ПРИЛОЖЕНИЕ А

(обязательное)

Исходный текст аппаратной части

Файл arduino.ino:

#include <SoftwareSerial.h>

const int rx = 10;

const int tx = 11;

const int speed = 9600;

const String HELP = "HELP";

const String MOVEMENT = "MOVEMENT";

const String SENSORS = "SENSORS";

const String WIFI\_SESSION = "WIFI\_SESSION";

int RIGHT\_UP = 7;

int RIGHT\_DOWN = 6;

int LEFT\_UP = 5;

int LEFT\_DOWN = 4;

bool wifiSession = false;

SoftwareSerial\* wifiModule = new SoftwareSerial(rx, tx);

void setup() {

Serial.begin(speed);

wifiModule->begin(speed);

wifiModule->listen();

pinMode (LEFT\_DOWN, OUTPUT);

pinMode (LEFT\_UP, OUTPUT);

pinMode (RIGHT\_DOWN, OUTPUT);

pinMode (RIGHT\_UP, OUTPUT);

}

void loop() {

String cmd = Serial.readString();

cmd = cmd.substring(0, cmd.length() - 2);

if (cmd == WIFI\_SESSION) {

wifiSession = true;

}

if (cmd.length() > 0) {

sendToWifiModule(cmd);

Serial.println("From serial: " + cmd);

}

if (wifiModule->available()) {

String cmd = wifiModule->readString();

Serial.println("From wifiModule: " + cmd);

if (wifiSession) {

Serial.println("wifi session enabled \n");

startWifiSession();

}

}

delay(100);

}

void startWifiSession() {

while(wifiSession) {

if (wifiModule->available()) {

String cmd = wifiModule->readString();

Serial.println("From wifiModule: " + cmd);

handleCmd(cmd);

}

}

}

void sendToWifiModule(String data) {

wifiModule->print("\*" + data);

}

void handleCmd(String cmd) {

if (cmd == "HELP") {

sendToWifiModule("- " + MOVEMENT + "\n- " + SENSORS);

}

else if (cmd == MOVEMENT) {

movementMenu();

sendToWifiModule(MOVEMENT + " exit");

}

else if (cmd == SENSORS) {

}

else if (cmd == "stop") {

wifiSession = false;

sendToWifiModule("WiFi session stopped");

}

else {

sendToWifiModule("Unknown command: " + cmd);

}

}

void movementMenu() {

sendToWifiModule("1.Forward\n2.Back\n3.Left\n4.Right\n5.Stop\n0.Exit");

while (1) {

if (wifiModule->available()) {

String cmd = wifiModule->readString();

if (cmd == "1") {

moveForward();

}

else if (cmd == "2") {

moveBack();

}

else if (cmd == "3") {

moveLeft();

}

else if (cmd == "4") {

moveRight();

}

else if (cmd == "5") {

moveStop();

}

else if (cmd == "0") {

return;

}

sendToWifiModule(cmd);

}

}

}

void moveForward() {

digitalWrite (RIGHT\_DOWN, LOW);

digitalWrite (RIGHT\_UP, HIGH);

digitalWrite (LEFT\_UP, HIGH);

digitalWrite (LEFT\_DOWN, LOW);

}

void moveBack() {

digitalWrite (RIGHT\_DOWN, HIGH);

digitalWrite (RIGHT\_UP, LOW);

digitalWrite (LEFT\_UP, LOW);

digitalWrite (LEFT\_DOWN, HIGH);

}

void moveLeft() {

digitalWrite (RIGHT\_DOWN, LOW);

digitalWrite (RIGHT\_UP, HIGH);

digitalWrite (LEFT\_UP, LOW);

digitalWrite (LEFT\_DOWN, LOW);

}

void moveRight() {

digitalWrite (RIGHT\_DOWN, LOW);

digitalWrite (RIGHT\_UP, LOW);

digitalWrite (LEFT\_UP, HIGH);

digitalWrite (LEFT\_DOWN, LOW);

}

void moveStop() {

digitalWrite (RIGHT\_DOWN, LOW);

digitalWrite (RIGHT\_UP, LOW);

digitalWrite (LEFT\_UP, LOW);

digitalWrite (LEFT\_DOWN, LOW);

}

Файл wifi.ino:

#include <ESP8266WiFi.h>

#include <SoftwareSerial.h>

String ssid = "";

String password = "";

const String GET\_IP\_ADDRESS = "GET\_IP\_ADDRESS";

const String SET\_IP\_ADDRESS = "SET\_IP\_ADDRESS";

const String SET\_SSID = "SET\_SSID";

const String GET\_SSID = "GET\_SSID";

const String GET\_LOCAL\_SSID = "GET\_LOCAL\_SSID";

const String SET\_PASSWORD = "SET\_PASSWORD";

const String GET\_PASSWORD = "GET\_PASSWORD";

const String CONNECT\_TO\_WIFI = "CONNECT\_TO\_WIFI";

const String WIFI\_SESSION = "WIFI\_SESSION";

const String HELP = "HELP";

const int COMMANDS\_LENGTH = 8;

String HELP\_COMMANDS[]={ GET\_IP\_ADDRESS, SET\_SSID, GET\_SSID, GET\_LOCAL\_SSID, SET\_PASSWORD, GET\_PASSWORD, CONNECT\_TO\_WIFI, WIFI\_SESSION };

const int STATE\_OK = 4;

IPAddress ip(192, 168, 1, 229);

IPAddress gateway(192, 168, 1, 1);

IPAddress subnet(255, 255, 0, 0);

SoftwareSerial\* controller = new SoftwareSerial(D9, D10);

WiFiServer wifiServer(80);

void setup() {

controller->begin(9600);

controller->listen();

wifiServer.begin();

}

void loop() {

String cmd = readFromController();

handleCommand(cmd);

delay(100);

}

void handleCommand(String cmd) {

if (cmd == "test") {

sendToController("test");

}

else if (cmd == SET\_IP\_ADDRESS) {

approve(cmd);

String newIp = readFromController();

int state = setIpAddress(newIp);

notifyController(state, cmd, getIpAddress());

}

else if (cmd == GET\_IP\_ADDRESS) {

String ipAddress = getIpAddress();

sendToController(ipAddress);

}

else if (cmd == SET\_SSID) {

approve(cmd);

delay(100);

String newSSID = readFromController();

int state = setSSID(newSSID);

notifyController(state, cmd, getSSID());

}

else if (cmd == GET\_SSID) {

String ssid = getSSID();

sendToController(ssid);

}

else if (cmd == GET\_LOCAL\_SSID) {

String ssid = getLocalSSID();

sendToController(ssid);

}

else if (cmd == SET\_PASSWORD) {

approve(cmd);

String newPassword = readFromController();

int state = setPassword(newPassword);

notifyController(state, cmd, getPassword());

}

else if (cmd == GET\_PASSWORD) {

String password = getPassword();

sendToController(password);

}

else if (cmd == CONNECT\_TO\_WIFI) {

approve(cmd);

int state = connectToHotspot();

sendToController(getIpAddress());

notifyController(state, cmd, getIpAddress());

}

else if (cmd == WIFI\_SESSION) {

approve(cmd);

notifyController(STATE\_OK, "WIFI\_SESSION", getIpAddress());

delay(200);

wifiSession();

}

else if (cmd == HELP) {

help();

}

else {

controller->print("Unknown command: " + cmd);

}

}

void wifiSession() {

while(1) {

WiFiClient client = wifiServer.available();

String wifiCmd = "";

if (client) {

while (client.connected()) {

while (client.available() > 0) {

char c = client.read();

if (c == '`') {

sendToController(wifiCmd);

String data = readFromController();

client.write(data.c\_str());

if (wifiCmd == "stop") {

client.stop();

return;

}

wifiCmd = "";

}

else {

wifiCmd += c;

}

}

delay(10);

}

}

client.stop();

}

}

void notifyController(int state, String cmd, String payload) {

state == STATE\_OK

? sendToController(cmd + " " + payload)

: reportError(cmd);

}

void reportError(String cmd) {

sendToController("Error when " + cmd);

}

void approve(String cmd) {

sendToController(cmd + "\_OK\n");

}

void sendToController(String data) {

controller->print(data.length() > 0 ? data : "NULL");

}

String readFromController() {

String data = controller->readString();

while (data[0] != '\*') {

data = controller->readString();

}

return data.substring(1, data.length());

}

int setSSID(String newSSID) {

ssid = newSSID;

return STATE\_OK;

}

String getSSID() {

return ssid;

}

String getLocalSSID() {

return WiFi.SSID();

}

int setPassword(String newPassword) {

password = newPassword;

return STATE\_OK;

}

String getPassword() {

return password;

}

int setIpAddress(String newIp) {

WiFi.config(ip, gateway, subnet);

return STATE\_OK;

}

String getIpAddress() {

return WiFi.localIP().toString();

}

int connectToHotspot() {

WiFi.begin(ssid.c\_str(), password.c\_str());

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

sendToController("Connection to " + ssid);

controller->flush();

}

sendToController("Connected to: " + ssid);

return STATE\_OK;

}

void help() {

for (int i = 0; i < COMMANDS\_LENGTH; i++) {

sendToController("\n" + HELP\_COMMANDS[i]);

}

}