

Омский Научный центр Сибирского отделения Российской академии наук
Региональная общественная организация «Омский совет ректоров»
Омское региональное отделение Всероссийской общественной организации
«Русское географическое общество»
Детская областная общественная организация
«Научное общество учащихся «Поиск»
Бюджетное общеобразовательное учреждение Полтавского района
«Полтавский лицей»

53-я

Межрегиональная научно-практическая конференция
школьников и учащейся молодежи

Тема: «Создание инкубатора, управляемого со смартфона на ОС
Android, на базе ESP32 и Arduino»
Учебно-исследовательская работа
Научное направление: информатика 10-11 класс

Выполнил:
ученик 10 класса
БОУ «Полтавский лицей»
Кондратенко Данила Александрович

Научный руководитель:
Школа №120
Шелихов Олег Юрьевич

Омск - 2020

Оглавление

Введение.....	3
Определения.....	4
Реализация.....	6
Строение инкубатора.....	6
Исполнительный блок инкубатора.....	6
Модуль дистанционного управления.....	7
Протокол дистанционного управления исполнительным модулем.....	9
Приложение для смартфона на базе Android.....	11
Заключение.....	11
Список литературы.....	13

Введение

В современном мире всё больше различных процессов возлагается на различного рода автоматику. Это позволяет исключить человеческий фактор там, где он может стоить очень дорого, а также облегчить человеческий труд, особенно на селе.

Кроме того, в настоящее время всё больше устройств управляются при помощи приложения для смартфона. Такой способ управления очень удобен, так как не требует создания отдельного пульта дистанционного управления.

Цель работы: создать инкубатор, управляемый со смартфона, на базе ESP32 и Arduino.

Задачи:

1. Изучить технологию инкубации яиц.
2. Разработать конструкцию инкубатора и собрать его.
3. Изучить язык C++ и библиотеки Arduino и ESP32.
4. Собрать исполнительный блок на базе Arduino.
5. Собрать модуль дистанционного управления на базе ESP32.
6. Изучить язык Java и интерфейс разработки приложений для Android.
7. Создать приложение, при помощи которого можно управлять инкубатором со смартфона.

Методы исследования:

- изучение справочной литературы;
- создание продукта в рамках данного исследования.

Определения

ESP32 — микроконтроллер, разработанный в китайской компании Espressif. Оснащён 32-битным процессором Tensilica Xtensa LX6 с тактовой частотой 240 МГц. Поддерживает Wi-Fi, что во многом определяет его применение.[1]

Arduino — платформа для разработки электронных устройств, в основном, на базе восьмибитных микроконтроллеров AVR ATmega, однако существуют варианты для иных микроконтроллеров, в том числе ESP32. Для программирования используется специальное расширение языка C++, именуемое Wiring, которое включает собственный набор классов для разработки приложений для данной платформы. [2]

C++ – компилируемый язык программирования, разработанный Бьярне Струstrupом на основе языка Си в 1983 году. Типизация статическая, т. е. Объявленная в коде переменная не способна менять свой тип в ходе работы программы. Большое внимание в языке уделяется развитию объектно-ориентированного программирования, т. е. Представлению программы в виде совокупности работающих объектов, относящихся к разным классам.

Стандартизован Международной организацией по стандартизации (ISO). Последний стандарт имеет название C++17. [3]

Android — операционная система для портативных компьютеров, разработанная компанией Google. Первая версия вышла в 2008 году. Основана на ядре Linux и включает в себя собственную реализацию виртуальной машины Java, которая необходима для запуска приложений на всех устройствах с предустановленным Android. [4]

Java — компилируемый объектно-ориентированный язык программирования со статической типизацией, созданный в корпорации Sun Microsystems в 1995 году, позднее купленный компанией Oracle.

Особенностью этого языка программирования является то, что код на этом языке преобразуется компилятором не в инструкции процессора, а в команды для виртуальной машины Java (т. н. байт-код), а значит, могут исполняться на любом процессоре, для которого есть её реализация.

Эта особенность определяет применение языка: он, в основном, используется для написания приложений под операционную систему Android.

[5]

Реализация

Для создания инкубатора я изучил книгу Отрыганьева «Технология инкубации», а также справочник «Домашние инкубаторы», изданный в Ростове-на-Дону. Руководствуясь этими двумя книгами, я и создал данный инкубатор.

Строение инкубатора

Данный инкубатор состоит из:

- инкубационной камеры круглой формы с лотками для яиц;
- мотора, поворачивающего инкубационную камеру;
- датчиков положения инкубационной камеры;
- датчиков температуры и влажности;
- электронагревателя;
- клапана-оросителя;
- вентилятора;
- вентиляционного клапана.

Всё это управляется исполнительным блоком, соединённым с модулем дистанционного управления, который подаёт на него команды, приходящие с внешних устройств.

Исполнительный блок инкубатора

Исполнительный блок инкубатора выполнен на базе платы Arduino Nano (микроконтроллер AVR ATmega328p). К нему подключены датчики положения инкубационной камеры, температуры и влажности, а также все компоненты данного инкубатора.

Работа инкубатора организована по следующему алгоритму:

1. Снимаются показания с датчиков температуры и влажности.

2. Проверяется текущая температура.

1. Если температура ниже заданной на $0,3^{\circ}\text{C}$, то включается электронагреватель.
2. Если температура равна или больше заданной, то электронагреватель выключается.
3. Если температура продолжает повышаться, то при достижении температуры 39°C включается звонок и передаётся сигнал на смартфон, а при достижении температуры 43°C открывается вентиляционный клапан.

3. Проверяется текущая влажность.

1. Если влажность ниже заданной на 10%, то включается клапан-ороситель.
2. Если влажность равна заданной, то клапан-ороситель выключается.
3. Если влажность превышает 80%, то включается звонок и открывается вентиляционный клапан.

Поворот инкубационной камеры с яйцами осуществляется по таймеру через равные промежутки времени, которые определяются количеством поворотов яиц в день.

Вентилятор работает постоянно, так как он осуществляет циркуляцию воздуха в инкубационной камере.

В инкубаторе предусмотрены режимы ручного и автоматического регулирования.

Команды с модуля дистанционного управления принимаются по UART¹.

1 UART — Universal Asynchronous Receiver-Transmitter (Универсальный асинхронный приёмопередатчик) — последовательный интерфейс передачи данных.

Кроме управления по Wi-Fi предусмотрено управление с панели на корпусе инкубатора.

Модуль дистанционного управления

Модуль дистанционного управления выполнен на базе микроконтроллера ESP32. Он принимает приходящие по Wi-Fi HTTP-запросы, передаёт их по UART на исполнительный модуль и возвращает ответ, пришедший с исполнительного модуля.

Протокол дистанционного управления исполнительным модулем

Дистанционное управление инкубатором осуществляется при помощи соответствующего модуля, который по UART передаёт текстовые команды. Каждая текстовая команда представляет собой строку, оканчивающуюся символами «возврат каретки» и «перевод строки» (т. е. последовательностью байт 0x0D 0x0A), где слова разделены пробелами. Первое слово — наименование команды, последующие слова — аргументы. Производимое действие определяется первым словом:

Первое слово	Назначение	Принимаемые аргументы
request_state	Запросить состояние инкубатора	Аргументов не принимает
request_config	Запросить настройки инкубатора	Аргументов не принимает
needed_temp	Установить необходимую температуру	Принимается целое число либо десятичная дробь с точкой в качестве десятичного разделителя
needed_humid	Установить необходимую влажность	Принимается целое число либо десятичная дробь с точкой в качестве десятичного разделителя
rotations_per_day	Установить количество поворотов яиц в день	Принимается целое число

Первое слово	Назначение	Принимаемые аргументы
switch_to_program	Включить программу регулирования	Принимается целое число; 0 — переход на ручное регулирование >0 — переход на автоматическое регулирование

Таблица 1. Команды инкубатора для управления по UART

В ответ на команды инкубатор возвращает последовательность текстовых команд, аналогичных посылаемым, но для устройства-приёмника. В ответ могут быть посланы следующие команды:

Первое слово	Назначение	Аргументы
<i>Ответ на команду request_state</i>		
current_temp	Текущая температура	Десятичная дробь с двумя знаками после точки
current_humid	Текущая влажность	Десятичная дробь с двумя знаками после точки
cooler	Состояние вентилятора	0, если вентилятор выключен; 1, если вентилятор включен.
heater	Состояние нагревателя	0, если нагреватель выключен; 1, если нагреватель включен.
wetter	Состояние оросителя	0, если ороситель выключен; 1, если ороситель включен.
chamber	Положение инкубационной камеры	-1, если камера отклонена на 45° против часовой стрелки от нейтрального положения; 0, если камера находится в нейтральном положении; 1, если камера отклонена на 45° по часовой стрелке от нейтрального положения; 2, если положение камеры невозможно определить, при этом могут быть неисправны датчики положения; 3, если положение камеры невозможно определить, при этом ни один датчик положения не срабатывает.
current_program	Номер текущей программы регулирования	Целое число; 0 — ручное регулирование

Первое слово	Назначение	Аргументы
uptime	Время, прошедшее с начала инкубации	Целое число
overheat	Перегрев инкубатора	Аргументов не принимает
<i>Ответ на команду request_config</i>		
needed_temp	Необходимая температура	Десятичная дробь с двумя знаками после точки
needed_humid	Необходимая влажность	Десятичная дробь с двумя знаками после точки
rotations_per_day	Количество поворотов в минуту	Целое число
number_of_programs	Количество программ регулирования	Целое число
<i>Ответ на остальные команды</i>		
success	Ответ получен, команда применена.	

Таблица 2. Команды для управляющего устройства

Приложение для смартфона на базе Android

Инкубатор управляется дистанционно при помощи приложения для ОС Android. При запуске этого приложения в главной активности (IncubatorStateActivity) появляется модель разрабатываемого инкубатора, а также экран, подобный реальному экрану инкубатора с тремя кнопками: **(–)**, **(M)**, **(+)**. Кнопка **(M)** переключает пункты меню: текущее состояние, необходимую температуру, необходимую влажность, количество поворотов яиц в день и текущую программу управления инкубатором.

Модель инкубатора реализована при помощи объектов класса ImageView, отвечающих за элементы инкубатора, в частности, за экран, кнопки, нагреватель и инкубационную камеру, а также при помощи объекта класса TextView, который отвечает за отображаемое на экране.

Для отправки сетевых запросов применяется класс NetworkRequestTask собственного написания, являющийся наследником класса AsyncTask. Он не-

обходим для того, чтобы не загружать основной поток ожиданием ответа от сети. [6]

Заключение

В результате исследования был создан полностью автоматический и дистанционно управляемый инкубатор, который способен поддерживать эмбриональное развитие птицы, а также приложение к нему, связывающееся с инкубатором по Wi-Fi и наглядно показывающее состояние инкубатора и процесс инкубации.

Дальнейшие исследования показали, что связь по Wi-Fi имеет важный недостаток: дальность распространения сигнала ограничена. Дистанционное управление на большем расстоянии требует постоянной связи с Интернетом, а это означает аренду доменного имени и хостинга, а также, возможно, и подключение дополнительного модуля для выхода в Интернет (хотя ESP32 позволяет подключаться к общей Wi-Fi-сети). При желании всё это можно реализовать.

Однако, несмотря на вышеуказанные замечания, можно считать, что наша цель достигнута.

Список литературы

1. ESP32 // Википедия [Электронный ресурс]. - Режим доступа: <https://ru.wikipedia.org/wiki/ESP32>
2. Arduino // Википедия [Электронный ресурс]. - Режим доступа: <https://ru.wikipedia.org/wiki/Arduino>
3. C++ // Википедия [Электронный ресурс]. - Режим доступа: <https://ru.wikipedia.org/wiki/C++>
4. Android // Википедия [Электронный ресурс]. - Режим доступа: <https://ru.wikipedia.org/wiki/Android>
5. Java // Википедия [Электронный ресурс]. - Режим доступа: <https://ru.wikipedia.org/wiki/Java>
6. Keeping your app responsive. // Android Developers [Electronic resource]. - URL: <https://developer.android.com/training/articles/perf-anr>