

## **РАЗРАБОТКА ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСА «НОМЕВРАИН» ДЛЯ УПРАВЛЕНИЯ И МОНИТОРИНГА IoT-СИСТЕМ**

Листопадов С., Сицко В.А., Хренков Д.Д.

*Институт информационных технологий БГУИР, г. Минск, Республика Беларусь*

*[s.listopadov@bsuir.by](mailto:s.listopadov@bsuir.by), [v.sitsko@bsuir.by](mailto:v.sitsko@bsuir.by)*

Данная статья посвящена разработке программно-аппаратного комплекса «HomeBrain» для управления и мониторинга IoT-систем. Данный комплекс позволяет автоматизировать контроль и управление различными процессами, а также может использоваться для создания комфортной среды для лиц с особыми потребностями. Например, контроль за домом или облегчение выполнения бытовых задач.

Ключевые слова: PHP; Symfony; JavaScript; Knockout.js; 3. Chart.js; MySQL; Redis; Arduino; C++; IoT; JWT.

В настоящее время IoT-системы являются одним из наиболее актуальных и востребованных направлений в области информационных технологий. Данные системы позволяют осуществлять контроль над определенным участком с использованием широкого спектра датчиков и управлять различными устройствами, такими как освещение, отопление, кондиционирование, системы безопасности, и др.

Актуальность проекта – IoT-системы становятся все более распространенными в современном мире. Они становятся все более доступными и дешевыми. Таким образом,

пользователи могут применить данные системы в областях ранее не доступных из-за высоких издержек приобретения оборудования и его обслуживания. Например, контроль за домом или облегчение выполнения бытовых задач.

Цель проекта – разработка программно-аппаратного комплекса «HomeBrain». Проект «HomeBrain» должен позволить пользователю устанавливать систему управления и контроля в различных областях с минимальными затратами.

«HomeBrain» – это программно-аппаратный комплекс, состоящий из веб-приложения, выступающего в роли интерфейса системы, сервера обработки входящих и исходящих запросов, связанных с IoT-системой, и логического модуля на базе Arduino, выступающего посредником между издателями и сервером обработки данных.

Издатель – датчик (сенсор) или устройство (девайс), подключенный к логическому модулю.

Для использования комплекса необходимо выполнить следующие шаги:

Зарегистрироваться на сайте проекта.

Зарегистрировать логический модуль. К данному модулю нужно подключить необходимые датчики и устройства.

В программном коде логического модуля необходимо указать данные зарегистрированного логического модуля (логин и пароль) и перепрограммировать логический модуль.

Создать на сайте проекта локацию. Для удобства восприятия информации Издатели сгруппированы в локации.

На странице локации создать издатель. Для корректной работы издателя необходимо указать настройки датчика.

Перезагрузить логический модуль.

Логический модуль, используя зарегистрированный логин и пароль, после запуска автоматически обращается к серверу веб-приложения для загрузки информации об издателях. Общий процесс инициализации логического блока выглядит следующим образом:

Логический блок отправляет POST-запрос на авторизацию серверу загрузки данных издателей. В качестве параметров запрос отправляет username и password, зарегистрированного логического блока.

После успешной авторизации логический блок принимает JSON Web Token (JWT) [1]. Данный токен будет необходим для всех взаимодействий с сервером обработки данных.

Блок обращается к серверу загрузки данных для получения информации об издателях всех видов.

После получения данных об издателях происходит инициализация сенсоров и девайсов.

Начинается циклический опрос датчиков согласно переданным настройкам. Одновременно блок ожидает команд для устройств.

Программа логического блока содержит набор функций для работы с различными типами издателей-датчиков (аналоговые, цифровые, I2C, USART и др.). В случае необходимости добавления своего обработчика издателя это можно сделать следующим образом:

Создайте строковое название вашей функции.

Создайте указатель на вашу функцию-обработчик.

Воспользуйтесь методом addHandler сервис-контейнера и в качестве параметра передайте указатель на функцию и строковое название.

Теперь данный обработчик доступен в качестве обработчика данных для любого выбранного датчика.

Полученные от датчиков данные группируются в отчет, передаваемый серверу обработки данных. Данные валидируются согласно настройкам и при необходимости на логический блок отправляется команда на активацию функции реакции на опасную ситуацию (при ее указании в настройках). Периодически поступившие данные сохраняются

в реляционную базу данных для формирования статистики, которую можно увидеть на веб-сайте (рис.1).

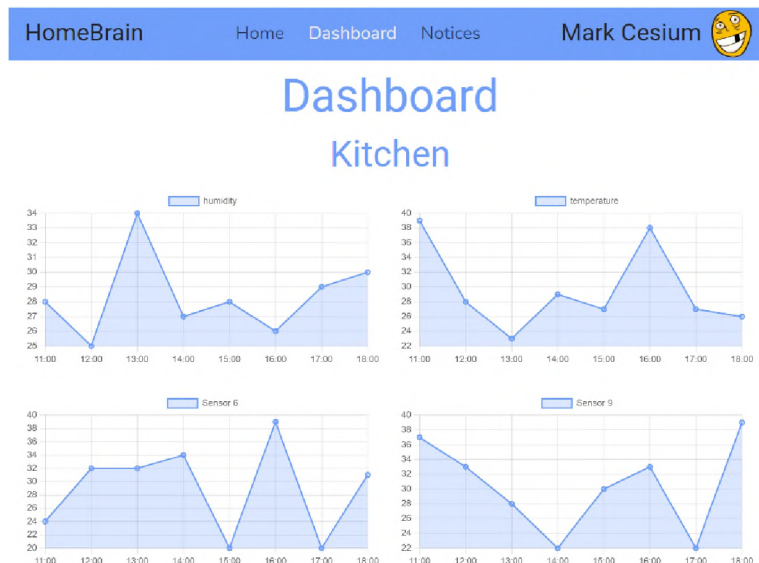


Рисунок 1 – Страница графиков «HomeBrain»

По умолчанию статистика с издателя (тип датчик) предоставляется за 8 часов. В интерактивном режиме данные поступают на страницу локаций (рис. 2).

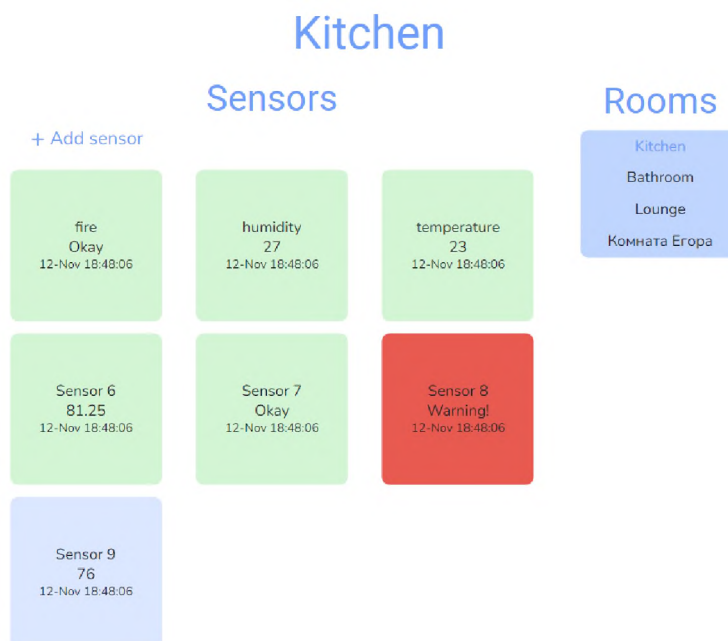


Рисунок 2. – Страница локаций «HomeBrain». Раздел «Sensors»

На странице локаций у пользователя два раздела: сенсоры и девайсы. Карточки сенсоров содержат в себе название, значение, выводимое в зависимости от настроек, дату последнего обновления значения. Также карточка подсвечивается определенным цветом, в зависимости от прохождения условия валидации значений, выставляемых пользователем. Карточки девайсов содержат в себе название девайса, поле ввода значения либо переключатель, в зависимости от настройки, кнопку отправить (рис. 3).

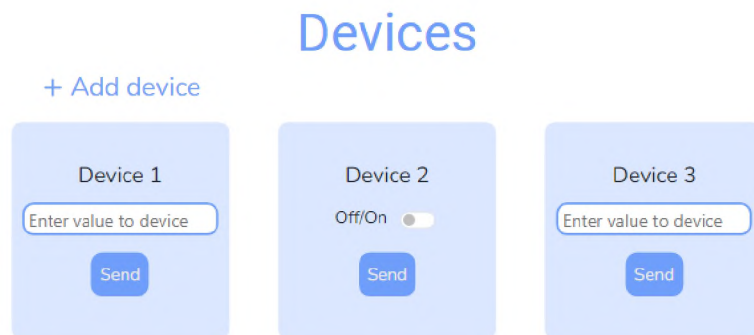


Рисунок 3 – Страница локалии «HomeBrain». Раздел «Devices»

Настройки издателя подразделяются на несколько типов: activation и validation.

Первая группа настроек необходима для настройки работы вашей издателя:

пин на вход – номер пина, с которого необходимо считывать значения (в системе – pin\_i, в HomeBrain – pin input);

пин на выход – номер пина, на который необходимо подать напряжение (в системе – pin\_o, в HomeBrain – pin output);

хэндлер – функция, обрабатывающая приходящие значения с платы перед отправкой (в системе – handler, в HomeBrain – handler);

время работы девайса – время, в течение которого будет работать девайс (в системе – d\_rt, в HomeBrain – device runtime);

пин девайса на выход – пин, на который необходимо подать напряжение при срабатывании «опасного» хэндлера (в системе d\_pin\_o, в HomeBrain – device pin out);

период опроса сенсора – задержка между запросами к сенсору за получением данных (в системе – srd, минимальное значение – 100 мс, в HomeBrain – sensor request delay).

Вторая группа настроек необходима для валидации значений, отправляемых издателем. Валидация доступна по следующим сценариям:

больше чем (в HomeBrain – greater than) – для прохождения валидации значение датчика должно быть больше, чем указанное в сценарии;

меньше чем (в HomeBrain – less than) – для прохождения валидации значение датчика должно быть меньше, чем указанное в сценарии;

равно (в HomeBrain – equals) – для прохождения валидации значение датчика должно быть равно указанному в сценарии;

не равно (в HomeBrain – not equals) – для прохождения валидации значение датчика должно быть не равно указанному в сценарии;

в промежутке (в HomeBrain – values range) – для прохождения валидации значение датчика должно быть между начальным и конечным числами, которые указаны в сценарии;

danger reaction (реакция на опасность) функция, которая будет активирована при не прохождении валидации (в HomeBrain – danger reaction).

Валидация работает по следующему принципу: если значение, присылаемое датчиком, не удовлетворяет сценарию – сервер определяет показания как не валидные, интерфейс подсвечивает блок датчика красным цветом, а логический модуль получает команду на активацию реакции на данное событие. При прохождении валидации блок будет подсвечен зеленым. В случае отсутствия настроек валидации блок отображается синим.

Для создания программно-аппаратного комплекса были использованы следующие технологии:

MySQL – самая популярная в мире база данных с открытым исходным кодом. По данным DB-Engines, MySQL занимает второе место [2] по популярности после Oracle Database. На MySQL работают многие наиболее посещаемые приложения, включая Facebook, Twitter, Netflix, Uber, Airbnb, Shopify и Booking.com.

Knockout – свободный JavaScript каркас веб-приложений, реализующий Model-View-View-Model шаблон с образцами [3].

Chart.js – JavaScript библиотека для построения графиков и диаграмм;

Symfony – PHP – фреймворк с открытым исходным кодом, предназначенный для разработки веб-сайтов с использованием архитектурной модели MVC.

Redis – нереляционная резидентная СУБД, хранящая данные в виде пар «ключ-значение».

Arduino – комплекс аппаратно-программных средств построения и прототипирования простых систем, моделей и экспериментов в области электроники, автоматике, автоматизации процессов и робототехники.

В результате реализации проекта был разработан программно-аппаратный комплекс «HomeBrain» для управления и мониторинга IoT-системы.

В ходе реализации комплекса были выполнены следующие задачи:

разработано программное обеспечение для логического блока;

разработаны серверные части комплекса для обработки входных данных и выходных данных логического модуля и веб-приложения;

разработан интерфейс веб-приложения, учитывающий особенности дизайна.

Проект находится на завершающей стадии тестирования и доработки перед выпуском альфа-версии. В дальнейшем планируется опытная эксплуатация и выпуск потребительской версии продукта.

### Литература

1. Веб-сайт стандарта JWT [Электронный ресурс]. – Режим доступа: <https://jwt.io/>. – Дата доступа: 20.11.2023.

2. Информационный веб-сайт DB-Engines [Электронный ресурс]. – Режим доступа: <https://iwt.io/>. – Дата доступа: 20.11.2023.

3. Веб-сайт библиотеки Knockout JS [Электронный ресурс]. – Режим доступа: <https://knockoutjs.com/>. – Дата доступа: 20.11.2023.

## **DEVELOPMENT OF HARDWARE AND SOFTWARE COMPLEX “HOMEBRAIN” FOR MANAGEMENT AND MONITORING OF IOT SYSTEMS**

Listopadov S., Sitsko V.A., Khrenkov D.D.

*Institute of information technologies BSUIR, Minsk, Republic of Belarus*

This article is devoted to the development of the “HomeBrain” hardware and software complex for managing and monitoring IoT systems. This complex allows you to automate the control and management of various processes, and can also be used to create a comfortable environment for people with special needs. For example, monitoring the house or making it easier to complete household tasks.

Keywords: PHP; Symfony; JavaScript; Knockout.js; 3. Chart.js; MySQL; Redis; Arduino; C++; IoT; JWT.