КОНТРОЛЛЕР MF167

Руководство пользователя

АО КАСКОД

2000

Санкт-Петербург

АО КАСКОД

196625, Санкт-Петербург, Павловск, Фильтровское шоссе, 3

тел.: (812) 476-0795, (812) 466-5784, факс: (812) 465-3519

E-mail : cascod@online.ru
kaskod@spb.cityline.ru

http://www.kaskod.ru

Принятые сокращения

АЦП	–	Аналого-цифровой преобразователь.
ОЗУ	–	Оперативное запоминающее устройство.
ПЗУ	–	Постоянное запоминающее устройство.
CAN	–	Controller Area Network (Контроллер CAN сети).
ШИМ	–	Широтно-импульсная модуляция.
PEC	–	Peripheral Event Controller (Периферийный контроллер событий).
CPU	–	Central Processing Unit (Центральное процессорное устройство).
CS	–	Chip Select (выбор микросхемы).
CAPCOM	–	Capture/Compare (Блок захвата/сравнения).
GPT	–	General Purpose Timer unit (Блок таймеров).
GPR	–	General Purpose Register (Регистры общего назначения).
nc	–	Свободный контакт .
GND	–	Общий провод питания.
VCC	–	Напряжение питания +5 вольт.
iRES	–	Сигнал “Сброс” контроллера.
Res	–	Сигнал “Сброс”.
WRH	–	Сигнал “Запись старшего байта”.
WRL	–	Сигнал “Запись младшего байта”.
R/W	–	Сигнал “Чтение/Запись”.
RD	–	Сигнал “Чтение”.
SCLK	–	Сигнал тактирования.
ALE	–	Сигнал разрешения адреса.
NMI	–	Немаскируемое прерывание.
AEN	–	Разрешение адреса.
Ax	–	Бит адреса x, где x=0-23.
Dy	–	Бит данных y, где y=0-15.
rst	–	Restart (сброс канала часового таймера).
лог.1	–	Уровень логической единицы.
лог.0	–	Уровень логического нуля.
REF	–	Опорное напряжение.

Содержание **Страница**

1. Назначение 6
2. Технические характеристики 7
3. Структурная схема контроллера 8
4. Распределение памяти контроллера 10
5. Последовательный порт 11
6. Часовой таймер 12
7. CAN интерфейс 14
8. АЦП 15
9. Математический сопроцессор 17
10. Работа в отладочном режиме 18
11. Программирование микросхем FLASH памяти 19
12. Старт контроллера из ПЗУ 20
13. Сброс контроллера 20
14. Питание контроллера 21
15. Подключение батареи супервизора 21
16. Внешние разъемы и переключатели 22
17. Комплект поставки 27
18. Варианты исполнения контроллера 27
19. Габаритные и установочные размеры 28
20. Приложения 29
    - а) преобразование кода Грея в двоичный весовой код 29
    - б) генерация шума (псевдослучайная последовательность) 29
    - в) формирование звука 29
    - г) работа с внешним АЦП микроконтроллера 31
    - д) работа с часовым таймером 31
    - е) работа с внутренним АЦП микроконтроллера 33
    - ж) стартовый загрузчик 33
    - з) генерация сигналов PWM 34
    - и) генерация сигналов высокоскоростного PWM 39
    - к) работа с CAN 42
    - н) система команд микроконтроллероов 80C16x 46
    - п) функциональная схема 51

1. Назначение

Контроллеры серии MF167 разработаны на базе 16-ти разрядных микроконтроллеров фирм Siemens (Infineon) **C167** и STMicroelectronics **ST10R167** и 32-х разрядного математического сопроцессора с плавающей точкой. Контроллеры MF167 предназначены для построения прецезионных цифровых систем реального времени:

– систем управления электродвигателями различных типов,

– систем питания различных типов,

– следящих систем,

– систем управления и синхронизации энергетических объектов,

– систем сбора и обработки информации,

– распределенных систем управления и т.д.

Общий вид контроллера MF167 представлен на рисунке 1.

Программное обеспечение, поставляемое в комплекте с контроллерами, позволяет разрабатывать и отлаживать программы в интерактивном режиме без использования дополнительного отладочного оборудования (внешние программаторы, эмуляторы, и т.д.). Контроллер подключается к PC компьютеру через интерфейс RS232.

Полноэкранный интерактивный отладчик контроллера позволяет использовать различные режимы отладки с полным отображением состояния контроллера на экране PC компьютера.

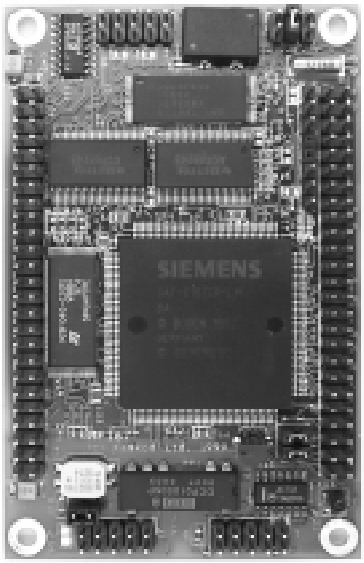


Рис. 1

2. Технические характеристики

- Время выполнения команды – 100 (80) нс (пересылка типа регистр-регистр).
- Максимальное время выполнения команд умножения 16 бит * 16 бит с результатом 32 бита – 500 (400) нс.
- Максимальное время выполнения команд деления 32/16 с результатом 16/16 – 1000 (800) нс.
- Операции умножения и деления прерываемы.

  Примечание: Значения указаны для тактовой частоты процессора 20 МГц.
  Значения в скобках приведены для частоты 25 МГц.
- Объем ПЗУ (FLASH) – до 1 Мбайт.
- Объем ОЗУ – 256 Кбайт.
- 32-х разрядный математический сопроцессор с плавающей точкой. (Время выполнения функций приведено в таблице (см. гл.9).
- 35 входов прерываний.
- Типовое время реакции на прерывание – 300 нс.
- 2 Кбайта внутренней памяти.
- 8-канальный блок PEC для пересылок типа память-память, память-порт, память-последовательный порт.
- Последовательный канал RS232 со скоростью передачи до 625 Кбит/с.
- Оптоизолированный CAN интерфейс (спецификация 2.0B) со скоростью передачи до 1 Мбит/с.
- Девять 16-разрядных таймеров-счетчиков с предделителями и возможностью каскадирования до 32 или 33 разрядов.
- 27 каналов обработки и формирования цифровых сигналов с разрешением 400 нс из них 8 каналов могут использоваться в режиме быстрого прерывания – 50 нс.
- Скоростной синхронный порт со скоростью передачи до 5 Мбит/с.
- 16 каналов 10-разрядного аналого-цифрового преобразователя (АЦП), которые могут использоваться как входы цифрового ввода. Время преобразования для одного канала АЦП – 9,75 мкс (при тактовой частоте процессора 20 МГц).
- 8-канальный 12-разрядный аналого-цифровой последовательный преобразователь (АЦП).
- Сторожевой 16-разрядный таймер (WatchDog).
- Супервизор питания (сохранение параметров в ОЗУ и часовом таймере при выключении питания).
- Часовой таймер (секунды, минуты, часы, дни, месяцы, годы) с разрешением 1 с.
- Стартовый (Bootstrap Loader) загрузчик. Позволяет работать с ОЗУ, программировать Flash прямо на контроллере. Для загрузки программы используется канал RS232.
- Питание от одного источника +5 В (напряжение +12 В для RS232 формируется на плате).
- Диапазон рабочих температур: от 0 ˚С до +70 ˚С (стандартный),
  от -40 ˚С до +85 ˚С (по заказу).

3. Структурная схема контроллера

Структурная схема контроллера приведена на рис. 2.

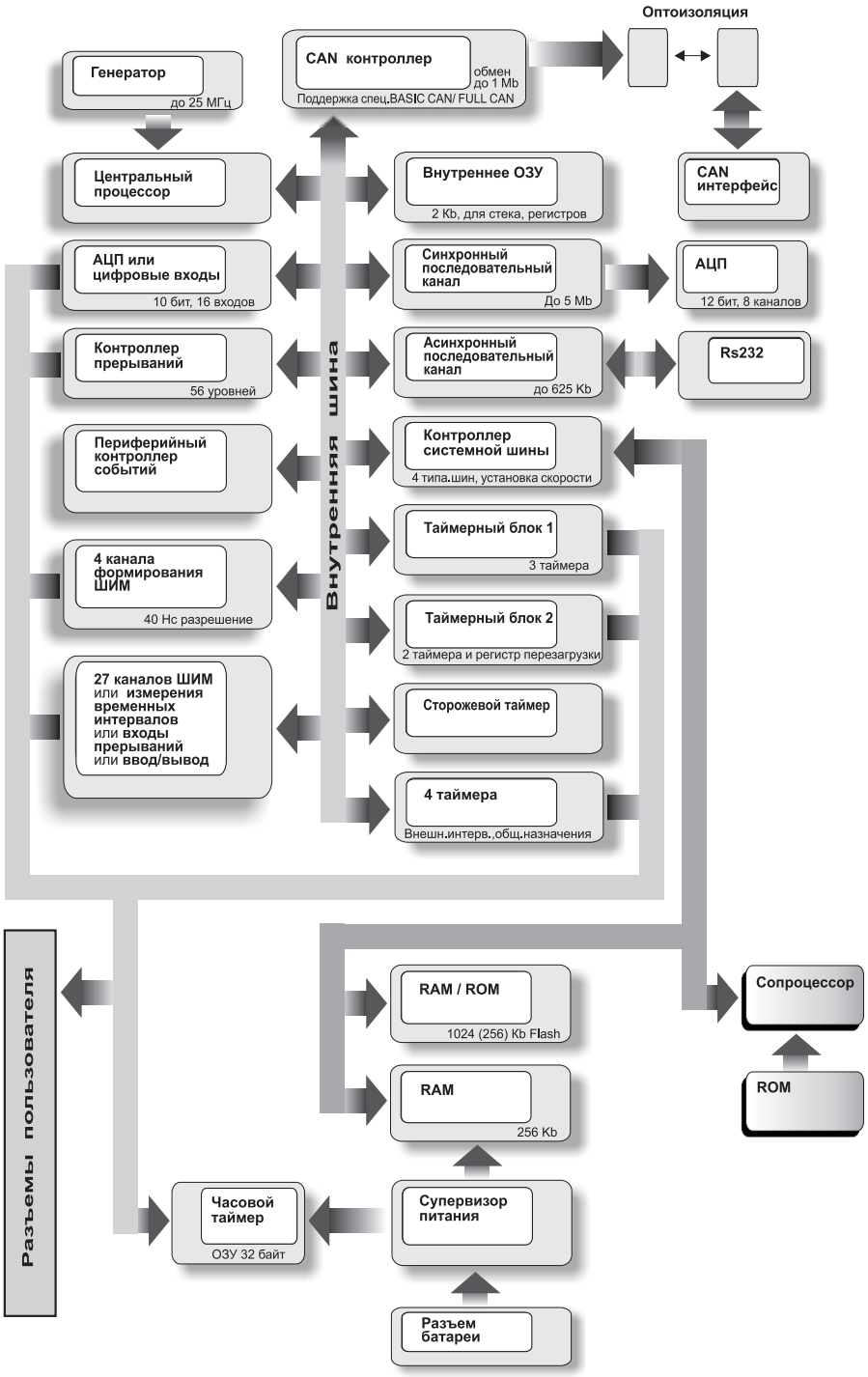


Рис. 2

Микроконтроллер **C167/ST10R167** состоит из следующих устройств:

ЦПУ – 16-разрядный процессор с шестнадцатью 16-разрядными регистрами общего назначения (GPR).

ОЗУ – оперативное запоминающее устройство в котором располагаются:

- системный стек;
- регистры CPU и внешних устройств;
- область битовых переменных.

Контроллер прерываний

– поддерживает 56 векторов прерываний.

Каждое из устройств, способных генерировать запрос прерывания, имеет управляющий регистр с 4 глобаль-ными и 2 групповыми битами уровня прерывания, битом запроса прерывания и разрешения прерывания. С контроллером прерываний объединен контроллер периферийных событий (PEC).

Кроме этого имеется вход немаскируемого прерывания NMI.

Контроллер шины – обеспечивает режимы работы внешней шины, различные по временным параметрам и наличию сигналов записи, чтения и готовности шины.

Режимы работы шины:

- 8-битовый немультиплексный;
- 16-битовый немультиплексный;
- 8-битовый мультиплексный;
- 16-битовый мультиплексный;

Сторожевой таймер – предназначен для восстановления работоспособности системы. Контрольный период этого таймера может быть изменен.

АЦП – 16-канальный 10-разрядный аналого-цифровой преобразователь. Может работать в следующих режимах:

- режим однократного преобразования для одного, выбранного канала;
- режим повторяющегося преобразования для одного, выбранного канала;
- режим однократного преобразования для каждого канала из выбранной группы;
- режим повторяющегося преобразования для выбранной группы каналов;
- режим автоматического запуска следующего преобразования после считывания данных;
- режим вставки преобразования для одного канала в режиме группового преобразования.

PWM – 4-канальный 50-ти наносекундный блок формирования ШИМ. Может работать в следующих режимах:

- режим стандартного генератора ШИМ;
- режим генератора симметричного ШИМ;
- режим модуляции одного канала другим;
- режим программного формирования одиночного импульса.

ASC0 – Последовательный порт. Может работать в следующих режимах:

- асинхронные режимы: семибитовый с битом паритета, восьмибитовый, восьмибитовый с битом будильника, восьмибитовый с битом паритета, девятибитовый.
- синхронный восьмибитовый.

SSC – Синхронный порт.
Может работать в следующих режимах:

Master – скорость передачи данных определяется контроллером;

Slave – скорость передачи задается внешним устройством.

Длина посылки программируется от 2 до 16 битов. Имеется возможность выбора последовательности передачи данных, начиная с младшего или старшего битов, синхронизирующего фронта или спада, пассивного состояния низкого или высокого уровня, что позволяет использовать на одной шине разнотипные устройства.

CAPCOM – состоит из 2-х блоков. Каждый блок состоит из 16 регистров захвата/сравнения, 2-х таймеров и 4-х регистров управления. Блоки CAPCOM позволяют формировать до 32 независимых каналов ШИМ. Для каждого регистра захват/сравнение устанавливается один из режимов работы:

- режим захвата и сравнения отключен;
- режим захвата по положительному перепаду;
- режим захвата по отрицательному перепаду;
- режим захвата по отрицательному и по положительному перепадам;
- режим сравнения с генерацией нескольких прерываний за период;
- режим двухрегистрового сравнения. Выход отключен;
- режим сравнения с генерацией нескольких прерываний за период. Выход отключен;
- режим сравнения с генерацией только одного прерывания за период. Выход отключен;
- режим сравнения с генерацией только одного прерывания за период. Выход устанавливается при равенстве значений в регистре и таймере и сбрасывается при переполнении таймера.

GPT – состоит из двух блоков таймеров общего назначения (GPT1 и GPT2).

Состав блока таймеров GPT1: три 16-разрядных таймера T2, T3 и T4. Каждый таймер может работать в следующих режимах:

- режим таймера;
- режим счетчика;
- режим старт/стопного таймера;
- режим каскадирования таймера T3 с одним из таймеров T2 или T4. Образуется 32-разрядный или 33-разрядный таймер.

Каждый таймер может вести счет на увеличение и на уменьшение.

Состав блока таймеров GPT2: два 16-разрядных таймера T5, T6 и регистр захвата CAPREL.

Каждый таймер может работать в следующих режимах:

- режим таймера;
- режим счетчика;
- режим старт/стопного таймера;
- режим каскадирования таймеров T5 и T6. Образуется 32-разрядный или 33-разрядный таймер.

Каждый таймер может вести счет на увеличение и на уменьшение.

4. Распределение памяти контроллера

Общий объем памяти контроллера может достигать 16 Мбайт. Пространство памяти разбито на 1024 страницы по 16 Кбайт или 256 сегментов по 64 Кбайт. Страничная адресация (по 16 Кбайт) осуществляется с помощью специальных регистров DPP0-DPP3. Сегментная адресация (по 64 Кбайт) используется в командах длинных вызовов подпрограмм и командах EXTS, EXTSR. Микроконтроллер позволяет изменять распределение памяти между устройствами, в том числе и динамически.

Внимание: Адресное пространство шины достигает 16 Мбайт только в том случае, если не используется CAN контроллер.

Порт P6 микроконтроллера используется в режиме внутреннего адресного дешифратора.

Назначение сигналов выборки устройств CS0-CS4:

Сигнал CS0 предназначен для выборки микросхем ПЗУ.

Сигнал CS1 предназначен для выборки микросхем ОЗУ.

Сигнал CS2 предназначен для обмена с сопроцессором.

Сигнал CS3 (P6.3) предназначен для ввода/вывода.

Сигнал CS4 (P6.4) предназначен для ввода/вывода.

Соответствие сигналов выборки устройств (CS0-CS2) регистрам конфигурации микроконтроллера:

CS0	BUSCON0		....	ПЗУ
CS1	BUSCON1,	ADDRSEL1	....	ОЗУ на плате
CS2	BUSCON2,	ADDRSEL2	....	Сопроцессор

5. Последовательный порт

Контролер имеет один асинхронный последовательный порт RS232, который используется отладчиком. Для подключения внешних линий связи используется разъем J9.

Разъем J9 предназначен для подключения кабеля RS232 к последовательному порту 0 микроконтроллера C167/ST10.

Разъем J9

1,2,7	–	соединены между собой
3	–	RXD (прием данных в контроллер).
4,6	–	соединены между собой
5	–	TXD (передача данных из контроллера)
8	–	свободный
9,10	–	общий (земля)

6. Часовой таймер

Для управления часовым таймером используются следующие биты порта микроконтроллера:

Синхронизирующие импульсы	(SCLK)	– бит порта P4.4
Данные таймера	(I/O)	– бит порта P4.7
Сброс таймера	(RST)	– бит порта P6.5

Подключение питающего напряжения не приводит к старту таймера. Для его запуска необходимо записать в регистр секунд любое правильное значение секунд.

Часовой таймер сохраняет работоспособность при подключенной батарее и отключенном напряжении питания контроллера.

Календарь таймера учитывает количество дней в месяцах, включая високосные годы.

Пример программы работы с часовым таймером приведен в приложении.

Часовой таймер содержит 7 регистров календаря и 31 байт статического ОЗУ. Обмен данными производится по синхронному последовательному интерфейсу одиночными байтами или блоками объемом до 31 байта. Сначала передается адрес команды (командный байт), определяющий к какому из сорока байтов данных происходит доступ, определяется режим передачи одного байта или блока, выбирается цикл чтение или запись.

При обмене данными сигнал RST устанавливается в состояние лог.1, затем подаются синхронизирующие импульсы SCLK.

Диаграммы передачи данных приведены на рисунке 4.

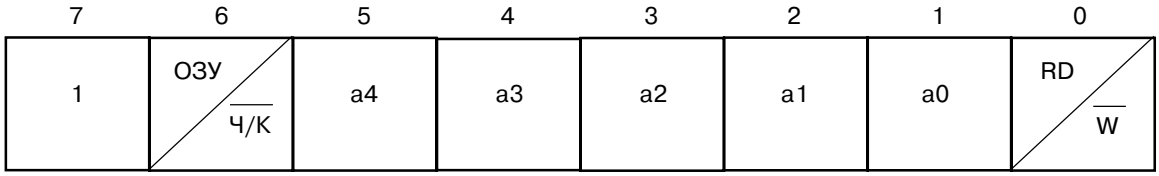


Рис. 3. Структура командного байта

Структура командного байта показана на рисунке 3.

Структура командного байта:

Бит 7:	логическая 1.
Бит 6:	логическая 1 – ОЗУ, логический 0 – часы/календарь.
Биты 5 - 1:	определяют регистр данных или режим передачи данных блоком (все – лог.1).
Бит 0:	логическая 1 – цикл чтения из часового таймера (ОЗУ), логический 0 – запись в часовой таймер (ОЗУ).

Старт командного байта происходит с бита 0.

Старт байта данных происходит с бита 0.

Соответствие командных байтов и байтов данных приведены на рисунке 5.

Часовой таймер содержит семь регистров с данными в двоично-десятичном формате.

Бит 7 регистра «секунды» указывает на состояние часов: лог.1 – часы стоят, лог.0 – часы работают.
Бит 7 регистра «часы» определяет режим: лог.1 – 01-12 часов, лог.0 – 00-24 часа.
Бит 7 регистра «контроль» – бит защиты записи. Перед циклом записи в регистры таймера или ОЗУ, бит защиты записи должен быть установлен в лог.0.
Биты 6-0 регистра контроля всегда в состоянии лог.0, при чтении должны быть прочитаны как лог.0.

При передаче блоком происходит последовательное чтение или запись восьми регистров часового таймера (включая регистр контроля) следующих за командным байтом, стартующих с бита 0, адреса 0. При установленном в лог.1 бите защиты записи приём данных в этом режиме не происходит.

Статическое ОЗУ (31 * 8 байт) адресуется последовательно в адресном пространстве часового таймера.

При передаче данных блоком происходит последовательное чтение или запись 31 регистра ОЗУ часового таймера, следующих за командным байтом, стартующих с бита 0, адреса 0.

Диаграмма передачи данных

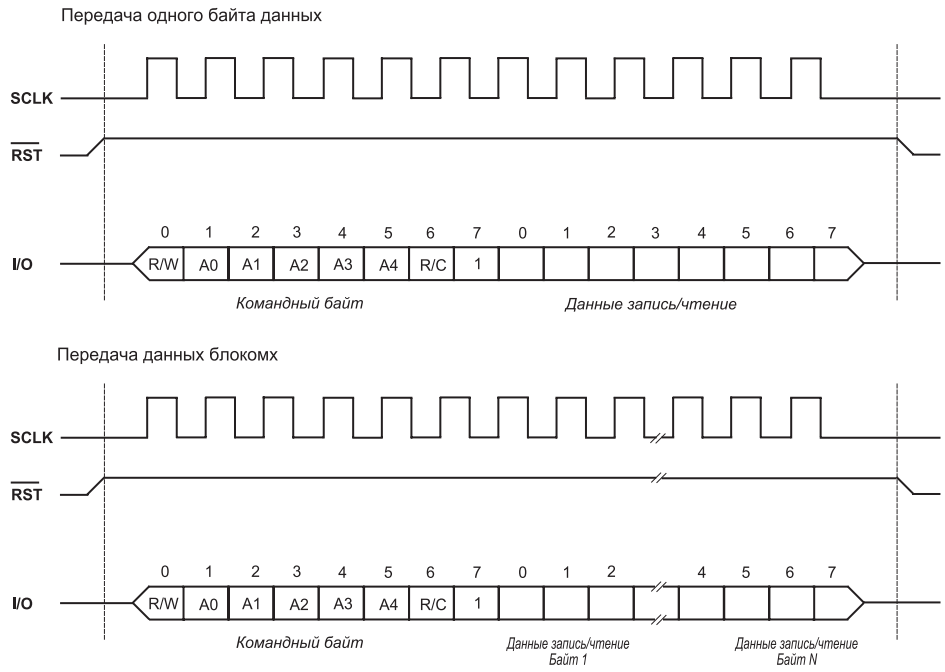


Рис. 4

Формат регистров часов реального времени

Командные регистры

А. Часы/Календарь

									Регистры данных									
Секунды	1	0	0	0	0	0	0	RD/W	00-59	Ch	10 сек			Секунды				
Минуты	1	0	0	0	0	0	1	RD/W	00-59	0	10 мин			Минуты				
Часы	1	0	0	0	0	1	0	RD/W	01-12 00-23	0	0	10 A/P	Час	Часы				
Число	1	0	0	0	0	1	1	RD/W	01-28/29 00-30 01-31	0	0	10 дата		Число				
Месяц	1	0	0	0	1	0	0	RD/W	01-12	0	0	0	10 мес	Месяц				
День	1	0	0	0	1	0	1	RD/W	01-07	0	0	0	0	0	День			
Год	1	0	0	0	1	1	0	RD/W	00-99	10 год					Год			
Контроль	1	0	0	0	1	1	1	RD/W		WP	0	0	0	0	0	0	0	
Не использ.	1	0	0	1	0	0	0	RD/W		TCS	TCS	TCS	TCS	DS	DS	RS	RS	
Режим передачи блоком	1	0	1	1	1	1	1	RD/W										

Б. ОЗУ

ОЗУ 0	1	1	0	0	0	0	0	RD/W	ОЗУ данные 0									
ОЗУ 30	1	1	1	1	1	1	0	RD/W	ОЗУ данные 30									
Режим передачи блоком	1	1	1	1	1	1	1	RD/W										

Рис. 5

7. CAN интерфейс

Контроллер MF167 имеет в своем составе оптоизолированный сетевой CAN интерфейс, предназначенный для построения мультипроцессорных систем реального времени.

На плате контроллера установлен буфер, который позволяет подключить контроллер к CAN шине, содержащей до 32 устройств (рис. 6). При использовании большего количества устройств на CAN шине необходим модуль расширителя.

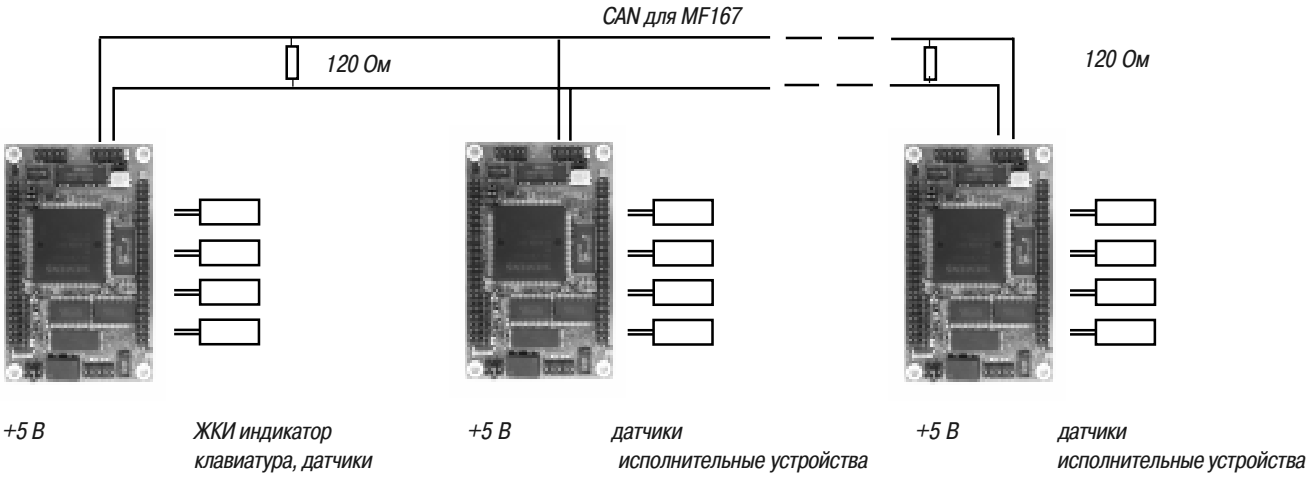


Рис. 6

CAN контроллер занимает 256 ячеек памяти в первом сегменте памяти, начиная с адреса EF00h. В этой области располагаются 15 регистров объектов-сообщений, регистры конфигурации, контроля и арбитража, регистр статуса/контроля (0EF00h), регистр прерываний (0EF02h), регистр битов таймера (0EF04h), регистры глобальной (длинной (0EF08h) и короткой (0EF06h)) маски, маска последнего сообщения (0EF0Ch). Для приема и передачи используются биты P4.5 и P4.6 порта микроконтроллера, поэтому при использовании CAN контроллера максимальный размер внешней памяти может быть не более 1 МБайт. Каждый из объектов-сообщений имеет бит XTD, определяющий работу со стандартным или расширенным идентификатором. Длина сообщения может быть от 1 до 8 байт. Объекты-сообщения с номерами с 1 по 14 могут работать на передачу и прием, сообщение с номером 15 имеет буферированный режим, то есть можно принять еще одно сообщение до того, как будет прочитано предыдущее.

Разъем J3 предназначен для подключения контроллера в CAN сеть.

Номер контакта	*Сигнал*
1	не используется
2	не используется
3	BUS.L
4	BUS.H
5	GND CAN
6	не используется
7	не используется
8	не используется
9	не используется
10	не используется

8. АЦП

На плате контроллера MF167 установлен 8-канальный 12-разрядный последовательный АЦП. Для управления сигналом CS АЦП используется порт P3-15. При обращении к АЦП $\overline{CS}$ устанавливается в «0».

Формат данных при обращении к регистру команд АЦП:

X	0	A2	A1	A0	Ref	PM1	PM0

PM0-PM1 – управление режимом питания АЦП. PM1 – “0”, PM0 – «0»
Ref – “0” – включить, «1» – выключить внутренний источник опорного напряжения АЦП, в этом режиме внешнее опорное напряжение может быть от 1,2 до 5 Вольт
A2-A0 – выбор канала АЦП для преобразования
000 – 1 канал
001 – 2 канал
……
111 – 8 канал
0 – всегда 0
X – значение произвольное
Ain1 – Ain8 – аналоговые входы

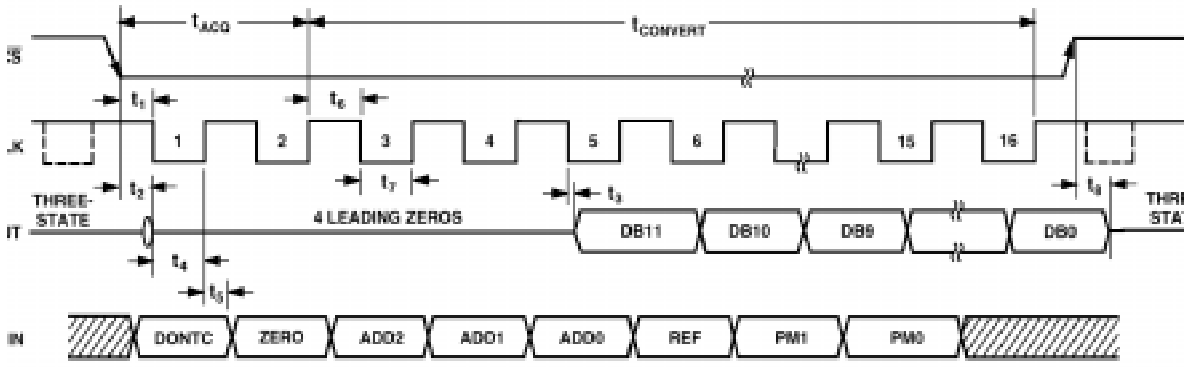
Диапазон входного напряжения всех входов от 0 до U_{Ref}.

Неиспользованные входные каналы должны быть связаны с aGND, чтобы избежать помех.

Внутреннее опорное напряжение АЦП 2,5 +- 0,05 В.

При записи в регистр управления АЦП необходимого номера канала синхронно происходит вывод из АЦП значения предыдущего записанного канала.

При включении питания по умолчанию в регистр управления записывается номер 1-го канала.



Разъем J6.

Номер контакта	*Сигнал*	*Номер контакта*	*Сигнал*
1	An 0	2	An 1
3	An 2	4	An 3
5	An 4	6	An 5
7	An 6	8	An 7
9	Ref	10	aGND
11	aGND	12	aGND
13	nc	14	nc
15	nc	16	nc
17	nc	18	nc
19	nc	20	nc
21	nc	22	nc
23	nc	24	nc
25	P6.3	26	P6.4
27	P7.0	28	P7.1
29	P7.2	30	P7.3
31	P7.4	32	P7.5
33	P7.6	34	P7.7
35	P8.0	36	P8.1
37	P8.2	38	P8.3
39	P8.4	40	P8.5
41	P8.6	42	P8.7
43	P3.13	44	P3.9
45	P3.15	46	P3.8
47	nc	48	+5v (VCC)
49	nc	50	GND

Примечание:

An x	–	Аналоговый вход (номер канала последовательного АЦП).
Ref	–	Опорное напряжение АЦП.
nc	–	Контакт свободный.
aGND	–	Аналоговая земля последовательного АЦП.
GND	–	Цифровая земля (общий провод).
+5v (VCC)	–	Напряжение питания контроллера.
P6.3 - P6.4	–	Порт P6 микроконтроллера.
P7.0 - P7.7	–	Порт P7 микроконтроллера.
P8.0 - P8.7	–	Порт P8 микроконтроллера.
P3.8, P3.9, P3.13, P3.15	–	Порт P3 микроконтроллера.

АЦП поддерживает диапазон уровней входного сигнала: от 0 до +2,5 В (при использовании внутреннего опорного напряжения АЦП), от 0 до 1,2 В - 5 В (при использовании внешнего опорного напряжения АЦП в зависимости от его величины).

Выходное опорное напряжение АЦП: 2,5 В , ±0,05 В, температурный коэффициент 50 ppm/C.

Примеры программ работы с АЦП приведены в приложении Е).

9. Математический сопроцессор

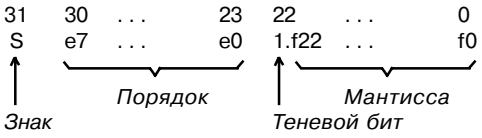
Контроллеры этой серии содержат в своём составе микроконтроллер C167/ST10R167 и 40-разрядный математический сопроцессор, позволяющий увеличить быстродействие и упростить разработку программ, содержащих большой объём вычислений.

Регистры сопроцессора находятся в адресном пространстве внешней памяти микроконтроллера C167/ST10R167, что обеспечивает удобный доступ к данным во время записи и чтения операндов.

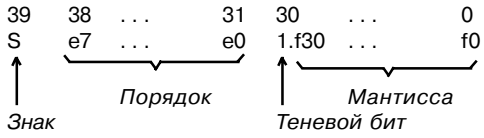
Вычисления внутри сопроцессора выполняются в расширенном 40-битовом формате.

Формат плавающих чисел сопроцессора соответствует стандарту **IEEE 754/854.**

32-х разрядный формат:



40 разрядный формат:



Для 32-х разрядных плавающих чисел диапазон значений экспоненты *e* находится в пределах *1<=e<=254*, для вычисления истинного значения экспоненты необходимо вычесть 127.

Экспонента	*Мантисса*	*Значение*
255	>0	Не определено
255	0	$(-1)^S$.бесконечность
1<=e<=254	Любое значение	$(-1)^S (1.f22-f0)2^{e-127}$
0	0	$(-1)^S$. 0

Математический сопроцессор представляет собой контроллер, выполняющий подпрограммы функций, которые загружаются во внутренне ОЗУ сопроцессора из отдельной Flash с помощью стартового загрузчика при инициализации (программном сбросе).

Обмен данными с 32-разрядными регистрами сопроцессора осуществляется по два 16-разрядных слова по одному адресу (младшее слово первое).

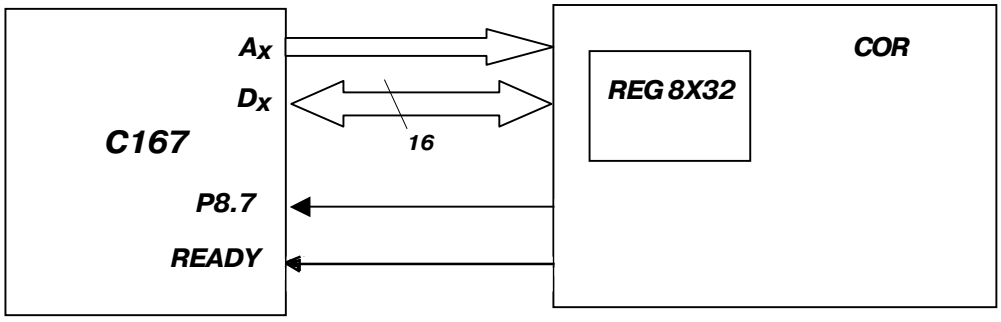


Таблица математических функций сопроцессора

Тип функции	*Время выполнения (наносекунды)*	*32-разрядн.код функции Старш. 16-бит*	*Младш. 16-бит*	
Одноoперандные:				
X=Fix(x)	100	0	15H	преобразовать плавающее в целое
Fx=Float(x)	100	0	17H	преобразовать целое в плавающее
Y=Cos(x)	700	0	01H	косинус X
Y=Sin(x)	600	0	02H	синус X
Y=Tg(x)	700	0	03H	тангенс X
Y=atan(x)	1100	0	10H	арктангенс X
Y=acos(x)	2100	0	12H	арккосинус X
Y=asin(x)	2000	0	13H	арксинус X
Y=Exp(x)	700	0	04H	e в степени X
Y=flog2(x)	900	0	09H	логарифм X по основанию 2
Y=flog10(x)	900	0	0AH	логарифм X по основанию 10
Y=flog(x)	900	0	0BH	натуральный логарифм
Z=logY(x)	1800	0	14H	логарифм Y по основанию X
Y=sqrt(x)	900	0	0CH	корень квадратный из X
Y=1/sqrt(x)	300	0	0DH	1 делить на корень квадратный из X
Y=1/x	200	0	0FH	1 делить на X
X=mod(x)	100	0	19H	модуль X
Двухоперандные:				
X,Y=Fix(x,y)	150	0	16H	преобразовать плавающие в целые X и Y
Fx,Fy=Float(x,y)	150	0	18H	преобразовать целые в плавающие X и Y
Z=atan(x/y)	1400	0	11H	арктангенс X/Y
Z=x^y	700	0	0EH	X в степени Y
Z=x/y	200	0	08H	X делить на Y
Z=x*y	100	0	05H	X умножить на Y
Z=x+y	100	0	06H	X плюс Y
Z=x-y	100	0	07H	X минус Y
Z=mod(x+y)	100	0	1CH	модуль суммы X и Y
Z=mod(x-y)	100	0	1DH	модуль разности X и Y
Z=(x+y)/2	100	0	1EH	сумма X и Y деленная на 2
X=neg(x)	100	0	1FH	преобразовать в отрицательное
Z=comp(x,y)	100	0	20H	сравнение двух операндов
Z=max(x,y)	100	0	21H	максимальное из X и Y
Z=min(x,y)	100	0	22H	минимальное из X и Y
Специальные:				
A(x),B(x)=Fase2(x)	1200	0	24H	2-х фазный конвертор, на выходе SIN(x) и COS(x)
A(x),B(x),C(x)=Fase3(x)	1800	0	23H	3-х фазный конвертор, на выходе 3 значения синуса сигнала со сдвигом 120 градусов
X,Y, Z=Fix(x,y,z)	150	0	25H	преобразовать плавающие в целые X, Y, Z
Ascii>flt	6000	0	1AH	Преобразовать последовательность ASCII символов в плавающее число
Flt>ascii	9000	0	1BH	Преобразовать плавающее число в последовательность ASCII символов

Примечание: Время выполнения плавающих операций не включает время записи и чтения операндов.

Адреса этих регистров располагаются в области внешней памяти процессора C167/ST10R167 (строб CS2). Для доступа к этим регистрам необходимо установить соответствующие регистры кофигурации шин в C167/ST10R167.

BUSCON2: 04BFh
ADDRSEL2: F007h

Считывание результата по сигналу READY при установке соответствующего бита в BUSCON2 (14BFh) или по перепаду сигнала (порт P8.7/CC23).

Считывание и запись в 32-битовые регистры осуществляется двумя 16-бит словами **по обному и тому же адресу** (младшее слово первое).

Регистры операндов 32 бита

Код функции – 1-е слово
0h – 2-е слово

	до выполнения	*после выполнения*
0F2 : 0010H	Код функции	0
0F2 : 0012H	1 операнд	результат 1
0F2 : 0014H	2 операнд	результат 2
0F2 : 0016H	3 операнд	результат 3
0F2 : 0018H	4 операнд	результат 4
0F2 : 001AH	5 операнд	результат 5
0F2 : 001CH	6 операнд	результат 6
0F2 : 001EH	7 операнд	результат 7

Работа с сопроцессором

Все математические функции работают с входными числами в 32 битовом формате с плавающей точкой.

Вычисления внутри функций осуществляется в 40-битовом формате с округлением до 32 разрядов в результате.

После выполнения команды и выдачи результата в регистры операндов происходит обнуление регистра комманд 0F2 : 0010h. Это обнуление происходит и при сбросе сопроцессора.

Последовательность действий:

Загрузить операнды
Загрузить код комманды
Фаза выполнения
Считывание результата
(по READY CS2 или по установке флага CC23ir)

Обращение к сопроцессору можно проводить с помощью команд непрерываемых последовательностей типа EXTS.

Пример:

```
\\ запись операндов
        exts    #F2h, #4        \\ уст. сегм. на F2.xxxx для 4 команд
        mov     12h, r2         \\ r2- операнд X младшая часть
        mov     12h, r3         \\ r3- операнд X старшая часть
        mov     14h, r4         \\ r4- операнд Y младшая часть
        mov     14h, r5         \\ r5- операнд Y старшая часть
\\ Код команды (функции) и чтение результата
        exts    #F2h, #2        \\ уст. сегм. на F2.xxxx для 2 команд
        mov     10h, r6         \\ r6- код комманды младшая часть
        mov     10h, zeros      \\ 0 - код комманды старшая часть
\\ Чтение результата  (BUSCON2=14BFh)
        exts    #F2h, #2        \\ уст. сегм. на F2.xxxx для 2 команд
        mov     r2,12h          \\ чтение результата младшая часть
        mov     r3,12h          \\ чтение результата старшая часть
```

Примечание: Для считывания результата использовать отдельную последовательность EXTS.

Для преобразования чисел используются функции Float, Fix

Float – Преобразование челочисленного знакового в плавающее

1234568 -> Float -> 1.234568e+6

Fix – Преобразование плавающего (вещественного) в знаковое целое

12.34568e+1 -> Fix -> 123
N Fix
Где N=1,..N

12.34568e+1 -> 100 fix -> 12345

Например: При вычислении Sin(x) имеем диапазон чисел от –1.0 до +1.0
При проведении опреации Fix диапазон будет состоять только из трех чисел: –1,0,+1

Для приведения диапазона используется N Fix
При задании N=32000 диапазон Sin будет от –32000 до 32000 с дисккретностью 1 бит

Эти функции существуют в одно-, двух- и трехоперандных вариантах

12345 -> x
12345 -> y
-> float(x,y) -> tang(x/y) -> atan(x) -> 10 fix(x) -> x=10

314159 -> x
10000 -> y
-> float(x,y) -> x/y (x=3.14159) —> cos(x)
-> 10000 fix(x) -> x=-100000

1570796 -> x
100000 -> y
-> float(x,y) -> x/y ->(x=1.570796) 3fase(x)
-> 100000 fix(x,y,z)
-> x=100000,y=-50000,z=-50000

Описание функций

Y=Cos(x) – 0 01h

Вычислить косинус вещественного числа X, заданного в радианах.

Входные величины – X=[6e-20,6e20] радиан
Регистр по адресу : f2.0012h
Выходные величины Y=[-1.0,1.0]
Регистр по адресу : f2.0012h

Y=Sin(x) – 0 02h

Вычислить синус вещественного числа X, заданного в радианах.

Входные величины – X=[6e-20,6e20] радиан
Регистр по адресу : f2.0012h
Выходные величины Y=[-1.0,1.0]
Регистр по адресу : f2.0012h

Y=Tan(x) – 0 03h

Вычислить тангенс вещественного числа X, заданного в радианах

Входные величины – X=[6e-20,6e20] радиан
Регистр по адресу : f2.0012h
Выходные величины Y= [tangensX]
Регистр по адресу : f2.0012h

Y=Exp(x) – 0 04h

Вычислить экспоненту вещественного числа X

Входные величины – X=[6e-20,6e20]
Регистр по адресу : f2.0012h
Выходные величины Y= [tangensX]
Регистр по адресу : f2.0012h

Z=x*y – 0 05h

Вычислить произведение двух вещественных чисел X и Y

Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0014h
Величина Z – регистр по адресу : f2.0012h после выполнения

Z=x+y – 0 06h

Вычислить сумму двух вещественных чисел X и Y

Результат Z – вешественное число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0014h
Величина Z – регистр по адресу : f2.0012h после выполнения

Z=x-y – 0 07h

Вычислить разность двух вещественных чисел X и Y

Результат Z – вещественное число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0014h
Величина Z – регистр по адресу : f2.0012h после выполнения

Z=x/y – 0 08h

Вычислить частное двух вещественных чисел X и Y

Результат Z – вещественное число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0014h
Величина Z – регистр по адресу : f2.0012h после выполнения

Y=log2(x) – 0 09h

Вычислить логарифм по основанию 2 вещественного числа X

Результат – вещественное число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0012h после выполнения

Y=log10(x) – 0 0Ah

Вычислить десятичный логарифм вещественного числа X

Результат – вещественное число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0012h после выполнения

Y=ln(x) – 0 0Bh

Вычислить натуральный логарифм вещественного числа X

Результат – вещественное число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0012h после выполнения

Y=sqrt(x) – 0 0Ch

Вычислить квадратный корень вещественного числа X

Результат – вещественное число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0012h после выполнения

Y=isqrt(x)=1/sqrt(x) – 0 0Dh

Вычислить инверсный квадратный корень вещественного числа X

Результат – вещественное число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0012h после выполнения

Z=x^y – 0 0Eh

Вычислить результат возведения в вещественную степень Y вещественного числа X

Результат – вещественное число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0014h
Величина Z – регистр по адресу : f2.0012h после выполнения

Y=1/x – 0 0Fh

Вычислить инверсию вещественного числа X

Результат – вещественное число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0012h после выполнения

Y=atan(x) – 0 10h

Вычислить арктангенс вещественного числа X

Результат в радианах – вещественное число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0012h после выполнения

Z=atan(x/y) – 0 11h

Вычислить арктангенс от частного вещественных чисел X и Y

Результат в радианах – вещественное число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0014h
Величина Z – регистр по адресу : f2.0012h после выполнения

Y=acos(x) – 0 12h

Вычислить арккосинус вещественного числа X в радианах

Результат – вещественное число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0012h после выполнения

Y=asin(x) – 0 13h

Вычислить арксинус вещественного числа X в радианах

Результат – вещественное число

Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0012h после выполнения

Z=logY(x) – 0 14h

Вычислить логарифм по вещественному основанию Y вещественного числа X

Результат – вещественное число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0014h
Величина Z – регистр по адресу : f2.0012h после выполнения

Y=fix(x) – 0 15h

Выделить целую часть вещественного числа X

Результат – целое знаковое число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0012h после выполнения
Множитель N – регистр по адресу : f2.0018h

X, Y=fix(x,y) – 0 16h

Выделить целую часть вещественных чисел X, Y

Результат – два целых знаковых числа
Величина x – регистр по адресу : f2.0012h
Величина y – регистр по адресу : f2.0014h
Величина X – регистр по адресу : f2.0012h после выполнения
Величина Y – регистр по адресу : f2.0014h после выполнения
Множитель N – регистр по адресу : f2.0018h

Y=float(x) – 0 17h

Преобразовать целое X в вещественное число Y

Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0012h после выполнения

(X,Y)=float(x,y) – 0 18h

Преобразовать целые x и y в вещественные числа X и Y

Величина x – регистр по адресу : f2.0012h
Величина y – регистр по адресу : f2.0014h
Величина X – регистр по адресу : f2.0012h после выполнения
Величина Y – регистр по адресу : f2.0014h после выполнения

Y=mod(x) – 0 19h

Вычислить модуль вещественного числа X

Результат – модуль вещественного числа
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0012h после выполнения

Z=mod(x+y) – 0 1Ch

Вычислить модуль суммы вещественных чисел X и Y

Результат – модуль вещественного числа
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0014h
Величина Z – регистр по адресу : f2.0012h после выполнения

Z=mod(x-y) – 0 1Dh

Вычислить модуль разности вещественных чисел X и Y

Результат – модуль вещественного числа
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0014h
Величина Z – регистр по адресу : f2.0012h после выполнения

Z=(x+y)/2 – 0 1Eh

Вычислить полусумму вещественных чисел X и Y

Результат – вещественное число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0014h
Величина Z – регистр по адресу : f2.0012h после выполнения

Y=neg(x) – 0 1Fh

Сменить знак вещественного числа X

Результат – вещественное число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0012h после выполнения

Z=comp(x,y) – 020h

Сравнить два вещественных числа X и Y

Результат – целое число:
Если x>y z=-1
Если x<y z=1
Если x=y z=0000 0000h

Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0014h
Величина Z – регистр по адресу : f2.0012h после выполнения

Z=max(x,y) – 0 21h

Выбрать максимальное из двух вещественных чисел X и Y

Результат – максимальное вещественное число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0014h
Величина Z – регистр по адресу : f2.0012h после выполнения

Z=min(x,y) – 0 22h

Выбрать минимальное из двух вещественных чисел X и Y

Результат – минимальное вещественное число
Величина X – регистр по адресу : f2.0012h
Величина Y – регистр по адресу : f2.0014h
Величина Z – регистр по адресу : f2.0012h после выполнения

ASCII>FLT – 0 1Ah

Преобразовать байтовую последовательность кодов ASCII символов в вещественное число

Заменить символами регистры по 16 бит (код символа в младших 8 битах)

	младш.	*старш.*
f2.0012	2	1
f2.0014	4	3
f2.0016	6	5
f2.0018	8	7
f2.001A	10	9
f2.001C	12	11
f2.001E	14	13

Код команды в f2.0010h
Результат в f2.0012h

FLT>ASCII – 0 1Bh

Преобразовать вещественное число в байтовую последовательность кодов ASCII символов

Считывание результата из регистра по 16 бит (код символа в младших 8 битах)

	младш.	*старш.*
f2.0012	2	1
f2.0014	4	3
f2.0016	6	5
f2.0018	8	7
f2.001A	10	9
f2.001C	12	11
f2.001E	14	13

Операнд в f2.0012h
Код команды в f2.0010h

Fase2(x) – 0 24h (двухфазный генератор)

Вычислить синус и косинус X (радианы)

Результат – вещественные числа
Величина X – регистр по адресу : f2.0012h
Величина sin(x) – регистр по адресу : f2.0012h после выполнения
Величина cos(x) – регистр по адресу : f2.0014h после выполнения

Fase3(x) – 0 23h (трехфазный генератор)

Вычислить синус вещественной величины X (радианы) с тремя фазами 0, 120, -120 градусов

Результат – вещественные числа
Величина X – регистр по адресу : f2.0012h
Величина sin(x) – регистр по адресу : f2.0012h после выполнения
Величина sin[(x)+120] – регистр по адресу : f2.0014h после выполнения
Величина sin[(x)-120] – регистр по адресу : f2.0016h после выполнения

X, Y, Z=fix(x,y,z) –0 25h

Выделить целую часть вещественных чисел X, Y , Z

Результат – три целых знаковых числа
Величина x – регистр по адресу : f2.0012h
Величина y – регистр по адресу : f2.0014h
Величина z – регистр по адресу : f2.0016h
Величина X – регистр по адресу : f2.0012h после выполнения
Величина Y – регистр по адресу : f2.0014h после выполнения
Величина Z – регистр по адресу : f2.0016h после выполнения
Множитель N – регистр по адресу : f2.0018h

Примеры программ

Пример инициализации:

```
InitCor :
        einit
        mov     addrsel2, #F007h        \\ инициализация CS2
        mov     buscon2, #4BFh
        exts    #F0h, #3                \\ сброс сопроцессора
        mov     ch, ones
        mov     Ah, ones
        mov     8h, ones
        exts    #F0h, #2
        mov     8h, zeros
        mov     8h, ones
        ( 5 ms ождать, загрузка сопроцессора)
        exts    #f0h,# 1                \\ подключение сопроцессора к C167
        mov     Ah,zeros
        mov     buscon2, #14BFh         \\ при работе с READY
        mov     ccm5, #3000h
        extr    #1
        blc     cc23ir
    // сопроцессор готов к работе
```

Пример для двухоперандной функции:

```
        pop     r2                      \\ xl
        pop     r3                      \\ xh
        pop     r4                      \\ yl
        pop     r5                      \\ yh
        pop     r6                      \\ tg2(x/y)
        exts    #F2h, #4                \\
        mov     12h, r2                 \\ 1 операнд младшая часть
        mov     12h, r3                 \\ 1 операнд старшая часть
        mov     14h, r4                 \\ 2 операнд младшая часть
        mov     14h, r5                 \\ 2 операнд старшая часть
        exts    #F2h,#4                 \\
        mov     10h, r6                 \\ код команды младшая часть
        mov     10h, zeros              \\ код команды старшая часть
\\ Чтение результата
        exts    #F2h, #2                \\
        mov     r2, 12h                 \\ чтение операнда младшая часть
        mov     r3, 12h                 \\ чтение операнда старшая часть
        push    r2
        push    r3
```

Пример для одноперандной функции:

```
                pop     r2              \\ opl
                pop     r3              \\ oph
                pop     r4              \\ cos(x)
                exts    #F2h, #4        \\
                mov     12h, r2         \\ 1 операнд младшая часть
                mov     12h, r3         \\ 1 операнд старшая часть
                mov     10h, r4         \\ код команды младшая часть
                mov     10h, zeros      \\ код команды старшая часть
        \\ Чтение результата
                exts    #F2h,#2         \\
                mov     r2, 12h         \\ чтение операнда младшая часть
                mov     r3, 12h         \\ чтение операнда старшая часть
                push    r3
                push    r2
```

Пример использования предыдущего значения

A()=((x1*y1)+y2)*y3+y4

Compute:

```
                mov     r2, opl
                mov     r3, oph
                mov     r4, #10h
                exts    #F2h, #4        \\
                mov     14h, r2         \\ y1 операнд младшая часть
                mov     14h, r3         \\ y1 операнд старшая часть
                mov     [r4], #17h      \\ код команды Float(x,y) младшая часть
                mov     10h, zeros      \\                старшая часть
                exts    #F2h, #2        \\
                mov     [r4], #5h       \\ (x1*y1) код команды (*) младшая часть
                mov     10h, zeros      \\           старшая часть
                mov     r2, opl
                mov     r3, oph
                exts    #F2h,#4         \\
                mov     14h,r2          \\ y2 операнд младшая часть
                mov     14h,r3          \\ y2 операнд старшая часть
                mov     [r4], #6h       \\ (+y2)  код команды (+) младшая часть
                mov     10h, zeros      \\           старшая часть
                exts    #F2h,#4         \\
                mov     14h,r2          \\ y2 операнд младшая часть
                mov     14h,r3          \\ y2 операнд старшая часть
                mov     [r4], #5h       \\ (*y3)  код команды (*) младшая часть
                mov     10h, zeros      \\           старшая часть
                exts    #F2h,#4         \\
                mov     14h,r2          \\ y2 операнд младшая часть
                mov     14h,r3          \\ y2 операнд старшая часть
                mov     [r4], #6h       \\ (+y4)  код команды (+) младшая часть
                mov     10h, zeros      \\           старшая часть
                exts    #F2h,#4         \\
                mov     [r4], #15h      \\ (Fix)  код команды  младшая часть
                mov     10h, zeros      \\           старшая часть
                mov     r2, 12h         \\ чтение A младшая часть
                mov     r3, 12h         \\ чтение A старшая часть
                pop     r3
                pop     r2
                rets
```

10. Работа в отладочном режиме

С контроллером MF167 поставляется программный инструмент разработки – полноэкранный отладчик, который позволяет выполнять отладку загруженного исполняемого кода. Программа, предназначенная для отладки, должна быть предварительно скомпилирована в Intel hex.83 коде или двоичном формате.

Отладчик загружается в основное ОЗУ контроллера.

Для запуска контроллера в отладочном режиме необходимо:

- подключить кабелем разъем J9 контроллера к последовательному порту RS232 PC-совместимого компьютера;
- установить перемычку на разъем J8;
- подать напряжение питания контроллера;
- запустить программу отладчика SFD7 (входит в комплект поставки);
- сконфигурировать имеющееся на плате ОЗУ в диапазон младших адресов с помощью системных регистров ADDRSELx:

Например:
SYSCON = 0180h BUSCON0 = 04BFh;
ADDRSEL1 = 0005h BUSCON1 = 04BFh;
ADDRSEL2 = 0000h BUSCON2 = 0000h;
ADDRSEL3 = 0000h BUSCON3 = 0000h;
ADDRSEL4 = 0FF04h BUSCON4 = 04BDh
Load Address = 00:D800h.

Системные регистры ADDRSEL2, BUSCON2 в отладчике можно не устанавливать, однако, для доступа к системным ресурсам необходимо запрограммировать эти регистры в пользовательской программе начальной инициализации.

11. Программирование микросхем FLASH памяти

Схема контроллера MF167 предусматривает возможность программирования установленных на плате микросхем Flash памяти. Программа, предназначенная для программирования во Flash память, должна быть предварительно скомпилирована в Intel hex.83 коде.

Для программирования Flash памяти необходимо:

- подключить кабелем разъем J9 контроллера к последовательному порту RS232 PC-совместимого компьютера;
- установить перемычку на разъем J8;
- подать питание на контроллер;
- запустить программу отладчика SFD7 (входит в комплект поставки);
- установить системные регистры отладчика SFD7:

SYSCON = 0180h; BUSCON0 = 04BFh;
ADDRSEL1 = 1805h BUSCON1 = 04BFh;
ADDRSEL2 = 0000h BUSCON2 = 0000h;
ADDRSEL3 = 0000h BUSCON3 = 0000h;
ADDRSEL4 = 0000h BUSCON4 = 0000h;
S0RIC = 0037h;
Load Address 18:F000h
VPP bit - - . -

12. Старт контроллера из ПЗУ

Для старта контроллера из ПЗУ необходимо:

- удалить перемычку с разъема J8;
- подать питание на контроллер или произвести сброс контроллера.

13. Сброс контроллера

Для сброса предназначен переключатель J11. Сброс производится замыканием контактов между собой.

Переключатель J11.

Номер контакта	*Сигнал*
1	Вход сброса
2	общий (GND)

14. Питание контроллера

Контроллер питается от внешнего источника постоянного тока +5 В ±5 % с типовым потреблением 300 мА. Плюсовой вывод источника подключается к контакту 48 разъема J6(J7), минусовой вывод источника подключается к контакту 50 разъема J6(J7).

15. Подключение батареи супервизора

Для сохранения данных в ОЗУ контроллера и работоспособности часового таймера при отключении основного источника питания к разъему J10 может быть подключена внешняя батарея напряжением 3.6 ± 0.6 В. Типовой ток потребления от батареи 40 мкА при отсутствии основного напряжения питания.

Разъем J10.

Номер контакта	*Сигнал*
1	Плюс батареи
2	Минус батареи

16. Внешние разъемы и переключатели

Расположение и назначение разъемов и переключателей на плате контроллера M167-2 представлено на рисунке 7.

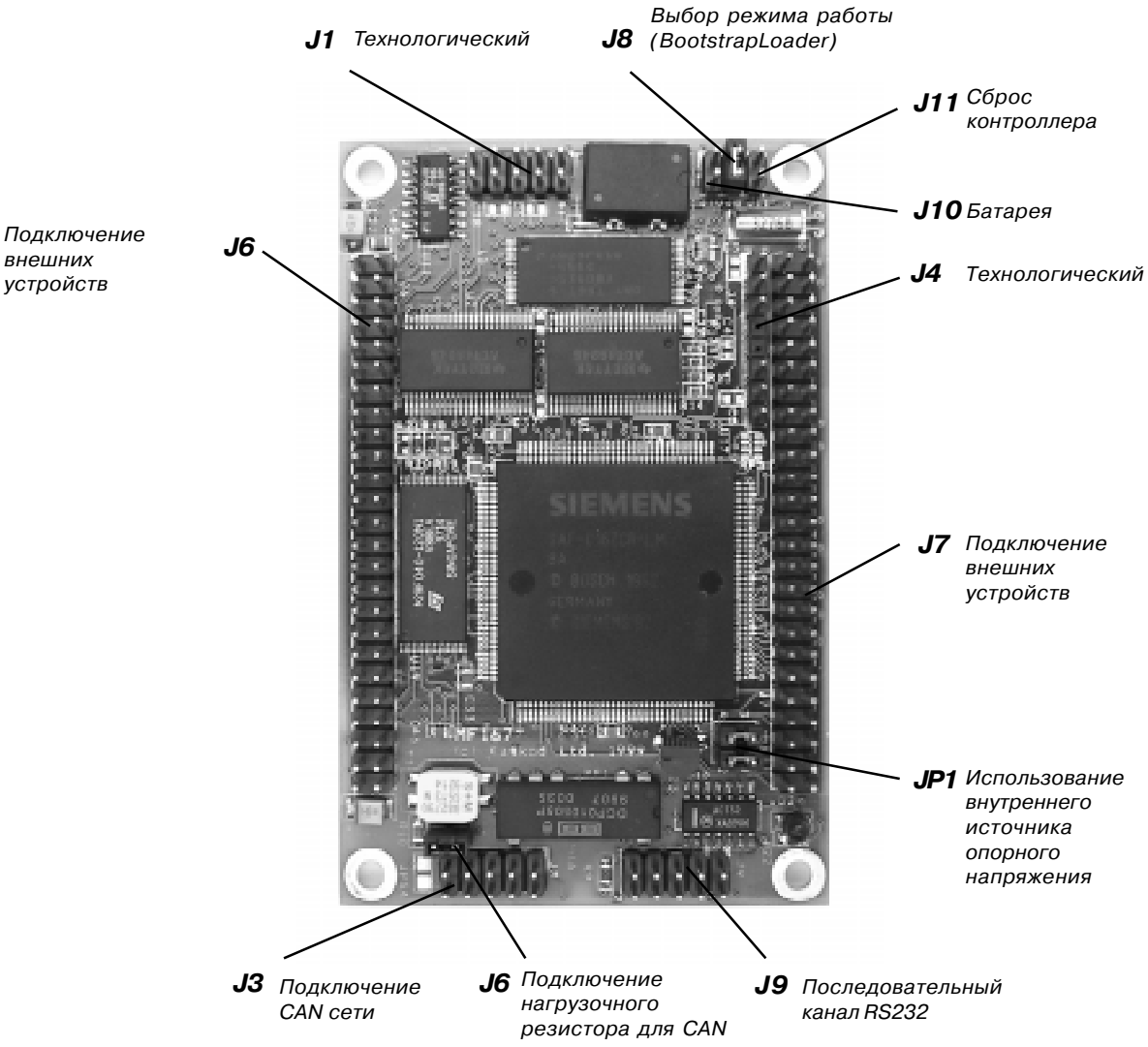


Рис. 7

Подключение внешних цепей к контроллеру осуществляется с помощью разъемов. Ниже приведены разъемы, переключатели и перемычки контроллера.

Разъем J7.

Номер контакта	*Сигнал*	*Номер контакта*	*Сигнал*
1	P5.0 Ain 1	2	P5.1 Ain 2
3	P5.2 Ain3	4	P5.3 Ain 4
5	P5.3 Ain 5	6	P5.5 Ain 6
7	P5.6 Ain 7	8	P5.7 Ain 8
9	P5.8 Ain 9	10	P5.9 Ain 10
11	P5.10 Ain 11	12	P5.11 Ain 12
13	P5.12 Ain 13	14	P5.13 Ain 14
15	P5.14 Ain 15	16	P5.15 Ain 16
17	GND (VAREF)	18	GND (VAREF)
19	VAREF	20	NMI
21	GND (VAREF)	22	GND (VAREF)
23	P2.0	24	P2.1
25	P2.2	26	P2.3
27	P2.4	28	P2.5
29	P2.6	30	P2.7
31	P2.8	32	P2.9
33	P2.10	34	P2.11
35	P2.12	36	P2.13
37	P2.14	38	P2.15
39	P3.0	40	P3.1
41	P3.2	42	P3.3
43	P3.4	44	P3.5
45	P3.8	46	P3.7
47	nc	48	+5v (VCC)
49	nc	50	GND

Примечание:

Ain x – Аналоговый вход (номер канала АЦП микроконтроллера). (Порт P5 микроконтроллера).
nc – Контакт свободен.
GND(VAREF) – Аналоговая земля АЦП микроконтроллера.
GND – Цифровая земля (общий провод).
+5v (VCC) – Напряжение питания контроллера.
NMI – Немаскируемое прерывание.
P2.0 - P2.15 – Порт P2 микроконтроллера.
P3.0 - P3.7 – Порт P3 микроконтроллера.
VAREF – Опорное напряжение АЦП микроконтроллера.
GND(VAREF) и GND могут быть соединены на плате перемычкой JP1 (3 и 4).

Разъем J6.

Номер контакта	*Сигнал*	*Номер контакта*	*Сигнал*
1	An 0	2	An 1
3	An 2	4	An 3
5	An 4	6	An 5
7	An 6	8	An 7
9	Ref	10	aGND
11	aGND	12	aGND
13	nc	14	nc

15	nc	16	nc
17	nc	18	nc
19	nc	20	nc
21	nc	22	nc
23	nc	24	nc
25	P6.3	26	P6.4
27	P7.0	28	P7.1
29	P7.2	30	P7.3
31	P7.4	32	P7.5
33	P7.6	34	P7.7
35	P8.0	36	P8.1
37	P8.2	38	P8.3
39	P8.4	40	P8.5
41	P8.6	42	P8.7
43	P3.13	44	P3.9
45	P3.15	46	P3.8
47	nc	48	+5v (VCC)
49	nc	50	GND

Примечание:

An x	–	Аналоговый вход (номер канала последовательного АЦП)
Ref	–	Опорное напряжение АЦП
nc	–	Контакт свободный
aGND	–	Аналоговая земля последовательного АЦП
GND	–	Цифровая земля (общий провод)
+5v (VCC)	–	Напряжение питания контроллера
P6.3 - P6.4	–	Порт P6 микроконтроллера
P7.0 - P7.7	–	Порт P7 микроконтроллера
P8.0 - P8.7	–	Порт P8 микроконтроллера
P3.8, P3.9, P3.13, P3.15	–	Порт P3 микроконтроллера
P8.7	–	Прерывание от сопроцессора (выход)

Разъем J3.

Разъем J3 предназначен для подключения контроллера к CAN сети.

Номер контакта	*Сигнал*
1	не используется
2	не используется
3	BUS.L
4	BUS.H
5	CAN
6, 7, 8, 9, 10	не используется

Разъем J9.

Разъем J9 предназначен для подключения кабеля RS232 к последовательному порту 0 микроконтроллера C167/ST10R167.

Разъем J9

1,2,7	–	соединены между собой.
3	–	RXD (прием данных в контроллер).
4,6	–	соединены между собой.
5	–	TXD (передача данных из контроллера).
8	–	свободный.
9,10	–	общий (земля).

Разъем J11.

Разъем предназначен для подключения кнопки сброса контроллера (при необходимости). Сброс производится замыканием контактов разъема.

Номер контакта	*Сигнал*
1	Вход сброса
2	общий (GND)

Разъем J10.

Разъем предназначен для подключения батареи супервизора.

Номер контакта	*Сигнал*
1	Плюс батареи
2	Минус батареи

Разъем J8.

Разъем J8 предназначен для выбора режима старта.

Перемычка установлена – отладочный режим.

Перемычка снята – режим старта программы из ПЗУ.

Разъем J4.

Разъем J4 технологический.

Разъем JP1.

Разъем JP1 предназначен для подключения внутреннего источника опорного напряжения величиной 2,5 В (перемычка между контактами 1 и 2 установлена). Если перемычка между контактами 1 и 2 не установлена, используется внешний источник опорного напряжения величиной от 0 до 5 В. Перемычка между контактами 3 и 4 установлена – соединены цифровая земля и аналоговая земля.

Схема соединения MF167 с PC компьютером для связи по RS232 показана на рисунке 8.

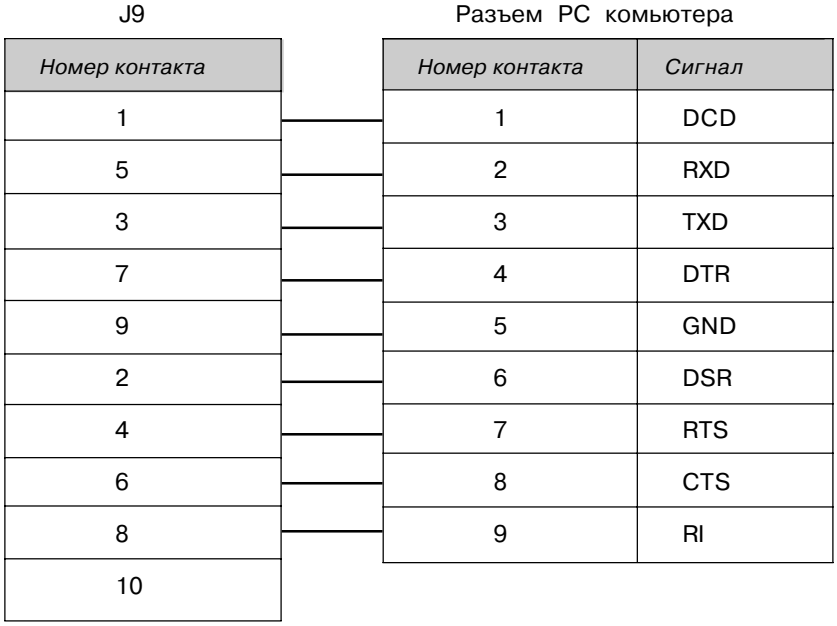


Рис. 8

17. Комплект поставки

1. MF167
2. Отладчик
3. Кроссассемблер
4. Руководство пользователя
5. Примеры программ для работы с периферийными устройтсвами

18. Варианты исполнения контроллера

Контроллер поставляется в следующих модификациях:

1. MF167	–	полный вариант, диапазон рабочих температур: от 0˚С до +70˚С.
Суффикс -EXT	–	диапазон рабочих температур: от -40˚С до +85˚С по всем контроллерам.
Суффикс -KIT	–	все ответные части разъемов и кабель RS232.
Суффикс -Flash	–	Flash память 1024 Кб.

Разъёмы:

Разъёмы J9, J3, J6, J7 прямые или угловые, или прямые вниз (по требованию заказчика).

Замечание: При заказе контроллеров необходимо соблюдать обозначения изделий данные выше. Расположение выходных разъемов оговаривается отдельно (прямые, угловые вниз, вверх).

19. Габаритные и установочные размеры

Габариты и установочные размеры платы показаны на рисунке 9.

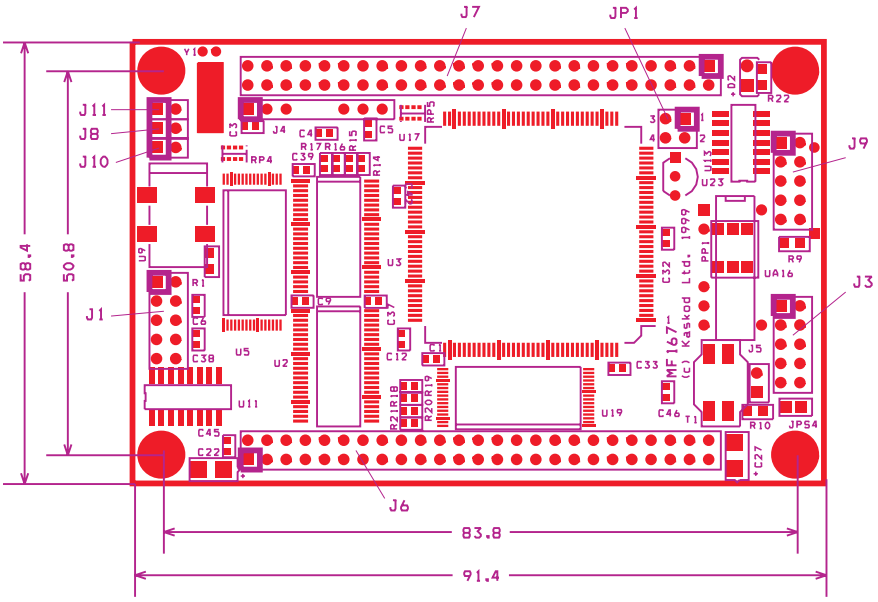
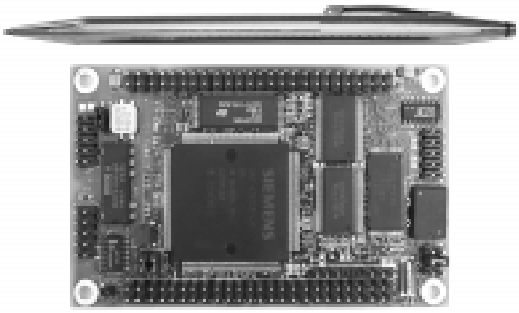


Рис. 9

Размеры приведены в миллиметрах.



Приложение П. Функциональная схема контроллера