Министерство цифрового развития, связи и

Массовых коммуникаций Российской Федерации

Ордена Трудового Красного Знамени федеральное государственное

Бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра «Сетевые информационные технологии и сервисы (СИТиС)»

**ОТЧЁТ**

**о прохождении учебной (технологической) практики**

**на кафедре Сетевых информационных технологий и сервисов (СИТиС)**

Выполнил: студ. гр. БСТ2001

Баранкин А. В.

Вариант №4

Проверила: спец. учебн. лаб. каф. СИТиС.

Тришина С.В.

Москва 2022

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**

**по учебной (технологической) практике**

Спроектировать модель микроконтроллерной схемы на базе Arduino для аквариума с автоматическим подогревом и аэрацией. Реализовать программу для демонстрации процесса работы схемы с получением доступа к базе данных с описанием всех данных. Продемонстрировать результаты работы с помощью отчёта и презентации.

**Цель учебной (технологической) практики**

Цель работы – получение навыков построения схем Arduino и работы с симуляторами схем, навыков выделение объектов, связей предметной области, навыков разработки БД и пользовательского интерфейса. Умение продемонстрировать результаты работы с помощью отчёта и презентации с видеороликом.

Для выполнения цели учебной (технологической) практики необходимо посредством изучения темы индивидуального задания выполнить следующие задачи:

**Задачи учебной (ознакомительной) практики:**

* Составить ER диаграмму предметной области.
* Определить составные части системы и взаимоувязать их в единый макет.
* Изучить и инсталлировать симуляторы для аппаратно-программных средств семейства Arduino.
* Составить схему взаимодействия в выбранном симуляторе.
* Спроектировать базу данных (БД) для предметной области.
* Спроектировать и реализовать графический интерфейс и взаимодействие с БД.

**Содержание**

[1. Описание предметной области 4](#_Toc107321838)

[2. Выделение сущностей предметной области и атрибутов сущностей 5](#_Toc107321839)

[3. Составление ER диаграммы предметной области 6](#_Toc107321840)

[4. Определение составных частей системы и их взаимоувязка в единый макет 7](#_Toc107321841)

[5. Изучение и инсталляция симуляторов для аппаратно-программных средств семейства Arduino. 9](#_Toc107321842)

[6. Выбор электронных компонентов, необходимых для функционирования прототипа макета 10](#_Toc107321843)

[7. Составление схемы взаимодействия в выбранном симуляторе 11](#_Toc107321844)

[8. Проектирование базы данных (БД) для предметной области 12](#_Toc107321845)

[9. Проектирование и реализация графического интерфейса и взаимодействие с БД……... 13](#_Toc107321846)

[Список использованных источников 14](#_Toc107321847)

[ПРИЛОЖЕНИЕ А 15](#_Toc107321848)

[ПРИЛОЖЕНИЕ Б 19](#_Toc107321849)

# **1. Описание предметной области**

Результатом работы должен являться аквариум с автоматическим подогревом и аэрацией, автоматическим кормлением и источником света. Для реализации следует продумать подходы к конструированию устройств, которые будут считывать показатели температуры и насыщенности кислородом воды, а затем регулировать состояние воды до необходимого; кормить рыбок через определённые интервалы времени, регулировать свет.

Также стоит усовершенствовать устройство, добавив возможности регулировки света и автоматического кормления рыбок.

# **2. Выделение сущностей предметной области и атрибутов сущностей**

Для упрощения понимания будущей модели аквариума необходимо выделить сущности предметной области и их атрибуты:

1. Аквариум: рыбки, вода, электроника;

2. Вода: температура, уровень кислорода;

3. Электроника: датчик измерения температуры воды, датчик уровня содержания кислорода в воде, терморегулятор, аэратор для аквариума, таймер, устройство для кормления, источник света;

4. Датчик измерения температуры воды: показатель температуры;

5. Датчик уровня содержания кислорода в воде: показатель уровня кислорода;

6. Терморегулятор: необходимый уровень температуры;

7. Аэратор для аквариума: необходимый уровень кислорода.

8. Таймер: время.

# **3. Составление ER диаграммы предметной области**

Ниже на рисунке 3.1 представлена ER диаграмма предметной области (схема взаимосвязей сущностей и их атрибутов).

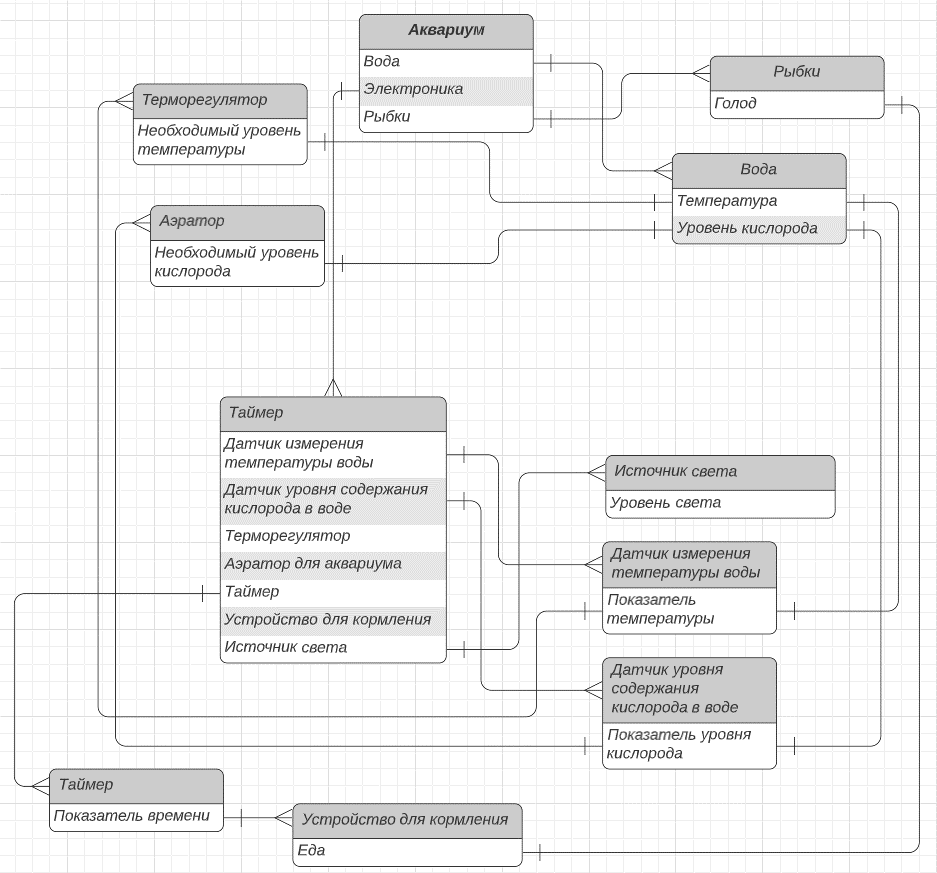


Рисунок 3.1 – ER диаграмма предметной области

# **4. Определение составных частей системы и их взаимоувязка в единый макет**

Для будущей реализации системы спланируем возможный составных частей системы. Чтобы наиболее просто представить вид будущего устройства, составим единый макет составных частей (чертёж). Этот чертёж представлен на рисунке 4.1 и рисунке 4.2.

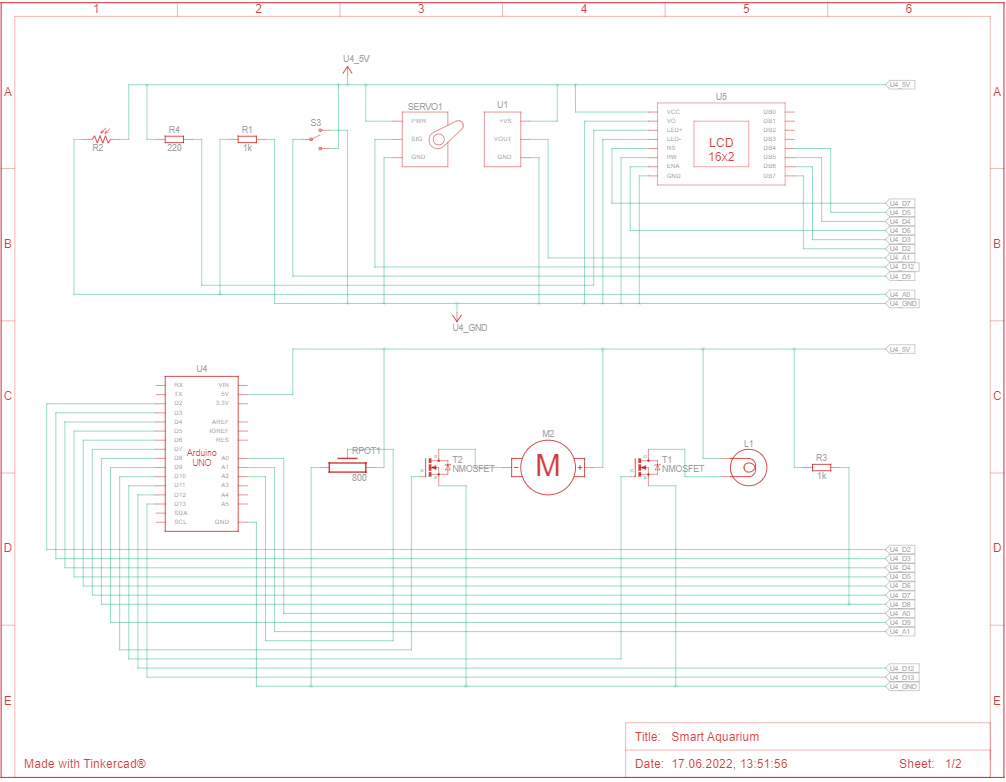


Рисунок 4.1 – Макет (чертёж) системы

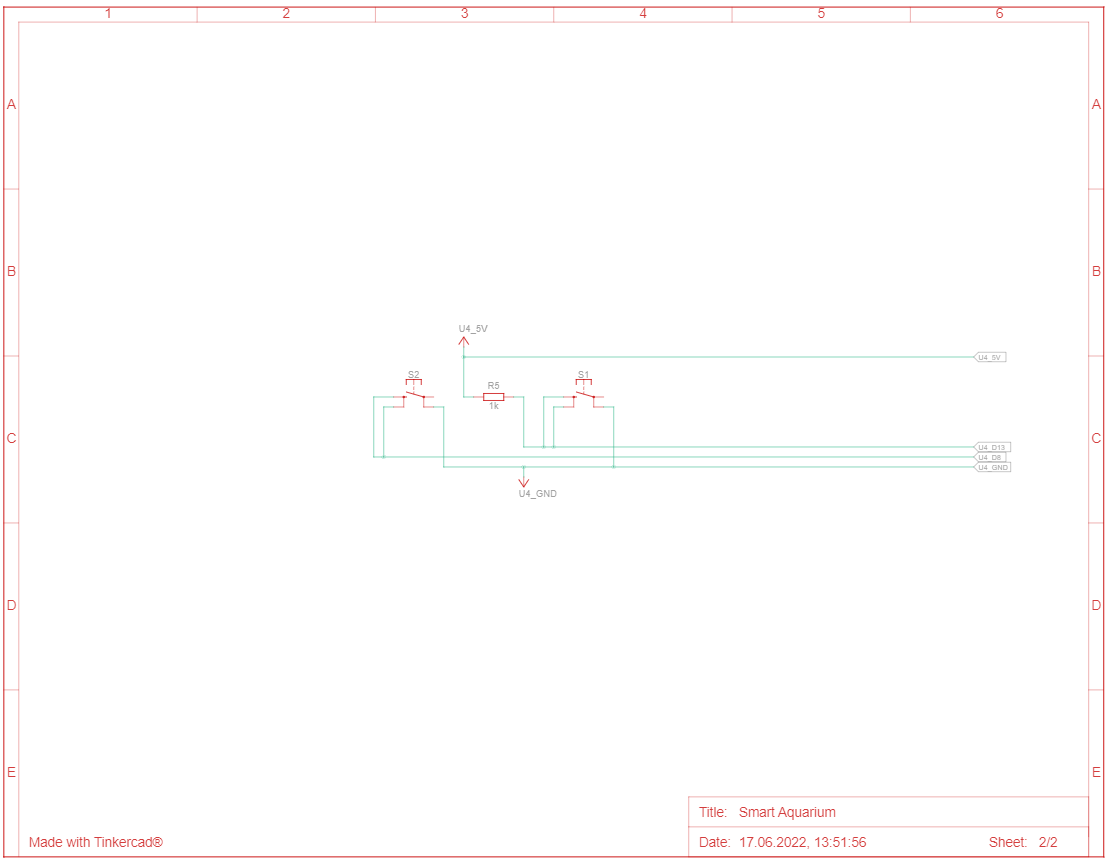


Рисунок 4.2 – Макет (чертёж) системы

# **5. Изучение и инсталляция симуляторов для** **аппаратно-программных средств семейства Arduino.**

В качестве симулятора для аппаратно-программных средств семейства Arduino был выбран симулятор Autodesk Tinkercad.

Данный симулятор очень удобен в использовании, поскольку не требует инсталляции, а запускается в браузере. На рисунке 5.1 представлена стартовая страница симулятора Autodesk Tinkercad.

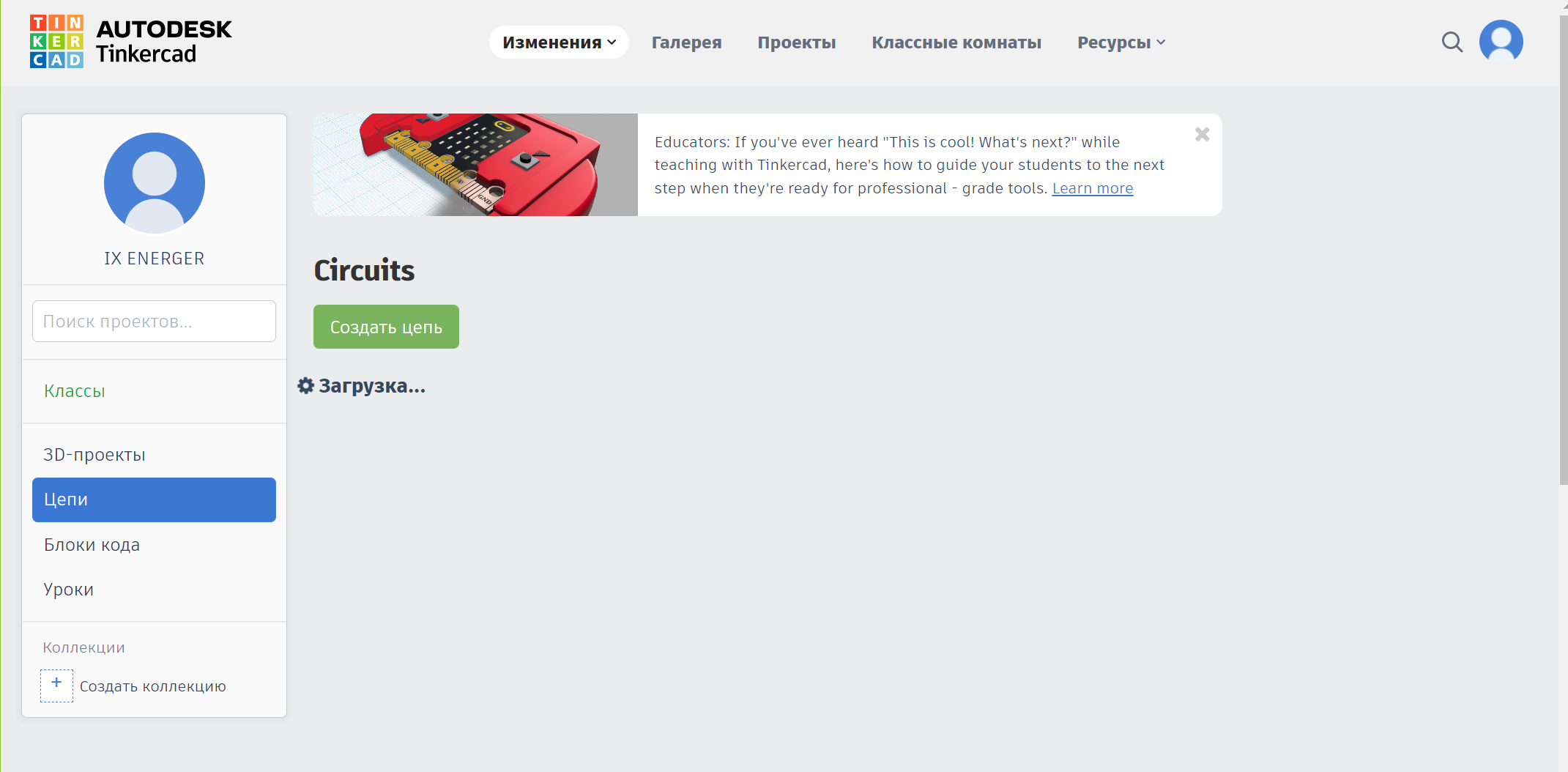


Рисунок 5.1 – Cтартовая страница симулятора Autodesk Tinkercad

# **6. Выбор электронных компонентов, необходимых для функционирования прототипа макета**

Будущий макет должны нести в себе следующие составные части системы: Arduino Uno R3 (1 шт.), ЖК-экран 16 x 2 (1 шт.), резистор с сопротивлением 220 Ом (1 шт.), лампа накаливания (1 шт.), резистор с сопротивлением 1 кОм (3 шт.), фоторезистор (1 шт.), датчик температуры [TMP36] (1 шт.), позиционный микросервопривод (1 шт.), двигатель постоянного тока (1 шт.), n-канальный МОП-транзистор (полевой транзистор с МОП-структурой) (2 шт.), кнопка (2 шт.), ползунковый переключатель (1 шт.), потенциометр с сопротивлением 800 Ом (1 шт.).

# **7. Составление схемы взаимодействия в выбранном симуляторе**

Исходя из списка необходимых электронных компонентов, построим схему в симуляторе Autodesk Tinkercad. Схема отображена на рисунке 7.1.

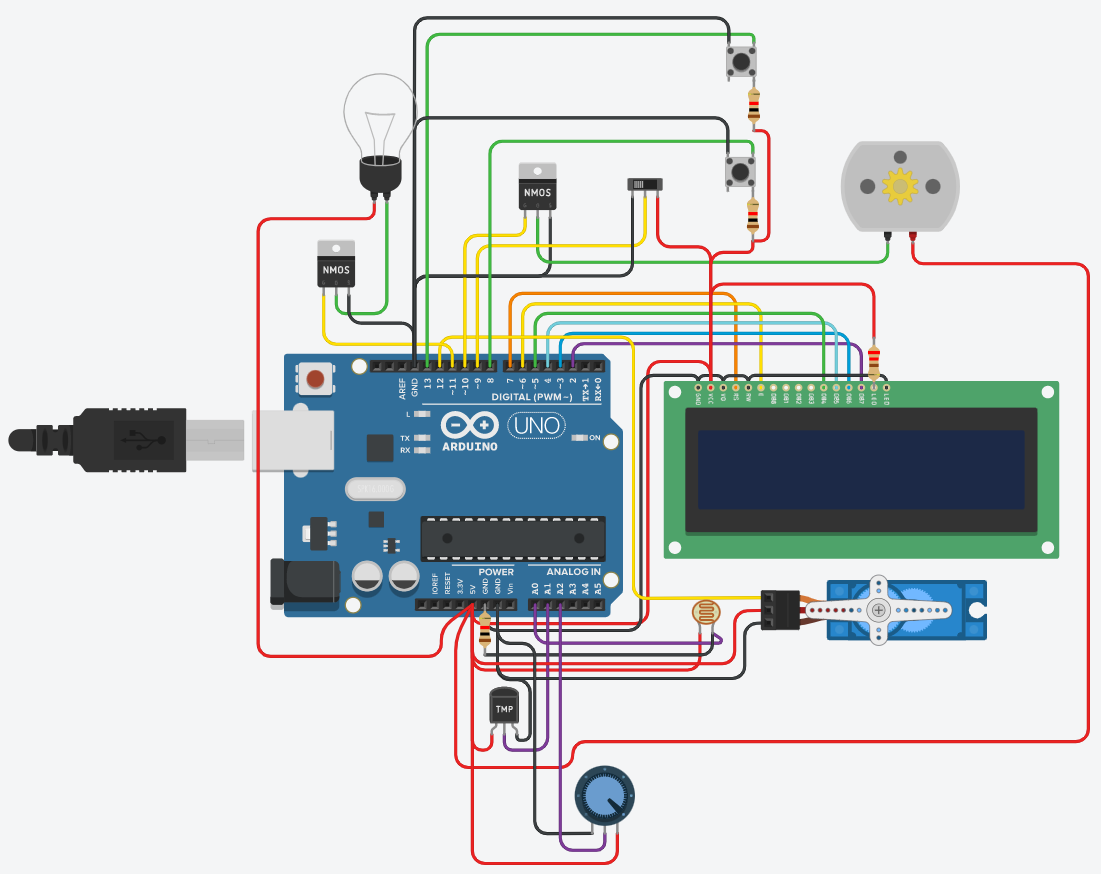


Рисунок 7.1 – Cхема взаимодействия в симуляторе Autodesk Tinkercad

**8. Проектирование базы данных (БД) для предметной области**

БД располагается в мониторе последовательного интерфейса в схеме. Интерфейс взаимодействия отображён на рисунке 8.1.

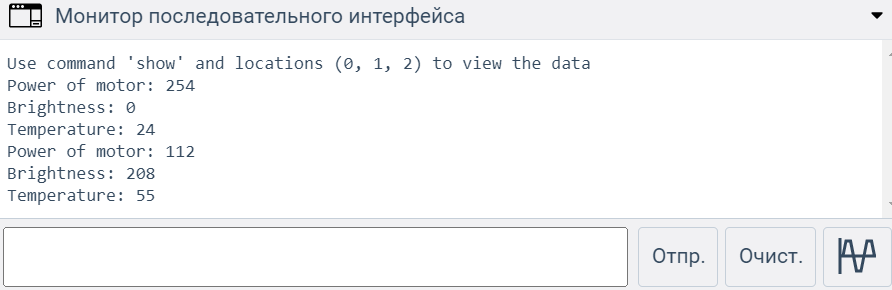


Рисунок 8.1 – Интерфейс взаимодействия с БД

Доступ к БД возможно получить при переходе к моделированию.

# **9. Проектирование и реализация графического интерфейса и взаимодействие с БД.**

С использованием языка программирования C# и .NET Framework был разработан интерфейс, показанный на рисунке 9.1.

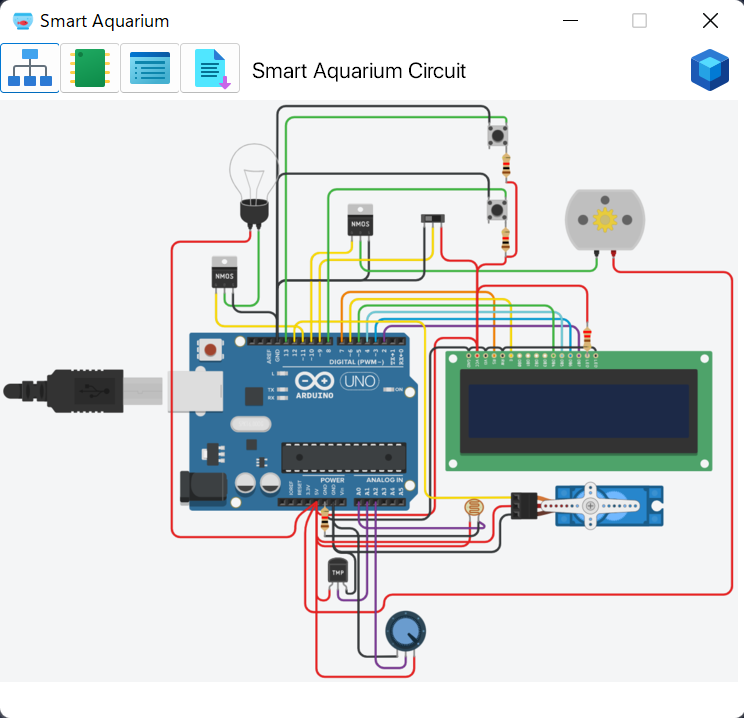


Рисунок 9.1 – Графический интерфейс

В программе предусмотрены возможности демонстрации изображения схемы Arduino, ER диаграммы и списка компонентов, открытия файла отчёта по проделанной работе, перехода к моделированию схемы и БД, расположенной непосредственно в мониторе последовательного интерфейса в схеме.

Исходные файлы проекта представлены на GitHub: <https://github.com/AlexanderBarankin/Interface-for-Arduino-Smart-Aquarium>.

# **Список использованных источников**

1. ГОСТ 7.32-2017 СИБИД. Отчет о научно-исследовательской работе. Структура и правила оформления (с Поправками).

[Правовой информационный ресурс]. – 2017. –

URL: <http://www.consultant.ru/document/cons_doc_LAW_292293/>

(дата обращения 30.06.2022).

2. Положение о порядке подготовки бакалаврской работы.

[Документ]. – 2015. –

URL:

(дата обращения 30.06.2022).

3. Lucid App.

[Онлайн-сервис для создания диаграмм и блок-схем]. – 2022. –

URL: https://lucid.app/

(дата обращения 07.06.2022).

4. Autodesk Tinkercad.

[Онлайн-сервис для работы с схемами Arduino и 3D графикой]. – 2022. –

URL: <https://www.tinkercad.com/dashboard>

(дата обращения 11.06.2022)

5. Stack Overflow.

[Интернет-форум по программированию]. – 2022. –

URL: https://ru.stackoverflow.com/

(дата обращения 16.06.2022)

6. Microsoft Docs.

[Техническая документация Майкрософт]. – 2022. –

URL: https://docs.microsoft.com/ru-ru/

(дата обращения 19.06.2022)

# **ПРИЛОЖЕНИЕ А**

**Код Arduino схемы**

#include <EEPROM.h>

#include <Servo.h>

#include <LiquidCrystal.h>

#define APP\_DELAY 100

#define MULTIPLIER 1

#define FEED\_DELTA 1

#define TIME\_RATE 1000

#define MIN\_TEMPERATURE 15

#define MIN\_LIGHT 200

#define MAX\_LIGHT 1000

#define TEMPERATURE A1

#define LIGHT A0

#define LAMP 11

#define MOTOR 10

#define SWITCH 9

#define INCREASE 13

#define DECREASE 8

#define POWER A2

Servo servo;

LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

byte degree[]{ 0x06, 0x09, 0x09, 0x06, 0x00, 0x00, 0x00, 0x00 };

int location;

String interface;

void setup()

{

pinMode(SWITCH, INPUT);

Serial.begin(9600);

lcd.begin(16, 2);

lcd.createChar(0, degree);

servo.attach(12); servo.write(0);

Serial.println("Use command 'show' and locations (0, 1, 2) to view the data");

}

int brightness = 0;

int temperature = 0;

int power = 128;

int value = 0;

unsigned feedRate = 3;

bool hasEaten = 0;

int last = 0;

void loop()

{

interface = Serial.readString();

auto time = (millis() / TIME\_RATE) - last;

power = analogRead(POWER);

EEPROM.update(0, power);

brightness = map(analogRead(LIGHT), 6, 675, 0, 256);

EEPROM.update(1, brightness);

temperature = (((analogRead(TEMPERATURE) \* 5.0) / 1024.0) - 0.5) \* 100;

EEPROM.update(2, temperature);

auto map\_temp = map(temperature, -40, 125, 0, 256);

if (interface.startsWith("show")) {

location = interface.substring(5,6).toInt();

if (location == 0) {

Serial.print("Power of motor: ");

Serial.println(EEPROM.read(location));

} else if (location == 1) {

Serial.print("Brightness: ");

Serial.println(EEPROM.read(location));

} else if (location == 2){

Serial.print("Temperature: ");

Serial.println(EEPROM.read(location));

} else {

Serial.println("No data");

}

}

analogWrite(LAMP, brightness > map\_temp ? brightness : map\_temp);

analogWrite(MOTOR, power);

if (time == feedRate)

{

servo.write(90);

last += time;

delay(500);

}

else servo.write(0);

lcd.clear();

lcd.setCursor(0, 0);

if (digitalRead(SWITCH))

{

lcd.print("STOP WATCH");

lcd.setCursor(0, 1);

lcd.print(feedRate);

lcd.print("s");

if (!digitalRead(INCREASE)) feedRate += MULTIPLIER;

if (!digitalRead(DECREASE)) feedRate -= MULTIPLIER;

return;

}

lcd.print(temperature);

lcd.write(byte(0));

lcd.print("C ");

lcd.print(time);

lcd.print(" ");

lcd.print(map(power, 0, 1022, 0, 100));

lcd.print("%");

lcd.setCursor(0, 1);

lcd.print("Light: ");

lcd.print(map(brightness, 0, 255, 0, 100));

lcd.print("%");

delay(APP\_DELAY);

}

# **ПРИЛОЖЕНИЕ Б**

**Код программы интерфейса**

namespace SmartAquarium

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void linkLabel1\_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)

{

System.Diagnostics.Process.Start(new System.Diagnostics.ProcessStartInfo { FileName = @"https://www.tinkercad.com/things/bhpYC8SFs1l-smart-aquarium/editel", UseShellExecute = true }); ;

}

private void button1\_Click(object sender, EventArgs e)

{

label1.Text = "ER-chart";

pictureBox1.Image = System.Drawing.Image.FromFile(@"Arduino\ER.png");

}

private void button3\_Click(object sender, EventArgs e)

{

label1.Text = "Smart Aquarium Circuit";

pictureBox1.Image = System.Drawing.Image.FromFile(@"Arduino\Circuit.png");

}

private void button4\_Click(object sender, EventArgs e)

{

label1.Text = "Components List";

pictureBox1.Image = System.Drawing.Image.FromFile(@"Arduino\Components.png");

}

private void button5\_Click(object sender, EventArgs e)

{

label1.Text = "Open report";

pictureBox1.Image = null;

}

}

}