**Министерство цифрового развития, связи и массовых**

**коммуникаций Российской Федерации**

**Ордена Трудового Красного Знамени**

**федеральное государственное бюджетное**

**учреждение высшего образования**

**«Московский Технический Университет Связи и Информатики»**

Кафедра Сетевых информационных технологий и сервисов (СИТиС)

**ОТЧЁТ**

**о прохождении учебной (технологической) практики**

**на кафедре Сетевых информационных технологий и сервисов**

**(СИТиС)**

Выполнил: студент группы БСТ2001

Савкин Д.И.

Вариант №17

Проверил: спец. учебн. лаб. каф. СИТиС

Тришина С.В.

Москва 2022

# ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ по учебной (ознакомительной) практике

Спроектировать фитолампу с освещением в зависимости от светового дня.

## Цель учебной (ознакомительной) практики

Изучить основные возможности микроконтроллерной схемы Arduino, научиться строить электронные схемы на базе существующих компонентов с применением навыков компоновки частей в самостоятельную систему с возможностью мониторинга характеристик окружающей среды, а также управлением пороговых значений сенсоров и других элементов.

## Задачи учебной (ознакомительной) практики

Обозначены следующие задачи:

* описать предметную область, выделить сущности и их атрибуты;
* составить ER-диаграмму предметной области;
* подготовить прототип основных частей системы, взаимоувязать их в единый макет;
* спроектировать БД для предметной области, подготовить формат хранения данных;
* разработать графический интерфейс, обеспечить взаимодействие с БД.

СОДЕРЖАНИЕ

1 Описание предметной области, выделение сущностей и их атрибутов 4

2 Составление ER-диаграммы предметной области 5

3 Подготовка прототипа основных частей системы, взаимоувязка их в единый макет 6

4 Проектирование БД для предметной области, подготовка формата хранения данных 8

5 Разработка графического интерфейса, обеспечение взаимодействия с БД 10

ЗАКЛЮЧЕНИЕ 15

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 16

# 1 Описание предметной области, выделение сущностей и их атрибутов

Результатом выполнения текущей работы является прототип фитолампы с освещением в зависимости от длины светового дня. Требуется разработать аналоговое устройство, работающего в роли датчика освещённости (светочувствительный элемент). Также, необходимо подобрать необходимые светоизлучающие элементы (светодиоды), длина волны света которых будет давать требуемую стимуляцию фотосинтеза на плантации, и скомпоновать их в матрицу, геометрия которой будет обеспечивать наибольшее световое покрытие. Необходимо реализовать графический интерфейс с возможностью управления электропитанием лампы, установкой её порогового значения включения\отключения и периодическим мониторингом уровня освещённости, интерпретируемым в виде линейной диаграммы. Продумать подключения светодиодного блока и датчика освещённости к плате микропроцессорного контроллера Arduino. Также, разработать протокол взаимодействия аппаратной и программной частей системы через COM-порты.

В виду комплексности структуры, необходимо разделить её на логические компоненты (сущности предметной области с их атрибутами):

* источник дневного света: яркость;
* светочувствительный элемент: сопротивление в зависимости от яркости падающих на сенсор лучей света;
* система мониторинга;
* светоизлучающие элементы: длина волны;
* записи об изменения состояния: время, описание события
* база данных.

# 2 Составление ER-диаграммы предметной области

На рисунке 2.1 представлена ER-диаграмма предметной области, связывающая обозначенные сущности и их атрибуты.

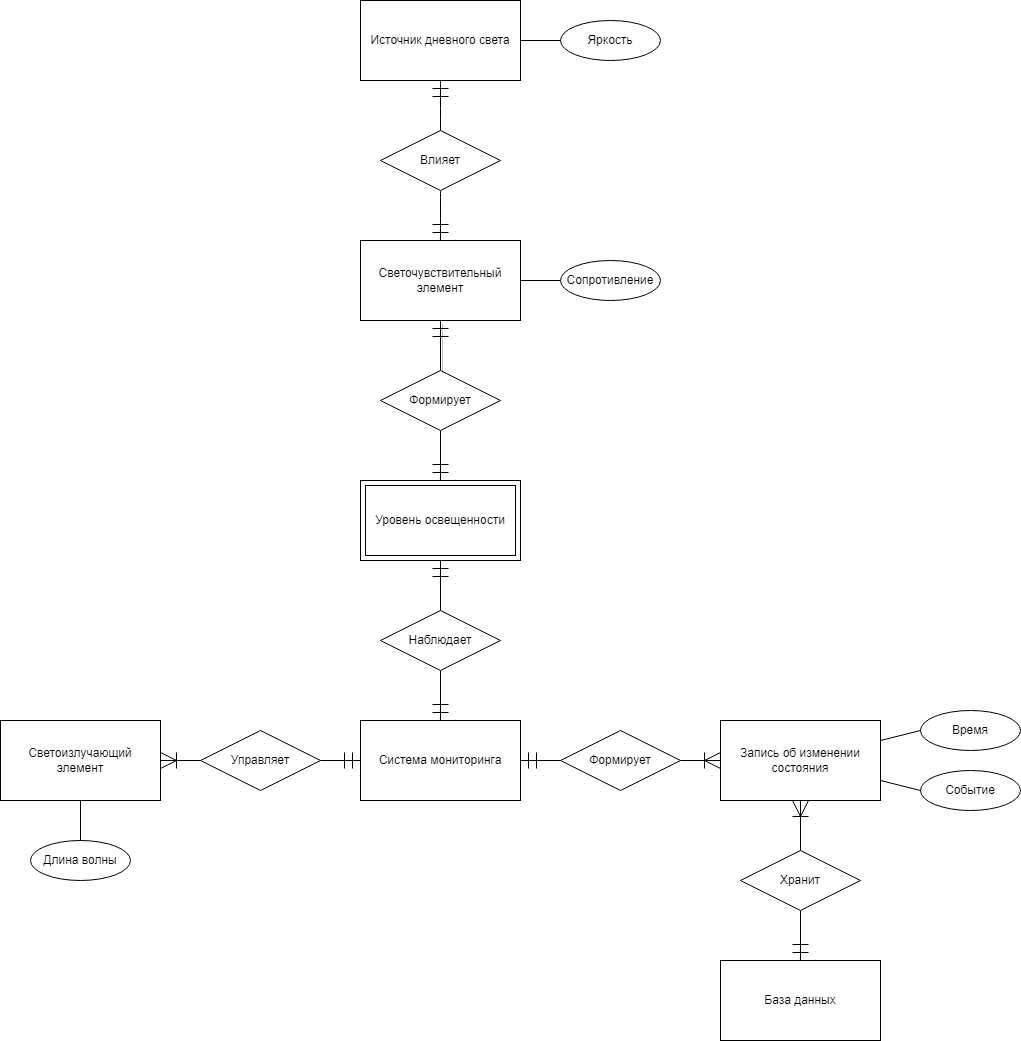


Рисунок 2.1 – ER-диаграмма предметной области

# 3 Подготовка прототипа основных частей системы, взаимоувязка их в единый макет

На рисунке 3.1 представлен единый макет системы, по совместительству являющийся принципиальной электрической схемой.

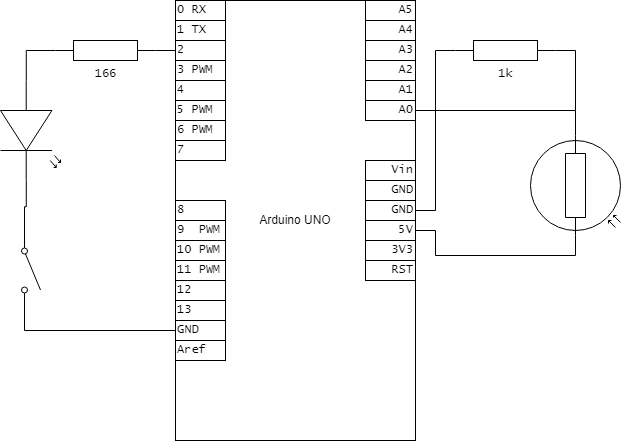


Рисунок 3.1 – Макет системы

Макет состоит из следующего набора электронных компонентов:

* микроконтроллерная плата Arduino UNO,
* ключ,
* светоизлучающий элемент (светодиод),
* резисторы: 166 и 1000 Ом,
* светочувствительный элемент (фоторезистор).

Для симуляции макета был разработан рабочий прототип в программном обеспечении SimulIDE (рисунок 3.2).

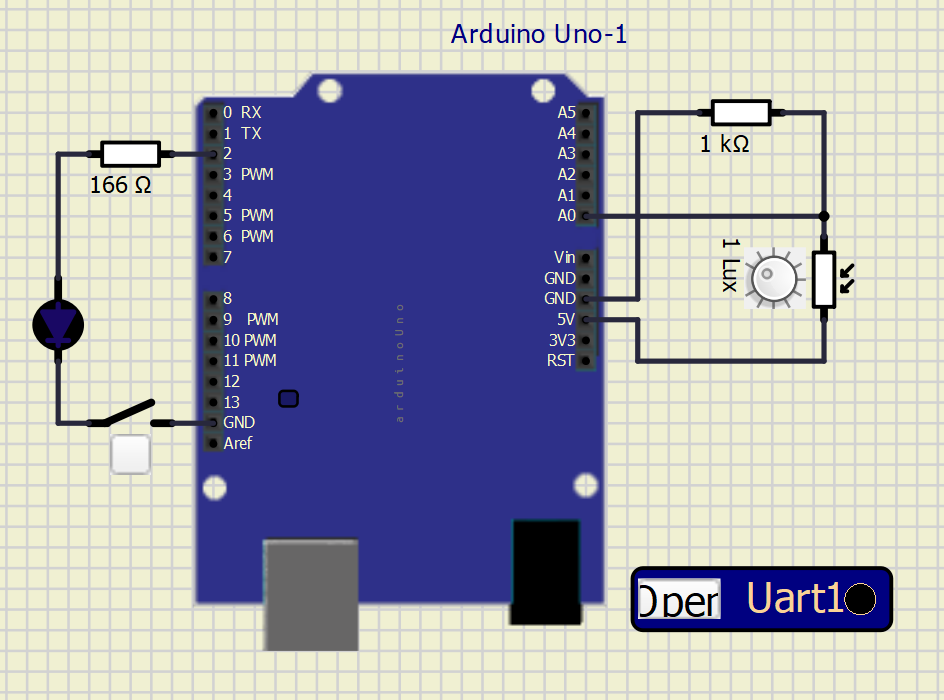


Рисунок 3.2 – Рабочий прототип в симуляторе

# 4 Проектирование БД для предметной области, подготовка формата хранения данных

Любая система реагирует на внутренние и внешние воздействия, изменяя как своё внутреннее состояние, так и состояние компонентов, управляемые ею. В рамках данной работы было необходимо спроектировать базу данных и подготовить формат хранения записей о тех событиях, которые происходят в системе. Хранение записей о событиях позволяет изучить ход работы системы, а так же провести аналитику о тех данных, процессах и показателях, которые происходили в конкретный момент времени.

В качестве базы данных был выбран текстовый файл по следующим причинам:

* нет необходимости устанавливать дополнительное программное обеспечение;
* простота использования;
* чётко определённый формат записей, позволяющий быстро произвести лексический и синтаксический анализы для последующей обработки данных во внешних системах.

Формат записей представляет собой одиночную строку, сформированную по следующему правилу формального определения синтаксиса EBNF:



Рассмотрим каждый идентификатор подробно:

* <record> представляет собой саму запись;
* <timestamp> представляет собой форматированную запись о времени события, которая включает себя сокращённое название месяца, день, час, минуту, секунду и миллисекунду;
* <log-level> может быть равен один из определённых правилом значений: INFO означает, что событие носит информационный характер (например, подключение к плате Arduino), DEBUG означает, что соответствующее описание события носит технический характер и может быть использовано в качестве отладки, ERROR означает, что произошла ошибка, информация о которой может быть использована как для обнаружения неисправностей, так и для аналитики стабильности работы системы и сопутствующих служб.

# 5 Разработка графического интерфейса, обеспечение взаимодействия с БД

Графический интерфейс позволяет конечному пользователю эффективно управлять системой без необходимости знать её техническую реализацию. В рамках данной работы было необходимо разработать графический интерфейс как центр управления системой, а также обеспечить взаимодействие с базой данных, рассмотренной в прошлой главе. За основу была взята платформа для разработки клиентских (пользовательских) приложений для десктопа (ПК) и встроенных систем на Java под названием JavaFX. В качестве языка программирования был взят язык Kotlin. Платформа JavaFX является кроссплатформенной, потому что работает на виртуальной машине Java. Это обеспечивает абсолютную совместимость и чёткую работоспособность на всех современных операционных системах без необходимости разрабатывать графический интерфейс под каждую платформу отдельно.

Графический интерфейс (далее – интерфейс, программа, приложение) представляет собой окно, разделённое на три вертикальные секции: форма для подключения к микроконтроллерной плате Arduino, группа основных элементов управления и телеметрии, строка состояния. Интерфейс представлен на рисунке 5.1.

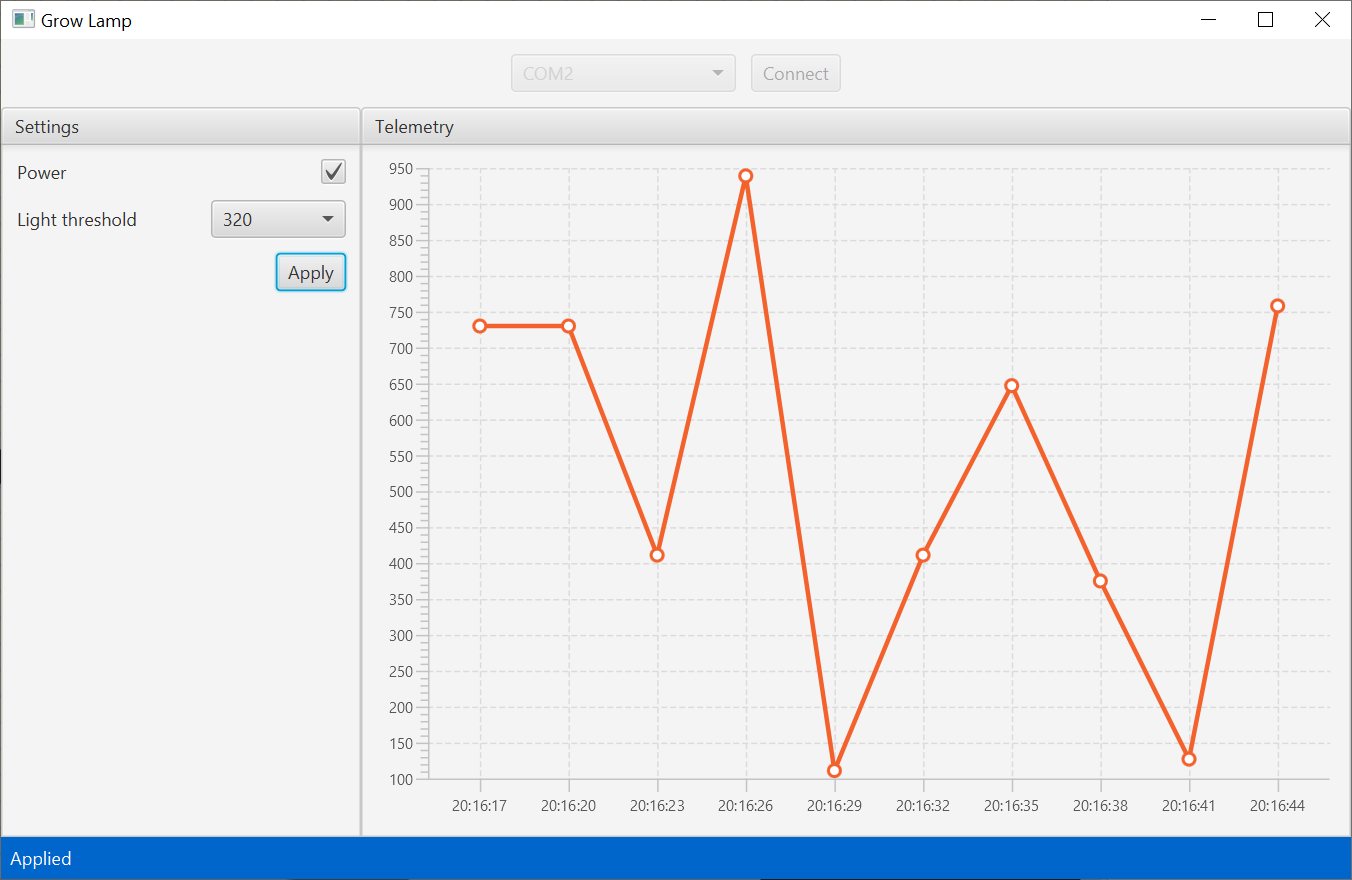


Рисунок 5.1 – Окно графического интерфейса

Форма для подключения к микроконтроллерной плате Arduino представляет собой выпадающий список из COM-портов, доступных для подключения, а так же саму кнопку подключения. Это показано на рисунке 5.2.

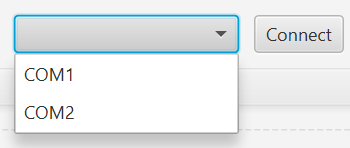


Рисунок 5.2 – Выпадающий список с доступными COM-портами

Группа основных элементов управления и телеметрии (центральная часть рисунка 5.1) представляет собой две вертикальные группы элементов: Settings и Telemetry. Группа Settings содержит в себе элементы для управления электропитанием лампы и установкой порогового значения освещённости в люксах. Выпадающий список имеет предопределённые значения, которые может выбрать пользователь (рисунок 5.3).

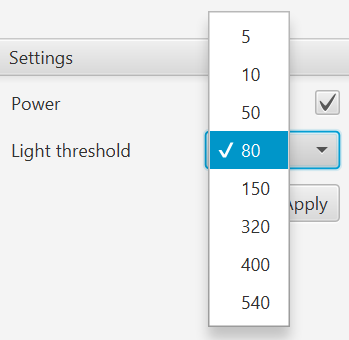


Рисунок 5.3 – Выпадающий список с предопределёнными  
пороговыми значениями освещённости

Группа Telemetry представляет собой линейную диаграмму, на оси ординат которой отмечены значения уровня освещённости в люксах, а на оси абсцисс – моменты времени, в которые были зафиксированы соответствующие значения уровня освещённости. Диаграмма обновляется каждые 3 секунды, показывая не более 10-ти последних измерений.

Программа взаимодействует с микроконтроллером Arduino через COM-порт, обмениваясь сообщениями по разработанному протоколу по модели «клиент-сервер».

В виду комплексности системы и необходимости в быстром расширении функционала с возможностью обратной совместимости был разработан протокол обмена сообщениями между программой и Arduino. Формат сообщений формируется по следующему правилу формального определения синтаксиса EBNF:



Каждый запрос к Arduino содержит в себе обязательную часть в виде значения идентификатора <request-type> (тип запроса) и необязательную часть в виде последовательности значений идентификатора <parameter> (параметры запроса). Тип запроса представляет собой значение длиной в один байт. В зависимости от типа запроса, далее могут идти необязательные параметры, размеры и последовательность которых определяется заранее установленными правилами передачи и обмена сообщениями. На текущий момент реализованы следующие типы запросов:

* READ\_THRESHOLD\_VOLTAGE (значение 0x1) позволяет получить установленное пороговое значение освещённости. В ответ будет получено двухбайтовое значение установленного порога.
* SET\_THRESHOLD\_VOLTAGE (значение 0x2) позволяет установить новое пороговое значение освещённости. Данный запрос не требует ответа.
* READ\_CURRENT\_VOLTAGE (значение 0x3) позволяет получить текущее значение аналогового выхода датчика освещённости. В ответе будет получено двухбайтовое значение текущего значения освещённости.

Взаимодействие с базой данных происходит в фоновом режиме во время исполнения программы. Записи создаются по ходу изменения внутреннего состояния как системы, так и интерфейса. Пример файла с записями (лог) после непродолжительной сессии взаимодействия с интерфейсом представлен на рисунке 5.4.

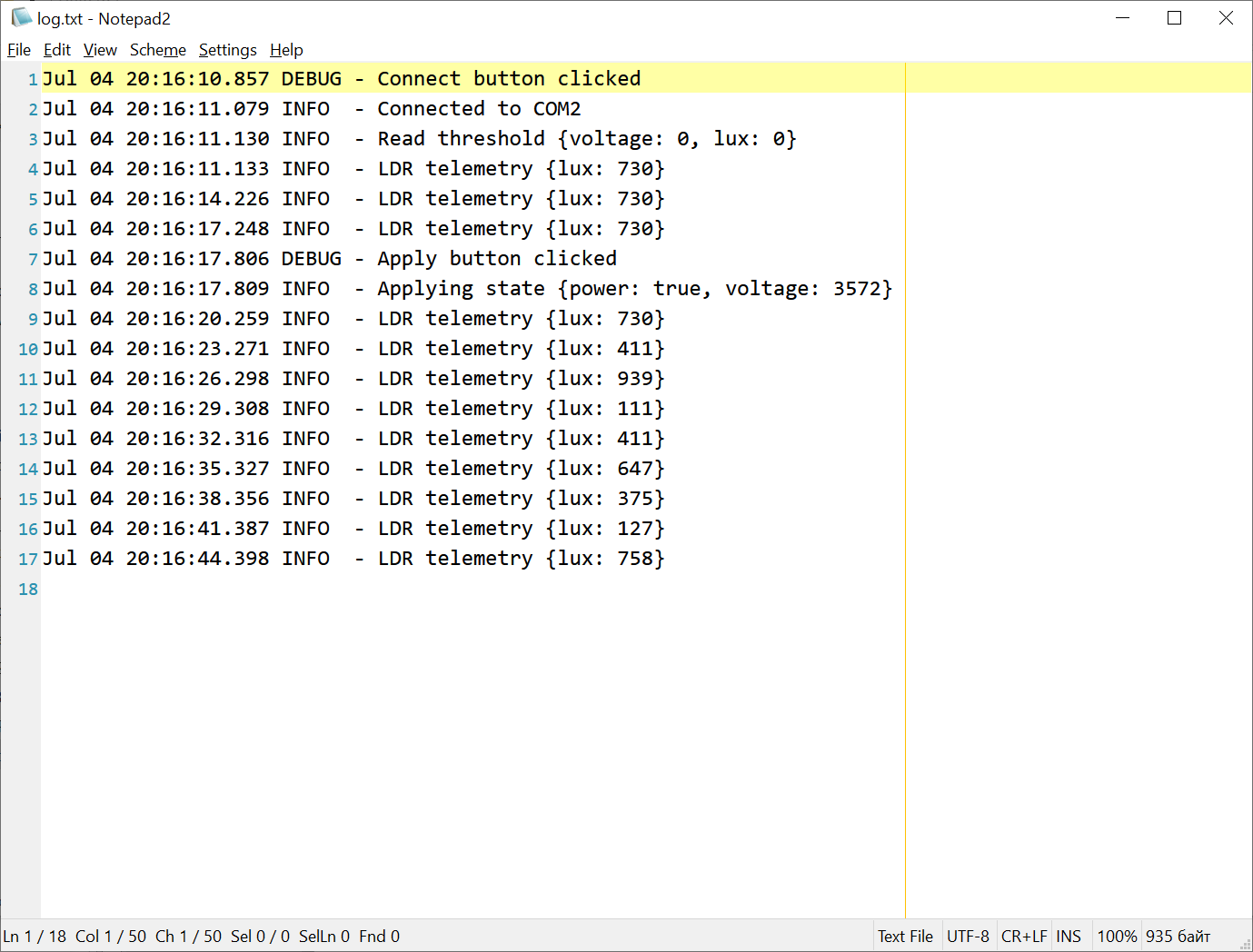


Рисунок 5.4 – Лог в результате работы программы

# ЗАКЛЮЧЕНИЕ

В данной работе был рассмотрен процесс проектирования фитолампы с освещением в зависимости от светового дня. Были установлены цели и поставлены соответствующие задачи. Произведено описание предметной области, выделены сущности с их атрибутами. Составлена ER-диаграмма предметной области, в которой сущности с их атрибутами связаны друг с другом. Подготовлен прототип основных частей системы, который реализован в среде виртуальной симуляции, определена база данных для хранения записей об изменения состояния системы, а так же формат этих записей согласно правилам формального определения синтаксиса EBNF. Разработан графический интерфейс, обеспечено взаимодействие интерфейса и базы данных. Проведены тесты системы по следующим параметрам: стабильность, быстродействие, информативность и затраченное время на выявление типичных проблем и неполадок системы с последующим устранением неисправностей.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчёт о научно-исследовательской работе. Структура и правила оформления (введён в действие Приказом Росстандарта от 24.10.2017 N 1494-ст) – URL: https://docs.cntd.ru/document/1200157208 (дата обращения: 04.07.2022). – Текст: электронный.

2 Система вопросов и ответов о программировании. – URL: https://stackoverflow.com/ (дата обращения: 04.07.2022).

3 Симулятор электронных схем в реальном времени. – URL: https://www.simulide.com/p/home.html (дата обращения: 04.07.2022).

4 Java библиотека для чтения и записи данных через COM-порты. – URL: https://fazecast.github.io/jSerialComm/ (дата обращения: 04.07.2022).