

Федеральное агентство по образованию
Южно-Уральский Государственный Университет

МИКРОПРОЦЕССОРНЫЕ СРЕСТВА В ЭЛЕКТРОПРИВОДЕ
Методические указания к курсовому проектированию

Микропроцессоры в электроприводе : Методические указания к курсовому проектированию. –

Описана последовательность проектирования систем на AVR-микроконтроллерах фирмы Atmel Приведены задания на проектирование, требования к оформлению проекта. Для студентов специальности 140604 «Электропривод и автоматика промышленных установок», изучающих курс «Микропроцессорные средства в электроприводе»

Введение

Проектирование микропроцессорных систем – многоэтапный процесс, результат которого сильно зависит от индивидуальных особенностей разработчика. Процесс проектирования требует самых разных навыков разработчика в области цифровой схемотехники, программирования, конструирования. Его очень трудно формализовать и в нем ярче всего проявляются как способности студента, так и его желания.

Задания на курсовое проектирования очень похожи, но индивидуальны.

В соответствии с вариантом задания в курсовом проекте необходимо:

- разработать структурную схему микропроцессорной системы,
- разработать алгоритм работы системы,
- разработать программы на языке ассемблера,

Несмотря на сходство заданий, результаты проектирования всегда очень различны. Микропроцессорную систему можно строить на различной элементной базе, от которой сильно зависит результат. Задание можно усложнять в соответствии с пожеланиями разработчика.

За основу проектируемой микропроцессорной системы рекомендуется взять AVR-микроконтроллер фирмы Atmel. Подробное описание микроконтроллеров в работе [1], все необходимые для проектирования сведения можно найти на русскоязычном сайте www.atmel.ru. В принципе возможно использование и изделий других производителей.

1. Задание на курсовой проект

Объектом разработки является микропроцессорная система измерения скорости вращения электропривода, снабженного импульсным датчиком скорости. Примерная структурная схема системы приведена на рис. 1.1.

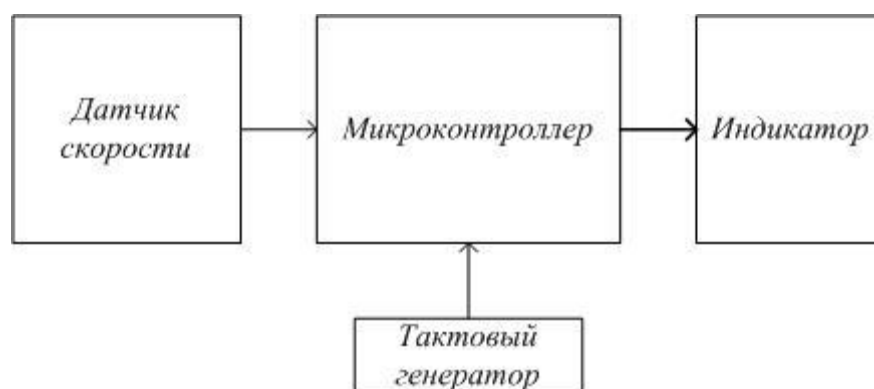


Рис. 1.1. Структура измерителя скорости вращения

В системе сигналы с датчика скорости поступают на микроконтроллер. Микроконтроллер обрабатывает сигналы, вычисляет скорость вращения и результаты вычислений выводит на индикатор. Тактовый генератор обеспечивает работу микроконтроллера.

В схеме используется оптоэлектронный датчик угловой скорости, называемый шифратором приращений или энкодером (*encoder*). В таком шифраторе в корпусе на валу крепится прозрачный стеклянный или пластиковый кодирующий диск на котором, в зависимости от типа датчика, нанесены одна, две или три дорожки непрозрачных меток (рис. 1.2).

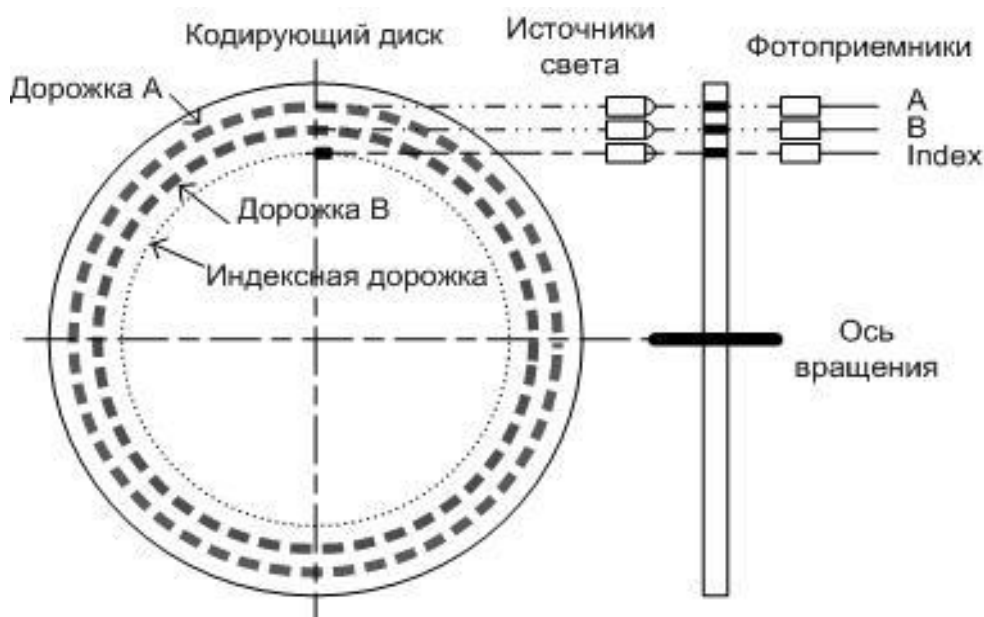


Рис. 1.2. Шифратор приращений

На три concentрические окружности диска нанесены метки. Число меток на внешних окружностях А и В равно. Это число k называется разрешающей способностью датчика. На третьей окружности I нанесена только одна индексная метка, фиксирующая точку отсчета. При вращении диска датчик формирует две сдвинутые по фазе на 90° периодические импульсные последовательности, соответствующие рядам меток А и В, и так называемый нулевой импульс, соответствующий индексной метке I. Сдвиг импульсных последовательностей А и В зависит от направления вращения. При движении по часовой стрелке сигнал А отстает от сигнала В, а при движении против часовой стрелки канал А сигнал А опережает сигнал В (рис.1.3).

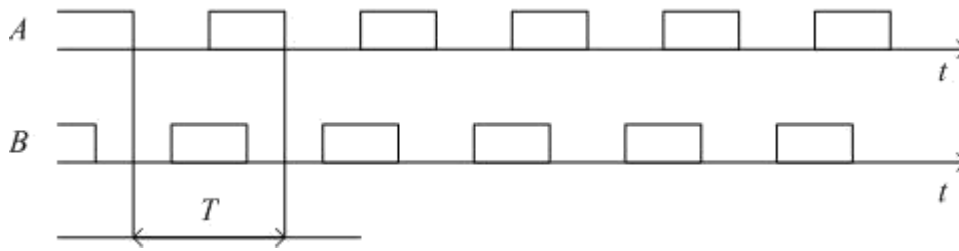


Рис. 1.3. Диаграммы сигналов на выходе импульсного датчика скорости

Частота f и период T следования импульсов на выходе датчика однозначно связана со скоростью вращения вала:

$$f = 1/T = n \cdot k / 60 ,$$

где n – скорость вращения вала (оборот/мин).

Знак скорости можно определить по фазовому сдвигу последовательностей А и В:

- если сигнал А опережает сигнал В, то скорость считается положительной,
- если сигнал А отстает от сигнала В, то скорость считается отрицательной.

Для определения знака скорости можно, например, фиксировать состояние сигнала В в момент появления переднего фронта сигнала А. Тогда при $V=1$ (рис.1.2) перемещение отрицательно, а при $V=0$ – положительно.

Варианты заданий на курсовое проектирование приведены в табл. 1.1. В каждом варианте определены все исходные данные для разработки:

- максимальная частота вращения вала – n (об/мин),
- единицы измерения частоты вращения (об/мин, об/с или рад/с),
- разрешающая способность датчика – k (имп/об),
- частота колебаний опорного кварцевого генератора – f_0 , (МГц),

Конкретное задание на проектирование формируется студентом в зависимости от уровня сложности решаемой задачи:

- 1 уровень. Система обрабатывает сигналы датчика, оценивает знак и величину скорости, результат формируется в двоичном коде и хранится в памяти данных микроконтроллера.
- 2 уровень. Система обрабатывает сигналы датчика, оценивает знак и величину скорости, результаты преобразуются в двоично-десятичный код и хранятся в памяти микроконтроллера.
- 3 уровень. Система обрабатывает сигналы датчика, оценивает знак и величину скорости, результаты выводятся на индикатор.
- 4 уровень. Система обрабатывает сигналы датчика, оценивает знак и величину скорости, результаты в двоично-десятичном коде по стандартному интерфейсу передаются на систему сбора информации и выводятся на индикатор.
- 5 уровень. Универсальная система. Параметры датчика программируются с цифровой клавиатуры.

Варианты заданий на курсовое проектирование

	f(об/мин)	единицы измерения	к (меток/ оборот)	f0 (кГц)
0	100	об/мин	8000	8000
1	200	об/с	7810	7910
2	300	об/мин	7620	7820
3	400	1/с	7430	7730
4	500	об/мин	7240	7640
5	600	об/с	7050	7550
6	700	об/с	6860	7460
7	800	1/с	6670	7370
8	900	об/мин	6480	7280
9	1000	об/с	6290	7190
10	1100	1/с	6100	7100
11	1200	1/с	5910	7010
12	1300	об/мин	5720	6920
13	1400	об/мин	5530	6830
14	1500	об/с	5340	6740
15	1600	1/с	5150	6650
16	1700	об/с	4960	6560
17	1800	об/мин	4770	6470
18	1900	об/с	4580	6380
19	2000	1/с	4390	6290
20	2100	1/с	4200	6200
21	2200	об/мин	4010	6110
22	2300	об/мин	3820	6020
23	2400	об/с	3630	5930
24	2500	об/с	3440	5840
25	2600	1/с	3250	5750
26	2700	1/с	3060	5660
27	2800	об/мин	2870	5570
28	2900	об/мин	2680	5480
29	3000	об/с	2490	5390
30	3100	1/с	2300	5300
31	3200	об/мин	2110	5210
32	3300	об/с	1920	5120
33	3400	об/с	1730	5030
34	3500	1/с	1540	4940
35	3600	1/с	1350	4850
36	3700	об/мин	1160	4760

Для построения схемы рекомендуется использовать микроконтроллеры фирмы Atmel. Элементная база для построения генератора тактовых импульсов и индикатора выбирается студентом самостоятельно. Возможно использование любых интегральных схем малой, средней и большой степени интеграции, светодиодных индикаторов отечественного и зарубежного производства.

2. Содержание проекта

Работа над курсовым проектом предусматривает решение следующих взаимосвязанных задач:

- ° разработка структурной схемы микропроцессорной системы;
- ° разработка алгоритма работы микропроцессорной системы;
- ° разработка и отладка программного обеспечения на языке ассемблера;
- ° разработка принципиальной схемы микропроцессорной системы;
- ° анализ результатов проектирования.

2.1. Разработка структурной схемы

Структурная схема определяет основные функциональные части изделия, их назначение и связи. Разработки такой схемы подразумевает определение оптимального состава аппаратных и программных средств для решения поставленной задачи. На этом этапе необходимо выбрать микроконтроллер для решения задачи и принять решение, какие функции системы будут реализованы аппаратными средствами, а какие – программными.

Для разработки схемы необходимо прояснить все требования технического характера:

Во первых, надо проанализировать все устройства, подключаемые к микроконтроллеру, их электрические характеристики, схемы включения, временные диаграммы и способы обработки информации.

Во -вторых, необходимо рассмотреть набор внутренних функциональных блоков микроконтроллера: порты ввода-вывода, аналоговые и аналого-цифровые элементы, систему прерываний, интерфейсы, таймеры и др.

В результате сопоставления требований устройства и необходимого для решения задачи набора функциональных блоков принимается решение об использовании того или иного микроконтроллера и модулей в нем.

Одна и та же задача может решаться разными способами, все из них можно рассмотреть в первом разделе работы, сопоставить и объяснить выбор одного из них.

На структурной схеме устройства необходимо показать все внешние устройства и используемые в системе функциональные блоки микроконтроллера, их взаимосвязь и параметры, обеспечивающие решение поставленной задачи. Структурная схема оформляется как чертеж с соблюдением всех требований стандартов.

Пример

При известной скорости вращения n (об/мин) и разрешающей способности k (имп/об) частота сигнала на выходе датчика $f = n \cdot k / 60$ (Гц); при скорости $n = 800$ об/мин и $k = 1000$ имеем: $f = 800 \cdot 1000 / 60 = 13333$ Гц.

Оценка скорости может производиться путем подсчета импульсов с выхода датчика за фиксированный интервал времени T , как это показано на рис. 2.1.

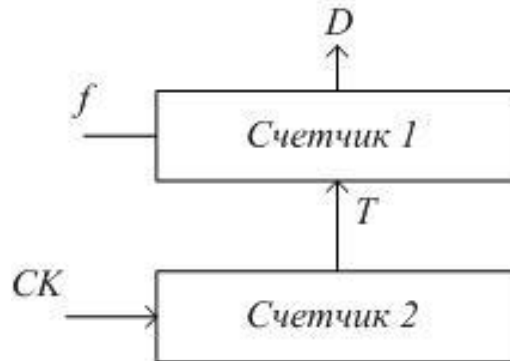


Рис. 2.1. Измеритель скорости с двумя счетчиками

Число, накопленное в счетчике 1 за время T , равно $D = f \cdot T$. Результат счета должен соответствовать измеряемой скорости и заданным требованиям точности. Если измеряемую скорость представить в радианах в секунду, то $D = 800 \cdot 6,28 / 60 = 83,7$ (1/с). Такой результат может быть записан в восьмиразрядном счетчике.

В результате интервал времени T можно рассчитать по формуле: $T = D / f$. При максимальной частоте сигнала датчика $f = 13333$ Гц получаем $T = 83,7 / 13333 = 0,006278$ сек.

Счетчик 2 на рис. 2.1 формирует временной интервал T , пересчитывая тактовую частоту $Ск$ процессорного ядра. При тактовой частоте 1 МГц для формирования интервала 0,006278 сек необходим счетчик с коэффициентом счета $Kс = 0,006278 \cdot 1000000 = 6278$, что можно получить в шестнадцатиразрядном счетчике без делителя..

На рис. 2.2 показан возможный вариант структурной схемы проектируемой системы.

В системе задействованы два таймера счетчика: восьмиразрядный таймер/счетчик 0 (Timer/counter 0) и шестнадцатиразрядный таймер/счетчик 1 (Timer/counter 1). Таймер/счетчик 0 подсчитывает импульсы с выхода датчика скорости, подключенного к контакту Т0. Таймер счетчик 1 осуществляет подсчет тактовых импульсов СК (Clock) частотой 1 МГц от генератора процессорного ядра. Используя внутренний регистр сравнения, в который загружено число 6278, он через интервалы времени 0,006278 сек формирует запросы на прерывания. Подпрограмма

обработки прерывания копирует содержимое счетчика 0 в регистр временного хранения *temp*, выводит данные из регистра *temp* в память, сбрасывает таймер/счетчик 0 и таймер/счетчик 1. В результате, в ячейке памяти сохраняется число, соответствующее скорости вращения датчика.

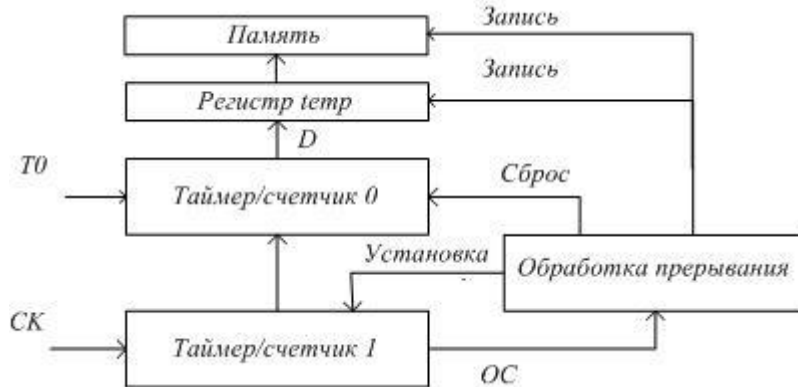


Рис. 2.2. Измеритель скорости. Схема структурная

2.2. Разработка схемы алгоритма

Схема алгоритма должна наглядно отображать все действия микроконтроллера по решению поставленной задачи.

Общий алгоритм может быть представлен отдельными, связанными между собой схемами, описывающими работу отдельных подпрограмм. При использовании прерываний каждая подпрограмма обработки прерывания должна иметь отдельную схему алгоритма. Все действия, изображенные на схеме алгоритма, должны быть понятны. В то же время, они не должны повторять текст программы. В пояснительной записке необходимо подробно обосновать разработанную схему.

Пример

В рассматриваемом случае, при структуре системы на рис. 2.2. алгоритм работы системы может включать в себя две программы:

- программа инициализации микроконтроллера (*main*),
- подпрограмма обработки прерывания (*int_os*).

При инициализации микроконтроллера в программе *main* (рис. 2.3) задаются режимы работы системы прерываний и таймеров/счетчиков, разрешаются прерывания, процессор переходит в режим ожидания прерываний.

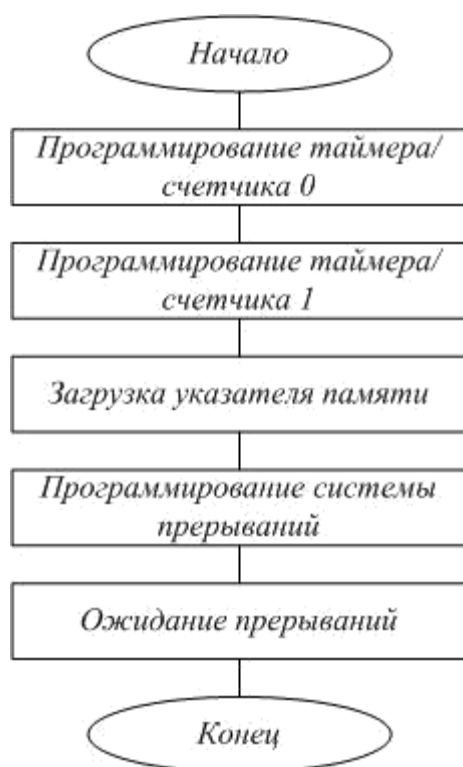


Рис. 2.3. Измеритель скорости. Пример схемы алгоритма основной программы

На приведенной схеме алгоритма:

- *программирование таймера счетчика 0 предполагает программное подключение его к контакту T0 и начальный сброс;*
- *программирование таймера/счетчика 1 означает загрузку в регистр сравнения числа 6278, подключение входа счетчика к тактовому генератору микроконтроллера без предделителя, установку бита маски прерываний;*
- *при загрузке указателя памяти необходимо выделить область памяти для размещения считываемых из счетчика 0 данных; указатель памяти должен указывать на первую ячейку этой области;*
- *программирование системы прерываний в данном случае сводится к глобальному разрешению прерываний;*
- *при ожидании прерываний процессор должен выполнять какую-либо работу; в простейшем случае это может быть бесконечный цикл, содержащий только пустые операции.*

На рис. 2.4 приведена подпрограмма обработки прерывания. В данной подпрограмме предусматривается сохранение содержимого таймера/счетчика 0 в регистре временного хранения temp, сброс таймера/счетчика 0, сброс таймера/счетчика 1 и запись новых данных в память.



Рис. 2.4. Пример алгоритма обработки прерывания *int_0c*

2.3. Разработка программы

Программа на языке ассемблера должна соответствовать разработанному алгоритму.

В пояснительной записке должно быть подробно обосновано использование тех или регистров ввода/вывода, выбор режимов работы функциональных блоков, программирование регистров. Используемые регистры ввода вывода необходимо изобразить в пояснительной записке, пояснить назначение отдельных разрядов, обосновать необходимость их установки или сброса в процессе выполнения программы.

Пример

Основная программа main Программирование таймера / счетчика 0

Для управления счетчиком необходимо использовать минимум два регистра (рис. 2.5):

- *регистр управления tccr0 (timer/counter0 control register),*
- *регистр данных tcnt0 (timer/counter0).*

	7	6	5	4	3	2	1	0
tccr0	-	-	-	-	-	cs02	cs01	cs00
tcnt0	-	-	-	-	-	-	-	-

Рис. 2.5. Регистры, участвующие в управлении таймером/счетчиком 0

Регистр *tccr0* управляет входом таймера/счетчика 0, в нем биты 2..0 - *cs02*, *cs01*, *cs00* – задают коэффициент деления предделителя таймера/счетчика 0 и тип внешнего сигнала (табл. 2.1).

Таблица 2.1

Коэффициенты деления предделителя таймера/счетчика 0

<i>cs02</i>	<i>cs01</i>	<i>cs00</i>	описание
0	0	0	остановка таймера
0	0	1	<i>ck</i>
0	1	0	<i>ck</i> / 8
0	1	1	<i>ck</i> / 64
1	0	0	<i>ck</i> / 256
1	0	1	<i>ck</i> / 1024
1	1	0	внешний контакт <i>t0</i> , задний фронт
1	1	1	внешний контакт <i>t0</i> , передний фронт

Для подключения внешнего контакта *T0* в регистр *tccr0* необходимо загрузить двоичное число *0b111* :

```
ldi var, 0b111; здесь var – временная переменная
out tccr0, var
```

Для начального сброса счетчика необходимо занести 0 в регистр *tcnt0*:

```
ldi var, 0
out tcnt0, var
```

Программирование таймера / счетчика 1

Работу таймера/счетчика 1 обеспечивается регистрами (рис. 2.6):

- управления таймера счетчика *tccr1a* (*timer/counter1 control register a*) и *tccr1b* (*timer/counter1 control register b*);
- данных *tcnt1h* (*timer/counter1 high*) и *tcnt1l* (*timer/counter1 low*);
- сравнения *ocr1ah* (*timer/counter1 output compare register a high*) и *ocr1al* (*timer/counter1 output compare register a low*);
- маски прерывания *timsk* (*timer/counter interrupt mask register*).

	7	6	5	4	3	2	1	0
<i>tccr1a</i>	<i>com1a1</i>	<i>com1a0</i>	<i>com1b1</i>	<i>com1b0</i>	<i>foc1a</i>	<i>foc1b</i>	<i>pwm11</i>	<i>pwm10</i>
<i>tccr1b</i>	<i>icnc1</i>	<i>ices1</i>	-	-	<i>ctc1</i>	<i>cs12</i>	<i>cs11</i>	<i>cs10</i>
<i>tcnt1h</i>								
<i>tcnt1l</i>								
<i>ocr1ah</i>								
<i>ocr1al</i>								
<i>tmsk</i>			<i>ticie1</i>	<i>ocie1a</i>	<i>ocie1b</i>	<i>toie1</i>		

Рис.2.6. Регистры таймера/счетчика 1

В регистре управления *tccr1b* находятся биты для подключения входа счетчика/таймера 1: биты 2..0 - *cs12*, *cs11*, *cs10* (*clock select1*, bit 2,1, and 0) (табл. 2.2)

Таблица 2.2

Управление входом таймера/счетчика 1

<i>cs12</i>	<i>cs11</i>	<i>cs10</i>	Описание
0	0	0	Останов таймера/счетчика 1.
0	0	1	СК
0	1	0	СК / 8
0	1	1	СК / 64
1	0	0	СК / 256
1	0	1	СК / 1024
1	1	0	Внешний контакт <i>t1</i> , задний фронт импульса
1	1	1	Внешний контакт <i>t1</i> , передний фронт импульса

Для подключения входа таймера/счетчика 1 к тактовому генератору процессорного ядра необходимо установить биты 2..0 в состояние 0b001:

```
ldi var,0b1
out tccr1b, var
```

Регистры данных *tcnt1h* и *tcnt1l* образуют 16-разрядный регистр, содержащий текущее значение 16- разрядного таймера/счетчика 1. Для начального сброса счетчика запишем в старший и младший байты счетчика число 0:

```
ldi var,0
out tcnt1h, var
out tcnt1l, var
```

Таймер/счетчик 1 поддерживает два выхода сравнения, используя регистры сравнения *ocr1a* и *ocr1b* в качестве источников данных, сравниваемых с содержимым таймера/счетчика 1. Для решения задачи достаточно одного 16-битного регистра сравнения *ocr1a*, состоящего из двух 8-битных регистров *ocr1ah* и *ocr1al*.

Загрузим в регистр сравнения *ocr1a* число $6278 = \$1886$:

```
ldi var,$18
out ocr1ah, var
ldi var,$86
out ocr1al, var
```

Загрузка указателя памяти

Для хранения результатом выделим 256 ячеек памяти данных с адресами $\$100..\$1FF$. Для адресации ячеек используем регистр косвенной адресации *X*, состоящий из двух 8-разрядных регистров *xl* и *xh*. Загрузим в указатель памяти число $\$100$:

```
ldi xl,0
ldi xh,1
```

Программирование системы прерываний

В системе предполагается использовать прерывания только от регистра сравнения таймера/счетчика 1. Для разрешения этого прерывания в регистре масок прерываний *timsk* необходимо установить бит 4 (*ocie1a*) – бит разрешения прерывания при сравнении в регистре *ocr1a*:

```
ldi var,0b10000
out timsk, var
```

Ожидание прерываний

Ожидание прерываний можно выполнить в замкнутом цикле с пустыми операциями, например:

```
тт: пор; где тт – метка начала
цикла jmp тт
```

Подпрограмма обработки прерывания

int_ос Чтение таймера / счетчика 0

Содержимое таймера/счетчика 0 копируется в регистр временного хранения temp и таймер счетчик 0 обнуляется:

```
in temp,tcnt0  
ldi var,0  
out tcnt0,var
```

Подпрограмма записи данных в память

При каждом чтении данные заносятся в новую ячейку памяти в выделенной области. Номер ячейки задается указателем памяти.

```
st x+,temp  
cpi xh,2  
brne pp  
ldi xh,1  
pp:
```

Сброс таймера / счетчика 1

В таймер/счетчик 1 заносится нулевое значение:

```
ldi var,0  
out tcnt1h,var  
out tcnt1l
```

2.3. Листинг программы

Листинг программы формируется в программе AVR Studio автоматически. При компиляции выявляются все синтаксические ошибки. В конце листинга следует сообщение об отсутствии ошибок.

Пример

AVRASM ver. 1.30 C:\Proba\encoder.asm Sun Feb 26 15:32:36 2006

```
; курсовой проект по дисциплине МП в ЭП  
; Иванов А.А., гр. ЗЭПЧ-31
```

```

; 1 марта 2006 года
.nolist
.def var=r20; определение регистров
.def temp=r21
.org 0 ; вектора прерываний
000000 940c 0030 jmp main
.org $12
000012 940c 0045 jmp int_oc
.org $30
main: ; основная программа
000030 e047 ldi var, 0b111; программирование таймера/счетчика 0
000031 bf43 out tccr0,var
000032 e040 ldi var,0
000033 bf42 out tcnt0, var
000034 e041 ldi var,0b1
000035 bd4e out tccr1b, var
000036 e040 ldi var,0 ; программирование таймера/счетчика 0
000037 bd4d out tcnt1h, var
000038 bd4c out tcnt1l, var
000039 e148 ldi var,$18; загрузка регистра сравнения
00003a bd4b out ocr1ah, var
00003b e846 ldi var,$86
00003c bd4a out ocr1al, var
00003d e0a0 ldi xl,0; загрузка указателя памяти
00003e e0b1 ldi xh,1
00003f e140 ldi var,0b10000; разрешение прерываний
000040 bf49 out tmsk, var
000041 9478 sei; глобальное разрешение прерываний
000042 0000 tm: por; ожидание прерываний
000043 940c 0042 jmp tm
int_oc: ; подпрограмма обработки прерываний
000045 b752 in temp,tcnt0; чтение таймера/счетчика 0
000046 e040 ldi var,0; обнуление таймера/счетчика 0
000047 bf42 out tcnt0,var
000048 935d st x+,temp; запись данных в память
000049 30b2 cpi xh,2
00004a f409 brne pp
00004b e0b1 ldi xh,1
pp:

```

```

00004c e040    ldi var,0; сброс таймера/счетчика 1
00004d bd4d    out tcnt1h,var
00004e bd4c    out tcnt1l,var
00004f 9518    reti ; выход из прерывания
        .exit

```

Assembly complete with no errors.

2.4. Отладка программы

Отладку программы следует производить на персональном компьютере с использованием интегрированной отладочной среды, соответствующей выбранному микроконтроллеру. В результате отладки должны быть созданы работоспособные файлы проекта. Листинг отлаженной программы необходимо распечатать и поместить в приложении.

Используя интегрированную отладочную среду необходимо смоделировать работу спроектированной системы и доказать её работоспособность. Отдельные окна программы, иллюстрирующие состояние микропроцессорной системы в различные моменты времени (состояние входов/выходов, регистров) должны быть приведены в тексте пояснительной записке.

Пример

Состояние регистров, памяти данных, регистров общего назначения и ввода/вывода после выполнения трех циклов прерывания показано на рис. 2.6-2.8.

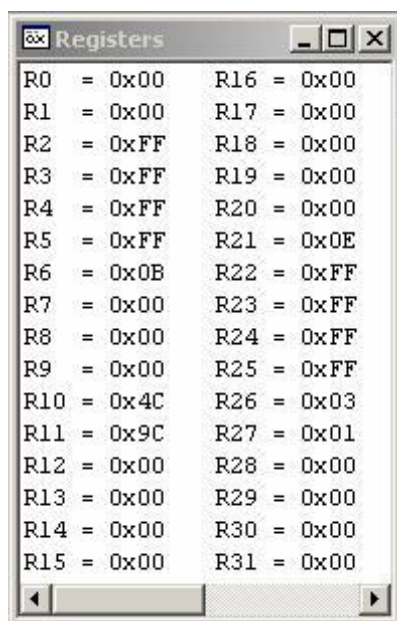


Рис. 2.6. Регистры общего назначения

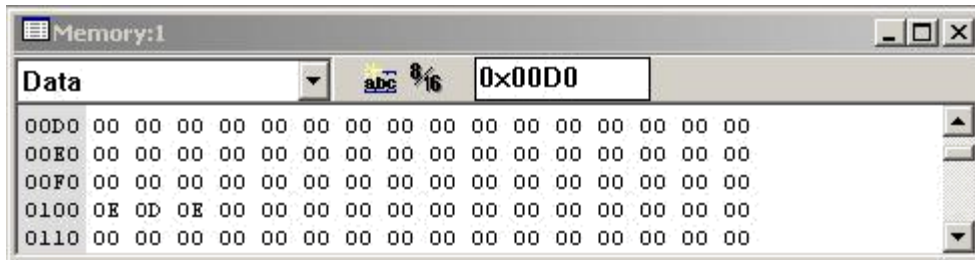


Рис. 2.7. Память данных

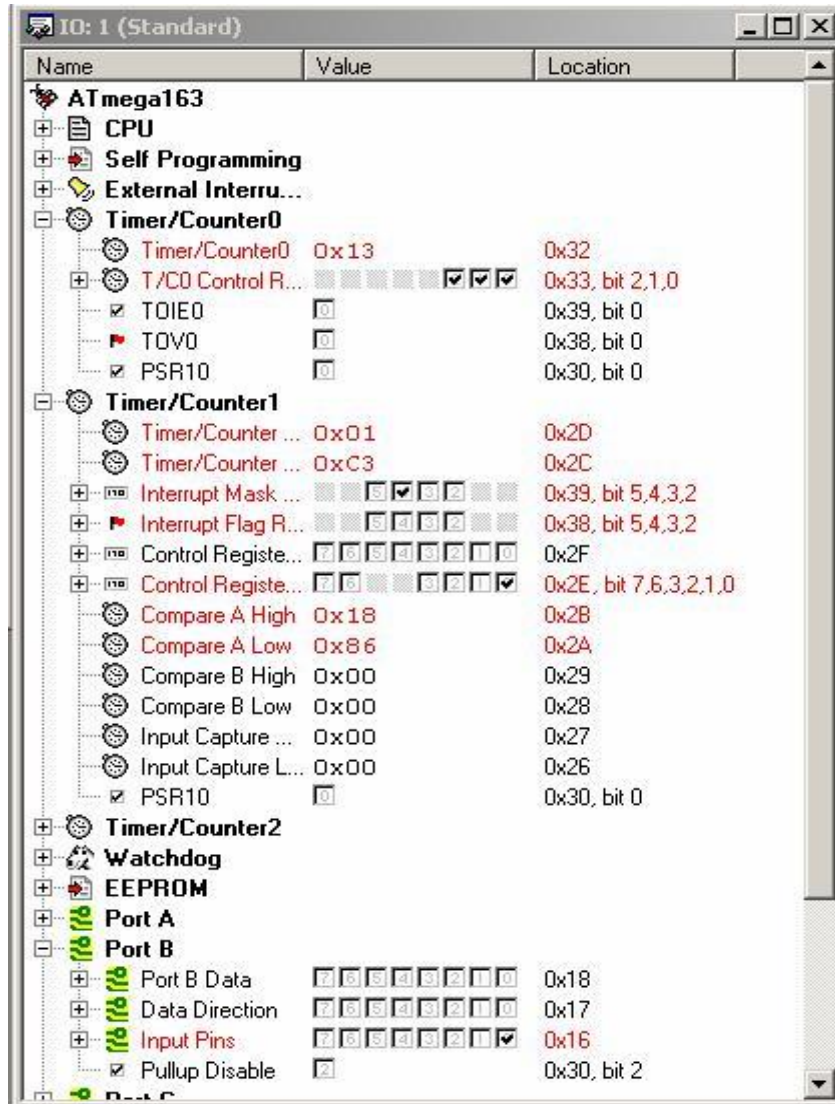


Рис. 2.8. Регистры ввода/вывода

2.5. Разработка принципиальной схемы

Принципиальная схема устройства содержит все элементы системы и связи между ними. Основной задачей при разработке принципиальной схемы является выбор компонентов и их электрическое согласование.

Основным элементом системы является микроконтроллер, электрические характеристики которого определяют требования к другим элементам. При

разработке принципиальной схемы необходимо учитывать тип его корпуса, однозначно определяющий нумерацию выводов.

Необходимо разобраться с назначением выводов, подключить питание (+5В) к микроконтроллеру, согласовать его выводы с другими элементами схемы. Уровни входных сигналов должны соответствовать уровням микросхем ТТЛ (логическая единица = 2,4...2,5В). Нагрузочную способность выходов можно принять равной 10 входам элементов ТТЛ, а выходы портов выдерживают ток до 20 мА, допуская прямое подключение светодиодов.

При включении микроконтроллера необходимо выбрать один из возможных генераторов синхронизации: внешний или внутренний. Внешний кварцевый резонатор необходимо подключить к контактам микроконтроллера.

Ко входу сброса микроконтроллера (#RESET) следует подсоединить кнопку или источник внешнего сигнала сброса.

Принципиальная схема должна быть оформлена в соответствии с ГОСТ «Правила выполнения электрических схем».

Пример

Принципиальная схема разрабатываемой системы приведена на рис. 2.9. В схеме импульсный датчик подключен к входу T 0 таймера/счетчика 0, кварцевый резонатор с резонансной частотой 1 МГц – к контактам XTAL1, XTAL2, кнопка сброса – к контакту Reset.

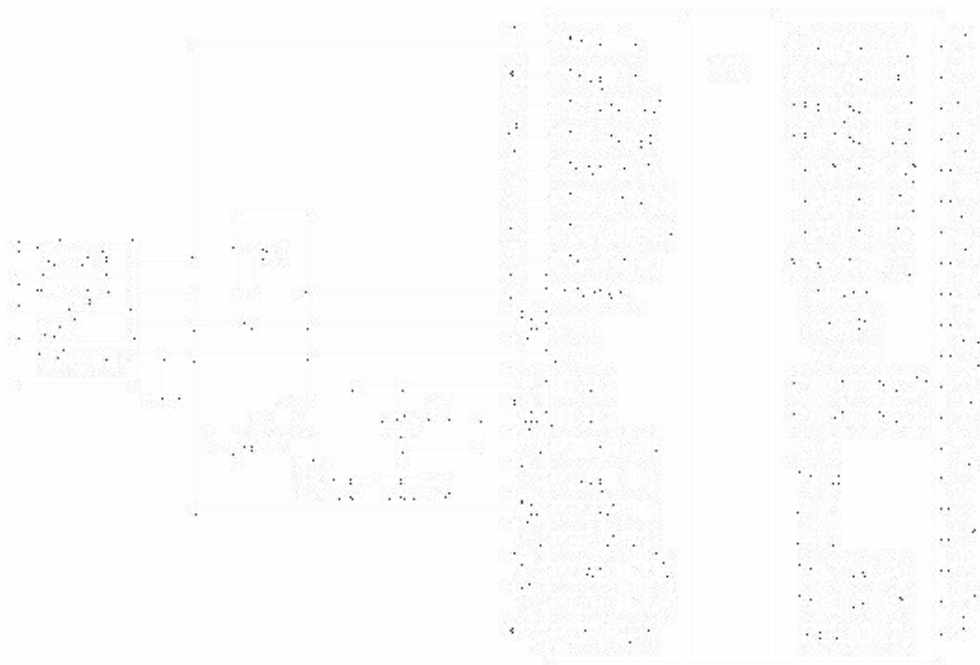


Рис. 2.9. Схема измерителя скорости

4. ОФОРМЛЕНИЕ ПРОЕКТА

4.1. Пояснительная записка

Пояснительная записка является документом, содержащим обоснование принятых при разработке решений, описание разработанного устройства.

Пояснительная записка курсового проекта должна содержать следующие разделы:

- ° титульный лист,
- ° оглавление,
- ° задание на проектирование,
- ° введение,
- ° разработка структурной схемы,
- ° разработка алгоритма,
- ° разработка программы,
- ° разработка принципиальной схемы,
- ° заключение
- ° список использованных источников информации.

Пояснительная записка должна содержать 15-20 страниц текста с иллюстрациями, выполненными на одной стороне листа белой бумаги формата А4. При оформлении с использованием оргтехники текст печатается через 1,5 межстрочных интервала, высота шрифта не менее 2,5 мм.

4.2. Оформление чертежей

Графический материал проекта должен содержать схему алгоритма, структурную схему и принципиальную схему системы, оформленные на листах стандартного формата.

ЛИТЕРАТУРА

1. Водовозов, А.М. Микроконтроллеры для систем автоматики: Учебное пособие/ А.М.Водовозов.- Вологда: ВоГТУ, 2002.- 128 с.
2. Жданкин, В. Поворотные шифраторы: основные типы и некоторые особенности применения/ В.Жданкин// Современные технологии автоматизации.- 2001.-№ 2.- С. 68 – 79 .
3. Жданкин, В. Поворотные шифраторы фирмы Pepperl+Fuchs/ В.Жданкин // Современные технологии автоматизации.- 2001.- № 3.- С. 6-24.
4. Водовозов, А.М. Цифровые элементы систем автоматики. Учебное пособие/ А.М.Водовозов. – Вологда: ВоГТУ, 2002.- 110 с.