

3. Несколько подсистем объединяются в решение комплексных задач, дают возможность универсального и гибкого подбора нужного функционала для автоматизации различных процессов, как например «Умный дом», улучшающий и оптимизирующий повседневный быт человека. Такого рода системы являются гибкими, их реализацию полностью контролирует пользователь. Более того, в силу независимости подсистем друг от друга система может функционировать, ещё будучи незаконченной. В случае же выхода из строя какого-либо модуля не придется ремонтировать весь комплекс целиком, достаточно будет заменить вышедшую из строя подсистему.

Данная концепция уже имеет частичную реализацию, например, для контроллеров Arduino разработаны платы-расширения. Будучи выпускаемыми промышленно, они решают не комплексные, а все же одиночные и довольно узкие задачи, имеют универсальные для всех моделей Arduino способы подключения, их комбинации позволяют разрабатывать очень широкий спектр систем. Однако на программном уровне работа с ними происходит через язык программирования C, являющийся низкоуровневым.

Заключение. Таким образом, исходя из проблем современных вариантов производства систем на базе контроллеров и на основе данных из информационных источников была разработана концепция систем-модулей. Она является компромиссом между качеством и доступностью промышленных систем, с одной стороны, и универсальностью сборки «с нуля», с другой стороны, где из обеих сторон используются положительные качества, и позволяющие избежать сложностей работы с учетом недостатков каждой из сторон.

1. Черняк, А.А. Система «Умный дом»/ А.А. Черняк. – Текст : непосредственный // Молодой ученый. – 2020. – С. 51–53. URL: <https://moluch.ru/archive/342/77055> (дата обращения: 21.03.2023).

2. Документация Arduino [Электронный ресурс]. – Режим доступа: <https://www.arduino.cc/> – Дата доступа: 15.03.2023.

ОСОБЕННОСТИ НАПИСАНИЯ ДРАЙВЕРОВ

Гончаренок Е.А., Козаченко А.П.,

*студенты 3 курса Белорусского государственного университета информатики
и радиоэлектроники, г. Минск, Республика Беларусь*

Научный руководитель – Хожевец О.А.

В настоящее время все большее распространение получают микроконтроллеры и подобные им программируемые устройства. Являясь в большинстве своем парой из самого устройства и прилагаемого к нему программного обеспечения, зачастую исходный код которых распространяется открыто, микроконтроллеры заложили основу для создания таких технологий как Internet of Things и Smart House.

Большинство скетчей, написанных для микроконтроллеров созданы для автономной работы платы, все же существует и используется возможность разрабатывать микроконтроллеры, в своей работе тесно взаимодействующие с ПК или другими устройствами. И если при работе с такими распространенными платами, как Arduino, Raspberry, STM-32 уже реализовано большое количество драйверов, то при разработке специфичных устройств на основе микроконтроллеров для непосредственного взаимодействия с ПК, принтерами, WiFi-модулями возникает необходимость разрабатывать драйвера самостоятельно.

В связи с этим целью данной статьи является обобщение и структурирование информации об основных компонентах, особенностях и принципах создания собственных драйверов для работы с устройствами со встроенной операционной системой Windows.

Материал и методы. Материалом для исследований послужили данные из книги «Программирование драйверов Windows» В.Коммисаровой (2007) [1] и примеры драйверов с официального сайта Arduino [2].

Результаты и их обсуждение. Главной проблемой выявления общих особенностей разработки драйверов является отсутствие полной и достоверной информации о том, что представляет из себя драйвер изнутри, есть ли в этом какая-либо общая концепция, и если есть, то какая. Все же, в разработке драйверов для взаимодействия с устройствами Windows, несмотря на их большое разнообразие, были выявлены общие особенности и структура.



Рисунок 1 – Схема реализации драйвера Windows

Любой драйвер под Windows должен иметь следующие реализованные функции для работы:

1. Функция инициализации – ключевая функция любого драйвера, представляет собой набор команд для правильной инициализации всех параметров и настройки точек входа каждой из функций драйвера, запускается при загрузке драйвера единожды.

2. Процедура добавления устройства – важный элемент начала взаимодействия компьютера и контроллера, так как данная процедура посылает уведомление о подключении устройства, за работу с которым отвечает данный драйвер. При подключении данная процедура создает объект «устройство», описывающий подключенное аппаратное устройство.

3. Процедура инициации ввода-вывода – реализация обмена данных между устройством и компьютером.

4. ISR-процедура обслуживания прерываний – реализация логики обработки прерываний, генерируемых подключенным устройством, именно этой процедуре процессором будет передаваться управление. Для оптимизации и уменьшения задержек она должна содержать минимум логики, более того, зачастую её реализация может быть не нужна, если работа с устройством исключает возможность возникновения прерываний.

5. DPC-процедура обработки прерываний – процедура, выполняющая основную часть работы по обработке прерывания, оставшуюся после ISR-процедуры. Она отслеживает окончание одной input-output процедуры и позволяет корректно перейти к следующей.

6. Процедуры диспетчеризации – основные процедуры, предоставляемые драйвером для чтения или записи на устройство, диспетчер ввода-вывода обращается к драйверу для выполнения операции именно через одну из его процедур диспетчеризации.

Кроме того, стоит упомянуть об опциональных подсистемах драйвера, как система обработки отмены ввода-вывода и система обработки ошибок.

В остальном же, любой драйвер может включать в себя любое количество функций и процедур, не выполняющих взаимодействие с внешним аппаратным или программным обеспечением, а занимающихся какими-то служебными задачами внутри драйвера, тем самым упрощая и структурируя код.

Заключение. Таким образом, на основываясь на анализе литературных источников и примерах кода уже написанных драйверов, была выявлено, что драйвер представляет собой комплексную систему, включающую в себя ряд элементов, выполняющих отдельные минимальные функции. Наиболее близкой аналогией драйверам в программировании более высокого уровня являются DLL-библиотеки, по сути реализующие набор базовых функций для дальнейшего использования сторонними программами. Драйвер, в отличие от DLL, является библиотекой-посредником между устройствами и операционной системой, позволяя обмениваться данными даже при существенных различиях в архитектуре и особенностях внутренней работы устройств. Результатом работы является схема реализации драйвера Windows с описанием всех ключевых подсистем и элементов драйверов Windows.

1. Комиссарова, В. Программирование драйверов для Windows / В. Комиссарова / – СПб.: БХВ-Петербург, 2007. – 256 с.
2. Документация Arduino [Электронный ресурс]. – Режим доступа: <https://www.arduino.cc/> – Дата доступа: 15.03.2023.

ПЕРСПЕКТИВЫ ТЕХНОЛОГИИ «УМНЫЙ ДОМ»

Гончаренок Е.А., Романов Н.С.,

студенты 3 курса Белорусского государственного университета информатики и радиоэлектроники, г. Минск, Республика Беларусь

Научный руководитель – Беккеров Д.Э.

В настоящее время все большее распространение получают микроконтроллеры и подобные им программируемые устройства. Являясь в большинстве своем парой из самого устройства и прилагаемого к нему программного обеспечения, зачастую исходный код которых распространяется открыто, микроконтроллеры заложили основу для создания такой технологии как «Умный дом».

Ранее основную массу устройств создавали радиолюбители, программисты и другие отдельные специалисты с достаточным уровнем знаний в электронике, в настоящее время существуют и промышленные разработки, такие как «Яндекс-Алиса» и другие. Это свидетельствует о том, что в современном мире технология «Умный дом» является коммерчески выгодной и перспективной. Поэтому целью данной работы является анализ перспектив технологии «Умный дом» на ближайшее время.

Материал и методы. Материалом для исследований послужили данные из статьи [1] и форум с официального сайта Arduino [2].

Результаты и их обсуждение. Анализ современных систем на основе микроконтроллеров в рамках технологии «Умный дом» показал ряд тенденций, по которым развивается данная технология.

С повсеместным распространением информационных технологий разработка автоматизированных систем получает все большее распространение. Для самостоятельного создания рабочих систем требуется все меньше навыков и знаний, так как информационная база большинства проектов на базе микроконтроллеров распространяется в открытом доступе, что ведет к расширению сфер применения подобных систем, в том числе в