Министерство образования Российской Федерации

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. БАУМАНА

Факультет «Информатика и системы управления»

Кафедра «Компьютерные системы и сети»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

курсовой работы

МК – система (ТАЙМЕР) для включения/выключения подключенных к нему электроприборов согласно заданному расписанию

Микропроцессорные системы

2005

# **Реферат**

РПЗ 30 с., 9 рис., 0 табл., 2 прил.

МИКРОСХЕМА, Таймер, ПОСЛЕДОВАТЕЛЬНЫЙ ИНТЕРФЕЙС, ЧАСТОТА, МИКРОКОНТРОЛЛЕР, COM-ПОРТ, РАСПИСАНИЕ, ПОСЛЕДОВАТЕЛЬНАЯ ПЕРЕДАЧА.

Объектом проектирования является разработка микроконтроллерной системы на основе AT90S8535 включения/выключения подключенных к нему электроприборов согласно заданному расписанию.

Цель работы – эскизный проект цифрового устройства ограниченной сложности.

Формирование одного из выходных сигналов происходит после совпадения текущего времени и времени в соответствующем расписании, заданном пользователем.

Устройство выполнено в виде одноплатного модуля, на который поступают сигналы из внешнего устройства через последовательный интерфейс. Формируемые выходные сигналы поступают от соответствующих выводов микроконтроллера (микроконтроллерной системы).

Материалы по курсовому проекту представлены в виде графической части и расчетно-пояснительной записки.

Задачи, решаемые в процессе проектирования: анализ альтернативных решений задачи; выбор схемотического решения и элементной базы; расчет потребляемой мощности устройства; разработка алгоритма управления и соответствующей программы микроконтроллера.

# **Содержание**

Реферат 2

Содержание 3

Введение 4

1. Анализ задания и Синтез структурно-функциональной схемы устройства 5

1.1 Анализ требований 5

1.2 Синтез структурно-функциональной схемы 6

1.2.1 Блок микроконтроллера 7

1.2.2 Блок коммутационных реле 8

1.2.3 Блок питания 8

2. Синтез принципиальной схемы устройства 10

2.1 Архитектура микроконтроллера 10

2.2 Выбор элементной базы 12

2.3 Тактирование системы 14

2.4 Подключение цепей питания 15

2.5 Устранение помех в цепях питания 15

3. Расчет потребляемой мощности 17

4. Описание алгоритмов программы 19

5. Отладка Программы для микроконтроллера 23

6. Программирование МК 24

Заключение 27

Список использованных источников 28

Приложение 1. 29

# **Введение**

Зачастую возникает ситуация, когда необходимо управлять некоторыми электроприборами не с помощью оператора, а по некоторому заранее заданному расписанию в соответствии с реальным временем. Очевидным решением этой задачи является использование таймера, на основе программируемого логического микроконтроллера. Это позволяет построить систему управления включением/выключением сразу нескольких подключенных к устройству приборов, что обеспечивает большую универсальность применения этого изделия.

Данную разработку можно использовать в качестве часов с будильником с возможностью задания нескольких расписаний, а также в качестве устройства управления различными устройствами, включение и выключение которых должны осуществляться в отсутствие оператора.

В область применения данной разработки входят все системы, в которых необходим автоматизированный запуск и выключение электрооборудования в заранее заданные моменты времени.

Разработанную микроконтроллерную систему можно представить как состоящую из двух частей:

* аппаратной части – представляющей собой коммутацию микроконтроллера, цепи блока питания и интерфейса приёма данных от внешнего устройства, обеспечивающего задание текущего времени, а также режимов работы каждого из подключенных электроприборов.
* программной части – представляющей собой программу, находящуюся в ПЗУ микроконтроллера и непосредственно выполняющую как счёт времени, так и запуск или отключение соответствующего электроприбора. Программа находится во FLASH-памяти микроконтроллера и функция изменения её пользователем не предусмотрена.

# **1.** **Анализ задания и Синтез структурно-функциональной схемы устройства**

## 1.1 Анализ требований

Согласно техническому заданию, необходимо разработать микроконтроллерную систему (таймер), осуществляющую включение и выключение подключенных к ней электроприборов согласно заданному пользователем расписанию. Загрузку расписаний необходимо осуществлять по последовательному каналу в формате ММ-ДД-ЧЧ-мм. Максимальное количество приборов, которые можно подключить, ограничено техническим заданием десятью.

Анализ требований к функциональности системы показывает, что разрабатываемая система представляет собой аппаратно-программный модуль, который является печатной платой, на которой находится микроконтроллер Atmel AT90S8535, осуществляющий счёт времени, сверку с ним загруженных расписаний и выдачу управляющих сигналов на реле включения/выключения соответствующего прибора. Программа микроконтроллера осуществляет обработку сигналов прерывания от таймера, вычисляя текущую дату и время в формате пригодном для сравнения с загруженным пользователем расписанием. При совпадении одного из загруженных расписаний с текущим временем осуществляется выдача соответствующего управляющего сигнала на один из выводов микроконтроллерной системы.

Пользователю предоставлены следующие функции:

* Корректировка даты 31->1. Это вызвано тем, что таймер считает, что каждый месяц содержит 31 день, поэтому, если в каком-либо месяце текущее число - 31-е, вместо 1-го, есть возможность простой командой это исправить.
* Установка даты-времени. Позволяет установить текущую дату и время.
* Установка расписаний. Позволяет для конкретного прибора установить дату и время его включения и выключения.

Основной режим работы микроконтроллера заключается в подсчете времени и сверки его с каждым заданным расписанием. Прерывание от таймера происходит каждую секунду, а от UART - при выборе пользователем одного из функций настройки устройства и подаче соответствующего сигнала через последовательный интерфейс.

Таким образом, при подаче на разработанное микроконтроллерное устройство питания, включается таймер и начинается счёт реального времени. Пользователю предоставляется возможность, выбрав соответствующий режим, установить или скорректировать текущую дату-время или задать расписание работы какого-либо прибора.

## 1.2 Синтез структурно-функциональной схемы

На рис. 1 приведена общая структурная схема функционирования программируемого логического микроконтроллера.

Рис. 1 Структурная схема программируемого логического микроконтроллера

Из рисунка 1 видно, что управление непосредственно включением и выключением электроприборов осуществляется через реле. Это позволяет осуществить управление электроприборами с питанием 220 В.

### **1.2.1 Блок микроконтроллера**

Данный блок представлен микроконтроллером Atmel AT90S8535, на вход РD0(RXD) которого поступают внешние управляющие сигналы от согласующего устройства, посредством которых задаётся режим работы микроконтроллера и устанавливается время и расписания. Также на микроконтроллер поступает питающее напряжение от блока питания.

При совпадении текущего времени и одного из времен в одном из заданных расписаний микроконтроллер выдаёт сигнал высокого (для включения электроприбора, при совпадении текущего времени с временем включения данного прибора в расписании) или низкого (для выключения электроприбора, при совпадении текущего времени с временем выключения данного прибора в расписании) уровня с соответствующего бита порта (PA0..PA7 для приборов с 1-го по 8-й соответственно и PB0, PB1 для 9-го и 10-го приборов соответственно).

После подачи пользователем сигнала выбора режима работы микроконтроллера на последовательный интерфейс, микроконтроллер переходит в режим ожидания ввода остальных данных соответствующих данному режиму (для режимов установки времени и установки расписания) или непосредственно выполняет подпрограмму для данного режима (режим коррекции времени).

При выборе пользователем режима коррекции времени (путем подачи на UART двоичного кода 0000 0001), микроконтроллер осуществляет увеличение текущего числа на единицу, при необходимости с соответствующим изменением текущего месяца, и возвращается в обычный режим работы.

При выборе режима установки времени (путем подачи на UART двоичного кода 0000 0010), микроконтроллер переходит в режим ожидания ввода нового значения месяца. После ввода, это значение становится текущим, а микроконтроллер ожидает ввода дня. Когда будет принято и установлено значение минут, система переходит в обычный режим работы.

При выборе режима установки расписания (путем подачи на UART двоичного кода 0000 0011), микроконтроллер ожидает ввода номера прибора, для которого будет задаваться расписание. После выбора прибора пользователь должен ввести дату и время включения выбранного прибора в формате ММ-ДД-ЧЧ-мм, а затем дату и время выключения данного прибора в том же формате. После ввода система переходит в обычный режим работы.

### **1.2.2 Блок коммутационных реле**

Блок коммутационных реле осуществляет сообщение портов МК и подключенных к нему электроприборов с целью обеспечить при необходимости подачу на прибор питания 220 В, 50 Гц.

При совпадении текущего времени с одним из расписаний, с одного из выводов микроконтроллера поступает сигнал на реле, соответствующее прибору, для которого сработало расписание. При поступлении сигнала, реле замыкает (для единичного сигнала включения прибора) или размыкает (для нулевого сигнала выключения прибора) цепь питания электроприбора в соответствии с типом сработавшего расписания.

### **1.2.3 Блок питания**

Данный блок представлен электротехническим устройством, осуществляющим преобразование входного переменного напряжения 220 В в постоянное напряжение 5 В, необходимое для питания входящих в состав системы микросхем.

Также в этом блоке содержится устройство бесперебойного питания, позволяющее подключить резервную аккумуляторную батарею, в случае отключения основного питания, для поддержания работы таймера. Но, очевидно, включенные приборы заработают только при подаче основного напряжения.

# **2. Синтез принципиальной схемы устройства**

## 2.1 Архитектура микроконтроллера

Основным узлом данного устройства является микроконтроллер AT90S8535 фирмы Atmel. Графическое обозначение микроконтроллера приведено на рис 2. AT90S8535 является маломощным, высокопроизводительным 8-ми разрядным микроконтроллером, изготовленным по технологии КМОП на основе AVR RISC архитектуры. МК спроектирован по новейшей технологии Atmel с высокой плотностью интеграции, а также удобной системой команд и набором выводов. В состав однокристального МК AT90S8535 входит:

* внутреннее ЗУ SRAM объемом 512 байт;

* 32 программируемых линий ввода/вывода;
* два 8-разрядных и один 16-разрядный таймер-счетчик;
* канал последовательного ввода/вывода;
* часы реального времени;
* 8Кб программируемая Flash память (до 1000 циклов перезаписи);

А также множество других полезных схем.

Наряду с высоким быстродействием, микроконтроллер имеет встроенную энергонезависимую память EEPROM.

Структурная схема микроконтроллера приведена на рис. 3

Все 4 порта являются двунаправленными. Один из 8–разрядных счётчиков позволяет организовать часы реального времени. EEPROM – энергонезависимая память, в которой будут хранится расписания. UART – последовательный интерфейс, что позволяет передавать сигналы на МК через COM-порт (RS-232).

Таким образом, выбранный микроконтроллер обеспечивает весь набор ресурсов, необходимых для работы системы. А дополнительные возможности (Flash память), предоставляемые микроконтроллерами семейства Atmel, будут использоваться для хранения кода исполнительной программы (программной части микроконтроллерной системы).

Для нормального функционирования системы задействованы 2 порта ввода/вывода микроконтроллера, плюс специальная функция порта D (обмен данных с UART). РA и РB задействованы для реализации выдачи сигналов управления электроприборами на соответствующие реле.

## 2.2 Выбор элементной базы

Для согласования уровней напряжений между COM-портом и микроконтроллером использовалась микросхема MAXIM MAX233, предназначенная специально для такого согласования и имеющая встроенные средства для такого преобразования уровней. Работая от стандартного напряжения 5 В, она преобразует его в +14/-14 В, необходимые интерфейсу RS-232.

MAX233 – универсальный приемопередатчик, предназначенный для сопряжения микропроцессорных элементов, для работы которых требуется напряжение 5 В и устройств с протоколом, имеющим опорное напряжение +14/-14 В. Для питания этой микросхемы необходимы те же 5 В, от которых питается микроконтроллер AT90S8535, что во многом обусловило наш выбор именно этого прибора. Его мощность составляет 890 мВт.

Типовая схема его включения показана на рисунке 4. Микросхема имеет несколько каналов согласования, но нам необходим только один. Этим обусловлены дополнительные перемычки выводов схемы.

Выводы R1Out и R1In соответственно выход и вход информационного сигнала, для которого осуществляется согласование уровней.

Для других участков схемы при выборе элементов использовались такие критерии как достаточное быстродействие, низкая потребляемая мощность.

В реле управления электроприборами использованы транзисторы КТ315, обладающие подходящей нагрузочной способностью по сравнению с другими сериями. К световым диодам особых требований не предъявлялось, поэтому были применены диоды АЛ307 (для индикации состояния каждого прибора (включен/выключен)).

В качестве источника резервного питания использованы 3 гальванические батареи напряжением по 1,5 В.

Также в блоке питания присутствует стабилизатор напряжения L7805 для сглаживания импульсов напряжения после диодного мостика, выбор которого обуславливается высокой точностью и качеством стабилизации (сглаживания), а также низкой потребляемой мощностью и отсутствием необходимости в каких-либо дополнительных электро-радио элементах, кроме сглаживающих конденсаторов. Схема его включения показана на рисунке 5. Последние две цифры обозначения показывают выходное напряжение прибора (5 В).

В реле были использованы тиристоры КУ208Г. Это кремниевые тиристоры структуры p-n-p-n, предназначенные для работы в качестве переключающих элементов средней мощности для коммутации цепей силовой электроники на переменном токе. Приборы обладают достаточным быстродействием (время включения не более 10 мс), низким сопротивлением в открытом состоянии по сравнению с другими подобными элементами.

## 2.3 Тактирование системы

Существует несколько способов задания тактовой частоты микроконтроллера AT90S8535, каждый из которых имеет свои достоинства и недостатки. Можно например использовать кварцевый резонатор или подавать тактовые импульсы от внешнего генератора. В данном курсовом проекте использован кварцевый резонатор. Этот способ позволяет очень точно задать тактовую частоту микроконтроллера (разброс частот обычно составляет не более 0.01%). Номиналы емкостей конденсаторов в данной схеме определяются производителем микроконтроллера для конкретной резонансной частоты кварца. Для проектируемого устройства был использован кварц – 4 МГц, и емкости конденсаторов – 0,1 мкФ, что является приемлемым для используемого типа микроконтроллера.

Другие способы синхронизации – использование RC-генератора. Это самый дешевый способ задания частоты, но наименее точный. И еще один способ – подача тактовых импульсов от внешнего генератора. При помощи внешнего тактового генератора можно задать любую частоту синхронизации.

В целом работа проектируемого устройства зависит не столько от тактовой частоты работы микропроцессора, сколько от частоты работы счетчика, организованного в качестве часов реального времени, с помощью подключения кварцевого резонатора частотой 32.768 КГц. Это частота рекомендована производителем микроконтроллера для организации на данном счётчике часов реального времени.

## 2.4 Подключение цепей питания

AT90S8535 имеет три источника сброса:

* сброс по включению питания. Процессор сбрасывается при подаче питания на выводы VCC и GND;
* внешний сброс. Процессор сбрасывается при подаче низкого уровня на вывод RESET на время более двух периодов тактовой частоты;
* сброс от сторожевого таймера.

Специальная схема, встроенная в микроконтроллер, - цепь сброса по включению питания, обеспечивает запрет включения процессора до тех пор, пока напряжение питания не достигнет безопасного уровня. Внешний сброс обрабатывается по низкому уровню на выводе. После снятия сигнала 0 с вывода RESET микроконтроллер запускается через некоторое время, как и в случае подачи питания.

## 2.5 Устранение помех в цепях питания

Для уменьшения наводок и скачков напряжений, связанных как с переходными процессами в шинах питания и земли, так и с работой диодного мостика применены 4 сглаживающих конденсатора различной ёмкости, а также стабилизатор напряжения, сглаживающий импульсы, для обеспечения более стабильного питания микроконтроллерной системы. Конденсаторы подключаются между шинами питания и земли и устанавливаются в непосредственной близости от обслуживаемых корпусов.

Для предотвращения сгорания электроприборов или компонентов блока питания вследствие чрезмерного увеличения тока в высоковольтной магистрали применены два плавких предохранителя: один в цепи питания приборов (на 5А), и второй в цепи блока питания (на 1А).

# **3.** **Расчет потребляемой мощности**

Суммарная максимальная мощность, потребляемая устройством РΣmax, может быть вычислена по следующей формуле:

РΣМАХ = РΣИС + РΣЭРЭ , (1)

 где РΣИС – суммарная мощность, потребляемая интегральными микросхемами; РΣЭРЭ – суммарная мощность, рассеиваемая электрорадиоэлементами.

Мощность, потребляемая интегральными микросхемами:

РΣИС = РΣИСстат + РΣИСдин , (2)

где РΣИСстат – статическая мощность;

РΣИСдин – динамическая мощность.

Для проектируемого устройства составляющие суммарной потребляемой мощности будут считаться по следующим образом:

Мощность, рассеиваемая ЭРЭ (резисторы имеют фиксированную рассеиваемую мощность):

РΣR = 20 шт \* 0,125 Вт = 2,5 Вт

Мощность, потребляемая МС является суммой потребляемых мощностей в статическом и динамических режимах.

Согласно документации ток потребляемый в активном динамическом режиме (при частоте 4 МГц) составляет около 12 мА. При напряжении 5 В потребляемая мощность = 60 мВт.

В статическом режиме каждый порт выдаёт ток около 3 мА. Т.о. суммарная рассеиваемая мощность составляет 150 мВт.

 Р∑имс = РL7805 + РMAX233 + РАТ90S8535

Р∑имс = 30 + 890 + 210 = 1,130 Вт;

Данный расчёт является приблизительным, поскольку работа схемы осуществляется в асинхронном режиме, и в каждый момент времени существует большое количество неактивных цепей.

Суммарная максимальная мощность, потребляемая микроконтроллерной системой:

Рмах = Р∑имс + РR = 3,63 Вт

# **4. Описание алгоритмов программы**

Общая схема алгоритма приведена на рисунке 4.

Программа для разрабатываемой системы состоит из нескольких основных частей:

* Блок начальной инициализации системы;
* Блок обработчиков прерываний;
* Блок обработки режимов микроконтроллера;
* Блок служебных подпрограмм.

Рассмотрим эти алгоритмы более подробно.

Блок начальной инициализации системы содержит операции для программирования таймера, UART-интерфейса и других различных настроек.

Инициализация UART заключается в установке скорости его работы в значение 9600 бит/с путём загрузки в регистр UBRR значения 25, а также в настройке его на чтение (а не на запись), что осуществляется записью в UCR значения 144.

Инициализация таймера заключается в задании режима его работы от внешнего резонатора (а не от внутреннего генератора частоты), разрешении прерываний таймера (по переполнению). Далее осуществляется запуск таймера.

Блок обработчиков прерываний обслуживает возникающие в системе прерывания в соответствии с заложенным алгоритмом. Выделено два обработчика прерываний: от таймера и от UART-интерфейса.

Обработчик прерывания UART осуществляет запоминание введённой пользователем по последовательному интерфейсу команды в одном из внутренних регистров процессора, что позволяет произвести её обработку в подходящий момент времени, не нарушая работы канала передачи.

Обработчик прерывания таймера осуществляет корректировку текущего времени, увеличивая счётчик секунд, минут, часов, дней и месяцев. Одна секунда отсчитывается таймером (частота работы которого задаётся от внешнего резонатора частотой 32768 Гц) за 32768 цикла. Использовать делители частоты при работе от внешнего резонатора, в отличие от работы от внутреннего генератора, нельзя, а частота кварца 32768 Гц рекомендована производителем МК для реализации часов реального времени.

Так как таймер 8-ми разрядный, то у нас есть возможность досчитать только до 256, таким образом, необходимо организовать ещё дополнительный множитель 128, что и организовано в регистре R28.

По прошествии одной секунды увеличивается значение секунд в регистре R27, а затем полученное значение сравнивается с 60-ю. При получении равенства, счётчик секунд обнуляется, а счётчик минут инкрементируется, с последующим сравнением количества минут с 60-ю и возможной инкрементацией счётчика часов. И так продолжается до тех пор, пока все значения (месяца, дня, часа, минуты, секунды) не будут посчитаны.

Блок обработки режимов микроконтроллера содержит три основные подпрограммы для каждого из возможных режимов работы микроконтроллера: режима корректировки времени, режима установки времени и режима установки расписания.

Режим корректировки времени осуществляет увеличение счётчика дней на 1, с последующей проверкой на достижение 32-х с возможной корректировкой месяца. Это может использоваться пользователем, когда прошлый месяц содержал только 30 дней, и первого числа следующего месяца часы показывают 31-е (так нет разграничения на 30-ти и 31-но дневные месяцы). Тогда выбором этого режима пользователь легко корректирует дату.

Режим установки времени позволяет установить текущие значения даты и времени, без проверки их правильности. После выбора этого режима программа ожидает от пользователя ввода текущего месяца, затем дня, часа и минуты. После выполнения всех действий и вступления изменений в силу, программа возвращается в нормальный режим работы.

Режим установки расписания позволяет пользователю задать дату и время включения и выключения выбранного им прибора. После выбора этого режима микроконтроллер ожидает ввода номера прибора, для которого будет задаваться расписание. После выбора прибора пользователь должен ввести дату и время включения выбранного прибора в формате ММ-ДД-ЧЧ-мм, а затем дату и время выключения данного прибора в том же формате. После ввода система переходит в обычный режим работы.

Блок служебных подпрограмм представляет собой остальные подпрограммы, необходимые для работы. К ним относятся подпрограмма предварительной очистки EEPROM (хотя её можно было отнести к блоку инициализации), подпрограмма ожидания (основной режим), а также подпрограмма, осуществляющая включение или выключение приборов при совпадении текущего времени с каким-либо расписанием.

После запуска прибора выполняется блок инициализации программы, осуществляющий загрузку начальных значений и установку режимов работы устройств микроконтроллера. Также производится запуск таймера.

Далее программа переходит в основной режим своей работы, в течение которого вызывается подпрограмма проверки расписания, осуществляющая сравнение текущей даты-времени со всеми значениями расписания в памяти и вызов при необходимости подпрограммы активации/деактивации подключенных устройств. Также основная программа постоянно опрашивает значение регистра, работающего с UART для обнаружения факта задания одного из режимов работы МК. При обнаружении этого факта управление передаётся подпрограмме выявления номера режима, и далее на обработчик выбранного режима.

Если режим не задан, то основная программа вновь осуществляет проверку расписания. Это необходимо делать довольно часто в связи с достаточно большим объемом программного кода, осуществляющего эту проверку, а также в связи с достаточно большим значением самих проверяемы значений (80 байт).

При подаче пользователем через последовательный интерфейс сигнала о выборе режима, генерируется прерывание от UART, что впоследствии быстро выявляется в основной программе.

# **5. Отладка Программы для микроконтроллера**

Успешная трансляция ещё не означает, что код не содержит ошибок. Она означает, что нет ошибок синтаксического плана, но сам алгоритм может работать совершенно неверно. Для его отладки необходимо проверить все его ветви и возможные переходы. Программа была отлажена с помощью Win32 AVR Studio 4. Листинг программы представлен в приложении 1.

Изначально проектировалась секция инициализации, и основная часть – режим ожидания. Запуск на выполнение позволял полностью отладить программу.

При проектировании обработчика прерывания таймера, как собственно и остальных участков программы отладка производилась в пошаговом режиме с постоянным контролем содержимого регистров портов и т.п. Иногда было необходимо вставлять дополнительные операторы, для выявления некоторых ситуаций (чаще всего это операторы присвоения). Такой подход позволял полностью выявить ошибки в коде.

При разработке подпрограммы работы с UART и обработки режимов микроконтроллера, возникла необходимость моделирования внешнего устройства (COM-порта), по которому пользователь будет посылать управляющие сигналы на МК. В этом случае производилась отладка кода до момента непосредственно ввода пользовательских данных (в пошаговом режиме или режиме «запуск до курсора), затем вручную менялось содержимое какого-либо порта или регистра и затем выполнение продолжалось далее в пошаговом режиме. Такой способ позволил довольно легко отладить участки кода, где требовалось непосредственное вмешательство пользователя в содержимое регистров и портов микроконтроллера.

# **6. Программирование МК**

МК AT90S8535 предоставляет 8 Кб перепрограммируемой FLASH – памяти. Для программирования может использоваться высоковольтный параллельный или низковольтный последовательный режимы программирования. Первый способ используется для программирования памяти с невозможностью её последующего изменения пользователем. Второй же предоставляет возможность загружать программу пользователем. Каждый метод предусматривает свой алгоритм загрузки программы, каждый из которых подробно рассмотрен в документации на микроконтроллер.

Параллельное программирование.

Схема для осуществления параллельного программирования изображена на рисунке 7. Сигналы:

 - 1 – Устройство готово к программированию

 - 0 - Разрешение вывода

 - Импульс записи

*BS -* Выбор байта (1 – старший)

XA0, XA1 - Выбор режима программирования

Data - Данные

На рисунке 8 довольно понятно показан алгоритм программирования FLASH-памяти в виде последовательности подаваемых сигналов.

Каждое действие инициируется сигналом XTAL1. Сначала в МК загружается команда записи. Затем младшая и старшая части адреса для загружаемых данных. Затем осуществляется сама загрузка данных, а затем подачей сигнала WR осуществляется непосредственно запись данных в память.

Последовательное программирование.

Схема для последовательного программирования изображена на рисунке 9.

Запись данных осуществляется по положительному фронту сигнала SCK. Инструкции загружаются сигналом MOSI. Любые загруженные данные можно посмотреть с выхода MISO, после соответствующей загруженной команды.

# **Заключение**

В результате проектирования была разработана МК – система (ТАЙМЕР) для включения/выключения подключенных к нему электроприборов согласно заданному расписанию.

Разработанное устройство, благодаря использованию современной элементной базы, отличается высокой точностью, надежностью и быстродействием. Высокая степень интеграции используемых радиоэлементных средств обеспечивает уменьшение массогабаритных параметров и потребляемой мощности.

# **Список использованных источников**

1. Хартов В.Я., Проектирование и отладка программ для микроконтроллеров AVR фирмы ATMEL: учебное пособие по курсу «микропроцессорные системы». Москва, изд-во «МГТУ им. Баумана», 2004. – 62 с.
2. Голубцов М.С., Кириченкова А.В. Микроконтроллеры AVR: от простого к сложному. Москва «СОЛОН-Пресс», 2005. – 304 с.
3. u208.pdf http://www.alkon.net Тиристоры серии у208, описание
4. L7800.pdf http://www.st.com Стабилизаторы L78хх, техническое описание
5. AT90S8535.pdf http://www.atmel.com 8-разрядный AVR МК AT90S8535
6. DOCCOM.pdf http://www.atmel.com Полудуплексное управление UART
7. http://www.atmel.ru/Articles/Atmel11.htm Справочник по ассемблеру для AVR
8. MAX233xPP.pdf http://www.maxim-ic.com MAX233xPP, техническое описание

# Приложение 1.

Листинг программы для МК

.include "D:\AVR Tools\AvrAssembler\Appnotes\8535def.inc"

.cseg

.def mes=r17

.def den=r18

.def chas=r19

.def min=r20

.def sek=r27

; R16 - временный регистр

; R21 - регистр работы с UART

; R22 - указатель на нужное расписание в EEPROM

; R23 - для флагов

; R24 - указатель на текущий приборорежим

; R25 - значение из памяти

; R26 - флаг выводного порта (N прибора > 8)

;-------------

.org $000

rjmp main

.org $004

rjmp T2ovf

.org $00b

rjmp uart\_dre

;-------------

uart\_dre: ; Обр-к прерывания UART

in r21, udr ; --Задай UDR--

reti

;-------------

T2ovf: ; Обр-к прерывания таймера (минуты)

In r23, sreg

push r23

inc r28

cpi r28, 0x80

brne goback

ldi r28, 0

inc sek

cpi sek, 60

brlo goback

ldi sek, 0 ; Корректировка переходов

inc min

cpi min, 60

brlo goback

ldi min, 0

inc chas

cpi chas, 24

brlo goback

ldi chas, 0

inc den

cpi den, 32

brlo goback

ldi den, 1

inc mes

cpi mes, 13

brlo goback

ldi mes, 1

goback:

pop r23

out sreg, r23

reti

;-------------

main:

;Инициализация

ldi r16, 0xdf

out spl, r16 ; Стек

rcall clear ; Очистка EEPROM

ldi mes, 1 ; Начальное время

ldi den, 1

ldi chas, 0

ldi min, 0

ldi sek, 0

ldi r28, 0

ldi r21, 0

ldi r24, 0

sei ; Разрешение прерываний

ldi r16, 25

out ubrr, r16 ; Скорость UART = 9600

ldi r16, 8

out assr, r16 ; Таймер - от кристалла

ldi r16, 0xff

out ddra, r16 ; Порт А на вывод

ldi r16, 3

out ddrb, r16 ; Биты 0 и 1 порта Б на вывод

ldi r16, 144

out ucr, r16 ; UART на чтение

ldi r16, 64

out timsk, r16 ; Разрешение прерывания таймера

ldi r16, 1

out tccr2, r16 ; Настройка и запуск таймера

;Работа

Wait: ; Цикл ожидания

rcall check

cpi r21, 0 ; Задали ли режим настройки

breq wait ; --Установи USR->RXC--

cpi r21, 1

breq rej1 ; Режим корректировки даты (31->1)

cpi r21, 2

breq rej2 ; Режим задания времени

cpi r21, 3

breq rej3 ; Режим задания расписания

rjmp wait

;------------

Rej1: ; Подпрога корректировки 31->1

ldi r21, 0

inc den

cpi den, 32 ; Корректировка переходов

brlo wait

ldi den, 1

inc mes

cpi mes, 13

brlo wait

ldi mes, 1

rjmp wait

;------------

Rej2: ; Подпрога установки времени

ldi r21, 70

mes2: ; Ожидание приёма месяца

cpi r21, 70

breq mes2

mov mes, r21

ldi r21, 70

den2: ; Ожидание приёма дня

cpi r21, 70

breq den2

mov den, r21

ldi r21, 70

chas2: ; Ожидание приёма часа

cpi r21, 70

breq chas2

mov chas, r21

ldi r21, 70

min2: ; Ожидание приёма минут

cpi r21, 70

breq min2

mov min, r21

ldi r21, 0

rjmp wait

;-------------

Rej3: ; Подпрога установки расписания вкл/выкл

ldi r21, 70

nom3: ; Ожидание приёма номера прибора

cpi r21, 70

breq nom3

mov r22, r21

subi r22, 1

lsl r22 ; Умножение на 8 (3 строки)

lsl r22

lsl r22

out eearl, r22 ; Указатель на нужный прибор

;----------

ldi r16, 1 ; Счётчик записи расписания

ldi r21, 70

vvod: ; Ожидание ввода элемента

cpi r21, 70

breq vvod

out eedr, r21 ; Значение - в регистр данных

ldi r21, 6

out eecr, r21 ; Записать в EEPROM

inc r22

ldi r21, 0

out eecr, r21

out eearl, r22 ; Увеличить указатель

ldi r21, 70

inc r16 ; Увеличить счётчик

cpi r16, 9 ; Оба ли расписания записаны

brne vvod

ldi r21, 0

rjmp wait

;----------

check: ; Подпрога проверки расписаний

cpi r24, 80 ; Последний ли прибор

brlo cont

ldi r24, 0 ; Сначала

cont: ; Проверка расписания прибора

rcall vvod1

cp r25, mes

brne back3

rcall vvod1

cp r25, den

brne back2

rcall vvod1

cp r25, chas

brne back1

rcall vvod1

cp r25, min

brne back0

RCALL TRY ; Звонок!!!

ret

vvod1: ; Подпрога чтения значений из памяти

out eearl, r24

ldi r16, 1

out eecr, r16

in r25, eedr ; В R25 значение

inc r24

ret

back3: ; Блок корректировки на след. прибор

inc r24

back2:

inc r24

back1:

inc r24

back0:

ret

; ПРОЦЕДУРА ЗВОНОК

TRY:

push r24

subi r24, 4

ldi r26, 0 ; Приборы 1-8

cpi r24, 0x40

brlo prta

subi r24, 0x40 ; Приборы 9 или 10

ldi r26, 0x0f

prta:

lsr r24 ; Корректировка адреса

lsr r24

lsr r24

brlo vikl ; Вкл/выкл?

rcall vibor

cpi r26, 0 ; Выбор порта

breq prta1

in r24, portb ; Вкл порт Б

or r24, r16

out portb, r24

pop r24

ret

prta1: ; Вкл порт А

in r24, porta

or r24, r16

out porta, r24

pop r24

ret

vikl:

rcall vibor

cpi r26, 0 ; Выбор порта

breq prta2

in r24, portb ; Выкл порт Б

com r16

and r24, r16

out portb, r24

pop r24

ret

prta2: ; Выкл порт А

in r24, porta

com r16

and r24, r16

out porta, r24

pop r24

ret

;Процедура выбора пина порта

vibor:

ldi r16, 1

cikl:

cpi r24, 0

breq next

dec r24

lsl r16

rjmp cikl

next:

ret

; Процедура очистки области расписаний EEPROM

clear:

ldi r16, 0

out eedr, r16

ldi r21, 6

cl1:

out eearl, r16

out eecr, r21

inc r16

cpi r16, 80

brne cl1

ret