

Использование инструментария LPCXpresso для разработки приложений на базе 32-разрядных микроконтроллеров NXP с ядрами ARM Cortex-M0 и Cortex-M3

Часть 3

Павел Редькин (г. Ульяновск)

Статья посвящена программно-аппаратному обеспечению LPCXpresso, предназначенному для разработки и отладки приложений для 32-разрядных микроконтроллеров семейств LPC11xx/13xx/17xx производства NXP с ядрами ARM Cortex-M0, Cortex-M3.

НАСТРОЙКИ ПРОЕКТОВ В LPCXPRESSO IDE

Параметры и настройки формирования

Рассмотренные выше операции по формированию и отладке пользовательского приложения выполнялись с параметрами и настройками созданного проекта, заданными по умолчанию. Однако пакет LPCXpresso IDE предоставляет разработчику возможность задавать эти параметры для каждого проекта путём использования соответствующих диалоговых окон.

Для получения доступа к параметрам и настройкам проекта необходимо отметить курсором в окне обозрения проводника проектов *Project Explorer* позицию табуляции этого проекта и выбрать в главном меню *Project > Properties*, после чего по умол-

чанию откроется раздел *Resource* диалогового окна *Properties* (см. рис. 15), где настройки исследуемого проекта сгруппированы в окне *Properties* в виде иерархического дерева с тематическим расположением узлов. В разделе *Resource* содержатся сведения о типе исследуемого ресурса (в нашем случае проекта), его названии, пути к нему, дате последней модификации, типе кодировки его текстовых файлов и т.п.

При выборе курсором в иерархическом дереве узла *Builders* открывается одноимённый раздел диалогового окна *Properties*. В этом разделе разработчик может выбрать, какие инструменты формирования (*Builders*) будут заданы в исследуемом проекте и в каком порядке они будут запускаться на выполнение. Используя соответствующие

кнопки, можно задать для использования новые (не отображённые в окне) инструменты формирования, а также импортировать их извне. Можно также сконфигурировать порядок и условия запуска на выполнение каждого инструмента формирования, предварительно отметив его строку табуляции курсором и щёлкнув на кнопке *Edit...*, после чего откроется окно *Configure Builder*.

При выборе узла *C/C++ Build* открывается одноимённый раздел диалогового окна *Properties*. В этом разделе разработчик может задать детали и особенности выполнения используемого для проекта инструмента формирования. В поле *Configuration*: окна этого и некоторых других разделов можно выбрать, добавить, удалить или переименовать конфигурацию компоновки проекта, для которой производится задание параметров. Активная на данный момент конфигурация компоновки отмечена в поле *Configuration* словом *[Active]*. Чтобы сделать активной другую конфигурацию, необходимо щёлкнуть на кнопке *Manage Configurations...*, после чего откроется одноимённое диалоговое окно с перечнем всех доступных конфигураций компоновки. В этом окне следует отметить позицию табуляции требуемой конфигурации курсором и воспользоваться кнопками *Set Active* и *OK*.

В поле *Makefile generation* включённая по умолчанию настройка *Generate Makefiles automatically* обеспечивает автоматическую генерацию make-файла проекта. Разделы *Build Variables*, *Discovery options*, *Environment* окна *Properties* предназначены для организации управления переменными формирования и процессом формирования (компиляции) для проектов в рабочей области. В разделе *MCU settings* можно задать (изменить) выбор МК для проекта.

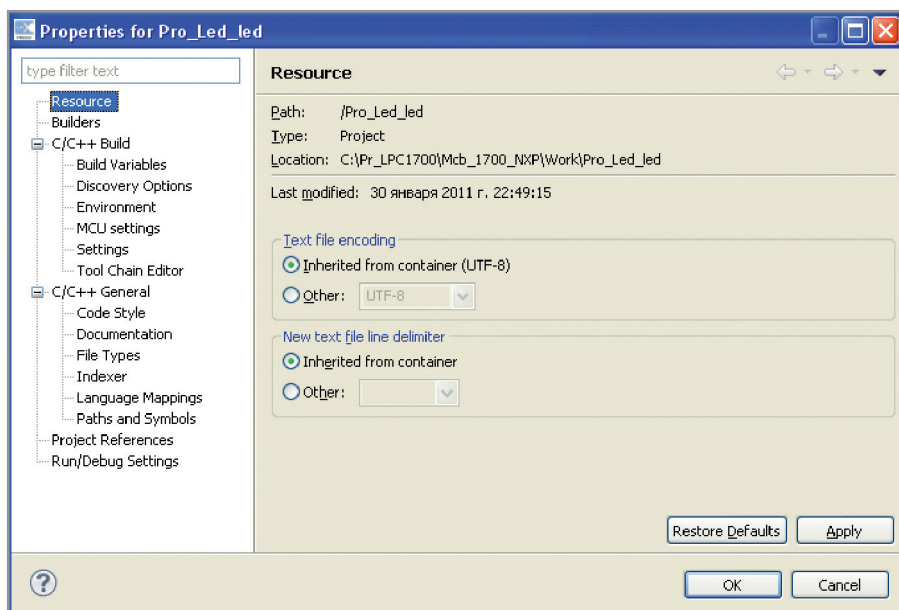


Рис. 15. Раздел *Resource* диалогового окна *Properties*

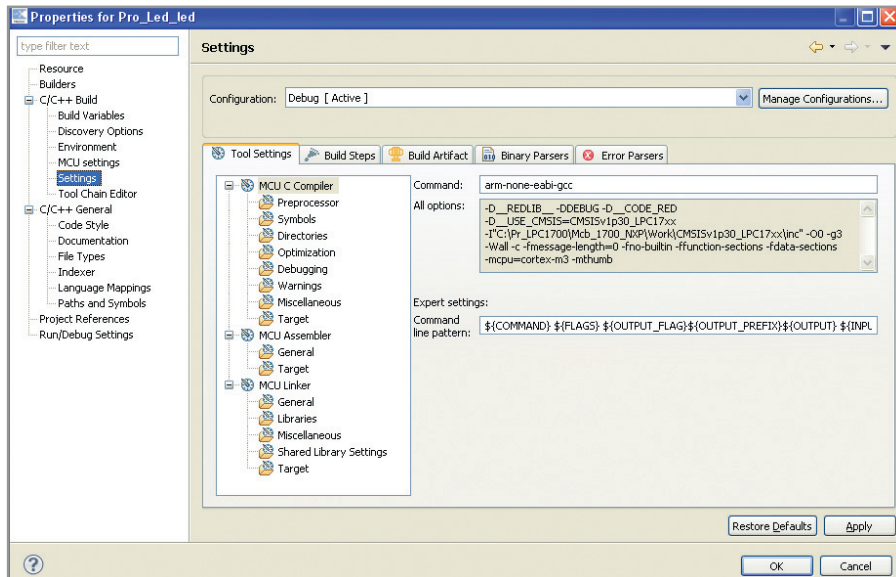


Рис. 16. Страница *Tool Settings* раздела *Settings* диалогового окна *Properties*

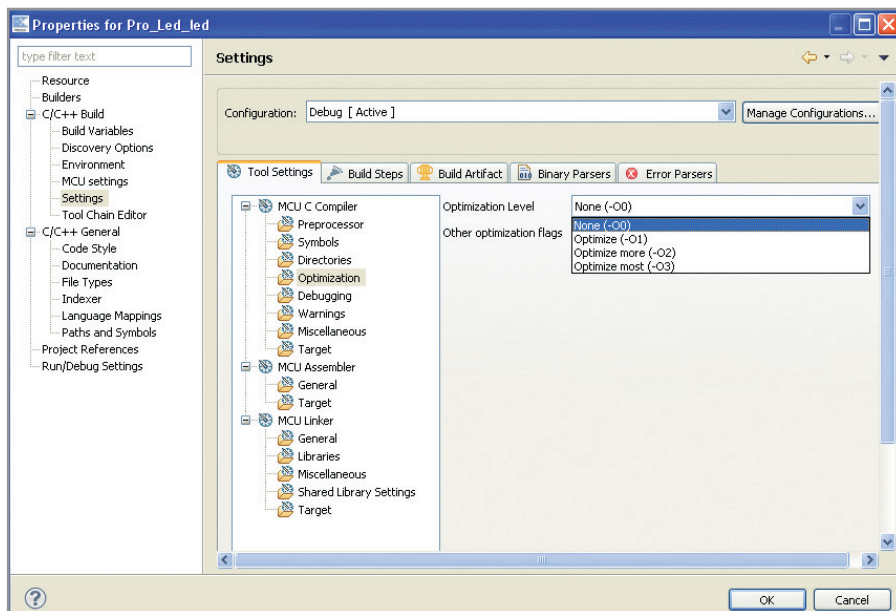


Рис. 17. Задание уровня оптимизации кода на C при компиляции *Optimization Level*

На открытой по умолчанию странице *Tool Settings* раздела *Settings* можно произвести настройку программного инструментария, используемого при работе с проектом: компилятора C, ассемблера и компоновщика (см. рис. 16). Настройки компилятора C (*MCU C Compiler*) включают узлы параметров, относящихся к работе пред-

процессора (*Preprocessor*), заданию относящихся к нему директив и символов (*Symbols*), указанию пути к местоположению каталога подключаемых файлов проекта в рабочей области (*Directories*), заданию уровня оптимизации кода на языке C при компиляции (*Optimization*), уровня (глубины) отладки (*Debugging*), перечня и характе-

Связь между уровнем оптимизации компилятора C с результирующим размером кода для проекта Pro_Led_led

Уровень оптимизации (Optimization Level)	Размер кода из файла Pro_Led_led.axf*		
	text	dec	hex
None (-O0)	2668	3192	c78
Optimize (-O1)	2516	3040	be0
Optimize more (-O2)	2404	2928	b70
Optimize most (-O3)	2372	2896	b50

*Реальный размер файла Pro_Led_led.axf значительно превышает приведённые в таблице значения, однако этот файл является входным для отладчика, но не является кодом «прошивки» флэш-памяти МК

ра генерируемых в ходе компиляции сообщений (*Warnings*), управляющих флагов для компилятора (*Miscellaneous*), а также заданию используемой проектом архитектуры МК целевой системы (*Target*).

Текущие заданные настройки компилятора C перечислены в поле *All options*: (см. рис. 16). По мнению автора, особый интерес для разработчика представляет параметр, задающий уровень оптимизации кода на языке C при компиляции (*Optimization Level*), значение которого можно выбрать из нескольких возможных вариантов, как показано на рисунке 17.

Напомним, что оптимизация кода при компиляции используется для минимизации его результирующего размера с целью экономии флэш-памяти МК. По утверждению производителя, более высокие уровни оптимизации в общем случае будут приводить к ускорению выполнения кода приложения, но также к увеличению размера этого кода [2]. Для отладки приложения производитель рекомендует задавать минимальный уровень оптимизации (-O0), а для выгрузки (при окончательной записи кода в МК) – более высокие уровни оптимизации. Влияние заданного уровня оптимизации на результирующий размер его кода можно оценить с помощью полученных автором экспериментальных данных, представленных в таблице. Приведённые в таблице значения взяты из информации, отображаемой по завершении формирования проекта в окне обозрения *Console*.

Доступ к настройкам ассемблера можно получить, выбрав соответствующий узел в дереве *MCU Assembler*. Текущие настройки ассемблера будут перечислены в поле *All options* открывшегося окна; они включают параметры общего назначения, относящиеся к заданию управляющих флагов и путей к местоположению подключаемых файлов (*General*), а также к заданию используемой архитектуры МК целевой системы (*Target*). В параметрах узла *General* можно задать такой параметр, как уровень отладки (*Debug level*), очевидно, представляющий собой степень детализации показа ассемблерных инструкций при пошаговой передаче управления в программе в ходе её отладки/выполнения.

Доступ к настройкам компоновщика можно получить, выбрав соответствующий узел в дереве *MCU Linker*.

Текущие настройки компоновщика будут перечислены в поле *All options* открывшегося окна; они включают узлы параметров общего назначения (*General*), задание библиотек (*Libraries*), управляющих флагов для компоновщика (*Miscellaneous*), настроек общедоступных библиотек (*Shared Library Settings*), а также используемой архитектуры МК целевой системы (*Target*).

В разделе *Settings*, помимо страницы *Tool Settings*, представлены страницы *Build Steps*, *Build Artifact*, *Binary Parsers* и *Error Parsers*. На странице *Build Artifact* можно задать тип, название и расширение файла, являющегося результатом формирования, страницы *Binary Parsers* и *Error Parsers* содержат

настройки, позволяющие задавать инструментальные анализаторы кода и ошибок, используемые при формировании.

В разделе *Tool Chain Editor* диалогового окна настроек проекта *Properties* можно выбрать набора программный инструментарий (поле *Current tool-chain*), а также указать инструмент формирования из состава выбранного набора (поле *Current builder*). Если в разделе *Tool Chain Editor* щёлкнуть на кнопке *Select Tools...*, то откроется одноимённое диалоговое окно, в котором можно выбрать для каждой используемой категории инструментальных средств формирования *Used tools* (компилятор C++, компилятор C, ассемблер, компоновщик, компоновщик

C++) соответствующий вариант из списка доступных инструментальных средств *Available tools*, как показано на рисунке 18.

Настройки и параметры узла C/C++ General диалогового окна *Properties* определяют, какой инструментарий отображения должен использоваться, чтобы определять вид и возможности редактора LPCXpresso IDE. К последним относятся генерация комментариев, подсветка синтаксиса и т.д.

В разделе *Code Style* узла C/C++ General можно задать параметры, относящиеся к стилю, особенностям представления, отступам и т.д. различных программных конструкций исходного текста на языке C. Текущие параметры исходного текста в виде примеров

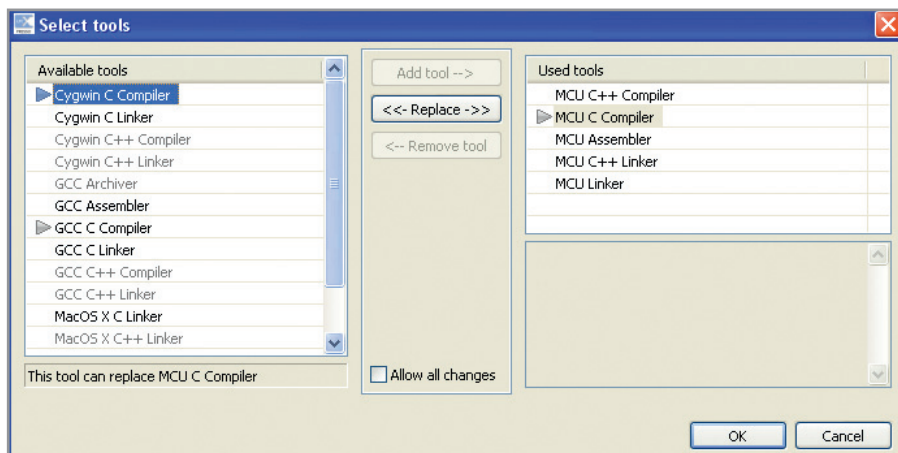


Рис. 18. Диалоговое окно выбора инструментальных средств *Select Tools*

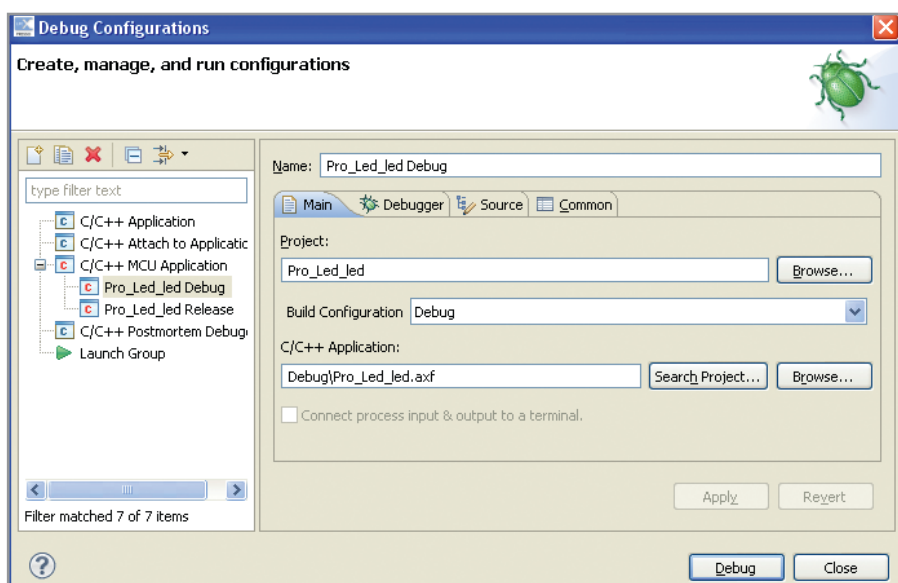


Рис. 19. Диалоговое окно конфигурации отладки *Debug configurations*

фрагментов программных конструкций можно наблюдать в поле *Preview*.

Раздел *Paths and Symbols* узла C/C++ General содержит несколько страниц. По умолчанию открыта страница *Includes*, с которой можно задать (изменить) пути к подключаемым файлам проекта. На странице *Symbols* можно задать (изменить) список символов препроцессора. На странице *Library paths* можно задать (изменить) пути к используемым в проекте библиотекам. На странице *Source location* можно задать (изменить) путь к исходным файлам проекта.

На странице *References* можно редактировать принадлежность опорной информации исследуемого проекта к различным конфигурациям компоновки этого проекта. Здесь отображаются все проекты, которые доступны для использования в качестве опорной информации. Напомним, что в качестве опорной информации для нашего проекта был задан библиотечный проект CMSISv1p30_LPC17xx. Помимо это-

го, на странице *References* отображаются все конфигурации исследуемого проекта; активная конфигурация отмечена словом *[Active]*. Чтобы задать принадлежность опорного проекта требуемой конфигурации компоновки исследуемого проекта, следует отметить его позицию табуляции галочкой в квадрате слева (может быть помечена только одна конфигурация).

В узле *Project References* может быть задана (отменена) принадлежность доступных опорных проектов исследуемому проекту в целом (ко всем его конфигурациям компоновки). В узле *Refactoring History* можно наблюдать в виде иерархического дерева хронологический отчет о действиях пользователя, касающихся всех модификаций исследуемого проекта.

Параметры и настройки конфигурации выполнения/отладки

Доступ к параметрам и настройкам конфигурации выполнения/от-

ладки загруженного в приложения возможен через меню *Run* перспективы *Debug (Run > Run Configurations... или Run > Debug Configurations...)*. Кроме того, можно выбрать в окне обозрения «быстрого» запуска *Quickstart Panel > Debug and Run > Open debug configurations*. После этого откроется одноимённое диалоговое окно конфигурации отладки (см. рис. 19).

На открытой по умолчанию странице *Main* окна конфигурации отладки *Debug Configurations* можно выбрать для задания параметров отладки любую из двух конфигураций компоновки приложения (*Debug* и *Release*) путём выбора её строки табуляции из выпадающего меню. Указанный выбор можно также произвести, отметив курсором позицию табуляции требуемой конфигурации компоновки в иерархическом дереве в левой части окна. На этой же странице можно задавать (идентифицировать) название выполняемого/отлаживаемого проекта, а также название используемого для отладки файла кода приложения.

На странице *Debugger* окна конфигурации отладки можно задавать параметры и настройки используемого отладчика. Включение настройки *Stop on startup at: main* позволяет после сброса целевого МК начать выполнение/отладку приложения с вызова основной функции *main* его управляющей программы. На странице *Source* окна конфигурации отладки можно задавать используемые для отладки исходные файлы и пути к ним. На странице *Common* окна конфигурации отладки можно задавать настройки общего назначения, связанные с отображением и сохранением ресурсов приложения.

ЛИТЕРАТУРА

1. Development Tools for ARM-based microcontrollers – Select from the best in support. August 2010. <http://ics.nxp.com/literature/other/microcontrollers/pdf/arm.mcu.tools.pdf>.
2. LPCXpresso. Getting started with NXP LPCXpresso. User guide. Rev. 7. 15 September 2010.
3. Редькин П.П. 32-битные микроконтроллеры NXP с ядром Cortex-M3 семейства LPC17xx. Полное руководство. Додэка-XXI, 2011.
4. <http://www.keil.com/mcb1700/mcb1760.asp>.



Поставки крупных ЖК-панелей выросли во II кв.

Поставки жидкокристаллических панелей крупного размера увеличились до 180,8 млн. во втором квартале 2011 г., что на 11% больше по сравнению с первым кварталом и на 6% больше по сравнению с прошлогодними показателями. При этом доходы производителей составили \$19,3 млрд., что на 11% лучше по сравнению с первым кварталом, но на 16% хуже по сравнению с прошлым годом, — отмечает аналитическое агентство DisplaySearch.



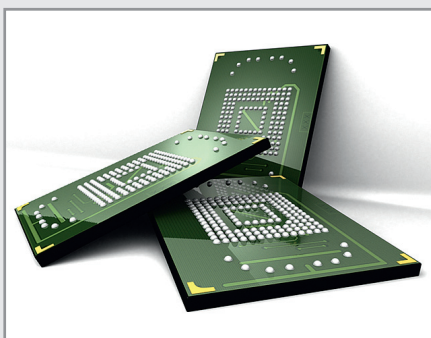
Согласно экспертам, отрасль ЖК-панелей более года страдала от перепроизводства. И первый квартал 2011 г. стал точкой минимума. Во втором квартале цены немного стабилизировались, а поставки и доходы увеличились. Тем не менее, аналитики считают, что это лишь кратковременное улучшение ситуации, и застойное состояние мировой экономики не позволит отрасли существенно подняться.

Большинство производителей ЖК-телевизоров снизили прогнозы касательно спроса на свою продукцию. Это позволит им сократить затраты и удерживать складские запасы на низком уровне.

<http://www.digitimes.com/>

В августе цены на NAND снижаются медленнее

Согласно данным DRAmEXchange, контрактные цены на NAND-память в первой половине августа снижались менее резко по сравнению с июлем. Так, MLC-микросхемы массового сегмента подешевели всего на 1...2%.



Отраслевые источники связывают улучшение ситуации на рынке с активизацией компании Apple. Разработчик легендарных iPad и iPhone с начала августа начал размещать объёмные заказы на флэш-память.

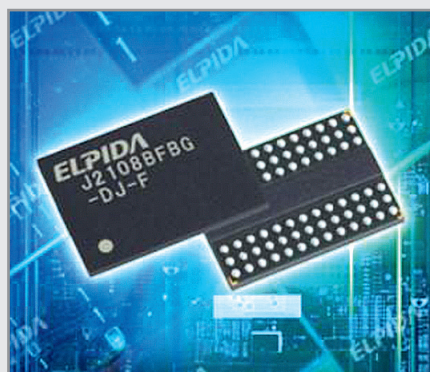
Кроме того, отмечается увеличение спроса на карты памяти для телефонов. Тем не менее, аналитики не берутся прогнозировать, сможет ли увеличение спроса успеть за ростом темпов производства. Если нет, то тенденция к снижению цен в отрасли может сохраниться. Ключевые игроки рынка, такие как Samsung Electronics, Toshiba и Micron, продолжают расширять объёмы производства NAND-памяти во втором полугодии.

По состоянию на начало августа контрактные цены на 32-Гбит MLC NAND-чипы упали на 0,5...1,5%. TLC-чипы такой же ёмкости подешевели на 7%. Средние цены на 16- и 64-Гбит MLC-микросхемы упали на 2 и 1% соответственно.

<http://www.digitimes.com/>

Elpida выпустила 25-нм DDR3-чипы

Японская компания Elpida Memory с гордостью заявила о запуске производства DRAM-чипов, выполненных по проектным нормам 25 нм. Также она утверждает, что её новые 2-Гбит микросхемы DDR3-памяти, которые функционируют с эффективной частотой 1866 МГц, являются самыми компактными в отрасли.



По сравнению с 30-нм решениями, 25-нм чипы отличаются на 15% меньшим током нагрузки в рабочем режиме, а экономия в режиме ожидания достигает 20%. Новые чипы Elpida нацелены на использование в настольных ПК и серверных приложениях, но производитель обещает также выпустить 25-нм решения для мобильных устройств.

Кроме того, Elpida планирует запустить производство 25-нм чипов ёмкостью 4 Гбит. Это может случиться уже в текущем году.

<http://www.elpida.com/>

Количество устройств с поддержкой 802.11n возрастёт на 465% к 2015 г.

Компания In-Stat провела исследование, согласно которому число устройств с поддержкой Wi-Fi 802.11n возрастёт к 2015 г. на 465%. За последний год во всём мире число приобретённых электронных устройств с поддержкой данного протокола составило 53 млн., а в следующем году аналитики прогнозируют рост их числа до 82 млн. К 2015 г. количество подобных решений, приобретаемых по всему миру, достигнет 300 млн. единиц в год.



Более ранние версии протокола Wi-Fi, такие как 802.11a/b/g, используют для связи частоту 2,4 ГГц. В отличие от них, Wi-Fi 802.11n работает на частоте 5 ГГц. Данный протокол позволяет устройствам передавать информацию на более высокой скорости и больших расстояниях. Теоретическая граница скорости передачи данных для 802.11n составляет 600 Мбит/с. Для сравнения, 802.11g позволяет устройствам обмениваться данными на скорости до 54 Мбит/с.

Важность протокола Wi-Fi для современной мобильной электроники сложно переоценить. Так, аналитики из In-Stat прогнозируют, что к 2013 г. 85% всех производимых e-ридеров будут поддерживать этот стандарт беспроводной связи. Кроме того, за 2013 г. по всему миру будет продано около 750 млн. смартфонов с поддержкой Wi-Fi различных версий.

Несмотря на то что на сегодняшний день именно 802.11n является последней версией протокола, различные производители под эгидой Wi-Fi Alliance готовятся в скором времени внедрить его замену. Она получит название 802.11ac, будет также работать на частоте 5 ГГц и позволит передавать данные на скорости до 1 Гбит/с. Новые спецификации стандарта будут внедрены в 2012 или 2013 г.

<http://www.cnet.com/>