

Всероссийский конкурс исследовательских и проектных работ школьников
«Высший пилотаж»

Инкубатор версия 3.0.
На пути к промышленному образцу
Направление «Технические и инженерные науки»

Автор: Семенкин Иван Максимович,
учащийся 6 класса,
ГБНОУ ПО «Губернский лицей»
Руководитель: учитель информатики
Гуляйкина Алина Игоревна

2024 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ3

ОПИСАНИЕ РАБОТЫ5

1 ЭТАП ПРОЕКТА – СОЗДАНИЕ КОРПУСА ИНКУБАТОРА5

2 ЭТАП ПРОЕКТА – СХЕМОТЕХНИКА6

**3 ЭТАП ПРОЕКТА – ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, WEB-ИНТЕРФЕЙС И
ОТКРЫТЫЕ ДАННЫЕ**8

ПРОБЛЕМЫ10

ТЕРМОРЕГУЛЯЦИЯ11

ВЫВОДЫ12

СТРУКТУРА КОМАНДНОЙ РАБОТЫ НАД ПРОЕКТОМ12

СПИСОК ЛИТЕРАТУРЫ14

Приложение 115

Приложение 217

ВВЕДЕНИЕ

В прошлом году я делал инкубатор. Изучил историю создания инкубаторов, посмотрел имеющиеся образцы на рынке, разобрался с режимами инкубации куриных и перепелиных яиц [1]. Это был мой первый опыт. Инкубатор был простой, но мог поддерживать температуру и влажность, что является необходимым условием для выведения цыплят. Я заложил 12 яиц и вывел перепелов, они прожили у нас примерно неделю, а потом мы их отдали соседям. При использовании стали видны недостатки инкубатора, такие как ненадежность системы и низкая выводимость птенцов.

Версии инкубатора

Я работал над двумя последовательными образцами, поэтому обозначил их как версия 1.0 и 2.0. В этом году при поддержке компании “Кодинсайд” которая входит в кружковое движение “НТИ” [2] я разработал версию инкубатора 3.0. Поддержка осуществлялась в научном отношении, техникой и основными элементами для разработки.

Цели и задачи:

Цель моей работы: создать улучшенную версию инкубатора (3.0), повысить надежность и удобство использования, расширить функциональные возможности. Я хочу, чтобы это был прототип промышленного образца, возможный в перспективе к серийному производству – дешевый и доступный умный инкубатор.

Задачи для достижения этой цели:

- a. Повысить вместимость корпуса инкубатора;
- b. Повысить прочность корпуса инкубатора;
- c. Повысить надежность схемотехники;
- d. Улучшить управление инкубатором;
- e. Повысить выводимость цыплят.
- f. Составить спецификацию материалов и смету к ней

Материалы

Для проекта потребовались разные материалы и элементы, но почти все мы нашли в магазинах города. Только греющие коврики были заказаны на АлиЭкспресс.

Для корпуса мне понадобились:

1. Экструдированный пенополистирол 1 м2,
2. Армированный скотч,
3. Клей ‘Титан’,
4. Просечная строительная сетка.

Для регулировки температуры и влажности:

5. Вентилятор
6. Греющие коврики

Для монтажа управляющего блока:

7. DHT22 - датчик температуры и влажности
8. Плата ESP8266
9. Соединительный провод
10. SSD1306 - дисплей
11. Электронные ключи
12. Монтажная коробка

Таблица 1. Смета на материалы

Пенопласт	142,00 Р
Электронный ключ	100,00 Р
Вентилятор	171,00 Р
ESP8266	114,31 Р
Монтажная плата	36,74 Р
Дисплей	135,41 Р
Провода	50,00 Р
Монтажная коробка	123,00 Р
Блок питания	300,00 Р
Магнитная лента	90,12 Р
Греющий коврик	116,00 Р
Датчик DHT 22	104,11 Р
Всего	1482,69 Р

Таким образом видно, что все материалы на изготовление инкубатора обошлись в сумму не более 1,5 тысяч рублей.

ОПИСАНИЕ РАБОТЫ

1 ЭТАП ПРОЕКТА – СОЗДАНИЕ КОРПУСА ИНКУБАТОРА

В прошлом году с изготовлением корпуса мне помог дедушка, а в этом году я вычертил детали корпуса в AutoCAD 2020, вырезал его, а потом склеил. Пенопласт я заменил на более плотный и твердый материал – экструдированный пенополистерол.

Для того чтобы сделать в дальнейшем поворот яиц необходимо оставить место для этого внутри корпуса, поэтому я увеличил габариты по отношению к предыдущей версии инкубатора. Теперь он стал более высоким и вместительным. Новые габариты: 29 см – глубина, 25 см - ширина, 27 см – высота. Пришлось отказаться от обзорного окошка, слишком большие были теплопотери через него. Стало не так удобно – не видно яиц, но для повышения выводимости это необходимая мера. Внутри предусмотрены опорные ребра для сетки.



Рисунок 1. Увеличенный корпус инкубатора версии 3.0



Рисунок 2. Сетка для яиц

2 ЭТАП ПРОЕКТА – СХЕМОТЕХНИКА

Основные элементы, необходимые для сборки управляющего блока.

1. ESP8266 — микроконтроллер с интерфейсом Wi-Fi. ESP8266 создан для использования в умных розетках, беспроводных сенсорах, носимой электронике и так далее. Можно сказать, что он появился на свет, чтобы стать мозгом «Интернета вещей».

2. NodeMCU v2 – это платформа разработки с открытым исходным кодом, которая подходит для создания прототипов IoT. Платформа широко распространена, ее можно купить в любом магазине радиодеталей, недорого.

Эта платформа была выбрана для использования в проекте, ее параметры:

- ▶ Напряжение питания: 5 В (от USB)
- ▶ Максимальный ток: 220 мА
- ▶ Габариты: 49 мм x 24.5 мм x 13 мм

Первым этапом на макетной плате я собрал прототип будущего контроллера, затем вычертил в программе Sprint Layout расположение элементов на монтажной плате и их коммутацию. Sprint Layout – это программа для проектирования монтажных плат, где можно добавить любые элементы, дорожки, места пайки и даже сверления [2].

Затем я взял монтажную плату, расположил элементы в соответствии с вычерченной монтажной схемой и соединил их между собой проводниками.

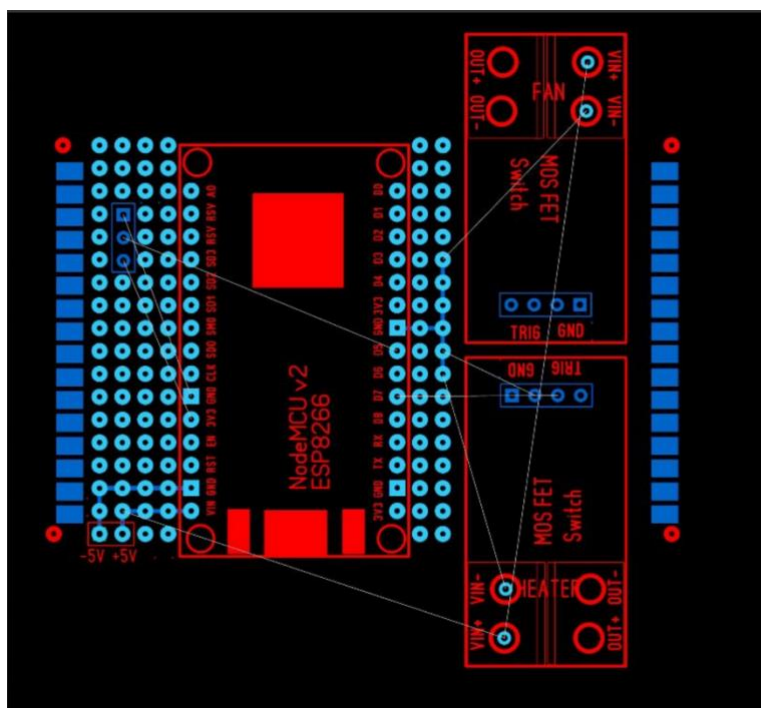


Рисунок 3. Вычерченная мной монтажная схема управляющего блока

Дальше подключил греющий коврик и датчик влажности к плате и всё заработало!

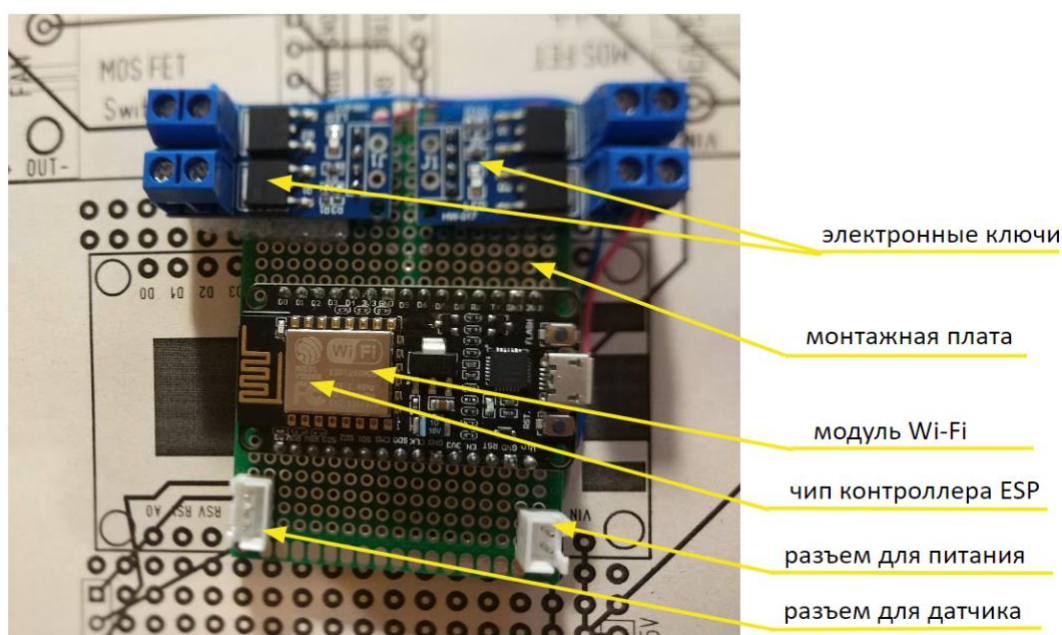


Рисунок 4. Монтажная плата в сборке

В предыдущей версии инкубатора все элементы не имели жесткого монтажа, теперь всё пропаяно, а монтажная плата компактно собрана в монтажной коробке. В корпусе - специальные посадочные места для вентилятора и датчиков. Дальше я планирую изготовить внешний корпус из фанеры и сделать углубления пазы для проводов, чтобы их тоже спрятать.

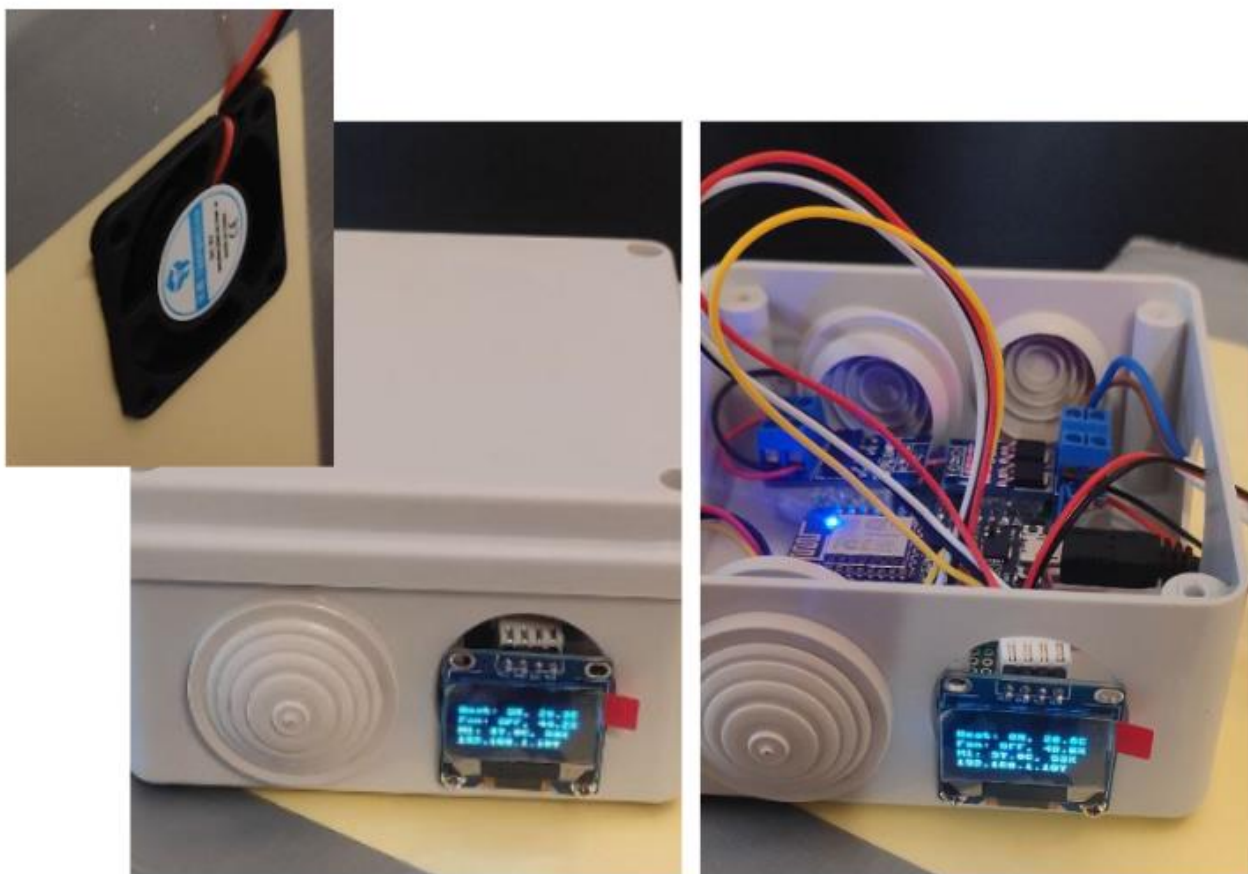


Рисунок 5. Монтажная коробка и посадочное место для вентилятора в корпусе

3 ЭТАП ПРОЕКТА – ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, WEB-ИНТЕРФЕЙС И ОТКРЫТЫЕ ДАННЫЕ

Написание управляющей программы для инкубатора — это отдельный большой этап работы, который длился параллельно и сопутствовал остальным этапам. Мы использовали профессиональный язык Python [3], который я учу со второго класса. Архитектором проекта был составлен план и я писал и несколько раз переписывал весь код по плану, который он составил. Весь код программы, представленный в приложении 2 написан мной.

На сайте моего инкубатора всего три страницы. На главной странице выводятся показания с датчика и показывается, какой режим включен. На странице настроек можно управлять режимами, вентилятором и подогревом. Третья страница текущей информации, в ней выводится год, месяц, дата, время окончания работы над программой. Код программы и внешний вид всех страниц приводится в Приложении 1 и Приложении 2.

Мой проект - с открытыми данными (open source). Я планирую делиться всеми наработками, чтобы любой мог сделать свой инкубатор. По ссылке на Яндекс-диске будут доступны: спецификация; чертежи; программный код; схемотехника.

СРАВНЕНИЕ РЕШЕНИЙ В ВЕРСИИ ИНКУБАТОРА 2.0 И 3.0

Сравнение текущих решений с решениями предыдущей версии иллюстрирует изменения и методы, которые для этого использовались. Перечисленные изменения позволили добиться поставленных мной целей.

1. Одноплатный компьютер Ардуино для учебных и любительских задач мы заменили на ESP, более надёжный и производительный, он также используется для задач промышленной автоматизации.

2. Использовали механическое реле для управления нагревателем, которые были шумные и малонадёжные, а теперь применяю электронное реле, которое бесшумно и не содержит механических частей, что повышает их надёжность.

3. Одно из главных изменений - инкубатор стал подключаться к сети, я использую концепцию “интернет вещей”. Традиционное управление с дисплея и переключателя прямо на корпусе инкубатора я заменил на управление через WEB-интерфейс.

4. Для нагрева использовалась лампочка накаливания, у которой нагрев является побочным эффектом, а теперь я использую специальные нагревательные коврики и ещё освободилось место.

5. Раньше я использовал вентилятор, который был под руками, это было нерационально, его хватало с большим избытком, теперь применяю вентилятор меньшего размера, менее шумный, который лучше подходит для данной задачи.

6. Программирование осуществлялось с помощью среды Ардуино, а здесь я использовал профессиональный язык Python.

Все изменения я собрал в таблицу.

Таблица 2. Сравнение версий инкубаторов 2.0 и 3.0

	Инкубатор 2.0	Инкубатор 3.0
Контроллер	Arduino Leonardo	NodeMCU ESP8266
Датчик температуры и влажности	DHT22	DHT22
Управление нагревателем и вентилятором	Механическое реле	Электронный бесшумный ключ
Управление инкубатором	Дисплей и переключатель	Веб интерфейс и дисплей
Питание	12V	5V
Обогреватель	Лампа накаливания 40Вт	Коврик нагревательный 15Вт
Контроль влажности	Вентилятор 60x60мм 12V	Вентилятор 40x40мм 5V

ПРОБЛЕМЫ

Не обошлось без проблем. С помощью тепловизора мы обнаружили значительный нагрев питающего провода более 70 °С, на фотографии он обозначен как бело-желтое пятно. Пришлось заменить питающий провод на провод большего сечения и поставить более мощный разъем.

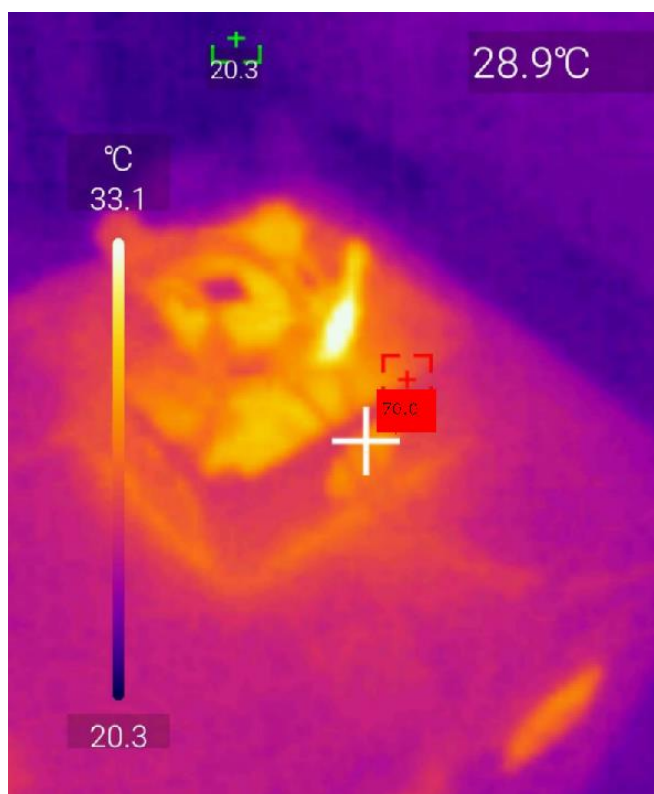


Рисунок 6. Фотография с тепловизора с показанием температуры перегрева

Еще одной проблемой стало то, что пенополистирол стал плавиться под греющими ковриками. Решением этой проблемы стал отрезок стеклоткани, который я вырезал и подложил под матику.



Рисунок 7. Места, где плавился пенополистирол

ТЕРМОРЕГУЛЯЦИЯ

Продолжением работы будет изучение процессов, происходящих внутри инкубатора. Я заметил, что нагрев не прекращается моментально после выключения нагревателя, температура продолжает расти ещё некоторое время. Мы произвели замеры и составили графики зависимости температуры нагрева и охлаждения от времени. В старой версии инкубатора амплитуда отклонения от нормированной температуры находится в диапазоне от 35,5 до 41,1 градуса. Мы улучшили терморегуляцию, но это тема отдельного исследования, которому я хочу посвятить следующую работу. Главный результат, что мы снизили амплитуду колебания температур с 5 градусов до 1.2 градуса, это должно повысить выводимость птенцов и я надеюсь следующая попытка вывести перепелов будет более удачной.

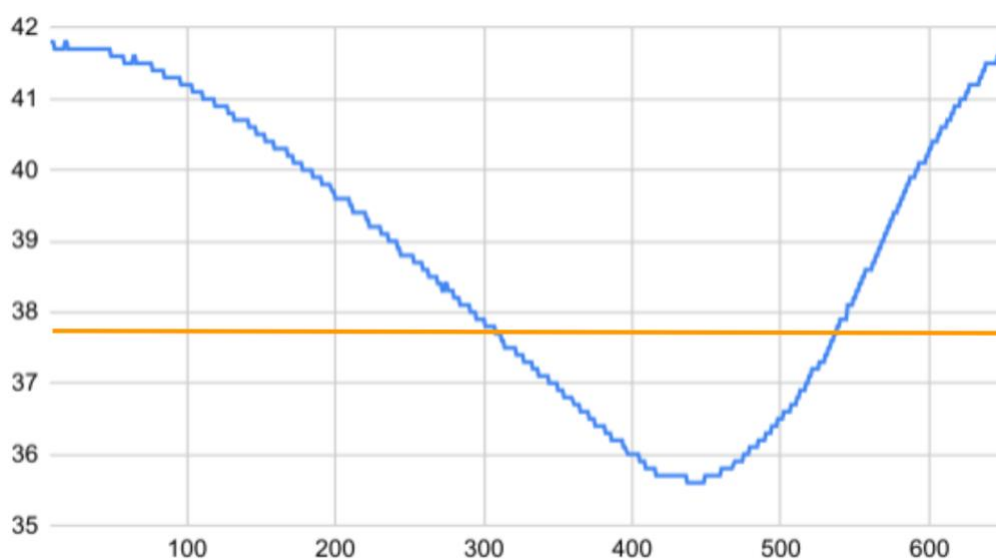


Рисунок 8. График температуры старой версии инкубатора 2.0

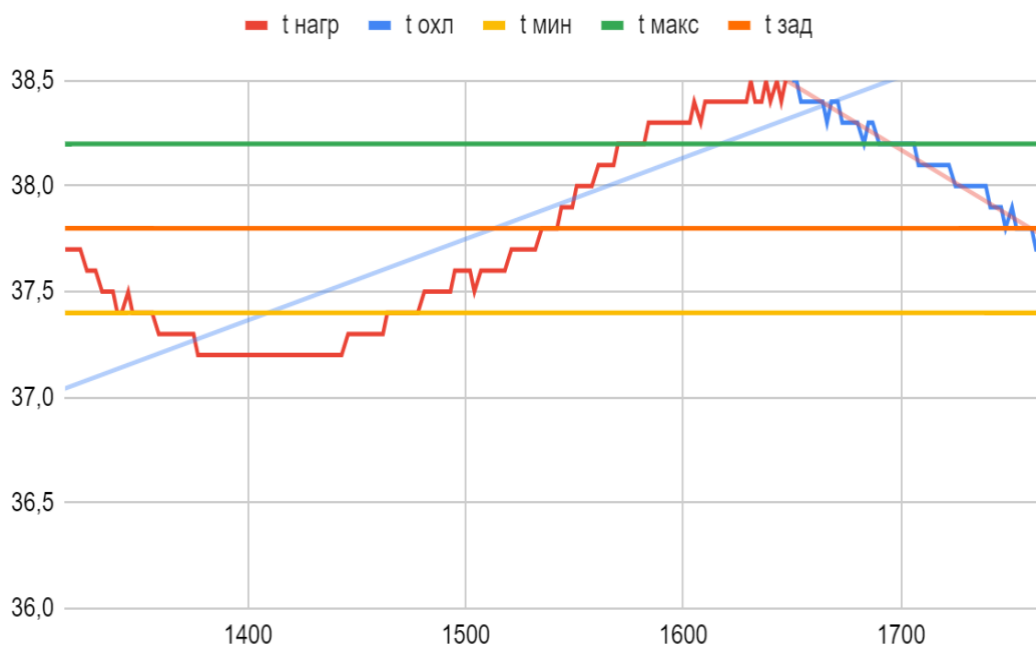


Рисунок 9. График температуры инкубатора версии 3.0

ВЫВОДЫ

Проделанная работа позволила повысить вместимость корпуса инкубатора, за счет использования пенополистирола повысить прочность корпуса. Надежность схемотехники была значительно повышена как за счет используемых элементов, так и за счет монтажа. Управление инкубатором стало значительно удобнее за счет использования Web-интерфейса.

Таким образом, все заявленные задачи были решены, и указанная цель достигнута.

ПЛАНЫ НА БУДУЩЕЕ

В моих ближайших планах:

- Сделать автоматический поворот яиц;
- Вырезать на лазере корпус из фанеры или даже попробовать сделать корпус обтекаемой формы;
- Создать красивые страницы для сайта;
- Реализовать трансляцию моментов выведения цыплят из скорлупы в социальные сети;
- Подключить для управления к платформе умного дома Home Assistant;
- Сделать уведомления через бот в Telegram;
- Опубликовать проект для публичного доступа (open source).

Я хочу, чтобы инкубатор стал устройством, которое будет не только простым в обслуживании и облегчит труд по выведению цыплят, но будет доступным и дешевым конструктором для каждого, модным и интересным гаджетом.

СТРУКТУРА КОМАНДНОЙ РАБОТЫ НАД ПРОЕКТОМ

Таблица 3. Структура и этапы работ

	Этапы работ	Иван	Другие участники
1	Проектирование корпуса	100%	0%
2	Сборка корпуса	100%	0%
3	Подбор элементной базы	0%	100%
4	Макетирование схемотехники	90%	10%
5	Разводка монтажной платы в Spring layout	30%	70%
6	Распайка компонентов монтажной платы	20%	80%
7	Разработка программной архитектуры	0%	100%
8	Реализация (кодирование) программного функционала на Python	60%	40%
9	Программирование страниц на HTML	80%	20%
1			
0	Отладка программы	100%	0%
1	Тестирование программы	100%	0%

1			
1 2	Монтаж датчика температуры и влажности и нагревателя	100%	0%
1 3	Комплексное испытание	100%	0%

В таблице я перечислил этапы и показал свое участие в них. Некоторые я завершил сам, а какие-то в команде, например, п.8. я программировал по разработанной архитектуре.

Но в результате работы над проектом вот мои использованные и вновь приобретенные навыки:

- Работа с AutoCAD - двухмерной системе проектирования и черчения;
- Работа в Spring layout 5 - программе для разводки печатных плат;
- Навык радиомонтажа и работы с паяльником;
- Навык программирования на языках Python и HTML.

СПИСОК ЛИТЕРАТУРЫ

1. Отрыганьев К.А. Технология инкубации / К. А. Орыганьев, В.М. Рошков - М.: Агропромиздаст. 2003 - 152с.
2. <https://map.kruzhok.org/catalog/2075>
3. Проектирование печатных плат в программе Sprint Layout 6 / М.Г. Царёв. – Ульяновск, 2016. – 97 с.
4. <https://micropython.org/>

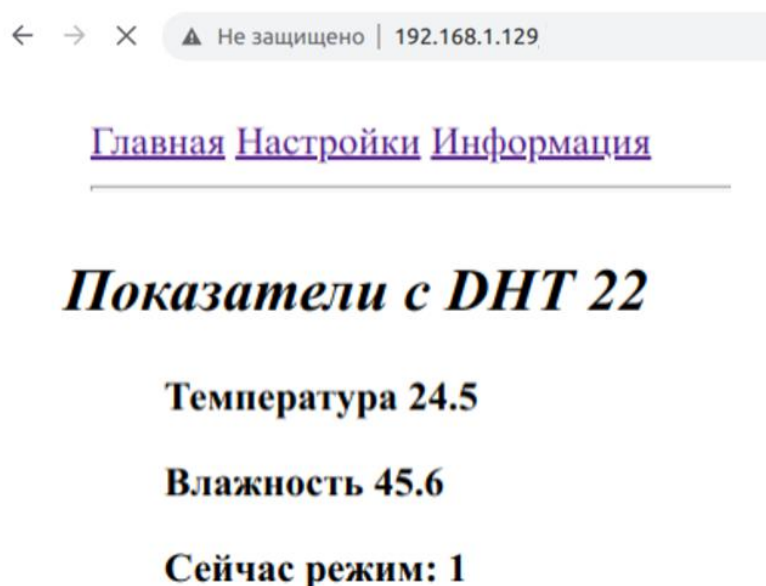


Рисунок 1. Главная страница web-интерфейса с меню и выводом текущих показателей

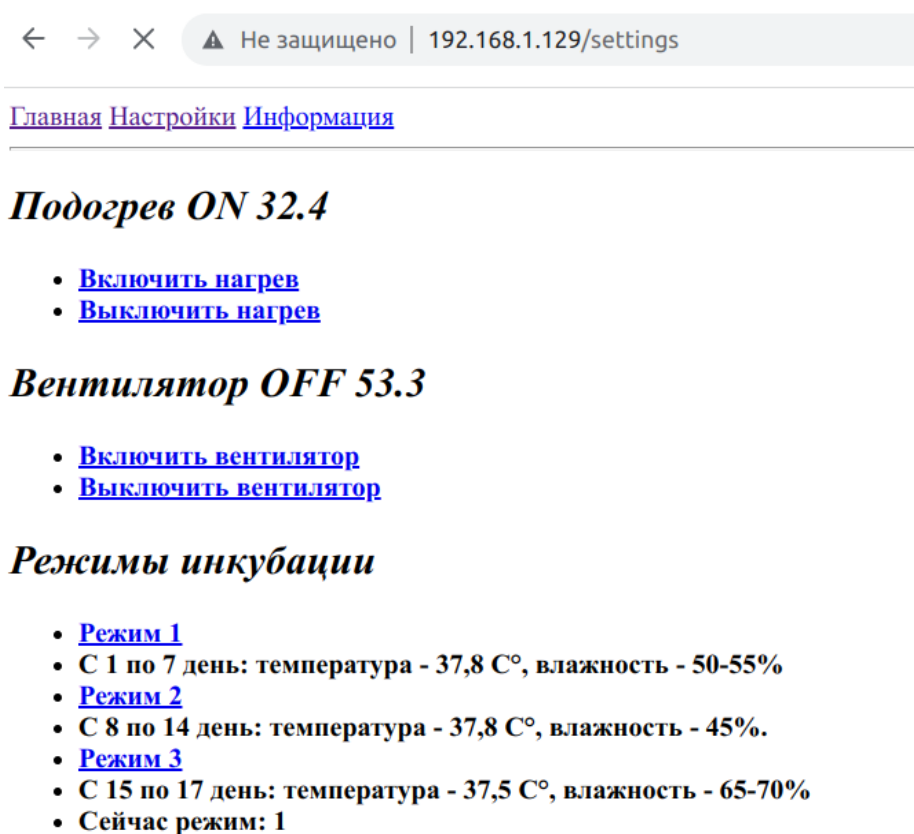


Рисунок 2. Вторая страница web-интерфейса – настройки

[Главная](#) [Настройки](#) [Информация](#)

- 13.11.2022.16.25
- Подключение к сети обязательно!

Рисунок 3. Третья страница web-интерфейса – информация

```

import uasyncio
import network
from machine import Pin
import machine
from time import sleep
import dht
from machine import Pin, I2C
import ssd1306
import json
import time

from nanoweb import Nanoweb

naw = Nanoweb()

def init():
    global mode, display
    display.text('Starting...', 0, 0, 1)
    try:
        file = open("config.json", "r")
        config = json.loads(file.read())
        print('config = ', config)
        mode = config['mode']
    except:
        mode = 1
    else:
        file.close()
    set_mode(mode)
    display.fill(0)

async def init_network():
    global ip_addr, mode

    wlan_id = "Ivans"
    wlan_pass = "parols01"

    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    wlan.disconnect()
    if not wlan.isconnected():
        wlan.connect(wlan_id, wlan_pass)
        print("Waiting for connection...")
        display.fill_rect(0, 0, 128, 11, 0)

```

```

display.text('Connecting...', 0, 0, 1)
while not wlan.isconnected():
    await uasyncio.sleep(1)

#wlan.disconnect()

ip_addr = wlan.ifconfig()[0]
print("Connected... IP: " + ip_addr)
display.fill_rect(0, 0, 128, 11, 0)

@naw.route("/")
def index(request):
    global temperature, humidity, mode

    sensor.measure()
    temperature = sensor.temperature()
    humidity = sensor.humidity()

    html = ""
    <!DOCTYPE HTML>
    <html>
        <head>
            <meta charset="utf-8">
            <meta http-equiv="refresh" content="2">
            <title>Инкубатор</title>
        </head>
        <body>
            <a href="/">Главная</a>
            <a href="/settings">Настройки</a>
            <a href="/inf">Информация</a>
            <hr>
            <h1>Только для перепелов!</h1>
            <hr>
            <h2><em>Показатели с DHT 22</em></h2>
            <ul>
                <li><b>Температура {temp} {heater_status}</b></li>
                <li><b>Влажность {hum} {fan_status}</b></li>
                <li><b>Сейчас режим: {mode}</b></li>
            </ul>
        </body>
    </html>
    """.format(temp = temperature,
              hum = humidity,
              fan_status = on_off_dict[fan.value()],
              heater_status = on_off_dict[heater.value()],

```

```

        mode = mode)
await request.write("HTTP/1.0 200 OK\r\n")
await request.write("Content-Type: text/html; charset=utf-8\r\n\r\n")
await request.write(html)

@naw.route("/settings")
def settings(request):
    on_off_dict = {1: "ON", 0: "OFF"}
    global mode

    html = ""
    <!DOCTYPE HTML>
<html>
<head>
    <meta charset="utf-8">
    <meta http-equiv="refresh" content="2">
    <title>Настройки (инкубатор)</title>
</head>
<body>
    <a href="/">Главная</a>
    <a href="/settings">Настройки</a>
    <a href="/inf">Информация</a>
    <hr>
    <h2><em>Подогрев {heater_status} {temp}</em></h2>
    <ul>
        <li><b><a href="/heater_on">Включить нагрев</a></b></li>
        <li><b><a href="/heater_off">Выключить нагрев</a></b></li>
    </ul>
    <h2><em>Вентилятор {fan_status} {hum}</em></h2>
    <ul>
        <li><b><a href="/fan_on">Включить вентилятор</a></b></li>
        <li><b><a href="/fan_off">Выключить вентилятор</a></b></li>
    </ul>
    <h2><em>Информация по выведению перепелов</em></h2>
    <ul>
        <li><b><a href="/inf1">Подробная информация</a> (подключение к сети
обязательно!)</b></li>
    </ul>
    <h2><em>Режимы инкубации</em></h2>
    <ul>
        <li><b><a href="/mode_1">Режим 1</a></b></li>
        <li><b>С 1 по 7 день: температура - 37,8 С°, влажность - 50-55%</b></li>
        <li><b><a href="/mode_2">Режим 2</a></b></li>
        <li><b>С 8 по 14 день: температура - 37,8 С°, влажность - 45%.</b></li>
        <li><b><a href="/mode_3">Режим 3</a></b></li>

```

```

        <li><b>С 15 по 17 день: температура - 37,5 С°, влажность - 65-70%</b></li>
        <li><b>Сейчас режим: 1</b></li>
    </ul>
</body>
</html>

```

```

"".format(
    temp = temperature,
    hum = humidity, fan_status = on_off_dict[fan.value()],
    heater_status = on_off_dict[heater.value()],
    mode = mode
)
await request.write("HTTP/1.0 200 OK\r\n")
await request.write("Content-Type: text/html; charset=utf-8\r\n\r\n")
await request.write(html)

```

```
@naw.route("/inf")
```

```
def inf(request):
```

```

html = ""
<!DOCTYPE HTML>
<html>
<head>
    <meta charset="utf-8">
    <meta http-equiv="refresh" content="2">
    <title>Информация (инкубатор)</title>
</head>
<body>
    <a href="/">Главная</a>
    <a href="/settings">Настройки</a>
    <a href="/inf">Информация</a>
    <hr>
    <ul>
        <li>28.11.2022.18.9</li>
        <li>Подключение к сети обязательно!</li>
    </ul>
</body>
</html>
""

```

```

await request.write("HTTP/1.0 200 OK\r\n")
await request.write("Content-Type: text/html; charset=utf-8\r\n\r\n")
await request.write(html)

```

```
@naw.route("/heater_on")
```

```

def heater_on(request):
    heater.value(1)
    await request.write("HTTP/1.0 302 OK\r\n")
    await request.write('Location: /settings\r\n')

@naw.route("/inf1")
def inf1(request):
    await request.write("HTTP/1.0 302 OK\r\n")
    await request.write('Location:
https://fermerznaet.com/pticevodstvo/perepela/inkubaciya.html#m5')

@naw.route("/heater_off")
def heater_off(request):
    heater.value(0)
    await request.write("HTTP/1.0 302 OK\r\n")
    await request.write('Location: /settings\r\n')

@naw.route("/fan_on")
def fan_on(request):
    fan.value(1)
    await request.write("HTTP/1.0 302 OK\r\n")
    await request.write('Location: /settings\r\n')

@naw.route("/fan_off")
def fan_off(request):
    fan.value(0)
    await request.write("HTTP/1.0 302 OK\r\n")
    await request.write('Location: /settings\r\n')

@naw.route("/mode_1")
def mode_1(request):
    set_mode(1)
    save_settings(mode)

    await request.write("HTTP/1.0 302 OK\r\n")
    await request.write('Location: /settings\r\n')

@naw.route("/mode_2")
def mode_2(request):
    set_mode(2)
    save_settings(mode)
    await request.write("HTTP/1.0 302 OK\r\n")
    await request.write('Location: /settings\r\n')

@naw.route("/mode_3")

```

```

def mode_3(request):
    set_mode(3)
    save_settings(mode)
    await request.write("HTTP/1.0 302 OK\r\n")
    await request.write('Location: /settings\r\n')

def set_mode(new_mode):
    global mode, target_temperature, target_humidity
    mode = new_mode
    if mode == 1:
        #C 1 по 7 день: температура - 37,8 C°, влажность - 50-55%.
        target_temperature = 37.8
        target_humidity = 53
    elif mode == 2:
        #C 8 по 14 день: температура - 37,8 C°, влажность - 45%.
        target_temperature = 37.8
        target_humidity = 45
    elif mode == 3:
        #C 15 по 17 день: температура - 37,5 C°, влажность - 65-70%.
        target_temperature = 37.5
        target_humidity = 67

def save_settings(mode):
    file = open("config.json", "w")
    file.write(json.dumps({'mode': mode}))
    file.close()

async def thermostat():
    global temperature, humidity, mode, ip_addr
    start_time = time.time()
    while True:
        sensor.measure()
        temperature = sensor.temperature()
        humidity = sensor.humidity()
        if temperature >= target_temperature:
            heater.value(0)
        elif temperature < target_temperature:
            heater.value(1)

        #print(0 - (start_time - time.time()), "\t", temperature)

        if humidity > target_humidity + delta_humidity:
            fan.value(1)
        elif humidity < target_humidity - delta_humidity:
            fan.value(0)

```

```

# Перегрев
#     if temperature > 39:
#         fan.value(1)
#         display.text('Overheat fan ON!!!', 0, 0, 1)
#     elif temperature < 38.5:
#         fan.value(0)
#         display.fill_rect(0, 0, 128, 11, 0)

#     if temperature < 38:
#         display.text('Undercooling!!!', 0, 0, 1)
#     elif temperature > 37.2:
#         display.fill_rect(0, 0, 128, 11, 0)

    led.value(not led.value())
    on_off_dict = {1: "ON", 0: "OFF"}
    display.fill_rect(0, 12, 128, 64, 0)
    display.text('Heat: {heat}, {temp}C'.format(heat = on_off_dict[heater.value()], temp =
temperature), 0, 16, 1)
    display.text('Fan: {fan}, {hum}%'.format(fan = on_off_dict[fan.value()], hum = humidity), 0,
28, 1)
    display.text('M{mode}: {temp}C, {hum}%'.format(mode = mode, temp = target_temperature,
hum = target_humidity), 0, 40, 1)
    display.text(ip_addr, 0, 52, 1)

    display.show()
    await uasyncio.sleep(2)

led = Pin(2, Pin.OUT)
sensor = dht.DHT22(Pin(14))
heater = Pin(13, Pin.OUT)
fan = Pin(12, Pin.OUT)
display = ssd1306.SSD1306_I2C(128, 64, I2C(sda=Pin(0), scl=Pin(2)))

mode = 1
target_temperature = 37.8
target_humidity = 53
delta_temperature = .3
delta_humidity = 3

humidity = 0
temperature = 0
ip_addr = ""
on_off_dict = {1: "ON", 0: "OFF"}

led.value(1)

```

```
#wait hardware init after hard reset  
#sleep(5)
```

```
led.value(0)
```

```
init()
```

```
loop = uasyncio.get_event_loop()  
loop.create_task(init_network())  
loop.create_task(naw.run())  
loop.create_task(thermostat())  
loop.run_forever()
```

РЕЦЕНЗИЯ

на проектную работу «Высший пилотаж - Пенза» 2025

Семенкина Ивана Максимовича
(фамилия, имя и отчество)

Тема проекта: «Инкубатор версия 3.0. На пути к промышленному образцу»

В работе Семенкина И.М. была выполнена модернизация и разработка нового инкубатора версии 3.0. Модернизация этого устройства позволяет не только снизить производительность за счет новой элементной базы, но и повысить надежность в качестве механических и тепловых характеристик. В связи с этим тема является актуальной.

В работе систематизировано большое количество справочного материала из различных источников. Достоинствами проектной работы является проведение теоретического и практического анализа инкубатора 2.0 и внесение изменений и корректировок в инкубатор 3.0. Материал в проектной работе изложен с соблюдением внутренней логики, между разделами прослеживается логическая взаимосвязь. В работе присутствуют ссылки на используемые источники.

Проектная работа полностью отвечает используемыми подходами и обоснованными решениями, раскрытых с необходимой полнотой. Графическая часть выполнена в достаточном объеме, работа раскрывает содержание проекта.

В работе решены все поставленные задачи. Семенкин И.М. показал умение пользоваться научно-технической литературой, решать инженерные и научные задачи, способность использовать имеющиеся знания при выполнении данного проекта. Работа выполнена качественно.

Опираясь на вышесказанное, считаю, что проектная работа заслуживает высокой и максимальной оценки жюри

Рецензент:
Руководитель проекта,
учитель информатики Губернского лицея



А.И. Гуляйкина