

СОДЕРЖАНИЕ

	Лист
1 Подключение микроконтроллера	2
2 Перевод микроконтроллера в режим последовательного программирования	2
3 Команды управления программированием	3
3.1 Сброс указателя памяти программ	5
3.2 Инкремент адреса.....	5
3.3 Загрузка адреса.....	5
3.4 Чтение адреса.....	6
3.5 Загрузка данных.....	6
3.6 Чтение данных.....	7
3.7 Старт программирования.....	7
3.8 Запись делителя частоты.....	8
3.9 Стирание сектора.....	8
3.10 Тест сектора на чистоту.....	9
3.11 Функция ЧТЕНИЕ ОЗУ+ПЕРЕФЕРИЯ.....	10
3.12 Функция ЗАПИСЬ ОЗУ+ПЕРЕФЕРИЯ.....	10
3.13 Функция ЗАПРОС ГОТОВНОСТИ.....	10
3.14 Функция ПЕРЕХОД CALL.....	10
3.15 Функция ПЕРЕХОД GOTO.....	10
3.16 Функция ВЕРСИЯ МИКРОКОНТРОЛЛЕРА.....	10
3.17 Функция СТЕРЕТЬ ВСЮ ПАМЯТЬ.....	11
4 Механизм восстановления синхронизации	11
5 Специальные библиотечные функции ПЗУ	11
6 Электрические характеристики для последовательного режима программирования	12

Справка №	Перв. Примен. ТСКЯ.431295.007
-----------	----------------------------------

Изм.	Лист	№ докум.	Подп.	Дата
------	------	----------	-------	------

Изм.	Лист	№ докум.	Подп.	Дата
Разраб.	Филатова			
Пров.	Шумилин			
Гл.констр	Какоулин			
Н.Контр.	Ануфриева			
Утв..	Какоулин			

Согласовано 4400 ВП МО РФ И.А.Фронтов

ТСКЯ.431295.007И

Микросхема
1886VE7У, 1886VE71У
Инструкция
по программированию

Лит	Лист	Листов
А	1	13

1 Подключение микроконтроллера

Для программирования внутренней EEPROM памяти программ микроконтроллеров 1886BE7У, 1886BE71У используется последовательный интерфейс ISP (Interface Serial Program). В этом режиме программирования задействованы выводы микроконтроллера и напряжения питания, приведённые в таблице 1.

Т а б л и ц а 1 – Выводы ISP интерфейса

Обозначение	В режиме программирования		
	Назначение	Тип	Описание
PA2	DT	вход/выход	Последовательные данные (18)
PA3	CK	вход	Последовательный синхросигнал (19)
PA1	CLK	вход	Источник синхронизации микроконтроллера (15)
TEST	TEST	вход	Вход для выбора тестового режима (9)
nMCLR	MCLRn	питание	Внешний сброс (10)
U _{cc}	U _{cc}	питание	Напряжение питания (11, 24)
GND	GND	общий	Общий (4, 8, 12)

2 Перевод микроконтроллера в режим последовательного программирования

Перевод микроконтроллера в режим последовательного программирования (ISP) осуществляется подачей напряжения питания U_{cc} на выводы TEST и MCLRn в следующей последовательности:

- 1 Подача U_{cc} на вывод TEST.
- 2 Подача U_{cc} на вывод nMCLR.

Время установки между событиями 1 и 2 должно быть не менее 1 микросекунды. В итоге счётчик команд будет указывать на адрес памяти программ 0xF000. В этой области памяти программ расположено загрузочное ПЗУ (Boot ROM). Этот программный код инициализирует USART/SCI для получения команд. Для выполнения программы из загрузочного ПЗУ необходимо обеспечить подачу синхросигнала на вывод PA1/TOCLK. Для того, чтобы USART/SCI был инициализирован для получения команд, необходимо выполнить следующую последовательность действий:

- 1 Запустить источник синхронизации микроконтроллера.
- 2 Ожидать 160 циклов микроконтроллера для инициализации USART/SCI загрузочным ПЗУ.
- 3 Послать команду по USART/SCI.

Инд. №	Подп. и дата	Взам. инв	Инд. №	Подп. и дата

Инд. №	Подп. и дата	Взам. инв	Инд. №	Подп. и дата
Из Лист	№ докум.	Подп.	Дата	

ТСКЯ.431295.007И

Лист

2

3 Команды управления программированием и специальные библиотечные функции

В ISP режиме модуль USART/SCI сконфигурирован как синхронное ведомое устройство. Микроконтроллер ожидает команду, которая выполняется загрузочным ПЗУ, расположенным в области программной памяти 0xF000-0xF7FF. Программно-аппаратные средства ISP поддерживают 20 команд, приведённых в таблице 2. Кроме того в ПЗУ расположены библиотечные функции приведенные в таблице 3.

Таблица 2 – Команды ISP

	Команда	Значение
1	СБРОС УКАЗАТЕЛЯ ПАМЯТИ ПРОГРАММ	0000 0000
2	ЗАГРУЗКА ДАННЫХ	0000 0010
3	ЧТЕНИЕ ДАННЫХ	0000 0100
4	ИНКРЕМЕНТ АДРЕСА	0000 0110
5	СТАРТ ПРОГРАММИРОВАНИЯ	0000 1000
6	ЗАГРУЗКА АДРЕСА	0000 1010
7	ЧТЕНИЕ АДРЕСА	0000 1100
8	СТОП ПРОГРАММИРОВАНИЯ	0000 1110
9	ЗАПИСЬ ДЕЛИТЕЛЯ ЧАСТОТЫ	0000 0001
10	Зарезервировано	0000 0011
11	Зарезервировано	0000 0111
12	СТИРАНИЕ СЕКТОРА	0000 1111
13	ТЕСТ СЕКТОРА НА ЧИСТОТУ	0000 0101
Расширение стандартного набора команд		
14	ЧТЕНИЕ ОЗУ+ПЕРЕФЕРИЯ	0001 0000
15	ЗАПИСЬ ОЗУ+ПЕРЕФЕРИЯ	0001 0001
16	ЗАПРОС ГОТОВНОСТИ	0001 0010
17	ПЕРЕХОД CALL	0001 0011
18	ПЕРЕХОД GOTO	0001 0100
19	ВЕРСИЯ МИКРОКОНТРОЛЛЕРА	0001 0101
20	СТЕРЕТЬ ВСЮ ПАМЯТЬ	0001 0110

Инв. №	Подп. и дата
Взам. инв	Инв. №
Подп. и дата	Подп. и дата
Инв. №	Инв. №

Из	Лист	№ докум.	Подп.	Дата
----	------	----------	-------	------

ТСКЯ.431295.007И

Лист

3

Т а б л и ц а 3 – Библиотечные функции ПЗУ

№	Название	Параметры	Описание
1	Стирание блока		Очистка содержимого памяти программ
2	Запись блока	(16'bDATA)	Запись памяти программ одним числом за один цикл
3	Проверка содержимого блока	(16'bDATA)	Проверка программ на равенство одному числу.
4	Стирание строки	(address)	Очистка содержимого строки памяти программ по адресу
5	Проверка содержимого строки	(address, 16'bDATA)	Проверка содержимого строки памяти программ на равенство одному числу.
6	Запись слова (через регистры)	(address, 16'bDATA)	Программирование ячейки памяти программ через управление регистрами блока.
7	Чтение слова (через регистры)	(address)	Чтение ячейки памяти программ через управление регистрами блока.
8	Вычисление контрольной суммы EEPROM	Стартовый адрес, длина, начальное значение, результат.	Вычисляет контрольную сумму всей памяти EEPROM.
9	Тестирование ядра		
10	Тестирование ОЗУ		
11	Тестирование EEPROM ПД		
12	Тестирование порта A		
13	Тестирование Timer0		
14	Тестирование LIN		
15	Тестирование USART_async		
16	Тестирование USART_sync_ms		
17	Тестирование USART_sync_sl		

Инв. №	Подп. и дата	Взам. инв	Инв. №	Подп. и дата
Из Лист	№ докум.	Подп.	Дата	

ТСКЯ.431295.007И

Лист

4

3.1 Сброс указателя памяти программ

Эта команда предназначена для очистки указателя адреса памяти программ. После этого указатель находится в исходном состоянии и указывает на адрес 0x0000. Временная диаграмма команды приведена на рисунке 1.

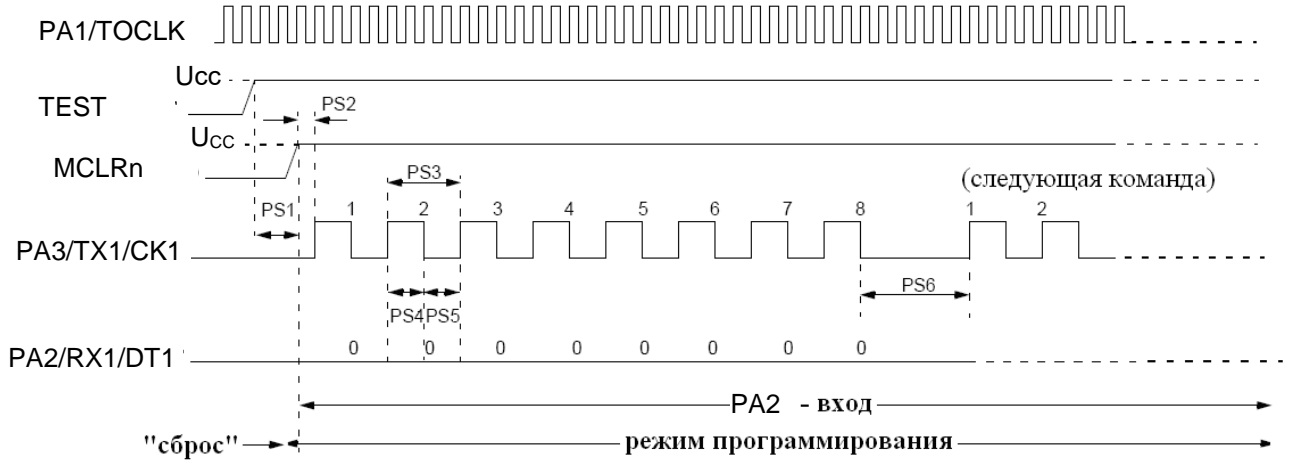


Рисунок 1 – Команда сброса указателя адреса памяти программ

3.2 Инкремент адреса

Эта команда инкрементирует указатель адреса памяти программ. Обычно эта команда следует за командами чтения или программирования. Временная диаграмма команды приведена на рисунке 2.

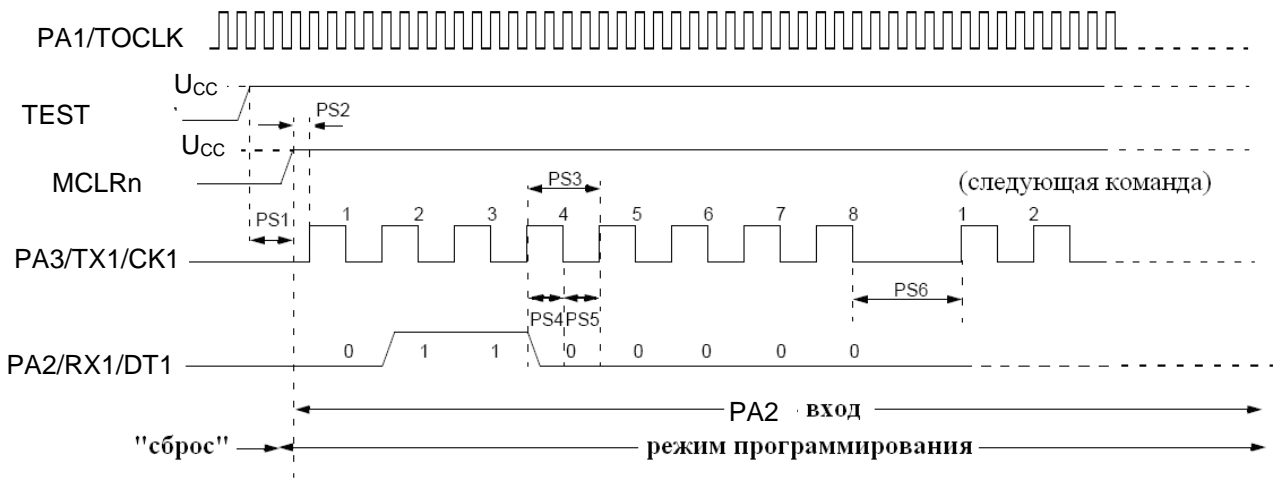


Рисунок 2 – Команда инкремента адреса

3.3 Загрузка адреса

Эта команда используется для загрузки указателя адреса программ определённым 16 разрядным значением. Обычно эта команда применяется для доступа к определённому адресному пространству. Временная диаграмма команды приведена на рисунке 3.

Инв. №	Подп. и дата
Взам. инв	Инв. №
Подп. и дата	Подп. и дата
Инв. №	Подп. и дата

Из	Лист	№ докум.	Подп.	Дата
----	------	----------	-------	------

ТСКЯ.431295.007И

Лист
5

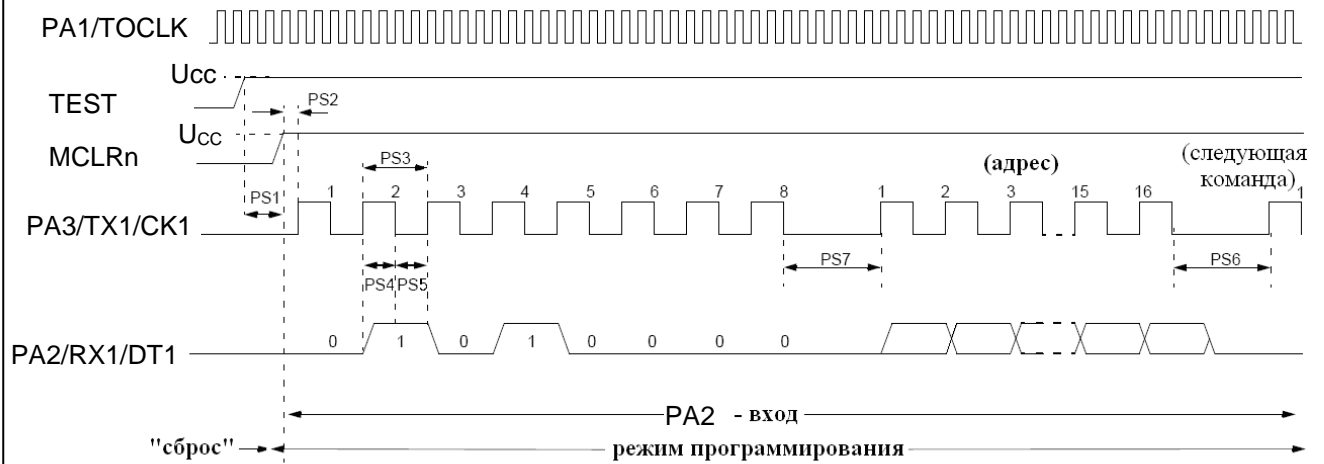


Рисунок 3 – Команда загрузки адреса памяти программ

3.4 Чтение адреса

Эта команда используется для получения текущего указателя адреса памяти программ. Временная диаграмма команды приведена на рисунке 4.

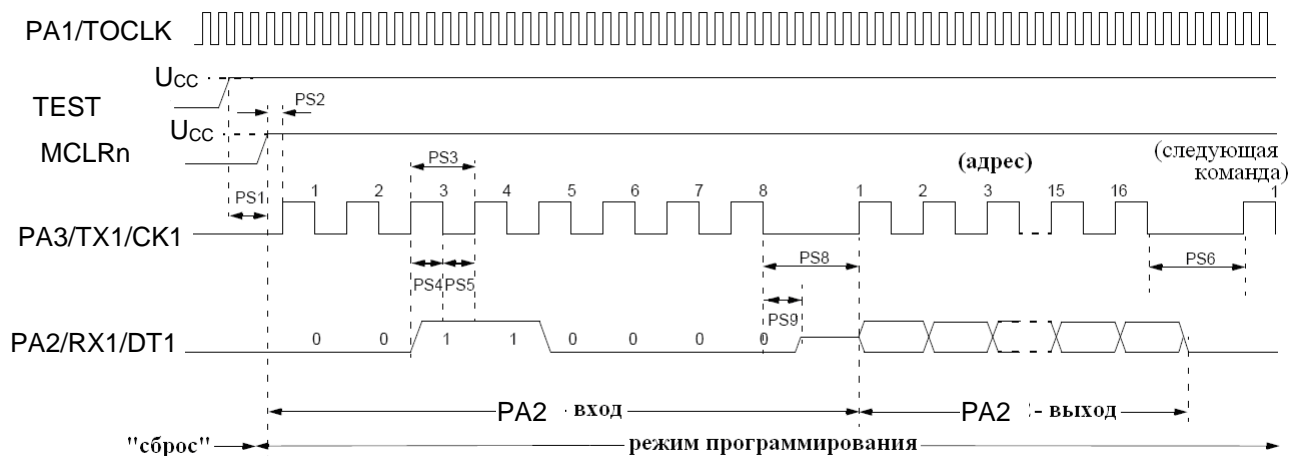


Рисунок 4 – Команда чтения адреса памяти программ

3.5 Загрузка данных

Эта команда загружает 16 разрядные данные, которые должны быть запрограммированы в память программ. Адрес памяти программ может быть модифицирован после загрузки данных. Данные не будут запрограммированы, пока не выполнится команда СТАРТ ПРОГРАММИРОВАНИЯ. Временная диаграмма команды приведена на рисунке 5.

Инв. №	Подп. и дата
Взам. инв	Инв. №
Подп. и дата	Подп. и дата
Инв. №	Инв. №

Из	Лист	№ докум.	Подп.	Дата
----	------	----------	-------	------

ТСКЯ.431295.007И

Лист
6

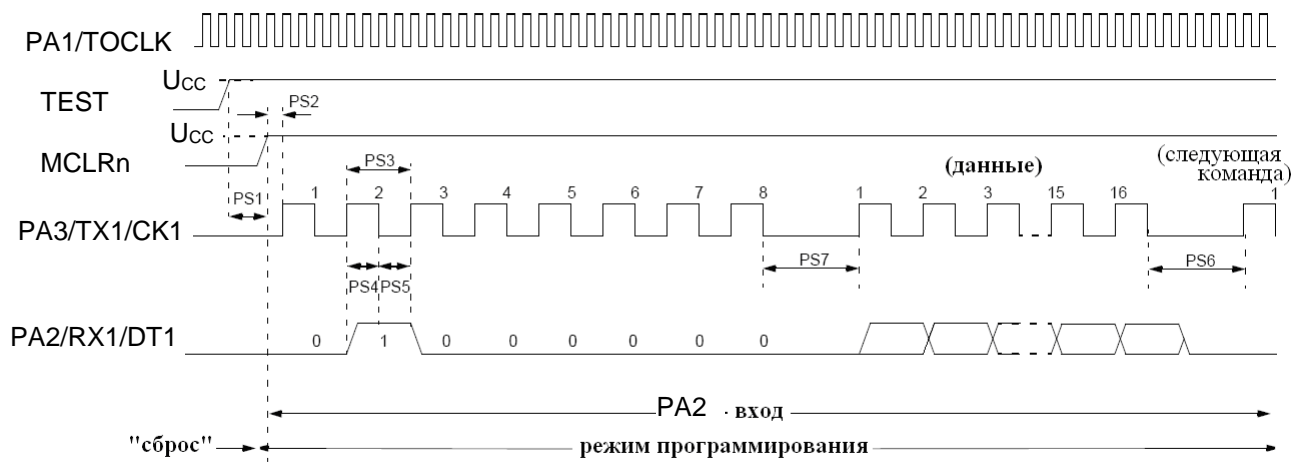


Рисунок 5 – Команда загрузки данных

3.6 Чтение данных

Эта команда читает данные, расположенные по адресу на который ссылается указатель адреса памяти программ. Обычно эта процедура используется в процедуре контроля правильности запрограммированных данных. Временная диаграмма команды приведена на рисунке 6.

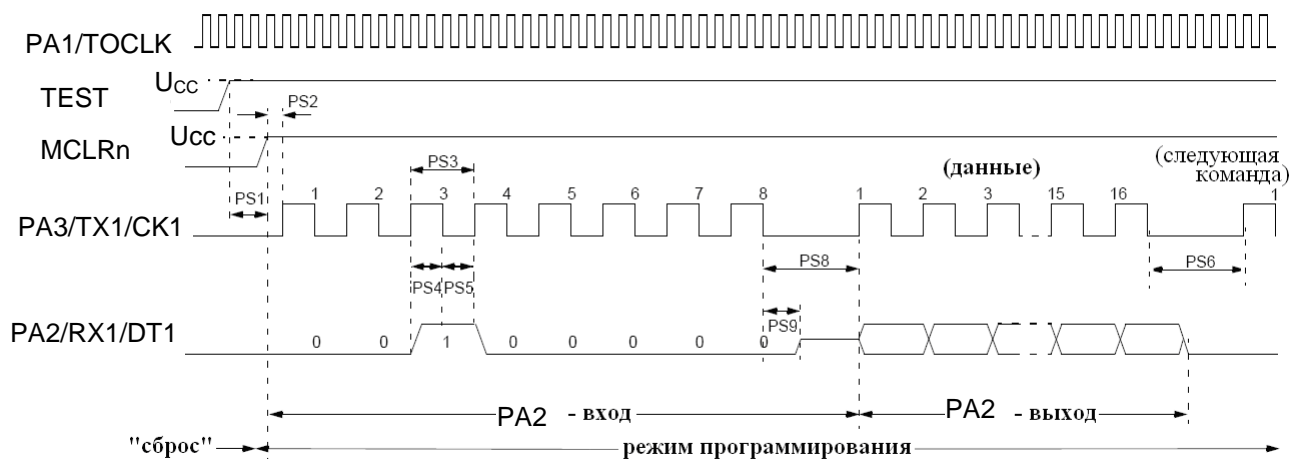


Рисунок 6 – Команда чтения данных

3.7 Старт программирования

Эта команда используется для программирования текущих данных (посланных последней командой ЗАГРУЗКА ДАННЫХ) по адресу памяти программ на который ссылается указатель адреса. Время программирования данных 200 микросекунд. После прохождения этого времени любая команда может быть послана и выполнена. Временная диаграмма команды приведена на рисунке 7.

Инв. №	Подп. и дата	Взам. инв	Инв. №	Подп. и дата

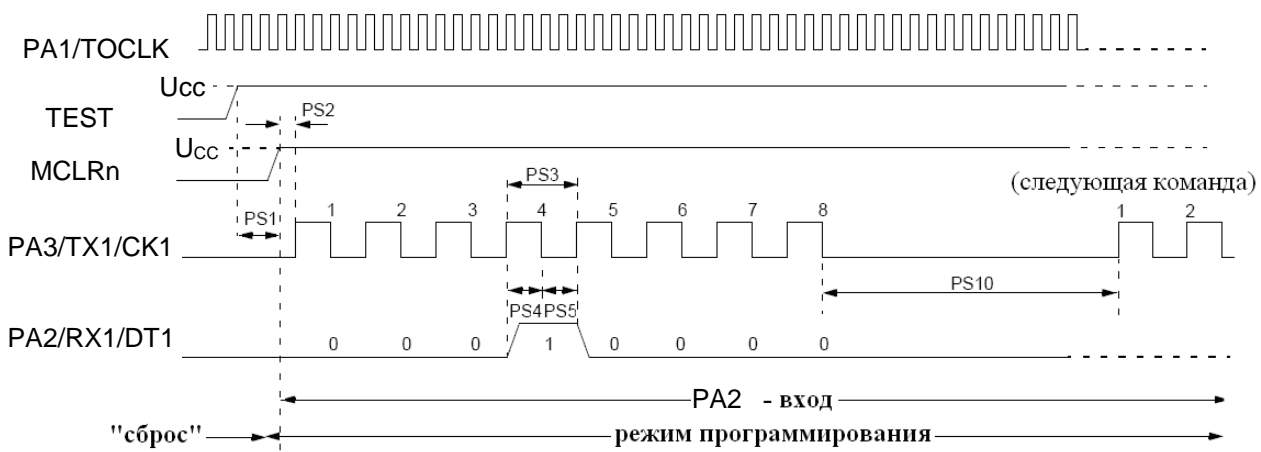


Рисунок 7 – Команда программирования

3.8 Запись делителя частоты

Эта команда предназначена для записи коэффициента деления частоты генератора в 8 разрядный регистр микроконтроллера (EEDIV) с целью выработки импульса записи и стирания памяти программ длительностью 5 мс. Делитель передается в первом байте 16 разрядных данных. Правило выбора коэффициента деления:

$$K = 5 \text{ мс} / 651 \cdot T_{osc}$$

Временная диаграмма команды приведена на рисунке 8.

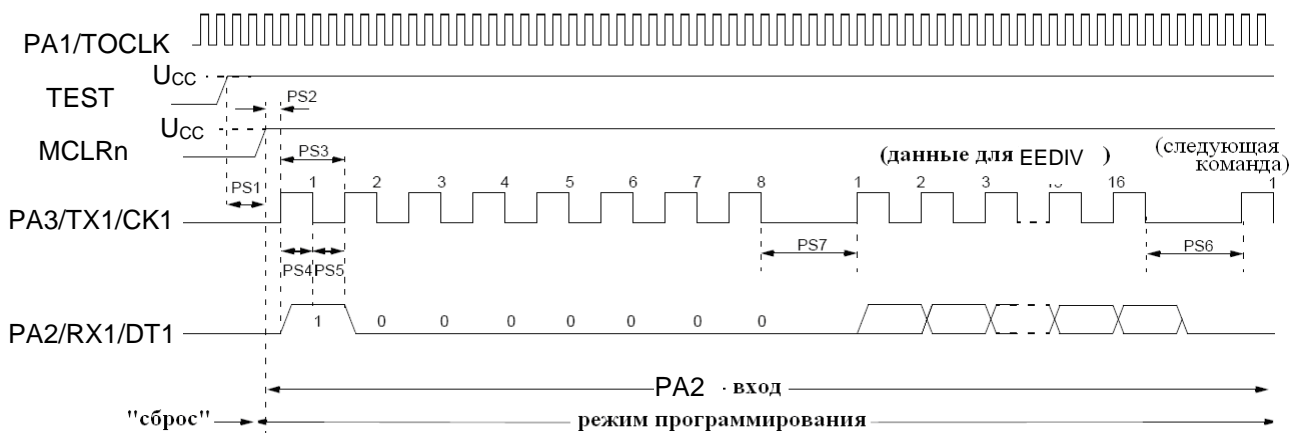


Рисунок 8 – Команда записи делителя частоты

3.9 Стирание сектора

Эта команда предназначена для запуска процедуры стирания EEPROM-памяти. Временная диаграмма команды приведена на рисунке 9.

Инв. №	Подп. и дата
Взам. инв	Инв. №
Подп. и дата	Подп. и дата
Инв. №	Подп. и дата

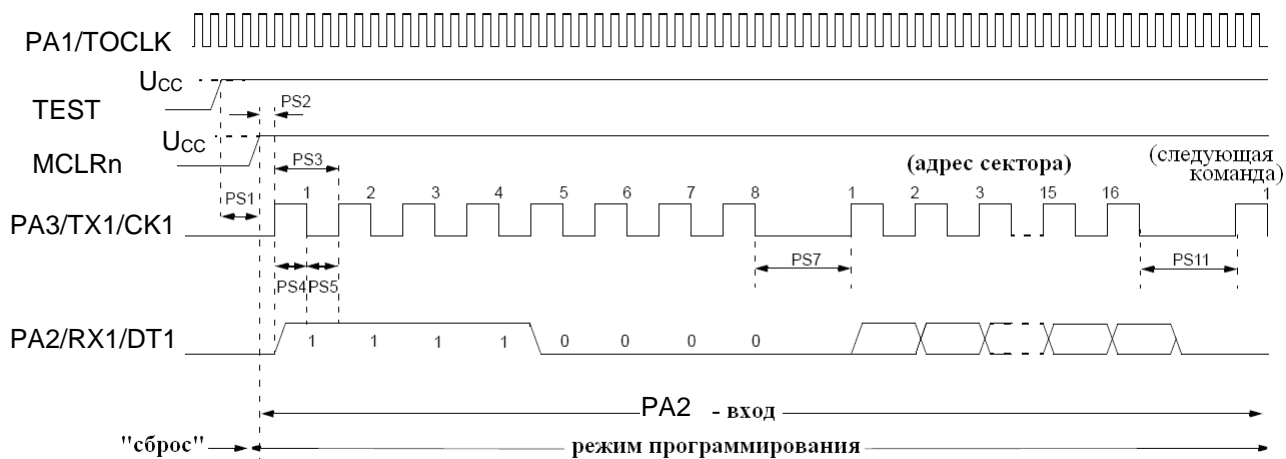


Рисунок 9 – Команда стирания сектора

3.10 Тест сектора на чистоту

Эта процедура проводится после операции стирания сектора, для проверки чистоты сектора.

В случае, если все ячейки памяти в адресном пространстве проверяемого сектора содержат значение 0x0000, сектор считается очищенным и из канала USART/SCI читаются 16 разрядные данные 0xFFFF, в противном случае читаются 0x0000. Временная диаграмма команды приведена на рисунке 10.

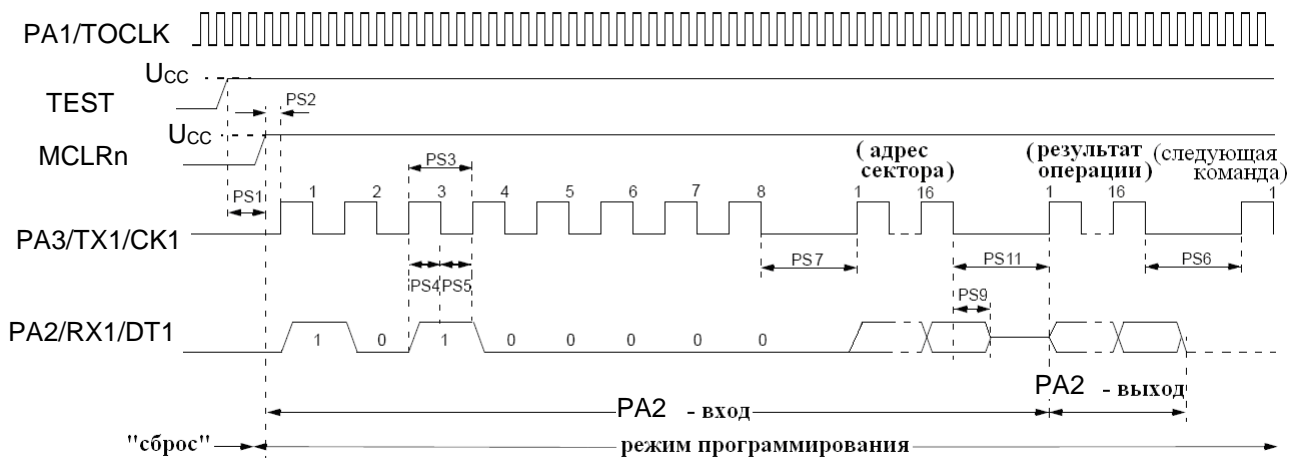


Рисунок 10 – Команда тест сектора на чистоту

Инв. №	Подп. и дата	Взам. инв	Инв. №	Подп. и дата

3.11 Функция ЧТЕНИЕ ОЗУ+ПЕРЕФЕРИЯ

Команда функции состоит из 4 частей:

1. Запись команды чтения ОЗУ (0x10).
2. Запись банка ОЗУ (регистр BSR).
3. Запись адреса ОЗУ.
4. Чтение полученных данных DATA.

3.12 Функция ЗАПИСЬ ОЗУ+ПЕРЕФЕРИЯ

Команда функции состоит из 4 частей:

1. Запись команды записи ОЗУ (0x11).
2. Запись банка ОЗУ (регистр BSR).
3. Запись адреса ОЗУ.
4. Запись данных DATA.

3.13 Функция ЗАПРОС ГОТОВНОСТИ

Команда функции состоит из 2 частей:

1. Запись команды Запрос Готовности (0x12).
2. Чтение данных DATA. Если принятое значение отличается от 0x55, то микроконтроллер не готов и синхронизация не восстановлена (ожидаемое значение при неготовности микроконтроллера 0xFF). После получения значения равного 0x55 микроконтроллер готов к обработке следующих команд.

3.14 Функция ПЕРЕХОД CALL

Команда функции состоит из 3 частей:

1. Запись команды Переход CALL (0x13).
2. Запись младшей половины адреса.
3. Запись старшей половины адреса.

После выполнения команды для восстановления синхронизации используется команда Запрос Готовности.

3.15 Функция ПЕРЕХОД GOTO

Команда функции состоит из 3 частей:

1. Запись команды Переход GOTO (0x14).
2. Запись младшей половины адреса.
3. Запись старшей половины адреса.

После выполнения команды для восстановления синхронизации используется команда Запрос Готовности.

3.16 Функция ВЕРСИЯ МИКРОКОНТРОЛЛЕРА

Команда функции состоит из 2 частей:

1. Запись команды Версия Микроконтроллера (0x15).
2. Чтение данных DATA.

Инд. №	Подп. и дата	Взам. инв	Инд. №	Подп. и дата
Инд. №	Подп. и дата	Взам. инв	Инд. №	Подп. и дата

Из	Лист	№ докум.	Подп.	Дата

ТСКЯ.431295.007И

Лист
10

3.17 Функция Стереть всю память

Команда функции состоит из 1 части:

1. Запись команды Стереть Всю Память (0x16).

После выполнения команды для восстановления синхронизации используется команда Запрос Готовности.

4 Механизм восстановления синхронизации

При выполнении команды Переход CALL или GOTO микропроцессорное ядро может уйти на подпрограммы не обеспечивающие обработку запросов по USART. В этом случае при приеме USART новых команд, ядро их не будет обрабатывать и они будут накапливаться в очереди USART. При переполнении очереди будет выставлен флаг OVERFLOW. И дальнейшие принимаемые данные будут игнорироваться. После того как будет выполнена подпрограмма, на которую был осуществлен переход CALL или GOTO, микроконтроллер вернется в обработчик USART. При этом он должен дождаться появления флага OVERFLOW. После чего очистить очередь, считывая ранее полученные команды и игнорируя их (всего будет три команды в очереди). После очистки очереди дождаться команды запроса готовности и вернуть значение результата выполнения подпрограммы, на которую был осуществлен переход CALL или GOTO.

5 Специальные библиотечные функции ПЗУ

Инв. №	Подп. и дата	Взам. инв	Инв. №	Подп. и дата	Тест	Адрес вызова	Параметры	Комментарии
					9	Тестирование ядра	—	—
10	Тестирование ОЗУ	—	—					
11	Тестирование EEPROM ПД	—	—					
12	Тестирование порта А	—	—					
13	Тестирование Timer0	—	—					
14	Тестирование LIN	—	—					
15	Тестирование USART_async	—	—					
16	Тестирование USART_sync_ms	—	—					
17	Тестирование USART_sync_sl	—	—					
					ТСКЯ.431295.007И			
Из	Лист	№ докум.	Подп.	Дата				
					Лист 11			

В памяти программ реализованы вспомогательные тестовые программы для проведения отбраковочного тестирования микросхем. Запуск данных программ производится в тестовом режиме с помощью функции переход CALL по соответствующему адресу. После запуска необходимо восстановить синхронизацию интерфейса программирования и считать результат выполнения программы. Данные программы применяются только для отбраковочного тестирования, при их вызове в рабочем режиме, они могут нарушить работу всей программы.

6 Электрические характеристики для последовательного режима программирования

Параметр	Характеристика	Мин.	Тип.	Макс.	Единицы измерен.
U_{CC}	Напряжение питания	4,5	5	5,5	В
f_c	Частота следования импульсов тактовых сигналов на выводах PA1	–	–	10	МГц
f_{c_PROG}	Частота следования импульсов тактовых сигналов на выводах PA1 при программировании, стирании и чтении EEPROM программ			8,0	МГц
T_{OSC}	Период тактовой частоты f_c	–	$1/f_c$	–	МГц
T_{CY}	Такт инструкции микроконтроллера	–	$4/f_c$	–	МГц
PS1	Время между установкой U_{CC} на выводах TEST и MCLRn	1	–	–	мкс
PS2	Время инициализации интерфейса	160	–	–	T_{CY}
PS3	Период частоты	1	–	–	T_{CY}
$t_{SU(TX-RX)}$	Время установления данных RX/DT относительно синхронизации TX/СК	17	–	–	нс
$t_{H(TX-RX)}$	Время удержания данных RX/DT относительно синхронизации TX/СК	17	–	–	нс
PS6	Задержка между последним спадом сигнала СК и первым фронтом СК следующей команды	160	–	–	T_{CY}
PS7	Задержка между последним спадом сигнала СК передачи байта команды и первым фронтом СК записываемого слова данных	160	–	–	T_{CY}
PS8	Задержка между последним спадом сигнала СК передачи байта команды и первым фронтом СК читаемого слова данных	160	–	–	T_{CY}
PS9	Время перенастройки вывода данных с выхода на вход	1	–	–	T_{CY}
PS10	Минимальное время программирования	4	–	–	мс
PS11	Время выполнения процедуры стирания сектора	4	–	–	мс

Инв. №	Подп. и дата
	Взам. инв
Инв. №	Инв. №
	Подп. и дата
Инв. №	Подп. и дата
	Взам. инв
Инв. №	Подп. и дата
	Взам. инв

				ТСКЯ.431295.007И		Лист
Из	Лист	№ докум.	Подп.	Дата		
						12

Лист регистрации изменений

Изм	Номера листов (страниц)				Всего листов (страниц) в докум.	№ докум.	Входящий № сопроводительного докум. и дата	Подп.	Дата
	измененных	замененных	новых	аннулированных					

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата
--------------	--------------	--------------	--------------	--------------

Изм	Лист	№ докум.	Подп.	Дата
ТСКЯ.431295.007И				
Лист 13				