

# **Модуль аналогового ввода-вывода**

## **КМ1604**

### **Руководство пользователя**



# Содержание

	страница
Содержание.....	3
1. Принятые сокращения .....	4
2. Назначение .....	5
3. Технические характеристики .....	6
4. Подключение модуля .....	7
5. Структурная схема модуля .....	8
6. Выбор базового адреса доступа к модулю.....	9
7. Установка номеров запросов прерывания .....	11
8. Доступ к модулю со стороны шины PC/104 (ISA) .....	12
9. АЦП .....	15
10. ЦАП.....	22
11. Сброс модуля .....	27
12. Питание модуля.....	27
13. Внешние разъемы и переключатели.....	28
14. Условия эксплуатации и хранения.....	34
15. Варианты исполнения модуля.....	35
16. Комплект поставки и маркировка модуля .....	36
17. Габаритные и установочные размеры. ....	37
Приложение.....	38

# 1. Принятые сокращения

модуль	–	Модуль KM1604.
АЦП	–	Аналого-цифровой преобразователь.
ЦАП	–	Цифро-аналоговый преобразователь.
CPU (ЦПУ)	–	Central Processing Unit (Центральное процессорное устройство).
CS	–	Chip Select (выбор микросхемы).
GPT	–	General Purpose Timer unit (Блок таймеров).
GPR	–	General Purpose Register (Регистры общего назначения).
nc	–	Свободный контакт (контакт не подключен).
GND	–	Цифровая земля (общий провод питания).
AGND	–	Аналоговая земля.
VCC	–	Напряжение питания +5 Вольт.
RESDRV	–	Сигнал “Сброс” модуля.
IRQn	–	Прерывание номер n
WR	–	Сигнал записи.
RD	–	Сигнал чтения.
IOW	–	Сигнал “Запись в порт”.
IOR	–	Сигнал “Чтение из порта”.
AEN	–	Сигнал защелки адреса.
BHE	–	Разрешение старшего байта.
IOCS16		I/O Chip Select 16
SAx	–	Бит адреса x, где x=0-19.
SDy	–	Бит данных y, где y=0-15.
лог.1	–	Уровень логической единицы.
лог.0	–	Уровень логического нуля.
CLK	–	Тактовые сигналы модуля

## 2. Назначение

Модуль KM1604 выполнен в стандарте PC/104 (IEEE-P996.1).

Плата предназначена для работы в составе систем поддерживающих формат шины PC/104.

Модуль KM1604 предназначен для для работы:

- с аналоговыми датчиками тока 0 – 24 мА, и/или 0 – 20 мА, и/или 4 – 20 мА,
- с устройствами имеющими аналоговые токовые входы 4 – 20 мА, 0 – 20 мА, 0 – 24 мА,
- с устройствами имеющими аналоговые входы 0...+5 Вольт, 0...+10 Вольт, -5...+5 Вольт, -10...+10 Вольт.

Модуль KM1604 может быть использован для построения:

- систем сбора и обработки информации,
- следящих систем,
- систем управления и синхронизации энергетических объектов,
- распределенных систем управления и т.д.

Общий вид модуля KM1604 представлен на рисунке 1.

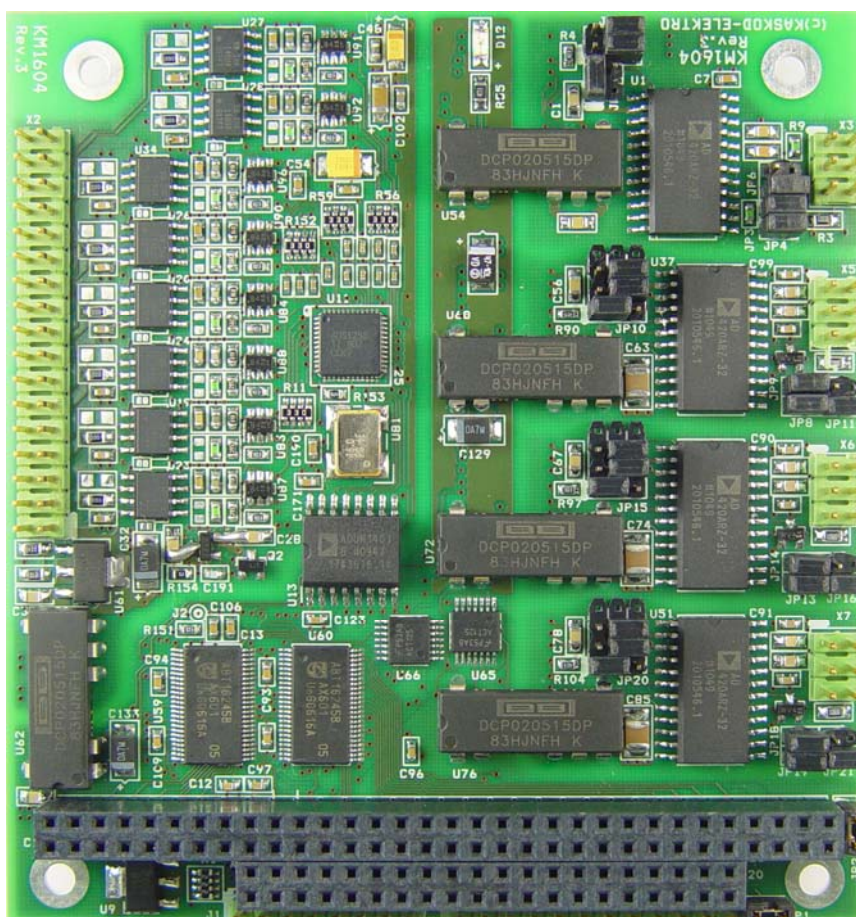


Рис. 1. Модуль KM1604

### 3. Технические характеристики

- Модуль аналогового ввода-вывода в формате PC/104.
- АЦП: 24 бит, 16 каналов:
  - Гальваноизолированный 16-канальный 24-разрядный АЦП, имеющий 16 аналоговых входов тока от 0 до 24 миллиампер с групповой изоляцией.
  - Скорость преобразования 8 микросекунд для одного канала
  - Входное сопротивление 249 Ом  $\pm 1,5\%$ .

*Замечание:*

По отдельному запросу возможно изменение входов тока на входы напряжения с диапазоном от 0 до 5 Вольт или от 0 до 10 Вольт.

- ЦАП: 16 бит, 4 канала:
  - 4 канала 16-разрядных цифро-аналоговых преобразователей (четыре индивидуально изолированных ЦАП).
  - Выходное напряжение – от минус 10 Вольт до 10 Вольт, или от минус 5 Вольт до 5 Вольт, или от 0 до 5 Вольт, или от 0 до 10 Вольт.
  - Выходной ток – от 4 до 20 миллиампер, или от 0 до 20 миллиампер, или от 0 до 24 миллиампер.
  - Скорость преобразования ЦАП 3 миллисекунды.
  - Режим выхода устанавливается переключателями на плате модуля.
- Возможность установки номера используемого прерывания.
- Возможность установки базового адреса.
- Сквозной разъем PC/104.
- 16-разрядный шинный интерфейс PC/104, позволяющий подключать различные модули в формате PC/104.
- Через переходную плату возможно подключение к шине ISA.
- Размер платы 90x96 мм.
- Напряжение питания 5,0  $\pm$  0,25 Вольт.
- Потребляемый ток - 1,2 Ампер (в режиме токовых выходов).
- Стартовый ток - 3,0 Ампер.
- Напряжение питания подключается через разъемы X1, X4 (PC/104).
- Расширенный диапазон рабочих температур: минус 40°C - +85°C.
- Расширенный диапазон рабочих температур: минус 55°C - +85°C.
- Вес 0,1 кг.

Примеры программ работы с модулем.

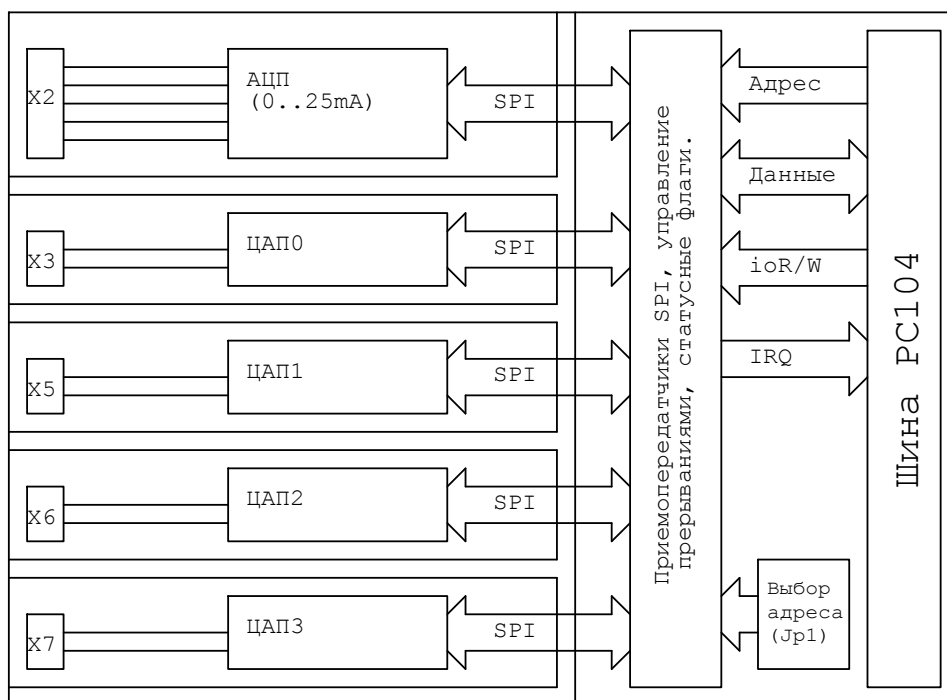
## 4. Подключение модуля

### Общие замечания по установке

- Сохраняйте модуль в антистатическом пакете до установки в систему!
  - Перед работой с модулем снимите с себя заряд статического электричества, соблюдая меры электрической безопасности.
  - Доставая модуль из пакета, старайтесь не дотрагиваться до выводов и компонентов.
  - Используйте антистатические маты и заземления.
  - Все изменения соединений при работе с модулем производите при отключенном питании.
1. Выключите аппаратуру.
  2. Снимите с себя заряд статического электричества, соблюдая меры электрической безопасности.
  3. Достаньте модуль из антистатического пакета.
  4. Перед установкой платы проверьте правильность установки переключателей.
  5. Удерживая модуль за края, установите его в систему или поместите на антистатическую поверхность.
  6. Подключите необходимые кабели. Убедитесь в правильной полярности соединений.
  7. Включите аппаратуру.

Модуль готов к работе.

## 5. Структурная схема модуля



Структурная схема платы KM1604

Рис. 2 Структурная схема модуля KM1604.

Примечания: пунктиром показаны изоляционные барьеры на плате.

На шине PC/104(ISA) плата представляется тремя регистрами:

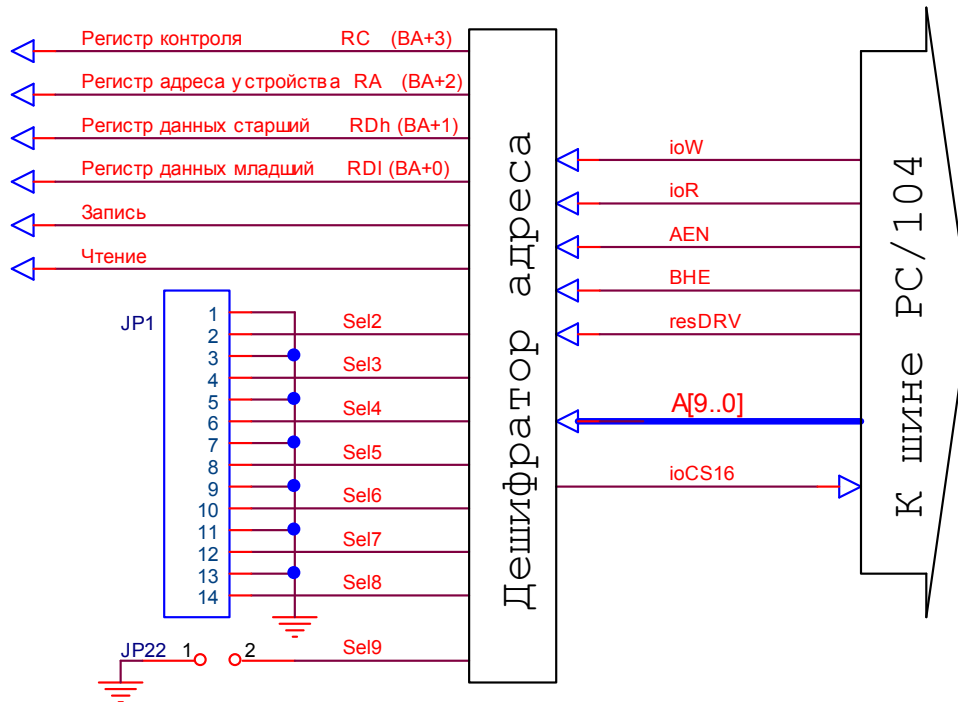
- **регистром данных,**
- **регистром адреса,**
- **регистром контроля.**



## 6. Выбор базового адреса доступа к модулю

Модуль аналогового ввода-вывода KM1604 подключается по шине PC/104 (ISA) в область портового пространства PC-совместимого устройства.

Для выбора базового адреса необходимо установить переключатели на переключателе JP1.



Этот переключатель позволяет задавать базовый адрес в диапазоне от 0h (0d) до 3FCh (1020d).

В этом диапазоне адресов могут быть и другие устройства (например, адреса адаптера джойстика = 200h - 207h; 320h - 32Fh XT Hard Disk и т.п.)

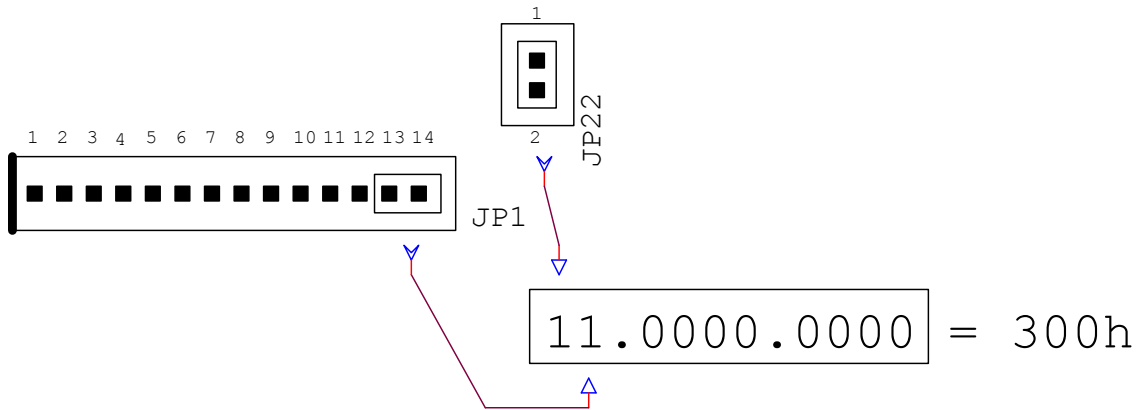
Поэтому, наиболее часто встречающейся ошибкой при первоначальной установке модуля является неправильная установка адреса и вызванный этим конфликт на шине PC.

Каждая пара контактов переключателя обеспечивает сравнение одного адресного сигнала, при этом замкнутому состоянию контактов (переключатель установлен), соответствует уровень логической единицы. Адресные сигналы начиная от A10 и больше не проверяются и не оказывают влияния на дешифрацию. Поэтому, если вы установили адрес платы, например 300h, то вы можете обращаться к ней и по адресу 700h, если только архитектура вашего ведущего модуля не блокирует доступ по таким адресам. Адреса A0 и A1 используются платой для разделения обращений к устройствам модуля, поэтому в расчете базового адреса платы приняты для простоты равными 0.

Выводы	JP1	Примечание
переключатель	1 - 2	устанавливает адрес A2
переключатель	3 - 4	устанавливает адрес A3
переключатель	5 - 6	устанавливает адрес A4
переключатель	7 - 8	устанавливает адрес A5
переключатель	9 - 10	устанавливает адрес A6
переключатель	11 - 12	устанавливает адрес A7
переключатель	13 - 14	устанавливает адрес A8
Выводы	JP22	Примечание
переключатель	1 - 2	устанавливает адрес A9

**Примеры установки базового адреса.**

Например мы хотим установить базовый адрес для платы равный 300h (0x300 или 768) в зависимости от используемой системы программирования. Для этого нам надо перевести заданный адрес в двоичное представление: 300h => 11.000.0000b , и, начиная со старшей части, устанавливать переключатели слева направо в те позиции, которым соответствуют единицы получившегося двоичного числа:



**Замечание:**

Максимальное количество одновременно подключенных плат на шине ограничивается свободным адресным пространством портов ввода-вывода и нагрузочной способностью шины вашего РС совместимого устройства. Использование одинаковых адресов для плат, установленных в одно устройство не допускается!

## 7. Установка номеров запросов прерывания

Устройства, установленные на модуле, могут генерировать запрос на прерывание:

- по завершению преобразования АЦП,
- по завершению передачи данных в АЦП,
- по завершению передачи данных в ЦАП.

Выбор вектора прерывания осуществляется установкой битов SellRQ3, SellRQ2, SellRQ1 и SellRQ0 в регистре контроля RC.

Номер выбранного прерывания определяется кодом, которому соответствует двоичное представление битов: 0011 - используется вход запроса №3, 0100 - №4, ..., 1111 - №15.

Прерывания с номерами 3, 4, 5, 6, 7, 9, 10, 11, 12, 14, 15 подключены к шине и могут быть использованы. Установка прерываний с номерами 0, 1, 2, 8, 13 равносильна запрету прерываний, поэтому отдельный бит запрета прерываний отсутствует. Плата не может разделять прерывания с другими платами, поэтому при выборе номера прерывания следует соблюдать осторожность для исключения конфликтов на шине.

Так как АЦП может работать в режиме автоматической конверсии данных, то в этом режиме после получения данных от АЦП также будет сгенерировано прерывание.

Прерывания запрещаются и разрешаются индивидуально в регистрах управления соответствующих АЦП и ЦАП, биты SellRQ3, SellRQ2, SellRQ1 и SellRQ0 определяют только канал запроса прерывания шины ISA и могут быть использованы для глобального запрещения прерываний.

Разделение прерываний с другими модулями не предусмотрено.

Снятие запроса прерывания происходит при чтении любого регистра, относящегося к соответствующему устройству. Или можно очистить флаги прерывания, сбрасывая биты разрешения прерываний в регистрах управления соответствующего устройства.

## 8. Доступ к модулю со стороны шины PC/104 (ISA)

На шине PC/104 (ISA) модуль представлен:

- **регистром данных RD**
- **регистром адреса RA.**
- **регистром контроля RC**

Регистр данных. Его адрес равен базовому адресу, устанавливаемому на модуле с помощью перемычек.

Регистр адреса. Его адрес равен базовому адресу +2. Определяет устройство, подключенное к регистру данных. Допускает одновременное использование с регистром контроля.

Регистр контроля. Его адрес равен базовому адресу +3

Представление регистров на шине ISA.

<b>регистр данных RD</b> (старший байт)	<b>регистр данных RD</b> (младший байт)
<b>регистр контроля RC</b>	<b>регистр адреса RA</b>

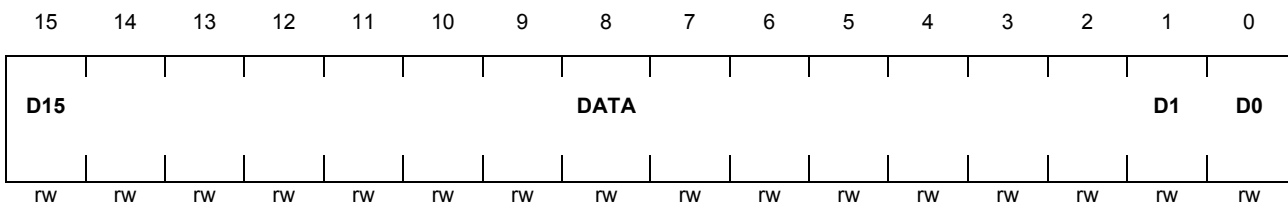
### Регистр данных RD

Регистр данных RD с адресом равным базовому адресу используется для обмена данными.

Его адрес устанавливается на модуле с помощью перемычек на переключателе JP1.

#### RD (БА)

Значение после сброса: **0000h**



Бит	Функция
<b>DATA</b>	Запись и чтение данных
<b>D0...D15</b>	

**D15** – Старший разряд данных **D0** – Младший разряд данных

Обращения к регистру данных — 16-битовые.

Регистр контроля предназначен контроля состояний АЦП и ЦАП, разрешения и выбора номера прерываний. Регистр контроля RC используется для управления устройствами модуля.

**Регистр контроля RC****RC (БА + 3)**                      Значение после сброса: **0000h**

7      6      5      4      3      2      1      0

IRQ	Busy DAC	Busy ADC	DRDY	Sel IRQ3	Sel IRQ2	Sel IRQ1	Sel IRQ0
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Бит	Функция
<b>Sel IRQ0</b> , <b>Sel IRQ1</b> , <b>Sel IRQ2</b> , <b>Sel IRQ3</b>	Используются для выбора соответствующих входов прерывания. Номер выбранного прерывания определяется кодом, которому соответствует двоичное представление битов: 0011 - используется вход запроса №3, 0100 - №4, ... 1111 - №15. Прерывания с номерами 3,4,5,6,7,9,10,11,12,14,15 подключены к шине и могут быть использованы. Установка прерываний с номерами 0, 1, 2, 8, 13 равносильна запрету прерываний, поэтому отдельный бит запрета прерываний отсутствует. Модуль не может разделять прерывания с другими Модулями, поэтому при выборе номера прерывания следует соблюдать осторожность для исключения конфликтов на шине.
<b>DRDY</b>	Сигнал DRDY с АЦП. Готовность данных.
<b>BusyADC</b>	Занятость канала обмена с АЦП. При чтении 0 канал готов к приему команды.
<b>BusyDAC</b>	Занятость канала обмена с ЦАП. При чтении 0 канал готов к приему команды.
<b>IRQ</b>	Установка этого бита в 1 сообщает о наличии необслуженного запроса прерывания.

**Регистр адреса RA**

Регистр адреса RA используется для выбора регистра подключенного к регистру данных.

**RA (БА + 2)**                      Значение после сброса: **0000h**

7      6      5      4      3      2      1      0

ADDR							
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Бит	Функция
<b>ADDR</b>	Адрес внутреннего устройства модуля (см. таблицу 1)

Возможно совместное использование регистра контроля и регистра адреса.

**RCRA (БА + 2)**                      Значение после сброса: **0000h**

15      14      13      12      11      10      9      8      7      6      5      4      3      2      1      0

IRQ	Busy DAC	Busy ADC	DRDY	Sel IRQ3	Sel IRQ2	Sel IRQ1	Sel IRQ0	ADDR								
r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w

Распределение устройств в зависимости от содержимого регистра адреса приведено в таблице 1

Таблица 1

ADDR	Активное устройство в регистре данных	
000h	Прием из АЦП байты 9-8	
001h	Прием из АЦП байты 7-6	
002h	Прием из АЦП байты 5-4	
003h	Прием из АЦП байты 3-2	
004h	Прием из АЦП байты 1-0	
005h	Резерв (не использовать)	
006h	Резерв (не использовать)	
007h	Резерв (не использовать)	
008h	Резерв (не использовать)	
009h	Резерв (не использовать)	
00Ah	Байты данных 8-7	
00Bh	Байты данных 6-5	
00Ch	Байты данных 4-3	
00Dh	Байты данных 2-1	
00Eh	Командный байт + байт данных 0	
00Fh	Регистр управления АЦП	
010h	Регистр данных ЦАП	
011h	Регистр управления ЦАП	
012h...0FEh	Резерв (не использовать)	
0FFh	Идентификатор платы 1101h	

## Идентификатор модуля

Идентификатор модуля предназначен для проверки типа установленного модуля и его версии. Тип модуля определяется старшим байтом и для этого модуля всегда 10h. Номер версии прошивки записан в младшем байте и может быть от 1...0FFh.

## 9. АЦП

АЦП: 24 бит, 16 каналов:

- Гальваноизолированный последовательный 16-канальный 24-разрядный АЦП, имеющий 16 аналоговых входов тока от 0 до 24 миллиампер с групповой изоляцией.
- Скорость преобразования 8 микросекунд для одного канала
- Входное сопротивление 249 Ом  $\pm 1,5\%$ .

Для обмена с АЦП используется последовательный канал (SPI) и несколько сигналов управления: Сброс (Reset), режим понижения энергопотребления (PWRDN), автосканирование.

### Регистр управления АЦП (00Fh)

Значение после сброса: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	-	-	-	-	Reset АЦП	Auto Start	PWRDN	IRQEN	Ext Start
											rw	rw	rw	rw	rw

Бит	Функция
<b>ExtStart</b>	Запуск приема от сигнала DRDY 1 Запуск АЦП через регистр RC 0 Нет запуска АЦП
<b>IRQEN</b>	Разрешение запросов на прерывание 1 0
<b>PWRDN</b>	Переключение АЦПх в режим пониженного энергопотребления 1 режим пониженного энергопотребления 0 Нормальный режим
<b>AutoStart</b>	Режим автосканирования 1 Запуск АЦП через регистр RC 0 Нет запуска АЦП
<b>ResetАЦП</b>	Сброс АЦП 1 0

### Регистр команд АЦП (00Eh)

Значение после сброса: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<b>C2</b>	<b>C1</b>	<b>C0</b>	<b>MUL</b>	<b>A3</b>	<b>A2</b>	<b>A1</b>	<b>A0</b>	<b>D7</b>	<b>D6</b>	<b>D5</b>	<b>D4</b>	<b>D3</b>	<b>D2</b>	<b>D1</b>	<b>D0</b>
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Бит	Функция
<b>D0...D7</b>	данные
<b>A0</b>	Адрес стартового байта
<b>A1</b>	

<b>A2</b>	
<b>A3</b>	
<b>MUL</b>	Многобайтовый обмен
<b>C0</b>	Выбор команды
<b>C1</b>	
<b>C2</b>	

Возможные команды, их значения и комбинации приведены в руководстве пользователя ADS1258 стр. 34 в разделе Commands and Register definitions. Биты командного байта 7...0 соответствуют битам 15...8 регистра команды АЦП. Младший байт этого регистра будет первым байтом параметров. В зависимости от типа команды и битов A[3..0] значение этого байта может быть занесено в любой из регистров АЦП. При передаче нескольких байтов сразу, вплоть до всех доступных регистров, каждый последующий байт данных находится на 1 адрес ниже:

Адрес	Старший байт	Младший байт
00Ah	Байт данных 7	Байт данных 8
00Bh	Байт данных 5	Байт данных 6
00Ch	Байт данных 3	Байт данных 4
00Dh	Байт данных 1	Байт данных 2
00Eh	Байт команды	Байт данных 0

Байт команды в обязательном порядке должен быть записан последним одновременно с байтом данных 0 (запись слова), так как запись этого байта начинает передачу данных в АЦП. Пример одновременной установки всех регистров АЦП:

Адрес	Старший байт	Младший байт
00Ah	000h (GPIOC 0FFh)	000h (GPIOD 000h)
00Bh	00h (MUXSG1 0FFh)	000h (SysRed 000h)
00Ch	000h (MUXDIF 000h)	01h (MUXSG0 0FFh)
00Dh	083h (Config1 083h)	000h (MUXSCH 000h)
00Eh	070h	00Ah (Config0 00Ah)

Возможна также и индивидуальная установка регистров, кроме регистра ID (идентификатора), то есть команду на установку выдать возможно, но результат не предскажем:

Адрес	Старший байт	Младший байт
00Ah	Безразличен, не пишем.	Безразличен, не пишем
00Bh	Безразличен, не пишем	Безразличен, не пишем
00Ch	Безразличен, не пишем	Безразличен, не пишем
00Dh	Безразличен, не пишем	Безразличен, не пишем
00Eh	064h	01h (MUXSG0 0FFh)

После установки/настройки регистров можно выдать команду на преобразование канала:

Адрес	Старший байт	Младший байт
00Ah	Безразличен, не пишем.	Безразличен, не пишем
00Bh	Безразличен, не пишем	Безразличен, не пишем
00Ch	Безразличен, не пишем	Безразличен, не пишем



00Dh	Безразличен, не пишем	Безразличен, не пишем
00Eh	080h (или 090h)	Безразличен, не пишем

... и на чтение полученного результата:

Адрес	Старший байт	Младший байт
00Ah	Безразличен, не пишем.	Безразличен, не пишем
00Bh	Безразличен, не пишем	Безразличен, не пишем
00Ch	Безразличен, не пишем	Безразличен, не пишем
00Dh	Безразличен, не пишем	Безразличен, не пишем
00Eh	030h	Безразличен, не пишем

... а после выполнения команды получаем данные:

Адрес	Старший байт	Младший байт
000h	Средний байт данных.	Младший байт данных.
001h	Статусный байт.	Старший байт данных.
002h	Неопределено.	Неопределено.
003h	Неопределено.	Неопределено.
004h	Неопределено.	Неопределено.

Для чтения всех регистров АЦП сразу выдаём команду:

Адрес	Старший байт	Младший байт
00Ah	Безразличен, не пишем.	Безразличен, не пишем
00Bh	Безразличен, не пишем	Безразличен, не пишем
00Ch	Безразличен, не пишем	Безразличен, не пишем
00Dh	Безразличен, не пишем	Безразличен, не пишем
00Eh	050h	Безразличен, не пишем

и после завершения команды получаем:

Адрес	Старший байт	Младший байт
000h	GPIOD (000h)	ID 08Bh
001h	SysRed (000h)	GPIOC (0FFh)
002h	MUXSG0 (0FFh)	MUXSG1 (0FFh)
003h	MUXSCH (000h)	MUXDIF (000h)
004h	Config0 (00Ah)	Config1 (083h)

Можно прочитать и один регистр, например регистр идентификатора:

Адрес	Старший байт	Младший байт
00Ah	Безразличен, не пишем.	Безразличен, не пишем
00Bh	Безразличен, не пишем	Безразличен, не пишем
00Ch	Безразличен, не пишем	Безразличен, не пишем
00Dh	Безразличен, не пишем	Безразличен, не пишем

00Eh	049h	Безразличен, не пишем
------	------	-----------------------

и после завершения команды получаем:

Адрес	Старший байт	Младший байт
000h	Неопределено.	ID 08Bh
001h	Неопределено.	Неопределено.
002h	Неопределено.	Неопределено.
003h	Неопределено.	Неопределено.
004h	Неопределено.	Неопределено.

После завершения обмена устанавливается бит запроса прерывания, если прерывание разрешено установкой бита № 1 (IRQEN - Разрешение запросов на прерывание) в регистре управления АЦП и выбран допустимый вход прерывания битами 3..0 регистра контроля. Можно так же пользоваться битом BusyADC (бит 5 регистра контроля), который устанавливается в 1 на время занятости канала и сбрасывается в 0 после завершения обмена.

### Типовая схема подключения АЦП.

АЦП предназначен для преобразования сигналов, поступающих от токовых датчиков. Типовая схема подключения токовых датчиков приведена на рисунке 3:

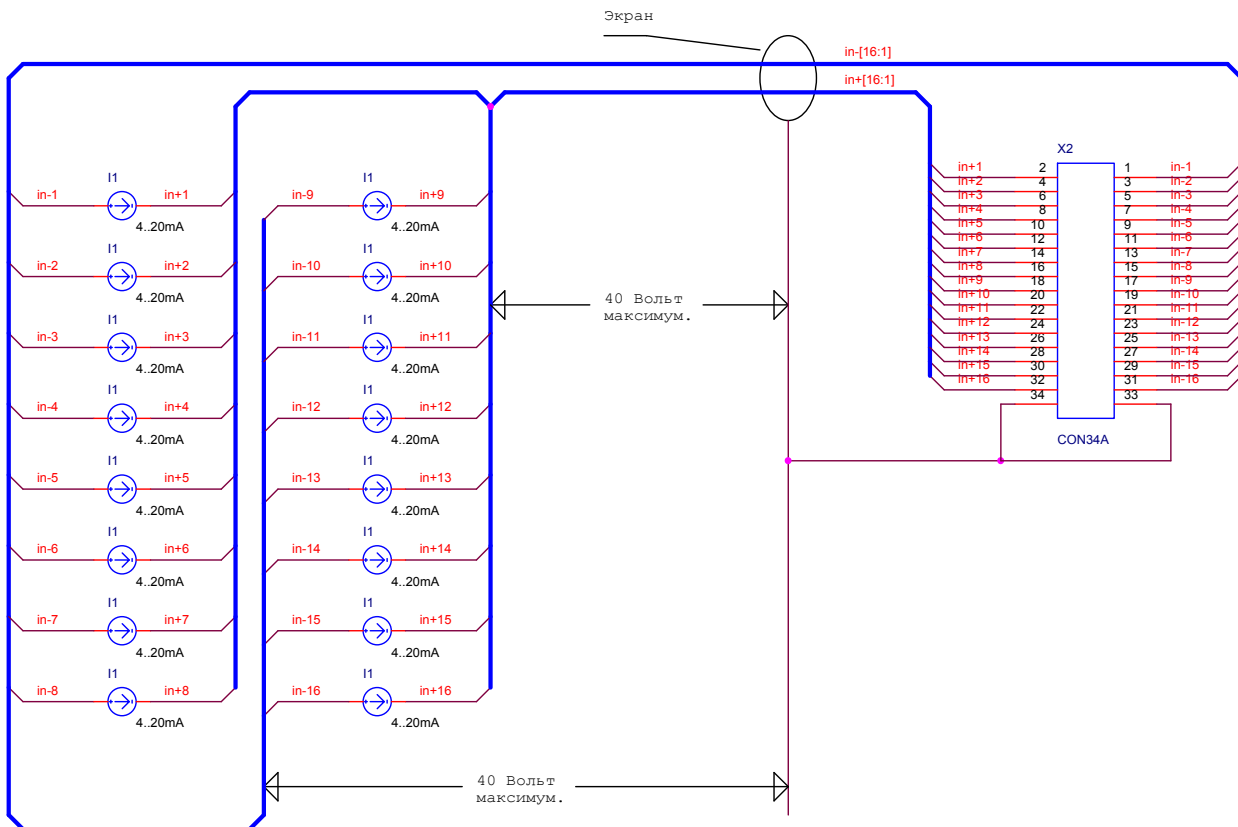


Рис. 3 Типовая схема подключения входных сигналов.

Примечание: При переполюсовке подключенных каналов работа остальных каналов может быть нарушена. Входное сопротивление каждого канала 250 Ом.

## Пример программы обмена с АЦП.

Функция ReadADC устанавливает коммутатор на заданный канал используя SetADCchan, запускает преобразование АЦП и считывает данные из него. В глобальной переменной baseAddr хранится адрес платы, установленный переключателями Jp1 +2, то есть адрес регистра адреса. Перед запуском преобразования необходимо инициализировать плату сбросив АЦП и флаги канала функцией initBoard, которая сбрасывает АЦП и очищает флаги канала обмена.

```
// АДРЕСА УСТРОЙСТВ НА ВНУТРЕННЕЙ ШИНЕ.

#define RADCL 0x0000 // ПРИЕМ ОТ АЦП БАЙТЫ 9-8
#define RADCM1 0x0001 // ПРИЕМ ОТ АЦП БАЙТЫ 6-7
#define RADCM2 0x0002 // ПРИЕМ ОТ АЦП БАЙТЫ 5-4
#define RADCM3 0x0003 // ПРИЕМ ОТ АЦП БАЙТЫ 3-2
#define RADCH 0x0004 // ПРИЕМ ОТ АЦП БАЙТЫ 1-0
// АДРЕСА 4 .. 9 ЗАРЕЗЕРВИРОВАНЫ
#define WADCL 0x000A // ПЕРЕДАЧА В АЦП БАЙТЫ 8-7
#define WADCM1 0x000B // ПЕРЕДАЧА В АЦП БАЙТЫ 6-5
#define WADCM2 0x000C // ПЕРЕДАЧА В АЦП БАЙТЫ 4-3
#define WADCM3 0x000D // ПЕРЕДАЧА В АЦП БАЙТЫ 2-1
#define WADCH 0x000E // ПЕРЕДАЧА В АЦП БАЙТЫ COM-0

#define ComADC 0x000F // АДРЕС РЕГИСТРА УПРАВЛЕНИЯ АЦП.

// ТАБЛИЧКА ПРЕОБРАЗОВАНИЯ НОМЕРА БИТА В ЕГО ПОЛОЖЕНИЕ В СЛОВЕ.
int unitary[16]={0x0001,0x0002,0x0004,0x0008,0x0010,0x0020,0x0040,0x0080,
                0x0100,0x0200,0x0400,0x0800,0x1000,0x2000,0x4000,0x8000};

// ГЛОБАЛЬНЫЕ ПЕРЕМЕННЫЕ:

static int baseAddr; // БАЗОВЫЙ АДРЕС ПЛАТЫ+2 ОТ УСТАНОВЛЕННОГО J3.

// -----

void wrIO (int data,int adr) // УСТАНОВКА РЕГИСТРА.
{
    outp(baseAddr,adr);
    outpw((baseAddr-2),data);
}

int rdIO (int adr) // ЧТЕНИЕ РЕГИСТРА.
{
    outp(baseAddr,adr);
    return(inpw(baseAddr-2));
}
```

```
INT RSTATUS (VOID)                // ЧТЕНИЕ РЕГИСТРА СТАТУСА.
{
    RETURN (INP (BASEADDR+1) );
}

// *****

INT NUM2UNIT (INT NUM)            // ПРЕОБРАЗОВАТЬ НОМЕР В ПОЗИЦИЮ БИТА.
{
    RETURN NUM[UNITARY];
}

// *****

VOID WAITADCSP1 (VOID)            // ДОЖДАТЬСЯ КОНЦА ПЕРЕДАЧИ В АЦП.
{
    WHILE (RSTATUS () &0x020);
}

VOID WAITADCCONV (VOID)          // ПОЛУЧИТЬ ГОТОВНОСТЬ АЦП.
{
    WHILE (RSTATUS () &0x010);
}

// СБРОСИТЬ СОБСТВЕННО АЦП И ОЧИСТИТЬ ФЛАГИ КАНАЛА.

VOID INITBOARD ()
{
    WRIO (0x0C000, wADCH);        // КОМАНДА СБРОС АЦП.
    WRIO (0x08000, wADCH); WAITADCCONV (); // СБРОС DRDY.
    WRIO (0x03000, wADCH); WAITADCSP1 (); // СЧИТАТЬ ДАННЫЕ.
}

VOID SETADCCHAN (INT CHAN)       // ВЫБРАТЬ КАНАЛ АЦП.
{
    WRIO (0x00000, wADCL);        // УСТАНОВИТЬ GPIOC И GPIOD.
    WRIO (NUM2UNIT (CHAN) &0x0FF00, wADCm1); // УСТАНОВИТЬ MUXSG1 И SysRED.
    WRIO (NUM2UNIT (CHAN) &0x000FF, wADCm2); // УСТАНОВИТЬ MUXDIF И MUXSG0.
    WRIO (0x00000, wADCm3);        // УСТАНОВИТЬ CONFIG1 И MUXSGH.
    WRIO (0x0700E, wADCH);        // УСТАНОВИТЬ CONFIG0 И ЗАПУСТИТЬ.
}

// ЕДИНИЧНОЕ ПРЕОБРАЗОВАНИЕ ЗАДАННОГО КАНАЛА АЦП.
// НОМЕР КАНАЛА В ВЫХОДНЫХ ДАННЫХ СБРОШЕН.
```

```
LONG READADC (INT CHAN)          // НОМЕР КАНАЛА АЦП.
{ LONG ADC;
  SETADCSCHAN(CHAN); WAITADCSP1(); // ВЫБРАТЬ КАНАЛ АЦП.
  WRIO(0x0008, ComADC); WAITADCconv(); // ЗАПУСТИТЬ ПРЕОБРАЗОВАНИЕ.
  WRIO(0x0000, ComADC);           // ОСТАНОВИТЬ ПРЕОБРАЗОВАНИЕ.
  WRIO(0x03000, wADCH); WAITADCSP1(); // СЧИТАТЬ ДАННЫЕ.
  ADC= ((LONG) (RDIO (RADCM1) &0x0FF) <<16 | ((LONG) RDIO (RADCL) &0xFFFFL) );
  ADC=ADC<<8;                      // РАСШИРЕНИЕ ЗНАКА.
  RETURN (ADC>>8);
}
```

## 10. ЦАП.

Модуль содержит четыре идентичных индивидуально изолированных канала ЦАП, обозначаемых соответственно канал 0, канал 1, канал 2 и канал 3.

Цифро-аналоговый преобразователь: 16 разрядов:

- Выходное напряжение – от минус 10 Вольт до 10 Вольт, или от минус 5 Вольт до 5 Вольт, или от 0 до 5 Вольт, или от 0 до 10 Вольт.
- Выходной ток – от 4 до 20 миллиампер, или от 0 до 20 миллиампер, или от 0 до 24 миллиампер.
- Скорость преобразования ЦАП 3 миллисекунды.
- Режим выхода устанавливается переключателями на плате модуля.

Для работы с ЦАП используются 2 регистра во внутреннем адресном пространстве: регистр с адресом 10h предназначен для передачи данных, а регистр управления с адресом 11h позволяет выбрать канал ЦАП, отображает биты ошибок, каналов разрешает запросы на прерывания при возникновении ошибок и завершении передачи данных. Флаг готовности ЦАП выведен в регистр контроля (бит №6) а запрос прерывания дублируется битом №7 в регистре контроля.

### Регистр управления ЦАП (011h)

Значение после сброса: 0000h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLT3	FLT2	FLT1	FLT0	x	x	x	IRQEN	IRQ FLT EN3	IRQ FLT EN2	IRQ FLT EN1	IRQ FLT EN0	EN3	EN2	EN1	EN0
													rw		rw

Бит	Функция
EN0	Разрешение работы канала 0 1 – работа разрешена 0 – работа запрещена
EN1	Разрешение работы канала 1
EN2	Разрешение работы канала 2
EN3	Разрешение работы канала 3
IRQFLTEN0	Разрешение запросов прерывания по ошибке канала 0
IRQFLTEN1	Разрешение запросов прерывания по ошибке канала 1
IRQFLTEN2	Разрешение запросов прерывания по ошибке канала 2
IRQFLTEN3	Разрешение запросов прерывания по ошибке канала 3
IRQEN	Разрешение / сброс запросов прерывания от готовности ЦАП
x	Зарезервирован
FLT0	Обрыв линии канала 0 (только чтение)
FLT1	Обрыв линии канала 1 (только чтение)
FLT2	Обрыв линии канала 2 (только чтение)
FLT3	Обрыв линии канала 3 (только чтение)

Для разрешения работы канала или(и) генерирования прерывания, установите соответствующий бит регистра. Запрос прерывания очищается автоматически при чтении регистра данных ЦАП или записи в его старшую часть. Запись в старшую часть этого регистра начинает передачу данных в выбранные битами 3..0 каналы. Также можно сбросить запрос прерывания установкой в 0 бита №8 регистра управления ЦАП.

Сигнал Fault появляется тогда, когда заданное значение тока в канале не может быть установлено. Такая ситуация обычно возникает при обрыве или отсутствии нагрузки. Этот сигнал доступен только в режиме токового выхода и в режиме выхода напряжения он всегда не активен (0).

Значения каналов могут быть установлены одновременно для всех или некоторых каналов. Для этого у соответствующих каналов устанавливаются биты EN.

**Регистр данных ЦАП (011h)**

Значение после сброса: **0000h**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Бит	Функция
D15...D0	Преобразованные данные (D0 - младший значащий разряд)

### Типовая схема подключения ЦАП.

Все каналы ЦАП отличаются между собой только разъёмами на которые они выведены и номерами переключателей установки режима работы.

Типовая схема подключения ЦАП приведена на рисунке 4.

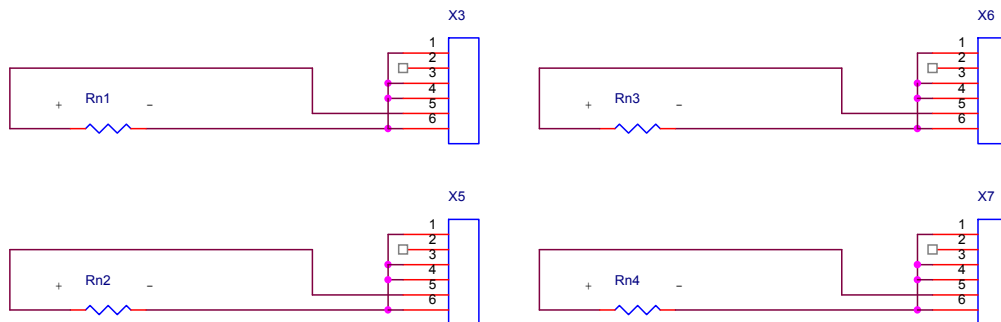


Рис. 4 Схема подключения ЦАП.

Разъёмы X3, X5, X6, X7, предназначены для подключения к выходным сигналам к ЦАП каналов 0, 1, 2, 3 соответственно.

Номер контакта	Сигнал	Номер контакта	Сигнал
1	AGND_DACx	2	U_OUT_DACx
3	AGND_DACx	4	AGND_DACx
5	I_OUT_DACx	6	AGND_DACx

**Примечание:**

AGND_DACx	Аналоговая земля ЦАП канала x, где x = 0, 1, 2 или 3.
U_OUT_DACx	Выход напряжения ЦАП канала x, где x = 0, 1, 2 или 3.
I_OUT_DACx	Выход тока ЦАП канала x, где x = 0, 1, 2 или 3.

**Примечание** Одновременное использование в одном канале выходов тока и напряжения невозможно.

Выходы канала 0 подключены к разъёму X3, канала 1 к X5, канала 2 к X6 и канала 3 к X7.

Для установки выходного тока 4-20 мА следует установить переключки в положение 4-6 и 1-3 для переключателей: канал 0 - JP2, канал 1 - JP7, канал 3 - JP12, канал 3 - JP17. Установка этих переключателей в положения 2-4 и 3-5 устанавливает диапазон выходного тока 0-20 мА, в положениях 2-4 и 1-3 устанавливается диапазон токов 0-24 мА, положения 4-6 и 3-5 активизируют выход напряжения. В режиме выхода напряжения доступны 4 выходных диапазона напряжений: 0...+5 Вольт, 0...+10 Вольт, -5...+5 Вольт и -10...+10 Вольт. Выбор диапазона выходного напряжения или тока осуществляется установкой переключек между обозначенными контактами на переключателях с соответствии с таблицей:

## Канал 0 (выходной разъем X3)

Переключатель \ Диапазон	JP5	JP2	JP3	JP4	JP6
1. 0...20 мА.	1-2	2-4 и 3-5	-	-	-
2. 0...24 мА.	1-2	1-3 и 2-4	-	-	-
3. 4...20 мА.	1-2	1-3 и 4-6	-	-	-
4. 0...5 Вольт.	1-2	3-5 и 4-6	Отсутствует.	Отсутствует.	1-2
5. 0...10 Вольт.			Отсутствует.	Отсутствует.	2-3
6. -5...+5 Вольт.			1-2	Отсутствует.	Отсутствует.
7. -10...+10 Вольт.			1-2	1-2	2-3

## Канал 1 (выходной разъем X5)

Переключатель \ Диапазон	JP10	JP7	JP8	JP9	JP11
1. 0...20 мА.	1-2	2-4 и 3-5	-	-	-
2. 0...24 мА.	1-2	1-3 и 2-4	-	-	-
3. 4...20 мА.	1-2	1-3 и 4-6	-	-	-
4. 0...5 Вольт.	1-2	3-5 и 4-6	Отсутствует.	Отсутствует.	1-2
5. 0...10 Вольт.			Отсутствует.	Отсутствует.	2-3
6. -5...+5 Вольт.			1-2	Отсутствует.	Отсутствует.
7. -10...+10 Вольт.			1-2	1-2	2-3



## Канал 2 (выходной разъем X6)

Переключатель \ Диапазон	JP15	JP12	JP13	JP14	JP16
1. 0...20 мА.	1-2	2-4 и 3-5	-	-	-
2. 0...24 мА.	1-2	1-3 и 2-4	-	-	-
3. 4...20 мА.	1-2	1-3 и 4-6	-	-	-
4. 0...5 Вольт.	1-2	3-5 и 4-6	Отсутствует.	Отсутствует.	1-2
5. 0...10 Вольт.			Отсутствует.	Отсутствует.	2-3
6. -5...+5 Вольт.			1-2	Отсутствует.	Отсутствует.
7. -10...+10 Вольт.			1-2	1-2	2-3

## Канал 3 (выходной разъем X7)

Переключатель \ Диапазон	JP20	JP17	JP18	JP19	JP21
1. 0...20 мА.	1-2	2-4 и 3-5	-	-	-
2. 0...24 мА.	1-2	1-3 и 2-4	-	-	-
3. 4...20 мА.	1-2	1-3 и 4-6	-	-	-
4. 0...5 Вольт.	1-2	3-5 и 4-6	Отсутствует.	Отсутствует.	1-2
5. 0...10 Вольт.			Отсутствует.	Отсутствует.	2-3
6. -5...+5 Вольт.			1-2	Отсутствует.	Отсутствует.
7. -10...+10 Вольт.			1-2	1-2	2-3

## Пример работы с ЦАП.

Функция SetDAC получает данные для записи в ЦАП и номер канала ЦАП. Если номер ЦАП больше 3, то данные записываются сразу во все каналы. Преобразование номера ЦАП в позицию бита в регистре управления осуществляется подфункцией num2unit. При работе функция запрещает прерывания от блока ЦАП устанавливая в 0 все биты регистра управления ЦАП, кроме битов EN[3..0], которые устанавливаются в зависимости от номера выбранного для преобразования канала ЦАП.

```
// АДРЕСА УСТРОЙСТВ НА ВНУТРЕННЕЙ ШИНЕ.
```

```
#DEFINE DATDAC 0x0010 // АДРЕС РЕГИСТРА ДАННЫХ ЦАП.
#DEFINE COMDAC 0x0011 // АДРЕС УПРАВЛЕНИЯ ЦАП.
```

```
// ТАБЛИЧКА ПРЕОБРАЗОВАНИЯ НОМЕРА БИТА В ЕГО ПОЛОЖЕНИЕ В СЛОВЕ.
```

```
INT UNITARY[16]={0x0001,0x0002,0x0004,0x0008,0x0010,0x0020,0x0040,0x0080,
0x0100,0x0200,0x0400,0x0800,0x1000,0x2000,0x4000,0x8000};
```

```
// ГЛОБАЛЬНЫЕ ПЕРЕМЕННЫЕ:
```

```
STATIC INT BASEADDR ; // БАЗОВЫЙ АДРЕС ПЛАТЫ+2 ОТ УСТАНОВЛЕННОГО JF1.

// -----

VOID WRIO (INT DATA, INT ADR) // УСТАНОВКА РЕГИСТРА.
{
    OUTP (BASEADDR, ADR) ;
    OUTPW ((BASEADDR-2), DATA) ;
}

// *****

INT NUM2UNIT (INT NUM) // ПРЕОБРАЗОВАТЬ НОМЕР В ПОЗИЦИЮ БИТА.
{
    RETURN NUM[UNITARY] ;
}

// *****

VOID WAITDACSPI (VOID) // ДОЖДАТЬСЯ КОНЦА ПЕРЕДАЧИ В ЦАП.
{
    WHILE (RSTATUS () & 0x040) ;
}

// УСТАНОВИТЬ ЦАП. ЕСЛИ НОМЕР ЦАП >= 4 УСТАНОВЛИВАЮТСЯ ВСЕ КАНАЛЫ.

VOID SETDACs (UNSIGNED INT DATA, UNSIGNED INT DAC)
{
    IF (DAC < 4) {DAC=NUM2UNIT (DAC) ;} // ВЫБРАТЬ КАНАЛ.
    ELSE {DAC=0x00F ;} // ВЫБРАТЬ ВСЕ КАНАЛЫ.
    WAITDACSPI () ; // КАНАЛ ГОТОВ К ПЕРЕДАЧЕ?
    WRIO (DAC, 0x011) ; // ВЫБРАТЬ КАНАЛ.
    WRIO (DATA, 0x010) ; // УСТАНОВИТЬ КАНАЛ.
}
```

## 11 Сброс модуля

Сброс модуля осуществляется сигналом RESET по шине PC/104 (ISA).

## 12. Питание модуля



### Внимание

все подключения и отключения к разъемам и все коммутации на переключателях должны осуществляться только при отключенном напряжении питания контроллера.

Модуль питается от внешнего источника постоянного тока  $+5\text{ В} \pm 5\%$  (включая выбросы при включении и выключении питания) с типовым потреблением 1,2 Ампер (в режиме токовых выходов).

Стартовый ток может достигать величины до 3,0 А для модуля KM1604–EXT(MIL) с диапазоном рабочих температур от минус 40°C до плюс 85°C и KM1604–MIL с диапазоном рабочих температур от минус 55°C до плюс 85°C.

Напряжение питания подается через разъемы X1, X4 шины PC/104.

Наличие напряжения питания +5 В индицируется свечением светодиода.

### 13. Внешние разъемы и переключатели.

Подключение внешних цепей к модулю осуществляется с помощью разъемов. Расположение разъемов и переключателей на плате модуля KM1604 представлено на рисунке 5.

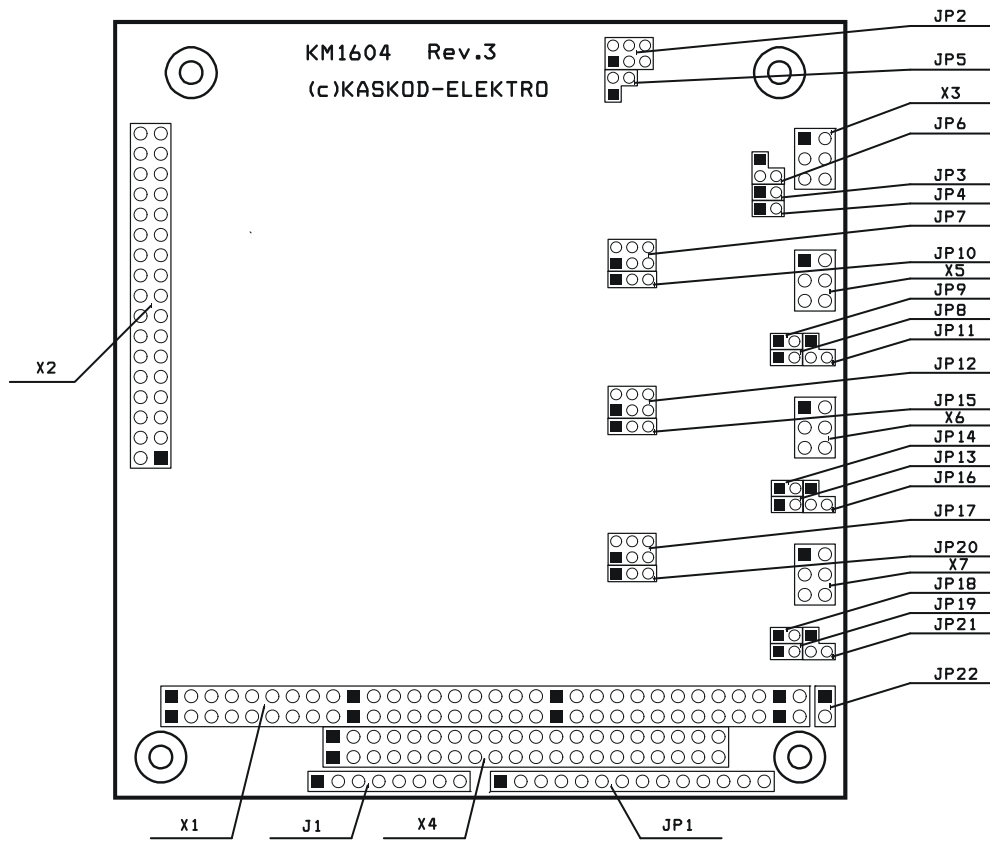


Рис. 5 Расположение разъемов на плате.

Типовое расположение разъемов и переключателей IDC-типа. Первый контакт имеет квадратную форму печатной площадки (рис. 6).

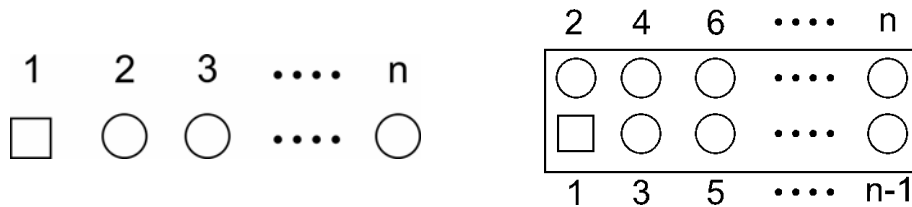


Рис. 6 Типовое расположение разъемов и переключателей IDC-типа.

**Переключатель J1**

Тип: 8-контактный однорядный разъем IDC типа.

Разъем технологический.

Первый контакт имеет квадратную форму печатной площадки на плате.

Номер контакта	Сигнал
1	технологический
2	технологический
3	VCC (+ 5 Вольт)
4	отсутствует
5	GND (Цифровая земля (общий))
6	технологический
7	технологический
8	GND (Цифровая земля (общий))

**Переключатель JP1**

Тип: 16-контактный переключатель IDC типа.

Разъем предназначен установки базового адреса.

Первый контакт имеет квадратную форму печатной площадки на плате.

Номер контакта	Сигнал
1	SeIA2
2	GND (Цифровая земля (общий))
3	SeIA3
4	GND (Цифровая земля (общий))
5	SeIA4
6	GND (Цифровая земля (общий))
7	SeIA5
8	GND (Цифровая земля (общий))
9	SeIA6
10	GND (Цифровая земля (общий))
11	SeIA7
12	GND (Цифровая земля (общий))
13	SeIA8
14	GND (Цифровая земля (общий))

**Переключатель JP22**

Тип: 2-контактный переключатель IDC типа.

Разъем предназначен установки базового адреса.

Первый контакт имеет квадратную форму печатной площадки на плате.

Номер контакта	Сигнал
1	SeIA9
2	GND (Цифровая земля (общий))

**Разъем X1.**

Разъем X1 предназначен для подключения к шине PC/104.

Тип: 64-контактный сквозной разъем PC/104L

Номер контакта	Название контакта	Сигнал	Номер контакта	Название контакта	Сигнал
A1	IOCHCHK*		B1	GND	GND
A2	SD7	SD7	B2	RESETDRV	RESET
A3	SD6	SD6	B3	+5V	VCC
A4	SD5	SD5	B4	IRQ9	IRQ9
A5	SD4	SD4	B5	-5V	nc
A6	SD3	SD3	B6	DRQ2	nc
A7	SD2	SD2	B7	-12V	nc
A8	SD1	SD1	B8	ENDXFR*	nc
A9	SD0	SD0	B9	+12V	nc
A10	IOCHRDY	nc	B10	KEY(2)	nc
A11	AEN	AEN	B11	SMEMW*	nc
A12	SA19	nc	B12	SMEMR*	nc
A13	SA18	nc	B13	IOW*	IOW
A14	SA17	nc	B14	IOR*	IOR
A15	SA16	nc	B15	DACK3*	nc
A16	SA15	nc	B16	DRQ3	nc
A17	SA14	nc	B17	DACK1*	nc
A18	SA13	nc	B18	DRQ1	nc
A19	SA12	nc	B19	REFRESH*	nc
A20	SA11	nc	B20	SYSCLK	nc
A21	SA10	nc	B21	IRQ7	IRQ7
A22	SA9	SA9	B22	IRQ6	IRQ6
A23	SA8	SA8	B23	IRQ5	IRQ5
A24	SA7	SA7	B24	IRQ4	IRQ4
A25	SA6	SA6	B25	IRQ3	IRQ3
A26	SA5	SA5	B26	DACK2*	nc
A27	SA4	SA4	B27	TC	nc
A28	SA3	SA3	B28	BALE	nc
A29	SA2	SA2	B29	+5V	VCC
A30	SA1	SA1	B30	OSC	nc
A31	SA0	SA0	B31	GND	GND
A32	GND	GND	B32	GND	GND

**Примечание:**

SD0...SD7	Бит данных
SA0...SA9	Бит адреса
AEN	Разрешение адреса
RESET	Сброс модуля
IRQ3...IRQ7, IRQ9	Прерывание 3...прерывание 7, прерывание 9

IOW	Сигнал “Запись в порт”.
IOR	Сигнал “Чтение из порта”.
VCC	Напряжение питания модуля +5 Вольт
GND	Цифровая земля (общий)

**Разъём X4.**

Тип: 40-контактный сквозной разъём PC/104

Разъём X4 предназначен для подключения к шине PC/104H.

Номер контакта	Название контакта	Сигнал	Номер контакта	Название контакта	Сигнал
C1	GND	GND	D1	GND	GND
C2	SBHE*	BHE	D2	MEMCS16*	nc
C3	LA23	nc	D3	IOCS16*	IOCS16
C4	LA22	nc	D4	IRQ10	IRQ10
C5	LA21	nc	D5	IRQ11	IRQ11
C6	LA20	nc	D6	IRQ12	IRQ12
C7	LA19	nc	D7	IRQ15	IRQ15
C8	LA18	nc	D8	IRQ14	IRQ14
C9	LA17	nc	D9	DACK0*	nc
C10	MEMR*	nc	D10	DRQ0	nc
C11	MEMW*	nc	D11	DACK5*	nc
C12	SD8	SD8	D12	DRQ5	nc
C13	SD9	SD9	D13	DACK6*	nc
C14	SD10	SD10	D14	DRQ6	nc
C15	SD11	SD11	D15	DACK7*	nc
C16	SD12	SD12	D16	DRQ7	nc
C17	SD13	SD13	D17	+5V	VCC
C18	SD14	SD14	D18	MASTER*	nc
C19	SD15	SD15	D19	GND	GND
C20	GND (KEY)	GND	D20	GND	GND

*Примечание:*

SD8...SD15	Бит данных
BHE	Разрешение старшего байта
IOCS16	
IRQ10...IRQ12	Прерывание 10, прерывание 11, прерывание 12
IRQ14, IRQ15	Прерывание 14, прерывание 15
VCC	Напряжение питания модуля +5 Вольт
GND	Цифровая земля (общий)

**Разъем X2.**

Тип: 34-контактный штыревой двухрядный разъем IDC-типа.

Разъем X1 предназначен для подключения входных сигналов к АЦП.

Номер контакта	Сигнал	Номер контакта	Сигнал
1	- Vin1	2	+ Vin1
3	- Vin2	4	+ Vin2
5	- Vin3	6	+ Vin3
7	- Vin4	8	+ Vin4
9	- Vin5	10	+ Vin5
11	- Vin6	12	+ Vin6
13	- Vin7	14	+ Vin7
15	- Vin8	16	+ Vin8
17	- Vin9	18	+ Vin9
19	- Vin10	20	+ Vin10
21	- Vin11	22	+ Vin11
23	- Vin12	24	+ Vin12
25	- Vin13	26	+ Vin13
27	- Vin14	28	+ Vin14
29	- Vin15	30	+ Vin15
31	- Vin16	32	+ Vin16
33	AGND	34	AGND

*Примечание:*

AGND_ADC	Аналоговая земля АЦП.
- Vin1...- Vin16	Минусовой вход для датчика тока (источника напряжения, для входов напряжения).
+ Vin1...+ Vin16	Плюсовой вход для датчика тока (источника напряжения, для входов напряжения).

**Разъем X3.**

Тип: 6-контактный штыревой двухрядный разъем IDC-типа.

Разъем X3 предназначен для подключения к выходным сигналам к ЦАП канала 0.

Номер контакта	Сигнал	Номер контакта	Сигнал
1	AGND_DAC0	2	U_OUT_DAC0
3	AGND_DAC0	4	AGND_DAC0
5	I_OUT_DAC0	6	AGND_DAC0

*Примечание:*

AGND_DAC0	Аналоговая земля ЦАП канал 0.
U_OUT_DAC0	Выход напряжения ЦАП канал 0.
I_OUT_DAC0	Выход тока ЦАП канал 0.



**Разъем X5.**

Тип: 6-контактный штыревой двухрядный разъем IDC-типа.

Разъём X3 предназначен для подключения к выходным сигналам к ЦАП канала 1.

Номер контакта	Сигнал	Номер контакта	Сигнал
1	AGND_DAC1	2	U_OUT_DAC1
3	AGND_DAC1	4	AGND_DAC1
5	I_OUT_DAC1	6	AGND_DAC1

*Примечание:*

AGND_DAC1	Аналоговая земля ЦАП канал 1.
U_OUT_DAC1	Выход напряжения ЦАП канал 1.
I_OUT_DAC1	Выход тока ЦАП канал 1.

**Разъем X6.**

Тип: 6-контактный штыревой двухрядный разъем IDC-типа.

Разъём X3 предназначен для подключения к выходным сигналам к ЦАП канала 2.

Номер контакта	Сигнал	Номер контакта	Сигнал
1	AGND_DAC2	2	U_OUT_DAC2
3	AGND_DAC2	4	AGND_DAC2
5	I_OUT_DAC2	6	AGND_DAC2

*Примечание:*

AGND_DAC2	Аналоговая земля ЦАП канал 2.
U_OUT_DAC2	Выход напряжения ЦАП канал 2.
I_OUT_DAC2	Выход тока ЦАП канал 2.

**Разъем X7.**

Тип: 6-контактный штыревой двухрядный разъем IDC-типа.

Разъём X3 предназначен для подключения к выходным сигналам к ЦАП канала 3.

Номер контакта	Сигнал	Номер контакта	Сигнал
1	AGND_DAC3	2	U_OUT_DAC3
3	AGND_DAC3	4	AGND_DAC3
5	I_OUT_DAC3	6	AGND_DAC3

*Примечание:*

AGND_DAC3	Аналоговая земля ЦАП канал 3.
U_OUT_DAC3	Выход напряжения ЦАП канал 3.
I_OUT_DAC3	Выход тока ЦАП канал 3.

## 14. Условия эксплуатации и хранения

Модуль KM1604 предназначен для работы в составе группы модулей формата PC/104 или ISA, через переходную плату PC104-ISA .

Напряжение питания подается через разъемы шины PC104 X1, X4. Наличие напряжения питания 5 В индицируется свечением светодиода.

Детали и сборочные единицы, взятые на специальный учёт в KM1604 отсутствуют.

Изделие удовлетворяет следующим требованиям эксплуатации:

- диапазон рабочих температур: от 0°С до плюс 70°С,
- диапазон температур хранения: от минус 40°С до плюс 70°С.

Изделие для расширенного диапазона рабочих температур удовлетворяет следующим требованиям эксплуатации:

- диапазон рабочих температур: от минус 40°С до плюс 85°С,
- диапазон температур хранения: от минус 55°С до плюс 85°С.

При необходимости большего диапазона рабочих температур и температур хранения обращайтесь к изготовителю.

## 15. Варианты исполнения модуля

Модуль поставляется в следующих модификациях:

Наименование	Описание
<b>KM1604</b>	АЦП: 24 бит, 16 каналов; ЦАП: 16 бит, 4 канала; диапазон рабочих температур: 0°C - +70°C
<b>KM1604-EXT</b>	АЦП: 24 бит, 16 каналов; ЦАП: 16 бит, 4 канала; диапазон рабочих температур: -40°C - +85°C
<b>KM1604-AD10DA1</b>	АЦП: 24 бит, 10 каналов (канал 1 - канал 10); ЦАП: 16 бит, 1 канал (канал 0); диапазон рабочих температур: -40°C - +85°C

**Внимание:** По умолчанию устанавливаются разъемы прямые вверх.  
Расположение разъемов оговаривается при заказе.

Возможное расположение разъемов:

- Разъемы **X2, X3, X5, X6, X7**:
  - прямые вверх;
  - угловые;
  - прямые вниз.
- Остальные разъемы и переключатели:
  - прямые вверх;
  - прямые вниз.

**Дополнительно можно заказать:**

- **-KIT**                    ответные части всех разъемов (кроме разъемов шины PC104);
- **-LAK**                    влагозащитное покрытие;

*Замечание:* При заказе модулей необходимо соблюдать обозначения изделий данные выше.

*Например:*

Код заказа	описание
<b>KM1604-EXT-KIT-LAK, угловые разъемы</b>	Модуль KM1604, АЦП: 24 бит, 16 каналов; ЦАП: 16 бит, 4 канала, расширенный температурный диапазон -40°C - +85°C, ответные части всех разъемов, влагозащитное покрытие, <b>с угловыми разъемами.</b>

## 16. Комплект поставки и маркировка модуля

### В комплект поставки входит:

- |    |                                    |   |       |
|----|------------------------------------|---|-------|
| 1. | Модуль KM1604 в выбранном варианте | - | 1 шт. |
| 2. | Компакт-диск                       | - | 1 шт. |

#### Примечание:

С партией модулей поставляется не более двух компакт-дисков .

### На компакт-диске:

- руководство пользователя.
- документация;
- примеры программ;

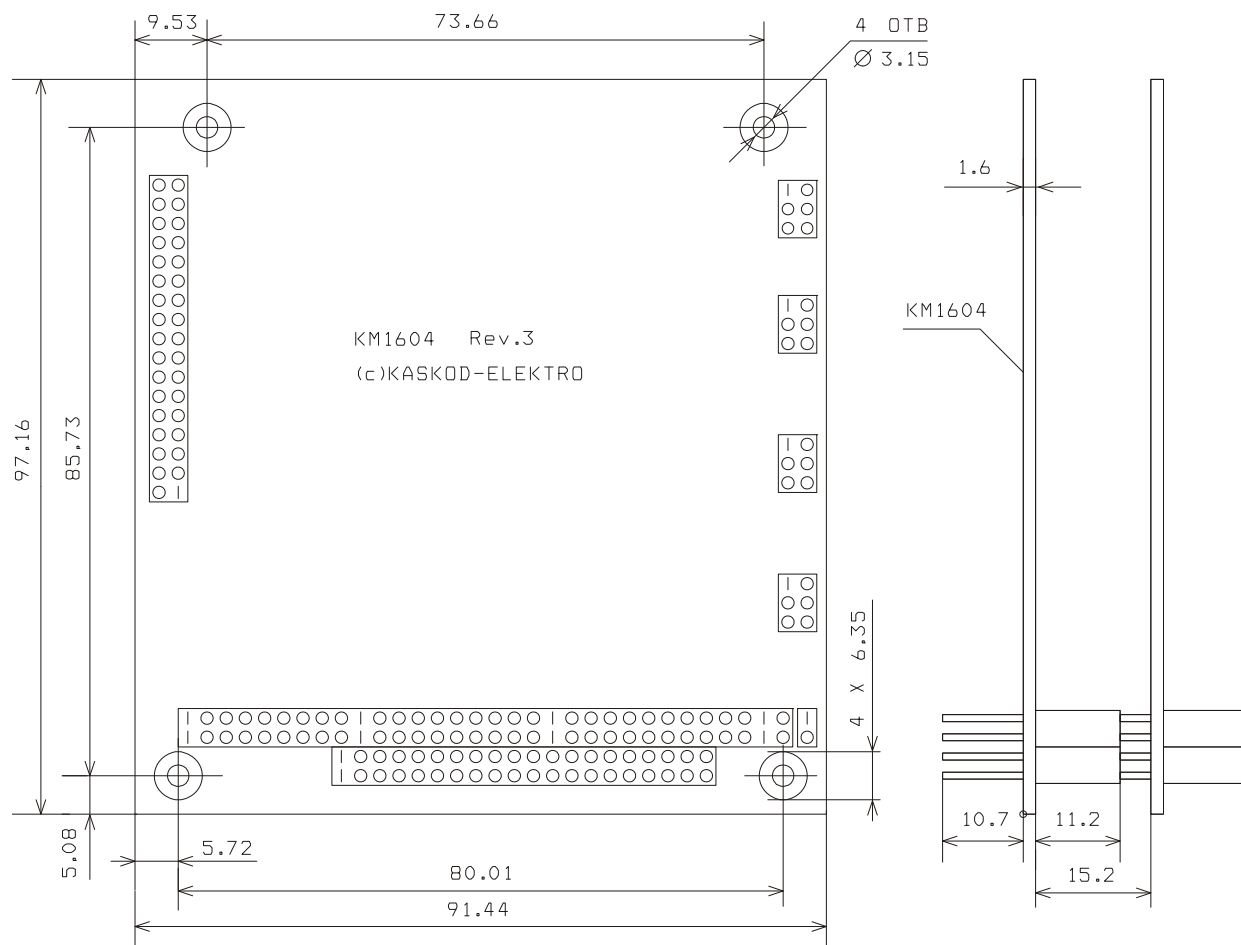
### Маркировка контроллера

Модуль KM1604 имеет маркировку на плате **KM1604**.

Серийный номер находится на плате и имеет вид:

- **S/N XXXXXX**, например: S/N 115356.

# 17. Габаритные и установочные размеры.



## Приложение

### Пример программы минитеста модуля.

#### Демонстрационная программа.

Демонстрационная программа предназначена для проверки работы входов платы КМ1604. После запуска программа запрашивает базовый адрес платы, проверяет его на допустимость (диапазон адресов 200h..3FCh), на наличие платы (наличие платы определяется по старшему байту идентификационного регистра) и циклически выводит на экран результаты преобразования для всех каналов. АЦП используется в режиме 16-ти недифференциальных входов с коммутатором. Программа позволяет устанавливать значения для каждого из ЦАПов поотдельности в двоичном виде (SetDAC), либо в диапазоне токов 4-20 мА (SetDACf). При попытке установки тока больше или меньше диапазона, величина тока будет установлена на минимальное или максимально возможное значение. Функции ожидания готовности вызывают системную функцию kbhit, что позволяет выйти из программы (нажать ctrl~с или ctrl~Break) при неисправной плате.

```
#INCLUDE <STDIO.H>
#include <I86.H>

// ГЛОБАЛЬНЫЕ КОНСТАНТЫ:

#define LEAVEKEY 27 // Код выхода из программы.
#define IMAX 25.0 // МАКСИМАЛЬНОЕ ЗНАЧЕНИЕ ТОКА.

// АДРЕСА УСТРОЙСТВ НА ВНУТРЕННЕЙ ШИНЕ.

#define RADCL 0x0000 // ПРИЕМ ОТ АЦП БАЙТЫ 9-8
#define RADCM1 0x0001 // ПРИЕМ ОТ АЦП БАЙТЫ 6-7
#define RADCM2 0x0002 // ПРИЕМ ОТ АЦП БАЙТЫ 5-4
#define RADCM3 0x0003 // ПРИЕМ ОТ АЦП БАЙТЫ 3-2
#define RADCH 0x0004 // ПРИЕМ ОТ АЦП БАЙТЫ 1-0
// АДРЕСА 4 .. 9 ЗАРЕЗЕРВИРОВАННЫ
#define WADCL 0x000A // ПЕРЕДАЧА В АЦП БАЙТЫ 8-7
#define WADCM1 0x000B // ПЕРЕДАЧА В АЦП БАЙТЫ 6-5
#define WADCM2 0x000C // ПЕРЕДАЧА В АЦП БАЙТЫ 4-3
#define WADCM3 0x000D // ПЕРЕДАЧА В АЦП БАЙТЫ 2-1
#define WADCH 0x000E // ПЕРЕДАЧА В АЦП БАЙТЫ СОМ-0

#define ComADC 0x000F // АДРЕС РЕГИСТРА УПРАВЛЕНИЯ АЦП.

#define DatDAC 0x0010 // АДРЕС РЕГИСТРА ДАННЫХ ЦАП.
#define ComDAC 0x0011 // АДРЕС УПРАВЛЕНИЯ ЦАП.

// ТАБЛИЧКА ПРЕОБРАЗОВАНИЯ НОМЕРА БИТА В ЕГО ПОЛОЖЕНИЕ В СЛОВЕ.
```

```
INT UNITARY [16]={0x0001,0x0002,0x0004,0x0008,0x0010,0x0020,0x0040,0x0080,
                0x0100,0x0200,0x0400,0x0800,0x1000,0x2000,0x4000,0x8000};

// ГЛОБАЛЬНЫЕ ПЕРЕМЕННЫЕ:

STATIC INT BASEADDR ; // БАЗОВЫЙ АДРЕС ПЛАТЫ+2 ОТ УСТАНОВЛЕННОГО J3.

// -----

VOID WRIO (INT DATA,INT ADR) // УСТАНОВКА РЕГИСТРА.
{
    OUTP (BASEADDR,ADR);
    OUTPW ((BASEADDR-2),DATA);
}

INT RDIO (INT ADR) // ЧТЕНИЕ РЕГИСТРА.
{
    OUTP (BASEADDR,ADR);
    RETURN (INPW (BASEADDR-2));
}

INT RSTATUS (VOID) // ЧТЕНИЕ РЕГИСТРА СТАТУСА.
{
    RETURN (INP (BASEADDR+1));
}

// *****

INT NUM2UNIT (INT NUM) // ПРЕОБРАЗОВАТЬ НОМЕР В ПОЗИЦИЮ БИТА.
{
    RETURN NUM[UNITARY];
}

// *****

VOID WAITDACSPI (VOID) // ДОЖДАТЬСЯ КОНЦА ПЕРЕДАЧИ В АЦП.
{
    WHILE (RSTATUS () & 0x040) КВНТ ();
}

VOID WAITADCSPI (VOID) // ДОЖДАТЬСЯ КОНЦА ПЕРЕДАЧИ В АЦП.
{
    WHILE (RSTATUS () & 0x020) КВНТ ();
}
```

```

}

VOID WAITADCCONV (VOID)          // Получить готовность АЦП.
{
    WHILE (RSTATUS () & 0x010) КВНIT ();
}

// СБРОСИТЬ СОБСТВЕННО АЦП И ОЧИСТИТЬ ФЛАГИ КАНАЛА.

VOID INITBOARD ()
{
    WRIO (0x0C000, WADCH);          // Команда сброс АЦП.
    WRIO (0x08000, WADCH); WAITADCCONV (); // Сброс DRDY.
    WRIO (0x03000, WADCH); WAITADCSPI (); // Считать данные.
}

VOID SETADCCHAN (INT CHAN)        // Выбрать канал АЦП.
{
    WRIO (0x00000, WADCL);          // Установить GPIOC и GPIOD.
    WRIO (NUM2UNIT (CHAN) & 0x0FF00, WADCM1); // Установить MUXSG1 и SysRED.
    WRIO (NUM2UNIT (CHAN) & 0x000FF, WADCM2); // Установить MUXDIF и MUXSG0.
    WRIO (0x00000, WADCM3);          // Установить CONFIG1 и MUXSGH.
    WRIO (0x0700E, WADCH);          // Установить CONFIG0 и запустить.
}

// ЕДИНИЧНОЕ ПРЕОБРАЗОВАНИЕ ЗАДАННОГО КАНАЛА АЦП.
// НОМЕР КАНАЛА В ВЫХОДНЫХ ДАННЫХ СБРОШЕН.

LONG READADC (INT CHAN)           // НОМЕР КАНАЛА АЦП.
{
    LONG ADC;

    SETADCCHAN (CHAN); WAITADCSPI (); // Выбрать канал АЦП.
    WRIO (0x0008, COMADC); WAITADCCONV (); // Запустить преобразование.
    WRIO (0x0000, COMADC);          // Остановить преобразование.
    WRIO (0x03000, WADCH); WAITADCSPI (); // Считать данные.
    ADC = ((LONG) (RDIO (RADCM1) & 0x0FF) << 16 | ((LONG) RDIO (RADCL)) & 0xFFFFL);
    ADC = ADC << 8;                  // Расширение знака.
    RETURN (ADC >> 8);
}

// УСТАНОВИТЬ ЦАП. ЕСЛИ НОМЕР ЦАП >= 4 УСТАНОВЛИВАЮТСЯ ВСЕ КАНАЛЫ.

VOID SETDACs (UNSIGNED INT DATA, UNSIGNED INT DAC)
{

```



```
IF (DAC<4) {DAC=NUM2UNIT(DAC);} // ВЫБРАТЬ КАНАЛ.
ELSE {DAC=0x00F;}; // ВЫБРАТЬ ВСЕ КАНАЛЫ.
WAITDACSPI(); // КАНАЛ ГОТОВ К ПЕРЕДАЧЕ?
WRIO(DAC,0x011); // ВЫБРАТЬ КАНАЛ.
WRIO(DATA,0x010); // УСТАНОВИТЬ КАНАЛ.
}

// *****
// УПРАВЛЕНИЕ ЭКРАНОМ ЧЕРЕЗ INT16.

CLS()
{
UNION REGS REGS;
REGS.W.AX = 0x0003;
#ifdef __386__
INT386 (0x10, &REGS, &REGS);
#else
INT86 (0x10, &REGS, &REGS);
#endif
}

VOID PRINTAT(INT X,INT Y)
{
UNION REGS REGS;
REGS.W.AX = 0x0200;
REGS.W.DX = Y<<8|X;
REGS.W.BX = 0x0000;

#ifdef __386__
INT386 (0x10, &REGS, &REGS);
#else
INT86 (0x10, &REGS, &REGS);
#endif
}

UNSIGNED INT GETCURSOR ()
{ RETURN *(INT FAR *) MK_FP (0,0x460);
}

VOID SETCURSOR (UNSIGNED INT N)
{
UNION REGS REGS;
REGS.W.AX = 0x0100;
REGS.W.CX = N;
}
```

```

#ifdef __386__
    INT386 (0x10, &REGS, &REGS);
#else
    INT86 (0x10, &REGS, &REGS);
#endif
}

GETXY (INT *X, INT *Y)
{

    UNION REGS REGS;
    REGS.W.AX = 0x0300;
    REGS.W.BX = 0x0000;
#ifdef __386__
    INT386 (0x10, &REGS, &REGS);
#else
    INT86 (0x10, &REGS, &REGS);
#endif
    *X= (REGS.W.DX&0X0FFF);
    *Y= (REGS.W.DX>>8&0X0FFF);
}

CURSOROFF ()
{
    SETCURSOR (GETCURSOR() | 0x2000);
}

CURSORON ()
{
    SETCURSOR (GETCURSOR() & 0x0F0F);
}

// *****

GETBASEADDR ()
{
    INT VALIDADDR, KEY, X, Y, X1, Y1;
    DO
    {
        VALIDADDR=0; PRINTF ("\N ВВЕДИТЕ АДРЕС ПЛАТЫ: "); FFLUSH(STDOUT);
        GETXY (&X, &Y); SCANF ("%X", &BASEADDR); GETXY (&X1, &Y1); IF (Y==Y1) Y--;
        PRINTAT (X+8, Y); FFLUSH (STDOUT);
    }
}

```

```

IF (BASEADDR&3)
{
    PRINTF (" БАЗОВЫЙ АДРЕС %XН НЕ КРАТЕН 4.\n",BASEADDR);VALIDADDR=1;
}
ELSE
{
    IF (BASEADDR<0x200)
    {
        PRINTF (" АДРЕС %XН В СИСТЕМНОЙ ОБЛАСТИ, ВЫ УВЕРЕНЫ?",BASEADDR);
        FFLUSH (STDOUT);KEY=GETCH ();
        IF (KEY==89 || KEY==121 || KEY==141 || KEY==173) PRINTF (" YES.\n");
        ELSE {VALIDADDR=1;PRINTF (" No.\n");}
    }
    IF (BASEADDR>0x003FF)
    {
        PRINTF (" УКАЗАННЫЙ АДРЕС СЛИШКОМ ВЕЛИК.\n");VALIDADDR=1;
    }
}
}
WHILE (VALIDADDR);
BASEADDR=BASEADDR+2;
}

INT SETDAC (UNSIGNED INT *DAC, INT N)
{ INT X,Y; INT CNT=N;
  UNSIGNED INT *DAC=DAC;
  WHILE (CNT--!=0) DAC++;
  PRINTAT (4,22);
  PRINTF ("ВВЕДИТЕ ШЕСТНАДЦАТЕРИЧНОЕ ЗНАЧЕНИЕ ДЛЯ ЦАП#%D(%5X): ",N+1,*DAC);
  CURSORON (); SCANF ("%X",DAC); CURSOROFF ();
  GETXY (&X,&Y);
  PRINTAT (61,22); PRINTF (" ВЫВОД В ЦАП..."); FFLUSH (STDOUT);
  SETDACs (*DAC,N);
  IF (Y==23)
  {
    PRINTAT (2,22);
    PRINTF ("                               ");
    PRINTF ("                               ");
    FFLUSH (STDOUT);
    DRAWDACs (DAC);
    RETURN 0;          // ЭКРАН ОЧИЩЕН.
  }
  RETURN 1;          // ТРЕБУЕТСЯ ПЕРЕРИСОВКА ЭКРАНА.
}

```

```

}

INT SETDACF (UNSIGNED INT *DAC, INT N)
{
  INT X, Y; INT CNT=N;
  UNSIGNED INT *DAC=DAC;
  FLOAT FDAC;
  WHILE (CNT--!=0) DAC++;

  FDAC= ((FLOAT) ((LONG) *DAC) *16.) /65535.+4.0;
  PRINTAT (4, 22);
  PRINTF ("ВВЕДИТЕ ЗНАЧЕНИЕ ТОКА ДЛЯ ЦАП#%D (%5.5F) : ", N+1, FDAC);
  CURSORON (); SCANF ("%F", &FDAC); CURSOROFF (); GETXY (&X, &Y);
  PRINTAT (61, 22); PRINTF (" ВЫВОД В ЦАП..."); FFLUSH (STDOUT);
  IF (FDAC<4.0) FDAC=4.0; IF (FDAC>20.0) FDAC=20.0;
  FDAC= ((FDAC-4.0) *65535.0) /16.0;
  *DAC= (UNSIGNED INT) FDAC;
  SETDACs (*DAC, N);
  IF (Y==23)
  {
    PRINTAT (2, 22);
    PRINTF ("");
    PRINTF ("");
    FFLUSH (STDOUT);
    DRAWDACs (DAC);
    RETURN 0;          // ЭКРАН ОЧИЩЕН.
  }
  RETURN 1;          // ТРЕБУЕТСЯ ПЕРЕРИСОВКА ЭКРАНА.
}

// ВЫВЕСТИ КАНАЛЫ НА ЭКРАН В ЗАДАнные ПОЗИЦИИ.

DRAWADC (FLOAT *I)
{
  INT CHAN = 0;
  INT X = 24;
  INT Y = 7;
  FLOAT I;
  DO
  {
    I=*I*IMAX/ (FLOAT) 0x07FFFFFFL;
    IF (CHAN==8) {x=53; y=-1;};
    PRINTAT (X, (CHAN+Y));
    IF (I<10.0&&I>0.) PRINTF ("%10LD (%5.5F МА)", (LONG) *I, I);
    ELSE PRINTF ("%10LD (%5.5F МА)", (LONG) *I, I);
  }
}

```

```
        FFLUSH (STDOUT) ; I++;
    }
    WHILE (CHAN++!=15);
}

// ВЫВЕСТИ НА ЭКРАН УСТАНОВЛЕННЫЕ ЗНАЧЕНИЯ ЦАП.

DRAWDACs (UNSIGNED INT *I)
{ INT CHAN = 0; FLOAT N;
  DO
  {
    N= ((FLOAT) ((LONG) *I) *16) /65535.+4.0;
    PRINTAt (1, ((CHAN<<1)+7));
    IF (N<10.0) PRINTF ("%5XH (%5.5F mA) ", *I, N);
    ELSE      PRINTF ("%5XH (%5.5F mA) ", *I, N);
    I++; FFLUSH (STDOUT);
  }
  WHILE (CHAN++!=3);
}

// ПРЕОБРАЗОВАНИЕ ВСЕХ КАНАЛОВ АЦП;

CONVALL (FLOAT *I, FLOAT K, INT FON) { INT CHAN = 0;
  DO
  {
    IF (FON!=0) *I= (FLOAT) READADC (CHAN);
    ELSE *I= (( (FLOAT) READADC (CHAN) - *I) *K) + *I;
    I++;
  }
  WHILE (CHAN++!=15);
}

// ВКЛЮЧИТЬ/ВЫКЛЮЧИТЬ ФИЛЬТРАЦИЮ ДАННЫХ ДЛЯ АЦП.

FLTOnOFF (INT *F)
{
  *F=!*F; PRINTAt (46, 5);
  IF (*F==0) PRINTF (" (ФИЛЬТР ВКЛЮЧЕН) ");
  ELSE      PRINTF ("                ");
  FFLUSH (STDOUT);
}

DRAWHELLO (INT *FLTOn, UNSIGNED INT *DACSAVE)
{
```

```

CLS ();
CURSOROFF ();
PRINTAT (40,1);
PRINTF (" ПРОВЕРКА РАБОТЫ ПЛАТЫ KM1604. КАСКОД.");
FFLUSH (STDOUT);
PRINTAT (5,4); PRINTF (" УСТАНОВКА ЦАП:"); FFLUSH (STDOUT);
PRINTAT (50,4); PRINTF (" ЧТЕНИЕ АЦП:"); FFLUSH (STDOUT);
PRINTAT (15,16);
PRINTF ("КЛАВИША F ВКЛЮЧАЕТ/ВЫКЛЮЧАЕТ ФИЛЬТР ДЛЯ АЦП."); FFLUSH (STDOUT);
PRINTAT (14,17);
PRINTF ("КЛАВИШИ 1..4 УСТАНОВКА СООТВЕТСТВУЮЩЕГО ЦАП (HEX)."); FFLUSH (STDOUT);
PRINTAT (13,18);
PRINTF ("КЛАВИШИ F1..F4 УСТАНОВКА СООТВЕТСТВУЮЩЕГО ЦАП (FLOAT)."); FFLUSH (STDOUT);
PRINTAT (15,20);
PRINTF ("НАЖМИТЕ НА КЛАВИШУ ESC ДЛЯ ЗАВЕРШЕНИЯ РАБОТЫ."); FFLUSH (STDOUT);
FLTONOFF (FLTON);
DRAWDACs (DACSAVE);
}

RUNPROGRAMM ()
{
CHAR REDRAW = 1;
CHAR KEY = 0;
INT FLTON = 0;
UNSIGNED INT DACSAVE [4]={0,0,0,0};
FLOAT ADCSAVE [16]={0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};
FLOAT ADCDIV = 0.05;
INITBOARD ();
SETDACs (0,-1);
PRINTF ("\n ИНИЦИАЛИЗАЦИЯ ПЛАТЫ..."); FFLUSH (STDOUT);
CONVALL (ADCSAVE, ADCDIV, 0);
CLS ();
DO
{
IF (REDRAW) {DRAWHELLO (&FLTON, DACSAVE); REDRAW=0;}
CONVALL (ADCSAVE, ADCDIV, FLTON);
DRAWADC (ADCSAVE);
IF (KBHIT ())
{
KEY=GETCH (); // PRINTAT (0,0); PRINTF (" KEY= %D", KEY); FFLUSH (STDOUT);
IF (KEY != 0)
{
IF (KEY == 102 || KEY == 160 || KEY == 70 || KEY == 128) FLTONOFF (&FLTON);
IF (KEY == 49 || KEY == 33) REDRAW=SETDAC (DACSAVE, 0);
}
}
}
}

```

```
IF (KEY == 50 || KEY == 34 || KEY == 64) REDRAW=SETDAC (DACSAVE, 1);
IF (KEY == 51 || KEY == 35) REDRAW=SETDAC (DACSAVE, 2);
IF (KEY == 52 || KEY == 36) REDRAW=SETDAC (DACSAVE, 3);
}
ELSE
{
KEY=GETCH();
IF (KEY == 59) REDRAW=SETDACF (DACSAVE, 0);
IF (KEY == 60) REDRAW=SETDACF (DACSAVE, 1);
IF (KEY == 61) REDRAW=SETDACF (DACSAVE, 2);
IF (KEY == 62) REDRAW=SETDACF (DACSAVE, 3);
}
}
}
WHILE (KEY != LEAVEKEY);
CURSORON(); PRINTAT (15,20);
PRINTF (" GOOD LUCK! ;) \N\n");
}

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

VOID MAIN(VOID)
{
BASEADDR = 0x302; // БАЗОВЫЙ АДРЕС ПО УМОЛЧАНИЮ;
GETBASEADDR();
PRINTF (" ПОИСК ПЛАТЫ...");FFLUSH(STDOUT);
IF (RDIO(0x0FF) >> 8 == 0x010) RUNPROGRAMM();
ELSE
{ BASEADDR=BASEADDR-2; PRINTF ("\N\n ПЛАТА KM1604 ПО АДРЕСУ ");
PRINTF ("%D (%XH) НЕ НАЙДЕНА.\N", BASEADDR, BASEADDR);
}
}
```