

УДК 519.71:519.72:004

**ПРОГРАММНО-ИНЖЕНЕРНАЯ РЕАЛИЗАЦИЯ  
МИКРОКОНТРОЛЛЕРНОГО УСТРОЙСТВА ДЛЯ УПРАВЛЕНИЯ ПРОЦЕССОМ  
ЗАРЯДА LI-ION БАТАРЕИ В УСЛОВИЯХ IOT**

*С.Ц. Манжикова, С.И. Гусев*

Приведены результаты программной реализации алгоритмов управления процессом заряда Li-Ion батареи в условиях IoT (Internet of things – интернета вещей) с использованием конкретных типов программируемых контроллеров и с учётом их технических характеристик и электронных параметров.

*Ключевые слова:* программируемый микроконтроллер; трансивер; интернет вещей; алгоритмы управления; сетевой протокол.

---

**SOFTWARE ENGINEERING IMPLEMENTATION OF MICROCONTROLLER  
DEVICES FOR PROCESS CONTROL CHARGE LI-ION BATTERIES IN CONDITIONS OF IOT**

*S.Ts. Manzhikova, S.I. Gusev*

The paper describes the results of software implementation, process control algorithms charge Li-Ion batteries under IoT (Internet of things - the things the Internet) using specific types of programmable controllers, and taking into account their technical characteristics and electronic parameters.

*Keywords:* programmable microcontroller; transceiver; internet of things; control algorithms; network protocol.

В работе [1] представлены результаты проектирования автоматической системы управления процессом зарядки литиевой батареи с применением технологии IoT. Всё более возрастающее практическое использование IoT требует производства специального оборудования [2]. Одной из важных задач при подключении к интернету вещей является подбор/поиск наиболее оптимального состава необходимых компонентов для реализации таких систем. Ниже приводятся технические и функциональные характеристики соответствующих устройств.

**Микроконтроллер** является основным блоком устройства [1]. Ядро микроконтроллера образуют блок процессора, объединяющий АЛУ с регистром признаков SREG и устройство управления, Flash-память программ объемом 8Кб, регистры общего назначения, память данных статического типа SRAM объемом 512 Кбайт.

Устройство управления представлено схемой синхронизации, регистром управления микроконтроллера MCUCR, генератором, а также регистром команд с дешифратором, программным счетчиком и указателем стека. Структурная схема используемого МК представлена на рисунке 1.

Между собой МК ATmega640/1280/1281/2560/2561 отличаются только размером памяти и количеством выводов. В частности, используемый в данной работе МК ATmega1281 имеет 128Кбайт flash-памяти, 4Кбайт EEPROM, 8Кбайт ОЗУ, 54 портов ввода/вывода, 2 серийных UART-интерфейса и 8 каналов АЦП.

Организация памяти МК AVR выполнена по Гарвардской архитектуре [3]. Память программ предназначена для хранения команд, управляющих функционированием МК. В памяти программ также хранятся различные константы, не меняющиеся во время работы программы. Память программ представляет собой электрически стираемое ППЗУ (FLASH-ПЗУ). Для адресации используется счетчик команд.

Память данных разделена на три части: регистровая память, оперативная память (статическое ОЗУ) и энергонезависимое EEPROM.

Другим не менее важным блоком является трансивер. Он управляет передачей полученных данных по интернет протоколу. Его основные функции:

- формирование прерывания для микроконтроллера;

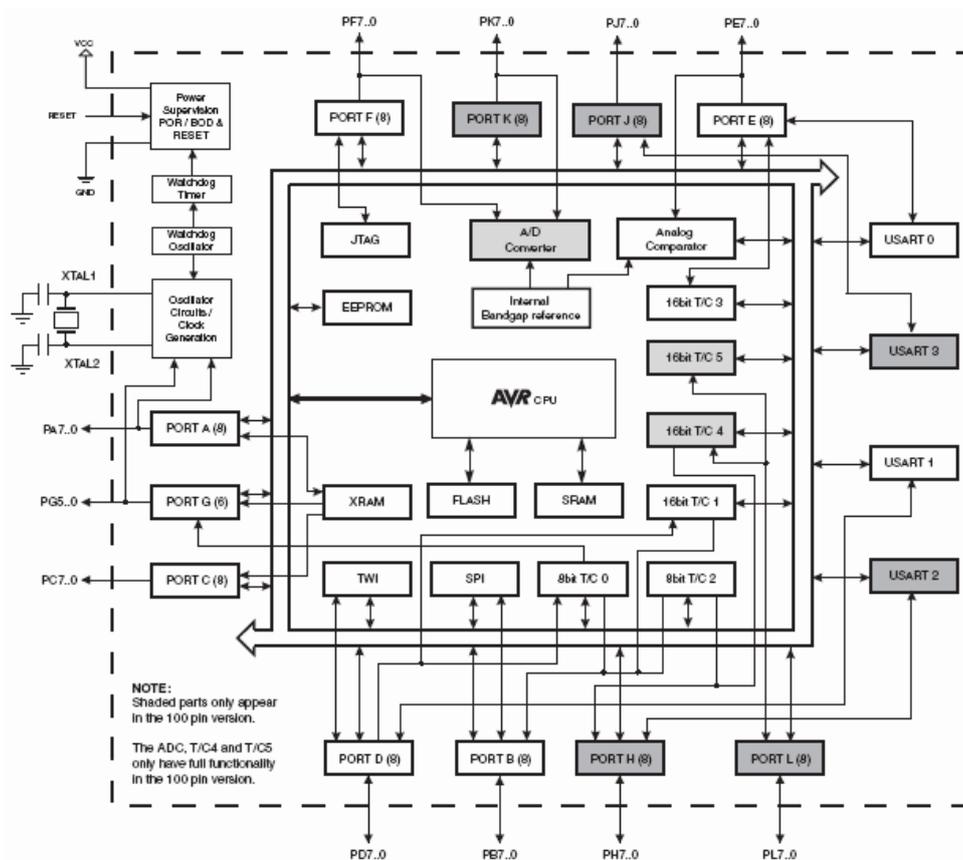


Рисунок 1 – Структурная схема МК ATmega1281

- пересылка данных на сервер/клиент;
- циклическая проверка целостности передаваемых данных.

Выбранный трансивер, разработанный фирмой TexasInstruments, работает в субгигагерцевом режиме, что позволяет использовать стек протокола 6LowPAN [4–6], благодаря чему трансивер потребляет относительно небольшую мощность [7]. Основная задача этого узла – работа со стеком протокола с помощью NanoStack 2.0. Таким образом, основная обработка информации проходит на МК ATmega1281, а CC1180 формирует прерывания из сети и отправляет информацию на сервер о состоянии батареи. Такая информация включает информацию о напряжении  $U_{\text{текущее}}$  на батарее и информацию о температуре  $t$  [1].

Для моделирования работы схемы были использованы выводы под интерфейсное подключение – UART\_RX и UART\_TX.

**Цифровой термометр** DS18B20 с программируемым разрешением [8] от 9 до 12-bit, которое может сохраняться в EEPROM памяти прибора.

DS18B20 обменивается данными по 1-Wire шине и при этом может быть как единственным устройством на линии, так и работать в группе. Все процессы на шине управляются центральным микропроцессором.

На рисунке 2 показана блок-схема цифрового термометра DS18B20. 64-битовый ROM запоминает уникальный последовательный код прибора. Оперативная память содержит 2-байтовый температурный регистр, который хранит значение температуры по окончании температурного преобразования; два однобайтовых регистра контроля температуры (триггерной схемы TH и TL), и регистр конфигурации. Регистр конфигурации позволяет пользователю устанавливать разрешающую способность цифрового преобразователя температуры к 9, 10, 11, или 12 битам – это и влияет на время конвертирования температуры. TH, TL и регистры конфигурации энергонезависимы (EEPROM), таким образом, они сохраняют данные, когда прибор выключен. DS18B20 использует исключительно 1-Wire протокол – при этом формируется соеди-

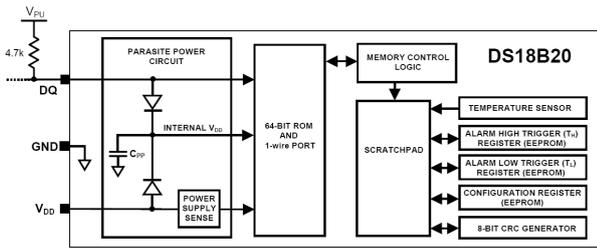


Рисунок 2 – Блок-схема цифрового термометра DS1820

нение, которое осуществляет коммуникацию на шине, используя всего один управляющий сигнал. Шина должна быть подключена к источнику питания через подтягивающий резистор. Так как все устройства связаны с шиной, используется соединение через Z-состояния или вход открытого стока. Используя эту шину, микропроцессор (устройство управления) идентифицирует датчики температуры и обращается к ним, используя 64-битовый код прибора. Поскольку каждый прибор имеет уникальный код, число приборов, к которым можно обратиться на одной шине, фактически не ограничено.

Другая особенность DS18B20 – способность работать без внешнего питания. Эта возможность предоставляется через подтягивающий резистор. Высокий сигнал шины заряжает внутренний конденсатор (CPP), который питает прибор, когда на шине низкий уровень. Этот метод носит название “Паразитное питание”. При этом максимальная измеряемая температура составляет +100 °С. Для расширения диапазона температур до +125 °С необходимо использовать внешнее питание.

Основные функциональные возможности DS18B20 определяет его температурный преобразователь. Разрешающая способность температурного преобразователя может быть изменена пользователем и составляет 9, 10, 11, или 12 битов, соответствуя приращениям (дискретности измерения) температуры 0,5 °С, 0,25 °С, 0,125 °С, и 0,0625 °С соответственно.

Подключение датчика к шине внешнего питания показано на рисунке 3.

АЦП – аналогово-цифровой преобразователь (ADC – Analog-to-Digital Converter) преобразует некий аналоговый сигнал в цифровой. Битность АЦП определяет точность преобразования сигнала. Время преобразования, соответственно, характеризует скорость работы АЦП. АЦП встроен во многих микроконтроллерах семейства AVR и упрощает использование микроконтроллера в различных схемах регулирования, где требуется оцифровывать аналоговый сигнал.

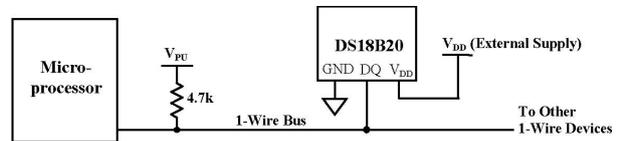


Рисунок 3 – Подключение температурного датчика DS1820 к микроконтроллеру

Из всех видов аккумуляторных батарей широкого применения Li-Ion и Li-Pol аккумуляторы выделяются самым высоким напряжением на ячейку (типичное значение 3,6 В) и самой высокой удельной емкостью (100–160 Вт\*час/кг).

Li-Ion аккумуляторы более требовательны к режиму заряда, теряют значительную часть ресурса при разряде ниже порогового значения и остаются пока более дорогими, чем NiCd или NiMH аккумуляторы. Кроме того, Li-Ion аккумуляторы оснащаются встроенными схемами защиты, что также сказывается на их стоимости. Наличие схем защиты актуально для предотвращения возможного взрыва от перегрева или образования газов, выделяющихся в ячейке при определенных условиях в процессе заряда.

**Условия заряда Li-Ion и Li-Pol аккумуляторов/батарей.** Рассмотрение этих условий позволило разработать алгоритм выработки управляющего воздействия (или просто управления) зарядом [1].

Одна из эксплуатационных проблем аккумуляторов многих типов – это чувствительность к температуре окружающей среды при заряде. Большинство разновидностей Li-Ion аккумуляторов допускают заряд током до 1С (С – номинальная емкость аккумулятора) при температуре от +5 до +45 °С. При температуре от 0 до +5 °С рекомендуется заряжать током 0,1 С. Заряд при отрицательной температуре не рекомендован. Оптимальная температура заряда аккумуляторов на основе лития составляет +15...+25 °С.

Процесс заряда разделяют на три этапа:

1. *Предварительный заряд.* Если аккумулятор глубоко разряжен, то величина зарядного тока на начальном этапе должна составлять 0,1 С. Типичное значение определения порога глубокой разряженности – 2,9 В. Некоторые модели аккумуляторов допускают и меньший уровень.
2. *Режим быстрого заряда постоянным током (основной режим).* Когда напряжение на ячейке превысит документированный нижний порог, заряд можно продолжить значительно

большей величиной тока. Заряд проводится при постоянной величине тока. Производители аккумуляторов рекомендуют ток зарядки 0,7С для быстрого заряда в простых зарядных устройствах. Аккумуляторы допускают заряд большей величиной тока (1.0...1.5С), но при этом необходимо контролировать температуру ячейки, чтобы не допустить ее перегрев с последующим выходом аккумулятора из строя. Напряжение на ячейке постепенно нарастает до номинального верхнего предела.

3. *Режим быстрого заряда постоянным напряжением до максимальной емкости.* Ток заряда при этом постепенно снижается. Погрешность величины напряжения на клеммах аккумуляторной батареи в процессе заряда постоянным напряжением не должна превышать  $\pm 50$  мВ.

**Программирование управляющего воздействия.** Этапы программирования можно разделить на два этапа: программирование МК ATmega1281 и трансивера CC1180. Каждый из них программируется отдельно как различные микроконтроллеры, с учетом соединения через последовательный канал UART.

При программировании МК, полученный в результате компиляции программы машинный код, загружается в память программ. Микроконтроллер поддерживает два режима программирования:

- режим параллельного программирования при высоком напряжении;
- режим программирования по последовательному каналу (рисунок 4).

Независимо от режима программирования FLASH-памяти осуществляется всегда побайтно.

Все модели микроконтроллеров поставляются со стертой памятью программ и памятью данных (во всех ячейках находится число \$FF) и пригодны к немедленному программированию.

Содержимое FLASH-памяти (памяти программ) может быть защищено от записи и/или чтения посредством программирования ячеек защиты (LockBits) LB1 и LB2. В исходном (незапрограммированном) состоянии в этих ячейках содержится

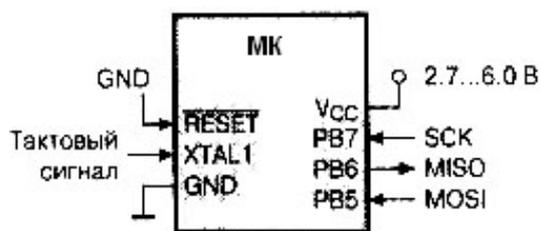


Рисунок 4 – Схема включения микросхемы в режиме программирования по последовательному каналу

“1”, после программирования – “0”. Стирание ячеек защиты (запись в них лог. “1”) может быть произведено только при выполнении команды “Стирание кристалла”, уничтожающей также содержимое FLASH-памяти. Программирование памяти программ и данных осуществляется побайтно путем отправки команд “Запись FLASH” (WriteFLASH) и “Запись EEPROM” (WriteEEPROM).

При записи в ячейку памяти следует быть уверенным, что программирование предыдущей ячейки завершилось. Для этого существует два способа. Первый (наиболее универсальный) способ – выдерживать паузу между отсылкой команд длительностью не меньше  $t_{WD\_PROG} = 4$  ms (при 5.0V). Недостаток этого способа – увеличение времени программирования.

Второй способ заключается в контролировании содержимого ячейки после отправки команды записи:

1. FLASH-память. До завершения записи ячейки при ее чтении возвращается значение “\$FF”, а после завершения – записанное значение;
2. EEPROM-память. Перед записью ячейки EEPROM-памяти автоматически выполняется стирание ее содержимого. Поэтому при чтении ячейки после отправки команды записи возвращаются значения: P1=\$00 – до завершения цикла стирания; P2=\$FF – после завершения цикла стирания; записанное значение – после завершения операции записи.

**Алгоритм программирования flash-памяти:**

1. Включение МК. К выводам VCC и GND прикладывается напряжение питания, в то время как на выводах SCK и RESET установлен низкий уровень.
2. Пауза >20 мс, затем передача команды “ProgrammingEnable”. Во время передачи третьего байта команды выполняется проверка эхо-ответа.
3. Flash-память записывается постранично. Размер страницы – 16 слов (32 байта). Программатор побайтно передает содержимое страницы в командах “LoadProgramMemoryPage” следующего формата: 0100 H000 xxxxxxxxxxxxxxxbbbbbiiiiiii, где “H”: 1 – записать старший байт, 0 – записать младший байт, “bbbb” – адрес слова внутри страницы, “iiiiiii” – данные, “x” – незначащие биты. При этом младший байт передается первым.
4. Когда страница будет передана целиком, программатор выдает команду “WriteProgramMemoryPage” формата: 0100 1100 xxxxxaaabbbbxxxxxxxxxxxx, где “aa” – наиболее значимые биты номера страницы, “bbbb” – наиболее значимые биты номера страницы.

5. Если необходимо записать несколько страниц, программатор должен сделать паузу не менее 4,5 мс. Это время необходимо для сохранения страницы во flash-памяти.
6. Производится проверка правильности записанной программы путем чтения содержимого flash-памяти и сравнения его с переданной программой.
7. По завершении программирования на выводе RESET устанавливается высокий уровень, что означает перевод МК в нормальный режим работы.  
Микроконтроллер ATmega1281 содержит в себе определенную программу, с помощью которой и происходит программирование МК. В режиме программирования повышенным напряжением на вход сброса (RESET) микроконтроллера подается напряжение 12В. Все AVR-микроконтроллеры поддерживают данный режим программирования.

#### *Литература*

1. Гусев С.И. Разработка микроконтроллерного программируемого устройства управления процессом заряда Li-Ion батареи в пространстве интернета вещей / С.И. Гусев, С.Ц. Манжикова // Вестник КPCУ. 2017. Т. 17. № 5.
2. Каталог беспроводного оборудования для интернета вещей.
3. URL: <http://www.euromobile.ru/iot-komponenty-dlya-interneta-veshhej/>
4. Гарвардская архитектура. URL: [https://ru.wikipedia.org/wiki/Гарвардская\\_архитектура](https://ru.wikipedia.org/wiki/Гарвардская_архитектура)
5. Mesh and IP Networks – 6LoWPAN – <http://www.ti.com/product/cc1180>
6. CC1180 Sub-1GHz 6LoWPAN Network – <http://www.ti.com/lit/gpn/cc1180>
7. ATmega1281 datasheet – <http://www.atmel.com/devices/atmega1281.aspx>
8. CC-6LoWPAN – Texas Instruments Embedded Processors – <http://processors.wiki.ti.com/index.php/CC-6LoWPAN>
9. Чернов Геннадий DS18B20 русское описание работы с датчиком температуры – [http://www.ugolok-mastera.ru/ds18b20-rus\\_pic\\_v2.pdf](http://www.ugolok-mastera.ru/ds18b20-rus_pic_v2.pdf)