#include <Ethernet.h>

#include <IRremote.h>

IRsend irsend;

boolean newInfo = 0; //переменная для новой информации

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

//вводим mac адрес, обычно он такого вида, может отличаться

IPAddress ip(192,168,1,175);

//вводим любой IP , который не совпадает с вашим

EthernetServer server(80); //инифиализация библиотеки Ethernet server library

/\*---------------- 2 часть ----------------\*/

void setup() {

pinMode (3, OUTPUT);//инициализируем 8 пин как выход для светодиода

Ethernet.begin (mac, ip);//запускаем сервер с указанными ранее MAC и вашим IP

pinMode (9, INPUT);

Serial.begin (9600);

server. begin();

}

/\*---------------- 3 часть ----------------\*/

void loop() {

EthernetClient client = server.available(); //принимаем данные, посылаемые клиентом

if (client)

{ //если запрос оканчивается пустой строкой

boolean currentLineIsBlank = true;

//ставим метку об окончании запроса (дословно: текущая линия чиста)

while (client.connected())

{ //пока есть соединение с клиентом

if (client.available())

{ //если клиент активен

char c = client.read(); //считываем посылаемую информацию в переменную "с"

if (newInfo && c == ' ')

{

//если переменная новой информации = 1 и "с", в которой записан запрос, равен пустой строке

newInfo = 0; //то обнуляем переменную поступления новой информации

}

if (c == '$')

{

// если переменная "с", несущая отправленный нам запрос, содержит символ $

// "$" подразумевает разделение получаемой информации (символов)

newInfo = 1; //то пришла новая информация, ставим метку новой информации в 1

}

//Проверяем содержание URL - присутствует $1 или $2

if (newInfo == 1)

{ //если есть новая информация

Serial.println (c);

if (c == '1')

{ //и "с" содержит 1

Serial.println ("Включить");

digitalWrite (3, HIGH); //то зажигаем светодиод, при зажиганий ИК светодиода отправляется ниже приведенный ИК сигнал

int khz = 38; // 38kHz carrier frequency for the NEC protocol

unsigned int irSignal[] = {8900, 4350, 650, 1550, 650, 1550, 650, 450, 650, 450, 700, 450, 650, 450, 650, 450, 700, 1500, 700, 1500, 700, 450, 650, 1550, 650, 450, 650, 1550, 650, 450, 700, 1500, 700, 450, 650, 450, 650, 450, 700, 450, 650, 450, 650, 1550, 650, 450, 650, 450, 700, 1550, 650, 1550, 650, 1550, 650, 1550, 650, 1550, 650, 450, 650, 1550, 650, 1550, 650, 450, 700}; //AnalysIR Batch Export (IRremote) - RAW

irsend.sendRaw(irSignal, sizeof(irSignal) / sizeof(irSignal[0]), khz); //Note the approach used to automatically calculate the size of the array.

delay(10); //In this example, the signal will be repeated every 5 seconds, approximately.

}

if (c == '2')

{ //если "с" содержит 2

Serial.println ("Выключить");

digitalWrite (3, LOW); //гасим светодиод

}

}

if (c == '\n') { //если "с" равен символу новой строки

currentLineIsBlank = true; //то начинаем новую строку

}

else if (c != '\r')

{

//иначе, если "с" не равен символу возврата курсора на начало строки

currentLineIsBlank = false; //то получаем символ на текущей строке

}

if (c == '\n' && currentLineIsBlank) { //выводим HTML страницу

/\*---------------- 4 часть ----------------\*/

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

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

client. println ("Connection: close");

//автоматическое обновление каждые 5 сек

client. println (); //Это не ошибка, так должно быть

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

client. println ("<html>"); //открытие тега HTML

client. println ("<head> "); //открытие тега Head

client. println ("<meta http-equiv='Content-Type' content='text/html ; charset=utf-8'/> ");

client. print ("<title>N\_I\_I Web Server</title>"); //название страницы

client. println ("</head>"); //заголовочная информация

client. println ("<body>");

client. print ("<H1>N\_I\_I Web Server</H1>"); //заголовк на странице

client. print ("<a href=\"/$1\"><button style=\"height:100px;width:100px;\">Включить</button></a>");

//кнопка включить

client. print ("<a href=\"/$2\"><button style=\"height:100px;width:100px;\">Выключить</button></a>");

//кнопка выключить

client. println ("<br> <br>"); //перенос на след. строчку

client. println ("Состояние = ");

client. println (digitalRead(9));

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

client. println ("</html>"); //закрываем тег HTML

break; //выход

}

}

}

delay (1); //время на получение новых данных

client. stop(); //закрываем соеднение

} }