



ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.434

РАЗРАБОТКА СПЕЦИАЛИЗИРОВАННОГО ЯЗЫКА ДЛЯ УДАЛЕННОГО ПРОГРАММИРОВАНИЯ МИКРОКОНТРОЛЛЕРОВ

В.В. МУРОМЦЕВ
В.В. ЛОМАКИН
Н.В. ЦОЦОРИНА

*Белгородский государственный
национальный исследовательский
университет*

e-mail:
muromtsev@bsu.edu.ru
lomakin@bsu.edu.ru
tsotsorina@bsu.edu.ru

В работе рассматривается специализированный язык для удаленного программирования микроконтроллеров и вопросы синтеза распознавателя данного языка по новой методике, основанной на трансформации диаграмм. Для апробации языка разработан прототип микроконтроллерного модуля, управляемого с помощью SMS.

Ключевые слова: специализированный язык программирования, распознаватель, микроконтроллер, AduC7020, Siemens TC 35i.

Введение.

Микроконтроллеры (МК) находят широкое применение при построении вычислительных систем, использующих беспроводные технологии передачи данных. Важным требованием при построении таких систем является возможность удаленного программирования МК, т.е. возможность загрузки программного обеспечения МК по беспроводному каналу. Для удаленного программирования обычно используют универсальные алгоритмические языки или простые командные языки. Использование командных языков не требует навыков программирования, но позволяет решать только задачи узкого класса. Использование универсальных алгоритмических языков не сужает класс задач, но требует привлечения к разработке высококвалифицированных программистов, что приводит к увеличению затрат на создание вычислительной системы. Расширить класс задач, которые могут быть решены без привлечения программистов можно за счет использования достаточно простого специализированного языка. Поэтому задача разработки такого языка, актуальна.

В работе рассматривается специализированный язык для удаленного программирования МК и вопросы синтеза распознавателя данного языка по новой методике, основанной на трансформации диаграмм. Для апробации языка разработан прототип микроконтроллерного модуля, управляемого с помощью SMS.

1. Структура аппаратного и программного обеспечения микроконтроллерного модуля

Модуль создан на базе МК ADuC7020 фирмы Analog Devices. Данный МК является 32-разрядным контроллером с ядром ARM7TDMI. Контроллеры ADuC702x широко применяются в промышленном оборудовании, средствах связи, медицине, автомобильной электронике и др. Особенностью МК серии ADuC702x является наличие в них прецизионной аналоговой части. ADuC7020 содержит встроенные 12-разрядные АЦП и ЦАП. Подробное описание МК можно найти на сайте производителя [1].

Для связи с контроллером используется GSM-модем Siemens TC35i, представляющий собой конструктивно законченное устройство передачи данных в стандарте GSM900/1800. Модем содержит полный интерфейс RS232. Управление модемом осуществляется с помощью AT-команд. Более детальную информацию о модеме можно найти на сайте производителя [2].

Для управления модулем используются текстовые команды, передаваемые с помощью SMS-сообщений. Результаты выполнения команд также передаются в виде SMS-сообщений. Все команды можно разделить на три вида: команды управления модулем; команды определения событий; команды смены состояний. Команды управления дают модулю задание выполнить то или иное действие. Команды определения событий задают правила: как модуль должен реагировать на те или иные изменения в окружающей среде. Команды смены состояний позволяют активизировать один из четырех возможных режимов работы модуля (рис.1).

При включении модуля активизируется режим выполнения команд. В этом режиме модуль реагирует на команды управления. При поступлении такой команды проверяется ее синтаксис. Если в команде обнаружена синтаксическая ошибка, то выводится соответствующее сообщение и команда не выполняется. Если ошибок нет, то команда выполняется.

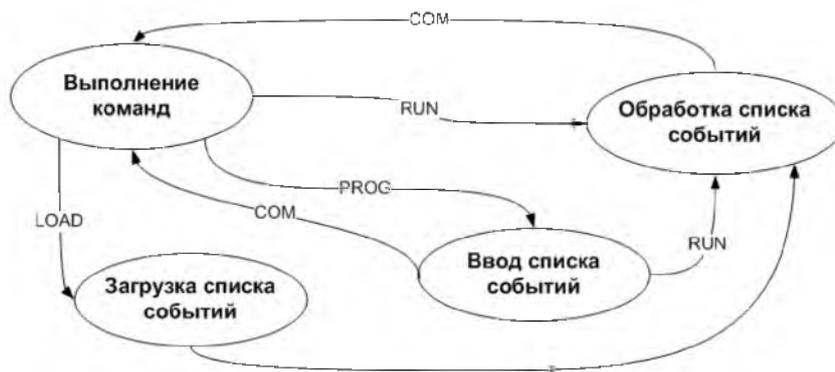


Рис. 1. Диаграмма состояний

В режиме ввода списка событий обрабатываются команды определения событий. При поступлении такой команды также проверяется синтаксис. При обнаружении синтаксической ошибки выводится соответствующее сообщение. Если ошибок нет, то команда запоминается в FLASH-памяти модуля. Команды вводятся последовательно. Для ввода сразу нескольких команд определения событий используется режим загрузки списка событий.



В режиме обработки событий модуль просматривает все определения событий, записанные в FLASH-память, и в том случае, когда условие некоторого события истинно, выполняет соответствующее действие. В этом режиме синтаксис команд не проверяется, т.к. в память могут быть записаны только синтаксически правильные команды.

2. Специализированный язык программирования

Перечислим основные команды управления микроконтроллерным модулем:

- Команда RD – считывает значение напряжения на заданной линии ввода.
- Команда ADC – считывает значение напряжения на заданном канале АЦП.
- Команда WR – устанавливает заданный уровень напряжения на заданной

линии вывода.

- Команда DAC – устанавливает заданное значение напряжения на заданном канале ЦАП.

- Команда OUT – отправляет заданное сообщение.

Синтаксис команды RD задается следующей БНФ:

<Команда RD>:=(“RD” | “rd”) (“<Десятичное число>”).

Синтаксическая переменная <Десятичное число> определяет целое положительное число в десятичной форме.

Все команды управления модулем имеют простой синтаксис и могут быть распознаны конечными автоматами. Синтез подобных распознавателей не представляет трудности, поэтому далее синтаксис команд данной группы рассматривать не будем.

Синтаксис команды определения события представляется следующей БНФ:

<Событие>:= <Условие наступления события>”:”<Действие>.

<Условие наступления события>:=<Лог.функция>.

<Лог.функция>:=“(”<Лог.функция>”)” |

<Лог.функция><Лог.операция><Лог.функция> |

”!”<Лог.функция> | <Лог.переменная>.

<Лог.операция>:=”&” | ”|”.

<Лог.переменная>:=<Команда RD><Знак операции><Уровень напряжения> | (<Команда ADC> | <Команда TIME>) <Знак операции> <Десятичное число>.

<Знак операции>:= ”>” | ”<” | ”=” | ”>=” | ”<=”.

<Уровень напряжения>:=<Высокий уровень> | <Низкий уровень>.

<Высокий уровень>:=”1” | ”H” | ”HI”.

<Низкий уровень>:=”0” | ”L” | ”LOW”.

<Команда TIME>:=”TIME” | ”time”.

<Действие>:= {<Команда вывода>”,”} <Команда вывода>.

<Команда вывода>:=<Команда WR> | <Команда DAC> | <Команда OUT>.

Команда TIME возвращает время, прошедшее с момента последнего возникновения события.

Синтаксис команды определения событий не может быть распознан конечным автоматом, т.к. язык, определенный синтаксической категорией <Лог.функция> не является регулярным. Для распознавания команды определения событий был синтезирован автомат, относящийся к классу односторонних недетерминированных распознавателей с магазинной памятью [3]. При этом использовалась разработанная методика, основанная на последовательной трансформации диаграмм. Для пояснения сути данной методики проиллюстрируем ее применение при синтезе языка, определяемого синтаксической категорией <Лог. функция>. Для сокращения записи введем следующие обозначения: <Лог. функция> – *L*; <Лог. операция> – \circ ; <Лог. переменная> – *p*. С учетом введенных обозначений синтаксис рассматриваемого языка определяется следующей КС-грамматикой:



$$L \rightarrow (L) | L \circ L | !L | p.$$

Вначале данная грамматика задается в виде КС-диаграммера [3,4], представленного на рис.2(а). Далее, по разработанным правилам удаления рекурсии диаграммер преобразуется в G-автомат [4]. G-автомат A это кортеж $(K, R, M, P, W, a_n, a_k)$, где K – множество состояний; R – входной алфавит; M – магазинный алфавит; P – множество операций; $W \subseteq \{((a,s),(b,o)) | (a,s) \in K \times R, (b,o) \in K \times P^*\}$ – переходы; a_n, a_k – начальное и конечное состояния. Для обозначения перехода $((a,s),(b,o))$ будем использовать запись $(a,b;s,o)$. Тройка $(A, \varpi, \nu) \in 2^K \times R^* \times M^*$, где 2^K – булеан множества K , ϖ – необработанная часть входной цепочки, ν – содержимое магазина называется конфигурацией G-автомата. Такт работы G-автомата определяется бинарным отношением (обозначение « \mapsto »), определенным на множестве конфигураций. Запись $T = (A, s\varpi, \nu) \mapsto (B, \varpi', \nu')$, где $s \in R$; $\varpi \in R^*$; $\nu, \nu' \in M^*$ означает, что выполнен такт T , то есть выполнены следующие действия:

- 1) $V \leftarrow \emptyset$; $B \leftarrow A$, здесь и далее символ « \leftarrow » означает операцию присваивания;
- 2) пока $V \neq B$ выполнять следующие действия:
 - а) $V \leftarrow B$;
 - б) $B \leftarrow B \cup \{b | (a,b;\varepsilon,o) \in W, a \in V\}$;
 - 3) $B \leftarrow \{b | (a,b;s,o) \in W, a \in V\}$.

G-автомат A допускает цепочку ϖ , если $(A_n, \varpi, \varepsilon) \mapsto^* (A_k, \varepsilon, \varepsilon)$, где $a_n \in A_n$, $a_k \in A_k$, « \mapsto^* » – транзитивное замыкание отношения « \mapsto ». G-автомат A допускает язык $L(A) = \{\varpi | \varpi \in R^*, (A_n, \varpi, \varepsilon) \mapsto^* (A_k, \varepsilon, \varepsilon)\}$

G-автомат, допускающий рассматриваемый язык логических выражений, представлен диаграммой на рис.2 (б). Диаграмма представляет собой ориентированный граф $G = (V, E)$, в котором $V = K$ – множество вершин, $E = W$ – множество дуг. Дуга $(a,b) \in E$, соответствующая переходу $(a,b;s,o)$, помечается цепочкой s,o . Если операция o состоит из нескольких операций, то символы, используемые для обозначения этих операций, разделяются запятой. В данном примере используются только три операции: Δ – не передвигать считывающую головку (не переходить к анализу следующего входного символа); $b+$ – занести символ b в магазин; $b-$ – удалить символ b из магазина. Если пометка дуги не содержит символ Δ , то по умолчанию, после выполнения соответствующего ей перехода, осуществляется передвижение считывающей головки к следующему входному символу.

Далее диаграмма упрощается путем замены группы параллельных переходов, содержащих операцию Δ одним безусловным переходом (в). После этого осуществляется перенос операций $b+$, $b-$ (г) и снова производится упрощение (д).

На последнем этапе удаляются спонтанные переходы (е).

Рассмотрим пример работы синтезированного G-автомата, представленного на рис.2 (е). Пусть на входе задана синтаксически правильная входная цепочка символов $p \circ (p \circ (p \circ p))$. В процессе распознавания данной цепочки автомат выполнит следующую последовательность тактов:



$$\begin{aligned}
 (L_H, p \circ (p \circ !(p \circ p)), \varepsilon) \mapsto (L_K, \circ(p \circ !(p \circ p)), \varepsilon) \mapsto (L_H, (p \circ !(p \circ p)), \varepsilon) \mapsto (L_H, p \circ !(p \circ p), b) \mapsto \\
 (L_K, \circ!(p \circ p), b) \mapsto (L_H, !(p \circ p), b) \mapsto (L_H, (p \circ p), b) \mapsto (L_H, p \circ p, bb) \mapsto \\
 (L_K, \circ p), bb) \mapsto (L_H, p), bb) \mapsto (L_K,), bb) \mapsto (L_K,), b) \mapsto (L_K, \varepsilon, \varepsilon)
 \end{aligned}$$

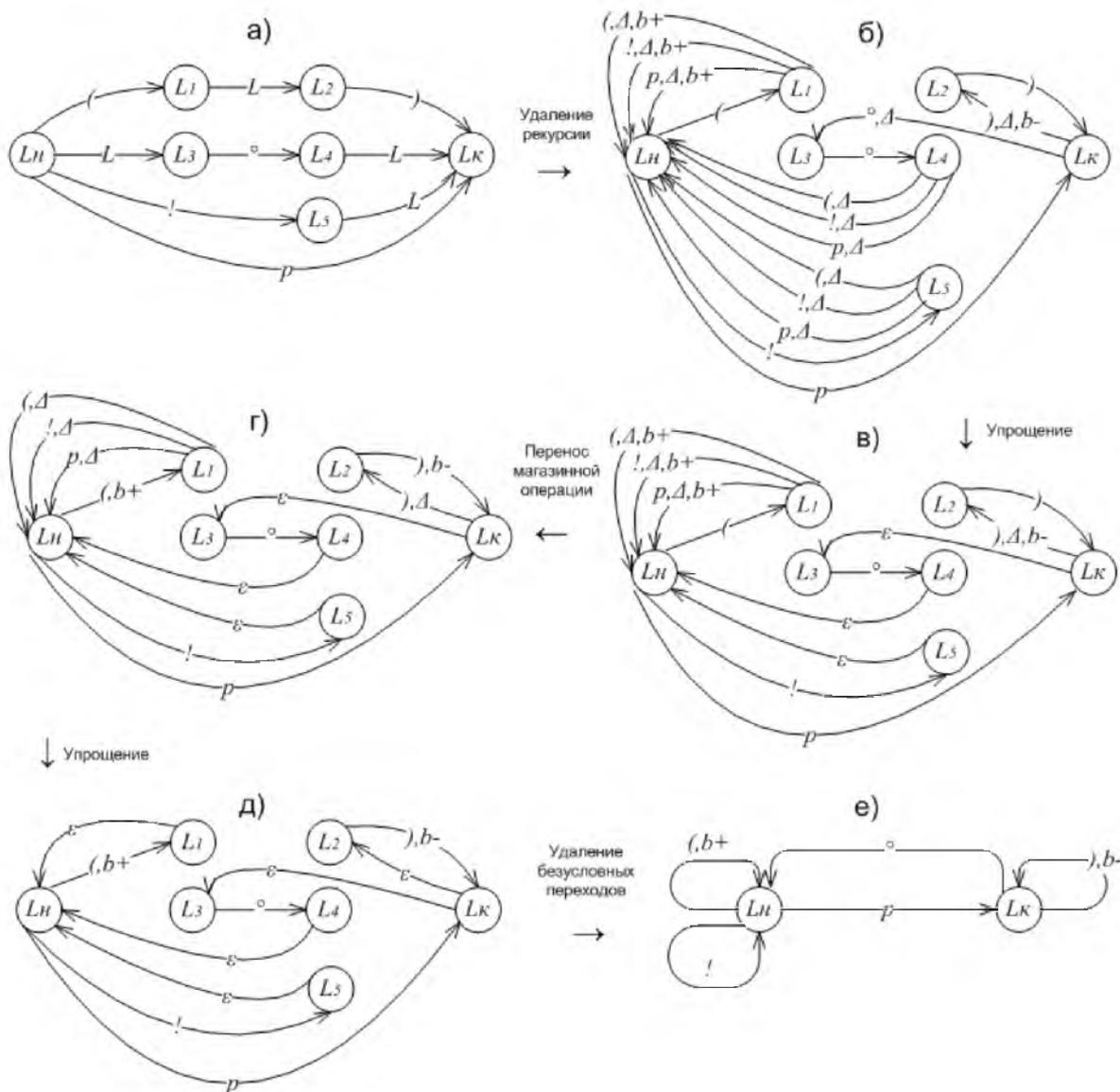


Рис. 2. Этапы синтеза G-автомата

Выводы.

Разработанный специализированный язык служит как для формирования команд управления, так и для определения простых алгоритмов. Использование такого языка позволяет снизить затраты при создании вычислительных систем, использующих беспроводные технологии передачи данных.

При разработке синтаксического анализатора языка использовалась новая формализованная методика, основанная на трансформации диаграмм. Применение данной методики позволяет ускорить процесс разработки синтаксических анализаторов.

Рассмотренный микроконтроллерный модуль может использоваться для создания законченных устройств. Поскольку модуль создан на базе МК, имеющего пре-



цизионную аналоговую часть, то он также может использоваться при проведении научных исследований.

Работа выполнена при финансовой поддержке ФЦП "Научные и научно-педагогические кадры инновационной России" на 2009-2013 годы. ГК № 14.740.11.0591 от 05.10.2010.

Список литературы

1. Analog Devices. Precision Analog Microcontroller, 12-Bit Analog I/O, ARM7TDMI@ MCU Analog Devices [Электронный ресурс]: технические характеристики прецизионных аналоговых микроконтроллеров компании Analog Devices. - Режим доступа: http://www.analog.com/static/imported-files/data_sheets/ADUC7019_7020_7021_7022_7024_7025_7026_7027_7028.pdf, свободный.
2. Siemens. TC 35i. Terminal. Hardware Interface Description [Электронный ресурс]: Описание GSM-модема Siemens TC 35i. - Режим доступа: <http://www.automation.siemens.com/mcms/topics/en/siplus/extreme/add-ons/gsm-kit/Documents/TC35iT-HW-Description.pdf>, свободный.
3. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции [Текст] т.1. Синтаксический анализ. - М.: Мир, 1978.
4. Муромцев, В.В. Автоматная модель для выполнения статистического моделирования при сжатии текстов на формальных языках [Текст] / В.В.Муромцев // Вопросы радиоэлектроники. Сер. ЭВТ. - 2007. - Вып.2. - С.139-149.

DEVELOPMENT OF SPECIALIZED LANGUAGE FOR REMOTE PROGRAMMING MICROCONTROLLER

V.V.MUROMTSEV

V.V.LOMAKIN

N.V.TSOTSORINA

*Belgorod National
Research University*

e-mail:

muromtsev@bsu.edu.ru

lomakin@bsu.edu.ru

tsotsorina@bsu.edu.ru

We consider a specialized language for remote programming of microcontrollers and the synthesis of a recognizer of the language of the new method, based on the transformation of the diagrams. To test the prototype of the language microcontroller module, managed by SMS.

Key words: specialized programming language, recognizer, microcontroller, AduC7020, Siemens TC 35i.