PB-DAC3

Opto-isolated 4-Channel Digital to Analog Piggyback
for VMOD-2 and IMOD

Order No. 5230-35/x
User's Manual
Publication No. 5230-UM-0101
Issue 2

Unpacking and Special Handling Instructions

This PepCard product is carefully designed for a long and fault-free life; nonetheless, its life expectancy can be drastically reduced by improper treatment during unpacking and installation.

Observe standard anti-static precautions when changing piggybacks, ROM devices, jumper settings, etc. If the product contains batteries for RTC or memory back-up, ensure that the board is not placed on conductive surfaces, including anti-static plastics or sponges. These can cause shorts and damage to the batteries or tracks on the board.

When installing the board, switch off the power mains to the chassis. Do not disconnect the mains as the ground connection prevents the chassis from static voltages, which can damage the board as it is inserted.

Furthermore, do not exceed the specified operational temperature ranges of the board version ordered. If batteries are present, their temperature restrictions must be taken into account.

Keep all of the original packaging material for future storage or warranty shipments. If it is necessary to store or ship the board, re-pack it as it was originally packed.

REVISION HISTORY
PB-DAC3 User's Manual
5230-UM-0101

Issue	Brief Description of Changes	PCB Index	Date of Issue
1	First Issue	01-01	October, 1993
2	Small changes throughout manual	01-01	March, 1994

PCB Index = Printed Circuit Board/Schematics Revision Number. This column provides a valid "from—to" range for the PCB index covered by each issue. These numbers combine to form the last four digits of the publication number. "/n" numbers suffixed to the index number show modifications to a local instruction note (addition of wires).

Date of issue = the release date of the issue. This date does not necessarily reflect the date the improvements were first made.

A thick bar to the left of any text indicates that the text has changed from the previous issue of the manual.

A double thin bar to the left of any text indicates that new text (paragraphs) has been added.
Changes and additions to drawings and boxed text, such as notes or jumper setting tables, cannot be marked in the above way; therefore this marking system is used only for changes to "normal" body text.

The information in this document is, to the best of our knowledge, entirely correct. However, **PEP *Modular Computers*** cannot accept liability for any inaccuracies, or the consequences thereof, nor for any liability arising from the use or application of any circuit, product, or example shown in this document.

PEP *Modular Computers* reserve the right to change, modify, or improve this document or the product described herein, as seen fit by **PEP *Modular Computers*** without further notice.

PEP *Modular Computers*® One Year Limited Warranty

We grant the original purchaser of **PEP** products the following hardware warranty. No other warranties that may be granted or implied by anyone on behalf of **PEP** are valid unless the consumer has the express written consent of **PEP *Modular Computers***.

PEP *Modular Computers* warrants their own products (excluding software) to be free from defects in workmanship and materials for a period of 12 consecutive months from the date of purchase. This warranty is not transferable nor extendible to cover any other consumers or long term storage of the product.

This warranty does not cover products which have been modified, altered, or repaired by any other party than **PEP *Modular Computers*** or their authorized agents. Furthermore, any product which has been, or is suspected of being damaged as a result of negligence, misuse, incorrect handling, servicing or maintenance; or has been damaged as a result of excessive current/voltage or temperature; or has had its serial number(s), any other markings, or parts thereof altered, defaced, or removed will also be excluded from this warranty.

A customer who has not excluded his eligibility for this warranty may, in the event of any claim, return the product at the earliest possible convenience, together with a copy of the original proof of purchase, a full description of the application it is used on, and a description of the defect; to the original place of purchase. Pack the product in such a way as to ensure safe transportation (we recommend the original packing materials), whereby **PEP** undertakes to repair or replace any part, assembly or sub-assembly at our discretion; or, to refund the original cost of purchase, if appropriate.

In the event of repair, refund, or replacement of any part, the ownership of the removed or replaced parts reverts to **PEP *Modular Computers***, and the remaining part of the original guarantee, or any new guarantee to cover the repaired or replaced items, will be transferred to cover the new or repaired items. Any extensions to the original guarantee are considered gestures of goodwill, and will be defined in the "Repair Report" returned from **PEP** with the repaired or replaced item.

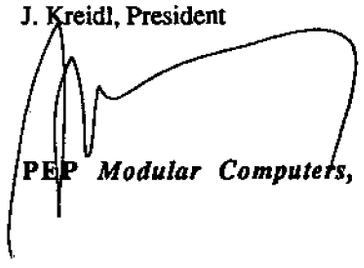
Other than the repair, replacement, or refund specified above, **PEP *Modular Computers*** will not accept any liability for any further claims which result directly or indirectly from any warranty claim. We specifically exclude any claim for damage to any system or process in which the product was employed, or any loss incurred as a result of the product not functioning at any given time. The extent of **PEP *Modular Computers*** liability to the customer shall not be greater than the original purchase price of the item for which any claim exists.

PEP *Modular Computers* makes no warranty or representation, either express or implied, with respect to its products, reliability, fitness, quality, marketability or ability to fulfill any particular application or purpose. As a result, the products are sold "as is," and the responsibility to ensure their suitability for any given task remains the purchaser's.

In no event will **PEP** be liable for direct, indirect, or consequential damages resulting from the use of our hardware or software products, or documentation; even if we were advised of the possibility of such claims prior to the purchase of, or during any period since the purchase of the product.

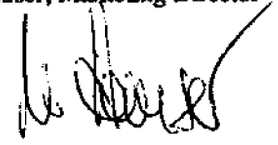
Please remember that no **PEP *Modular Computers*** employee, dealer, or agent are authorized to make any modification or addition to the above terms, either verbally or in any other form written or electronically transmitted, without the consent and counter-signature of the under-mentioned.

J. Kreidl, President



PEP *Modular Computers*,

N. Hauser, Marketing Director

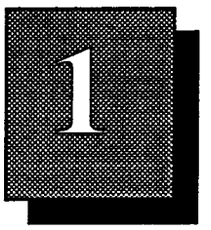


G. Rücker, Technical Director

Friday, July 29, 1988

TABLE OF CONTENTS

This page has been intentionally left blank



1. INTRODUCTION

1.1 Product Overview

The PB-DAC3 is an opto-isolated 4 channel D/A output piggyback for the VMOD-2 and IMOD. The output channels are independent with 12-bit resolution and are galvanically isolated from the system supply. Either unipolar or bipolar conversion can be selected. The range of the output distributor is programmable from 0V to 10V (±10V). An optional current output version from 0-20mA is also available.

1.2 Ordering Information

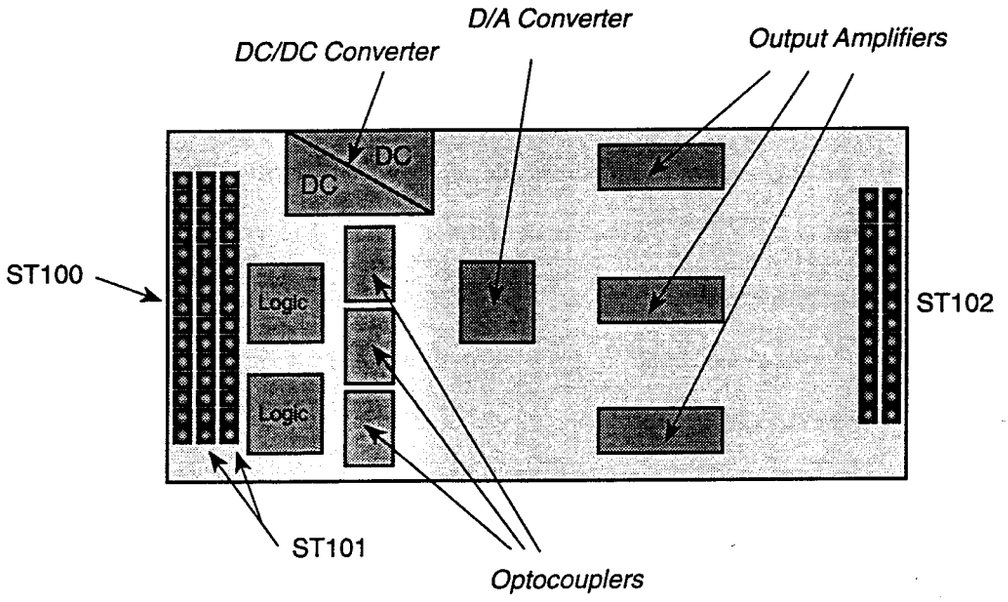
Name	Description	Order Number
PB-DAC3	4 channel voltage output, 12-bit resolution, unipolar or bipolar voltage output, EEPROM with calibration values	5230-35
PB-DAC3	4 channel current output, 12-bit resolution, 0-20mA current output, EEPROM with calibration values	5230-35/1

1.3 Specifications

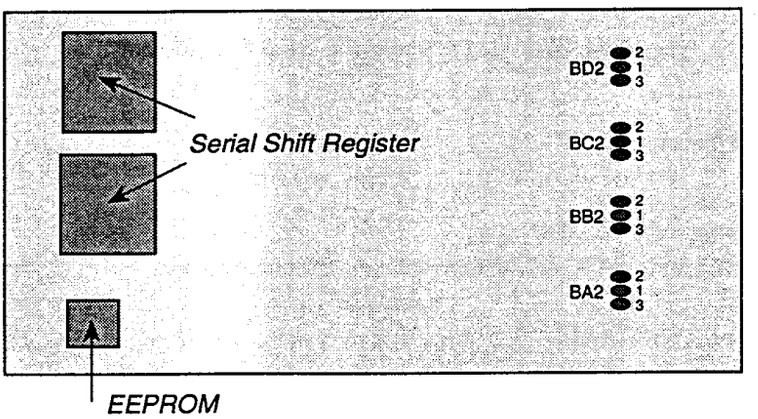
D/A converter chip	AD7568B
Number of channels	4
Resolution	12-bit
Serial data transfer time	4µs per data word
Rise time	0.4V per µs
Linearity error	± 0.75 LSB
Differential linearity	± 0.9 LSB
Voltage output ranges:- Unipolar	0-10V optional for each output via solder jumpers, full scale programmable by Reference
Bipolar	±10V optional for each output via solder jumpers, full scale programmable by Reference
Current output range	0-20mA per channel, programmable by Reference
Reference voltage	10V
Maximum current for voltage output	2mA per channel
External voltage for current output	5-24V DC
EEPROM	93C46, 128 byte programmable calibration data
Galvanic isolation from the system	500V DC from the system
ID Byte	$EA
Temperature range Standard Extended	 0...+70°C -40...+85°C
Power requirement	@5V : 290mA

1.4 Board Overview

Component Side



Solder Side



1.5 Features

- *Four independent D/A outputs*
- *12-bit resolution*
- *Galvanic isolation from the system (500V DC)*
- *Unipolar or bipolar voltage outputs, selectable by solder jumpers*
- *Adjustable voltage outputs, 0 to 10V, ±10V, full scale programmable by Reference*
- *Reference voltage for the output channels programmable in Vref/4096 steps from 0 to 10V*
- *Current outputs 0-20mA*
- *Pre-load register for synchronous update*
- *Hardware and software resets to zero, available for all channels, including bipolar*
- *Power-up condition of all channels is zero*
- *128-byte programmable EEPROM with on-board calibration data*

1.6 Glossary of Terms

Rise Time	The time taken for the signal pulse to come up.
Differential Linearity	Output signal accuracy, proportional to the derivative of the input signal, measured from the Least Significant Bit.
Linearity Error	The deviation of the actual curve characteristic from the linear approximation of a Hall generator, given in terms of the Least Significant Bit.
EEPROM	Electrically Erasable Programmable Read Only Memory.

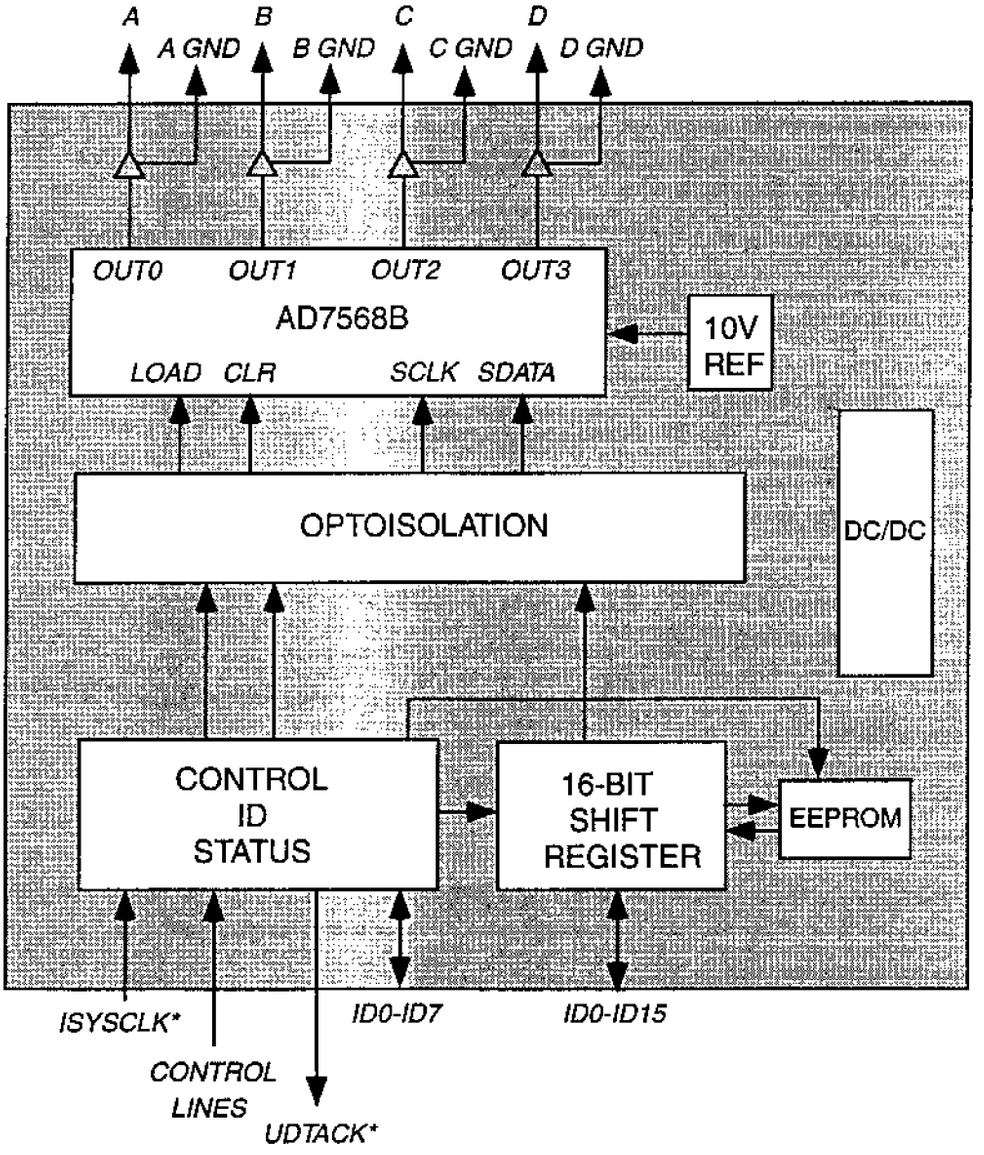
1.7 Related Publications

VMEbus Specifications Revision C1
Data Sheet for the AD7568B from Analog Devices
Data Sheet for the 93C46 EEPROM from Microchip/National Semiconductors

2. FUNCTIONAL DESCRIPTION

This chapter describes the functionality of the main blocks of the PB-DAC3 piggyback.

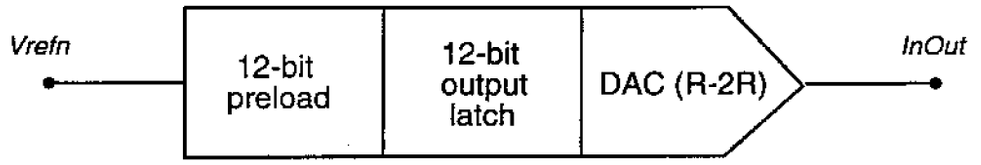
Figure 2.0.0.1 PB-DAC3 Block Diagram



2.1 Converter Principle

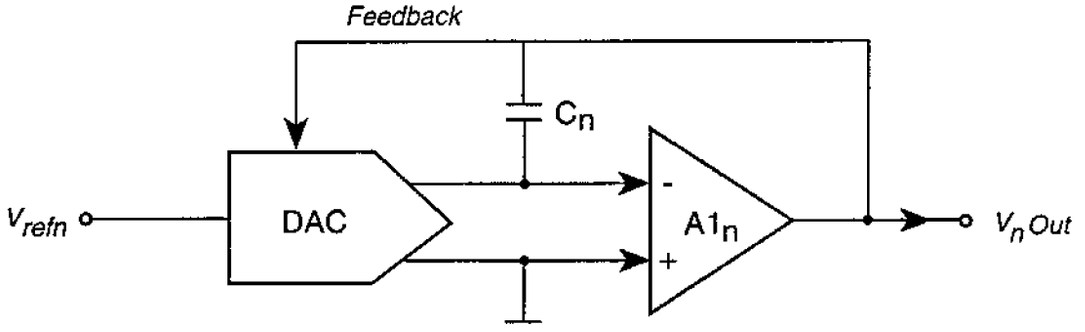
The D/A conversion is carried out using the AD7568 chip. It contains 8 independent 12-bit current output D/A converters. A standard R-2R ladder network is used for the conversion of the reference voltage. The switching time is therefore substantially reduced (0.5µs).

Figure 2.1.0.1: Conversion Principle of the AD7568 (1 of 8)



Using the following circuit the current is converted into a loaded unipolar voltage.

Figure 2.1.0.2: PB-DAC3 I/O Conversion Principle (1 of 8)

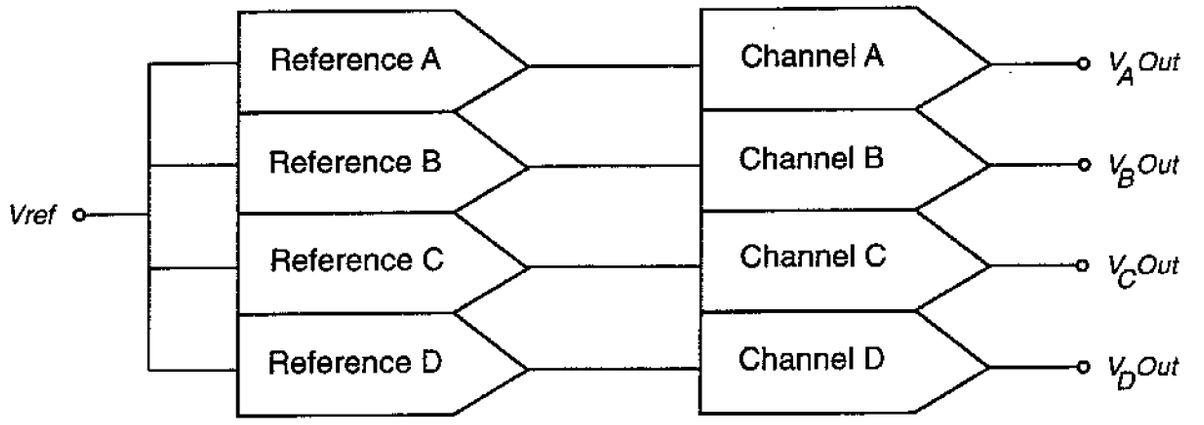


A1 n = Amplifier LM124/LT1014

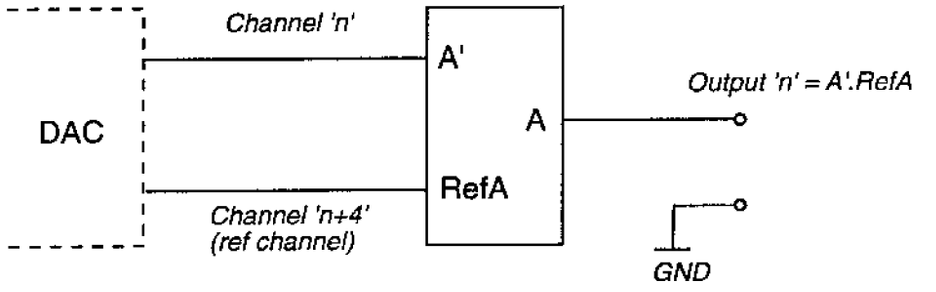
C n = Phase compensator

In order that the adjustment range of V_nOut can be modified and also to correct the rise error of the whole circuit, 4 from 8 of the analog channels are used for the reference voltage of the 4 remaining analog outputs.

Figure 2.1.0.3: Reference Control

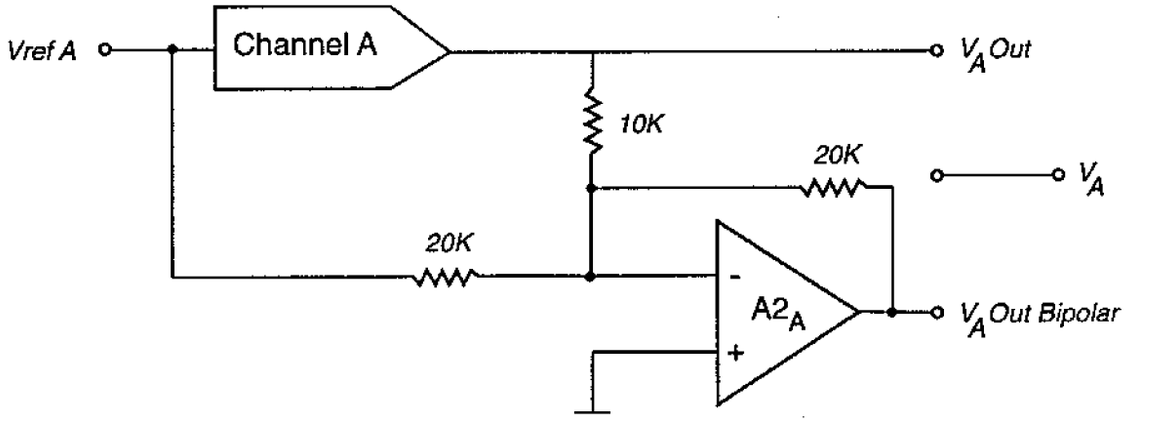


Using this circuit layout, a 12-bit programmable reference voltage for each of the 4 analog outputs is produced, using the following relationship:



Using a further amplifier the unipolar output voltage can also be transferred into a bipolar output voltage. The selection of the output voltage is made using solder jumpers.

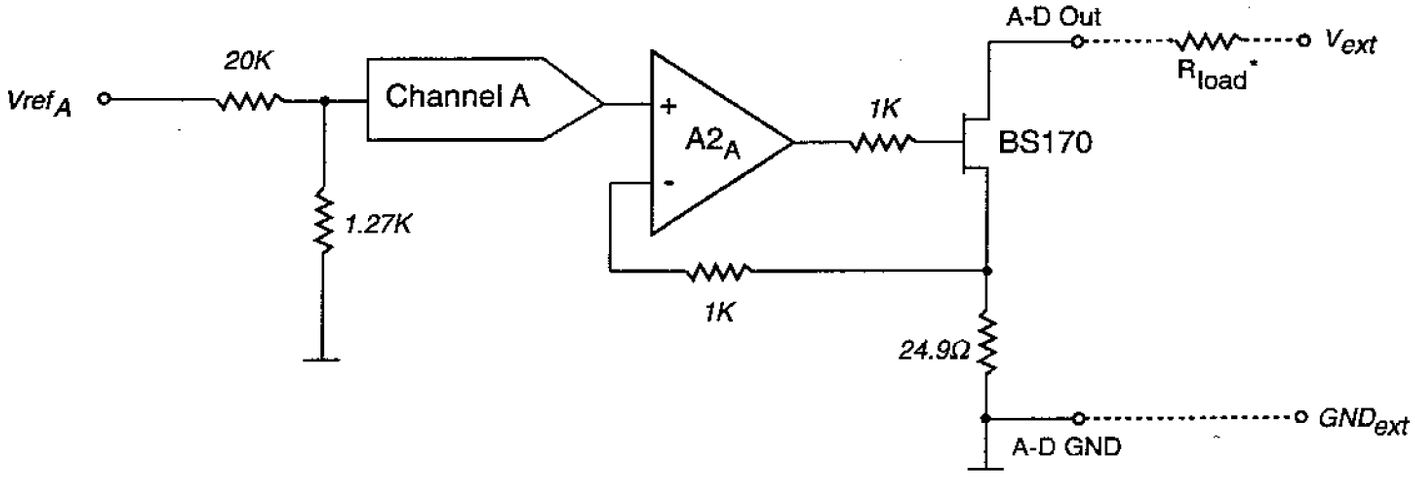
Figure 2.1.0.4: Unipolar and Bipolar Selection (1 of 4)



$A2_A$ = Amplifier LM124/LT1014

The current output is variable. Using a component change the amplifier A2 from Figure 2.1.0.4 can be used to control a MOSFET output stage. The externally supplied current flow is controlled using a low valued precision resistor (24.9Ω/0.1%).

Figure 2.1.0.5: Current Output A (1 of 4)



* With V_{ext} = 24V, R_{load} max = 1K.

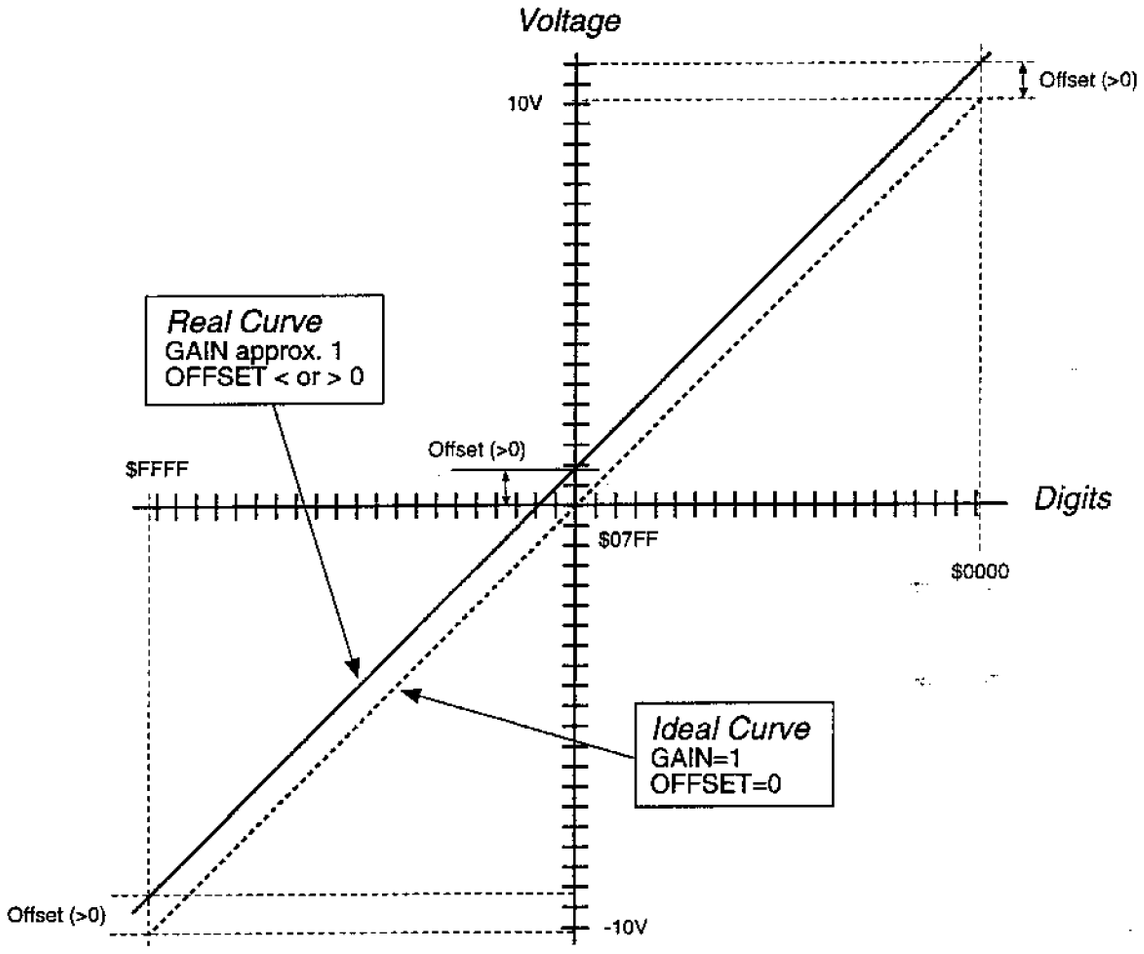
The data transfer across the optoisolation barrier is controlled by the serial interface of the AD7568. Using the 4MHz SCLK a data transfer rate of 4µs is achieved for the 16-bit data word.

The assigned time of the analog outputs is influenced by the rise time of the operational amplifier stage. With a rise time of 0.4V/µs, a minimum hub of 1.6V per access is possible.

2.2 Calibration Principle

The Figure below depicts the PB-DAC3 output characteristics, showing the relationship between the ideal and assigned functions. The assigned slope can be calibrated so that it can resemble the ideal slope = 1, ±0.5LSB. The relevant calibration of the reference channel is stored in the EEPROM for both unipolar and bipolar modes.

Figure 2.2.0.1: PB-DAC3 Characteristics (Bipolar)



The resultant offset cannot be configured in the hardware. The size of the offset is, however, stored in EEPROM in order that software compensation can be achieved.

2.3 Logic Interface

The logic interface consists of two shift registers and the PAL logic. The main task of the logic interface is to oversee the 16-bit parallel serial conversion and data control to the converter and to and from the EEPROM.

As well as this the clock signals are generated, together with the status register. The status register also enables the EOS (End Of Shift) signal and also the EOP (End Of Programming) signal to be accessed when programming.

Another important task performed by the logic interface is the generation of an ID byte. This built-in test feature allows an interrogation of the VMOD-2/IMOD. It supplies an ID for each of the fitted piggybacks. If this is integrated into the application software, it can be used to check that any given tasks are valid for the fitted piggyback. Offsets $7F (for location A) and $FF (for location B), allow a software check of which piggybacks are fitted.

The PB-DAC3 has a "$EA" Byte.

Other ID Bytes are:

$EE	*PB-BIT*	*BITBUS communications controller*
$EF	*PB-DIO4*	*Digital I/O piggyback (high voltage)*
$F0	*PB-CNT*	*Counter piggyback*
$F1	*PB-DAC/2*	*D to A converter piggyback*
$EA	*PB-DAC3*	*D to A output piggyback*
$F2	*PB-DIO*	*Digital I/O piggyback*
$F3	*PB-DIN/3*	*Digital input piggyback*
$F4	*PB-ADC/2*	*A to D converter piggyback*
$EB	*PB-ADC3*	*A to D input piggyback*
$F5	*PB-CIO/2*	*Counter/I/O piggyback*
$F7	*PB-SIO4*	*Quad serial piggyback RS232*
$E7	*PB-SIO4A*	*Quad serial piggyback RS422/RS485*
$F8	*PB-DOUT*	*12 channel high voltage digital output*
$ED	*PB-DOUT2*	*16 channel high voltage digital output*
$F9	*PB-DIN2*	*Digital input piggyback*
$FB	*PB-DIO-2*	*Digital I/O piggyback*
$FC	*PB-REL*	*Relay piggyback*
$FD	*PB-DIO-3*	*Digital I/O piggyback*
$FE	*PB-STP*	*Stepper motor controller piggyback*

3. CONFIGURATION

The PB-DAC3 has 4 solder jumpers on the solder side of the board, one for each channel. These allow selection of the voltage outputs in unipolar or bipolar mode.

Figure 3.0.0.1 PB-DAC3 Jumper Layout (Solder Side)

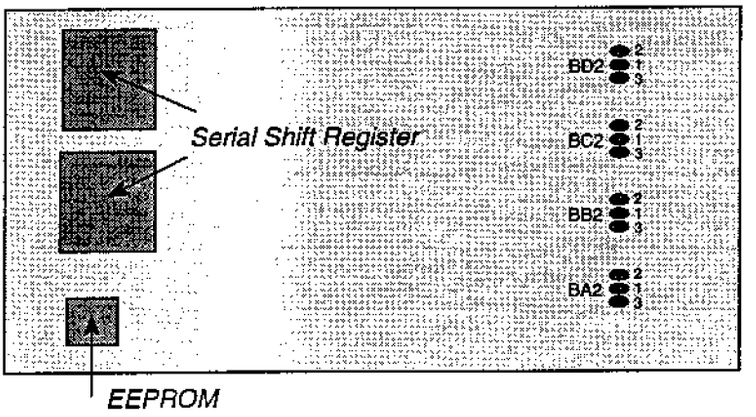


Table 3.0.0.2 PB-DAC3 Default Jumper Settings

Voltage Version:

Channel	Jumper	Default Setting	Function
A	BA2	1-3	Bipolar voltage output
B	BB2	1-3	Bipolar voltage output
C	BC2	1-3	Bipolar voltage output
D	BD2	1-3	Bipolar voltage output

Current Version:

Channel	Jumper	Default Setting	Function
A	BA2	open	Current output
B	BB2	open	Current output
C	BC2	open	Current output
D	BD2	open	Current output

Default settings are shown in *italic* in the following section.

3.1 Jumpers BA2-BD2: Output Selection

Voltage Version:

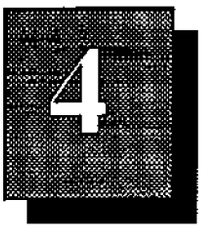
These jumpers can be configured to set a bipolar or unipolar voltage output for each channel.

Channel	Jumper	Setting	Function
A	BA2	1-2	Unipolar voltage output
		1-3	*Bipolar voltage output*
B	BB2	1-2	Unipolar voltage output
		1-3	*Bipolar voltage output*
C	BC2	1-2	Unipolar voltage output
		1-3	*Bipolar voltage output*
D	BD2	1-2	Unipolar voltage output
		1-3	*Bipolar voltage output*

Current Version:

Only one jumper setting is available for the current output version of the PB-DAC3.

Channel	Jumper	Setting	Function
A	BA2	*open*	*Current output*
B	BB2	*open*	*Current output*
C	BC2	*open*	*Current output*
D	BD2	*open*	*Current output*



4. PROGRAMMING

The base address of the PB-DAC3 in the upper piggyback position (location A) is the same as that set for the VMOD-2/IMOD used (default: $87FE2400/$F70000), whereas the base address in the lower piggyback position (location B) is +$80 of that set for the VMOD-2/IMOD (default: $87FE2480/$F70080).

4.1 PB-DAC3 Address Map

Address	Byte/Word	Read/Write	Function
BASE+$00	Word	Write	DAC output register A-D DAC reference register A-D
BASE+$02	Word	Write	Preload register A-D Preload register for reference A-D
BASE+$03	Byte	Read Write	Status register (Read) DAC clear register (Write)
BASE+$20	Word	Read/Write	EEPROM communications register
BASE+$22	Word	Write	EEPROM programming register
BASE+$7F	Byte	Read	ID register

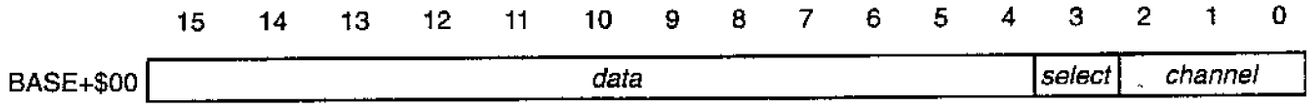
4.1.1 DAC Output Register ($00)

The four analog outputs, together with their respective reference channels, are accesses through this register. Using a 'word' write access to this register, a 12-bit word is sent to the respective pre-load register of the chosen channel, and thereafter, all DAC output registers are synchronously updated.

Note the function of the *Select* bit. If it is set to 1, the data and channel information is not used and only a synchronous update of all channels is performed.

WARNING!

It is necessary to initialize all pre-load registers prior to accessing this DAC output register for the first time. The default setting of the preload register is $FFF, which is set to full power. For setting use the preload register address.



Register Description

Name	Value	Description
data *bits 15-4*	$0 - $FFF	12-bit conversion data
select *bit 3*		Selects whether the actual data is adopted in the corresponding pre-load register. Once this bit is set, a synchronous update of all DAC outputs is triggered.
	0	Channel value adopted
	1	Channel value not adopted
channel *bits 2-0*	0	Output A
	1	Output B
	2	Output C
	3	Output D
	4	Reference Channel A
	5	Reference Channel B
	6	Reference Channel C
	7	Reference Channel D

Example 1

```
            .
            .
            .
wait:   btst.b   #2,BASE+$03        DAC ready?
        bne.s    wait
        move.w   #$8002,BASE+$00    set $800 to channel C
            .
            .
            .
```

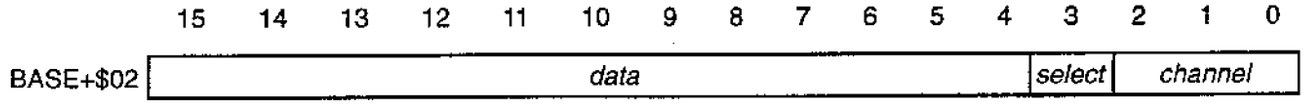
Example 2

```
                .
                .
                .
wait:       btst.b    #2,BASE+$03         DAC ready?
            bne.s     wait
            move.w    #$0008,BASE+$00     synchronous update of all channels
            .
            .
            .
```

4.1.2 DAC Preload Register ($02)

The four analog preload registers and their respective reference channels are accessed through this register. Using a 'word' write access to this register a 12-bit word is sent to the respective pre-load register of the chosen channel. To allow this, the *Select* bit has to be set to 0. Please note that the default setting of the pre-load register is $FFF after start-up, which is the maximum condition. Initialization is necessary as the next step in the process. Access to the DAC through this register has no influence on the actual output status.

The same register configuration is used as for the DAC Output Register, described above.



4.1.3 Status Register ($03)

The status register is a double function register. A read access to the status register monitors the physical status of shift activities and programming duration. A write access triggers the software reset on the DAC outputs.

	7	6	5	4	3	2	1	0
BASE+$03	*not used*			*P-Busy*	*not used*	*S-Busy*	*not used*	

Register Description

Name	Value	Description
P-Busy *bit 4*	 0 1	Indicates when the programming process is complete. Relevant only after programming access has been made to the EEPROM. EEPROM programming busy EEPROM programming ready (default)
S-Busy *bit 2*	 0 1	State of the serial buffer. Datashift ready (default) Datashift busy
not used *bits 7-5, 3, 1* *bit 0*	 1 0	

Example 1

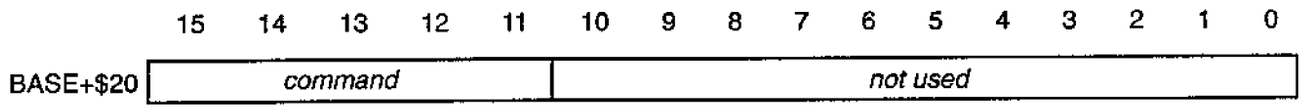
```
                .
                .
                .
wait:       btst.b    #2,BASE+$03        DAC ready?
            bne.s     wait
            move.b    #0,BASE+$03        set all DAC output registers to zero
                .
                .
                .
```

4.1.4 EEPROM Communications Register ($20)

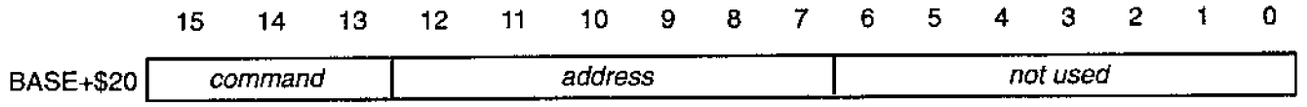
This register handles the data exchange with the EEPROM. This requires a 'WORD' write and read access. Before reading a particular address in the EEPROM, a *read* command has to be written to the required address on this register.

Write (*Two Cases*):

1) Write for EN/DIS protection command



2) Write for readout command

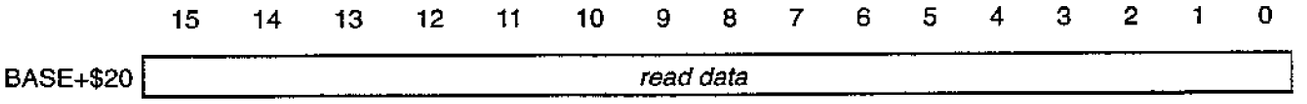


Register Description

Name	Value	Description
command *bits 15-11* *or* *bits 15-13*	%10011 %10000 %110	Disable program protection Enable program protection Read stored data
address *bits 12-7*	 0-63	These bits can only be interpreted with a read command Address 0-63 (dec)
not used *bits 10-0* *or* *bits 6-0*	Free	

Read:

After a *read* command, it can be determined using the status bit *S-Busy*, when the expected data can be read (WORD).



Example 1

```
                .
                .
                .
wait:       btst.b    #2,BASE+$03         S-Busy?
            bne.s     wait
            move.w    #$C000,BASE+$20     read command for EPROM address offset 0
wait2:      btst.b    #2,BASE+$03         S-Busy?
            bne.s     wait2
            move.w    BASE+$20,d0         get EEPROM read result
                .
                .
                .
```

Address offset 1 -> read command = $C080
Address offset 63 -> read command = $DF80

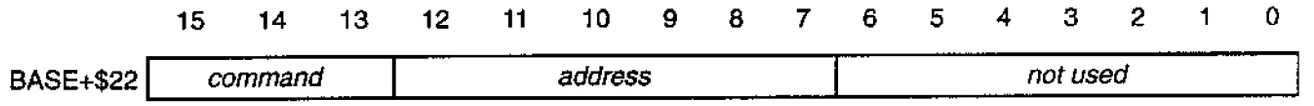
Example 2

```
                .
                .
                .
wait:       btst.b    #2,BASE+$03         S-Busy?
            bne.s     wait
            move.w    #$9800,BASE+$20     set EEPROM in programming mode
                .
                .

wait2:      btst.b    #2,BASE+$03         S-Busy?
            bne.s     wait2
            move.w    #$8000,BASE+$20     set EEPROM in protected mode
                .
                .
                .
```

4.1.5 EEPROM Programming Register ($22)

This register is used for data exchange in the programming mode. This requires two consecutive word write accesses to the register. The first access sends the *write* command and the address that is to be overwritten.

First value write:

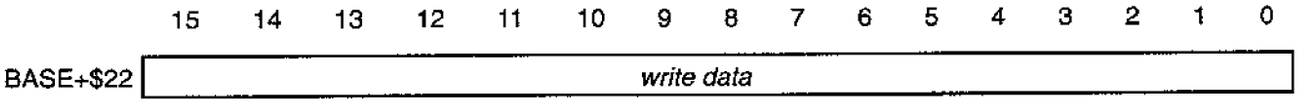


Register Description

Name	Value	Description
command *bits 15-13*	%101	Write the following data
address *bits 12-7*	0-63	Address 0 to 63 (dec)
not used *bits 6-0*	Free	

Second value write:

After a *write* command, it must be determined using the status bit *S-Busy*, when the next write access can take place.



After the data has been transferred, the start of the program can be determined with the status register's *P-Busy* flag. The PB-DAC3/EEPROM can only be accessed again after the *P-Busy* flag has reverted to 1 **or** a programming time of at least 5ms has passed.

Example

```
                .
                .
wait:       btst.b    #2,BASE+$03        S-Busy?
            bne.s     wait
            move.w    #$B000,BASE+$22    write command to first user location
wait2:      btst.b    #2,BASE+$03        S-Busy?
            bne.s     wait2
            move.w    #$1234,BASE+$22    program $1234
wait3:      btst.b    #4,BASE+$03        P-Busy?
            bne.s     wait3
                .
                .

or          wait at least 5ms
```


Note

Due to the fact that the EEPROM contains calibration data programmed by PEP, it is not recommended that data be written to the EEPROM, even in free locations.

4.1.6 ID Register ($7F)

This register contains the ID byte, allowing automatic recognition of the piggyback fitted to the VMOD-2 or IMOD. This feature is particularly useful in a system with several piggybacks fitted.

PB-DAC3 ID byte = $EA

For more information on the ID byte, please see the ID Byte section in the Functional Description chapter.

4.2 Calculation of the Analog Output

Voltage Output	Voltage Output	Current Output
Unipolar Range 0-10V/Reference 10V	Bipolar Range ±10V/Reference 10V	Unipolar ONLY! Range 0-20mA/Reference 0.5V
$000=0V $800=5V - 0.5LSB $FFF=10V - 1LSB	$000=10V - 1LSB $800=0V $FFF=-10V + 1LSB	$000=0mA $333=4mA $FFF=20mA - 1LSB

The above values correspond to the values from a channel with 0 LSB offset. If a channel is assigned a single offset, the compensation is calculated using the channel-specific data of the EEPROM.

Example:

Bipolar Offset = +2LSB

$002 = 10V - 1LSB *Compensated*
$801 = 0V *Compensated*
$FFF = -10V + 2LSB *Non compensated*

4.3 EEPROM Data Structure

The EEPROM 93C46 provides a 64-word address, word access, permanently programmable memory. The address assignments to the individual analog outputs are shown below.

Address*	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
$0	*reserved*								*channel*				*mode*				
$1	*unipolar offset (20mA offset)*								*unipolar reference (20mA reference)*								channel A
$2	*bipolar offset (reserved)*								*bipolar reference (reserved)*								
$3	*reserved*								*reserved*								
.																	
.																	
.																	
.																	
$C	*reserved*								*channel*				*mode*				
$D	*unipolar offset (20mA offset)*								*unipolar reference (20mA reference)*								channel D
$E	*bipolar offset (reserved)*								*bipolar reference (reserved)*								
$F	*reserved*								*reserved*								
$10	*user*																
.																	
.																	
$3F																	

* Read command for accessing an EEPROM address:

EEPROM Read Command = $C000 + (Address <<7);

Note

The parameters in brackets above refer to those specific to the 20mA version of the PB-DAC3.

EEPROM Description

Name	Value	Description
channel *bits 7-4*	0 1 2 3	Channel A Channel B Channel C Channel D
mode *bits 3-0*	0 1 2	Unipolar Bipolar Current
unipolar offset *bits 15-8*		A signed byte value for the LSB offset at the unipolar, or current, output. **Example** $FF = -1LSB
unipolar reference *bits 7-0*		A byte value for the difference between the full power value and the reference voltage for 0-10V, or 0-20mA. **Example** Contents = $31 Reference value = $FFF - $31 = $FCE
bipolar offset *bits 15-8*		A signed byte value for the LSB offset at the bipolar voltage output, or a reserved byte for the current output version. **Example** $02 = +2LSB
bipolar reference *bits 7-0*		A byte value for the difference between the full power value and the reference voltage for ±10V, or a reserved byte for the current output version. **Example** Contents = $2F Reference Value = $FFF - $2F = $FD0
user *bits 15-0*		96 bytes free for user-specific data.

Note

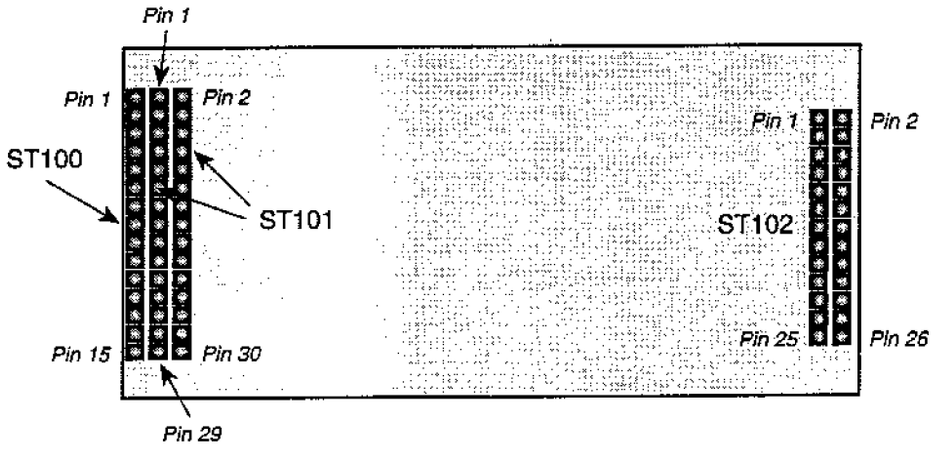
Care must be taken when editing the EEPROM. PEP accepts no responsibility for the erasure of calibration data.

5. PINOUTS

The PB-DAC3 has three sets of connectors. ST100 is the rearmost row of 15 pins directly next to ST101 with 30 pins. At the front end of the piggyback there is ST102 with 26 pins.

5.1 Main Board

Figure 5.1.0.1: Board Connector Overview



5.1.1 ST100 Connector

The ST100 connector is fitted into the socket row BU0A or BU0B on the IMOD and VMOD-2, depending if there is one or two piggybacks to be fitted.

Pin #	Signal
1	GND
2	N/C
3	N/C
4	N/C
5	N/C
6	IDS1*
7	ID15
8	ID14
9	ID13
10	ID12
11	ID11
12	ID10
13	ID9
14	ID8
15	GND

* *Active Signal Low*

5.1.2 ST101 Connector

The ST101 connector is fitted into the socket row BU1A or BU1B on the IMOD and VMOD-2 (depending if there is one or two piggybacks to be fitted) to the VMEbus interface logic.

Signal	Pin #	Pin#	Signal
GND	1	2	Vcc
N/C	3	4	N/C
R/W*	5	6	CLK
RESET*	7	8	UDTACK*
N/C	9	10	CS*
N/C	11	12	N/C
ID7	13	14	IDS0*
ID6	15	16	N/C
ID5	17	18	IA6
ID4	19	20	IA5
ID3	21	22	N/C
ID4	23	24	N/C
ID1	25	26	N/C
ID0	27	28	IA1
GND	29	30	VCC

* *Active Signal Low*

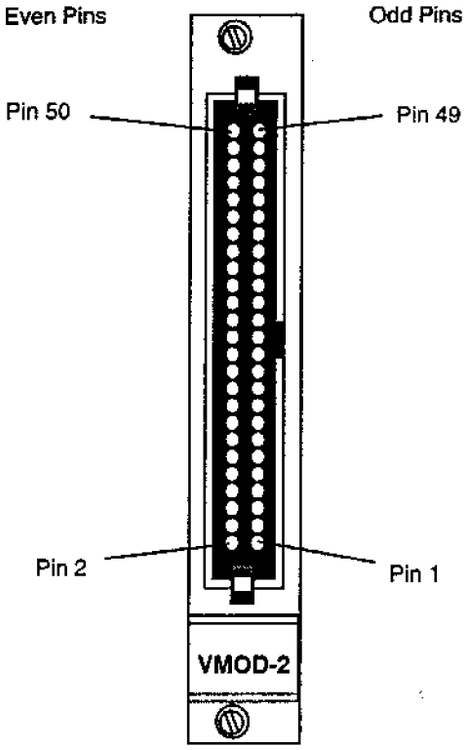
5.1.3 ST102 Connector

The ST102 connector is fitted into the socket row BU2A or BU2B on the IMOD and VMOD-2 (depending if there is one or two piggybacks to be fitted), directly to half of the 50-way VMOD-2 front panel connector, and ultimately to the user's external interfaces.

Signal	Pin#	Pin#	Signal
N/C	1	2	N/C
N/C	3	4	N/C
N/C	5	6	N/C
N/C	7	8	N/C
N/C	9	10	AGND
AOUT	11	12	AGND
N/C	13	14	BGND
BOUT	15	16	BGND
N/C	17	18	CGND
COUT	19	20	CGND
N/C	21	22	DGND
DOUT	23	24	DGND
N/C	25	26	N/C

5.2 VMOD-2/IMOD Front Panel

Figure 5.2.0.1: VMOD-2/IMOD Front Panel



Piggyback A

Pin #	Signal	Pin #	Signal
50	N/C	49	N/C
48	N/C	47	N/C
46	N/C	45	N/C
44	N/C	43	N/C
42	N/C	41	AGND
40	AOUT	39	AGND
38	N/C	37	BGND
36	BOUT	35	BGND
34	N/C	33	CGND
32	COUT	31	CGND
30	N/C	29	DGND
28	DOUT	27	DGND
26	Ext. Reset GND	25	Ext. Reset VCC

Piggyback B

Pin #	Signal	Pin #	Signal
24	N/C	23	N/C
22	N/C	21	N/C
20	N/C	19	N/C
18	N/C	17	N/C
16	N/C	15	AGND
14	AOUT	13	AGND
12	N/C	11	BGND
10	BOUT	9	BGND
8	N/C	7	CGND
6	COUT	5	CGND
4	N/C	3	DGND
2	DOUT	1	DGND

The relevant half of the VMOD-2 and IMOD front panel 50-way connector (pins 1..24 for lower position and pins 27..50 for upper) assumes the relationship of PB-DAC3 signals and its ST102.

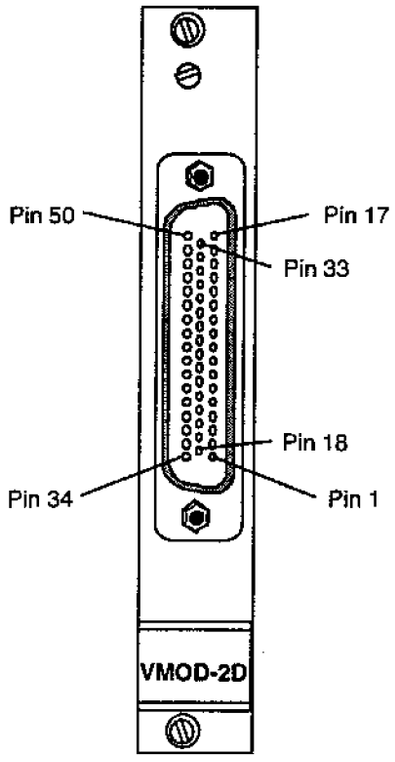
An optional 50-way header behind the front panel connector has an identical pin-out to the front panel connector. It is provided for applications where the flat band cable is to be routed internally, or where an alternative front panel is to be fitted and used. In some cases, cables can be routed through the systems interior i.e. to the back panel (from this optional connector) and some from the external connector on the front panel. In doing so take care not to exceed the fan out ability of the piggyback's driver circuits.

Note

With systems that have more than one of this type of connector , or which use several VMOD-2 or IMODs with various piggybacks, it is advisable to put a drop of paint on the back of the mating connector and on the front panel of the VMOD-2 or IMOD, for correct connection. The connector splits virtually in half (pins 1-24 and 27-50) for connection to the rear piggyback location.

5.3 VMOD-2D Front Panel

Figure 5.3.0.1: VMOD-2D Front Panel



Piggyback A

Pin #	Signal	Pin #	Signal	Pin #	Signal
50	N/C	33	N/C	17	N/C
49	N/C	32	N/C	16	N/C
48	N/C	31	N/C	15	N/C
47	AGND	30	AGND	14	AOUT
46	N/C	29	BOUT	13	BGND
45	BGND	28	CGND	12	N/C
44	COUT	27	N/C	11	CGND
43	DGND	26	DGND	10	DOUT
42	Ext. Reset GND			9	Ext. Reset VCC

Piggyback B

Pin #	Signal	Pin #	Signal	Pin #	Signal
		25	N/C		
41	N/C	24	N/C	8	N/C
40	N/C	23	N/C	7	N/C
39	N/C	22	AGND	6	N/C
38	AOUT	21	N/C	5	AGND
37	BGND	20	BGND	4	BOUT
36	N C	19	COUT	3	CGND
35	CGND	18	DGND	2	N/C
34	DOUT			1	DGND

5.4 VMOD-2 / VMOD-2D Pinout Relationship

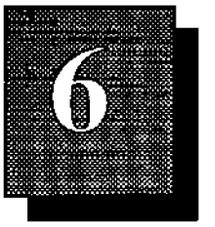
Piggyback A

Signal	VMOD-2 Pin #	VMOD-2D Pin #	Signal	VMOD-2 Pin #	VMOD-2D Pin #
N/C	50	50	N/C	49	17
N/C	48	33	N/C	47	49
N/C	46	16	N/C	45	32
N/C	44	48	N/C	43	15
N/C	42	31	AGND	41	47
AOUT	40	14	AGND	39	30
N/C	38	46	BGND	37	13
BOUT	36	29	BGND	35	45
N/C	34	12	CGND	33	28
COUT	32	44	CGND	31	11
N/C	30	27	DGND	29	43
DOUT	28	10	DGND	27	26

Signal	VMOD-2 Pin #	VMOD-2D Pin #	Signal	VMOD-2 Pin #	VMOD-2D Pin #
Ext. Reset GND	26	42	Ext Reset VCC	25	9

Piggyback B

Signal	VMOD-2 Pin #	VMOD-2D Pin #	Signal	VMOD-2 Pin #	VMOD-2D Pin #
N/C	24	25	N/C	23	41
N/C	22	8	N/C	21	24
N/C	20	40	N/C	19	7
N/C	18	23	N/C	17	39
N/C	16	6	AGND	15	22
AOUT	14	38	AGND	13	5
N/C	12	21	BGND	11	37
BOUT	10	4	BGND	9	20
N/C	8	36	CGND	7	3
COUT	6	19	CGND	5	35
N/C	4	2	DGND	3	18
DOUT	2	34	DGND	1	1



6. INSTALLATION

6.1 VMEbus Connection

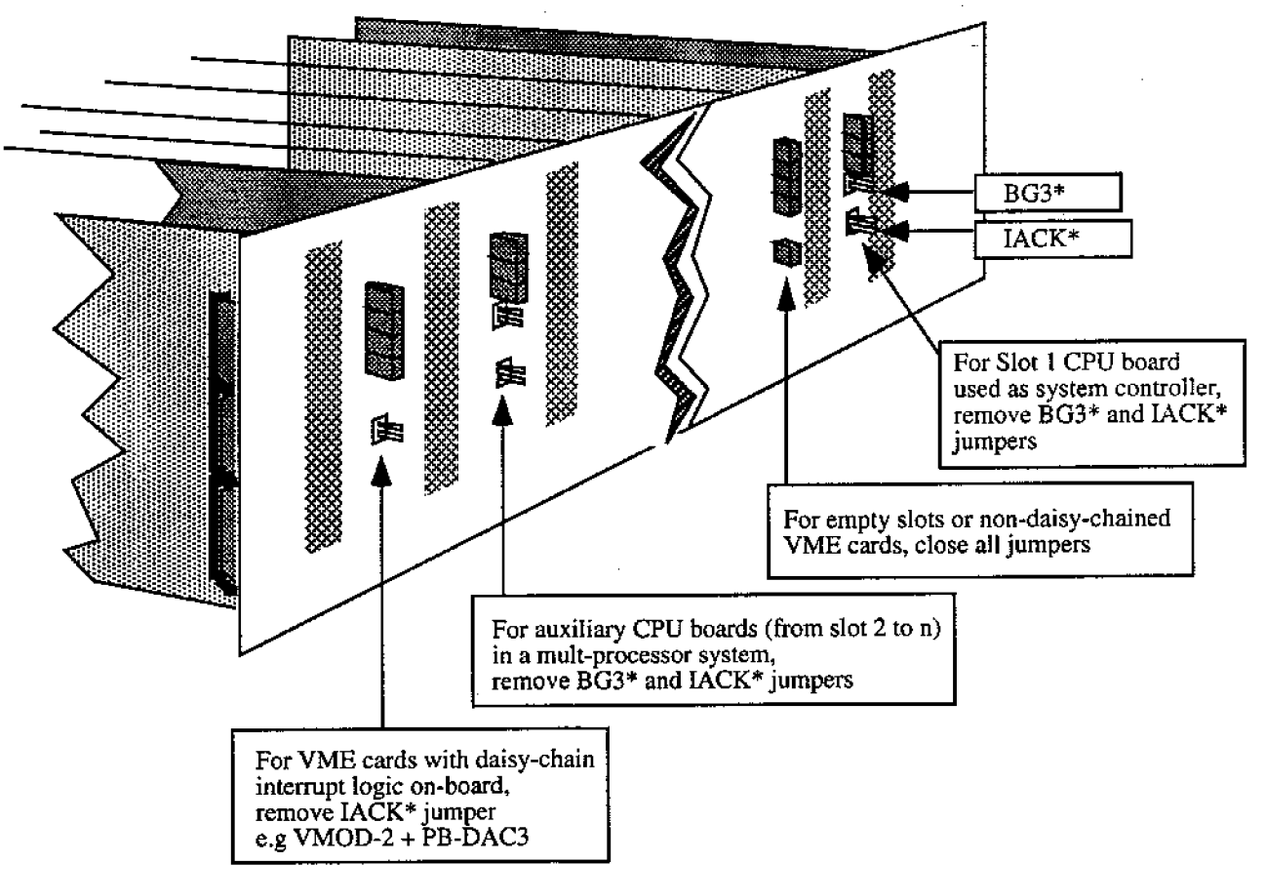
Caution!

Before installing or removing any VMEbus boards, always turn off the power to the bus and any external peripherals.

Inserting or removing modules, while the power is on, could result in damage to the VME module or peripherals interface.

Please refer to VMOD-2/IMOD user's manual for details on installing or removing.

Figure 6.1.0.1: The VMEbus Backplane



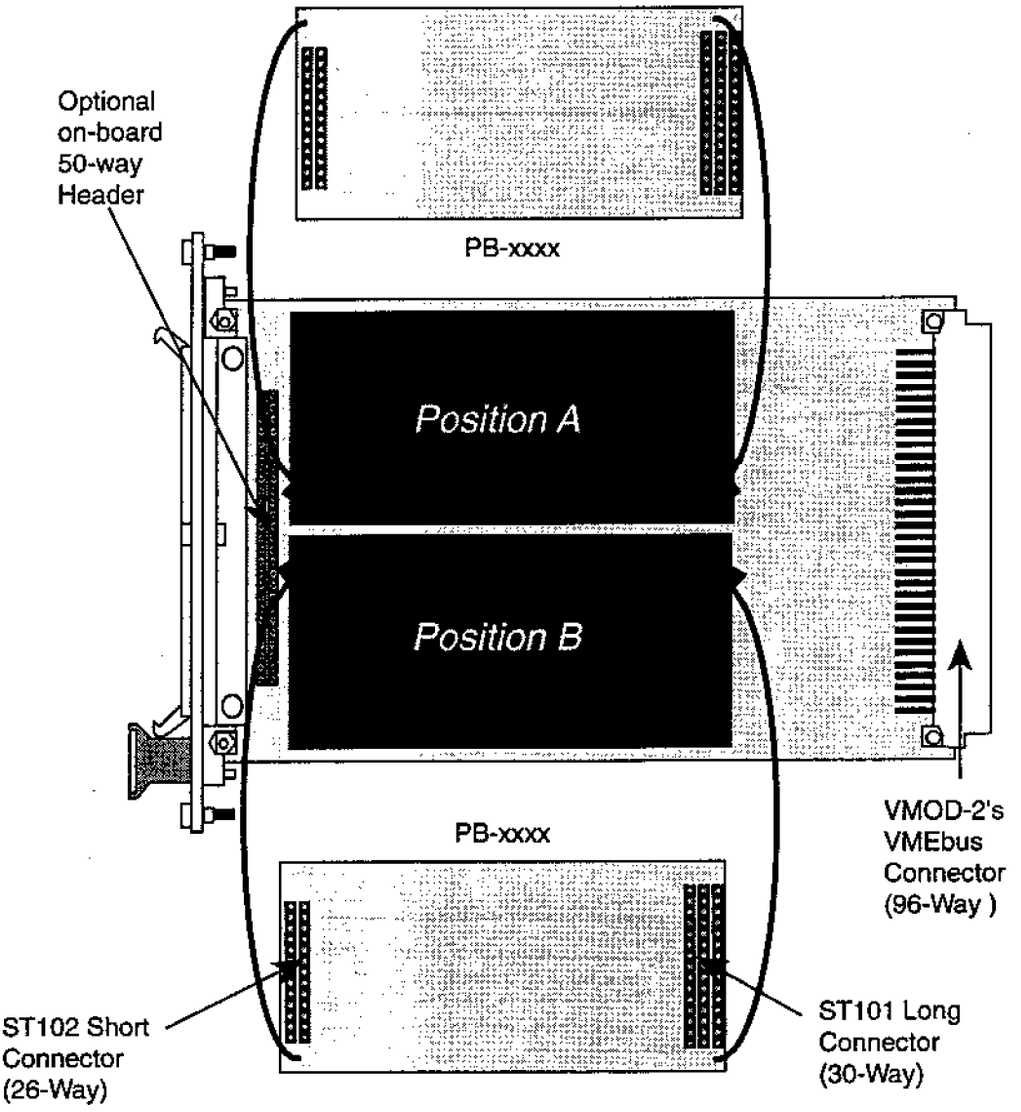
6.2 Installing the PB-DAC3

The PB-DAC3 may be plugged into any free piggyback position (A or B) on the VMOD-2 or IMOD. Please ensure the correct location before fitting.

Note

1. One connector on the PB-DAC3 has fewer pins than the other.
2. ST101 has two-rows which are to fit the front two-rows of the VMOD-2 amd IMOD three-row interface socket. Take care to ensure that the piggyback is in its correct position.

Figure 6.2.0.1: PB-DAC3 Installation Overview



Mechanical fastening and support is provided by the two interface connectors. In addition, the piggyback can be attached to the motherboard by screws and stand-off pillars at the front end of the piggyback, and at the corresponding location on the VMOD-2 and IMOD at the two holes provided for this purpose.

6.3 General Notes for Using the System

Having designed a system, it is necessary to keep it in good working order. The three biggest risks to the system occur when:

- *Connecting peripherals, disk-drives, printers, terminals and external power sources;*
- *Adding or changing modules, address settings and locations, etc;*
- *Becoming complacent and not referring to the manuals when altering or adding modules.*

These risks can be reduced by:

- *Checking the electrical compatibility of all devices to be connected;*
- *Ensuring that they are powered from the same mains supply branch (phase) and grounded to the same reference point;*
- *Shutting down all power before making or breaking any connections to modules or attachments to the system, including power to the peripherals;*
- *Observing sensible static protection procedures before handling any modules, piggybacks or memory IC's;*
- *Keeping all manuals available by the system and refer to them when required.*

Some tips are:-

PEPCards are not over sensitive to static, but it is generally advisable to observe normal antistatic procedures.

When configuring the module, it should not be taken out of the original packing unless necessary. The clear packs can be opened and the jumpers set, piggybacks added, etc. without removing the card. This prevents inadvertent shorting of any on-board devices.

When inserting modules into a system, the power should be turned off and the mains lead not removed! The ground wire prevents the rack floating with dangerous static voltages, which could destroy circuits on the module being inserted.

The front panel of the module and the shell of the connector should be touched to any part of the rack before fitting. This discharges any static from the user.

Modules should not be pulled straight out of a rack and the back of the front panel checked to see if there are cables to unplug (such as the VSBC-1's 40-pin parallel on-board headers). Any leads should be disconnected from a module before unscrewing the front panel and removing from the rack. It should be ensured that, when fitted, these cables have enough play to allow the modules to be removed far enough to detach these cables. Modules should be put into the rack before connecting any front-panel connectors.

The "pulled" jumpers should be parked on to one of the pins they would normally bridge, so they can be quickly replaced.

It should be remembered to check the mains input voltage selector switch before installing or using any PSU!

A record should be kept of settings and a copy forwarded with any board returned to PEP for failure analysis.

This page has been intentionally left blank

APPENDIX A. COMPLEX EXAMPLES

A.1 Write Channels

A.1.1 C-Program Listing

```
#include <errno.h>
#include <modes.h>
#include <sgstat.h>
#include <stdio.h>

/*****************************
*          Address Offset Table           *
 *****************************/

#define DACOUT        ( 0x00 )              /* word write */
#define DACSET        ( 0x02 )              /* word write */
#define STATUS          0x03                /* byte read/write */
#define EEPROM        ( 0x20 )              /* word read/write */
#define EEP_PROG      ( 0x22 )              /* word write */
#define ID              0x7F                /* byte read */

/******************
*       definitions        *
*****************/

#define READY         0xfa        /* STATUS register READY check  */
#define PBDAC3_ID     0xEA        /* PB-DAC3 identification byte */
#define REF           4           /* Reference Channel Offset */

#define A_CHA         0
#define B_CHA         1
#define C_CHA         2
#define D_CHA         3

#define A_REF         4
#define B_REF         5
#define C_REF         6
#define D_REF         7

char name[] = {'A','B','C','D'};
char *str[] = {"Unipolar","Bipolar","Current"};

static unsigned char *babase=0x87fe2400;    /* default base address of VMOD-2 */
static unsigned short *wabase=0x87fe2400;
```

```
/************************
 *   PB-DAC3 Functions     *
 ************************/

DAC_ready()
{
        for(count=0;count<100;count++)
        {
             status=*(babase+STATUS);     /* check if shift register is ready */
             if ((status&0xff)==READY)
             break;
        }
}


DAC_clear()
{
        *(babase+STATUS)=8;               /* clear all outputs to zero */
}


DAC_update()
}
        *(wabase+DACOUT)=8;         /* latch all preload registers to output */
}


DAC_out(channel,value)
{
char status;
short code;

        code=(value<<4);
        code=(code&0xfff0)+channel;       /* build channel code */
        DAC_check();
        *(wabase+DACOUT)=code;            /* write out channel code */
}


DAC_set(value,channel)

short value,channel;
{
char status;
short code;

        code=(value<<4)+channel;          /* build channel code */
        DAC_check();
        *(wabase+DACSET)=code;            /* write out channel code */
}
```

```
/***************
* Main Program  *
***************/
main()
{
          int repeat,cc,fullscale_error[4];
          short eeprom_command,eeprom_read,reference[4];
          short chan,value;
          char mode[4];

          /* PB-DAC3 ID Check */

          if (*(babase+ID)!=PBDAC3_ID)
          {
               printf("\nError: Wrong PB-DAC3 ID found! Correct one is $EA!\n");
               printf(" Read: $%x\n\n",*(babase+ID));
               exit(0);
          }

          /* Initialize PB-DAC3 from EEPROM data set */

          eeprom_command=0xc000;
          for (cc=0;cc<4;cc++)
          {
               DAC_ready();
               *(wabase+EEPROM)=eeprom_command;

               DAC_ready();
               eeprom_read=*(wabase+EEPROM);

               mode[cc]=(char)(eeprom_read&0x0f);
               printf("PB-DAC3 Channel %c, Mode %d=%s\n",namee[cc],mode[cc],
                                        str[mode[cc]]);

               eeprom_command=eeprom_command+0x0080;
               if (mode[cc]==1)
               {
                    eeprom_command=eeprom_command+0x0080;
                    DAC_preload(cc,0x800);
               }
               else DAC_preload(cc,0);

               DAC_ready();
               *(wabase+EEPROM)=eeprom_command;

               DAC_ready();
               eeprom_read=*(wabase+EEPROM);

               reference[cc]=0xfff-(eeprom_read&0xff);
               eeprom_read=*(wabase+EEPROM);

               fullscale_error[cc]=(int)(eeprom_read>>8);
               printf("PB-DAC3 Reference Channel %c: $%3x\n",name[cc],
                                        reference[cc]);
               printf("PB-DAC3 Fullscale Error Channel %c: %dLSB\n",name[cc],
                                        fullscale_error[cc]);
```

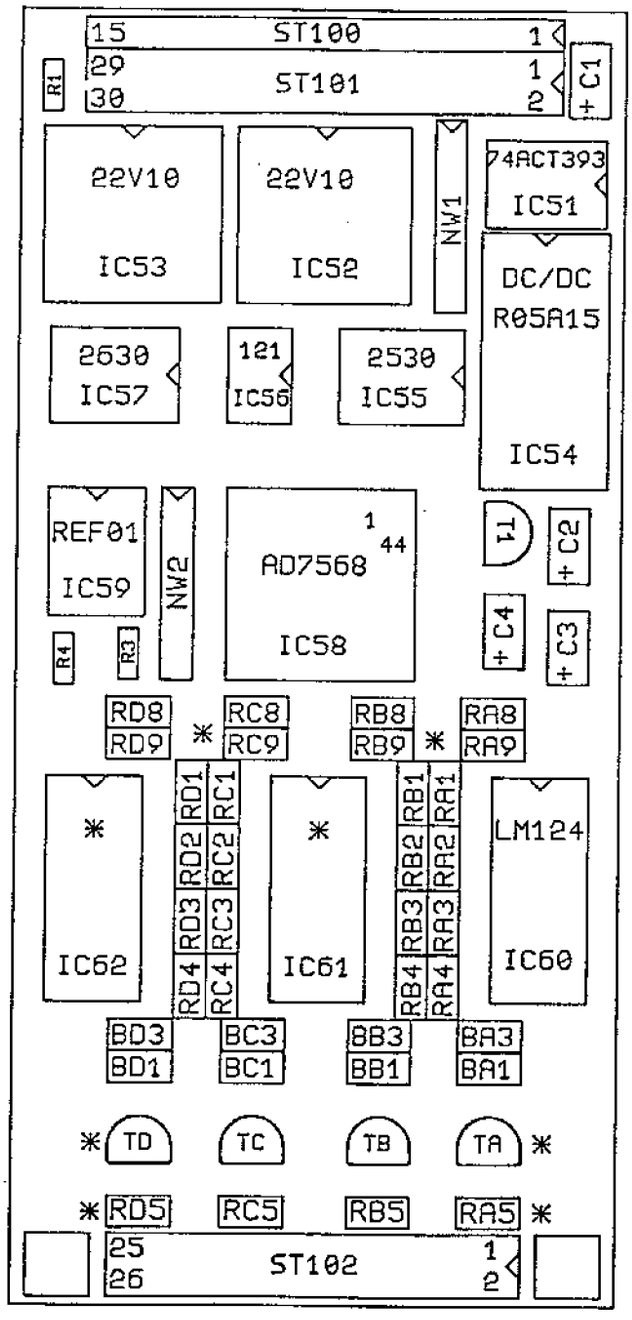
```
            DAC_preload(cc+REF,reference[cc]);
            printf("\n");
            eeprom_command=eeprom_command+0x0080;
            eeprom_command=eeprom_command+0x0080;
            if (mode[cc]!=1) eeprom_command=eeprom_command+0x0080;
        }

        DAC_update;
        printf("Latched preloaded registers to DAC outputs!\n");

        /* Write out a value to channel A */

        chan=A_CHA;
        value=0x800;                    /* any half range value */
        DAC_out(chan,value);
        printf("Set channel A to half range\n\n");

        exit(((();
}


/* program end  */
```

A.1.2 Assembler Program

```
* Channel A maximum/minimum_endless toggle program example

BASE      equ         $fe2400             base address VMOD2 default
DACOUT    equ         $00                 output register
STATUS    equ         $03                 status register offset

          move.l      #$FFF0,d2           maximum A=0
          move.l      #$0000,d1           minimum A=0

full:     cmpi.b      #$fa,BASE+STATUS    register ready ?
          bne.s       full                no, wait
          move.w      d2,BASE+DACOUT      write out full scale
          move.l      #2,d0               sleep wake up
          os9         F$Sleep             by tick timer
          bra.s       zero
          nop                             dummy nop

zero:     cmpi.b      #$fa,BASE+STATUS    end of conversion ?
          bne.s       zero                no, wait
          move.w      d1,BASE             write out zero scale
          move.l      #2,d0               sleep wake up
          os9         F$Sleep             by tick timer
          bra.s       full


* end of file
```

This page has been intentionally left blank

ORIGINAL



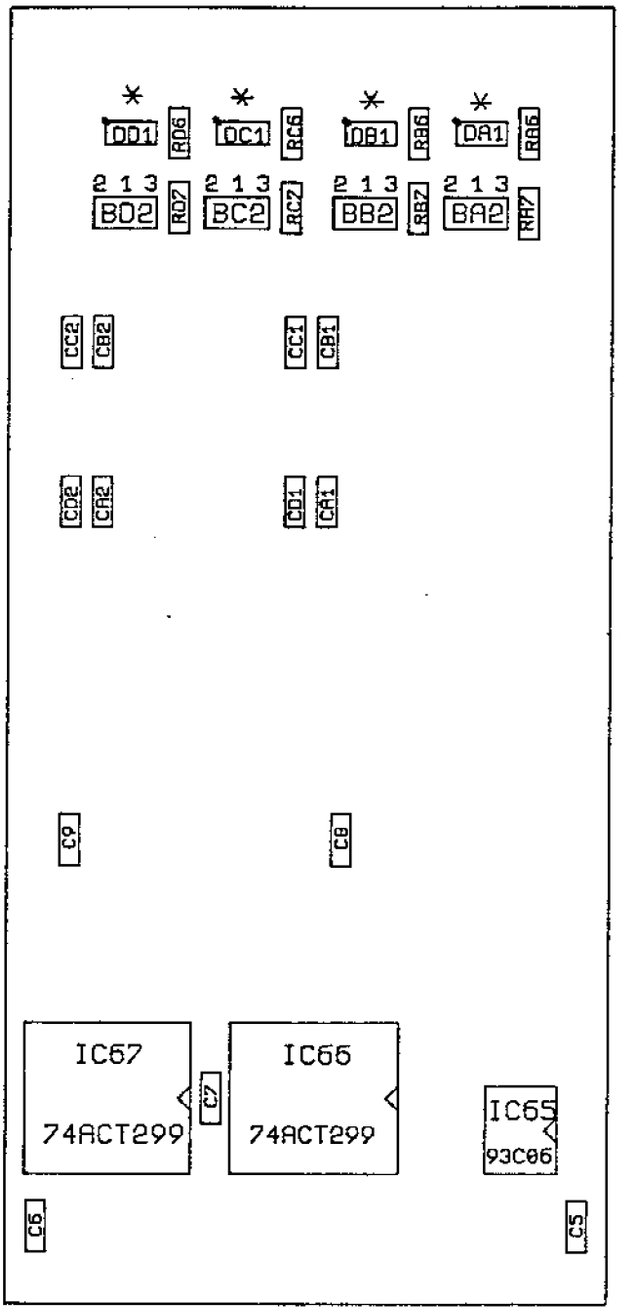
B-Seite / B-side

* entspr. Bestell-Nr / according to order-no 5230-35/x

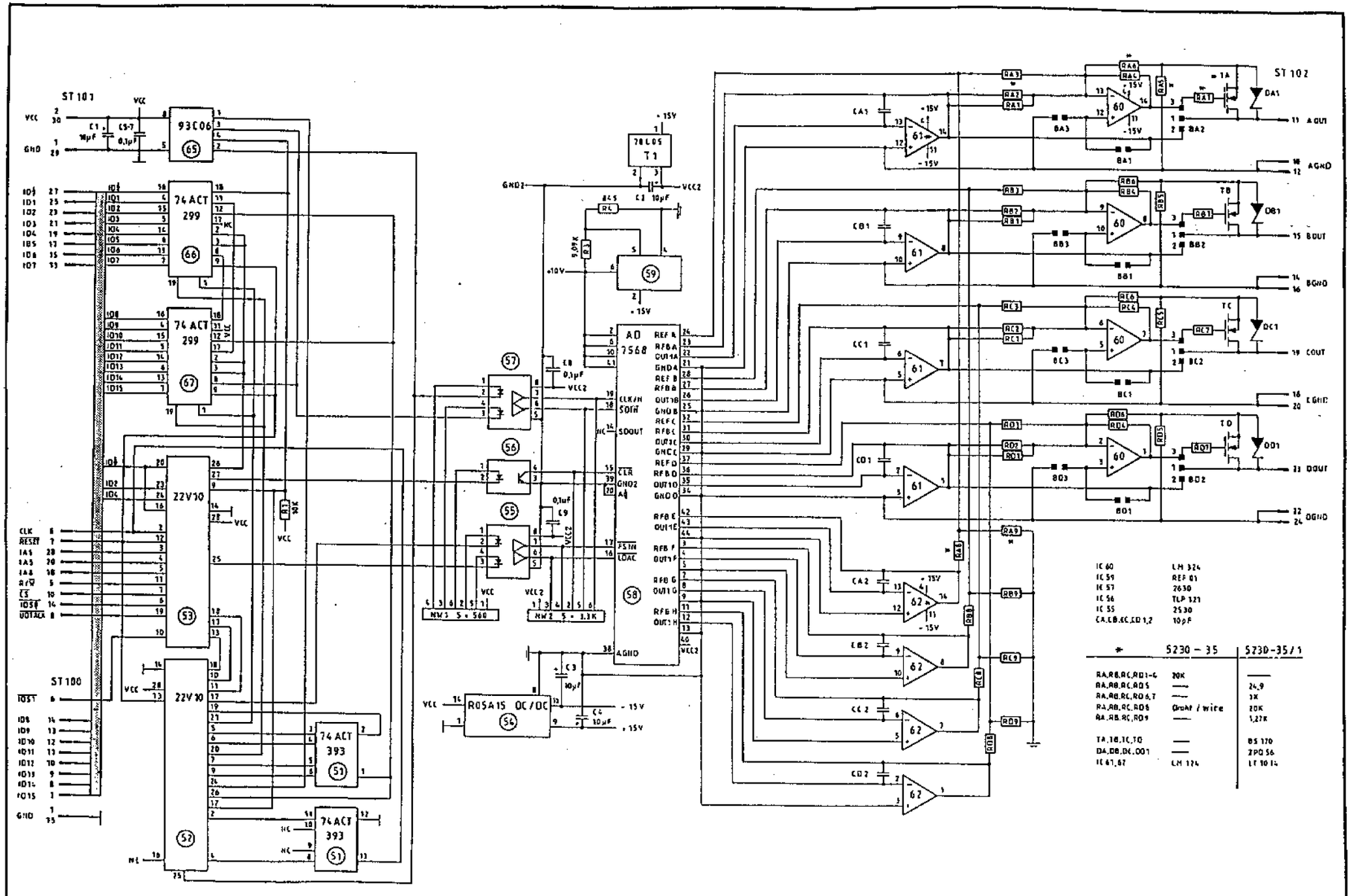
Modular Computers®	PB - DAC 3		31.112 - 1240.1
	Date	Aug 05,1993	Index 01
			Page 1 of 2

ORIGINAL

L-Seite / L-side



Modular Computers®	PB – DAC 3		31 112 –1240 1
	Date	Aug 05,1993	Index 01
			Page 2 of 2



* entspr. Bestell-Nr. / according to order-no 5230 - 35/x

ORIGINAL

Modular Computers®

PB – DAC 3		31.112 – 1240.1	
Date	Aug 05, 1993	Index	01