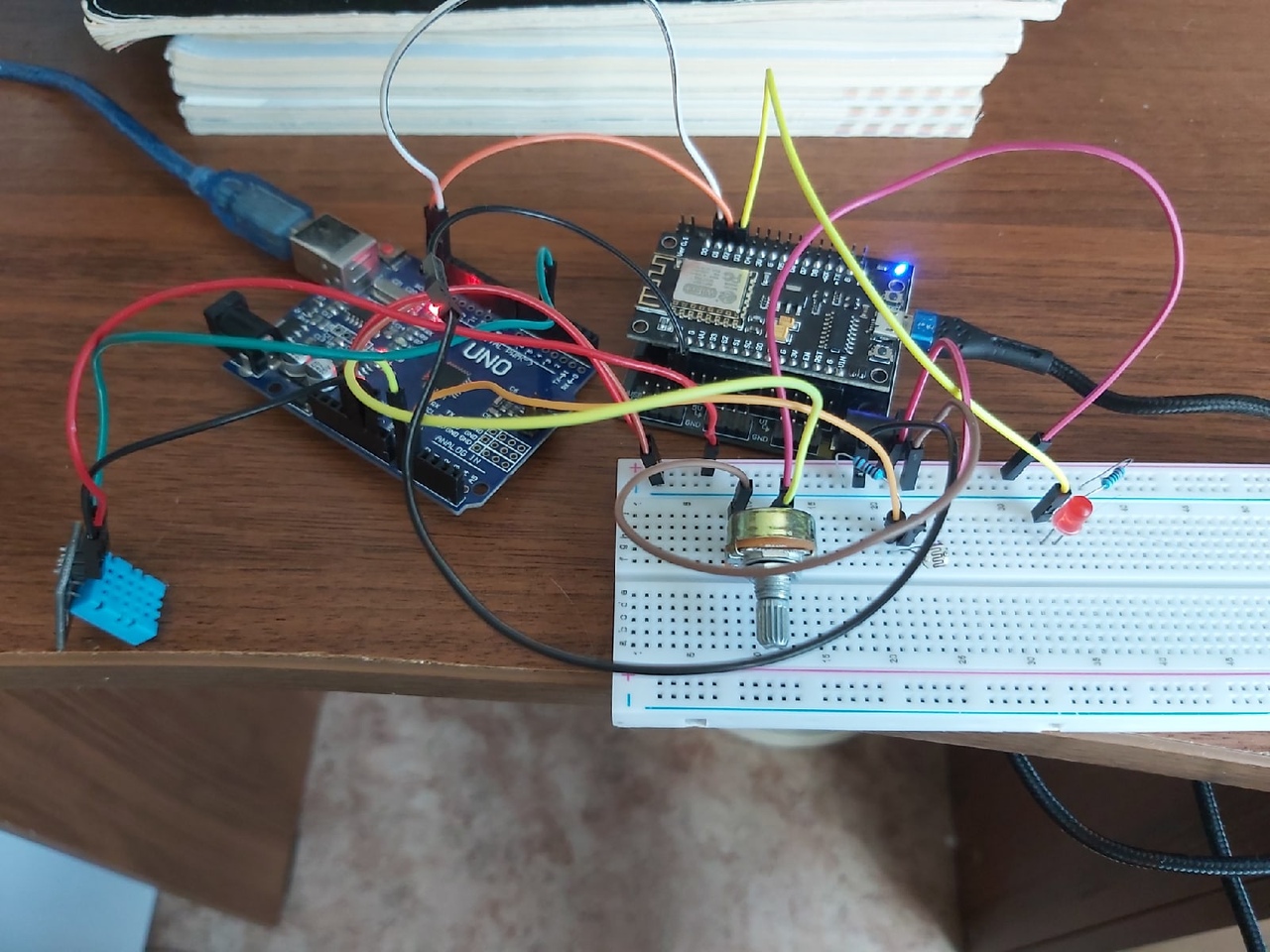


Рисунок 3 собранная схема.

Приложение 1:

#include <Wire.h> // I2C Библиотека

#include <dht11.h>

#define DHT\_PIN 5

#define POT\_PIN A0

#define PR\_PIN A1

dht11 dht;

uint8\_t dataI2C [4]; // массив данных

enum dataVal{temperature,humidity,photoResistor,pot}; // перечисление

void setup() {

Serial.begin(9600);

Wire.begin(55); // задаем I2C адресс 55

Wire.onRequest(requestEvent); // регистрируем запрошенное событие

}

void loop() {

dht.read(DHT\_PIN);

}

void requestEvent () { // функция для передачи данных по I2C

//dht.read(DHT\_PIN);

dataI2C [temperature] = dht.temperature;

dataI2C [humidity] = dht.humidity;

dataI2C [photoResistor] = analogRead(PR\_PIN)/10.24f;

dataI2C [pot] = analogRead(POT\_PIN)/10.24f;

for(uint8\_t i = 0; i < sizeof(dataI2C);i++)

{

Wire.write(dataI2C[i]);

}

}

Приложение 2:

#include <ESP8266WiFi.h>

#include <Wire.h> // I2C Библиотека

// Для подключения к WIFI

#define \_SSID "TP-Link\_6A88"

#define \_PASSWORD "07741203"

#define ADDR 55 // Адресс Arduino

#define SCL D2 // Пины

#define SDA D1

#define ERROR\_SIGN\_PIN D3

int8\_t dataI2C [4] = {-1,-1,-1,-1}; // получаемые данные

struct // структура для ошибок

{

bool val = false;

bool i2c = false;

bool dweet = false;

}errorFlag;

enum dataVal{temperature,humidity,photoResistor,pot}; //перечисление

WiFiClient esp; // вайфай на ESP

String name = "Test\_0"; // индификатор для dweet

void setup(void) {

Serial.begin(9600);

pinMode(ERROR\_SIGN\_PIN,OUTPUT);

WiFi.begin(\_SSID,\_PASSWORD); // подключение ESP к WiFi

Serial.println("https://dweet.io/follow/" + name); //где смотреть данные

Wire.begin(SDA,SCL); // Инициализируем I2C

while (WiFi.status() != WL\_CONNECTED) // ожидания подключения к WiFi

{

delay(500);

Serial.println(F("Connecting to WiFi.."));

}

}

void loop(void) {

Wire.requestFrom(ADDR,sizeof(dataI2C)); // запрос по I2C по адрессу ADDR, количество битов sizeof(dataI2C)

if(Wire.available()) // если есть соединение по I2C то читаем даные

{

for(uint8\_t i = 0; i < sizeof(dataI2C);i++)

{

dataI2C[i]= Wire.read();

}

errorFlag.i2c = false;

}

else

{

Serial.println(F("I2C No signal"));

errorFlag.i2c = true;

}

if(!errorFlag.i2c)// если нет ошибки с принятием данных по I2C

{

bool\* flagErrorVal = new bool;// вспомагательная переменная для вылова ошибок

for(uint8\_t i = 0; i < sizeof(dataI2C);i++) // проверка на значение на ошибки

{

if (dataI2C[i] < 0) // если значение меньше нуля то поднимаем флаг ошибки

{

Serial.print(F("Error value "));

Serial.print(dataI2C[i]);

Serial.print(F(" number of parameter "));

Serial.println(i);

\*flagErrorVal = true;

errorFlag.val = true;

}

if(!\*flagErrorVal)

{

errorFlag.val = false;

}

if (i == (sizeof(dataI2C)-1))\*flagErrorVal = false;

}

delete flagErrorVal; // уничтожаем вспомогательную переменую

}

else // в ином случае идем в конец

{

goto \_end;

}

if (errorFlag.val) goto \_end; // если ошибка значения то идем в конец

//инфомация с датчиков в порт

Serial.print(dataI2C[temperature]);

Serial.print('\t');

Serial.print(dataI2C[humidity]);

Serial.print('\t');

Serial.print(dataI2C[photoResistor]);

Serial.print('\t');

Serial.println(dataI2C[pot]);

if (esp.connect("www.dweet.io", 80)) // делаем запрос на www.dweet.io , если удалось поключится к серверу

{

String internetRequest = "GET /dweet/for/" + name + "?"; // создаем строку запроса

internetRequest += "Temperature=" + String(dataI2C[temperature]); // температура

internetRequest += "&Humidity=" + String(dataI2C[humidity]); // влажность

internetRequest += "&Photoresistor=" + String(dataI2C[photoResistor]); // Фоторезистор

internetRequest += "&Potentiometer=" + String(dataI2C[pot]); // потенциометр

internetRequest += " HTTP/1.1\r\n";

internetRequest += "Host: dweet.io\r\n\r\n";

esp.println(internetRequest); // отправляем

Serial.print(internetRequest);// дублируем в последовательный порт

errorFlag.dweet = false;

}

else //если неудалось поключится к серверу

{

Serial.println(F("dweet io no conection"));

errorFlag.dweet = true;

}

\_end:

digitalWrite(ERROR\_SIGN\_PIN, errorFlag.val || errorFlag.i2c || errorFlag.dweet); // зажигаем сигнальную колону при ошибке

delay(5000);

}