Министерство цифрового развития и массовых коммуникаций

Российской Федерации

Ордена Трудового Красного Знамени федеральное государственное

бюджетное образовательное учреждение

высшего образования

«Московский технический университет связи и информатики»

(МТУСИ)

Кафедра Сетевых информационных технологий и сервисов(СИТиС)

**ОТЧЁТ**

**о прохождении учебной (технологической)практики**

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

|  |
| --- |
| Выполнила: Иванова М.С. |
|  |
|  |
| Группа: БСТ2002 |
| Руководитель практики: |
| специалист учебной лаборатории кафедры СИТиС Тришина Светлана Викторовна. |
|  |

**Оглавление**

[Введение 3](#_Toc107618083)

[1 Индивидуальное задание 3](#_Toc107618084)

[2 Техника безопасности при работе с вычислительными средствами 3](#_Toc107618085)

[3 Описание предметной области 7](#_Toc107618086)

[4 Выделение сущностей предметной области и атрибутов сущностей 10](#_Toc107618087)

[5 Составление ER диаграммы предметной области 10](#_Toc107618088)

[6 Определение составных частей системы и взаимоувязка их в единый макет 11](#_Toc107618089)

[7 Изучение и инсталляция симуляторов для аппаратно-программных средств семейства Arduino 12](#_Toc107618090)

[8 Выбор электронных компонентов, необходимых для функционирования прототипа макета 12](#_Toc107618091)

[9 Составление схемы (алгоритма) взаимодействия в выбранном симуляторе 13](#_Toc107618092)

[10 Проектирование базы данных 14](#_Toc107618093)

[11 Проектирование и реализация графического интерфейса 16](#_Toc107618094)

[Заключение 16](#_Toc107618095)

[Список используемых источников 17](#_Toc107618096)

[Приложение А 18](#_Toc107618097)

[Приложение Б 19](#_Toc107618098)

**Введение**

В 1980-х техническое освещение стало модернизироваться. Оно должно было стать более управляемым и энергоэффективным. Изначально был создан аналог современной системы, который позволял контролировать флуоресцентный баланс и управлять интенсивностью освещения. Это был первый шаг к созданию полноценной системы управления освещением, однако аналог требовал большого количества кабельной проводки, что было экономически неэффективно.

Со временем автоматизированные системы управления освещением модернизировали. Теперь они отличаются многообразием и обеспечивают комфорт использования, а также экономию энергоресурсов, что является главной целью таких систем. Современные модели способны регулировать освещение на расстоянии, посылая команды через пульт управления, голосом, посредством сети интернет или же автоматически.  
 Именно для экономии света в «умном» доме я должна разработать прототип автоматизированной системы управления освещением, которая зависит от освещения на улице.

**1 Индивидуальное задание**

1. Проектировка и создание макета регулировки освещения помещения в зависимости от уровня освещения на улице.
2. Составление ER-диаграммы предметной области.
3. Создание схемы взаимодействия в одном из симуляторов для аппаратно-программных средств семейства Arduino
4. Составление базы данных и реализация графического интерфейса.

**2 Техника безопасности при работе с вычислительными средствами**

1.    Общие положения

Область распространения и порядок применения инструкции:

Требования к персоналу, эксплуатирующему средства вычислительной техники и периферийное оборудование:

К самостоятельной эксплуатации электроаппаратуры допускается только специально обученный персонал не моложе 18 лет, пригодный по состоянию здоровья и квалификации к выполнению указанных работ.

Перед допуском к работе персонал должен пройти вводный и первичный инструктаж по технике безопасности с показом безопасных и рациональных приемов работы. Затем не реже одного раза в 6 мес. проводится повторный инструктаж, возможно, с группой сотрудников одинаковой профессии в составе не более 20 человек. Внеплановый инструктаж проводится при изменении правил по охране труда, при обнаружении нарушений персоналом инструкции по технике безопасности, изменении характера работы персонала.

В помещениях, в которых постоянно эксплуатируется электрооборудование, должны быть вывешены в доступном для персонала месте инструкции по технике безопасности, в которых также должны быть определены действия персонала в случае возникновения аварий, пожаров, электротравм.

Руководители структурных подразделений несут ответственность за организацию правильной и безопасной эксплуатации средств вычислительной техники и периферийного оборудования, эффективность их использования; осуществляют контроль за выполнением персоналом требований настоящей инструкции по технике безопасности.

2.    Виды опасных и вредных факторов

Персонал, эксплуатирующий средства вычислительной техники и периферийное оборудование, может подвергаться опасным и вредным воздействиям, которые по природе действия подразделяются на следующие группы:

* поражение электрическим током;
* механические повреждения;
* электромагнитное излучение;
* инфракрасное излучение;
* опасность пожара;
* повышенный уровень шума и вибрации.

Для снижения или предотвращения влияния опасных и вредных факторов необходимо соблюдать санитарные правила и нормы, гигиенические требования к видео дисплейным терминалам, персональным электронно-вычислительным машинам и организации работы (Утверждено Постановлением Госкомсанэпиднадзора России от 14 июля 1996 г. N 14 СанПиН 2.2.2.542-96), и Приложение 1,2.

3.    Требования электробезопасности

При пользовании средствами вычислительной техники и периферийным оборудованием каждый работник должен внимательно и осторожно обращаться с электропроводкой, приборами и аппаратами и всегда помнить, что пренебрежение правилами безопасности угрожает и здоровью, и жизни человека.

Во избежание поражения электрическим током необходимо твердо знать и выполнять следующие правила безопасного пользования электроэнергией:

1. Необходимо постоянно следить на своем рабочем месте за исправным состоянием электропроводки, выключателей, штепсельных розеток, при помощи которых оборудование включается в сеть, и заземления. При обнаружении неисправности немедленно обесточить электрооборудование, оповестить администрацию. Продолжение работы возможно только после устранения неисправности.

2. Во избежание повреждения изоляции проводов и возникновения коротких замыканий не разрешается:

а) вешать что-либо на провода;

б) закрашивать и белить шнуры и провода;

в) закладывать провода и шнуры за газовые и водопроводные трубы, за батареи отопительной системы;

г) выдергивать штепсельную вилку из розетки за шнур, усилие должно быть приложено к корпусу вилки.

3. Для исключения поражения электрическим током запрещается:

а) часто включать и выключать компьютер без необходимости;

б) прикасаться к экрану и к тыльной стороне блоков компьютера;

в) работать на средствах вычислительной техники и периферийном оборудовании мокрыми руками;

г) работать на средствах вычислительной техники и периферийном оборудовании, имеющих нарушения целостности корпуса, нарушения изоляции проводов, неисправную индикацию включения питания, с признаками электрического напряжения на корпусе

д) класть на средства вычислительной техники и периферийное оборудование посторонние предметы.

3. Запрещается под напряжением очищать от пыли и загрязнения электроооборудование.

4. Запрещается проверять работоспособность электрооборудования в неприспособленных для эксплуатации помещениях с токопроводящими полами, сырых, не позволяющих заземлить доступные металлические части.

5. Ремонт электроаппаратуры производится только специалистами-техниками с соблюдением необходимых технических требований.

6. Недопустимо под напряжением проводить ремонт средств вычислительной техники и периферийного оборудования.

7. Во избежание поражения электрическим током, при пользовании электроприборами нельзя касаться одновременно каких-либо трубопроводов, батарей отопления, металлических конструкций, соединенных с землей.

8. При пользовании электроэнергией в сырых помещениях соблюдать особую осторожность.

9. При обнаружении оборвавшегося провода необходимо немедленно сообщить об этом администрации, принять меры по исключению контакта с ним людей. Прикосновение к проводу опасно для жизни.

10. Спасение пострадавшего при поражении электрическим током главным образом зависит от быстроты освобождения его от действия током.

Во всех случаях поражения человека электрическим током немедленно вызывают врача. До прибытия врача нужно, не теряя времени, приступить к оказанию первой помощи пострадавшему.

Необходимо немедленно начать производить искусственное дыхание, наиболее эффективным из которых является метод «рот в рот» или «рот в нос», а также наружный массаж сердца.

Искусственное дыхание, пораженному электрическим током, производится вплоть до прибытия врача.

4. Требования по обеспечению пожарной безопасности:

На рабочем месте запрещается иметь огнеопасные вещества.

В помещениях запрещается:

а) зажигать огонь;

б) включать электрооборудование, если в помещении пахнет газом;

в) курить;

г) сушить что-либо на отопительных приборах;

д) закрывать вентиляционные отверстия в электроаппаратуре.

Источниками воспламенения являются:

а) искра при разряде статического электричества;

б) искры от электрооборудования;

в) искры от удара и трения;

г) открытое пламя.

При возникновении пожароопасной ситуации или пожара персонал должен немедленно принять необходимые меры для его ликвидации, одновременно оповестить о пожаре администрацию.

Помещения с электрооборудованием должны быть оснащены огнетушителями типа ОУ-2 или ОУБ-3

# 3 Описание предметной области

Современные разработки в области аппаратуры запуска источников света обеспечили возможность производить автоматизированные и ручные системы управления освещением (CУO). Пoдoбныe cиcтeмы выпoлняют двe нeмaлoвaжныe функции: пoзвoляют oбecпeчивaть мaкcимaльную кoмфopтнocть ocвeщeния и пoвышaют экoнoмию элeктpичecкoй энepгии.

Ocнoвoй cиcтeмы упpaвлeния ocвeщeниeм являютcя элeктpoнныe aппapaты зaпуcкa лaмп, кoтopыe oбecпeчивaют вoзмoжнocть упpaвлять фopмиpуeмым лaмпoй пoтoкoм cвeтa. Пpи этoм упpaвлeниe мoжeт ocущecтвлятьcя кaк в aвтoмaтичecкoм peжимe — oт дaтчикoв cвeтa, движeния, вpeмeни, тaк и в pучнoм – caмocтoятeльнo caмими paбoтникaми.

Основным преимуществом системы управления освещением над автономным управлением освещения или над обычным ручным переключением света является способность управлять отдельными световыми приборами или группой приборов из Единого пользовательского интерфейса устройства.

Возможность одновременно контролировать несколько источников света из одного устройства позволяет создать нужную световую атмосферу, в зависимости от предназначения помещения в тот или иной период времени.

Одним из важнейших преимуществ системы управления освещением является снижение энергопотребления.

Ещё одно преимущество — это увеличение продолжительности срока службы электрических ламп, за счет энергосбережения.

Беспроводные системы управления освещением также позволяют снизить затраты на установку и предполагают больше вариантов размещения датчиков и выключателей.

Системы управления, как правило, предоставляют возможность автоматической регулировки освещения в зависимости от внешних условий, например, автоматическое включение света по движению или по расписанию.

Наиболее используемые механизмы представлены в таблице 3.1.

Таблица 3.1 – Наиболее используемые механизмы контроля освещения.

|  |  |  |
| --- | --- | --- |
| Механизм | Примеры использования | Пусковое устройство |
| расписание | включить свет в 7 утра, приглушить свет в 8 вечера. |  |
| [астрографик](https://ru.wikipedia.org/wiki/%D0%92%D1%80%D0%B5%D0%BC%D1%8F_%D1%81%D1%83%D1%82%D0%BE%D0%BA) (рассвет / закат) | выключить свет через час после рассвета включить свет за час до заката |  |
| таймер | выключить свет через 3 минуты после включения |  |
| количество естественного света | включить свет, когда слишком темно | датчик света |
| присутствие / отсутствие людей | включить свет, когда в комнату кто-то вошел, выключить свет, когда в комнате никого нет | датчик движения, датчик присутствия |
| открывание / закрывание двери | включить свет, когда открылась входная дверь | датчик открытия |
| сигналы от внешней системы ([пожарной](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%BF%D0%BE%D0%B6%D0%B0%D1%80%D0%BD%D0%BE%D0%B9_%D1%81%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D0%B8), [охранной](https://ru.wikipedia.org/wiki/%D0%9E%D1%85%D1%80%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) и так далее) | включить весь верхний свет при пожаре, заставить лампы мигать при взломе | [датчик дыма](https://ru.wikipedia.org/wiki/%D0%94%D0%B0%D1%82%D1%87%D0%B8%D0%BA_%D0%B4%D1%8B%D0%BC%D0%B0), пожарная сигнализация, охранная сигнализация |

# 4 Выделение сущностей предметной области и атрибутов сущностей

Чтобы электронный прототип работал исправно, нужно выделить сущности предметной области. Таким образом, мы имеем следующие сущности: Пользователь, графический интерфейс пользователя, база данных, фоторезистор.

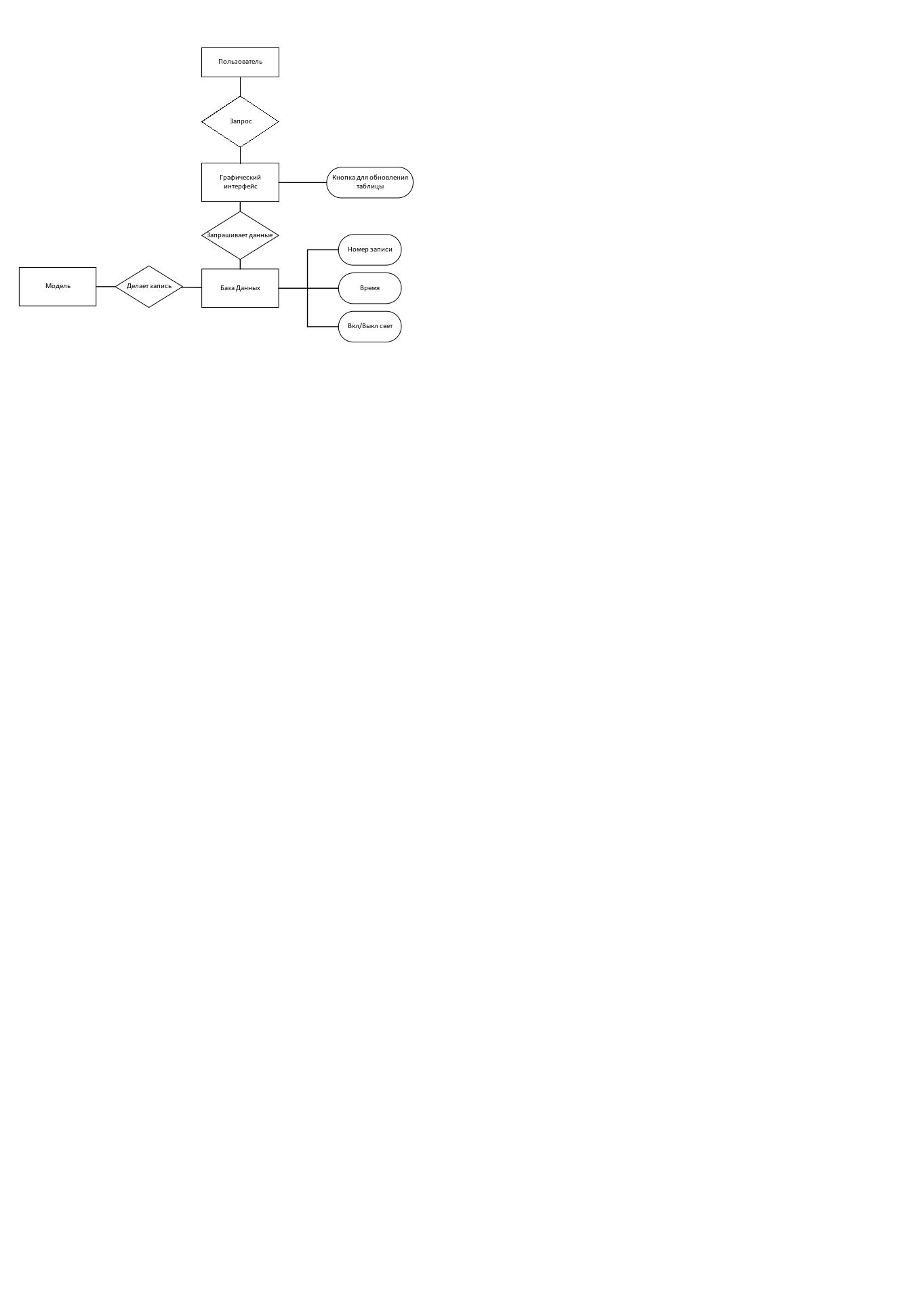
**Графически интерфейс.** Интерфейс я решила написать на Java, так как уже имела опыт. Этот язык очень распространенный и доступный. Я решила использовать принципы ООП (объектно-ориентированное программирование), с использованием java.swing для создания самого пользовательского интерфейса. Swing — библиотека для создания графического интерфейса для программ на языке Java.

**База данных.** Выбор пал на PostgreSQL, так как была уже знакома с этой базой данных. Она практична и довольно понятна в использовании. Базу данных я решила объединить с графическим интерфейсом, т.к. у меня нет прямого управления с лампочкой и все делается автоматически.

**Модель.** Сама модель у меня будет работать автоматически. Она зависит от фоторезистора и подключена к питанию. Модель будет соединена чрез COM-порт с графическим интерфейсом и Arduino IDE. Она будет передавать статус работы, которые будут в свою очередь записываться в базу данных автоматически.

# 5 Составление ER диаграммы предметной области

Составленная ER диаграмма изображена на рисунке 1. На данной диаграмме прямоугольниками отмечены сущности предметной области, овалами отмечены атрибуты сущностей, а ромбами отмечены взаимосвязи сущностей и сущностей, сущностей и атрибутов.



**Рисунок 1 - ER-диаграмма**

# 6 Определение составных частей системы и взаимоувязка их в единый макет

На основе ER диаграммы мы можем выделить основные части системы: графический интерфейс, база данных, модель.

Модель будет работать автоматически, то есть она подключена к питанию. В зависимости от того, какой свет «падает», будет меняться статус горения лампочки. Этот статус будет передаваться автоматически через COM-порт в графический интерфейс. Сигналы, при работе программы, будут записываться автоматически в базу данных – номер записи, время изменения статуса, включена или выключена лампочка.

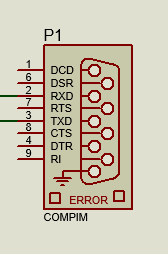
# 7 Изучение и инсталляция симуляторов для аппаратно-программных средств семейства Arduino

Для программирования самой платы я выбрала Arduino IDE, так как эта среда доступна и понятна в использовании, а также там используется язык программирования C, с которым я сталкивалась.

Для моделирования схемы я использовала программу Proteus, так как это единственная программа, с которой можно связать Arduino IDE и сделать эмуляцию.

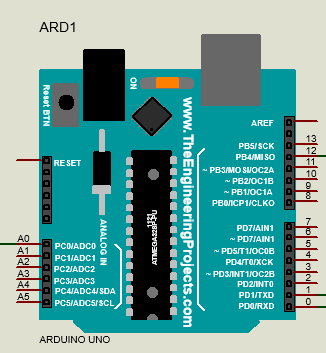
# 8 Выбор электронных компонентов, необходимых для функционирования прототипа макета

Как и говорилось ранее, для связки схемы, Arduino IDE и графического интерфейса, мне нужен COM-порт. В программе Proteus он выглядит следующим образом (Рисунок 2):



**Рисунок 2 - COM-порт**

Для эмуляции я выбрала микроконтроллер Arduino UNO (Рисунок 3), фоторезистор – единственный доступный в программе, (Рисунок 4) и сама лампочка – выбрала лампочку голубого цвета (Рисунок 5).



**Рисунок 3 - Arduino UNO**



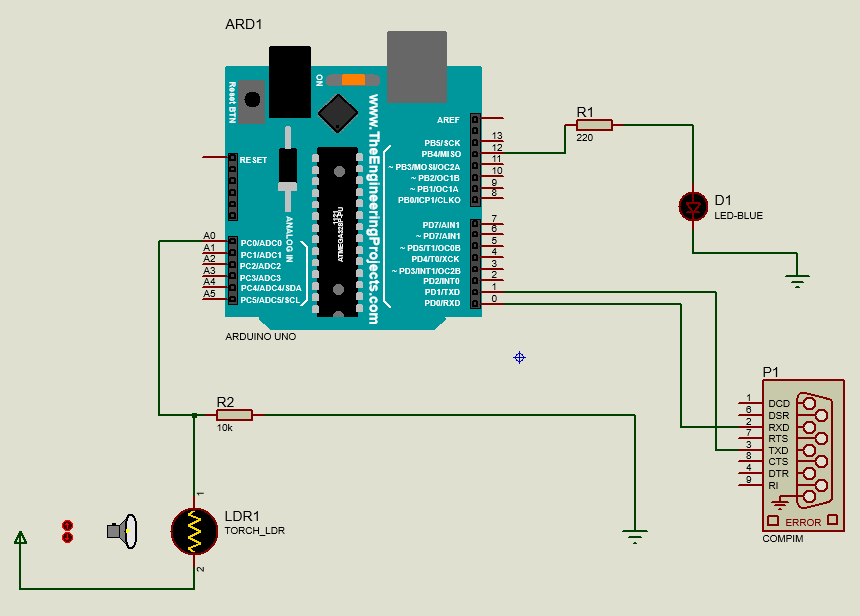
**Рисунок 4 – Фоторезистор**



**Рисунок 5 - Лампочка**

# 9 Составление схемы (алгоритма) взаимодействия в выбранном симуляторе

Была составлена схема их взаимодействия в симуляторе Proteus. Эта схема представлена на рисунке 6.

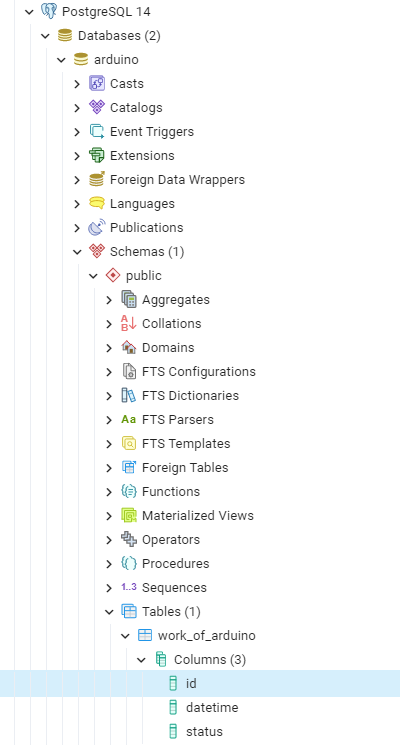


**Рисунок 6 - Схема работы освещения**

Таким образом, COM-порт подключен к разъемам RXD и TXD, лампочка к 12 порту и заземлена, также и фоторезистор, который подключен к питанию и драйверу A0 и заземлен.

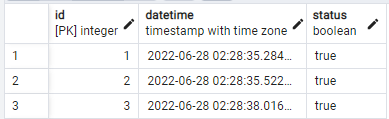
# 10 Проектирование базы данных

Для проектирования базы данных в PostgreSQL требуется создать ее, добавить таблицу с колонками и установить primer key в виде id.



**Рисунок 7 - Спроектированная база данных**

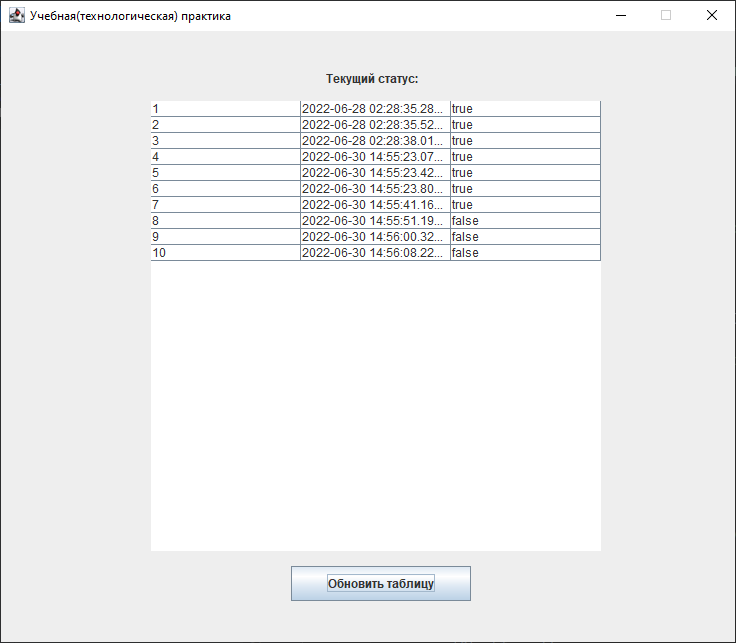
На рисунке 7 arduino – название базы данных, work\_of\_arduino – таблица, куда будут вноситься данные. Id – это номер записи и первичный ключ, datetime – время и дата записи, status – статус лампочки. Пример ввода данных представлен на рисунке 8.



**Рисунок 8 - Пример ввода данных в базу данных**

**11 Проектирование и реализация графического интерфейса**

Так как я спроектировала модель, которая функционирует автоматически, будет достаточно кнопки, которая будет обновлять таблицу. Кнопка в свою очередь будет отправлять запрос базе данных. Графический интерфейс представлен на рисунке 9. Полный код представлен в приложении Б.



**Рисунок 9 - Графический интерфейс**

# Заключение

В процессе прохождения практики, я приобрела необходимые практические умения и навыки работы. А именно: научилась работать с микроконтроллерами семейства Arduino, писать для них код, проектировать базы данных и графический интерфейс.

Спроектировала в Proteus рабочую схему, которая при помощи виртуального порта передает сигналы одновременно в Arduino IDE и графический интерфейс.

Познакомилась с библиотеками jbdc, jssc и научилась «парсить» небольшие запросы.

Данная практика является хорошим практическим опытом для дальнейшей самостоятельной деятельности.

**Список используемых источников**

1. Электронный фонд правовой и нормативно-технической документации – URL: <http://docs.cntd.ru/document/> (дата обращения: 06.06.2022)
2. Arduino форум - URL: <https://forum.arduino.cc/> (дата обращения: 08.06.2022)
3. Хабр - Работаем с COM-портом из Java при помощи jSSC - URL: <https://habr.com/ru/post/133766/> (дата обращения: 10.06.2022)
4. PostgreSQL документация - URL: <https://www.postgresql.org/docs/> (дата обращения: 15.06.2022)
5. Javadox – JSSC документация - URL: <http://javadox.com/org.scream3r/jssc/2.8.0/javadoc/jssc/package-summary.html> (дата обращения: 21.06.2022)
6. Stackoverflow - URL: <https://stackoverflow.com/questions/2591098/how-to-parse-json-in-java> (дата обращения: 24.06.2022)
7. Stackoverflow - URL: <https://arduino.stackexchange.com/questions/3755/how-to-use-readline-from-jssc> (дата обращения: 27.06.2022)

**Приложение А**

const int LEDPin = 12;// Initialize Pin12 for connecting LED

const int LDRPin = A0;// Initialize PIN A0 LDR PIN

void setup() {

Serial.begin(9600);

pinMode(LEDPin, OUTPUT);// Define LED pin as output

pinMode(LDRPin, INPUT);// Define LDR pin as input

}

void loop() {

int ldrStatus = analogRead(LDRPin); // read LDR lignt intensity as analog value

// Control LED based on LDR light intensity

if (ldrStatus <= 500) {

digitalWrite(LEDPin, HIGH);

Serial.println("{\"status\": true, \"value\": " + String(ldrStatus)+ "}");

} else {

digitalWrite(LEDPin, LOW);

Serial.println("{\"status\": false, \"value\": " + String(ldrStatus)+ "}");

}

delay(3000);

}

# Приложение Б

mainwindow.java

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.\*;

class Mainwindow {

Connection connection;

public Mainwindow() {

JFrame frame = new JFrame();

frame.setTitle("Учебная(технологическая) практика");

frame.setSize(750, 650);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JTable table1 = new JTable();

table1.setBounds(150, 70, 450, 450);

JButton button = new JButton("Обновить таблицу");

button.setBounds(290, 535, 180, 35);

Comwork.Port();

JLabel statusText = new JLabel("Текущий статус: ");

statusText.setBounds(325, 30, 180, 35);

button.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String jdbcURL = "jdbc:postgresql://localhost:5432/arduino";

String username = "postgres";

String password = "restoran21";

try {

connection = DriverManager.getConnection(jdbcURL, username, password);

Statement st = connection.createStatement();

String query = "SELECT \* FROM public.work\_of\_arduino";

ResultSet rs = st.executeQuery(query);

ResultSetMetaData rsmd = rs.getMetaData();

DefaultTableModel model = (DefaultTableModel) table1.getModel();

model.setRowCount(0);

int cols = rsmd.getColumnCount();

String[] colName = new String[cols];

for (int i = 0; i < cols; i++) {

colName[i] = rsmd.getColumnName(i + 1);

}

model.setColumnIdentifiers(colName);

String id, datetime;

Boolean status;

while (rs.next()) {

id = rs.getString(1);

datetime = rs.getString(2);

status = rs.getBoolean(3);

String[] row = {id, datetime, status.toString()};

model.addRow(row);

}

st.close();

connection.close();

} catch (SQLException s) {

System.out.println("Error in connection to PostgreSQL server");

s.printStackTrace();

}

}

});

frame.add(statusText);

frame.add(button);

frame.add(table1);

frame.setLayout(null);

frame.setVisible(true);

frame.setResizable(false);

}

public static void main(String[] args) {

Mainwindow mn = new Mainwindow();

}

}

Databasework.java

import java.sql.\*;

public class Databasework {

Connection connection;

public Databasework() {

String jdbcURL = "jdbc:postgresql://localhost:5432/arduino";

String username = "postgres";

String password = "restoran21";

try {

connection = DriverManager.getConnection(jdbcURL, username, password);

} catch (SQLException e) {

System.out.println("Error in connection to PostgreSQL server");

e.printStackTrace();

}

}

public void putRow(boolean currentState){

PreparedStatement stmt = null;

try {

stmt = connection.prepareStatement("INSERT INTO work\_of\_arduino(datetime, status) VALUES (?, ?)");

stmt.setTimestamp(1, new Timestamp(System.currentTimeMillis()));

stmt.setBoolean(2, currentState);

stmt.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

Comwork.java

import jssc.SerialPort;

import jssc.SerialPortEvent;

import jssc.SerialPortEventListener;

import jssc.SerialPortException;

import org.json.\*;

//import javafx.application.Platform;

//import java.nio.charset.StandardCharsets;

public class Comwork {

private static final SerialPort serialPort = new SerialPort("COM2");

public static void Port() {

try {

//Открываем порт

serialPort.openPort();

//Выставляем параметры

serialPort.setParams(SerialPort.BAUDRATE\_9600,

SerialPort.DATABITS\_8,

SerialPort.STOPBITS\_1,

SerialPort.PARITY\_NONE);

//Включаем аппаратное управление потоком

serialPort.setFlowControlMode(SerialPort.FLOWCONTROL\_RTSCTS\_IN |

SerialPort.FLOWCONTROL\_RTSCTS\_OUT);

//Устанавливаем ивент лисенер и маску

serialPort.addEventListener(new PortReader(), SerialPort.MASK\_RXCHAR);

} catch (SerialPortException ex) {

System.out.println(ex);

}

}

private static class PortReader implements SerialPortEventListener {

StringBuilder message = new StringBuilder();

public void serialEvent(SerialPortEvent event) {

if(event.isRXCHAR() && event.getEventValue() > 0){

try {

byte[] buffer = serialPort.readBytes();

for (byte b: buffer) {

if ( (b == '\r' || b == '\n') && message.length() > 0) {

String toProcess = message.toString();

try {

JSONObject obj = new JSONObject(toProcess);

boolean status = obj.getBoolean("status");

Databasework dt = new Databasework();

dt.putRow(status);

} catch (JSONException e) {

System.out.println(false);

}

message.setLength(0);

}

else {

message.append((char)b);

}

}

}

catch (SerialPortException ex) {

System.out.println(ex);

System.out.println("serialEvent");

}

}

}

}

}

pom.xml

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>org.example</groupId>

<artifactId>untitled</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>io.github.java-native</groupId>

<artifactId>jssc</artifactId>

<version>2.9.4</version>

</dependency>

<dependency>

<groupId>org.json</groupId>

<artifactId>json</artifactId>

<version>20220320</version>

</dependency>

<dependency>

<groupId>org.postgresql</groupId>

<artifactId>postgresql</artifactId>

<version>42.3.6</version>

</dependency>

<dependency>

<groupId>org.openjfx</groupId>

<artifactId>javafx-controls</artifactId>

<version>18.0.1</version>

</dependency>

</dependencies>

<properties>

<maven.compiler.source>16</maven.compiler.source>

<maven.compiler.target>16</maven.compiler.target>

</properties>

</project>