

Беспроводная синхронизация бортовых вычислительных устройств при помощи WiFi

Романов А.М.^{1*}, Гринголи Ф.^{2**}, Сикора А.^{3***}

¹МирЭА - Российский технологический университет", проспект Вернадского, 78,
Москва, 119454, Россия

²Университет Брешии, пл. Спедали Цивили, 1, Брешия, 25100, Италия

³Университет Оффенбурга, Бадштрассе, 24, Оффенбург, 77652, Германия

*e-mail: romanov@mirea.ru

**e-mail: francesco.gringoli@unibs.it

***e-mail: axel.sikora@hs-offenburg.de

Статья поступила 09.10.2019

Аннотация

Работа посвящена обеспечению точной беспроводной синхронизации бортовых устройств управления малоразмерными беспилотными летательными аппаратами (МБПЛА). Решение данной проблемы востребовано в таких областях, как групповой мониторинг окружающей среды, обслуживание складского хозяйства, организация шоу и театральных представлений. В данной работе предлагается новый подход к синхронизации, который основан на анализе потенциала входа радиоприемника интегрированной микросхемы WiFi, реализованной по технологии система-на-кристалле. Преимуществом данного подхода является отсутствие необходимости в демодуляции и декодировании принимаемых пакетов, что позволяет реализовывать его с использованием малогабаритных бюджетных компонент. В работе, на примере микросхемы Cypress CYW43438, описываются схемотехнические и программные решения, необходимые для решения задачи

беспроводной синхронизации бортовых вычислительных устройств МБПЛА. Результаты проведенных экспериментальных исследований демонстрируют ошибку взаимной синхронизации двух бортовых вычислительных устройств менее 10 мкс.

Ключевые слова: беспроводная синхронизация, WiFi, джиттер, малоразмерные беспилотные летательные аппараты, программируемые логические интегральные схемы.

Введение

В последние годы все большее практическое применение в различных областях экономики находят малоразмерные беспилотные летательные аппараты (МБПЛА) и в первую очередь квадрокоптеры. Изначальным толчком к этому послужило бурное развитие рынка радиомоделей, предназначенных для досуга, а также конструкторов для любителей моделизма. Это позволило существенно снизить цены на все основные комплектующие, а также отработать технологии управления, обеспечивающие надежность, достаточную для коммерческого применения. Первой областью, которая взяла на вооружение такие радиомодели стала профессиональная фотосъемка. На сегодняшний день МБПЛА аппараты для коммерческого применения можно уже выделить в отдельную группу. Они используются для мониторинга состояния окружающей среды [1], аэрофотосъемки [2], ликвидации последствий стихийных бедствий [3] и т. д. Причем, во многих случаях их управление осуществляется уже не по радиоканалу, а автономно при помощи бортового устройства управления. В то же время, сочетание растущей сложности

решаемых задач с достаточно серьезными ограничениями по грузоподъемности, радиусу действия и времени автономной работы, заставляет разработчиков смотреть в сторону использования не одного беспилотного аппарата, а их группы, действующей в рамках единого закона управления. В отдельных случаях мониторинга состояния окружающей среды [4], групповое управление может быть реализовано достаточно просто, так как после распределения задач и формирования траекторий полета каждый робот может действовать независимо от других участников группы. В то же время, более сложные задачи мониторинга требуют поддержания строя [<http://trudymai.ru/published.php?ID=85774>

5], и как следствие синхронизации действий между различными аппаратами. Синхронизация между аппаратами требуется при решении задач по переноске тяжёлых объектов [6, 7], актуальной, например, при обслуживании складских хозяйств. Ещё одной областью применения взаимной синхронизации является организация шоу и театральных представлений с использованием МБПЛА [8], где роботы должны не только взаимодействовать между собой, но и перемещаться синхронно с музыкой и применяемыми визуальными эффектами (видео, дым, управляемый свет и т. д.).

На открытом воздухе задача синхронизации может решаться при помощи спутниковых систем единого времени [9]. Однако, данный подход не подходит для закрытых помещений, что ограничивает его применение в складской логистике и театральной технике. Существующие наземные аналоги спутниковых навигационных систем потенциально предоставляют крайне высокую, а для

большинства задач — избыточную, точность синхронизации (менее 1 нс)
[<http://trudymai.ru/published.php?ID=87284>

10], но в то же время, являются крайне дорогостоящими, что делает их применение в большинстве описанных задач экономически нецелесообразным. Таким образом, актуальной является разработка новых решений беспроводной синхронизации бортовых вычислителей МБПЛА гражданского применения, обеспечивающих точность не хуже 10 мкс (что достаточно для решения большинства задач группового управления движением МБПЛА) и обладающих простотой, малыми габаритами и низкой себестоимостью.

Анализируя элементную базу автономных МБПЛА, предназначенных для гражданского применения, можно сделать вывод, что формирование целевой траектории и стратегии управления аппаратом в целом в них реализуется на базе одноплатных компьютеров, наиболее распространённым из которых является Raspberry Pi. Этот компьютер сочетает в себе достаточно высокую вычислительную мощность, малые массогабаритные размеры, низкую стоимость и наличие на борту средств радиообмена по технологиям Bluetooth (IEEE 802.15.1) и WiFi (IEEE 802.11).

Целью данной работы является разработка решения для взаимной беспроводной синхронизации бортовых вычислительных устройств на базе Raspberry Pi по каналу WiFi, обладающего малыми габаритными размерами и низкой себестоимостью.

1. Проблемы беспроводной синхронизации по каналам WiFi

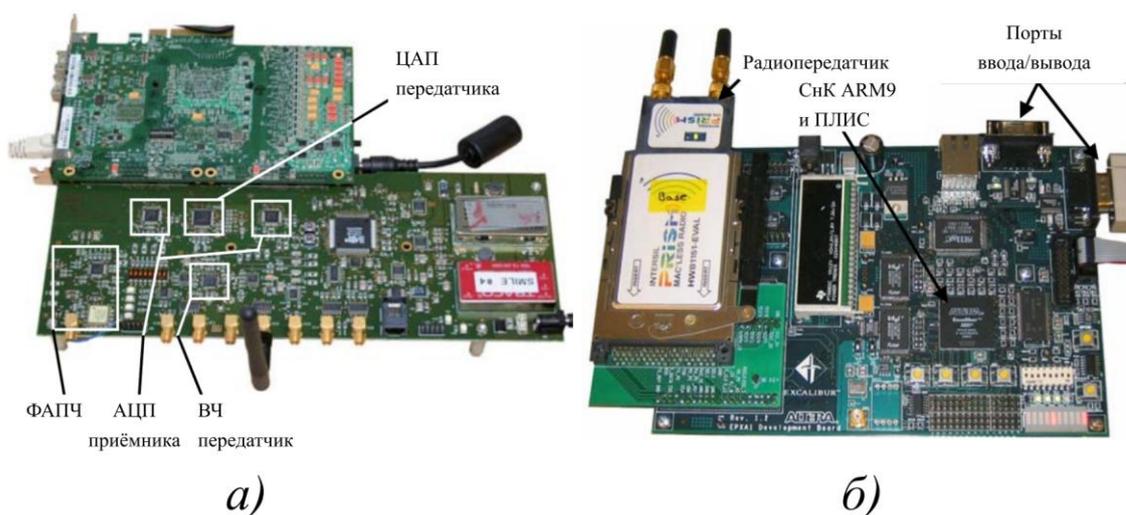
В системах беспроводной синхронизации наибольшую точность дают методы,

[<http://trudymai.ru/published.php?ID=58802>

11]. Суть этих методов состоит в вычислении на основе меток времени приёма и отправки одного и того же пакета на разных устройствах, величины транспортных задержек, фазового сдвига и, в отдельных случаях, коэффициента расхождения частот тактовых генераторов участников информационного обмена. Ключевыми параметрами, влияющими на точность такой синхронизации, являются точность и джиттер определения моментов времени приёма и отправки пакетов.

Наилучшие результаты в этой области демонстрируют специализированные заказные приёмо-передающие устройства [<http://trudymai.ru/published.php?ID=58802>

11-13], построенные на базе дорогостоящих программируемых логических интегральных схем (ПЛИС), таких как Altera (Intel) Startix или Xilinx Virtex. Помимо высокой цены, такие устройства имеют существенные габариты (рис. 1), что делает



их малоприспособными для МБПЛА.

Рис. 1. Специализированные заказные приёмо-передающие устройства стандарта

IEEE802.11: а) Проект Smile [13]; б) Excalibur EPXA1 [12]

Можно отметить, что по совокупности своих характеристик, они, в первую очередь, предназначены для проведения исследований, а не для применения в промышленных образцах. Ключевой особенностью таких устройств с точки зрения синхронизации является, регистрация меток времени непосредственно после демодуляции радиосигналов, что позволяет минимизировать задержки и возможный джиттер.

В то же время, в промышленности и потребительских устройствах наибольшее распространение получили интегрированные микросхемы беспроводной связи (ИМБС), построенные по технологии системы-на-кристалле (СнК). Такие микросхемы включают в себя несколько процессоров и цифроаналоговую часть, необходимую для модуляции, демодуляции и усиления радиосигналов. Ярким примером ИМБС является микросхема Cypress CYW43438, используемая в одноплатном компьютере Raspberry Pi 3В (рис. 2).

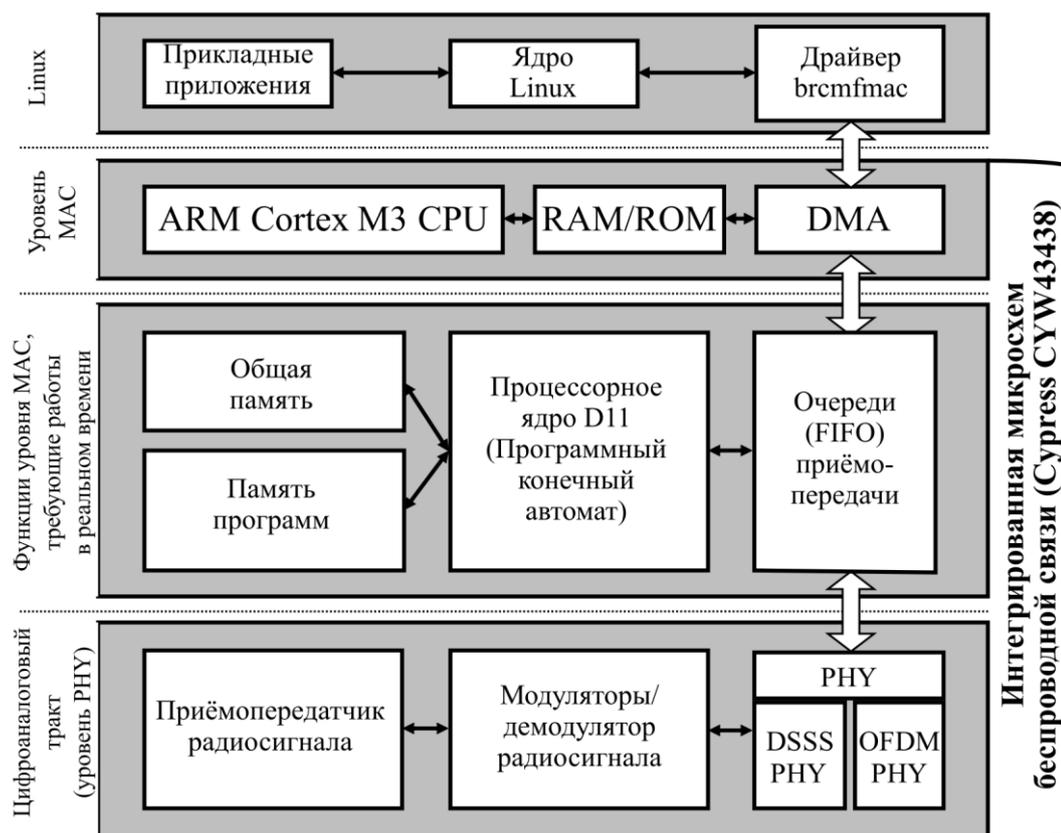


Рис. 2. Структурная схема интегрированной микросхемы беспроводной связи Cypress CYW43438, используемой в одноплатном компьютере Raspberry Pi 3B

Как видно из (рис. 2), в состав Cypress CYW43438 входят два процессора (ARM Cortex M3 и D11), которые реализуют уровень MAC (кодирование и декодирование пакетов, а также обеспечение правильного порядка информационного обмена в сетях стандарта IEEE 802.11) и цифроаналоговый тракт, который реализует уровень PHY (модуляция и демодуляция, а также усиление радиосигналов). Использование СнК является более технологичным, чем применение отдельных микросхем MAC и PHY и позволяет создавать бюджетные и энергоэффективные устройства беспроводной связи. В тоже время, такая архитектура ограничивает возможности доступа разработчика непосредственно к PHY и существенно усложняет получение точных меток времени для принимаемых и отправляемых пакетов, необходимых в задачах взаимной синхронизации и локализации

промышленных устройств. Попытка же программно регистрировать метки времени на выходе ИМБС через стандартный интерфейс ухудшает точность синхронизации до 234 мкс [14], при этом, джиттер существенно зависит как от загрузки сети, так и от загрузки процессора. Это связано с тем, что процессор СнК, отвечающий за обработку принимаемых пакетов, связан с центральным процессором одноплатного компьютера через интерфейс прямого доступа к памяти (DMA), характеризующимся нестабильными запаздываниями. Также, существенные задержки вносятся на уровне драйвера и ядра операционной системы [14].

Ряд современных ИМБС, включая такие как Broadcom BCM4712 и Cypress CYW43438, обладают встроенным функционалом, позволяющим регистрировать метки времени по внутренним часам микросхемы, однако, из-за высокого джиттера штатного DMA интерфейса, эти метки времени можно использовать только для синхронизации нескольких ИМБС между собой, но не для синхронизации бортовых вычислителей, в состав которых входят эти ИМБС.

Основная часть данной работы посвящена обеспечению синхронизации бортовых вычислителей, построенных на одноплатных компьютерах Raspberry Pi 3B, при помощи WiFi и решению задачи синхронизации ИМБС с центральным процессором бортового вычислителя, как ключевой проблемы, необходимой для достижения поставленной цели.

2. Новый метод синхронизации интегрированных микросхем беспроводной связи с бортовым вычислительным устройством

Как было отмечено выше, современные микросхемы беспроводной связи

стандарта IEEE 802.11 не имеют специализированного интерфейса реального времени для их синхронизации с внешними устройствами. Для решения этой проблемы в данной работе предлагается новый подход, основанный на анализе потенциала входа радиоприемника микросхемы Cypress CYW43438.

В основе предлагаемого подхода лежит наблюдение о том, что вход радиоприёмника ИМБС изменяет свой потенциал на близкий к 0В в момент передачи (рис. 3), в то время как в режиме приёма потенциал этого входа приблизительно составляет 600 мВ.

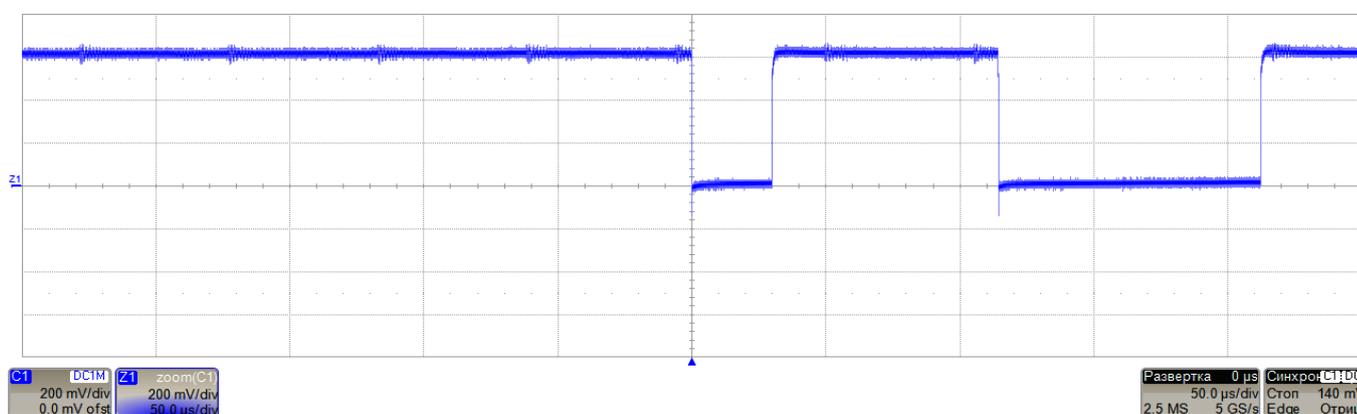


Рис. 3. Осциллограмма потенциала на входе приёмник интегрированной микросхемы беспроводной связи стандарта IEEE 802.11 (Cypress CYW43438)

Наиболее вероятной причиной этого изменения является желание разработчиков оградить приёмник от получения своих же собственных пакетов. Управление этим механизмом осуществляется аппаратно в цифроаналоговом тракте, отвечающем за физический уровень стандарта IEEE 802.11, поэтому потенциально имеющим меньший джиттер, по сравнению с интерфейсами, реализованными программно на базе процессорных ядер, входящих в состав микросхемы.

Предлагаемый метод синхронизации ИМБС с бортовым вычислительным устройством включает в себя несколько этапов (рис. 4).

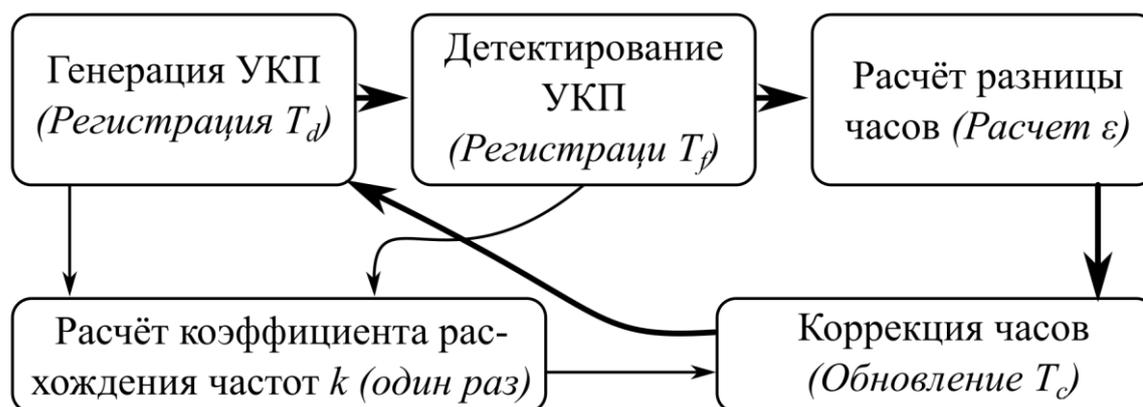


Рис. 4. Основные этапы предлагаемого метода синхронизации интегрированной микросхемы беспроводной связи стандарта IEEE 802.11 с бортовым вычислительным устройством

На первом этапе каждого цикла синхронизации при помощи ИМБС формируется исходящий ультракороткий пакет (УКП), длина которого существенно отличается в меньшую сторону от пакетов, используемых при радиопередаче по стандарту IEEE 802.11, и регистрируется T_d , время начала его передачи по встроенным часам ИМБС. Далее, постоянно анализируя уровень на входе радиоприёмника ИМБС при помощи бюджетной ПЛИС, установленной на плате расширения бортового вычислительного устройства, УКП выделяется из всего потока передаваемой информации по его длительности и регистрируется T_f , момент времени начала его передачи по внутренним часам ПЛИС.

На основании 100-1000 последовательно полученных пар T_d и T_f методом наименьших квадратов строится их линейная аппроксимация (1) и находится коэффициент расхождения частот k .

$$T_d(T_f) \approx k T_f + b \quad (1)$$

Данное значение k рассчитывается на этапе пусконаладочных работ один раз для каждого конкретного бортового вычислительного устройства и не меняется в

процессе работы.

Дальнейшие действия по синхронизации выполняются циклически и направлены на вычисление фазового сдвига T_c между внутренними часами ИМБС по которым выставляются метки времени T_d и синхронизированными с ними часами ПЛИС T_{fd} , чье значение в каждый момент времени вычисляется по формуле (2)

$$T_{fd}(T_f) = k(T_f + T_c) \quad (2)$$

На первом цикле значение T_c принимается равным 0. В дальнейшем оно вычисляется по формулам (3-4).

$$\varepsilon_n = T_{dn} - T_{fd}(T_{fn}) \quad (3)$$

$$T_{dn+1} = T_{dn} + \text{round}\left(\frac{\varepsilon_n}{k}\right), \quad (4)$$

где T_{dn} – это значение T_d полученное на n-ом цикле синхронизации;

T_{fn} – это значение T_f полученное на n-ом цикле синхронизации;

T_{cn} – это значение T_c полученное на n-ом цикле синхронизации;

round – функция округления до ближайшего целого, используемая для приведения T_c в целочисленный формат, в котором хранится текущее значение часов;

ε_n – ошибка синхронизации на начало n-ого цикла.

На практике, сразу по нескольким причинам, может оказаться невозможным получить метки времени T_d и T_f в течение одного или нескольких подряд идущих циклов и, тем более, гарантировать постоянство периода циклов синхронизации. Во-первых, команда на формирование УКП и выгрузку метки времени T_d

осуществляется по интерфейсу МАС-уровня, который, в совокупности с взаимодействующим с ним драйвером операционной системы, обладает существенным джиттером [14]. Во-вторых, даже если управляющая команда вовремя поступила процессору ИМБС, отвечающему за формирование УКП, это не даёт гарантии, что УКП может быть немедленно отправлен, так как в этот момент радиоэфир может быть занят другим устройством. Также передатчик может быть занят отправкой данных прикладного приложения, не связанных с задачами синхронизации. Наконец, значение T_f может быть получено некорректно, например, в результате ложного срабатывания на случайную помеху. Последний случай, тем не менее, можно достаточно легко диагностировать, если в течение нескольких подряд идущих циклов наблюдаются резкие скачки ошибки синхронизации ε большой амплитуды, несоизмеримые с амплитудой изменения ε на предыдущих циклах. В рамках разработанного алгоритма синхронизации, если пара T_d и T_f не была получена на каком-либо цикле синхронизации или была признана ошибочной алгоритмом диагностики, данный цикл синхронизации пропускается, а значение T_c остается неизменным до начала следующего цикла.

Как видно, точное поддержание периода цикла синхронизации для предложенного метода не требуется. Напротив, в ряде случаев он может автоматически увеличиваться в несколько раз за счёт пропуска циклов, на которых не удалось рассчитать корректное значение T_c . В то же время, чрезмерное увеличение времени между обновлением T_c также нежелательно, так как в этом случае, в значениях T_{fd} будет неминуемо накапливаться ошибка, связанная с

ограниченной точностью вычисления k и нестабильностью тактовых генераторов, как ИМБС, так и внешнего устройства, с которым осуществляется синхронизация.

После того, как решена задача синхронизации ИМБС с бортовым вычислительным устройством синхронизация нескольких бортовых вычислительных устройств по беспроводному WiFi каналу является тривиальной. Согласно стандарту IEEE 802.11 точка доступа WiFi регулярно осуществляет безадресную широковещательную передачу пакетов-маяков (beacon frames) для оповещения других устройств о своём присутствии. Данные пакеты содержат время отправки по часам точки доступа и по сути являются источником единого времени для всех устройств сети. В момент приёма пакета-маяка ИМБС регистрирует для него метку времени по встроенным часам ИМБС. Таким образом на каждом цикле отправки пакета маяка можно определить смещение и коэффициент расхождения частот между часами ИМБС и общим временем точки доступа. С использованием этих значений на каждом из бортовых устройств управления можно получить единое время точки доступа. Данный метод синхронизации не учитывает различие задержек распространения сигнала от точки доступа до разных бортовых устройств управления, так как для расстояний менее 100 м (дальность работы подавляющего большинства точек доступа WiFi) эти задержки не превышают 350 нс и могут считаться несущественными.

В следующих разделах рассмотрим прикладные аспекты реализации предложенного метода синхронизации ИМБС Cypress CYW43438 с бортовым вычислительным устройством на базе одноплатного компьютера Raspberry Pi 3B.

3. Генерация ультракоротких пакетов

Первой проблемой на пути реализации предложенного метода синхронизации является необходимость формирования УКП и регистрации меток времени в момент его фактической отправки. Стандартный функционал ИМБС не позволяет этого сделать, поэтому в рамках данной работы было разработано модифицированное программное обеспечение для микросхемы CYW43438. Чтобы лучше понять суть внесенных изменений, рассмотрим структурную схему, представленную на рис. 2.

Функционал MAC микросхемы Cypress CYW43438 реализован в два уровня с использованием двух независимых процессоров. Процессор ARM Cortex M3 обеспечивает те функции информационного обмена, регламентированные стандартом IEEE 802.11, которые наименее критичны ко времени исполнения. Второй процессор, отмеченный на рис. 2 как D11, отвечает за реализацию программного конечного автомата ИМБС и тех функций, которые требуют работы в режиме реального времени. Именно этот процессор в составе ИМБС Cypress CYW43438 отвечает за взаимодействие с РНУ [15].

В то время как код самого драйвера `brctmfmac` для операционной системы Linux является открытым, программное обеспечение для обоих процессоров ИМБС Cypress предоставляет только в виде заранее скомпилированных бинарных образов. Тем не менее, благодаря проектам Nexmon [16] и OpenFWWF [17], существует возможность для ИМБС Cypress и Broadcom модифицировать функционал MAC уровня [18]. Также, инструментарий Nexmon позволяет добавлять новые системные вызовы в программный код ИМБС, а потом активировать их при помощи команд

бортового вычислительного устройства с использованием утилиты *nexutil* [15]. На основе этих инструментов, в рамках данной работы было разработано новое программное обеспечение для микропроцессора D11, которое обеспечивает формирование УКП и регистрацию времени их отправки.

Разработанное программное обеспечение интегрируется в программный конечный автомат, реализованный на процессоре D11, и проверяет, используется ли в данный момент радиоканал для приёма или отправки пакетов. В случае, если была получена команда на формирование УКП, в ближайший момент времени, когда радиоканал окажется свободен, инициируется отправка исходящего пакета на скорости 6 Мб/с при помощи функции немедленной передачи. Момент времени начала передачи фиксируется по внутренним часам D11, хранящимся в регистре PSM_clock. После того, как будет передана преамбула пакета и разделитель начала кадра (SFD), вызывается автоматическое прерывание передачи, что позволяет сгенерировать пакет существенно короче тех, которые используются при стандартном радиобмене в сетях WiFi. Здесь важно отметить, что формируемые УКП не содержат контрольной суммы, в результате чего они не принимаются другими устройствами, автоматически отсеиваясь на уровне их микросхем беспроводной связи.

На рис. 5 показана осциллограмма потенциала входа радиоприёмника ИМБС в момент формирования УКП.

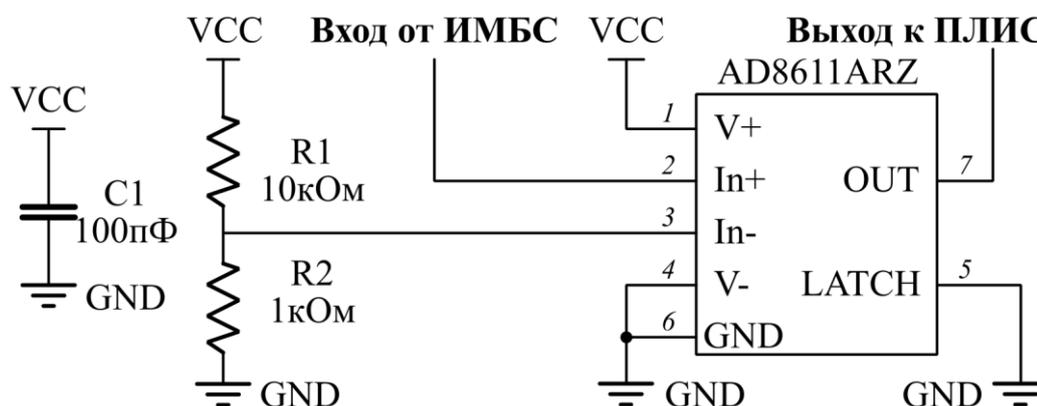


Рис. 6. Схема согласования сигнала на входе радиоприёмника интегрированной микросхемы беспроводной связи стандарта IEEE 802.11 (Cypress CYW43438) и дискретного входа ПЛИС

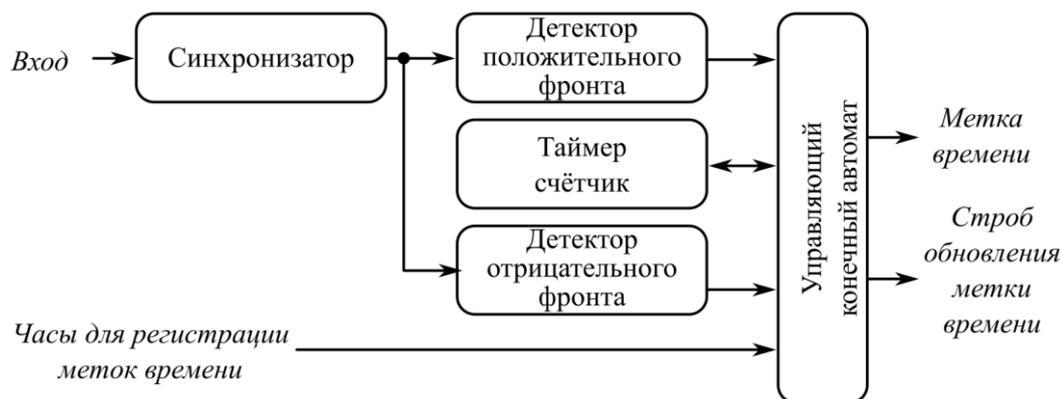


Рис. 7. Структура ядра ПЛИС для декодирования ультра-коротких пакетов

Сигнал с выхода компаратора переводится в частотный домен тактового генератора ПЛИС при помощи синхронизатора и поступает на детекторы отрицательного и положительного фронтов, которые подключены к управляющему конечному автомату. Данный конечный автомат имеет два состояния: ожидания и измерения. Если в режиме ожидания регистрируется отрицательный фронт входного сигнала, конечный автомат сохраняет метку времени прихода фронта по часам, поступающим на вход модуля, запускает таймер счётчик и переходит в режим измерения.

В режиме измерения происходит ожидание положительного фронта сигнала на выходе компаратора. Как только такой сигнал регистрируется, таймер-счётчик

останавливается, а его значение сравнивается с верхней и нижней границами допустимой длины УКП. Если измеренное значение удовлетворяет условию, то метка времени на выходе ядра изменяется на зарегистрированную в момент детектирования отрицательного фронта сигнала компаратора, и формируется строб, информирующий другие ядра ПЛИС о приходе УКП. Вне зависимости от результата измерения длительности передаваемого пакета по положительному фронту сигнала с компаратора, управляющий конечный автомат снова переходит в режим ожидания.

Ресурсоёмкость разработанного ядра ПЛИС при синтезе для микросхем разных производителей представлена в табл. 1.

Таблица 1. Ресурсоёмкость разработанного ядра ПЛИС для детектирования и простановки меток времени ультра-коротких пакетов

| Семейство микросхем | Ресурсоёмкость | Максимальная частота |
|-----------------------------|----------------|----------------------|
| Xilinx Artyx-7 | 102 LUT6 | 231.72 МГц |
| Altera (Intel) Cyclone IV E | 173 LUT4 | 222.67 МГц |
| Lattice ICE-40 | 307 LUT4 | 65.42 МГц |

Низкие требования предложенного решения к ресурсам ПЛИС позволяют реализовывать его, в том числе, с использованием ультракомпактных бюджетных микросхем Lattice ICE40, чьи минимальные габариты составляют 1.40x1.48x0.45 мм, а розничная стоимость за 1 микросхему на портале Digi-Key не превышает 3\$ по состоянию на сентябрь 2019 г.

5. Экспериментальные исследования

Для проведения экспериментальных исследований был разработан стенд (рис. 8, а), включающий в себя одноплатный компьютер Raspberry Pi 3B v1.2, оснащенный ИМБС Cypress CYW43438, плату согласования на базе

быстродействующего компаратора Analog Devices AD8611ARZ и отладочную плату Digilent Nexys 4 DDR на базе ПЛИС Xilinx Artyx-7. Подключение платы компаратора к входу радиоприёмника CYW43438 осуществлялось путем пайки в точке, указанной на рис. 8, б.

Связь между ПЛИС и Raspberry Pi 3b осуществлялась по интерфейсу Ethernet и протоколу UDP. Такой способ взаимодействия был выбран исходя из простоты его организации в рамках оборудования, имеющегося в наличии. Предложенный в работе метод синхронизации не предъявляет каких-либо существенных требований к пропускной способности или детерминизму интерфейса между ПЛИС и центральным процессором бортовой системы управления, что позволяет в дальнейшем заменить Ethernet на более дешевые в реализации I²C, SPI или UART.

Алгоритм синхронизации и все необходимые для него вычисления (1-4) были реализованы в среде Octave, запущенной на Raspberry Pi 3b. В качестве операционной системы встраиваемого компьютера был использован стандартный дистрибутив Rasbian в версии от 2018-11-13 с ядром 4.14.79-v7+ без использования патчей реального времени. Для передачи управляющих команд и чтения меток времени, зарегистрированных ИМБС, использовалась утилита nexutil из инструментария nexmon [16]. Для синхронизации программного обеспечения Raspberry Pi 3b ПЛИС формировала дискретный сигнал на одном из его входов, по которому на уровне ядра операционной системы был настроен обработчик прерывания.

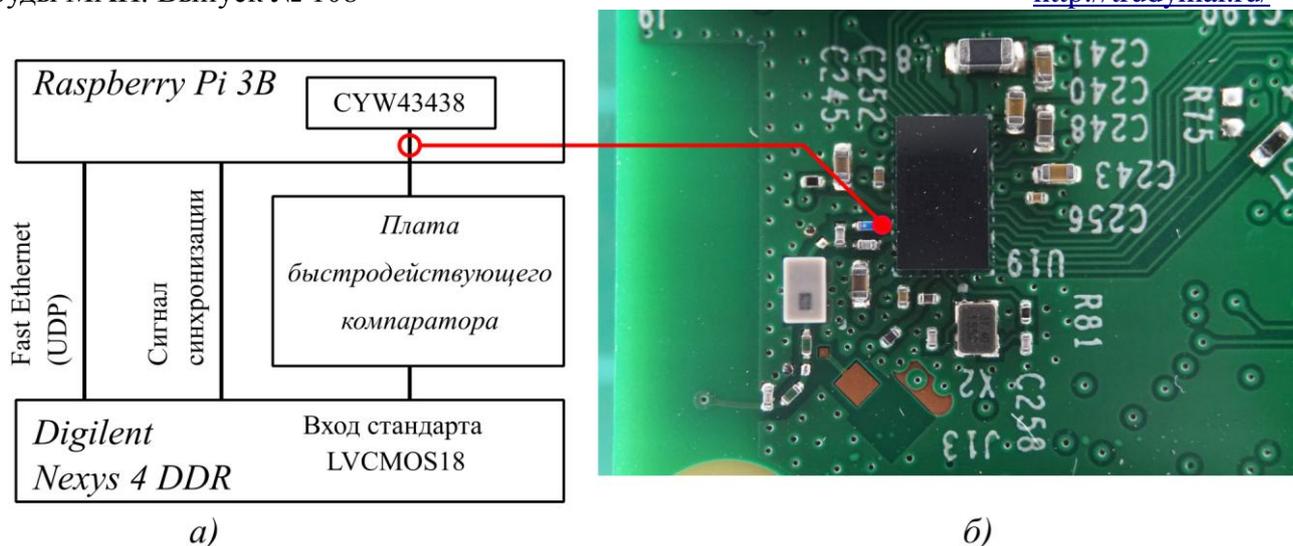


Рис. 8. а) Структурная схема экспериментального стенда. б) Точка подключения компаратора к входу радиоприёмника CYW43438 на плате Raspberry Pi 3B

Для реализации всех компонент аппаратно-программного обеспечения ПЛИС, использованных в ходе экспериментов, включая ядра для аппаратной реализации протокола UDP, из всех ресурсов ПЛИС было использовано только 936 LUT6 и 1 блок управления тактовыми сигналами. Модули блочной памяти и аппаратные умножители не использовались. Таким образом, можно констатировать, что ресурсоёмкость предложенных решений на несколько порядков меньше, чем требуется для реализации РНУ и MAC на базе ПЛИС [13, 14], которые необходимы в случае решения поставленной задачи методами, предполагающими полное декодирование входящих пакетов на базе ПЛИС.

В рамках эксперимента было собрано два идентичных стенда (рис. 8), которые были подключены к общей точке доступа. Для каждого из стендов была получена экспериментальная зависимость $T_d(T_f)$ по которой, методом наименьших квадратов, было определено значение k , которое составило 0.02000003524 и 0.02000019317 для первого и второго стендов соответственно.

С использованием полученных значений k были произведены эксперименты по синхронизации часов ИМБС и ПЛИС длительностью 2500 циклов. Общая продолжительность эксперимента составила 2334 с (38 мин 54 сек).

На рис. 9 представлено изменение ошибки синхронизации ε в ходе эксперимента. Как показал анализ результатов эксперимента для каждого из стендов, среднеквадратическая ошибка синхронизации составила менее 1 мкс а абсолютная ошибка синхронизации не превышала 6 мкс.

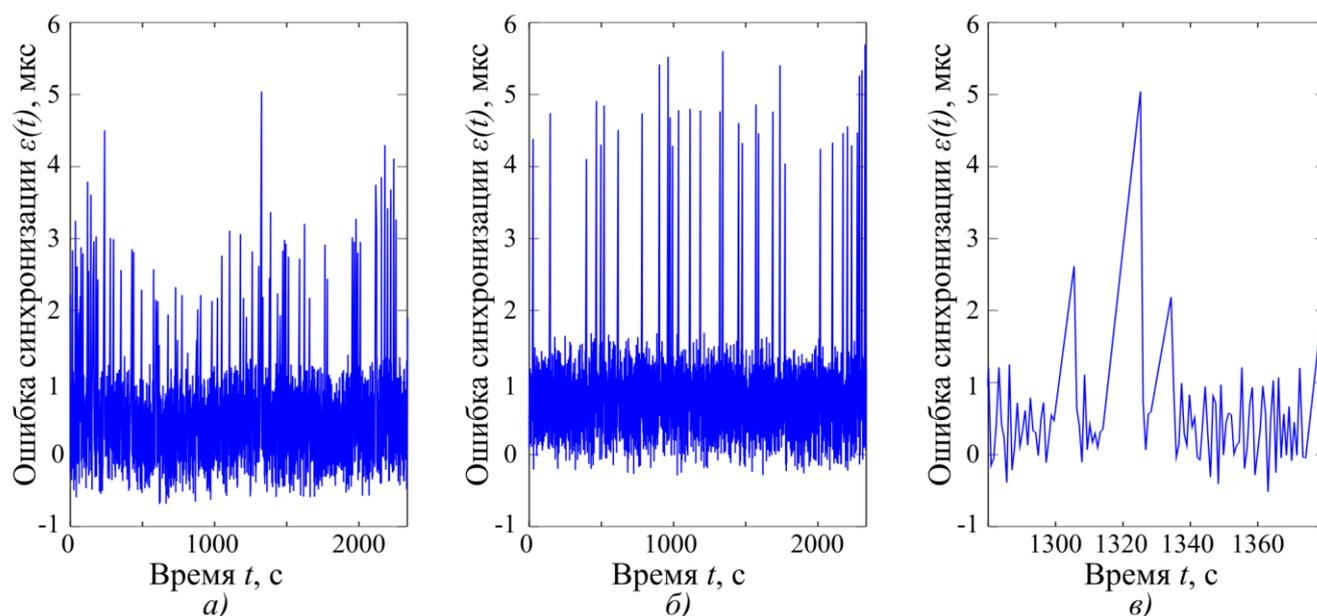


Рис. 9. Ошибка синхронизации $\varepsilon(t)$ в ходе эксперимента: а) за все время эксперимента для стенда 1; б) за все время эксперимента для стенда 2; в) в момент пика $\varepsilon(t)$ для стенда 1

В то же время, при более пристальном рассмотрении моментов времени, где ошибка синхронизации была максимальной (рис. 9, в), можно заметить, что они попадают на те моменты, когда программному обеспечению D11 не удалось сформировать корректный УКП, что приводило к существенному увеличению времени между моментами синхронизации. Это говорит в пользу того, что данная ошибка была вызвана не случайными задержками, а расхождением частот тактовых

генераторов ИМБС и ПЛИС, которые не удалось компенсировать из-за неточностей в вычислении k .

На рис.10 представлена осциллограмма, демонстрирующая фактический джиттер взаимной синхронизации двух бортовых вычислительных устройств между собой. Данная осциллограмма отображает сигналы синхронизации Raspberry Pi 3В, сгенерированные ПЛИС на каждом из стендов в момент положительного фронта сигнала синхронизации первого стенда. Как можно увидеть, фактическая ошибка взаимного синхронизации не превышает 10 мкс.

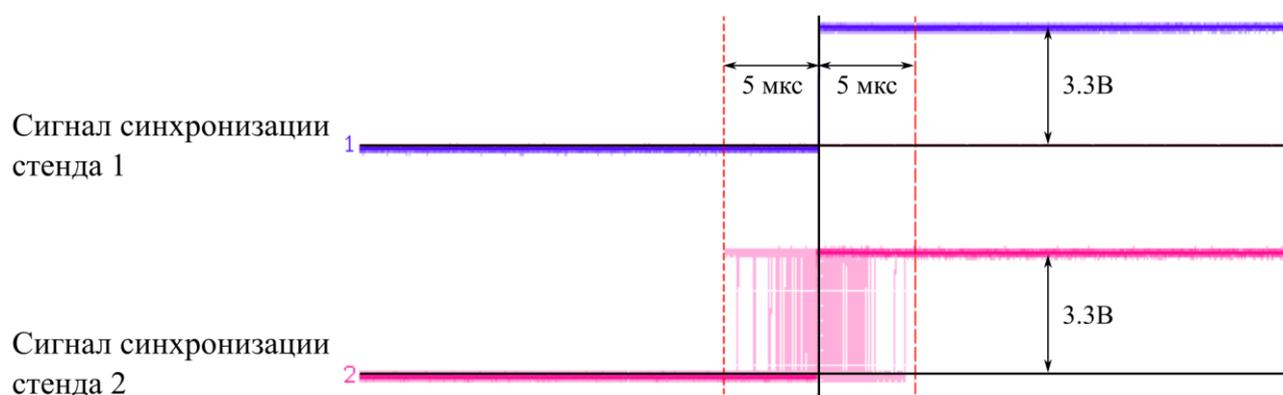


Рис. 10. Осциллограмма сигналов синхронизации Raspberry Pi 3В, сгенерированных ПЛИС на каждом из стендов

Заключение

Будущее применения МБПЛА несомненно заключается в развитии технологий группового управления. Взаимная синхронизация бортовых устройств управления различных МБПЛА является ключевой составляющей этих технологий. В данной работе предлагается новый подход к синхронизации, который основан на анализе потенциала входа радиоприемника интегрированной микросхемы WiFi, реализованной по технологии система-на-кристалле. Преимуществом данного подхода является отсутствие необходимости в демодуляции и декодировании

принимаемых пакетов, что позволяет реализовывать его с использованием малогабаритных бюджетных компонент. В работе, на примере микросхемы Cypress CYW43438, описываются схемотехнические и программные решения, необходимые для решения задачи беспроводной синхронизации бортовых вычислительных устройств МБПЛА. Результаты проведенных экспериментальных исследований демонстрируют ошибку взаимной синхронизации двух бортовых вычислительных устройств менее 10 мкс.

Оценивая полученную точность синхронизации с практической точки зрения, можно сделать вывод, что она достаточна для большинства прикладных задач в области группового управления МБПЛА, хоть и не является рекордной для решений на основе IEEE 802.11. В то же время, с точки зрения массогабаритных размеров, стоимости и доступности компонентов, предложенный в данной работе подход практически не имеет конкурентов.

Сведения об источнике финансовой поддержки работы

А.М. Романов выполнял работу при поддержке Министерства науки и высшего образования РФ и Немецкой службы академических обменов (DAAD) по программе «Михаил Ломоносов» (Personal ref. no: 91725531, Funding ID: 57447934, № 8.13409.2019/13.2).

Библиографический список

1. Добрынин Д.А. Применение малых БПЛА мультикоптерного типа для локального мониторинга объектов окружающей среды // Робототехника и техническая

кибернетика. 2014. № 1. С. 33 - 37.

2. Нелидина М.Ю., Ткачева О.А. Квадрокоптер - как способ получения пространственной информации в землеустройстве // Международная научно-практическая конференция «Результаты научных исследований»: сборник статей, (15 февраля 2016, Тюмень). – Уфа: Аэтерна, 2016. Ч. 3. С. 190 - 193.

3. Nedjati A., Vizvari B., Izbirak G. Post-earthquake response by small UAV helicopters // Natural Hazards, 2016, vol. 80, no. 3, pp. 1669 - 1688.

4. Дивеев А.И., Конырбаев Н.Б. Управление группой квадрокоптеров методом вариационного аналитического программирования // Труды МАИ. 2017. № 96. URL: <http://trudymai.ru/published.php?ID=85774>

5. Иванов Д.Я. Формирование строя группой беспилотных летательных аппаратов при решении задач мониторинга // Известия Южного федерального университета. Технические науки. 2012. Т. 129. № 4. С 219 – 224.

6. Lim Y.H. et al. Implementation of load transportation using multiple quadcopters // International Conference on Advanced Intelligent Mechatronics (AIM), IEEE, 2017, pp. 639 - 644. DOI: [10.1109/aim.2017.8014089](https://doi.org/10.1109/aim.2017.8014089)

7. Aghdam A.S., Menhaj M.B., Suratgar A. A. Non-rigid Load Transportation by Multiple Quadrotors // 6th RSI International Conference on Robotics and Mechatronics (ICRoM), IEEE, 2018, pp. 88 - 93. DOI: [10.1109/ICRoM.2018.8657570](https://doi.org/10.1109/ICRoM.2018.8657570)

8. Dinh H.T., Torres M.H.C., Holvoet T. Dancing UAVs: Using linear programming to model movement behavior with safety requirements // International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, 2017, pp. 326 - 335. DOI:

10.1109/ICUAS.2017.7991352

9. Ковальский А.А., Афонин Г.И., Терещенко С.В. Предложения по основным направлениям модернизации системы единого времени наземного автоматизированного комплекса управления космическими аппаратами // Труды МАИ. 2017. № 97. URL: <http://trudymai.ru/published.php?ID=87284>

10. Кишко Д.В. Синхронизация передающих устройств распределенных радиотехнических систем навигации и посадки летательного аппарата // Труды МАИ. 2015. № 82. URL: <http://trudymai.ru/published.php?ID=58802>

11. Mahmood A. et al. Clock synchronization over IEEE 802.11, A survey of methodologies and protocols // IEEE Transactions on Industrial Informatics, 2016, vol. 13, no. 2, pp. 907 – 922. DOI: [10.1109/TII.2016.2629669](https://doi.org/10.1109/TII.2016.2629669)

12. Kannisto J. et al. Software and hardware prototypes of the IEEE 1588 precision time protocol on wireless LAN // 14th IEEE Workshop on Local & Metropolitan Area Networks, IEEE, 2005, pp. 6. DOI: [10.1109/LANMAN.2005.1541513](https://doi.org/10.1109/LANMAN.2005.1541513)

13. Exel R. Clock synchronization in IEEE 802.11 wireless LANs using physical layer timestamps // International Symposium on Precision Clock Synchronization for Measurement, Control and Communication Proceedings, IEEE, 2012. pp. 1-6. DOI: [10.1109/ISPCS.2012.6336622](https://doi.org/10.1109/ISPCS.2012.6336622)

14. Ferrari P. et al. Experimental characterization of uncertainty sources in a software-only synchronization system // Transactions on Instrumentation and Measurement, IEEE, 2012, vol. 61, no. 5, pp. 1512 - 1521. DOI: [10.1109/TIM.2011.2180974](https://doi.org/10.1109/TIM.2011.2180974)

15. Schulz M. et al. Shadow Wi-Fi: Teaching Smartphones to Transmit Raw Signals and to

Extract Channel State Information to Implement Practical Covert Channels over Wi-Fi // Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, ACM, 2018, pp. 256 - 268.

16. Schulz M., Wegemer D., Hollick M. Using NexMon, the C-based WiFi firmware modification framework // Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks, ACM, 2016, pp. 213 - 215. DOI: [10.1145/2939918.2942419](https://doi.org/10.1145/2939918.2942419)

17. Gringoli F., Nava L. Openfwfwf: Open firmware for wifi networks, HTTP, 2010, available at: <http://netweb.ing.unibs.it/openfwfwf>

18. Gringoli F. et al. Experimental QoE evaluation of multicast video delivery over IEEE 802.11 aa WLANs // IEEE Transactions on Mobile Computing, 2018, pp. 99. DOI: 10.1109/TMC.2018.2876000

19. Agnes Fain, Wolfgang Meryk. Implementing Wireless LAN Interface in an FPGA // Xcell Journal, 2009. no. 69, available at: <https://www.xilinx.com/publications/archives/xcell/Xcell69.pdf>

20. Mango Communications, Inc., Mango Communications 802.11 Reference Design HTTP, 2019, available at: <http://mangocomm.com/802.11>