

УДК 004.4

Костин Даниил Александрович

*Приамурский государственный университет им.Шолом-Алейхема
студент*

Баженов Руслан Иванович

*Приамурский государственный университет им.Шолом-Алейхема
к.п.н., доцент, зав.кафедрой информационных систем, математики и
методик обучения*

Разработка цифрового автомата, управляющего работой автомобильной сигнализации

Аннотация

В статье показана разработка автомобильной сигнализации на основе цифрового автомата. В тексте статьи описана схема функционирования прототипа автомобильной сигнализации, приведена модель цифрового автомата, управляющая поведением описанного устройства, и код реализации автомата (с++ для AVR микроконтроллера).

Ключевые слова: теория автоматов, автоматное программирование, автомобильная сигнализация.

Kostin Daniil Aleksandrovich

*Sholom-Aleichem Priamursky State University
student*

Bazhenov Ruslan Ivanovich

*Sholom-Aleichem Priamursky State University
Candidate of pedagogical sciences, associate professor, Head of the Department
of Information Systems, Mathematics and teaching methods*

Development of a digital machine that controls the car alarm

Abstract

The article shows the development of car alarms based on the digital machine. The text of the article describes the scheme of functioning prototype car alarm is given a digital machine model, control the behavior of the apparatus described , and the implementation of machine code (C ++ for the AVR microcontroller).

Keywords: automata theory, automata-based programming, car alarm.

Применение автоматного программирования при разработке устройств со сложным поведением позволяет создать качественное программное обеспечение. Такой подход обеспечивает отделение описания логики

поведения от описания его семантики. При этом описание логики жестко структурировано. Эти свойства делают автоматное описание сложного поведения наглядным и ясным [5].

Принципы визуализации состояний сложных устройств (как один из этапов разработки) были подробно изложены Д.Харелом [1]. Общая концепция выбранного подхода проработана А.А.Шалыто [2, 3]. Ряд программных моделей конкретных устройств, разработанных на основе теории автоматов разработал Р.И. Баженов и др. [4-13]. Применительно к встраиваемым системам на базе микроконтроллеров необходимо отметить цикл статей В. Татарчевского [14], а также труды Н.А.Зубкова, В.М.Иськив, В.В.Чугунова [15] и других авторов [16-18].

В данной статье представлена управляющая программа для прототипа автомобильной сигнализации. Для формулирования задачи опишем схему работы устройства:

- При поступлении питания устройство входит в режим ожидания.
- С пульта кнопкой «1» производится постановка на охрану, а кнопкой «2» снятие с охраны.
- В режиме охраны сигнализация чувствительна к ударам.
- Сильный удар приводит к переходу в режим тревоги, сопровождаемый сиреной.
- При слабом ударе устройство подает короткий сигнал и переходит в режим опасности.
- Если в течение 30 секунд последует еще один удар (сильный или слабый), срабатывает тревога.
- Иначе сигнализация возвращается в основной режим охраны.

В соответствии с заданным поведением разработаем граф состояний (рис. 1)

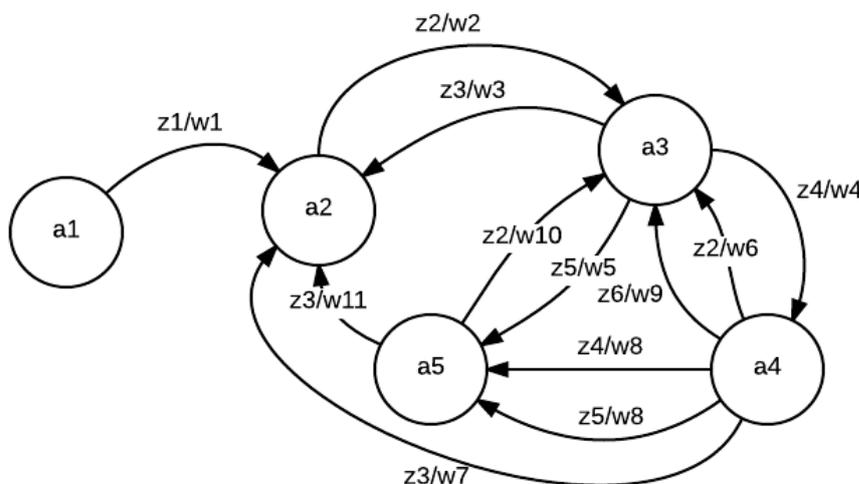


Рисунок 1 - Граф состояний

Пояснения к обозначениям рисунка даны в соответствующих таблицах (табл. 1, табл. 2, табл. 3).

Таблица 1 - Внутренние состояния

Состояние
a_1 – не инициализировано
a_2 – режим ожидания
a_3 – режим охраны
a_4 – режим опасности
a_5 – режим тревоги

Таблица 2 - Входные сигналы

Сигнал
z_1 – подача питания
z_2 – кнопка постановки на охрану
z_3 – кнопка снятия с охраны
z_4 – слабый удар
z_5 – сильный удар
z_6 – 30-секундный таймер режима опасности

Таблица 3 - Выходные сигналы

Сигнал	Сигнал
w_1 – инициализировать	w_7 – отменить опасность, снять с охраны
w_2 – включить охрану	w_8 – отключить таймер, включить тревогу
w_3 – выключить охрану	w_9 – отключить таймер, поставить на охрану
w_4 – включить опасность	w_{10} – отключить тревогу, поставить на охрану
w_5 – включить тревогу	w_{11} – отключить тревогу, снять с охраны
w_6 – отменить опасность	

Введем в программу описанные состояния и сигналы в виде перечислений:

```
enum states { uninitialized, standby, guard, disturbed, alarm };
enum events { power_on, button_on, button_off, weak_knock, strong_knock,
             calm_timer };
enum commands { initialize, enable_guard, disable_guard, make_disturbed,
              enable_alarm, make_calm, make_calm_and_disable_guard,
              disable_timer_and_make_alarm, disable_timer_and_enable_guard,
              disable_alarm_and_enable_guard, disable_alarm_and_disable_guard };
```

Программа выполняется в цикле. Текущее состояние хранится в глобальной переменной *state*. Любое из описанных входных событий влечет за собой вызов подпрограммы «*automat*»:

```
void automat(int event) {
```

```
switch (state) {
  case uninitialized:
    switch (event) {
      case power_on:
        state = standby;
        execute(initialize);
        break;
    }
    break;
  case standby:
    switch (event) {
      case button_on:
        state = guard;
        execute(enable_guard);
        break;
      case button_off:
        state = standby;
        execute(disable_guard);
        break;
    }
    break;
  case guard:
    switch (event) {
      case button_on:
        state = guard;
        execute(enable_guard);
        break;
      case button_off:
        state = standby;
        execute(disable_guard);
        break;
      case weak_knock:
        state = disturbed;
        execute(make_disturbed);
        break;
      case strong_knock:
        state = alarm;
        execute(enable_alarm);
        break;
    }
    break;
  case disturbed:
    switch (event) {
      case button_on:
```

```
    state = guard;
    execute(disable_timer_and_enable_guard);
    break;
case button_off:
    state = standby;
    execute(make_calm_and_disable_guard);
    break;
case calm_timer:
    state = guard;
    execute(make_calm);
    break;
case weak_knock:
    state = alarm;
    execute(disable_timer_and_make_alarm);
    break;
case strong_knock:
    state = alarm;
    execute(disable_timer_and_make_alarm);
    break;
}
break;
case alarm:
    switch (event) {
        case button_on:
            state = guard;
            execute(disable_alarm_and_enable_guard);
            break;
        case button_off:
            state = standby;
            execute(disable_alarm_and_disable_guard);
            break;
    }
    break;
}
}
```

В этой подпрограмме происходит переход в новое состояние и исполнение команды в соответствии с текущим состоянием и входным сигналом.

Конечный вид устройства, работающего под управлением разрабатываемой программы показан на рис. 2.

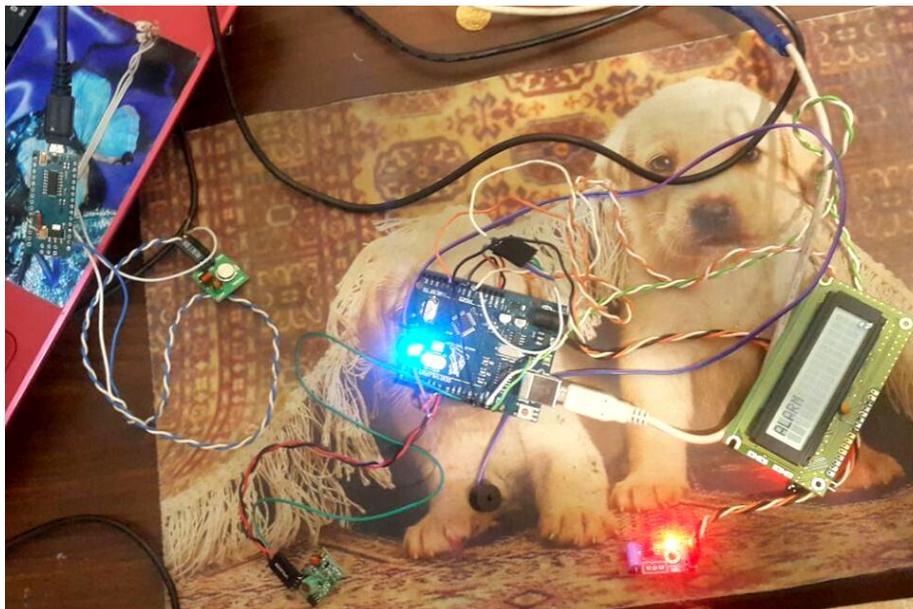


Рисунок 2 - Запрограммированное устройство (состояние тревоги)

Таким образом, была разработана программа управления автомобильной сигнализацией для микроконтроллера. Автоматное программирование позволило избежать запутанного спагетти-кода, реализованный алгоритм структурирован и ясен, а возможность допустить ошибку существенно уменьшена. Результаты исследования могут быть использованы для обучения прикладной теории автоматов.

Библиографический список

1. Harel D. Statecharts: A visual formalism for complex systems. Rehovot: The Weizmann Institute of Science.
2. Поликарпова Н.И., Шалыто А.А., Автоматное программирование. 2008. с. 11-12.
3. Вавилов К.В., Шалыто А.А. Что плохого в неавтоматном подходе к программированию контроллеров? // Промышленные АСУ и контроллеры. 2007. № 1. с. 49-51.
4. Баженов Р.И. Методические рекомендации для выполнения курсовой работы по дисциплине «Теория автоматов». Биробиджан, 2008. 20 с.
5. Баженов Р.И., Афанасьева М.А. Разработка программной модели контроля дверей холодильника на основе теории автоматов // Журнал научных публикаций аспирантов и докторантов. 2014. №4(94). с. 306-308.
6. Баженов Р.И., Балашова С.В., Моисеев В.В. Разработка компьютерной модели MP3-плеера на основе switch-технологии // Журнал научных публикаций аспирантов и докторантов. 2014. №6(96). С. 246-248.
7. Векслер В.А., Баженов Р.И. Формирование модели обучения взрослых основам информационных технологий: региональный аспект. Биробиджан, 2014.
8. Баженов Р.И. Организация научно-исследовательской работы студентов

- по дисциплине «Теория автоматов» // Современная педагогика. 2014. № 5 (18). С. 20.
9. Баженов Р.И., Гринкруг Л.С. Информационная система по расчету и распределению нагрузки профессорско-преподавательского состава ФГБОУ ВПО «Приамурский государственный университет им. Шолом-Алейхема» // Информатизация и связь. 2012. № 5. С. 75-78.
 10. Баженов Р.И., Гринкруг Л.С. Информационная система абитуриент-деканат ФГБОУ ВПО «Приамурский государственный университет им. Шолом-Алейхема» // Информатизация и связь. 2013. № 2. С. 97-99.
 11. Волков Е.И., Баженов Р.И. Разработка программной системы «Сканер социальных связей» // Современные научные исследования и инновации. 2014. № 5-1 (37). С. 22.
 12. Дзиковский Ф.Г., Баженов Р.И. Компьютерная игра по элементарным преобразованиям матриц // Современные научные исследования и инновации. 2014. № 7 (39). С. 23-31.
 13. Манойленко И.Г., Кардаш А.С., Баженов Р.И. Разработка модели управления холодильником на основе switch-технологии // Nauka-Rastudent.ru. 2015. № 2 (14). С. 17.
 14. Татарчевский В. Применение SWITCH-технологии при разработке прикладного программного обеспечения для микроконтроллеров. Часть 1 // Компоненты и технологии. 2006. № 11. URL: http://www.kite.ru/articles/circuit/2006_11_164.php (дата обращения 08.11.2015).
 15. Зубков Н.А., Иськив В.М., Чугунов В.В. Применение switch-технологии при разработке прикладного программного обеспечения для микроконтроллеров avr // Сборник научных трудов Sworld. 2010. Т. 7. № 4. С. 39-40.
 16. Милехин Д.А., Смагин Ю.С., Гоман А.А., Шатковский О.Ю. Унификация алгоритмов функций логики микропроцессорных систем централизации с использованием switch-технологии // Проектирование и технология электронных средств. 2007. № 4. С. 59-63.
 17. Шифрин Б.М. Применение switch-технологии для разработки логической модели управления процессом загрузки шпона // Известия Санкт-Петербургской лесотехнической академии. 2009. № 186. С. 151-155.
 18. Reddy U.S. Automata-Theoretic Semantics of Idealized Algol with Passive Expressions // Electronic Notes in Theoretical Computer Science. 2013. Т. 298. № 4. С. 325-348.