Муниципальный форум

Для обучающихся 5-11 классов образовательных учреждений ЗАТО

Железногорск в 2019 году

**Система «Умный аквариум»**

Направление: Техносалон  
«Информационные системы и технологии

в науке, технике и образовании»

Проектно-исследовательская работа

***Авторы:***

Савин Савелий Евгеньевич,

МБОУ Гимназия №91, 8 класс

evgen0805@yandex.ru

Грибанов Елисей Борисович

МБОУ СОШ № 97,7 класс

***Руководители:***

Сиротинина И.В.,

учитель информатики

irina-sirotinina@yandex.ru

Одинцова Т.А.,

учитель проектной деятельности

МБОУ Гимназия №91

***Консультанты:***

Кириллов К.Ю.,

Филиппов М.С.,

Святка И.В.

Железногорск – 2019

Тезисы

Актуальность. Возможность создания на некоторое время микросреды для рыб без участия человека актуально для многих аквариумистов. Система «Умный аквариум» позволяет в определенное время суток проверять температуру воды, включать и выключать свет, кормить аквариумных рыбок. На сегодняшний день существуют множество автоматизированных систем, позволяющих поддерживать климатические условия в аквариуме, но их стоимость достаточно высока и, в большинстве своем, они содержат избыточный функционал.

Цель: Создать автоматическую систему «Умный аквариум» для поддержания климатических условий аквариума.

Задачи:

1. Проанализировать существующие решения.
2. Выбрать микроконтроллер для реализации проекта. Определить входные и выходные характеристики системы.
3. Изучить особенности построения микроконтроллера на базе плат Arduino и их модулей, основы микросхемотехники.
4. Разработать алгоритм работы системы «Умный аквариум».
5. Изучить особенности среды программирования на базе платформы Arduino.
6. Разработать схему подключения датчиков, модулей.
7. Собрать блок управления системы «Умный аквариум».
8. Написать програму управелния системой.
9. Создать 3D- модель для корпуса (блока управления и кормушки) системы «Умный аквариум» и распечатать на 3D принтере.
10. Протестировать работу системы «Умный аквариум». Исправить неполадки в работе.

Объект исследования: Поддержка микроклимата аквариума без участия человека

Предмет исследования: автоматичсекя система «Умный аквариум».

Методы: анализ собранной информации; структурно-функциональный метод; информационное и предметное моделирование; наблюдение; проверочные и воспроизводящие эксперименты.

Вывод. Разработана и собрана система «Умный аквариум» на базе микроконтроллера Arduino UNO.Разработан алгоритм работы. Написана программа, согласно которой выполняются следующие действия:- включение и выключение света; кормление рыб; измерение температуры воды; вывод информации на дисплей;созданы и распечатаны 3D модели корпуса блока управления и кормушки.

1. Обзор готовых систем «Умный аквариум» <URL:http://www.aquacoral.ru/aquacoral/price-aqua-na-zakaz.html>
2. Обзор аналогов Arduino URL: <https://arduinoplus.ru/vse-analogi-arduino/#___Netduino>

[Введение…………………………………………………………………………………………………. 3](#_Toc771673)

[Основная часть 4](#_Toc771674)

[1 этап. Проведение анализа существующих систем «Умный аквариум». 4](#_Toc771675)

[2 этап. Выбор микроконтроллера. 4](#_Toc771676)

[3 этап. Определение входных характеристик. 5](#_Toc771677)

[4 этап. Определение выходных характеристик. 6](#_Toc771678)

[5 этап. Разработка алгоритма работы системы 9](#_Toc771679)

[6 этап. Создание схемы и заказ деталей. 9](#_Toc771680)

[7 этап. Сборка блока управления. 10](#_Toc771681)

[8 этап. Разработка. 11](#_Toc771682)

[9 этап. Печать 3D моделей и их сборка. 12](#_Toc771683)

[10 этап. Тестирование и устранение неисправностей. 14](#_Toc771684)

[Заключение 15](#_Toc771685)

[Список литературы 16](#_Toc771686)

[Приложение 1.Характеристики модулей. 17](#_Toc771687)

[Приложение 2. Блок-схема алгоритма. 19](#_Toc771688)

[Приложение 3. Листинг программы. 24](#_Toc771689)

# Введение

Возможность создания на некоторое время микросреды для рыб без участия человека актуально для многих аквариумистов.

Система «Умный аквариум» позволяет в определенное время суток проверять температуру воды, включать и выключать свет, кормить аквариумных рыбок и т.д. На сегодняшний день существуют множество автоматизированных систем, позволяющих автономно или с участием человека поддерживать климатические условия для жизни рыб в аквариуме, но их стоимость достаточно высока и, в большинстве своем, они содержат избыточный функционал для обычного любителя-аквариумиста. Аквариумистика - очень популярное хобби. Она занимает 4 место в России. Поэтому существует необходимость создания бюджетной системы «Умный аквариум» с определенными характеристиками, нужными конкретному владельцу аквариума.

Цель: Создать автоматическую систему «Умный аквариум» для поддержания климатических условий аквариума.

Задачи:

1. Проанализировать существующие решения.
2. Выбрать микроконтроллер для реализации проекта.
3. Определить входные и выходные характеристики системы.
4. Изучить особенности построения микроконтроллера на базе плат Arduino и их модулей.
5. Изучить основы микросхемотехники.
6. Разработать алгоритм работы системы «Умный аквариум».
7. Изучить особенности среды программирования на базе платформы Arduino.
8. Разработать схему подключения датчиков, модулей.
9. Собрать блок управления системы «Умный аквариум».
10. Написать програму управелния системой.
11. Создать 3D- модель для корпуса (блока управления и кормушки) системы «Умный аквариум»;
12. Распечатать корпус на 3D принтере.
13. Протестировать работу системы «Умный аквариум».
14. Исправить возникшие неполадки в работе системы.

Объект исследования: Поддержка микроклимата аквариума без участия человека

Предмет исследования: автоматичсекя система «Умный аквариум».

Методы*:*

1. Анализ собранной информации;
2. Структурно-функциональный метод:
   1. изучение строения, структуры системного объекта;
   2. исследование его элементов и их функциональных характеристик;
   3. анализ изменения этих элементов и их функций;
3. Информационное и предметное моделирование;
4. Наблюдение;
5. Проверочные и воспроизводящие эксперименты.

# Основная часть

### 1 этап. Проведение анализа существующих систем «Умный аквариум».

Система «Умный аквариум» - система его жизнеобеспечения. Благодаря которой, мы можем оставить своих питомцев под присмотром умной автоматики. На сегодня существует множество автоматических систем. Можно выбирать степень автоматизации и количества анализируемых/управляемых параметров. Умный аквариум - это своего рода система [умный дом](http://www.aquacoral.ru/aquacoral/smart-house.html).

Наиболее известные изготовители: «AquaDigitalLife», Bluenero, Tetra, Sera. [1],[2]

Но стоимость автоматических систем этих фирм очень дорога. Очень много функций систем не нужны для всех аквариумов, так как разные условия содержания аквариумных рыб.

Систему «Умный аквариум» в интернете есть варианты реализации, мы решили создать свой.

### 2 этап. Выбор микроконтроллера.

После изучения материала по данной теме для основы системы «Умный аквариум» была выбрана плата «Arduino UNO» (рис.3).

В тоже время в интернете существует много схем создания системы «Умный аквариум» на базе микроконтроллеров [3]:

* [Netduino](https://arduinoplus.ru/vse-analogi-arduino/#___Netduino)
* [Ti MSP430 LaunchPad](https://arduinoplus.ru/vse-analogi-arduino/#Ti_MSP430_LaunchPad)
* [Teensy](https://arduinoplus.ru/vse-analogi-arduino/#Teensy)
* [Particle Photon](https://arduinoplus.ru/vse-analogi-arduino/#Particle_Photon)



Рис.3.Arduino UNO

Критерии выбора:

* 1. Приемлемая стоимость.
  2. Доступность компонентов.
  3. Простота настройки программной части проекта.
  4. Множество библиотек для управления типовыми устройствами.

Разработка приложений на базе плат Arduino осуществляется в специальной среде программирования Arduino IDE. Среда предназначена для написания и загрузки собственных программ в память микроконтроллера. Среда разработки Arduino состоит из редактора программного кода, области сообщений, окна вывода текста, панели инструментов и панели меню. Базовая структура программы для Arduino состоит из двух обязательных частей: функций setup() и loop(). Перед функцией setup() идет объявление переменных, подключение вспомогательных библиотек. Функция setup() запускается один раз после каждого включения питания или сброса платы. Она используется для инициализации переменных, установки режима работы портов и т. д. Функция loop() в бесконечном цикле последовательно исполняет описанные команды. [4]

### 3 этап. Определение входных характеристик.

4% россиян отдали предпочтение аквариумным рыбкам. Они крайне неприхотливы и занимают минимум вашего времени. Они выполняют больше декоративные функции, а также являются отличным способом расслабиться и успокоится после тяжелого дня. Оборудовав аквариум фильтром для воды, правильно подобрав соседство аквариумным жителям и покупая им корм, вы не будете испытывать почти никаких хлопот с вашими плавающими друзьями. [5]

Параметры микроклимата особенно важны для жизнедеятельности обитателей аквариума. Для определения входных характеристик необходимо определить климатические параметры среды аквариума. В данной работе мы определили два климатических критерия: освещенность, то есть включение и выключение света в аквариуме в определенное программой время и температура воды.

Для создания заданных параметров нам необходимы следующие датчики. Датчик измерения температуры, часы реального времени, двойное реле (необходимо для включения и выключения света). Все модули должны находиться в распределительной коробке, для этого мы измерили габариты аквариума Rio 180 LED и толщину стенок:

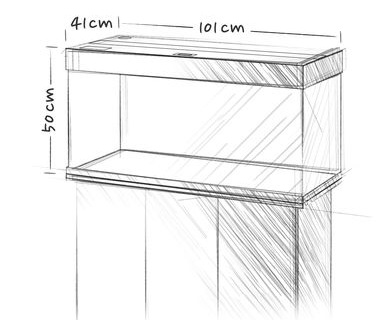
****

Рис.1 Габариты аквариума

|  |  |
| --- | --- |
| **Толщина боковых стенок** | **1,3 см** |
| **Толщина передней стенки** | **1,4 см** |

Данные характеристики необходимы для создания 3D модели корпуса и кормушки.

### 4 этап. Определение выходных характеристик.

Выходными параметрами в данном проекте являются: выбор способа визуального контроля системы времени и температуры. Для этого нам нужен дисплей, отражающий измеряемые параметры (температуру воды и реальное время), а так же сервопривод, участвующий в процессе кормления.

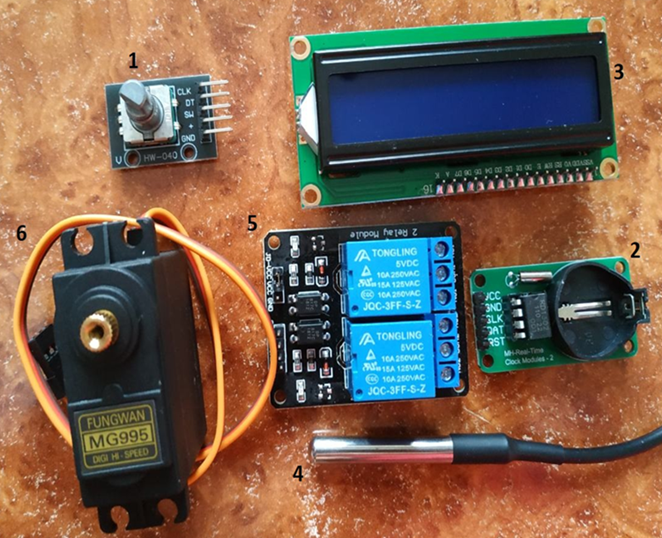


Рис. 2 Модули системы «Умный аквариум»

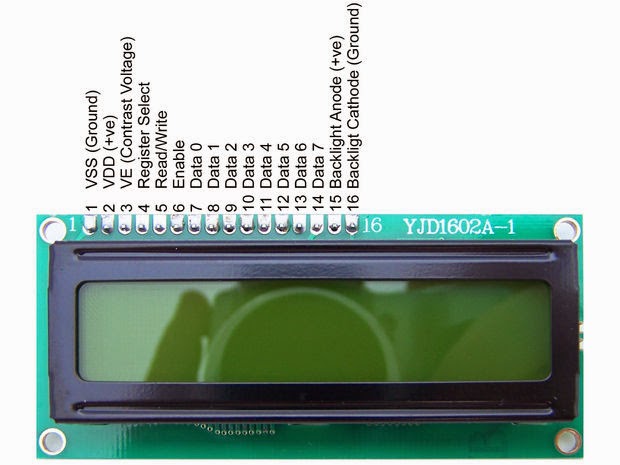
1. *Энкодер* – используется для изменения реального времени;
2. *Часы реального времени* – используется для определения реального времени;
3. *Дисплей* – отражает измеряемые параметры;
4. *Датчик измерения температуры*;
5. *Двойное реле* – необходимо для включения/выключения света [10];
6. *Сервопривод* – участвует в процессе кормления.

Модуль **энкодер** HW-040 является механическим датчиком угла поворота, он преобразует угол поворота вращающегося объекта (например вала) в электрические сигналы сдвинутые на 90 градусов относительно друг друга. Данный модуль имеет три вывода - CLK, DT и SW. Сигналы, сдвинутые на 90 градусов относительно друг друга появляются именно на выводах CLK и DT при вращении по/против часовой стрелки, вывод SW используется для получения состояния центральной оси энкодера, которая работает как кнопка.

Контроллер Arduino не имеет своих собственных часов. Поэтому для реализации нашего проекта необходимы **часы реального времени** DS 1302. [7],[ 8]

**Жидкокристаллический дисплей** (Liquid Crystal Display) **LCD 1602** является хорошим выбором для вывода строк символов в различных проектах. Он стоит недорого, есть различные модификации с разными цветами подсветки. Самым главным недостатком этого экрана является тот факт, что дисплей имеет 16 цифровых выводов, из которых обязательными являются минимум 6. Поэтому использование этого LCD экрана без i2c добавляет серьезные ограничения для плат Arduino Uno или Nano. [6]

Краткое описание пинов LCD 1602:

[](https://arduinomaster.ru/wp-content/uploads/2017/11/patillaje-LCD-1602.jpg)

Каждый из выводов имеет свое назначение:

1. Земля GND;
2. Питание 5 В;
3. Установка контрастности монитора;
4. Команда, данные;
5. Записывание и чтение данных;
6. Enable;

7-14. Линии данных;

1. Плюс подсветки;
2. Минус подсветки.

**Датчик температуры** в Arduino — один из самых распространенных видов сенсоров. Разработчику проектов с термометрами на Arduino доступно множество разных вариантов, отличающихся по принципу действия, точности, конструктивному исполнению. Цифровой датчик DS18B20 является одним из наиболее популярных температурных датчиков, часто он используется в водонепроницаемом корпусе для измерения температуры воды или других жидкостей. DS18B20 – это цифровой температурный датчик, обладающий множеством полезных функций. По сути, DS18B20 — это целый микроконтроллер, который может хранить значение измерений, сигнализировать о выходе температуры за установленные границы (сами границы мы можем устанавливать и менять), менять точность измерений, способ взаимодействия с контроллером и многое другое. Все это в очень небольшом корпусе, который, к тому же, доступен в водонепроницаемом исполнении. [9]

**Серводвигатель** MG995 поставляется с проводом длиной 30 см и 3-мя ‘S’ контактами типа мама.

Выходной вал сервопривода поворачивается приблизительно на 120 градусов (60 градусов в каждом направлении). Для управления сервами MG995 можно использовать любые контроллеры с питанием логики 5 В, в том числе и Arduino.

Сервомашинка изготавливается в пластиковом корпусе. На выходе стоит редуктор с металлическими шестернями. В комплекте поставляются пластиковые качалки различных форм-факторов.

### 5 этап. Разработка алгоритма работы системы

Рис.4 Алгоритм работы системы «Умный аквариум»

### 6 этап. Создание схемы и заказ деталей.

Для выполнения 5 этапа заказаны датчики и плата на общую сумму 2400 рублей:

1. Arduino Uno;
2. RTC 1302 часы реального времени;
3. DS1820 водонепроницаемый датчик температуры;
4. Двойное реле;
5. ЖК дисплей 1602;
6. Енкодер;
7. Сервопривод MG995;
8. Пластик для 3D модели;
9. Расходный материал (клей для 3D модели, резистор, соединительные провода).

Состав платформы и характеристики модулей указаны в Приложении 1.

Составлен общий алгоритм работы системы «Умный аквариум» (рис.4).

### 7 этап. Сборка блока управления.

На рисунке 5 представлена схема подключения датчиков к микроконтроллеру.

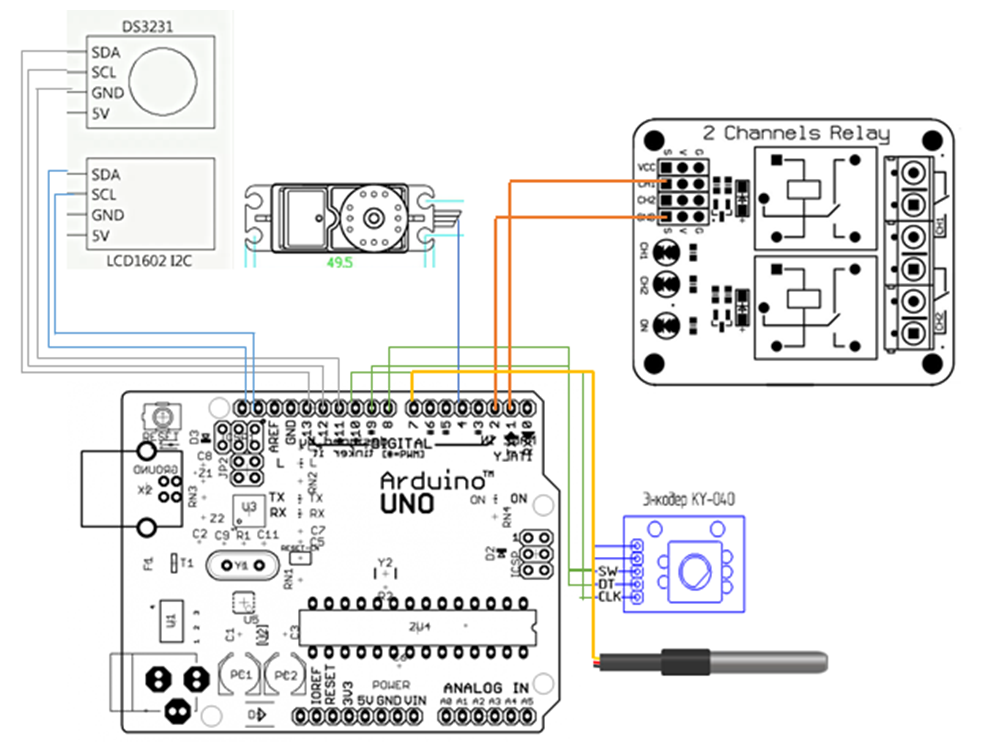


Рис. 5 Схема подключения датчиков к микроконтроллеру

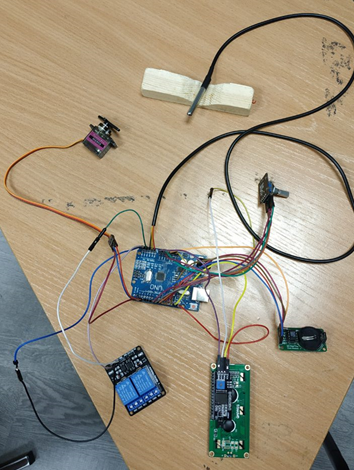


Рис. 6 подключение модулей к микроконтроллеру

На этапе 7 выполнения проекта осуществлено подключение всех датчиков к микроконтроллеру (рис.6).

### 8 этап. Разработка программы.

Интерфейс среды разработки Arduino содержит следующие основные элементы: текстовый редактор для написания кода, область для вывода сообщений, текстовая консоль, панель инструментов с традиционными кнопками и главное меню. Данный софт позволяет компьютеру взаимодействовать с Arduino как для передачи данных, так и для прошивки кода в контроллер.

Написание программ

Программы, создаваемые в среде разработки Arduino, иногда еще называют скетчами. Скетчи пишутся в текстовом редакторе и сохраняются в файлах с расширением .ino. Встроенный текстовый редактор имеет стандартные инструменты копирования, вставки, поиска и замены текста. Область сообщений в окне программы является, своего рода, обратной связью для пользователя и информирует его о событиях (в том числе и об ошибках), возникающих в процессе записи или экспорта написанного кода. Консоль отображает в виде текста поток выходных данных среды Arduino, включая все сообщения об ошибках и пр. генерируемую ею информацию. В нижнем правом углу окна программы показывается модель текущей платы и последовательный порт, к которому она подключена. Кнопки на панели инструментов предназначены для создания, открытия, сохранения и прошивки программ в устройство. Отдельная кнопка запускает программу SerialMonitor.

Для работы модулей в системе «Умный аквариум» составлены следующие алгоритмы.

***Алгоритм работы сервопривода:***

1. Если время кормления, то

2. Сервопривод двигается на 135 градусов и подаётся корм;

3. Сервопривод возвращается на 135 градусов в начальное положение.

Листинг программа работы сервопривода см. Приложение 3.

***Алгоритм работы лампы освещения:***

1.Если время включения, то

2.Включение лампы;

3.Если время выключения, то

4.Выключение лампы;

5.Если температура ниже заданного уровня, то

6.Выключение.

Листинг программа работы лампы освещения см. Приложение 3.

***Алгоритм работы датчика температуры:***

1.Считывание температуры;

2.Вывод температуры на экран.

Листинг программа работы датчика температуры см. Приложение 3.

***Алгоритм настройки времени:***

1.Долгое нажатие.

2.Начинается настройка реального времени.

3.1 нажатие выставляются часы, 2 нажатия выставляются минуты, 3 нажатия выставляются секунды.

4.Вывод на дисплей реального времени.

Листинг программа настройки времени см. Приложение 3.

***Общий алгоритм см. Приложение 2***.

### 9 этап. Печать 3D моделей и их сборка.

В ходе разработки проекта нам стало известно, что нужных по размеру деталей найти невозможно поэтом мы решили использовать технологию 3D печати.

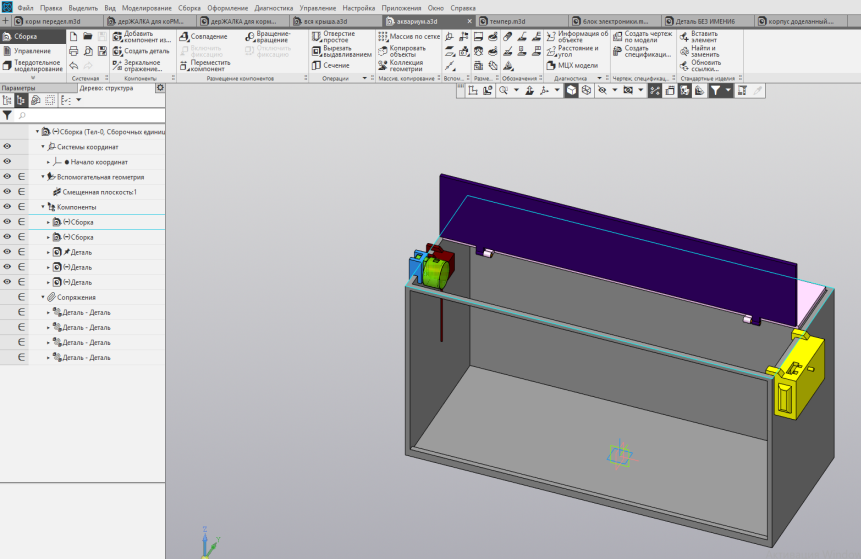
Почему Компас 3D?

Во-первых, эта программа адаптирована под ПК.

Во-вторых, имеет несложный и понятный интерфейс.

В-третьих, 3D-моделист нашей команды имеет опыт работы в этой программе.

Само создание 3D модели довольно долгий и трудоёмкий процесс, поэтому на создание был выделен большой отрезок времени.



Желтым показан блок управления системой, зеленым кормушка для рыб, красный – датчик температуры.

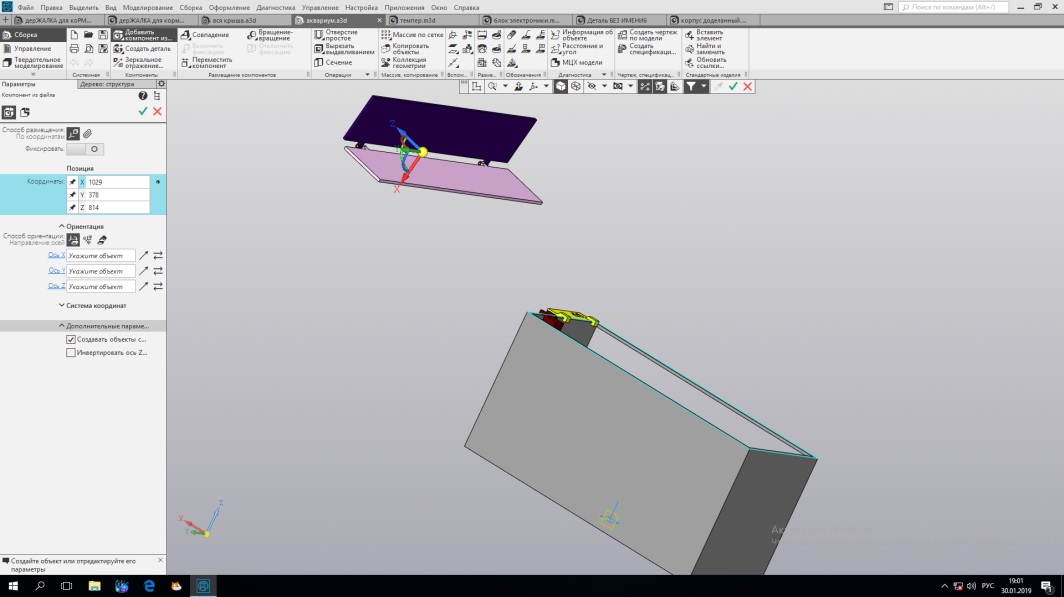


Рис. 7 Создание эскизов деталей в программе Компас 3D

На рисунке 7 конечный результат модели.

Следующим этапом было поэтапная отрисовка, т.е. описание каждой детали:

1. Основная деталь − сам аквариум. На него крепятся все основные детали: блок управления, кормушка, датчик температуры.
2. Блок управления. Необходимо было указать точные размеры, чтобы разъёмы под микроконтроллер подошли.
3. Изучив аналоги кормушек мы выбрали более оптимальный вариант для реализации.
4. Датчик температуры. Его отрисовка заняла меньше всего времени, так как он представлял собой простой параллелепипед с выпиравшем длинным цилиндром.

5 Верхняя крышка не несла ни какого функционала, но была отрисована для полноты модели.

Следующим этапом был выбор материала.

Для печати был выбран пластик PLA, так как PLA-пластик – является биоразлагаемым, биосовместимым и не несёт никакой опасности для рыбок, так как ПЛА-пластик производят из кукурузы или сахарного тростника. Сырьем для получения служат также картофельный и кукурузный крахмал, соевый белок, целлюлоза. [11]

Последний этап - подготовка деталей к печати.

1. Чтобы не пришлось перепечатывать детали, мы убедились, что все размеры введены правильно;
2. Создали gcode файл (Gcode это формат, с которым работаю 3D принтеры). При создание gcode были применены такие настройки: печать пластик pla, печать на принтере геркулес, 15% заполнение, 2 периметра, 8 милиметров юбка (для усиления адгезии (адгезия - сцепление [поверхностей](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B2%D0%B5%D1%80%D1%85%D0%BD%D0%BE%D1%81%D1%82%D1%8C) разнородных твёрдых и/или жидких тел.).
3. Печать. В результате получились 2 детали – корпус системы и кормушка (рис. 8).

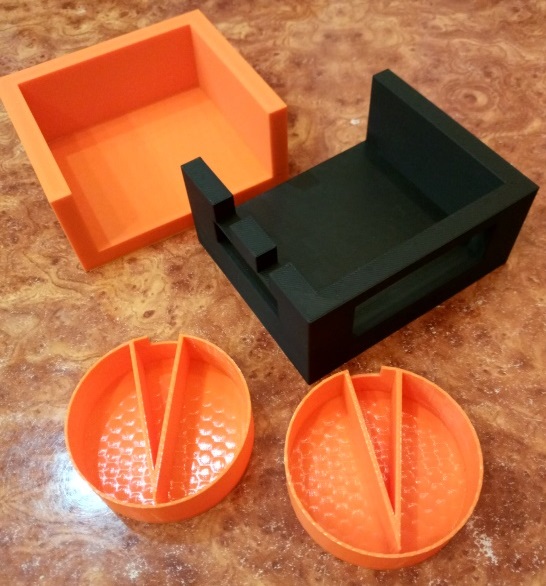


Рис. 8 Корпус системы управления и кормушка для рыб.

### 10 этап. Тестирование и устранение неисправностей.

В ходе подключения и тестирования системы обнаружилось:

1. При программировании блока изменения времени было найдено много ошибок – устранено с помощью руководителя проекта;
2. Для использования датчика температуры необходим резистор 4.7КОм – устранено;
3. После печати 3D модели обнаружены ошибки в размере – устранено.

### Заключение

Разработана и собрана система «Умный аквариум» на базе микроконтроллера Arduino UNO (рис. 9).

Разработан алгоритм работы.

Написана программа, согласно которой выполняются следующие действия:

- включение и выключение света;

- кормление рыб;

- измерение температуры воды;

- вывод информации на дисплей;

- созданы и распечатаны 3D модели корпуса блока управления и кормушки.

В ходе выполнения проекта появились новые идеи о возможности задавать время включения и выключения света и кормления, а также вывод информации на мобильные устройства.

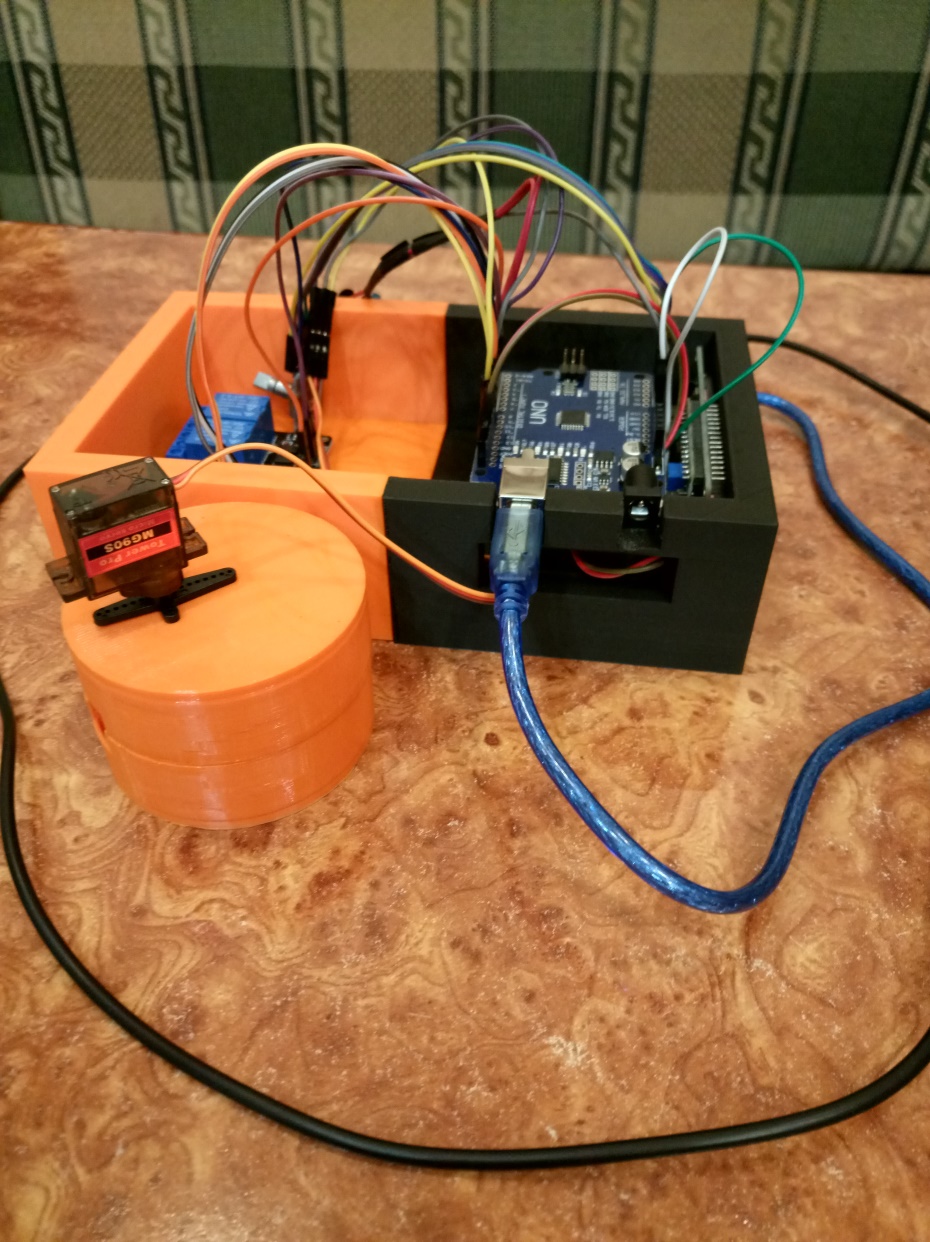


Рис. 9 Система «Умный аквариум».

### Список литературы

1. Обзор готовых систем «Умный аквариум» URL: <https://www.aquatrace.ru/category/kat00000000206/>, https://market.yandex.ru/catalog--akvariumy/56245/list?glfilter=7893318%3A10744795&onstock=1&local-offers-first=0 (Дата обращения 20.01.2019г.)
2. Обзор готовых систем «Умный аквариум» <URL:http://www.aquacoral.ru/aquacoral/price-aqua-na-zakaz.html> (дата обращения 20.01.2019г.)
3. Обзор аналогов Arduino URL: <https://arduinoplus.ru/vse-analogi-arduino/#___Netduino> (Дата 20.01.2019г.)
4. Характеристика платы Arduino Uno URL: <http://arduino.ru/Hardware/ArduinoBoardUno> (Дата 20.01.2019г.)
5. Опрос Зоогид URL: <http://www.zoogid.com/article/top-samyh-populyarnyh-domashnih-zhivotnyh> (Дата 30.01.2019г.)
6. Описание датчиков URL: <https://arduinomaster.ru/datchiki-arduino/lcd-i2c-arduino-displey-ekran/> (дата обращения 02.02.2019)
7. Описание датчиков URL: <http://kip-world.ru/podklyuchenie-chasov-realnogo-vremeni-ds1302-k-arduino.html>(Дата 30.01.2019г.)
8. Описание датчиков URL: <https://arduinomaster.ru/datchiki-arduino/arduino-chasy-rtc-ds1307-ds1302-ds3231/>(Дата 30.01.2019г.)
9. Описание датчиков URL: <https://arduinomaster.ru/datchiki-arduino/arduino-ds18b20/>(Дата 30.01.2019г.)
10. Описание датчиков URL: http://robotchip.ru/obzor-modulya-rele-2-x-kanalny/(Дата 30.01.2019г.)
11. Описание пластика URL: <https://3dtoday.ru/wiki/PLA_plastic/> (Дата обращения 08.02.2019)

# Приложение 1.Характеристики модулей.

**Краткие характеристики Arduino Uno**

| **Микроконтроллер** | **ATmega328** |
| --- | --- |
| Рабочее напряжение (логическая уровень) | 5 В |
| Входное напряжение (рекомендуемое) | 7-12 В |
| Входное напряжение (предельное) | 6-20 В |
| Цифровые Входы/Выходы | 14 (6 из которых могут использоваться как выходы ШИМ) |
| Аналоговые входы | 6 |
| Постоянный ток через вход/выход | 40 мА |
| Постоянный ток для вывода 3.3 В | 50 мА |
| Флеш-память | 32 Кб (ATmega328) из которых 0.5 Кб используются для загрузчика |
| ОЗУ | 2 Кб (ATmega328) |
| EEPROM | 1 Кб (ATmega328) |
| Тактовая частота | 16 МГц |
| Размеры | 1.85 см x 4.2 см |

**Технические характеристики сервопривода MG995**

* Масса: 55 грамм;
* Размеры: примерно 40.7 х 19.7 х 42.9;
* Крутящий момент: 8.5 кг х см (при 4.8 В питания), 10 кг х см (при 6 В);
* Скорость: 0.2 с/60º (при 4.8 В), 0.16 с/60º (при 6 В);
* Рабочее питание: 4.8 - 7.2 В;
* Ширина мертвой зоны: 5 мкс;
* Диапазон рабочих температур: 0 ºC – 55 ºC.

**Характеристика ЖК дисплея Arduino LCD 1602**

* Напряжение питания 5В;
* Формат 16х2 символов;
* Диапазон рабочих температур от -20С до +70С, диапазон температур хранения от -30С до +80 С;
* Угол обзора 180 градусов.

**Технические характеристики дисплея LCD 1602:**

* Символьный тип отображения, есть возможность загрузки символов;
* Светодиодная подсветка;
* Контроллер HD44780;
* Напряжение питания 5В;
* Формат 16х2 символов;
* Диапазон рабочих температур от -20С до +70С, диапазон температур хранения от -30С до +80 С;
* Угол обзора 180 градусов.

**Технические характеристики датчика времени DS 1302**

* Частота - 32.768 кГц;
* Точность – 5 сек в сутки;
* Поддерживаемые протоколы - I2C, SPI
* Питание – 5В;
* Диапазон рабочих температур от -40С до 85С;
* 56 байт памяти;
* Литиевая батарейка LIR2032;
* Реализует 12-ти и 24-х часовые режимы;
* Поддержка интерфейса I2C.

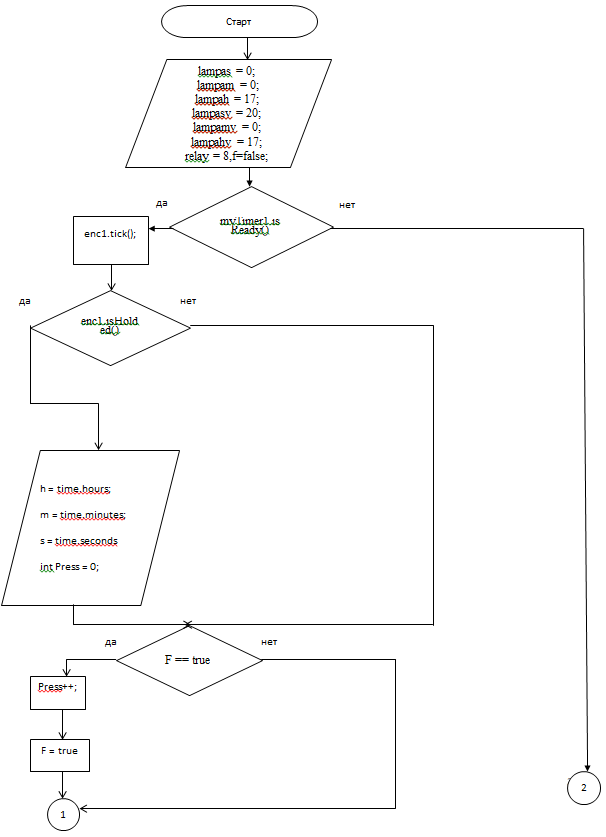
**Технические характеристики и особенности цифрового датчика температуры DS18b20**

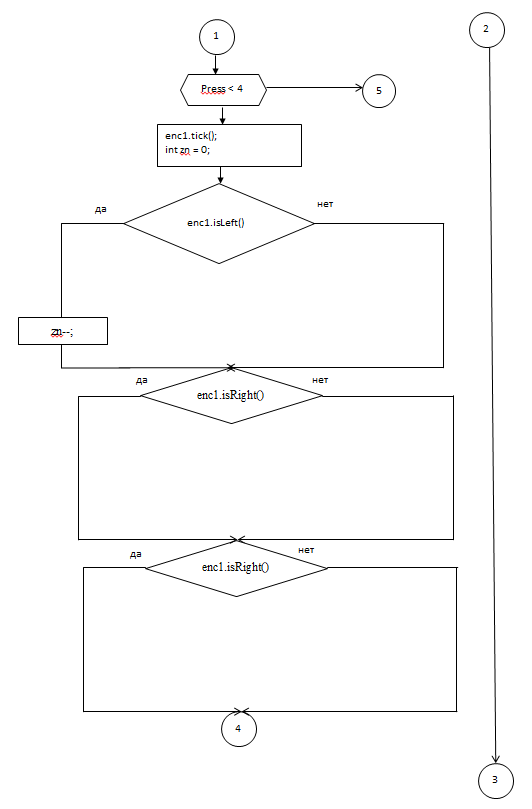
* Погрешность измерения не больше 0,5 С (для температур от -10С до +85С), что позволяет точно определить значение температуры. Не требуется дополнительная калибровка.
* Температурный диапазон измерений лежит в пределах от -55 С до +125 С.
* Датчик питается напряжением от 3,3В до 5В.
* Можно программно задать максимальную разрешающую способность до 0,0625С, наибольшее разрешение 12 бит.
* Присутствует функция тревожного сигнала.
* Каждое устройство обладает своим уникальным серийным кодом.
* Не требуются дополнительные внешние элементы.
* Можно подключить сразу до 127 датчиков к одной линии связи.
* Информация передается по протоколу 1-Wire.
* Для присоединения к микроконтроллеру нужны только 3 провода.

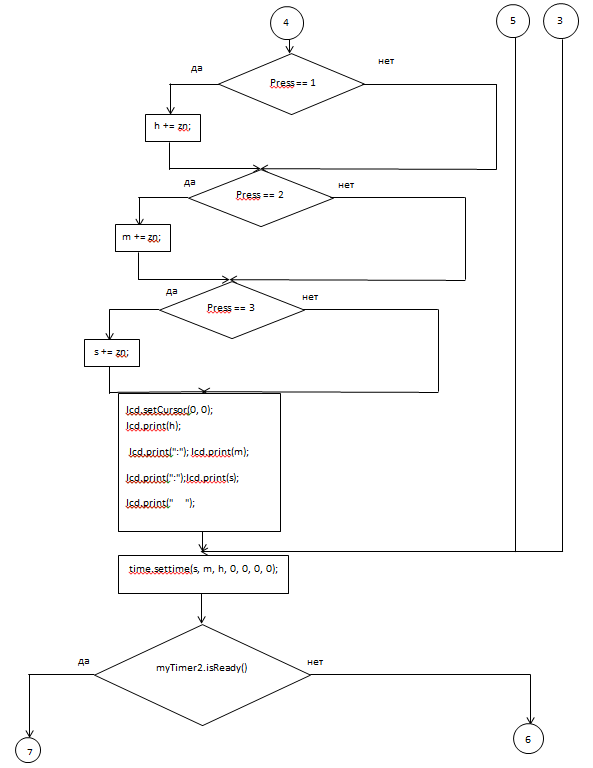
**Технические характеристики двойного реле**

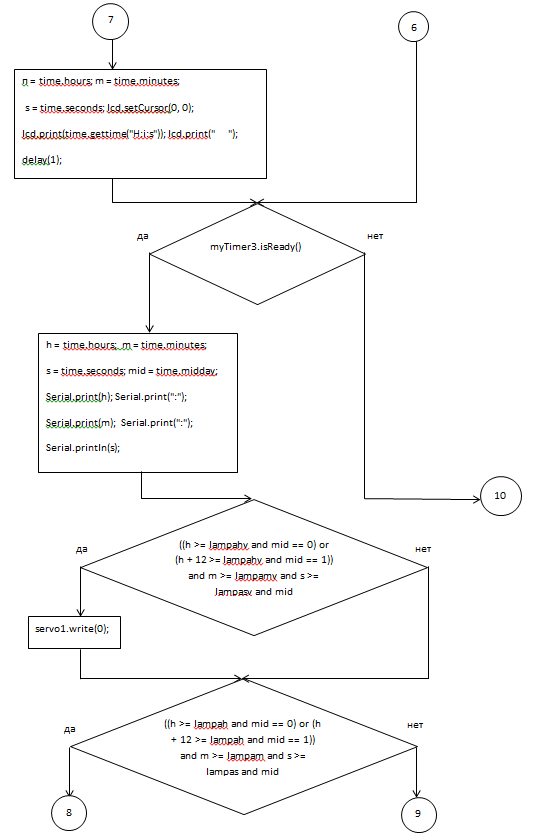
* Напряжение питания: 5 В;
* Потребляемый ток: 30 мА … 40 мА;
* Сигнал включение: 0 В (низкий уровень);
* Оптическая изоляция: есть;
* Количество реле: 2 шт.;
* Тип реле: электромеханическое;
* Коминальный ток нагрузки: 10 А;
* Коммутируемое напряжение: 250VAC, 30VDC;
* Габариты: 50.5мм x 32.5мм x 17м.

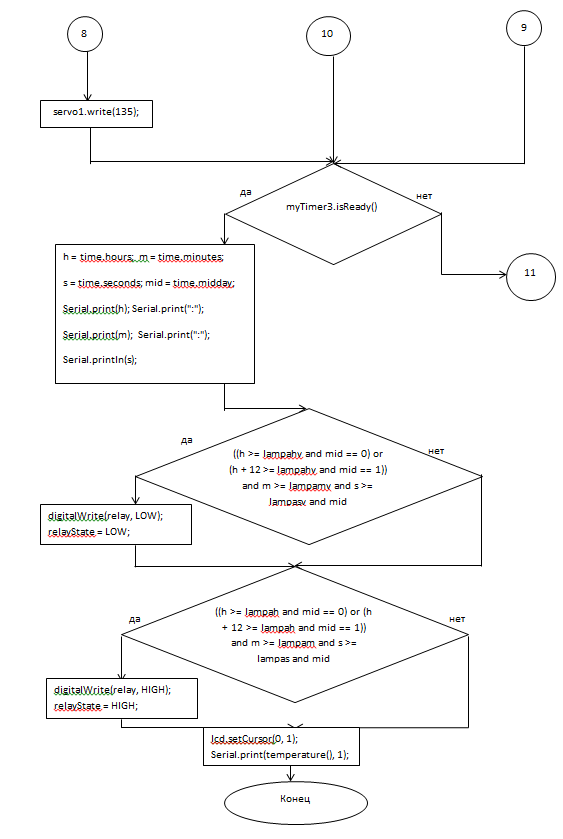
# Приложение 2. Блок-схема алгоритма.











# Приложение 3. Листинг программы.

#include <Servo.h>

#include <Wire.h>

#include <OneWire.h>

#include <GyverTimer.h> //Таймер

#include <LiquidCrystal\_I2C.h> //Дисплей

#include <iarduino\_RTC.h> //Часы

#include <GyverEncoder.h> //Энкодер

Servo servo1;

#define ONE\_WIRE\_BUS 2 // Указываем пин подключения data-вывода датчика температуры

#define term\_power 4

#define CLK 10 //Порты энкодера

#define DT 9

#define SW 8

#define RST 11 // Порты RTC

#define CLKtime 12

#define DAT 13

#define SCL A5 //ЖК Дисплей

#define SDA A4

OneWire oneWire(ONE\_WIRE\_BUS); // Сообщаем библиотеке об устройстве, работающем по протоколу 1-Wire

DallasTemperature sensors(&oneWire); // Связываем функции библиотеки DallasTemperature с нашим 1-Wire устройством (DS18B20)

GTimer\_ms myTimer1(500);//объявление таймеров

GTimer\_ms myTimer3(100);

GTimer\_ms myTimer2(1000);

Encoder enc1(CLK, DT, SW); //Энкодер

iarduino\_RTC time(RTC\_DS1302, RST, CLKtime, DAT); //RTC

LiquidCrystal\_I2C lcd(0x27, SDA, SCL); // Устанавливаем дисплей

int h, m, s, mid;

bool F = false;

int lampas = 0, lampam = 0, lampah = 17;

int lampasv = 20, lampamv = 0, lampahv = 17;

int relay = 8;

volatile byte relayState = LOW;

void setup() {

servo1.attach(3);

servo1.write(0);

Serial.begin(9600);

time.begin();//запускаем часы

lcd.init();//инициализация экрана

lcd.backlight();

enc1.setDirection(REVERSE);//настройка энкодера

enc1.setType(TYPE1);

pinMode(relay, OUTPUT);

Serial.begin(115200);

sensors.begin();

pinMode(term\_power, OUTPUT);

}

float temperature() { // Измеряем температуру 10 секунд

digitalWrite(term\_power, HIGH); // Включаем питание датчика температуры

delay(100); // Задержка перед первым измерением

sensors.requestTemperatures(); // Запрос на измерение температуры (1-й ошибочный)

delay(500); // Задержка перед поторным измерением

sensors.requestTemperatures(); // Запрос на измерение температуры (повторный)

float t = float(sensors.getTempCByIndex(0)); // Получаем значение температуры

digitalWrite(term\_power, LOW); // Отключаем питание датчика температуры

delay(9400); // Задержка, чтобы датчик не нагревался от частых измерений

return (t); // Возвращаем значение температуры в место вызова функции

void loop() {

if (myTimer1.isReady()) {

enc1.tick();

//Serial.println(1);

if (enc1.isHolded()) {//на энкодере долгое нажатие

//Serial.println(2);

h = time.hours;

m = time.minutes;

s = time.seconds;

int Press = 0;

if (F == true)

{

Press++;

}

F = true;

//настройка времени

while (Press < 4) {//переход от часов к минутам и от минут к секундам

//Serial.println(3);

enc1.tick();

int zn = 0;

if (enc1.isLeft()) {

zn--;

}

if (enc1.isRight()) {

zn++;

}

if (enc1.isPress()) {

Press++;

}

if (Press == 1) {

h += zn;

}

if (Press == 2) {

m += zn;

}

if (Press == 3) {

s += zn;

}

// Serial.print(h);

// Serial.print(":");

// Serial.print(m);

// Serial.print(":");

// Serial.println(s);

//выввод на экран время

lcd.setCursor(0, 0);

lcd.print(h);

lcd.print(":");

lcd.print(m);

lcd.print(":");

lcd.print(s);

lcd.print(" ");

}

time.settime(s, m, h, 0, 0, 0, 0);

}

}

//вывод текущего времени на жкран

if (myTimer2.isReady())

{

h = time.hours;

m = time.minutes;

s = time.seconds;

lcd.setCursor(0, 0);

lcd.print(time.gettime("H:i:s"));

lcd.print(" ");

delay(1);

}

if (myTimer3.isReady())

{

h = time.hours;

m = time.minutes;

s = time.seconds;

mid = time.midday;

Serial.print(h);

Serial.print(":");

Serial.print(m);

Serial.print(":");

Serial.println(s);

if (((h >= lampahv and mid == 0) or (h + 12 >= lampahv and mid == 1)) and m >= lampamv and s >= lampasv and mid)

{

servo1.write(0);//Возвращение в начальное положение

}

else {

if (((h >= lampah and mid == 0) or (h + 12 >= lampah and mid == 1)) and m >= lampam and s >= lampas and mid)

{

servo1.write(135);//Подача корма

}

}

}

if (myTimer3.isReady())

{

h = time.hours;

m = time.minutes;

s = time.seconds;

mid = time.midday;

Serial.print(h);

Serial.print(":");

Serial.print(m);

Serial.print(":");

Serial.println(s);

if (((h >= lampahv and mid == 0) or (h + 12 >= lampahv and mid == 1)) and m >= lampamv and s >= lampasv and mid)

{

digitalWrite(relay, LOW);

relayState = LOW; //выключить лампы

}

else {

if (((h >= lampah and mid == 0) or (h + 12 >= lampah and mid == 1)) and m >= lampam and s >= lampas and mid)

{

digitalWrite(relay, HIGH);

relayState = HIGH;//включить лампу

}

}

}

lcd.setCursor(0, 1);

Serial.print(temperature(), 1);

}