

Ethernet-мосты компании WIZnet: перспективные возможности

Сергей ДОЛГУШИН
dsa@efo.ru

Проводные сети Ethernet остаются популярными и востребованными, несмотря на бурное развитие беспроводных технологий, предлагаемых им на замену. Микросхемы компании Wiznet являются одним из самых простых в освоении и применении решений для проводного Ethernet. Уже более 10 лет компания выпускает аппаратные мосты Ethernet, аппаратный стек TCP/IP которых включает поддержку протоколов TCP, UDP, IPv4, ICMP, IGMP и PPPoE. Этого вполне достаточно для реализации типовых задач по удаленному сбору данных, мониторингу и управлению. Сегодня речь пойдет о перспективных микросхемах Wiznet и их основных возможностях, а также о базовых принципах работы с мостом W5500 на примере его подключения к микроконтроллеру Microchip (Atmel) SAM D21 (ARM Cortex M0).

Удаленный сбор данных, мониторинг, управление и другие аналогичные задачи становятся все более популярными. В частности, этому способствует активная реклама «Интернета вещей» (IoT). Очевидно, что в устройствах IoT наибольшее применение должны получить беспроводные решения, тем не менее проводные сети не утратят своих позиций в ближайшем будущем. Развитая инфраструктура позволяет просто и дешево обеспечить подключение любых устройств (счетчиков расхода ресурсов, контрольно-кассового оборудования) к локальным сетям и Интернету. Кроме того, в случае проводного соединения можно не беспокоиться об устойчивости канала обмена и возможных сбоях в передаче из-за воздействия внешних условий, что не всегда может быть обеспечено при использовании беспроводных подключений. Пять лет назад встроенная в микроконтроллеры поддержка Ethernet была редкостью, а сегодня многие производители предусматривают такую возможность в своих микросхемах.

Компания Wiznet специализируется на разработке и выпуске микросхем для Ethernet-приложений, а также коммуникационных модулей на их базе. Основной особенностью микросхем Wiznet является аппаратная реализация стека TCP/IP. На первый взгляд, интегрированное решение выглядит предпочтительнее, чем использование внешних микросхем Wiznet. Но нет, здесь и сказывается наличие аппаратной поддержки TCP/IP. В большинстве существующих решений аппаратно выполняется только MAC-уровень, стек TCP/IP реализуется программными средствами. В частности, это объясняет причины, по-

чему поддержка интерфейса Ethernet добавляется в микроконтроллеры с ядром не младше Cortex M4. Это оставляет решениям от Wiznet достаточно большую нишу для применения. Например, в модернизируемых устройствах, где не требуется увеличение производительности микроконтроллера (МК) и только необходимо добавить сетевой интерфейс.

Итак, теперь расскажем о том, какие аппаратные мосты от Wiznet можно рассматривать в качестве перспективы для новых проектов. На текущий момент имеет смысл выбирать между мостами W5500 и W5300, о них и пойдет речь.

В общих чертах, все Ethernet-мосты Wiznet, выпускаемые в настоящее время, имеют общую внутреннюю структуру (рис. 1, 2). Основой моста служит аппаратный блок Ethernet, куда входит MAC-уровень и чей аппаратный стек TCP/IP включает поддержку протоколов TCP, UDP, IPv4, ICMP, IGMP и PPPoE. Встроенный физический уровень PHY W5500 и W5300 обеспечивает возможность работы с сетями Ethernet 10BaseT/100BaseTX. Отличительной чертой встроенного PHY W5300 является поддержка функции auto-MDIX (определение типа кабеля — прямой или перекрестный), физический уровень W5500 эту функцию не поддерживает. Из внешних компонентов с этой стороны требуются только трансформатор и разъем. Микросхема W5300, кроме встроенного PHY, имеет дополнительный внешний интерфейс MII (media independent interface) для подсоединения внешней микросхемы физического уровня. Это полезно, если есть необходимость подключения к сети, например, по оптической линии.

Управление мостами, а также прием и передача данных осуществляются по интерфейсу SPI в случае W5500 или по параллельному 8/16-разрядному интерфейсу у W5300. Максимальная рабочая частота SPI W5500 установлена равной 80 МГц. В реальной ситуации можно ориентироваться на следующие показатели пропускной способности в зависимости от частоты SPI: при 24 МГц она достигает 8 Мбит/с, при 40 МГц — 16 Мбит/с при работе с одним сокетом на прием и передачу (показатели приводятся в качестве оценки возможности по информации от пользователей). В случае W5300, при использовании 16-разрядной шины данных, пропускная способность будет ограничена пропускной способностью самой сети 100 BaseTx. Режимы работы микросхем, установка их в то или иное состояние осуществляются через регистры управления.

Для приема и передачи данных выделена специальная область памяти, объем которой у W5500 равен 32 кбайт, у W5300 — 128 кбайт. Общий объем памяти распределяется поровну, на прием и передачу. Обе микросхемы поддерживают одновременную работу с восемью сокетами. В зависимости от задачи и количества активных сокетов последним можно назначить различный рабочий объем памяти на прием и передачу. Для W5500 сокету резервируется объем 1, 2, 4, 8 или 16 кбайт; по умолчанию каждому из них присвоено по 2 кбайт памяти на прием и передачу. Для W5300 каждому сокету выделяется область памяти в диапазоне 0–64 кбайт, на прием и передачу. При распределении памяти между сокетами основным условием является кратность восьми общей

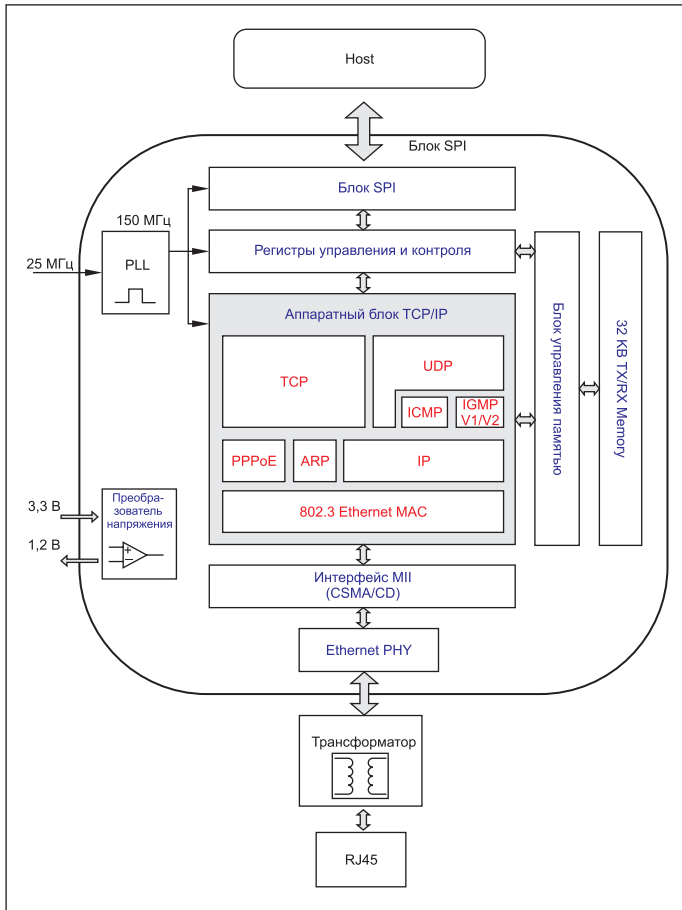


Рис. 1. Структурная блок-схема моста W5500

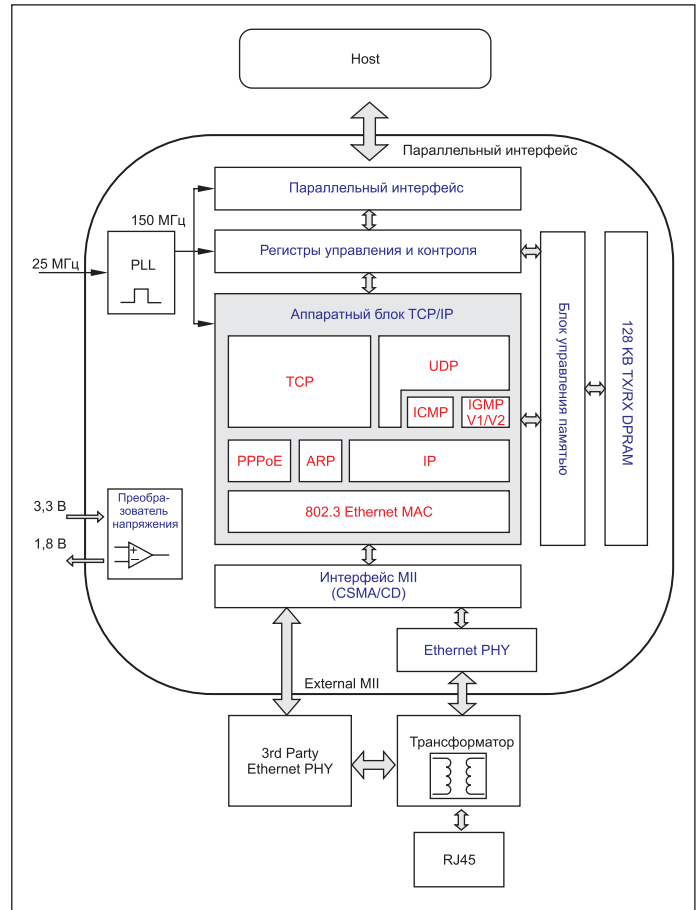


Рис. 2. Структурная блок-схема моста W5300

выделенной памяти. По умолчанию каждый имеет 8 кбайт памяти на прием и передачу.

Рабочие характеристики микросхем следующие. Номинальное напряжение питания W5300 и W5500 составляет 3,3 В. Обе микросхемы могут работать с внешними устройствами с выходами 3,3 и 5 В. Рабочий температурный диапазон лежит в пределах $-40...+85$ °С.

Типовое энергопотребление W5300 составляет 175 мА при ручном выборе типа сети и 180 мА при включении функции автоматического определения сети блоком PHY, или 65 мА при использовании внешней микросхемы PHY. Специальных режимов по снижению энергопотребления в W5300 нет.

W5500 имеет максимальное потребление в режиме передачи по сети 100 BaseTx, и оно может достигать 132 мА. Есть режим Power Down, при котором выключается блок PHY, типовое потребление в этом случае равно 13 мА. Перевод микросхемы в режим Power Down и выход из него осуществляется программным путем.

На базе этих микросхем Wiznet предлагает несколько мезонинных модулей, их можно использовать на этапе макетирования и/или в мелкосерийных проектах. Далее кратко расскажем о самых популярных из них.

Мезонинный модуль WIZ830MJ (рис. 3) реализован на базе моста W5300. Модуль со-

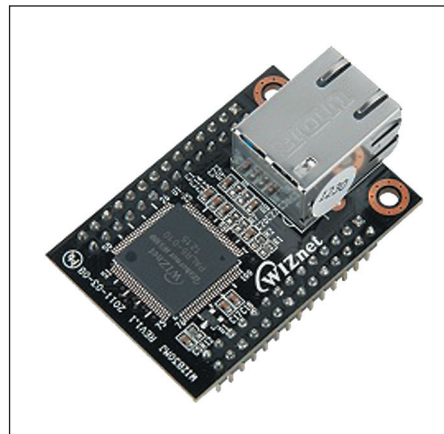


Рис. 3. Модуль WIZ830MJ

держит все необходимые компоненты для работы моста и Ethernet-разъем со встроенным трансформатором. Все управляющие действиям W5300 линии, 16 разрядов шины данных, а также питание выведены на два двухрядных штыревых разъема, шаг выводов разъемов составляет 2,54 мм.

Миниатюрный мезонинный модуль WIZ850io (рис. 4) на базе W5500 — самое недорогое готовое решение для реализации Ethernet-соединения. Линии SPI, сброс и питание выведены на два однорядных штыревых разъема (рис. 5), шаг выводов — 2,54 мм.

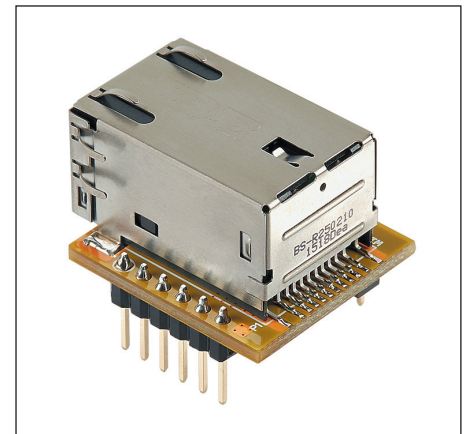


Рис. 4. Модуль WIZ850io

Модуль полностью совместим по выводам с модулем WIZ820io, реализованным на предшественнице W5500 — микросхеме W5200, не рекомендованной к использованию в новых разработках.

Еще один популярный модуль на микросхеме W5500 — это WIZ550io (рис. 6). Основным отличием его от WIZ850io является дополнительный микроконтроллер, обеспечивающий запуск моста W5500 в рабочий режим после подачи питания на модуль. В памяти микроконтроллера «предпрошиты» сетевые настройки, включая уникальный

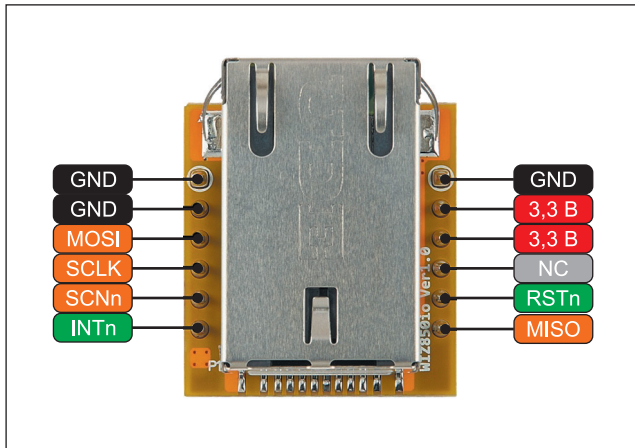


Рис. 5. Назначение выводов модуля Wiz850io

Таблица. Подключение платы XPlained Pro ATSAMD21-XPRO к модулю Wiz850io

	Vcc	Gnd	SCLK	MISO	MOSI	CS	RST
Wiz850io	3V3	GND	SCLK	MISO	MOSI	SCn	RSTn
ATSAMD21-XPRO	PWR2	PWR4	Ext1 18	Ext1 17	Ext1 16	Ext1 15	Ext1 7

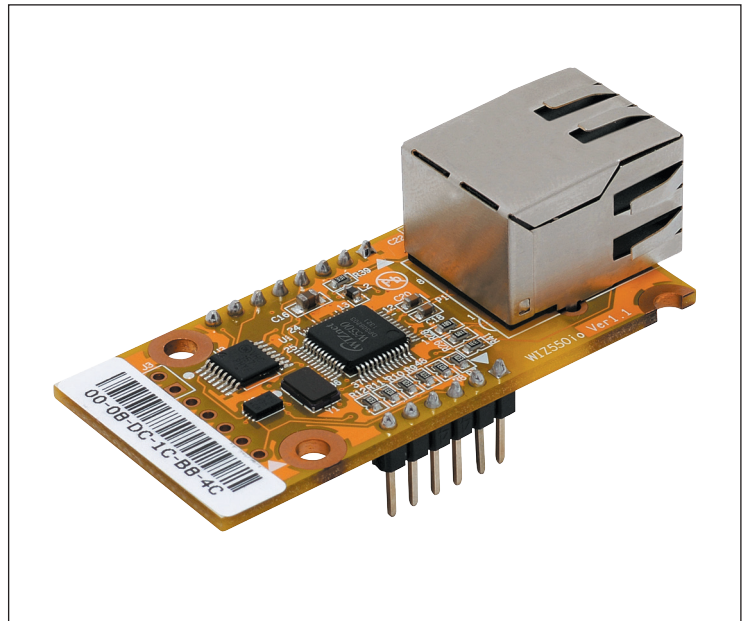


Рис. 6. Модуль WIZ550io

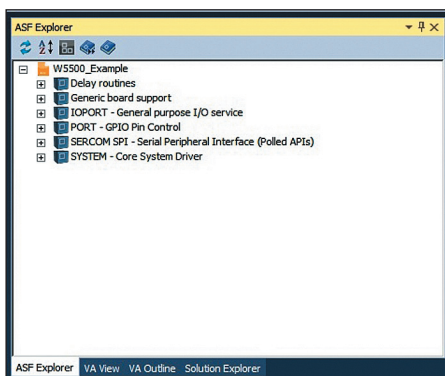


Рис. 7. Драйверы и функции, добавленные в проект через ASF

MAC-адрес, после загрузки которых в W5500 последний становится доступен по сети.

Базовые принципы работы с микросхемой W5500 покажем на примере реализации организации обмена между МК и ПК. В качестве МК будет использоваться ARM Cortex-M0 SAM D21 Microchip (Atmel), а Ethernet-мост W5500 служит для организации сетевого соединения. Для примера возьмем отладочную плату Microchip (Atmel) XPlained Pro ATSAMD21-XPRO и модуль Wiznet Wiz850io. В таблице показан порядок подключения выводов модуля и отладочной платы. В качестве среды разработки для микроконтроллера (МК) SAM D21 воспользуемся Atmel Studio 7 и Atmel Software Framework (ASF).

Создание шаблона проекта в ASF полностью аналогично тому, как это описано в статье [1]. Для работы с W5500 понадобятся те же самые периферийные модули и библиотеки, что были использованы в примере на рис. 7. Создав шаблон проекта, добавляем в него файлы библиотеки Wiznet ioLibrary

для работы с мостами [2]. Для адаптации библиотеки Wiznet для SAM D21 необходимо изменить функции чтения/записи (файл *W5500.c*), функцию установки линий SPI в активное/неактивное состояние CS (*wizchip_conf.c*) и управление сбросом RST (реализовано в *main.c*). Подробно на этом останавливаться не будем, итоговую реализацию можно посмотреть в проекте, который доступен в исходном виде по ссылке [3].

Работу с Ethernet-мостом начинаем с подачи сигнала сброса на линию RST:

```
ioport_set_pin_level(RST_Pin, 0);
delay_us(500);
ioport_set_pin_level(RST_Pin, 1);

delay_ms(1);
```

Длительность импульса сброса должна быть 500 мкс. Затем в течение 1 мс происходит установка ФАПЧ в рабочий режим. После завершения данного этапа W5500 готов для инициализации.

В начальную конфигурацию входят установка значения приемных и передающих буферов и назначение сетевых параметров:

```
// размеры всех буферов равны 2 кбайтам
uint8_t txsize[8] = {2,2,2,2,2,2,2,2};
uint8_t rxsize[8] = {2,2,2,2,2,2,2,2};
wiz_NetInfo gWIZNETINFO = { .mac = {0x00, 0x08, 0xdc, 0x00, 0xab, 0xcd}, // MAC-адрес
    .ip = {192, 168, 22, 3}, // IP
    .sn = {255, 255, 255, 0}, // Маска подсети
    .gw = {192, 168, 22, 1}, // Основной шлюз
    .dns = {0,0,0,0}, // в примере не используется
    .dhcp = NETINFO_STATIC; // в примере не используется
    wizchip_init(txsize, rxsize); // устанавливаем размеры буферов
    network_init(); // загружаем сетевые настройки // в регистры GAR, SIPR, SUBR // и SHAR
```

После назначения сетевых параметров W5500 будет доступен по сети, что можно проверить командой ping 192.168.22.3.

Основной рабочий цикл, в самом простом виде, будет выглядеть следующим образом:

```
switch(getSn_SR(sn)) // sn — номер сокета
{
    case SOCK_CLOSED:
        if((ret = socket(sn, Sn_MR_TCP, port, 0x00)) != sn) return ret;
        // открываем сокет SN в режиме TCP, с номером порта port
        // и флагом для W5500: SF_BROAD_BLOCK,
        SF_MULTI_BLOCK, SF_IPv6_BLOCK, и/или SF_UNI_BLOCK
        break;
        default:
        break;

    case SOCK_INIT:
        if((ret = listen(sn)) != SOCK_OK) return ret;
        break;

    case SOCK_ESTABLISHED:
        if((size = getSn_RX_RSR(sn)) > 0)
        {
            if(size > DATA_BUF_SIZE) size = DATA_BUF_SIZE;
            ret = recv(sn, buf, size);

            if(ret <= 0) return ret;
            sentsize = 0;

            while(size != sentsize)
            {
                ret = send(sn, buf, size);
                if(ret < 0)
                {
                    close(sn);
                    return ret;
                }
                sentsize += ret;
            }
        }
        break;

    case SOCK_CLOSE_WAIT:
        if((ret = disconnect(sn)) != SOCK_OK) return ret;
        break;
}
return 1;
}
```

В цикле постоянно проверяем регистр состояния сокетов Sn_Sr. Изменения его статуса происходят при изменении в командном регистре Sn_Cr W5500 или при приеме специальных пакетов SYN или FIN.

В первоначальном состоянии сокет закрыт и имеет статус SOCK_CLOSED. Получив

этот статус, можно открыть нужный нам сокет функцией `socket()` в режиме TCP.

Успешные инициализация и открытие сокета будут подтверждены сменой состояния регистра `Sn_Sr`, который будет `SOCK_INIT`. Функцией `listen()` переводим W5500 в режим ожидания запроса на установление соединения от удаленного клиента.

При получении такого запроса W5500 перейдет в состояние `SOCK_ESTABLISHED`, что говорит об установлении соединения с клиентом. Пока мост находится в этом состоянии, он может принимать и передавать данные по сети. В описанном примере мы принимаем данные от клиента и передаем их обратно. Наличие и количество данных проверяем чтением регистра `Sn_Rx_RSR`, принимаем их функцией `recv()` и отправляем обратно функцией `send()`.

При получении запроса FIN от клиента на закрытие сокета W5500 переходит в состояние `SOCK_WAIT_CLOSE`. Здесь можно принимать и передавать данные. Функцией `disconnect()` передаем в командный регистр

команду `DISCON`. При использовании этой команды W5500 подтверждает закрытие сокета отправкой пакета FIN клиенту. Регистр состояния меняет статус на `SOCK_CLOSED`. Закрыть сокет также можно командой `CLOSE`, причем W5500 не передает никаких подтверждающих пакетов клиенту.

Этот простой пример демонстрирует базовые возможности приема и передачи данных по сети в режиме TCP-сервер. С его помощью можно быстро протестировать возможности Ethernet-моста и понять, насколько он удовлетворяет поставленным задачам. Полный проект для SAM D21 приведен по ссылке [3].

Несмотря на то, что сегодня многие производители предлагают Ethernet в составе собственных контроллеров, аппаратные решения от Wiznet не теряют своей привлекательности. В первую очередь это касается небольших проектов и/или когда необходимо получить готовое и работоспособное решение в очень короткий срок. Приведенный пример для МК SAM D21 показывает, как просто и быстро можно реализовать сетевое соединение с по-

мощью Ethernet-моста W5500. Кроме того, в интегрированных решениях мы привязаны к тому набору периферии, который дает производитель. В случае связки «МК + Ethernet-мост» подобрать МК с нужными интерфейсами гораздо проще. Так, в задачах автоматизации и управления может потребоваться большое количество последовательных каналов передачи, например RS-232/422/485, связка нового МК Atmel (Microchip) SAM G55 и W5500 позволит легко реализовать сетевой конвертер Ethernet на семь последовательных каналов. ■

Литература

1. Долгушин С. Графический контроллер EVE FT800 FTDI и микроконтроллер SAMD21 Atmel. Работаем с графическими изображениями // Компоненты и технологии. 2014. № 8.
2. [www.github.com/Wiznet/ioLibrary_Driver](https://github.com/Wiznet/ioLibrary_Driver)
3. www.mymcu.ru/articles/realizatsiya-ethernet-c-pomoshchyu-mosta-wiznet-w5500-i-mikrokontrollera-microchip-atmel-sam-d21.html