

ЗМІСТ

1.Вступ.....	4
2.Аналіз технічного завдання.....	5
3. Розробка функціональної схеми і вибір елементної бази	8
4.Створення бота на базі месенджера Telegram	12
5.Розробка алгоритму роботи.....	14
6.Написання текстів програми	15
7.Висновок.....	19
8.Література	20

					2019.КП.0501.306.18.00.00 ПЗ					
Змн.	Арк.	№ докум.	Підпис	Дата	Pet Feeder з управлінням через Telegram			Літ.	Арк.	Акрушіє
Розроб.		Магальяс Р.С.								
Перевір.		Нелопитко								
Реценз.								ТК ТНТУ КІ-306		
Н. Контр.										
Затверд.					Пояснювальна записка					

1 ВСТУП

Інтернет речей (Internet of Things – IoT) перетворює звичні для нас речі у нові пристрої, створюючи як розумні годинники, так і розумні міста. Він під'єднує далекі від Інтернету засоби до мережі та надає їм нові функції.

Термін “Інтернет речей” (або англійською “Internet of Things”, скорочено – IoT) вперше було сформульовано в кінці XX-го століття, у 1999 році. Це концепція комунікації об'єктів (“речей”), які використовують технології для взаємодії між собою та з навколишнім середовищем. Також ця концепція передбачає виконання пристроями певних дій без втручання людини. Таким чином, всі пристрої в будинках, в автомобілях, на користувачеві виконують обробку інформації, її аналіз та обмін між собою та, залежно від результатів, приймають рішення і виконують певні дії.

Сфера IoT – один із головних світових трендів. Старі-добрі пристрої стають частиною Інтернет мережі і виконують нові функції. Недарма цю галузь вважають рушієм 4-ї індустріальної революції, яка зараз триває у світі.

Pet Feeder (годівниця для тварин) з управлінням через бота у месенджері Telegram також є одним із IoT пристроїв. За допомогою цієї годівниці можна годувати тварин не залежно від того де буде знаходитись користувач, йому знадобиться тільки доступ до інтернету.

Месенджер Telegram, який було вибрано, як платформу для спілкування пристрою з людиною є доволі популярним сьогодні. Telegram працює на iOS, Android, Windows Phone, Windows, Mac OS і Linux, а також у вигляді веб-версії. Така мультиплатформенність дозволяє користувачу не бути залежним від додатку на телефоні, а мати доступ до пристрою за допомогою будь-якого приладу підключеного до інтернету.

					2019.КП.0501.306.18.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

2 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

Метою даного курсового проекту є розробка пристрою на основі Arduino для годування домашніх тварин на відстані за допомогою бота у месенджері Telegram (з Android/IOS приладів або з комп'ютера підключеного до інтернету).

На рисунку 2.1 зображена структурна схема елементів мікропроцесорної системи.

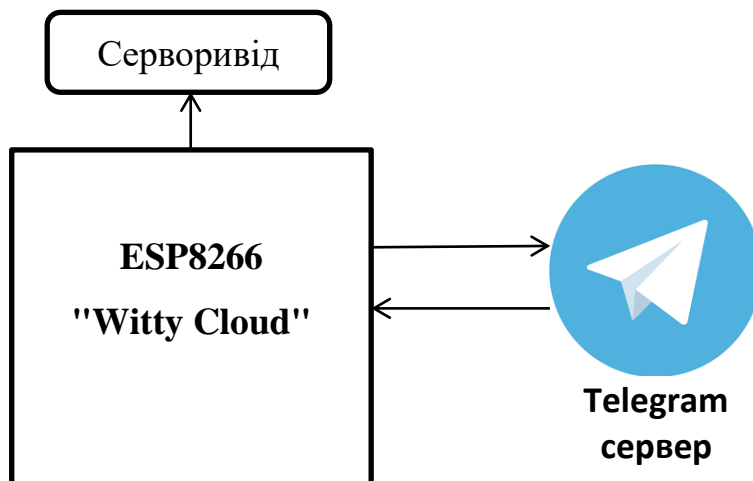


Рисунок 2.1 - Структурна схема мікропроцесорної системи

Дана система складається з самого пристрою і Web-частини.

Елементи пристрою:

-Arduino ESP8266 "Witty Cloud" - призначений для обробки даних та надсилання сингалів, а також для з'єднання з Web частиною і обміном інформацією з нею;

-Сервопривід – для дікриття дверки подачі корму.

Web-частина – це бот на базі месенджера Telegram який працює на API серверів Telegram за допомогою якого і буде здійснюватись управління

пристроєм через інтернет. На рисунку 2.2 зображене вікно Telegram бота який відкритий у месенджері на базі Android.

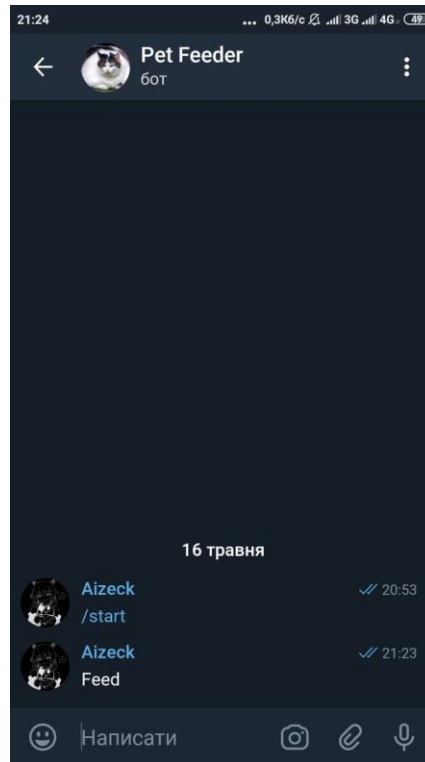


Рисунок 2.2 - Вікно Telegram бота(Android)

Для виконня пристроєм годування потрібно відправити боту повідомлення “Feed”.

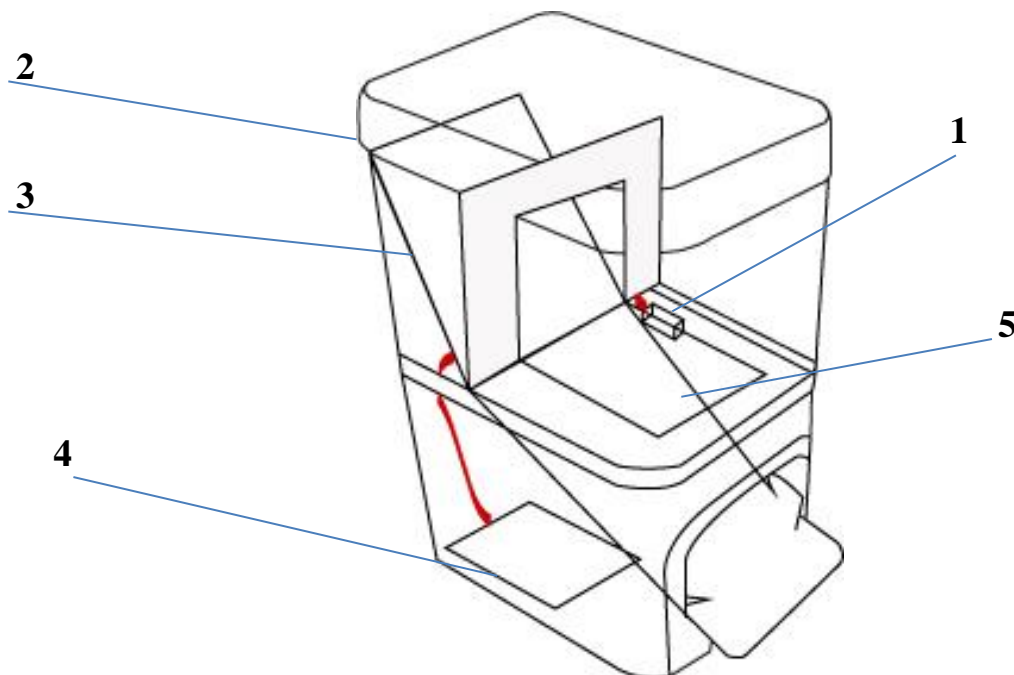


Рисунок 2.3 - 3D модель пристрою

					2019.КП.0501.306.18.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

Опис рисунку 2.3:

- 1.Сервопривід.
- 2.Знімна кришка.
- 3.Відділ для корму.
- 4.Плата управління.
- 5.Дверка яка затримує корм.

Порядок роботи з пристроєм

1. Підєднання Arduino до комп'ютера і налаштування параметрів WiFi мережі за допомогою Arduino IDE, а саме: ввести логін, пароль і Telegram ID у відповідні місця у коді.

2. Підєднання пристрою до електричної мережі.

3. Наповнення відділу для корму.

4. Вхід у месенджер телеграм.

5. Пошук бота у месенджері (@PFATbot або за посиланням t.me/PFATbot).

6. Для виконання одноразового годування потрібно надіслати боту команду “/feed”.

					2019.КП.0501.306.18.00.00 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

3 РОЗРОБКА ФУНКЦІОНАЛЬНОЇ СХЕМИ І ВИБІР ЕЛЕМЕНТНОЇ БАЗИ

Arduino ESP8266 "Witty Cloud"

Система розробляється на базі мікропроцесорної платформи

Arduino ESP8266 "Witty Cloud". Дана плата використовується у пристрої для управління сервоприводом та зв'язком з сервером бота месенджера Telegram.

Плата Arduino ESP8266 "Witty Cloud" представляє з себе систему для розробки на базі чіпа ESP8266(версії ESP12E), який є UART-WiFi модулем з ультранизким енергоспоживанням.

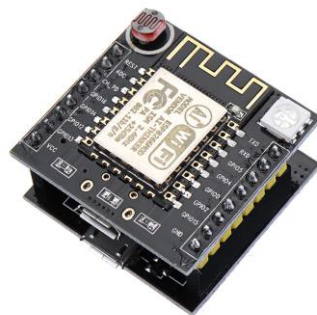


Рисунок 3.1 - Зовнішній вигляд Arduino ESP8266 "Witty Cloud"

Характеристики:

- WiFi стандарту 802.11 b / g / n
- підтримка STA / AP / STA + AP режимів
- вбудований стек протоколів TCP / IP з підтримкою багатокористувацьких підключень(до 5 одночасно)
- шифрування: WPA / WPA2
- живлення: 3.7 - 12V / 220 мА, живлення від USB з наданням налагоджувального інтерфейсу
- споживання: ~70 мА (240 мА максимум), очікування <200 мкА
- швидкість передачі: 110-460800 б/сек
- підтримка UART / GPIO інтерфейсів передачі даних
- перепрошивка з хмари або через USB

					2019.КП.0501.306.18.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

- відстань між контактними пінами: 28 мм
- діапазон робочих температур: -40 ~ +125 °C
- маса: 18 г
- розмір: 30 x 31x 18 мм

На рисунку 3.2 можна побачити призначення виводів плати ESP8266 "Witty Cloud". А саме, модуль має 12 портів вводу-виводу спільного призначення. Деякі з портів мають додаткові функції:

- D6-8, 4 – для RGB світодіода розташованого на платі
- D2 вивід кнопки яка розташована на платі
- A0 вивід фоторезистора з плати
- порт перезапуску плати

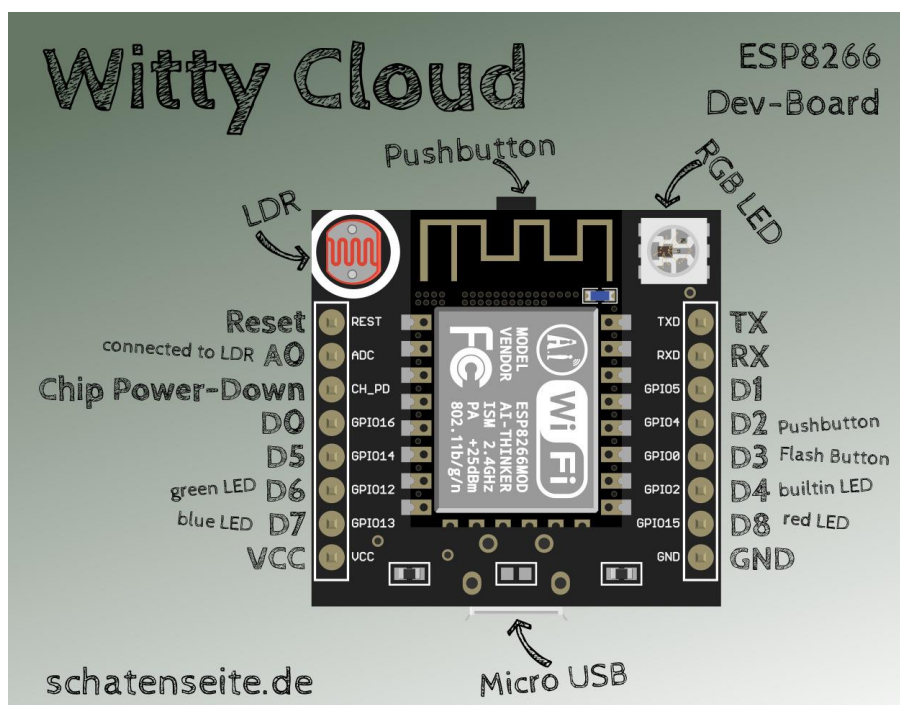


Рисунок 3.2 - Схема виводів плати Arduino ESP8266 "Witty Cloud"

Сервопривід TowerProSG90 microservo 9g

Сервопривід використовується у даній системі для відкриття та закриття дверки через яку подається корм.

Сервопривід – це привід з управлінням через негативний зворотний зв'язок, що дозволяє точно керувати параметрами руху. Сервоприводом є будь-який тип механічного приводу, що має в складі датчик (положення, швидкості, зусилля і т.п.) і блок керування приводом, автоматично підтримує

					2019.КП.0501.306.18.00.00 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

необхідні параметри на датчику і пристрої згідно заданому зовнішньому значенням. Зовнішній вигляд сервоприводу зображений на рисунку 3.3.



Рисунок 3.3 - Зовнішній вигляд сервоприводу

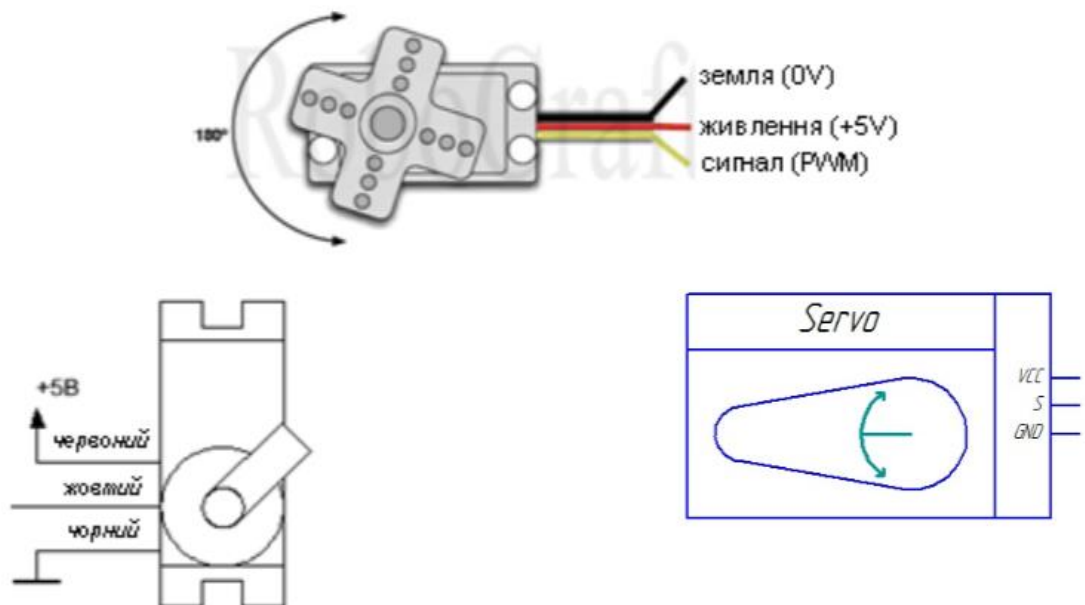


Рисунок 3.4 - Сервопривід та його умовне позначення на схемі

Сервопривід TowerProSG90 microservo 9g має три виводи:

- VTG – живлення;
- GNR – сигнал;
- SIG(S) – земля.

Характеристики TowerProSG90 microservo 9g:

					2019.КП.0501.306.18.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

- На рисунку 3.5 зображена схема підключення сервоприводу TowerProSG90 до плати управління Arduino ESP8266 "Witty Cloud".



The diagram shows a Witty Pi board with the following connections for a servo motor:

- VCC:** Connected to the servo's positive terminal (+).
- GND:** Connected to the servo's ground terminal (GND).
- PWM:** Connected to the servo's signal terminal (pulse) using a green wire.

The servo motor is labeled "Servo" and has a horn with a pulley. The Witty Pi board is labeled "Witty Pi" and "Gizwits ESP-12F".

Рисунок 3.6 - Функціональна схема

4 СТВОРЕННЯ БОТА НА БАЗІ МЕСЕНДЖЕРА TELEGRAM

Telegram — це месенджер, програмне забезпечення для смартфонів, планшетів та ПК, яке дозволяє обмінюватися текстовими повідомленнями та різноманітними файлами, зокрема графічними файлами та відеофайлами, а також безкоштовно телефонувати іншим користувачам програми.

За допомогою спеціального API сторонні розробники можуть створювати «ботів», спеціальні акаунти, керовані програмами. Типові боти відповідають на спеціальні команди в персональних і групових чатах, також вони можуть здійснювати пошук в інтернеті або виконувати інші завдання, застосовуються задля розваг або в бізнесі.

Для створення бота потрібно увійти в Telegram.

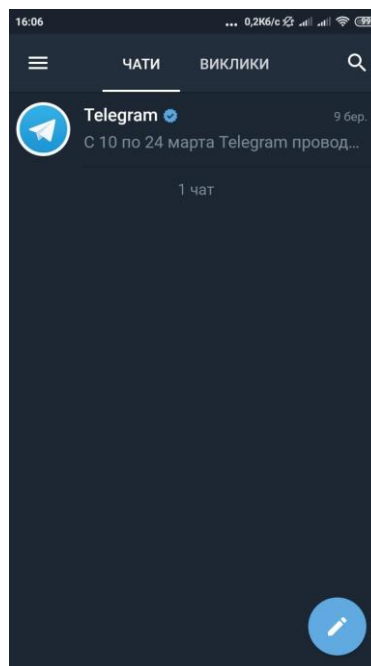


Рисунок 4.1 - Вікно месенджера Telegram X(Android)

Для отримання власного токена бота (API) потрібно знайти бота @botfather (рисунок 4.2).

Звертаємось до нього, прописуємо команду /newbot – створення нового бота. Далі вказуємо ім'я бота та його користувацьке ім'я. Після чого отримаємо токен власного бота (API).

					2019.КП.0501.306.18.00.00 ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

Цей токен використовується у коді програми для підключення ESP8266 до серверу Telegram.



Рисунок 4.2 - Бот @botfather

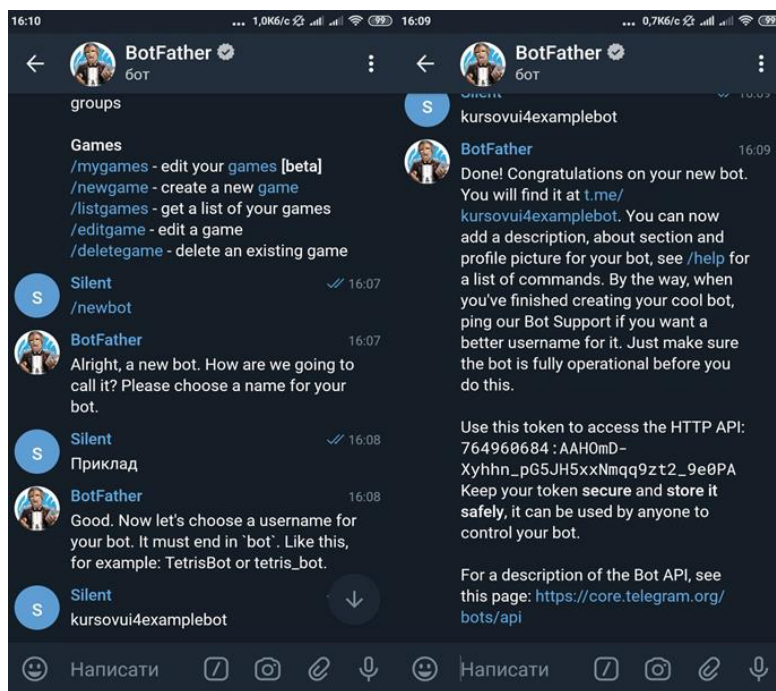


Рисунок 4.3 - Команда /newbot та API нового бота

					2019.КП.0501.306.18.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

5 РОЗРОБКА АЛГОРИТМУ РОБОТИ

Алгоритм роботи Pet Feeder з управлінням через Telegram:

1. Ініціалізація бібліотек.
2. Оголошуються та ініціалізуються змінні та об'єкт сервоприводу.
3. Ініціалізація порта сервоприводу.
4. Ініціалізація констант WiFi та API бота.
5. Підключення та початкова перевірка WiFi з'єднання.
6. Опис методу повороту сервоприводу для виконання годування.
7. Перевірка надходження повідомлень боту.
8. Якщо повідомлення “/feed”, то виконується годування (відкриття та тряска віконцем з якого надходить корм) та надсилається повідомлення у відповідь “Годування проведено”.
9. Повернення до пункту 7.



Рисунок 5.1 - Блок-схема алгоритму

6 НАПИСАННЯ ТЕКСТІВ ПРОГРАМИ

```
#include <ESP8266WiFi.h>           //
#include <ESP8266WiFiMulti.h>       // Підключення WiFi бібліотек
#include <ESP8266HTTPClient.h>      //
#include <WiFiClientSecureBearSSL.h> //
#include <Servo.h>                  // Підключення бібліотеки сервоприводу

Servo servo1;                      // створення об'єкту сервоприводу
int servoPIN=D0;                   // контакт сервоприводу

int Y0=0;                          // кут сервоприводу в закритому положенні
int Y1=45;                         // кут сервоприводу в відкритому положенні
int t=40;                          // затримка для режиму тряски

const uint8_t fingerprint[20] = {0xBB, 0xDC, 0x45, 0x2A, 0x07, 0xE3, 0x4A,
0x71, 0x33, 0x40, 0x32, 0xDA, 0xBE, 0x81, 0xF7, 0x72, 0x6F, 0x4A, 0x2B,
0x6B};

ESP8266WiFiMulti WiFiMulti;
HTTPClient https;

std::unique_ptr<BearSSL::WiFiClientSecure>client(new
BearSSL::WiFiClientSecure);

#define btnSend D2

boolean pressedButton = false;
boolean answer = false;
boolean firstTime=true;

String payload;
String messageId = "";
String timeMessage = "time";
String timeNewMessage = " ";

String token = "719976365:AAF4ydWGCUKXceXer0bQwAHFz4-
hTyk7E5w"; //ТОКЕН бота
String userId = "229136130"; //Telegram ID користувача

void setup() {
    pinMode(btnSend, INPUT);

    Serial.begin(115200);
```

					2019.КП.0501.306.18.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

```

Serial.println();
Serial.println();
Serial.println();

for (uint8_t t = 4; t > 0; t--) {
    Serial.printf("[SETUP] WAIT %d...\n", t);
    Serial.flush();
    delay(1000);
}

WiFi.mode(WIFI_STA);

WiFiMulti.addAP("arduinoobot", "arduinoobot"); // логін та пароль до WiFi
}

void sendBotMessage(String message) {
    HTTPS_GET("https://api.telegram.org/bot" + token +
"/sendMessage?chat_id=" + userId + "&parse_mode=html&text=" + message);
}

String readBotMessage() {
    HTTPS_GET("https://api.telegram.org/bot" + token + "/getUpdates");
    return answerBot();
}

void HTTPS_GET(String messageLink) {
    Serial.print("[HTTPS] begin...\n");
    if (https.begin(*client, messageLink)) {
        Serial.print("[HTTPS] GET...\n");
        int httpCode = https.GET();

        if (httpCode > 0) {
            Serial.printf("[HTTPS] GET... code: % d\n", httpCode);
            if (httpCode == HTTP_CODE_OK || httpCode ==
HTTP_CODE_MOVED_PERMANENTLY) {
                payload = https.getString();
                Serial.println(payload);
            }
            else {
                Serial.printf("[HTTPS] GET... failed, error: % s\n",
https.errorToString(httpCode).c_str());
            }
            https.end();
        }
        else {
            Serial.printf("[HTTPS] Unable to connect\n");
        }
    }
}

String answerBot() {

```

```

int posId = payload.lastIndexOf("{\"id\":");
userMessageId = payload.substring(posId + 6);
posId = userMessageId.indexOf("\"");
userMessageId = userMessageId.substring(0, posId - 1);

int posDate = payload.lastIndexOf("\"date\":");
timeNewMessage = payload.substring(posDate + 6);
posDate = timeNewMessage.indexOf("\"");
timeNewMessage = timeNewMessage.substring(0, posDate - 1);

int posText = payload.lastIndexOf("\"text\":");
String MessageText = payload.substring(posText + 8);
posText = MessageText.indexOf("\"");
MessageText = MessageText.substring(0, posText);

// if (timeMessage == "time"); MessageText= " ";
return MessageText;
}

void loop() {

    if ((WiFiMulti.run() == WL_CONNECTED)) {

        client->setFingerprint(fingerprint);

        if (firstTime) {sendBotMessage("Привіт, годівниця готова до роботи!
Напишіть /feed для разового годування");
//повідомлення при ввімкненні пристрою
firstTime=false;
delay(3000);
}

        String botCommand = readBotMessage();

        if (userMessageId.equals(userId) && timeNewMessage!=timeMessage) {
            timeMessage=timeNewMessage;
            // if owner's command and new message

            if (botCommand.equals("/feed"))
            {
                Open () ;
                sendBotMessage("Годування проведено");
            }
        }
    }
}

```

```

else if (botCommand.equals("/start")) {
    String welcome = "Для годування відправте повідомлення /feed";
    // початкове повідомлення
    sendBotMessage(welcome);
}
}
Serial.println("Wait 7s before next round...");
delay(3000);
}
}

void Open() //метод повороту сервоприводу
{
    servo1.attach(servoPIN);
    int pos;
    for(pos = Y0; pos < Y1; pos += 5)
    // цикл плавного відкривання з кроком 5 градусів
    {
        servo1.write(pos);
        delay(20);
    }
    delay(20);

    for(pos = Y1; pos >= Y0; pos -= 8) // цикл закривання з режимом тряски
    {
        servo1.write(pos);
        delay(t);
        servo1.write(pos-5);
        delay(t);
        servo1.write(pos);
        delay(t);
    }
    for(int i = 0; i <= 3; i += 1) //додаткова тряска
    {
        servo1.write(14);
        delay(400);
        servo1.write(Y0);
        delay(200);
    }
    delay(1500);
    servo1.detach();
}

```


7 ВИСНОВОК

У процесі виконання курсового проекту було розроблено Pet Feeder з управлінням через месенджер Telegram.

Пристрій віддалено годує тварин за допомогою бота у месенджері Telegram, а також даний проект може бути використаний як приклад для створення більш складних пристроїв, які можуть керуватись віддалено, через месенджер Telegram.

Під час виконання даної роботи, я навчився створювати та описувати функціональні схеми, складати алгоритми та програми на мові C для плат Arduino.

					2019.КП.0501.306.18.00.00 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

8 ЛІТЕРАТУРА

1.Arduino [Електронний ресурс] - Режим доступу до ресурсу: www.arduino.cc Дата доступу: 28.05.2019. – Заголовок з екрану.

2.ARDUI.NO.UA [Електронний ресурс] - Режим доступу до ресурсу: arduino.ua Дата доступу: 29.05.2019. – Заголовок з екрану.

3.ArduinoMaster [Електронний ресурс] - Режим доступу до ресурсу: arduino-master.ru Дата доступу: 03.06.2019. – Заголовок з екрану.

4.MYSKU [Електронний ресурс] - Режим доступу до ресурсу: mysku.ru Дата доступу: 03.06.2019. – Заголовок з екрану.

5. Internet of Things [Електронний ресурс] - Режим доступу до ресурсу: iot.lviv.ua Дата доступу: 04.06.2019. – Заголовок з екрану.

6. SCHATENSEITE [Електронний ресурс] - Режим доступу до ресурсу: www.schatenseite.de Дата доступу: 04.06.2019. – Заголовок з екрану.

7.Telegram Messenger [Електронний ресурс] - Режим доступу до ресурсу: telegram.org Дата доступу: 02.06.2019. – Заголовок з екрану.

8.GitHub [Електронний ресурс] - Режим доступу до ресурсу: github.com Дата доступу: 04.06.2019. – Заголовок з екрану.

					2019.КП.0501.306.18.00.00 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20