

УДК 681.58:681.32

А.П. ПЛАХТЕЕВ, П.А. ПЛАХТЕЕВ

Харьковский национальный автомобильно-дорожный университет, Украина

## АНАЛИЗ И СИНТЕЗ МИКРОКОНТРОЛЛЕРНЫХ УСТРОЙСТВ РАСПРЕДЕЛЕННЫХ СИСТЕМ УПРАВЛЕНИЯ И СБОРА ДАННЫХ

Рассматривается протокол ADAM ASCII взаимодействия модулей в сети RS485, структуры специализированных микроконтроллерных модулей. Приведены форматы обмена модулей, особенности реализации на основе AVR микроконтроллеров. Проведена оценка размера программ модулей.

*микроконтроллерный модуль, протокол ADAM ASCII, распределенная система*

### Введение

**Актуальной задачей** является построение распределенных систем управления и сбора информации значительной протяженности на основе проводного интерфейса RS485. Системы включают процессорные и периферийные микроконтроллерные модули, например, серий ADAM4000 и ADAM5000 компании Advantech [1], i-7000 и i-8000 компании ICP-DAS [2]. Номенклатура периферийных модулей достаточно широкая и постоянно расширяется, обеспечивая выполнение простых типовых функций сбора, передачи данных и управления. Для решения конкретных задач число модулей может оказаться значительным, а стоимость неприемлемой.

Взаимодействие модулей осуществляется по простому протоколу ADAM ASCII. Увеличение числа модулей удлиняет цикл сбора данных и управления, что требует увеличения скорости обмена, а это приводит к уменьшению длины сегмента.

**Целью исследования** является оптимизация объема и стоимости оборудования распределенных систем на основе синтеза специализированных многофункциональных модулей, способных к сетевому взаимодействию с унифицированными модулями по протоколу ADAM ASCII.

**Задачами исследования** являются: анализ протокола обмена; оптимизация набора сообщений для модулей; синтез аппаратно-программных средств

модулей; анализ эффективности предлагаемых решений на основе микроконтроллеров.

Распределенные системы включают ведущие (Master) и ведомые (Slave) устройства. Наиболее сложно организовано взаимодействие элементов при наличии нескольких Master –  $M_0 \dots M_n$  (Multi-Master) и множества Slave-  $S_0 \dots S_m$  устройств (рис. 1) ввиду неизбежных конфликтов на шине.

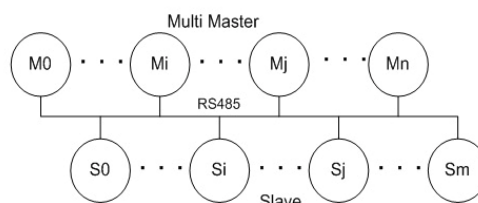


Рис. 1. Распределенная система Multi-Master

Рассмотрим более простую и распространенную шинную организацию распределенной системы (рис. 2) с одним ведущим устройством  $M$  и множеством Slave устройств (периферийных модулей, в дальнейшем - модулей) -  $S_0 \dots S_m$ .

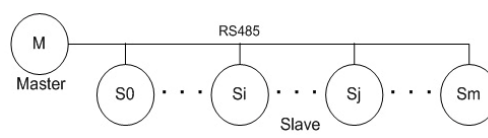


Рис. 2. Распределенная система с одним ведущим устройством

Взаимодействие ведущего модуля с ведомыми

модулями вида запрос-ответ (рис. 3) требует:

- присвоения адресов ведомым модулям  $A_1..A_k$ ;
- определения форматов запросов (command);
- определения форматов ответов (response).

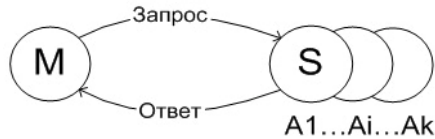


Рис. 3. Взаимодействие вида запрос – ответ

Продолжительность обмена (Т) определяется временем передачи запроса ( $T_3$ ), ответа ( $T_{от}$ ), выполнения операции ( $T_{оп}$ ) и ограничено временем  $T_{out}$ :

$$T = T_3 + T_{оп} + T_{от}, \quad (1)$$

$$T \leq T_{out}$$

Циклы обмена имеют различные варианты завершения (рис. 4) в зависимости от состояния элементов сети:

- а) нормальное завершение по времени,  $T < T_{out}$ 
  - запрос воспринят;
  - введен неверный запрос (запрос искажен);
- б) пропуск запроса,  $T > T_{out}$ ;
- в) искажение ответа,  $T < T_{out}$ ;
- г) ошибочное восприятие запроса,  $T < T_{out}$ .

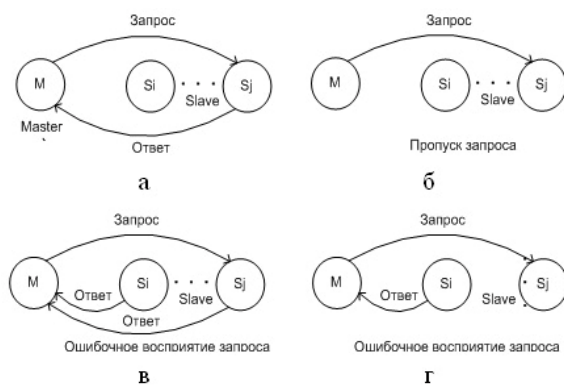


Рис. 4. Варианты завершения циклов обмена

Контроль за достоверностью данных и идентификацией завершения циклов обмена возлагается на аппаратные и программные средства различных уровней протокола обмена. Наиболее просто реализуется механизм адресного подтверждения операций обмена, контрольных данных. Вместе с тем, увеличение длины запросов и ответов, введение

контрольных циклов увеличивает продолжительность циклов обмена, цикла управления в целом, либо ограничивает число модулей в сети. Сохранение функциональности при уменьшении числа модулей достигается использованием многофункциональных специализированных модулей.

При использовании текстового протокола ADAM ASCII модель распределенной системы  $G$  может быть определена следующим образом

$$G = \langle N, v, A, S, R, F \rangle,$$

где  $N \leq 255$  - число модулей;

$v \in \{3, 4, 5, 6, 7\}$  - код скорости  $V (V = 150 \cdot 2^{v-3}$ , бод);

$A = \{A_1, \dots, A_i, \dots, A_N\}$  - множество адресов модулей  $G$ ;

$S = \|S_{ik}\|$  - множество команд всех модулей;

$R = \|R_{jl}\|$  - множество ответов всех модулей;

$F$  - формат данных.

Строки команд  $S_{ik}$  (ответов  $R_{ik}$ ) имеют форматы различной длины  $m_{ik}^C$  ( $m_{ik}^R$ ):

$$S_{ik} = \langle B, A_i, D_{ik}, cr \rangle, \quad m_{ik}^C = \text{Len}(D_{ik}) + 4;$$

$$S_{ik} = \langle B, A_i, cr \rangle, \quad m_{ik}^C = 4;$$

$$S_{ik} = \langle B, D_{ik}, cr \rangle, \quad m_{ik}^C = \text{Len}(D_{ik}) + 2;$$

$$R_{ik} = \langle B, A_i, D_{ik}, cr \rangle, \quad m_{ik}^R = \text{Len}(D_{ik}) + 4; \quad (2)$$

$$R_{ik} = \langle B, A_i, cr \rangle, \quad m_{ik}^R = 4;$$

$$R_{ik} = \langle B, cr \rangle, \quad m_{ik}^R = 2.$$

Каждая команда или запрос имеют идентификатор  $B$  и признак  $cr$  (обычно код 13), служащие для разделения общего потока символов в сети.

Строки параметров  $D_{ik}$  в командах и запросах имеют переменную длину  $\text{Len}(D_{ik})$  в зависимости от формата данных и содержания параметра. Если содержанием параметра являются числовые данные в диапазоне  $0..65535$ , то его представление в шестнадцатеричном формате имеет  $\text{Len}(D_{ik}) = 4$ , в десятичном (инженерном) формате  $\text{Len}(D_{ik}) = 5$ .

Множество команд  $i$ -го модуля

$$S_i = S_i^0 \cup S_i^1,$$

где:  $S_i^0 = \{S_{i1}^0, \dots, S_{iq}^0\}$  – фиксированное множество команд протокола общего назначения ( контроль присутствия, конфигурация, запрос имени и версии ПО модуля);

$$S_i^1 = \{S_{i1}^1, \dots, S_{iL_i}^1\}$$

– собственное множество команд модуля. Число команд меняется в широких пределах для различных типов модулей.

Модель сети для модуля  $i$  - поток данных

$$\dots, \langle S_i, R_i \rangle, \langle S_{i+1}, R_{i+1} \rangle, \dots$$

с интенсивностью

$$\lambda = \frac{\sum (m_i^C + m_i^R)}{T_C}, \quad (3)$$

как средним числом символов команд и ответов в единицу времени, зависящим от состояния элементов системы и сети (рис. 4) и продолжительности  $T_C$  цикла управления или сбора данных. Очевидно, что скорость обмена  $V$  должна выбираться из условия

$$\lambda \times n \leq V,$$

а с учетом (3) получим

$$(m^C + m^R) \cdot N \cdot n \leq V \cdot T_C.$$

Для типовых значений  $n = 10$  и  $V = 9600$ , получим  $\lambda \leq 960$ . Тогда для цикла управления  $T_C = 1$  сек и средних значений  $m^C + m^R = 12$  (16), в сеть может быть включено до 80 (60) модулей.

При синтезе специализированных модулей необходима оптимизация систем команд и ответов по критерию  $\min(\lambda)$ .

В соответствии с этой моделью построены несколько специализированных модулей с наименованиями – ‘PP01’, ‘AP05’, ‘AP06’, ‘AP07’.

Набор запросов и ответов специализированного модуля с сетевым адресом  $A = '01'$  и идентификатором ‘PP01’: чтение параметра T; чтение параметра

V; чтение параметра Z; чтение имени модуля; установка параметра T; установка параметра V; установка параметра Z. В качестве параметров в командах содержится трехразрядное десятичное число  $\langle D2..0 \rangle$  - ‘000’...’999’ и 3-разрядный шестнадцатеричный код  $\langle H2..0 \rangle$  - ‘000’...’FFF’. В табл. 1 приведены строки запросов и ответов без завершающего байта 0x0D (cr), который учитывается в оценке числа передаваемых ( $m^C$ ) и принимаемых ( $m^R$ ) символов циклов обмена. Время передачи запроса и приема ответа  $T$  определяется как

$$T_i = \frac{n}{V} \times (m_i^C + m_i^R). \quad (4)$$

Таблица 1

Форматы обмена модуля ‘PP01’

№	Запрос	Ответ	$m^C$	$m^R$	$T, \text{мс}$
1	\$01T	!01T<D2..0>	5	8	13,54
2	\$01V	!01V<D2..0>	5	8	13,54
3	\$01Z	!01Z<H2..0>	5	8	13,54
4	\$01M	!01AP05	5	8	13,54
5	#01T<D2..0>	!01	8	4	12,5
6	#01V<D2..0>	!01	8	4	12,5
7	#01Z<H2..0>	>	8	2	10,24

Проведенная по (2) оценка времени обмена с  $V = 9600$  бит/сек и  $n = 10$  (1 стартовый бит, 8 информационных бит, 1 стоповый бит) в табл.1 позволяет определить  $\max(m^C + m^R) = 13$  и максимальное время обмена  $T_i$  для любого из рассматриваемых циклов запрос-ответ. Время выполнения операции (1) можно ограничить 1-2 мс, т.е. длительность цикла запрос-ответ  $T < 15,5$  мс. Тогда можно выбрать временной интервал  $T_{out} = 16$  мс, в течение которого цикл запрос-ответ должен быть завершен.

Общая структура микроконтроллерного устройства распределенной системы (рис. 5) включает: схему драйвера сопряжения с последовательным интерфейсом RS485; собственно микроконтроллер; устройства согласования (VC) для аналоговых и цифровых сигналов.



Рис. 5. Общая структура микроконтроллерного устройства

На выбор микроконтроллера оказывают влияние различные факторы: эффективность архитектуры МК и функциональная достаточность; доступность и стоимость МК; перспективы поддержки линейки МК производителем и дилерской сетью; доступность и стоимость средств разработки от производителя и сторонних фирм; качество документации и технической поддержки разработчика и др.

Опыт разработки микроконтроллерных устройств и анализ состава серийно- выпускаемых модулей показывает, что 8-разрядные микроконтроллеры успешно выполняют коммуникационные функции и обработку данных. В частности, модули ADAM4000 построены на основе 8- разрядных микроконтроллеров фирм Motorola - MC68HC705C8 с 8К памяти программ и Winbond - W78C58BP с 32К Flash- памяти программ.

В настоящее время быстро заполняют рынок 8-разрядные AVR микроконтроллеры фирмы Atmel семейств: tiny, mega, Xmega [3].

В табл. 2 приведены характеристики нескольких моделей AVR- микроконтроллеров, предназначенных для тяжелых условий эксплуатации в автомобильных приложениях.

Таблица 2

Микроконтроллеры AVR для автомобильных приложений

mega	Flash Кбайт	SRAM Кбайт	EEPROM байт	I/O pin
ATmega164P	16	1	512	32
ATmega168	16	1	512	23
ATmega324P	32	2	1024	32
ATmega644P	64	4	2048	32
ATmega88	8	1	512	23

В основу разработок положен функционально развитый микроконтроллер ATmega8 фирмы Atmel, современные версии ATmega88 которого работают в расширенном автомобильном диапазоне (-40..+150°C).

Функциональная схема устройства на основе МК Atmel ATmega8/88 для распределенной системы приведена на рис. 6. ATmega8/88 содержит: 3 таймера-счетчика, АЦП, контроллеры последовательных интерфейсов USART, SPI, TWI, 6 каналов ШИМ и пр.

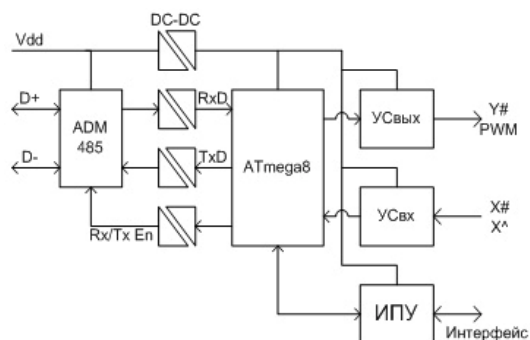


Рис. 6. Построение устройства на основе МК ATmega8

Схема включает элементы гальванической развязки МК от интерфейса RS485 (D+ и D-) и связанного с ним драйвера ADM485. В общем случае, МКУ имеет цифровые выходы Y# и каналы ШИМ (PWM), связанные с устройством согласования (УСвых); цифровые X# и аналоговые X^ входы, связанные с устройством согласования (УСвх); интеллектуальные периферийные устройства (ИПУ) с собственными интерфейсами.

На основе этой схемы с различными УСвых, УСвх и ИПУ можно построить ряд специализированных модулей для распределенных систем. Специализация модулей определяется также программой микроконтроллера.

На рис.7 приведена структура модуля 'AP05' управления мощным коллекторным двигателем постоянного тока (ДПТ) через симисторный фазовый регулятор (СФР). Датчик скорости V используется для организации контура обратной связи регулятора скорости.

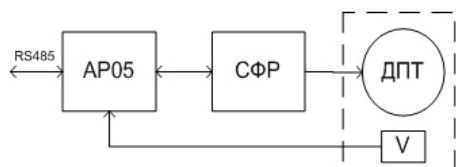


Рис. 7. Модуль управління двигателем

Модуль ‘AP05’ має набір запитів і відповідей (табл. 3) для установки затримки  $T$  включення СФР, контролю установленної затримки  $T$ , контролю швидкості обертання  $V$ . Цикл обміну включає передачу 10-13 символів.

Таблиця 3

Формати обміну модуля ‘AP05’

№	Запит	Відповідь	Операція
1	\$05T	<120	читання затримки
2	\$05V	<400	читання швидкості
4	\$05M	!05AP05	читання ідентифікатора
5	#05T255	!05	Установка затримки

Модуль ‘AP06’ (рис. 8) керує кутовим положенням двох органів управління за допомогою шагових двигачів ШД1,2. Драйвери двигачів ДШД забезпечують сопряження МК з ШД. Одночасно модуль опитує два датчики температури по інтерфейсам 1-Wire.

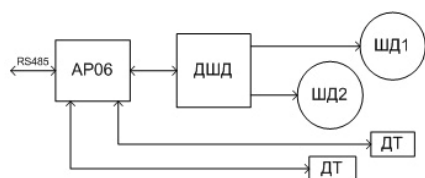


Рис. 8. Модуль управління ШД і контролю температури

Модуль ‘AP06’ має набір запитів і відповідей (табл. 4) для читання коду температури з одного з двох датчиків DS18B20, контролю установленної затримки  $T$ , установки кутів і напрямлення повороту ШД1, ШД2. Цикл обміну включає передачу 12-14 символів.

Модуль ‘AP07’ (рис. 9) являється мостом між інтерфейсом RS485 мережі, інтерфейсом RS232 пірометра Raytek і інтерфейсом SEI енкодерів M600. С його допомогою забезпечується доступ до елементів з несовместимими інтерфейсами.

Таблиця 4

Формати обміну ‘AP06’

№	Запит	Відповідь	Операція
1	\$06D1	<<+019	Чтение из DS18B20 №1
2	\$06D2	<<+107	Чтение из DS18B20 №2
4	\$06M	!06AP06	Чтение идентификатора
5	#06S1+050	!06	Поворот ШД1 на +50°
6	#06S2-050	!06	Поворот ШД2 на -50°

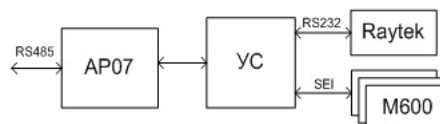


Рис. 9. Модуль согласования интерфейсов

Модуль ‘AP07’ має набір запитів і відповідей (табл. 5) для читання коду температури з пірометра, читання кута повороту з енкодера. Цикл обміну включає передачу 12-14 символів.

Таблиця 5

Формати обміну модуля ‘AP07’

№	Запит	Відповідь	Операція
1	\$07T	!T0019.9	читання з пірометра
2	\$07P	<33886	читання з енкодера M600
4	\$07M	!06AP07	читання ідентифікатора

Таким образом, базовая схема МКУ позволяет реализовать широкий спектр функций, определяемый разрабатываемым программным обеспечением. Сложность функций ограничивается объемом памяти программ и данных, а также быстродействием микроконтроллера.

Программные средства имеют две группы квази-параллельных функциональных компонентов – анализа запросов и формирования ответов; выполнения операций в соответствии с запросами. Схема обработки запросов в микроконтроллере включает селектор запросов, выделяющий из потока входных данных, адресованные данному модулю, запросы с контролем времени приема команд (1). Выполняется контроль синтаксиса запроса, формирование сообщения об ошибке для неопознанного запроса. Для корректных запросов  $R_1 \dots R_p$  управление передается соответствующим модулям обработки запросов.

Граф обработки потока входных данных включает состояние  $a0$  ожидания символа  $x \in B$  для перехода

да в состояние  $a1$  ожидания адреса  $A_i$ . Если адрес не опознан, или время ожидания истекло (условие  $T \mid /A_i$ ), то цикл начинается сначала ( $a0$ ). При совпадении с адресом модуля в состоянии  $a2$  начинается прием параметра команды ( $D$ ) до получения символа  $x = cr$  с выдачей команды на обработку ( $a4$ ), или  $x \in B$  для перехода в состояние  $a3$  сброса принятых символов и ожидания адреса  $A_i$  ( $a1$ ). Здесь  $T$  – условие превышения времени ожидания (таймаут). Такой функциональный элемент является составной частью программ всех специализированных модулей.

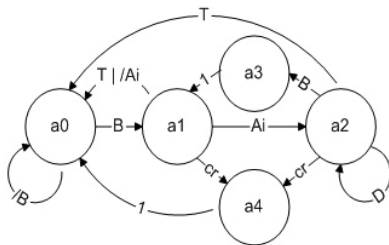


Рис. 11. Граф обработки потока входных данных в микроконтроллере

Время выполнения программы, объем программы, время и стоимость разработки программы определяется используемым языком программирования и соответствующим компилятором.

Наибольшей степенью оптимизации по размеру и скорости выполнения обладают программы, которые создаются на языках низкого уровня: Assembler, Algorithm Builder. Однако время и стоимость разработки таких программ, их верификация при увеличении сложности становятся во многих случаях неприемлемыми.

Среди языков высокого уровня преимущественное распространение получил язык C, поддерживаемый рядом коммерческих и свободно распространяемых компиляторов и интегрированных сред разработки (IDE).

Коммерческие компиляторы IAR Systems (IDE) для языка C имеют высокую стоимость, однако позволяют использовать обширные библиотеки и создавать компактный код, что является определяю-

щим фактором для микроконтроллерных устройств с ограниченным объемом Flash памяти программ.

Более доступны компиляторы ICC, Фитон и CodeVision, но они уступают IAR C. Свободно распространяемый компилятор WinAVR (GNU C) имеет худшие показатели по размеру генерируемого кода. Кроме того, он содержит ряд ошибок, что затрудняет создание надежного программного обеспечения на его основе.

Программы в памяти микроконтроллеров модулей (табл. 5) занимают часть объема, что позволяет развивать функциональные возможности модулей.

Таблица 5  
Размер программ модулей

№	Модуль	МК	Размер программы, слов (% Flash)
1	AP05	ATmega8	1850 (45%)
2	AP06	ATmega8	1900 (46%)
3	AP07	ATmega8	1080 (26%)

Таким образом, AVR микроконтроллеры могут быть основой для создания специализированных модулей для распределенных систем с последовательным интерфейсом RS485 и протоколом ADAM ASCII, обладающие невысокой стоимостью. Алгоритмы, заложенные в программы могут использоваться при проектировании элементов распределенных систем на основе других интерфейсов и протоколов.

## Литература

1. ADAM4000. User's Manual. – [Электронный ресурс]. – Режим доступа: <http://www.Advantech.com>.
2. Remote I/O Modules i-7000 series. – [Электронный ресурс]. – Режим доступа: <http://www.ICPDAS.com>
3. Сайт компании Atmel. – [Электронный ресурс]. – Режим доступа: <http://www.Atmel.com>.

Поступила в редакцию 22.02.2008

**Рецензент:** д-р техн. наук, проф. Б.М. Конорев, Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков.