

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ «ИНЖЕНЕРНАЯ ШКОЛА № 1581»

Разработка автоматической кормушки на базе
ArduinoIDE.

Разработал:

ученик 11 «Л» класса

ГБОУ Инженерная школа №1581

Андреев Глеб Владимирович

Научный руководитель:

Преподаватель ГБОУ Инженерная школа
№1581

Ольховская Елена Александровна

Москва 2024.

Аннотация.

Цель работы:

Спроектировать и собрать рабочую кормушку. Написать программу для кормушки на языке Kotlin.

Задачи:

1. Провести анализ существующих вариантов кормушки;
2. Изучить язык программирования Kotlin;
3. Изучить среду разработки Arduino IDE;
4. Написать программу, позволяющую взаимодействовать с кормушкой через локальную сеть при помощи мобильного приложения;
5. Разработать программное обеспечение для платы «Arduino», отправляющую данные в приложение по локальной сети;
6. Найти корпус для кормушки с подходящим механизмом;
7. Собрать рабочий прототип кормушки;

Методы и инструменты:

1. Использование языка программирования Kotlin использования среды разработки Android Studio. Микроконтроллер Arduino esp8266
2. Использование среды разработки Arduino IDE

Результаты:

Собран рабочий прототип кормушки способный взаимодействовать с мобильным приложением на Android.

Выводы:

1. Получены навыки работы в средах Android Studio и Arduino IDE
2. Освоены базовые навыки работы с Kotlin
3. Получен опыт работы с микроконтроллером esp8266.

Оглавление.

1.ВВЕДЕНИЕ	4
2.АНАЛИТИЧЕСКИЙ РАЗДЕЛ	5
2.1 Поилка	5
2.2 Динамик	7
2.3 Корпус	7
3.КОНСТРУКЦИОННЫЙ РАЗДЕЛ	8
3.1 Электронные компоненты	8
4.ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ	11
4.1 Фаза кормления	11
4.2 Контекстная диаграмма	11

1. ВВЕДЕНИЕ.

В настоящее время многие люди проводят большую часть времени вне дома, однако такая проблема как надобность в кормлении, например, кошки может помешать планам хозяина, из-за чего он будет вынужден прийти домой раньше обычного.

Уезжая на выходные загород или уходя в отпуск, хозяева питомцев просят друзей или родственников проследить за их домашним животным. Иногда случается так, что своего четвероного друга оставить не с кем.

В связи с подобными случаями целью моего проекта стала разработка автоматической кормушки, которая позволяла бы осуществлять процесс кормления автоматически на расстоянии.

В данный момент существует множество аналогичных устройств, но их цена и реализация кажутся мне несоизмеримыми. Рассмотрим пару из таких устройств.

1) “Xiaomi Mijia Smart Pet Feeder XWPF01MG” (рис. 1.1)

Кормушка выполнена из качественных материалов, имеет свое мобильное приложение и поддержку умного дома. Я считаю приложение этой кормушки довольно мало функциональным, так как оно позволяет лишь настраивать таймер выдачи еды, без просмотра статуса продуктов и просмотра камеры наблюдения. Поилка - отсутствует, камера - отсутствует. Цена - 10 500 руб.



Рисунок 1.1 - фотография “Xiaomi Mijia Smart Pet Feeder XWPF01MG”

2) KitFort KT-2081 (рис. 1.2)

В отличие от Xiaomi обладает камерой для просмотра за питомцем, однако у этой кормушки нет приложения и все так же нет поилки, что очень важно, если хозяин решит оставить своего питомца одного надолго. Цена - 12 000 руб.



Рисунок 1.2 - KitFort KT-2081

2.АНАЛИТИЧЕСКИЙ РАЗДЕЛ

2.1 Поилка

Практически все аналогичные автоматические кормушки не оснащены поилкой. Я считаю, что поилка в кормушке обязательно должна быть так как жидкость необходима для нормального функционирования организма.

Поэтому важно отслеживать не только количество корма, потребляемого кошкой, но и жидкости.

Рассмотрим 2 Варианта поилки – «фонтанчик» и обычная механическая поилка.

1) Поилка – фонтанчик имеет ряд плюсов, однако рассматривая ее как вариант поилки в автоматической кормушке возникает ряд недостатков. Работает поилка по принципу циркуляции воды с помощью помпы по замкнутому контуру

- Сборка герметичного прототипа выйдет гораздо трудозатратнее
- Оставлять фонтан все время включенным будет непрактично.

Более того, кошки и собаки пьют воды столько, сколько ему нужно организму, поэтому жестко контролировать подачу жидкости нет необходимости.

2) Поилка, работающая по принципу сообщающихся сосудов, будет более надежной. В ней отсутствуют минусы, указанные в предыдущей вариации поилки.



Рисунок 2.1 - поилка-фонтанчик



Рисунок 2.2 - “обычная” поилка

2.2 Динамик

Кормушка KitFort КТ-2081 (рис. 1.2), как и многие другие оснащена динамиком, с помощью которого, во время питания животного будет воспроизводиться запись голоса хозяина. На мой взгляд, данная опция не имеет смысла, поэтому было принято решение отказаться от её использования в данном проекте.

2.3 Корпус

В качестве прототипа была выбрана кормушка «Smart Pet Feeder» с простым механизмом работы и встроенным преобразователем переменного тока 5 вольт, что необходимо для питания Arduino. В дальнейшем будем использовать готовый корпус и электродвигатель, которым оснащена данная модель.



Рис. 2.3 – Кормушка «Smart Pet Feeder».

3.КОНСТРУКЦИОННЫЙ РАЗДЕЛ

3.1 Электронные компоненты

1 ESP32-CAM - камера (рис. 3.3)

2 ESP8266 - управляющая плата (рис 3.2)

3 HC-SR04 - ультразвуковой датчик (рис. 3.1)

4 Электродвигатель (рис 3.4)

5 Кнопка

6 Соединительные провода



Рисунок 3.1



Рисунок 3.2



Рисунок 3.3

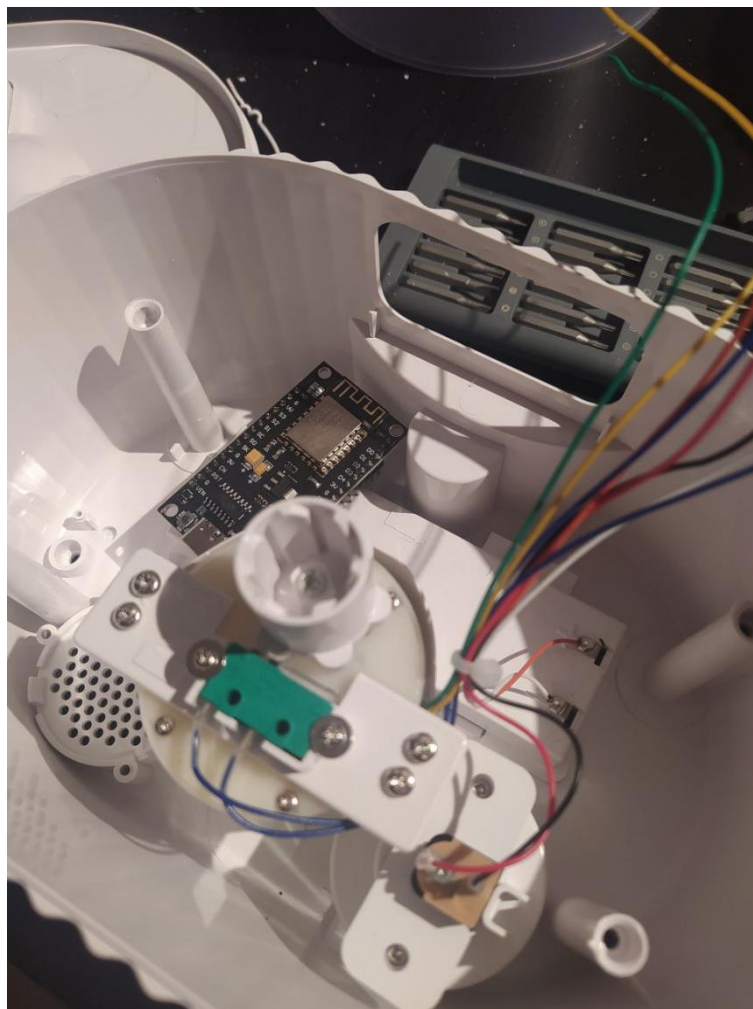


Рисунок 3.4

4.ТЕХНОЛОГИЧЕСКИЙ РАЗДЕЛ.

4.1 Фаза кормления

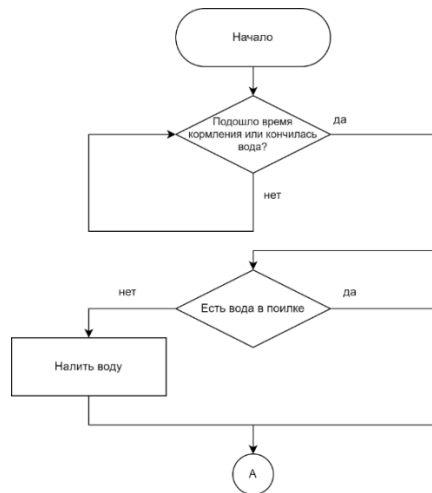


Рисунок 2.1

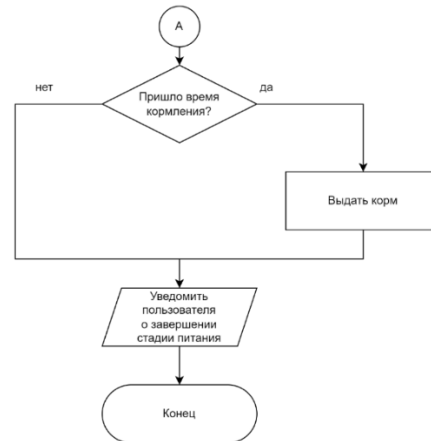


Рисунок 2.2

Процесс кормления осуществляется следующим образом:

1. Кормушка считывает значения с датчиков и определяет уровень жидкости и корма. Если уровень того или иного содержимого ниже минимального, пользователю придет соответствующее уведомление в приложении. Если уровень в норме - кормушка готова к выдаче корма и воды.
2. В назначенное время кормушка выдает заданное количество корма, присылает пользователю сколько корма осталось и уведомляет о том, что процесс кормления завершен. (рис. 2.2)

4.2 Контекстная диаграмма



Рисунок 2.3 - контекстная диаграмма

Приложение.

Фрагмент кода платы.

```
#define butt 8
#define ledOk 2
#define ledOut 3
#define motor 4
#define uFoodIn 5
#define uFoodOut 6
#define key 7

void setup() {
  pinMode(motor, OUTPUT);
  pinMode(uFoodOut, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if (butt == 1){
    while(key != 0){
      digitalWrite(motor, 1);
      delay(50);
    }
    digitalWrite(motor, 0);
    delay(100);
  }

  float duration, distance;
  digitalWrite(uFoodOut, LOW);
  delayMicroseconds(2);

  digitalWrite(uFoodOut, HIGH);
  delayMicroseconds(10);
  digitalWrite(uFoodOut, LOW);

  duration = pulseIn(uFoodIn,
HIGH);
  distance = (duration / 2) * 0.0344;

  if (distance >= 400 || distance <=
2){
    Serial.print("Distance = ");
    Serial.println("Out of range");
  }
  else {
    Serial.print("Distance = ");
    Serial.print(distance);
    Serial.println(" cm");
    delay(500);
  }
  delay(500);
}
```

Фрагмент кода камеры.

```
#define butt 8
#define ledOk 2
#define ledOut 3
#define motor 4
#define uFoodIn 5
#define uFoodOut 6
#define key 7

void setup() {
  pinMode(motor, OUTPUT);
  pinMode(uFoodOut, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  if (butt == 1){
    while(key != 0){
      digitalWrite(motor, 1);
      delay(50);
    }
    digitalWrite(motor, 0);
    delay(100);
  }

  float duration, distance;
  digitalWrite(uFoodOut, LOW);
  delayMicroseconds(2);

  digitalWrite(uFoodOut, HIGH);
  delayMicroseconds(10);
  digitalWrite(uFoodOut, LOW);

  duration = pulseIn(uFoodIn,
HIGH);
  distance = (duration / 2) * 0.0344;

  if (distance >= 400 || distance <=
2){
    Serial.print("Distance = ");
    Serial.println("Out of range");
  }
  else {
    Serial.print("Distance = ");
    Serial.print(distance);
    Serial.println(" cm");
    delay(500);
  }
  delay(500);
}
```