Done at Home
Программирование микроконтроллеров avr

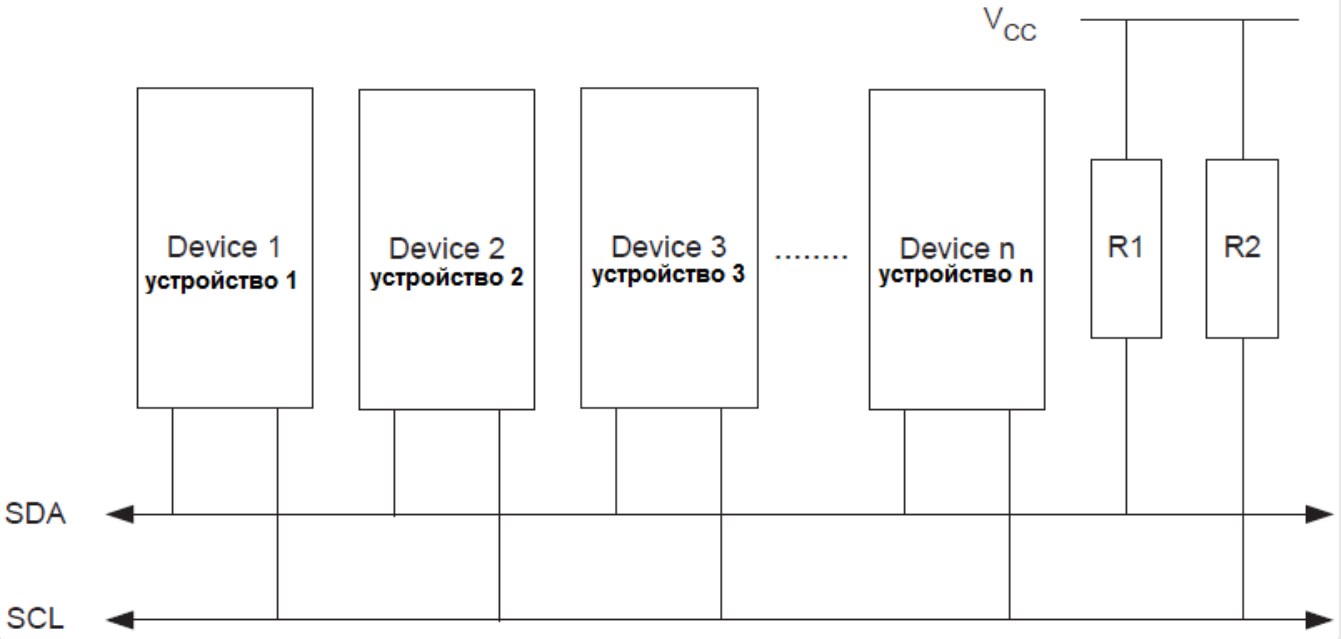
AVR микроконтроллеры для начинающих (урок 11) I2C(TWI)-интерфейс

admin | 16.02.2014 0 Comment

AVR микроконтроллеры для начинающих (урок 11) I2C(TWI)-интерфейс

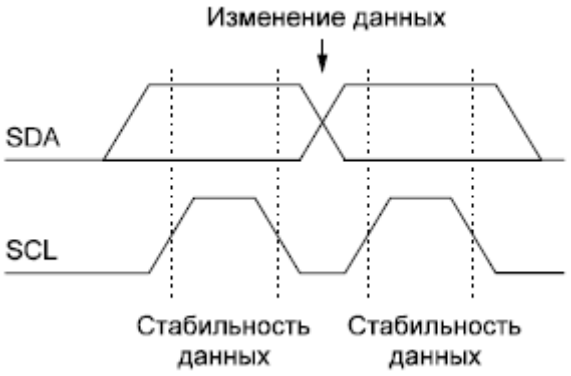
I2C интерфейс представляет собой две двунаправленные линии связи – SDA и SCL. По SDA передаются данные, по SCL тактовый сигнал. Обе линии подтянуты через резисторы к плюсу питания. Фирма Philips за использование названия этого интерфейса требует лицензионных отчислений, поэтому в микроконтроллерах Atmel используется собственное название TWI – two-wire interface («двухпроводной интерфейс»).

Вот некоторые достоинства шины I2C:
· Требуется только две линии – линия данных (SDA) и линия синхронизации (SCL). Каждое устройство, подключенное к шине, может быть программно адресовано по уникальному адресу. В каждый момент времени существует простое отношение ведущий/ведомый: ведущие могут работать как ведущий-передатчик и ведущий-приёмник.
· Шина позволяет иметь несколько ведущих, предоставляя средства для определения коллизий и арбитраж для предотвращения повреждения данных в ситуации, когда два или более ведущих одновременно начинают передачу данных. В стандартном режиме обеспечивается передача последовательных 8-битных данных со скоростью до 100 кбит/с, и до 400 кбит/с в «быстром» режиме.
· Встроенный в микросхемы фильтр подавляет всплески, обеспечивая целостность данных.

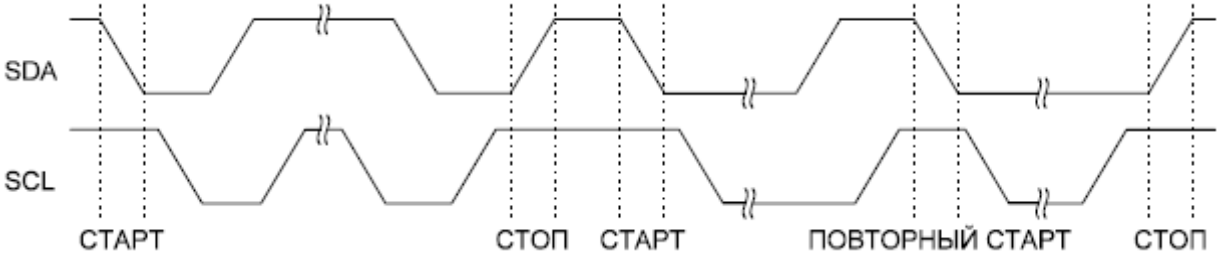


Формат передачи данных

Интерфейс I2C является синхронным, поэтому каждый передаваемый бит данных на линии SDA сопровождается импульсом на линии синхронизации SCL. Уровень данных должен быть стабильным, когда на линии синхронизации присутствует лог. «1» Исключением для этого правила является генерация условий (СТАРТ/СТОП).

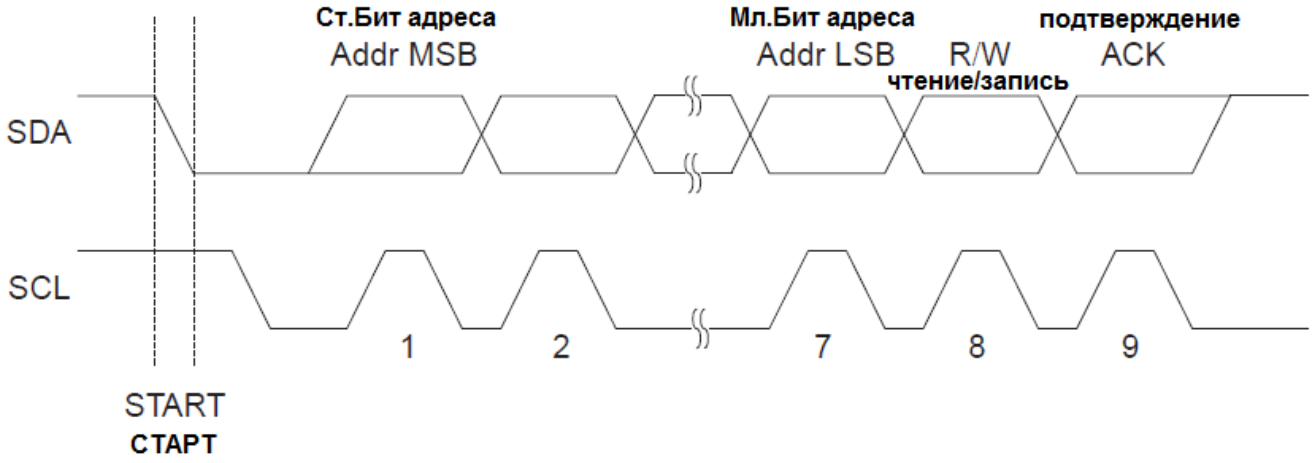


Ведущее устройство инициирует и заканчивает передачу данных. Передача инициируется, когда ведущий формирует условие СТАРТ на шине, и прекращается, когда ведущий формирует на шине условие СТОП. Между условиями СТАРТ и СТОП шина считается занятой и в этом случае никакой другой мастер не может осуществлять управляющие воздействия на шине. Существуют особые случаи, когда новое условие СТАРТ возникает между условиями СТАРТ и СТОП. Данный случай именуется как “Повторный старт” и используется при необходимости инициировать мастером новый сеанс связи, не теряя при этом управление шиной. После “Повторного старта” шина считается занятой до следующего СТОП. Это идентично поведению после СТАРТ.



Формат адресного пакета

Все передаваемые адресные пакеты по шине I2C состоят из 9 бит, в т.ч. 7 бит адреса, один бит управления для задания типа операции чтения или записи (R/W) и один бит подтверждения (ACK). Если бит R/W = 1, то будет выполнена операция чтения, иначе – запись. Если подчиненный распознает, что к нему происходит адресация, то он должен сформировать низкий уровень на линии SDA на 9-ом цикле SCL (формирование бита подтверждения). Если адресуемое подчиненное устройство занято или по каким-либо другим причинам не может обслужить ведущее устройство, то на линии SDA необходимо оставить высокий уровень во время цикла подтверждения. Ведущий после этого может передать условие СТОП или “Повторного старта” для инициации новой пере-
дачи.

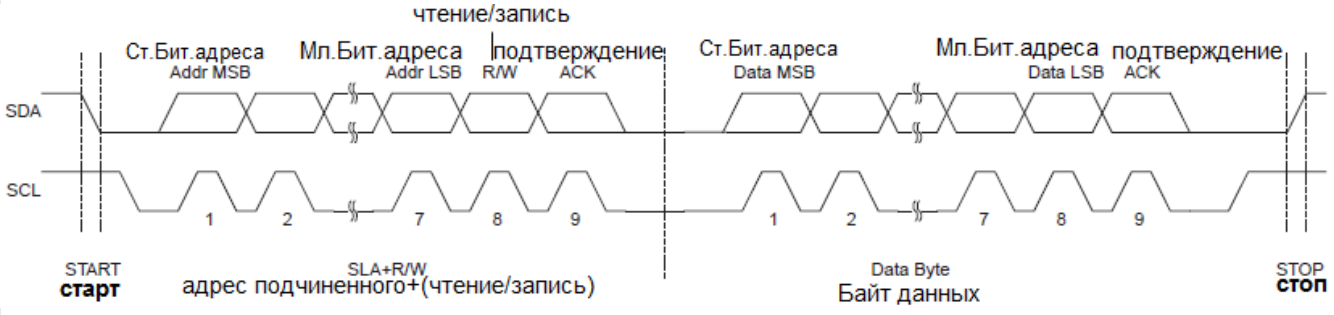


R/W – определяет последующее направление передачи данных. Бит квитирования – это ответ ведомого устройства на принятый адрес. Если адрес распознан, ведомый выдает на линию SDA низкий уровень. В противном случае на линии удерживается высокий уровень.
ACR - бит подтверждения. Низкий уровень означает ответ от устройства к которому обращается ведущий. Высокий уровень говорит о том, что устройство занято.

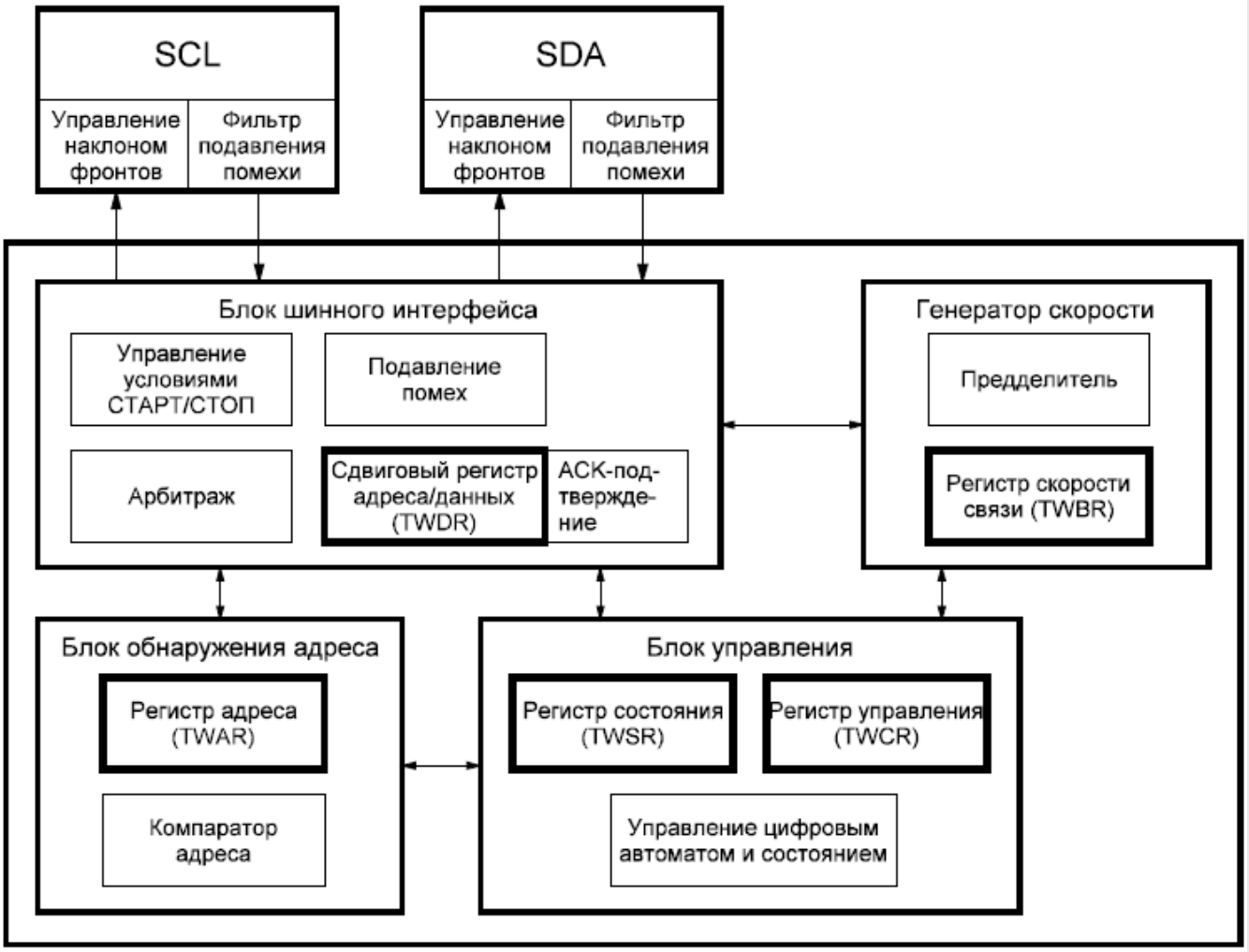
Пакеты данных

Пакеты данных состоят из байта данных и бита квитирования, то есть тоже имеют длину 9 бит. После приема каждого байта данных, принимающее устройство (приемник) отвечает передающему устройству (передатчику), устанавливая на линии SDA низкий уровень (это и есть бит квитирования). Если принимающее устройство получило последний байт или больше не может продолжать прием данных, оно должно “оставить” на линии SDA высокий уровень.

Типичная передача данных



Функциональная схема блока шины I2C



Регистры I2C (TWI)

В выделенных жирной линией прямоугольниках находятся регистры настройки I2C модуля в avr микроконтроллере. Для того чтобы дальше разбираться с I2C модулем, нужно ознакомиться с его регистрами. Разбор регистров будет вестись на примере микроконтроллера atmega32. В других микроконтроллерах возможны небольшие отличия.

Регистр скорости передачи TWBR (TWI Bit Rate Register)

Bit	7	6	5	4	3	2	1	0	
	TWBR7	**TWBR6**	**TWBR5**	**TWBR4**	**TWBR3**	**TWBR2**	**TWBR1**	**TWBR0**	**TWBR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Бит 7:0 — биты этого регистра определяют частоту работы модуля I2C. Частота также зависит от тактовой частоты работы микроконтроллера, и значения в младших битах(TWPS0,TWPS1) регистра TWSR.

Частота SCL сигнала, тактовая частота микроконтроллера, и значение регистров TWBR и TWSR связаны следующим соотношением:

$$\text{SCL frequency} = \frac{\text{CPU Clock frequency}}{16 + 2(\text{TWBR}) \cdot 4^{TWPS}}$$

Регистр данных TWDR (TWI Data Register)

Bit	7	6	5	4	3	2	1	0	
	TWD7	TWD6	TWD5	TWD4	TWD3	TWD2	TWD1	TWD0	TWDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	1	1	1	1	

Бит 7:0 — биты этого регистра хранит данные, которые мы либо хотим передать ведомому, либо получили от ведущего.

Регистр адреса TWAR (TWI Address Register)

Bit	7	6	5	4	3	2	1	0	
	TWA6	TWA5	TWA4	TWA3	TWA2	TWA1	TWA0	TWGCE	TWAR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	1	1	1	0	

Бит 7:1 — биты хранения значения адреса, по которому будет отзываться микроконтроллер, находясь в режиме ведомого .
Бит 0 — бит разрешения на отклик во время общих вызовов.

Статусный регистр TWSR (TWI Status Register)

Bit	7	6	5	4	3	2	1	0	
	TWS7	TWS6	TWS5	TWS4	TWS3	–	TWPS1	TWPS0	TWSR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	1	1	1	1	1	0	0	0	

Бит 7:3 — биты содержат статусный код. Биты доступны только для чтения, статусный код устанавливается TWI модулем аппаратно, после выполнения различных операций. Например, формирование состояния СТАРТ, передачи пакета данных и так далее. По значению статусного кода можно судить о результате операции. Выполнилась ли она успешно или нет.
Бит 2 — бит зарезервирован и читается как 0.
Бит 1:0 — биты влияющие на частоту SCL (зависимость наглядна в формуле для вычисления SCL)

TWPS1	TWPS0	Prescaler Value
0	0	1
0	1	4
1	0	16
1	1	64

Регистр управления TWCR (TWI Control Register)

Bit	7	6	5	4	3	2	1	0	
	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	TWCR
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Бит 7 — бит флага прерывания TWI модуля. Этот бит устанавливается аппаратно, когда TWI модуль завершает текущую операцию (формирование состояния СТАРТ, передачи адресного пакета и так далее). При этом если установлен бит глобального разрешения прерываний (бит I регистра SREG) и разрешены прерывания TWI модуля, то вызывается соответствующий обработчик.

Бит TWINT очищается программно, записью единицы. При выполнении обработчика прерывания этот бит не сбрасывается аппаратно, как в других модулях. Сброс флага TWINT запускает работу TWI модуля, поэтому все операции с регистром данных, статуса или адреса, должны быть выполнены до его сброса. Пока бит TWINT установлен, на линии SCL удерживается низкий уровень.

Бит 6 — бит разрешения подтверждения. Если бит TWEA установлен в 1, TWI модуль формирует сигнал подтверждения (ACK), когда это требуется. А требуется это в трех случаях: ведущее или ведомое устройство получило байт данных, ведомое устройство получило общий вызов, ведомое устройство получило свой адрес.

Бит 5 — флаг состояния СТАРТ. Когда этот бит устанавливается в 1, TWI модуль проверяет не занята ли шина и формирует состояние СТАРТ. Если шина занята, он будет ожидать появления на ней состояния СТОП и после этого выдаст состояние СТАРТ. Бит TWSTA должен быть очищен программно, когда состояние СТАРТ передано.

Бит 4 — флаг состояния СТОП. Когда этот бит устанавливается в 1 в режиме ведущего, TWI модуль выдает на шину состояние СТОП и сбрасывает этот бит. В режиме ведомого установка этого бита может использоваться для восстановления после ошибки. При этом состояние СТОП не формируется, но TWI модуль возвращается к начальному не адресованному состоянию.

Бит 3 — флаг конфликта записи. Этот флаг устанавливается аппаратно, когда выполняется запись в регистр данных (TWDR) при низком значении бита TWINT. То есть когда TWI модуль уже выполняет какие-то операции.

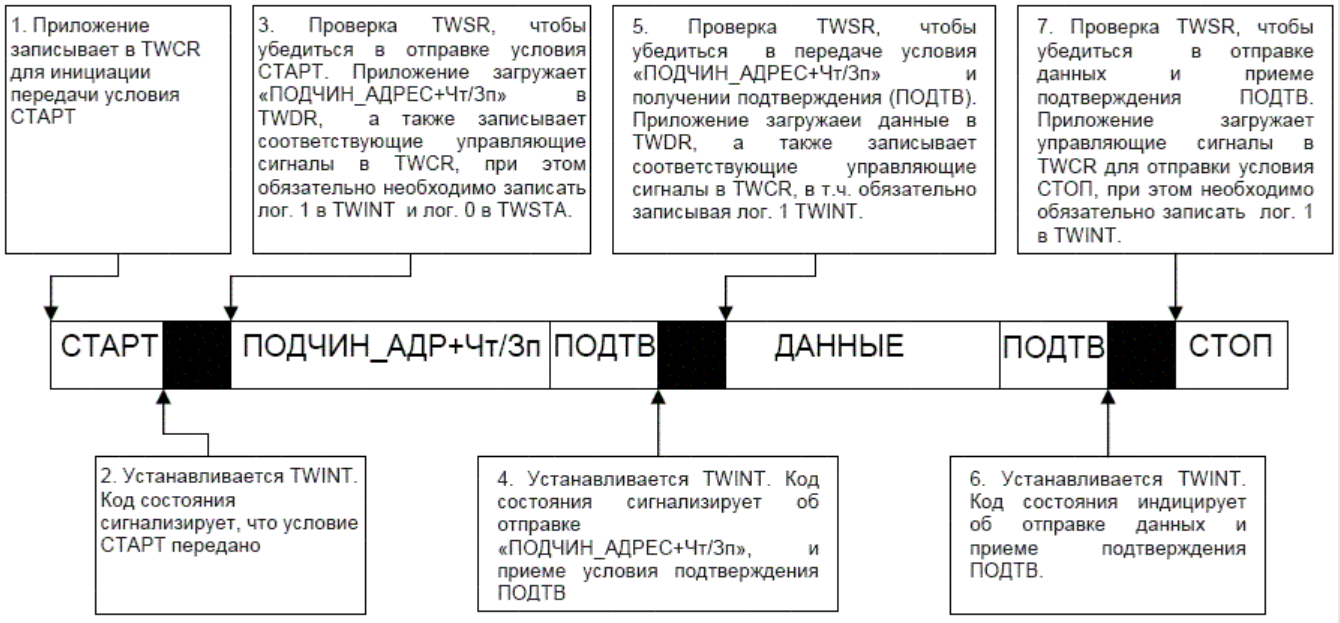
Флаг TWWC сбрасывается аппаратно, когда запись в регистр данных выполняется при установленном флаге прерывания TWINT.

Бит 2 — бит разрешения работы TWI модуля. Когда бит TWEN устанавливается в 1, TWI модуль включается и берет на себя управление выводами SCL и SDA. Когда бит TWEN сбрасывается, TWI модуль выключается.

Бит 1 — бит зарезервирован и читается как 0.

Бит 0 — разрешение прерывания TWI модуля. Когда бит TWIE и бит I регистра SREG установлены в 1 – прерывания модуля TWI разрешены. Прерывания будут вызываться при установке бита TWINT.

Типичная передача данных (подробнее)



Пример кода для СИ и Ассемблера

№	Пример кода на Ассемблере	Пример кода на Си	Комментарий
1	ldi r16, (1<<TWINT)|(1<<TWSTA)|(1<<TWEN) out TWCR, r16	TWCR = (1<<TWINT)| (1<<TWSTA)|(1<<TWEN)	Передача условия СТАРТ
2	wait1: in r16,TWCR sbrs r16,TWINT rjmp wait1	while (!(TWCR & (1<<TWINT))) ;	Ожидание установки флага TWINT. Этим индицируется завершение передачи условия СТАРТ
3	in r16,TWSR andi r16, 0xF8 cpi r16, START brne ERROR	if ((TWSR & 0xF8) != START) ERROR();	Проверка кода состояния TWI. Маскир. бит предделителяЕсли код состояния не равен СТАРТ, то переход на ERROR
	ldi r16, SLA_W out TWDR, r16 ldi r16, (1<<TWINT) | (1<<TWEN) out TWCR, r16	TWDR = SLA_W; TWCR = (1<<TWINT) | (1<<TWEN);	Загрузка ПОДЧИН_АДР + ЗАПИСЬ в регистр TWDR. Сброс бита TWINT в TWCR для начала передачи адреса
4	wait2: in r16,TWCR sbrs r16,TWINT rjmp wait2	while (!(TWCR & (1<<TWINT))) ;	Ожидание установки флага TWINT. Этим сигнализируется завершение передачи ПОДЧИН_АДР + ЗАПИСЬ и получение/неполучение подтверждения (ПОДТВ/НЕТ ПОДТВ).
5	in r16,TWSR andi r16, 0xF8 cpi r16, MT_SLA_ACK brne ERROR	if ((TWSR & 0xF8) != MT_SLA_ACK) ERROR();	Проверка значения регистра состояния.Маскирование бит предделителяЕсли состояние отличается от MT_SLA_ACK, то переход на ERROR
	ldi r16, DATA out TWDR, r16 ldi r16, (1<<TWINT) | (1<<TWEN) out TWCR, r16	TWDR = DATA; TWCR = (1<<TWINT) | (1<<TWEN);	Загрузка данных в TWDRСброс флага TWINT в TWCR для начала передачи данных
6	wait3: in r16,TWCR sbrs r16,TWINT rjmp wait3	while (!(TWCR & (1<<TWINT))) ;	Ожидание установки флага TWINTЭтим индицируется, что данные были переданы и принято/не принято подтверждение (ПОТДВ/НЕТ ПОДТВ).
7	in r16,TWSR andi r16, 0xF8 cpi r16, MT_DATA_ACK brne ERROR	if ((TWSR & 0xF8) != MT_DATA_ACK) ERROR();	Проверка значения регистра состояния TWI. Маскирование бит предделителя. Если состояние отличается от MT_DATA_ACK, то переход на ERROR
	ldi r16, (1<<TWINT)|(1<<TWEN)| (1<<TWSTO) out TWCR, r16	TWCR = (1<<TWINT)|(1<<TWEN)| (1<<TWSTO);	Передача условия СТОП

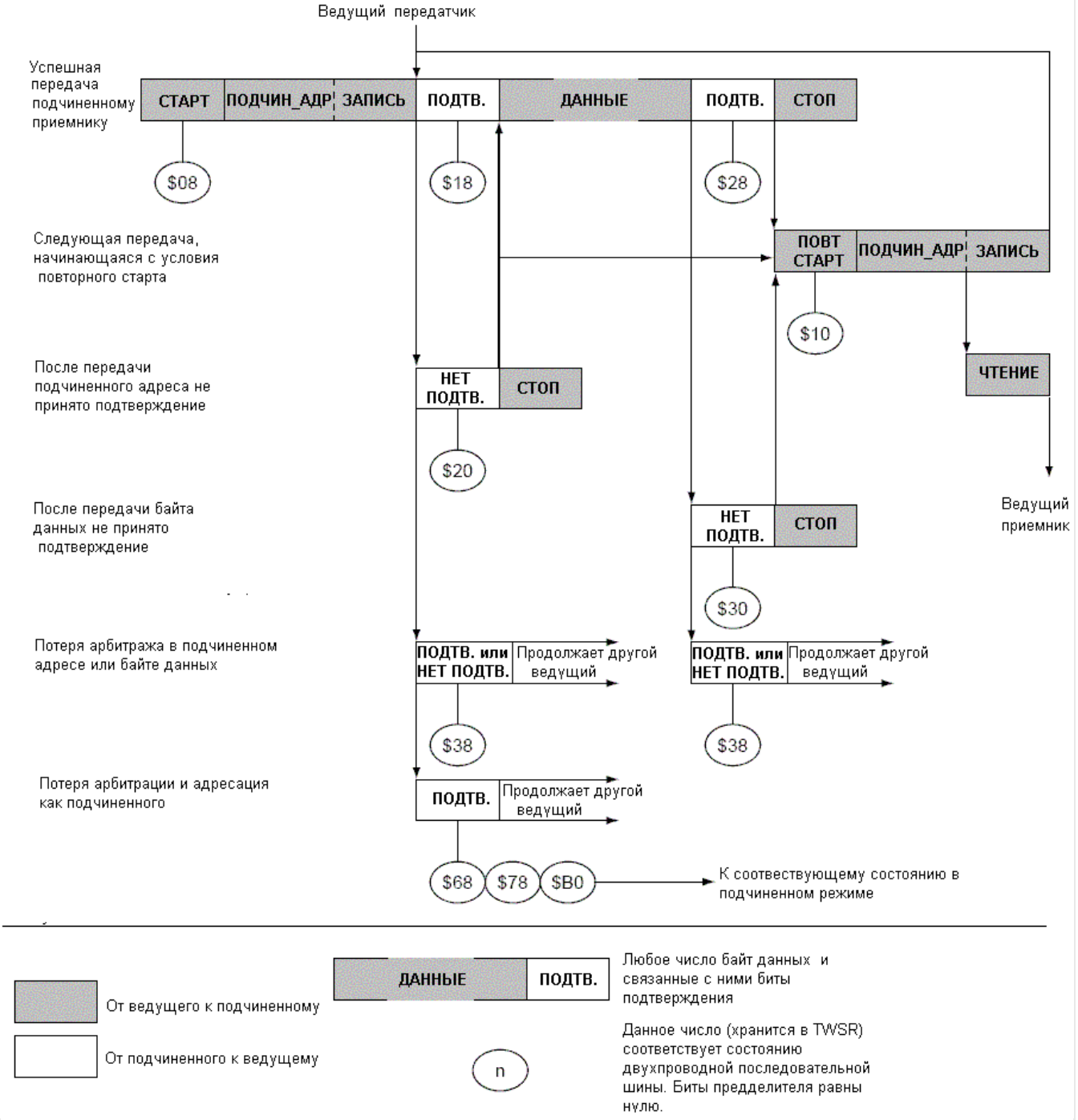
Теперь разберемся с возможными режимами работы нашего модуля I2C.

Есть четыре возможных варианта :

Ведущий передатчик [Master Transmitter (MT)], Ведущий приемник [Master Receiver (MR)],
Ведомый передатчик [Slave Transmitter (ST)], Ведомый приемник [Slave Receiver (SR)].

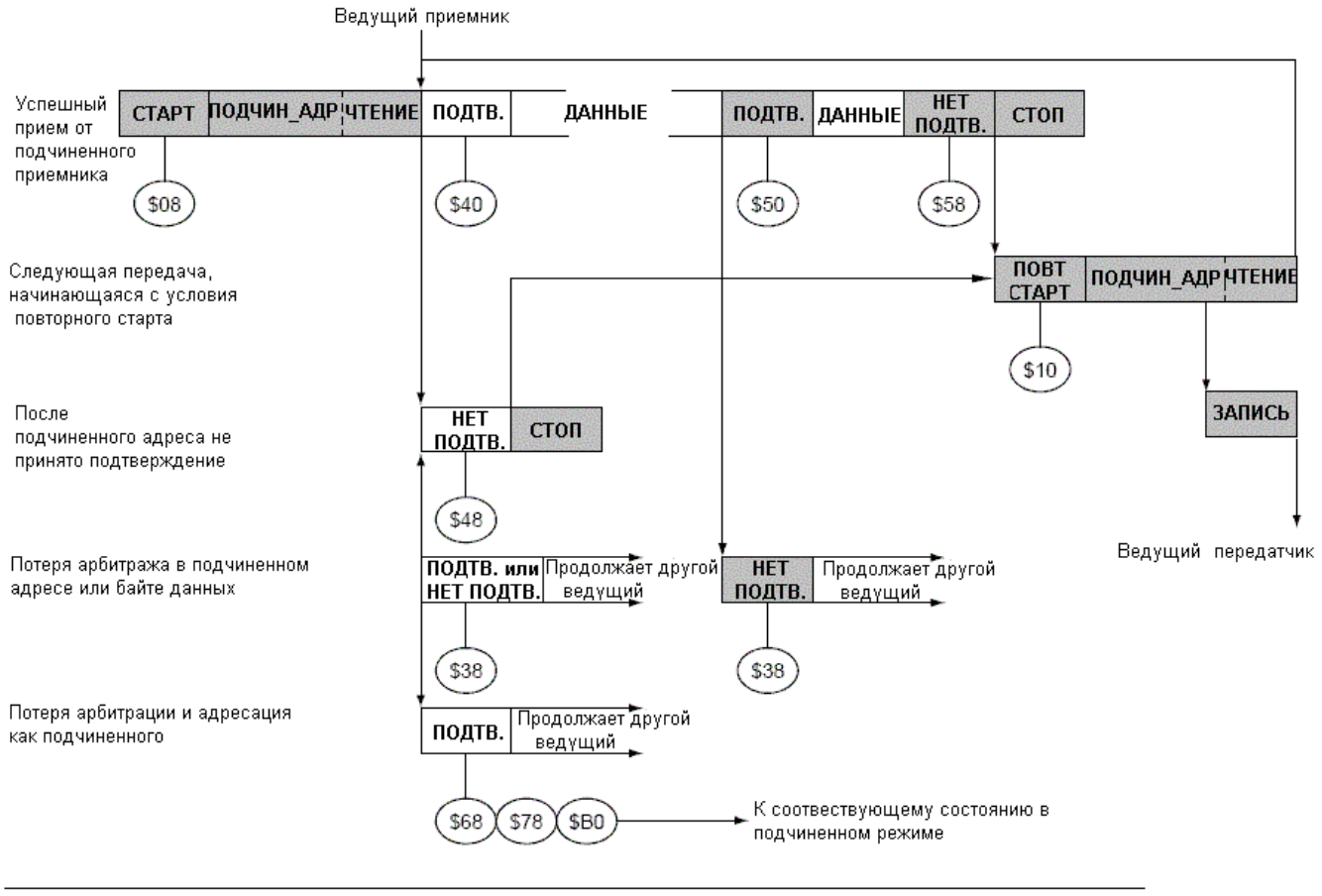
Ведущий передатчик [Master Transmitter (MT)]

Код состояния (TWSR), биты предделителя равны 0	Состояние двухпроводной последовательной шины и схемы двухпроводного последовательного интерфейса	Программные действия					Следующее действие, выполняемое схемой TWI
		Вйиз TWDR	В TWCR				
			STA	STO	TWINT	TWEA	
$08	Передано условие СТАРТ	Загрузка ПОДЧИН_АДР+ЗАПИСЬ	0	0	1	x	Передается ПОДЧИН_АДР + ЗАПИСЬ Принимается ПОДТВ. или НЕТ ПОДТВ.
$10	Передано условие ПОВТОРНЫЙ СТАРТ	Загрузка ПОДЧИН_АДР+ЗАПИСЬ	0	0	1	X	Передается ПОДЧИН_АДР + ЗАПИСЬ; Принимается ПОДТВ или НЕТ ПОДТВ
		или ПОДЧИН_АДР+ЧТЕНИЕ	0	0	1	X	Передается ПОДЧИН_АДР + ЧТЕНИЕ; Переход на режим ведущего приемника
$18	Передано ПОДЧИН_АДР+ЗАПИСЬ и принято ПОДТВерждение	Загрузка байта данных или	0	0	1	X	Передается байт данных, принимается или не принимается ПОДТВерждение
		действия без загрузки TWDR или	1	0	1	X	Передается ПОВТОРНЫЙ СТАРТ
		действия без загрузки TWDR или	0	1	1	X	Передается условие СТОП и сбрасывается флаг TWSTO
		действия без загрузки TWDR	1	1	1	X	Вслед за условием СТАРТ передается условие СТОП и сбрасывается флаг TWSTO
$20	Передано ПОДЧИН_АДР+ЗАПИСЬ и принято НЕТ ПОДТВ	Загрузка байта данных или	0	0	1	X	Передается байт данных, принимается или не принимается ПОДТВерждение
		действия без загрузки TWDR или	1	0	1	X	Передается ПОВТОРНЫЙ СТАРТ
		действия без загрузки TWDR или	0	1	1	X	Передается условие СТОП и сбрасывается флаг TWSTO
		действия без загрузки TWDR	1	1	1	X	Вслед за условием СТАРТ передается условие СТОП и сбрасывается флаг TWSTO
$28	Передается байт данных; принимается ПОДТВерждение	Загрузка байта данных или	0	0	1	X	Передается байт данных, принимается или не принимается ПОДТВерждение
		действия без загрузки TWDR или	1	0	1	X	Передается ПОВТОРНЫЙ СТАРТ
		действия без загрузки TWDR или	0	1	1	X	Передается условие СТОП и сбрасывается флаг TWSTO
		действия без загрузки TWDR	1	1	1	X	Вслед за условием СТАРТ передается условие СТОП и сбрасывается флаг TWSTO
$30	Передается байт данных; принимается НЕТ ПОДТВерждения	Загрузка байта данных или	0	0	1	X	Передается байт данных, принимается или не принимается ПОДТВерждение
		действия без загрузки TWDR или	1	0	1	X	Передается ПОВТОРНЫЙ СТАРТ
		действия без загрузки TWDR или	0	1	1	X	Передается условие СТОП и сбрасывается флаг TWSTO
		действия без загрузки TWDR	1	1	1	X	Вслед за условием СТАРТ передается условие СТОП и сбрасывается флаг TWSTO
$38	Потеря арбитража при передачи ПОДЧИН_АДР + ЗАПИСЬ или байта данных	Нет действий с TWDR или	0	0	1	X	Освобождается двухпроводная шина и вводится неадресуемый подчиненный режим
		нет действий с TWDR	1	0	1	X	Передача условия СТАРТ после освобождения шины



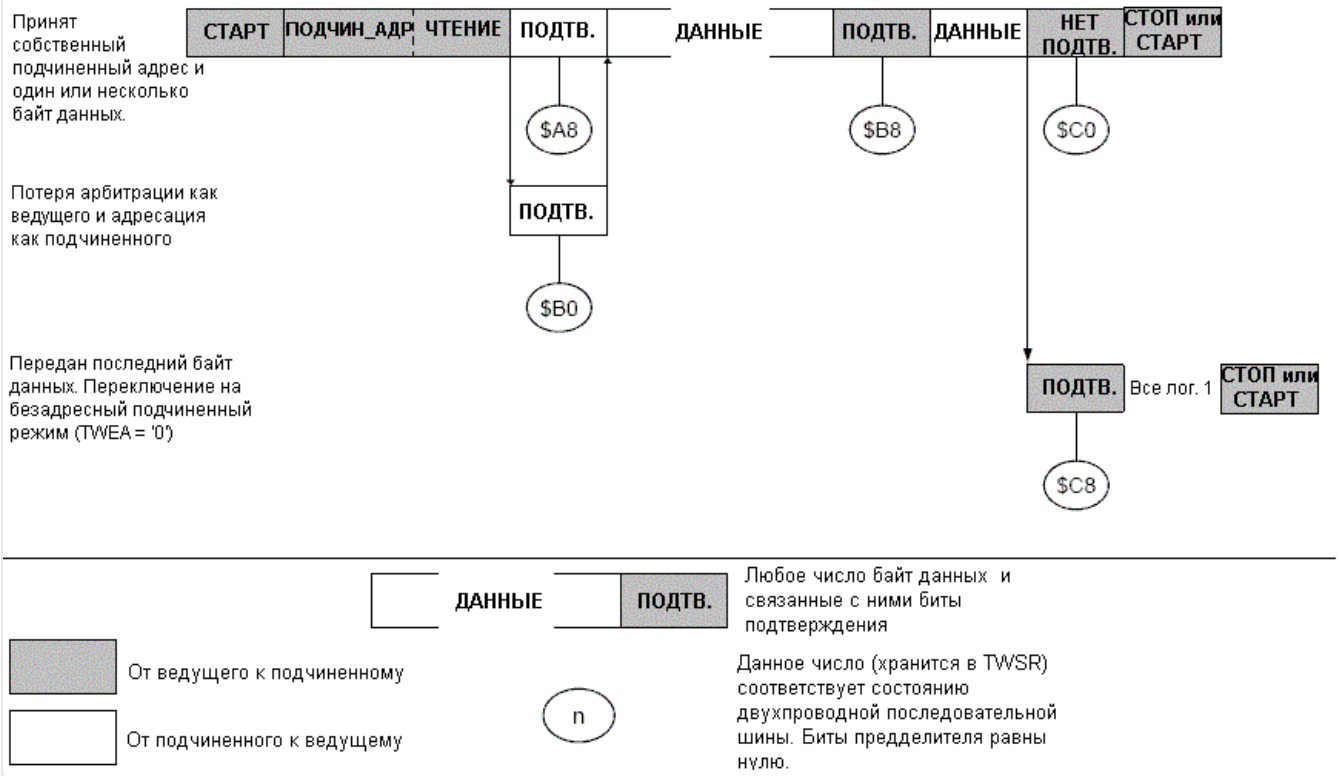
Ведущий приемник [Master Receiver (MR)]

Код состояния (TWSR), биты предделителя равны 0	Состояние двухпроводной последовательной шины и схемы двухпроводного последовательного интерфейса	Программные действия					Следующее действие, выполняемое схемой TWI
		В/из TWDR	В TWCR				
			STA	STO	TWINT	TWEA	
$08	Передано условие СТАРТ	Загрузка ПОДЧИН_АДР+ЧТЕНИЕ	0	0	1	x	Передается ПОДЧИН_АДР + ЧТЕНИЕ Принимается ПОДТВ. или НЕТ ПОДТВ.
$10	Передано условие ПОВТОРНЫЙ СТАРТ	Загрузка ПОДЧИН_АДР+ЧТЕНИЕ	0	0	1	X	Передается ПОДЧИН_АДР+ЧТЕНИЕ, принимается ПОДТВ. или НЕТ ПОДТВ.
		или ПОДЧИН_АДР+ЗАПИСЬ	0	0	1	X	Передается ПОДЧИН_АДР+ЗАПИСЬ, переключение на режим «Ведущий передатчик»
$38	Потеря арбитрации во время передачи бит ПОДЧИН_АДР + ЧТЕНИЕ или бита НЕТ ПОДТВ.	Действия без загрузки TWDR или	0	0	1	X	Шина освобождается и вводится безадресный подчиненный режим
		действия без загрузки TWDR	1	0	1	X	Условие СТАРТ передается после освобождения шины
$40	Передано ПОДЧИН_АДР+ЧТЕНИЕ и принято ПОДТВ	Действия без загрузки TWDR или	0	0	1	0	Принимается байт данных, возвращается бит НЕТ_ПОДТВ.
		действия без загрузки TWDR	0	0	1	1	Принимается байт данных, возвращается бит ПОДТВ.
$48	Передано ПОДЧИН_АДР+ЧТЕНИЕ и принято НЕТ ПОДТВ	Действия без загрузки TWDR или	1	0	1	X	Передается ПОВТОРНЫЙ СТАРТ
		действия без загрузки TWDR или	0	1	1	X	Передается условие СТОП и сбрасывается флаг TWSTO
		действия без загрузки TWDR	1	1	1	X	Вслед за условием СТАРТ передается условие СТОП и сбрасывается флаг TWSTO
$50	Принят байт данных и возвращается ПОДТВерждение	Чтение байта данных или	0	0	1	0	Принимается байт данных и возвращается НЕТ ПОДТВ
		чтение байта данных	0	0	1	1	Принимается байт данных и возвращается ПОДТВ
$58	Принят байт данных и возвращается НЕТ ПОДТВерждения	Чтение байта данных или	1	0	1	X	Передается ПОВТОРНЫЙ СТАРТ
		чтение байта данных или	0	1	1	X	Передается СТОП и сбрасывается флаг TWSTO
		чтение байта данных	1	1	1	X	Вслед за условием СТАРТ передается условие СТОП и сбрасывается флаг TWSTO



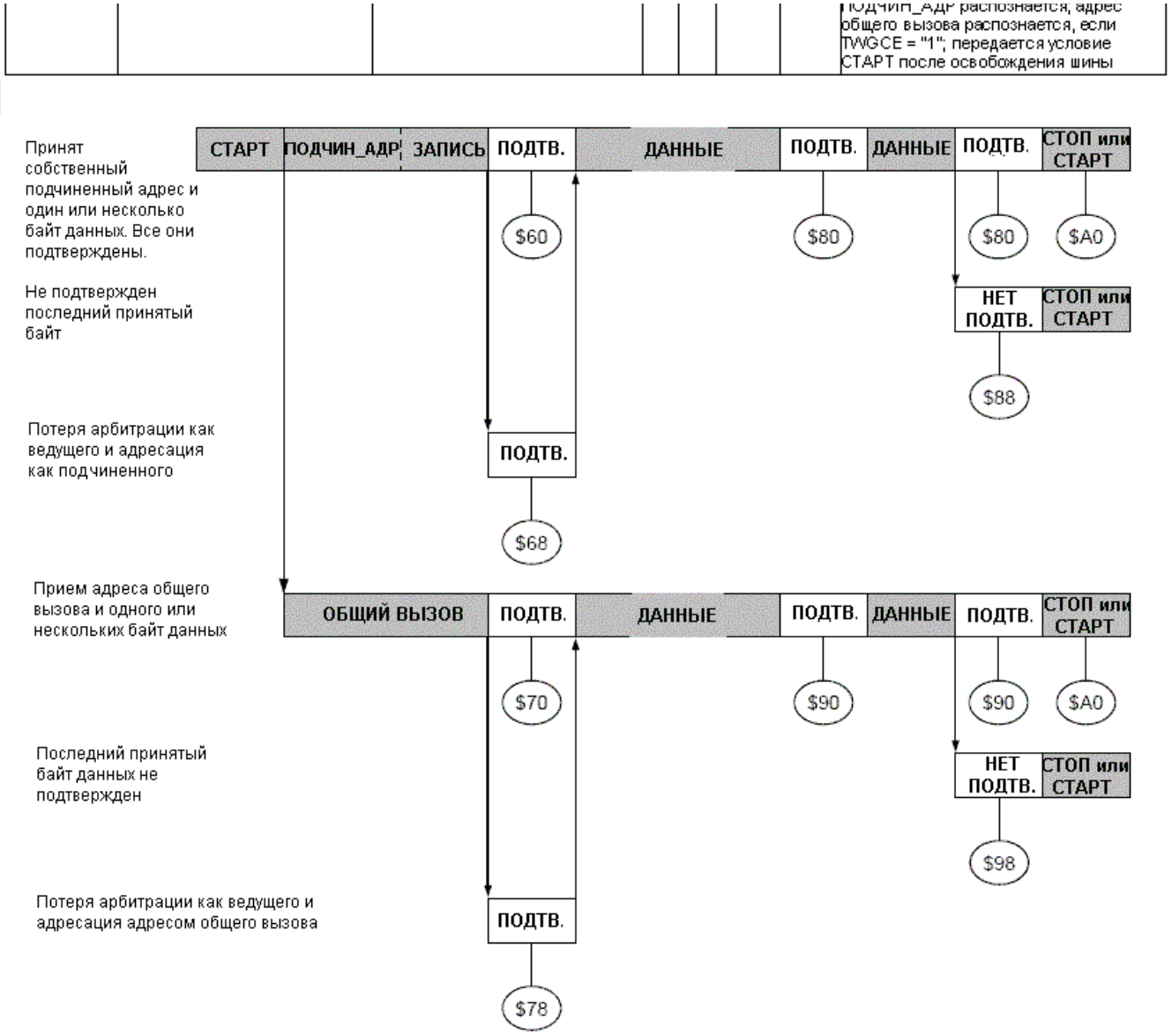
Ведомый передатчик [Slave Transmitter (ST)]

Код состояния (TWSR), биты предделителя равны 0	Состояние двухпроводной последовательной шины и схемы двухпроводного последовательного интерфейса	Программные действия					Следующее действие, выполняемое схемой TWI
		В/из TWDR	В TWCR				
			STA	STO	TWINT	TWEA	
$A8	Принимается собственный ПОДЧИН_АДР+ЧТЕНИЕ; возращено ПОДТВ.	Загрузка байта данных или	x	0	1	0	Передается последний байт данных и должно быть принято НЕТ ПОДТВ
		загрузка байта данных	x	0	1	1	Передается байт данных и должно быть принято ПОДТВ
$B0	Потеряна арбитрация во время передачи ПОДЧИН_АДР+ЧТЕНИЕ/ ЗАПИСЬ как ведущего, принят собственный ПОДЧИН_АДР+ЧТЕНИЕ, возвращено ПОДТВ	Загрузка байта данных или	x	0	1	0	Передается последний байт данных и должно быть принято НЕТ ПОДТВ
		загрузка байта данных	x	0	1	1	Передается байт данных и должно быть принято ПОДТВ
$B8	Передан байт данных из TWDR; принято ПОДТВерждение	Загрузка байта данных или	x	0	1	0	Передается последний байт данных и должно быть принято НЕТ ПОДТВ
		загрузка байта данных	x	0	1	1	Передается байт данных и должно быть принято ПОДТВ
$C0	Передан байт данных из TWDR; принято НЕТ ПОДТВерждения	Действия без загрузки TWDR или действия без загрузки TWDR или действия без загрузки TWDR или действия без загрузки TWDR	0	0	1	0	Переключение на безадресный подчиненный режим; не распознается собственный ПОДЧИН_АДР или адрес общего вызова
			0	0	1	1	Переключение на безадресный подчиненный режим; собственный ПОДЧИН_АДР распознается; адрес общего вызова распознается, если TWGCE = "1"
			1	0	1	0	Переключение на безадресный подчиненный режим; не распознается собственный ПОДЧИН_АДР или адрес общего вызова; передается условие СТАРТ после освобождения шины
			1	0	1	1	Переключение на безадресный подчиненный режим; собственный ПОДЧИН_АДР распознается; адрес общего вызова распознается, если TWGCE = "1"; передается условие СТАРТ после освобождения шины
$C8	Передан последний байт данных из TWDR (TWEA = "0"); принято ПОДТВерждение	Действия без загрузки TWDR или действия без загрузки TWDR или действия без загрузки TWDR или действия без загрузки TWDR	0	0	1	0	Переключение на безадресный подчиненный режим; не распознается собственный ПОДЧИН_АДР или адрес общего вызова
			0	0	1	1	Переключение на безадресный подчиненный режим; собственный ПОДЧИН_АДР распознается; адрес общего вызова распознается, если TWGCE = "1"
			1	0	1	0	Переключение на безадресный подчиненный режим; не распознается собственный ПОДЧИН_АДР или адрес общего вызова; передается условие СТАРТ после освобождения шины
			1	0	1	1	Переключение на безадресный подчиненный режим; собственный ПОДЧИН_АДР распознается; адрес общего вызова распознается, если TWGCE = "1"; передается условие СТАРТ после освобождения шины



Ведомый приемник [Slave Receiver (SR)]

Код состояния (TWSR), биты предделителя равны 0	Состояние двухпроводной последовательной шины и схемы двухпроводного последовательного интерфейса	Программные действия						Следующее действие, выполняемое схемой TWI
		В/из TWDR	В TWCR					
			STA	STO	TWINT	TWEA		
$60	Принимается собственный ПОДЧИН_АДР+ЗАПИСЬ; возращено ПОДТВ.	Действия без загрузки TWDR или	x	0	1	0		Принимается байт данных и возвращается НЕТ ПОДТВерждение
		действия без загрузки TWDR	x	0	1	1		Принимается байт данных и возвращается ПОДТВерждение
$68	Потеряна арбитрация во время передачи ПОДЧИН_АДР+ЧТЕНИЕ/ЗАПИСЬ как ведущего, принят собственный ПОДЧИН_АДР+ЗАПИСЬ, возвращено ПОДТВ	Действия без загрузки TWDR или	x	0	1	0		Принимается байт данных и возвращается НЕТ ПОДТВерждение
		действия без загрузки TWDR	x	0	1	1		Принимается байт данных и возвращается ПОДТВерждение
$70	Принят адрес общего вызова; возвращено подтверждение	Действия без загрузки TWDR или	x	0	1	0		Принимается байт данных и возвращается НЕТ ПОДТВерждение
		действия без загрузки TWDR	x	0	1	1		Принимается байт данных и возвращается ПОДТВерждение
$78	Потеряна арбитрация во время передачи ПОДЧИН_АДР+ЧТЕНИЕ/ЗАПИСЬ как ведущего, принят адрес общего вызова, возращено ПОДТВ	Действия без загрузки TWDR или	x	0	1	0		Принимается байт данных и возвращается НЕТ ПОДТВерждение
		действия без загрузки TWDR	x	0	1	1		Принимается байт данных и возвращается ПОДТВерждение
$80	Предварительная адресация собственным ПОДЧИН_АДР+ЗАПИСЬ; приняты данные; возвращено ПОДТВерждения	Чтение байта данных или	x	0	1	0		Принимается байт данных и возвращается НЕТ ПОДТВерждение
		чтение байта данных	x	0	1	1		Принимается байт данных и возвращается ПОДТВерждение
$88	Предварительная адресация собственным ПОДЧИН_АДР+ЗАПИСЬ; приняты данные; возвращено НЕТ ПОДТВерждения	Чтение байта данных или	0	0	1	0		Переключение на безадресный подчиненный режим; не распознается собственный ПОДЧИН_АДР или адрес общего вызова
		чтение байта данных или	0	0	1	1		Переключение на безадресный подчиненный режим; собственный ПОДЧИН_АДР распознается; адрес общего вызова распознается, если TWGCE = "1"
		чтение байта данных или	1	0	1	0		Переключение на безадресный подчиненный режим; не распознается собственный ПОДЧИН_АДР или адрес общего вызова; передается условие СТАРТ после освобождения шины
		чтение байта данных	1	0	1	1		Переключение на безадресный подчиненный режим; собственный ПОДЧИН_АДР распознается; адрес общего вызова распознается, если TWGCE = "1"; передается условие СТАРТ после освобождения шины
$90	Предварительная адресация адресом общего вызова; приняты данные; возвращено ПОДТВерждение	Чтение байта данных или	x	0	1	0		Принимается байт данных и возвращается НЕТ ПОДТВерждение
		чтение байта данных	x	0	1	1		Принимается байт данных и возвращается ПОДТВерждение
$98	Предварительная адресация адресом общего вызова; приняты данные; возвращено НЕТ ПОДТВерждения	Чтение байта данных или	0	0	1	0		Переключение на безадресный подчиненный режим; не распознается собственный ПОДЧИН_АДР или адрес общего вызова
		чтение байта данных или	0	0	1	1		Переключение на безадресный подчиненный режим; собственный ПОДЧИН_АДР распознается; адрес общего вызова распознается, если TWGCE = "1"
		чтение байта данных или	1	0	1	0		Переключение на безадресный подчиненный режим; не распознается собственный ПОДЧИН_АДР или адрес общего вызова; передается условие СТАРТ после освобождения шины
		чтение байта данных	1	0	1	1		Переключение на безадресный подчиненный режим; собственный ПОДЧИН_АДР распознается; адрес общего вызова распознается, если TWGCE = "1"; передается условие СТАРТ после освобождения шины
$A0	Принято условие СТОП или повторный СТАРТ во время подчиненной адресации	Нет действий	0	0	1	0		Переключение на безадресный подчиненный режим; не распознается собственный ПОДЧИН_АДР или адрес общего вызова
			0	0	1	1		Переключение на безадресный подчиненный режим; собственный ПОДЧИН_АДР распознается; адрес общего вызова распознается, если TWGCE = "1"
			1	0	1	0		Переключение на безадресный подчиненный режим; не распознается собственный ПОДЧИН_АДР или адрес общего вызова; передается условие СТАРТ после освобождения шины
			1	0	1	1		Переключение на безадресный подчиненный режим; собственный



Пример:

Напишем 2-а кода: для ведущего и ведомого. Ведущий будет каждую секунду отправлять ведомому поI2C интерфейсу число насчитанных секунд. Ведомый будет принимать это значение и выводить его на PORTA.

Ведущий (MASTER) :

```
1  #define F_CPU 1000000UL
2  #include <avr/io.h>
3  #include <util/delay.h>
4
5  //адрес ведомого с которым хотим пообщаться
6  #define RTC_ADDR 0b11111111
7  //макрос адреса + бит чтения
8  #define SLA_R   RTC_ADDR|0b00000001
9  //макрос адреса + бит записи
10 #define SLA_W   RTC_ADDR&0b11111110
11
```

```
// отправка команды СТАРТ
void I2C_StartCondition(void)
{
TWCR = (1<<TWINT)|(1<<TWSTA)|(1<<TWEN);
while (!(TWCR & (1<<TWINT)));//ожидание установки бита TWIN
}

// отправка СТОП
void I2C_StopCondition(void)
{
TWCR = (1<<TWINT)|(1<<TWSTO)|(1<<TWEN);
}

//отправка байта
void I2C_SendByte(unsigned char c)
{
TWDR = c;//загрузка значения в регистр данных
TWCR = (1<<TWINT)|(1<<TWEN);//начаало передачи байта данных
while (!(TWCR & (1<<TWINT)));//ожидание установки бита TWIN
}

//инициализация I2C как передатчика
void I2C_Init (void)
{
TWBR=0xFF;//скорость передачи
}

//отправка SLA_W + байт данных
void I2C_SendPocket (unsigned char value)
{
I2C_StartCondition(); // генерируем условие СТАРТ
I2C_SendByte(SLA_W); //оправляем адрес устройства+бит запись
I2C_SendByte(value);//отправляем байт данных
I2C_StopCondition();//генерируем условие СТОП
}

int main(void)
{
I2C_Init();//инициализация модуля
char i=0;//переменная для передачи
while (1)
{
I2C_SendPocket (i);
_delay_ms(1000);
i+=1;
}
}
```

Ведомый (Slave) :

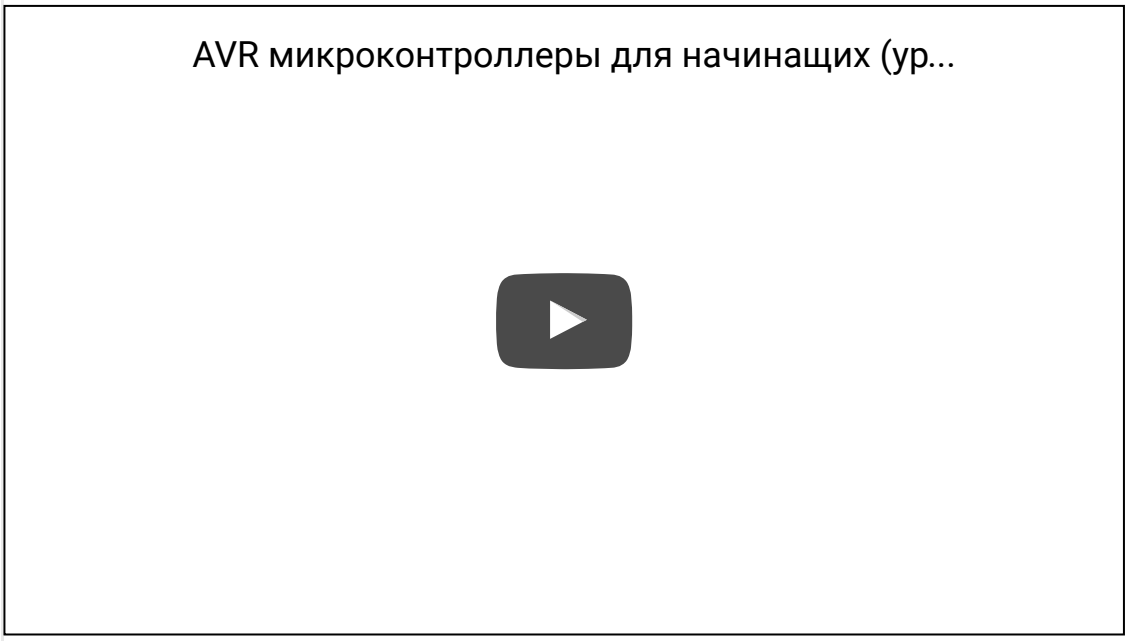
```
#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>

//адрес на который отвечает приемник
#define ME_ADDR 0b11111111

//чтение статуса
unsigned char I2C_GetStatus(void)
{
unsigned char status;//переменная хранения
status = TWSR & 0xF8;//маска
return status;
}

//инициализация I2C как приемника
void I2C_Init (void)
{
```

```
20 TWAR=ME_ADDR;// адрес для обращения
21 TWCR=(1<<TWEA)//разрешаем подтверждение
22 |(1<<TWEN)//включаем модуль TWI
23 |(1<<TWIE);//разрешаем прерывание
24 }
25
26 ISR(TWI_vect)
27 {
28 if(I2C_GetStatus()==0x80)//пришли данные от ведущего
29 PORTA=TWDR;//читаем значение
30 TWCR|=(1<<TWEN);//сбрасываем флаг
31 }
32
33 int main(void)
34 {
35 I2C_Init();//инициализация модуля
36 DDRA=0xFF;//порт на выход
37 sei();//разрешение глобальных прерываний
38 while(1);
39 }
```



Рубрика: Все посты AVR Уроки AVR

на авто в BG