

Яндекс.Директ



Грузовой лифт в Краснодаре

Цена от 60 000руб произ-
водство под ваш размер +
монтаж! Замер бесплатно

Подъемник в шахту

Цепной подъемник

Мини подъемник Замер

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

[Адрес и телефон](#)

×

NRF24L01. Datasheet PDF

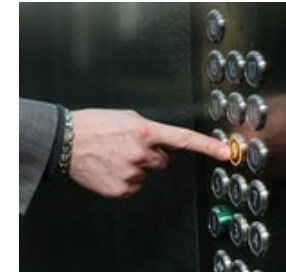
Гарантия. Выгодные цены.
Без посредников. Большой опыт
работы. Заказать: 18+

О компании Гарантия

Доставка Контакты

ru.icpowerooo.cn

×



Лифты под ключ в Краснодаре

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

Модели Сертификаты

Оставить заявку Контакты

esd-lift.ru [Адрес и телефон](#)

×

AVR. Учебный курс. Скелет программы

AVR. Учебный курс | 10 Июль 2008 | DI HALT | 67 Comments

При написании прошивки надо очень внимательно подходить к процессу организации архитектуры будущей программы. Программа должна быть быстрой, не допускать задержек главного цикла и легко расширяться. Оптимально использовать аппаратные ресурсы и стараться выжать максимум возможного из имеющихся ресурсов.

Вообще, архитектура программ это отдельная тема и ближе к концу курса, в его Сильной части я подробнее рассказываю о разных типах организации прошивки. Можешь забежать вперед и поглядеть, что да как.

В ассемблерной же части, я расскажу о одном из самых простых вариантов — флаговом автомате, а позже, когда ты уже будешь вовсю ориентироваться в моем коде, дам пример на основе конвейерного диспетчера, с подробным описанием его работы.

Суперцикл

Все программы на микроконтроллерах обычно зацикленные. Т.е. у нас есть какой то главный цикл, который вращается непрерывно.

Структура же программы при этом следующая:

- **Макросы и макроопределения**
- **Сегмент ОЗУ**
- **Точка входа** — ORG 0000
- **Таблица векторов** — и вектора, ведущие в секцию обработчиков прерываний
- **Обработчики прерываний** — тела обработчиков, возврат отсюда только по RETI
- **Инициализация памяти** — а вот уже отсюда начинается активная часть программы
- **Инициализация стека**
- **Инициализация внутренней периферии** — программирование и запуск в работу всяких таймеров, интерфейсов, выставление портов ввода-вывода в нужные уровни. Разрешение прерываний.
- **Инициализация внешней периферии** — инициализация дисплеев, внешней памяти, разных аппаратных примочек, что подключены к микроконтроллеру извне.
- **Запуск фоновых процессов** — процессы работающие непрерывно, вне зависимости от условий. Такие как сканирование клавиатуры, обновление экрана и так далее.
- **Главный цикл** — тут уже идет вся управляющая логика программы.
- **Сегмент ЕЕПРОМ**

Начинается все с макросов, их пока не много, если что по ходу добавим.

1	.include "m16def.inc" ; Используем ATmega16
2	
3	:= Start macro.inc =====
4	.macro OUTI
5	LDI R16,@1
6	.if @0 < 0x40
7	OUT @0,R16
8	.else
9	STS @0,R16
10	.endif
11	.endm
12	
13	.macro UOUT
14	.if @0 < 0x40
15	OUT @0,@1
16	.else
17	STS @0,@1
18	.endif
19	.endm
20	:= End macro.inc =====

В оперативке пока ничего не размечаем. Нечего.

1	; RAM =====
2	.DSEG
3	; END RAM =====

С точкой входа и таблицей векторов все понятно, следуя нашему давнему шаблону, берем его оттуда:

```
1      ; FLASH =====
2      .CSEG
3      .ORG $000      ; (RESET)
4      RJMP  Reset
5      .ORG $002
6      RETI           ; (INT0) External Interrupt Request 0
7      .ORG $004
8      RETI           ; (INT1) External Interrupt Request 1
9      .ORG $006
10     RETI           ; (TIMER2 COMP) Timer/Counter2 Compare Match
11     .ORG $008
12     RETI           ; (TIMER2 OVF) Timer/Counter2 Overflow
13     .ORG $00A
14     RETI           ; (TIMER1 CAPT) Timer/Counter1 Capture Event
15     .ORG $00C
16     RETI           ; (TIMER1 COMPA) Timer/Counter1 Compare Match A
17     .ORG $00E
18     RETI           ; (TIMER1 COMPB) Timer/Counter1 Compare Match B
19     .ORG $010
20     RETI           ; (TIMER1 OVF) Timer/Counter1 Overflow
21     .ORG $012
22     RETI           ; (TIMER0 OVF) Timer/Counter0 Overflow
23     .ORG $014
24     RETI           ; (SPI,STC) Serial Transfer Complete
```

25	.ORG \$016
26	RETI ; (USART,RXC) USART, Rx Complete
27	.ORG \$018
28	RETI ; (USART,UDRE) USART Data Register Empty
29	.ORG \$01A
30	RETI ; (USART,TXC) USART, Tx Complete
31	.ORG \$01C
32	RETI ; (ADC) ADC Conversion Complete
33	.ORG \$01E
34	RETI ; (EE_RDY) EEPROM Ready
35	.ORG \$020
36	RETI ; (ANA_COMP) Analog Comparator
37	.ORG \$022
38	RETI ; (TWI) 2-wire Serial Interface
39	.ORG \$024
40	RETI ; (INT2) External Interrupt Request 2
41	.ORG \$026
42	RETI ; (TIMER0 COMP) Timer/Counter0 Compare Match
43	.ORG \$028
44	RETI ; (SPM_RDY) Store Program Memory Ready
45	
46	.ORG INT_VECTORS_SIZE ; Конец таблицы прерываний

Обработчики пока тоже пусты, но потом добавим

1	; Interrupts =====
2	; End Interrupts =====

Инициализация ядра. Память, стек, регистры:

1	Reset:	LDI R16,Low(RAMEND)	; Инициализация стека
2		OUT SPL,R16	; Обязательно!!!
3			
4		LDI R16,High(RAMEND)	
5		OUT SPH,R16	
6			
7		; Start coreinit.inc	
8	RAM_Flush:	LDI ZL,Low(SRAM_START)	; Адрес начала ОЗУ в индекс
9		LDI ZH,High(SRAM_START)	
10		CLR R16	; Очищаем R16
11	Flush:	ST Z+,R16	; Сохраняем 0 в ячейку памяти
12		CPI ZH,High(RAMEND)	; Достигли конца оперативки?
13		BRNE Flush	; Нет? Крутимся дальше!
14			
15		CPI ZL,Low(RAMEND)	; А младший байт достиг конца?
16		BRNE Flush	
17			
18		CLR ZL	; Очищаем индекс
19		CLR ZH	
20		CLR R0	
21		CLR R1	
22		CLR R2	
23		CLR R3	
24		CLR R4	

25	CLR	R5
26	CLR	R6
27	CLR	R7
28	CLR	R8
29	CLR	R9
30	CLR	R10
31	CLR	R11
32	CLR	R12
33	CLR	R13
34	CLR	R14
35	CLR	R15
36	CLR	R16
37	CLR	R17
38	CLR	R18
39	CLR	R19
40	CLR	R20
41	CLR	R21
42	CLR	R22
43	CLR	R23
44	CLR	R24
45	CLR	R25
46	CLR	R26
47	CLR	R27
48	CLR	R28
49	CLR	R29
50	; End coreinit.inc	

Всю эту портянку можно и нужно спрятать в inc файл и больше не трогать.

Секции внешней и внутренней инициализации переферии пока пусты, но ненадолго. Равно как и запуск фоновых программ. Потом я просто буду говорить, что мол добавьте эту ботву в секцию Internal Hardware Init и все :)

1	; Internal Hardware Init =====
2	
3	; End Internal Hardware Init =====
4	
5	; External Hardware Init =====
6	
7	; End Internal Hardware Init =====
8	
9	; Run =====
10	
11	; End Run =====

А теперь, собственно, сам главный цикл.

1	; Main =====
2	Main:
3	
4	JMP Main
5	; End Main =====

Все процедуры располагаются в отдельной секции, не смешиваясь с главным циклом. Так удобней, потом можно их по частям вынести в библиотечные файлы и разделить исходник на несколько файлов. Но мы пока это делать не будем. Разделим их просто логически.

1	; Procedure =====
2	
3	; End Procedure =====

[Ну и вот тебе файл с уже готовым проектом под этот шаблон](#)

[← Assembler](#) [← AVR](#) [← Макро язык](#) [← Программирование](#) [← Трюки](#) [← Флаги](#)

67 thoughts on “AVR. Учебный курс. Скелет программы”

Mazayac

22 Август 2008 в 16:56

«Суть в том, чтобы в автоматическом режиме, каждые несколько миллисекунд вызывать прерывание, в котором происходит подсчет импульсов от одометров»

Очень сложный, и неудачный подход.

Для подсчета импульсов с энкодеров в разы проще и эффективнее использовать прерывание по изменению состояния входа.

Пришел импульс — сгенерировалось прерывание, посчитали его быстренько и работаем спокойно дальше. Итого: 1 импульс на энкодере — одно прерывание, а не постоянная беготня с проверкой порта и сравнением с предыдущим состоянием :)

★ DI HALT

22 Август 2008 в 21:39

Да можно и так. Я подумал об этом в последний момент, но у меня на инты повешан драйвер движка, когда я его туда вешал я даже не думал про энкодеры на редуктор, Точнее у меня на них был собрана макетка, и я решил оставить обратную совместимость и ничего не менять. Так что чистейшая ошибка планирования, когда итоговая конструкция допридумывается походу дела такое сплошь и рядом :)))) Но зато появился кусок извратского кода для изучения. Тоже полезно, заодно показал как логика работает. :)

Mazayac

22 Август 2008 в 23:56

«на инты повешан драйвер движка»

Пора на Атмегу88 переходить :) У нее все порты (B, C, D) умеют внешние прерывания обслуживать, 24 входа для прерываний — с ума можно сойти.

★ DI HALT

23 Август 2008 в 0:04

Не продают у нас такое. А заказывать больно накладно выходит.

Mazayac

23 Август 2008 в 1:33

У нас тоже не продают. Приходится заказывать из Москвы, хорошо еще что удастся «прицепляться» к заказам для фирмы и не платить за доставку.

Lomaster

18 Ноябрь 2009 в 18:53

Не разобрался: прерывание по изменению состояния входа на ATМega8 можно только на 4 и 5 (INT0 и INT1) ноге делать?

★ **DI HALT**

18 Ноябрь 2009 в 21:41

Да. На мега88 есть PCINT которая на дофига ногах есть.

Stebanoid

27 Сентябрь 2008 в 21:58

Я что-то недопонял, или у вас тут

ANDI R16,11<<LGEAR ; Побитовое И регистра R16 и числа "00010000"

и тут

ANDI R17,11<<ODOL ; Побитовое И регистра R17 и числа "00010000"

опечатка?

11 вместо 1.

★ **DI HALT**

27 Сентябрь 2008 в 22:03

Да опечатка. Щас поправлю.

testicq

3 Февраль 2009 в 11:37

Очепятка:

«Каждый его биты заведуют режимами работы.»

RunningWolf

25 Ноябрь 2009 в 16:16

Народ, проясните пожалуйста следующий момент про одометры:

У DI HALT'a на схеме фотодиоды непосредственно к ногам МК подходят. На фотодиоды меня жаба задушила, взял фототранзисторы подешевле =) А они, злыдни, нужного эффекта не дают. Светишь, не светишь, все равно сигнал на МК не приходит. Кроме того, на разных ресурсах все норовят ОУ использовать для подобных датчиков. (Уточнение, буду использовать именно как датчик отражения.) Собственно вопрос: Можно ли фотодатчик подключать напрямую к ногам МК? Если «да» то какие характеристики у него должны быть?

P.S. Тихо радуюсь за себя =) За месяц с нуля при отсутствии радиолюбительского опыта удалось из радиоуправляемой машинки сделать робота с контактными датчиками, управляемого с установленного на нем КПК и удаленно по WiFi с компа. В целом все живет, сейчас занимаюсь отладкой. Платформа (p/u машинка)оказалась неудачной: моторчик фиговый и без датчика оборотов предсказать, сколько он проезжает за одно и то же время — нереально. А без этого нормального управления не сделать.

Кстати, попутный вопрос: Какие бы максимально просто определять положение передних (рулевых) колес? Сложность: конкретного положения им не задать — поворачиваться они могут только постепенно в процессе движения вперед или назад. Т.о. хочется знать, а в каком они положении в каждый момент времени. Пока пришла только одна мысль: переменный рез. и к АЦП но это как-то... не просто. К тому же движения рулевой рейки там минимальные (меньше 1 см в каждую сторону). В общем буду рад идеям.

★ DI HALT

25 Ноябрь 2009 в 17:23

Ты бы замерил его характеристики (хотя бы сопротивление) на свету и не на свету. Возможно ты пытаешься светить ему не тем спектром. Мои, например, совершенно не реагируют на зеленый и синий светодиоды, но вот зато на красный сразу же.

RunningWolf

25 Ноябрь 2009 в 20:34

Спасибо за ответ.

Не поверишь — замерял =)

Транзистор ИК, свечу я на него тоже ИК-диодом. Сопротивление меняется от 15-11 кОм до 7-4 кОм. Пробовал мерять напряжение — не меняется 8-[] Стабильно показывает 5 с копейками, которые через него бегут.

Доп. инф.: Стоит он между ногой МК и землей. Если ему ноги закоротить — МК сигнал принимает. Видимо, все-таки сопротивление у него дюже большое. М.б. постоянный резистор параллельно воткнуть? Вот только не соображу, как посчитать. Или как-то по-другому надо?

★ DI HALT

26 Ноябрь 2009 в 0:59

Изменения очень невелики. Поэтому у тебя и не хватает. Варианты тут такие:

1. Подобрать делитель так, чтобы малейшее изменение вызывало переход через логический уровень.
2. Смасштабировать через ОУ.
3. Можно попробовать измерительный мост приспособить, но не уверен.

RunningWolf

26 Ноябрь 2009 в 3:49

Спасибо огромное.

Извини, еще попристаю.

Можно для идиотов чуть поподробнее:

- 1) Какие изменения будут достаточными для Atmega8?
- 2) Если взять не транзистор, а диод, станет ли лучше?
- 3) Мне казалось, что транзистор и диод — бинарные элементы. Т.е. ток либо идет и идет весь либо не идет. С чего оно ведет себя, как фоторезистор?
- 4) Делитель, это параллельный резистор? А можно формулу расчета или конкретный номинал для меня?
- 5) Что есть «измерительный мост»?

6) М.б. порекомендуешь конкретные ИК-диоды/транзисторы, которые можно ставить между МК и землей без танцев? Если прямо из каталога Ч&Д — вообще будет супер.

7) Твой-то робот как? С момента последней статьи год прошел.

P.S. Понимаю, что Гугл рулит, но во-первых от живого человека ответы приятнее. Во-вторых, большей частью в результатах валится промышленная реклама. В третьих, стиль у тебя понятный.

★ DI HALT

26 Ноябрь 2009 в 4:40

В даташите есть раздел:

Pin Thresholds and Hysteresis: там график зависимости лог уровня от напряжения питания. Так вот на 5 вольтах все что ниже 1.4 вольт — это ноль. А все что выше 1.8 вольт единица. Между ними граничное значение где может считываться и так и так.

Так вот, твой транзистор и резистор должны образовать делитель напряжения. Читай основы на пальцах в 1...4 части про делитель было расписано. В общем, сопротивление должно быть подобрано так, чтобы изменение верхнего плеча делителя (твой транз) таскало напругу через эту границу в 1.8-1.4 вольта.

5) таки в гугл

6) модели не помню, я их обычно выпаиваю из старых дисководов.

7) стоит :)

RunningWolf

27 Ноябрь 2009 в 4:27

И еще раз спасибо.

Делитель помог — заработало. Завтречка вместо трех последовательных прикуплю один и запихаю наконец в работа.

RunningWolf

29 Ноябрь 2009 в 9:00

Похвастаюсь немного =)

Одометр довел до рабочего состояния и вставил в платформу. Работает на отражении от метки на колесе.

Сегодня добил софтовые задачи:

1) Поддержки заданной скорости (задаю желаемую скорость, а прошивка рулит ШИМОМ)

2) Проезда заданного расстояния.

Все стало ездить гораздо предсказуемее =)

Надо еще китайский магнит на руле на серву поменять и будет мне счастье.

R_ura

2 Март 2010 в 0:53

Все это хорошо, хоть и весьма усложнено. Но это теория. На практике будет «дребезг контактов», датчик может оказаться в пограничном положении. Надо писать функцию проверки истинности импульса. Вообще-то не пойму, зачем вам для простого счета импульсов нужны именно прерывания?? В каждом цикле можно опрашивать порты, как обычные входы. Есть сигнал > процедура проверки > подтвердился факт импульса > установился флаг события, уложился в РАМ (либо сразу вызвался счетчик), далее счетчик проверил флаг импульса. Если флаг установлен > добавил себе единичку или вычел, смотря в какую сторону ехал.. В следующем цикле этот флаг сбросился там же, где и устанавливался или в самом счетчике после счета.. Счетчики тоже проще организовать РАМ. Опрос порта займет несколько микросекунд, остальное время можно МК нагружать чем хочешь. А по вашему методу у вас очень скоро просто закончатся и регистры и счетчики.))

★ DI HALT

2 Март 2010 в 9:43

А если мы делаем что то медленное и опрос пропустит пару импульсов? Это же критично!

Дребезга тут нет, т.к. датчик работает быстро и весь остальной цикл это дело сглаживает

R_ura

2 Март 2010 в 12:17

Что можно делать настолько медленное? Может и неправильно с «академической» точки зрения (я не программист), но у меня большинство внешних событий вызывает установку соответствующего флага в RAM, а в основной программе в каждом цикле проверяется его состояние. Так очень удобно — если у меня по , скажем, импульсу_X происходит и счет в счетчике, и проверка выхода счетчика за пределы переполнения, и защиты всевозможные. Так же в дальнейшем к этому флажку можно легко подвязаться любой подпрограммой. Получается очень удобно добавлять новые возможности. Есть флаги, которые устанавливаются и сбрасываются основной логикой программы, а уж опираясь на них можно ыешать кучу , так сказать «сервисных» и контрольных функций. Если все эти программы обработок и контроля вызывать сразу после прихода импульса, то получится достаточно громоздкая и неуклюжая конструкция. Конечно, если нужна очень высокая скорость, то импульсы надо считать по прерываниям. Но вот где взять в Меge8 8 внешних прерываний, да так чтобы и осталось еще что полезное из периферии свободным.

В моем случае, импульс успевает 50 раз проверится на истинность, посчитаться, еще 50 раз провериться на заднем фронте (спаде). А

★ DI HALT

2 Март 2010 в 13:18

В данном случае это трудно представить. Но если поставить оптический энкодер с частотой выдачи импульсов в килогерцы? Тогда проблема вылезит в полный рост.

R_ura

2 Март 2010 в 15:02

Можно просто чаще опрашивать входа, скажем за цикл не один раз, а десять.

Опрос-то всего-ничего:

```
sbis pinX, x
```

```
rjmp a1
```

```
—
```

```
a1: sbis pinX, x
```


rjmp a2

—

a2:

Это практически не занимает ни время МК, ни его ресурсы. Понятно, что это до определенного потолка быстродействия, ну так и по прерывания такая же картина. Вроде бы примитивно, но зато удобно, и регистры все свободны. У меня на программу в 18кБ из регистров занято штук шесть. Да и вся периферия практически свободна, можно еще функций навешивать и навешивать.

★ DI HALT

2 Март 2010 в 18:32

Но есть риск прозевать событие. Неважно по какой причине. Не попало в опрос и все тут. С прерываниями такого не выйдет.

А чего жалеть регистры? Вот когда будет поджимать тогда и разберемся :)

R_ura

2 Март 2010 в 12:52

Di, кстати, а как проверить истинность импульса (в смысле, что не короткая помеха проскочила) если считать их по прерываниям? Как я понимаю, после перехода в обработчик прерывания, флаг прерывания сбрасывается, и что дальше? Как проверить, что это именно импульс? Обычно перепроверяют либо через некоторый промежуток времени, либо N-ное кол-во раз. А с прерыванием что-то не могу придумать. Сдается мне, что никак. Ну для машинки такой метод годится, а как быть, если до датчика 20 м кабеля? Там же помехи, наводки могут такого «насчитать», что за голову схватишься! Прошу прощения за вездливость, просто всегда интересно мнение более опытного товарища, а то у меня ведь — так, самодеятельность.

★ DI HALT

2 Март 2010 в 13:17

Если есть риск того, что это помеха то только повторной проверкой.

Из прерывания можно по таймеру сделать через некоторое время проверку. Т.е. суть прерывания поймать событие в любом случае. А уж как оно будет обрабатываться потом это дело другое.

Blacky

15 Март 2010 в 20:41

DI HALT, я заметил, что в твоём коде ты (вроде) не делаешь задержек после того, как поймал нажатие кнопки. Она у тебя не дребезжит? Или ты как-то борешься с этим аппаратно? Триггер? Вообще, мне интересно)

★ DI HALT

15 Март 2010 в 20:44

Дребезг мешает только тогда, когда надо подсчитывать число нажатий или что то подобное. Если важен только лишь факт нажатия кнопки (единоразовый), то его можно не учитывать.

Blacky

15 Март 2010 в 21:02

А сколько времени (в среднем) примерно длится дребезг при «нормальном» нажатии у «нормальной» кнопки?

★ DI HALT

15 Март 2010 в 21:19

Нормальной это какой? У китайской SWT я дребезг не смог увидеть даже на осциллографе 40мгц. А у советской ТМ-2 успокаивать приходилось чуть ли не в 3мс.

Blacky

15 Март 2010 в 22:08

Нормальной — это значит средненькой такой по цене кнопочки в магазине. Твоей советской ТМ-2 наверняка уже больше 20ти лет и использовалась она не раз. Но то что у новых зарубежных кнопок дребезга практически нет, радует. Не так все плохо оказывается, как пугают в книжках.

★ **DI HALT**

16 Март 2010 в 1:29

ТМ-2 тоже новая. Просто у ней конструкция более расхлябаная.

Blacky

16 Март 2010 в 1:41

Ну дак значит российская, а не советская. Советский союз был давно)

★ **DI HALT**

16 Март 2010 в 1:47

Ну валяться на складе с 89 года можно было без особых проблем. По крайней мере она была не паяная, не окисленная, а на корпусе клеймо 89 год. =))))

dima_m

7 Июнь 2010 в 13:48

DIHALT привет. Почему например файл с макросами располагается вначале? Например можно ли его расположить после прерываний? На что это повлияет? И вообще если поменять структуру расположения вложенный файлов, это допускается? Или только так как указано выше?

★ **DI HALT**

7 Июнь 2010 в 14:46

Компилятор парсит исходник сверху вниз и если он, не найдя макросов, нарвется на их использование будет эррор. Плюс я не знаю как он отреагирует на макросы в исходном коде. Может вкомпилит их все. Поэтому я располагаю их за пределами .CSEG

shiva

12 Ноябрь 2010 в 15:39

Секции внешней и внутренней инициализации периферии пока пусты, но ненадолго.

Далее идет разметка кода:

```
; Internal Hardware Init =====
```

```
; End Internal Hardware Init =====
```

```
; External Hardware Init =====
```

```
; End Internal Hardware Init =====
```

```
; Run =====
```

```
; End Run =====
```

в 4-ой строке должно быть не End Internal Hardware Init, а End External Hardware Init.

в файле проекта также

shiva

12 Ноябрь 2010 в 15:43

у Белова перед циклом программы идет инициализация портов. у Ревича тоже. инициализацию пихать в секцию
; Internal Hardware Init =====

; End Internal Hardware Init =====

?

и еще совсем уж глупый вопрос — что произойдет если порты вообще не инициализировать?

★ DI HALT

12 Ноябрь 2010 в 15:58

Что такое инициализация? Это настройка периферии в нужный режим работы. Только и всего. Ну и избавление от умолчаний. Т.е. если после сброса порт находится уже в нужном нам режиме, то это совсем не значит, что его не нужно заново в этот режим выставить. Кто знает как отработает схема сброса и будет ли при следующем сбросе или сбое порт в нужном режиме? Поэтому то все каждый раз заново и выставляют при старте. Чтобы точно знать, что у нас все настроено правильно

avk1963

11 Февраль 2011 в 21:02

DI HALT, подскажите, пожалуйста, где можно почитать про инициализацию USB в AVR (1286, 1287, ...) на ассемблере. На Вашем сайте (за сайт — огромное спасибо) я этого что-то не нашел. Так, чтобы было достаточно подробно описано:

1. Включение внутреннего стабилизатора напряжения
2. Настройка интерфейса блока PLL
3. Разрешение работы блока PLL и ожидание входа его в состояние захвата (lock)
4. Разрешение работы USB-интерфейса
5. Настройка USB-интерфейса (скорость, настройка конечных точек и др.)
6. Ожидание информационного подключения USB VBUS
7. Присоединение к шине

На сайте Atmel примеры есть на Си (не всегда работающие, несмотря что фирменные), а вот на Asm — нет.

★ **DI HALT**

11 Февраль 2011 в 21:14

Не знаю, никогда не интересовался USB плотно.

ingor

10 Декабрь 2010 в 11:55

Добрый день!

У меня снова повился вопрос: Делаю схему на ATtiny13A (ATtiny13V) питание +2 В. В даташите для питания 1,8-2,7В рекомендованно использовать тактовую частоту ≤ 4 МГц. Внутренний генератор можно настроить только на 4,8 МГц. Как сделать 4 МГц? С помощью OSCCAL? Ток я не понял, что это за зверь и как он работает. Объясните пожалуйста... если можно с примером..

Заранее спасибо. :)

★ **DI HALT**

11 Декабрь 2010 в 6:22

Можешь попробовать через oscal загнать в минус. Просто вписав туда минимальное значение.

либо через CLKPRE поделить частоту. Тебе надо точно 4мгц или 2х хватит?

ingor

11 Декабрь 2010 в 12:56

Добрый день продолжается!

Спасибо за подсказку. Я пропустил описание CLKPR когда смотрел описание. В запаре был. Мне достаточно м 2 МГц. Но тут дилема. Два варианта реализации:

1. При программировании установить FUSE биты CKSEL 01 получив тем самым 4,8 МГц, а потом при запуске МК в системе программно задать CLKPR — 0001 поделить $4,8/2 = 2,4$ МГц.
2. Не париться с Фьюзами (ещё ни разу с ними дела не имел) и сразу в программе задать CLKPR — 0010 поделив $9,6/4 = 2,4$ МГц.

А дилема в чём? Прокатит ли вариант 2 при питании 2 вольта? Я, в общем, склоняюсь к варианту 1.

★ DI HALT

12 Декабрь 2010 в 2:58

Оба варианта одинаковые. 2 проще и безопасней.

gks5

5 Июнь 2011 в 10:29

Имеются опечатки, возможно, это метки для защиты копирайта, например, на

<http://shop.easyelectronics.ru/index.php?productID=147>

адаптера

мы можешь

Имхо, логическая опечатка:

Хотя в разделе

<http://easyelectronics.ru/category/avr-uchebnyj-kurs> Стартовая инициализация

«убиваем все регистры от первого до последнего» в цикле,

в разделе

<http://easyelectronics.ru/avr-uchebnyj-kurs-vtoraya-programma.html#more-30>

это не используется, идет тупой перебор регистров.

Вопрос по существу, можете сделать сравнительный анализ программаторов отладчиков (расхвалить свой товар)

Отладочная плата PinBoard <http://shop.easyelectronics.ru/index.php?productID=147>
<http://www.electronsik.ru/card/otladochniy-komplekt-stk500-dlya-avr-59500>
<http://www.new-technik.ru/product/stk500/>

★ DI HALT

5 Июнь 2011 в 12:29

у STK500 очень хороший программатор с возможностью HVPROG. И возможность втыкать разные МК.

Больше там ничего особо нет. JTAG отладчик придется покупать отдельно, дисплей отдельно. На самой плате из органов управления только кнопки и светодиоды. Был бы он за 1500 еще можно было купить, но не за 5000.

Тот что по второй ссылке — это просто программатор, но весьма неплохой.

tabarigen

26 Июль 2011 в 22:15

RunningWolf 25 Ноя 2009 16:16

Народ, проясните пожалуйста следующий момент про одометры:

У DI HALT'a на схеме фотодиоды непосредственно к ногам МК подходят. На фотодиоды меня жаба задушила, взял фототранзисторы подешевле =)=

=====

извините за дурацкий вопрос.. но разве к этой статье схема приложена??

★ DI HALT

27 Июль 2011 в 3:41

Когда то давно статья была существенно другой. Потом была почти полностью переписана под другую платформу

SerpSB

16 Сентябрь 2011 в 21:32

Чтобы обнулялась последняя ячейка RAM, в текст программы и в файл шаблона надо внести исправления, на которые указывал Andriy в предыдущей статье, а именно заменить:

CPI ZL,Low(RAMEND) на CPI ZL,Low(RAMEND+1)

Dima030888

2 Декабрь 2011 в 1:52

У меня вопрос. Скорее всего он тупой)) просто я изучаю весь курс поп порядку. В макросах ты используешь конструкцию if-then-else. Но на скока я понял в ассемблере нет таких команд. ведь у тебя целая статья про типовые конструкции и про то как создать if then else через БРАНЧИ. я посмотрел файл m16def.inc но там ничего не написано про это. это случайно не ммакросы (.if и .else)? И вообще что означает точка перед командой?

★ DI HALT

2 Декабрь 2011 в 17:14

эти иф елзе относятся к компилятору, а не командам процессора. Благодаря им он на этап компиляции определяет какие куски кода компилировать, а какие выбросить, что куда подставить и все подобное. На выходе же получается чистый ассемблерный листинг.

Dima030888

6 Декабрь 2011 в 1:53

ОК. этот момент я понял. т.е в случае сооружения структуры из бранчей в код будет прописано и ИФ и ЭЛСЕ. а здесь в нашем случае пропишется либо ИФ, либо ЭЛСЕ, в зависимости от начальных условий. В данном случае от переменных, которые мы подставляем в

макрос. ок. ТОГДА ОПЯТЬ ТУПОЙ ВОПРОС. нам же главное результат, так? почему я не могу соорудить такой макрос, который бы мне , к примеру сравнивал величины. и делал бы либо то, либо другое. Я понимаю, что получается не совсем программа, а, по большому счету код будет прописан «в лоб», только пишем не мы а компилятор. т.е не будет логики. прокатит только для отдельно взятого случая. ток вот ни пофигу ли нам??? в частных случаях прокатило бы и это. или чего то изначально не понимаю?)) заранее спасибо

★ DI HALT

6 Декабрь 2011 в 14:18

Макросы оперируют только тем, что можно вычислить на этапе компиляции. Если у тебя масса логики решается уже тут — то ее надо тут и сделать. Т.к. появляется читаемость программы, а все что надо посчитать будет посчано компилером. Но это же все только константы. Ты можешь не пользуя макросами все сам посчитать и подставить.

А когда прога будет уже зашита, то например ты не сможешь сделать переход на основании пришедшей в порт циферки.

Vitalii

19 Март 2013 в 0:40

DI HALT, доброго времени суток.

Я тут пытаюсь соорудить одно устройство на ATtiny2313, использую этот шаблон, но авр студия ругается, пишет вот такие замечания:

error: Overlap in.cseg: addr=0x14 conflicts with 0x14:0x15 (это на строку: «OUT SPL,R16»)

error: Overlap in.cseg: addr=0x16 conflicts with 0x16:0x17 (это на строку: «OUT SPH,R16»)

и так далее. всего только на сам шаблон без самой программы 11 ошибок.

Подскажи, в чем причина? пока самому не получается разобраться. спасибо.

★ DI HALT

19 Март 2013 в 0:42

1. Убедись, что адрес который пишется в SPL указан как RAMEND и инклюдник от тини2313 взят, а не от чего то другого.
2. У тини2313 нет SPH регистра. Ее озу укладывается в 255 байт.

recceer

21 Август 2013 в 12:20

В коде «Инициализация ядра. Память, стек, регистры» есть ошибка. Когда чистим SRAM от начала (SRAM_START) до конца (RAMEND), то последняя ячейка (RAMEND) не очищается. Почему? Да потому, что выполнилось сравнение что Z=RAMEND и заканчиваем цикл без обнуления этой ячейки памяти RAMEND. Поэтому надо вставить после цикла еще одну строку для обнуления.

=====

; Start coreinit.inc

RAM_Flush: LDI ZL,Low(SRAM_START) ; Адрес начала ОЗУ в индекс

LDI ZH,High(SRAM_START)

CLR R16 ; Очищаем R16

Flush: ST Z+,R16 ; Сохраняем 0 в ячейку памяти

CPI ZH,High(RAMEND) ; Достигли конца оперативки?

BRNE Flush ; Нет? Крутимся дальше!

CPI ZL,Low(RAMEND) ; А младший байт достиг конца?

BRNE Flush

=====еще одна строка для обнуления ячейки памяти RAMEND

ST Z,R16 ; Сохраняем 0 в последнюю ячейку памяти (RAMEND)

;=====еще одна строка для обнуления ячейки памяти RAMEND

Предлагаю автору проверить и поправить в коде источника.

Wan-Derer

21 Октябрь 2013 в 21:59

Что-то большинство коментов от какой-то другой статьи. Ты бы их поносил, всё равно бессмысленны без контекста. И название «Скелет программы», а ссылка /avr-uchebnyj-kurs-vtoraya-programma.html — не соответствует :)

Jazon

28 Сентябрь 2014 в 23:10

Что-то у меня разрыв шаблона.

Комментарии и статья как-то связаны???

★ DI HALT

29 Сентябрь 2014 в 5:32

Может уже и нет. Статьи иногда переписываются практически полностью.

broakelon

16 Ноябрь 2014 в 21:15

Решил писать программку по данному шаблону, все куски кода из статьи вписывал в тойже последовательности, инклюдка для мег8, но на кусок с таблицей прерываний авр-студия выдает при компиляции матюки

.....\Kom2x2_14.asm(144): error: Overlap in .cseg: addr=0x14 conflicts with 0x14:0x15

и так далее до \Kom2x2_14.asm(171): error: Overlap in .cseg: addr=0x28 conflicts with 0x28:0x29

При убивании строчки .ORG INT_VECTORS_SIZE компилит без ошибок.

Собственно немогу понять чего он на нее ругается?

★ DI HALT

16 Ноябрь 2014 в 21:37

У тебя где то ORGi наложились друг на друга. Проверь это внимательно. Они должны быть строго по порядку возрастания, а у тебя где то перепутались. В результате возникла ошибка перекрытия Overlap.

broakelon

16 Ноябрь 2014 в 21:53

Странно — кроме этой таблички там пока нет никаких ORGov, перепроверил инклюдки — там вообще нет ORGov
Собственно строка .ORG INT_VECTORS_SIZE мне не совсем понятна — «INT_VECTORS_SIZE» это должен быть адрес в памяти?
и этим именем он должен где-то быть назван с помощью .def ? или я ошибаюсь?

★ DI HALT

16 Ноябрь 2014 в 21:56

Либо ORG возник после физического кода, т.е. ты навтыкал кода (тех же векторов) скажем на 10 байт, а ORG предписывает на 9 и возникло перекрытие.

Да это адрес в памяти, он для каждого контроллера свой и прописан где-то в def файлах. Если не найдешь, то проверь, что используешь assembler2 и файлы из него, т.к. в ассемблер1 таких дефайнов много меньше (да и сам ассемблер1 намного слабей чем версия 2)

Таблицу векторов правильно нарисовал? А то если ты ее копипастнул откуда-то, то там может быть больше векторов чем есть на самом деле, т.к. она может быть с другого контроллера. Вот тебе и оверлап вылез.

broakelon

16 Ноябрь 2014 в 22:10

Спасибо за терпение Di, разобрался — копипастнул табличку из статьи — а она под мегу16 — у меня же мега8 — вот и матюкалось — исправил — работает, учим дальше RTOS.))

Full_30h

9 Декабрь 2015 в 10:21

Не очень понимаю зачем так самозабвенно обнуляются регистры? Есть там мусор, ну и хер с ним. А берётся регистр в оборот так всё равно туда либо что то предварительно грузится. Или это чисто академический пример?

maks.portnyagin

10 Март 2016 в 7:36

Ді доброго времени. Могли бы помочь перенсти RTOS на мегa328? Вектора поправил, uart поправил, а вот с таймерами разобраться не могу.