# Министерство образования и науки Российской Федерации

# Федеральное государственное бюджетное образовательное учреждение

# высшего профессионального образования

# «Ижевский государственный технический университет имени М. Т. Калашникова»

# Кафедра «Программное обеспечение»

Отчет по лабораторной работе №2

по дисциплине: «Конструирование ПО»

на тему: «Автоматизированная система управления микроклиматом теплиц»

Выполнил

студент гр. Б08-191-2 Э.Ф.Ахмерова

Принял В.Г. Власов

Ижевск

2016

ТРЕБОВАНИЯ

1. Блок управление датчиками.
   1. Сбор показаний датчиков с заданной периодичностью (Т).
   2. Принятие решения в соответствии с настройками.
      1. Если данные с датчиков вышли за границы (мин и макс) настроек, то включается соответствующее устройство. Устройство работает, пока не будет достигнуто среднее значение настроек.
   3. Включение/настройка других устройств.
   4. Отправка данных с заданной периодичностью (Т).
      1. Если Т опрашивания датчиков меньше Т отправки данных, то данные с датчиков усредняются. Т опрашивания датчиков не может быть больше Т отправки данных.
      2. Данные о состоянии устройства отправляются при смене состояния (включении/выключении).
   5. Принятие данных (настроек) из базы данных
   6. Обновление своих настроек.
2. БД, таблицы:

* Датчики (ИД, название)
* Устройства (ИД, название)
* Настройки (ИД, название)
* Показания датчиков (ИД датчика, значение, время)
* Состояния устройств (ИД устр., состояние, время)
* Настройки (ИД настройки, значение, ИД датчика)
* Предупреждения (Название, время).

1. Пользовательское приложение
   1. При подключении к интернету загрузка данных из бд.
   2. Установка необходимых настроек, отправка настроек в бд.
   3. Составление отчетов.
   4. Просмотр текущего состояния.
   5. Запрос кода подтверждения управляющего.

1. ВВЕДЕНИЕ

1.1 Назначение системы

Данная система необходима для увеличения производительности теплиц, путем автоматизации процесса выращивания сельхозпродукции, уменьшения влияния человеческого фактора, мониторинга микроклимата.

1.2 Область применения системы

Автоматизированная система управления микроклиматом применяется в теплицах или тепличных комплексах для обеспечения оптимального микроклимата для различных видов культур. Система управления состоит из набора датчиков, передающих различные показания окружающей среды; набора устройств, осуществляющих регулировку и корректировку климата, посева и т.д., и управляющего блока. Помимо системы управления существует также устройство-сервер, принимающий какие-либо показания и сохраняющий их в базу данных, и устройство-монитор, который отображает полученные от сервера данные, формирует статистику.

Основные функциональные возможности системы:

1. Считывание показаний датчиков
2. Включение, настройка других устройств
3. Сохранение необходимых показаний в базу данных
4. Отображение данных, формирование статистики, графиков и т.д.

1.3 Определения, акронимы, аббревиатуры

Работник – человек, осуществляющий основные операции в теплице, может просматривать состояние системы.

Агроном – человек, осуществляющий задание, редактирование настроек регулировки климата, может просматривать состояние системы.

Блок управления (БУ) – устройство-микроконтроллер, осуществляющее управление датчиками и устройствами.

1.4 Ссылки

1.5 Обзор системы

Система делится на два главных модуля: БУ и пользовательский интерфейс, которые связаны между собой удаленным хранилищем (базой данных).

БУ собирает показания датчиков с заданной периодичностью, принимает решение в соответствии с настройками, включает, настраивает другие устройства, отправляет данные в базу данных, принимает настройки из базы данных.

Пользовательское приложение загружает данные из БД, составляет отчеты, отправляет настройки в БД.

2. ОБЩЕЕ ОПИСАНИЕ СИСТЕМЫ

2.1 Системный контекст

2.2 Режимы и состояния системы

Система может находиться в следующих состояниях:

1. Состояния блока управления.

* ожидание опроса датчиков;
* опрос датчиков;
* проверка соответствия показаний датчиков настройкам;
* включение исполнительных устройств;
* передача данных в базу данных;
* получение настроек из базы данных

1. Состояния пользовательского приложения.

* загрузка данных из БД;
* передача настроек в БД;
* составление отчета;
* ввод настроек.

2.3 Основные функциональные возможности системы



2.4 Основные условия системы

Блок управления должен иметь постоянный доступ к интернету.

2.5 Основные ограничения системы

2.6 Характеристики пользователя

1. Работник. Просматривает состояние климата в теплице.
2. Агроном. Имеет возможность изменять настройки микроклимата и также просматривать состояние климата в теплице.

2.7 Допущения и зависимости

2.8 Оперативные сценарии

Система предусматривает следующий сценарий. БУ опрашивает датчики, принимает решение о необходимости включить соответствующие исполнительные устройства. Также он отправляет данные в базу данных с заданной периодичностью и получает новые настройки. Работник запускает приложение с интернет подключением и запрашивает отчет на заданный период, либо текущее состояние. Приложение загружает данные из БД и составляет отчет. Агроном авторизуется в системе с помощью заданного кода подтверждения и может устанавливать новые настройки и параметры микроклимата.

ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ, УСЛОВИЯ И ОГРАНИЧЕНИЯ СИСТЕМЫ

* 1. Физические
     1. Конструкция

Блок управления состоит из микроконтроллера Arduino и подключенных к нему датчиков и исполнительных устройств. БУ подключается к интернету с помощью сетевого кабеля.

3.1.2 Износостойкость

3.1.3 Адаптируемость

Блок управления позволяет увеличивать количество датчиков и исполнительных устройств.

3.1.4 Условия окружающей среды

3.2 Рабочие характеристики системы

3.3 Защита системы

База данных имеет защищенное соединение. Изменение настроек возможно только при ввода кода подтверждения.

3.4 Информационный менеджмент

3.5 Работа системы

3.5.1 Эргономика системы

3.5.2 Эксплуатационная технологичность

3.5.3 Надежность системы

Система защищена от постороннего ввода настроек.

3.6 Стратегия и регулирование

3.7 Устойчивость жизненного цикла системы

КЛАССЫ ПРОЕКТИРОВАНИЯ



Интерфейсы:

1. Интерфейс котроллера для получения служб.

interface Controller

{

DataBaseConnector GetDataBaseConnector();

SensorManager GetSensorManager();

}

1. Интерфейс для соединения с базой данных.

interface ControllerDataBaseConnector

{

void SendSensorData(Sensor sensor);

void SendDeviceData(ImplDevice device);

List<Setting> GetSetting();

}

1. Интерфейс для управления датчиками и устройствами.

interface SensorManager

{

List<Sensor> ReadSensorsData();

void AnalyseSensorsData(List<Sensor> sensorList);

void Implement(List<ImplDevice> deviceList);

}

1. Интерфейс приложения для получения служб.

interface Application

{

DataBaseConnector GetDataBaseConnector();

DataHandler GetDataHandler();

SettingManager GetSettingManager();

}

1. Интерфейс для соединения с базой данных.

interface DataBaseConnector

{

List<Sensor> GetSensorsData(long timeBegin, long timeEnd);

List<ImplDevice> GetDevicesData(long timeBegin, long timeEnd);

void SendSettings(List<Setting> settings);

}

1. Интерфейс для обработки данных.

interface DataHandler

{

void GenerateClimateState(List<Sensor> sensors, List<ImplDevice> devices);

void GenerateCurrentState(Sensor sensor, ImplDevice device);

}

1. Интерфейс для изменения настроек.

interface SettingManager

{

List<Setting> InputSettings();

int InputCode();

bool IsRightCode(int code);

}

UML ДИАГРАММА КЛАССОВ КОДИРОВАНИЯ



Классы кодирования программы соответствуют интерфейсам, по которым построены классов проектирования. Для системы также разработаны классы, реализующие интерфейсы, и другие вспомогательные классы классы:

1. Класс Датчик.

class Sensor

{

public int id;

public string name;

public long time;

public double value;

}

1. Класс Исполнительное устройство.

class ImplDevice

{

public int id;

public string name;

public long time;

public bool state;

}

1. Класс Настройка.

class Setting

{

public int id;

public string name;

public double minValue;

public double maxValue;

}

1. Класс ApplicationImpl – реализация интерфейса Application.
2. Класс DataBaseConnectorImpl – реализация интерфейса DataBaseConnector.
3. Класс DataHandlerImpl – реализация интерфейса DataHandler.
4. Класс SettingManagerImpl – реализация интерфейса SettingManager.
5. Класс ControllerImpl – реализация интерфейса Controller.
6. Класс ControllerDataBaseConnectorImpl – реализация интерфейса ControllerDataBaseConnector.
7. Класс SensorManagerImpl – реализация интерфейса SensorManager.