ДОДАТОК А

Текст програми

```
#include <Arduino.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include "soc/soc.h"
#include "soc/rtc_cntl_reg.h"
#include "esp_camera.h"
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>
#include <ESP32Servo.h>
#include <SPI.h>
#include <MFRC522.h>

const char* ssid = "Alehio";
const char* password = "prikolist";
String BOTtoken = "7684928596:AAFvxZLyReVJJAhKCZKwarBwX5–FA2RbIVc";
String CHAT_ID = "–4911883012";

Servo myservo;
#define SERVO_PIN 12
#define FLASH_LED_PIN 4
#define BUZZER_PIN 1
#define BUTTON_PIN 3
#define MOTION_PIN 4
```

```
#define SS_PIN 2
#define RST_PIN 15
MFRC522 mfrc522(SS_PIN, RST_PIN);

#define UID_SIZE 4
#define MAX_CARDS 20
byte cardUIDs[MAX_CARDS][UID_SIZE];
String cardNames[MAX_CARDS];
bool awaitingCard = false;
bool awaitingRemove = false;
String pendingCardName = "";

bool sendPhoto = false;
WiFiClientSecure clientTCP;
UniversalTelegramBot bot(BOTtoken, clientTCP);
bool flashState = LOW;

int botRequestDelay = 1000;
unsigned long lastTimeBotRan;

#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    –1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27
#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
```

```
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22

void configInitCamera() {
camera_config_t config;
config.ledc_channel = LEDC_CHANNEL_0;
config.ledc_timer = LEDC_TIMER_0;
config.pin_d0 = Y2_GPIO_NUM;
config.pin_d1 = Y3_GPIO_NUM;
config.pin_d2 = Y4_GPIO_NUM;
config.pin_d3 = Y5_GPIO_NUM;
config.pin_d4 = Y6_GPIO_NUM;
config.pin_d5 = Y7_GPIO_NUM;
config.pin_d6 = Y8_GPIO_NUM;
config.pin_d7 = Y9_GPIO_NUM;
config.pin_xclk = XCLK_GPIO_NUM;
config.pin_pclk = PCLK_GPIO_NUM;
config.pin_vsync = VSYNC_GPIO_NUM;
config.pin_href = HREF_GPIO_NUM;
config.pin_sccb_sda = SIOD_GPIO_NUM;
```

```
config.pin_sccb_scl = SIOC_GPIO_NUM;
config.pin_pwdn = PWDN_GPIO_NUM;
config.pin_reset = RESET_GPIO_NUM;
config.xclk_freq_hz = 20000000;
config.pixel_format = PIXFORMAT_JPEG;
config.grab_mode = CAMERA_GRAB_LATEST;
if(psramFound()){
config.frame_size = FRAMESIZE_UXGA;
config.jpeg_quality = 10;
config.fb_count = 1;
} else {
config.frame_size = FRAMESIZE_SVGA;
config.jpeg_quality = 12;
config.fb_count = 1;
}
esp_camera_init(&config);
}

void unlockDoor() {
myservo.write(90);
}

void lockDoor() {
myservo.write(0);
}

void handleNewMessages(int numNewMessages) {
```

```
for (int i = 0; i < numNewMessages; i++) {
String chat_id = String(bot.messages[i].chat_id);
if (chat_id != CHAT_ID) continue;
String text = bot.messages[i].text;
String from_name = bot.messages[i].from_name;
if (text == "/start") {
String welcome = "Радий вас бачити, " + from_name + "\n";
welcome += "Доступні команди: \n";
welcome += "/photo : робить нове фото\n";
welcome += "/flash : вмикає фонарик \n";
welcome += "/lock : зачиняє двері \n";
welcome += "/unlock : відчіняє двері \n";
welcome += "/list : список доданих карт \n";
welcome += "/add : додати нову картку \n";
welcome += "/remove : видалити картку \n";
bot.sendMessage(CHAT_ID, welcome, "");
}
if (text == "/flash") {
flashState = !flashState;
digitalWrite(FLASH_LED_PIN, flashState);
}
if (text == "/photo") {
sendPhoto = true;
}
if (text == "/lock") {
lockDoor();
}
```

```
if (text == "/unlock") {
unlockDoor();
}
if (text == "/list") {
String list = "Список доступних карт::\n";
for (int i = 0; i < MAX_CARDS; i++) {
if (cardNames[i] != "") {
list += cardNames[i] + "\n";
}
}
bot.sendMessage(CHAT_ID, list, "");
}
if (text == "/add") {
bot.sendMessage(CHAT_ID, "Введіть ім'я карти і після цього піднесіть її", "");
awaitingCard = true;
} else if (awaitingCard && pendingCardName == "") {
pendingCardName = text;
}
if (text == "/remove") {
bot.sendMessage(CHAT_ID, "Введіть ім’я картки для видалення:", "");
awaitingRemove = true;
} else if (awaitingRemove) {
for (int i = 0; i < MAX_CARDS; i++) {
if (cardNames[i] == text) {
cardNames[i] = "";
for (int j = 0; j < UID_SIZE; j++) cardUIDs[i][j] = 0xFF;
```

```
bot.sendMessage(CHAT_ID, "Картку видалено", "");
break;
}
}
awaitingRemove = false;
}
}
}

String sendPhotoTelegram() {
const char* myDomain = "api.telegram.org";
String getBody = "";
camera_fb_t * fb = esp_camera_fb_get();
esp_camera_fb_return(fb);
fb = esp_camera_fb_get();
if(!fb) return "Помилка";
if (clientTCP.connect(myDomain, 443)) {
String head = "—boundary\r\nContent–Disposition: form–data;
name=\"chat_id\"; \r\n\r\n" + CHAT_ID + "\r\n—boundary\r\nContent–
Disposition: form–data; name=\"photo\"; filename=\"photo.jpg\"\r\nContent–
Type: image/jpeg\r\n\r\n";
String tail = "\r\n—boundary—\r\n";
size_t totalLen = fb–>len + head.length() + tail.length();
clientTCP.println("POST /bot" + BOTtoken + "/sendPhoto HTTP/1.1");
clientTCP.println("Host: " + String(myDomain));
clientTCP.println("Content–Length: " + String(totalLen));
clientTCP.println("Content–Type: multipart/form–data;
```

```
boundary=boundary");
        clientTCP.println();
        clientTCP.print(head);
        clientTCP.write(fb–>buf, fb–>len);
        clientTCP.print(tail);
        esp_camera_fb_return(fb);
        while (clientTCP.connected()) {
        if (clientTCP.available()) getBody += (char)clientTCP.read();
        }
        clientTCP.stop();
        }
        return getBody;
        }

        int findCardIndex(byte uid[UID_SIZE]) {
        for (int i = 0; i < MAX_CARDS; i++) {
        bool match = true;
        for (int j = 0; j < UID_SIZE; j++) {
        if (cardUIDs[i][j] != uid[j]) {
        match = false;
        break;
        }
        }
        if (match) return i;
        }
        return –1;
        }
```

```
void addCard(byte uid[UID_SIZE], String name) {
for (int i = 0; i < MAX_CARDS; i++) {
if (cardNames[i] == "") {
for (int j = 0; j < UID_SIZE; j++) {
cardUIDs[i][j] = uid[j];
}
cardNames[i] = name;
return;
}
}
}

void setup() {
WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
Serial.begin(115200);
myservo.attach(SERVO_PIN);
lockDoor();
pinMode(FLASH_LED_PIN, OUTPUT);
pinMode(BUZZER_PIN, OUTPUT);
pinMode(BUTTON_PIN, INPUT_PULLUP);
pinMode(MOTION_PIN, INPUT);
digitalWrite(FLASH_LED_PIN, flashState);
configInitCamera();
WiFi.mode(WIFI_STA);
WiFi.begin(ssid, password);
clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT);
```

```
while (WiFi.status() != WL_CONNECTED) delay(500);
SPI.begin();
mfrc522.PCD_Init();
}

void loop() {
if (digitalRead(BUTTON_PIN) == LOW) {
unlockDoor();
delay(5000);
lockDoor();
}
if (digitalRead(MOTION_PIN) == HIGH) {
sendPhotoTelegram();
delay(30000);
}
if (sendPhoto) {
sendPhotoTelegram();
sendPhoto = false;
}
if (millis() > lastTimeBotRan + botRequestDelay) {
int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
while (numNewMessages) {
handleNewMessages(numNewMessages);
numNewMessages = bot.getUpdates(bot.last_message_received + 1);
}
lastTimeBotRan = millis();
}
```

```
        if (mfrc522.PICC_IsNewCardPresent() &&
mfrc522.PICC_ReadCardSerial()) {
        byte *uid = mfrc522.uid.uidByte;
        if (awaitingCard && pendingCardName != "") {
        addCard(uid, pendingCardName);
        bot.sendMessage(CHAT_ID, "Картку додано", "");
        pendingCardName = "";
        awaitingCard = false;
        } else {
        int index = findCardIndex(uid);
        if (index >= 0) {
        unlockDoor();
        delay(5000);
        lockDoor();
        } else {
        digitalWrite(BUZZER_PIN, HIGH);
        delay(500);
        digitalWrite(BUZZER_PIN, LOW);
        sendPhotoTelegram();
        bot.sendMessage(CHAT_ID, "Невідома картка", "");
        }
        }
        mfrc522.PICC_HaltA();
        }
        }
```

ДОДАТОК Б

Демонстраційний матеріал

№ документа	Позначення	Найменування	Додаткові відомості
		Текстові документи	
1	ГЮИК. 418250.010 ПЗ	Пояснювальна записка	А4, 68 с.
		Додаткові матеріали	
		Текст програми	А4, 11 с.
		Демонстраційний матеріал	А4, 9 с.

Змін.	Арк.	Номер докум.	Підпис	Дата	ГЮИК. 418250.010 ВД			
Розроб.		Мартинюк О.В.			Розроблення системи автоматизації доступу у виробниче приміщення з використанням технологій IIoT	Літера	Аркуш	Аркушів
Перевір.		Стародупцев				Н	1	1
						Кафедра КІТАР ХНУРЕ		
Н.контр.		Демська Н.П.						
Затв.		Невлюдов І.Ш.						