

Яндекс.Директ



### Грузовой лифт в Краснодаре

Цена от 60 000руб производство под ваш размер + монтаж! Замер бесплатно

Подъемник в шахту

Цепной подъемник

Мини подъемник Замер

[zavod-ptm.ru](http://zavod-ptm.ru) Адрес и телефон

×



### Грузовые лифты - подъемники!

Мы завод! Работаем без посредников! Подъемники от 59 000р! Узнать подробнее

Наша продукция О нас

Наши контакты

[грузовой-лифт.рф](http://грузовой-лифт.рф)

Адрес и телефон

×



### Светодиодная подсветка LED!

Низкие цены. Гарантия 3 года. Доставка 0 руб. Настройка. Установка.Звоните!

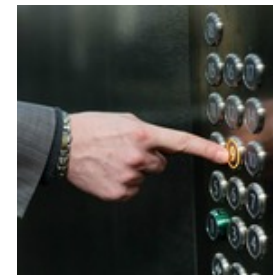
Светодиодный экран

Бегущая строка

Медиафасад LED табло

[ledimperial.ru](http://ledimperial.ru) Адрес и телефон

×



### Лифты под ключ в Краснодаре

Поставка, монтаж, сервис **лифтов** от производителя. Гарантия 24 мес! Звоните!

Модели Сертификаты

Оставить заявку Контакты

[esd-lift.ru](http://esd-lift.ru) Адрес и телефон

×

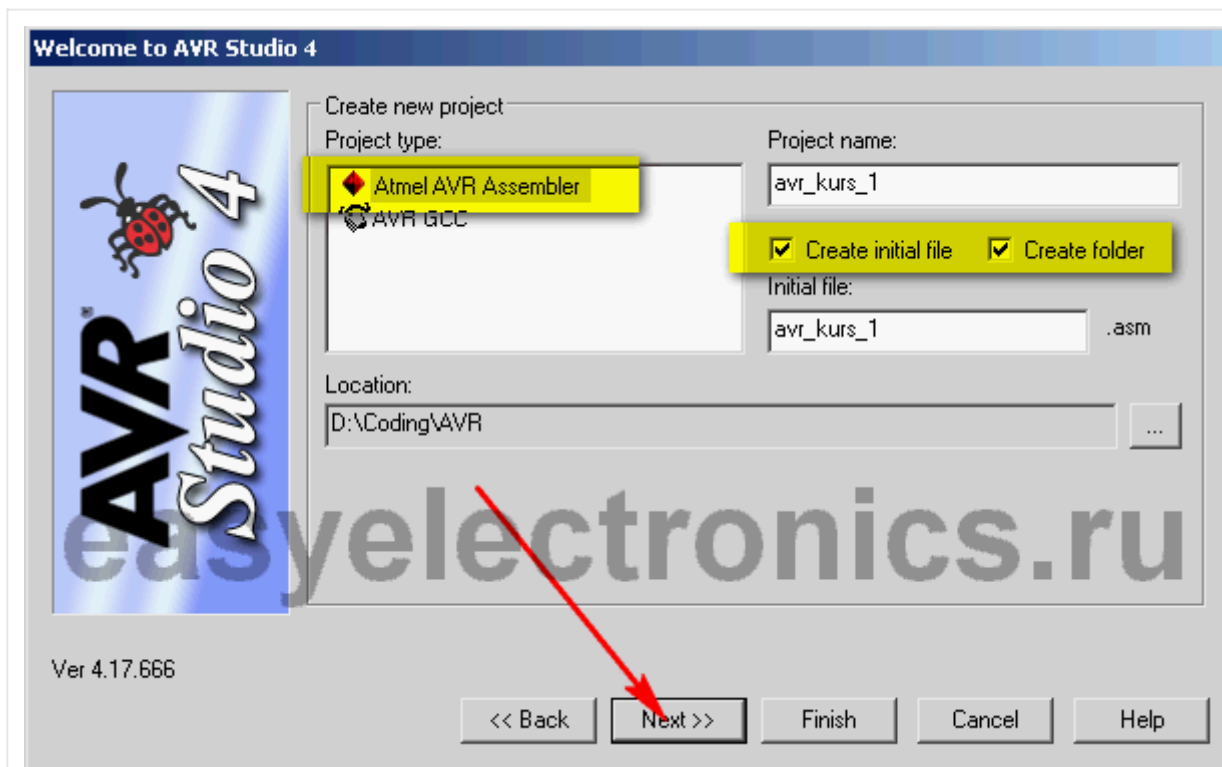
## AVR. Учебный курс. Простейшая программа.

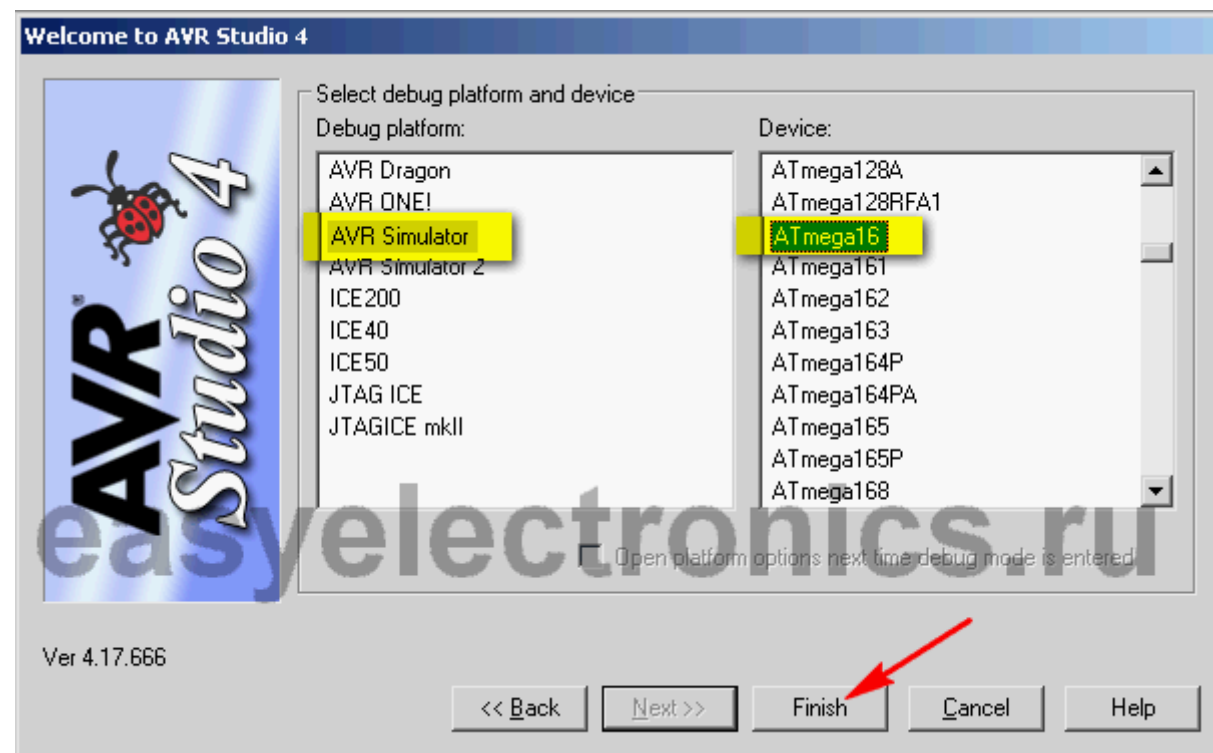
AVR. Учебный курс | 6 Июль 2008 | DI HALT | 233 Comments

Итак, прежде чем что то делать надо понять как вообще выполняется программа в контроллере, как работает ядро процессора. Для этого нам хватит AVR Studio и его эмулятора. Не очень интересно, может даже занудно, но если этот шаг пропустить, то дальнейшие действия будут как бег в темноте.

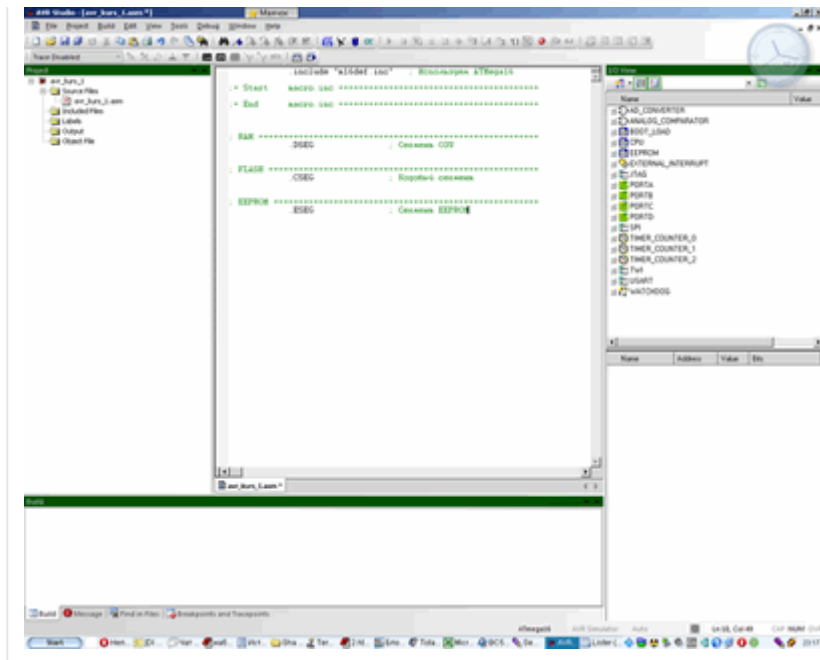
Поскольку в демоплате Pinboard используется процессор ATmega16, то рассматривать мы будем именно его. Впрочем, как я уже говорил, для других контроллеров AVR это также будет справедливо. Отличия, конечно, есть, но они не существенные.

Запускаем AVR Studio (далее просто студия) и в выскочившем мастере сразу же создаем проект:





Откроется окно:



## Увеличить

Оно может выглядеть чуток не так. Дело в том, что интерфейс студии очень легко конфигурируется и перекраивается под себя. Так что можно перетаскать панели как нам удобно. Что я и сделал.

Систему команд хорошо бы распечатать себе на листочке. Их там всего около 130, кратким списком (тип команды, что делает и какие операнды) занимает пару листов формата А4. Учить не надо, прочитать раз на десять, чтобы помнить знать что у нас есть. Даже я периодически подглядываю в систему команд, хотя пишу на ассемблере уже много лет. Я же команды которые буду использовать буду описывать по мере появления.

В центральном окне пишем код:

```
1          .include "m16def.inc"    ; Используем ATmega16
2      ;= Start macro.inc =====
```

```

3
4     ; Тут будут наши макросы, потом.
5
6     ;= End macro.inc =====
7
8
9     ; RAM =====
10         .DSEG                ; Сегмент ОЗУ
11
12
13     ; FLASH =====
14         .CSEG                ; Кодовый сегмент
15
16
17     ; EEPROM =====
18         .ESEG                ; Сегмент EEPROM

```

Это вроде шаблона для начала любого проекта. Правда кода тут 0 байт :) Только директивы.

Обрати также внимание на оформление кода. Дело в том, что в ассемблере нет никаких структурных элементов — весь код плоский, тупо в столбик. Так что поэтому чтобы он был читабелен его надо в обязательно порядке разбивать на блоки, активно пользоваться табуляцией и расставлять комментарии. Иначе уже через пару дней ты не сможешь понять, что вообще написал.

В коде будут активно использоваться макросы. Поначалу они будут писаться в основном файле, в секции macro.inc, но потом я их вынесу в отдельный файл, чтобы не мешались. Так удобней.

Как я уже говорил, в микроконтроллере куча отдельных видов памяти. И тут используется гарвардская архитектура. Т.е. исполняемый код в одном месте, а переменные в другом. Причем это разные адресные пространства. Т.е. адрес 0000 в сегменте данных это не то же что и адрес 0000 в сегменте кода. Это дает большое преимущество в том, что данные не могут испортить код, но не дает писать полиморфные программы. Такая архитектура является традиционной для микроконтроллеров.

У обычного PC компа используется другая архитектура — Фон Неймановская. Там данные и код находятся в одном адресном пространстве. Т.е., скажем, с адреса 0000 по 0100 идет код, а с 100 до FFFF данные.

В нашей же программе и код и данные располагаются на одном листе, но чтобы компилятор понял где у нас что они выделяются директивами сегментации. Пока нас интересуют директива .CSEG после нее начинается исполняемый код. Там и будем писать нашу программу.

Возьмем и напишем:

1	; FLASH =====	
2	.CSEG	; Кодовый сегмент
3	NOP	
4	NOP	
5	NOP	

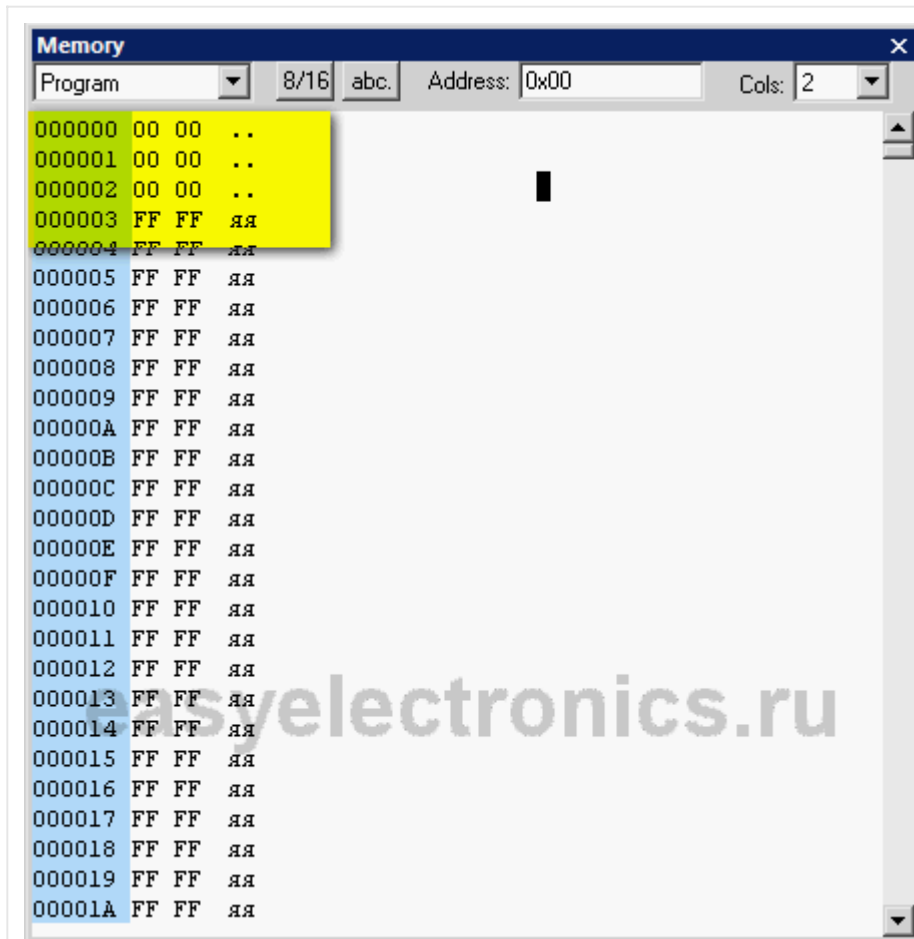
Что мы сделали? А ничего! Команда NOP это команда затычка. Она не делает ничего, просто занимает 2 байта и 1 такт.

Речь не о команде NOP (там и обсуждать то нечего), а о том как оно все будет выполняться.

Запускай симуляцию (Ctrl+F7) когда пробежит прогресс бар компиляции/симуляции и возле первого NOP возникнет стрелочка нажми ALT+O — выскочит диалог настройки симуляции. Там тебе надо там только выставить частоту 8МГц. Почему 8МГц? Просто на [Pinboard](#) частота главного проца по дефолту 8МГц. Если у тебя свои идеи на этот счет — можешь поправить как угодно. На симуляцию это не влияет.

Вернемся к нашей симуляции. Давай посмотрим как выглядит наша программа с точки зрения машинных кодов, как размещается в памяти. Интерес, по большей части, чисто теоретический, редко когда пригождается и по этому во многих учебных курсах по AVR на это даже внимание не заостряют. А зря! Т.к. упускается глубинное ощущение кода. Открой окно просмотра программ. View — Memory или Alt+4.

Там выбери тип памяти Programm. Это и есть наш Flash сегмент. Выставим в списке cols две колонки, чтобы было наглядней.

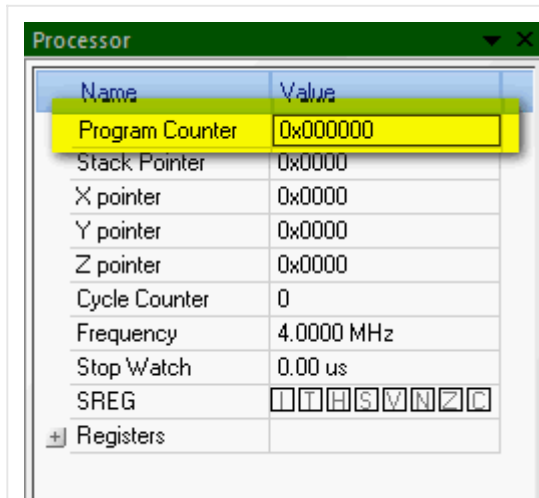


Это наша память программ. На голубом фоне идут адреса, они, как видишь, в словах. Т.е. двум байтам соответствует один адрес. Но, на самом деле это интерпретация адресации компилятором, микроконтроллер же может оперировать в памяти программ с точностью до байта.

0000 это код команды NOP. У нас три команды, поэтому шесть нулей.

Обрати внимание на то, что команды идут с адреса 0000. Это потому, что мы не указали ничего иного. А кодовый сегмент начинается с адреса 0000.

Теперь смотрим на программный счетчик **Program Counter**, он показывает адрес инструкции которая будет выполнена.



Поскольку мы только стартовали, то он равен нулю — нулевая инструкция. Нажми F11, процессор сделает шаг, программный счетчик изменится на 1 и покажет следующую инструкцию. И так до тех пор, пока не дойдет до третьего NOP, а что у нас после него? А после него у нас FF до конца памяти. FF это несуществующая инструкция, на ней виртуальный контроллер перезагрузится с ошибкой invalid opcode, а реальный контроллер ее проигнорирует, пробежав по ним, как по NOP, до конца памяти.

Сбрось симуляцию микроконтроллера (Shift+F5) и вручную выстави в Program Counter адрес 0x000002 — проц сам перепрыгнет на последнюю команду NOP. Если менять Program Counter то проц будет выполнять те команды, которые мы ему укажем. Но как это сделать в реальном контроллере? В него то мышкой не залезешь!



Программный счетчик меняется командами переходов. Их много (условные, безусловные, относительные), о них я расскажу подробнее чуть позже, пока же приведу один пример.

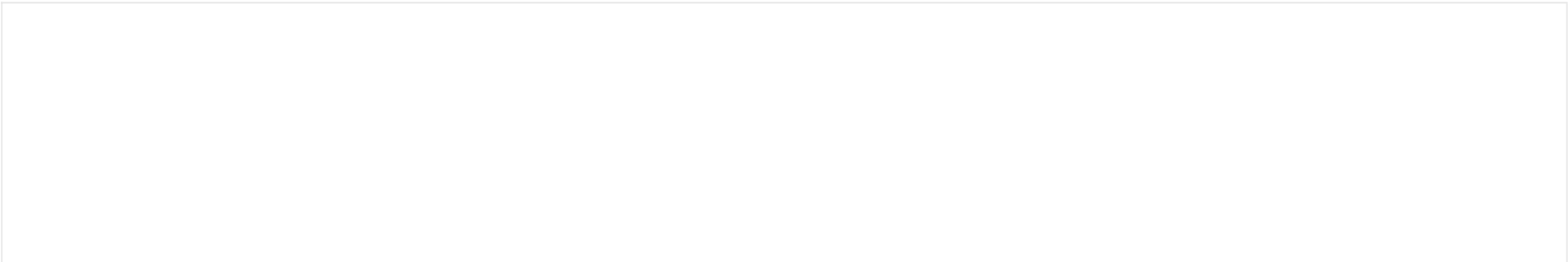
Добавим в наш код команду JMP и ее аргумент — адрес 0x000001.

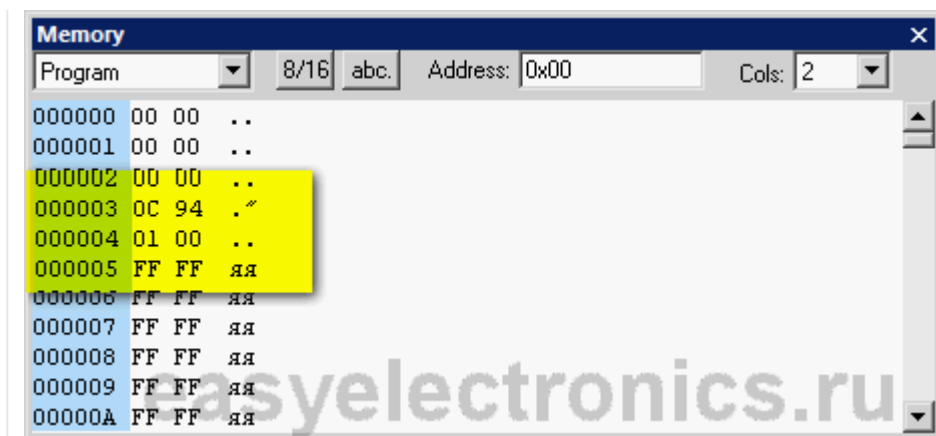
1	.CSEG	; Кодовый сегмент
2	NOP	
3	NOP	
4	NOP	
5	JMP 0x000001	

Команда JMP, как и все команды перехода, работает просто — записывает в Program Counter свой аргумент. В нашем случае — 0x000001.

Перекомпиль проект и посмотри на то, как меняется Program Counter (далее буду звать его PC) и вообще как теперь идет процесс выполнения программы. Видишь, после JMP в программный счетчик заносится новый адрес и процессор сразу же перепрыгивает в начало кода, но не на первую, а на вторую инструкцию. Программа зациклилась.

Это называется абсолютный переход. Он может сигануть куда угодно, в любую область памяти программ. Но за такую дальнобойность приходится платить. Если заглянешь в память, то увидишь там такую картину:





0C 94 — это код нашей команды, а 01 00 адрес перехода. Длина команды стала четыре байта. Два байта ушло на адрес. Вот, кстати, особенность памяти в том, что там данные записываются зеркально, т.е. младший байт числа по младшему адресу (порядок little-endian).

Но поскольку дальнобойные команды применяются редко, основной командой перехода в AVR является относительный переход RJMP. Основное отличие тут в том, что в PC не записывается не точный адрес куда надо перейти, а просто к PC прибавляется смещение. Вот так:

1	.CSEG	; Кодовый сегмент
2	NOP	
3	NOP	
4	NOP	
5	RJMP PC+2	
6	NOP	
7	NOP	
8	RJMP PC-6	
9	NOP	

По идее, должно работать напрямую, т.е. RJMP +2. Но компилятор такую запись не понимает, он оперирует абсолютными адресами, приводя их потом в нужные смещения машинного кода. Поэтому применим макроопределение PC — это текущее значение счетчика в данном месте программы. Т.е.

1	JMP PC
---	--------

Наглухо зациклит программу в этой строчке.

Благодаря относительному переходу, смещение можно записать в те же два байта, что занимает команда. При этом код относительного перехода выглядит так Sx xx, где xxx это смещение от +/-2047 команд. Что обычно хватает с лихвой. В памяти же команда перехода выглядит как xx Sx, то есть байты переставлены.

Таким образом, длина команды у относительного перехода составляет 2 байта, против четырех у абсолютного. Экономия!!! Учитывая что в обычной программе подобные переходы на каждом шагу.

Вот только что же, каждый раз самому вручную высчитывать длину перехода? Ладно тут программка в три команды, а если их сотни? Да если дописал чуток то все переходы заново высчитывать?

Угу. Когда то давно, когда компиляторов еще не было так и делали. Сейчас же все куда проще — компилятор все сделает сам. Ему только надо указать откуда и куда. Делается это с помощью меток.

1	.CSEG ; Кодовый сегмент
2	M1: NOP
3	NOP
4	NOP
5	RJMP M2

6		NOP
7	M2:	NOP
8		RJMP M1
9		NOP

Метки могут быть из букв или цифр, без пробелов и не начинаться с цифр. Заканчиваются двоеточием. По факту, метка означает текущий адрес в словах. Так что ее можно использовать и в операциях. Надо только помнить, что она двубайтная, а наш контроллер однобайтный. Разобрать двубайтное значение по байтам можно с помощью директивы компилятора Low и High

Например,

1		.CSEG	; Кодовый сегмент
2	M1:	NOP	
3		NOP	
4		LDI ZL,low(M2)	; Загрузили в индекс
5		LDI ZH,High(M2)	
6			
7		IJMP	
8			
9		NOP	
10		NOP	
11		NOP	
12	M2:	NOP	
13		RJMP M1	
14		NOP	

Команда LDI загружает непосредственное значение в регистр старшей (от R16 до R31) группы. Например, LDI R17,3 и в R17 будет число 3. А тут мы в регистры R30 (ZL) загрузили младший байт адреса на который указывает метка M2, а в R31 (ZH) старший байт адреса. Если протрассируешь выполнение (F11) этого кода, то увидишь как меняются значения регистров R30 и R31 (впрочем 31 может и не поменяться, т.к. там был ноль, а мы туда ноль и запишем — адрес то мал). Смену значений регистров можно поглядеть в том же окне где и Program Counter в разделе Registers.

Команда JMP это косвенный переход. Т.е. он переходит не по адресу который заложен в коде операции или идет после него, а по адресу который лежит в индексной регистровой паре Z. Помните я говорил, что шесть последних регистров старшей регистровой группы образуют три регистровые пары X,Y,Z используются для адресации? Вот это я и имел ввиду.

После JMP мы переходим на нашу же M2, но хитровывернутым способом. Зачем вообще так? Не проще ли применить RJMP и JMP. В этом случае да, проще.

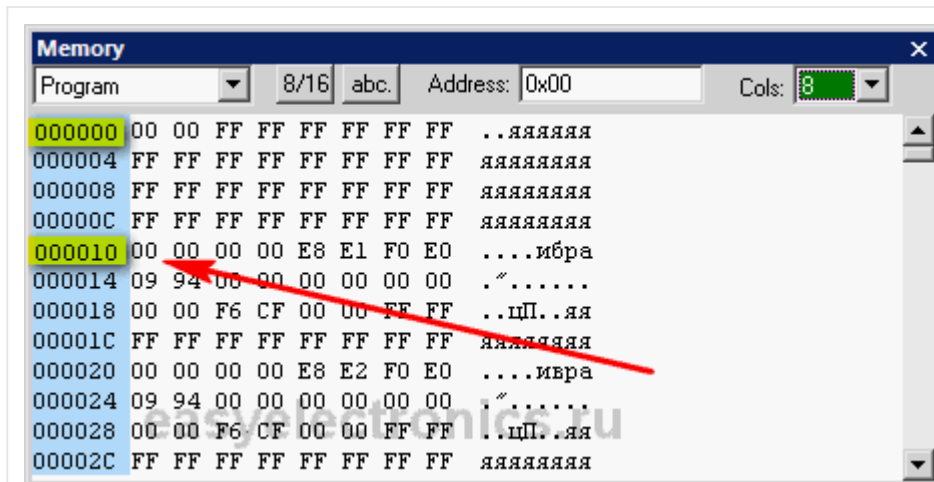
Но вот благодаря косвенному переходу мы можем **программно** менять точку перехода. И это просто зверский метод при обработке всяких таблиц и создании switch-case структур или конечных автоматов. Примеры будут позже, пока попробуйте придумать сами.

Давай подвинем нашу кодовую конструкцию в памяти на десяток байт. Добавь в код директиву ORG

1	; FLASH =====		
2		.CSEG	; Кодовый сегмент
3		NOP	
4			
5		.ORG 0x0010	
6	M1:	NOP	
7		NOP	
8		LDI ZL,low(M2)	; Загрузили в индекс
9		LDI ZH,high(M2)	
10			
11		IJMP	

12		
13		NOP
14		NOP
15		NOP
16	M2:	NOP
17		RJMP M1
18		NOP

Скомпиль и запусти. Открой память и увидишь что в точке с адресом 0000 у нас стоит наш NOP (Точка входа должна быть, иначе симулятор скукожит от такого когнитивного диссонанса. А вот дальше сплошные FF FF и лишь начиная с нашего адреса 0x0010 пошли коды остальных команд.



То есть мы разбили код, заставив директивой ORG компилятор перетаскать его дальше, начиная с 0x0010 адреса.

Зачем это нам потребовалось? Ну, в первую очередь, это требуется чтобы перешагнуть таблицу векторов прерываний. Она находится в первых адресах.

Во-вторых, такая мера нужна чтобы записать код в конец флеша, при написании бутлоадеров (о них тоже дальше в курсе будет).

А еще таким образом можно прятать в коде данные, перемешивая данные с кодом. Тогда дизассемблирование такой программы превратится в ад =) своего рода, обфускация публичной прошивки, сильно затрудняющая реверс инженеринг.

Да, у кристалла есть биты защиты. Выставил которые и все, программу можно считать только спилив крышку кристалла с помощью электронных микроскопов, что очень и очень недешево. Но иногда надо оставить прошивку на виду, но не дать сорцов. И вот тогда превращение прошивки в кашу очень даже поможет =)

Ну или написать ее на Си ;)))))) (Есть вариант шифровки прошивки и дешифровки ее на лету бутлоадером, но это отдельная тема. О ней может быть расскажу)

Еще, там, в студии, же где и программный счетчик, есть ряд интересных параметров.

Во-первых, там отдельно вынесены наши индексные пары X,Y,Z и можно не лазать на дно раздела Registers.

Во-вторых, там есть Cycle counter — он показывает сколько машинных циклов протикал наш контроллер. Один машинный цикл в AVR равен одному такту. Frequency содержит частоту контроллера в данный момент, но мы задаем ее вручную.

А в-третьих, есть Stop Watch который показывает время выполнения. В микросекундах. Но можно переключить и в миллисекунды (пошарь в контекстном меню).

Если тебе кажется, что все слишком просто и я чрезмерно все разжевываю. Хы, не расслабляйся, у меня сложность растет по экспоненте. Скоро пойдет работа на прерываниях, а потом будем писать свою операционную систему :))))))

[◀ Assembler](#) [◀ AVR](#) [◀ Программирование](#)

**233 thoughts on “AVR. Учебный курс. Простейшая программа.”**

## Beams

3 Август 2008 в 20:19

«Если вместо стройной иерархии структур в нашей голове образовалась каша, не стоит волноваться — это нормально. Постепенно она утрясётся и все структуры встанут на свои места, так что оставим их созревать, а сами сосредоточимся на текущих проблемах.» Крис Касперски.

Думается мне само то вставить в эпиграф ;)

---

## ★ DI HALT

3 Август 2008 в 22:28

Примерно так. Я действую тем же способом, что и учился сам. Взял и поставил задачу, а уж как она решается, из каких кусков делается... Разберемся по ходу дела.

---

## Arhangelsk

5 Август 2008 в 0:26

«Поехали!»! Все. Начали. Можно пробовать эмулировать схему в протеусе.

PS: А когда планируется первая «протеусная» статья?

---

## ★ DI HALT

5 Август 2008 в 0:51

Могу после 6го выложить чтонибудь. ПОка запара — статья в «Хакер» висит, а дедлайн ооочень скоро :)

---

## Softy

17 Август 2008 в 15:03



*Что такое стек? Это область памяти, работающая по принципу первый вошел-последний вышел.А не наоборот, последний вошёл, первый вышел (LIFO)? Хотя по логике вроде тоже самое.*

---

★ **DI HALT**

17 Август 2008 в 16:44

Ээээ а чем отличается? Вопрос лишь в формулировке. Да это оно.

---

**sinobi**

24 Август 2008 в 19:20

Интересно о ассемблере пишут,а о компиляторе нет ни слова,непонятно.Или я где-то просмотрел?Люди которые не знают как компильнуть как это будут делать это?

---

★ **DI HALT**

24 Август 2008 в 19:36

По моему я упоминал, что все работы ведутся в авр студии. Там полный GUI так что никаких ключей и гемороя. Там же и отладка.

---

**Shorst**

15 Сентябрь 2008 в 19:28

Вопрос по строчке OUTI SPL,low(RAMEND): функция low(RAMEND) выбирает из RAMEND младший байт и по нему устанавливает конец ОЗУ МК, а старший байт подразумевается равным нулю.Т.к. значение RAMEND равно \$045f,то мы ограничиваем размер ОЗУ стеком до 95 байт (5F)?

---

★ **DI HALT**

15 Сентябрь 2008 в 19:35

Эт косяк. Я думал что уже везде поправил. Сей кусок кода была скопипащен из кода для Тини2313 у которой 128байт ОЗУ и значит там не требуется второй байт для адресации. У меги 8 уже 1кб. Конечно же надо еще загрузить старший байт

OUTI SPH,high(RAMEND)

Копипаст зло!

---

## VVV

16 Октябрь 2008 в 5:26

Возможно я чето пропустил, по этому не до конца понимаю почему все-таки ты всегда зажигаешь светодиоды именно нолем? Здесь есть какие-то преимущества или это просто дело привычки?

И еще, где можно посмотреть перечень контролеров Atmel (можно и не только Atmel) и их возможности? А то я пробовал что-то разузнать на их сайте — ничего для «домашнего» применения не встретил и ваше там только какие-то супермодные все...

---

## ★ DI HALT

16 Октябрь 2008 в 6:17

1) Дело привычки — C51 не умел тянуть ноги вверх

UPD:

Потер те две причины. Они к подтягу на 1 %) Остается только привычка. А вообще, последнее время, на реальных схемах делаю так, как удобней разводить. Иногда бывает и так и эдак в одной и той же схеме. А чтобы не путаться, то увязываю все в макросы. Типо ON/OFF

Блин гоню с недосыпа.

[promelec.ru/catalog\\_info/48/70/212/73/](http://promelec.ru/catalog_info/48/70/212/73/)

ну вот например.

Там же, в каталоге промэлектроники можно позырить такие же таблицы и на пики, и на МСП430 и на C51

---

**@mva**

30 Июль 2010 в 19:12

кстати, на STK500 (думаю и на 600 тоже) диоды тоже нулём зажигаются. Так что не у одного тебя такие привычки :)

---

**vanchester**

19 Октябрь 2008 в 9:53

DI HALT, нужна помощь.

Прошил в ATmega8535 твою программу Mega8Robot.asm, предварительно подправив inc-файл и прерывания. Воткнул в собранную схемку — все работает с точностью до наоборот (команда CBI заставляет светодиод светиться, команда SBI — гаснуть)... (для прошивки использую uniprof, программатор собрал по схемке, которая у тебя на сайте)

Например, такая программка:

```
.include «m8535def.inc»
```

```
.def acc = R16 ; Accumulator
```

```
.MACRO outi
```

```
LDI R16,@1
```

```
OUT @0,R16
```

```
.ENDMACRO
```

```
.CSEG
```

```
.ORG 0x0030
```

```
RJMP Reset
```

```
Reset: OUTI SPL,low(RAMEND)
OUTI SPH,High(RAMEND)
OUTI DDRB, 0xFF
main:
OUTI PORTB, 0x00
RJMP main
```

В симуляторе все как положено, все выводы PORTB «выключенные», а реально если между выводами и Vcc поставить диоды, они светятся. Попробовал заменить 0x00 на 0xFF, светодиоды перестают светиться. Где я чего напутал? Спасибо.

---

### ★ DI HALT

19 Октябрь 2008 в 10:03

а ты диод как ставишь? Тут ситуация такая:

Если диод подключен между +5 и выводом мк то от 0 на выходе он будет светиться.  
(ток идет с +5 через диод на 0 на ножке)

Если же включить между выводом и GND то будет светиться от 1.  
(Ток идет с +5 на ножке через диод на землю.)

Проследи путь тока через диод и все станет ясно :)

---

### **vanchester**

19 Октябрь 2008 в 11:20

Теперь понятно :)

---

**Sergey**

24 Ноябрь 2008 в 0:36

DI HALT возник вопрос по программе из статьи «Домашний терминатор» журнала Хакер.

Посмотрел текст прграммы:

1. ждем нажатия кнопки
  2. если да, то зажигаем лампу и подаем 1 к левому и правому двиг. и разрешения
- и т.д.

Потом зашил скомпиленный файл .hex, проверил па макетке — никаких реакций на нажатие кнопки (PB0 подключаем к земле) и последующие нажатия кнопок датчика касания.

Скомпилировал файл с прошивкой из исходных текстов — размер оказался другим (больше). Проверил на макетке. При подачи питания едет (есть разрешения и 1 на двигателях) нажимаю кнопку правый двигатель останавливается, не реагирует на датчики касания.

Подскажи где ошибка. Спасибо.

---

**Timy**

9 Январь 2009 в 0:47

долго мучил DiHalt-a ))) потом в сети нашол :

---

**Ridik911**

14 Апрель 2009 в 1:13

так.. получается у нас для определения будет ли порт вход или выход — ДДР если 1 то выход, если 0 то вход?  
например

```
ldi r16,0xff ;весь порт на выход
```

```
ldi r17,123
```

```
out ddrc,r16 ;вкл. порт на выход
```

```
out portc,r17;выводим число
```

я правильно понимаю надеюсь.. только студия тупит чота а мож я... когда уже непосредственно вывожу на ПОРТЦ некоторое число до 255 то оноже идет и по ПИНЦ.. иль так задумано (хапсим не поддерживает новую стройку студии)

---

### ★ DI HALT

14 Апрель 2009 в 1:35

Так и есть. Значение DDR определяет направление порта. 0 вход, 1 выход.

Если у тебя линия на выход настроена, то выдача значения в PORT одновременно выходит и в PIN, что естественно. Т.к. PIN это значение которое непосредственно на ноге.

Т.е. грубо говоря так:

DDR — направление

PORT — что мы хотели получить на выходе

PIN — что у нас реально получилось.

Т.е. пример, ты делаешь DDR на выход, выдаешь в PORT 1, ножка пытается установиться в единицу. Но вот засада — она на ноль оказалась по ошибке закреплена. В итоге транзистор сгорает, нога безвольно повисает в нуле, и несмотря на то что в PORT=1 из PIN считается 0, т.к. ножка сгорела и в единицу встать не может.

---

### Ridik911

14 Апрель 2009 в 21:10

>Но вот засада — она на ноль оказалась по ошибке закреплена. В итоге транзистор сгорает, нога безвольно повисает в нуле, и несмотря на то что в PORT=1 из PIN считается 0, т.к. ножка сгорела и в единицу встать не может.

—

это какбы контрпример таксказать..

а так понял. спасибо... надежды ждать разработчиков хапсим нет.. такчто запустил для проверки в VMLAB (чтото типо хапсим для

кодевизон).. только никак не получается подключить диоды нога-земля (получается нога-Vdd(+5)) — вообще инвертированность дает

.POWER VDD=5 VSS=0 ; Power nodes

.CLOCK 32k ; Micro clock

.STORE 250m ; Trace (micro+signals) storage time

D1 VDD PD0 ; LED code.

D2 VDD PD1 ; (since no more LEDs are available

...

D8 VDD PD7 ;in the control panel)

, но для проверки и наглядности достаточно

---

## Security

25 Апрель 2009 в 3:21

незнал куда написать, но проблема в программировании.

подключил 12-ти разрядный АЦП(AD7892-2) к микроконтроллеру (АТmega16), к порту Ц(младшие 8) и к порту Б(старшие 4). выставляю данные на выход АЦП (на входы портов), читаю порты:

```
buffer[0]=PINC;
```

```
buffer[1]=PINB & 0x0F;
```

получаю например такой результат:

```
buffer[0]: 0x42
```

```
buffer[1]: 0x00
```

потом беру меряю тестером напруги на ножках микроконтроллера:

порт C = 0x78

порт B = 0x00

я так понимаю что непраьно я считываю инфу с портов. оба порта настроены так:

DDRC=0x00; // порт C вход

PORTC=0xFF; // подтягивающие резисторы подключены.

подскажите в чем я ошибся. или предложите альтернативу решения данной проблемы.

---

### ★ DI HALT

25 Апрель 2009 в 3:28

Считываешь правильно и настроил порт ты тоже как надо.

Т.е. у тебя считанное не равно тому что реально на входе? А ты точно уверен в том, что выходное значение АЦП не плавает? Отключи АЦП и подай какойнибудь левый код вручную.

---

### Security

25 Апрель 2009 в 22:45

«Т.е. у тебя считанное не равно тому что реально на входе?» да именно.

выход АЦП не плавает (тыкал в каждый вывод счетчик, на ТТЛ, т.е. скорость реакции бешанная 5нс) ниче не щитает. четко показывает «1» «0».

А не может быть такой хитрости, что порт C вообще не предназначен для считывания инфы, а только на вывод? Просто в примере по которому я делаю, используется вместо порта C порт A, но я подумал, что это не принципиально и подпаял на C.

---

### ★ DI HALT

25 Апрель 2009 в 22:50

Кажись понял в чем косяк! На этом порту ЕМНИП висит JTAG и он по дефолту вроде бы включен. Выруби нах в фузах бит JTAGEN

---

### Security



25 Апрель 2009 в 23:18

Пусть за мной будет замечена гомосятина, но я тебя расцелую!!! ;)

Да, по умолчанию оказалось JTAG включен. Скинул флаг начало читать то что подает АЦП.

Огромное спасибо!

---

### ★ DI HALT

25 Апрель 2009 в 23:24

Старые грабли. Я сам никогда мегу16 не юзал, а про них где то упоминалось уже.

---

### bdp

14 Май 2009 в 20:58

Проемулировал программку в AVRStudio, и тут такая лажа приключилась с кнопкой старта:

Значения DDRB,0 = 0 и PORTB,0 = 1 а вот значение PINB,0 осталось равным «0», и поэтому програма проходила дальше (а кнопку то я типа и не нажимал). Таже лажа с контролем припятствий.

Насколько я помню по предидущей лекции это глюк самой AVRStudio, или я гдето лоханулся.

Набранный код перепроверил три раза O\_o

---

### ★ DI HALT

14 Май 2009 в 21:53

Да, студия сама PIN не выставляет. Откуда ей знать какой он должен быть? Его надо вручную задавать.

---

### SIM

8 Август 2009 в 3:54

В предыдущем примере где на ноги выводили похожее на 0101010 (hex AA) небыло теблицы прерываний. Начинало работать сразу основное тело программы?

Программа начиналась по адресу

.ORG 0x0030

я так понимаю при включении микроконтроллер кидается на адрес 0x0000 там «пор» и так ползет 30 тактов до адреса 0x0030? или сразу как-то переходит?

2. Вход reset я так понял может использоваться как порт ввода-вывода? В том же Attiny2313 он помечен как (RESET/dW) PA2. Как тогда сбросить контроллер? непонятно пока что :-)

---

### ★ DI HALT

8 Август 2009 в 14:53

Да там стартует на 0000 потом ползет через всю таблицу прерываний до 30 адреса и оттуда выполняет.

Но по правильному бы лучше сделать так:

ORG 0000

RJMP RESET

[таблица векторов прерываний]

ORG [end vectors size]

RESET:

Сейчас подправлю там текст проги, чтобы сразу было как принято.

2. Да его можно использовать как порт, при этом теряется возможность внешнего сброса и **ТЕРЯЕТСЯ ВОЗМОЖНОСТЬ ПРОШИВКИ ЧЕРЕЗ SPI** только параллельным программатором на 12вольтах.

---

### RunningWolf

19 Октябрь 2009 в 1:51

Во-первых, огромное спасибо автору. Хорошая подача материала, действительно помогло перейти от раздумий «хорошо бы...» к делу не имея никаких навыков в электронике.

Респект!

А теперь по теме: =)

Собрал программатор Громова. Воткнул USB-COM — получил COM26. Черт! Uniprof только до COM5. Ладно, бог с ним — попробовал avrdude — фиг вам! А, чтоб тебя..!

Достал с антресолей маму с PIII и COM-портами на маме, воткнул программатор — как ни странно, заработало сразу. Ура!

Т.к. на кошках тренироваться скучно, решил прошить программку из этой статьи.

Подключил, правда, только:

— кнопку «Старт» (PB0)

— кнопку «Левый передний датчик» (PC0)

— светодиоды: отладочный(PC3), левый разрешен(PB2), левый направление(PD2, PD3)

Прошил, включил и... оно не работает. МК: Atmega8-16PU. После запуска загораются вообще все светодиоды и на кнопки не реагирует.

Подумав, пошел в обход и таки получил, простую прогу, которая зажигает желаемые лампы при нажатии на те или иные кнопки.

Т.о. проблемы либо в схеме, либо в программке, либо в Меге либо, во мне =)

По проге из этой статьи:

1) Входы подтянуты к 1-це. Это логический 0 или логическая 1?

2) Светодиоды подключены к +5В. Когда контроллер чисты (не прошит) — не горит ничего. Т.е. на всех выходах — логическая 1.

Судя по тому, что при запуске проги из этой статьи сразу загораются все светодиоды — значит на всех выходах сразу устанавливаются нули. Т.е. проходят CBI для каждого порта.

Где они есть в этой программке? Это макросы LIGHT\_ON, LDR\_DISABLE, L\_Run\_F, L\_Run\_B, L\_Stop. Т.е. в самых разных частях.

Такое впечатление, что оно после запуска крутится в главном цикле, игнорируя все условия.

3) Изначально все входы находятся в одном состоянии (либо 0, либо 1, в зависимости от ответа на вопрос 1 =)). Нажатие кнопки меняет состояние порта на противоположное (пока кнопка нажата). Команда SBIC проверяет вход=0. SBIS проверяет, что вход = 1. Как тогда в

одном месте для проверки нажатия используется SBIC, а в другом — SBIS? Ведь изменение уровня от кнопки в обоих случаях одинаковое. Или как?

---

### ★ DI HALT

19 Октябрь 2009 в 2:48

«Собрал программатор Громова. Воткнул USB-COM — получил COM26»

А толку, Громов все равно не работает с этими конвертерами. Хотя номер порта можно и сменить в диспетчере оборудования.

1) Это вход с подтягом. Они подтянуты к 1це. Т.е. когда кнопка не нажата на порту 1. Нажатая кнопка садит порт на 0

2) Когда контроллер чист на всех выводах HiZ т.е. считай что они вообще никуда не подключены.

А теперь смотрим что в программе:

Мы при инициализации порты светодиодов сразу же выставляем на выход:

```
SBI DDRD,LDR0 ; А тут..
```

```
SBI DDRD,LDR1 ; микросхемой..
```

```
SBI DDRB,LDRE ; двигатели....
```

```
SBI DDRC,RDR0
```

```
SBI DDRC,RDR1
```

```
SBI DDRB,RDRE
```

Но! При старте во всех регистрах PORT по дефолту стоят нули. А когда мы DDR устанавливаем в 1, то порт становится на выход и в результате (т.к. PORT = 0) на ноге у него получается 0, вот диоды и зажигаются все сразу.

3) Ты не совсем вкурил в алгоритм. SBIS и SBIC это не команды проверки кнопки. Это проверка условия.

В первом случае (кнопка) нам надо понять когда она нажата и по этому факту выскочить из цикла.

А во втором случае нам удобней отслеживать другое условие, т.е. пока не нажата пропускаем переход на StopAll. Как только нажата — переходим и стопорим все движки.

использовать можно sbic и sbis в зависимости от того как нам удобней в данный момент.  
т.е. пропустить команду если бит есть или пропустить команду если бита нет.

---

### **RunningWolf**

20 Октябрь 2009 в 3:59

Спасибо за ответ. Картинка стала полнее. Однако предложенный тобой код все равно не хочет жить. И есть у меня подозрение, что не живет он из-за того, что подтяг не работает.

Вот мой код:

```
.include «m8def.inc»
```

```
.MACRO outi  
LDI R16,@1  
OUT @0,R16  
.ENDMACRO
```

```
.CSEG  
.ORG 0x0000  
RJMP RESET
```

```
.ORG 0x0030
```

```
RESET:
```

```
CBI DDRB,0  
SBI PORTB,0
```

```
CBI DDRC,0  
SBI PORTC,0
```

```
SBI DDRC,3  
SBI DDRB,2
```

```
SBI DDRD,2  
SBI DDRD,3
```

```
; Init stack  
OUTI SPL,low(RAMEND)  
OUTI SPH,High(RAMEND)
```

```
Main:  
;SBIS PINB,0  
SBIC PINB,0  
RJMP Pause
```

```
; Off light  
SBI PORTC,3  
SBI PORTB,2
```

```
; On Light  
CBI PORTD,2  
CBI PORTD,3
```

```
Pause:  
;SBIS PINC,0  
SBIC PINC,0  
RJMP Main
```

```
SBI PORTD,2  
SBI PORTD,3
```

```
CBI PORTC,3  
CBI PORTB,2
```

```
RJMP Pause
```

Как оно живет «в натуре»: После запуска загорается все (PD2, PD3, PC3, PB2). При замыкании на землю PB0 остаются гореть только PD2, PD3. При заземлении PC0 наоборот горят только PC2, PB3.

Если я правильно понимаю, то при наличии подтяга (он, вроде обеспечивается инструкциями CBI DDRB,0 и SBI PORTB,0) на порту, если его никто не трогает, всегда висит 1-ца. Т.е. пока держишь землю — идет ноль. Отключил землю — порт ушел опять в 1. Так? Если да, то не понятно как может крутиться блок Pause (горят только PC2, PB3), когда я уже отпустил кнопку (снял землю) на PC0? Ведь тогда PC0 опять встанет в 1, SBIC PINC,0 не сработает и при следующем же цикле оно должно уйти по RJMP Main. Так?

С другой стороны, если подтяга нет, то не понятно, как крутится блок Main (горят только PD2, PD3)? Ведь опять-таки тогда должны все время срабатывать оба условия и гореть все диоды.

Вобщем, что-то я не догоняю :( Не дай помереть дураком =)

Подтяг проверяется просто — в свободном состоянии на выводе должен быть +5в

---

★ **DI HALT**

20 Октябрь 2009 в 4:35

Так, стоп. Где ты у меня паузу увидел?

---

**RunningWolf**

20 Октябрь 2009 в 18:56

Ты про «блок Pause»? Это я логику своей проги (см. пред. пост) имел в виду.

У тебя, я так понимаю, циклы только:

- 1) Ожидание нажатия старт (Main: ... RJMP Main)
- 2) Ожидание сигнала от датчиков (Seek: ... RJMP Seek)
- 3) Общий цикл (Main: ... последний RJMP Main)

Я про свою прогу. Никак не могу понять, как оно уходит в цикл Pause когда PC0 уже НЕ заземлен. А в реальности он туда уходит (стоит только коснуться землей PC0 и затем убрать землю), т.к. PB2 и PC3 продолжают гореть, а PD2, PD3 — погашены.

---

★ **DI HALT**

21 Октябрь 2009 в 4:22

Дребезг контактов учел?

---

**RunningWolf**

20 Октябрь 2009 в 4:01

А про USB-COM и Громова. Вроде кто-то писал, что иногда оно работает. Надо же было попробовать =) Да и все-равно оно надо для USBASP.



---

**SergeyDon**

30 Октябрь 2009 в 20:19

чем отличаются например команды: JMP и CALL? ну кроме количества тактов исполнения.  
ставил и так и так работает...

---

**★ DI HALT**

30 Октябрь 2009 в 20:29

Принципиально отличаются.

JMP это просто переход.

А CALL это переход с запоминанием точки перехода и по команде RET мы в эту точку вернемся.

Мало того, часто делая CALL мы можем сорвать стек. Так что за каждым CALL должен следовать RET На какую глубину ушли из нее же возвращаемся.

---

**SergeyDon**

30 Октябрь 2009 в 21:51

СПС.

я разобрался!

просто я стек не инициализировал, и у меня при вызове что JMP, что CALL

при достижении Ret возвращалось на адрес 0x0000.

на сайте daw.ru там даташит с переводом на русский, так там вообще нет про стек и его инициализацию... ничего нет!

---

**Sergas5**

3 Февраль 2010 в 11:40

Привет всем! DI HALT, о какой отладочной плате идет речь? О той, которая в разделе «Демоплата» или которая в разделе «Создание печатной платы методом лазерного утюга». Кстати, где можно прочитать о последней — что-то нигде не нашел. Заранее спасибо!!!

---

### ★ DI HALT

3 Февраль 2010 в 14:07

О любой. Изначально было о той что в статье про ЛУТ.

А нигде. Тогда еще не было четкой логики сайта поэтому и не было отдельной статьи.

---

### A\_ndrej

6 Февраль 2010 в 22:54

Не могу понять.

В вашей программе в этом куске:

» ; Датчики касания

.equ L\_FNT = 0 ; Правый передний

.equ R\_FNT = 5 ; Левый передний

.equ L\_BCK = 4 ; Левый задний

.equ R\_BCK = 3 ; Правый задний

; Кнопка запуска

.equ ST\_BTN = 0

; Двигатели.

.equ LDR0 = 2 ; Левый двигатель бит 0

.equ LDR1 = 3 ; Левый двигатель бит 1

.equ LDRE = 2 ; Левый двигатель разрешение

```
.equ RDR0 = 2 ; Правый двигатель бит 0  
.equ RDR1 = 1 ; Правый двигатель бит 1  
.equ RDRE = 1 ; Правый двигатель разрешение
```

```
; Сигнальные лампы  
.equ LIGHT = 3
```

/\* Откуда я взял эти цифры? А поглядите на номера портов к которым прицеплены все эти датчики и двигатели. Вот! Т.е. я просто обозвал биты как мне было удобно. Чтобы не держать в памяти где у меня что подключено. \*/ »

я не понимаю, не откуда взялись эти цифры(уже понял), а как эти биты не путаются с такими же битами других портов? Ведь в каждом порту есть биты 0...7. Как же ваши биты не путаются с остальными?

---

#### ★ DI HALT

6 Февраль 2010 в 23:56

Ну так я же явно указываю имя порта когда обращаюсь к этим битам

---

#### A\_ndrej

8 Февраль 2010 в 15:30

Как указываете?

В этом же куске кода?

В упор не вижу.

---

#### ★ DI HALT

8 Февраль 2010 в 21:18

Ну вот, например:

CBIDDDRC,L\_FNT

DDRC нам какбэ намекает, что мы пишем в регистр DDR порта C

---

### **MWAndry**

7 Февраль 2010 в 22:50

DI-HALT, объясни пожалуйста, зачем ты в определениях обозвал регистр R16

.def ACC = R16 ; Accumulator

Ведь дальше по коду это имя не встречается.

В макросе OUTI ты используеш R16 а не ACC.

Это очепятка или я что-то пропустил?

---

### **★ DI HALT**

7 Февраль 2010 в 23:39

Да это привычка из C51 ассемблера. Там был ACC как основной регистр. А тут как то забылось уже.

---

### **MWAndry**

8 Февраль 2010 в 0:32

Тобиш если я эту строку удалю из кода, ни на что это не повлияет?

---

### **★ DI HALT**

8 Февраль 2010 в 0:54

Скорей всего нет.

---

**MWAndry**

8 Февраль 2010 в 1:09

Спасибо за опаретивный ответ.

Отдельное спасибо вообще за все учебные статьи. Спаял програматор Громова, удалось прошить контроллер.

Но что то не так со статьей про UART. Жму «читать полностью» а мне выбрасывается вот такой символ «<I» и ничего больше.

---

★ **DI HALT**

8 Февраль 2010 в 1:40

Какая именно статья глючит? Только что проверил статью «Передача данных через UART» нормально она открывается.

---

**MWAndry**

8 Февраль 2010 в 12:59

С рабочего компьютера всё открылось и у меня. Значит домашняя сеть барахлит. Звоняйте за беспокойство.

---

**MWAndry**

8 Февраль 2010 в 1:18

Оперативный ответ. Звиняйте, часто очепятываюсь.

---

**A\_ndrej**

8 Февраль 2010 в 20:41

Есть один вопрос.

Хочу в качестве своей первой программы сделать спидометр на моторолер.

Поцеплю на диск колеса магнит, на барабан геркончи или датчик Холла, и буду ловить их показания АТмега восьмеркой.

После прочитаного, казалось, что может быть проще)))

А когда начал разбираться то появилось сколько подводных камней, что я теперь не вылажу с нета в поисках каких то аналогий. А везде всякие спидометры и тахометры настроены только на свои параметры колеса, и даются только hex файлы.

Хожу теперь в ступоре, по пол ночи не могу заснуть(все это в голове копошится, а отказываться от идеи не хочется(уже сколько времени на нее потратил).

Как я понимаю для того что бы измерить скорость, нужно измерить период приходящего импульса.

Я думаю что один таймер нужно запитать от внешнего источника(от геркона), а второй пустить от встроенного генератора.

Когда первый импульс с геркона приходит, то второй таймер запускается и начинает считать.

Как только приходит второй импульс то второй таймер выдает свои подсчеты(время между импульсами), и начинает считать заново до прихода нового импульса.

Все это потом считается по формуле  $V=L/t$ ? где V- скорость, L-окружность колеса(то есть пройденный путь),t- время(результат подсчета второго таймера).

Так, все понятно, а как это расписать в коде ума не приложу.

Может что подскажете?

Или ссылочку какую то дадите с примером?

P.S. Еще подскажите пожалуйста как делить и умножать в AVR?

---

## ★ DI HALT

8 Февраль 2010 в 21:10

Проще наверное даже так сделать: геркон на прерывание INT. По первому импульсу запускаешь таймер, по второму останавливаешь и считаешь натиканное.

Городить вычисления думаю тебе не нужно. У тебя же не высокоточная схема. Проще сделай табличные вычисления. Забабахай себе таблицу скоростей с шагом в 0.5км/ч под твоё колесо или как тебе ещё надо. И последовательно сравнивай натиканное по таблице пока не будет больше/меньше (смотря с какой стороны пойдёшь сравнивать) это и будет текущая скорость. Будет быстрее и компактней.

---

**A\_ndrej**

8 Февраль 2010 в 21:23

Спасибо.

Как отсеживать понял. А как изобразить в коде сравнение с заданными таблице значениями?

Это надо чтобы ставился флаг C, а потом отлавливать его командами BRSH и BRLO?

Если да, то как сделать так чтобы флаг C ставился, и куда его вообще ставить чтобы потом можно было сравнивать с заранее заданными значениями?

---

### ★ DI HALT

8 Февраль 2010 в 21:40

Да. Делаешь CP Rn,Rn двух регистров где у тебя табличное и текущее. И на основании этой команды выскочат флаги, ну а там BR\*\* и все.

---

### A\_ndrej

8 Февраль 2010 в 21:44

Аааааааааа!!!

---

### A\_ndrej

8 Февраль 2010 в 21:47

Дошло))))

Большое спасибо))

---

### A\_ndrej

9 Февраль 2010 в 1:58

А как задать таймер?

Что то типа:

ldi temp, 0x01; без деления в предделителе  
out TCCR0, temp  
?

Так как прерывания по переполнению не надо, то разрешение на прерывание по переполнению не ставить?  
Чтобы сбить таймер в ноль надо в предыдущем коде в temp записать ноль, или  
написать clr TCCR0?  
А как снять количество натиканных импульсов с таймера?  
Написать out temp, TCCR0 или как?  
Если можете — ответьте.  
Не обижайтесь пожалуйста на бесконечные расспросы :-), — хочется все таки разобраться.

---

### ★ DI HALT

9 Февраль 2010 в 2:36

Про таймеры есть отдельная статья. там все расписано.

У таймера дофига регистров. TCCR0 это только один из них- управляющий. Есть еще регистры масок прерываний, и еще два управляющих регистра. ;)))

А счет идет в счетном регистре TCNT И вот оттуда надо считать. Ну и про сброс предделителя не стоит забывать ;)

---

### Igor

10 Февраль 2010 в 13:45

День добрый.

У меня есть один вопрос.

Можно ли реализовать как на некоторых китайских игрушках.

Несколько кнопок и задаю программую,кнопка вперед,влево,право и тд. после нажимаем



старт и танк исполняет то, что там напрограммировал. для каждого нажатия к примеру:  
20 см вперед или назад, 10 град. поворот влево или право и тд.  
Реально это или нет?

---

★ **DI HALT**

2 Апрель 2010 в 22:16

Реально. Но придется писать что то вроде небольшой виртуальной машины.

---

**Dmitri**

2 Апрель 2010 в 21:51

Команда JMP 0x000001 не работает что-то в mega8. Здесь говорят <http://www.avrfreaks.net/index.php?name=PNphpBB2&file=printview&t=33273&start=0> что она не айс и её закрыли.

---

★ **DI HALT**

2 Апрель 2010 в 22:14

В Мега8 нет такой команды. Там нет и особой нужды сигать дальше чем RJMP — память коротка.

---

**krestic**

4 Апрель 2010 в 2:33

Чудища мне что коменты тут совсем не к этой статье:) Диоды, «Домашний терминатор»- а в статье только про НОПы...  
А в чем великий смысл ставить такое количество нулей в 0x000001 ? Почему именно шестизначное число, а не восьми или двух?

---

★ **DI HALT**

4 Апрель 2010 в 8:51

Привычка адреса указывать так.

---

### **Shuterkent**

17 Апрель 2010 в 0:41

Читал я читал множество статей твоих прочитал ДиХалт но так и не вкурил, допустим мне нужно выставить к примеру Порт PB4 на вход, так вот как я понимаю мне нужно регистр DDRB выставить на 0, а порт в зависимости, что я хочу там увидеть или PORT равен 1, или PORTB=0, и еще правильно ли я понял, если у нас в регистре PORTB равен 0 то при нажатие на кнопку он менятся на 1???. А теперь собственно вопрос что я должен увидеть в AVR studio, когда соответствинные регистры выставлены, и как понять что это 1 и что это 0 и где???. Просто я только начинаю разбираться в микроконтроллерах, а сайт у тебя классный очень много интересного даже его выкачал себе)))!

---

### **★ DI HALT**

17 Апрель 2010 в 10:21

На вход DDR=0 это всегда.

А PORT определяет режим работы входа.

1 — «активный» вход с подтягом к +5, например для отслеживания кнопок.

0 — пассивный вход с Hi-Z. Для отслеживания шин всяких.

Реальное Значение из порта БЕРЕТСЯ В РЕГИСТРЕ PIN!!!

---

### **Shuterkent**

17 Апрель 2010 в 18:11

тоесть, если я выставил DDR=0, PORT=1 то только тогда я могу подключить кнопку, а реальное значение вкл. она или не вкл. я просматриваю в PIN. Правильно я понял???

---

## ★ DI HALT

17 Апрель 2010 в 18:38

Да

---

### Shuterkent

17 Апрель 2010 в 18:53

Вот только начал читать про макроассамблер там где простейшая программа, так вот я не понял как работает логика контроллера, а может и не увидел. Ну к примеру есть у меня:

Порт первый

DDR=0 Port=1-это я подкл. кнопку

Порт второй

DDR=1 Port=0-это я подкл. светодиод

Так вот это первоначальное состояние.

Так вот нажал я кнопку, мне нужно что б светодиод засветился так что Port=0 должен выставиться на Port=1.

Как это пишется с помощью ассамблера??!

---

## ★ DI HALT

17 Апрель 2010 в 19:40

Эм... вообще PORT это целый байт. И в нем 8 бит, а всего таких портов у МК несколько может быть. Так что у тебя на одну ногу кнопка, на другую светодиод. А говоря про DDR=0 я имел ввиду КОНКРЕТНЫЙ БИТ порта.

Например DDRD.7=0, а DDRD.6=1 и одна нога работает на вход, другая на выход.

Соответственно на ассемблере настройка порта таким образом делается так:

LDI R16,0<<7|1<<6  
OUT DDRD,R16

---

### **Shuterkent**

17 Апрель 2010 в 22:07

аааа понял! спасибо

---

### **Kots**

21 Апрель 2010 в 17:33

DI HALT, а какой прогой выделение жёлтым организуется? Оч удобно и наглядно, а сам вкурить метод никак не могу. =)

---

### **★ DI HALT**

21 Апрель 2010 в 17:51

FastStone Viewer — рисование — выделение.

---

### **Kots**

21 Апрель 2010 в 22:05

Спасибо)

---

### **sanjok**

1 Май 2010 в 20:57

можно поподробнее о регистрах R0-R31, не могу понять с чем их есть,  
это временная память которая работает с всеми функциями,  
я должен постоянно в нее загружать, а потом с ней работать, и обратно в оперативку?

---

### ★ DI HALT

1 Май 2010 в 21:05

Это 31 ячейка которыми ты можешь распоряжаться как угодно, но поскольку только с ними можно выполнять математические и логические операции, то они используются как временные оперативные регистры, а результаты сохраняются в оперативку, т.к. регистров на все не хватит.

---

### **sanjok**

1 Май 2010 в 23:18

спасибо

---

### **Sergas5**

8 Май 2010 в 12:21

DI HALT, а где можно взять систему команд для микроконтроллера желательно на русском языке?

---

### ★ DI HALT

8 Май 2010 в 15:19

Я пользуюсь той, что в книге Евстифеева Контроллеры AVR серии Мегы как то так. (у меня в разделе книги вроде бы еще есть ссылка). Вообще хорошая книжка — перевод даташита по сути.

---

### **k0ndr**

18 Май 2010 в 22:46

Странно... При запуске кода с командами LDI и JMP студия (v.4.18.700) выдаёт такую ошибку:

D:\Program Files\Atmel\Projects\3.asm(4): error: Undefined symbol: ZL

D:\Program Files\Atmel\Projects\3.asm(4): error: Invalid register

D:\Program Files\Atmel\Projects\3.asm(5): error: Undefined symbol: ZH

D:\Program Files\Atmel\Projects\3.asm(5): error: Invalid register

D:\Program Files\Atmel\Projects\3.asm(15): No EEPROM data, deleting D:\Program Files\Atmel\Projects\3.eep

Assembly failed, 4 errors, 0 warnings

Что может быть не так?

---

### ★ DI HALT

19 Май 2010 в 1:38

Команды покажи? Может у тебя inc файл не подгрузился и символические имена не подцепились?

---

### SphiNx

15 Декабрь 2010 в 0:14

Аналогичная проблема, понимает только

ldi r27,low(m1)

а через XH никак.

версия 4.18.716

---

### k0ndr

19 Май 2010 в 1:54

Оу, точно, когда перезапускал студию, забыл inc подгрузить. Спасибо, заработало нормально.

---

### dima\_m

12 Июнь 2010 в 13:25

Чисто условный пример, чтоб понять как идет запись данных во флеш память.

1. Допустим я хочу записать число по адресу \$0000. Как я понял оно запишется в ячейку 0, а не в 1?
2. И если число увеличивать, то после 255, данные начнут записываться в ячейку 1?
3. То есть запись будет идти как обычно только задом наперед (little-endian) ?

ячейка ячейка

\$0000 — 0 1

\$0001 — 2 3

\$0002 — 4 5

---

### ★ DI HALT

12 Июнь 2010 в 14:25

Адресация флеша идет в словах (компилятору так удобней) т.е. 0000 это адрес первых двух байт (начала) 0001 адрес вторых двух байт и так далее.

В реальности (когда мк сам обращается к данным во флеше) же адрес побайтный, 16ти разрядный.

Когда указываешь адрес 0000 то запишется в реальный адрес 0000 в 0001 ты записать отдельный байт напрямую не сможешь. Только сгруппировав его в слово с другим байтом, либо вручную вписав в хекс.

2. Не понял вопроса.

Да, там младший байт по младшему адресу.

---

### dimam

13 Июнь 2010 в 9:16

2. Вопрос некорректный, ладно опустим его.

---

**SergeyDon**

11 Январь 2011 в 1:02

разум отказывается въезжать:

«...Т.е. двум байтам соответствует один адрес. Но, на самом деле это интерпретация адресации компилятором, микроконтроллер же может оперировать в памяти программ с точностью до байта...»

пример:

0000 \$A1 \$F0 // коды от балды

0001 \$00 \$0C // тоже от балды

0002 \$94 \$02 // аналогично

0003 \$A2 \$FF // ну вы поняли

1. каким образом например прочитатть только младший байт по адресу 0001 (который в примере =\$0C)?

2. если контроллер может оперировать с точностью до байта, то тогда должна быть возможность перехода на «половинчатые» адреса... (например в промежутке между 0001 и 0002, таким образом код следующей команды будет интерпретироваться как \$0C \$94 — а это уже RJMP... вроде так делали защиту кода от «ламеров дизасемблеристов» на Z80).

3. если в код примера добавить строчку чтения байта по адресу на который указывает метка то там будит обломс...:

M1:

NOP

NOP

LDI ZL,low(M2) ;

LDI ZH,High(M2)

lpm r16,Z+ // тут обломс.... ожидаем в R16 код комады NOP, а там лажа!



JMP // а тут отработываем как надо ;)

NOP

M2:

NOP

RJMP M1

NOP

--==--

LPM — чтение из памяти программ и инкремент.

до выполнения инкремента Z=\$09, после Z=\$0A. Вроде все логично прибавили 1 и команда JMP отработала как надо.

а вот в R16 считали лабуду с непонятного адреса... опытные люди подсказали что нужно \*2 но разве это правильно?

P.S. что-то тут коменты 90% вообще отношения к теме не имеют... какие-то порты... компиляторы... программаторы... коды своих нерабочих творений по выкладывали... BCE на форум :)

---

### ★ DI HALT

11 Январь 2011 в 9:18

LDI ZL,low(M2\*2)

LDI ZH,High(M2\*2)

Вот так надо. Т.к. у компилятора адреса в словах. Умножая их на два мы переводим их в байты и тогда с ними можно работать. И тогда ты загрузишь в R16 код NOP

---

### SergeyDon

11 Январь 2011 в 16:57

да про умножить на 2 я понял уже...

зачем это сделано? только с толку сбивает.

есть метка на адрес, пусть она и остается меткой на адрес...

директива .org 0x0010 реально в памяти начинается с адреса 0x0020!

только студия так извращается?

---

### ★ DI HALT

11 Январь 2011 в 17:01

Просто метки в студии они для компилятора и переходов. А переходы адресуются в словах, потому так и сделано.

---

### ★ DI HALT

11 Январь 2011 в 9:21

1.

В словах только компилятор оперирует. Видимо для его удобства. В реальных адресах карта памяти будет выглядеть так:

0000 \$A1 \$F0 // коды от балды

0002 \$00 \$0C // тоже от балды

0004 \$94 \$02 // аналогично

0006 \$A2 \$FF // ну вы поняли

И можно будет взять любой байт какой захочешь. А компилятор (через метки или еще какие способы вручную скормить ему адрес) дает тебе только адрес в словах.

---

### SergeyDon

11 Январь 2011 в 17:04

«Команда JMP, как и все команды перехода, работает просто — записывает в Program Counter свой аргумент. В нашем случае — 0x000001.»

странно но в PC записывается 0x000002; и вообще что-то я в PC записать нечетное число никак не могу! даже в hex правка не помогает все время четное тыкает (я в протеусе проверяю).

через стек тоже адрес умножает на 2:

```
LDI R16,0x11
```

```
PUSH R16
```

```
LDI R16,0x00
```

```
PUSH R16
```

```
RET // в PC грузит 0x000022!
```

надо в реальном МК попробовать...

---

#### ★ DI HALT

11 Январь 2011 в 19:37

В AVR Studio только что записал в PC 1 и сразу же произошел переход на первый вектор прерывания (фактический адрес в байтах 0x0002 (следующий после Reset) PC то щелкает словами. Т.е. 1 ведет на адрес 0x0002 и так далее.

---

#### ★ DI HALT

11 Январь 2011 в 19:38

Меньше пользуйся всякими левыми глючными симуляторами :)

---

**A\_ndrej**

11 Июль 2010 в 18:47

Подскажите пожалуйста можно ли писать так:

in XL,ICR1L

in XH,ICR1H

ldi Y,868

ср Y,X

В смысле можно ли оперировать сразу X или Y не расписывая на старшие и младшие разряды.

И нет ли ошибки в выражении ldi Y,868

или вообще в том что я накалякал?

---

#### ★ DI HALT

11 Июль 2010 в 19:30

Нельзя. Проц то у нас 8ми разрядный. Он не может сравнить два байта сразу. Да и регистра такого Y или X нету как таковых. Они сделаны лишь для удобства — не более чем просто обозначение. На деле есть только XL XH (которые тоже не более чем обозначение, т.к. это R26,R27)

А вот IN XL,ICR1L вполне возможно. Т.к. это тоже самое что и  
IN R26,ICR1L

---

#### A\_ndrej

11 Июль 2010 в 19:44

Спасибо, дошло)

---

#### A\_ndrej

11 Июль 2010 в 19:49

Спасибо, дошло)

А как тогда выполнить сравнение с числом

di Y,868

ср Y,X

?

Разделять его на два разряда?

---

### ★ DI HALT

11 Июль 2010 в 20:31

Сравнивать побайтно, сначала младшие . а потом старшие. Старший сравнивать через команду CPC, чтобы с учетом переноса было.

---

### A\_ndrej

11 Июль 2010 в 20:45

Щас мозг лопнет:-)

---

### ★ DI HALT

11 Июль 2010 в 20:49

Смотри. Есть X и Y

CP XL,YL

CPC XH,YH

BREQ ....

И все, и не надо лопать мозг. Таким образом, кстати, можно щелкать сколько угодно байтные переменные. Просто наращивая цепочку CPC операторов.

---

**A\_ndrej**

11 Июль 2010 в 21:02

А как игреку присвоить число 868? преобразовать в двоичный код — и разделить на 11 и 01100100.

ldi YL,1101100100

ldi YH,11

Так что ли?

---

★ **DI HALT**

11 Июль 2010 в 21:09

Ага делишь число просто на биты и все. Но есть способ проще — доверь это компилятору!

LDI YL,Low(868)

LDI YH,High(868)

И все, компилятор сам разложит все по байтам.

---

**A\_ndrej**

11 Июль 2010 в 21:27

Понял.

Блин, ну и головоломка:-)

Это я пытаюсь писать прогу для спидометра которую закинул полгода назад.

Использую вывод ICP1 для геркона и сравниваю то что нащелкал счетчик до захвата, с просчитанными цифрами чтобы индцировать скорость на экране.

А вот когда мне нужно будет сравнить не двух разрядное число с таким же 2х разрядным показателем счетчика, а одноразрядное число(например 23) с двух разрядным показанием счетчика? Как быть?

Также:

LDI YL,Low(23)

LDI YH,High(23)

просто он поставит в YH сплошные ноли, или надо делать как то по другому?

---

#### ★ DI HALT

11 Июль 2010 в 21:29

>просто он поставит в YH сплошные ноли

А почему тебя это смущает? ;)))) Так и должно быть!

---

#### A\_ndrej

11 Июль 2010 в 21:33

Да не, просто интересно, пусть ставит, лишь бы работало:-)

---

#### Anhanger

15 Июль 2010 в 0:15

Объясните, пожалуйста...начинающему)

Почему может не сбрасываться память в студииABP 4?

Допустим, вбиваю код 3 NOPа, в «программ мемори» в трех ячейках 6 байтов нулей после запуска эмуляции. Далее стираю один из NOP, т е остается их два. Нажимаю РЕЗЕТ, затем снова ЭМуляцию. По идее, должно остаться четыре байта нулей...(?) А-н-нет, их остается все также шесть. Почему)? (криво установилась прога или криво с ней работаю)?)

---

#### ★ DI HALT

15 Июль 2010 в 0:23

А перекомпиляцию проекта кто будет делать?

---

**Anhanger**

15 Июль 2010 в 0:45

Что нужно для этого нажать)?

---

**★ DI HALT**

15 Июль 2010 в 0:50

F7

---

**Anhanger**

15 Июль 2010 в 1:01

Нажимал(((

Напишите, пожалуйста, тогда последовательность нажатия клавиш

Допустим, запущена эмуляция с тремя этими командами НОП. Далее жму РЕЗЕТ, далее СТОП, далее Ф7, далее КОНТРЛ Ф? или  
Старт Дебаггинг.

ПЕРЕД нажатием на резет удаляю третью команду НОП)

---

**★ DI HALT**

15 Июль 2010 в 1:07

В эмуляции удали любую команду NOP и попробуй исполнить эмуляцию на еще один шаг дальше. Компилятор спросит у тебя разрешение перекомпилировать проект (вопрос Rebuild project?) отвечай да и проект будет перекомпилирован, а дебаг запущен заново. Тут то та команда и исчезнет

---

**Anhanger**

15 Июль 2010 в 1:18



...Простите за мое невежество...

1. Проект с тремя НОП запущен
  2. Удаляю один последний НОП
  3. Нажимаю Ф11
  4. Прога спрашивает «РеБилд?» — нажимаю ДА
  - 5 Проект перезапускается.
- 5а В Программ мемори остаются все те же три строки нулей(

---

★ **DI HALT**

15 Июль 2010 в 1:40

Ааа ну есть такой прикоп. Меньше не больше. А напиши туда какую нибудь другую команду — она отобразится. Это, кстати, прикоп авр студии. Она не стирает память перед запуском, оставляя там предыдущие значения. Особенно это вставляет в оперативке (RAM). КОгда будешь работать с ОЗУ учитывай этот факт. И стирай память ВСЮ в самом начале работы, специальным циклом инициализации памяти (тупо записывая туда в цикле нули). Ну там дальше будет пример.

---

**Anhanger**

15 Июль 2010 в 1:45

Большое Спасибо, DI HALT!

---

★ **DI HALT**

15 Июль 2010 в 1:42

Можешь сохранить проект, закрыть студию, открыть проект заново. И запустить. Будет два нопа.

---

**koyot**

25 Август 2010 в 10:40

А где можно почитать про этот “Домашний терминатор”? а то в коментах в основном про него, а в статье ни слова)

---

★ **DI HALT**

25 Август 2010 в 11:54

Раздел Робототехника на сайте

---

**koyot**

2 Сентябрь 2010 в 10:33

можно ли как то сделать что бы в AVR Studio содержимое регистров, памяти и портов отображалось в двоичном виде? и где посмотреть содержимое стека?

---

★ **DI HALT**

2 Сентябрь 2010 в 10:42

Значения портов видно в двоичном виде в окне IOView

Регистры в двоичном виде можно увидеть если открыть спец окно Registry (Alt+0) и там в контекстном меню настроить.

Память вроде бы нельзя в двоичном показать.

Стек наблюдается в конце памяти.

---

**Mercury**

20 Октябрь 2010 в 18:02

а как сделать так, чтобы в окне кода проекта отображались блоки как на скринах выше?? а то у меня там все пусто((

---

★ **DI HALT**

20 Октябрь 2010 в 18:09

Какие блоки кода?

---

**Mercury**

20 Октябрь 2010 в 18:14

cseg

eseg

сегменты правильнее..

---

★ **DI HALT**

20 Октябрь 2010 в 18:24

Ручками вписываются. Какие проблемы то?

---

**I-kaktus**

31 Декабрь 2010 в 5:08

DI, у меня возник вопрос:

В примере про использование команды LDI, если поменять регистр с Z (30,31) на X(26,27) или Y(28,29), то при выполнении команды JMP переход идет к 0x000000

---

**I-kaktus**

31 Декабрь 2010 в 5:11

Рано отправил, забыл в начале второй строчки написать «Почему», а в конце поставить знак вопроса))))

---

★ **DI HALT**

31 Декабрь 2010 в 6:11

Ну так погляди в даташите описание команды JMP и поймешь почему. Она работает только с Z парой.

---

**I-kaktus**

31 Декабрь 2010 в 15:33

Понял, спс. Я только разбираться начал во всем этом, поэтому хочется чтобы недопоняток не оставалось.

---

**Zubilo**

29 Март 2011 в 22:34

поправьте если ошибаюсь, но инструкция

.ORG 0x0010

должна нас привести на позицию с адресом 16

ведь 0x0010 — это хекс

---

**★ DI HALT**

30 Март 2011 в 1:54

А я где то утверждал иначе?

---

**Zubilo**

3 Апрель 2011 в 23:43

прошу прощения, видимо меня переклинило =)

---

**Antinomy**

21 Февраль 2012 в 9:06

Да, ошибка в примере в материале «Давай подвинем нашу кодовую конструкцию в памяти на десяток байт.» в то время как речь идёт о 10h байт. Исправьте, пожалуйста, на 16. Также в ранней статье не исправили сообщённую вам другим посетителем ошибку, я позволю себе то исправление поднять.

---

### **Kostik28**

11 Апрель 2011 в 19:36

хотел спросить в разделе AVR. Учебный курс. Подпрограммы и прерывания, но чёт там не получается... глюк наверн вот мой быдлокод. почему во время счета не срабатывает прерывание? ссылка на листинг <http://easyelectronics.ru/repository.php?act=view&id=47>

---

### **★ DI HALT**

11 Апрель 2011 в 19:47

А сами импульсы считает? Флаги выставляются?

---

### **Kostik28**

11 Апрель 2011 в 19:58

да, это всё работает, старший регистр не проверял (устал тыкать:)), а младший считает

---

### **★ DI HALT**

11 Апрель 2011 в 20:22

А зачем много тыкать? Вручную в счетном регистре выставил 254 и два раза протыкнул — вот тебе и переход в старший разряд. Поставил за пару раз до переполнения/сравнения — вот тебе и прерывающее событие. Заодно увидишь что происходит

---

**Kostik28**

11 Апрель 2011 в 20:46

до 256 я дошёл, но прерывания нет

---

**Kostik28**

11 Апрель 2011 в 23:17

прерывания работают, спасибо:)

но происходит затык в этом месте, в реале контроллер здесь встанет?

Wait\_uart\_00: ; передача по UART (во время прерывания по совпадению)

sbis ucscr,udre

rjmp Wait\_uart\_00

out udr,temp

Wait\_uart\_10:

sbis ucscr,udre

rjmp Wait\_uart\_10

out udr,XL

Wait\_uart\_20:

sbis ucscr,udre

rjmp Wait\_uart\_20

out udr,XH

---

**444c-c4**

11 Апрель 2011 в 21:44

Уважаемые, может для кого-нибудь ситуация прозрачна? Tn2313, Asm, AVRStud. MCUCR — 0,

GIMSK — 0xC0 (прерывание вызывается все время, пока 0 на ПД2 или ПД3), на соответствующих векторах прерываний есть команды

перехода для выполн. прерываний.

Подпрограмма прерывания заканчивается: 1-когда уровень на ПД2 становится 1; 2-когда подпрограмма прерывания получает команду reti? Если подпрограмма прерывания по reti

возвращается к выполнению основн. проги, при этом SREG7 — 1, почему нулевой уровень на ПД2, не вызывает вновь выполнения прерывания?

Может кто подскажет как здесь делать гиперссылку? Всего наилучшего.

---

**bdp**

6 Июнь 2011 в 1:47

Решил повторно перечитать Ваш курс (тем более он обновился).

Ну чтож — как всегда супер, так держать!!!

А теперь ложка дегтя :)

Директива .ORG указывает компилятору начало следующего куска кода. По вашему примеру программа сначала делает NOP а потом прыгает через 10 слов(через таблицу векторов прерываний) на продолжение куска программы. И тут кизяк припрятался :) — компилятор то понял где следующий кусок кода, а контролерное ядро откуда будет знать про это.

Вся проблемность усугубляется полным провтиком симулятора при движении по коду, но вот если открыть окно дизасемблера и потыкать F11, там все корректно указывает на подкрившийся кизяк.

К слову — в старом курсе Ви правильно указывали в поле адресе RJMP main, где main метка адреса после таблицы векторов прерываний.

---

**★ DI HALT**

6 Июнь 2011 в 1:54

***>По вашему примеру программа сначала делает NOP а потом прыгает через 10 слов(через таблицу векторов прерываний) на продолжение куска программы.***

Это вы сами уже домыслили. Я такого не говорил. Цель была показать как ORG гоняет код по памяти. в дальнейших примерах везде и RJMP и так далее. Кроме того число 10 взято просто от балды, т.к. таблица векторов она поболее будет.

---

### **bdp**

6 Июнь 2011 в 2:13

у про число 10 я понял :)

А за домыслы извини, меня на это сподвигнула фраза:

>А еще таким образом можно прятать в коде данные, перемешивая данные с кодом.

Если честно, реверсный инженеринг не сильно збивает с толку простое запихивание данных между частями кода, ни и тому подобное.

Я видел как ребята в поисках дыр (не преступники, а нормальная кантора которую нанимают для поиска дыр) используют инструменты которые лехко разгребают такой хлам.

---

### **Alex0720**

14 Август 2011 в 14:32

Доброго времени суток.

Цитата: «Если заглянешь в память, то увидишь там такую картину:.... ОС 94 — это код нашей команды, а 01 00 адрес перехода.»

А теперь вопрос: Почему на код:

NOP

RJMP 0x000002

RJMP 0x000000

в окне Memory/Disassembler видим???

11: NOP

+00000000: 0000 NOP No operation

12: RJMP 0x000002



+00000001: C000 RJMP PC+0x0001 Relative jump  
13: RJMP 0x000000  
+00000002: CFFD RJMP PC-0x0002 Relative jump

Почему коды команд RJMP разные??? Как я понимаю коды команд это что то из разряда забитого гвоздиком?

---

### ★ DI HALT

14 Август 2011 в 16:31

Так и есть. Тут команды двубайтные и операнд вписан в код команды. Т.е. касаемо RJMP то код там такой 1100xxxxxxxxxx где xxxxxx это смещение перехода. Т.е. он вписан в код команды или, другими словами, у нас есть 4096 команды группы RJMP каждая со своим фиксированным смещением перехода.

---

### Alex0720

14 Август 2011 в 18:17

Спасибо за ответ. С этим понятно. Кажется. =) Но попробую потупить еще раз.. =)

Рассмотрим подробно строку 12 Disassembler-a:

Даем команду программе сделать переход на строку кода 0x000002(RJMP 0x000002). И при прогоне он таки туда и переходит. Но...

Посчитаем переход для RJMP:  $PC + 0 \times 0001 = 00000001 + 0001 = 00000010$ . С учетом кода команды — 1100\_0000\_0000\_0010(C002h).

А AVR утверждает что — C000h.

Со вторым RJMP такая же засада.. =(

---

### ★ DI HALT

14 Август 2011 в 18:50

Ну там не буквально xxx, а  $xxxx+1$  вот и получается, что для PC+1 смещение будет указано в коде как 0000

---

**Gena**

26 Август 2011 в 15:59

Вопрос про RJMP.

Цитата: “Если заглянешь в память, то увидишь там такую картину:.... ОС 94 — это код нашей команды, а 01 00 адрес перехода.”

Почему именно «адрес перехода» если это команда относительного перехода и в команде должно указываться смещение перехода.

Или смещение высчитывается в момент компиляции и хранится в команде как адрес перехода?

Изменил пример:

```
.include «tn2313def.inc» ; Используем tiny2313
```

```
.CSEG ; Кодовый сегмент
```

```
M1: NOP
```

```
NOP
```

```
LDI ZL,low(M2) ; Загрузили в индекс
```

```
LDI ZH,High(M2)
```

```
JMP
```

```
NOP
```

```
NOP
```

```
NOP
```

```
M2: NOP
```

```
RJMP 3
```

```
NOP
```

```
NOP
```

```
NOP
```

```
NOP
```

```
NOP
```

```
JMP M1
```

При компилировании почему-то говорит что команды JMP не бывает. При трассировке переход RJMP идет на M1, а не на три команды вперед. Отрицательные значения смещения не принимает. Я знаю, что на практике указывают метки а не числа.

---

### **Gena**

26 Август 2011 в 16:13

Ха. Переходит вообще на третью команду, будто это не смещение, а абсолютный адрес.

---

### **X-Ray**

16 Сентябрь 2011 в 21:57

В AVR Studio 5, при нажатии Alt+O не появляется окно настройки симуляции и негде выставить частоту. Можно вручную вписать частоту в окошке Processor. Влияет ли выставленная частота на еще что то кроме расчета времени обработки операций?

---

### **★ DI HALT**

16 Сентябрь 2011 в 22:04

Только на время обсчета

---

### **X-Ray**

16 Сентябрь 2011 в 23:08

Про то что работает во время просчета я понял. Но у меня это сочетание не работает ни как, может они упростили это и теперь можно просто записать в графе Frequency?

---

### **X-Ray**

16 Сентябрь 2011 в 23:08

Сори, не понял ответ))

---

### ★ DI HALT

17 Сентябрь 2011 в 11:51

Возможно. Поставь да проверь, там же видно время stop watch

---

### X-Ray

16 Сентябрь 2011 в 22:47

И еще в AVR Studio 5, адреса указываются не словами а байтами. Поэтому где мы смещаем код на 0x0010, в памяти он отображается на 0x0020.

Смотрим скриншот <https://picasaweb.google.com/lh/photo/BUQd-77AWuGRNSURndHPXA?feat=directlink> если плохо видно, слева вверху около фотографии есть кнопка сохранить. Там полное разрешение

---

### Anton

4 Ноябрь 2011 в 18:56

DI HALT, у вас на 4 рисунке в окне кода включена нумерация строк. Подскажите как это сделать, в studio 4.19 не могу найти

---

### ★ DI HALT

4 Ноябрь 2011 в 19:05

Хоспади, да с чего вы все берете, что это результат студии. Это же не скриншот. В студии нельзя включить нумерацию. Та что у меня на сайте, так она генерируется автоматически, плагином к движку сайта и к AVR Studio отношения не имеет совершенно.

---

### number\_nine

9 Декабрь 2011 в 21:37

Приветствую!

Вопрос по директиве .ORG. При пошаговой трассировке, после прохождения директивы .ORG Program Counter становится равным 0x000010, а Cycle Counter — 16. Контроллер «прощелкивает» адреса или загружает откуда-то значение программного счетчика? Если «прощелкивает» — почему не выполняется код таблицы прерываний? Если загружает — то откуда он берет значение, заданное директивой .ORG? В окне 메모гу я ничего не обнаружил... Прощу прощения за корявый вопрос, заранее спасибо!

---

### ★ DI HALT

9 Декабрь 2011 в 22:22

ORG существует ТОЛЬКО для компилятора и программиста, который указывает компилятору, что код надо разместить здесь.

Контроллер прощелкивает адреса строго по очереди, либо сразу заносит нужный адрес если ему попадается команда перехода. Таблица векторов не выполняется, т.к. в нулевом векторе обычно пишут RJMP main. Ну а саму майн посредством ORG смещают за таблицу векторов.

---

### number\_nine

9 Декабрь 2011 в 22:32

Всё понял, упустил из виду gjmp main. В общем, я так и думал, смутило то, что значение программного счетчика поменялось сразу, хотя счетчик циклов должен был насторожить :) Спасибо за ответ, думаю, вопрос далеко не последний!

---

### Vobar

14 Январь 2012 в 2:40

Привет всем!

Начал делать диплом ( в техникуме ). Вначале нужно реализовать ИК-связь между пультом и кондиционером. Я для начала попытался просто создать меандр с максимальной частотой. Программа получилась такая: контроллер (ATtiny13a)

```
sbi 0x16, 0 ;установка режима работы потра В/В
```

```
sbi 0x17, 0
```

```
por
```

```
por  
cbi 0x17, 0  
rjmp pc-4
```

В AVR Studio на операцию генерации уходит 0.83 микросекунды, а на осциллограмме длина периода — 7 микросекунд при частоте 9.6 МГц от внутреннего генератора. Осциллограф — С1-93, перед измерением калибровался. Я практически на 100% уверен, что частота для тактировки выбрана такая, т.к. после перезаписи фьюзов из (1:0) в (0:1) импульсы стали в 2 раза короче. Откуда такое расхождение?

---

### **akocur**

17 Январь 2012 в 19:56

Я пользуюсь программой AVR STUDIO 5.0.1223 Вопрос в следующем: когда запускаю симуляцию, то в окне MEMORY адрес не по порядку идет, а через один. Это в том случае когда выставляю в две колонки <http://narod.ru/disk/37987044001/AVR.jpg.html> Когда же выставляю в одну колонку то получается вот так <http://narod.ru/disk/37987495001/AVR1.jpg.html> Хотя в примерах выше адрес в окне MEMORY по другому. Может я что то не так настроил?

---

### **★ DI HALT**

17 Январь 2012 в 20:05

Я юзаю авр студию 4.19. Пятая версия — сырое, глючное, неприспособленное к жизни говно. Оно еще и старые контроллеры не поддерживает в полной мере. Не говоря уж про старые отладочные средства.

---

### **sea**

22 Январь 2012 в 14:01

Уважаемый DI\_HALT ! Микроконтроллер ATmega128. Объем программы: Code=2260, Data=2092, Used=4352. Всё компилируется нормально.

Но, как стоит мне добавить любую одну инструкцию на место «Здесь вся программа», даже в самом начале где от rcall до метки print не изменяется разрыв до 2Кб.

Сразу компиляция не проходит = Error: Operand(s) out of range in LDI R16,0x100' ?

Я где-то понимаю что это связано с 2Кб страницей, но Z может адресовать 64Кб напрямую, тем более метка metfont находится рядом !

В чём причина ошибки компиляции, дальнейшее увеличение программы вводит в ступор ?

```
.DEF rga=r16
```

```
...
```

Здесь вся программа.

```
...
```

```
print: ld rga,y+ ;печать символа из Y, по адресу X
```

```
printat: ldi rgb,8 ;печать символа в rga, по адресу X
```

```
mul rga,rgb
```

```
movw zl,r0
```

```
ldi rga,low(metfont)*2
```

```
add zl,rga
```

```
ldi rga,high(metfont)*2
```

```
adc zh,rga
```

```
fontm01: lpm rga,z+
```

```
fontm04: st x+,rga
```

```
dec rgb
```

```
breq fontm03
```

```
cpi rga,0
```

```
brne fontm01
```

```
fontm02: lpm rga,z+
```

```
cpi rga,0
```

```
brne fontm04
```

```
fontm03: ret
```

```
metfont: .include «font_4x8.asm»
```

---

### ★ DI HALT

22 Январь 2012 в 22:25

Чет не догоняю. А какой радостит о он ругается на LDI??? А выше или ниже LDI что?

---

### sea

22 Февраль 2012 в 20:04

Я разобрался, ошибка: вместо `ldi rga,high(metfont)*2` надо писать: `ldi rga,high(metfont*2*)` !!! Но есть такой вопрос: в ATtiny2313 пробовал использовать 4-разрядный счётчик USI, как счётчик делитель для прерываний, подав на PORTB7 импульсы с определённой частотой (так как оба TC0 и TC1 заняты под ШИМ) ? В AVRStudio не работает, но в pdf написано, что можно !

---

### Aleksandr88

10 Май 2012 в 19:34

Подскажите как в си сделать прыжок так как в Ассемблере «RJMP PC+2» если писать так `asm(«RJMP PC+2»);` студия ругается O\_o.

---

### ★ DI HALT

10 Май 2012 в 19:41

Напрямую никак. Только ассемблерной вставкой. Но зачем?

---

### Aleksandr88

10 Май 2012 в 19:54

Ну я вот и хочу ассемблерную вставку сделать в студию `asm(«RJMP PC-2»);` но она ругается на «PC» есть возможность этого добиться ?



---

★ **DI HALT**

10 Май 2012 в 19:56

Не, я понял про ассемблерную вставку. Но во первых, она делается не так. Там свой формат записи. Во вторых, не факт, что он знает определение PC. А в третьих, зачем вам именно эта вставка? Наверняка можно решить без нее.

---

**Aleksandr88**

10 Май 2012 в 20:13

Ну суть дела в том. Пару дней назад решил изучать возможность организации режима многозадачности. Прочила часть ваших статей на этом сайте про (конвертерную обработку данных) и подумал добавить туда не много многозадачности используя Program counter для переключения между процессами. Конечно есть готовые навороченные OS но от той мысли сколько кода надо изучить чтобы полностью с ними разобраться становится плохо=) Как посоветуете переходить между процессами ?

---

★ **DI HALT**

10 Май 2012 в 21:05

Ну точно не так. Во первых на Си для этого есть переход по указателю на функцию. В статьях где тот же диспетчер описан на сях используется именно он. Во вторых тут в полный рост встает вопрос сохранения контекста. Потому как тупо спрыгнув в другую задачу ты теряешь стек и локальные переменные.

---

**Aleksandr88**

11 Май 2012 в 15:45

1. В диспетчере как я понял (может не до конца) обработка выполняется последовательно только после завершения первого процесса осуществляется переход на второй?
  2. А как допустим организовывается функция Delay(); не используя цикл, то есть чтоб проц просто в холостую не крутился в цикле а во время задержки переходил в (2) процесс и по истечению времени задержки от таймера переходил в процесс (1)?
-

## ★ DI HALT

11 Май 2012 в 17:23

1) Да

2) У меня никак. Я обычно разбиваю задачу на две. До задержки и после задержки. И делаю вызов второй половины через службу таймера. Можно сделать и на одной, через конечный автомат и локальные static переменные.

---

## ataman365

6 Октябрь 2012 в 20:00

Подскажите пожалуйста, как выставить частоту в AtmelStudio 6

---

## Spaun

20 Февраль 2013 в 11:35

Просто тыкаете курсор в частоту(в цифры) и меняете.

---

## alexeytulavifov

8 Декабрь 2012 в 0:02

Ребята, никак не пойму вот что. А что, во флеш-памяти команда задом наперед хранится? Т е младший байт спереди, а старший — сзади? В hex файлах старший и младший байты местами переставлены. Это зачем так? Расшифровывать их так что ли проще? А почему бы тогда и в справочниках (типа Евстифеева) не указывать коды операций сразу задом наперед, чтобы люди с дизассемблированием не мучались?

---

## ★ DI HALT

8 Декабрь 2012 в 0:29

Младший байт по младшему адресу. На самом деле так логичней. т.к. выборка адресации проще. Особенно при многобайтной математике, т.к. при выборке сначала оперируем младшей парой байт, потом просто инкрементируем адрес и берем старшую пару байт, добавляем к ней переносы и так до конца. И не важно сколько байт впереди. Есть архитектуры где наоборот, но это редкость.

А реальный код читается в обратную сторону, сначала старший байт, потом младший, так человеку привычней. От того и поворачивается.

---

**alexeytulavifov**

9 Декабрь 2012 в 6:14

Спасибще, Дихалт, растолковал. Выходит это все условности, так же как + и — в электротехнике...

---

**else-2**

14 Январь 2013 в 1:13

Купил на sparkfun.com платку на atmega32u4, программирующуюся по ЮСБ.

Avr studio 6 поставлена поверх winavr.

Вбиваю пример с тремя пор.

При запуске симуляции, студия пишет, что она не поддерживает этот контроллер.

В интернетах нашел только вот это

<http://www.avrfreaks.net/index.php?name=PNphpBB2&file=viewtopic&p=967773>

где говорят, что в ближайшее время поддержки не будет.

Хочу начать писать на ассемблере, и мне важно программирование по usb, так как пайка программатора мне кажется слишком трудоемкой.

Краем глаза на том же форуме видел, что кто-то для этого контроллера советовал simulator-2 для 4й версии студии. Подскажите, кто знает, поможет-ли?

Спокойно-ли можно снести более позднюю версию студии и поставить более раннюю, или будут проблемы?

Что в итоге делать?

---

## ★ DI HALT

14 Январь 2013 в 1:18

Студия начиная с версии 4.19 не поддерживает winavr для си там теперь есть avrtoolchain то же самое, только вид сбоку. Но отличия некоторые все же есть изза чего ряд программ под винавр не компилируется.

Вообще все студии нового типа (от 5 и выше) УГ, т.к. атмел тупо забил на поддержку старых контроллеров и средств отладки. Сноси этот шлак и ставь студию 4.

---

## else-2

17 Январь 2013 в 0:30

Тогда какие же у них новые контроллеры, на что они вообще нацелены теперь? Понятно, что они не прекратят производство старых, ну или мне так хочется думать. С чем связано такое кардинальное прекращение поддержки «старого» модельного ряда?

---

## Spaun

20 Февраль 2013 в 11:20

День добрый.

Первое — спасибо за труд, всё очень доходчиво написано.

Теперь вопрос по статье выше: «Давай подвинем нашу кодовую конструкцию в памяти на десяток байт .ORG 0x0010» — сдвиг на 16 шагов и потом мы окажемся на 20 байте (адреса ведь в словах?)

Просто только начал изучение, поставил Atmel Studio 6 — а там 1 адрес соответствует 1 байту (маленький затык у меня случился=)

---

## broneko

5 Июль 2013 в 20:56

вот читаю я это все, читаю, читаю.. потом перечитываю, перечитываю.. такая тоска накатывает, что не смогу дожить до того момента, когда смогу работать с МК, а мне 20 лет всего. это нормально? :)

я так понимаю, лучше немного изучить основы Ассемблера, а потом читать курс по AVR?

---

**Boris86**

12 Декабрь 2013 в 13:32

Тоска это когда тебе 27, сидишь на попе ровно без денег, учишь эту тему и понимаешь, что до результатов еще далеко. А кушать хочется сейчас. И сознание тебя уничтожает.

20 — это нормально, более чем нормально

---

**BIYur**

19 Июль 2013 в 17:54

Помогите разобраться, почему не хочет работать моя простейшая программа, которая зажигает светодиод при нажатой кнопке и гасит при отпущенной:

(PinBoadr 1.1)

```
ldi r17, 0b11111111
```

```
out ddrb, r17 ; Порт В на выход
```

```
ldi r16, 0b00000000
```

```
out ddrc, r16 ; Порт С на вход
```

```
out PortC, r17 ; Подтяжка порта С
```

```
Main:
```

```
sbic pinc, 4 ; Проверим состояние входа кнопки PC4
```

```
rjmp LEDOff ; Переход, если не нажата (на PC4 высокий уровень)
```

```
out PortB, r17 ; Включим светодиод
```

```
RJMP Main
```

```
LEDOff:
```

```
out PortB, r16 ; Выключим светодиод
```

```
rjmp Main
```

---

## ★ DI HALT

19 Июль 2013 в 20:37

Классические вилы. Порт C по дефолту работает на жтаг и потому те выводы его , что служат жтагу (TDI TDO ЕСЛ TMS) не могут использоваться как порт. Жтаг можно отключить либо фуз битами. Либо поправив регистр MCUCSR поставив там бит JTD

---

## VeniaminCaver

8 Ноябрь 2013 в 2:48

Окей, вроде понял — хочу опять попросить проверить то, что я понял=)

Итак — ну на счёт элементарного — вроде все понятно — единственно : на меге 8 типа не работает команда JMP, а вместо неё работает RJMP и её можно юзать и как JMP и как RJMP. То есть теперь нет JMP а есть «RJMP метка» «RJMP адрес» «RJMP смещение».

По поводу старших регистров и меток:

Скажем ситуация — нам нужно ждать какого-то события, а потом, при наступлении этого события мы идём молотить программу дальше. Что мы делаем — ставим метку M1 на команду 0002. Код её расположения среди команд процессора есть последовательность нолей и единиц на 2 байта — то есть «00010011» «00000110» — к примеру, означает, что это адрес команды процессора 0002. Ниже мы юзаем LDI — это такая хрень, которая задаёт регистру(у которого есть 8 байт — то есть грубо 8 ячеек, в которых может быть 0 или 1) значение в виде 8 бит. ZI -это старший регистр ZH тоже. Они могут образовывать регистровую пару — как раз для таких случаев. Так вот мы через LDI прописываем этим двум регистрам 2 части нашего адреса метки — то есть по одиночке это ничего — а вместе дают адрес инструкции процессора. Соответственно мы ставим ниже переход IJMP который хватает из регистров по байту — ставит их рядом — понимает что это адрес метки M1 и бежит туда — затем всё повторяется — и пока это всё повторяется — мы как бы ждём события. Затем мы внезапно получаем сигнал от датчика. Меняем биты в регистре(было бы круто, если бы биты на этом регистре можно было сразу извне запитать в нужной последовательности — которая бы дала 2 байта нового адреса) и 2 новых байта означают уже адрес метки M2 — соответственно прога прыгает на M2 и там уже дальше пошло-поехало. Можно так сделать несколько циклов по ожидании каких то данных с датчиков или ещё откуда.

---

## Ana-bio-z

9 Февраль 2014 в 20:23

Крепкого тебе здоровья, уважаемый Di Halt! По поводу директивы .ORG было написано следующее — «Точка входа должна быть, иначе симулятор скукожит от такого когнитивного диссонанса. А вот дальше сплошные FF FF и лишь начиная с нашего адреса 0×0010 пошли коды остальных команд». Я хоть и верю на слово, но любопытства ради все проверяю. Сделал так:

```
.include «m16def.inc» ; Используем ATmega16
;= Start macro.inc =====

; Тут будут наши макросы, потом.

;= End macro.inc =====

; RAM =====
.DSEG ; Сегмент ОЗУ

; FLASH =====
.CSEG
.ORG 0x000010
M1: NOP
NOP
NOP
LDI ZL,low(M2)
LDI ZH,high(M2)
IJMP
NOP
M2: NOP
RJMP M1
NOP
```

```
;EEPROM =====  
.ESEG ; Сегмент EEPROM
```

И ничего, даже без NOP по адресу 0x00 симулируется без проблем. В во флеше с 0x00 по 0x0F отображаются значения FF, а «точка входа» — (первый NOP — 00 00) располагается по адресу — 0x10. Можешь прокомментировать почему?

И еще пара вопрос полного новичка...

1. Обозначения 0x00 и 00h это одно и тоже? И что тогда — \$00?
2. Можно ли как-то сбрасывать memory program перед перекомпиляцией проекта после удаления части текста программы? А то каждый раз сохранять проект и перезапускать студию чтобы чтобы изменения попали в память как-то не удобно.

Спасибо за прекрасный сайт!

---

### ★ DI HALT

9 Февраль 2014 в 20:41

По дефолту org ноль, но у тебя же явно адрес 10 прописан в коде, так чего удивляешься? Стимулятор на ff варнинги дает. По крайней мере давал в старых версиях.

1. Это одно и то же. Разные написания просто.
2. Написать программу инициализации памяти. Полезно и в железе, приколов меньше будет при отладке.

---

### Ana-bio-z

9 Февраль 2014 в 21:41

AVR Studio: Version 4.19 build 730 — ни варнингов, ни ероров :)

1. Попробовал такое: 0x10 и такое: \$10 компилятор принимает, а на такое: 10h или 0x10h ругается.
2. В ответ на комменты к этой статье ты уже писал про специальный цикл инициализации памяти пример которого будет в дальнейшем. Я думал это касается оперативки. Пока до туда не дочитал, поэтому интересно для флеша он тоже подойдет?



---

## ★ DI HALT

10 Февраль 2014 в 0:06

А зачем стирать флеш? При компиляции он сам обновляется.

---

## Ana-bio-z

10 Февраль 2014 в 3:32

Так вот то-то и оно, что не обновляется! То есть обновляется в том случае, если какие либо строки были заменены на другие. А если после компиляции просто удалить одну или несколько строк в редакторе кода и перекомпилировать проект, то в окне memory (program) все остается по прежнему, как до удаления и перекомпиляции. Однако, при этом симуляция выполнения программы происходит уже с внесенными изменениями. Об этом выше уже писал уважаемый Anhanger 15 Июля 2010 в 0:15 и далее.

Было 3 NOP, удаляем один, перекомпилируем, запускаем симуляцию заново. В окне редактора кода остается 2 NOPа, а в окне флеша, как было три с прошлого раза, так и осталось. Тем не менее, симулятор щелкает два NOPа подряд и дальше пробегает по FFкам до конца флеша. Вопрос, можно ли перед перекомпиляцией принудительно очищать флеш в студии чтоб с толку не сбивала?

---

## ★ DI HALT

10 Февраль 2014 в 3:36

F12 вроде бы. Build clean должно помочь. Оно грохает все промежуточные файлы.

---

## Valina

19 Март 2014 в 10:43

Привет! Пробую писать как у вас в уроке, только на atmel studio. У меня проблемка, не пойму почему команде пор приписывается в памяти программ не нули. Вот картинка <http://yadi.sk/d/yovtZaCEKndcC>

---

## ★ DI HALT

19 Март 2014 в 10:51

Пользуйся 4й студией и будет тебе счастье. бя вся кривая и глючная.

---

**Valina**

19 Март 2014 в 10:55

спасибо! Ссыль не кинешь? Спасибо за курс )))

---

★ **DI HALT**

19 Март 2014 в 10:59

<https://dl.dropboxusercontent.com/u/12226548/Files/AvrStudio4.17%20Setup.exe>

---

**Valina**

19 Март 2014 в 11:01

Огромное Спасибо! )))

---

**Valina**

19 Март 2014 в 15:08

чёт вначале буксую ((( первый пример с метками запускаю, он выдаёт ошибку и больше не запускается отладчик, даже если прогу меняешь. Не подскажешь в чём беда?

---

★ **DI HALT**

19 Март 2014 в 15:21

Какую ошибку?

---

## Valina

19 Март 2014 в 15:33

C:\avr\Mega16\DiHalt\Avr1prog.asm(5): error: Invalid register

---

## ★ DI HALT

19 Март 2014 в 15:48

inc файл видимо не подключился. Там все эти символические имена типа ZL прописаны. Может пути не верно идут. Или русские символы есть в пути установки студии.

---

## Valina

19 Март 2014 в 15:34

C:\avr\Mega16\DiHalt\Avr1prog.asm(5): error: Undefined symbol: ZH

---

## Valina

19 Март 2014 в 17:49

AVRASM: AVR macro assembler 2.1.41 (build 1792 Jul 21 2009 12:30:27)

Copyright (C) 1995-2009 ATMEL Corporation

C:\avr\Prog1\Prog1.asm(4): error: Undefined symbol: ZL

C:\avr\Prog1\Prog1.asm(4): error: Invalid register

C:\avr\Prog1\Prog1.asm(5): error: Undefined symbol: ZH

C:\avr\Prog1\Prog1.asm(5): error: Invalid register

C:\avr\Prog1\Prog1.asm(15): No EEPROM data, deleting C:\avr\Prog1\Prog1.eep

где копать? переустановил студию на всякий случай, поменял путь, названия написал тока английские, ошибка та же. Я правильно понимаю, что студия не может найти файл, которые описывает регистры? Может мне надо переключиться чуть, чтобы потом увидеть очевидное? Что подскажешь, или лучше мне пойти передохнуть до завтра?

---

★ **DI HALT**

19 Март 2014 в 18:22

.include «m16def.inc» ; Используем ATmega16

вот эта строка или подобная есть в самом начале?

---

**Valina**

20 Март 2014 в 9:40

Спасибо за подсказки ))) С утра голова у меня лучше варит ))) Я ещё раз переустановил студию, всё удалил, теперь всё работает! Ура!

---

**Valina**

20 Март 2014 в 10:41

не понимает символы((( <http://yadi.sk/d/yovtZaCEKndcC>

---

★ **DI HALT**

20 Март 2014 в 10:46

не подцепились inc файлы. Они там прописаны. Можешь не использовать символические константы, а использовать вместо них числа. Вручную указывая адреса конца памяти.

---

**Valina**

20 Март 2014 в 10:51

а как подцепить inc файлы? Вручную указывал адреса в прошлом уроке, было всё ок.

---

**Valina**

20 Март 2014 в 11:49

Подцепил, написал вначале проги `.include «m16def.inc»`

---

★ **DI HALT**

20 Март 2014 в 12:00

А сами файлы должны быть в студии и в свойствах проекта должны быть прописаны пути туда. Вообще по умолчанию это обычно так. Если нет, то можешь инклюд сделать через полный путь. Ну и сама студия не ругается что файла нет?

---

**Valina**

20 Март 2014 в 12:42

они у меня во включенных файлах, только не пойму почему сразу привязки нет, ведь как я понимаю выбираем мк перед созданием проекта именно для этих целей.

---

**kia**

27 Март 2014 в 1:17

Не могли бы поделиться списком системы команд для работы, или же ссылкой на них.

---

★ **DI HALT**

27 Март 2014 в 3:23

Он есть в даташите в конце самом. Раздел `instructions set`

---

**kia**

27 Март 2014 в 23:55

Спасибо! )))

---

### **Aid1992**

25 Ноябрь 2014 в 21:38

DI HALT а не могли бы Вы выложить сюда список команд Ассемблера, о которых Вы пишете в этом уроке. Было бы полезно для изучения, распечатал и все перед глазами. Спасибо.

---

### **★ DI HALT**

26 Ноябрь 2014 в 4:35

А зачем? Все команды с описанием есть в каждом даташите. Там прям таблица всех команд.

---

### **Aid1992**

26 Ноябрь 2014 в 8:02

Да? Почему-то никогда не обращал внимания. Надо будет основательнее покопаться в даташите 16-ой АтМеги :) Спасибо.

---

### **Hottabitch**

6 Сентябрь 2015 в 2:55

Привет DI. Вопрос что не так с адресами на картинке? Студия 6.2.

<http://itimages.ru/image/view/2965595/c1f1e840>

---

### **★ DI HALT**

6 Сентябрь 2015 в 2:58

Все так. Просто студия 4 умела показывать либо в словах, либо в байтах (у меня в словах, переключает кнопочка 8/16, что выше ), а студи 6 видимо только в байтах.

---

### **Hottabitch**

6 Сентябрь 2015 в 17:00

Тогда другой вопрос, программа должна начать записываться в ячейки памяти с 0x0010 после данной команды .ORG 0x0010

но как видно из скрина выше, код записывается с 0x0020 ячейки. Чего я недопонимаю?

---

### **Hottabitch**

6 Сентябрь 2015 в 17:18

Вроде разобрался. Как я понял в разделах памяти .DSEG и .ESEG команда .ORG считает по байтам, а в .CSEG по словам. Так?

---

### **★ DI HALT**

8 Сентябрь 2015 в 0:40

Да. Если где то разница в адресах ровно в два раза, то это просто указан адрес в словах или в байтах.

---

### **zhituha**

31 Октябрь 2015 в 23:38

Я правильно понимаю, что конструкция RJMP 0x000001 сделает ровно то же самое, что и JMP, но чуть дешевле? Или симулятор обманывает меня?

---

### **★ DI HALT**

1 Ноябрь 2015 в 4:40

Дешевле, да. Но вот незадача RJMP умеет ходить не дальше чем на 127 команд вверх или вниз, потом у него штаны рвутся. А JMP в пределах всей памяти. Когда пишешь большие программы, то напарываешься на это ограничение.

---

### **zhituha**

1 Ноябрь 2015 в 15:11

Данке. И вот не очень понимаю поведение симулятора:

.CSEG ; Кодовый сегмент

NOP

.ORG 0x0010

NOP

В памяти вижу, что код сместился чуть дальше:

prog 0x0000 00 00 ..

prog 0x0002 ff ff яя

....

prog 0x001C ff ff яя

prog 0x001E ff ff яя

prog 0x0020 00 00 ..

Это почему так?

---

### **★ DI HALT**

1 Ноябрь 2015 в 23:26

Ну так все верно. У тебя в коде все адреса всегда в словах. А в памяти включено отображение в байтах. Он сместился не чуть дальше, а ровно в два раза дальше.



