

ПРИНЦИПЫ ПРОЕКТИРОВАНИЯ СЛОЖНЫХ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

Васьков А.С.

Научный руководитель: канд. техн. наук, доц. Смирнов Б.И.
 Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»
 им. В.И. Ульянова (Ленина) (СПбГЭТУ), Россия
 E-mail: mc_hale@rambler.ru

Аннотация — Рассмотрена идея моделирования сложных микропроцессорных устройств с богатой периферией и/или взаимодействием между множеством микроконтроллеров с использованием современных технологий и подходов. Кратко описаны наиболее используемые последовательные интерфейсы.

1. Введение

При проектировании сложного микропроцессорного устройства часто возникают ситуации, когда необходимо разбить устройство на функциональные составляющие, и организовать их правильное взаимодействие. В докладе приводится общая идеология построения сложных мультипроцессорных систем, исследованы варианты их моделирования с использованием последовательных интерфейсов в пакете *Proteus*.

2. Основная часть

При проектировании сложных микропроцессорных систем могут возникнуть две обыденные проблемы. Первой проблемой является недостаточная производительность одного микропроцессорного устройства для обеспечения всех функциональных модулей. Вторая проблема возникает из-за большого объема машинного кода с множеством безусловных переходов, так называемого, спагетти-кода (*spaghetti code*), который усложняет понимание работы программы и затрудняет её сопровождение. В языках высокого уровня подобная задача решена объектно-ориентированным программированием (ООП), где каждый класс представляет сущность, выполняющую возложенные на нее задачи. В языках ООП для организации приемлемой архитектуры используют паттерны проектирования (*design patterns*), которые предназначены для обеспечения слабых связей (*weak reference*). Слабые связи позволяют создать систему независимых друг от друга компонентов, которые могут обрабатывать данные друг друга по определенным правилам (интерфейсам), при этом функциональное изменение одного компонента не повлияет на работу другого. Такой подход полезен и в микропроцессорном проектировании, где необходимо контролировать память, задержки, и конфликты. Обеспечив слабые связи можно работать над новой функциональностью, не заботясь о нарушении работы других функциональных модулей, что многократно увеличит производительность разработчика, а также избавит от возможных непредвиденных ошибок. Слабые связи, паттерны проектирования и основные принципы ООП рассмотрены в [1].

Самым простым примером использования паттернов проектирования может быть паттерн «Наблюдатель» (*Observer*), где реализуется связь «один ко многим». Главный контроллер (субъект) принимает сигналы, полученные с датчиков, например, через АЦП по интерфейсу *SPI* [2], и оповещает ведомые контроллеры (наблюдатели), которых «интересует» обновление данных, полученных с субъекта (например, для вычисления других характеристик и передачи данных общему серверу). Наблюдатели независимы друг от друга и в случае появления новой характеристики, достаточно лишь создать новое событие и подписать на него контроллер, который будет обрабатывать это событие. Прежние контроллеры будут при этом работать без изменений. Вероятность сбоев в работе невысока, так как наблюдатели сами выбирают какие характеристики им необходимо принимать и, в общем случае, отсутствует обращение наблюдателей к субъекту. В результате, новая функциональность не будет влиять на исходную, и достаточно лишь добавить новый обработчик события, который должен соответствовать общим спецификациям моделируемой системы (в ООП языках за это отвечают интерфейсы).

Обеспечить необходимые связи информационного обмена между микроконтроллерами позволяют последовательные интерфейсы. Существует несколько вариантов последовательных интерфейсов, наиболее часто применяемых сегодня: *UART*, *I²C*, *SPI*, *USB*. Наиболее распространенный интерфейс *UART* (*Universal Asynchronous Receiver/Transmitter*) является наиболее простым из них и поддерживается почти во всех микроконтроллерах. При передаче данных по интерфейсу *UART* используются два канала передачи данных.

В докладе рассматриваются мультипроцессорные модели, включающие 4 микроконтроллера семейства 8051, связанные между собой по последовательному интерфейсу. Для моделирования использован пакет *Proteus 7.8*. Рассмотрены варианты использования интерфейсов *UART*, *I²C* и *SPI* и применения шаблонов проектирования типа *Observer*.

Построенная модель показала высокую скорость симуляции (в режиме анимации, т.е. воспроизведения работы устройства в режиме реального времени), устойчивость в работе благодаря слабым связям и невысокую загрузку процессорного ядра. Модели предполагается использовать в учебном процессе университета.

3. Заключение

В докладе описаны принципы построения сложных устройств, и причины, по которым рекомендуется придерживаться идей паттернов проектирования, которые применяются во многих областях, и реализация их основной идеи — слабых связей сильно облегчит проектирование сложных устройств. Описанные принципы были реализованы в моделях мультипроцессорных систем в пакете *Proteus*.

4. Список литературы

- [1] Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Дж. Влиссидес. — СПб.: Питер, 2007. — 336 с.
- [2] Магда Ю. Микроконтроллеры серии 8051: практический подход / Ю. Магда. — М.: ДМК Пресс, 2008. — 228 с.

THE PRINCIPLES OF DESIGNING OF COMPLEX MICROPROCESSOR SYSTEMS

Vaskov A.S.

Scientific adviser: Smirnov B.I.
 Saint Petersburg Electrotechnical University "LETI",
 Russia

Abstract — The idea of the complex microprocessor devices design with a large periphery and/or an interaction between multiple microcontrollers with the latest technologies and approaches is considered. The most useful interfaces are described.