Главное управление по образованию Витебского областного исполнительного комитета

**Учреждение образования  
 «Полоцкий государственный экономический колледж»**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА  
по учебному предмету «Основы алгоритмизации**

**и программирования»**

Тема: «Программно-аппаратный комплекс «Умный дом». Модуль взаимодействия с пользователями»

|  |  |  |
| --- | --- | --- |
| Выполнила:  учащаяся 2 курса группы П21 специальности  2-40 01 01 «Программное обеспечение информационных технологий» | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(Подпись)* | М.В. Долгая |
| Руководитель:  преподаватель | ­­­­­  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(Подпись)* | С.Б. Готовская |

Дата представления \_\_\_.\_\_\_.\_\_\_\_

Отметка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Полоцк 2023

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc137735297)

[1 АНАЛИЗ ИСХОДНЫХ ДАННЫХ И ПОСТАНОВКА ЗАДАЧИ 6](#_Toc137735298)

[1.1 Анализ предметной области 6](#_Toc137735299)

[1.2 Анализ функциональных требований к системе 9](#_Toc137735300)

[2 ПРОЕКТИРОВАНИЕ ПРОГРАММНО-АППАРТНОГО КОМПЛЕКСА 10](#_Toc137735301)

[2.1 Проектирование аппаратной части 10](#_Toc137735302)

[2.2 Проектирование структуры хранения данных 12](#_Toc137735303)

[2.3 Проектирование программного модуля 13](#_Toc137735304)

[2.3.1 Проектирование функциональной структуры 13](#_Toc137735305)

[2.3.2 Проектирование корпуса 13](#_Toc137735306)

[3 РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА 15](#_Toc137735307)

[3.1 Реализация программно-аппаратного комплекса 15](#_Toc137735308)

[3.1.1 Реализация функциональной части проекта 15](#_Toc137735309)

[3.1.2 Реализация нефункциональной части проекта 15](#_Toc137735310)

[3.2 Тестирование программно-аппаратного комплекса 16](#_Toc137735311)

[ЗАКЛЮЧЕНИЕ 20](#_Toc137735312)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 21](#_Toc137735313)

[ПРИЛОЖЕНИЕ A Техническое задание 22](#_Toc137735314)

[ПРИЛОЖЕНИЕ Б Листинг кода программы 25](#_Toc137735315)

[ПРИЛОЖЕНИЕ В Содержание электронного носителя 36](#_Toc137735316)

ВВЕДЕНИЕ

Компьютеры позволяют автоматизировать огромное количество операций, которые ранее были под управлением людей. Скорость обработки данных компьютерами раскрывает широкие возможности, применимые в различных сферах человеческой деятельности, оставляя людям лишь самые важные функции по настройке и управлению различных цифровых систем.

Программно-аппаратный комплекс (ПАК) – это набор различных программных и технических средств. Они работают совокупно для выполнения смежных задач. Чтобы оперативно и результативно развить системы IT-инфраструктуры без дополнительного привлечения специалистов и при этом сократить расход, необходимо использовать ПАК [7].

Аппаратная часть данного комплекса – это устройство, которое отвечает за сбор и обработку информации, то есть компьютер. Техническая часть – это специальное программное обеспечение.

Программно-аппаратные комплексы по своей природе универсальны, и они могут быть использованы в различных сферах деятельности.

Например, это:

− торговля, как розничная, так и оптовая;

− информационные ресурсы;

− банковские организации;

− сфера производства;

− предприятия;

− учебные заведения;

− коммуникация и связь;

− здравоохранительные учреждения;

− туризм.

Применение программно-аппаратных комплексов дает возможность не только значительно уменьшить используемые средства, но и повысить показатели эффективности труда и рентабельности.

Программно-аппаратные комплексы гарантируют:

− простоту управления, которое выполняется при поддержке любого интернет-браузера;

− минимальные расходы, исходя из понятного и простого интерфейса;

− контролирование работоспособности дает возможность восстановить работу устройства, если возникла неполадка;

− уменьшение возможности сбоя в системе;

− эффективное реагирование на вирусные атаки.

Ниже представлены преимущества и недостатки готовых ПАК перед комплексами собственной сборки.

*Плюсы:*

1. Готовое оборудование, на развёртывание которого уйдёт минимум времени.

2. Все компоненты комплекса проходят многоступенчатое тестирование на совместимость и отказоустойчивость. Специалисты компаний, выпускающих ПАК, используют оригинальное программное обеспечение (ПО) для максимальной оптимизации взаимодействия всех частей комплекса.

3. Минимальное количество обслуживающего персонала. Многие ПАК не требуют от пользователя глубоких знаний IT-технологий, а техподдержка от производителей способна помочь в решении большинства типовых проблем без вызова специалиста.

4. Гибкость. Готовый комплекс аппаратных и программных средств, позволяющих выполнять расширенный спектр операций в своей области привлекает покупателей универсальностью.

*Минусы:*

1. Стоимость готовых решений может быть выше, чем разработанные для конкретного процесса комплексы программно-аппаратные от IT-специалистов внутри компании.

2. Гибкость сборных ПАК может быть недостаточна для решения определённых задач, где требуется уникальный подход.

3. Несанкционированное вмешательство в аппаратную или программную части комплекса зачастую приводит к ограничению гарантийного обслуживания.

Цель курсового проектирования − разработать программно-аппаратный комплекс «Умный дом. Модуль взаимодействия с пользователями».

Объектом данного курсового проектирования является автоматизация дома.

Предметом курсового проектирования является разработка программно-аппаратного комплекса «Умный дом. Модуль взаимодействия с пользователями».

# АНАЛИЗ ИСХОДНЫХ ДАННЫХ И ПОСТАНОВКА ЗАДАЧИ

## Анализ предметной области

В соответствии с заданием на курсовое проектирование темой является программно-аппаратный комплекс «Умный дом. Модуль взаимодействия с пользователями».

«Умный дом» − это автоматизированная интеллектуальная система, созданная с целью управления инженерными элементами и системами помещения. Система домашних устройств, способных выполнять действия и решать определённые повседневные задачи без участия человека или под его определенным контролем при помощи специальных средств. Домашняя автоматизация в современных условиях − чрезвычайно гибкая система, которую пользователь конструирует и настраивает самостоятельно или с помощью компаний установки, в зависимости от собственных потребностей [8].

Большое количество устройств, объединенных в одну систему, обеспечивают безопасность жилья, комфорт и экономию ресурсов. Кроме этого, важным преимуществом такой системы является централизованное управление всеми задействованными в комплексе устройствами.

Система «Умный дом» имеет свои подсистемы, состоящие из связанных по смыслу их работы устройств:

− системы освещения;

− электросистемы;

− системы вентиляции;

− климат контроль;

− системы видеонаблюдения;

− пожарная сигнализация;

− охранная сигнализация;

− системы контроля доступа (распознает хозяев);

− контроль нагрузок и аварийных состояний.

Система управления «Умным домом» выполняет ряд определенных функций:

1. Управление различным оборудованием дома из одной точки.

2. Выполнение сценариев. Сценарий – это совокупность заранее согласованных с пользователем запрограммированных действий, которые система умный дом выполняет по команде.

3. Автоматическое управление умным домом.

Существует несколько различных видов систем. Кроме того, само оборудование может быть беспроводное или проводное.

Система состоит из таких элементов:

1. Контроллер «Умного дома» (главный и дискретные модуляторы ввода/вывода);

2. Модули расширения и связи (коммутаторы, роутеры, GPS/GPRS модули);

3. Элементы коммутации электрической цепи (реле, диммеры, блоки питания);

4. Измерительные приборы, датчики и сенсоры (датчики движения, температуры, света и др.);

5. Элементы управления системой (пульты, сенсорные панели, КПК, планшеты);

6. Исполнительные механизмы (клапаны воды, механизмы вентиляции, механизмы газа, ролеты и т.д.).

«Умный дом» имеет ряд преимуществ: позволяет экономить до 10-18% электроэнергии, повышать комфорт, безопасность и т.д. Принцип работы «Умного дома» заключается в центральном компьютере, принимающем сигналы от командных устройств, затем эти сигналы передаются исполнительным системам. Управление различными устройствами осуществляется простейшей системой автоматики.

Очень важно, чтобы алгоритмы взаимодействия подсистем в доме были гибкими, и могли приспосабливаться под изменяющиеся нужды владельца дома [10].

Для создания данного устройства были использованы различные модули, а в качестве платы была выбрана плата Arduino UNO. Для взаимодействия пользователей с устройством были использованы SPI OLED-дисплей и инфракрасный пульт дистанционного управления (ИК-пульт), инфракрасный датчик, модуль часов реального времени, датчик температуры.

Arduino/Genuino Uno - это устройство на основе микроконтроллера ATmega328 (datasheet). В его состав входит все необходимое для удобной работы с микроконтроллером: 14 цифровых входов/выходов (из них 6 могут использоваться в качестве ШИМ-выходов), 6 аналоговых входов, кварцевый резонатор на 16 МГц, разъем USB, разъем питания, разъем для внутрисхемного программирования (ICSP) и кнопка сброса. Для начала работы с уcтройством достаточно просто подать питание от AC/DC-адаптера или батарейки, либо подключить его к компьютеру посредством USB-кабеля [9].

OLED (Organic Light-Emitting Diode) − данная технология зачастую применяется при изготовлении крупноразмерных жидкокристаллических панелей.

OLED панели состоят из огромного количества пикселей, из которых и формируется общая картинка.

OLED-дисплей работает без подсветки, потому что он излучает свой собственный видимый свет. Таким образом, он может отображать глубокие уровни черного и может быть тоньше и легче, чем жидкокристаллический дисплей (LCD). В условиях низкой освещенности (например, в темной комнате) OLED-экран может обеспечивать более высокий коэффициент контрастности, чем ЖК-дисплей, независимо от того, используются ли на ЖК-дисплее люминесцентные лампы с холодным катодом или светодиодная подсветка [11].

Инфракрасный пульт дистанционного управления использует световые сигналы, посылаемые с передатчика, расположенного на одном конце пульта дистанционного управления, на приемник в другом электронном устройстве.

Передатчик на конце пульта дистанционного управления передает электромагнитный сигнал с длиной волны, которая несколько длиннее видимого света. ИК-пульт дистанционного управления излучает этот «невидимый» свет в двоичном коде. Каждая «команда» имеет определенный код. Для считывания данных кодов был использован ИК-датчик [12].

Электронная схема, предназначенная для учета хронометрических данных (текущее время, дата, день недели и др.), представляет собой систему из автономного источника питания и учитывающего устройства. Модуль DS3231 (модуль часов реального времени) представляет собой обыкновенные часы. Данный модуль был использован для установки пользователем конкретного времени включения/включения освещения [13].

DS18B20 – это цифровой температурный датчик, обладающий множеством полезных функций. DS18B20 – это целый микроконтроллер, который может хранить значение измерений, сигнализировать о выходе температуры за установленные границы (сами границы можно устанавливать и менять), менять точность измерений, способ взаимодействия с контроллером и многое другое.

Микросхема имеет три выхода, из которых для данных используется только один, два остальных – это земля и питание. Число проводов можно сократить до двух, если использовать схему с паразитным питанием и соединить 5 В с землей. К одному проводу с данными можно подключить сразу несколько датчиков DS18B20 и в плате Arduino будет задействован всего один пин. Датчик температуры был использован для вывода информации о температуре в помещении пользователю [14].

По средствам вышеперечисленных модулей пользователь сам может управлять устройством «умного дома», что упрощает и автоматизирует его использование.

Исходя из сделанных выводов, данное программное средство должно обеспечивать взаимодействие с пользователями для управления устройством «умный дом».

## Анализ функциональных требований к системе

Целью курсового проектирования является разработка программно-аппаратного комплекса «Умный дом. Модуль взаимодействия с пользователями».

Функциями данного модуля являются:

− выбор сценариев устройства при помощи пульта;

− контроль температуры при помощи дисплея;

− контроль времени при помощи дисплея;

− контроль выполнения сценариев с помощью дисплея;

− взаимодействие с пользователем, для установки включения/выключения освещения в определенное время;

− вывод информационных сообщений.

Программное средство должно использовать 2 вида пользователей: Пользователь, Контроллер. Каждому из пользователей предоставлены свои возможности.

В качестве графической части были разработаны диаграмма вариантов использования и структурная электрическая схема подключение устройства.

В процессе проектирования программы был использован язык программирования С++, средой программирования является Arduino IDE 2.0.4.

# ПРОЕКТИРОВАНИЕ ПРОГРАММНО-АППАРТНОГО КОМПЛЕКСА

## Проектирование аппаратной части

Исходя их функциональных требований к разрабатываемому устройству «Умный дом. Модуль взаимодействия с пользователями», в состав устройства необходимо включить следующие компоненты:

− плата Arduino Uno − для осуществления функций контроллера;

− датчик температуры DS18B20 − для контроля температуры;

− часы реального времени RTC\_DS1307 − для контроля времени;

− OLED дисплей − для вывода информации пользователю;

− ИК-датчик − для считывания кодов с пульта управления.

Изначально для обеспечения работы системы была выбрана плата Arduino Nano, однако ее размеры не позволили вместимость полного кода. Равноценной заменой данного модуля является плата Arduino Uno, которая была использованная в данном проекте.

По результатам проектирования была создана структурная электрическая схема подключения устройства, которая представлена в графической части.

Из данной схемы видно, какие входы и выходы задействованы для обеспечения работы устройства:

* использовано 4 аналоговых входов (А0, A3, А4, А5);
* использовано 10 цифровых входов/выходов (D2, D4, D5, D6, D7, D8, D9, D10, D11, D13);
* использован 1 выход для заземления (GND);
* использован 1 выход для питания на 5 В (5V).

На рисунке 2.1 изображены цвета и номера проводов, использованных для подключения системы.

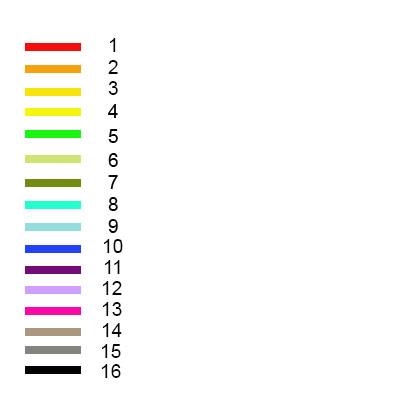


Рисунок 2.1 – Числовая и цветовая нумерация использованных проводов

Провода с номерами имеют следующее назначения:

* 1 – провод, предназначенный для подачи питания в 5 вольт (5V);
* 2 – провод, предназначенный для соединения OLED с пином на Arduino (D9/DS);
* 3 – провод, предназначенный для соединения OLED с пином на Arduino (D8/RES);
* 4 – провод, предназначенный для соединения соединения линии данных модуля реального времени на LCD с пином линии данных на Arduino× (A4/SDA);
* 5 – провод, предназначенный для соединения OLED с пином на Arduino (D13/D0);
* 6 – провод, предназначенный для получения сигнала с датчика температуры из цифрового входа/выхода D2;
* 7 – провод, предназначенный для передачи сигнала на реле из цифрового входа/выхода D4;
* 8 – провод, предназначенный для соединения OLED с пином на Arduino (D11/D1);
* 9 – провод, предназначенный для получения сигнала с инфракрасного датчика из цифрового входа/выхода A3;
* 10 − провод, предназначенный для передачи сигнала на реле из цифрового входа/выхода D6;
* 11 – провод, предназначенный для получения сигнала с фоторезистора из цифрового входа/выхода A0;
* 12  – провод, предназначенный для передачи сигнала на реле из цифрового входа/выхода D5;
* 13 − провод, предназначенный для соединения OLED с пином на Arduino (D10/CS);
* 14 − провод, предназначенный для передачи сигнала на реле из цифрового входа/выхода D7;
* 15 − провод, предназначенный для соединения линии синхронизации на модуле реального времени с пином линии синхронизации на Arduino (A5/SCL);
* 16 − провод, предназначенный для заземления (GND).

## Проектирование структуры хранения данных

Для разработки программно-аппаратного комплекса были созданы переменные для хранения данных, приведенные в таблице 2.1.

Таблица 2.1 − Переменные разрабатываемого модуля

|  |  |  |
| --- | --- | --- |
| Название | Тип | Назначение |
| autoLighting | byte | Предназначена для управления авто освещением и освещением по времени |
| autoVentilation | byte | Предназначена для управления авто вентиляцией |
| autoHeating | byte | Предназначена для управления авто обогревом |
| enablingTime | byte | Предназначена для установки освещения по времени |
| printTime | byte | Предназначена для вывода времени на дисплей |
| Hour | String | Предназначена для записи установленного времени в часах и его вывода на экран |
| Minut | String | Предназначена для записи установленного времени в минутах и его вывода на экран |
| inputMinut | byte | Предназначена для установки требуемого времени в минутах |
| inputHour | byte | Предназначена для установки требуемого времени в часах |

## Проектирование программного модуля

### Проектирование функциональной структуры

В результате проведенного анализа исходных данных сформулированы требования к разрабатываемому модулю.

Модуль должен обладать следующими возможностями:

− выбор сценариев устройства при помощи пульта;

− контроль температуры при помощи дисплея;

− контроль времени при помощи дисплея;

− контроль выполнения сценариев с помощью дисплея;

− взаимодействие с пользователем, для установки включения/выключения освещения в определенное время;

− вывод информационных сообщений.

При создании программы использовались функции, представленные в таблице 2.2.

Таблица 2.2 – Описание функций

|  |  |  |  |
| --- | --- | --- | --- |
| Название функции | Назначение | Возвращаемое значение | Параметры |
| 1 | 2 | 3 | 4 |
| void clear\_screen(void); | Очистка дисплея | ­− | − |
| void drawTime (String hour, String minut); | Вывод установленного времени | ­− | String hour – значение установленного часа, String minut – значение установленных минут |
| void setup(); | Описание модулей экрана, датчиков, назначение кнопок пульта | − | − |
| void loop(); | Считывание кодов кнопок пульта, управление устройствами, автоуправление, вывод температуры и времени | − | − |

### Проектирование корпуса

При проектировании программно-аппаратного комплекса необходимо создать удобный и интуитивно-понятный пользовательский корпус, для обеспечения простой работы с устройством.

После сборки программно-аппаратного комплекса был создан макет будущего корпуса в приложении Blender, представленный на рисунке 2.2.

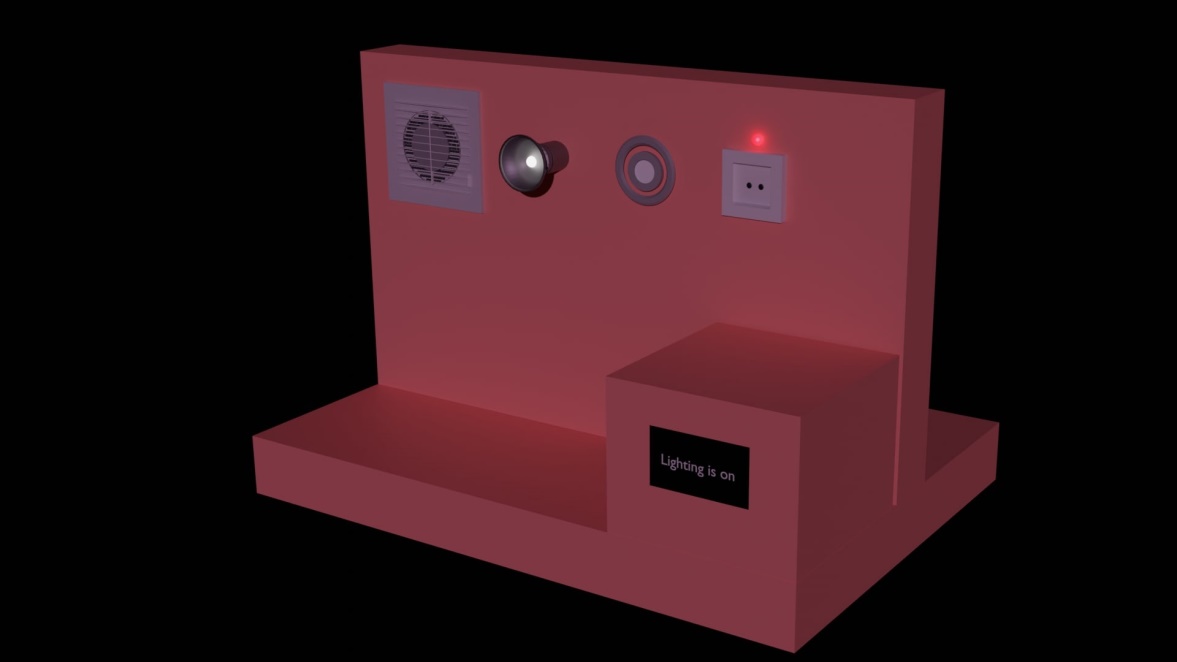


Рисунок 2.2 − Корпус ПАК в Blender

На данном рисунке видно, что основная стенка корпуса, на которой находятся устройства, располагается на специальной подставке, выходящей с обеих сторон. Однако для экономии материала было принято решение уменьшить подставку в два раза, оставив для устойчивости конструкции только переднюю часть.

На основной части корпуса располагаются различные устройства:

* вентиляция − для охлаждения помещения;
* инфракрасная лампочка − для обогрева помещения;
* обычная лампочка − для освещения;
* розетка − для питания различных устройств.

Так же на корпусе располагается отдельная часть, внутри которой находится само устройство управления. Наружу были выведены отдельные части устройства:

* дисплей − для вывода информационных сообщений;
* датчик температуры − для считывания температуры в помещения;
* фоторезистор − для контроля освещения в помещении;
* ИК-датчик − для считывания кодов кнопок с пульта.

Данный программно-аппаратный комплекс управляется пультом.

# РЕАЛИЗАЦИЯ И ТЕСТИРОВАНИЕ ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА

## Реализация программно-аппаратного комплекса

В результате реализации курсового проектирования был разработан проект «Программно-аппаратный комплекс «Умный дом». Модуль взаимодействия с пользователями».

### Реализация функциональной части проекта

Для работы с проектом было выделено два пользователя: «Пользователь» и «Контроллер» со своими функциональными возможностями.

Функциональные возможности «Пользователя»:

− включение приборов с помощью пульта управления;

− выключение приборов с помощью пульта управления;

− установка временного освещения для включения/выключения освещения с помощью пульта управления;

− выбор различных сценариев с помощью пульта управления.

Функциональные возможности «Контроллера»:

* определение и вывод с помощь модуля RTC DS 1307 времени;
* определение и вывод с помощью датчика DS18B20 температуры помещения;
* считывание кода операции и выполнения сценариев с помощью ИК-датчика;
* вывод сообщений о выполнении сценариев и включении/выключении приборов. Вывод всех показаний и сообщений осуществляется на дисплей по интерфейсу I2C;
* очистка экрана при помощи пульта управления.

Листинг кода программы представлен в приложении Б.

### Реализация нефункциональной части проекта

Реализация нефункциональной части проекта заключалась в разработке электрической схемы устройства, монтаже электронных компонентов и изготовлении корпуса.

Первичная проверка работоспособности схемы осуществлялась по технологии навесного монтажа с помощью макетных плат и соединительных проводов, входящих в комплект поставки модулей Arduino. После анализа схемы на работоспособность была произведена окончательная сборка с частичным монтажом.

Кроме основного корпуса также был собран меньший корпус, где находиться сам контроллер, размером 100×150×100 мм.

Весь корпус выполнен из материала на пластиковой основе толщиной в 4 мм. В передней стенке вырезаны отверстия для размещения освещения, вентиляции, инфракрасной лампочки и розетки. А также корпус был обклеен пленкой с разработанным дизайном. В заднюю стенку вмонтирован разъем для подключения источника питания. Во второй части корпуса вырезаны отверстия для дисплея, датчика температуры, ИК-датчика и фоторезистора. Одна из боковых стенок меньшей части сделана съемной для обеспечения возможности доступа к электрической схеме с целью просмотра или внесения изменений в схему.

Для питания в боковой стенке основного корпуса вставлен разъем на 220 B, а внутри корпуса был установлен блок питания на 5-9 В.

## Тестирование программно-аппаратного комплекса

Тестирование − это процесс исполнения программной части проекта с целью обнаружения ошибок. Одним из способов изучения поставленного вопроса является исследование стратегии тестирования.

Существует несколько методов тестирования:

* тестирование программ методом «чёрного ящика» (Black box testing);
* тестирование софта методом «белого ящика» (White box).

Тестирование по принципу «белого ящика» (структурное тестирование) применяется, когда известна внутренняя структура программы. Затем исследуются внутренние элементы программы и связи между ними. Данное тестирование характеризуется степенью, в которой тесты выполняют или покрывают логику (исходный [текст](http://baza-referat.ru/%D0%A2%D0%B5%D0%BA%D1%81%D1%82)) программы [5].

Тестирование по принципу «чёрного ящика» (функциональное тестирование) применяется, когда известны функции программы. Затем исследуются работа каждой функции на всей области определения. При тестировании «черного ящика» рассматриваются системные характеристики программ, игнорируется их внутренняя логическая структура. Данный метод необходим для проверки функциональности, производительности, проверки специфики работы на определенных классах входных данных [6].

Таким образом, для разработанного проекта «Умный дом. Модуль взаимодействия с пользователями» тестирование производилось по принципу «чёрного ящика», так как исследовался объект, внутренне устройство, которого неизвестно.

Для проверки разработанной системы необходимо проверить работоспособность проекта:

* проверка на запуск;
* проверка на вывод времени;
* проверка на вывод температуры помещения;
* проверка работы ИК-датчика;
* проверка вывода информационных сообщений;
* проверка установки времени включения/выключения освещения;
* проверка очистки экрана.

Результатом проведения тестирования проекта являются данные, представленные в таблице 3.1. Результат каждого отдельного теста считается не успешным, если тест выполнился без ошибки, и успешным в противном случае.

Таблица 3.1 − Результаты тестирования программы

|  |  |  |  |
| --- | --- | --- | --- |
| Тестовый случай | Выполняемые действия | Фактический результат | Результат теста |
| 1 | 2 | 3 | 4 |
| Проверка на запуск | Подключаем проект к источнику питания | Проект начнет работу | Неуспешно |
| Проверка работы ИК-датчика | Нажимаем на рабочую кнопку пульта | Включение сценария, связанного с нажатой кнопкой | Неуспешно |
| Проверка на вывод времени | Ждем вывода на дисплей | Вывод времени на дисплей | Неуспешно |
| Проверка на вывод температуры помещения | Ждем вывода на дисплей | Вывод температуры помещения на дисплей | Неуспешно |
| Проверка вывода информационного сообщения | Включаем освещение | Вывод на дисплей сообщения, что освещение включено | Неуспешно |
| Проверка вывода информационного сообщения | Включаем вентиляцию | Вывод на дисплей сообщения, что вентиляция включена | Неуспешно |
| Проверка вывода информационного сообщения | Включаем обогрев | Вывод на дисплей сообщения, что обогрев включен | Неуспешно |
| Проверка вывода информационного сообщения | Включаем авто управление | Вывод на дисплей сообщения, что автоуправление включено | Неуспешно |
| Проверка вывода информационного сообщения | Выключаем авто управление | Вывод на дисплей сообщения, что автоуправление выключено | Неуспешно |

Продолжение табл. 3.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Проверка вывода информационного сообщения | Выключаем розетку | Вывод на дисплей сообщения, что розетка включена | Неуспешно |
| Проверка вывода информационного сообщения | Сохраняем установленное время включения освещения | Вывод на дисплей сообщения, что время включения сохранено | Неуспешно |
| Проверка вывода информационного сообщения | Сохраняем установленное время выключения освещения | Вывод на дисплей сообщения, что время выключения сохранено | Неуспешно |
| Проверка вывода информационного сообщения | Включаем вентиляцию и повышаем температуру до 27 градусов | Вывод на дисплей сообщения, что вентиляция включена | Неуспешно |
| Проверка вывода информационного сообщения | Включаем обогрев и понижаем температуру до 15 градусов | Вывод на дисплей сообщения, что обогрев включен | Неуспешно |
| Проверка вывода информационного сообщения | Устанавливаем время включения освещения и дожидаемся данного времени | Вывод на дисплей сообщения, что освещение включено | Неуспешно |
| Проверка вывода информационного сообщения | Устанавливаем время выключения освещения и дожидаемся данного времени | Вывод на дисплей сообщения, что освещение выключено | Неуспешно |
| Проверка установки времени включения/выключения освещения | Нажимаем кнопку настройки времени | Вывод на дисплей настраиваемого времени | Неуспешно |
| Проверка установки времени включения/выключения освещения | Нажимаем кнопку прибавления минут | Вывод на дисплей настраиваемого времени, увеличенного на минуту | Неуспешно |
| Проверка установки времени включения/выключения освещения | Нажимаем кнопку уменьшения минут | Вывод на дисплей настраиваемого времени, уменьшенного на минуту | Неуспешно |

Продолжение табл. 3.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Проверка установки времени включения/выключения освещения | Нажимаем кнопку прибавления часов | Вывод на дисплей настраиваемого времени, увеличенного на час | Неуспешно |
| Проверка установки времени включения/выключения освещения | Нажимаем кнопку уменьшения часов | Вывод на дисплей настраиваемого времени, уменьшенного на час | Неуспешно |
| Проверка установки времени включения/выключения освещения | Нажимаем кнопку прибавления минут после 59 | Вывод на дисплей настраиваемого времени, где количество минут равно нулю | Неуспешно |
| Проверка установки времени включения/выключения освещения | Нажимаем кнопку прибавления часов после 23 | Вывод на дисплей настраиваемого времени, где количество часов равно нулю | Неуспешно |
| Проверка установки времени включения/выключения освещения | Нажимаем кнопку уменьшения минут после нуля | Вывод на дисплей настраиваемого времени, где количество минут равно 59 | Неуспешно |
| Проверка установки времени включения/выключения освещения | Нажимаем кнопку уменьшения часов после нуля | Вывод на дисплей настраиваемого времени, где количество часов равно 23 | Неуспешно |
| Проверка очистки экрана | Нажимаем кнопку очистки экрана | Очистка экрана | Неуспешно |

По результатам проведенного тестирования ошибок в работе устройства выявлено не было, соответственно проект является полностью функционально работоспособным и готовым к эксплуатации.

# ЗАКЛЮЧЕНИЕ

За период работы над проектом «Программно-аппаратный комплекс «Умный дом». Модуль взаимодействия с пользователями» были изучены номенклатура и характеристика разрабатываемого ПО, методы проектирования и среды, используемые для его разработки, предметная область «умный дом».

В ходе реализации курсового проекта были решены следующие задачи:

* изучена предметная область;
* выявлены требования к разрабатываемой системе;
* собрано устройство и написана для него прошивка;
* произведено тестирование устройства;
* разработана необходимая программная документация.

В рамках курсового проектирования была достигнута главная цель, которая заключалась в разработке программно-аппаратного комплекса «Умный дом. Модуль взаимодействия с пользователями». Разработанное устройство является стабильным в работе, имеет незначительный вес и компактные размеры, а также удобный корпус.

Устройство может использоваться широким кругом пользователей без специальной предварительной подготовки.

При необходимости функционал автоматизированной системы может быть расширен за счет добавления wi-fi или bluetooth модуля, а также за счет создания мобильного приложения, с помощью которого можно осуществлять отправку информации на мобильное устройство.

Работа над курсовым проектом позволила приобрести знания в области разработки встраиваемых систем и программирования для платформ Arduino. Были решены все проблемы, которые возникали на различных этапах разработки.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Программа и методика испытаний. Требования к содержанию и оформлению: ГОСТ 19.301-2001 ЕСПД – Минск: Изд-во стандартов, 1982.
2. Руководство оператора. Требования к содержанию и оформлению: ГОСТ 19.505-79 ЕСПД – Минск: Изд-во стандартов, 1982.
3. Техническое задание. Требования к содержанию и оформлению: ГОСТ 19.201-78 ЕСПД – М.: Издательство стандартов, 1987.
4. Требования к дипломному (курсовому) проекту: СТП ПГЭК 2.01 – УО «ПГЭК».
5. Тестирование по стратегии белого ящика [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/Тестирование_по_стратегии_белого_ящика>. – Дата доступа: 10.06.2023.
6. Тестирование по стратегии чёрного ящика [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/Тестирование_по_стратегии_чёрного_ящика>. – Дата доступа: 10.06.2023.
7. Программно-аппаратный комплекс [Электронный ресурс] − Режим доступа: <https://fb.ru/article/251563/chto-takoe-pak-i-dlya-chego-on-nujen?ysclid=livpjplk80246096631> − Дата доступа: 29.04.2023.

8. «Умный дом» [Электронный ресурс] − Режим доступа: <https://gigaboom.ru/articles/raznye-stati/komu-podhodit-i-kak-rabotaet-sistema-umnyi-dom/?ysclid=lix35tsdmu348804562> − Дата доступа: 01.05.2023.

9. Arduino Uno [Электронный ресурс] − Режим доступа: <http://arduino.on.kg/Arduino-UNO?ysclid=lix3quq6ua412539795> − Дата доступа: 15.05.2023.

10. Использование «умного дома» [Электронный ресурс] − Режим доступа: <https://dzen.ru/a/YG7bm1Nd4xyxE8k9> − Дата доступа: 02.05.2023.

11. OLED-дисплей [Электронный ресурс] − Режим доступа: <https://alphapedia.ru/w/OLED> − Дата доступа: 15.05.2023.

12. Инфракрасный пульт [Электронный ресурс] − Режим доступа: <https://led-displays.ru/ir_remote_theory.html?ysclid=lj15qlt4es807294673> − Дата доступа: 15.05.2023.

13. Модуль реального времени [Электронный ресурс] − Режим доступа: https://arduinoplus.ru/chasi-na-arduino/?ysclid=lj165hjkmo26383070 − Дата доступа: 15.05.2023.

14. Датчик температуры [Электронный ресурс] − Режим доступа: https://dzen.ru/a/XywndNlcwBBgAnz3 − Дата доступа: 15.05.2023.

# ПРИЛОЖЕНИЕ A (обязательное) Техническое задание

**А Введение**

Наименование проекта – «Программно-аппаратный комплекс «Умный дом». Модуль взаимодействия с пользователями». Устройство предназначено для автоматизации управления устройствами «умного» дома.

**А1 Основание для разработки**

Основанием для разработки является задание на курсовое проектирование, выданное Долгой Марии Викторовны, утвержденное приказом по УО «Полоцкий государственный экономический колледж» от 20 апреля 2023 года.

**А2 Назначение разработки**

К пользованию системой допускаются лица, ознакомившиеся с руководством по эксплуатации. Специальные требования к пользователю не предъявляются. Система может эксплуатироваться, как и в организациях, так и в помещениях частной формы владения.

**А3 Требования к программе**

**А3.1 Требования к функциональным характеристикам**

Проект должен обеспечивать выполнение следующих функций:

− выбор сценариев устройства при помощи пульта;

− контроль температуры при помощи дисплея;

− контроль времени при помощи дисплея;

− контроль выполнения сценариев с помощью дисплея;

− взаимодействие с пользователем, для установки включения/выключения освещения в определенное время;

− вывод информационных сообщений.

**А3.2 Условия эксплуатации**

Специальные требования по условиям эксплуатации проекта отсутствуют.

**А3.4 Требования к составу и параметрам технических средств**

Для обеспечения бесперебойной работы системы необходим постоянный источник питания в 5V.

**А3.5 Требования к информационной и программной  
совместимости**

Не установлены.

**А3.6 Требования к маркировке и упаковке**

Разработанный проект должен иметь малые габариты, эстетический вид. Устройства управления и электроника должны быть смонтированы в пластиковом корпусе с максимально допустимыми размерами 200×500×400 мм. Одна из стенок должна быть съемной с целью возможного доступа к контроллеру и датчикам.

Компакт диск должен быть промаркирован в соответствии с  СТП ПГЭК 2.01.

**А4 Требования к программной документации**

Программная документация должна содержать пояснительную записку и техническое задание.

Программная документация должна быть оформлена в соответствии с требованиями Стандарта предприятия СТП ПГЭК 2.01 «Программное обеспечение информационных технологий. Требования к дипломному (курсовому) проекту».

**А5 Технико-экономические показатели**

Улучшение технико-экономических показателей достигается за счет простоты использования проекта.

**А6 Стадии и этапы разработки**

Стадии и этапы разработки системы в таблице А.1.

Таблица А.1 – Сроки выполнения индивидуального задания на курсовой проект

|  |  |
| --- | --- |
| Этапы курсового проектирования | Срок выполнения этапа |
| Введение. Описание предметной области | 20.04.2023 |
| Постановка задачи | 28.04.2023 |
| Разработка технического задания | 06.05.2023 |
| Разработка навигационной схемы приложения | 13.05.2023 |
| Проектирование программного обеспечения | 20.05.2023 |
| Реализация и тестирование программного обеспечения | 05.06.2023 |
| Разработка программной документации | 09.06.2023 |
| Оформление пояснительной записки, графической части и приложений | 13.06.2023 |

**А7 Порядок контроля и приемки**

**А7.1 Порядок контроля**

Контроль выполнения осуществляется преподавателем дисциплины «Основы алгоритмизации и программирования».

**А7.2 Порядок защиты**

Защита осуществляется преподавателем дисциплины «Основы алгоритмизации и программирования».

**А7.3 Срок защиты**

Срок защиты 21.06.2023

**А8 Примечание**

В процессе выполнения работы возможно уточнение отдельных требований технического задания по взаимному согласованию преподавателя и исполнителя.

# ПРИЛОЖЕНИЕ Б (обязательное) Листинг кода программы

**Листинг Б – Модуль работы контроллера ymniu\_dom.ino**

/\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Модуль ymniu\_dom.ino

гр. П21

Разработчик: Долгая М.В.

Модифицирован: 2 июня 2023

--------------------------------------------------------------

Модуль работы контроллера

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*/

#include "U8glib.h"

#include <OneWire.h>

#include <DallasTemperature.h>

#include <IRremote.hpp>

#include "iarduino\_RTC.h"

#define ONE\_WIRE\_BUS 2

#define IR\_RECEIVE\_PIN A3

#define PIN\_LED2 6

#define PIN\_LED3 7

#define PIN\_LED1 4

#define PIN\_LED4 5

#define PIN\_PHOTO\_SENSOR A0

#define LINE\_MAX 128

#define ROW\_MAX 64

OneWire oneWire(ONE\_WIRE\_BUS);

iarduino\_RTC time(RTC\_DS1307);

DallasTemperature sensors(&oneWire);

U8GLIB\_SSD1306\_128X64 display(13, 11, 10, 9, 8);

uint8\_t screen[ROW\_MAX][LINE\_MAX];

byte autoLighting = 0;

byte autoVentilation = 0;

byte autoHeating = 0;

byte enablingTime = 0;

byte printTime = 0;

String Hour = "00";

String Minut = "00";

byte inputMinut = 0;

byte inputHour = 0;

void clear\_screen(void)

{

uint8\_t i, j;

for( i = 0; i < ROW\_MAX; i++ )

for( j = 0; j < LINE\_MAX; j++ )

screen[i][j] = 0;

}

void drawTime(String hour, String minut)

{

  display.firstPage();

  do {

display.setFont(u8g\_font\_unifont);

display.setPrintPos(20, 15);

display.print("Time Lamp: ");

display.setPrintPos(40, 35);

display.print(hour + ":" + minut);

} while( display.nextPage() );

}

void setup()

{

pinMode(4,OUTPUT);

digitalWrite(4,HIGH);

time.begin();

IrReceiver.begin(IR\_RECEIVE\_PIN);

pinMode(PIN\_LED1, OUTPUT);

pinMode(PIN\_LED2, OUTPUT);

pinMode(PIN\_LED3, OUTPUT);

digitalWrite(PIN\_LED1, HIGH);

digitalWrite(PIN\_LED2, HIGH);

digitalWrite(PIN\_LED3, HIGH);

display.firstPage();

  do {

    display.setFont(u8g\_font\_unifont);

    clear\_screen();

  } while( display.nextPage() );

}

void loop()

{

if (IrReceiver.decode())

{

IrReceiver.resume();

if(IrReceiver.decodedIRData.command == 20)

{

display.firstPage();

do {

display.setFont(u8g\_font\_unifont);

display.setPrintPos(40, 35);

display.print("Hello!");

} while( display.nextPage() );

enablingTime = 0;

autoLighting = 0;

autoVentilation = 0;

autoHeating = 0;

printTime = 0;

}

if(IrReceiver.decodedIRData.command == 23)

{

 display.firstPage();

 do {

    display.setFont(u8g\_font\_unifont);

    clear\_screen();

  } while( display.nextPage() );

  printTime = 0;

}

if(IrReceiver.decodedIRData.command == 1)

{

  digitalWrite(PIN\_LED2, LOW);

  enablingTime = 0;

  printTime = 0;

  display.firstPage();

do {

display.setFont(u8g\_font\_unifont);

display.setPrintPos(5,35);

display.print("Lighting is on");

} while( display.nextPage() );

}

if(IrReceiver.decodedIRData.command == 2)

{

  enablingTime = 0;

  autoLighting = 0;

  printTime = 0;

  digitalWrite(PIN\_LED2, HIGH);

  display.firstPage();

  do {

    display.setFont(u8g\_font\_unifont);

    clear\_screen();

  } while( display.nextPage() );

}

if(IrReceiver.decodedIRData.command == 15)

{

  printTime = 0;

  drawTime(Hour, Minut);

}

if(IrReceiver.decodedIRData.command == 16)

{

if(inputHour != 23)

{

inputHour += 1;

}

else

{

inputHour = 0;

}

if(inputMinut < 10)

{

Minut = "0" + String(inputMinut);

}

else

{

Minut = String(inputMinut);

}

if(inputHour < 10)

{

Hour = "0" + String(inputHour);

}

else

{

Hour=String(inputHour);

}

drawTime(Hour, Minut);

}

if(IrReceiver.decodedIRData.command == 17)

{

if(inputHour != 0)

{

inputHour -= 1;

}

else

{

inputHour = 23;

}

if(inputMinut < 10)

{

Minut = "0" + String(inputMinut);

}

else

{

Minut = String(inputMinut);

}

if(inputHour < 10)

{

Hour = "0" + String(inputHour);

}

else

{

Hour = String(inputHour);

}

drawTime(Hour, Minut);

}

if(IrReceiver.decodedIRData.command == 18)

{

if(inputMinut != 59)

{

inputMinut += 1;

}

else

{

inputMinut = 0;

}

if(inputMinut < 10)

{

Minut = "0" + String(inputMinut);

}

else

{

Minut = String(inputMinut);

}

if(inputHour < 10)

{

Hour = "0" + String(inputHour);

}

else

{

Hour = String(inputHour);

}

drawTime(Hour, Minut);

}

if(IrReceiver.decodedIRData.command == 19)

{

if(inputMinut != 0)

{

inputMinut -= 1;

}

else

{

inputMinut = 59;

}

if(inputMinut < 10)

{

Minut = "0" + String(inputMinut);

}

else

{

Minut = String(inputMinut);

}

if(inputHour < 10)

{

Hour = "0" + String(inputHour);

}

else

{

Hour = String(inputHour);

}

drawTime(Hour, Minut);

}

if(IrReceiver.decodedIRData.command == 21)

{

enablingTime = 2;

}

if(IrReceiver.decodedIRData.command == 11)

{

sensors.requestTemperatures();

printTime = 0;

display.firstPage();

do {

display.setFont(u8g\_font\_unifont);

display.setPrintPos(20, 15);

display.print("Temperature ");

display.setPrintPos(40, 35);

display.print(sensors.getTempCByIndex(0));

display.drawCircle(82, 27, 2);

display.print(" C");

} while( display.nextPage() );

}

if(IrReceiver.decodedIRData.command == 26)

{

autoLighting = 1;

autoHeating = 1;

autoVentilation = 1;

printTime = 0;

display.firstPage();

do {

display.setFont(u8g\_font\_unifont);

display.setPrintPos(20, 15);

display.print("Auto control");

display.setPrintPos(40, 35);

display.print("is on");

} while( display.nextPage() );

}

if(IrReceiver.decodedIRData.command == 14)

{

autoLighting = 0;

autoHeating = 0;

autoVentilation = 0;

printTime = 0;

display.firstPage();

do {

display.setFont(u8g\_font\_unifont);

display.setPrintPos(20, 15);

display.print("Auto control");

display.setPrintPos(40, 35);

display.print("is off");

} while( display.nextPage() );

}

if(IrReceiver.decodedIRData.command == 3)

{

digitalWrite(PIN\_LED3, LOW);

  enablingTime = 0;

  autoHeating = 0;

  printTime = 0;

  display.firstPage();

do {

display.setFont(u8g\_font\_unifont);

display.setPrintPos(35, 15);

display.print("Heating");

display.setPrintPos(40, 35);

display.print("is on");

} while( display.nextPage() );

}

if(IrReceiver.decodedIRData.command == 4)

{

enablingTime = 0;

autoHeating = 0;

printTime = 0;

digitalWrite(PIN\_LED3, HIGH);

display.firstPage();

do {

    display.setFont(u8g\_font\_unifont);

    clear\_screen();

  } while( display.nextPage() );

}

if(IrReceiver.decodedIRData.command == 5)

{

digitalWrite(PIN\_LED1, LOW);

enablingTime = 0;

  autoVentilation = 0;

  printTime = 0;

  display.firstPage();

do {

display.setFont(u8g\_font\_unifont);

display.setPrintPos(20, 15);

display.print("Ventilation");

display.setPrintPos(40, 35);

display.print("is on");

} while( display.nextPage() );

}

if(IrReceiver.decodedIRData.command == 6)

{

enablingTime=0;

autoVentilation=0;

printTime=0;

digitalWrite(PIN\_LED1, HIGH);

display.firstPage();

do {

    display.setFont(u8g\_font\_unifont);

    clear\_screen();

  } while( display.nextPage() );

}

if(IrReceiver.decodedIRData.command == 7)

{

digitalWrite(PIN\_LED4, LOW);

enablingTime = 0;

printTime = 0;

display.firstPage();

do {

display.setFont(u8g\_font\_unifont);

display.setPrintPos(20, 15);

display.print("The socket");

display.setPrintPos(40, 35);

display.print("is on");

} while( display.nextPage() );

}

if(IrReceiver.decodedIRData.command == 8)

{

enablingTime = 0;

printTime = 0;

digitalWrite(PIN\_LED4, HIGH);

display.firstPage();

do {

    display.setFont(u8g\_font\_unifont);

    clear\_screen();

  } while( display.nextPage() );

}

if(IrReceiver.decodedIRData.command == 31)

{

autoLighting = 2;

display.firstPage();

do {

display.setFont(u8g\_font\_unifont);

display.setPrintPos(20, 15);

display.print("Time saved");

display.setPrintPos(40, 35);

display.print("Enabling");

} while( display.nextPage() );

}

if(IrReceiver.decodedIRData.command == 30)

{

autoLighting = 3;

display.firstPage();

do {

display.setFont(u8g\_font\_unifont);

display.setPrintPos(25, 15);

display.print("Time saved");

display.setPrintPos(40, 35);

display.print("Shutdown");

} while( display.nextPage() );

}

if(IrReceiver.decodedIRData.command == 24)

{

 printTime = 1;

}

}

sensors.requestTemperatures();

if(autoHeating == 1)

{

if(sensors.getTempCByIndex(0) <= 15)

{

digitalWrite(PIN\_LED3, LOW);

display.firstPage();

do {

display.setFont(u8g\_font\_unifont);

display.setPrintPos(35, 15);

display.print("Heating");

display.setPrintPos(40 ,35);

display.print("is on");

} while( display.nextPage() );

}

if(sensors.getTempCByIndex(0) > 15)

{

  digitalWrite(PIN\_LED3, HIGH);

}

}

if(autoVentilation==1)

{

if(sensors.getTempCByIndex(0) >= 27)

{

  digitalWrite(PIN\_LED1, LOW);

  display.firstPage();

do {

display.setFont(u8g\_font\_unifont);

display.setPrintPos(20, 15);

display.print("Ventilation");

display.setPrintPos(40, 35);

display.print("is on");

} while( display.nextPage() );

}

if(sensors.getTempCByIndex(0) < 27)

{

  digitalWrite(PIN\_LED1, HIGH);

}

}

if(autoLighting==1)

{

int val = analogRead(PIN\_PHOTO\_SENSOR);

  if (val <= 500)

{

digitalWrite(PIN\_LED2, LOW);

}

else

{

digitalWrite(PIN\_LED2, HIGH);

 }

}

if(enablingTime == 2)

{

  if(String(time.gettime("H") ) == Hour && String( time.gettime("i") ) ==

Minut && String(time.gettime("s") ) == "00")

{

    if(autoLighting == 2)

{

digitalWrite(PIN\_LED2, LOW);

display.firstPage();

do {

display.setFont(u8g\_font\_unifont);

display.setPrintPos(5, 35);

display.print("Lighting is on");

} while( display.nextPage() );

}

else

{

digitalWrite(PIN\_LED2, HIGH);

display.firstPage();

do {

display.setFont(u8g\_font\_unifont);

display.setPrintPos(5, 35);

display.print("Lighting is off");

} while( display.nextPage() );

}

}

}

if(printTime == 1)

{

display.firstPage();

do {

display.setFont(u8g\_font\_unifont);

display.setPrintPos(25, 20);

display.print(time.gettime("d-m-Y") );

display.setPrintPos(25, 40);

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

display.setPrintPos(50, 60);

display.print(time.gettime("D") );

} while( display.nextPage() );delay(1);

}

}

# ПРИЛОЖЕНИЕ В (обязательное) Содержание электронного носителя

На прилагаемом электронном носителе информации расположены следующие директории и файлы:

1. Директория «*DOC*» – файл «*П21\_Долгая\_М.В.\_Проект\_  
   «Программно\_аппаратный\_комплекс\_Умный\_дом.\_Модуль\_взаимодейств-ия\_с\_пользователями.docx»* представляет собой текстовый документ с пояснительной запиской и графическим материалом.
2. Директория «PRG» содержит файл с прошивкой.
3. Файл «*Readme.txt*» содержит описание содержания электронного носителя.