

УДК 004.414.28 + 004.415.52

А.И. Левтеров, П.А. Плахтеев

Харьковский автомобильно-дорожный университет, Харьков

АВТОМАТНАЯ МОДЕЛЬ УЗЛА СЕНСОРНОЙ СЕТИ

Рассматривается модель узла сенсорной сети в виде множества автоматов. Граф переходов преобразуется к удобному для программной реализации виду. Проведены оценки затрат ресурсов для типовых модулей программ, построенных в различных средах. Проведена верификация программной реализации автоматной модели с использованием симулятора микроконтроллерной платформы.

Ключевые слова: узел сети, протокол обмена, состояние, автоматная модель, граф переходов.

Введение

Сетевые технологии, позволяющие взаимодействовать объектам реального мира, ведут к глубоким рыночным и технологическим переменам. На сегодняшний день более 99% объектов реального мира не охвачено сетями. К 2020 году, по прогнозам Cisco, к Интернету подключатся до 37 миллиардов всевозможных устройств и объектов. Получают повсеместное распространение бортовые сенсорные сети "умных" транспортных средств, машин, механизмов, автономных робототехнических комплексов. Важнейшее условие реализации этого – программируемые узлы сенсорных сетей на новейшей технологической базе, выводящие управляемость, контролируемость и масштабируемость сети на уровень, необходимый для поддержки чрезвычайно большого числа соединений. Функциональное многообразие узлов, жесткие ограничения по энергопотреблению, весогабаритным параметрам, стоимости и высокие требования к надежности обуславливают необходимость построения адекватных моделей функционирования микроконтроллерных узлов проводных и беспроводных сенсорных сетей совершенствования существующих и поиска новых решений на основе современных технологий, в том числе использование теории цифровых автоматов для построения программ.

Целью статьи является решение задачи описания модели функционирования узлов проводных сенсорных сетей в терминах конечных автоматов и методика ее реализации микроконтроллерными средствами.

Анализ последних исследований и публикаций. Существует множество проводных сенсорных сетей на основе интерфейсов RS232/485 (EIA/TIA-232/485) и подобных, в которых используются текстовые протоколы: Modbus ASCII [1], Овен, ADAM ASCII, DALI. Они реализованы в серийных модулях таких производителей, как Siemens (www.automation.siemens.com), компаний ICP DAS (www.icpdas.com), Advantech (www.advantech.com), Овен (www.owen.ru) и др. Многие датчики, исполнительные устройства имеют собственные текстовые протоколы обмена, которые требуют преобразования для использования в сенсорных сетях. Узлы, совместимые с RS232/485, могут быть построены на основе распространенных микроконтроллеров со встроенным асинхронным интерфейсом (UART). Кроме информационного обмена микроконтроллеры выполняют функции опроса датчиков и управления исполнительными устройствами, что позволяет улучшить весогабаритные и энергетические параметры, снизить стоимость конечных узлов сети.

Процессы обработки сообщений узлами сенсорных сетей имеют много общего с лексическим анализом в компиляторах и интерпретаторах языков программирования компьютеров, где производится распознавание и выделение лексем из входной последовательности символов [2] с использованием моделей конечных автоматов. Компилятор IAR Visual State (www.iar.com) использует автоматное представление алгоритма функционирования программ. В автоматном программировании [3] могут

использоваться как конечные автоматы, так и автоматы более сложной структуры. Полностью выполнение кода в автоматном стиле представляет собой цикл (возможно, неявный) шагов автомата.

Платформенно-независимое функциональное описание узлов в виде конечного автомата может дать возможность реализации требуемых задач на различных микроконтроллерах с ограниченными ресурсами.

Постановка задачи. Реализация конечным узлом сети протокола обмена включает выделение и обработку множества $C = \{C_1, \dots, C_i, \dots, C_k\}$ сообщений и формирование ответных сообщений из известного множества $R = \{R_1, \dots, R_i, \dots, R_k\}$. Число сообщений определяется числом и функциональностью узлов сети. Таким образом, ведущий узел реализует циклы обмена $\langle R_i, C_i \rangle$, а ведомый – $\langle C_i, R_i \rangle$. Все сообщения множества $M = C \cup R = \{M_1, \dots, M_i, \dots, M_k\}$ имеют формат

$$M_i = \langle DC_i, P_{i,1}, P_{i,1}, \dots, P_{i,N_i}, CR \rangle, \quad (1)$$

где DC_i – разделитель сообщения i (Delimiter Char); $P_{i,j}$ – параметр j сообщения i ($j=1..N_i$); N_i – число параметров сообщения i ; CR – код завершения сообщений.

Поскольку

$$\begin{aligned} (\forall i \in \overline{1, k} \forall j \in \overline{1, k} \exists i \neq j) DC_i &= DC_j \\ (\forall i \in \overline{1, k}) DC_i &\in D, \\ D &= \{D_1, \dots, D_n\}, \end{aligned} \quad (2)$$

множество M можно разделить на n групп, соответствующих множеству разделителей D .

Модель узла сенсорной сети представим множеством автоматов $1, 2, \dots, n$ (рис. 1), каждый из которых выполняет обработку соответствующих групп сообщений и управляется последовательностью символов X . Автоматы могут быть как отдельными элементами узла, так и реализованы совместно.

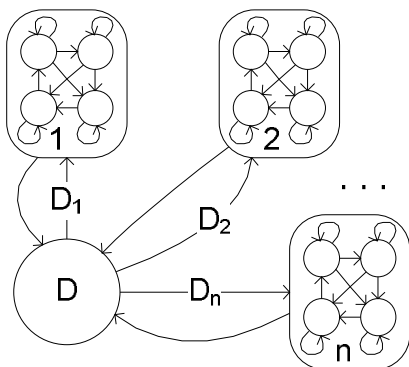


Рис. 1. Автоматная модель узла сенсорной сети

Начальное состояние D служит для обнаружения разделителей и передачи дальнейшей обработки соответствующему автомату. После успешной обработки или обнаружения ошибок управление возвращается состоянию D . Пусть $D=0$, тогда множест-

ва состояний автоматов не пересекаются и кодируются произвольными числами из ряда $-1, 2, 3, \dots$

Как известно, конечные автоматы описываются функциями переходов ϕ из текущего состояния $S(t)$ в $S(t+1)$:

$$S(t+1) = \phi[S(t), x(t)], \quad (3)$$

и выходов ξ, λ действий $Y(t)$:

$$Y(t) = \xi[S(t)], \quad (4)$$

$$Y(t) = \lambda[S(t), x(t)] \quad (5)$$

для автоматов Мура (4) и Мили (5). В общем случае, автомат Мили имеет меньшее число состояний.

Функции автоматов (3) – (5) могут быть реализованы аппаратными (FPGA), программными (MCU, CPU, DSP) и программно-аппаратными средствами (FPGA с аппаратными ядрами CPU). Для реализации функций программными средствами необходимо учитывать последовательный характер вычислений.

Селектор состояний

Рассмотрим реализацию переходов для простейшей автоматной модели, представленной графом на рис. 2. Здесь из входного потока

$$X = \langle x_0, x_1, x_2, \dots, x_{q-1}, x_q, x_{q+1}, \dots, x_{2q-1}, \dots \rangle$$

выделяются пакеты длины q символов:

$$\langle x_0, \dots, x_{q-1} \rangle, \langle x_q, \dots, x_{2q-1} \rangle, \dots$$

В данном случае $q=8$.

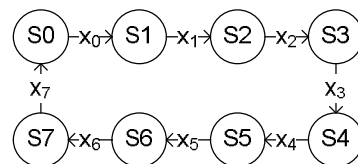


Рис. 2. Граф переходов простейшего автомата

Пусть коды состояний S_0, S_1, \dots, S_{q-1} определяются как:

$$(\forall i \in \overline{0, q-1}) S_i = i, \quad (6)$$

тогда на рис. 2 реализуется последовательность состояний автомата $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 0 \rightarrow \dots$

Программная реализация модели включает n секций программного кода, связанных с состояниями автомата, где реализуются преобразования эквивалентные функциям переходов и выходов канонических конечных автоматов. Текущее состояние определяет переход к соответствующей секции в каждом цикле работы автоматной модели. Введем состояние S выбора секции программного кода по текущему состоянию $S_0..S_7=0..7$, тогда графическое описание (рис. 3) будет реализовывать последовательность состояний $S-0-S-1-S-\dots-6-S-7-S-0-\dots$.

Операторы выбора, позволяющие определить положение фрагмента обработки текущего состояния S , с помощью псевдокодов записываются следующим образом:

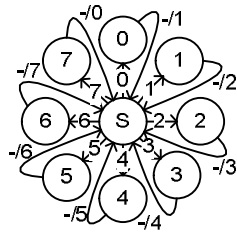


Рис. 3. Граф программной реализации автоматной модели

1. if ... then ...
 2. if ... then ... else ...
 3. case S
- ```

S1: ...
...
Sq: ...
END-CASE
ELSE ...
}

```

Оценки затрат памяти программ и времени выполнения полного цикла для программной реализации в IDE Arduino, WinAVR и Algorithm Builder (www.algrom.net), полученные в результате экспериментов с AVR микроконтроллерами, приведены в табл. 1.

Таблица 1

Параметры программной реализации

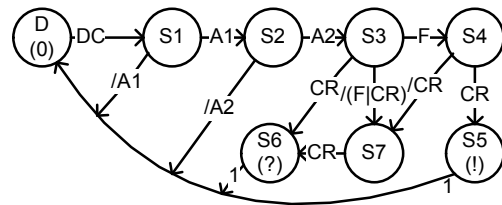
| № | Реализация            | Программа, слов | Цикл, мкс |
|---|-----------------------|-----------------|-----------|
| 1 | IDE Arduino           | 290             | 18        |
| 2 | WinAVR                | 167             | 21,75     |
| 3 | Algorithm Builder (1) | 112             | 6,5       |
| 4 | Algorithm Builder (2) | 89              | 5,5       |

Экспериментальные оценки среднего времени выбора состояния – 2.25 мкс, 2.7 мкс, 0.81 мкс и 0.71 мкс для реализаций 1 – 4 при тактовой частоте 16 МГц. Такие значения позволяют автоматной модели выполнять переходы на достаточно высокой скорости.

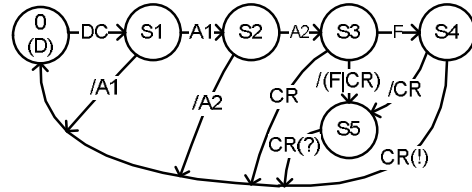
### Обработка сообщения

Рассмотрим эквивалентное представление автоматов Мура и Мили для обработки сообщения вида (1) с 3 параметрами: A1, A2 – адрес, F- функция. Для A1, A2 требуется полное совпадение, а отличие или отсутствие F приводит к сообщению об ошибке, как и появление дополнительных параметров, нарушающих формат сообщения (рис. 4). Модель не предусматривает ограничения времени ожидания ответного сообщения. Эквивалентным представлением автомата является таблица переходов и выходов. Для модели автомата Мура построена табл. 1. Выполняемые действия в ней определяются только текущим состоянием автомата.

Используя селектор состояний (рис. 3) и условия переходов из табл. 2 можно построить граф последовательной программной реализации автомата обработки сообщений заданного вида (рис. 5).



а



б

Рис. 4. Графическое описание автомата Мура (а) и Мили (б)

Таблица 2

Переходы и выходы автоматной модели

| №  | S <sub>t</sub> | X       | S <sub>t+1</sub> | Y <sub>t</sub> |
|----|----------------|---------|------------------|----------------|
| 1  | 0              | DC      | S1               | -              |
| 2  | S1             | /A1     | 0                | -              |
| 3  |                | A1      | S2               |                |
| 4  | S2             | /A2     | 0                | -              |
| 5  |                | A2      | S3               |                |
| 6  | S3             | F       | S4               | -              |
| 7  |                | CR      | S6               |                |
| 8  |                | /(F CR) | S7               |                |
| 9  | S4             | CR      | S5               | -              |
| 10 |                | /CR     | S7               |                |
| 11 | S5             | -       | 0                | ”d,d,...,CR”   |
| 12 | S6             | -       | 0                | ”e,e,...,CR”   |
| 13 | S7             | CR      | S6               | -              |

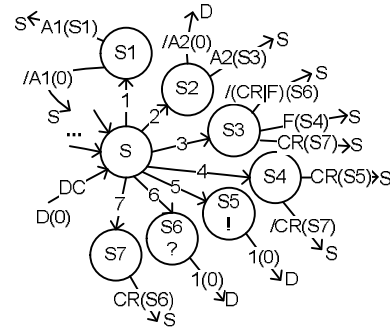


Рис. 5. Граф цикла обмена для команды вида – (A1)(A2)(F)(CR)

Если в состояниях S1..S4 и S7 переход определяется очередным принятым символом, то из состояний S5, S6 выполняется безусловный переход. Разделим эти группы состояний и получим схему алгоритма программной реализации автомата (рис. 6). Построенная по этому алгоритму программа в среде Algorithm Builder занимает 221 слово, что позволяет разместить ее в микроконтроллерах с минимальными ресурсами и повысить энергоэффективность. Аналогичная программа для IDE Arduino требует 1834 слова памяти программ, а для WinAVR – занимает 306 слов. Эти программы являются шаблонами для построения узлов с более сложным поведением.

Выбор шаблона определяется используемой в узлах сети микроконтроллерной платформой.

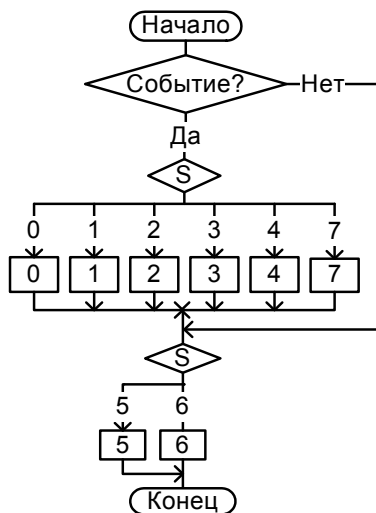


Рис. 6. Схема алгоритма программной реализации автоматной модели

### Верификация автоматной модели узла сенсорной сети

Верификация является необходимым этапом разработки автоматных программ [3, 4]. Использование симуляторов (Proteus и др.) позволяет уменьшить затраты и ускорить верификацию. Предлагается использование структуры (рис. 7), обеспечивающей визуализацию диаграммы последовательностей состояний посредством формирования кодов состояний для групп или отдельных переходов на дополнительных линиях портов и преобразования их в аналоговый сигнал U(S).

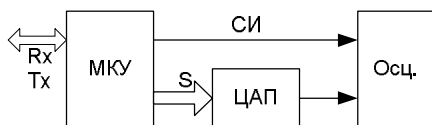


Рис. 7. Верификация автоматной модели

Это позволяет анализировать диаграммы состояний с помощью осциллографа. На рис. 8 приведены результаты моделирования такого способа верификации автоматной модели. В качестве примера обработки команд "\$05M\r" и цепи переходов используется циклическая входная последовательность символов:

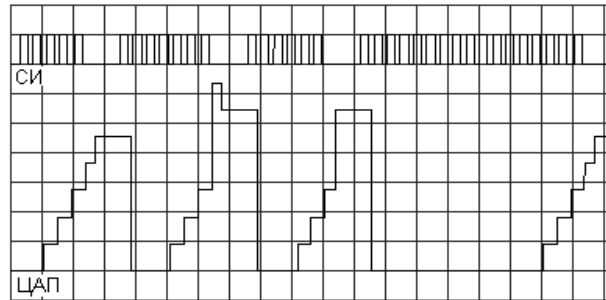
"\$05M\r01\$052\r23\$05\r45>+1.2345\r7\$05M\r".

Последовательность СИ определяет моменты поступления символов последовательности.

Визуальный образ последовательностей состояний, особенно на специально подобранных тестах, позволяет облегчить обнаружение нарушения условий функционирования.

### Выводы

Автоматные модели узлов сенсорных сетей позволяют получить компактное описание протоколов сенсорных сетей и их преобразование в программы как на языках высокого, так и низкого уровней.



а) 01234 5 001237 6 0001236 0 0 000000000012345  
б) \$05M\r 01\$ 052\r 23\$05\r 4 5 +1.2345\r7 \$05M

Рис. 8. Диаграмма последовательностей состояний (а) для входной последовательности (б)

Полученные шаблоны таких программ могут использоваться множеством связанных автоматов, а полученные оценки требуемых ресурсов – определить требования к аппаратной платформе.

Контроль диаграмм состояний в визуальной форме в симуляторах и в аппаратной реализации позволяет упростить верификацию программ и контроль в реальном масштабе времени.

### Список литературы

1. Modicon Modbus. Protocol Reference Guide. [Электронный ресурс]. – Режим доступа к ресурсу: [www.modbus.org/docs/PI\\_MBUS\\_300.pdf](http://www.modbus.org/docs/PI_MBUS_300.pdf).
2. Грис Д. Конструирование компиляторов для цифровых вычислительных машин / Д. Грис. – М.: Мир, 1975. – 543 с.
3. Поликарпова Н.И. Автоматное программирование[2] / Н.И. Поликарпова, А.А. Шальто. – СПб.: Питер, 2009. – 176 с.
4. Егоров К.В. Методика верификации автоматных программ / К.В. Егоров, А.А. Шальто // Информационно-управляющие системы. – 2008. – № 5. – С. 15-21.

Поступила в редколлегию 13.08.2013

Рецензент: д-р техн. наук, проф. В.С. Харченко, Национальный аэрокосмический университет им. Н.Е. Жуковского "ХАИ", Харьков.

### АВТОМАТНА МОДЕЛЬ ВУЗЛА СЕНСОРНОЇ МЕРЕЖІ

А.І. Левтеров, П.А. Плахтеев

Розглядається модель вузла сенсорної мережі у вигляді множини автоматів. Граф переходів перетворюється до зручного для програмної реалізації виду. Проведено оцінки витрат ресурсів для типових модулів програм, побудованих у різних середовищах. Проведена верифікація програмної реалізації автоматної моделі з використанням симулятора мікроконтролерної платформи.

Ключові слова: вузол мережі, протокол обміну, стан, автоматна модель, граф переходів.

## **AUTOMATE MODEL OF A SENSOR NETWORK NODE**

A.I. Levterov, P.A. Plakhteyev

*A model of the sensor network node in a form of a set of automata is being considered. A transition graph is being transformed into convenient for software implementation form. An estimation of resources consumption for typical program modules developed in different environments has been performed. The program implementation of automate model has verified with microcontroller platform simulator.*

**Keywords:** *network node, protocol, state, automata model, transition graph.*