

УДК 004.3

## СОЗДАНИЕ ЭЛЕМЕНТОВ СМАРТ-СИСТЕМ НА ПРИМЕРЕ ЛАБОРАТОРНОГО МАКЕТА УМНОГО ДОМА

**А. Г. САВИЦКИЙ***(Представлено: А. О. ЛУКЬЯНОВ)*

*В данной статье описаны структура и состав макета умного дома, а также программные технологии, применяемые в реализации. При этом поясняется выбор отдельных компонентов и решений, которые применены. Перечислены сферы применения и преимущества от использования интерактивного макета.*

**Введение.** С развитием технологий наше повседневное окружение становится все более умным и автоматизированным. Умные дома и интернет вещей (IoT) переписывают правила игры, предлагая инновационные решения для повышения комфорта, безопасности и энергоэффективности нашего жилища. Для этих целей в настоящее время активно применяются отладочные платы и комплекты разработчиков, которые не позволяют проследить взаимодействие различных компонентов в составе системы умного дома для выполнения возложенных на неё задач. Целью статьи является описание комплексного подхода к созданию макета домашней автоматизации для использования в процессе обучения.

**Постановка задачи.** В первую очередь необходимо определиться с контроллером для сбора данных, управления и предоставления интерфейса взаимодействия с пользователем. Для этой цели часто используются одноплатные компьютеры и микроконтроллеры. Рассматривая микроконтроллеры можно отметить высокое быстродействие в задачах реального времени и относительно простую управляющую программу для обеспечения взаимодействия всех подключённых устройств. Однако при использовании микроконтроллеров осложняется задача сбора большого количества данных с возможностью их дальнейшего анализа и обработки. Также, при использовании микроконтроллера, могут возникнуть ограничения при создании пользовательского интерфейса.

**Методы исследований.** Применяя одноплатный компьютер в качестве основы для макета появляется возможность организовать полноценный сбор данных с дальнейшей обработкой и визуализацией в интерфейсе пользователя. При этом ресурсов этого типа платформы достаточно для обработки данных с различных датчиков и взаимодействия с исполнительными устройствами без видимой задержки. Также, благодаря использованию высокоуровневых языков программирования для реализации управляющего алгоритма, есть возможность для построения более сложных алгоритмов обработки получаемых данных. Ко всем преимуществам одноплатных компьютеров можно отнести и наличие большого количества портов ввода/вывода, что позволяет напрямую подключать датчики и исполнительные устройства без использования различных конвертеров интерфейсов.

**Результаты и их обсуждение.** В процессе проектирования макета было рассмотрено несколько одноплатных компьютеров, после чего выбор сделан в пользу Raspberry Pi 3B - это мощная, экономичная одноплатная платформа, которая предоставляет широкие возможности для создания проектов в области "Интернета вещей" (IoT), домашней автоматизации и образовательных решений. Благодаря встроенным интерфейсам, таким как GPIO, I2C и SPI, а также поддержке Wi-Fi и Bluetooth. Что позволяет легко подключать различные устройства и управлять ими удалённо через интернет [1].

Следующим этапом проектирования макета был выбор интерфейса взаимодействия контроллера с периферийными узлами: датчиками и исполнительными устройствами. Для этой цели наиболее оптимальным решением являются шины для связи между отдельными компонентами внутри электронных устройств: SPI, I2C и UART. Для обеспечения наименьшего количества соединительных проводов и повышения наглядности структуры макета выбрана шина I2C, так как она обладает значительными преимуществами для внутриплатной связи на короткие расстояния. Во-первых, его гибкость позволяет организовать связь между несколькими ведущими и ведомыми устройствами, что упрощает добавление новых компонентов. Функция адресации упрощает подключение устройств, устраняя необходимость в дополнительных линиях выбора. Протокол также прост в реализации, так как требует всего двух сигнальных линий. Для повышения надёжности, I2C использует механизм подтверждения (ACK/NACK) для обработки ошибок. Кроме того, он адаптируется к устройствам с разной скоростью работы [2].

Однако часть устройств не поддерживают данный интерфейс, используя логический или аналоговый вход/выход. Для взаимодействия с такими устройствами применён расширитель портов и АЦП с I2C интерфейсом. Также допускается подключение этих устройств напрямую к контроллеру через его интерфейс ввода/вывода общего назначения.

Особенность шины I2C заключается в необходимости подтяжки к напряжению питания линий данных и тактирования с использованием резисторов номиналом от 1,5 кОм до 10 кОм. Однако на линии могут одновременно использоваться устройства в виде готовых модулей с уже установленными подтягивающими резисторами. Поэтому для предотвращения чрезмерного снижения сопротивления резисторов подтяжки и организации одновременной работы устройств с различными логическими уровнями используется коммутатор шины на 8 каналов, что позволяет подключить каждое устройство на отдельный канал.

Далее вкратце рассмотрим используемые в макете датчики и исполнительные устройства. Так наряду со стандартным набором датчиков влажности и температуры, атмосферного давления, определения движения и открывания двери присутствуют датчики освещенности, влажности почвы и потребляемой мощности. Исполнительные устройства представлены в виде диммируемых светодиодных лент, адресной светодиодной подсветки, сервопривода открывания двери, а также вентилятора охлаждения и водяного насоса. Помимо этого, макет содержит элементы непосредственного взаимодействия с пользователем: символьный дисплей и матричную клавиатуру, которые продемонстрированы на рис.1. Широкий набор различных узлов системы позволяет максимально охватить взаимодействие компонентов в рамках одной системы и наглядно продемонстрировать происходящие в ней процессы.

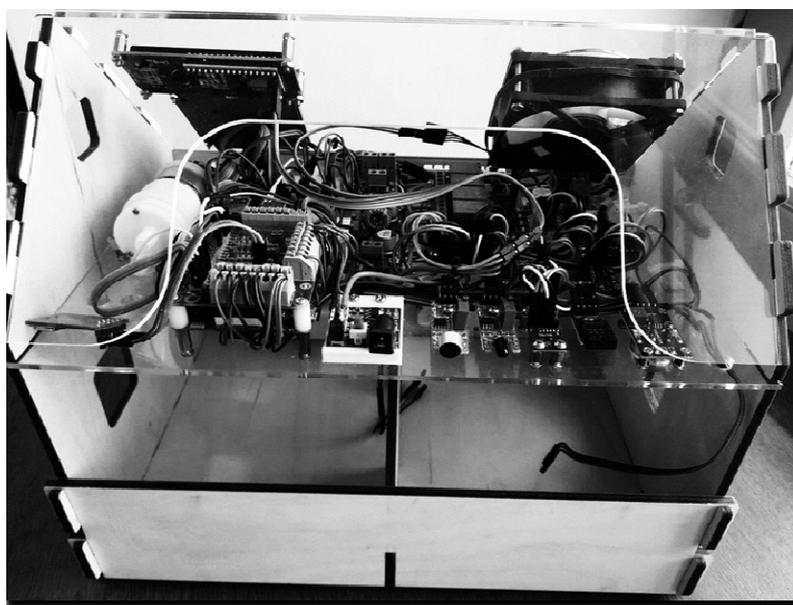


Рисунок 1. – Коммутация компонентов в готовом макете



Рисунок 2. – Символьный дисплей и матричная клавиатура на макете

После проектирования аппаратной части рассмотрим используемые программные компоненты для обеспечения взаимодействия всех элементов системы. Основным языком программирования при создании управляющего алгоритма выбран Python. Этому поспособствовало простота интеграции с различными компонентами системы, такими как база данных, веб интерфейс и аппаратные интерфейсы одноплатного компьютера, а также лёгкость написания достаточно сложных алгоритмов обработки и анализа полученных данных. При этом в наличии достаточно большая база библиотек для взаимодействия с различными аппаратными узлами системы.

Для сбора и последующего отображения данных с датчиков применяется реляционная база данных MariaDB. Преимущества использования данной БД включают её открытый исходный код, что обеспечивает высокую гибкость. Это позволяет пользователям изменять и адаптировать платформу под свои нужды, а также добавлять плагины и расширения для повышения функциональности. MariaDB также предлагает высокую безопасность, включая защиту от SQL-инъекций, шифрование данных и различные методы аутентификации, что важно для защиты чувствительной информации пользователей. Кроме того, она совместима с множеством языков программирования, таких как C++, Java, Python и Ruby, что позволяет разработчикам работать в привычной среде. Из недостатков следует обратить внимание на снижение производительности при выборке большого количества элементов на одноплатных платформах вследствие ограничения пропускной способности карт памяти, на которых располагаются все системные файлы [3].

Для реализации веб-интерфейса используется веб-фреймворк Flask, а для наглядного представления данных с датчиков в виде графиков и диаграмм применяется пакет Grafana.

Flask является микро веб-фреймворком, реализованном на Python. Он относится к микрофреймворкам, так как не требует специальных инструментов и библиотек. Flask не содержит уровня абстракции базы данных, проверки форм и других компонентов, ранее существующих в виде сторонних библиотек. При этом Flask поддерживает расширения, которые добавляют необходимую функциональность. Существуют расширения для объектно-ориентированного отображения баз данных, проверки форм, обработки загрузок, различных технологий аутентификации и так далее [4].

Grafana — это мощная платформа для визуализации данных, которая предлагает множество преимуществ. Во-первых, она поддерживает интеграцию с различными источниками данных, такими как Prometheus, InfluxDB и MySQL, что позволяет пользователям извлекать и отображать данные из различных систем на одной панели управления.

Еще одно важное преимущество Grafana — это возможность настройки панелей управления. Пользователи могут создавать и настраивать панели управления по своему усмотрению, включая различные типы визуализаций, такие как графики, гистограммы и тепловые карты. Это обеспечивает гибкость в представлении данных и позволяет адаптировать интерфейс под конкретные потребности [5].

Также для взаимодействия с макетом разработано мобильное приложение для операционной системы Android. Взаимодействие приложение с макетом организовано через протокол MQTT. Данный протокол обеспечивает надежную передачу данных в реальном времени с минимальной нагрузкой на сеть, что делает его идеальным для использования в умных домах и системах IoT. Благодаря простоте и гибкости, этот протокол поддерживает широкий спектр приложений — от домашней автоматизации до промышленных систем мониторинга [6]. На контроллере запущен MQTT-брокер. Применение данного вида связи позволяет получать информацию об изменении состояния системы и передавать управляющие команды с минимальными задержками.

Выводы. Для учебного макета умного дома был выбран одноплатный компьютер Raspberry Pi 3B, что позволило организовать полноценный сбор данных с дальнейшей обработкой и визуализацией в интерфейсе пользователя. Использование шины I2C для взаимодействия с периферийными устройствами обеспечило наглядность структуры макета и минимизацию количества соединительных проводов.

Применение высокоуровневых языков программирования и реляционной базы данных MariaDB позволило создать сложные алгоритмы обработки данных и обеспечить их хранение. Веб-фреймворк Flask и пакет Grafana обеспечили наглядное представление данных и удобный интерфейс для пользователя. Разработка мобильного приложения для Android с использованием протокола MQTT позволила организовать взаимодействие с макетом с минимальными задержками.

#### ЛИТЕРАТУРА

1. Документация по Raspberry Pi [Электронный ресурс]. – Режим доступа: <https://www.raspberrypi.com/documentation/> – Дата доступа: 24.09.2024.
2. Преимущества и ограничения связи I2C [Электронный ресурс]. – Режим доступа: <https://www.totalphase.com/blog/2016/08/advantages-limitations-i2c-communication/> – Дата доступа: 24.09.2024.

3. Почему MariaDB? Преимущества перед MySQL [Электронный ресурс]. – Режим доступа: <https://www.totalphase.com/blog/2016/08/advantages-limitations-i2c-communication/> – Дата доступа: 27.09.2024.
4. Преимущества использования Python Flask [Электронный ресурс]. – Режим доступа: <https://pythongeeks.org/what-is-python-flask/> – Дата доступа: 26.09.2024.
5. 10 функций Grafana, которые необходимо знать для эффективного мониторинга [Электронный ресурс]. – Режим доступа: <https://opensource.com/article/20/2/grafana-features> – Дата доступа: 26.09.2024
6. Основы MQTT. Полное руководство по MQTT для начинающих и экспертов [Электронный ресурс]. – Режим доступа: <https://www.hivemq.com/mqtt/> – Дата доступа: 28.09.2024.