8. *Бродин Б.В., Шагурин И.И.* Микроконтроллеры: Справочник. М.: ЭКОМ, 1999. 395 с.
9. Программируемые логические ИМС на КМОП-структурах и их применение. / П.П. Мальцев, Н.И. Гарбузов, А.П. Шарапов, А.А. Кнышев. М.: Энергоатомиздат, 1998. 158 с.
10. *Соловьев В.В., Васильев А.Г.* Программируемые логические интегральные схемы и их применение. Мн.: Беларуская наука, 1998. 270 с.
11. *Лаптев В.* Цифровой измеритель температуры на базе AVRмикроконтроллера и RC-цепочки. Электронные компоненты, 2001. № 2. С. 46—49.

ПРОГРАММИРОВАНИЕ МИКРОКОНТРОЛЛЕРА

Митарова Д.М.

Митарова Джамиля Магомедовна – студент,
кафедра информационных сетей и техносферной безопасности,
Российский государственный социальный университет, г. Москва

Аннотация: *в статье анализируется, какие задачи вы возлагаете на микроконтроллер, и как он будет их выполнять, определяется заложенной в него программой – программой, которую для микроконтроллера составляете вы сами.*
Программа (в переводе это слово означает – «предписание») – предварительное описание предстоящих событий или действий.
К примеру, мы хотим, чтобы микроконтроллер помигал светодиодом. Довольно простая задача, но, тем не менее, для того чтобы микроконтроллер выполнил ее, мы, предварительно, должны шаг за шагом описать все действия микроконтроллера — написать программу, которую он должен выполнить для получения нужного нам результата – мигающий светодиод.
Ключевые слова: *микроконтроллер, язык С++.*

ВВЕДЕНИЕ

Какие задачи вы возлагаете на микроконтроллер, и как он будет их выполнять, определяется заложенной в него программой – программой которую для микроконтроллера составляете вы сами.

Программа (в переводе это слово означает – “предписание”) – предварительное описание предстоящих событий или действий [1].

К примеру, мы хотим, чтобы микроконтроллер помигал светодиодом. Довольно простая задача, но тем не менее, для того, чтобы микроконтроллер выполнил ее, мы, предварительно, должны шаг за шагом описать все действия микроконтроллера — написать программу, которую он должен выполнить для получения нужного нам результата – мигающий светодиод.

Что-то вроде такого:

Зажечь светодиод:

настроить вывод микроконтроллера, к которому подключен светодиод, для работы на вывод информации

подать на этот вывод логический уровень, который позволит зажечь светодиод

Подождать некоторое время:

перейти к подпрограмме формирующей паузу (которую тоже нужно “прописать”)

по выполнению подпрограммы паузы вернуться в основную программу

Погасить светодиод:

подать на вывод микроконтроллера логический уровень, гасящий светодиод

и так далее.

Алгоритм – набор инструкций, описывающих порядок действия для достижения нужного результата [2].

Если в программе мы подробнейшим образом прописываем все действия микроконтроллера, то в алгоритме, — мы определяем порядок действий микроконтроллера, на основе которых мы потом создадим программу. По аналогии с вышеприведенном примером:

Зажечь светодиод

Подождать некоторое время

Погасить светодиод

и так далее.

Таким образом, алгоритм – это предшественник программы. И чем тщательно и продумано будет создан алгоритм, тем проще будет создавать программу.

1 Языки программирования

К сожалению, если любимой собачке мы можем подавать команды на человеческом языке, то общение с микроконтроллером должно происходить на языке, который понятен ему — языке микроконтроллерных команд.

Команды для микроконтроллера имеют вид набора единичек и нулей, типа:

00110101 011000100

так называемые – коды команд, а коды команд – это язык, который понимает микроконтроллер. А для того, чтобы перевести наш язык общения на язык микроконтроллера – в эти самые наборы нулей и единичек, существуют специальные программы [4].

Эти программы позволяют описать порядок работы для микроконтроллера на более-менее понятном для нас языке, а затем перевести этот порядок на язык понятный микроконтроллеру, в результате чего получается так называемый машинный код – последовательность команд и инструкций (те самые нули и единички) которые только и понимает микроконтроллер. Текст программы, написанный программистом, называется исходным кодом. Перевод программы с языка программирования (исходного кода) на язык микроконтроллера (машинный код) производится трансляторами. Транслятор превращает текст программы в машинные коды, которые потом записываются в память микроконтроллера [5].

В таких программах порядок работы микроконтроллера описывается специальным языком – языком программирования.

Язык программирования – это способ передачи команд, инструкций, чёткого руководства к действию для микроконтроллера [3].

Из множества языков программирования можно выделить два типа:

- языки программирования низкого уровня
- языки программирования высокого уровня

Чем они отличаются. А отличаются они своей близостью к микроконтроллеру.

На заре зарождения микропроцессорной техники, программы писали в машинных кодах, то есть весь алгоритм работы последовательно прописывали в виде нулей и единичек [7]. Вот так, примерно, могла выглядеть программа:

01010010

01000110

10010011

Трудно, даже профессионалу, разобраться в такой комбинаций из двух цифр. Для облегчения своей жизни, программисты стали создавать первые языки программирования. Так вот, чем ближе язык программирования к такому набору нулей и единиц тем больше он “низкого уровня”, а чем дальше от них – тем больше “высокого уровня”.

2Программированиемикроконтроллера

Разрабатываемое устройство вычисляет две независимые величины E и F по следующим формулам: Σ = = n i i A n E 1 1 ; Σ = = n i i B n F 1 1 , где A и F – аналого- вое сигналы, B и E – цифровые сигналы, n – количество введенных на данный момент значений. Для программирования микроконтроллера можно использовать язык Си, либо, ассемблер. Программирование производилось на языке ANSI C ввиду его наглядности и неизменности стандартов [8].

2.1. Особенности программирования на языке C++

Программирование микроконтроллеров на Си имеет ряд особенностей, связанных со спецификой управления реальным объектом. Во-первых, программа для микроконтроллера никогда не должна заканчиваться, а значит помимо главной программы main(), обязательным является наличие главного бесконечного цикла while(1) внутри main() [10]. Таким образом, код будет выполняться пока на микроконтроллер подается питание. Второй особенностью работы с микроконтроллерами на языке Си являются побитовые операции, которые встречаются очень редко при классическом программировании. Все «переключатели» и переменные микроконтроллера находятся внутри 8-битных регистров и очень часто бывает необходимо взаимодействовать только с одним битом регистра, а все остальные оставить, как было и не учитывать [13].

2.2. Управление битами регистра

Стандартные порты ввода-вывода организованы таким образом, что каждому выводу микроконтроллера соответствует один бит в трех 8- ЭВМ и программная обработка данных 31

битных регистрах. В микроконтроллерах AVR, порты (регистры, связанные с физическими контактами) именуются буквами (A,B,C,…), а контакты цифрами от 0 до 7. Каждому порту соответствуют следующие регистры:

- DDRx – регистр направления передачи сигнала (0 – Выход, 1 – Вход);
- PORTx – регистр значения (для выхода) и подтяжки (для входа);
- PINx – регистр состояния (реальный логический уровень на контакте). Подтяжка (Pull-up) – Подключение контакта к питанию или земле через высокоомный(около 40 КОм) резистор [15].

ЗАКЛЮЧЕНИЕ

Алгоритм управления был написан на одном из самых распространенных языков, следовательно модификация этого устройства под любые нужды не составит никакого труда. Устройство можно настроить на работу с полноценным знаковым дисплеем, а так же, благодаря аппаратному UART, легко организовать связь с ПК через RS-232 или USB. С помощью микроконтроллера можно управлять шаговыми двигателями и сервоприводами. Столь широкие возможности по обработке данных и управлению физическими объектами нашли применение во многих отраслях науки и промышленности.

Список литературы

1. *Прокопенко В.С.* Программирование микроконтроллеров ATMEL на языке С. СПб.: КОРОНА-ВЕК, 2012. 307 с.
2. *Дхананья Гадре, Нигуал Мэлхотра.* Занимательные проекты на базе микроконтроллеров tinyAVR. СПб.: БХВ-Петербург, 2012. 330 с.
3. Википедия – свободная энциклопедия. [Электронный ресурс]. // Wikimedia Foundation, Inc.: [сайт], 2001. Режим доступа: http://ru.wikipedia.org/ (дата обращения: 30.08.2013).
4. Краткий Курс – Самоучитель AVR, ATmega и ATtiny. [Электронный ресурс], 2007. Режим доступа: http://123avr.com/ (дата обращения: 30.08.2013).
5. *Белов А.В.* Программирование микроконтроллеров для начинающих и не только. Книга + виртуальный диск / А.В. Белов. СПб.: Наука и техника, 2016. 352 с.
6. *Брей Б.* Применение микроконтроллеров PIC 18. Архитектура, программирование и построение интерфейсов с применением С и ассемблера / Б. Брей. СПб.: КОРОНА-Век, 2008. 576 с.
7. *Брей Б.* Применение микроконтроллеров PIC 18. Архитектура, программирование и построение интерфейсов с применением С и ассемблера / Б. Брей. СПб.: КОРОНА-Век, 2014. 576 с.
8. *Иванов В.Б.* Программирование микроконтроллеров для начинающих. Визуальное проектирование, язык С, ассемблер / В.Б. Иванов. СПб.: КОРОНА-Век, 2015. 176 с.
9. *Иванов В.Б.* Программирование микроконтроллеров для начинающих: Визуальное проектирование / В.Б. Иванов. СПб.: Корона-Век, 2010. 176 с.
10. *Каспер Э.* Программирование на языке Ассемблера для микроконтроллеров семейства i8051 / Э. Каспер. М.: ГЛТ, 2012. 192 с.
11. *Магда Ю.С.* Программирование и отладка С/С++ приложений для микроконтроллеров ARM / Ю.С. Магда. М.: ДМК, 2014. 168 с.
12. *Магда Ю.С.* Программирование и отладка С/С++ приложений микроконтроллеров ARM / Ю.С. Магда. М.: ДМК Пресс, 2012. 168 с.
13. *Прокопенко В.С.* Программирование микроконтроллеров ATMEL на языке С / В.С. Прокопенко. СПб.: Корона-Век, 2013. 320 с.
14. *Ревич Ю.В.* Практическое программирование микро-контроллеров Atmel AVR на языке ассемблера / Ю.В. Ревич. СПб.: BHV, 2012. 352 с.
15. *Хелибайк Ч.* Программирование PIC- микроконтроллеров на PICBasic / Ч. Хелибайк. М.: Додэка XXI, 2007. 336 с.