Задача: организовать управление процессом по средствам Telegram бота. Получение данных с датчиков, включение и отключение приборов.

# СХЕМА

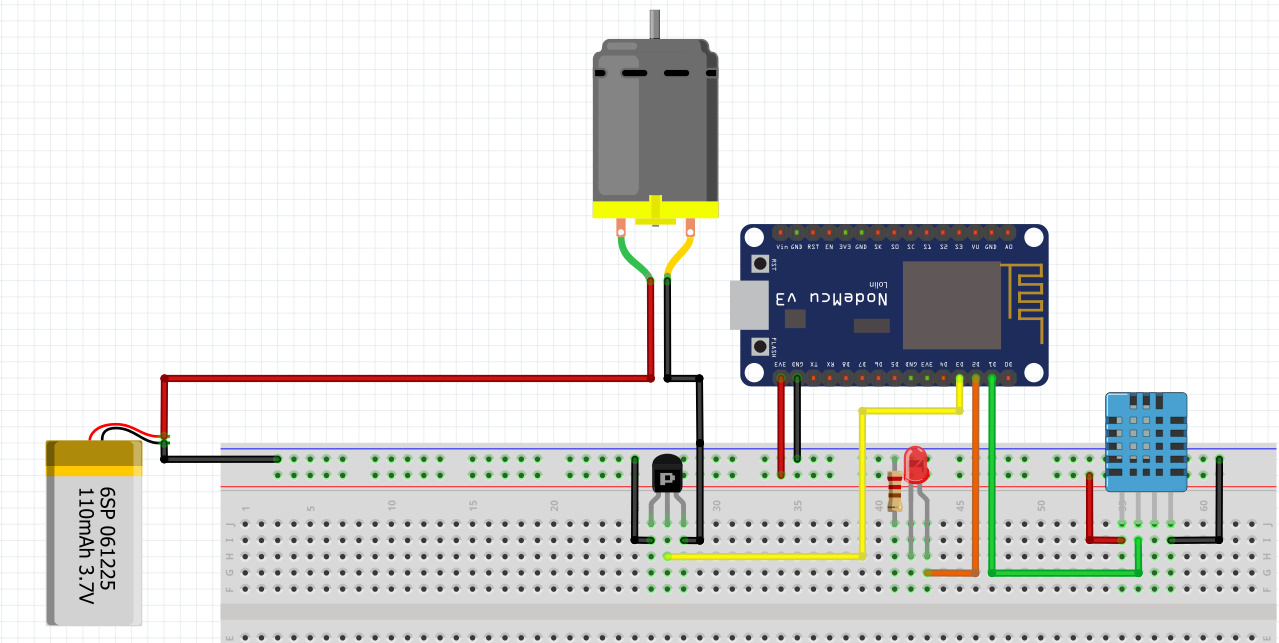


Рисунок 1.

На схеме изображена монтажная схема, состоящая из датчика DHT11, светодиода (LED), вентилятор (FAN) показанный в виде двигателя постоянного тока, источник постоянного тока в виде батарейки и n-p-n транзистор КТ815Г соединения по схеме с общим коллектором. Транзистор необходим для включения вентилятора.

# ПРОГРАММА

Создания бота:

1. Открой мессенджер. В поиске найди главного бота — «@botfather». Заведи с ним диалог и нажми кнопку «START»
2. Выбери пункт «/newbot».
3. Придумай имя своему боту.
4. Теперь придумай уникальный логин бота. Он обязательно должен заканчиваться на «bot».
5. Сделано! Теперь скопируй специальный токен, который тебе выдал @BotFather. Он потребуется для авторизации. Токен — это уникальный ключ для доступа к сервису. Его должен знать только ты — не сообщай его никому!
6. Кликни на ссылку, которую тебе дал @ BotFather, чтобы начать разговор с ботом.
7. Добавь своего бота в контакты и нажми кнопку «START». Ты увидишь клавиатуру с надписями «/ledon» и «/ledoff». Открой Serial Monitor и понажимай на эти кнопки. В Serial Monitor будут выводиться соответствующие сообщения.

Также для работы программы нужно скачать две библиотеки:

1. ArduinoJson ссылка: https://github.com/bblanchon/ArduinoJson
2. UniversalTelegramBot ссылка: <https://github.com/witnessmenow/Universal-Arduino-Telegram-Bot>

Программный код:

#include <ESP8266WiFi.h>

#include <WiFiClientSecure.h>

#include <UniversalTelegramBot.h>

#include <dht11.h>

// Для подключения к WIFI

#define \_SSID "TP-Link\_6A88"

#define \_PASSWORD "07741203"

#define BOT\_TOKEN "1138203036:AAFEosSBUTexaTVQSkDZAvNvrOaTp8G-tD0"

#define FAN\_PIN D3

#define LED\_PIN D2

#define DHT\_PIN D1

WiFiClientSecure esp; // вайфай на ESP

UniversalTelegramBot bot(BOT\_TOKEN,esp);

int8\_t numNewMessages; //переменная для хронения количесвто новых сообщений

struct // структура для разрешений

{

bool fan;

bool led;

} permition;

dht11 dht;

void setup() {

Serial.begin(9600);

pinMode(FAN\_PIN,OUTPUT);

pinMode(LED\_PIN,OUTPUT);

esp.setInsecure();

WiFi.mode(WIFI\_STA);// устанавливаем вайфай в режим станции

WiFi.disconnect();

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

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

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

{

delay(500);

Serial.print('.');

}

Serial.println("");

Serial.println("Connected");

}

void loop() {

dht.read(DHT\_PIN); // чтение DHT

numNewMessages = bot.getUpdates(bot.last\_message\_received + 1); // читаем количество сообщений

for(int i = 0; i < numNewMessages; i++) // Обработка сигнала

{

String chat = String(bot.messages[i].chat\_id);

String txt = bot.messages[i].text;

if (txt == "/start" || txt == "Main Menu")

{

String keyboardJson = "[[\"/LED\"],"; // 1 строка кнопки

keyboardJson += "[\"/FAN\"],"; // 2 строка кнопки

keyboardJson += "[\"/DHT\"]]"; // 3 строка кнопки

bot.sendMessageWithReplyKeyboard(chat, " Main Menu", "", keyboardJson, true);

Serial.println("Start");

// обнуление разрешений

permition.fan = false;

permition.led = false;

}

if(txt == "/FAN")

{

String keyboardJson = "[[\"/ON\",\"/OFF\"],";

keyboardJson +="[\"Main Menu\"]]"; // возврат в главное меню

bot.sendMessage(chat,"FAN = " + String(digitalRead(FAN\_PIN)? "ON" : "OFF"),"");

bot.sendMessageWithReplyKeyboard(chat, "FAN Menu", "", keyboardJson, true);

permition.fan = true;

}

else if (txt == "/LED")

{

String keyboardJson = "[[\"/ON\",\"/OFF\"],";

keyboardJson +="[\"Main Menu\"]]"; // возврат в главное меню

bot.sendMessage(chat,"LED = " + String(digitalRead(LED\_PIN)? "ON" : "OFF"),"");

bot.sendMessageWithReplyKeyboard(chat, "LED Menu", "", keyboardJson, true);

permition.led = true;

}

else if (txt == "/DHT")

{

bot.sendMessage(chat,"Temperature = " + String(dht.temperature) + "\n" + "Humidity = " + String(dht.humidity) + "\n","");

}

else if (txt == "/ON")

{

if(permition.led) digitalWrite(LED\_PIN,true);

if(permition.fan) digitalWrite(FAN\_PIN,true);

}

else if (txt == "/OFF")

{

if(permition.led) digitalWrite(LED\_PIN,false);

if(permition.fan) digitalWrite(FAN\_PIN,false);

}

}// конец обработки программы

}// конец loop