

Лабораторная работа №9

Устройства I²C

Оглавление

Интерфейс I2C	2
Инициализация и завершение передачи данных	3
Адресация устройств на шине I ² C	3
Подтверждение приема	4
Режим конкуренции	8
Арбитраж	8
Синхронизация	9
Память FRAM	10
Системный супервизор FM31xx	11
Общее описание	11
Основные возможности	12
Описание регистров	14
Описание выводов	15
Адреса I2C	15
Взаимодействие с энергонезависимой памятью	16
Драйвер мастера I ² C для LPC2292	18
Инициализация	18
Установка адреса	18
Прием байта	18
Передача байта	19
Формирование стоп состояния	19
Литература	19

Интерфейс I2C

I2C - двухпроводный последовательный интерфейс, разработанный корпорацией Philips. В Первоначальном техническом требовании к интерфейсу максимальная скорость передачи данных составляла 100 Кбит/с. Однако позже появились стандартные более скоростные режимы работы шины I2C (400Кбит/с и 1 Мбит/с). К одной шине I2C могут быть подключены устройства с различными скоростями доступа, если скорость передачи данных будет удовлетворять требованиям самого низкоскоростного устройства.

Протокол передачи данных по шине I2C разработан таким образом, чтобы гарантировать надежный качественный прием/передачу данных. При передаче данных одно устройство является "Ведущим", которое инициирует передачу данных и формирует сигналы синхронизации. Другое устройство "Ведомое", которое может начать передачу данных только по команде ведущего шины. Модуль SSP микроконтроллеров PIC16CXXX полностью поддерживает режим ведомого I2C, за исключением поддержки адреса общего вызова (режим ведущего реализуется программно). Модуль MSSP аппаратно поддерживает режим ведущего/ведомого I2C. адрес общего вызова и скорость обмена данными до 1 Мбит/с. Скорость передачи данных 1 Мбит/с используют некоторые микросхемы последовательной EEPROM памяти. В таблице 1 представлены основные термины, связанные с шиной I2C.

Каждое устройство на шине I2C имеет уникальный адрес. Когда ведущий инициирует передачу данных, то сначала передается адрес устройства, к которому выполняется обращение. Остальные устройства проверяют переданный ведущим адрес. В состав байта адреса устройства входит бит направления передачи данных (выполняется чтение из ведомого или запись). Ведомый и ведущий шины всегда находятся в противоположном режиме работы, что можно представить в виде двух состояний:

- Ведущий передатчик - ведомый приемник.
- Ведомый передатчик - ведущий приемник.

В обоих случаях ведущий формирует тактовый сигнал.

Вывод тактового сигнала (SCL) и данных (SDA) должны иметь выход с открытым коллектором, чтобы выполнять требования "монтажного И" на шине. Для формирования высокого уровня сигнала на линиях к ним подключаются подтягивающие резисторы. Число устройств, которые могут быть подключены к шине I2C, ограничивается только максимальной емкостью шины (400пФ) и способностью адресации этих устройств.

Таблица 1 основные термины

Термин	Описание
Передатчик	Устройство, передающее данные по шине I ² C.
Приемник	Устройство, принимающее данные с шины I ² C.
Ведущий	Устройство, инициирующее передачу данных и формирующее тактовый сигнал.
Ведомый	Устройство, к которому обращается ведущий.
Конкуренция	Более чем один ведущий на шине. Несколько ведущих могут пытаться передать данные без разрушения текущего сообщения.
Арбитраж	Процедура, гарантирующая, что только один ведущий управляет шиной.
Синхронизация	Процедура синхронизации тактовых сигналов от двух или более устройств.

Инициализация и завершение передачи данных

В то время, когда передача данных на шине I²C отсутствует, сигнал синхронизации (SCL) и данных (SDA) имеют высокий логический уровень за счет подтягивающих резисторов. Биты START и STOP формируются ведущим для определения начала и окончания передачи данных соответственно. Бит START формируется переходом сигнала SDA из высокого уровня в низкий при высоком уровне сигнала SCL. Бит STOP формируется переходом SDA из низкого уровня в высокий при высоком уровне SCL. На рисунке 1 показано формирование битов START и STOP. Ведущий шины формирует биты START и STOP для указания начала и завершения передачи данных. При передаче данных сигнал SDA может изменяться только, когда SCL имеет низкий логический уровень.

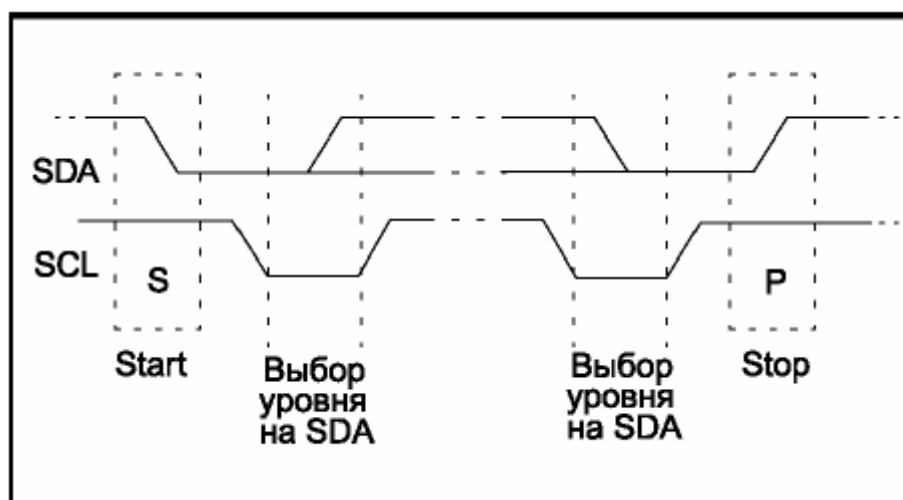


Рисунок 1 Состояния СТАРТ и СТОП

Адресация устройств на шине I²C

Для адресации устройств используется два формата адреса: простой 7-разрядный формат с битом чтения/записи R/W (см. рис. 2); 10-разрядный формат, передается два байта. В первом байте передается: пять битов, определяющих, что это 10-разрядный адрес; два старших бита адреса: бит записи/чтения. Во втором байте передается 8 младших бит адреса (см. рис. 3).



Рисунок 2 7 разрядная адресация

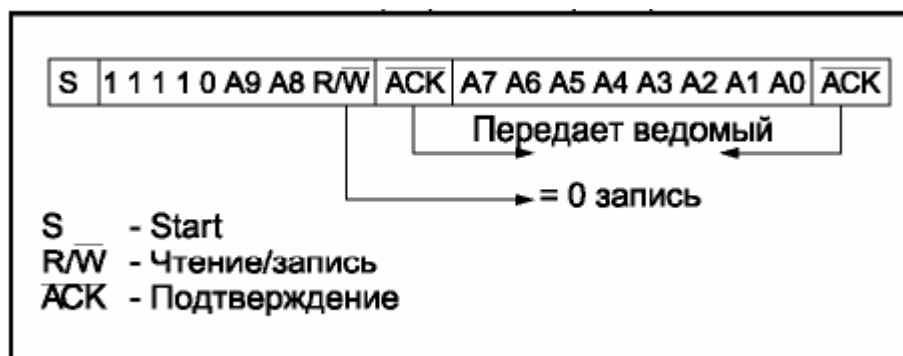


Рисунок 3 10 разрядная адресация

Подтверждение приема

При передаче данных после каждого переданного байта приемник должен подтвердить получения байта сигналом ACK. (см. рис. 4). Если ведомый не подтверждает получение байта адреса или данных, ведущий должен прервать передачу сформировав сигнал STOP (ведомый должен отпустить SDA для формирования STOP ведущим).

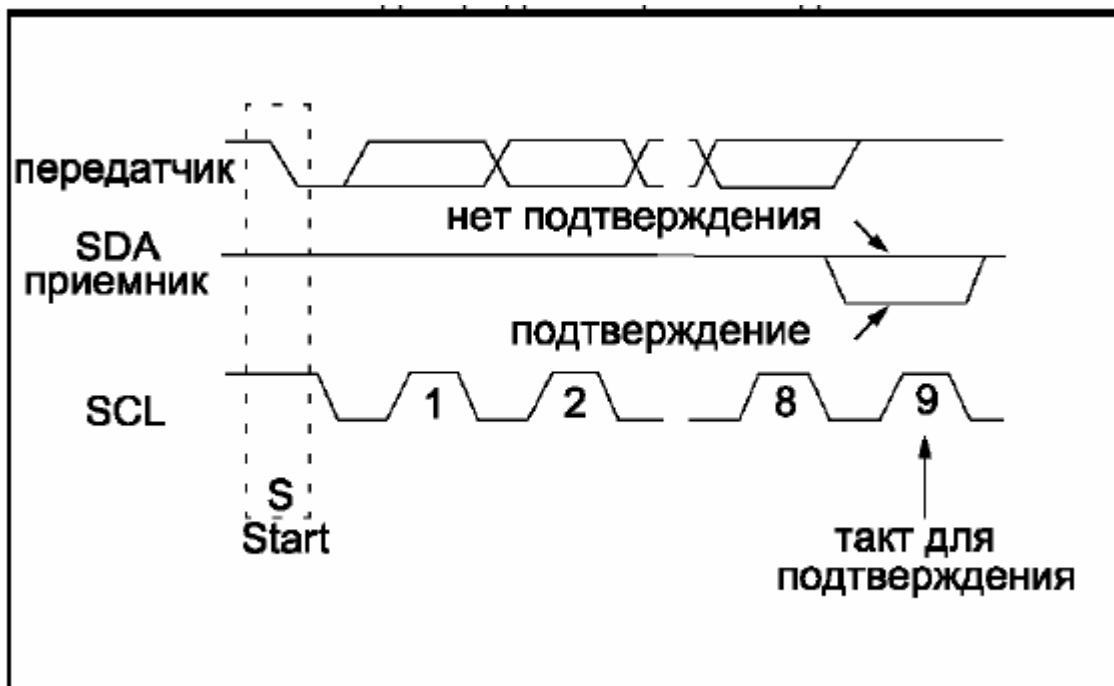


Рисунок 4 Подтверждение приема ведомым

Когда ведущий шины принимает данные, то на каждый принимаемый байт формируется бит подтверждения, если принятый байт не последний. Для сообщения ведомому о том, что ведущий прекращает принимать данные по приему последнего байта -ACK не формируется. Ведомый отпускает SDA, чтобы ведущий смог передать бит STOP. Ведущий может формировать бит STOP на месте бита подтверждения.

Если ведомому необходимо задержать передачу данных, то он может удерживать SCL в низком логическом уровне. Передача данных продолжится, когда ведомый отпустит SCL. Это позволяет ведомому подготовить новые данные для передачи. Методика задержки передачи данных может использоваться и при передаче отдельных битов (см. рис. А-5).



Рисунок 5 Ожидание передачи данных

На рисунках 6 и 7 показаны последовательности приема и передачи данных ведущим шины.

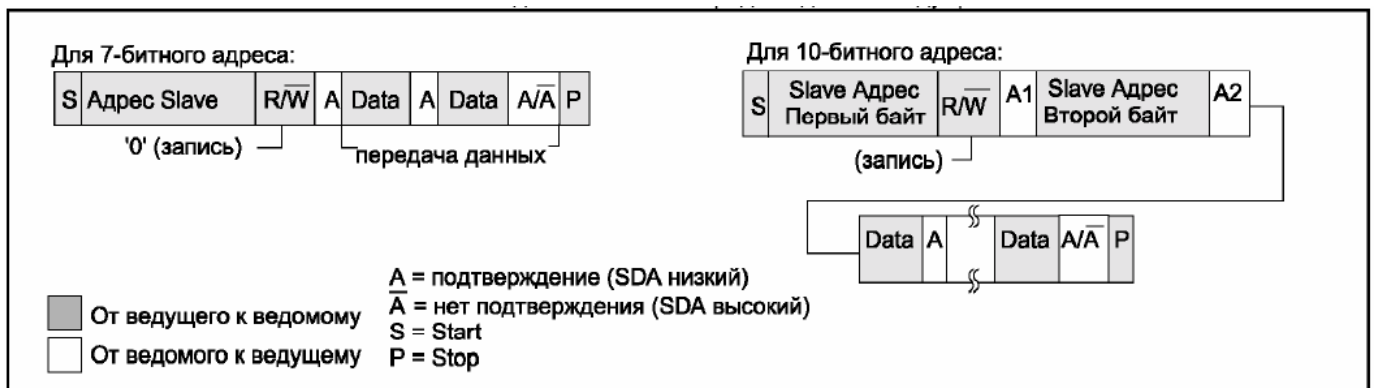


Рисунок 6 Последовательность передачи данных ведущим

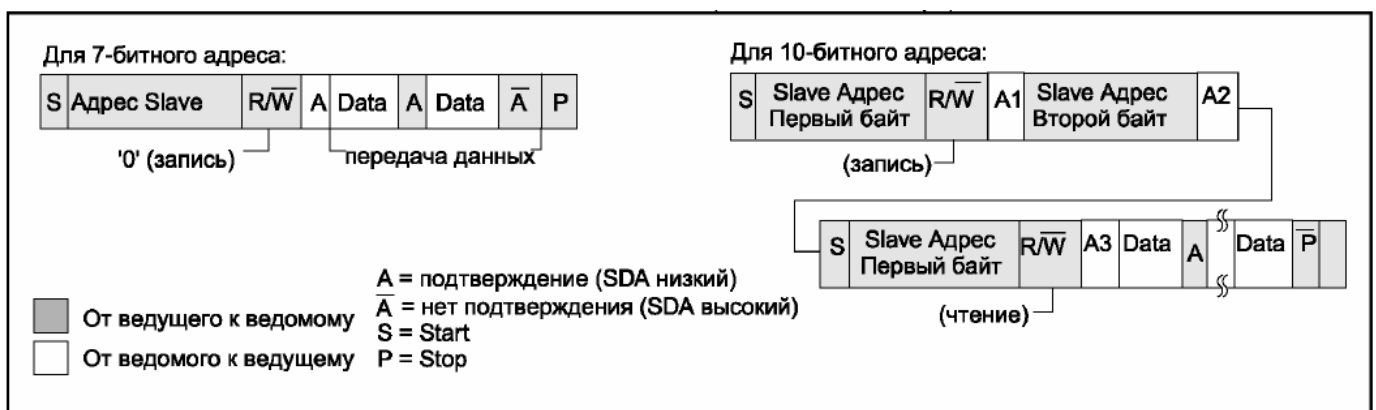


Рисунок 7 Последовательность приема данных ведомым

Когда ведущему шины необходимо продолжить обмен данными (при формировании бита STOP управление шиной теряется) может быть передан бит повторный START. Условие повторный START идентично условию START (на SDA формируется переход с высокого логического уровня в низкий при высоком уровне сигнала на SCL), но формируется после передачи бита подтверждения. Это позволяет ведущему продолжить обмен с текущим устройством или адресовать новое (см. рис. 8).

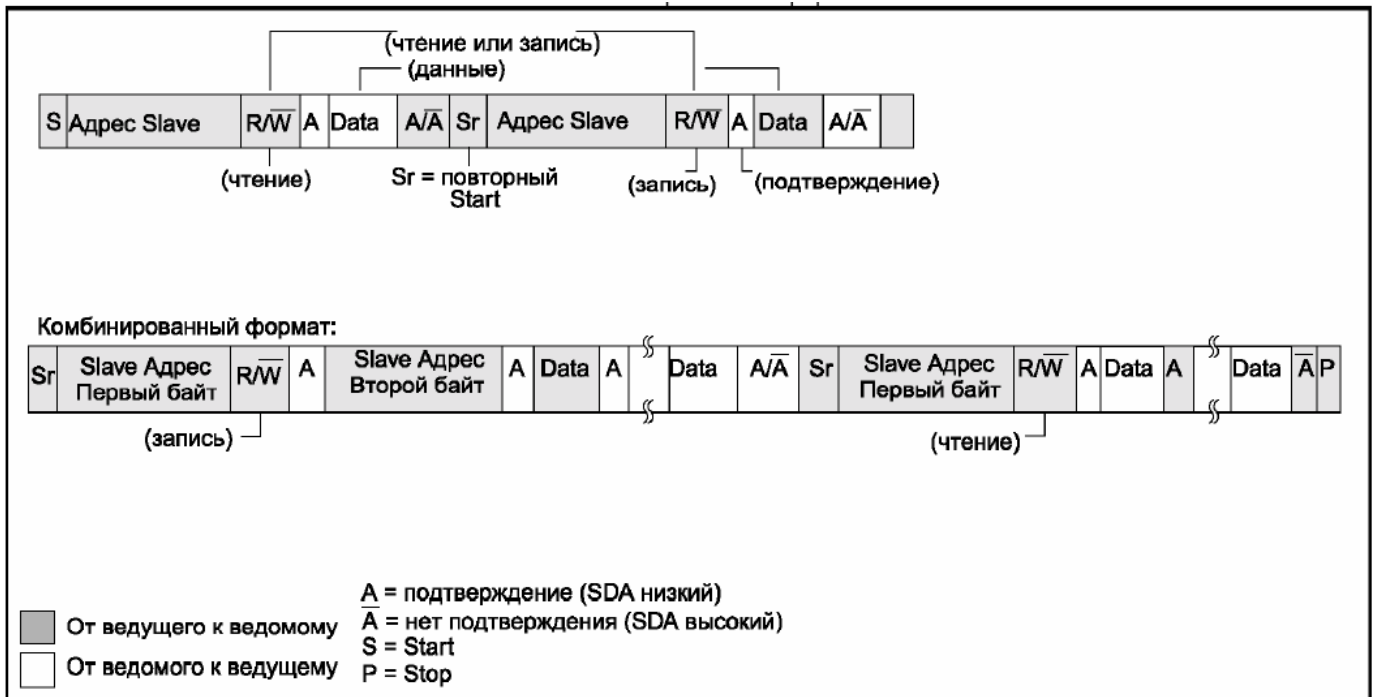


Рисунок 8 Комбинированный формат

Режим конкуренции

Протокол передачи данных I2C позволяет иметь более одного ведущего на шине. Для разрешения конфликтов на шине при инициализации передачи используются функции арбитража и синхронизации.

Арбитраж

Арбитраж выполняется на линии SDA при высоком уровне сигнала на SCL. Устройство, которое формирует на линии SDA высокий уровень, когда другое устройство передает низкий, теряет право быть ведущим и должно перейти в режим ведомого. Ведущий, потерявший инициативу на шине I2C, может формировать тактовые импульсы до конца байта, в котором потерял управление шиной. Когда несколько ведущих адресуют одно и то же устройство, то арбитраж выполняется при передаче данных.

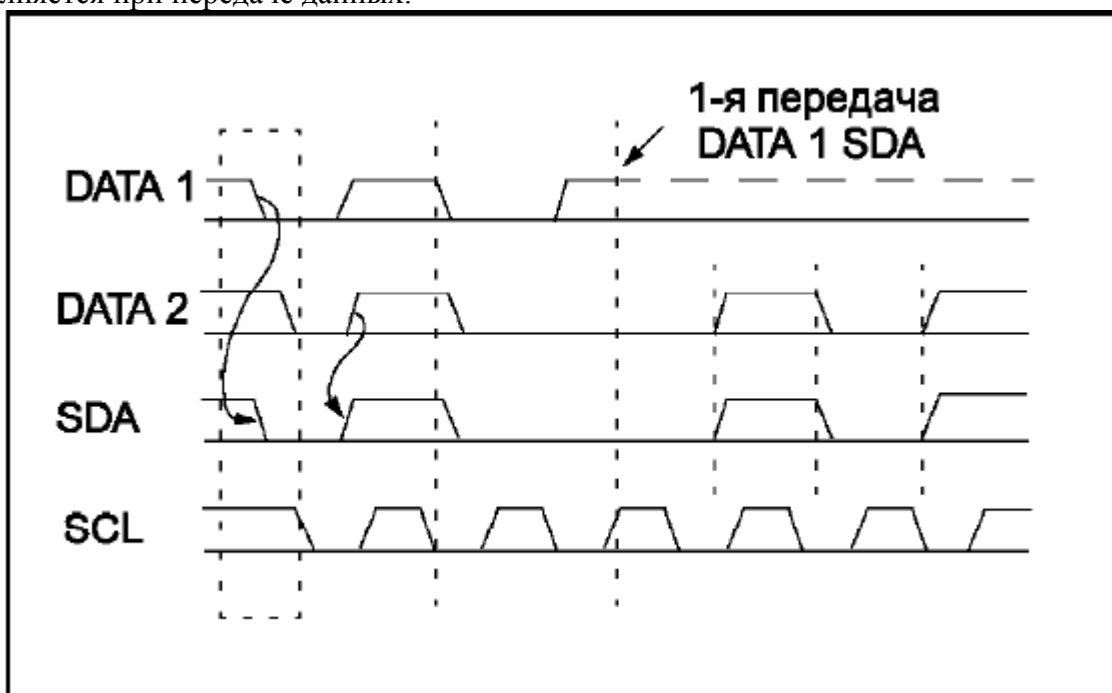


Рисунок 9 Арбитраж на шине I2C (два ведущих)

Ведущий, потерявший арбитраж, должен немедленно перейти в режим ведомого, поскольку он может быть адресован текущим ведущим.

Арбитраж не допускается между:

- Битами повторный START;
- Битом STOP и битом данных;
- Битами повторный START и STOP.

Ведущий шины должен гарантировать отсутствие указанных условий.

Синхронизация

Синхронизация тактового сигнала выполняется, когда устройства начинают арбитраж. Синхронизация реализуется за счет включения линии SCL по схеме "монтажное И". Переход сигнала на SCL с высокого логического уровня в низкий заставляет устройства, выполняющие арбитраж, начать отсчет длительности низкого логического уровня. После того, как тактовый сигнал устройства перешел в низкий уровень, оно будет удерживать этот уровень на SCL до тех пор, пока тактовый сигнал не перейдет в высокий уровень, но на SCL может быть по-прежнему низкий уровень, если другое устройство формирует низкий логический уровень. Низкий уровень на SCL удерживается устройством с минимальной частотой тактового сигнала передачи данных. Устройства с меньшей длительностью низкого уровня на SCL переходят в состояние ожидания, пока на SCL не появится высокий логический уровень сигнала. Затем все устройства начинают отсчет длительности высокого уровня сигнала. Устройство, с минимальной длительностью высокого уровня сигнала, первым переведет SCL в низкий уровень (см. рис. 10).

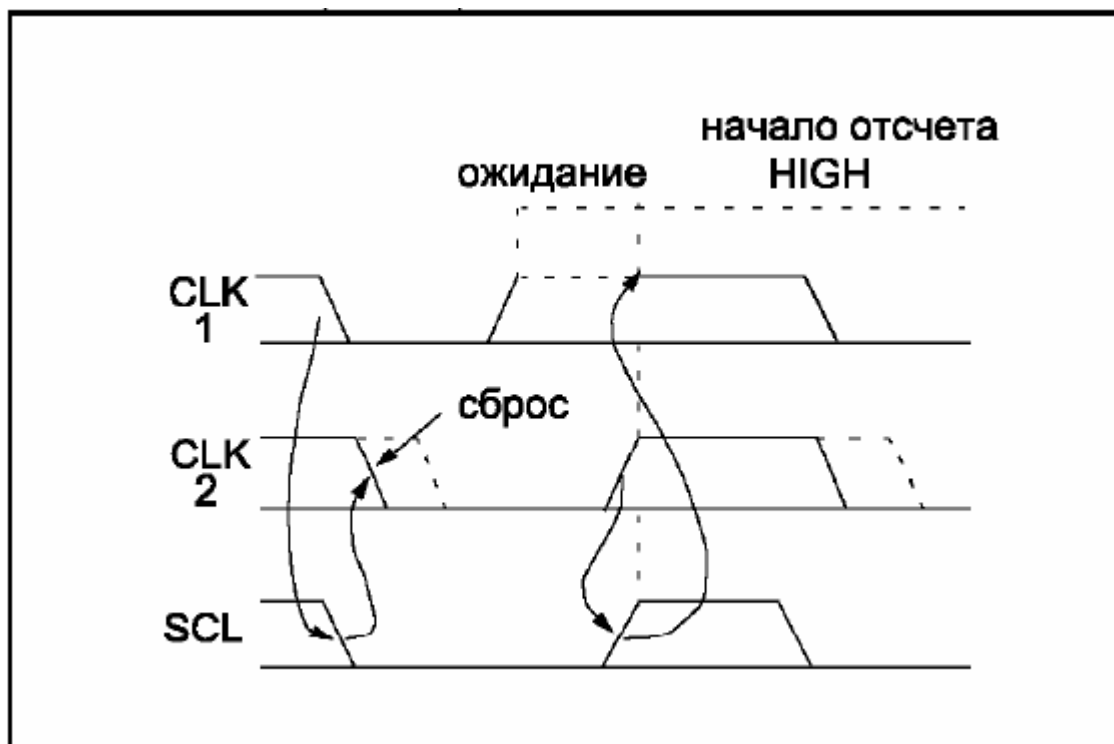


Рисунок 10 Синхронизация тактового сигнала на шине I2C

Память FRAM

Сегнетоэлектрическое ОЗУ (далее FRAM) обладает уникальными свойствами, которые отличают ее от других видов запоминающих устройств. Традиционные полупроводниковые запоминающие устройства можно разделить на две основные группы – энергозависимые и энергонезависимые. К энергозависимой памяти относятся статические оперативные запоминающие устройства (СОЗУ) и динамические оперативные запоминающие устройства (ДОЗУ). Их общим свойством является нарушение содержимого ячеек памяти после снятия напряжения питания. С прикладной точки зрения ОЗУ очень просты в использовании и обладают высоким быстродействием чтения и записи, но также имеют досадную особенность терять данные при исчезновении питания.

Энергонезависимая память (ЭНП) не теряет данных при снятии питания. Однако все основные типы ЭНП имеют общие истоки, которые берут свое начало от постоянных запоминающих устройств (ПЗУ). Тем, кто знаком с этой технологией знает насколько сложно осуществить запись информации в ПЗУ, а выполнить запись мгновенно вообще не возможно. Все последующие приемники этой технологии связаны проблемой сложности записи в них новой информации. В настоящее время известны следующие разновидности этой технологии: электрически перепрограммируемое ПЗУ - ЭППЗУ (морально устаревшая технология), электрически стираемое перепрограммируемое ПЗУ – ЭСПЗУ и флэш-память. Технологии на основе ПЗУ обладают медленной записью, подвержены существенному износу при записи, ограничивая количество циклов программирования, и требуют много энергии для программирования.

Отличием FRAM является использование технологии ОЗУ, при этом сохраняя энергонезависимость подобно ПЗУ. Таким образом, FRAM заполняет пробел между двумя категориями и создает нечто новое – энергонезависимое ОЗУ. Технология FRAM

Ядром сегнетоэлектрической FRAM-технологии от Ramtron являются сегнетоэлектрические кристаллы, которые позволяют законченным FRAM-изделиям работать подобно ОЗУ, при этом обеспечивая энергонезависимость хранения данных.

Когда электрическое поле прикладывается к сегнетоэлектрическому кристаллу, центральный атом движется в его направлении. Т.к. атом перемещается в пределах кристалла он проходит энергетический барьер, сопровождаемый спонтанной поляризацией. Внутренняя схема позволяет определить величину заряда и состояние памяти. Если электрическое поле отведено от кристалла, то центральный атом остается в том же положении, определяя состояние памяти. Поэтому, FRAM не нуждается в регенерации и после отключения питания сохраняет свое содержимое. Все происходит быстро и без износа!

FRAM-технология совместима со стандартной промышленной технологией КМОП. Сегнетоэлектрическая тонкая пленка размещена над основными КМОП слоями и сжата между двумя электродами. Технологический процесс сборки завершают металл для внешнего подключения и пассивация.

Технология FRAM от Ramtron имеет также историю развития. Первоначально, архитектура FRAM требовала два транзистора и два конденсатора (2Т/2С), что привело к относительно большим размерам ячейки памяти. Недавние улучшения сегнетоэлектрических материалов и технологии позволили избавиться от необходимости применения опорного конденсатора в каждой ячейки массива сегнетоэлектрической памяти. Новая одностранзисторная-одноконденсаторная архитектура от Ramtron работает подобно ДОЗУ, используя один конденсатор в качестве общего опорного конденсатора для каждого столбца массива памяти, тем самым позволив в два раза уменьшить требуемый размер ячейки по сравнению с архитектурой 2Т/2С. Новая архитектура существенно улучшает влияние кристалла и уменьшает производственную стоимость конечных изделий – микросхем FRAM-памяти.

Ramtron также стремится уменьшать шаг технологической сетки, чтобы снизить себестоимость ячеек FRAM памяти. Так недавний переход на 0.35мкм технологию позволил снизить

потребляемую мощность и увеличить размер матрицы по сравнению с предшествующими поколениями FRAM памяти, выполненных по 0.5 мкм технологии.

Ближайшей перспективой совершенствования архитектуры FRAM-памяти является также использование архитектуры 1Т/1С, но с размещением сегнетоэлектрического конденсатора над транзистором. Это будет способствовать дальнейшему уменьшению размеру ячеек памяти и переходу на шаг технологической сетки до 0.1мкм. Достижение этих результатов позволит в будущем преодолеть, пожалуй, единственный недостаток по сравнению с существующими популярными технологиями ЭНП – ограниченный размер памяти (до 32 кБ).

Все рассмотренные технологии FRAM-памяти находят применение во многих приложениях, с которыми люди сталкиваются ежедневно. FRAM содержится в большом количестве изделий и приложений по всему миру от офисных копиров и высокопроизводительных серверов до автомобильных бортовых самописцев (черных ящиков) и электронных развлекательных устройств.

Системный супервизор FM31xx

Общее описание

Системный супервизор (Processor Companion) фирмы Ramtron является многофункциональным устройством содержащим в себе следующие элементы:

- Схема сброса;
- Монитор питания;
- Часы реального времени;
- Счетчик событий;
- Энергонезависимую память FRAM;
- Сторожевой таймер;
- Уникальный серийный номер.

Системный супервизор взаимодействует с процессором через интерфейс I²C.

Устройство работает при питании от 2.7 до 5.5В. Представители семейства FM31xxx отличаются размером энергонезависимой памяти: 4 кб, 16 кб, 64 кб и 256 кб. Быстрота записи и неограниченное число циклов записи делают возможным использование данной памяти как расширенного ОЗУ или для энергонезависимого хранения данных. Данная память обладает большей степенью энергонезависимости, чем память с батарейным резервированием питания.

Схема сброса предназначена для формирования качественного сигнала RESET после включения питания, после нажатия кнопки RESET (JP1) или после выключения питания. Проблема состоит в том, что при старте контроллера после включения питания или при выключении питания возможны различные переходные процессы, могущие привести к некорректному исполнению программ или порче содержимого ОЗУ. Схема сброса обеспечивает формирование сигнала RESET на время, достаточное для окончания всех переходных процессов.

В устройстве есть возможность хранить 64-разрядный код (серийный номер) с возможностью блокировки записи для сохранения его неизменности.

Программируемый монитор питания позволяет вырабатывать сигнал RESET при понижении напряжения питания ниже установленного порога на 100 мс. Имеется флаг для индикации источника сброса.

Сторожевой таймер является счетчиком с программируемым периодом и сигналом сброса. При переполнении счетчика сторожевого таймера вырабатывается сигнал RESET. Интервал времени программируемый и может меняться в пределах от 100 мс до 3 сек.

Часы реального времени представляют информацию о времени и дате в двоично-десятичном формате. Часы могут непрерывно питаться от внешней батареи или конденсатора в качестве

резервного источника. Часы используют кварцевый резонатор 32768 Гц и имеют режим калибровки для программной настройки точности хода.

Энергонезависимая память FRAM является перепрограммируемым электрически стираемым постоянным запоминающим устройством. Объем памяти FRAM, установленной в стенде SDK-2.0, составляет от 512 байт до 256 Кбайт. Количество циклов записи и чтения не ограничено, время хранения информации – до 10 лет.

Счетчик событий позволяет считать импульсы с помощью двух 16-ти разрядных энергонезависимых счетчиков событий, подсчитывающих число нарастающих или спадающих фронтов на соответствующих входах.

Основные возможности

Высокая степень интеграции для замены нескольких устройств

- Последовательная энергонезависимая память
- Часы реального времени (ЧРВ)
- Формирование сигнала сброса при понижении напряжения
- Сторожевой таймер
- Упреждающая сигнализация о нарушении питания/NMI
- Два 16-разрядных счетчика событий
- Последовательное число с блокировкой записи

Сегнетоэлектрическое энергонезависимое ОЗУ

- Версии различной емкости памяти: 4 кб, 16 кб, 64 кб и 256 кб
- Неограниченное количество циклов чтение/запись
- 10 летний срок хранения информации
- Запись без задержки (NoDelay™)

Часы-календарь реального времени

- Ток потребления от резервного источника до 1 мкА
- Представление реального времени в диапазоне от секунд до столетия в двоично-десятичном формате

- Учет високосности до 2099 г.

- Использует стандартный кварцевый резонатор 32768 Гц (6 пф)

- Программная калибровка

- Поддержка батареи или конденсатора в качестве резервного источника

Функции совместной работы с процессором системы

- Формирование сигнала сброса с активным низким уровнем при снижении уровня VDD и переполнении сторожевого таймера

- Программируемый порог VDD для формирования сброса

- Фильтрация и защита от дребезга ручного сброса

- Программируемый сторожевой таймер

- Два счетчика событий фиксируют системные вторжения или другие события

- Компаратор для простоты генерации упреждающего сигнала о снижении

нестабильной точки питания -64-разр. программируемый последовательный код с защитой от записи

Быстродействующий двухпроводной последовательный интерфейс

- Максимальная тактовая частота последовательной шины до 1 МГц

- Поддержка тактовых частот предшествующих устройств 100 кГц и 400 кГц

- Выводы для задания адреса устройства позволяют использовать до 4 устройств на одной шине

- Управление ЧРВ, супервизором через двухпроводной интерфейс

Простота использования

- Работа при питании 2.7...5.5В
- Малогабаритный 14-выв. корпус SOIC
- Малый рабочий ток
- Рабочая температура -40°C...+85°C

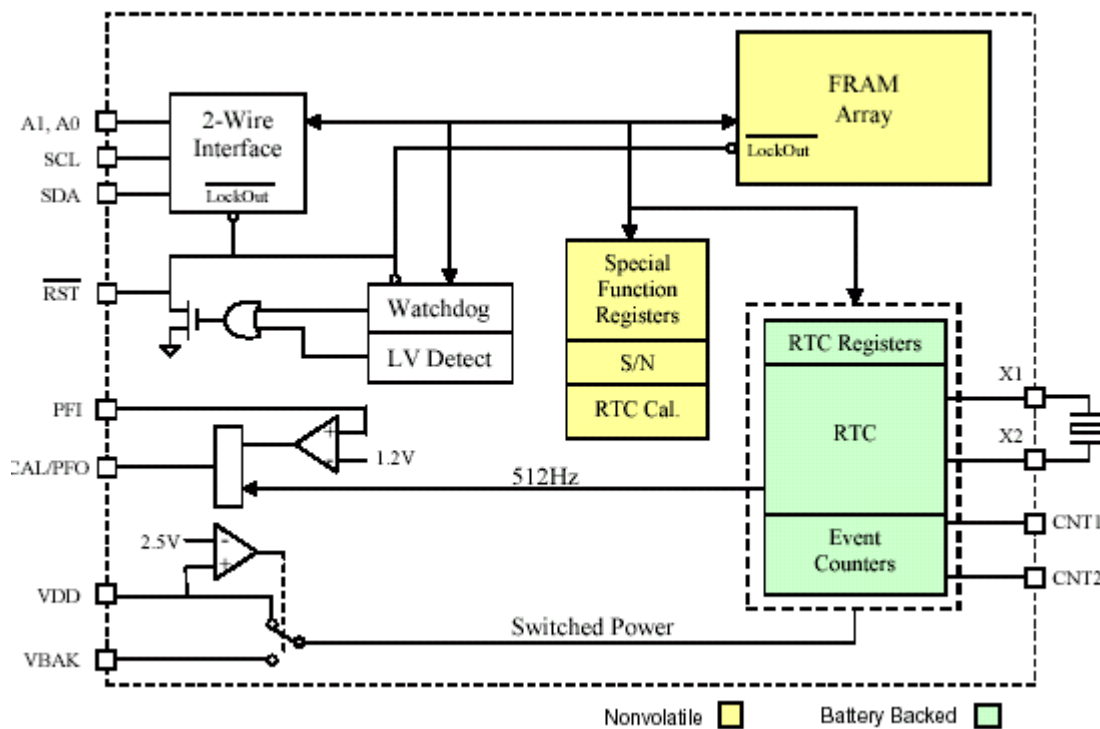


Рисунок 11 Структурная схема FM31xxx

Описание регистров

Все устройства FM31xxx доступны как набор 8-разрядных регистров.

Address	Data							
	D7	D6	D5	D4	D3	D2	D1	D0
18h	Serial Number Byte 7							
17h	Serial Number Byte 6							
16h	Serial Number Byte 5							
15h	Serial Number Byte 4							
14h	Serial Number Byte 3							
13h	Serial Number Byte 2							
12h	Serial Number Byte 1							
11h	Serial Number Byte 0							
10h	Counter 2 MSB							
0Fh	Counter 2 LSB							
0Eh	Counter 1 MSB							
0Dh	Counter 1 LSB							
0Ch					RC	CC	C2P	C1P
0Bh	SNL	-	-	WP1	WP0	VBC	VTP1	VTP0
0Ah	WDE	-	-	WDT4	WDT3	WDT2	WDT1	WDT0
09h	WTR	POR	LB	-	WR3	WR2	WR1	WR0
08h	10 years				years			
07h	0	0	0	10 mo	months			
06h	0	0	10 date		date			
05h	0	0	0	0	0	day		
04h	0	0	10 hours		hours			
03h	0	10 minutes			minutes			
02h	0	10 seconds			seconds			
01h	/OSCEN	reserved	CALS	CAL4	CAL3	CAL2	CAL1	CAL0
00h	reserved	CF	reserved	reserved	reserved	CAL	W	R

Рисунок 12 Карта памяти FM31xxx

Таблица 2 Значения по умолчанию

Адрес	Hex значение
18h	0x00
17h	0x00
16h	0x00
15h	0x00
14h	0x00
13h	0x00
12h	0x00
11h	0x00
0Bh	0x00
0Ah	0x1F
0h	0x80

Описание выводов

Таблица 3 Описание выводов

Наименование вывода	Описание
CNT1, CNT2	Входы счетчиков событий
A0, A1	Входы задания адреса устройства
CAL/PFO	Выход калибровки часов/предупреждения о нарушении питания
/RST	Вход/выход сброса
PFI	Вход компаратора упреждающего контроля понижения питания
X1, X2	Выводы подключения кварцевого резонатора
SDA	Ввод-вывод последовательных данных
SCL	Вход тактирования последовательной связи
VBAK	Резервное батарейное питание
VDD	Напряжение питания
VSS	Общий

Адреса I2C

С точки зрения интерфейса I2C, FM31xxx представлен как два устройства с различными адресами.

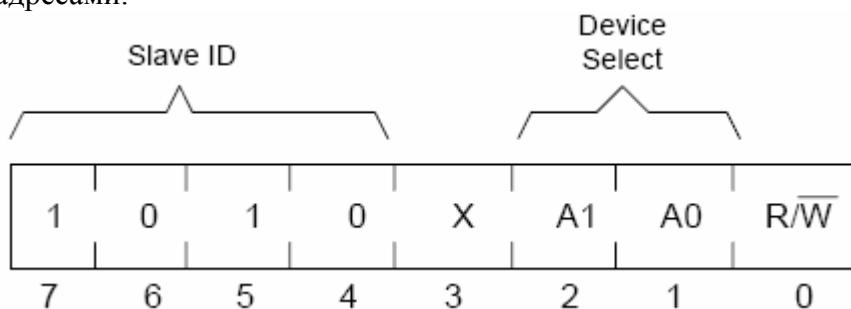


Рисунок 13 I2C адрес памяти энергонезависимой памяти

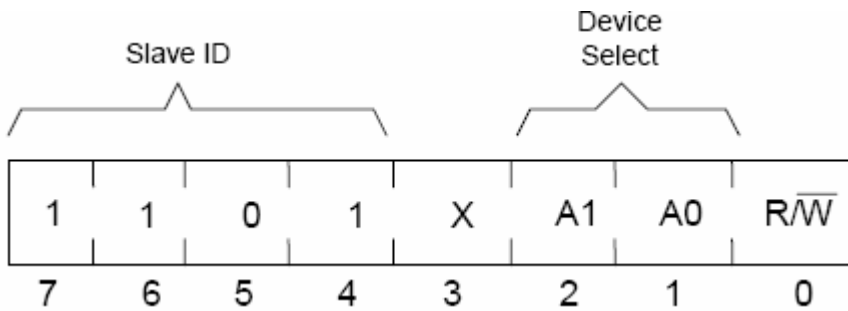


Рисунок 14 I2C адрес регистров

Взаимодействие с энергонезависимой памятью

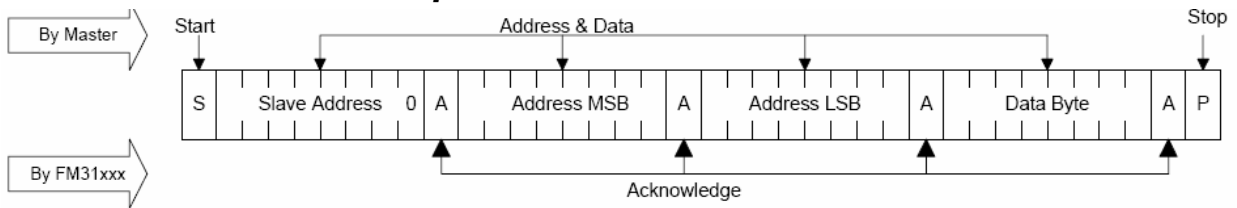


Рисунок 15 Запись одиночного байта в FRAM

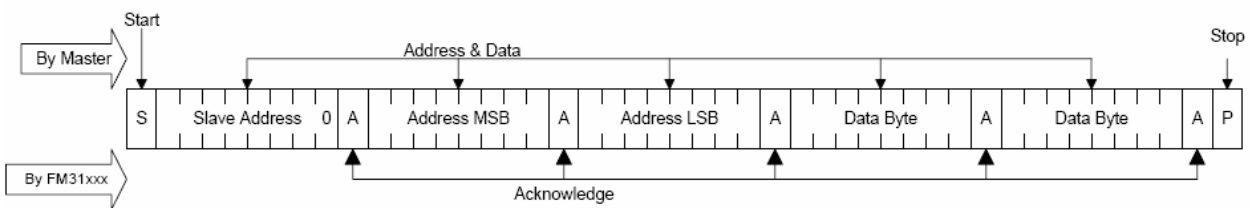


Рисунок 16 Запись нескольких байтов в FRAM

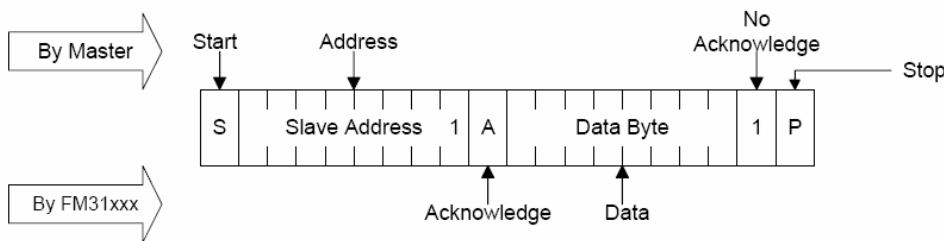


Рисунок 17 Чтение FRAM с текущего места

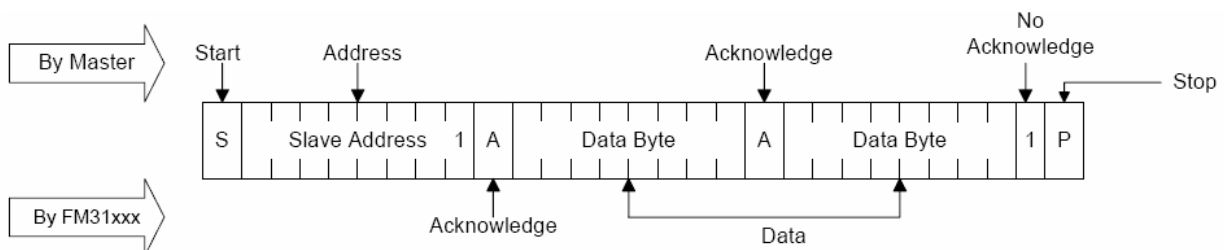


Рисунок 18 Чтение памяти с указанного адреса

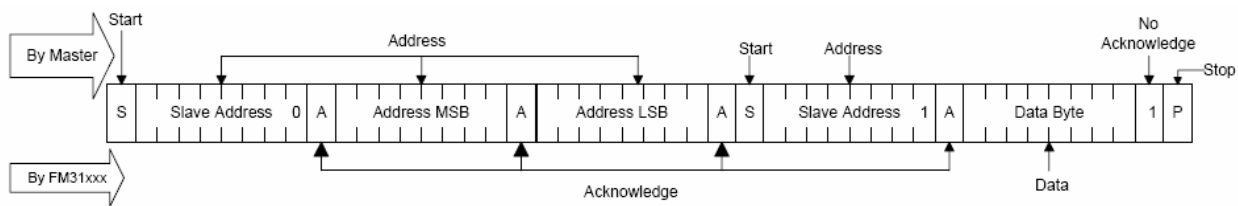


Рисунок 19 Случайный доступ к памяти

Тип	Первый байт адреса								Второй байт адреса						
	X	A14	A13	A12	A11	A10	A9	A8	A7	A5	A4	A3	A2	A1	A0
FM31256	X	A14	A13	A12	A11	A10	A9	A8	A7	A5	A4	A3	A2	A1	A0
FM3164	X	X	X	A12	A11	A10	A9	A8	A7	A5	A4	A3	A2	A1	A0
FM3116	X	X	X	X	X	A10	A9	A8	A7	A5	A4	A3	A2	A1	A0
FM3104	X	X	X	X	X	X	X	A8	A7	A5	A4	A3	A2	A1	A0

Рисунок 20 Двухбайтовый адрес FRAM

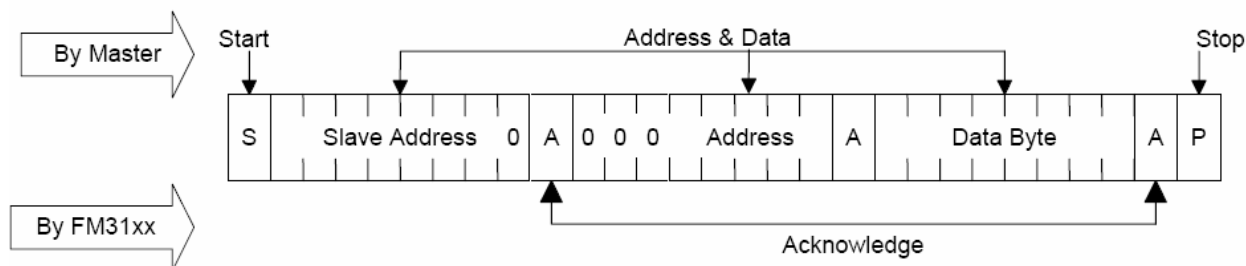


Рисунок 21 Запись байта в регистр

Драйвер мастера I²C для LPC2292

Инициализация

```
void init_i2c( void )
{
    I2CONCLR = 0xFF;

    I2SCLL=15;          //speed at 400 kHz for a VPB Clock Divider = 4 (12 MHz)
    I2SCLH=15;
    I2CONSET = 0x40;   //Active Master Mode on I2C bus
}
```

Установка адреса

```
int send_slave_addr(unsigned char addr)
{
    if( ! ( addr & 0x01 ) ) //test if it's a master address
    {
        I2CONSET = STA;

        while( I2STAT != 0x08 );    //Set and wait the start

        I2DAT      = addr; // Charge slave Address
        I2CONCLR = SIC | STAC; // Clear i2c interrupt bit to send the data

        while( I2STAT != 0x18 && !( I2CONSET & SI ) );    //wait the ACK
    }
    else //it's a slave address
    {
        I2CONSET = STA;
        I2CONCLR = SIC;

        while( I2STAT != 0x10 && !( I2CONSET & SI ) );
        //Set and wait the start

        I2DAT      = addr; // Charge slave Address
        I2CONCLR = SIC | STAC; // Clear i2c interrupt bit to send the data

        while( I2STAT != 0x40 && !( I2CONSET & SI));    //wait the ACK
    }

    return OK;
}
```

Прием байта

```
int read_i2c( unsigned char * data)
{
    int res = OK;

    I2CONCLR = SIC;          //ACK from the master
    while( I2STAT != 0x50 && !( I2CONSET & SI ) ); //wait the data

    // Read the data...
    *data = I2DAT;
    return res;
}
```

Передача байта

```
int write_i2c(unsigned char data)
{
    I2DAT    = data; // Charge Data
    I2CONCLR = SIC; // Clear i2c interrupt bit to send the data

    while( I2STAT !=0x28 && !( I2CONSET & SI ) ); //wait the ACK

    return OK;
}
```

Формирование стоп состояния

```
int stop_i2c( void )
{
    volatile unsigned long i;

    I2CONCLR = SIC;
    I2CONSET = STO;
    while( ( I2CONSET & STO ) );
    for( i = 0; i < 10000; i++);
    return OK;
}
```

Литература

1. Учебный стенд SDK-2.0. Инструкция по эксплуатации
2. LPC2119/2129/2194/2292/2294 USER MANUAL. Philips Semiconductors.
3. http://www.itt-ltd.com/reference/ref_i2c.html
4. <http://www.gaw.ru/html.cgi/txt/interface/iic/index.htm>
5. <http://www.microchip.ru/lit/?mid=1x0>