

УДК 62-529

С.М. Лісовець, к.т.н., доц.; І.Л. Ківа, к.т.н., доц.

ЗАСТОСУВАННЯ ETHERNET-КОНТРОЛЕРА ENC28J60 З ІНТЕРФЕЙСОМ SPI ПРИ ПОВБУДОВІ МАЛИХ І СЕРЕДНІХ СИСТЕМ АВТОМАТИЗАЦІЇ

Київський національний університет технологій та дизайну, м. Київ, ser.lis.290171@gmail.com

Проаналізовано роботу Ethernet-контролера ENC28J60 з інтерфейсом SPI. Показано, що застосування контролера такого типу або аналогічних йому контролерів дозволяє здійснювати швидкий і надійний обмін даними як між рівнями АСКТП і АСКП, так і всередині цих рівнів, причому вартість як апаратної, так і програмної реалізації такого обміну даними буде досить низькою.

Ключові слова: комунікаційний інтерфейс, пакет, реєстр, система керування, технологічна операція, технологічний процес.

Вступ. Сучасні системи автоматичного керування передбачають, з одного боку, повний контроль за ходом виконання певної технологічної операції або певного технологічного процесу (це рівень автоматизованих систем керування технологічними процесами – АСКТП). З іншого боку, вони передбачають двосторонній обмін даними із вищестоячими засобами контролю (це рівень автоматизованих систем керування підприємством – АСКП).

Для здійснення такого обміну (між рівнями АСКТП і АСКП) існують різні промислові комунікаційні інтерфейси – зокрема, RS-485, CAN, Ethernet і деякі інші. В теперішній час, зокрема, все більш широке застосування має комунікаційний інтерфейс Ethernet – такий інтерфейс має кілька суттєвих переваг.

По-перше, практично будь-яка сучасна операційна система має підтримку комунікаційного інтерфейсу Ethernet і відповідні API для роботи з ним. По-друге, доступ до Ethernet-пристрою можна здійснювати практично з будь-якого місця Ethernet-мережі. По-третє, з точки зору промислової автоматизації мінімальної швидкості Ethernet-з'єднання 10 Мбіт/с досить для керування практично будь-якою технологічною операцією або технологічним процесом. В-четвертих, логіка роботи UDP/IP- або TCP/IP-стеку є досить чіткою і зрозумілою.

Постановка завдання. Полягає в дослідженні різних режимів роботи Ethernet-контролера ENC28J60 (див. рис. 1), які повністю підтримують стандарт IEEE 802.3, дозволяють вирішувати колізії, дозволяють задавати напівдуплексний або повнодуплексний режим обміну даними, дозволяють відкидати помилкові пакети тощо.

Контролер ENC28J60 підтримує інтерфейс Ethernet стандарту 10BASE-T і, відповідно до цього, швидкість обміну даними 10 Мбіт/с. Також контролер ENC28J60 підтримує фізичний (PHY) і каналний (MAC) рівні моделі OSI, а його підключення до мережі Ethernet зазвичай здійснюється за допомогою рознімання RJ-45. Вся пам'ять в контролері ENC28J60 поділяється на реєстри для керування фізичним рівнем моделі OSI, реєстри керування і буфер для даних.

Для обміну даними з контролером ENC28J60 зазвичай використовується мікроконтролер, який має інтерфейс SPI (загалом 4 провідника), а швидкість обміну даними може досягати 10 Мбіт/с. Контролер ENC28J60 підтримує режим роботи 0 інтерфейсу SPI, який полягає в тому, що зчитування даних здійснюється при низькому рівні (лог. 0) тактового сигналу SCK ($CPOL = 0$) по задньому фронту тактового сигналу SCK ($CPHA = 0$).

Загалом контролер ENC28J60 підтримує 7 наборів інструкцій для обміну даними: зчитування керуючих реєстрів (Read Control Register), зчитування буферної пам'яті (Read Buffer Memory), запис керуючих реєстрів (Write Control Register), запис буферної пам'яті (Write Buffer Memory), встановлення бітового поля (Bit Field Set), очищення бітового поля (Bit Field Clear) і програмного скидання (System Command (Soft Reset)).

Перед початком обміну даними (як на зчитування, так і на запис) на вхід \overline{CS} подається низький рівень (лог. 0). Після цього подається перший байт, який складається з поля Opcode (3 біта), яке задає певний набір інструкцій, і з поля Argument (5 біт), яке задає за необхідності потрібний керуючий реєстр. Після цього за потреби зчитуються або записуються безпосередньо байти даних Data.

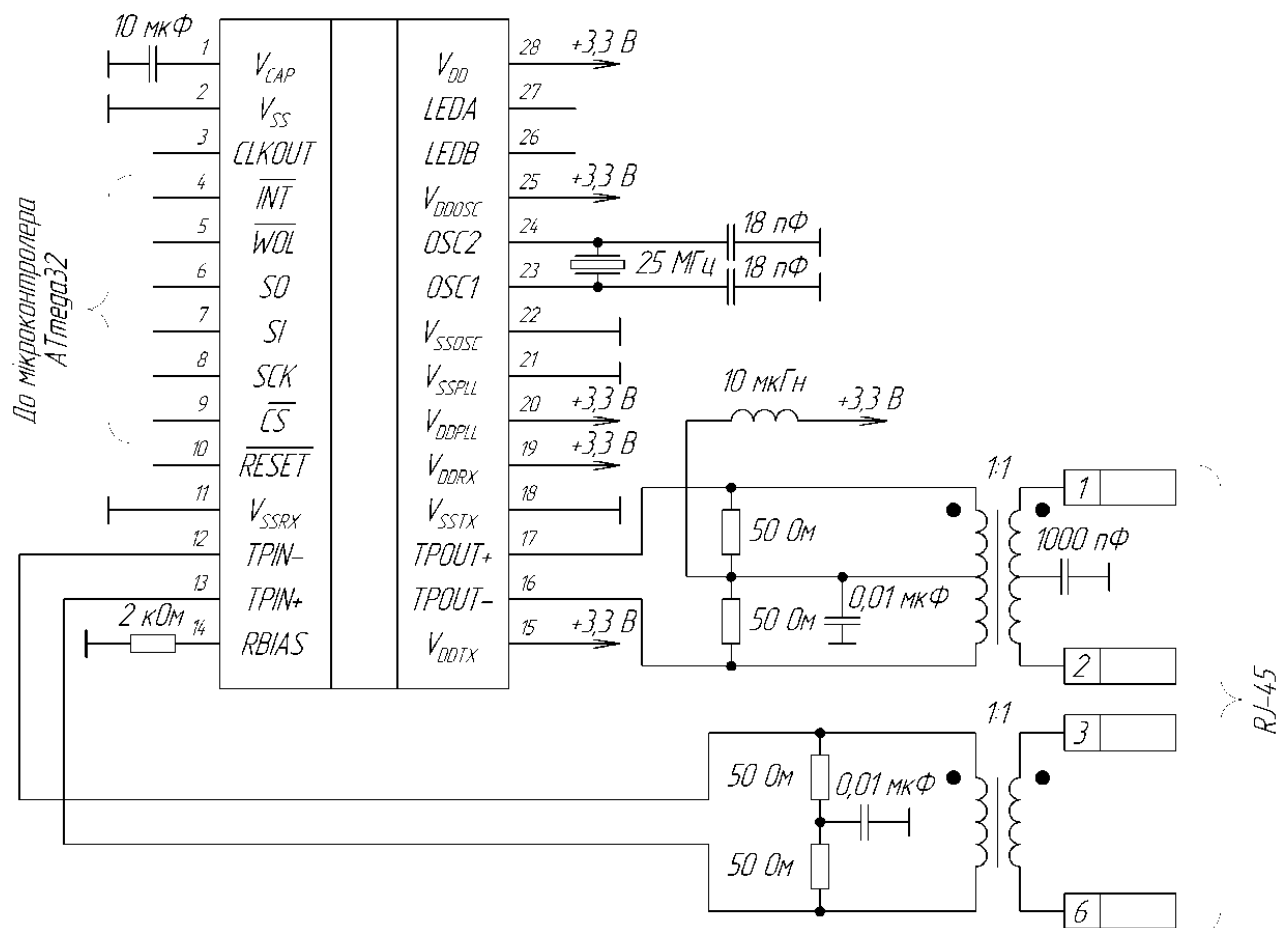


Рис. 1. Підключення Ethernet-контролера ENC28J60

Вся пам'ять Ethernet-контролера ENC28J60 інтерпретується як статична, причому вона ділиться на три умовні частини: пам'ять керуючих регістрів, Ethernet-буфер і пам'ять фізичних регістрів. Доступ до пам'яті керуючих регістрів можна отримати безпосередньо через інтерфейс SPI, а доступ до пам'яті фізичних регістрів можливий тільки через МІІ (Medium Independent Interface).

Загалом використовується 103 керуючих регістра з адресами від 00_{16} до 19_{16} і від $1B_{16}$ до $1F_{16}$, які розміщені в чотирьох банках з адресами від 00_{16} до 03_{16} , причому регістри з адресами від $1B_{16}$ до $1F_{16}$ співпадають між собою в будь-якому з банків.

Ethernet-буфер має розмір 8192 байта. Він використовується як для передачі, так і для прийому даних (див. рис. 2). Причому розмір областей пам'яті в цьому буфері, які використовуються відповідно для передачі і прийому даних, може бути програмно змінений – головне, щоб ці області між собою не перекривалися.

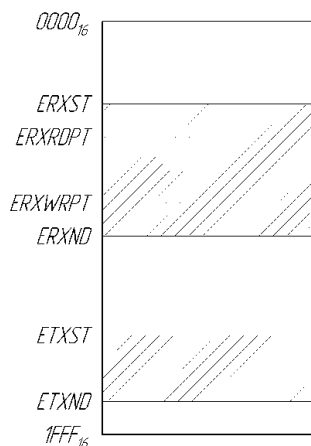


Рис. 2. Загальний устрій Ethernet-буфера

Зокрема, реєстр ETXST задає початок пакета, який буде відправлятися. А реєстр ETXND – кінець цього пакета. Наприклад, якщо необхідно відправити пакет розміром 256 байт, то в реєстр ETXST можна записати $1A00_{16}$, а в реєстр ETXND – $1AFF_{16}$. Аналогічно наведеному вище, реєстр ERXST задає початок кільцевого буфера, де буде розміщений пакет, який приймається. А реєстр ERXND – кінець цього буфера. Так як розмір пакета, який буде прийматися, наперед невідомий, то в такому буфері може одночасно знаходитися кілька пакетів.

Реєстр EWRPT (на рис. 2 не показаний) задає адресу в буфері, з якої дані будуть записуватися в буфер мікроконтролером. А реєстр ERDPT (на рис. 2 також не показаний) задає адресу в буфері, з якої дані будуть зчитуватися з буфера мікроконтролером. Крім того, реєстр ERXWRPT задає адресу в буфері, з якої буде розміщений новий прийнятий пакет. А реєстр ERXRPT задає адресу в буфері першого незчитаного пакета.

Крім того, використовується 9 фізичних реєстрів з адресами 00_{16} , 01_{16} , 02_{16} , 03_{16} , 10_{16} , 11_{16} , 12_{16} , 13_{16} і 14_{16} . Всі вони мають розрядність 16 біт. Переважна більшість з цих реєстрів не потребує зміни свого вмісту, але певні налагодження є необхідними.

Розв’язок поставленого завдання. Для дослідження Ethernet-контролера ENC28J60 з інтерфейсом SPI була використана макетна плата AVRPLC16 v6 виробництва MikroElektronika D.O.O. (м. Белград, Сербія), основу якої складає мікроконтролер ATmega32 виробництва Atmel Corporation (див. рис. 3).



Рис. 3. Зовнішній вигляд макетної плати AVRPLC16 v6

Розробка і налагодження програмного коду для мікроконтролера ATmega32 здійснювалося в безкоштовному середовищі програмування Atmel Studio 7.0 на мові програмування C. А безпосередньо програмування мікроконтролера ATmega32 здійснювалося за допомогою вбудованого в макетну плату AVRPLC16 v6 програматора за допомогою розробленого MikroElektronika D.O.O. програмного забезпечення.

Причому при програмуванні Ethernet-контролера ENC28J60 особлива увага приділялася документу ENC28J60 Rev. B7 Silicon Errata, в якому описані апаратно-програмні помилки, які мають місце при роботі Ethernet-контролера ENC28J60. Тому для коректної роботи Ethernet-контролера ENC28J60 ці помилки треба було “обходити”, коригуючи програмний код.

Спрощена структура каналу обміну даними між макетною платою AVRPLC16 v6 із Ethernet-контролером ENC28J60 і персональним комп’ютером показана на рис. 4.

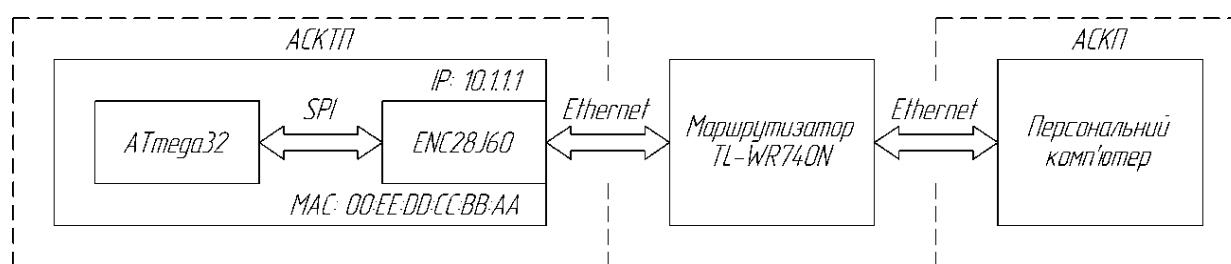


Рис. 4. Спрощена структура каналу обміну даними між макетною платою AVRPLC16 v6 із Ethernet-контролером ENC28J60 і персональним комп’ютером

Налагодження Ethernet-контролера ENC28J60 було виконано в наступній послідовності.

В реєстр ERXST було записано 0000_{16} , а в реєстр ERXND було записано $0FFF_{16}$ – таким чином, розмір буфера для прийому даних становив 4096 байт. В реєстр ERXRDPT також було записано 0000_{16} . В реєстр ETXST було записано 1000_{16} , а в реєстр ETXND було записано $1FFF_{16}$ – таким чином, розмір буфера для передачі даних також становив 4096 байт.

Для того, щоб приймати ширококомовні пакети і пакети на власну MAC-адресу, відповідно біти BCEN і UCEN реєстра ERXFCON були встановлені в лог. 1. А для того, щоб приймати пакети тільки з коректною контрольною сумою, біт CRCEN реєстра ERXFCON також було встановлено в лог. 1.

Для того, щоб дозволити прийом пакетів канальним рівнем (MAC), біт MARXEN реєстра MACON1 було встановлено в лог. 1. А для того, щоб дозволити апаратне керування прийомом і передачею кадрів, біти RXPAUS і TXPAUS реєстра MACON1 також було встановлено в лог. 1. Також для того, щоб дозволити роботу канального рівня (MAC), біт MARST реєстра MACON2 було скинуто в лог. 0.

Для того, щоб короткі пакети доповнювалися до довжини 60 байт і після цього до них додавалася коректна контрольна CRC-сума, біти PADCFG2 і PADCFG1 реєстра MACON3 було скинуто в лог. 0, а біт PADCFG0 реєстра MACON3 було встановлено в лог. 1. Причому, для додавання коректної контрольної CRC-суми також було встановлено біт TXCRCEN реєстра MACON3. Крім того, для роботи в повнодуплексному режимі біт FULDPX реєстра MACON3 було встановлено в лог. 1.

При цьому максимальний розмір кадра було встановлено таким, що дорівнював 1518 байт. Для цього в реєстр MAMXFLH було записано значення 05_{16} , а в реєстр MAMXFLL – значення EE_{16} .

Для того, щоб встановити період часу між кінцем попереднього і початком наступного з пакетів, біти BBIPG4, BBIPG2 і BBIPG0 реєстра MABBIPG було встановлено в лог. 1. Так як Ethernet-контролер ENC28J60 підтримує стандарт 10BASE-T (802.3i), то період часу між імпульсами при застосуванні манчестерського кодування (частота несучої 10 МГц) становив 0,1 мкс. Таким чином, період часу між пакетами становив

$$(2^4 + 2^2 + 2^0) \cdot 4 \cdot 0,1 \text{ мкс} = 9,6 \text{ мкс}.$$

Крім того, в реєстр MAIPGH було записане значення $0C_{16}$, а в реєстр MAIPGL – значення 12_{16} . Також для роботи в повнодуплексному режимі біт PDPXMD реєстра PHCON1 було встановлено в лог. 1.

Після цього для Ethernet-контролера ENC28J60 було задана MAC-адреса 00:EE:DD:CC:BB:AA шляхом запису значення AA_{16} в реєстр MAADR0, значення BB_{16} в реєстр MAADR1, значення CC_{16} в реєстр MAADR2 і так далі.

Таким чином, фізичний (PHY) і канальний (MAC) рівні контролера ENC28J60 було налагоджено.

Для перевірки обміну даними Ethernet-контролера ENC28J60 із зовнішніми пристроями (тобто між рівнями АСКТП і АСКП) на ньому було реалізовано простий UDP-сервер. Суть роботи такого UDP-сервера полягала в тому, що Ethernet-контролер ENC28J60 повинний був тільки приймати запити і відповідати на них. Причому сам запити він не формував і не передавав.

Як відомо, UDP (User Datagram Protocol) – найпростіший протокол транспортного рівня, який дозволяє здійснювати досить швидкій і надійний обмін даними за допомогою невеликих за розміром повідомлень (“дейтаграм”). Особливо UDP-пакети підходять для передачі даних в реальному часі від первинних засобів вимірювання до вторинних і навпаки, так як можуть забезпечити меншу часову затримку, ніж TCP-пакети.

Для реалізації UDP-сервера було реалізовано спрощений стек протоколів фізичного, канального, мережевого, транспортного і прикладного рівнів (див. рис. 5) [1-5].

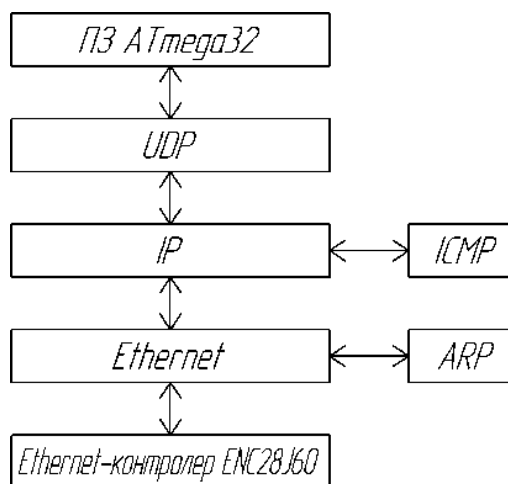


Рис. 5. Спрощений стек протоколів

Для коректної роботи каналного рівня (Ethernet) було реалізовано протокол ARP (Address Resolution Protocol). Основна робота цього протоколу – перетворення IP-адреси (мережевий рівень) в MAC-адресу (каналний рівень). Так як робилося припущення, що макетна плата AVRPLC16 v6 із Ethernet-контролером ENC28J60 знаходиться в промисловій локальній мережі, то Ethernet-контролеру ENC28J60 було призначено IP-адресу 10.1.1.1 (маска підмережі 255.0.0.0).

Крім того, для коректної роботи мережевого (IP) рівня було реалізовано протокол ICMP (Internet Control Message Protocol). Основна робота цього протоколу – передача повідомлень про помилки і інші виняткові ситуації, які виникли при обміні даними. В основному він використовувався для перевірки працездатності Ethernet-контролера ENC28J60.

Мікроконтролер ATmega32 мав вбудований 8-канальний 10-розрядний аналого-цифровий перетворювач із швидкодією до 15 тис. виб./с. На входи цього аналого-цифрового перетворювача подавалася тестова напруга, яка оцифровувалася і за допомогою UDP-сервера передавалася до персонального комп'ютера (див. рис. 4).

Аналіз роботи реалізованого на Ethernet-контролері ENC28J60 UDP-сервера показав достатньо велику надійність і малу часову затримку передачі “дейтаграм”, незважаючи навіть на відсутність підтвердження прийому цих “дейтаграм”. Таким чином, Ethernet-контролер ENC28J60 або аналогічні йому контролери можна застосовувати для швидкого і надійного обміну даними як між рівнями АСКТП і АСКП, так і всередині цих рівнів.

Висновки. В результаті безпосереднього програмування Ethernet-контролера ENC28J60 на мові програмування C було реалізовано найпростіший UDP-сервер, який дозволяв здійснювати обмін даними між макетною платою AVRPLC16 v6 із Ethernet-контролером ENC28J60 і персональним комп'ютером, імітуючи передачу даних в реальному часі від первинних засобів вимірювання до вторинних і навпаки. Було показано, що аналогічним чином за допомогою Ethernet-контролера ENC28J60 можна реалізувати більш складні технології обміну даними, такі як UDP-клієнт, TCP-сервер, TCP-клієнт, Web-сервер тощо. Також було показано, що використання Ethernet-контролерів типу ENC28J60 або аналогічних їм за технічними характеристиками і безпосереднє їх програмування під певні задачі дозволяє створювати розподілені системи автоматичного керування і здійснювати швидкий і надійний обмін даними між рівнями АСКТП і АСКП.

Список літературних джерел

1. Белов А.В. Разработка устройств на микроконтроллерах AVR: шагем от «чайника» до профи. Книга + видеокурс. – СПб.: Наука и Техника, 2013. – 528 с.: ил. + CD.
2. Мортон Дж. Микроконтроллеры AVR. Вводный курс. / Пер. с англ. – М.: Издательский дом «Додэка-XXI», 2006. – 272 с.: ил. (Серия «Мировая электроника»).
3. Петин В.А. Проекты с использованием контроллера Arduino. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2015. – 464 с.: ил. – (Электроника).
4. Прокопенко В.С. Программирование микроконтроллеров ATMEL на языке C. – К.: “МК-Пресс”, СПб.: “КОРОНА-ВЕК”, 2012. – 320 с., ил.
5. Ревич Ю.В. Практическое программирование микроконтроллеров Atmel AVR на языке ассемблера. – 3-у изд., испр. – СПб.: БХВ-Петербург, 2014. – 368 с: ил. – (Электроника).