

Санкт-Петербургский государственный политехнический университет
Факультет технической кибернетики
Кафедра автоматики и вычислительной техники

Работа допущена к защите
Зав. кафедрой

_____ В.Ф. Мелехин
“ ___ ” _____ 2009 г.

ВЫПУСКНАЯ РАБОТА БАКАЛАВРА

Тема: Разработка радио-системы сбора и первичной обработки информации на основе микросхемы CYWUSB6953 компании Cypress

Направление: 220200 - Автоматизация и управление

Выполнил студент гр. 4081/2 _____ О. В. Ненашев

Руководитель, канд. тех. наук _____ Н. Н. Балтруков

Санкт-Петербург
2009

Санкт-Петербургский государственный политехнический университет
Факультет технической кибернетики
Кафедра автоматики и вычислительной техники

УТВЕРЖДАЮ

“ _ ” _____ 2009 г.

Зав. кафедрой _____

ЗАДАНИЕ

по выпускной бакалаврской работе
студенту Ненашеву Олегу Вячеславовичу

1. Тема работы Разработка радио-системы сбора и первичной обработки информации на основе микросхемы CYWUSB6953 компании Cypress

2. Срок сдачи студентом законченной работы _____

3. Исходные данные к проекту (работе) Разработка должна производиться для микросхем CYWUSB6953 при помощи сред разработки PSoC Designer версий 4.3 и 5.0 на отладочных стендах CY3653. Требуется изучить технологию беспроводной связи WirelessUSB и исследовать возможности PРоC-кристаллов, разработать ряд систем по сбору дискретной и аналоговой информации и модифицировать протокол WirelessUSB так, чтобы обеспечить его совместимость со средой разработки PSoC Designer 5.0.

4. Содержание расчётно-пояснительной записки Анализ технического задания и постановка задач исследования; исследование технологии беспроводной связи WirelessUSB; изучение возможностей PРоC-кристалла по сбору и первичной обработке аналоговой и дискретной информации; разработка модификации протокола WirelessUSB; разработка описаний лабораторных работ по изучению возможностей применения микросхемы CYWUSB6953

5. Дата выдачи задания _____

Руководитель _____ (Балтруков Н.Н.)
(подпись руководителя) (Фамилия и инициалы)

Задание принял к исполнению “ _ ” _____ 2009 г.

_____ (Ненашев О.В.)
(подпись руководителя) (Фамилия и инициалы)

РЕФЕРАТ

с. 2, рис. 34, табл. 22

БЕСПРОВОДНАЯ СВЯЗЬ, СИСТЕМА СБОРА ИНФОРМАЦИИ, ТЕХНОЛОГИЯ WIRELESSUSB, PSoC, PROC, ПЕРЕДАТЧИКИ, ПРОТОКОЛ CYWUSB6953, ЛАБОРАТОРНАЯ РАБОТА.

Произведен обзор современных технологий беспроводной связи. Рассмотрены средства проектирования PРоC-кристаллов. Изучены принципы передачи информации в WirelessUSB, организация приёмопередатчиков, алгоритмы и программная реализация протокола. Выявлены недостатки фирменного протокола. Разработаны демонстрационные примеры организации беспроводной связи на базе микросхемы CYWUSB6953.

Изучена структура PРоC-кристаллов. Изучены возможности реализации стандартных пользовательских модулей в среде PSoC Designer. Произведён анализ возможностей реализации систем сбора и обработки информации на базе PРоC. Произведена разработка ряда систем сбора информации с использованием беспроводного канала связи для передачи на ЭВМ. Разработана терминальная программа для ЭВМ. Найдены ошибки в программном обеспечении PSoC Designer для аналоговой подсистемы микросхем CYWUSB6953. Разработана модификация протокола WirelessUSB. Разработаны описания лабораторных работ на базе созданных проектов.

Содержание

Введение	7
1. Анализ технического задания и постановка задач исследования	10
1.1. Обзор современных технологий беспроводной связи	10
1.2. Введение в технологию WirelessUSB компании Cypress	16
1.3. Средства проектирования PROC кристаллов	20
1.4. Задачи разработки	23
2. Исследование технологии беспроводной связи WirelessUSB	24
2.1. Принципы кодирования и частотного разделения каналов	24
2.2. Исследование организации приёмопередатчиков WirelessUSB	26
2.3. Исследование протокола передачи данных в WirelessUSB	30
2.4. Исследование программной реализации протокола WirelessUSB	35
3. Изучение возможностей PРоC-кристалла по сбору и первичной обработке аналоговой и дискретной информации	44
3.1. Архитектурные особенности PРоC-кристаллов	44
3.2. Пользовательские модули и возможность их реализации в среде PSOC Designer 4.3	50
3.3. Проблемы использования аналоговой подсистемы кристалла CYWUSB6953 в среде проектирования PSoC Designer	57
4. Разработка модификации протокола WirelessUSB	59
4.1. Краткая характеристика разработанного протокола	59
4.2. Программная реализация модифицированного протокола	62
4.3. Сравнительный анализ модифицированного протокола с базовым	65
Заключение	68
Список использованных источников	70

Приложение 1. Исследование принципов реализации радиоканала на базе PРоC кристалла CYWUSB6953 и протокола Wireless USB	71
Приложение 2. Исследование средств предварительной обработки дискретной и аналоговой информации для передачи по радиоканалу на основе кристалла CYWUSB6953	84
Приложение 3. Описание терминальной программы ComViewer	98

Перечень условных обозначений

АЦП - Аналого-Цифровой Преобразователь

УАП - Упрощённая Аналоговая Подсистема

ПМ - Пользовательский Модуль

ШИМ - Широтно-Импульсная Модуляция

DBB - Digital Basic Block (Цифровой базовый блок)

DCB - Digital Communication Block (Цифровой коммуникационный блок)

PSoC - Programmable System on Chip (Программируемая система на кристалле)

PRoC - Programmable Radio on Chip (Программируемое радио на кристалле)

DSSS - Direct-sequence spread spectrum (широкополосная модуляция с прямым расширением спектра)

FHSS - Frequency Hopping Spread Spectrum (метод частотных скачков)

GFSK - Gaussian Frequency Shift Keying (Частотная манипуляция с распределением по гауссовому закону)

UWB -Ultra-Wide Band (сверхширокий диапазон)

PSD - Среда Разработки PSoC Designer

PSE - Среда Разработки PSoC Express

HID - Human Interface Device (устройство пользовательского интерфейса)

MID - Manufacturing ID (серийный номер микросхемы)

Введение

В настоящее время беспроводная связь находит применение в самых разных областях: от организации домашних сетей до управления мобильными объектами. Основное достоинство беспроводных каналов связи – отсутствие проводов. Благодаря этому достигается простота и быстрота организации сети, в том числе при отсутствии прямой видимости или при большом расстоянии между устройствами. Среди различных способов беспроводной связи наиболее развита радиосвязь, и в дальнейшем под беспроводной связью понимается именно она.

Если расстояние между источником информации и системой её обработки велико, то рядом с источником устанавливается так называемая система сбора информации, в задачи которой входят съём данных, первичная их обработка и передача в систему управления. Такие системы должны быть достаточно дешёвы, надёжны и просты в эксплуатации. Одним из возможных решений по связи систем сбора информации и её обработки является использование радиоканала. При этом упрощаются монтаж системы, её наращивание и замена модулей. Критериями эффективности беспроводной связи в системах сбора являются скорость и дальность передачи информации, стоимость самого передатчика, его энергопотребление и помехозащищённость.

В настоящее время существует множество различных интерфейсов беспроводной связи. Однако, большинство из них требуют использования передатчика как отдельного модуля, что удорожает и усложняет систему в целом. Компания Cypress одной из первых (в 2002 г.) предложила семейство микросхем PRoC (Programmable Radio on Chip), где цифровая часть передатчика встроена в управляющий контроллер. Данные устройства основаны на технологии передачи WirelessUSB, также разработанной данной компанией.

Микросхемы PРоС построены по технологии PSoC (Programmable System on Chip). “Программируемая система на кристалле” включает, кроме ядра контроллера, набор конфигурируемых цифровых и аналоговых блоков, которые реализуют различные функции (счётчики, коммуникационные интерфейсы, АЦП и т.д.). Система на основе PSoC получается очень гибкой, так как возможна динамическая переконфигурация её элементов. Микросхемы PРоС поддерживаются такими средствами проектирования, как PSoC Designer от компании Cypress.

Беспроводная связь на основе WirelessUSB используется для дистанционного управления различными устройствами, в средствах сигнализации и защиты и промышленных системах сбора информации в распределённых системах. Сама технология достаточно нова, и в настоящее время микросхемы PРоС используются мало. Отдельные передатчики получили большее распространение. Прежде всего, это вызвано недостатком информации о технологии и малым её продвижением со стороны компании Cypress.

Тем не менее, заявленные параметры технологии (скорость до 1 Мбит/с при дальности до 50м) при малой стоимости устройств делают рассматриваемую технологию достаточно перспективным направлением, поэтому актуальна задача исследования возможностей реализации беспроводных систем сбора информации на основе WirelessUSB.

При этом, особенно актуально исследование микросхем CYWUSB6953, которые отличаются низкой стоимостью (вплоть до 2.5 USD). Кроме того, в контроллер интегрирован приёмопередатчик WirelessUSB LS, который реализует радиоканал со скоростью до 64 кбит/с и дальностью до 50 метров, чего вполне достаточно для многих задач. Микросхема CYWUSB6953 является полноценным контроллером:

на ней могут быть реализованы аналого-цифровые преобразователи и стандартные последовательные интерфейсы (UART, SPI, I2C). Поэтому, в работе делается упор именно на этот тип микросхем.

1. Анализ технического задания и постановка задач исследования

1.1. Обзор современных технологий беспроводной связи

В настоящее время существует множество различных технологий беспроводной передачи данных на основе радиоканала. Для их классификации можно использовать различные критерии. Пример классификации каналов связи по некоторым из них приведён на рис. 1.1.



Рис. 1.1. Классификация каналов связи

В целом, технологии беспроводной передачи данных можно разделить на две большие группы, используя классификационный признак “активность канала”. В одной из них обеспечивается установление прямого соединения на все время сеанса связи, независимо от реальной загрузки канала - так же, как это происходит в кабельных сетях с коммутацией каналов. Такие технологии обеспечивают синхронную связь - на одном конце происходит передача, а на другом, в то же самое время, - прием.

Технологии другой группы аналогичны системам с коммутацией пакетов - в них не обеспечивается синхронность, но зато и соединение устанавливается только на время реальной передачи, поэтому канал используется более эффективно.

При использовании **узкополосного** канала связи передача ведется в малом диапазоне вблизи строго определенной частоты. Отсюда - взаимные наводки, необходимость "делить" эфир, невозможность работы двух устройств в непосредственной близости друг от друга.

Широкополосные технологии используют значительно более широкий диапазон, чем при обычных методах модуляции, обладают более высокой помехоустойчивостью по отношению к узкополосным шумам и более экономно используют спектр сигнала. Поэтому, большинство массовых технологий (в т.ч. и WirelessUSB) относятся к данному типу. Широкополосные технологии могут использовать следующие способы передачи информации:

DSSS (Direct-sequence spread spectrum) - передача широкополосного сигнала по методу прямой последовательности. При этом вся используемая «широкая» полоса частот делится на некоторое число подканалов (например, на 11 по стандарту 802.11). Каждый передаваемый бит информации разбивается на сигналы в каждом из подканалов (на "чипы"). При приеме последовательность чипов декодируется по тому же алгоритму, что и при кодировке, и таким образом полезный сигнал удается выделить на фоне шума. Другая пара приемник-передатчик может использовать другой алгоритм кодировки - декодировки, поэтому возможна параллельная работа каналов при одинаковых частотах. Данный метод кодировки информации обеспечивает хорошую защиту канала как от "прослушивания", так и от узкополосных помех, создаваемых узкополосными устройствами. При этом мощность сигнала каждого из чипов мала, за счёт чего не оказывается влияние на сами узкополосные устройства.

FHSS (Frequency Hopping Spread Spectrum) - передача широкополосного сигнала по методу частотных скачков.

При кодировке по методу частотных скачков (FHSS) вся отведенная для передачи полоса частот разделяется на подканалы (по стандарту 802.11 их 79). В каждый момент времени каждый передатчик использует только один из подканалов, переходя с одного подканала на другой через определенные промежутки времени. Эти скачки происходят синхронно на передатчике и приемнике в заранее зафиксированной псевдослучайной последовательности, следовательно, не зная последовательности переключений, принять сигнал нельзя. Кроме того, так как частота передачи постоянно меняется, то сообщение, потерянное на одной частоте из-за помехи, может быть повторно передано на другой.

GFSK (Gaussian Frequency Shift Keying) - метод частотной манипуляции с низкочастотным фильтром на входе, обеспечивающий сглаживание формы входных импульсов по гауссовому закону. Данный метод используется в таких технологиях, как Bluetooth и DECT.

CSS (Chip Spread Spectrum)- Линейно-частотная модуляция. Мощность сигнала "размывается" по спектру, и при воздействии помех фиксированной частоты теряется только часть передаваемого сигнала, так что двоичная информация может быть, затем восстановлена в приемнике. Однако, при таком способе ширина канала намного больше, чем для остальных технологий.

Сверхширокополосные (UWB) передатчики могут использовать диапазоны в несколько Гигагерц (для WUSB- от 3.1 до 10.6 ГГц). Как известно, высокочастотные сигналы в атмосфере быстро рассеиваются, что обеспечивает низкий уровень помех. Это позволяет снизить стоимость передатчиков и существенно повысить максимальную скорость передачи данных (например, технология WUSB может обеспечивать скорость до 480 Мбит/с). Однако, по той же самой причине, дальность работы высокочастотных передатчиков мала [6].

Существует большое количество число различных коммерческих стандартов связи. Однако, большинство из них не получили большого распространения в условиях сильной конкуренции. Сравнительная характеристика некоторых наиболее популярных технологий приведена в таблице 1.1.

Коммерческие стандарты связи. Таблица 1.1

Технология	В	Обозначение IEEE	Скорость передачи, бит/с	Дальность передачи, м	Частотный диапазон	Стоимость передатчика, \$
Bluetooth	1.x 2.x	802.15.1	450к 2.1М	10(100) 10	ISM ISM	>8
ZigBee		802.15.4	250к	75	ISM 433, 868 МГц	5
WirelessUSB	LS	-	62.5к	10	ISM	2.5
	LR	-	62.5к	50	ISM	3
	LP	-	1000к	50	ISM	4.5
Wi-Fi		802.11	2М	100	ISM	15
		802.11a	54М	100	5 ГГц	
		802.11b	11М	100	ISM	
		802.11g	54М	100	ISM	
Wi-Max		802.16	70М	3-4к	10-66 ГГц	?
		802.16a	70М		2-10 ГГц	
WUSB		802.15.4a	500М	10	3.1-10.6 ГГц	15

Сравнительная диаграмма технологий по скорости и дальности передачи приведена на рис. 1.2.

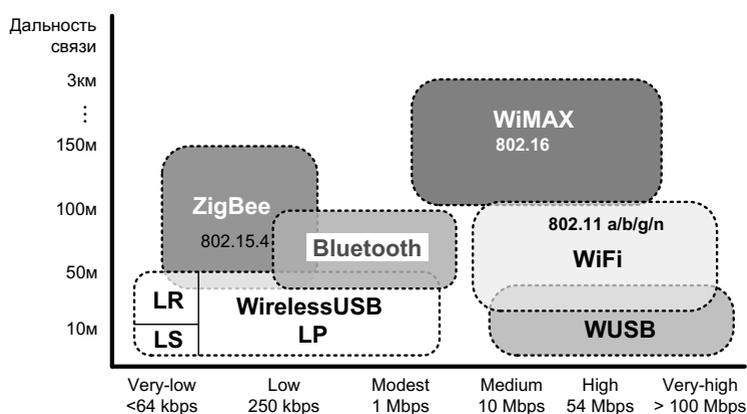


Рис. 1.2. Сравнительная диаграмма технологий беспроводной связи

Диапазоны 433-435 и 868-870 МГц

Два достаточно узких нелицензируемых диапазона для радио-связи. Разработка устройств, работающих в этих диапазонах, затруднена из-за высокой популярности диапазона (например, для радио-переговорных устройств) и, как следствие, высокого уровня помех. Поэтому, передатчики в этом диапазоне должны обладать достаточно большой мощностью, что приводит к повышенному энергопотреблению микросхем. Поэтому, для рассматриваемой задачи использование данного диапазона не является оптимальным.

Широкополосные технологии в диапазоне ISM

ISM (Industrial, Scientific and Medical) – это диапазон, выделенный под коммерческое использование практически во всём мире. Область частот лежит в пределах от 2.4 ГГц до 2.483 ГГц. При использовании данного диапазона не требуются затраты на лицензирование и перенастройку устройств под региональные стандарты, именно поэтому он используется большинством современных стандартов. Однако, в ISM действуют ограничения на мощность излучения одним передатчиком (500 мВт). К тому же, для сертификации стандарта требуется совместимость (возможность параллельной работы) с ранее зарегистрированными технологиями.

При использовании частот из диапазона ISM и соблюдении требований по мощности существует возможность осуществлять передачу на расстоянии до ста метров, чего вполне достаточно для большинства задач.

В диапазоне ISM использование узкополосных устройств затруднено, так как многие приборы (например, СВЧ-печи) излучают в этом же диапазоне, поэтому все стандарты в этом диапазоне являются широкополосными.

Наиболее популярными технологиями передачи в диапазоне ISM являются Bluetooth, Zigbee, WirelessUSB (Cypress) и Wi-fi версий b/g (на этом стандарте основаны сети WLAN). Кроме того, существуют ещё несколько десятков менее известных технологий. Поэтому, в диапазоне ISM встаёт проблема совместной работы многих устройств (например, первая версия Wi-fi (802.11) оказалась несовместимой с уже залицензированным Bluetooth, из-за чего так и не вышла в свет).

В настоящее время стандарты из диапазона ISM занимают следующие ниши:

1) Wi-fi – высокоскоростные беспроводные сети.

В других целях технология применяется мало из-за высокой цены устройства, высокого энергопотребления и ориентированности на связь "точка- много точек" (требуется наличие "точки доступа").

2) Bluetooth- носимая аппаратура.

Применяется там, где использование Wi-fi слишком дорого, а скорости других технологий недостаточно. У данной технологии развит сетевой протокол, но в большинстве случаев данная технология используется для связи лишь двух устройств. В настоящее время Bluetooth постепенно вытесняется с рынка другими технологиями из за высокой стоимости и большого энергопотребления передатчиков.

3) ZigBee и WirelessUSB- низкоскоростные каналы связи.

Рынок низкоскоростных устройств в данный момент делят между собой две молодых технологии: ZigBee и WirelessUSB. Вторая технология известна меньше, т.к. не была сертифицирована IEEE. Однако, по всем параметрам технологии близки друг к другу. Обе обладают очень низким энергопотреблением, цены приёмопередатчиков также близки. Преимуществом WirelessUSB является большое число независимых каналов, но у ZigBee более развит сетевой протокол.

Сверхширокополосная связь (UWB)

Данный стандарт разрабатывался рядом компаний под руководством Intel в рамках спецификаций IEEE 802.15.4a и был представлен в феврале 2004-го года. Технология UWB (Ultra-Wide Band) основана на передаче множества закодированных импульсов не гармонической формы очень малой мощности (0,05 мВт) и малой длительности в широком диапазоне частот (от 3,1 до 10,6 ГГц). Передача данных на расстояние до 5 метров осуществляется со скоростью от 400 до 500 Мбит/сек. На базе стандарта UWB базируются такие популярные технологии, как WiMAX и WUSB.

Стоимость сверхширокополосных приёмопередатчиков очень высока, а дальность передачи невелика. Высокоскоростные UWB-устройства хорошо подходят для работы с видеопотоками и прочими задачами, где требуется большая скорость передачи данных.

Среди рассмотренных технологий WirelessUSB является одной из наиболее привлекательных для использования в системах сбора информации с малой скоростью передачи данных. Его преимуществами является использование ISM-диапазона, благодаря чему не требуется лицензирование устройств, низкая стоимость и большое число каналов связи, что позволяет одновременно использовать большое число устройств в ограниченном пространстве.

1.2. Введение в технологию WirelessUSB компании Cypress

Внедрение WirelessUSB началось в 2002 году, когда компания Cypress Semiconductor представила новую беспроводную технологию, работающую в диапазоне ISM. Для упрощения разработки средств обмена информацией по радиоканалу компанией Cypress был предложен специализированный протокол, названный WirelessUSB.

Технология WirelessUSB позволяет организовывать каналы связи типов "точка-точка" и "точка-много точек" (рис. 1.3). Основное применение - беспроводные мосты и устройства сбора информации.

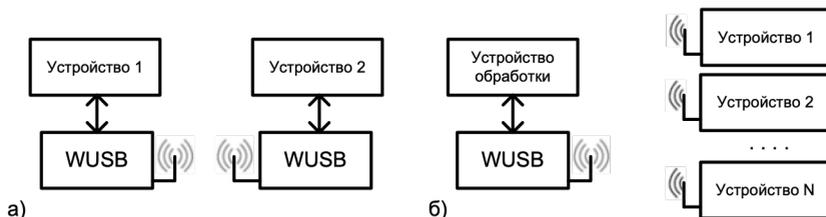


Рис. 1.3. Возможные организации канала связи (а- "точка-точка", б- "точка- много точек").

Главной особенностью нововведения стало интегральное исполнение цифровой части приёмопередатчика на одном кристалле с фирменным PSoC-контроллером. Новая линейка продуктов была названа PRoC (Programmable Radio on Chip). Такая система позволила обеспечить высокую степень интеграции продуктов и, следовательно, их низкую стоимость.

Концепция Cypress с первых дней существования привлекла внимание разработчиков, и, в настоящее время, выпускается большое количество специализированных контроллеров со встроенными модулями беспроводной связи.

Первое семейство специализированных кристаллов, CY694x (CY6941 Optical Mouse, CY6942 Keyboard, CY6943 USB Bridge) должно было составить конкуренцию Bluetooth и другим подобным стандартам на рынке беспроводного соединения периферии и персонального компьютера. По таблице, приведённой ниже, видно, что уже первые модели WirelessUSB конкурентоспособными по сравнению с существовавшими тогда стандартами связи.

Сравнительные характеристики Bluetooth и CY694x, 2003 год.

Таблица 1.3 [1]

Параметр	WirelessUSB	Bluetooth
Частотный диапазон	ISM	ISM
Номинальная дальность передачи, м	10	10
Максимальная скорость передачи, кбит/с	62.5	227
Количество коммутируемых устройств	>7	128
Задержка передаваемого сигнала, мс	8-20	30-40
Стоимость передатчика, \$	4	15
Энергопотребление передатчика, мВт	30-40	>100

В дальнейшем компанией Cypress был расширен ассортимент кристаллов, использующих WirelessUSB, стоимость которых постоянно снижалась. В конце 2003-го года был представлен передатчик LS (CY6953) с дальностью 50м при скорости 15.6 кбит/с, а уже в 2004-м появилось семейство LR (CY6935) со скоростью 62.5 кбит/с.

В 2006-м году было представлено семейство LP (CY6936). В нём была не только увеличена скорость передачи данных, но и снижено энергопотребление (особенно в спящем режиме). Наиболее новым семейством является CyFi, появившиеся в конце 2008-го года. Данное семейство приёмопередатчиков является дальнейшим развитием LP. В табл. 1.4 перечислены основные семейства микросхем WirelessUSB.

Семейства микросхем WirelessUSB. Таблица 1.4

Семейство	Назначение
CY694x	Семейство старых микросхем, применяемых для готовых изделий
CYWUSB695x	Программируемый контроллер с приёмопередатчиком из семейства LS
CYWUSB693x	Программируемый контроллер с приёмопередатчиком из семейства LR
CYRF693x	Приёмопередатчики из семейства LP
CYRF793x	Приёмопередатчики из семейства CyFi

Среди существующих на сегодняшний день технологий WirelessUSB для задач сбора информации наиболее интересны семейства LS и LP, которые интегрируются в управляющий контроллер.

В PRoC приёмопередатчик WirelessUSB является специальным системным ресурсом кристалла. Внутри кристалла реализована цифровая часть радио-устройства, управление которым производится с помощью интерфейса SPI (рис. 1.4). Вне кристалла располагается цепь сопряжения с антенной.

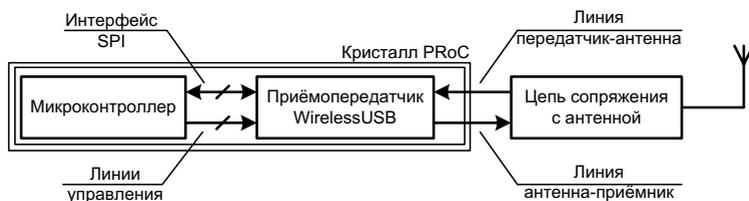


Рис. 1.4. Типовая схема включения передатчика.

Контроллеры PROC поставляются компанией Cypress как в виде отдельных контроллеров, так и уже готовых сборок (SIP-модулей), которые включают контроллер, цепь сопряжения и антенну. Следует отметить, что дальность передачи по радиоканалу, приводимая в документации, рассчитана именно для стандартных сборок. При необходимости можно повысить дальность передачи путём добавления усилителя или увеличения площади антенны [8].

В настоящее время технологию WirelessUSB используют такие фирмы, как Atmel, NEC, Logitech, Unigen и ещё около 50-ти производителей [1]. Компанией Atmel выпускаются передатчики ATR2406, которые являются практически полной копией WirelessUSB LP.

1.3. Средства проектирования PROC кристаллов

PSoC-кристаллы компании Cypress являются специализированными микроконтроллерами, позволяющими реализовать расширенную аналого-цифровую периферию различного назначения в рамках одного кристалла. Очевидно что, расширение функциональности должно сопровождаться дополнительными сложностями процесса проектирования и отладки. Поэтому, требуются специализированные средства разработки и отладки, позволяющие упростить и частично автоматизировать процесс разработки устройств на базе PSoC.

Введение радиопередатчика (**PRoC**) ещё более усложняет устройство, поэтому требуется наличие специализированных средств отладки радиоканала. Компания Cypress поставляет полный комплект средств разработки: среды проектирования, программаторы, отладчики и наборы разработки для предварительного исследования.

Наиболее простой в использовании средой проектирования является PSoC Express (PSE), часто называемая графическим генератором приложений. Особенностью данной среды является то, что весь код программ генерируется автоматически на основе заданных пользователем алгоритмов. Несомненно, такой механизм проектирования удобен для быстрой разработки простых проектов при малой квалификации разработчиков.

Среда PSoC Express содержит модули для реализации беспроводной связи между устройствами в различных режимах. Однако, единственной возможностью является использование внешнего передатчика семейств WirelessUSB LR или LP. Микросхемы PRoC данной средой не поддерживаются, что исключает возможность применения данной среды при разработке беспроводных устройств на базе CYWUSB6953.

Основным средством разработки систем является интегрированная среда проектирования **PSoC Designer**. Она позволяет выполнить весь процесс проектирования минисистемы на высоком уровне, используя специальный графический интерфейс, встроенную библиотеку пользовательских модулей и широко развитую справочную службу. Кроме того, при дополнительной аппаратной поддержке, в рамках среды осуществляется прошивка кристалла и обеспечивается возможность выполнения отладки на основе внутрисхемного эмулятора.

Основными подсистемами PSoC Designer являются:

1. Редактор устройства (Device Editor), предназначенный для конфигурирования программируемых блоков, портов ввода-вывода, определения системных параметров;

2. Редактор приложения (Application Editor), предназначенный для написания программной части проекта на языке C или ассемблере. При использовании языка C требуется лицензионная версия компилятора ImageCraft;

3. Отладчик (Debugger), позволяющий при наличии аппаратного внутрисхемного эмулятора использовать множество стратегий отладки, такие как, точки останова, отслеживание переменных, трассирование и другие.

Существуют несколько версий среды PSoC Designer. Рассматриваемые в работе PРоC-микросхемы и пользовательские модули для них впервые появились в версии PSD 4.2 и были без изменений перенесены в версию 4.3. В данной версии для компиляции Си-программ использовался компилятор ImageCraft 1.47, в более поздних версиях среды используется 1.69. Поэтому, для PSD 4.4 и выше возникли проблемы совместимости старых программ с новыми средами разработки [13].

Пользовательские модули в данной среде обновляются автоматически, а корректировка пользовательского кода остаётся за разработчиком. В частности, по результатам предварительного исследования, открытая версия протокола WirelessUSB не совместима с PSD 4.4 и PSD 5.0.

Исходя из вышесказанного, существует необходимость использовать старые версии сред разработки, которые в свою очередь не полностью поддерживают современные средства отладки (например, внутрисхемный эмулятор ICE-Cube), что сильно усложняет процесс разработки.

Число отладочных модулей для технологии WirelessUSB невелико. Для микросхем PRoC предназначен **набор CY3653** с базовой микросхемой CYWUSB6953. Он включает два модуля, на которые монтируются микросхемы. На каждом модуле располагаются наборное поле, жидкокристаллический индикатор, набор варисторов, кнопок и светодиодов, а также преобразователь интерфейса UART в RS-232. Модули могут питаться от девятивольтовой батарейки, что удобно при проверке работы беспроводного канала связи.

Для эмуляции контроллера PRoC используется специальная микросхема, поставляемая в наборе CY3653. Она подключается через специальный шлейф к универсальному эмулятору ICE-Cube (CY3215-DK). Отладка программы при этом производится в среде PSoC Designer. Для наблюдения доступны все регистры контроллера и оперативная память.

Из-за возникающих задержек отладка протокольной части программы затруднена. Поэтому, для семейства приёмопередатчиков WirelessUSB LS предлагается модуль WirelessUSB Listener (CY3632L), который производит наблюдение за каналом и дешифрацию пакетов протокола.

1.4. Постановка задач исследования

На основании вышеизложенного обзора можно сформулировать следующие задачи разработки.

При проведении исследования требуется подробно изучить аппаратную и программную составляющие технологии WirelessUSB. При изучении аппаратного уровня следует изучить как общие принципы передачи данных в интерфейсе, так и существующие на сегодняшний момент семейства приёмопередатчиков. Далее следует произвести исследование программной части: принципа организации сети, методики выбора канала, соединения устройств, протокола обмена пакетами. По итогам исследования следует разработать несколько практических примеров организации обмена данными между устройствами как с использованием протокола WirelessUSB, так и без него, и на их основе разработать лабораторную работу.

Следует исследовать возможности использования микросхем PРоC в задачах сбора и обработки информации. Для этого нужно произвести исследование внутренней структуры PРоC-кристалла, особенности организации цифровой и аналоговой блоков подсистем, возможностей реализации на их основе пользовательских модулей для обработки дискретной и аналоговой информации. Полученные результаты реализовать в виде лабораторной работы.

Как уже было сказано, возникают проблемы совместимости протокола WirelessUSB с современными средствами разработки. Поэтому, была поставлена задача разработки модификации протокола для WirelessUSB 1.1, который используется в устройствах с передатчиками семейств WirelessUSB LS и LR. При этом требуется обеспечить совместимость протокола со средой PSoC Designer 5.0, максимально реализовав функциональность исходного протокола.

2. Исследование технологии беспроводной связи WirelessUSB

2.1. Принципы кодирования и частотного разделения каналов

Технология Wireless USB предполагает использование большого числа независимых каналов передачи информации в диапазоне ISM при исключении влияния приемопередатчиков иных стандартов, обеспечивающих беспроводную связь в заданном диапазоне, а также прочего оборудования, в том числе и бытового, например, СВЧ-печей.

Формирование сигнала в WirelessUSB предполагает побитное преобразование кода по алгоритму DSSS (расширение спектра прямой последовательностью). Каждый информационный бит преобразуется в кодовую последовательность, называемую кодом псевдо-шума. В WirelessUSB длина последовательности (количество “чипов”) равно 64.

Зная код псевдо-шума и частоту канала, приёмник может выделить информационный сигнал, а для других принимающих устройств сигнал будет восприниматься как шум. Такой подход позволяет снизить мощность передатчика, так как не требуется “перекрывать” естественный уровень шумов. Кроме того, при таком исполнении каналы Wireless-USB практически не оказывают влияния на другие стандарты связи, работающие в диапазоне ISM.

Условием выбора кодов псевдо-шума являются высокая автокорреляция и низкая корреляция между собой. Первое условие позволяет приёмнику быстро выделять информационный сигнал, а выполнение второго условия позволяет организовать большое количество каналов связи на одной и той же частоте.

Передатчики WirelessUSB в качестве опорного сигнала используют 64-разрядные коды Голда, которые наилучшим образом соответствуют критериям, приведённым выше. Всего существует 49 различных 64-разрядных кодов Голда.

Принцип коррекции ошибок в коде Голда приведен на рис. 1.5

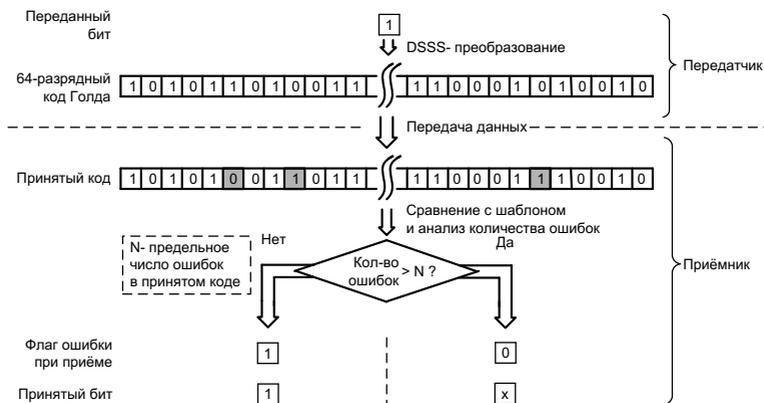


Рис. 2.1. Исправление ошибок в коде Голда [1]

Если кодовое расстояние между принятой комбинацией и ожидаемой не превышает заданного (пользователем) порога, то полученный бит сохраняется в буфере приёмника. В противном случае выставляется флаг о неправильном приёме бита. После окончания приёма полученная информация передаётся на управляющее устройство, где производится корректировка неправильно принятых битов.

В WirelessUSB диапазон ISM разбивается на 78 радио-каналов шириной в 1 МГц, в каждом из которых может использоваться 49 различных кода Голда. Таким образом, формируется 3822 (78*49) независимых канала связи, причём разделение производится на физическом уровне и не требует затрат программного времени.

Кроме DSSS, в WirelessUSB присутствует механизм GFSK (Гауссова частотная модуляция). Этот алгоритм управляет изменением несущей частоты канала в пределах выделенной полосы (1 МГц), что позволяет уходить от узкополосных помех внутри полосы. Изменение частоты производится псевдо-случайным образом по нормальному закону распределения.

2.2. Исследование организации приёмопередатчиков WirelessUSB

В настоящее время в PRoC используются три основных семейства радио-передатчиков WirelessUSB: LS, LR и LP. LS - основное, наиболее простое и дешёвое семейство, у семейства LR повышена дальность передачи, LP совмещает в себе свойство дальности передачи с пониженным энергопотреблением, особенно в спящем режиме. В табл. 2.1 приведены сравнительные характеристики трёх семейств.

Сравнительные характеристики различных семейств приёмопередатчиков WirelessUSB. Таблица 2.1

Параметр	Семейство приёмопередатчиков		
	LS	LR	LP
Частотный диапазон	ISM		
Номинальная дальность передачи, м	10 (50)	50	50
Максимальная скорость передачи, кбит/с	62.5 (15.6)	62.5	250 (DSSS), 1000 (GFSK)
Мощность передатчика, дБм	До 0	До 0	До +4
Чувствительность приёмника, дБм	-90	-95	-97
Максимальный потребляемый ток в режиме передачи, мА	69.1	69.1	21 (для -5 дБм)
Максимальный потребляемый ток в режиме приёма, мА	57.7	57.7	21
Максимальный потребляемый ток в спящем режиме, мА	0.24	0.24	0.0008
Внешний интерфейс	SPI		

Структурная схема приёмопередатчиков семейств LS и LR приведена на рис. 2.2. На рисунке видны все этапы преобразования передаваемого кода (сериализация, преобразование DSSS и GFSK-модуляция), описанные ранее. Для принимаемой и передаваемой информации используются отдельные линии. В радио-модуль также входит средство определения интенсивности принимаемого сигнала по сравнению с уровнем шумов (RSSI).

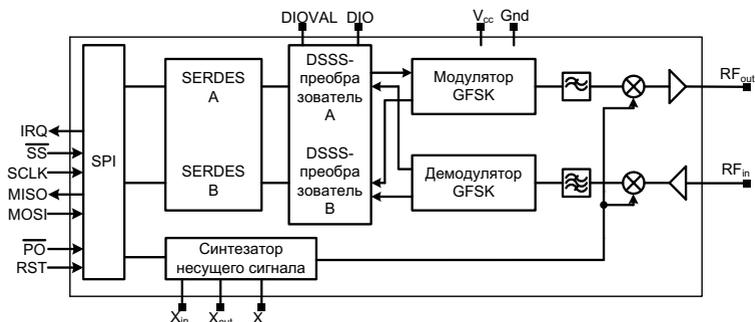


Рис. 2.2. Структура приёмопередатчика из семейств LS и LR

Для приёмопередатчиков семейств LR и LS существует 4 режима работы, обеспечиваемых устройством управления блоками.

- Одиночный канал при 64-чиповом кодировании. Передача производится на скорости 15.6 кбит/с. При этом достигается наибольшая помехозащищённость, что позволяет передавать информацию на расстояние до 50-ти метров.

- Одиночный канал при 32-разрядном кодировании и удвоенной скорости передачи данных. Информация передаётся со скоростью 62.5 кбит/с.

- Одиночный канал при 32-разрядном кодировании и удвоенной частоте выборки. Передача информации на скорости 31.25 кбит/с. За счёт удвоенной выборки обеспечивается устойчивость к помехам.

- Двойной канал при 32-разрядном кодировании. Позволяет одновременно производить приём и передачу информации на скорости 31.25 кбит/с. При этом каналы работают на одной частоте, но с различными псевдокодами.

Как видно из табл. 2.1, различие в параметрах между семействами LR и LS состоит только в чувствительности передатчика, что позволяет передавать информацию на расстояние до 50-ти метров при 32-битном кодировании. В остальном семейства являются идентичными.

Семейство **LP** принадлежит к новому поколению передатчиков WirelessUSB. В них существенно улучшены энергетические и скоростные характеристики устройства, но сохранена совместимость со всеми режимами работы передатчиков типов LS и LR. Кроме того, добавлены новые высокоскоростные режимы работы. Структурная схема модуля LP приведена на рис. 2.3.

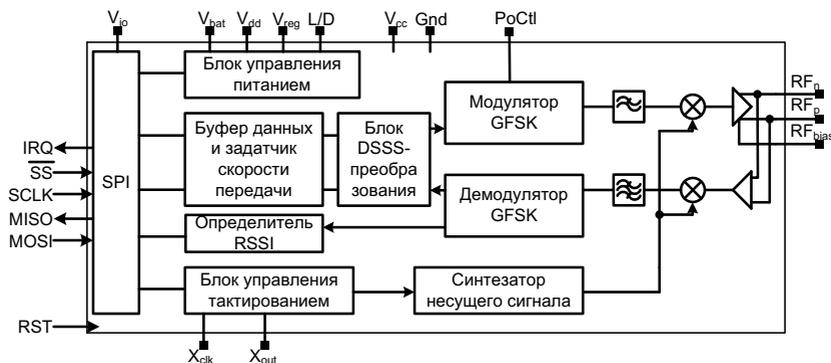


Рис. 2.3. Приёмопередатчик семейства LP

Семейство LP обладает более чувствительным приёмником и более мощным передатчиком (вплоть до 4 Дбм, что является максимально допустимым в диапазоне ISM), благодаря чему скорость передачи возросла на порядок.

Так как PРоC-устройства предназначены для встраиваемых систем, то компанией Cypress было уделено большое внимание энергопотреблению устройств. Как видно из схемы на рис. 2.3, был добавлен блок контроля питания, который позволяет управлять напряжением питания устройства, по-отдельности включать и отключать модули приёмника и передатчика. Была существенно снижена потребляемая мощность устройства, особенно в спящем режиме (до нескольких микроватт).

Устройства WirelessUSB LP поддерживают два режима передачи данных: чистый DSSS (до 250 кбит/с), и с добавлением GFSK (до 1 Мбит/с). При этом, скорость передачи при передаче может динамически меняться (15.6 кбит/с – 1 Мбит/с), а при приёме определяется автоматически.

CyFi

Семейство приёмопередатчиков CyFi является прямым развитием передатчиков WirelessUSB LP. Была сохранена внутренняя структура передатчика, возможные режимы и дальность передачи. Однако, за счёт использования новых технологий удалось снизить потребляемую устройством мощность в полтора раза. Фактически, CyFi является не новой технологией, а лишь ребрендингом старой.

Среди рассмотренных типов передатчиков наиболее привлекательным для исследования является наиболее новое семейство CyFi, поскольку в пресс-релизах компании Cypress говорится о будущей реализации микросхем PProC на его основе. Однако, для проведения практических экспериментов и последующей разработки необходимо наличие самих кристаллов. Единственным доступным на сегодняшний день представителем PProC является микросхема CYWUSB6953 на базе приёмопередатчика WirelessUSB LS.

Хотя семейство приёмопередатчиков WirelessUSB LS достаточно старо (почти 6 лет), они остаются конкурентными за счёт низкой цены контроллеров PProC. В поставленной задаче микросхем CYWUSB6953 допустимо, так как при их помощи возможно изучение принципов беспроводной передачи в WirelessUSB.

2.3. Исследование протокола передачи данных в WirelessUSB

При разработке устройств, использующих беспроводную связь, возникает множество проблем, связанных с устойчивостью работы канала связи. Например, одновременное использование канала двумя системами может привести к их некорректной работе. Кроме того, в большинстве разработок требуется контроль целостности сети и квитирование сообщений. Протокол WirelessUSB предназначен для решения этих проблем.

Использование протокола WirelessUSB в PРоC упрощает использование беспроводной связи: в протоколе автоматически разрешаются конфликты между устройствами, осуществляется контроль правильности и коррекция ошибок, и т.п.

Протокол WirelessUSB во многом схож с протоколом интерфейса USB, популярного в настоящее время. Как и в стандартном USB-протоколе, существуют три режима работы устройства: MASTER, SLAVE и DUAL.

SLAVE-устройства в WirelessUSB-сети не могут самостоятельно использовать канал, так как им требуется обязательная привязка к Master-устройствам, однако, именно Slave-устройства являются инициаторами обмена информацией. Привязка выполняется в двух режимах: стандартном и с использованием специального канала. (KISS-BIND), когда после разрыва связи производится автоматическая попытка её восстановления.

MASTER-устройство задаёт канал передачи и код псевдо-шума. Именно это устройство обрабатывает запросы на привязку от SLAVE-устройств. На одном канале может быть только одно MASTER-устройство. В противном случае система может работать со сбоями.

DUAL-устройства могут выполнять как роль SLAVE, так и MASTER, но только одну одновременно. Классический вариант- горячий резерв для мастерских устройств. Устройство в режиме SLAVE контролирует активность Мастера и в случае его выхода из строя переходит в состояние MASTER.

Протокол WirelessUSB позволяет организовать каналы связи типа "точка-точка" с двунаправленной передачей данных и "точка-много точек" для однонаправленной передачи (см. рис. 1.3). Существует модификация протокола для двунаправленной связи многих устройств.

При проектировании устройств с радио-связью возможно использование собственного протокола, но при этом требуется учитывать возможность одновременной работы со стандартным WirelessUSB. Плохая реализация протокола может привести к сбоям в работе обеих систем (например, подключение к каналу, уже занятому другим устройством).

Процесс работы протокола WirelessUSB можно условно разделить на следующие этапы:

- Инициализация ведущего устройства и занятие им свободного канала;
- Инициализация ведомого устройства и поиск им своего ведущего;
- Периодический обмен сообщениями;
- Контроль ошибок и их коррекция.

Контроль состояния линии и принимаемых пакетов осуществляется автоматически. О возникновении исключительных ситуаций пользователь может узнать по флагам, выставляемым функциями протокола. Далее рассмотрим алгоритмы выполнения основных этапов организации связи.

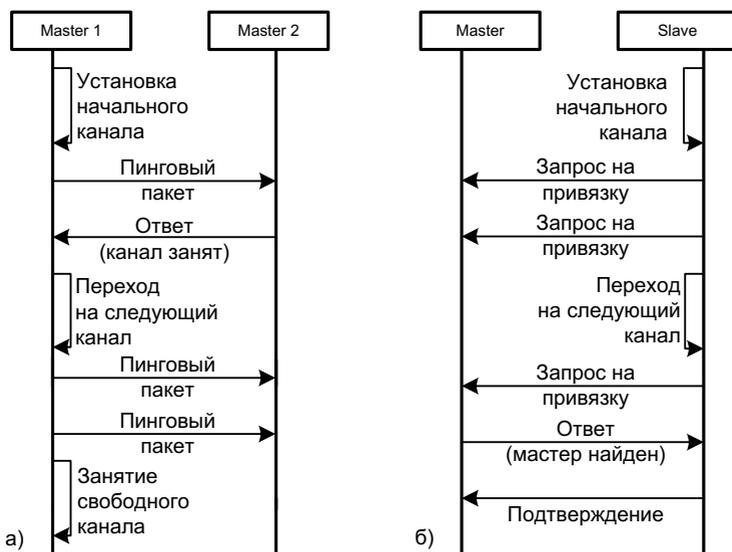


Рис. 2.4. Коммутация устройств в протоколе WirelessUSB (а- поиск мастером канала, б- привязка Slave-устройства)

Поиск свободного канала - первый этап организации связи устройств. Стартовый канал поиска определяется заводским номером передатчика (для начального распределения каналов). Устройство посылает нулевой запрос мастеру (ping). Если по истечении некоторого времени ответ не приходит, то канал считается свободным, и устройство занимает его. В противном случае производится переход на следующий канал, и операция повторяется.

Существует два режима связи устройств: обычный и KISS-BIND. В первом случае slave-устройство перебирает каналы связи, пока не находит соответствующее ему master-устройство. В KISS-BIND поиск ведётся только на каналах, выделенных под привязку устройств. При этом, по умолчанию, интенсивность передатчиков снижается, и поэтому устройства связываются лишь на близком расстоянии (kiss- "поцелуй").

Методы передачи информации для ролей Master и Slave различаются (рис. 2.5). В отличие от сетей радиально-базисного типа, весь обмен информацией осуществляется под управлением Slave-устройства, но по-прежнему возможен только между ведущим и ведомым устройствами.

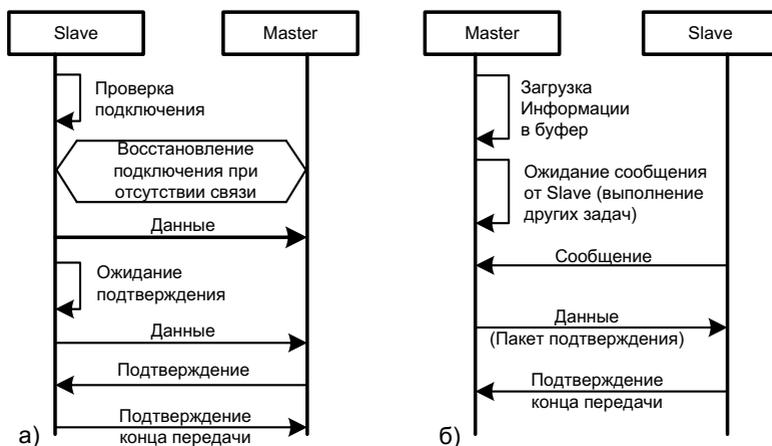


Рис. 2.5. Передача информации в WirelessUSB (а- Slave->Master, б- Master->Slave)

Вначале Slave-устройство проверяет состояние соединения. При его отсутствии производится попытка восстановления связи. Если связь восстановлена, то передаётся информационный пакет, после чего Slave ожидает ответа от мастерского устройства. Пакет подтверждения (ответ мастера) может быть пустым или содержать информацию, с помощью которой передаются сообщения по направлению Master->Slave. В любом случае, Slave-устройство обрабатывает пакет и посылает сообщение о конце цикла передачи.

Кроме аппаратной коррекции отдельных битов информации, в WirelessUSB производится **контроль правильности передачи пакета и коррекция ошибок**. Структура информационного пакета, применяемая в протоколе, приведена на рисунке 2.6.

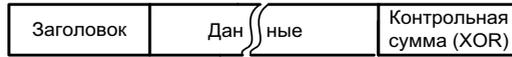


Рис. 2.6. Структура информационного пакета в протоколе WirelessUSB

В каждое сообщение добавлены два поля: заголовок и контрольная сумма. Заголовок пакета постоянен и используется для разделения информационных пакетов. Содержимое поля контрольной суммы вычисляется как “побитное исключающее или” (XOR) и используется для определения значений неправильно принятых бит. Поэтому, восстановить информацию возможно, если в каждом из разрядов находится не более одного искажённого бита. Пример коррекции ошибок в пакете приведён ниже на рис. 2.7.

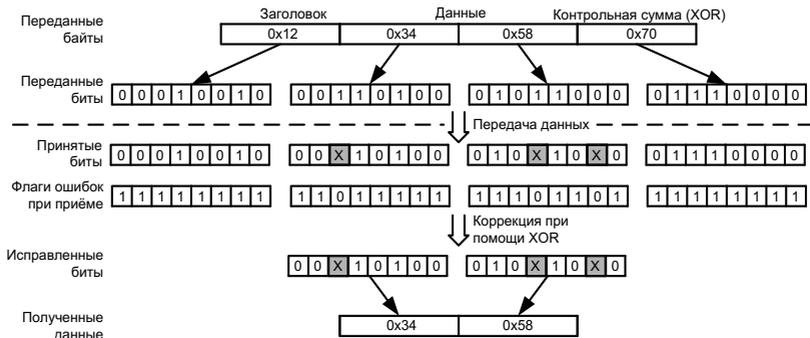


Рис. 2.7. Коррекция ошибок в пакете данных с помощью контрольной суммы

С ростом длины пакета возрастает вероятность возникновения неисправимой ошибки, и компания Cypress рекомендует использовать пакеты размером не более 20 байт.

По документации, вероятность неправильного приёма пакета составляет 10^{-4} при отсутствии сторонних сетей и 10^{-3} при их наличии. Однако, всегда может быть реализован механизм переспроса, который практически исключит возможность ошибки при передаче информации.

2.4. Исследование программной реализации протокола WirelessUSB

В данном разделе рассматривается программная реализация протокола WirelessUSB версии 1.1, который поставляется вместе с отладочными модулями CY3653, на которых производится разработка. Данная версия протокола предназначена для организации беспроводного канала связи по принципам "точка - точка" и "точка - много точек" между PRoC-контроллерами с приёмопередатчиками семейств LS и LR. Прочие версии протокола во многом схожи с рассматриваемой, различия состоят в интерфейсе с передатчиком и параметрах по-умолчанию.

Протокол WirelessUSB 1.1 реализован на языке Си в виде трёх файлов: "protocol.h", "protocol.c" и "Is_config.h". Также, для реализации задержек в протоколе требуется восьмиразрядный таймер, функции для настройки которого реализованы в файлах "timer.c" и "timer.h".

Для настройки протокола используется файл "Is_config.h". С его помощью можно задать режим работы протокола, тип устройства и тому подобное. Настройки задаются через директиву "#define", поэтому при изменении настроек достаточно произвести повторную компиляцию проекта. Основные настройки протокола приведены в табл. 2.2.

Настройки протокола WirelessUSB. Таблица 2.2 (начало)

Параметр	Принимаемые значения	Назначение
BIND_NO_HOP	-	Если опеределён то при команде перехода на следующий канал производится переход на 6 каналов.
DEFAULT_POWER_LEVEL	0-7	Настройка мощности сигнала по-умолчанию.
BIND_POWER_LEVEL	0-7	Настройка мощности передатчика во время привязки устройств.
KISS_BIND	-	Если параметр определён, то проверка связи и её восстановление производится по механизму Kiss-Bind.
PROTOCOL_ROLE_#X	#X= [SLAVE, MASTER, DUAL]	Роль устройства протоколе. Используется для маскирования функций, не используемых для выбранной роли.

Настройки протокола WirelessUSB. Таблица 2.2 (продолжение)

Параметр	Принимаемые значения	Назначение
RADIO_SLEEP_ENABLED	-	Разрешает радио переходить в спящий режим по окончанию передачи.
SLAVE_NUM_TRANSMITS	1-255	Количество попыток передачи информации Slave-устройством до ошибки передачи.
DEVICE_TYPE	1-7	Тип устройства. Привязка возможна только для устройств с одинаковым типом.
MAX_DATA_LEN	1-20 (рек.)	Максимальная длина пакета.
APP_TX(RX)_PACKET_SIZE	1-20 (рек.)	Используются, если принимаемые и передаваемые пакеты различаются по длине.
DEFAULT_CHANNEL	0-77	Канал, занимаемый при включении устройства.
DEFAULT_PN_CODE	0-48	Код псевдо-шума, который устанавливается при инициализации передатчика.

Кроме настроек, в составе WirelessUSB определены переменные, которые могут быть использованы для доступа к ресурсам протокола. Названия переменных и их назначение приведены в табл. 2.3.

Объекты протокола WirelessUSB. Таблица 2.3

Переменная	Тип	Назначение
sys_params	struct	Данная структура содержит информацию о передатчике и текущем канале.
tx_length	uint8	Длина передаваемого пакета.
valid_buffer	uint8[]	Битовая карта правильно принятых бит информации.
#X_#Y_data_toggle	uint8	Набор счётчиков приёма и передачи информационных сообщений (data).
#Y_packet	union	Наборы структур, хранящих принятый и посылаемый пакеты.
rx_data_length	uint8	Размер принятого сообщения в байтах.
bind_state	uint8	Текущее состояние соединения.
current_role	uint8	Текущая роль устройства. Доступен только для Dual-устройств.
rss_i_counter	uint8	Счётчик, используемый протоколом для вычисления относительной силы сигнала (rss_i)
rss_i_checked	uint8	Флаг действительности уровня сигнала (rss_i)
pn_code_table	uint8	Таблица используемых кодов псевдо-шума.
#X- роль устройства в протоколе (master или slave)		
#Y- направление передачи (rx-приём, tx- передача)		

Структура **sys_params** хранит информацию о текущих настройках приёмопередатчика и идентификаторе микросхемы (MID). Данный идентификатор для любого приёмопередатчика уникален, поэтому по нему может быть. Содержимое данной структуры приведено ниже:

```
sys_params
{
    .network_id- поле идентификаторов передатчика.
    {
        .channel - номер канала.
        .pn_code - код псевдошума.
        .seed     - зерно контрольной суммы.
        .pin      - номер pin, используемый в пакете.
        .signature - флаг модификации параметров
    }
    .bridge_mid -поле MID передатчика
    {
        .mid_1 - 1-й байт
        .mid_2 - 2-й байт
        .mid_3 - 3-й байт
        .mid_4 - 4-й байт
    }
}
```

Структуры **rx_packet** и **tx_packet** принадлежат к объединению **LS_DATA_PACKET**. Формат передаваемых сообщений в протоколе **WirelessUSB** фиксирован, поэтому возможно использование структуры-буфера, которое упростит обращение к отдельным полям отправляемого сообщения. Именно за это и отвечает **LS_DATA_PACKET**, в котором определены структуры для следующих типов пакетов:

```
union LS_DATA_PACKET
{
    .bind_request      - запрос на привязку.
    .bind_response     - ответ при привязке.
    .connect_request   - запрос на соединение.
    .connect_response  - ответ при соединении.
    .ping              - пакет-проверка связи.
    .data               - информационный пакет.
    .ack                - пакет подтверждения приёма.
    .null               - нулевой пакет.
}
```

Каждый из перечисленных типов пакетов имеет собственную структуру. Каждый из пакетов имеет заголовок со стандартными данными (MID передатчика, канал ведущего устройства и т.п.), после которого следует массив передаваемой информации и контрольная сумма. Следует отметить, что контрольная сумма используется только в пакетах, где идёт передача большого числа данных (не менее 8 байт). Для прочих пакетов производится проверка только по битовой карте.

Функции протокола WirelessUSB определены в файлах “protocol.c” и “protocol.h”. Как уже было сказано, существуют три режима работы протокола: MASTER, SLAVE и DUAL. Для каждого из них в протоколе реализуется свой набор функций. Каждая функция ограничена директивой препроцессора #ifdef для проверки заданной в настройках роли устройства. Поэтому, компилятором будут проигнорированы функции, не используемые в выбранном режиме протокола, что снижает размер итоговой подпрограммы протокола.

Для всех ролей существует набор базовых функций, таких как инициализация, доступ к переменным протокола и информации о принятых и полученных пакетах. Базовые функции протокола и их описание приведены в табл. 2.4.

Основные функции протокола WirelessUSB. Таблица 2.4

Функция	Описание
UINT8 protocol_init (void)	Запуск радио и инициализация установленной роли протокола. Для DUAL-устройство после инициализации требуется задать исходную роль. Функция всегда возвращает значение LS STATUS SUCCESS.
UINT8 protocol_get_rx_packet_length (void)	Возвращает длину последнего принятого сообщения.
void protocol_send_ack (void)	Отправка сообщения, подтверждающего приём информации (GOOD ACK).
void * protocol_get_tx_pkt (void)	Возвращает указатель на структуру tx_packet.
void * protocol_get_rx_pkt (void)	Возвращает указатель на структуру rx_packet.

Остальные функции протокола применяются только для одной из ролей устройства. Для роли DUAL, когда устройство может переключаться между ведущим и ведомым режимами протокола, доступны все приведённые ниже функции. Описание функций для ведущего устройства приведено в табл. 2.5.

Функции протокола WirelessUSB для ведущего устройства. Таблица 2.5

Функция	Описание
void protocol_set_role_master (void)	Функция устанавливает значения структуры sys_params, задаёт исходные параметры радиопередатчика и активирует режим поиска свободного канала (master_ping_mode). Для устройств Dual также производится переключение в роль Master.
UINT8 master_ping_mode (UINT8 reason)	Поиск незанятого канала. Параметр reason задаёт канал, с которого начинается поиск. (см. алгоритм на рис. 2.4)
UINT8 master_bind_mode (void)	Переход в режим коммутации со Slave-устройством. (см. алгоритм на рис. 2.4)
UINT8 master_data_mode (void)	Проверка наличия новой информации в буфере приёма радио. (см. алгоритм на рис. 2.5) и её обработки.
UINT8 master_send_packet (UINT8 data_length)	Отправка данных длиной data_length из буфера tx_packet. Передача производится в ответ на сообщение, полученное от Slave-устройства. Всегда возвращает TRUE (сохранение совместимости с режимом Slave).
UINT8 master_is_tx_buffer_empty (void)	Проверяет, пуст ли буфер передачи (tx_packet). Требуется для контроля завершения передачи информации. Возвращает TRUE, если буфер пуст, иначе- FALSE.
UINT8 master_read_rx_packet (UINT8 *data_ptr, UINT8 num bytes)	Считывает информацию указанной длины (обычно длины пакета) в указанный буфер. Возвращает длину считанной информации.

Функция **master_ping_mode()** работает в соответствии с алгоритмом, приведённым на рис. 2.4. Если параметр reason равняется PING_REASON_NORMAL, то поиск начинается с верхнего канала (первого канала по-умолчанию), иначе поиск начинается с текущего номера канала.

Функция возвращает значение `LS_STATUS_SUCCESS`, как только канал будет найден. Если все каналы заняты, то устройство остаётся в режиме поиска канала неограниченно долго.

При вызове функции `master_bind_mode()` (см. алгоритм на рис. 2.4) приёмопередатчик переводится на выделенные протоколом каналы привязки и в течении некоторого времени (зависит от номера канала) ожидает запроса на привязку от устройства с ролью Slave. Если запрос не приходит, то функцией возвращается значение `LS_STATUS_FAILED`.

Если же запрос пришёл, то Master сообщает свои параметры, по которым может быть найден (рабочий канал, PN-код и MID), после чего вызывается `master_ping_mode()` для возвращения устройства на рабочий канал.

Функция обработки принятой информации `master_data_mode` (см. алгоритм на рис. 2.5) проверку наличия новой информации в буфере, проверку формата сообщения и контрольной суммы. Также при её помощи осуществляется механизм квитирования при обмене информацией.

Если в буфере приёма радио имеется новая информация, то устройство посылает подтверждение приёма (`ack`) или данные, после чего ожидает ответное подтверждение. Если подтверждение не приходит, то производится проверка и попытка восстановления связи. Функция возвращает следующие значения:

`DATA_RECEIVED`- принят информационный пакет

`ACK_RECEIVED`- принято подтверждение о приёме ответного пакета

`DATA_IDLE`- новой информации не поступило

Функция `master_data_mode()` в программе должна вызываться регулярно, иначе ведущее устройство может не успеть ответить за время

ожидания Slave-устройства, что приведёт к сбою в работе ведомого устройства.

Описание функций протокола для ведомого устройства приведено в табл. 2.6.

Функции протокола WirelessUSB для ведомого устройства. Таблица 2.6

Функция	Описание
void protocol_set_role_slave (void)	Инициализация структуры sys_params.
UINT8 slave_bind (void)	Функция отвечает за привязку ведомого Slave-устройства к ведущему. Реализована отдельно для стандартного режима и KISS-BIND.
UINT8 slave_reconnect (void)	Функция реализует поиск Master-устройства по принятым от него параметрам. На установленном канале посылается сообщение CONNECT_REQ. Если на сообщение приходит ответ, то соединение считается установленным. Функция возвращает LS_STATUS_SUCCESS. Если ответ отсутствует, то проверяются соседние каналы, а потом- ещё раз исходный. При отсутствии ответа возвращается LS_STATUS_FAILED.
UINT8 slave_send_packet (UINT8 data_length)	Функция выполняет отправку сообщения длины data_length из буфера tx_packet. Первым делом производится проверка активности соединения. В случае разрыва производится попытка возобновления связи. Далее происходит формирование пакета, при этом производится контроль длины сообщения. Далее пакет передаётся по радио, после чего ожидается подтверждение приёма. Функция возвращает следующие значения: TX_TIMEOUT_ERR- не удалось установить связь с Master-устройством. INVALID_DATA- длина передаваемого сообщения больше длины пакета. DATA_SENT- передача данных произведена успешно. ACK_DATA- передача произведена успешно, и в ответ на посланное сообщение MASTER-устройство отправило информационный пакет.

В стандартном режиме функция **slave_bind()** перебирает выделенные каналы привязки, отправляя для каждого из них запрос на соединение. Если ни на одном из каналов устройство не получит ответа, то функция возвращает LS_STATUS_FAILED.

Если ответ получен, из принятого пакета `bind_response` считываются параметры Master-устройства, производится переход на исходный канал, и вызывается функция `slave_reconnect()` для установления соединения с Мастером на рабочем канале. Если соединение установлено, то возвращается `LS_STATUS_SUCCESS`, иначе- `LS_STATUS_FAILED`.

Кроме рассмотренных ранее функций, в роли Dual в протоколе используется функция **`protocol_get_role()`**, которая возвращает текущую роль устройства.

Для демонстрации возможностей технологии WirelessUSB был разработан ряд тестовых примеров, которые приведены в приложении 1. Примеры оформлены в виде лабораторной работы, что позволяет использовать их в образовательных целях.

В данной лабораторной работе рассматривается организация беспроводного канала связи между микросхемами CYWUSB6953 как с использованием протокола WirelessUSB, так и без него. В работе показывается преимущество второго способа при совместной работе нескольких сетей WirelessUSB за счёт автоматического устранения конфликтов по частотам.

Исследование показало, что рассматриваемая версия протокола неудобна для практического использования. Во-первых, файлы `protocol.h` и `protocol.c` очень громоздки, и их модификация и отладка сильно затруднены. Кроме того, в протоколе используются сложные структуры данных и статические массивы, которые по-разному компилируются в различных версиях PSoC Designer. Поэтому, при разработке модификации протокола следует произвести переработку кода с целью его упрощения и универсализации.

При проведении исследования было установлено, что протокол корректно работает при малом размере программы. Однако, при увеличении сложности системы периодически возникают потери связи, которые исправляются только многократным перепрограммированием устройства. Причину установить не удалось, однако было сделано предположение, что причиной является некорректная работа с флэш-памятью при установке соединения. Поэтому, при разработке модификации протокола потребуется также проверить алгоритмы работы функций соединения и попробовать определить причину сбоев в исходном протоколе.

3. Изучение возможностей PРоС-кристалла по сбору и обработке первичной аналоговой и дискретной информации

3.1. Архитектурные особенности PРоС-кристаллов

PРоС-кристаллы компании Cypress основаны на технологии PSoC. “Программируемая система на кристалле” включает, кроме ядра контроллера, набор конфигурируемых цифровых и аналоговых блоков, которые реализуют различные функции (счётчики, коммуникационные интерфейсы, АЦП и т.д.).

В PSoC возможна динамическая переконфигурация элементов, за счёт чего достигается большая гибкость системы. Основные блоки контроллера (вычислительное ядро, память и т.п.) обладают фиксированной структурой, что позволяет добиться высокой производительности системы в целом.

Ряд востребованных элементов в PSoC-контроллере (например, передатчик WirelessUSB или контроллер I2C) имеют сложную структуру, и поэтому на кристалле также реализуются в виде модулей фиксированной структуры, далее называемых системными ресурсами. К данным ресурсам относятся следящий таймер, таймер спящего режима, схема контроля питающего напряжения, контроллер прерываний, драйверы портов ввода-вывода, схемы генераторов тактовых импульсов, а также контроллеры коммуникационных интерфейсов, таких как I2C, WirelessUSB и USB.

Обобщённая структура PSoC-кристалла приведена на рис. 3.1.

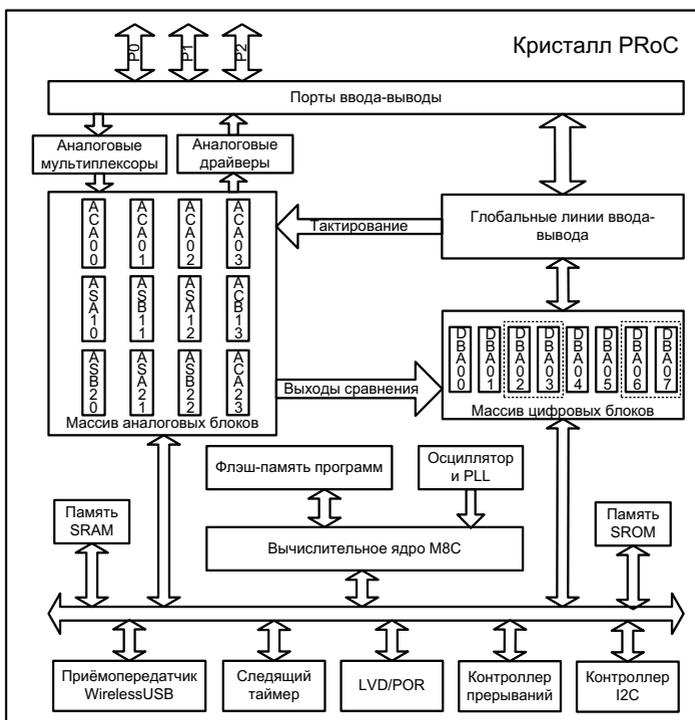


Рис. 3.1. Структура кристалла PRoC-контроллера

Вычислительное ядро M8C имеет восьмиразрядную гарвардскую архитектуру с производительностью до 4 MIPS (миллионов операций в секунду). Тактовая частота вычислительного ядра лежит в диапазоне от 93,7 кГц до 24 МГц. В функции ядра входит последовательное выполнение команд программы, хранящейся в энергонезависимой флэш-памяти.

Кроме хранения программ, **флэш-память** может хранить и пользовательские данные. В частности, ведомые устройства протокола WirelessUSB сохраняют во флэш-память параметры ведущего устройства для ускорения подключения после запуска. Доступ к памяти производится посредством специальных функций, хранимых в ROM. Во флэш-память встроены механизмы защиты, которые обеспечивают сохранность кода.

Цифровая подсистема PSoC позволяет реализовать периферийные устройства для преобразования цифровых сигналов. При помощи данной подсистемы могут быть реализованы различные счетчики, таймеры, ШИМ, генераторы случайных последовательностей и т.д. В PSoC конфигурируемые модули (блоки) разбиты на "ряды". Структура ряда цифровых блоков фиксирована, и количество блоков в системе определяется числом рядов. Всего их может быть от одного до четырёх.

Цифровая подсистема состоит из четырех компонентов (рис. 2.7.):

- глобальные цифровые связи (Global Digital Interconnect, GDI);
- связи в массиве цифровых блоков (Array Digital Interconnect, ADI);
- связи внутри ряда цифровых блоков (Row Digital Interconnect, RDI);
- цифровые базовые (DBB) и коммуникационные (DCB) блоки.

Цифровые блоки PSoC (рис. 2.11.) подразделяются на два типа - базовые (Digital Basic Blocks, DBB) и коммуникационные (Digital Communication Blocks, DCB). В каждом ряде присутствует по два блока каждого типа. Первый тип блоков позволяет реализовать такие периферийные модули, как таймеры, счетчики и ШИМ, а также модуль контроля с помощью циклического избыточного кода (CRC) и генератор псевдослучайной последовательности (PRS). Коммуникационные блоки, помимо указанных функций, позволяют создавать контроллеры интерфейса SPI или полнодуплексный UART.

Каждый цифровой блок построен на базе семи 8-битных регистров - трех конфигурационных, определяющих тип выполняемой блоком функции и конфигурацию входов и выходов блока, а также трех регистров данных и регистра управления функцией. Структурная схема цифрового блока приведена на рис. 3.2.

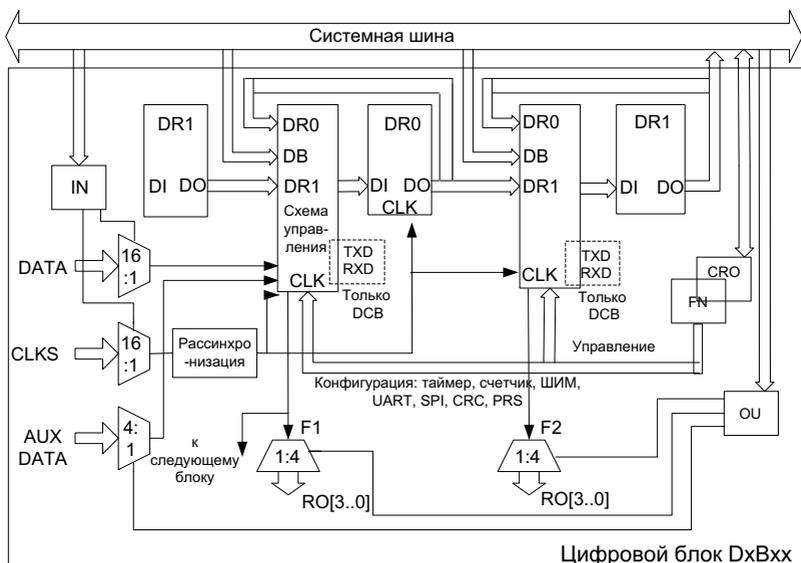


Рис. 3.2. Структура цифрового блока PSOC

Аналоговая система в PSOC бывает двух типов: полной и упрощённой. Полная аналоговая система включает блоки на переключаемых конденсаторах, за счёт чего на её основе могут быть построены фильтры, усилители, компараторы, а также ЦАП и АЦП различных типов. В результате, при помощи кристаллов с данным типом аналоговой подсистемы могут быть реализованы достаточно сложные аналоговые тракты преобразования сигналов, за счёт чего достигается экономия на внешних элементах.

В микросхемах PSoC системный ресурс приёмопередатчика WirelessUSB требует больших аппаратных затрат, в то время как требуется обеспечить низкую стоимость устройства. Поэтому, для большинства PSoC-микросхем (в т.ч. и CYWUSB6953) разработчики решили использовать упрощённую аналоговую подсистему (УАП). Архитектура УАП (рис. 3.3) отличается от базовой как системой ввода сигналов, так и содержанием самой аналоговой колонки.

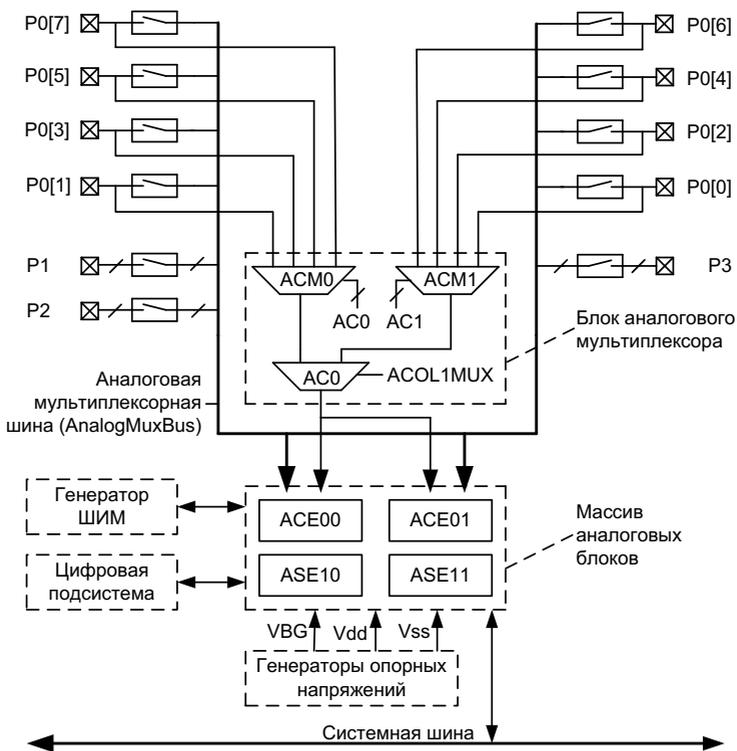


Рис. 3.3. Структура упрощённой аналоговой подсистемы

Если в полной аналоговой подсистеме для каждой колонки используется отдельный мультиплексор, то в на две колонки используется только один (AC1), к которому может быть подключена любая из колонок.

Также, в упрощённой аналоговой подсистеме появилась новая линия ввода- аналоговая мультиплексорная шина (AnalogMuxBus). Она позволяет подключать аналоговые блоки к любому из выводов микросхемы. Поэтому, в кристаллах с УАП могут быть реализованы устройства с большим числом каналов, например, многоканальный АЦП или многовходовые сенсорные органы управления.

Структура **аналоговых колонок** в УАП (рис. 3.4.) кардинально отличается от базовой модели. Из стандартного аналогового СТ - блока удалены матрица резисторов и маломощный компаратор. В результате он трансформирован в СЕ - блок, функции которого ограничиваются выполнением операции сравнения. При этом, сохранились блоки аппаратного преобразования сигналов на выходах компараторов (LUT). Вместо блока на переключаемых конденсаторах (SC) введён тип SE, построенный на управляемом источнике тока с цепью управления зарядом конденсатора. В современных реализациях УАП количество аналоговых колонок равняется двум.

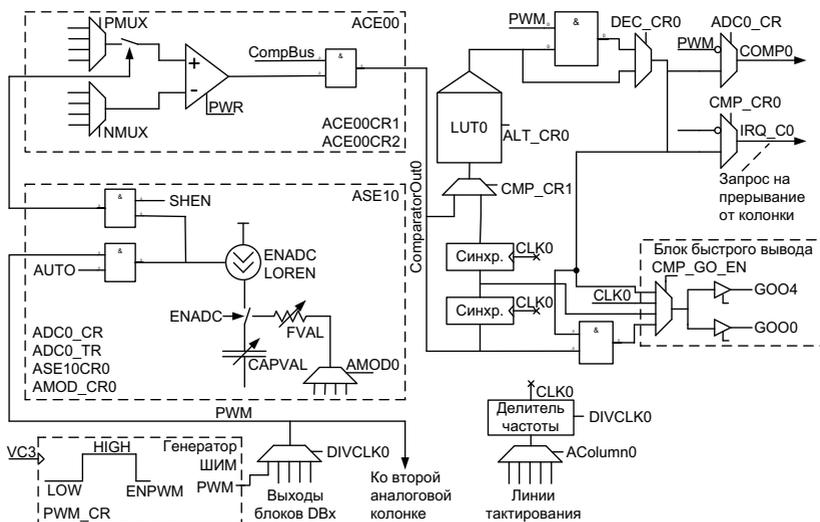


Рис. 3.4. Структура аналоговой колонки типа SE

Блок ACE представляет собой компаратор с мультиплексированными входами. Выход компаратора подаётся на логику линий сравнения (LUT- Look-Up-Table), которая формирует сигнал на выходной линии сравнения (COMPx). Кроме того, возможны прямое подключение выхода компаратора к глобальной шине ввода-вывода и генерация прерываний по результатам сравнения.

Всего в упрощённой аналоговой системе присутствует две линии сравнения. Настройки логики преобразования LUT на шинах компараторов задаются регистрами ALT_CR0 и ALT_CR1.

Блок ASE является управляемым источником тока с подключенным к нему конденсатором. При включённом источнике, напряжение на конденсаторе линейно зависит от времени, что может быть использовано для построения АЦП интегрирующего типа. Также в состав блока ASE входят цепи управления разрешением работы источника и разряда конденсатора.

В упрощенную аналоговую подсистему включен внутренний **генератор ШИМ** для реализации сигналов управления. Такой блок на две колонки один, поэтому на кристалле с упрощенной подсистемой может быть, например, сконфигурирован лишь один АЦП. Генератор ШИМ (тактируется сигналом VC3 и может управлять работой источника тока блока ASE и сбросом заряда конденсатора в нём).

3.2. Пользовательские модули и возможность их реализации в среде PSoC Designer 4.3

Для реализации типовых устройств в среде проектирования PSoC Designer предусмотрены так называемые пользовательские модули (ПМ). Каждый пользовательский модуль представляет из себя набор подпрограмм, реализующий настройку аппаратной части модуля, стандартные функции взаимодействия с модулем и т.п.

Среда PSoC Designer реализует графический интерфейс, который позволяет разместить пользовательские модули на цифровом и аналоговом массивах, подключить их входы и выходы к необходимым линиям и задать исходные параметры модуля. Код подпрограмм на ассемблере генерируется автоматически. Большинство функций

модифицировать нельзя, но для некоторых (например, для обработчиков прерываний) в коде при помощи специальных макросов объявлены области, в которые пользователь может внести необходимые дополнения.

Условно все пользовательские модули можно разделить на 4 группы: цифровые, коммуникационные, аналоговые и интерфейсные.

Цифровые пользовательские модули для своего функционирования используют цифровые блоки кристалла. Для микросхемы CYWUSB6953 доступны следующие ПМ:

- счётчики с разрядностью 8, 16, 24 и 32 бита;
- таймеры с разрядностью 8, 16, 24 и 32 бита;
- ШИМ с разрядностью 8 и 16 бит;
- двухфазный ШИМ с мёртвой зоной;
- логические буферы и инверторы.

Таймеры или **счётчики** являются наиболее часто используемыми цифровыми устройствами. В частности, в протоколе WirelessUSB при помощи таймера реализуются задержки при ожидании ответа от устройства, поэтому необходимо изучить структуру данного модуля. Счётчик с разрядностью 8 бит строится на основе одного цифрового блока. При использовании разрядности выше 8 бит, для организации модуля объединяются несколько соседних блоков при помощи цепей переноса. Схема восьмиразрядного блока таймера/счётчика приведена на рис. 3.5.

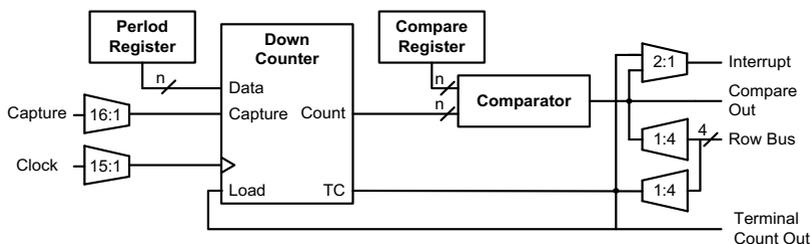


Рис. 3.5. Схема блока таймера/счётчика

Разрешение счёта capture может быть подано с входных и выходных линий цифрового ряда, выходов сравнения аналоговых колонок или константных сигналов. Тактирование производится при помощи линий тактирования микросхемы и цифрового ряда.

Коммуникационные пользовательские модули служат для реализации стандартных интерфейсов обмена с внешними по отношению к микросистеме устройствами. В среде проектирования PSoC Designer для CYWUSB6953 коммуникационные модули позволяют организовать следующие интерфейсы:

- I2C;
- I2C с поддержкой протокола доступа к внешней памяти;
- WirelessUSB;
- приёмопередатчик UART;
- отдельные модули приёмника и передатчика UART;
- IrDA-приёмник и передатчик;
- SPI в режимах ведущего и ведомого устройств .

В PSoC кристаллах интерфейсы I2C и WirelessUSB реализуются на кристалле аппаратно в виде отдельных системных ресурсов. Остальные пользовательские коммуникационные модули используют специальные блоки типа DCB (Digital Communication Block).

Коммуникационный модуль WirelessUSB реализует взаимодействие с системным ресурсом приёмопередатчика через программную реализацию последовательного интерфейса SPI. Коммуникационный модуль обеспечивает управление интерфейсом и низкий уровень программной реализации радиоканала. В модуль входят функции инициализации передатчика, управления режимом работы, выбора канала, отправки и приёма сообщений. Более подробно пользовательский модуль рассмотрен в приложении 1.

Аналоговые пользовательские модули в УАП предназначены для преобразования аналоговых сигналов в дискретную информацию. Другие возможности использования упрощённой подсистемы не предусмотрены. Число доступных аналоговых ПМ в среде PSoC Designer невелико: аналоговый компаратор, а также восьми- или десятиразрядный аналого-цифровые преобразователи.

В упрощённой аналоговой подсистеме **компаратор** реализуется при помощи одного СЕ-блока. Структурная схема компаратора приведена на рис. 3.6.

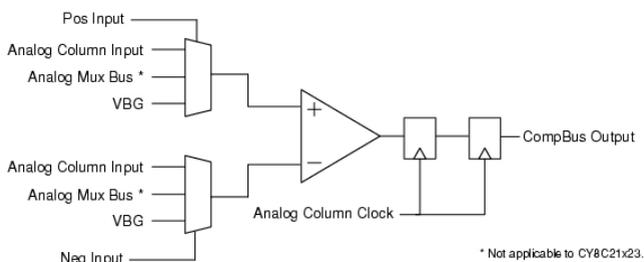


Рис. 3.6. Структурная схема компаратора

Структура компаратора позволяет настроить не только источник входного сигнала, но и дифференциальный сигнал. За счёт этого можно производить сравнение внешних аналоговых сигналов без использования аналого-цифрового преобразователя. Использование триггеров на выходе компаратора обеспечивает фильтрацию сигнала для исключения помехи.

АЦП последовательного счёта используется только в микросхемах с упрощённой аналоговой подсистемой (семейства CY7C603xx, CY8C21xxx, CYWUSB6953). АЦП данного использует два аналоговых блока (СТ и SC) одного столбца, один цифровой блок для накопления результата и выделенный ШИМ-генератор для тактирования. Структурная схема преобразователя приведена на рис. 3.7.

За то время, пока сигнал ШИМ находится в состоянии «0», накапливающий конденсатор разряжается, а конденсатор, удерживающий входной сигнал, заряжается. Момент спада сигнала ШИМ означает окончание преобразования.

В аналоговой подсистеме существует только один генератор ШИМ ADCPWM, используемый обоими аналоговыми столбцами, поэтому только один модуль такого типа может входить в проект. Разрядность преобразователя определяется выбором периода управляющего ШИМ и позволяет получать результат с разрядностью до 12 бит. Однако точность преобразователя составляет максимум 10 бит.

Достоинством данного АЦП данного типа является линейность преобразования и простота реализации. Так как на вход АЦП могут быть поданы эталонные сигналы (например, VBG), то возможна калибровка модуля для повышения его точности. Фактически структура упрощённой аналоговой подсистемы сложна ровно настолько, чтобы реализовать данный тип АЦП.

Интерфейсные пользовательские модули позволяют упростить взаимодействие со стандартными элементами управления (например, памятью) средствами индикации, такими как светодиоды или жидкокристаллический индикатор (ЖКИ). Пользовательские модули данного типа не используют конфигурируемых блоков или системных ресурсов микросхемы. К интерфейсным пользовательским модулям относят:

- Модуль доступа к флэш-памяти контроллера
- Программная реализация интерфейса I2C
- ЖКИ
- Семисегментный индикатор
- Светодиод

3.3. Проблемы использования аналоговой подсистемы кристалла CYWUSB6953 в среде проектирования PSoC Designer

По итогам исследования была произведена разработка ряда проектов, демонстрирующих возможности CYWUSB6953 по сбору и первичной обработке информации. Описания данных проектов в виде лабораторной работы приведены в приложении 2. В работе рассмотрены достаточно распространённые примеры использования беспроводной связи: манипулятор с дискретным (клавиатура) и непрерывным (двухосный джойстик) управлением. При этом, использована возможность динамической переконфигурации устройства.

В разработанных системах реализован беспроводной канал передачи между двумя микросхемами CYWUSB6953, после чего информация передаётся на инструментальную ЭВМ, где производится отображение полученной информации при помощи терминальной программы ComViewer, описание которой приведено в приложении 3.

При разработке проектов было обнаружено, что аналоговая подсистема CYWUSB6953 не полностью поддерживается средой PSoC Designer. В частности, в редакторе межсоединений отсутствует аналоговая мультиплексорная шина, и её настройка возможна только из пользовательской программы путём записи нужных значений в регистры управления.

Реализация аналогового ввода через аналоговые мультиплексоры также оказалась затруднена. Оказалось, что независимо от настроек, сделанных при помощи редактора межсоединений, в сгенерированном коде инициализации в регистры управления мультиплексорами записываются нули, и на входы аналоговых колонок остаётся подключённым вывод P0.0 микросхемы. Для исправления этой ситуации

также потребовалась запись в регистры правильных значений после модификации программы.

Таким образом, ни один из способов ввода аналоговой информации в среде проектирования PSoC Designer 4.3 не поддерживается полностью. В последующих версиях среды эти недостатки также не были устранены.

С другой стороны, в материалах компании Cypress указывается, что невозможно одновременное использование двух АЦП в упрощённой аналоговой подсистеме, так как блок генератора ШИМ в УАП всего один [8]. Однако, оказалось возможным подключить к одному генератору сразу два АЦП. Таким образом, на микросхеме CYWUSB6953 возможно реализовать двухканальный аналого-цифровой преобразователь с независимой настройкой каналов. Такая возможность расширяет диапазон возможного применения микросхемы, хотя реализация такого модуля требует использования всех ресурсов аналоговой подсистемы.

4. Разработка модификации протокола WirelessUSB

4.1. Краткая характеристика разработанного протокола

Результаты исследования технологии WirelessUSB показали, что программная реализация протокола, предложенная компанией Cypress, имеет ряд недостатков. Наиболее существенной проблемой является несовместимость протокола с новыми версиями среды PSoC Designer. Служба технической поддержки компании Cypress указывает, что несовместимость вызвана различной компиляцией статических массивов, но при исследовании протокола были выявлены и другие проблемы. Недостаток средств отладки и их неприменимость для отладки протокола исключил возможность поиска ошибок в исходном протоколе.

Поэтому, было принято решение реализовать протокол "с нуля", руководствуясь алгоритмами работы протокола WirelessUSB и постепенно реализуя его функциональность. При этом, с точки зрения пользователя, пространство имён базового протокола должно быть сохранено.

С точки зрения автора, наиболее эффективным решением была бы реализация протокола в виде пользовательского модуля для среды PSoC Designer. В таком случае, можно было бы упростить настройку протокола за счёт графического интерфейса, исключить возможные ошибки при настройке таймера, приёмопередатчика и флэш-памяти, реализовать автоматическую модификацию кода при изменении настроек и пользовательского модуля. Однако, среда PSoC Designer не имеет средств для создания пользовательских модулей, а анализ файловой структуры среды проектирования показал, что введение пользовательского модуля "вручную" крайне затруднительно. Поэтому, протокол решено было выполнить на языке Си в виде отдельных файлов.

Разработка и отладка протокола была произведена в среде PSoC Designer 4.3. После этого, была проверена корректность его работы в последней версии среды PSD 5.0 для двух доступных компиляторов: Imagecraft и HI-TECH. Были реализованы следующие возможности протокола WirelessUSB:

а) Общие свойства:

- многоканальность работы протокола,
- поддержка ролей MASTER, SLAVE и DUAL,
- алгоритмы взаимодействия устройств соответствуют разделу. 2.3,
- исправление ошибок приёма бит в соответствии с кодами Голда,
- проверка целостности пакетов по контрольной сумме,
- псевдо-случайный выбор начального канала.

б) Ведущее устройство:

- поиск незанятого канала,
- использование специального канала для привязки устройств,
- буферизация ответного сообщения для последующей передачи,
- обработка всех типов пакетов в одной функции.

в) Ведомое устройство:

- поиск ведомым устройством ведущего по типу устройства,
- сохранение информации о ведущем устройстве во флэш-памяти,
- многократная посылка пакетов при отсутствии ответа,
- перепоключение к ведущему устройству при потере связи.

При разработке протокола упор был сделан на реализацию устойчивого обмена в наиболее простом режиме. Не были реализованы следующие возможности, опционально доступные в базовом протоколе:

- перевод передатчика в спящий режим на время бездействия
- привязка устройств по алгоритму KISS-BIND
- контроль относительной интенсивности сигнала

При разработке протокола особое внимание было уделено проверке и исключению потенциальных проблем, которые были определены в разделе 2.4. Наиболее вероятными источниками сбоев считались следующие участки протокола: таблица кодов псевдо-шума, сложное объединение `LS_DATA_PACKET` и работа с энергонезависимой флэш-памятью.

Для каждой из возможных проблем была разработана программа для проверки предположения.

Проблема с неправильной интерпретацией кодов псевдо-шума различными компиляторами подтверждена компанией Cypress. Поэтому, требовалось лишь получить подтверждение данной проблемы. Оказалось, что компиляторы Imagecraft 1.69 и HI-TECH действительно неправильно заполняют массив кодов псевдо-шума, так как при считывании их значения не соответствуют заданной в программе таблице и, соответственно, кодам Голда. Поэтому, не следует ожидать корректного приёма данных приёмником WirelessUSB.

Оказалось, что для всех используемых компиляторов не возникает проблем при работе с объединениями (`union`), которые используются для описания структур пакетов с целью упрощения обращения к отдельным полям. Поэтому, структура `LS_DATA_PACKET` можно перенести в модификацию протокола без изменений.

При инициализации протокола для ведомого устройства из флэш-памяти считываются параметры ведущего устройства при последнем успешном соединении. Для проверки правильности считанных настроек используется бит `signature`. Если он установлен, то производится подключение по считанным параметрам. После перепрограммирования информация о подключении в ведомом устройстве стирается, и поиск ведущего идёт на каналах привязки. Поэтому, после включения

ведомого устройства с очищенной памятью требуется также перезагрузка ведущего. Исключить данную ситуацию можно, например, при помощи совмещения рабочих каналов и каналов привязки.

4.2. Программная реализация модифицированного протокола

Как уже было сказано, в базовой версии протокол WirelessUSB реализован в виде трёх больших файлов размером до двух тысяч строк. С точки зрения разработчика, такая реализация крайне неудобна для восприятия и модификации. Поэтому, было решено разбить протокол по логическим составляющим меньшего размера. Файловая структура модифицированного протокола приведена в табл. 4.1.

Файловая структура модифицированного протокола. Таблица 4.1

Имя файла	Описание
WirelessUSB_m.h	Основной включаемый файл протокола. Подключает прочие заголовочные файлы и содержит пользовательские настройки протокола WirelessUSB
prot_params.h	Файл содержит параметры протокола WirelessUSB: информацию о состоянии приёмопередатчика, буфера приёма и передачи
prot_config.h	Файл определяет внутренние настройки протокола, изменение которых пользователем нежелательно.
prot_base.h	В файлах объявлены и определены основные функции протокола WirelessUSB (табл. 2.3), которые используются для всех ролей устройств
prot_base.c	
prot_master.h	В файлах объявлены функции протокола для различных ролей устройства. Файлы подключаются в соответствии с выбранной ролью устройства.
prot_slave.h	
prot_dual.h	
prot_master.c	В файлах определены функции протоколов для различных ролей. Список функций для каждой из ролей и их описания приведены в таблицах 2.4-2.6
prot_slave.c	
prot_dual.c	
prot_packets.h	В данном заголовочном файле объявлены структуры всех типов пакетов протокола.
prot_support.h	В данных файлах объявлены и определены функции, обеспечивающие работу протокола: проверку контрольной суммы, исправление ошибок и т.п.
prot_support.c	
timer.h	Содержат подпрограммы управления таймером для реализации временных задержек и таймаутов при работе
timer.c	

На рис. 4.1. приведена файловая структура протокола с указанием расположения стандартных функций по файлам.

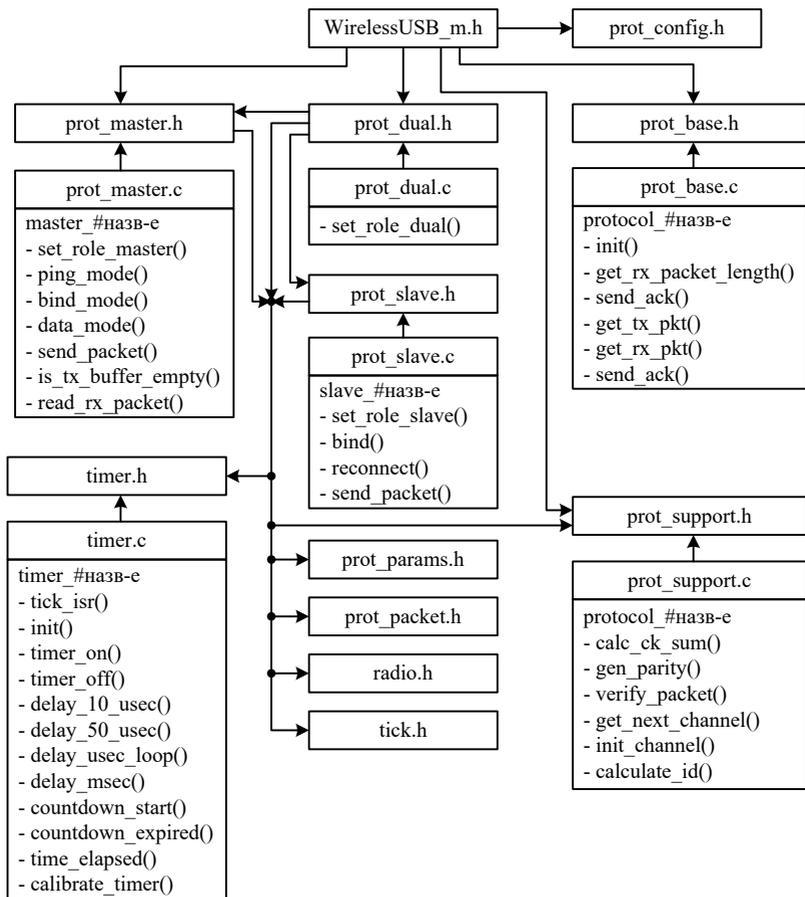


Рис. 4.1. Файловая структура протокола

Файлы radio.h и tick.h описывают функции пользовательских модулей WirelessUSB и Timer8, которые используются в протоколе. Данные функции генерируются средой автоматически и к протоколу не относятся. Подробно функции данных пользовательских модулей рассмотрены в приложении 1.

Поскольку ставилась задача максимально приблизить модификацию протокола к базовой версии, все функции, параметры и внутренние настройки базового протокола были сохранены. Поэтому, для пользовательской программы неважно, какая версия протокола используется, что упрощает перенос проектов на разработанную версию протокола. За счёт исключения части опций протокола, было снижено число пользовательских настроек протокола:

Настройки модифицированного протокола. Таблица 4.2.

Параметр	Принимаемые значения	Назначение
DEFAULT_POWER_LEVEL	0-7	Настройка мощности сигнала в рабочем режиме.
BIND_POWER_LEVEL	0-7	Настройка мощности передатчика во время привязки устройств.
PROTOCOL_ROLE_#X	#X=[SLAVE, MASTER, DUAL]	Роль устройства протоколе. Используется для маскирования функций, не используемых для выбранной роли.
SLAVE_NUM_TRANSMITS	1-255	Количество повторных попыток передачи информации Slave-устройством до возникновения ошибки передачи.
DEVICE_TYPE	1-7	Тип устройства. Привязка возможна только для устройств с одинаковым типом.
MAX_DATA_LEN	1-255	Максимальная длина информационного пакета.
DEFAULT_CHANNEL	0-77	Канал, занимаемый при включении устройства. При входе в режим привязки, канал меняется.
DEFAULT_PN_CODE	0-48	Код псевдо-шума, который устанавливается при инициализации передатчика.
BIND_ON_ALL_CHANNELS	-	Если параметр определён, то привязка осуществляется на всех каналах

Все функции модифицированного протокола удовлетворяют алгоритмам, рассмотренным в п. 2.3. Функции, обеспечивающие работу протокола и таймера, были перенесены из исходного протокола без изменений.

Таблица кодов псевдо-шума при помощи ассемблерной вставки с прямым указанием адреса была размещена в той же области флэш-памяти, что и структура `sys_params`. Тем самым была обеспечена работа протокола во всех рассмотренных средах проектирования. При этом, размер доступной для пользовательских программ памяти не изменился.

Для упрощения процедуры перепрограммирования в протокол был добавлен режим `BIND_ON_ALL_CHANNELS`, в котором поиск мастера для привязки осуществляется по всем каналам. Благодаря этому, при отсутствии информации о подключении во флэш-памяти, ведомое устройство подключится к первому же мастеру, обладающему заданным типом устройства.

4.3. Сравнительный анализ модифицированного протокола с базовым

После разработки было произведено комплексное тестирование протокола. Для начала, он был проверен на фирменных демо-примерах и разработанных ранее проектах. При испытаниях файлы протокола `WirelessUSB` были заменены файлами `WirelessUSB_mk`. Компиляция проектов прошла успешно, устройства работали корректно. Это означает, что модифицированный протокол полностью совместим с базовым на функциональном уровне.

Вторым этапом проверки было измерение скоростных параметров протокола. Для этого был использован проект, в котором реализуется беспроводной мост для последовательного интерфейса RS-232 (COM). При помощи данного моста были связаны два порта одного компьютера, на котором была установлена программа `COM Port Toolkit`, позволяющая проверить предельную скорость обмена по интерфейсу без потери данных [7].

При испытаниях оказалось, что предельная скорость передачи для обеих версий протокола при настройках по-умолчанию колеблется в пределах 40-45 килобит в секунду. Так как скорость передачи по радиоканалу у приёмопередатчиков WirelessUSB LS равняется 62.5 килобитам в секунду, то издержки при использовании протокола составляют около тридцати процентов. Несмотря на это, использование протокола по причинам, приведённым в разделе 2.3, является необходимым. Поэтому, можно констатировать то, что разработанная модификация не хуже базовой по скоростным характеристикам.

При сравнении размеров подпрограмм протоколов, скомпилированных ImageCraft 1.47 в среде проектирования PSoC Designer 4.3, оказалось, что модифицированная версия протокола занимает 3045 байт памяти программ, в то время как базовая- 3612. В основном, различие вызвано переносом статических массивов в пользовательскую память и исключением части дополнительных возможностей протокола. С другой стороны, модифицированная версия обеспечивает устойчивую связь и может быть использована при недостатке памяти при реализации сложных проектов.

Для более компиляторов Imagecraft 1.69 и HI-TECH размер программы модифицированного протокола составил 2940 и 2230 байт соответственно. Предельная скорость передачи при этом не изменилась. По результатам видно, что использование более новых версий сред проектирования и компиляторов позволяет существенно повысить эффективность использования ресурсов при программировании на языке высокого уровня. Это является дополнительным подтверждением актуальности проведённой разработки.

В целом, модифицированный протокол соответствует базовому протоколу в обычном режиме. Главным недостатком его является то, что

радио-передатчик не переводится в режим сна после завершения отправки пакета. Это должно сильно повысить потребляемую микросхемой мощность. Кроме того, не был реализован режим KISS-BIND, который упрощает привязку устройств, когда в радиусе передачи присутствует несколько ведущих устройств требуемого типа. Однако, реализация указанных выше возможностей базового протокола WirelessUSB вполне возможна в следующих версиях протокола.

Введённый режим BIND_ON_ALL_CHANNELS применим только для разработки и отладки устройств, когда требуется многократное перепрограммирование. В остальных случаях, использование данного режима нежелательно, так как запросы на соединение отправляются на рабочих каналах, и от ведущих устройств требуются затраты времени на обработку, что снижает пропускную способность уже работающих каналов связи.

Заключение

В проведённой работе было произведено исследование возможностей реализации беспроводных систем сбора и обработки первичной информации на основе технологии WirelessUSB компании Cypress. В работе было определено место данной технологии среди существующих на рынке, выделены её основные преимущества при реализации низкоскоростных устройств.

В работе были рассмотрены принципы организации беспроводной связи в технологии WirelessUSB, существующие на сегодняшний день решения и средства разработки для них. Среди всех существующих семейств наибольший интерес вызвали микросхемы PРоC, где цифровая часть приёмопередатчика интегрирована в управляющий контроллер. До сих пор такое решение остаётся уникальным. Однако, с ростом технологических возможностей и потребностей в миниатюризации устройств следует ожидать развития данной концепции. Дополнительное преимущество PРоC даёт наличие конфигурируемых модулей, что повышает гибкость систем и позволяет добиться высокой эффективности использования кристалла. Исходя из вышесказанного, выбранная тематика работы актуальна и возможно дальнейшее развитие.

Вплоть до настоящего времени не существует никаких методических материалов по данной технологии за исключением простейших демонстрационных примеров работы беспроводной связи. Поэтому, одной из задач было составление обобщённого описания технологий WirelessUSB, а также исследование возможностей применения микросхем PРоC в задачах сбора и первичной обработки информации. Для этого потребовалось изучить структуру контроллеров PРоC, возможностей реализации периферийных устройств на базе конфигурируемых модулей и т.п.

Все эти задачи были успешно выполнены, и по результатам работы был разработан ряд относительно простых устройств сбора и обработки информации. Также было разработано описание данных проектов в виде лабораторных работ, которые также могут быть использованы в методических целях.

После проведения предварительного исследования были обнаружены проблемы совместимости протокола WirelessUSB с последними средами проектирования, которые необходимы для проведения отладки пользовательских программ. Поэтому, была поставлена задача разработки модификации протокола, совместимого с современной средой PSoC Designer 5.0. Данная задача также была успешно выполнена.

В разработанной модификации была реализована лишь базовая функциональность протокола WirelessUSB. В новой версии не поддерживаются такие возможности, как автоматический перевод передатчика в спящий режим, привязка по методу KISS-BIND и контроль относительной интенсивности сигнала. С другой стороны, был введён новый режим работы протокола, при котором привязка устройств может производиться на всех каналах, что удобно при разработке и отладке устройств. В рамках разработанной модификации возможна реализация всех возможностей базового протокола, что может быть предметом дальнейших исследований.

Список использованных источников

1. Баскаков С., Оганов В. Передача данных на базе технологии WirelessUSB // Электронные компоненты.- 2005.- № 3.- С. 3-5
2. Ворона В.А. Радиопередающие устройства. Основы теории и расчёта.- М.: Горячая линия – Телеком, 2007
3. Грушвицкий Р., Мурсаев А., Угрюмов Е. Проектирование систем на микросхемах программируемой логики.- СПб.: БХВ-Петербург, 2002
4. Кривченко И. Системы на кристалле // Компоненты и технологии.- 2001.- № 6.- С. 15-12
5. Романченко В. Wireless USB // Коммуникационное оборудование.- 2005.- № 1.- С. 10-11
6. Шахнович И.В. Современные технологии беспроводной связи.- М.: ТЕХНОСФЕРА, 2006
7. COM Port Toolkit //www.compt.ru/
8. PSoC Mixed Signal Array. Technical Reference Manual
//<http://www.cypress.com/?rID=3208>
9. Maximizing Range in WirelessUSB Systems
//www.cypress.com/app_notes/AN4008.pdf
10. radio_GetMid function bug
//www.cypress.com/?id=4&rID=31032
11. Wireless Binding Methodologies
//www.cypress.com/app_notes/AN6066.pdf
12. WirelessUSB LS. Theory of Operation
//www.cypress.com/app_notes/AN033B.pdf
13. WirelessUSB problems with PSD 4.4
//www.cypress.com/?id=4&rID=30153

Приложение 1. Исследование принципов реализации радиоканала на базе PРоС кристалла CYWUSB6953 и протокола Wireless USB

Целью работы является изучение основных принципов организации радиоканала на основе PРоС кристалла CYWUSB6953 и стандартного протокола WirelessUSB 1.1.

Общие положения

В лабораторной работе рассматриваются простейшие примеры организации беспроводной связи на базе технологии WirelessUSB с использованием средств, предоставленных средой PSoC Designer.

Задачей работы является осуществление беспроводной связи двух устройств как с использованием протокола WirelessUSB, так и без него. При этом, передача инициируется извне нажатием кнопки и индицируется на приёмном устройстве переключением светодиодов (рис. П1.1).

В данной работе используются микросхемы CYWUSB6953, содержащие приемо-передатчик типа LS. В качестве среды проектирования используется система PSoC Designer, а в качестве отладочных модулей – две платы CY3653. Структура разрабатываемой системы приведена ниже:

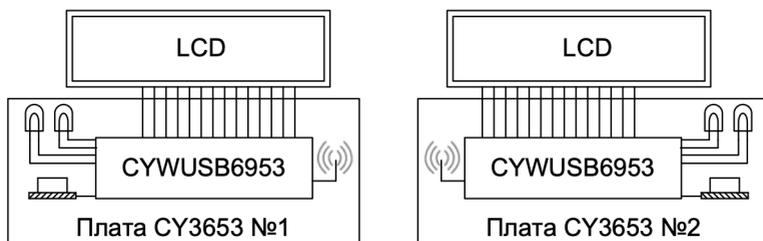


Рис. 1. Схема лабораторного стенда

Микросхемы CYWUSB6953 принадлежат к подсемейству PProC (Programmable Radio-on-Chip) из семейства программируемых систем на кристалле PSoC компании Cypress. Цифровая подсистема содержит один ряд (по два блока типов DBB и DCB), аналоговая система является упрощённой. Из системных ресурсов в кристалле присутствуют приёмопередатчик WirelessUSB LS, контроллер I2C, сторожевой таймер и блок контроля питания.

В CYWUSB6953 цифровая часть радиопередатчика интегрирована в кристалл и не требует использования дополнительных системных ресурсов. Однако, протокол WirelessUSB для своей работы требует наличия 8-разрядного таймера.

Микросхема CYWUSB6953 поддерживает два режима питания: 3.3 и 2.7 Вольт, при этом частота тактирования (SysClk) может быть 6, 12 и 24 МГц. Кроме того, в кристалл встроены три программируемых делителя частоты (VC1, VC2 и VC3), выходы которых могут быть использованы для тактирования пользовательских модулей. Также в устройстве присутствует умножитель частоты (SysClk*2), который позволяет получить большую частоту, чем может обеспечить встроенный генератор. Настройки микросхемы CYWUSB6953 могут быть заданы в среде PSoC Designer. Доступные настройки и их описание приведены в табл. 1. Настройки по-умолчанию выделены в таблице жирным шрифтом.

Настройки микросхемы CYWUSB6953. Таблица 1

Параметр	Возможные значения	Описание
Power Setting/ SysClk freq	2.7В /12 МГц 2.7В /6 МГц 3.3В /6 МГц 3.3В /24 МГц	Задание питающего напряжения платы и соответствующей ему тактовой частоты генератора.
CPU_Clk	SysClk/1...256 (2)	Частота тактирования процессорного ядра.
Sleep_Timer	1, 8, 64, 512 Гц	Частота перезагрузки таймеров спящем режиме.
VC1=SysClk/N	N=1...16 (1)	Настройка первого делителя тактовой

		частоты.
VC2=VC1/N	N=1..16 (1)	Настройка второго делителя тактовой частоты. Источник- VC1.
VC3 Source	SysClk, VC1, VC2	Выбор источника синхронизации для третьего делителя тактовой частоты.
VC3 Divider	1..256 (1)	Настройка третьего делителя частоты.
SysClk Source	Int, Ext	Настройка источника синхронизации. может быть взят встроенный генератор или внешний (сигнал принимается с P1.4)
SysClk*2 Disable	Yes, No	Разрешение или запрещение работы встроенного умножителя частоты.
Trip Voltage LVD (SMP)	2.45B (2.55B) 2.92B (3.02B) 3.02B (3.10B) 3.13B (3.25B)	Настройки монитора питания. LVD- уровень, при котором срабатывает детектор питающего напряжения, и схема переводится в состояние перезагрузки. Она остаётся в нём до тех пор, пока напряжение питания не превысит уровень SMP.
LVD ThrottleBack	Enable, Disable	Разрешает автоматический переход на режим питания 2.7В (6 МГц) при низком уровне питающего напряжения.
Watchdog Enable	Enable, Disable	Разрешение работы стерегущего таймера.

Основной реализации беспроводной связи в PRoC является пользовательский модуль WirelessUSB. Он обеспечивает программный интерфейс работы со встроенным в кристалл радиоустройством. Приёмопередатчики WirelessUSB конфигурируются при проектировании устройства и не могут быть изменены при помощи функций, генерируемых при сборке проекта. Настройки приёмопередатчиков семейства LS и их описание приведены в приложении (табл. 2). В большинстве случаев могут использоваться настройки по-умолчанию.

Для управления приёмопередатчиком в PRoC используется интерфейс SPI. Однако, в CYWUSB6953 отсутствует контроллер SPI, и производится программное управление интерфейсом. В среде PSoC Designer модуль WirelessUSB генерирует функции управления радиоустройством, описание которых приведено в приложении к работе (табл. 3).

Протокол WirelessUSB существенно упрощает работу с беспроводным каналом связи и повышает его надёжность. Он обеспечивает выполнение алгоритмов выбора рабочего канала, привязки устройств и обмена данными между ними. Именно он производит автоматический контроль структуры сети (наличие связи) и коррекцию ошибок в принятых пакетах.

Сам протокол достаточно сложен, но, с точки зрения пользователя, передача данных между устройствами может быть реализована всего лишь несколькими командами:

protocol_init() – инициализация протокола для любой роли устройства

slave_send_packet () - отправка сообщения Slave-устройством

slave_bind () – запуск алгоритма привязки у Slave-устройства
(вызывается после инициализации устройства)

master_data_mode () – обслуживание канала Master-устройством

Более подробно функции протокола описаны в разделе 2.4. При использовании протокола WirelessUSB следует помнить следующие моменты:

1) Требуется разработать, как минимум, два устройства (ведущее и ведомое), так как этого требует методика организации связи в протоколе.

2) Обмен информацией инициируется Slave-устройством, поэтому для реализации двунаправленного обмена ведомое устройство должно периодически производить отсылку, например, пустых пакетов, чтобы ведущее устройство могло ответить на них своим информационным пакетом.

3) Обработка принимаемой информации в Master-устройстве производится функцией `master_data_mode()`. Поэтому, эта функция

должна вызываться регулярно, иначе ведущее устройство может не успеть ответить за время ожидания Slave-устройства.

Программа работы

1.1. Создайте новый проект `Radio_exchange` в директории `C:/cypress_lab/student/фамилия/lab7/`. За основу возьмите кристалл `CYWUSB6953-48LFXC`. Для генерации основной функции проекта (`main`) выберите язык `C`.

1.2. Добавьте в устройство модуль радиопередатчика `WirelessUSB`. Для этого перейдите в редактор устройства (`Device Editor`) и выберите режим выбора пользовательских модулей (`User Module Selection View`). Далее выберите модуль `WirelessUSB LS` и добавьте в проект (`Select`).

1.3. Выберите два цифровых вывода микросхемы (например, `P1.1` и `P1.2`) и переименуйте их в `LED1` и `LED2` и перенастройте их на выход, установив мощный (`Strong`) уровень сигналов.

1.4. Для ввода информации с кнопки выберите вывод `P1.0` микросхемы и переименуйте его в `SW`. В настройках режима переведите порт в состояние `High Z`.

1.5. Добавьте в проект модуль `LCD` и подключите его ко второму порту. Переименуйте добавленный модуль в `LCD`.

1.6. Настройка системных ресурсов платы закончена. Сгенерируйте проект (`Config\Generate application`) и перейдите в редактор программы (`Application Editor`).

1.7. Добавьте в программу инициализацию приёмопередатчика `WirelessUSB`. Для этого следует последовательно в функции `main` (файл “`main.c`”) до основного цикла программы вызвать следующие операции:

```
radio_Start()           // Запуск передатчика;
radio_SetChannel(uchar) // Установка номера канала (1-78);
radio_SetPnCode(uchar)  // Установка кода псевдошума (0-7).
```

1.8. Создайте функции опроса нажатия кнопки и переключения светодиодов. Примеры их реализации приведены ниже:

```

UINT8 SW_up;
UINT8 SW_pressed(void)           // Функция опроса нажатия кнопки
{
    UINT8 port_shadow;
    port_shadow = SW_Data_ADDR;
    if(port_shadow &= SW_MASK) {
        if (button_up == FALSE)    return 0;
        else {
            button_up = FALSE;
            return 1;
        }
    }
    else {
        button_up = TRUE;
        return 0;
    }
}

UINT8 flag_LED=0;
//Функция переключения светодиодов
void switch_LED(void) {
    if (flag_LED == 1){
        LED2_Data_ADDR &= ~LED4_MASK; //Выключение LED 2
        LED1_Data_ADDR |= LED1_MASK;  //Включение LED 1
        flag_LED = 0;
    }
    else{
        LED1_Data_ADDR &= ~LED1_MASK; //Выключение LED 1
        LED2_Data_ADDR |= LED4_MASK;  //Включение LED 2
        flag_LED = 1;
    }
}

```

1.9. Добавьте в программу вывод на ЖКИ информации о выбранном номере канала и коде псевдошума. Пример вывода численной информации на экран приведён ниже:

```
void LCD_write (UINT8 Y, UINT8 X, number)
{
    UINT8 temp_string[10];
    // Преобразование числа number в строку для десятичной системы счисления
    itoa((char *)&temp_string, number, 10);
    // Установка начальной позиции вывода на дисплее
    LCD_Position(Y,X);
    // Вывод информации на дисплей
    LCD_PrString((char *)&temp_string);
}
```

1.10. Добавьте в программу периодический опрос нажатия кнопки и проверку принятых сообщений. Если кнопка нажата, то по радиоканалу должна передаваться некоторая информация (например, байт 0x55), а при приёме новых данных должны переключаться светодиоды. Для отправки сообщения используйте функцию `radio_SendData()`, а для выявления приёма сообщения - `radio_bReadData()`. Пример последовательности опроса кнопки, передачи и приёма сообщений приведён ниже:

```
// отправка сообщения при нажатии кнопки
if (SW_pressed())
{
    tx_count++;
    tx_buf[0] = 0x55;
    radio_SendData(1, (UINT8 *)&tx_buf);
}

// проверка приёма сообщения и реакция на него
if ( radio_bReadData (1,(UINT8*)&rx_buf, (UINT8 *)&valid_buf, RX_TIMEOUT))
```

```

{
    if (rx_buf[0] == 0x55)
    {
        rx_count++;
        switch_LED();
    }
}

```

1.11. Запрограммируйте микросхему и убедитесь в правильности работы устройств. При каждом нажатии кнопки на одной плате, на второй светодиоды должны переключиться. Данный проект не использует протокол WirelessUSB, поэтому в радиусе 30 м. от передатчика возможны ложные срабатывания иных абонентов в диапазоне ISM.

2.1. Изучите демо-пример C_Example_Protocol_Tutorial расположенный в папке C:/cypress_lab/help/lab7/. В нём Master- и Slave-устройства переключают светодиоды при приёме сообщения. Сама передача запускается нажатием кнопки на отладочной плате. В данной части работы требуется последовательно реализовать проект с аналогичной функциональностью.

2.2. Создайте новый проект WirelessUSB_Pattern в редакторе PSoC Designer, взяв за основу кристалл CYWUSB6953-48LFXC. Для генерации основной функции проекта (main) выберите язык C.

2.3. Добавьте в проект восьмиразрядный таймер и переименуйте его в Tick. Задайте следующие параметры таймера:

User Module Parameters	Value
Clock	CPU_32_KHz
Capture	Low
TerminalCountOut	None
CompareOut	None
Period	32
CompareValue	0
CompareType	Less Than
InterruptType	Terminal Count
ClockSync	Sync to SysClk
TC_PulseWidth	Full Clock
InterruptAPI	Enable
IniDispatchMode	ActiveStatus
InvertCapture	Normal

Рис. 2. Настройки таймера

2.4. Добавьте в проект модуль LCD и подключите его ко второму порту.

2.5. Добавьте в проект модуль WirelessUSB и переименуйте его в radio (такое имя использует протокол). Проверьте настройки модуля:

User Module Parameters	Value
DataRate(Kbps)	64
NonBlockingCalls	Enable
ChipErrorThreshold	3
DefaultTxPowerLevel	7
EofBits	3
CrystalStartUpTime(msec)	5
CrystalAdjust	0

Рис. 3. Настройки модуля WirelessUSB

2.6. Добавьте в проект (Project->Add to Project->Files) файлы protocol.c, protocol.h, timer.c, timer.h, cypdef.h, Is_config.h, flash.h и flash.asm, находящиеся в папке C:/cypress_lab/help/lab7/source. В этих файлах реализован протокол WirelessUSB и их изменение может привести к некорректной работе всех устройств в лаборатории.

После выполнения этого пункта вы получите заготовку, требуемую для любого проекта с использованием протокола WirelessUSB 1.1. Рекомендуется её сохранить и использовать в дальнейших работах как шаблон.

2.7. Используя полученный шаблон, создайте проекты для ведущего (WirelessUSB_Master) и ведомого (WirelessUSB_Slave) устройств. Определение роли устройства, как и настройка основных параметров протокола, производится в файле Is_config.h. Описание параметров протокола приведены в приложении к работе (табл. 1.5).

2.8. Для Slave-устройства реализуйте периодическую отправку пустых пакетов (slave_send_packet(0)) для обеспечения передачи данных от мастерского устройства.

2.9. Реализуйте поведение устройств, описанное в общих положениях к работе. Для этого введите функцию опроса нажатия кнопки. Если кнопка нажата, то по радиоканалу должно передаваться информационное сообщение.

2.10. Для обоих устройств выведите на LCD информацию о номере канала и коде псевдо-шума, а также о количестве отправленных и принятых пакетов.

2.11. Запрограммируйте устройства и добейтесь их правильной работы в обоих направлениях (от ведущего к ведомому и наоборот).

Приложение к лабораторной работе

Настройки модуля приёмопередатчика WirelessUSB. Таблица 2

Параметр	Принимаемые значения	Назначение
DataRate	16, 32, 64	Скорость передачи канала, кбит/с
NonBlockingCalls	Disable , Enable	Разрешает или запрещает прерывание приёма информации. Если устройство работает только на отсылку информации, то данный параметр позволяет исключить временные затраты на обработку принятой информации.
ChipErrorThreshold	1..15 (3)	Максимальное кодовое расстояние между принятым кодом и эталонным кодом Голда, при котором бит считается правильно принятым.
DefaultTxPowerLevel	0..7 (7)	Уровень интенсивности передатчика, присваиваемый по умолчанию.
EofBits	1..7 (3)	Количество ошибочных бит, принятых подряд, после которого возникает прерывание EOF.
CrystalStartupTime(msec)	0..20 (5)	Время в миллисекундах, требуемое кварцевым генератором для стабилизации частоты. Это время определяет задержку в команде запуска передатчика.
CrystalAdjust	0..31 (0)	Данный параметр задаётся для подстройки нагрузочной ёмкости, подключённой к тактовому генератору.

Функции управления радио-передатчиком WirelessUSB. Таблица 3
(начало)

Функция	Описание
void radio_Start (void)	Функция запускает (или перезапускает) радио, восстанавливая параметры по-умолчанию. После этого программой выжидается время CrystalStartupTime, требуемое для стабилизации кристалла импульсного генератора.
void radio_SetChannel (Byte Channel)	Устанавливает частоту работы передатчика, соответствующую номеру канала (от 0 до 78).
void radio_SetPnCode (const BYTE* pbPnCode)	Функция загружает в радио-передатчик 64-разрядный код псевдо-шума, находящийся по адресу pbPnCode.
void radio_SendData (BYTE bLength, BYTE* pbData)	Функция загружает в буфер радио-передатчика пакет данных длиной bLength, находящийся по адресу pbData.
BYTE radio_bReadData (BYTE bLength, BYTE *pbData, BYTE *pbValid, WORD wTimeout)	<p>Функция считывает принятые данные в буфер, пока:</p> <ul style="list-style-type: none"> - передатчиком не будет сгенерировано исключение EOF, т.е. не будут считаны все принятые данные. - не будет достигнута максимальная длина сообщения bLength. - не пройдет время wTimeout, данное на операцию обмена информацией. (wTimeout задается в интервалах времени по 15 мкс). <p>Информация записывается по адресам, начиная с pbData. При этом, возможен контроль принимаемого сообщения при помощи маски pbValid. Функция возвращает длину принятого сообщения в байтах.</p>
BYTE radio_StartNonBlockingRead (BYTE bLength, BYTE *pbData, BYTE *pbValid)	<p>Функция активирует режим непрерывного приёма информации. Функция активирует прерывание от радио-приёмника. Считывание данных с интерфейса производится до тех пор, пока не будут получены bLength байт информации.</p> <p>Сначала, подобно функции radio_bReadData() считываются данные из буфера и возвращается длина считанного сообщения, дальнейшее считывание и запись в указанный буфер производится через систему прерываний.</p>
void radio_StopNonBlockingRead (void)	Функция запрещает прерывания, генерируемые приёмником, отключая тем самым режим непрерывного считывания информации.
BYTE radio_bCheckRxInt (void)	Проверка достижения конца пакета (EOF) в режиме непрерывного приёма информации. Возвращает длину (в байтах) принятой информации.
BYTE radio_bCheckRxInProgress (void)	Проверка активности приёмника. Возвращает 1, если идёт приём информации, и 0, когда приём не ведётся.
void radio_Sleep (void)	Функция переводит радио-передатчик в спящий режим для уменьшения потребляемой энергии.

Функции управления радио-передатчиком WirelessUSB. Таблица 3
(продолжение)

Функция	Описание
void radio_Wakeup (void)	Вывод передатчика из спящего режима с сохранением его настроек. При этом производится задержка для стабилизации тактового генератора.
void radio_GetMid (BYTE *pbMid)	Функция считывает четырёхбайтовый уникальный номер передатчика (MID- Manufacturing ID) и записывает его по указанному адресу.
BYTE radio_bGetRssi (void)	Функция возвращает показатель силы принимаемого сигнала относительно уровня шума. Возвращаемое значение находится в диапазоне от 0 до 31.
void radio_SetTxPowerLevel (BYTE bPowerLevel)	Установление заданного уровня интенсивности сигнала передатчика. Параметр bPowerLevel должен находиться в диапазоне от 0 до 7.
BYTE radio_bGetTxPowerLevel (void)	Функция возвращает текущий уровень интенсивности передатчика.

Варианты заданий. Таблица 4

№ варианта	Номер канала	Код псевдошума
1	5	0
2	10	1
3	15	2
4	20	3
5	25	4
6	30	5
7	35	6
8	40	7

Приложение 2. Исследование средств предварительной обработки дискретной и аналоговой информации для передачи по радиоканалу на основе кристалла CYWUSB6953

Целью работы является исследование методов обработки цифровых и аналоговых сигналов в микросхемах CYWUSB6953 и передачи их по беспроводному каналу связи.

Общие положения

В лабораторной работе рассматриваются достаточно распространённые примеры использования беспроводной связи, такие как реализация пульта дискретного управления на примере клавиатуры и реализация аналогового управления на примере двухосного джойстика.

Реализуемые системы состоят из двух элементов: устройства сбора информации и её отображения, между которыми осуществляется беспроводная связь по протоколу WirelessUSB. В рассматриваемых проектах, информация передаётся на компьютер для последующей обработки. Общий вид системы можно представить следующим образом:



Рис. 1. Схема разрабатываемой системы

Устройство сбора информации производит снятие данных, их первичную обработку, после чего пересылает информацию на устройство-приёмник, которое передаёт данные на инструментальный компьютер, служащий средством визуализации полученных данных.

Подсистеме сбора информации в протоколе целесообразно присвоить роль Slave, так как именно оно является источником информации (и активирует передачу). Такая организация позволяет увеличить количество устройств сбора данных, подключая их к одному ведущему устройству.

Связь с инструментальным компьютером осуществляется Master-устройством через стандартный коммуникационный интерфейс (COM). В качестве внешнего интерфейса в PSoC Master-устройстве используется протокол UART, который преобразуется в RS-232 с помощью интегральной схемы MAX3221, установленной на плате отладочного модуля CYWUSB3653.

Отображение принимаемых данных на инструментальной ПК осуществляет программа ComViewer. В ней предусмотрены режимы “клавиатура” и “джойстик”, которые позволяют контролировать правильность считывания и передачи информации. Для корректной работы программы должен соблюдаться формат сообщения, который приведён в табл. 1.

Формат сообщений, обрабатываемый программой. Таблица 1

Смещение	Передаваемая информация
0	Префикс. Служит для синхронизации при непрерывном потоке информации по RS-232
1	Идентификатор устройства.
2 - N+1	N байтов информации

В первой части работы источником данных является 16-кнопочная клавиатура, электрическая схема которой приведена на рис. 2. Требуется организовать опрос нажатия клавиш, формирование информационного пакета и его передачу.

Рассматриваемая в работе клавиатура состоит из шестнадцати кнопок, расположенных в узлах матрицы 4x4. При замыкании кнопки образуется связь между проводниками строки (ROW) и колонки (COLUMN), на пересечении которых расположена кнопка. Строки и колонки через подтягивающие резисторы R₁..R₈ подключены к линии V_PULL.

Все девять линий клавиатуры выведены на монтажное поле отладочной платы.

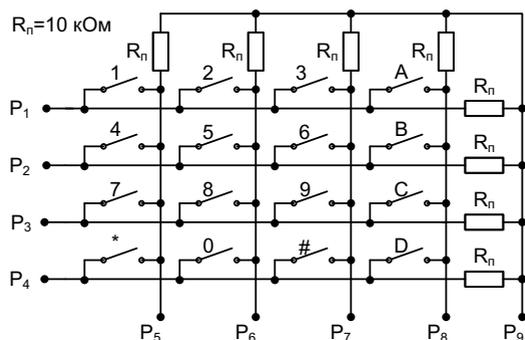


Рис. 2. Схема клавиатуры

В программе ComViewer для работы с клавиатурой следует использовать режим “Keyboard”. При этом на экране отображается одна или две панели размером 4x4, на которых отображается нажатие клавиш на устройствах с заданными идентификаторами. Кроме того, на экране присутствует дополнительная панель 18x7, в котором находится курсор, управляемый нажатием клавиш на клавиатуре. Реализуемый закон движения описан в табл. 2.

Реакция курсора в панели на нажатие клавиш клавиатуры. Таблица 2

Клавиша	Направление движения курсора
2	Вверх
4	Влево
6	Вправо
8	Вниз

При опросе клавиатуры в каждом пакете передаётся информация о состоянии клавиш. При этом информационное поле содержит два байта, каждый бит в которых отвечает за одну кнопку. Соответствие битов клавишам приведено в табл. 3.

Соответствие битов информационного поля клавишам. Таблица 3

Смещение	2							3								
Номер бита	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Клавиша	1	2	3	A	4	5	6	B	5	6	7	C	*	0	#	D

Во второй части работы рассматривается ввод аналоговой информации в устройство. В качестве источника сигналов используется двухосный джойстик FJM10K1, выполненный при помощи двух реостатов номиналом 100 кОм. Джойстик установлен на монтажном поле отладочной платы и подключён к уровням Vcc и Gnd, поэтому диапазон выходного сигнала соответствует полному диапазону АЦП. Это позволяет подключить выходы джойстика напрямую к АЦП не используя внешних преобразователей.

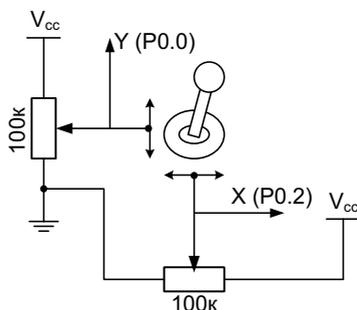


Рис. 3 Схема джойстика

Упрощённая аналоговая система, используемая в кристаллах CY8C21x и CYWUSB6953, сильно отличается от систем, рассмотренных ранее. В ней возможна реализация компаратора и АЦП последовательного счёта с

разрядностью, не превышающей 10. В лабораторной работе используется восьмиразрядный преобразователь.

При реализации АЦП разрядность задаётся соотношением периода тактирования счётчика и длительностью единичного импульса ШИМ. Поэтому, для получения восьмиразрядного АЦП с полным диапазоном измерения требуется соблюсти следующее соотношение:

$$256 * T_{\text{COUNTER}} = T_{\text{VC3}} * (\text{PWM High})$$

В среде PSoC Designer при построении АЦП последовательного счёта используются модули ADC8 и ADC10. В первом разрядность фиксирована и равна восьми, во втором может изменяться от одного до десяти. Возможные настройки АЦП приведены в табл. 4.

Настройки АЦП последовательного счёта. Таблица 4

Параметр	Возможные значения	Назначение
Data Clock	SysClk*2, VC1, VC2, VC3, CPU_32_kHz, Row_0_Output_x, Row_0_Input_x, Row_0_Broadcast	Определяет источник тактирования счётчика, накапливающего результат измерения.
ADC Input	PortMUX, ACE0x, AnalogMUXBus, VBG	Определяет источник аналогового сигнала. Кроме аналоговой шины и мультиплексора, АЦП может быть подключён выходу компаратора из противоположной колонки и уровню VBG (последнее подключение используется для калибровки АЦП).
PWM High	1, 2, 4, 8, 16 периодов VC3	Время удержания высокого уровня на выходе генератора шим в периодах VC3.
PWM Low	0, 1, 2, 3 периода VC3	Время удержания низкого уровня на выходе генератора шим в периодах VC3. в этот промежуток преобразование не ведётся, и результат может быть считан со счётчика.

В программном обеспечении АЦП предусмотрены два режима работы: полный диапазон входных напряжений (от Vss до Vdd) и неполный (от Vss до Vdd-1). Диапазон работы определяется в функции инициализации АЦП ADC_Start().

Также существует функция калибровки АЦП ADC_bCal(). Например, в восьмиразрядном АЦП при напряжении питания 2.7В уровень VBG (1.3В) должен соответствовать коду 7Ah. Тогда после запуска АЦП следует вызвать следующую функцию:

```
ADC_bCal(0x7A, ADC_CAL_VBG);
```

Полный список команд управления АЦП и их описание приведены в табл. 5.

Функции управления модулем ADC. Таблица 5

Функция	Описание
ADC_Start()	Начальная инициализация АЦП и включение питания блоков АЦП.
ADC_Stop()	Выключение питания блоков АЦП.
ADC_bCal()	Калибровка АЦП. Функция настраивает ёмкость конденсатора так, чтобы напряжение на эталонном источнике после преобразования соответствовало заданному коду.
ADC_StartADC()	Запуск непрерывного преобразования в АЦП. По окончании обработки генерируется прерывание, по которому производится обработка данных с АЦП.
ADC_StopADC()	Запрещает преобразование, в том числе и текущее.
ADC_flsDataAvailable()	Проверяет наличие новой информации. Возвращает TRUE, если появились новые данные. Иначе- FALSE.
ADC_ClearFlag()	Сброс флага приёма новых данных.
ADC_bGetData()	Возвращает результат последнего преобразования.
ADC_bGetDataClearFlag()	Считывает результат последнего преобразования и сбрасывает флаг приёма новой информации.

После каждого преобразования в обработчике прерывания от АЦП результат буферизуется, и выставляется флаг наличия новых данных. Далее, флаг может быть проверен в теле программы функцией ADC_flsDataAvaible(), после чего из буфера можно извлечь новые данные.

В стандартной реализации модуля АЦП последовательного счёта выполняется непрерывное преобразование входных сигналов, прерывание генерируется после окончания каждого измерения. Для реализации однократного преобразования можно вызвать функцию `ADC_StopADC()` в обработчике прерывания.

При рассмотрении аналогового ввода в работе ставится задача снятия координат двухосного джойстика. Для её выполнения требуется реализация двух каналов обработки аналоговой информации. Самый простой способ – динамическая переконфигурация линий аналогового ввода и переключение канала после каждого преобразования.

В программе `ComViewer` для работы с джойстиком следует использовать режим “Joystick”. В нём на экране отображается координатная сетка `0..255`, на которой курсором указывается текущее местоположение джойстика. В этом режиме длина информационного поля равна двум байтам. Каждый байт должен содержать результаты аналого-цифрового преобразования для одной из осей (первый байт- ось Y, второй- ось X).

Программа работы

1.1. Создайте в директории `C:/cypress_lab/student/фамилия/lab8/` новый проект `keyboard_slave`, взяв за основу шаблон `WirelessUSB_Pattern`, созданный в предыдущей лабораторной работе. Первая система должна производить опрос клавиатуры и передавать информацию на ведущее устройство. Поэтому, установите роль устройства `Slave` в файле `Is_config.h`.

1.2. В Редакторе Устройства (`Device Editor`) присвойте свободным выводам имена и настройте направление передачи сигналов (Y1..4- на вход, X1..4- на выход). Пример подключения приведён ниже:

Name	Port	Select	Drive	Interrupt
Port_0_0	P0[0]	StdCPU	Strong	DisableInt
Port_0_2	P0[2]	StdCPU	Strong	DisableInt
Port_0_6	P0[6]	StdCPU	High Z	RisingEdge
Y1	P1[0]	StdCPU	Strong	DisableInt
Y2	P1[1]	StdCPU	Strong	DisableInt
Y3	P1[2]	StdCPU	Strong	DisableInt
X1	P1[4]	StdCPU	High Z	DisableInt
X2	P1[5]	StdCPU	High Z	DisableInt
X3	P1[6]	StdCPU	High Z	DisableInt
X4	P1[7]	StdCPU	High Z	DisableInt
LCDD4	P2[0]	StdCPU	Strong	DisableInt
LCDD5	P2[1]	StdCPU	Strong	DisableInt
LCDD6	P2[2]	StdCPU	Strong	DisableInt
LCDD7	P2[3]	StdCPU	Strong	DisableInt
LCDE	P2[4]	StdCPU	Strong	DisableInt
LCDR5	P2[5]	StdCPU	Strong	DisableInt
LCDRw	P2[6]	StdCPU	Strong	DisableInt
Y4	P2[7]	StdCPU	Strong	DisableInt

Рис. 4. Пример настроек межсоединений

1.3. Добавьте в программу функцию, реализующую один из методов опроса клавиатуры. Пример функции опроса “бегущим нулём” для одной строки приведен ниже:

```

UINT8 B1,B2;           // первый и второй байты карты нажатий
void Check_keys_Y1(void)
{
    UINT8 port_shadow=0;
    UINT8 buf=0;

    Y1_Data_ADDR |= Y1_MASK; // установка всех вводов клавиатуры в “1”
    Y1_Data_ADDR &= ~Y1_MASK; // установка Y1 в 0
    timer_delay_msec(1);

    port_shadow = X1_Data_ADDR; // Считывание информации из первой строки
    buf=255-port_shadow;       // Инверсия данных

    (т.к. 0 обозначает нажатие, а 1-нет)
    if((buf & X1_MASK))B1+=8; // Корректировка карты нажатий в соответствии

```

```

if((buf & X2_MASK))B1+=128; // с полученными данными
if((buf & X3_MASK))B2+=8; //
if((buf & X4_MASK))B2+=128; //
Y1_Data_ADDR |= Y1_MASK; // Установка Y0 в 1

```

1.4. В программе обеспечьте вывод информации о нажатых клавишах на дисплей, формирование и передачу данных на Master-устройство после каждого опроса. Оцените интервал времени между опросами.

1.5. Запрограммируйте устройство и проверьте правильность его работы. Объясните, почему возникают большие временные задержки между опросами.

2.1. Создайте второй проект COM_master на основе шаблона WirelessUSB_Pattern. Присвойте устройству роль Master. В файле настроек протокола Is_config.h задайте начальные номер канала и код псевдошума соответственно табл. П1.4. Для этого следует изменить параметры DEFAULT_CHANNEL и DEFAULT_PN_CODE из файла настроек.

2.2. Добавьте в проект модуль TX8, который обеспечивает однонаправленную передачу по интерфейсу RS-232. Задайте следующие настройки передатчика:

TX8	
User Module Parameters	Value
Clock	VC3
Output	Row_0_Output_3
TX Interrupt Mode	TXRegEmpty
ClockSync	Unsynchronized
Data Clock Out	None
InterruptAPI	Enable
IntDispatchMode	ActiveStatus

Рис. 5. Настройки модуля передатчика

2.3. Подключите линию Row_0_Output_3 к выводу Port_2_7 микросхемы. Для этого, сначала подключите вывод к шине глобальных выводов, а потом соедините используемые линии локальной и глобальной шин как показано ниже:


```

UINT8 i;
if(status & DATA_RECEIVED)
{
    app_unload_master_packet();
    ID=rx_packet.ack_data.app_packet[1];
    B1=rx_packet.ack_data.app_packet[2];
    B2=rx_packet.ack_data.app_packet[3];
    // Передача 4-х байт из буфера.
// Контроль конца передачи производится функцией автоматически
    for (i=0; i<4;i++) TX8_PutChar(rx_packet.ack_data.app_packet[i]);
}

```

2.6. В программе выполните вывод на дисплей информации о принятых данных, номере канала и коде псевдошума.

2.7. Запрограммируйте устройство и проверьте правильность работы системы. Для этого используйте отладочную программу ComViewer. Если происходит сбой передаваемых пакетов, то канал перегружен. В этом случае следует уменьшить частоту опроса клавиатуры или увеличить скорость передачи данных. Убедитесь, что теперь задержка между опросами клавиатуры мала.

3.1. Для исследования ввода аналоговой информации создайте проект joystick_slave , взяв за основу созданный ранее шаблон WirelessUSB_Pattern. Временно исключите (закомментируйте) из программы инициализацию радио-передатчика WirelessUSB.

3.2. Добавьте в проект модуль АЦП и назовите его ADC. Задайте такие настройки модуля и делителей частот, чтобы получить восьмиразрядный АЦП с полным диапазоном преобразования, например, как показано ниже. В настройках ввода выберите аналоговую шину (AnalogMUXBus). Пример настроек микросхемы приведён на следующей странице:

Global Resources	Value
Power Setting [Vcc / SysClk freq]	3.3V / 24MHz
CPU_Clock	SysClk/8
Sleep_Timer	512_Hz
VC1= SysClk/N	2
VC2= VC1/N	12
VC3 Source	VC2
VC3 Divider	128
SysClk Source	Internal 24_MHz
SysClk*2 Disable	No
Trip Voltage [LVD (SMP)]	3.13V (3.25V)

User Module Parameters	Value
Data Clock	VC2
ADC Input	AnalogMUXBus
PwM_High	2 VC3 Periods
PwM_Low	1 VC3 Period

Рис. 8. Настройки микросхемы и АЦП

3.3. Сгенерируйте проект. Добавьте в код программы функции инициализации и калибровки АЦП. При калибровке в качестве эталонного напряжения возьмите уровень VBG. В настройках платы CYWUSB6953 найдите текущий уровень напряжения питания и рассчитайте соответствующий ему код.

3.4. Задайте АЦП однократный режим преобразования. Для этого в обработчике прерывания АЦП (файл `adcint.asm`) следует добавить вызов функции `ADC_StopADC` с помощью команды `lcall`.

3.5. Подключите аналоговую шину к выводу P0.2, выставив в единицу соответствующий бит в регистре `MUX_CR0`.

3.6. Реализуйте в программе опрос готовности АЦП. Если преобразование закончено, следует считать данные в буфер и переключить канал АЦП. Пример опроса АЦП для аналоговой мультиплексорной шины приведён ниже:

```

UINT8 bData, chan;
// проверка готовности АЦП
if(ADC_fIsDataAvailable())
{
    // считывание данных с АЦП и сброс флага готовности
    bData = ADC_bGetDataClearFlag();

    // проверка номера текущего канала
    if (chan==0)
    {
        LCD_write (1,3,bData);
        chan=1;
        // Подключение P0.2 к аналоговой шине
        MUX_CR0=4;
    }
    else
    {
        LCD_write (1,12,bData);
        chan=0;
        // Подключение P0.0 к аналоговой шине
        MUX_CR0=1;
    }
    // Запуск нового преобразования АЦП
    ADC_StartADC();
}

```

3.7 Настройте программу так, чтобы первому каналу соответствовало перемещение по оси Y (0 - низ, 255 - верх), второму- по оси X (0 - лево, 255 - право).

3.8. Скомпилируйте проект, загрузите его в плату и убедитесь в правильности работы полученного устройства. При изменении положения джойстика код должен меняться в диапазоне 0-255 в зависимости от положения ручки.

3.9. В редакторе устройства (Device Editor) подключите АЦП к аналоговому мультиплексору вместо аналоговой шины. Сгенерируйте проект. При инициализации в регистры управления мультиплексором AMX_IN и ABF_CR0 занесите такие значения, чтобы подключить вывод P0.2 к выходу мультиплексора.

3.10. Реализуйте переключение каналов ввода для аналогового мультиплексора подобно алгоритму для аналоговой мультиплексорной шины.

3.11. Скомпилируйте проект и убедитесь в правильности работы АЦП по критериям из п. 3.8.

3.12. В программе восстановите инициализацию радиопередатчика (уберите комментарий). В настройках протокола задайте режим работы Slave.

3.13. В основном цикле программы добавьте периодическую отправку пустых сообщений. Реализуйте вывод на дисплей информации о текущем состоянии связи.

3.14. По завершении очередного цикла преобразования АЦП произведите отображение результатов на дисплее и передачу данных по радиоканалу в заданном формате.

3.15. В качестве ведущего устройства используйте разработанное ранее ведущее устройство для клавиатуры.

3.16. С помощью программы ComViewer убедитесь в правильности определения положения джойстика. Визуально оцените полную задержку на считывание и передачу информации и укажите возможности её уменьшения.

Источником информации для всех режимов являются данные, получаемые через COM-порт ЭВМ. Программа поддерживает все стандартные режимы работы последовательного интерфейса, которые может обеспечить пользовательский модуль UART в микросхеме. Окно настроек COM приведено ниже:

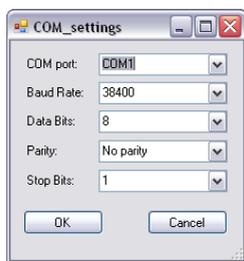


Рис. 2. Окно настройки последовательного порта

Передача информации в программу должна осуществляться пакетами со следующей структурой:

Формат сообщений, обрабатываемый ComViewer. Таблица 1

Смещение	Передаваемая информация
0	Префикс нового пакета 0x7F.
1	Идентификатор устройства.
2 - N+1	N байтов информации

Для всех режимов жёстко заданы форматы информационных пакетов. Примеры для клавиатуры и джойстика приведены в таблицах П2.2 и П2.3.

Переключение режима просмотра возможно в любой момент времени. Примеры окон отображения для различных режимов приведены ниже:

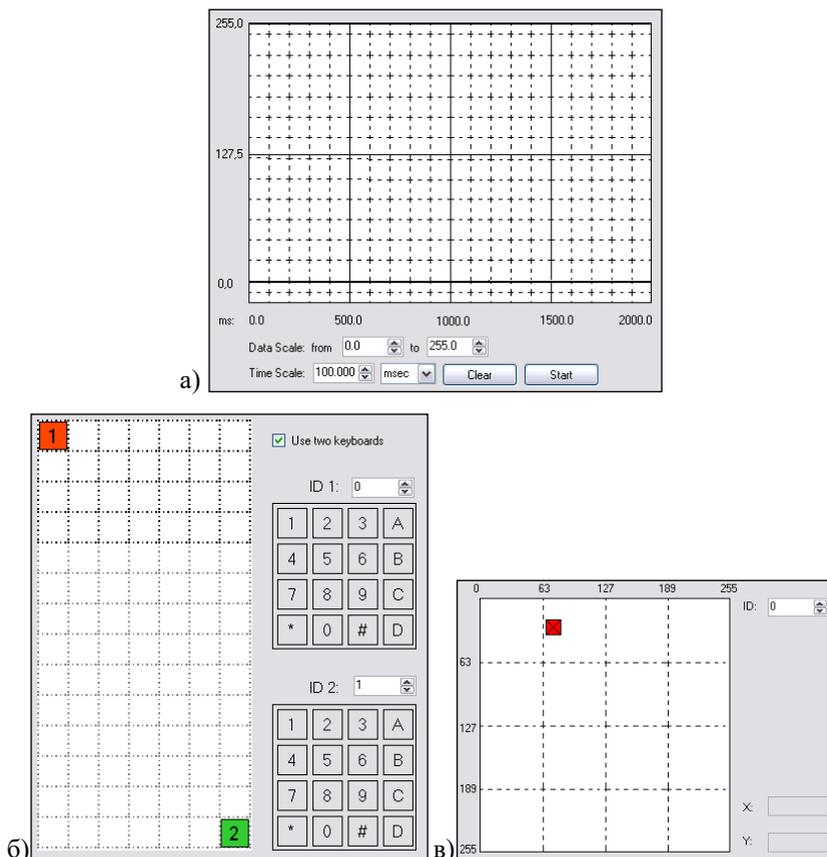


Рис. 3. Окна отображения для различных элементов контроля
(а- графопостроитель, б- клавиатура, в- джойстик)

Отображение полученных данных производится сразу же после получения пакета информации. При этом, для клавиатуры и джойстика возможна фильтрация получаемых сообщений по идентификатору устройства-отправителя.