

# **Модуль EDSP32**

**Версия 4.0**

**Руководство пользователя**

**АО КАСКОД**

**1999**

**Санкт-Петербург**

## **АО КАСКОД**

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

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

E-mail : [cascod@online.ru](mailto:cascod@online.ru)  
[kaskod@spb.cityline.ru](mailto:kaskod@spb.cityline.ru)

<http://www.kaskod.ru>

<b>Содержание</b>	<b>Страница</b>
Введение .....	4
Спецификация EDSP32 .....	5
Структура модуля EDSP32 .....	6
<b>Раздел 1.</b>	
1.1. Варианты подключения .....	8
1.2. Описание разъемов: .....	8
– последовательный порт RS-232/RS-485 .....	9
– многофункциональный последовательный порт TMS320C32 .....	9
– внешняя 32 разрядная буферированная шина TMS320C32 .....	10
– шина PC/104 .....	11
– порт MPSD .....	12
– питание .....	12
1.3. Описание джамперов .....	13
1.4. Конфигурация платы: .....	15
– автономный режим .....	15
– режим сопроцессора .....	15
1.5. Конфигурация стартового загрузчика .....	15
<b>Раздел 2.</b>	
2.1. Адреса устройств шины PC/104: .....	16
– базовый адрес EDSP32 .....	16
– адреса двухпортового ОЗУ .....	16
– адреса регистра управления .....	16
2.2. Адреса устройств TMS320C32: .....	17
– интерфейс памяти TMS320C32 .....	17
– карта памяти .....	18
– формат данных .....	19
– конфигурация строб-сигналов .....	19
– начальные установки TMS320C32 .....	20
– примеры управления режимами TMS320C32 через шину PC/104 .....	20
2.3. Двухпортовое ОЗУ .....	22
<b>Раздел 3.</b>	
Примеры программ: .....	24
Программирование Flash памяти .....	24
Интерфейса RS-232/RS-485 .....	25
<b>Раздел 4.</b>	
Распределенная система разработки и отладки программ для EDSP32 .....	26
<b>Приложения</b> .....	<b>27</b>

## Введение

---

**EDSP32** представляет собой модуль для цифровой обработки сигналов выполненный на высокопроизводительном 32-х разрядном сигнальном процессоре с плавающей точкой TMS320C32 (изготовитель фирма *Texas Instruments*), в конструктиве PC/104 .

Плата обеспечивает 2 режима работы:

- **режим сопроцессора** (параллельная работа с PC-компьютером через двухпортовое ОЗУ.  
Загрузка программ из двухпортового ОЗУ.  
Загрузка программ из Flash памяти.  
Загрузка программ через последовательный порт или через внешнюю 32-х разрядную шину);
- **автономный режим** работы (работа платы без PC-компьютера с загрузкой программы из Flash памяти, установленной на плате, либо через последовательный порт, либо внешнюю 32-х разрядную шину).

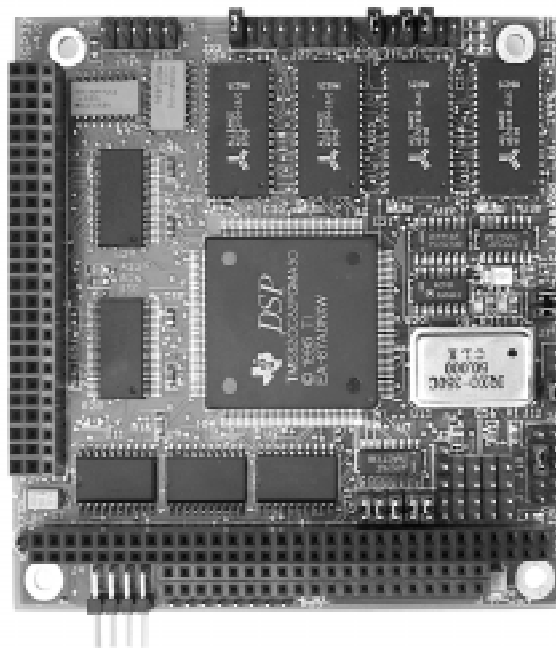


Рис. 1.  
Внешний вид  
модуля EDSP32

## Спецификация EDSP32

---

- 32-х разрядный RISC процессор TMS320C32 с плавающей точкой, 24-х разрядной адресной шиной ;
- время выполнения команды процессора :
  - 50ns/40МГц TMS320C32-40;
  - \*40ns/50МГц TMS320C32-50;
  - \*33ns/60МГц TMS320C32-60;
- два банка внутреннего ОЗУ 256 x 32 бит;
- внутренняя кэш память программ 64 x 32 бит;
- два банка внешнего статического ОЗУ без тактов ожидания общим объемом до 1 Мбайт (2 x 128K x 32 бит);
- двухпортовое ОЗУ объемом 8Kx16 бит без тактов ожидания, которое обеспечивает связь DSP процессора с шиной PC/104, где DSP процессор является ведомым (режим slave) через окно размером 128K в области памяти PC, базовый адрес окна выбирается джамперами;
- FLASH память объемом 128 Кбайт, 3 такта ожидания;
- скоростной 15 Мбит/с синхронный последовательный канал;
- RS232/RS485 буфер для построения программно-асинхронного последовательного канала из синхронного со скоростью – 9600бод;
- два внутренних 32-х разрядных таймера;
- два канала прямого доступа с возможностью задания приоритета;
- возможность начальной загрузки программ с шины PC/104, из FLASH памяти, с 32-х разрядной внешней шиной или по последовательному каналу;
- внешняя 32-х разрядная буферизированная шина в пространстве адресов IOSTRB DSP процессора т.е 128K x 32 для подключения дополнительной памяти , АЦП/ЦАП или других внешних устройств;
- два режима пониженного питания, задаваемых по команде процессора:
  - спящий режим (IDLE2),
  - режим с пониженной тактовой частотой (LOWPOWER);
- MPSD интерфейс;
- питание: 5 В ± 5%, I<sub>ср.</sub> = 800 мА ;
- размер платы 90 x 96 mm (PC/104);
- рабочая температура 0°C ... +70°C.

\* – возможные дополнительные опции

# Структурная схема модуля

Структурная схема модуля приведена на рисунке 2.

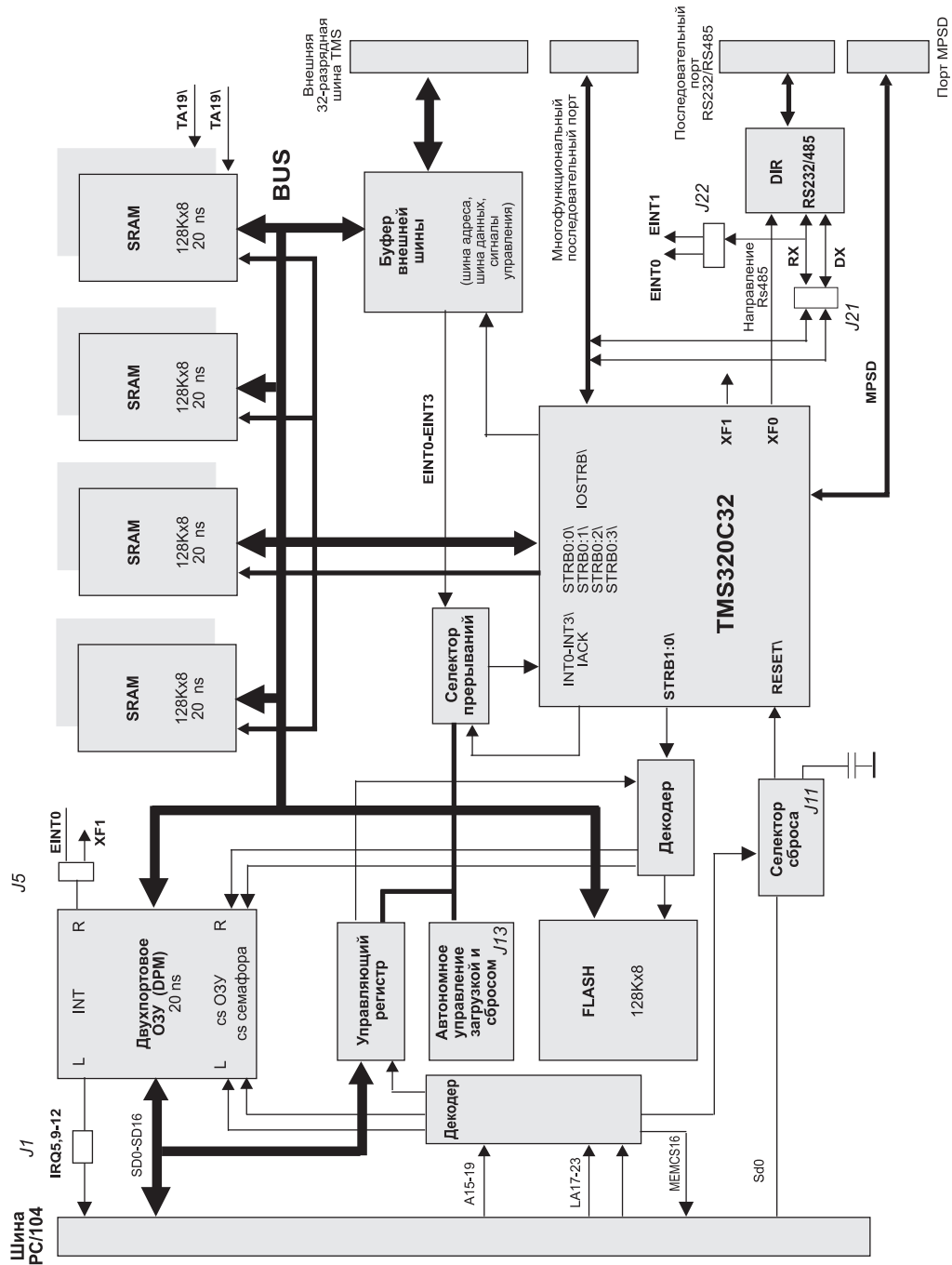


Рис. 2.  
Структурная  
схема

К шине DSP процессора подключены банки внешнего ОЗУ, двухпортовое ОЗУ (DPM), FLASH и буферы внешней шины.

Внешнее ОЗУ адресуется в пространстве адресов строга STRB0; FLASH и DPM адресуется в пространстве адресов строга STRB1, в зависимости от установленного бита <BOOT> управляющего регистра.

Внешняя буферизованная 32-х разрядная шина адресуется в пространстве адресов строга IOSTRB.

От процессора на внешний разъем через последовательные сопротивления выведен скоростной последовательный канал.

К сигналам последовательного канала DX0, DR0 через джампер подключается буфер RS232/485 интерфейса, при этом приемный сигнал (DR0) запараллеливается джамперами на одно из двух внешних прерываний (EINT0, EINT1), что позволяет программным образом преобразовывать синхронный канал в асинхронный и наоборот.

С помощью флага процессора XF0 задается направление (прием/передача) для буфера в режиме RS485.

Доступ к внешней памяти (ОЗУ) осуществляется с максимальным быстродействием процессора (0WS).

С помощью двухпортового ОЗУ (DPM) осуществляется начальная загрузка программ с помощью стартового загрузчика (Bootstraploader) и обмен данными между DSP процессором и шиной PC/104. Обмен с DPM может осуществляться при помощи «семафоров», прерываний или программного опроса (конфигурируется джампером J5 на плате).

От шины PC/104 при помощи регистра управления осуществляется управление сбросом, начальной загрузкой и битом BOOT.

Для задействования сигналов прерывания EINT0-EINT3 используется сигнал процессора  $\overline{ACK}$ , вырабатываемый после выполнения соответствующей команды процессора. По сбросу сигналами прерывания INT0-INT3 процессора управляет регистр управления.

В случае работы платы без шины PC/104 (автономно) управление сигналами прерываниями (загрузкой) осуществляется с помощью джамперов J13 селектора загрузки и J11 селектора сброса.

# Раздел 1.

## 1. 1. Варианты подключения

- непосредственно на шине PC/104 компьютера;
- с помощью переходной платы в PC-компьютер на шину ISA;
- автономно в режиме микрокомпьютера (с загрузкой программы из Flash).

## 1. 2. Описание разъемов

Расположение разъемов и джамперов представлено на рис. 3.

Затемненные области обозначают разъемы. Метками обозначен первый контакт как у разъемов, так и у джамперов.

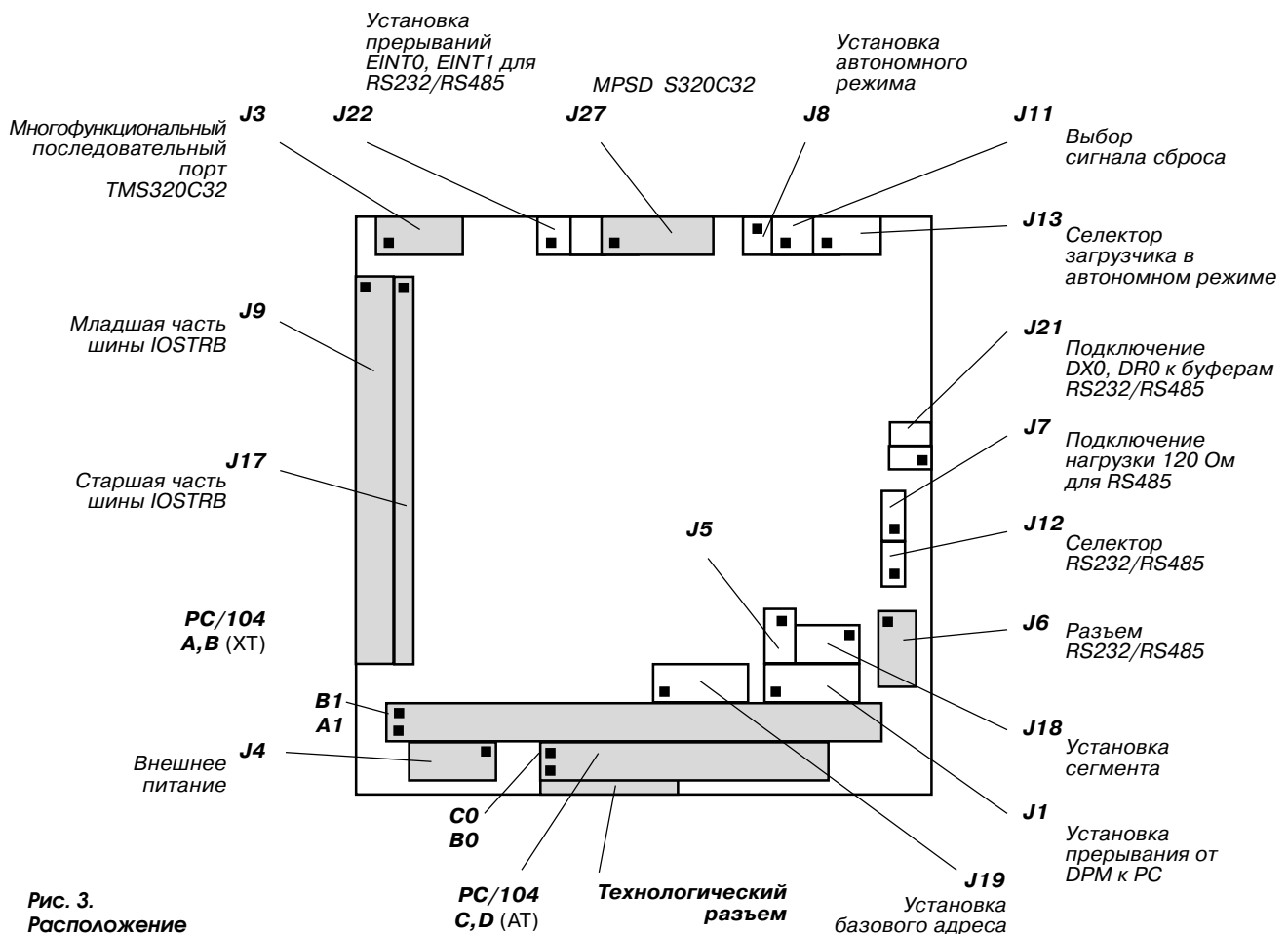


Рис. 3.  
Расположение разъемов и джамперов



**J6 – Последовательный порт RS232/RS485**

Таблица 1

Контакт	Обозначение	Описание
1	1-2-7	Перемычка 1-2-7
2	1-2-7	Перемычка 1-2-7
3	RIN / ARX	Прием 232/ прием 485 "А"
4	BTX	Передача 485 "В"
5	TOUT /ATX	Передача 232/ передача 485 "А"
6	BRX	Прием 485 "В"
7	1-2-7	Перемычка 1-2-7
8	N.C.	Не задействован
9	GND	Общий
10	GND	Общий

- Примечание:**
- выбор RS232 или RS485 осуществляется **J12**;
  - выходы имеют защиту от короткого замыкания и перегрева;
  - входы и выходы защищены от ESD  $\pm 10$  Кв.
- J7** – позволяет подключить для согласования 120 Ом параллельно линии RS485 непосредственно на плате.

**J3 – Многофункциональный последовательный порт TMS320C32**

Таблица 2

Контакт	Обозначение	Описание
1	RESET DSP	Сброс процессора (o)
2	DR0	Передача
3	DX0	Прием
4	CLKR0	Сигнал тактового синхронного приема
5	CLKX0	Сигнал тактовой синхронной передачи
6	FSR0	Сигнал блочного синхронного приема
7	FSX0	Сигнал блочной синхронной передачи
8	TCLK0	Таймер 0 (i/o)
9	TCLK1	Таймер 1 (i/o)
10	GND	Общий

- Примечание:** Последовательно со всеми выводами порта включены резисторы 33 Ом. Временные диаграммы, нагрузка и конфигурация соответствует описанию "TMS320C3X User's GUIDE".

**J9 – Внешняя 32-разрядная буферизированная шина (IOSTRB)**

Младшая часть

Таблица 3

Контакт	Обозначение	Описание
1,2	GND	Общий
3	$\overline{\text{EINT0}}$	Внешнее прерывание 0 (i)
4	$\overline{\text{EINT1}}$	Внешнее прерывание 1 (i)
5	$\overline{\text{EINT2}}$	Внешнее прерывание 2 (i)
6	$\overline{\text{EINT3}}$	Внешнее прерывание 3 (i)
7-23	EA0-EA16	Буферизированная адресная шина (o)
24-26	-	
27	XF0	Флаг XF0 (i/o)
28	TCLK0	Сигнал таймера0 (i/o)
29	CNTRL0	Буферизированный сигнал строба <b>iostrb\ (o)</b>
30	CNTRL1	Буферизированный сигнал чтения-записи <b>r/w(o)</b>
31	CNTRL2	Буферизированный сигнал сброса <b>tres\ (o)</b>
32,33	ED0,ED1	Буферизированный шина данных младшие 16 бит (i/o/z)
34,35	ED3,ED2	-“-
36,37	ED5,ED4	-“-
38,39	ED7,ED6	-“-
40,41	ED9,ED8	-“-
42,43	ED11,ED10	-“-
44,45	ED13,ED12	-“-
46,47	ED15,ED14	-“-
48	$\overline{\text{RDY}}$	Готовность (i)
49	+5V	Питание
50	GND	Общий

- Примечание:**
1. (i/o/z)- входы/выходы/третье состояние.  
EINT0-EINT3 четыре внешних входа прерываний.
  2. Буферизированные сигналы :  
 $I_{\text{OH}} = -32\text{mA}$  ,  $I_{\text{OL}} = -64\text{mA}$   
 $V_{\text{IH}} = 2\text{V min}$  ,  $V_{\text{IL}} = 0.8\text{V max}$

**J17 – Внешняя 32-разрядная буферизированная шина (IOSTRB)**

Старшая часть

Таблица 4

Контакт	Обозначение	Описание
1	GND	Общий
2	CNTRL3	Подтверждение прерывания <b>jack\ (o)</b>
3	-	
4	-	
5	ED16	Буферизированная шина данных старшие 16 бит (i/o/z)
6	ED17	Буферизированная шина данных старшие 16 бит (i/o/z)
7	ED18	Буферизированная шина данных старшие 16 бит (i/o/z)
8	ED19	Буферизированная шина данных старшие 16 бит (i/o/z)
9	ED20	Буферизированная шина данных старшие 16 бит (i/o/z)
10	ED21	Буферизированная шина данных старшие 16 бит (i/o/z)
11	ED22	Буферизированная шина данных старшие 16 бит (i/o/z)
12	ED23	Буферизированная шина данных старшие 16 бит (i/o/z)
13	ED24	Буферизированная шина данных старшие 16 бит (i/o/z)

14	ED25	Буферизированная шина данных старшие 16 бит (i/o/z)
15	ED26	Буферизированная шина данных старшие 16 бит (i/o/z)
16	ED27	Буферизированная шина данных старшие 16 бит (i/o/z)
17	ED28	Буферизированная шина данных старшие 16 бит (i/o/z)
18	ED29	Буферизированная шина данных старшие 16 бит (i/o/z)
19	ED30	Буферизированная шина данных старшие 16 бит (i/o/z)
20	ED31	Буферизированная шина данных старшие 16 бит (i/o/z)
21	TCLK1	Сигнал таймера 1 (i/o)
22	XF1	Флаг XF1 (i/o)
23	-	
24	-	
25	+5V	Питание

### Шина PC/104

Таблица 5

A	Сигнал	B	Сигнал	C	Сигнал	D	Сигнал
1	-	1	GND	0	GND	0	GND
2	SD7	2	SRESET	1	SBHE	1	MEMC16\
3	SD6	3	+5V	2	BA23	2	-
4	SD5	4	IRQ9	3	BA22	3	IRQ10
5	SD4	5	-5V	4	BA21	4	IRQ11
6	SD3	6	-	5	BA20	5	IRQ12
7	SD2	7	-12V	6	BA19	6	-
8	SD1	8	-	7	BA18	7	-
9	SD0	9	+12V	8	BA17	8	-
10	-	10	(KEY)	9	MEMR	9	-
11	-	11	-	10	MEMW	10	-
12	SA19	12	-	11	SD8	11	-
13	SA18	13	-	12	SD9	12	-
14	SA17	14	-	13	SD10	13	-
15	SA16	15	-	14	SD11	14	-
16	SA15	16	-	15	SD12	15	-
17	SA14	17	-	16	SD13	16	+5V
18	SA13	18	-	17	SD14	17	-
19	SA12	19	-	18	SD15	18	GND
20	SA11	20	-	19	(KEY)	19	GND
21	SA10	21	-				
22	SA9	22	-				
23	SA8	23	IRQ5				
24	SA7	24	-				
25	SA6	25	-				
26	SA5	26	-				
27	SA4	27	-				
28	SA3	28	BALE				
29	SA2	29	+5V				
30	SA1	30	-				
31	SA0	31	GND				
32	GND	32	GND				

В таблице прочерком указаны не используемые сигналы.

**Примечание:** SRESET подключен через резистор 20K к GND;

MEMW, MEMR, SBHE – через резистор 20K к +5V.

**J27 – Порт MPSD** (не буферизированный)

Таблица 6

Контакт	Обозначение	Описание
1	EMU1	(i)
3	EMU0	(i)
5	EMU2	(i)
7	+5V	Выход питания +5В
8	N.C.	Не задействован
9	EMU3	(o)
11	H3	Процессорные такты (o)
2,4,6,10,12	GND	Общий питания

**Примечание:** см. "TMS320C3X User's Guide" .

**J4 – Питание**

Таблица 7

Контакт	Обозначение	Описание
1,7	GND	Общий
2,8	+5V	+5В
<b>3</b>	<b><math>\overline{\text{RST}}</math></b>	<b>Внешний сброс процессора (i)</b>
4	+12V*	+12В шины PC104 (транслирование)
5	-5V*	-5В шины PC104 (транслирование)
6	-12V*	-12В шины PC104 (транслирование)

**Примечание:**  $\overline{\text{RST}}$  активен, когда замкнуты контакты **1-2 J11**.

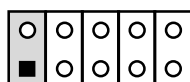
\*+12,-5,-12 платой не используются.

### 1.3. Описание джамперов

Ориентация джамперов соответствует расположению платы разъемом PC/104 вниз как на рис. 3.

#### J1 – Выбор номера прерывания PC/104 (IRQ5,9-12)

Сигнал прерывания вырабатывается двухпортовым ОЗУ на шину PC104 при обращении TMS320C32 по адресу 1FFEh. При считывании двухпортового ОЗУ по BASE+1FFEh с шины PC104 происходит сброс прерывания в ноль.



5 9 10 11 12 – номер IRQ

\* штриховка обозначает установленный джампер.

#### J8 – Селектор автономного режима загрузки:

- разомкнут – автономный режим;
- замкнут – управление загрузкой и сбросом от PC.

#### J11 – Селектор сброса (от PC/104 или внутренний)

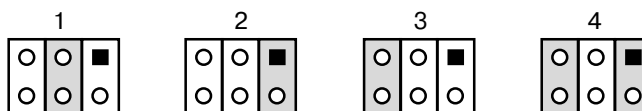


Сброс от PC



Сброс внутренний

#### J13 – Селектор загрузчика при автономном режиме

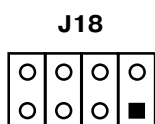


- |   |   |   |
|---|---|---|
| 1 | - | INT2 BOOT3 900000H STRB1 FLASH                |
| 2 | - | INT1 BOOT2 810000H IOSTRB (параллельный порт) |
| 3 | - | INT3 SERIAL PORT 32 BIT                       |
| 4 | - | INT1,INT3 BOOT2 810000H IOSTRB с XF0,XF1      |

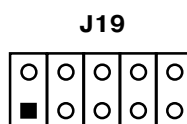
#### J18 – Установка адреса окна PC/104 выше первого мегабайта

#### J19 – Установка адреса окна в пределах первого мегабайта

- базовый адрес окна памяти со стороны хост процессора устанавливается с помощью комбинации джамперов J18 и J19



A23,A22,A21,A20



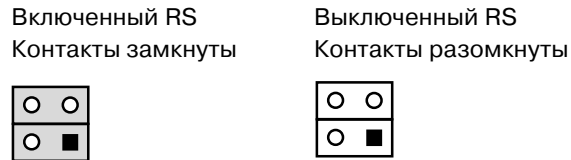
A15:A16,A17,A18,A19

**Установленный джампер соответствует лог.нулю !!!**

Пример установки адреса:

J18 = 0 0 0 0	J19 = 0 : 0 0 1 1	\ 0C:0xxx
1 0 1 0	1 : 0 0 0 0	\ A0:8xxx
1 1 1 1	1 : 1 0 0 0	\ F1:8xxx 16 Мб

### J21 – Подключение буфера RS232/RS485



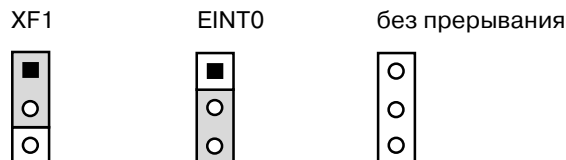
### J22 – Выбор прерывания EINT0, EINT1 для RS232/RS485



### J5 – Селектор XF1 или EINT0 при работе с двухпортовым ОЗУ по прерываниям

При обращении к двухпортовому ОЗУ по адресу BASE+1FFFh устанавливается сигнал прерывания TMS320C32 (ноль).

При считывании ячейки по этому адресу TMS320C32 сигнал прерывания устанавливается в единицу (неактивен).



**J7** – При замыкании контактов подключается 120 Ом между выводами А и В RS485 канала.

### J12 – Выбор RS232/RS485

- |                       |          |                 |
|-----------------------|----------|-----------------|
| – контакты замкнуты   | – RS232  | J7– разомкнуть; |
| – контакты разомкнуты | – RS485. |                 |

## 1.4. Конфигурация платы

Плата работает в следующих режимах:

### Автономный режим

В автономном режиме плата работает независимо от шины PC104, а программа загружается либо из FLASH памяти, либо по последовательному каналу, либо из параллельного порта в зависимости от установленных джамперов на плате .

- **J5** разомкнуть;
- **J8** разомкнуть;
- **J11** разомкнуть – внутренний сброс при включении питания;
- **J13** установить джампер в положение 1 (загрузка из Flash).

### Режим сопроцессора

- **J8** установить переключку;
- **J11** в положение **3-4** программный или аппаратный сброс от PC104;
- **J5** в одно из положений:
  - 1** – при работе с двухпортовым ОЗУ по прерыванию используя **XF1**;
  - 2** – при работе с двухпортовым ОЗУ по прерываниям используя **EINT0**;
  - 3** – либо разомкнутое состояние;

*Примечание:* При работе с прерыванием необходимо в инициализации процессора перед разрешением прерываний вставить команду чтения ячейки памяти с адресом 901FFFh – сброс прерывания от DPM.

## 1.5. Конфигурация стартового загрузчика(BootstrapLoader)

(биты **INT0-INT3** см. в разделе «Регистры управления»)

<b>N</b>	<b><math>\overline{INT0}</math></b>	<b><math>\overline{INT1}</math></b>	<b><math>\overline{INT2}</math></b>	<b><math>\overline{INT3}</math></b>	<b>Описание</b>
1	1	0	1	1	BOOT2 810000H - IOSTRB
2	1	1	0	1	BOOT3 900000H - DPM/FLASH
3	1	1	1	0	Последовательный порт
4	1	0	1	0	BOOT2 810000H с XF0/XF1-IOSTRB

Таблица 9

- 1 – загрузка программы через внешнюю 32-х разрядную буферизованную шину;
- 2 – загрузка программы из двухпортового ОЗУ или FLASH памяти;
- 3 – загрузка программы через последовательный порт J3;
- 4 – загрузка программы через внешнюю 32-х разрядную буферизованную шину с рукопожатием по флагам XF0, XF1.

# Раздел 2.

## 2. 1. Адреса устройств шины PC/104

### Базовый адрес EDSP32

Плата работает на шине через двухпортовое ОЗУ в окне памяти хост процессора, которое может устанавливаться во всем адресном пространстве шины PC/104. При установке базового адреса необходимо убедиться, что эта область памяти свободна или освободить ее с помощью менеджера памяти EMM386 (QEMM386).

*Например:* При установке базового адреса 0C:8000 дописать строку в config.sys  
(DEVICE= ..\EMM386.EXE x=C800-CFFF)

*Например:* Если J18 – все разомкнуты, J19 – все замкнуты, установлен адрес окна F00000h PC компьютера, что соответствует сегменту памяти между 15-16 Мб. При этом необходимо убедиться, что в этом пространстве адресов нет памяти. При ее наличии воспользоваться “сетапом” PC для образования окна в области 15-16 Мб.

**Примечание:** Окно памяти устанавливается с дискретностью 32 Кб.

### Адреса двухпортового ОЗУ

**адреса в два раза больше со стороны PC, чем со стороны DSP!!**  
т.к. адреса смешены на один (Sa0 использован для определения младшего байта)

<b>BASE+0000h</b>	–	<b>BASE+3FFAh</b>	– адреса ячеек памяти двухпортового ОЗУ
<b>BASE+3FFEH</b>	–	<b>Установка прерывания</b>	
<b>BASE+3FFCH</b>	–	<b>Сброс прерывания</b>	
<b>BASE+6000h</b>	–	Семафор 0	
<b>BASE+6002h</b>	–	Семафор 1	
<b>BASE+6004h</b>	–	Семафор 2	
<b>BASE+6006h</b>	–	Семафор 3	
<b>BASE+6008h</b>	–	Семафор 4	
<b>BASE+600Ah</b>	–	Семафор 5	
<b>BASE+600Ch</b>	–	Семафор 6	
<b>BASE+600Eh</b>	–	Семафор 7	

### Адреса регистра управления

**BASE+5000h** – адрес регистра управления загрузкой;

Таблица 10

SD0	SD1	SD2	SD3	SD4	SD5	SD6	SD7
INT0	INT1	INT2	INT3	-	-	-	BOOT



INT0-INT3: установка стартового загрузчика (см. Табл. 9).

- BOOT:
- 1 – блокирование обращения к DPM, разрешение обращения к FLASH памяти (запись, чтение), разрешение загрузки программы из FLASH памяти при старте DSP процессора .
  - 0 – разрешение обращения к DPM и загрузки программы при старте из DPM при старте DSP процессора, при этом обращение к FLASH памяти блокируется.

**BASE+4000h** – адрес защелки бита SD0 установки “старт/стоп”:

- SD0= 0 - сброс ;
- SD0= 1 - старт ;

## 2. 2. Адреса устройств TMS320C32

### Интерфейс памяти TMS320C32

001000-7FFFFFFh	–	STRB0
808000-8097FFh	–	RESERVED в кристалле TMS320C32
810000-82FFFFh	–	IOSTRB область 128K x 32 портовой разъем
87FE00-87FEFFh	–	256 внутренней RAM0
87FF00-87FFFFh	–	256 внутренней RAM1
880000-8FFFFFFh	–	STRB0
900000-FFFFFFh	–	STRB1

#### – STRB0:0-3

020000-03FFFFh	–	SRAM банк 0 128Kx32
0A0000-0BFFFFh	–	SRAM банк 1 128Kx32

#### – STRB1:0

при BOOT=0 (DPM и FLASH разделяются адресным дешифратором)

- FLASH 128Kx8
  - D00000-D1FFFFh – чтение и запись во FLASH
- DPM 8K x 16
  - 900000-901FFFh – чтение/запись
  - 901FFE запись – выставить прерывание PC
  - 901FFF чтение – сброс запроса прерывания от PC
  - 902000-902007h адреса 8-ми **семафоров**;

при BOOT=1 (обращение только к FLASH в адресном пространстве STRB1)

- FLASH
  - D00000-D1FFFFh – чтение и запись во FLASH(многозначность)
  - 900000-91FFFFh – начальная загрузка (многозначность)

**Карта памяти**

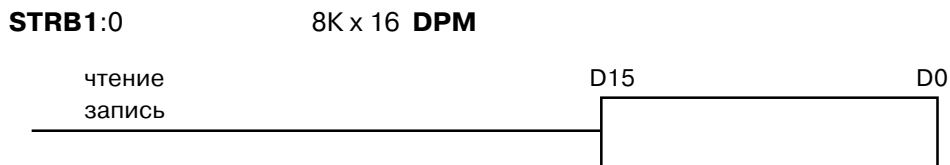
(DSP процессор используется в режиме микрокомпьютера с 32-х разрядным кодом программ: MKBL/MP=1, PRGW=0).

000000	–	Стартовый загрузчик TMS (BootstrapLoader)
000FFF		
001000	–	Зарезервировано (STRB0)
01FFFF		
020000	–	Внешнее <b>RAM0</b> (128Kx32) ( <b>STRB0</b> )
03FFFF		
040000	–	Зарезервировано (STRB0)
09FFFF		
0A0000	–	Внешнее <b>RAM1</b> (128Kx32) (STRB0)
0BFFFF		
0C0000	–	Зарезервировано (STRB0)
7FFFFFFF		
800000	–	Зарезервировано 32K
807FFF		
808000	–	Регистры TMS 6K
8097FF		
809800	–	Зарезервировано
80FFFF		
810000	–	Внешняя шина (128Kx32) ( <b>IOSTRB</b> ) [ <b>BOOT2</b> ]
82FFFF		
830000	–	Зарезервировано
87FDFF		
87FF00	–	RAM Block 1 (256 Word Internal)
87FE00		
87FF00	–	RAM Block 1 (256 Word Internal)
87FFFF		
900000	–	DPM ( <b>STRB1</b> ) [ <b>BOOT3</b> ]
901FFF		
902000	–	Семафоры DPM (STRB1)
902007		
902008	–	Зарезервировано
CFFFFFFF		
D00000	–	FLASH
D1FFFF		
D20000	–	Зарезервировано
FFFFFFF		

Зарезервированные области памяти не используются.

**Формат данных**

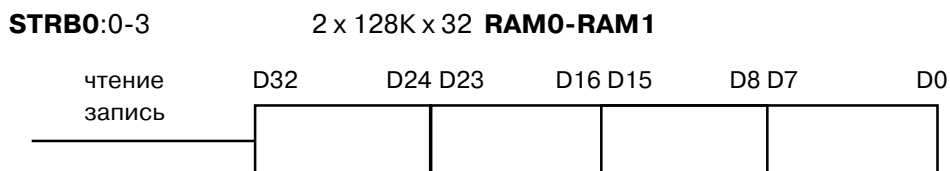
16-разрядное двухпортовое ОЗУ использует младший строб сигналов STRB1n. Разряды с D16-D32 не значащие. Данное устройство выбрано при установленном бите BOOT=0.



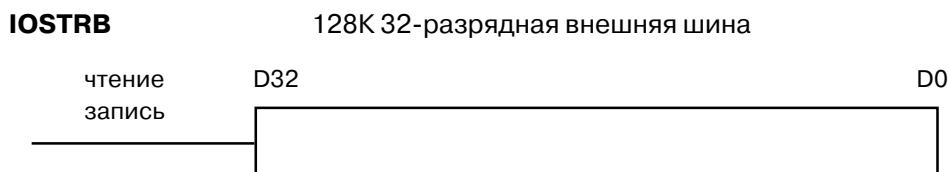
8-разрядная FLASH использует также младший сигнал STRB1.



Два банка внешнего ОЗУ используют все сигналы STRB0.



IOSTRB используется только для буферизированной внешней шины на все 32 разряда.

**Конфигурация строб-сигналов**

Строб-сигналы процессора конфигурируются как 32-х разрядная шина и 32-х разрядные данные.

	strb config	mem width	data size					
STRB0 CNTRL REG	XXX	0	XXXX	1	1	1	1	XXX
	mem data width size							
STRB1 CNTRL REG	XXX	XXXX	1	1	1	1	XXX	

## Начальные установки TMS320C32

### – Установка строб-сигналов:

**\$078;** = **IOSTRB** Control :

WTCNT=	<b>011</b>	(3 wait cycls)
SWW=	<b>11</b>	(RDY/ AND RDYint/)

**\$0F0518;** = **STRB0** Control :

HIS=	<b>0</b>	
NOHOLD=	<b>0</b>	
BNCMP=	<b>00101</b>	(512K)
WTCNT=	<b>000</b>	(0 wait cycls)
SWW=	<b>11</b>	(RDY/AND RDYint/)
Data Type Size=	<b>11</b>	(32bits)
Physical Memory		
Width=	<b>11</b>	(32 bits)
Sign		
Ext/Zero Fill=	<b>0</b>	(Sign Ext)
STRB Config=	<b>0</b>	
STRB Switch=	<b>1</b>	

**\$0F0B18;** = **STRB1** Control:

BNCMP=	<b>01011</b>	(8K)
WTCNT=	<b>000</b>	(0 wait cycls)

**Примечание:** При работе с FLASH необходимо понижать скорость STRB1 до 3-х тактов ожидания.

SWW=	<b>11</b>	(RDY/ AND RDYint/)
Data Type Size=	<b>11</b>	(32 bits)
Physical		
Memory Width=	<b>11</b>	(32 bits)
Sign		
Ext/Zero Fill=	<b>0</b>	(Sign Ext)

### – Установка флагов:

**IOF=6**    **XF0** – Устанавливается на вывод.  
**XF1** – Устанавливается на ввод.

## Примеры управление режимами TMS320C32 через шину PC/104

### **Управление от PC/104 сбросом, загрузкой и стартом**

- **SET RESET** ( RESET=0, BOOT=0, INT0-INT3=1 );
  - 0000h по адресам BASE+0..BASE+3FFEh    \\ очистить DPM
  - 0000h по адресу BASE+4000h            \\ TMS сброс - SD0 ->0
  - 000fh по адресу BASE+5000h            \\ boot -> 0 , intx=1
- записать в двухпортовое ОЗУ – блок программы для загрузчика с соответствующим заголовком.
- **SET START** ( Set RESET=1 , INT2= импульс );
  - 0001h по адресу BASE+4000h            \\ старт TMS - SD0 ->1
  - 000bh по адресу BASE+5000h            \\ int2=0 (загрузка из DPM)
  - 000fh по адресу BASE+5000h            \\ int2=1

**Пример загрузки (загрузка из DPM):**

**SET RESET;**     \\ см. выше  
                  \\ начало блока

DATA	ADDRES	
0010h	BASE+00	Width memory source
0000h	BASE+02	
0078h	BASE+04	IOSTRB
0000h	BASE+06	
05F8h	BASE+08	STRB0 ( 4F05F8h )
000Fh	BASE+10	
0BF8h	BASE+12	STRB1
000Fh	BASE+14	
0004	BASE+16	Size of block
0000	BASE+18	
0000h	BASE+20	Dest; addr;
0002h	BASE+22	
F864h	BASE+24	Dest; STRB0+low addr;
0F05h	BASE+26	
		USER'S PROGRAMM START:
4080h	BASE+28	LDI 4080 , AR0
0868h	BASE+30	
0009h	BASE+32	LSH 9 , AR0
09E8h	BASE+34	
C000h	BASE+36	STI R0, *AR0
1540h	BASE+38	
0000h	BASE+40	BR 20000h -
6002h	BASE+42	
		START. ADDR. EXT RAM
		USER'S PROGRAM END;
0000h	BASE+44	Size of block = 0
0000h	BASE+46	
0000h	BASE+48	Size of block = 0
0000h	BASE+50	

**SET START ;**     \\ конец блока  
                  \\ см. выше

- Процессор устанавливается на сброс.
- Двухпортовое ОЗУ заполняется кодом начальной инициализации загрузчика в соответствии с описанием процессора для размера шины 16 бит.
- Далее следует исполняемый код.
- Коды конца блока.
- После снятия сброса процессора происходит начальная загрузка программы из двухпортового ОЗУ во внутреннее ОЗУ платы или процессора, после чего начальный загрузчик передает управление процессору для выполнения программы.

## 2.3. Двухпортовое ОЗУ

Двухпортовое ОЗУ является асинхронным устройством, которое обеспечивает двухсторонний доступ со стороны PC-компьютера (левый порт) и DSP процессора (правый порт) к одной и той же области памяти, т.е. независимое чтение и запись в любую область памяти.

Возможен арбитраж по семафорам и прерываниям.

### – Арбитраж по прерываниям

Для установки флага прерывания с правой стороны необходимо записать данные в левый порт по адресу 3FFeH. Этот флаг сбрасывается после чтения правого порта по адресу 1FFfH.

Для установки флага прерывания с левой стороны необходимо записать данные в правый порт по адресу 1FFeH. Этот флаг сбрасывается после чтения левого порта по адресу 3FFCh.

16-битовая ячейка по этому адресу используется как обычная ячейка памяти.

### – Арбитраж по семафорам

Двухпортовое ОЗУ имеет 8 дополнительных бинарных ячеек памяти (флаги-семафоры).

Использование этих флагов позволяет слева и справа двухпортового ОЗУ установить необходимый приоритет доступа к областям памяти. Например, семафор может быть использован устройством для запрещения доступа другому устройству к области двухпортового ОЗУ (защита блока данных и т.п.).

### – Доступ к семафору

При записи в семафор используется только младший бит D0 . Если «0» записывается в неиспользуемый адрес семафора, то флаг семафора будет установлен в «0», и в «1» с другой стороны . Этот семафор может быть изменен только с той стороны, где флаг семафора установлен в «0».

Когда записана «1» в этот же семафор, флаг семафора будет установлен в «1» с двух сторон двухпортового ОЗУ.

Сторона, которая записала «0» в семафор, не дает установить флаг семафора в «0» с другой стороны.

«0» записанный в этот же семафор с другой стороны будет запоминаться в триггере семафорного запроса, пока семафор свободен с первой стороны.

При чтении семафорного флага, его величина распространяется на все биты слова, т.е. если флаг «1» – читаем «1» во всех разрядах.

### – Архитектура семафоров

Семафорная логика устанавливает 8 триггеров, которые независимы от памяти.

Эти триггеры могут быть применены для индикации использования разделенных ресурсов.

Семафор производит аппаратную поддержку для использования метода под названием *Token Passing Allocation*. В этом методе состояние семафорного триггера используется как token, для индикации использования разделенных ресурсов.

Если левый процессор хочет использовать эти ресурсы, он запрашивает token установкой семафорного триггера.

Затем этот процессор проверяет установку семафорного флага.

Если флаг «0», он продолжает работу с необходимой частью памяти.

Если нет – это значит, что правый процессор установил триггер первым, т.е. правый использует распределенные ресурсы.

Левый процессор может повторить запрос семафорного флага или отложить этот запрос для выполнения другой задачи, и затем снова попытаться повторить запрос через последовательную установку и проверку флага до получения управления.

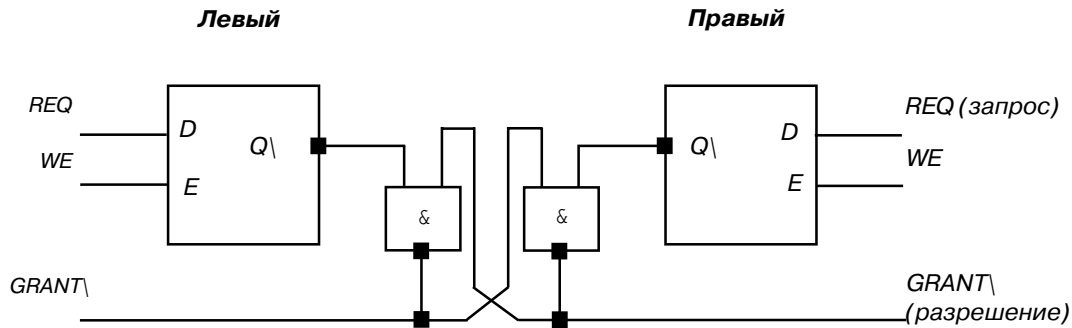


Рис.4. Семафорная защелка

Порт	Запись	Левый i/o	Правый i/o	Состояние
-	-	1	1	Семафор свободен
Левый	0	0	1	Левый порт получает разреш. доступ.
Правый	0	0	1	Нет изменения. Правый порт не имеет доступ к семафору
Левый	1	1	0	Правый порт получает разреш. доступ.
Левый	0	1	0	Нет изменения. Левый порт не имеет запись в семафор
Правый	1	0	1	Левый порт получает разреш. доступ.
Левый	1	1	1	Семафор свободен
Правый	0	1	0	Правый порт получает разреш. доступ.
Правый	1	1	1	Семафор свободен
Левый	0	0	1	Левый порт получает разреш. доступ.
Левый	1	1	1	Семафор свободен

Таблица 11

## Раздел 3.

---

---

### Примеры программ

#### Программирование FLASH памяти

Для программирования необходимо:

- Установить режим сопроцессора.
- Установить базовый адрес и загрузить **soft32-1.exe**.
- Открыть файл **Flash29.seq**, предназначенный для программирования FLASH памяти (open flash29 <enter>, e <enter>).
- В отмеченную строку текста вписывается имя двоичного файла **\*.pgm**, предназначенного для программирования, затем запускается трансляция (<**F10**>, ОК <enter>).

Сейчас на это место вписан файл t.pgm (демонстрация синтеза звука струны гитары при помощи цифрового фильтра).

Программа записи во FLASH автоматически формирует заголовки, необходимые для загрузки в режиме Boot loader из FLASH, производит очистку, программирование и проверку FLASH.

После программирования выйти из Forth системы.

При переходе в автономный режим установить загрузку из Flash.



## Программирование интерфейса RS232/RS485

Пример программы (TRS232.SEQ) преобразования последовательного канала TMS320C32 в стандартный асинхронный канал RS232/RS485 для связи с PC. Программа использует прерывания от приемника последовательного канала TMS и внешнее прерывание INT1 (необходимо правильно установить перемычки на плате).

Пристыковывается к файлу SOFT32-1.EXE .

### **Главные слова:**

<b>InitAll</b> ( — )	полная инициализация канала, переменных, прерываний;
<b>rs-in</b> ( — d )	прием байта по RS;
<b>rs-out</b> ( d — )	передача байта по RS;
<b>test-rsout</b> ( — )	тестовая программа; в цикле передает один и тот же байт C0h в RS;
<b>test-mirro</b> ( — )	тестовая программа; принимает, затем передает это же значение обратно, заполняя при этом буфер TDUAL-RAM первыми 100 значениями;

## Раздел 4.

### Распределенная система разработки и отладки для EDSP32

Система разработки – распределенный компилятор FORTH32.

Распределенный компилятор предназначен для написания и отладки программ, которые впоследствии будут прожигаться во FLASH память или загружаться в ОЗУ платы EDSP32.

Отладочная плата EDSP32, устанавливаемая в PC (PC104) может быть использована в качестве мощного вычислительного сопроцессора в компьютере.

Плата устанавливается в системную шину PC-IBM компьютера через переходную плату PC104/ISA.

Пользователь работает на PC в интерактивной среде, имеющей редактор и транслятор. Память делится на две части:

- область переменных (ОЗУ);
- область программ и констант (далее для прожигания во FLASH).

Обе эти области находятся во внешнем ОЗУ платы. Первая область будет называться областью ОЗУ, а вторая – областью FLASH. При трансляции исходного модуля код строится в памяти платы EDSP32. Пользователь может запускать слова (программы) на выполнение и следить за результатами. На экране PC выводится информация о количестве свободной памяти в ОЗУ и FLASH, а также состояние его стека данных. Отлаженный модуль может быть сохранен в виде файла \*.rgm для дальнейшего прожигания во FLASH пользовательской платы на базе кристалла TMS320C32.

Устанавливаемая в PC плата EDSP32 может быть использована в качестве мощного вычислительного сопроцессора внутри компьютера и полезна для создания пакетов обработки сигналов на PC. Так же плата EDSP32 может работать автономно по программе, записанной во FLASH память.

# Приложения

---

---