DAC488 User's Manual

12-Bit Digital-to-Analog Converter Systems; Models DAC488/2 and DAC488/4

the smart approach to instrumentation ™

IOtech, Inc.
25971 Cannon Road
Cleveland, OH 44146-1833
Phone: (440) 439-4091
Fax: (440) 439-4093
E-mail: sales@iotech.com
Internet: www.iotech.com

DAC488 User's Manual
12-Bit Digital-to-Analog Converter Systems
Models DAC488/2 and DAC488/4

p/n 107-0901 Rev. 1.0

Released Per EO#1927R11

June 1998

Warranty

Your IOtech warranty is as stated on the product warranty card. You may contact IOtech by phone, fax machine, or e-mail in regard to warranty-related issues.
Phone: (440) 439-4091, fax: (440) 439-4093, e-mail: sales@iotech.com

Limitation of Liability

IOtech, Inc. cannot be held liable for any damages resulting from the use or misuse of this product.

Copyright, Trademark, and Licensing Notice

All IOtech documentation, software, and hardware are copyright with all rights reserved. No part of this product may be copied, reproduced or transmitted by any mechanical, photographic, electronic, or other method without IOtech's prior written consent. IOtech product names are trademarked; other product names, as applicable, are trademarks of their respective holders. All supplied IOtech software (including miscellaneous support files, drivers, and sample programs) may only be used on one installation. You may make archival backup copies.

FCC Statement

IOtech devices emit radio frequency energy in levels compliant with Federal Communications Commission rules (Part 15) for Class A devices. If necessary, refer to the FCC booklet *How To Identify and Resolve Radio-TV Interference Problems* (stock # 004-000-00345-4) which is available from the U.S. Government Printing Office, Washington, D.C. 20402.

CE Notice

Many IOtech products carry the CE marker indicating they comply with the safety and emissions standards of the European Community. As applicable, we ship these products with a Declaration of Conformity stating which specifications and operating conditions apply.

Warnings and Cautions

Refer all service to qualified personnel. This caution symbol warns of possible personal injury or equipment damage under noted conditions. Follow all safety standards of professional practice and the recommendations in this manual. Using this equipment in ways other than described in this manual can present serious safety hazards or cause equipment damage.

This warning symbol is used in this manual or on the equipment to warn of possible injury or death from electrical shock under noted conditions.

This ESD caution symbol urges proper handling of equipment or components sensitive to damage from electrostatic discharge. Proper handling guidelines include the use of grounded anti-static mats and wrist straps, ESD-protective bags and cartons, and related procedures.

Calibration Notice

As applicable, IOtech calibrates its hardware products to published specifications. Periodic hardware calibration is not covered under the warranty and must be performed by qualified personnel as specified in this manual. Improper calibration procedures may void the warranty.

Quality Notice

IOtech has maintained ISO 9001 certification since 1996. Prior to shipment, we thoroughly test our products and review our documentation to assure the highest quality in all aspects. In a spirit of continuous improvement, IOtech welcomes your suggestions.

Introduction to this Manual

This manual covers the setup and operation of the DAC488/4 and the DAC488/2 digital-to-analog converter systems. This manual is organized as follows:

- ***Chapter 1 - DAC488 Overview*** gives an overview of the basic features of a DAC488 system. DAC488 accessories and specifications are also provided.
- ***Chapter 2 - DAC488 Setup*** explains in detail the various hardware features of the DAC488 unit.
- ***Chapter 3 - DAC488 Power & Assembly*** provides detailed instructions on line-voltage selection and fuse replacement, rack-mount and bench-top installation, as well as power-up activation.
- ***Chapter 4 - DAC488 Operation*** explains the DAC488 theory of operation. Descriptions of the DAC488 command groups, triggering, internal buffer, control modes, as well as IEEE 488 bus support, are provided.
- ***Chapter 5 - IEEE 488 Background*** describes the history and the basic concepts of IEEE 488 operation, including the various IEEE 488 bus commands and bus lines.
- ***Chapter 6 - DAC488 Calibration*** explains the manual and automatic procedures for calibrating the DAC488 unit.
- ***Chapter 7 - DAC488 Commands*** discusses the entire command set covering the DAC488/4 and DAC488/2 units. The command syntax, groups, and reference are provided. The description format of the individual commands includes the command type, syntax, description, and an example program excerpt.
- ***Chapter 8 - Troubleshooting*** provides a reference for possible solutions to technical problems. Before calling for technical assistance, refer to this chapter.
- The ***Appendix*** provides background information concerning the IEEE 488 bus, the serial bus, and ASCII controls.
- The ***Index*** provides a comprehensive alphabetical listing of the main terms and topics in this manual. Also, the *Abbreviations* on the last pages of this manual, provides an overall list of abbreviations, including acronyms and ASCII control codes, as an additional reference for this manual and for other related literature.

CAUTION

Using this equipment in ways other than described in this manual can cause personal injury or equipment damage. Before setting up and using your equipment, you should read *all* documentation that covers your system. Pay special attention to cautions and warnings formatted like this one.

Table of Contents

1 - DAC488 Overview

2 - DAC488 Setup

3 - DAC488 Power & Assembly

4 - DAC488 Operation

5 - IEEE 488 Background

6 - DAC488 Calibration

7 - DAC488 Commands

8 - Troubleshooting

A - Appendix

Notes

DAC488 Overview 1

Introduction

The DAC488 is a multiple output Digital-to-Analog Converter (DAC) interface for the IEEE 488 bus. Each port uses a 12-bit plus-sign D/A converter and is isolated from IEEE common and the other ports by up to 500 volts. Each port can be programmed for full-scale output of ±1 volt, ±5 volts and ±10 volts. Voltage values may be specified in volts or bits in increments of 1 part in 4096 plus-sign. The resolution of the ranges is 250 µVolts, 1.25 mVolts and 2.5 mVolts respectively.

Four operational modes are available to allow each port to be controlled directly from the IEEE bus or in an automatic manner. An internal 8,191 sample buffer is included which can be loaded with waveform voltages. These voltages may then be output in sequence either on a periodic interval or in a triggered fashion. The DAC488 sample buffer is automatically saved in Non-Volatile RAM (NV-RAM) as it is loaded, allowing a once only configuration of the device for many applications. All configuration settings may also be saved in Non-Volatile RAM for use as the permanent defaults.

Each of the ports can be triggered from three different sources: An external trigger, a trigger command, or a Group Execute Trigger (**GET**) bus command. Since each port can be triggered independently, different trigger sources may be selected for each port.

The interface also provides 8 TTL level digital inputs and 8 digital outputs. The digital outputs may be internally configured as either TTL level or 100 mA outputs to allow interfacing with solenoids, relays or other devices. An external trigger/SRQ input is also provided which can trigger the interface or generate an SRQ on either a falling or rising edge of a TTL compatible signal. All digital signals as well as +5 Volts and ground are accessible via a 20-pin card edge connector (Standard version) or a DB25 pin connector (CE version) located on the rear panel.

Calibration may be done automatically by connecting the interface to a Keithley Model 199 DMM. No mechanical adjustments are required. After calibration, a complete calibration report will be printed if the interface is connected to an IEEE printer.

Throughout this manual, the term “DAC488” refers to either a DAC488/4 or a DAC488/2.

Accessories

Additional accessories that can be ordered for the DAC488 include:

- ***CA-1:*** Power Cable
- ***CA-7-1:*** 1.5 foot IEEE 488 Cable
- ***CA-7-3:*** 6 foot shielded IEEE 488 Cable
- ***CA-7-4:*** 6 foot reverse entry IEEE 488 Cable
- ***CA-8-20:*** 6 foot, 20 conductor ribbon cable with a card edge connector on one end, the other end unterminated.
- ***CA-35-2:*** 2 foot, DB25 male-to-female pin Connector
- ***CA-35-6:*** 6 foot, DB25 male-to-female pin Connector
- ***CN-40-12:*** DAC488 Analog Output Connector
- ***CN-6-20:*** 20-pin card edge Connector.
- ***CN-20:*** Right Angle IEEE 488 adapter, male and female
- ***CN-22:*** IEEE 488 multi-tap bus strip, four female connectors in parallel
- ***CN-23:*** IEEE 488 panel mount feed-through connector, male and female

Specifications

CAUTION

Please read this manual carefully! If equipment is used in any manner not specified in this manual, the protection provided by the equipment may be impaired.

WARNING

Do not use the DAC488 unit outdoors! This unit is intended for indoor use only! Unexpected outdoor conditions could result in equipment failure, bodily injury or death!

Digital I/O

Terminal Installation Category: *Standard:* Not Applicable. *CE:* Category 1.
Transistor-Transistor Logic (TTL) Levels: Digital outputs will drive 2 TTL loads or sink 100mA (selectable using internal configuration jumper).
Digital Inputs and Digital Outputs: 8 inputs, TTL level compatible; 8 outputs, internally selectable TTL level compatible or open collector with 100 mA @ 50 V drive capability.
Service Request/External Trigger Input: 1 line, TTL level compatible.
Connectors: *Standard:* One 20-pin card edge connector. *CE:* One DB25 (25-pin) connector.

Analog Output

WARNING

There may be up to 200 V isolation potential between analog common and earth (CE version) or up to 500 V isolation potential (Standard version). Avoid contact with analog output terminals and the cables connected to them. Do not apply power until after the AC line and analog output connections have been properly made. The AC line must remain connected to the unit whenever the analog output terminals are connected to a device exceeding 60 VDC or 30 Vrms common mode voltage. Contact with voltage potentials could cause bodily injury or death!

CAUTION

Solid wire is recommended for the analog output terminal connections. If stranded wire is used, then strip the wire 6 mm, and twist or tin the stripped end. After tightening the connector, check that no loose wire strands are present.

Terminal Installation Category: *Standard:* Not Applicable. *CE:* Category 1.

DC Output Voltage and Resolution

1V Range: ± 1.02375 V, 250 µV/bit

5V Range: ± 5.11875 V, 1.25 mV/bit

10V Range: ± 10.2375 V, 2.5 mV/bit.

DC Output Current and Output Resistance: 10 mA maximum; and <500 mΩ (typical)

Accuracy (25 ± 5°C; Iout = 1 mA):

1V Range: ±(0.05% of setting + 1 mV)

5V Range: ±(0.05% of setting + 3 mV)

10V Range: ±(0.05% of setting + 10 mV)

Warm Up Time to Rated Accuracy: 1 hour

Temperature Coefficient (0° to 20° C, and 30°to 50° C): ±(0.002% of setting + 100 µV) / °C

Channel to Channel Isolation: *Standard:* 500 V max, 100,000 V-Hz.

CE: 200 volts max, 100,000 V-Hz

Channel to Digital Low Isolation: *Standard:* 500 V max, 100,000 V-Hz.

CE: (Channel Common to Earth) 200 V max, 100,000 V-Hz

Connectors: One 12-pin quick disconnect analog.

IEEE 488

CAUTION

The IEEE 488 terminal must only be used to control a non-isolated IEEE 488 system. The common mode voltage (cable shell to earth) must be zero.

Terminal Installation Category: *Standard:* Not Applicable. *CE:* Category 1.

Implementation: SH1, AH1, T4, TE0, L4, LE0, SR1, RL0, PP0, DC1, DT1,E1.

Terminators: Programmable CR, LF, LF CR and CR LF with EOI.

Programmable: IEEE Bus Terminators, EOI Control, SRQ Mask, Port Voltage, Gain, Offset, Digital I/O lines, Output Format, Internal Buffer, Mode.

Connector: Standard IEEE 488 connector with metric studs.

General

WARNING

Service: This product contains no operator serviceable parts. Fuse replacement and the changing of selected line voltage must be performed by qualified service personnel. Never open the DAC488 case while it is connected to the AC line, or when analog output terminals are connected to a device exceeding 60 VDC or 30 Vrms common mode voltage!

Terminal Installation Category: *Standard:* Not Applicable. *CE:* Category 2 for Line Voltage Input terminal. All other terminals are Category 1.

Dimensions: 425 mm wide x 45 mm high x 309 mm deep (16.75" x 1.75" x 12")

Weight: 3.32 kg. (7.31 lbs)

Environment: *Standard:* Indoor use, 0° to 50°C; 0 to 95% RH (non-condensing).
CE: Indoor use at altitudes below 2000 meters, 0° to 40°C; 0 to 80% RH up to 31°C decreasing linearly 4% RH/°C to 40°C.

Controls: Power Switch (external), line voltage selection switch (internal), mode and IEEE address DIP switch (external), calibration enable switch (external), digital output port configuration jumper (internal)

Indicators: LEDs for TALK, LISTEN, SRQ, ERROR, TEST and POWER

Power: Internally selectable 105-125 V or 210-250 V; 50-60 Hz, 35 VA Max

Fuse: 3/8A, Slo Blo, 3AG (for 105-125 V power line) or 3/4A, Slo Blo, 3AG (for 210-250 V power line)

CAUTION

Line Voltage: The protective conductor terminal on the AC line connector must be connected to an external protective earthing system. Failure to make such a connection will impair protection from shock. If common mode voltages connected to the analog input terminals exceed 60 VDC or 30 Vrms, the AC line connector must remain connected even when the unit is powered “off.” If it becomes necessary to disconnect the AC line connector, you must first disconnect all power to the analog input terminals.

CAUTION

Fuse Failure: Fuse failure indicates a possible problem within the device circuitry. If a fuse blows, contact a qualified service representative. Replacement fuses are to be installed by qualified service personnel with the unit disconnected from the power source and with all other terminals disconnected. If the line voltage selector is changed, then the fuse designated for that line voltage must be used.

DAC488 Setup 2

The Package

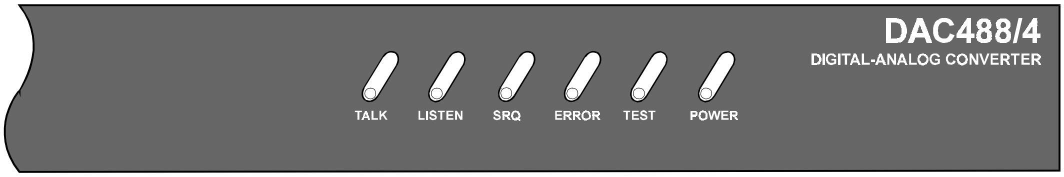
All DAC488 components are carefully inspected prior to shipment. When you receive your digital-to-analog converter system, carefully unpack all items from the shipping carton and check for any damage which may have occurred during shipment. Promptly report the damage to the shipping agent and your sales representative. Retain all shipping materials in case you must return the unit to the factory.

Every DAC488 package includes the following items listed by part number:

- ***DAC488/2 or DAC488/4:*** Digital-to-Analog Converter with Digital I/O and IEEE 488
- ***WINDOWS PANELS:*** Programmed Disks in IBM format, including Installation Card and Licensing Agreement (*PR-10*)
- ***107-0901:*** *DAC488 User's Manual*
- ***PR-2:*** Warranty Card
- ***DAC488-003:*** Accessories Kit for DAC488, which includes the following:
  - ***CA-1:*** Power Cable
  - ***CA-35-2 or CA-35-6:*** DB25 Male-to-Female Connector, 2 or 6 feet respectively (CE version)
  - ***CN-6-20:*** 20-pin Card Edge Connector (Standard version)
  - ***CN-40-12:*** 12-pin Analog Output Connector
  - ***FE-1:*** Rubber Feet (4)
  - ***EN-6:*** Rack Ears (2)
  - ***HA-41-6:*** Rack Screws (4)
  - ***PR-9:*** Rack Mounting Instructions
  - ***FU-1-.375:*** 3/8A Replacement Fuse
  - ***FU-1-.75:*** 3/4A Replacement Fuse

Hardware Setup

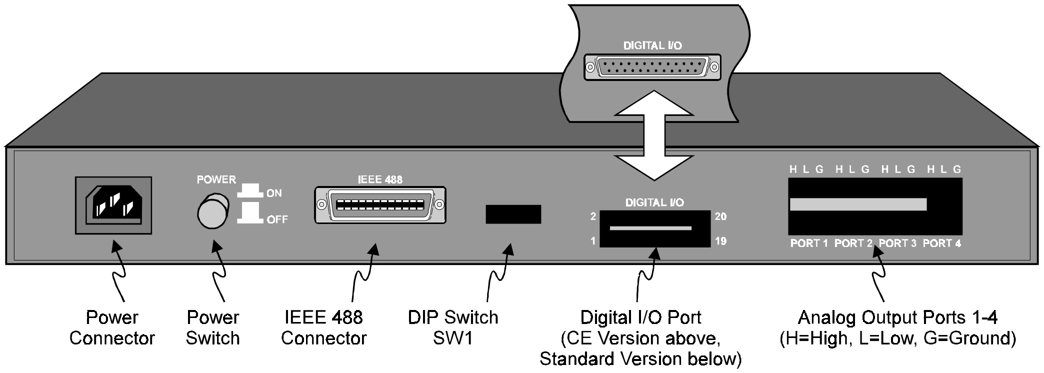
Front & Rear Panel Layouts



DAC488 Front Panel Indicators

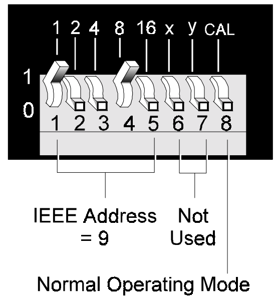
Six LED indicators on the DAC488 front panel, display the unit status. The following list describes the functions of these indicators:

- ***TALK:*** ON when DAC488 is in the Talker state; OFF when in the Idle or Listener state.
- ***LISTEN:*** ON when DAC488 is in the Listener state; OFF when in the Idle or Talker state.
- ***SRQ:*** ON when DAC488 generated a service request; OFF when no SRQ is pending.
- ***ERROR:*** ON when an error has occurred; OFF when no error condition exists.
- ***TEST:*** ON when used in conjunction with the *Test* (**W**) command to verify that communication has been established with the DAC488. The TEST LED will flash when the DAC488 is calibrating in the auto-calibration mode. The TEST LED will then remain lit when finished calibrating in the auto-calibration mode.
- ***POWER:*** ON when power is being applied while the power switch is in the ON position; OFF when power is not present.


DIGITAL I/O
POWER
ON
OFF
IEEE 488
DIGITAL I/O
2
20
1
19
H L G H L G H L G H L G
PORT 1 PORT 2 PORT 3 PORT 4
Power Connector
Power Switch
IEEE 488 Connector
DIP Switch SW1
Digital I/O Port (CE Version above, Standard Version below)
Analog Output Ports 1-4 (H=High, L=Low, G=Ground)

DAC488 Rear Panel Layout

Default Settings



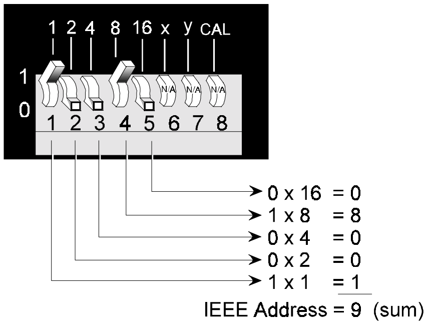
SW1 Factory Default Settings

The DAC488 unit may be operated from either 110 VAC or 220 VAC. The operating voltage is set by an internal switch. The factory set operating voltage appears on the label placed over the power jack on the rear panel. To change the operating voltage, see section *Line Voltage Selection* in Chapter 3.

The DAC488 unit has one 8-microswitch DIP switch (SW1) accessible from the rear panel. This DIP switch determines the IEEE 488 bus address and its operating mode. The switch is read only when the unit is powered on and should be set prior to applying power. The figure illustrates the factory default for the DIP microswitch settings. To modify the default settings, disconnect the power cord from the AC line and change the microswitch settings using a small screwdriver. The enclosure does not need to be opened to change the DIP microswitch settings.

IEEE 488 Bus Address Selection

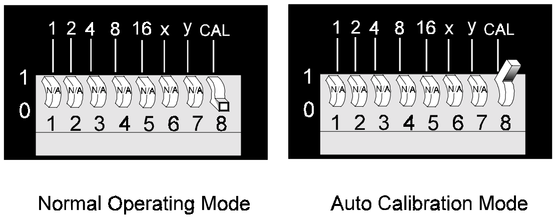
The IEEE 488 bus address is set by positioning the rear-panel DIP microswitches 1 through 5. The address can be set from 0 through 30 and is read only at power on. The address is selected by simple binary weighting with microswitch 1 being the Least Significant Bit (LSB) and microswitch 5 the Most Significant Bit (MSB). The factory default setting is address 9, as shown in the diagram. If address 31 is selected, it defaults to address 30 because the IEEE 488 standard has reserved address 31.



SW1 IEEE Bus Address Selection

Mode Selection

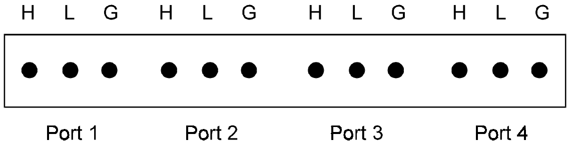
The DAC488 can be operated in one of two modes: *Normal operating mode* or *automatic calibration mode*, which are selected by the rear-panel DIP microswitch 8. The factory default setting for microswitch 8 is *normal operating mode*, as shown in the diagram.



SW1 View for Startup Mode Selection

Analog Output Ports

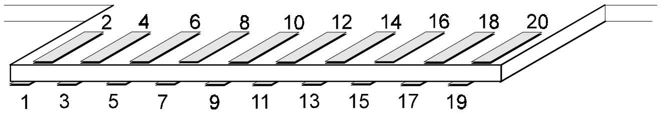
The DAC488 can be considered as multiple isolated IEEE 488 to Analog Converters. The unit occupies one IEEE 488 bus address. Each port has a low (L), high (H), and ground (case ground) line. The case ground line may be connected to the shield of shielded cable if this type of cable is used to carry the analog signals. The pinouts for all analog ports are labeled on the rear panel of the unit. Each analog output is capable of sourcing and sinking a maximum current of 10 mA.



DAC488 Analog Output Connector Pinout

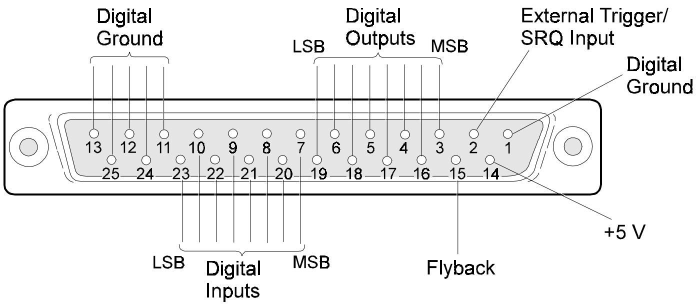
Digital I/O Port

The DAC488 has eight digital input lines and eight digital output lines. At power on, the configuration of digital output lines is determined by the options saved as the startup configuration. In order to set the digital output lines to a particular state upon power on, the *System Defaults* (**S**) command must be used after all the lines have been configured. For example, if it is desired to have digital output line 1 set high and the other lines set low at power on, the lines would be set to this configuration using the *Digital Output* (**D**) command. Then this configuration would be saved using the *System Defaults* (**S**) command.



Note: Refer to the chart below for pin descriptions.

DAC488 Rear Panel Digital I/O Connector Pinout
(Standard Version)



Note: Refer to the chart below for pin descriptions.

DAC488 Rear Panel Digital I/O Connector Pinout
(CE Version, Using DB25 Female I/O Port)

Digital I/O Port Pinouts

Connector Pin		Connector Pin Description	Connector Pin		Connector Pin Description
Standard (20-pin)	CE Version (DB25)		Standard (20-pin)	CE Version (DB25)	
1	23	Digital Input Bit 1 (LSB)	17	15	Flyback
2	10	Digital Input Bit 2	18	2	Ext. Trigger/SRQ Input
3	22	Digital Input Bit 3	19	14	+5 V (≤ 50 mA load)
4	9	Digital Input Bit 4	20	1	Digital Ground
5	21	Digital Input Bit 5		11	Digital Ground
6	8	Digital Input Bit 6		12	Digital Ground
7	20	Digital Input Bit 7		13	Digital Ground
8	7	Digital Input Bit 8 (MSB)		24	Digital Ground
9	19	Digital Output Bit 1 (LSB)		25	Digital Ground
10	6	Digital Output Bit 2			
11	18	Digital Output Bit 3			
12	5	Digital Output Bit 4			
13	17	Digital Output Bit 5			
14	4	Digital Output Bit 6			
15	16	Digital Output Bit 7			
16	3	Digital Output Bit 8 (MSB)			

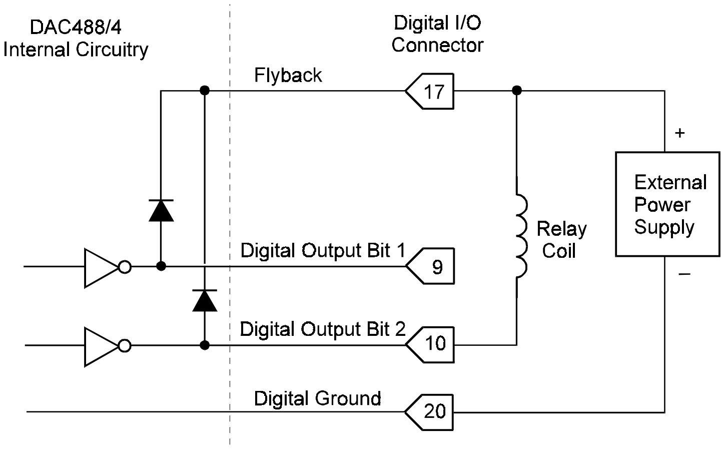
Transistor-Transistor Logic (TTL) Levels

The digital output lines will drive 2 TTL loads. All digital input lines are less than 1.5 TTL loads. All inputs are protected against damage due to high static voltages. Normal precautions should be taken to limit the input voltages to the range of -0.3 to +7.0 volts. All I/O lines are referenced to Digital Ground (Standard Version Pin 20, CE Version Pin 1).

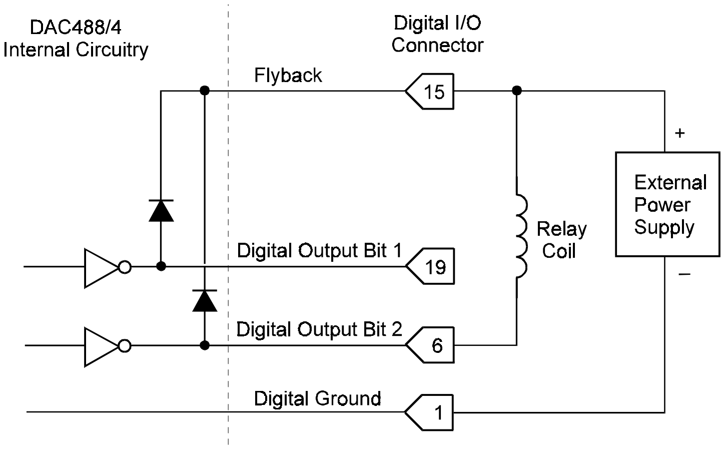
High Voltage/High Current Digital Outputs

In addition to interfacing with TTL logic levels, the digital output lines can be configured as high voltage/high current outputs. These outputs can sink up to 100 mA at 50 VDC through the use of open collector drivers with integral diodes for inductive load transient suppression. This allows for interfacing the digital outputs with relays, lamps and solenoids.

As shown in the following schematics, the gates driving the digital outputs (when configured for high voltage/high current operation) contain internal diodes to suppress inductive transients. The cathodes of these diodes are tied together and connected to the Flyback pin on the digital I/O port. By connecting the Flyback pin to the positive end of the supply driving the relays, the internal diodes will be connected in parallel with the relay coils and thus prevent inductive spikes from damaging the internal circuitry of the DAC488.

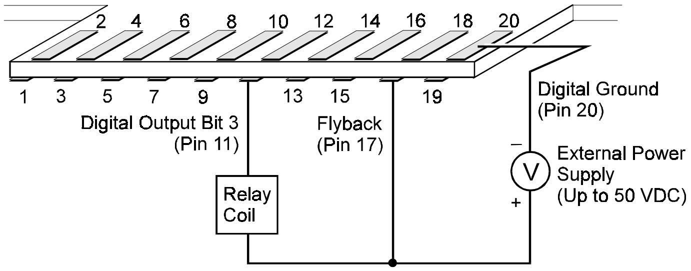


DAC488 High Voltage/High Current Internal Circuitry Schematic (Standard Version)

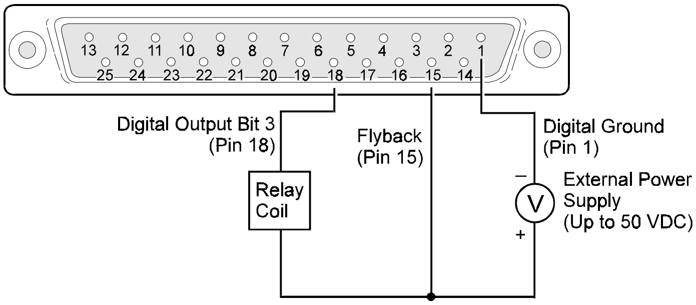


DAC488 High Voltage/High Current Internal Circuitry Schematic (CE Version)

When used with relays or solenoids, the Flyback pin on the edge card connector should be connected to the positive supply lead of the power supply used with the external devices being driven, as shown in the following figures. To configure the digital output lines for this purpose, it is necessary to open the enclosure and reposition the configuration jumper. For instructions on how this is done, see section *To Configure the Digital Output Lines* in the following text.



Edge Connector Using a Flyback Line
(Standard Version)

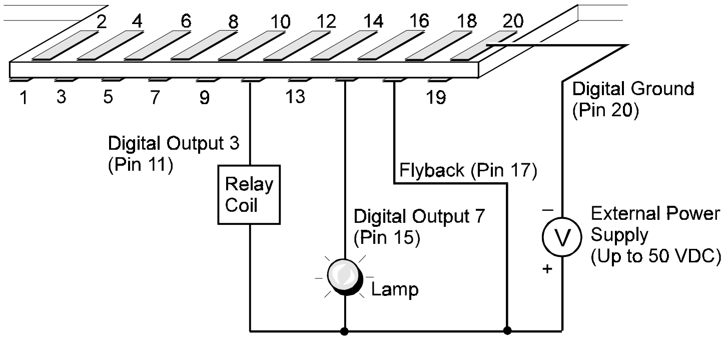


Pin Connector Using a Flyback Line
(CE Version, Using DB25 Female I/O Port)

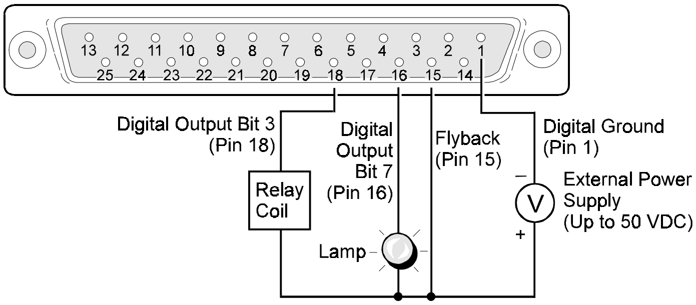
Once the digital output lines have been configured for high level operation, they can be used to drive devices such as relays, solenoids and displays. For example, a typical application may require a lamp and a relay to be driven by the Standard version DAC488. For this example, the relay is connected to digital output pin 11 and the lamp is connected to digital output pin 15, as shown in the following diagram.

Using this same example for the CE version DAC488, the relay is connected to digital output pin 18 and the lamp is connected to digital output pin 16, also shown in the next diagram.

Since a relay is used for this application, the Flyback terminal is connected to the positive terminal of the external power supply. Note also that the Ground lead of the power supply should be connected to pin 20 (Digital Ground) on the Standard card edge connector, or to pin 1 (Digital Ground) on the CE pin connector.



Edge Connector Using Lines for a Lamp and Flyback
(Standard Version)



Pin Connector Using Lines for a Lamp and Flyback
(CE Version, Using DB25 Female I/O Port)

External Trigger/SRQ Input

The External Trigger/SRQ input (Standard pin 18; CE pin 2) can be used to trigger the DAC488 once it has been properly armed. The DAC488 can be programmed to trigger on a positive-going or negative-going edge by using the *External Trigger Mask* (`Q`) command. Any TTL level signal may be used as a trigger pulse. A trigger pulse may also be used to generate an SRQ by using the *Service Request Mask* (`M`) command. The maximum rate at which the DAC488 can be triggered is 1 trigger pulse every 1 millisecond.

To Configure the Digital Output Lines

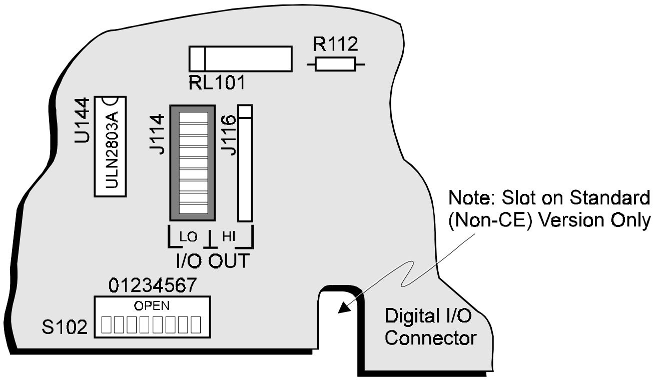
WARNING

⚠ **Service: This product contains no operator serviceable parts. Fuse replacement and the changing of selected line voltage must be performed by qualified service personnel. Never open the DAC488 case while it is connected to the AC line, or when analog output terminals are connected to a device exceeding 60 VDC or 30 Vrms common mode voltage!**

CAUTION

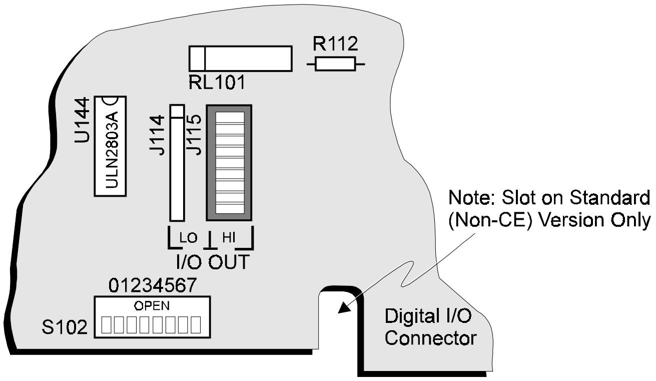
⚠ **Do not connect external high level devices to the digital output lines unless they have first been configured for this purpose. Otherwise the interface may be damaged.**

1. First, disconnect any cables from the *analog output terminals*. Next, disconnect the power cord from the *AC line* and from the interface. Then disconnect any other cables prior to disassembly.
2. Place the interface on a flat surface. Remove the six screws on top of the case and remove the top cover.
3. Located in front of the rear-panel DIP switch (SW1) are three sockets and a configuration jumper. This jumper is factory set to configure the digital output lines for low level TTL logic levels, as shown in the figure.
4. To configure the digital output lines as high voltage/high current outputs, first remove the configuration jumper.
5. Next, reinsert it so that the center socket is now connected to the right socket, as shown in the second figure.
6. Once the jumper has been repositioned for your particular application, make note of the new setting for later reference.
7. Carefully reassemble the unit.



Note: Slot on Standard (Non-CE) Version Only

Digital Output Configuration Jumper Configured for TTL Level Output (Standard and CE Versions, Factory Setting)



Note: Slot on Standard (Non-CE) Version Only

Digital Output Configuration Jumper Configured for High Level Output (Standard and CE Versions)

Notes

DAC488 Power & Assembly 3

Introduction

Internal Components

The heart of the DAC488 is a 6809 microprocessor [U101] supported by 16K bytes of firmware EPROM [U102] and 32K bytes of battery-backed-up RAM [U103]. Interface to the IEEE 488 bus is accomplished by a 9914 interface chip [U105]. A 65C22 [U109] generates real-time interrupts for the firmware operating system and also drives the front panel LED indicators.

Isolation of the analog ports is accomplished using separate bipolar power supplies for each port. Opto-Isolators are used to isolate the digital and analog circuitry. High voltage/high current output capability is available on the digital output port through a 74HCTLS240 and a ULN2803A peripheral driver [U144].

Calibration of the analog ports through program control is accomplished by the use of two 8-bit D/A converters on each analog circuit. One converter is used to adjust the gain and the other converter to adjust the offset.

Factory Service

Chapter 8 provides a troubleshooting reference for possible solutions to technical problems. Before calling for technical assistance, refer to this chapter.

However, if problems in using the DAC488 still persist, you should contact the factory. Many problems can be resolved by discussing the problems with the appropriate applications department. If the problem cannot be solved by this method, you will be instructed as to the proper return procedure.

Power Line & Fuse Configuration

The power configuration of the DAC488 unit consists of selecting the line voltage and replacing the fuses. Each DAC488 unit has a factory default to operate at 105-125 volts AC. However, each unit may be operated at either 105-125 or 210-250 VAC.

WARNING

Do not use the DAC488 unit outdoors! The unit is intended for indoor use only! Outdoor conditions could result in equipment failure, bodily injury or death!

WARNING

Never disassemble the DAC488 case while it is connected to the AC power line, or when analog output terminals are connected to a device exceeding 60 VDC or 30 Vrms common mode voltage! Internal voltage potentials exist which could cause bodily injury or death!

To change the operating voltage of the DAC488 unit, it is necessary to open the enclosure. However, before modifying the voltage, disconnect any input or output connections from the rear panel of the affected unit and then disconnect the power cord from the power line terminal.

Line voltage must be set for 105-125 or 210-250 VAC to match the power being supplied to the DAC488 unit. If the line voltage is changed, the fuse must also be changed. Refer to the following text for the line voltage switch and fuse locations.

Line Voltage Selection

CAUTION

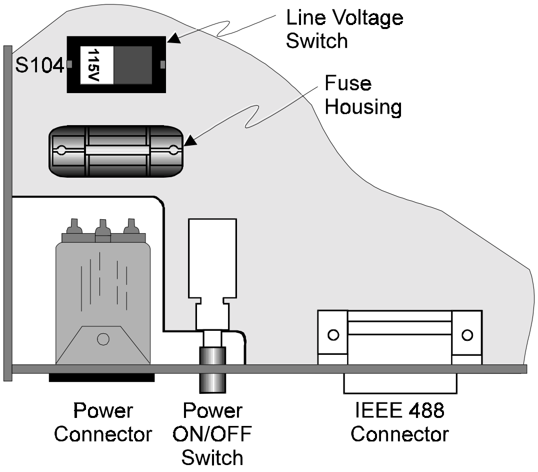
Service: This product contains no operator serviceable parts, or operator replaceable parts within the case. Fuse replacement and the changing of selected line voltage must be performed by qualified service personnel. Never open the DAC488 case while it is connected to the AC line, or when analog output terminals are connected to a device exceeding 60 VDC or 30 Vrms common mode voltage. Contact with voltage potentials could cause bodily injury or death!

As already mentioned, the DAC488 may be operated with 105-125 or 210-250 VAC, 50-60 Hz power, as set by its internal line-voltage switch (labelled S104). Each unit is shipped from the factory with this operating voltage setting marked on its rear panel. If this is not the appropriate power setting to be supplied to the unit, then the line voltage and power fuse must be changed to avoid damage to the unit. The line-voltage selection procedure is outlined in the following steps.

Note: If disassembly or disconnections are necessary, first turn off the power, then disconnect the analog output cables, next disconnect the AC power line, and then any other cables, prior to disassembly.

To Select the Line Voltage

1. Turn off the power, then disconnect the analog output cables. Next, disconnect the AC power line cord, and then all other test cables from the unit.
2. Place the unit on a flat surface. Remove the six screws on top of the case and remove the top cover.
3. Located next to the main power supply transformer is the line voltage selection switch (labelled S104). Using a small screwdriver, insert the tip of the screwdriver into the slot of the switch and slide the switch to the left or right *until it "clicks" into place* with the desired line voltage selection visible.



Top View of DAC488 Circuit Board

CAUTION

It is possible to place the line voltage switch (S104) in a partial position which could cause equipment damage or malfunction. When changing the position of the line voltage selection switch (S104), make sure the switch is completely positioned to the 115 V or 220 V selection. The switch will "click" into place when properly positioned.

4. Install a power line fuse appropriate for the line voltage. See section *Fuse Replacement – Step 3,* following this section.

CAUTION

Fuse Failure: Fuse failure indicates a possible problem within the device circuitry. If a fuse blows, contact a qualified service representative. Replacement fuses are to be installed by qualified service personnel with the unit disconnected from the power source and with all other terminals disconnected. If the line voltage selector is changed, then the fuse designated for that line voltage must be used.

5. Make note of the new voltage setting for later reference.
6. Carefully reassemble the unit.

Note: For re-assembly, first reconnect the AC power line, next reconnect the analog output terminals, and then any other cables, prior to reapplying power to the entire system.

Fuse Replacement

CAUTION

Service: This product contains no operator serviceable parts, or operator replaceable parts within the case. Fuse replacement and the changing of selected line voltage must be performed by qualified service personnel. Never open the DAC488 case while it is connected to the AC line, or when analog output terminals are connected to a device exceeding 60 VDC or 30 Vrms common mode voltage. Contact with voltage potentials could cause bodily injury or death!

The DAC488 unit contains an internal AC line fuse. The fuse is located next to the internal line-voltage switch (S104). You may replace the fuse by using the procedures found in the following text.

Note: If disassembly or disconnections are necessary, first turn off the power, then disconnect the analog output cables, next disconnect the AC power line, and then any other cables, prior to disassembly.

To Replace the Fuse

1. Turn off the power, then disconnect the analog output cables. Next, disconnect the AC power line cord, and then all other cables from the unit.
2. Place the unit on a flat surface. Remove the six screws on top of the case and remove the top cover.
3. Located next to the line-voltage selection switch (labelled S104) is the power fuse. Gently pull upward on the plastic fuse housing. The entire housing with the fuse inside should be removed.
4. Open the fuse housing by pushing up on the tab on the bottom of the housing. Remove the fuse, and replace it with the proper type using the following list as a guide:
    - *For line voltage 105-125 V*, use fuse type 3/4 A, Slo Blo, 3AG
    - *For line voltage 210-250 V*, use fuse type 3/8 A, Slo Blo, 3AG

CAUTION

Fuse Failure: Fuse failure indicates a possible problem within the device circuitry. If a fuse blows, contact a qualified service representative. Replacement fuses are to be installed by qualified service personnel with the unit disconnected from the power source and with all other terminals disconnected. If the line voltage selector is changed, then the fuse designated for that line voltage must be used.

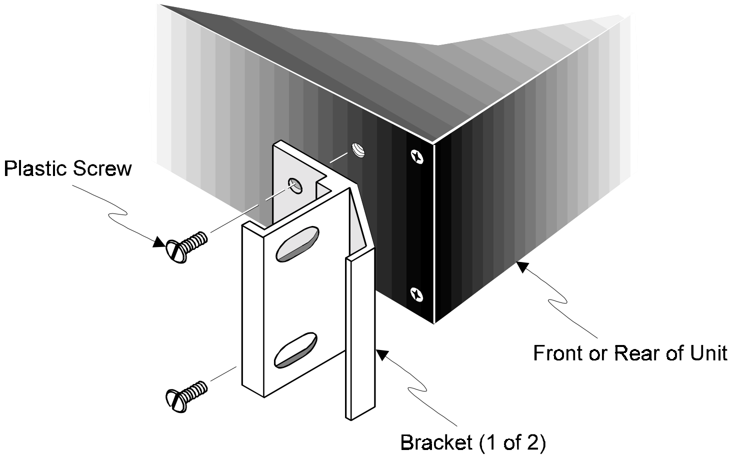
5. Close the housing. Insert the fuse into the fuse holder.
6. Make note of the new fuse rating for later reference. If you have also changed the operating line-voltage selection, return to the previous section *Line Voltage Selection – Step 5*.
7. Carefully reassemble the unit.

Note: For re-assembly, first reconnect the AC power line, next reconnect the analog output terminals, and then any other cables, prior to reapplying power to the entire system.

Rack-Mount & Bench-Top Assembly

The DAC488 includes accessories for rack-mount or bench-top assembly.

Rack Mount



Attaching a Rack-Mount Bracket

If rack-mount assembly is required, remove the two plastic screws from the pre-drilled holes on each side of the unit. Since the unit can be mounted with the front or rear panel facing the front of the rack fixture, remove only those screws from the set of holes that will be toward the front of the rack. Attach the two rack ears using the enclosed screws.

Bench Top

If bench-top assembly is required, install the self-adhesive rubber feet on the bottom of the unit approximately one inch from each corner.

Power-Up & Performance Tests

Power-Up Activation

CAUTION

Line Voltage: The protective conductor terminal on the AC line connector must be connected to an external protective earthing system. Failure to make such a connection will impair protection from shock.

LED Patterns for Error Conditions

Condition	**LED Indicators**					
	TALK	**LISTEN**	**SRQ**	**ERROR**	**TEST**	**POWER**
Normal (No Errors)	(Off)	(Off)	(Off)	(Off)	(Off)	ON
ROM Error	ON	ON	ON	ON	ON	ON
RAM Error	FLASHING	FLASHING	FLASHING	FLASHING	FLASHING	ON
NV-RAM Checksum Error	(Off)	(Off)	(Off)	ON	(Off)	ON
No Power	(Off)	(Off)	(Off)	(Off)	(Off)	(Off)

With the power cord plugged in and connected to the DAC488, turn on the unit by depressing the rear-panel power switch. All of the front-panel LED indicators should light up for approximately one second while the DAC488 performs an internal ROM and RAM self-check. At the end of this self-check all of the LED indicators should turn off except for *POWER*.

Otherwise, the DAC488 may be in an error condition according to the one of the four following LED patterns:

- ***If all of the LED indicators remain on*:** Then a ROM error has occurred.
- ***If the POWER LED indicator remains on while the rest of the LED indicators flash continuously:*** Then a RAM error has occurred. Try cycling the power to the DAC488 to determine that the error is repeatable.
- ***If the POWER and ERROR LED indicator remain on while the rest of the LED indicators are off:*** Then a checksum error on Non-Volatile RAM occurred, and the saved configurations may be lost. The checksum error condition may be cleared by doing a save of a configuration using the *System Defaults* (**S**) command. The error condition may also be cleared by using the *Query Error Status* (**E?**) command.
- ***If all of the LED indicators are off:*** Then there may not be any power being supplied to the DAC488. In this event, check to make sure that the AC power cable is securely connected at both ends. Otherwise, there may be a problem with the fuse. For more information, see section *Power Line & Fuse Configuration* earlier in this chapter.

Programming Examples

The DAC488 programming examples use the Personal488 PC/IEEE 488 controller interface and the BASIC Keyboard Controller (KBC) program. This KBC program **KYBDCTRL.BAS** is included on the disk and is listed below.

KYBDCTRL.BAS Program

```
10 ' Personal488 Keyboard Controller Program
20 '
30 ' For use with the IOtech Personal488
40 ' IEEE 488 interface
50 '
100 OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
110 IOCTL#1,"BREAK"
120 PRINT#1,"RESET"
130 OPEN "\DEV\IEEEIN" FOR INPUT AS #2
140 '
150 ON ERROR GOTO 300
160 PRINT#1,"ERROR OFF"
170 '
180 LINE INPUT "CMD> ",CMD$
190 PRINT#1,CMD$
200 '
210 IF IOCTL$(2)<>"1" THEN 180
220 PRINT INPUT$(1,2);
230 GOTO 210
290 '
300 ' Error Handler
310 '
320 IOCTL#1,"BREAK"
330 PRINT#1,"STATUS"
340 INPUT#2,ST$
350 PRINT CHR$(7);"Error #";MID$(ST$,15,2);": ";MID$(ST$,27)
360 RESUME NEXT
```

The Keyboard Controller (KBC) program allows direct communication with the DAC488 (or any IEEE 488 device on the bus) via the keyboard, and displays any responses on the screen. This program is a convenient method of exercising the DAC488 and becoming familiar with the commands and their actions.

To Run the Keyboard Controller Program

1. First, set the rear-panel DIP switch for bus address 9.
2. Connect an IEEE 488 cable from the IEEE 488 connector on the DAC488 to the IEEE 488 connector on the Personal488 PC/IEEE 488 controller interface card.
3. Turn on the DAC488 by depressing the rear-panel power switch.
4. From the MS-DOS prompt on your computer screen, run the Keyboard Controller Program **KYBDCTRL.BAS** and verify that the **CMD>** prompt appears on the screen.

DAC488 Command Responses

```
(1)     CMD>

(2)     CMD>OUTPUT 09;W1X

(3)     CMD>OUTPUT 09;W0X

(4)     CMD>ENTER 09
              A1 C0 P1 R0 V+00.00000

(5)     CMD>OUTPUT 09;P1 C0 A0 R3 V5.678

(6)     CMD>ENTER 09
              A0 C0 P1 R3 V+05.67750

(7)     CMD>OUTPUT 09;P1 C0 A0 R3 V4.321

(8)     CMD>ENTER 09
              A0 C0 P2 R2 V+04.32000
```

The above is a listing of the commands given to the DAC488 and the response received. Some commands and command responses are separated by extra vertical and horizontal spaces for clarity although these spaces will not actually appear when using the program. These eight command responses are explained as follows:

- ***Line 1:*** Initial command prompt from the Keyboard Controller program.
- ***Line 2:*** Instruct the DAC488 to turn on its *TEST* LED indicator. The *TEST* LED indicator should light up.
- ***Line 3:*** Instruct the DAC488 to turn off its *TEST* LED indicator. The *TEST* LED indicator should turn off.
- ***Line 4:*** Retrieve the DAC488 default status: Autorange on, Direct control mode, Port 1 selected, Ground range, and 0.0 volts.
- ***Line 5:*** Program the DAC488 Port 1 for 5.678 volts. A digital multimeter (DMM) connected to Port 1 should read 5.677 volts (± 10 mV) after this command string is executed.
- ***Line 6:*** Retrieve the DAC488 status: The programmed 5.678 volts was rounded to 5.67750 volts.
- ***Line 7:*** Program the DAC488 Port 2 for 4.321 volts. A digital multimeter (DMM) connected to Port 2 should read 4.320 volts (± 10 mV) after this command string is executed.
- ***Line 8:*** Retrieve the DAC488 status: The programmed 4.321 volts was rounded to 4.32000 volts.

Performance Verification

To verify that each DAC488 port is functioning within the specifications, the following procedure may be used. If the port voltage is measured using a Keithley 199 DMM/Scanner, the reading should fall between the low and high limits listed in the tables. The procedure is given for Port 1 although each port may be tested by the *Port Select* (**P**) command. It is assumed that the DAC488 is connected to the Keithley 199 DMM/Scanner using the calibration cable described in Chapter 6.

To Verify the Port Voltage – 10-Volt Range

1. Send the following command string to the DAC488: **`C0 P1 A0 R3 V10 X`** where:

   - **`C0`** = Direct Control Mode
   - **`P1`** = Port 1
   - **`A0`** = Autorange Off
   - **`R3`** = 10-Volt Range
   - **`V10`** = Output +10 Volts
   - **`X`** = Execute

2. Send the following command string to the 199 DMM: **`F0 R3 Z0 N1 S1 X`** where:

   - **`F0`** = DC Volts
   - **`R3`** = 30-Volt Range
   - **`Z0`** = Zero Disabled
   - **`N1`** = Scanner Channel 1
   - **`S1`** = 5 ½ Digit Resolution
   - **`X`** = Execute

3. For each test, the specific *Value Output* (**`V`**) command, its commanded output voltage, and its corresponding Low Limit and High Limit, are as follows:

• **`V10`** = Output +10 V	+9.985 ≤ Reading ≤ +10.015 Volts
• **`V5`** = Output +5 V	+4.990 ≤ Reading ≤ +5.010 Volts
• **`V1`** = Output +1 V	+0.994 ≤ Reading ≤ +1.006 Volts
• **`V0`** = Output 0 V	-0.005 ≤ Reading ≤ +0.005 Volts
• **`V-1`** = Output -1 V	-1.006 ≤ Reading ≤ -0.994 Volts
• **`V-5`** = Output -5 V	-5.010 ≤ Reading ≤ -4.990 Volts
• **`V-10`** = Output -10 V	-10.015 ≤ Reading ≤ -9.985 Volts

To Verify the Port Voltage – 5-Volt Range

1. Send the following command string to the DAC488: **`C0 P1 A0 R2 V5 X`** where:

   - **`C0`** = Direct Control Mode
   - **`P1`** = Port 1
   - **`A0`** = Autorange Off
   - **`R2`** = 5-Volt Range
   - **`V5`** = Output +5 Volts
   - **`X`** = Execute

2. Send the following command string to the 199 DMM: **`F0 R3 Z0 N1 S1 X`** where:

   - **`F0`** = DC Volts
   - **`R3`** = 30-Volt Range
   - **`Z0`** = Zero Disabled
   - **`N1`** = Scanner Channel 1
   - **`S1`** = 5 ½ Digit Resolution
   - **`X`** = Execute

3. For each test, the specific *Value Output* (**`V`**) command, its commanded output voltage, and its corresponding Low Limit and High Limit, are as follows:

• **`V5`** = Output +5 V	+4.9920 ≤ Reading ≤ +5.0080 Volts
• **`V2.5`** = Output +2.5 V	+2.4945 ≤ Reading ≤ +2.5055 Volts
• **`V1`** = Output +1 V	+0.9960 ≤ Reading ≤ +1.0040 Volts
• **`V0`** = Output 0 V	-0.0030 ≤ Reading ≤ +0.0030 Volts
• **`V-1`** = Output -1 V	-0.9960 ≤ Reading ≤ -1.0040 Volts
• **`V-2.5`** = Output –2.5 V	-2.4945 ≤ Reading ≤ -2.5055 Volts
• **`V-5`** = Output -5 V	-4.9920 ≤ Reading ≤ -5.0080 Volts

To Verify the Port Voltage – 1-Volt Range

1. Send the following command string to the DAC488: **`C0 P1 A0 R1 V1 X`** where:
   - **`C0`** = Direct Control Mode
   - **`P1`** = Port 1
   - **`A0`** = Autorange Off
   - **`R1`** = 1-Volt Range
   - **`V1`** = Output +1 Volt
   - **`X`** = Execute
2. Send the following command string to the 199 DMM: **`F0 R2 Z0 N1 S1 X`** where:
   - **`F0`** = DC Volts
   - **`R2`** = 3-Volt Range
   - **`Z0`** = Zero Disabled
   - **`N1`** = Scanner Channel 1
   - **`S1`** = 5 ½ Digit Resolution
   - **`X`** = Execute
3. For each test, the specific *Value Output* (**`V`**) command, its commanded output voltage, and its corresponding Low Limit and High Limit, are as follows:

• **`V1`** = Output +1 V	+0.9980 ≤ Reading ≤ +1.0020 Volts
• **`V0.5`** = Output +0.5 V	+0.4985 ≤ Reading ≤ +0.5015 Volts
• **`V0.1`** = Output +0.1 V	+0.0989 ≤ Reading ≤ +0.1011 Volts
• **`V0`** = Output 0 V	-0.0010 ≤ Reading ≤ +0.0010 Volts
• **`V-0.1`** = Output –0.1 V	-0.1011 ≤ Reading ≤ -0.0989 Volts
• **`V-0.5`** = Output –0.5 V	-0.5015 ≤ Reading ≤ -0.4985 Volts
• **`V-1`** = Output -1 V	-1.0020 ≤ Reading ≤ -0.9980 Volts

Data Transfer Speeds

The DAC488 has been measured to perform at the approximate speeds listed below. All explanations discuss only one port, although the information given is valid for any port. These values are relative to the controller used. The speeds you encounter may vary.

- ***Output of Data to the Analog Ports:*** When sending data to the analog outputs on a port of the DAC488, the outputs may be changed about once every 4 milliseconds when specifying voltages in bits, or once every 25 milliseconds when specifying voltages in volts.

- ***Output of Data to the Digital Output Port:*** Data may be sent to the digital output port at an approximate rate of once every 2.5 milliseconds.

- ***Input of Data from the Digital Input Port:*** The digital input port may be read at an approximate rate of once every 75 milliseconds.

DAC488 Operation 4

Introduction

The DAC488 provides multiple independently programmable 12-bit plus-sign Digital-to-Analog Converters (DACs). Three voltage ranges, ±1, ±5, and ±10, are available on a per port basis as well as an autorange feature which automatically selects the range giving the best resolution for the specified voltage. The DAC ports may be programmed in terms volts or bits. The resolution per bit on each range is: 250 µV/bit (±1 V range), 1.25 mV/bit (±5 V range), and 2.5 mV/bit (±10 V range). Volts are programmed with decimal floating point numbers in the range of ±10.2375 or ±1.2375E+1. Voltages may also be programmed in decimal bits (range = ±4095) or 16-bit hexadecimal two's complement numbers (range = F001 to 0FFF).

Command Groups

Operation of the DAC488 is accomplished using a set of commands that configure the entire interface as well as each DAC port. The DAC488 commands can be divided into two main groups: *System commands* and *port commands.*

System & Port Commands

System commands affect the operation of the entire interface and are not specific to a given port. Meanwhile, port commands only affect the operation of the selected port. The port is selected using the *Port Select* (`Pn`) command. The 15 system commands and the 11 port commands are listed below:

System Commands		Port Commands	
Syntax	**Name**	**Syntax**	**Name**
`@`	Command Trigger	`A`	Autorange
`D`	Digital Output	`B`	Buffer Data
`E?`	Query Error Status	`C`	Control Mode
`G`	GET Trigger Mask	`F`	Buffer Definition
`K`	End-Or-Identify Control	`H`	Offset Calibration
`M`	Service Request Mask	`I`	Interval
`O`	Output Format	`J`	Gain Calibration
`P`	Port Select	`L`	Buffer Location
`Q`	External Trigger Mask	`N`	Number of Cycles
`S`	System Defaults	`R`	Range Select
`T`	Command Trigger Mask	`V`	Value Output
`U`	User Status		
`W`	Test		
`X`	Execute		
`Y`	Bus Terminator		

Command Support

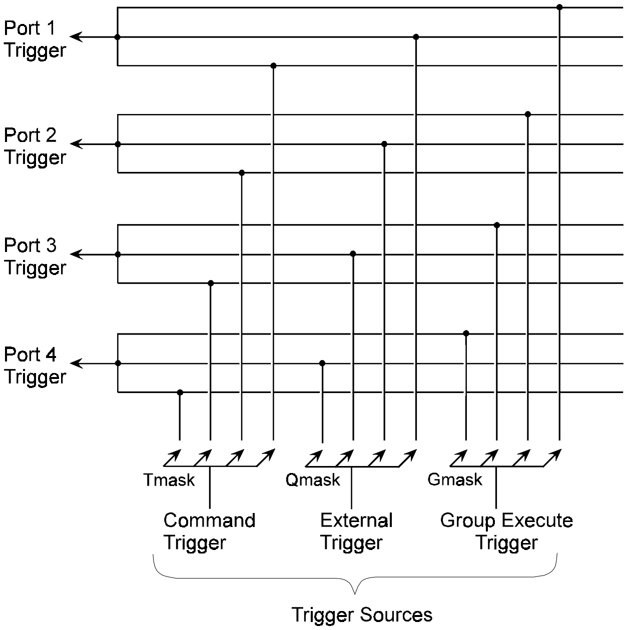
Support for the above commands, includes the *Query* (**?**) command extension, and the use of the Serial Poll Status Byte. Refer to Chapter 7 for more details.

Port Triggering

Triggering is the process of changing a DAC output at the occurrence of a trigger event. The three trigger events which can trigger a DAC port are: a *Trigger On* (@) command, an IEEE Group Execute Trigger (**GET**) or an external trigger pulse applied to the external trigger/SRQ input. Any DAC port may be configured to trigger on one or more of these trigger sources. These trigger sources and their relationship to a DAC port are shown in the following diagram.

As shown in the diagram, the trigger sources are routed to the ports by using the trigger mask commands. The trigger mask commands are illustrated as switches. The trigger mask commands are used to enable a trigger source to trigger the selected DAC port. All trigger signals are then combined to allow a port to be triggered upon the occurrence of any of the three trigger events.

Triggering is provided as a means of synchronizing multiple DAC ports. Voltages may be output at two or more ports simultaneously by programming each port for the desired voltages and trigger event(s). When the trigger event occurs, the programmed voltages will appear at the ports simultaneously.



DAC488/4 Trigger Structure

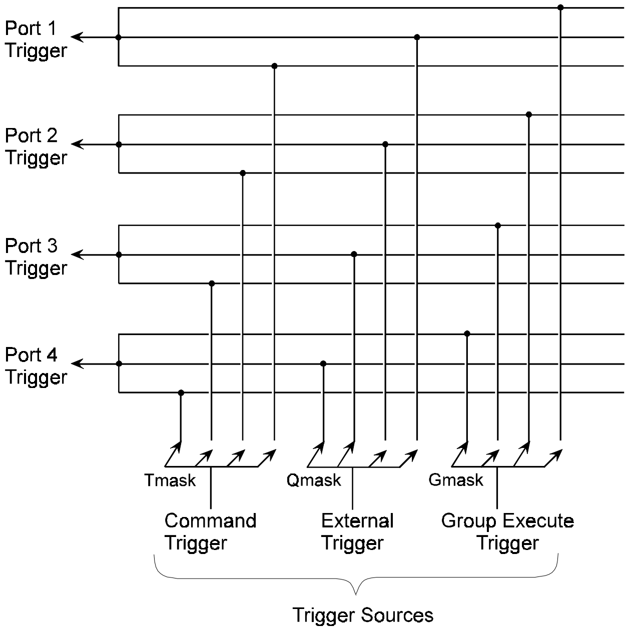
For example, a typical configuration may require Port 1 to have all trigger sources disabled, Port 2 configured to trigger on an IEEE Group Execute Trigger, Port 3 to trigger on an external trigger, and Port 4 to trigger on any of the three trigger sources.

The commands to configure the ports for this configuration are shown below:

```
PRINT#1,"OUTPUT09;G0 Q0 T0 X"
PRINT#1,"OUTPUT09;G2 X"
PRINT#1,"OUTPUT09;Q4 X"
PRINT#1,"OUTPUT09;G8 Q8 T8 X"
```

- ***Line 1:*** **`G0`** disables all ports from triggering on an IEEE Group Execute Trigger, **`Q0`** disables all ports from triggering on an external trigger, **`T0`** disables all ports from triggering on a command trigger (@), and **`X`** executes the commands.
- ***Line 2:*** **`G2`** enables port 2 for triggering on an IEEE Group Execute Trigger and **`X`** executes the command.
- ***Line 3:*** **`Q4`** enables port 3 for triggering on an external trigger and **`X`** executes the command.
- ***Line 4:*** **`G8`** enables port 4 for triggering on an IEEE Group Execute Trigger, **`Q8`** enables port 4 for triggering on an external trigger, **`T8`** enables port 4 for triggering on an a command trigger (@), and **`X`** executes the commands.

After these commands are executed, the trigger sources will be routed to the ports as illustrated in the figure.



DAC488/4 Trigger Structure - Typical Configuration

Trigger Processing & Trigger Overrun

Operation of the DAC488 is controlled by an internal 1 millisecond timer. Each DAC port is updated at a maximum rate of once every millisecond. When a trigger event occurs, the DAC will output the programmed voltage within 1 millisecond of receiving the trigger. If a second trigger event occurs before the DAC has finished processing the first trigger, the second trigger will be held pending and a trigger overrun error will occur. The error condition may be cleared by issuing the Error Query command (E?) and then reading the response from the DAC488. After the first trigger is processed, the pending trigger will then be processed. Any additional triggers which may occur while a trigger is held pending will be ignored.

Internal Buffer

Buffer Data

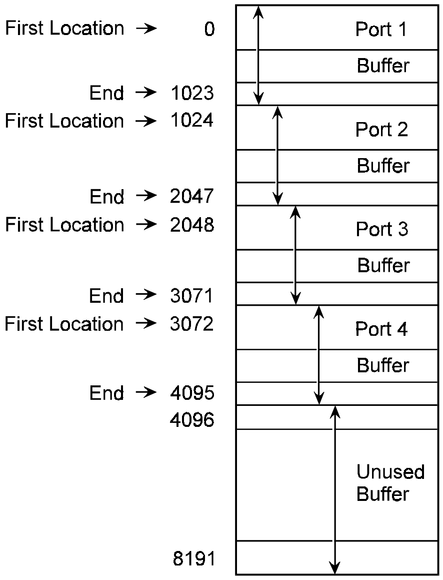
The DAC488 contains an internal buffer consisting of 8,192 locations numbered 0 to 8,191. The buffer is shared by all ports. Each port may be given a different section of the buffer or ports may use the same buffer locations without conflict.

This buffer may be loaded with voltage values to be output when the Stepped or Waveform modes are used. All data in the internal buffer is saved in Non-Volatile RAM. Therefore buffer data which was previously loaded will be available at power on. Examples showing the use of the internal buffer are given in the upcoming descriptions of Stepped and Waveform Control modes.

The following diagram shows the factory default allocation of the internal buffer to each of the DAC ports on the DAC488/4. On the DAC488/2, only the Port 1 and Port 2 sections of the internal buffer are allocated.

Buffers are defined using the *Buffer Definition* (**`F`**) command. The parameters of this command specify the starting location (**`start`**) and the number of locations (**`size`**) in the defined buffer. Once the buffer is defined it may be loaded with voltage values using the *Buffer Data* (**`B`**) command.

Before using the Buffer Data command, the *Buffer Location* (**`L`**) command should be used to set the location pointer to the desired value. The location pointer points to the area where data will be stored when the Buffer Data command is used. After each use of the Buffer Data command, the location pointer is incremented.



DAC488 Internal Buffer - Default Allocation

Buffer Save-and-Restore Program

This BASIC program **DACBUFR.BAS** may be used to save the contents of the DAC488 internal buffer to disk. It can also be used to load the saved values from disk into the internal buffer.

```
10      'BASIC program to save and restore the DAC488 buffer contents
20      '
30      'Initialize the IOtech IEEE488 Driver
40 OPEN "\dev\ieeeout" FOR OUTPUT AS #1
50 IOCTL #1, "BREAK"
60 PRINT #1, "RESET"
70 OPEN "\dev\ieeein" FOR INPUT AS #2
75 ON TIMER(5) GOSUB 2000
80 CLS
90 PRINT "DAC488/4 Waveform Buffer Save and Restore Program"
100 PRINT : PRINT " 1 - Save the Waveform Buffer "
110 PRINT " 2 - Restore the Waveform Buffer "
120 PRINT " 3 - Exit the program"
130 PRINT : INPUT "Select the desired function "; FUN
140 IF FUN < 1 OR FUN > 3 THEN PRINT "Invalid selection!": GOTO 100
150 IF FUN = 3 THEN END
160 IF FUN = 1 THEN GOSUB 500 ELSE GOSUB 1000
170 GOTO 80
500     ' Save DAC488/4 waveform buffer
510     '
515 WCNT = 0: TIMER ON
520 OPEN "buffer.sav" FOR OUTPUT AS #3
530 PRINT #1, "output 09;p1c0l0x"
540 PRINT "Saving the DAC488/4 waveform buffer"
550 FOR LOP = 1 TO 2048
560 PRINT #1, "output 09;b?b?b?b?"
570 PRINT #1, "enter 09"
580 LINE INPUT #2, SS$
590 PRINT #3, SS$
600 NEXT LOP
610 CLOSE #3
620 TIMER OFF
630 RETURN
1000    ' Restore DAC488/4 waveform buffer
1010    '
1020 OPEN "buffer.sav" FOR INPUT AS #3
1030 PRINT #1, "output 09;p1c0l0x"
1040 PRINT "Restoring the DAC488/4 waveform buffer"
1045 WCNT = 0: TIMER ON
1050 FOR LOP = 1 TO 2048
1060 LINE INPUT #3, SS$
1070 RS$ = MID$(SS$, 1, 12) + "x" + MID$(SS$, 13, 12) + "x" + MID$(SS$, 25, 12) + "x" +
MID$(SS$, 37, 12) + "x"
1080 PRINT #1, "output 09;"; RS$
1090 NEXT LOP
1100 CLOSE #3
1110 TIMER OFF
1120 RETURN
2000 PRINT "Working - "; : WCNT = WCNT + 1
2010 IF WCNT > 6 THEN PRINT : WCNT = 0
2020 RETURN
```

Control Modes

Four modes of DAC port operation are available: Direct, Indirect, Stepped, and Waveform. Each port is independent and may be operated in a different mode. The modes of operation are described below.

Direct Control Mode

Each of the DAC ports may be directly controlled from the IEEE 488 bus. In the Direct Control mode, a DAC voltage is output upon receipt of the *Execute* (**`X`**) command. Direct control is accomplished by selecting the DAC port, the range or autorange, specifying the DAC output voltage, and issuing the Execute command.

One use of the Direct mode is to output voltages at one or more DAC ports directly under program control from an IEEE controller. To output 4 volts on port 1 the following command string could be used:

```
PRINT#1,"OUTPUT09;C0 P1 A0 R2 V4 X"
```

In this example, **`C0`** selects Direct Control mode, **`P1`** selects port 1, **`A0`** disables Autoranging, **`R2`** selects the ±5 Volt range and **`V4`** is the voltage to be output (4 volts). This command string causes the DAC488 to output 4 volts at port 1.

Indirect Control Mode

Indirect control implies that the DAC output will change only when a trigger event occurs. Indirect control is accomplished by selecting the DAC port, the range or autorange, specifying the DAC output voltage and the desired trigger source. When the trigger event occurs, the programmed voltage will be output.

One use of the Indirect mode is to output voltages at one or more DAC ports upon the occurrence of a trigger event. The command string shown below causes the DAC488 to output 7.5 volts at Port 2 upon receipt of an external trigger signal.

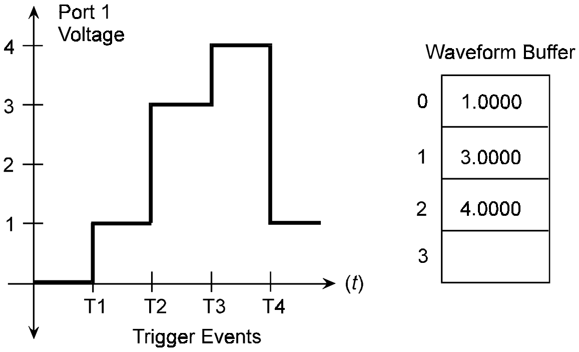
```
PRINT#1,"OUTPUT09;C1 P1 Q1 A0 R2 V4 X"
```

In this example, **`C1`** selects the Indirect control mode, **`P1`** selects port 1, **`Q1`** enables port 1 to trigger on the positive edge of an external trigger sense pulse, **`A0`** disables autoranging, **`R2`** selects the ±5 Volt range and **`V4`** is the voltage to be output (4 volts).

Stepped Control Mode

The Stepped control mode may be used to control the DAC in a stepped manner. In this mode, a sequence of DAC voltages are loaded into the DAC488 internal buffer. After the voltages are loaded into the buffer, the DAC output can then be stepped through each of the values by using any of the three trigger sources. When the last voltage in the buffer is output, the DAC488 will automatically return to the first location defined by the *Buffer Definition* (**`F`**) command thereby allowing the sequence to be repeated.

One application of Stepped mode may be to use the DAC488 to output a series of test voltages which may control another device. The example shows how the DAC488 would be configured to output the next voltage in a sequence of voltages each time an external trigger is received.



Waveform Buffer	
0	1.0000
1	3.0000
2	4.0000
3	

Example of Stepped Mode

In this example, the voltage at port 1 is stepped to the next value in the buffer each time an external trigger is received. Each T on the graph represents a trigger event. The voltage sequence is 1 volt, 3 volts, 4 volts.

The commands required to load the internal buffer for this example are:

```
PRINT#1,"OUTPUT09;C2 P1 F0,3 L0 Q1 X"
PRINT#1,"OUTPUT09;B1,1X  B2,3X  B2,4X"
PRINT#1,"OUTPUT09;L0 X"
```

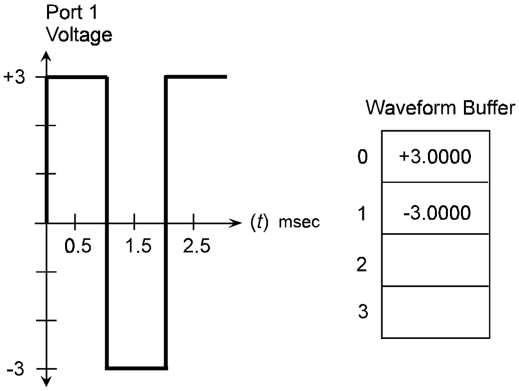
- ***Line 1:*** **`C2`** selects Stepped Control mode, **`P1`** selects port 1, **`F0,3`** defines a buffer with the first location as location 0 and the number of values in the sequence to 3, **`L0`** sets the location pointer back to 0 (first location) and **`Q1`** enables port 1 to trigger on an external trigger.
- ***Line 2:*** **`B1,1X`** selects the ±1 volt range, 1 volt for first point, **`B2,3X`** selects the ±5 volt range, 3 volts for second point and **`B2,4X`** selects the ±5 volt range, 4 volts for third point.
- ***Line 3:*** **`L0`** sets the location pointer back to location 0 before the port is triggered. If this is not done, the location pointer will start at location 3, not location 0.

The output of the sequence is controlled by the rate at which external triggers occur. In the graph previously shown, each T represents an external trigger event.

Waveform Control Mode

The Waveform control mode may be used to control the DAC in an automatic manner. A buffer is defined using the *Buffer Definition* (**`F`**) command and voltage values are loaded into the buffer using the *Buffer Location* (**`L`**) command and the *Buffer Data* (**`B`**) command. Once the DAC488 is triggered, these preloaded voltage values are then output at a regular interval. The time interval between the output of each value is set by using the *Interval* (**`I`**) command. The number of cycles of the waveform is set by using the *Number of Cycles* (**`N`**) command.

One application of waveform mode may be to use the DAC488 as a precise function generator. The waveform example shows how the DAC488 would be configured to output a 6 volt peak to peak 500 Hz square wave centered at zero.



Example of Waveform Mode

More complex waveforms may be generated by loading the buffer with values computed using a high-level language program. Sample programs are included with the DAC488 which will load the buffer with voltage values to generate a Sine, Triangle, Ramp and other waveforms.

The commands required to load the internal buffer for this example are:

```
PRINT#1,"OUTPUT09;C3 P1 F0,2 G1 L0 I1 X"
PRINT#1,"OUTPUT09;B2,3X  B2,-3X N0 X"
PRINT#1,"OUTPUT09;L0 X"
PRINT#1,"OUTPUT09;TRIGGER"
```

- ***Line 1:*** **`C3`** selects Waveform control mode, **`P1`** selects port 1, **`F0,2`** defines a buffer with the first location as location 0 and the number of points in the sequence to 2, **`G1`** selects trigger on **`GET`**, **`L0`** sets the location pointer to 0 (the first location) and **`I1`** sets the time interval between points to 1 millisecond.
- ***Line 2:*** **`B2,3X`** selects the ±5 volt range, 3 volts for first point and **`B2,-3X`** selects the ±5 volt range, -3 volts for second point, **`N0`** sets the number of cycles to continuous.
- ***Line 3:*** **`L0`** sets the buffer location pointer back to location 0 before the port is triggered. If this is not done, the location pointer will start at location 2, not location 0.
- ***Line 4:*** When the IEEE Group Execute Trigger is executed, the DAC488 will output the waveform at port 1 continuously.

Multiple Port Synchronization

Multiple DAC port output changes may be synchronized in any of the triggered modes of operation (Indirect, Stepped and Waveform). This is done by using the trigger mask commands to enable the ports to trigger on the same trigger event(s).

Synchronization in Waveform mode is automatically accomplished by programming multiple ports with the same time interval using the *Interval* (**`I`**) command and the same buffer size using the *Buffer Definition* (**`F`**) command. The BASIC program **`DAC4WAVE.BAS`** included on the program disk shows how multiple, synchronized waveforms may be generated. This program is included below.

Synchronized Waveforms Program

The following BASIC program **`DAC4WAVE.BAS`** is used with the DAC488 to generate a waveform at a user selectable frequency and 20 volts peak-to-peak centered at 0 volts:

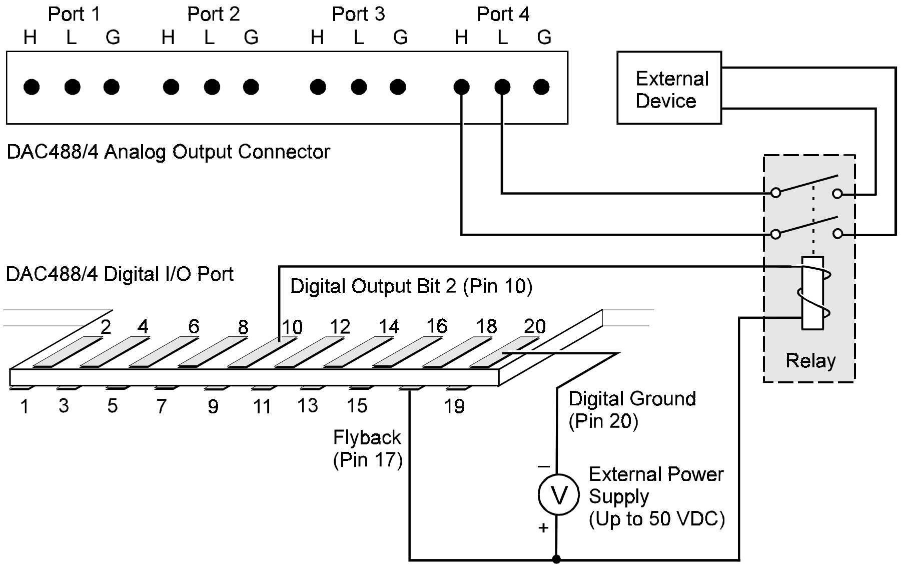
```
10 REM Basic program to load the DAC488/4 buffer w multiple waveforms
20 REM Initialize the IOtech IEEE488 Driver
30 CLS : KEY OFF
40 OPEN "\dev\ieeeout" FOR OUTPUT AS #1
50 IOCTL #1, "BREAK"
60 PRINT #1, "RESET"
70 OPEN "\dev\ieeein" FOR INPUT AS #2
80 PRINT "DAC488 Waveform Demonstration Program"
90 PRINT
100 PRINT"Set address of DAC488 to 09 and connect to controller"
110 PRINT
120 PRINT"Hit any key to start"
130 IF INKEY$="" THEN GOTO 130
140 PRINT:PRINT #1, "CLEAR 09"
150 PRINT #1, "output 09;P3XE?" 'Determine if DAC488/4 or DAC488/2
160 PRINT #1, "ENTER 09"
170 INPUT#2,A$
180 IF A$ = "E0" THEN DAC2$ = "N" ELSE DAC2$ = "Y"
190 IF DAC2$ = "Y" THEN MODEL$ = "2" ELSE MODEL$ = "4"
200 PRINT "Initializing the DAC488/";MODEL$;" this will take a while"
210 REM First a 256 point sinewave
220 PRINT #1, "output 09;c3l0x"
230 POINTS = 256: REM number of points in the sine wave
240 PI = 3.1415926535#
250 INC = (2! * PI) / POINTS
260 FOR LOOP = 1 TO POINTS
270 PVAL = 4095*SIN(LOOP * INC)
280 PRINT #1, "output 09;b3,#";CINT(PVAL);"x":NEXT
290 REM Now a triangle wave
300 INC = 4096 / 64
310 FOR LOOP = 0 TO 4095 STEP INC
320 PRINT #1,"output 09;b3,#";LOOP;"x":NEXT
330 FOR LOOP = 4095 TO -4095 STEP -INC
340 PRINT #1,"output 09;b3,#";LOOP;"x":NEXT
350 FOR LOOP = -4095 TO 0 STEP INC
360 PRINT #1,"output 09;b3,#";LOOP;"x":NEXT
370 REM Now a square wave
380 PRINT #1,"output 09;b3,#4095xb3,#-4095x"
390 REM Main Loop
400 CLS
```

```
410 IF MODEL$ = "2" THEN GOTO 420 ELSE GOTO 440
420 INPUT" Select a port 1,2  ";PORT
430 GOTO 450
440 INPUT" Select a port 1-4  ";PORT
450 IF PORT < 1 OR PORT > 4 GOTO 410
460 IF PORT > 2 AND DAC2$ ="Y" THEN GOTO 410
470 PRINT
480 INPUT"Select an interval in milliseconds (1-255) ";INTRVL
490 IF (INTRVL <= 0) OR (INTRVL > 255) GOTO 480
500 PRINT" Select a waveform (1-6) ";WAVE
510 PRINT" 1 - Sinewave "
520 PRINT" 2 - Rectified Sine Wave"
530 PRINT" 3 - Triangle "
540 PRINT" 4 - Ramp "
550 PRINT" 5 - Inverted Ramp "
560 PRINT" 6 - Square Wave "
570 INPUT WAVE
580 IF WAVE < 1 OR WAVE > 6 GOTO 500
590 IF WAVE = 1 GOTO 660
600 IF WAVE = 2 GOTO 700
610 IF WAVE = 3 GOTO 740
620 IF WAVE = 4 GOTO 780
630 IF WAVE = 5 GOTO 820
640 IF WAVE = 6 GOTO 860
650 PRINT: PRINT: GOTO 410
660 REM sinewave output routine
670 PRINT #1,"output 09;p";PORT"c3f0,256l0i";INTRVL;"n0t0x"
680 PRINT #1,"output 09;t"; (2^(PORT-1));"x@"
690 GOTO 650
700 REM rectified sinewave output routine
710 PRINT #1,"output 09;p";PORT"c3f0,128l0i";INTRVL;"n0t0x"
720 PRINT #1,"output 09;t"; (2^(PORT-1));"x@"
730 GOTO 650
740 REM triangle wave output routine
750 PRINT #1,"output 09;p";PORT"c3f256,256l256i";INTRVL;"n0t0x"
760 PRINT #1,"output 09;t"; (2^(PORT-1));"x@"
770 GOTO 650
780 REM ramp wave output routine
790 PRINT #1,"output 09;p";PORT"c3f256,64l256i";INTRVL;"n0t0x"
800 PRINT #1,"output 09;t"; (2^(PORT-1));"x@"
810 GOTO 650
820 REM inverted ramp wave output routine
830 PRINT #1,"output 09;p";PORT"c3f384,64l384i";INTRVL;"n0t0x"
840 PRINT #1,"output 09;t"; (2^(PORT-1));"x@"
850 GOTO 650
860 REM square wave output routine
870 PRINT #1,"output 09;p";PORT"c3f512,2l512i";INTRVL;"n0t0x"
880 PRINT #1,"output 09;t"; (2^(PORT-1));"x@"
890 GOTO 650
```

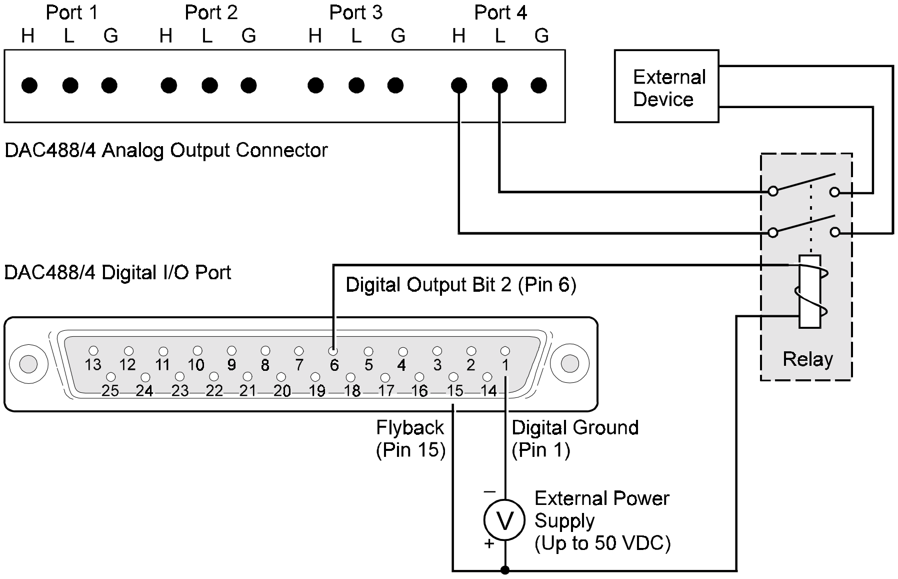
Digital I/O Port

The digital I/O port on the DAC488 may be used to read eight TTL level inputs and control eight TTL level outputs. The digital output port may also be configured for high voltage/high current operation. Once the output port is configured for high level operation, the output can control relays and other high level devices.

One use of the high level outputs may be to energize a relay under program control. The following Standard and CE diagrams each show how a digital output line can control a relay which in turn, connects the DAC Port 4 output voltage to an external volt controlled device. The *Digital Output* (**D**) command would then be used to control the relay. Sending the command **D0X** will energize the relay and a **D8X** will deenergize the relay.



Relay Control Using Digital Output
(Standard Version)



Relay Control Using Digital Output
(CE Version)

User-Defined System Defaults

The *System Defaults* (**`S`**) command is used to save the Calibration Constants in the Non-Volatile RAM. It may also be used to save a particular configuration of the DAC488 as a power on default configuration. All system and port commands may be used to configure the DAC488 and then this particular configuration saved as the power on default configuration.

An application may require the DAC488 to power on in the following configuration:

- Autorange disabled
- Waveform Control Mode
- Interval = 1 millisecond
- Port 2 selected
- Range = ±10 volts
- Trigger on Group Execute Trigger

The commands to configure the DAC488 in the above configuration and then to save this configuration as the power on default would be as follows:

```
PRINT#1,"OUTPUT09;A0 C3 I1 P2 R3 G2 X"
PRINT#1,"OUTPUT09;S1 X"
```

- ***Line 1:*** **`A0`** disables autoranging, **`C3`** selects Waveform control mode, **`I1`** sets the interval to 1 millisecond, **`P2`** selects port 2, **`R3`** sets the range to ±10 volts, **`G2`** enables port 2 to trigger on **`GET`**.
- ***Line 2:*** **`S1`** saves this configuration as the power on default configuration.

IEEE 488 Bus Support

The DAC488 implements many of the capabilities defined by the IEEE 488 1978 specification. These capabilities are discussed in the following sections. However, the *bus lines* (uniline) and *bus commands* (multiline) that the DAC488 does *not* support, or respond to, include the following:

REN	Remote Enable (uniline)	**PPD**	Parallel Poll Disable
GTL	Go to Local	**PPU**	Parallel Poll Unconfigure
LLO	Local Lock Out	**TCT**	Take Control
PPC	Parallel Poll Configure		

Bus Lines

The following IEEE 488 bus management lines are supported by DAC488:

- ***Interface Clear* (`IFC`):** **`IFC`** places the DAC488 in the Talker/Listener Idle State.
- ***Service Request* (`SRQ`):** Whenever the DAC488 generates a Service Request (**`SRQ`**), a Serial Poll will return a Serial Poll Status byte of at least 64 (decimal) showing that the **`SRQ`** was generated by the DAC488. The following BASIC program illustrates one approach for handling service requests:

```
PRINT#1,"SPOLL09"            Serial poll the DAC488
INPUT#2,SRQ09
IF SRQ09 < 64 THEN GOTO 500  SRQs disabled
IF SRQ09 = 65 THEN GOTO 600  DAC port 1 ready routine
IF SRQ09 = 66 THEN GOTO 700  DAC port 2 ready routine
IF SRQ09 = 68 THEN GOTO 800  DAC port 3 ready routine
```

Note: To allow the DAC488 to detect each of the above conditions, the *Service Request Mask* (**`M`**) command must be enabled for each condition. More extensive tests of the *Serial Poll Status Byte* are needed if multiple **`SRQ`** conditions are selected. For more details, refer to the Service Request Mask (**`M`**) command in Chapter 7.

Bus Commands

The following IEEE 488 bus commands are supported by DAC488:

- ***Device Clear* (`DCL`):** Device Clear resets all devices, including the DAC488, to its power on state. The DAC488 will be in the Listener Idle and Talker Idle state. The power on default settings for each command will be programmed.
- ***Group Execute Trigger* (`GET`):** When the DAC488 recognizes a **`GET`**, and the *GET Trigger Mask* (**`G`**) command has been used to arm the DAC488, it outputs the programmed voltage or voltage sequence (depending on which mode the DAC488 is in) at the selected port(s).
- ***My Listen Address* (`MLA`):** When the DAC488 is addressed to Listen, it accepts characters from the Active Talker and interprets these characters as commands and command parameters.
- ***My Talk Address* (`MTA`):** When the DAC488 is addressed to Talk, it returns a string specifying the selected port along with the selected range and output voltage which was programmed for that port. The DAC488 can also return a status string for each port or the entire interface by using the *User Status* (**`U`**) command. Information about commands previously executed or data stored in the internal buffer may be returned by using the *Query* (**`?`**) command.
- ***Selected Device Clear* (`SDC`):** Selected Device Clear resets a single device, in this case the DAC488, to its power on state. The DAC488 will be in the Listener Idle and Talker Idle state. The power on default settings for each command will be programmed.
- ***Serial Poll Disable* (`SPD`):** Disables the DAC488 from responding to Serial Polls by the controller.
- ***Serial Poll Enable* (`SPE`):** When Serial Poll Enabled, the DAC488 sets itself to respond to a Serial Poll with its Serial Poll Status byte if addressed to Talk. When the Serial Poll byte is accepted by the controller, any pending **`SRQ`**s are cleared. The DAC488 will continue to try to output its Serial Poll response until it is Serial Poll Disabled by the controller.
- ***Unlisten* (`UNL`): `UNL`** places the DAC488 in the Listener Idle State.
- ***Untalk* (`UNT`): `UNT`** places the DAC488 in the Talker Idle State.

Bus Addresses

The IEEE 488 interface standard permits up to 15 devices to be configured within one system. Each of these devices must have a unique address to avoid confusion. In a similar fashion, every building in town has a unique address to prevent one home from receiving another home's mail. Exactly how each device's address is set is specific to the product's manufacturer. Some are set by DIP switches in hardware, others by software. Consult the manufacturer's instructions to determine how to set the address.

Bus addresses are sent with bus (multiline) commands from the Active Controller. These commands include My Listen Address (**`MLA`**), My Talk Address (**`MTA`**), Talk Address Group (**`TAG`**), and Listen Address Group (**`LAG`**).

Bus Terminators

The terminating characters sent on output by the DAC488 are set by using the *Bus Terminator* (**`Y`**) command. Upon power on, the unit's IEEE 488 bus terminators are determined by the power-on configuration saved using the *System Defaults* (**`S`**) command. The factory default terminator selection is **`CR LF`** (carriage-return line-feed). The power on default terminators can be changed to any of the four available options listed under the *Bus Terminator* (**`Y`**) command, described in Chapter 7. The four available commands and the associated terminator options with them, are: (1) **`Y0 = CR LF`**, (2) **`Y1 = LF CR`**, (3) **`Y2 = CR`** only, and (4) **`Y3 = LF`** only. If it is desired to have particular power on default terminator(s), the terminator(s) may be selected using the Bus Terminator command and then this selection may be saved using the *System Defaults* (**`S`**) command.

The DAC488 ignores all terminators received from the bus controller. Only the *Execute* (**`X`**) command is used to signal the DAC488 that a command string has been completed. The only exception to this rule is the *Trigger On* (**`@`**) command which is processed immediately upon receipt.

IEEE 488 Background 5

History

The IEEE 488 bus is an instrumentation communication bus adopted by the Institute of Electrical and Electronic Engineers in 1975 and revised in 1978. Prior to the adoption of this standard, most instrumentation manufacturers offered their own versions of computer interfaces. This placed the burden of system hardware design on the end user. If his application required the products of several different manufacturers, then he might need to design several different hardware and software interfaces. The popularity of the IEEE 488 interface (sometimes called the General Purpose Interface Bus or GPIB) is due to the total specification of the electrical and mechanical interface as well as the data transfer and control protocols. The use of the IEEE 488 standard has moved the responsibility of the user from the design of the interface, to the design of the high level software that is specific to the measurement application.

General Bus Structure

The main purpose of the IEEE 488 interface is to transfer information between two or more devices. A device can either be an instrument or a computer. Before any information transfer can take place, it is first necessary to specify which will do the talking (send data) and which devices will be allowed to listen (receive data). The decision of who will Talk and who will Listen usually falls on the System Controller which is, at power on, the Active Controller.

The System Controller is similar to a committee chairman. On a well run committee, only one person may speak at a time and the chairman is responsible for recognizing members and allowing them to have their say. On the bus, the device which is recognized to speak is the Active Talker. There can only be one Talker at a time if the information transferred is to be clearly understood by all. The act of "giving the floor" to that device is called Addressing to Talk. If the committee chairman cannot attend the meeting, or if other matters require his attention, he can appoint an acting chairman to take control of the proceedings. For the IEEE 488 interface, this device becomes the Active Controller.

At a committee meeting, everyone present usually listens. This is not the case with the IEEE 488 interface. The Active Controller selects which devices will Listen and commands all other devices to ignore what is being transmitted. A device is instructed to Listen by being Addressed to Listen. This device is then referred to as an Active Listener. Devices which are to ignore the data message are instructed to Unlisten.

The reason some devices are instructed to Unlisten is quite simple. Suppose a college instructor is presenting the day's lesson. Each student is told to raise their hand if the instructor has exceeded their ability to keep up while taking notes. If a hand is raised, the instructor stops his discussion to allow the slower students the time to catch up. In this way, the instructor is certain that each and every student receives all the information he is trying to present. Since there are a lot of students in the classroom, this exchange of information can be very slow. In fact, the rate of information transfer is no faster than the rate at which the slowest note-taker can keep up. The instructor, though, may have a message for one particular student. The instructor tells the rest of the class to ignore this message (Unlisten) and tells it to that one student at a rate which he can understand. This information transfer can then happen much quicker, because it need not wait for the slowest student.

The IEEE 488 interface transfers information in a similar way, with a method of data transfer called *handshaking*. For data transfer on the IEEE 488 interface, the Active Controller must:

- Unlisten all devices to protect against eavesdroppers.
- Designate a device who will talk by addressing it to Talk.
- Designate all the devices who will listen by addressing them to Listen.
- Indicate to all devices that the data transfer can take place.

Bus Lines & Bus Commands

Bus Management Lines

For the general control and coordination of bus activities, five hardware lines on the IEEE 488 interface are used for bus management. Signals on these lines are often referred to as uniline (single line) commands. The signals are active low (i.e., a low voltage represents an *asserted* logic of "1", and a high voltage represents an *unasserted* logic of "0").

- ***Interface Clear* (IFC):** The **IFC** line is used only by the System Controller. It is used to place all bus devices in a known state. Although device configurations vary, the **IFC** command usually places the devices in the Talk and Listen Idle states (neither Active Talker nor Active Listener).

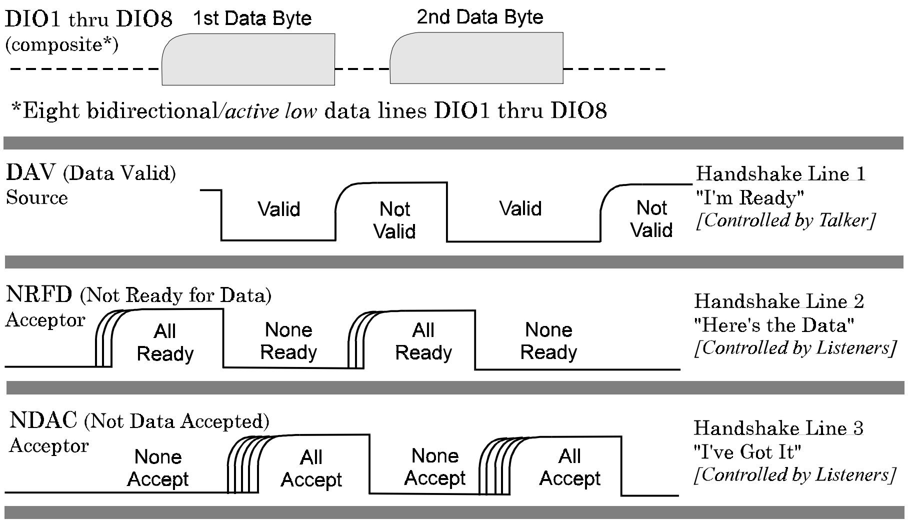
- ***Remote Enable* (REN):** When the System Controller sends the **REN** command, bus devices will respond to remote operation. Generally, the **REN** command should be issued before any bus programming is attempted. Only the System Controller has control of the Remote Enable line.

- ***Attention* (ATN):** The **ATN** line is one of the most important lines for bus management. If Attention is asserted, then the information contained on the data lines is to be interpreted as a multiline command. If it is not, then that information is to be interpreted as data for the Active Listeners. The Active Controller is the only bus device that has control of this line.

- ***End Or Identify* (EOI):** The **EOI** line is used to signal the last byte of a multibyte data transfer. The device that is sending the data asserts **EOI** during the transfer of the last data byte. The **EOI** signal is not always necessary, for the end of the data may be indicated by some special character such as carriage return (**CR**) and/or line feed (**LF**). The Active Controller also uses **EOI** to perform a Parallel Poll by simultaneously asserting **EOI** and **ATN**.

- ***Service Request* (SRQ):** When a device desires the immediate attention of the Active Controller, it asserts **SRQ**. It is then the responsibility of the controller to determine which device requested service. This is accomplished with a Serial Poll or a Parallel Poll.

Handshake Lines



IEEE 488 Bus Handshaking

The IEEE 488 interface uses three handshake lines in an "I'm ready - Here's the data - I've got it" sequence. This handshake protocol assures reliable data transfer, at the rate determined by the slowest Listener. One line is controlled by the Talker, while the other two are shared by all Active Listeners. The handshake lines, like the other IEEE 488 lines, are active low.

- ***Data Valid* (DAV):** The **DAV** line is controlled by the Talker. The Talker verifies that **NDAC** is asserted (active low) which indicates that all Listeners have accepted the previous data byte transferred. The Talker then outputs data on the bus and waits until **NRFD** is unasserted (high) which indicates that all addressed Listeners are ready to accept the information. When **NRFD** and **NDAC** are in the proper state, the Talker asserts **DAV** (active low) to indicate that the data on the bus is valid.
- ***Not Ready for Data* (NRFD):** The **NRFD** line is used by the Listeners to inform the Talker when they are ready to accept new data. The Talker must wait for each Listener to unassert this line (high) which they will do at their own rate when they are ready for more data. This assures that all devices that are to accept the information are ready to receive it.
- ***Not Data Accepted* (NDAC):** The **NDAC** line is also controlled by the Listeners. This line indicates to the Talker that each device addressed to Listen has accepted the information. Each device releases **NDAC** (high) at its own rate, but the **NDAC** will not go high until the slowest Listener has accepted the data byte.

Data Transfer Lines

To transfer bit-parallel/byte-serial information between devices on the bus, the IEEE 488 interface provides eight data lines (**DIO1** through **DIO8**). These data lines are bidirectional and are active low.

Bus Command Groups

Bus (multiline) commands are bytes sent by the Active Controller over the data bus with Attention (**ATN**) asserted. These commands are sent to all devices and are divided into the following 5 groups:

- ***Addressed Command Group* (ACG):** These commands affect only those devices which have previously been addressed to be a Listener. There are 5 bus line addressed commands: **GET**, **GTL**, **PPD**, **SDC**, and **TCT**.
- ***Universal Command Group* (UCG):** These commands cause every instrument on the bus to carry out the bus function specified (if the instrument is capable of it). There are 5 bus line universal commands: **DCL**, **LLO**, **PPU**, **SPD**, and **SPE**.
- ***Listen Address Group* (LAG):** These commands address to Listen specified bus devices. There are 31 (0 to 30) Listen addresses associated with this group. The 3 most significant bits of the data bus are set to **001** while the 5 least significant bits are the address of the device being told to Listen.
- ***Talk Address Group* (TAG):** These commands address to Talk specified bus devices. There are 31 (0 to 30) Talk addresses associated with this group. The 3 most significant bits of the data bus are set to **010** while the 5 least significant bits are the address of the device being told to Talk.
- ***Secondary Command Group* (SCG):** These commands are used to specify a subaddress or subfunction within a given bus device. These are any one of the 32 possible commands (0 to 31) in this group. They must immediately follow a Talk or Listen address.
- Three bus commands not found in the above 5 groups are: **MLA**, **MTA**, and **PPC**.

All of the IEEE 488 bus commands are further described individually, as follows:

- ***Device Clear* (DCL):** This command causes all bus devices to be initialized to a pre-defined or power up state.
- ***Group Execute Trigger* (GET):** This command usually signals a group of devices to begin executing a triggered action. This allows actions of different devices to begin simultaneously.
- ***Go To Local* (GTL):** This command allows the selected devices to be manually controlled.
- ***Local Lock Out* (LLO):** This command prevents manual control of the instrument's functions.
- ***My Listen Address* (MLA):** This command addresses a device to Listen. The device accepts data from the Active Talker and outputs this data through the serial interface. It substitutes the selected serial terminators for the received IEEE 488 bus terminators.
- ***My Talk Address* (MTA):** This command addresses a device to Talk. The device retrieves data from the serial input buffer and outputs it to the IEEE 488 bus. It substitutes the selected IEEE 488 bus terminators for the received serial terminators. The device will continue to output serial input buffer data as long as the IEEE 488 controller allows.
- ***Parallel Poll Configure* (PPC):** This command configures devices capable of performing a Parallel Poll via the data bit they are to assert in response to a Parallel Poll.
- ***Parallel Poll Disable* (PPD):** This command disables the Parallel Poll response of selected devices.
- ***Parallel Poll Unconfigure* (PPU):** This disables all devices from responding to a Parallel Poll.
- ***Selected Device Clear* (SDC):** This command causes a single device to be initialized to a pre-defined or power up state.

- ***Serial Poll Disable* (`SPD`):** The command disables all devices from sending their Serial Poll status byte.
- ***Serial Poll Enable* (`SPE`):** This command, when **`ATN`** is unasserted, will cause a device that is addressed to Talk, to output its Serial Poll status byte.
- ***Take Control* (`TCT`):** This command passes bus control responsibilities from the current Controller to another device which has the ability to control.
- ***Unlisten* (`UNL`):** This command places the device in the Listen Idle state.
- ***Untalk* (`UNT`):** This command places the device in the Talk Idle state.

More On Service Requests

Most of the commands covered, both uniline and multiline, are the responsibility of the Active Controller to send and the bus devices to recognize. Most of these happen routinely by the interface and are totally transparent to the system programmer. Other commands are used directly by the user to provide optimum system control. Of the uniline commands, **`SRQ`** is very important to the test system and the software designer has easy access to this line by most devices. Service Request is the method by which a bus device can signal to the Controller that an event has occurred. It is similar to an interrupt in a microprocessor based system.

Most intelligent bus peripherals have the ability to assert **`SRQ`**. A digital multimeter (DMM) might assert it when its measurement is complete, if its input is overloaded or for any of an assortment of reasons. A power supply might **`SRQ`** if its output has current limited. This is a powerful bus feature that removes the burden from the System Controller to periodically inquire, "Are you done yet?". Instead, the Controller says, "Do what I told you to do and let me know when you're done" or "Tell me when something is wrong."

Since **`SRQ`** is a single line command, there is no way for the Controller to determine which device requested the service without additional information. This information is provided by the multiline commands for Serial Poll and Parallel Poll.

- ***Serial Poll:*** Suppose the Controller receives a service request. For this example, let us assume there are several devices which could assert **`SRQ`**. The Controller issues a Serial Poll Enable (**`SPE`**) command to each device sequentially. If any device responds with **`DIO7`** asserted, it indicates to the Controller that it was the device that asserted **`SRQ`**. Often times the other bits will indicate why the device wanted service. This Serial Polling sequence, and any resulting action, is under control of the software designer.
- ***Parallel Poll:*** The Parallel Poll is another way the Controller can determine which device requested service. It provides the *who*, but not necessarily the *why*. When bus devices are configured for Parallel Poll, they are assigned one bit on the data bus for their response. By using the Status bit, the logic level of the response can be programmed to allow logical **`OR/AND`** conditions on one data line by more than one device. When **`SRQ`** is asserted, the Controller (under the user's software) conducts a Parallel Poll. The Controller must then analyze the eight bits of data received to determine the source of the request. Once the source is determined, a Serial Poll might be used to determine the why.

Of the two polling types, the Serial Poll is the most popular due to its ability to determine the who and why. In addition, most devices support Serial Poll only.

Notes

DAC488 Calibration 6

Introduction

Each port of the DAC488 should be calibrated every 12 months or if the performance verification report (described later in this chapter) shows that the DAC488 is not operating within the specifications. Calibration of the DAC488 involves computing the correct Calibration Constants for the offset and gain for each range and polarity. These constants are then stored in the DAC488 Non-Volatile RAM for subsequent use. When a DAC port is commanded to output a voltage, the Calibration Constants are sent to internal D/A converters which set the offset and gain for the given range and polarity. The DAC488 may be calibrated by the following two methods:

1. Calibration may be performed by a host controller using the BASIC calibration program included with the DAC488. The program will then print a calibration report when finished.
2. Calibration may be performed automatically by the DAC488 acting as a bus controller connected to a Keithley 199 DMM. A calibration report will then be printed if the DAC488 is connected to an IEEE printer.

The calibration enable switch on the rear panel must be in the closed (depressed) position to allow the new Calibration Constants to be saved in the Non-Volatile RAM. If the switch is not in this position, these constants may be changed but they will not be saved. If the DAC488 is then turned off, the Calibration Constants will revert to their last saved values. After calibration is completed, the calibration switch should be set to the open position, (switch cap flush with the read panel) to prevent accidental changing of these constants.

Calibration Procedures

The calibration procedures used to compute the Gain and Offset Calibration Constants, as explained below, are the same as those used in the BASIC and in the automatic calibration programs. The example shows how to compute the positive gain, negative gain, and offset calibration constants for the ±10 volt range of Port 1. The equipment needed to calibrate the DAC488 manually are as follows:

- A Voltmeter with a full scale error of 0.01% or better.
- An IEEE controller or a PC with an IEEE interface.
- An IEEE 488 cable.
- Test Leads.
- The analog output connector included with the DAC488.

Commands that must be sent are listed after each step. If you are using the Personal488 interface, you can send commands to the DAC488 directly by using the Keyboard Controller program listed in the “Power-Up & Performance Tests” section in Chapter 3. For example, to send the command **`W1X`** (This will turn on the TEST LED on the DAC488 front panel), you would type **`OUTPUT 09;W1X`** after the **`CMD>`** prompt, then press the **`<Enter>`** key.

Manual Calibration Procedure

Part I: Connections & Preparations

1. Connect the DAC488 to the IEEE controller using an IEEE 488 cable.
2. Connect the negative input of the voltmeter to the Port 1 L (Low) output.
3. Connect the positive input of the voltmeter to the Port 1 H (High) output.
4. Set the IEEE bus address of the DAC488 to 9.
5. Turn the DAC488 on, and allow it to warm up for *at least one hour* before beginning the calibration procedure.

Part II: Acquiring the Gain Calibration Constants

1. After the DAC488 has warmed up for *at least one hour*, initialize the DAC488 by sending an IEEE 488 Interface Clear (**`IFC`**). Command sent to the DAC488:

   ```
   CLEAR 09
   ```

2. Select Port 1, Direct Control Mode, Autoranging disabled, select the +10 volt range, set the Offset to 0, set the Positive and Negative Gain constants to 128, program the port to output 0 volts. Command sent to the DAC488:

   ```
   OUTPUT 09; P1 C0 A0 R3 H0 J128,128 V0 X
   ```

3. Measure and record the port 1 output voltage. Call this voltage **`ZVAL`**.
4. Program the DAC488 port 1 to output 10 volts. Command sent to the DAC488:

   ```
   OUTPUT 09; V10 X
   ```

5. Measure and record the port 1 output voltage. Call this voltage **`PVAL`**.
6. Program the DAC488 port 1 to output -10 volts. Command sent to the DAC488:

   ```
   OUTPUT 09; V-10 X
   ```

7. Measure and record the port 1 output voltage. Call this voltage **`NVAL`**.
8. Set the Positive and Negative Gain constants to maximum (255) and program the port to output 10 volts. Command sent to the DAC488:

   ```
   OUTPUT 09; J255,255 V10 X
   ```

9. Measure and record the port 1 output voltage. Call this voltage **`MVAL`**.
10. Set the Positive and Negative Gain constants to minimum (0) and program the port to output 10 volts. Command sent to the DAC488:

    ```
    OUTPUT 09; J0,0 V10 X
    ```

11. Measure and record the port 1 output voltage. Call this voltage **`NOVAL`**.
12. The *Gain Calibration Constants* are computed using the following formulas:
    - Gain adjustment per bit = (**`MVAL`** - **`NOVAL`**) / 256
    - Pos. Error (in volts) = **`PVAL`** - **`ZVAL`** - 10
    - Pos. Gain Calibration Constant = (128 - Pos. Error) / Gain adjust per bit
    - Neg. Error (in volts) = **`NVAL`** - **`ZVAL`** - 10
    - Neg. Gain Calibration Constant = (128 + Neg. Error) / Gain adjust per bit
13. Once the Gain Calibration Constants have been computed, the positive and negative gains should be set to these new values. This can be done using the *Gain Calibration* (**`J`**) command. For example, if the positive and negative Gain Calibration Constants are 131 and 148 respectively, the command **`J131,148 X`** should be sent to the DAC488.

After setting the new gain values, the Offset Calibration Constant should be computed. (The Offset Calibration Constant is affected by the Gain Calibration Constant.)

Part III: Acquiring the Offset Calibration Constant

1. After setting the new gain values, set the Offset of Port 1 to minimum (-255) and program Port 1 to output 0 volts. Command sent to the DAC488:

   `OUTPUT 09; H-255 V0 X`

2. Measure and record the port 1 output voltage. Call this voltage **`NOFFVAL`**.

3. Set the Offset of Port 1 to maximum (+255) and program Port 1 to output 0 volts. Command sent to the DAC488:

   `OUTPUT 09; H255 V0 X`

4. Measure and record the port 1 output voltage. Call this voltage **`POFFVAL`**.

5. Set the Offset of Port 1 to 0 and program Port 1 to output 0 volts. Command sent to the DAC488:

   `OUTPUT 09; H0 V0 X`

6. Measure and record the port 1 output voltage. Call this voltage **`ZERR`**.

7. The *Offset Calibration Constant* is computed using the following formulas:

   - Offset adjustment per bit = (**`POFFVAL`** - **`NOFFVAL`**) / 512
   - Offset Calibration constant = Negative **`ZERR`** / Offset adjustment per bit

8. The Offset Calibration Constant should now be set to the newly computed value by using the *Offset Calibration* (**`H`**) command. For example, if the Offset Calibration Constant is 18, the command **`H18 X`** should be sent to the DAC488.

BASIC Calibration Program

The following BASIC program (filename **`DACCAL.BAS`** or **`CALIBR8.BAS`**), which is included on disk, will calibrate the DAC488 using a Personal488 IEEE 488 interface card, Driver488 software, and a Keithley Instruments Model 196 DMM. Other DMMs (digital multimeters) may be used although the program will need to be modified. All device dependent commands are contained in subroutines thus allowing these commands to be easily changed to suit the DMM being used.

```
10      'DAC488 Calibration Program
20 KEY OFF
30      '       Requires a Keithley 196 DMM or equivalent
40      '
50 KAD$="26":DAD$="09"
60 DIM PFS!(4,3):DIM NFS!(4,3):DIM OFS!(4,3):REM error values
70 DIM GAIN(4,3,2):DIM OFFSET(4,3)
80 DIM RNG(3):RNG(1) = 1:RNG(2) = 5:RNG(3) = 10
90     'Establish communications with the IOtech Personal488
100 OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
110 OPEN "\DEV\IEEEIN" FOR INPUT AS #2
120     'Reset the Personal488
130     IOCTL#1,"BREAK"
140     IOCTL#1,"RESET"
150     IOCTL#1,"FILL ERROR"
160 CLS
170 PRINT"                      DAC488 Calibration Program
180 PRINT
190 PRINT"Ensure DAC488 is connected to IEEE488 bus using address";DAD$
200 PRINT"Ensure 196 is connected to IEEE488 bus using address";KAD$
210 PRINT"Press any key to continue "
220 IF INKEY$="" THEN GOTO 220
230     '
240     'Initialize the DAC488 and the Keithley 196
250     '
```

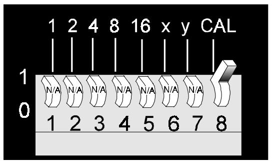
```
260 'Determine if unit is a DAC488/2 or DAC488/4
270 PRINT#1,"OUTPUT ";DAD$;";P3XE?"
275 PRINT#1,"ENTER 09"
280 INPUT#2,A$
290 IF A$ = "E0" THEN DAC2$ = "N" ELSE DAC2$ = "Y"
300 PRINT#1,"OUTPUT ";DAD$;";P1C0A0R3V10X"
310 PRINT
320 PRINT#1,"CLEAR ";KAD$;",";DAD$
330 PRINT#1,"OUTPUT ";KAD$;";F0R3Z0P0S2T1B0G1M0K0Y0X"
340 IF DAC2$ <> "Y" THEN GOTO 370
350 PRINT#1,"OUTPUT ";DAD$;";P1C0A0R3V10XP2C0A0R3V10X"
360 GOTO 380
370 PRINT#1,"OUTPUT";DAD$; ";P1C0A0R3V10XP2C0A0R3V10XP3C0A0R3V10XP4C0A0R3V10XM0Y2X"
380     'Calibrate all DAC ports
390 '
400 IF DAC2$ = "Y" THEN LASTPORT = 2 ELSE LASTPORT = 4
410 FOR PORT = 1 TO LASTPORT
420 PRINT#1,"OUTPUT ";DAD$;";P";PORT;"A0R3V#0X"
430 GOSUB 600:REM do the port
440 NEXT PORT
450 PRINT "Calibration Complete!  Save Calibration Constants in RAM? (Y/N)"
460 INPUT RESP$
470     'Print Calibration Reports
480     '
490 IF LEFT$(RESP$,1) <> "Y" AND LEFT$(RESP$,1) <> "y" THEN GOTO 510
500 PRINT #1,"output ";DAD$;";s3x"
510 PRINT "Print Verification report ?"
520 INPUT RESP$
530 IF LEFT$(RESP$,1) = "Y" OR LEFT$(RESP$,1) = "y" THEN GOSUB 1820
540 PRINT "Print Calibration report ?"
550 INPUT RESP$
560 IF LEFT$(RESP$,1) = "Y" OR LEFT$(RESP$,1) = "y" THEN GOSUB 1550
570 PRINT#1,"CLEAR ";KAD$;",";DAD$
580 KEY ON
590 END
600    'Calibrate a port subroutine
610    '
620 PRINT "Connect the output of port ";PORT;" to the DMM input"
630 PRINT"Press any key to continue "
640 IF INKEY$="" THEN GOTO 640
650 RANGE =3:FULLPOS =10!:FULLNEG =-10!
660 GOSUB 740:REM DO THE RANGE
670 RANGE =2:FULLPOS =5!:FULLNEG =-5!
680 GOSUB 740:REM DO THE RANGE
690 RANGE =1:FULLPOS =1!:FULLNEG =-1!
700 GOSUB 740:REM DO THE RANGE
710 RANGE =0:FULLPOS =0!:FULLNEG =0!
720 GOSUB 740:REM DO THE RANGE
730 PRINT "Finished Calibrating DAC port #";PORT:RETURN
740    'Calibrate a range subroutine
750    '
760 PRINT "Calibrating DAC port #";PORT;" ";FULLPOS;"volt range."
770 PRINT#1,"OUTPUT ";DAD$;";R";RANGE;"H0J128,128V0X"
780 IF RANGE < 2 THEN PRINT #1,"OUTPUT";KAD$;";R2X" ELSE PRINT #1,"OUTPUT";KAD$;";R3X"
790 IF RANGE = 0 THEN PADJ = 127:NADJ=127:GOTO 1110
800 GOSUB 1520:REM delay
810 PRINT#1,"ENTER ";KAD$
820 INPUT#2,ZEROVAL$:ZVAL!=VAL(ZEROVAL$)
830 PRINT#1,"OUTPUT ";DAD$;";V#4000X"
840 GOSUB 1520:REM delay
850 PRINT#1,"ENTER ";KAD$
860 INPUT#2,POSVAL$:PVAL!=VAL(POSVAL$)
870 PRINT#1,"OUTPUT ";DAD$;";V#-4000X"
880 GOSUB 1520:REM delay
890 PRINT#1,"ENTER ";KAD$
900 INPUT#2,NEGVAL$:NVAL!=VAL(NEGVAL$)
910 REM ROUTINE TO CALCULATE GAINBIT
920 '
930 PRINT#1,"OUTPUT ";DAD$;";R";RANGE;"V#4000J255,255X"
940 GOSUB 1520:REM delay
```

```
950 PRINT#1,"ENTER ";KAD$
960 INPUT#2,MOSTGAIN$:MVAL!=VAL(MOSTGAIN$)
970 '
980 PRINT#1,"OUTPUT ";DAD$;";R";RANGE;"V#4000J0,0X"
990 GOSUB 1520:REM delay
1000 PRINT#1,"ENTER ";KAD$
1010 INPUT#2,NOGAIN$:NOVAL!=VAL(NOGAIN$)
1020 GAINBIT! = (MVAL!-NOVAL!)/256
1030 PERR = PVAL!-ZVAL!-FULLPOS
1040 NERR = NVAL!-ZVAL!-FULLNEG
1050 PADJ = CINT(128 - (PERR / GAINBIT))
1060 IF PADJ < 0 THEN PADJ = 0
1070 ELSE IF PADJ > 255 THEN PADJ = 255
1080 NADJ = CINT(128 + (NERR / GAINBIT))
1090 IF NADJ < 0 THEN NADJ = 0
1100 IF NADJ > 255 THEN NADJ = 255
1110 PRINT#1,"OUTPUT ";DAD$;";J";PADJ;",";NADJ;"X"
1120 'ROUTINE TO CALCULATE OFFBIT VALUE
1130 '
1140 PRINT#1,"OUTPUT ";DAD$;";R";RANGE;"V0H-255X"
1150 GOSUB 1520:REM delay
1160 PRINT#1,"ENTER ";KAD$
1170 INPUT#2,NOFF$:NOFFVAL!=VAL(NOFF$)
1180 PRINT#1,"OUTPUT ";DAD$;";R";RANGE;"V0H255X"
1190 GOSUB 1520:REM delay
1200 PRINT#1,"ENTER ";KAD$
1210 INPUT#2,POFF$:POFFVAL!=VAL(POFF$)
1220 OFFBIT!=(POFFVAL!-NOFFVAL!)/512
1230 PRINT#1,"OUTPUT ";DAD$;";H0V0X"
1240 GOSUB 1520:REM delay
1250 PRINT#1,"ENTER ";KAD$
1260 INPUT#2,ZEROVAL$:ZERR!=VAL(ZEROVAL$)
1270 IF (ZERR!/OFFBIT!) < 0 THEN ZADJ= CINT( -(ZERR!/OFFBIT!)) ELSE ZADJ= -
CINT(ZERR!/OFFBIT!)
1280 IF ZADJ < -255 THEN ZADJ = 255 ELSE IF ZADJ>255 THEN ZADJ = 255
1290 PRINT#1,"OUTPUT ";DAD$;";H";ZADJ;"X"
1300    'Measure error after calibration'
1310    '
1320 PRINT#1,"OUTPUT ";DAD$;";R";RANGE;"V0X"
1330 GOSUB 1520:REM delay
1340 PRINT#1,"ENTER ";KAD$
1350 INPUT#2,ZEROVAL$:OFS!(PORT,RANGE) = VAL(ZEROVAL$)
1360 IF RANGE = 0 THEN GOTO 1450
1370 PRINT#1,"OUTPUT ";DAD$;";V#4000X"
1380 GOSUB 1520:REM delay
1390 PRINT#1,"ENTER ";KAD$
1400 INPUT#2,POSVAL$:PFS!(PORT,RANGE)=((VAL(POSVAL$)/FULLPOS) * 100)-100
1410 PRINT#1,"OUTPUT ";DAD$;";V#-4000X"
1420 GOSUB 1520:REM delay
1430 PRINT#1,"ENTER ";KAD$
1440 INPUT#2,NEGVAL$:NFS!(PORT,RANGE)=((VAL(NEGVAL$)/FULLNEG) * 100)-100
1450 GAIN(PORT,RANGE,1) = PADJ
1460 GAIN(PORT,RANGE,2) = NADJ
1470 OFFSET(PORT,RANGE) = ZADJ
1480 RETURN
1490 PRINT "Error calibrating Port #";PORT;" ";FULLPOS;" RANGE:return
1500    'Delay routine allows readings to stabilize
1510    '
1520 FOR DELAY = 0 TO 2
1530 NEXT DELAY
1540 RETURN
1550    'Print Calibration Report'
1560    '
1570 INPUT "Enter the units serial number";SERNUM
1580 LPRINT CHR$(12):LPRINT:LPRINT
1590 DA$ = DATE$
1600 LPRINT"                     Calibration Report              ";DA$
1610 LPRINT
1620 IF DAC2$ <> "Y" THEN GOTO 1650
1630 LPRINT"Manufacturer: IOtech    Model: DAC488/2     Serial#";
1640 GOTO 1660                   CALIBRATION REPORT              ";DA$
1650 LPRINT"Manufacturer: IOtech    Model: DAC488/4     Serial#";
1660 LPRINT USING "######";SERNUM
```

```
1670 LPRINT
1680 LPRINT"Port  Range  %Err +FS  %Err -FS  Offset Err  J Vals  H Val"
1690 FOR PORT = 1 TO LASTPORT
1700 LPRINT
1710 FOR RANGE = 0 TO 3
1720 LPRINT USING "  #  ## V  +#.###  +#.###  +##.###^^^^  ### ###
+###";PORT,RNG(RANGE),PFS!(PORT,RANGE),NFS!(PORT,RANGE),
OFS!(PORT,RANGE),GAIN(PORT,RANGE,1),GAIN(PORT,RANGE,2), OFFSET(PORT,RANGE);
1730 IF ABS(PFS!(PORT,RANGE)) < .1 AND ABS(NFS!(PORT,RANGE)) < .1 AND
ABS(OFS!(PORT,RANGE)) < .001 THEN LPRINT "Passed" ELSE LPRINT "FAILED!
1740 NEXT RANGE
1750 NEXT PORT
1760 LPRINT
1770 LPRINT"      Calibrator:"
1780 LPRINT
1790 LPRINT"      Comments:"
1800 LPRINT CHR$(12);
1810 RETURN
1820 'ROUTINE TO PRINT PERFORMANCE VERIFICATION IN SAME FORMAT
1830 'AS AUTO CAL PROGRAM
1840 '
1850 FOR PORT = 1 TO LASTPORT
1860 LPRINT CHR$(12)
1870 PRINT "Connect the output of port ";PORT;" to the DMM input."
1880 PRINT"Press any key to continue "
1890 IF INKEY$="" THEN GOTO 1890
1900 IF DAC2$ <> "Y" THEN GOTO 1930
1910 LPRINT"IOtech  DAC488/2   Performance Verification   Port #";PORT
1920 GOTO 1940
1930 LPRINT"IOtech  DAC488/4   Performance Verification   Port #";PORT
1940 LPRINT"Nominal Actual Error Result  Nominal Actual Error Result"
1950 '
1960 ' 10 VOLT RANGE
1970 '
1980 PRINT#1,"OUTPUT ";KAD$;";F0R3Z0P0S2T1B0G1M0K0Y0X"
1990 RANGE = 3:V1 = 10: VSTEP = -1:OFSP=.005
2000 LPRINT
2010 GOSUB 2130
2020 RANGE = 2:V1 = 5: VSTEP = -.5:OFSP=.003
2030 LPRINT
2040 GOSUB 2130
2050 PRINT#1,"OUTPUT ";KAD$;";R2X":OFSP=.001
2060 RANGE = 1:V1 = 1: VSTEP = -.1
2070 LPRINT
2080 GOSUB 2130
2090 PRINT
2100 NEXT PORT
2110 KEY ON
2120 RETURN
2130 'SUBROUTINE TO PRINT REPORT AFTER CALIBRATION
2140 '
2150 FOR VOLTS! = V1 TO 0 STEP VSTEP
2160 '
2170 PRINT#1,"OUTPUT ";DAD$;";R";RANGE;"P";PORT;"V";VOLTS!;"X"
2180 GOSUB 1520 ' DELAY
2190 PRINT#1,"ENTER ";KAD$
2200 INPUT#2,ACTPOS$:ACTPOSV!=VAL(ACTPOS$) 'ACTPOS=ACTUAL POSITIVE VALUE READ
2210 PRINT#1,"OUTPUT ";DAD$;";R";RANGE;"P";PORT;"V-";VOLTS;"X"
2220 GOSUB 1520 ' DELAY
2230 PRINT#1,"ENTER ";KAD$
2240 INPUT#2,ACTNEG$:ACTNEGV!=VAL(ACTNEG$)  'ACTNEG = ACTUAL NEGATIVE VALUE READ
2250 POSERROR! = ABS(ACTPOSV! - VOLTS!)
2260 IF POSERROR! <=(VOLTS!*.001 + OFSP) THEN PRESULT$="Pass" ELSE PRESULT$="Failed"
2270 NEGERROR! = ABS(ACTNEGV! - (-1*VOLTS!))
2280 IF NEGERROR! <=((1*VOLTS!)*.001 + OFSP) THEN NRESULT$="Pass" ELSE NRESULT$="Failed"
2290 '
2300 LPRINT USING  "##.####   ##.####    #.####  \       \ ###.####   ###.####   #.####  \
\";VOLTS!,ACTPOSV!,POSERROR!,PRESULT$,-VOLTS!,ACTNEGV!,NEGERROR!,NRESULT$
2310 '
2320 NEXT VOLTS!
2330 RETURN
```

Automatic Calibration Procedure

Part I: Connections & Preparations

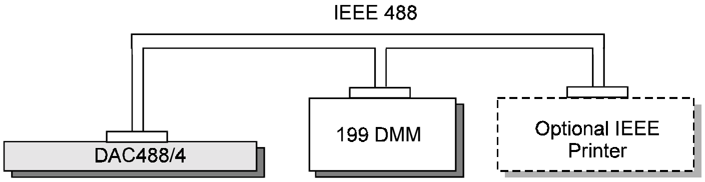


SW1 set for Auto Calibration

Fully automatic (or *Stand-Alone*) calibration of the DAC488 may be performed by connecting the DAC488 to a Keithley Instruments 199 DMM/Scanner. In the automatic calibration mode, the DAC488 acts as a bus controller. Automatic calibration requires the DAC488 to be connected to a Keithley 199 DMM both on the IEEE 488 bus and with the DAC ports connected to the scanner inputs using the calibration cable wired as described in the following text. An optional IEEE printer may also be connected on the bus and when calibration is finished the DAC488 will print a calibration report. Any other device which functions as a bus controller must be disconnected from the bus during the DAC488 auto-calibration.

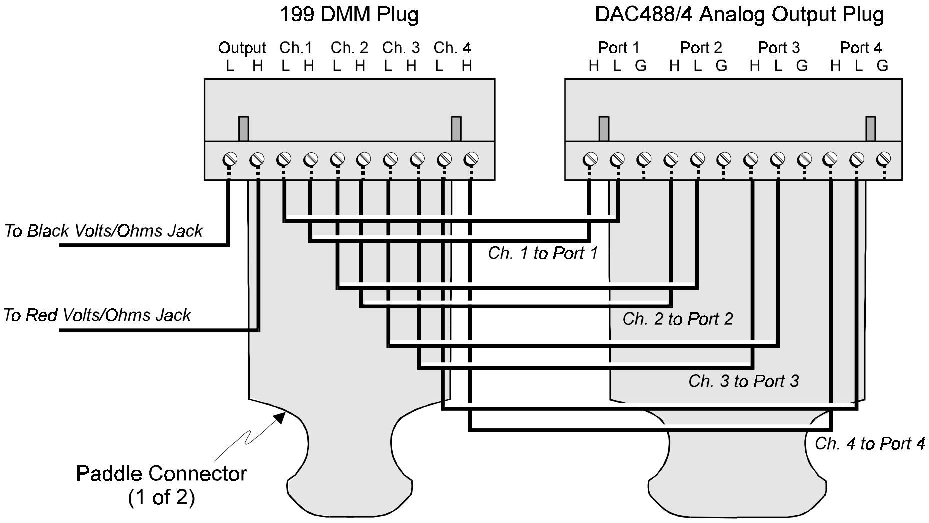
The IEEE address of the 199 must be set to 26 and the address of the printer must be 5. The address of the DAC488 is internally set to 0 regardless of the rear panel address switch settings.

1. With the unit off, set the rear-panel DIP microswitch 8 in the up position to enable the automatic calibration mode, as shown in the figure.
2. The Keithley Model 199 and optional printer must then be connected to the IEEE 488 connector on the DAC488 rear panel, as shown below. Any other IEEE controllers should be disconnected from the IEEE bus.



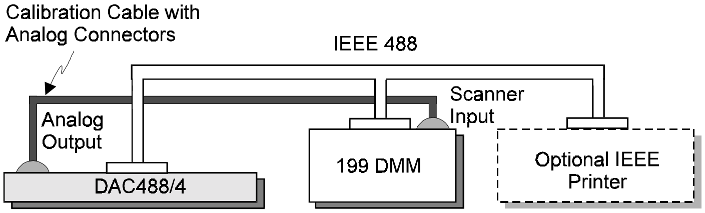
Auto-Calibration IEEE 488 Bus Configuration

3. You must now make a calibration cable using the analog connectors for the DAC488 and the 199. Both connectors have screw terminals therefore no soldering is required. The wiring for this cable is shown in the next diagram. If you are calibrating a DAC488/2, omit the wiring for ports 3 and 4.



Calibration Cable Wiring

4. Once the calibration cable has been made, the analog output of all DAC488 ports can then be connected to the 199, to complete the configuration in the following figure.



Complete Auto-Calibration Configuration

Part II: Automatic Calibration Process

When power is applied to the DAC488, its TEST LED will flash indicating that auto-calibration is taking place, then will turn off when the calibration has finished. If any errors occur during calibration the ERROR LED will flash and the calibration procedure will be aborted.

During auto-calibration, several messages will be displayed on the 199 in upper case characters. Each message will be displayed for 2 seconds. The messages will be displayed in the same order as listed below:

1. **`DAC488/4:`** This message is displayed if the DAC488/4 is properly connected to the 199, or:
   `DAC488/2:` This message is displayed if the DAC488/2 is properly connected to the 199.
2. **`AUTO_CAL:`** This message is displayed after message 1 to show that the DAC488 is in auto-calibration mode.
3. **`NO_SCANNER:`** This message is displayed if the 199 does not have a scanner installed.
4. **`NO_CONNECT:`** This message is displayed if the calibration cable is not connected or the wiring is incorrect.
5. **`CALIBRATE:`** This message is displayed if the calibration enable switch is in the depressed position which will allow the new Calibration Constants to be saved in the Non-Volatile RAM.
6. **`CAL_OFF:`** This message is displayed if the calibration enable switch is not in the depressed position. The DAC488 will not perform the auto-calibration. However, the present Calibration Constants will be used in the verify procedure. This approach may be used to verify the calibration of the unit without recalibrating it.
7. **`VERIFY:`** This message is displayed when the DAC488 has finished calibrating all ports and is ready to print a calibration verification report. If a printer is connected, a verification report is printed listing all voltages tested. Any failures are noted on the report. A summary report is also printed which lists the Calibration Constants for each port and range.
8. **`NO_PRINTER:`** This message is displayed if the DAC488 does not find a printer connected at bus address 5.
9. **`CAL_PASSED:`** This message is displayed if no errors occur during the verification tests.
10. **`CAL_FAILED:`** This message is displayed if any errors occur during the verification tests.

Note: All lines sent to the printer are terminated with a carriage return (**`CR`**).

Part III: Automatic Calibration Report

After the automatic calibration finishes, a calibration report will be printed if the DAC488 is connected to an IEEE printer. The following example shows the calibration report of the three volt ranges for Port 1:

```
            DAC488/4         Performance Verification          Port # 1

Nominal   Actual    Error     Result  Nominal    Actual     Error     Result

10.0000   10.0000   0.0000    Pass    -10.0000   -10.0000   0.0000    Pass
9.0000    8.9979    0.0021    Pass    -9.0000    -8.9976    0.0024    Pass
8.0000    7.9997    0.0003    Pass    -8.0000    -7.9994    0.0006    Pass
7.0000    7.0003    0.0003    Pass    -7.0000    -7.0001    0.0001    Pass
6.0000    6.0011    0.0010    Pass    -6.0000    -6.0009    0.0009    Pass
5.0000    4.9999    0.0001    Pass    -5.0000    -4.9997    0.0003    Pass
4.0000    4.0004    0.0004    Pass    -4.0000    -4.0002    0.0002    Pass
3.0000    2.9998    0.0003    Pass    -3.0000    -2.9996    0.0004    Pass
2.0000    2.0005    0.0005    Pass    -2.0000    -2.0004    0.0004    Pass
1.0000    0.9993    0.0007    Pass    -1.0000    -0.9992    0.0008    Pass
0.0000    0.0000    0.0000    Pass    -0.0000    -0.0000    0.0000    Pass

5.0000    5.0000    0.0000    Pass    -5.0000    -5.0000    0.0000    Pass
4.5000    4.4989    0.0011    Pass    -4.5000    -4.4988    0.0012    Pass
4.0000    3.9998    0.0003    Pass    -4.0000    -3.9997    0.0003    Pass
3.5000    3.5002    0.0002    Pass    -3.5000    -3.5001    0.0001    Pass
3.0000    3.0005    0.0005    Pass    -3.0000    -3.0005    0.0005    Pass
2.5000    2.4999    0.0001    Pass    -2.5000    -2.4999    0.0001    Pass
2.0000    2.0001    0.0001    Pass    -2.0000    -2.0002    0.0002    Pass
1.5000    1.4999    0.0001    Pass    -1.5000    -1.4998    0.0002    Pass
1.0000    1.0002    0.0002    Pass    -1.0000    -1.0002    0.0002    Pass
0.5000    0.4996    0.0004    Pass    -0.5000    -0.4996    0.0004    Pass
0.0000    0.0000    0.0000    Pass    -0.0000    -0.0000    0.0000    Pass

1.0000    1.0001    0.0001    Pass    -1.0000    -1.0000    0.0000    Pass
0.9000    0.8998    0.0002    Pass    -0.9000    -0.8997    0.0003    Pass
0.8000    0.8000    0.0000    Pass    -0.8000    -0.7999    0.0001    Pass
0.7000    0.7001    0.0001    Pass    -0.7000    -0.7000    0.0000    Pass
0.6000    0.6001    0.0001    Pass    -0.6000    -0.6000    0.0000    Pass
0.5000    0.5000    0.0000    Pass    -0.5000    -0.4999    0.0001    Pass
0.4000    0.4001    0.0001    Pass    -0.4000    -0.3999    0.0001    Pass
0.3000    0.3000    0.0000    Pass    -0.3000    -0.2998    0.0002    Pass
0.2000    0.1995    0.0005    Pass    -0.2000    -0.1993    0.0007    Pass
0.1000    0.0997    0.0003    Pass    -0.1000    -0.0995    0.0005    Pass
0.0000    0.0000    0.0000    Pass    -0.0000    -0.0000    0.0000    Pass
```

Automatic Calibration Report Format

A report similar to the one above will be printed for all four ports. A description of each item in the auto-calibration report is given below:

- ***Nominal:*** The voltage the DAC was programmed to output.
- ***Actual:*** The actual voltage measured by the DMM.
- ***Error:*** The difference between the Nominal and the Actual values.
- ***Result:*** Indicates Pass if the port is functioning within the stated specifications.

After the last port is tested, a summary report will be printed similar to that below:

```
                                         Calibration Report

Manufacturer:                   Model: DAC488/4                         Serial#


Port    Range    %Err +FS    %Err -FS    Offset Err       J    Vals    H Val

1       1        +0.003      -0.003      +28.000E-06    136    146     +15      Passed
1       2        -0.002      +0.000      -20.000E-06    118    126     +17      Passed
1       3        +0.001      -0.001      +0.000E+00     116    123     +18      Passed

2       1        -0.004      +0.003      -37.000E-06    112    98      -1       Passed
2       2        -0.001      +0.000      +0.000E+00     120    104     +3       Passed
2       3        -0.002      -0.001      -20.000E-06    119    102     +5       Passed

3       1        -0.001      +0.001      -27.000E-06    115    111     +11      Passed
3       2        -0.003      +0.001      -20.000E-06    118    113     +8       Passed
3       3        -0.001      +0.001      +0.000E+00     112    106     +1       Passed

4       1        +0.003      -0.007      +41.000E-06    123    113     +8       Passed
4       2        -0.002      +0.003      -50.000E-06    122    111     +18      Passed
4       3        +0.001      -0.001      -10.000E-06    127    114     +30      Passed
```

Automatic Calibration Summary Report

A description of each item in the auto-calibration summary report is given below:

- ***Port:*** This column shows which of the four ports is being tested.
- ***Range:*** This column shows which of the three volt ranges is being used, where 1 represents the ±1 volt range, 2 represents the ±5 volt range, and 3 represents the ±10 volt range.
- ***%Err +FS:*** This is the percent of Error at positive Full Scale (FS) Reading. This is computed using the following formula:

  `(FS Reading - FS Voltage)/(FS Voltage) x 100% = %Err +FS`

  For example, if a DAC port is programmed for 10 volts on the 10 volt scale and the voltmeter reading 10.0001 volts, then the percent of Error would be:

  `(10.0001 - 10.0000)/(10.0000) x 100% = +0.001%`

- ***%Err - FS:*** This is the percent of Error at negative Full Scale (FS) Reading. This is computed using the following formula:

  `(FS Reading - FS Voltage)/(FS Voltage) x 100% = %Err -FS`

  For example, if a DAC port is programmed for -5 volts on the 5 volt scale and the voltmeter reading -5.0002 volts, then the percent of Error would be:

  `(-5.0002 -  -5.0000 )/( -5.0000 ) x 100% = -0.004%`

- ***Offset Err:*** This column shows the measured output voltage when the port is programmed to output 0 volts.
- ***J Vals:*** These two columns show the positive and negative Gain Calibration Constants for each port and range. The positive Gain Calibration Constant is printed first.
- ***H Val:*** This column shows the Offset Calibration Constant for each range and port.

DAC488 Commands 7

Introduction

Control of the DAC488 is implemented with 26 commands, described here in detail. Examples are given for the commands using a PC functioning as an IEEE 488 controller by using the Personal488 PC/IEEE 488 board and associated Driver488 software. All examples are given using GW-BASIC or BASICA. Commands may be given using upper or lower case letters. The DAC488 bus address should be set to 9 for all examples.

In order to establish communication with Driver488 from GW-BASIC or BASICA, the following sequence must be used:

```
OPEN "\DEV\IEEEOUT" FOR OUTPUT AS #1
IOCTL#1,"BREAK"
PRINT#1,"RESET"
OPEN "\DEV\IEEEIN" FOR INPUT AS #2
```

All of the command examples that follow assume that the driver has been properly opened and reset by the above sequence.

Most of the instructions offer a *Query* (**?**) command extension. This extension can be used to determine the present configuration or mode of a command previously executed. Any number of these Query commands may be combined into one string to allow the user to construct a specialized status command requesting the DAC488 to return only that information which is of interest for a given application.

Note: It is necessary that the Execute command (X) follow all command strings sent to the DAC488. No commands are executed until an X is received by the DAC488. The only exceptions to this rule are the Query (?) commands and the CommandTrigger (@), which do not have to be followed by an X.

Command Syntax

Operation of the DAC488 is accomplished using a set of commands that configure the entire interface and commands that configure each DAC port. The syntax rules for using the commands are listed below.

- Each command consists of one alphabetic character followed by one or two numbers. The alphabetic character is referred to as the command and the number(s) following the command are the command parameters. Multiple parameters must be separated by a comma.

Examples	
A0	**A** specifies the Autorange command.
	0 is the parameter which specifies autorange disabled.
F0,20	**F** specifies the Buffer Definition command.
	0 is a parameter which specifies the starting location as location 0.
	20 is another parameter which specifies size of this particular buffer as 20 (20 locations).

- Commands are interpreted and processed as they are *received but are not executed until the Execute (**X**) command is issued*. (The exceptions to this rule are commands issued with the *Query* (**?**) parameter and the *Trigger On* (**@**) command which do not need to be followed by the *Execute* (**X**) command.)

Examples	
A0X	This command (disable autorange) will be executed immediately upon receipt of the **X.**
A0	The command will be interpreted,but not executed.
X	Upon receipt of the X, the Autorange command will be executed.

- Voltage values may be specified using scientific notation. (The **E** may be in upper or lower case).

Examples	
V5.6	These 4 commands specify the same voltage of 5.6 volts.
V0.56E1	
V56E-1	
V.056e+2	

- If bits are used to specify a voltage, the numeric bit value is preceded by a number sign (**#**).

Examples	
R3V#4000	The voltage specified is 4000 x 2.5 mV = 10 volts.
R3V#-3356	The voltage specified is -3356 x 2.5 mV = -8.39 volts.

- Hexadecimal values are preceded by a dollar sign (**$**) and followed by a **Z**.

Example	
R3V#$ACDZ	The voltage specified is 2765 x 2.5 mV = 6.91 volts. (**$ACD** = 2765)

- Commands may be entered in upper or lower case.

Example	
A0	These 2 commands are interpreted the same way.
a0	

- Spaces between commands and parameters are ignored.

Examples	
P1	This is interpreted the same way as **P 1**.
F0,20X	This is interpreted the same as **F 0 , 2 0 X**.

- Commands may be sent individually or in a string with other commands.

Example	
PRINT#1,"OUTPUT09;C0" **PRINT#1,"OUTPUT09;P1"** **PRINT#1,"OUTPUT09;V2X"**	These three lines of a program would have the same effect as the single line below: **PRINT#1,"OUTPUT09;C0 P1 V2 X"**

- Due to the structure of the command interpreter, each command may be issued only once within a string. However, the same command may be issued more than once in a string if each usage of the command is followed by the *Execute* (**X**) command.

Example	
PRINT#1, **"OUTPUT09;P1 V3 X P2 V5 X"**	You can program port 1 for 3 volts and port 2 for 5 volts using one command string as shown.

Command Groups

As introduced in Chapter 4, all of the DAC488 commands can be divided into two main groups: *System commands* and *port commands*.

System Commands

System commands affect the operation of the entire interface and are not specific to a given port. The 15 system commands are listed below:

@	Trigger On @
D	Digital Output
E?	Query Error Status
G	GET Trigger Mask
K	End-Or-Identify Control
M	Service Request Mask
O	Output Format
P	Port Select

Q	External Trigger Mask
S	System Defaults
T	Command Trigger Mask
U	User Status
W	Test
X	Execute
Y	Bus Terminator

Port Commands

Port commands only affect the operation of the selected port. The port is selected using the Port Select (**Pn**) command. The 11 port commands are listed below:

A	Autorange
B	Buffer Data
C	Control Mode
F	Buffer Definition
H	Offset Calibration
I	Interval

J	Gain Calibration
L	Buffer Location
N	Number of Cycles
R	Range Select
V	Value Output

Command Support

Support for the above 26 commands, includes the *Query* (**?**) command extension, and the use of the Serial Poll Status Byte. Refer to the upcoming “Command Reference” section in this Chapter, for more details.

Command Summary

The following summary presents all 26 of the DAC488 system and port commands in alphabetical order according to their command syntax.

Command		Syntax	Description
`@`	**Trigger On Command** (System Command)	`@`	Command trigger. Triggers the ports specified in the *Command Trigger Mask* (**`T`**). This command does NOT require the *Execute* (**`X`**) command to operate.
`A`	**Autorange** (Port Command)	**`An`**	Where **`n = 0`** or **`1`**.
		`A0`	Disable autoranging for selected DAC port.
		`A1`	(Default) Enable autoranging for selected DAC port.
		`A?`	Returns current autorange setting.
`B`	**Buffer Data** (Port Command)	**`Brng,volts`**	Write volts value in buffer location for selected DAC port.
		`Brng,#bits`	Write decimal bit value in buffer location for selected DAC port.
		`Brng,#$hexZ`	Write hexadecimal bit value in buffer location for selected DAC port.
		`B?`	Returns the range and value at the location pointer for the selected DAC port in the format specified by the Output Format (**`On`**) command.
		Notes:	Using this command causes the location counter to increment after the value has been written to or read from the buffer. Factory default buffer values are zero, range 0. **`rng`** = 0-3, **`volts`** = number without a V prefix.
`C`	**Control Mode** (Port Command)	**`Cn`**	Where **`n = 0, 1, 2,`** or **`3`**
		`C0`	(Default) Direct mode trigger on Execute (**`X`**) command.
		`C1`	Indirect mode, trigger on **`GET`**, External trigger or `@`.
		`C2`	Stepped mode, trigger on **`GET`**, External trigger or `@`.
		`C3`	Waveform mode, trigger on **`GET`**, External trigger or `@`.
		`C?`	Returns current control mode for selected port.
		Notes:	Selecting a control mode stops prior mode activity and rearms port for selected mode.
`D`	**Digital Output** (System Command)	**`Dval`**	Outputs the value on the digital output port.
		`D?`	Returns the current value of digital output port.
		Notes:	The default output value is 0. (**`val`** = 0 to 255)
`E?`	**Query Error Status** (System Command)	**`E?`**	Returns error status code which indicates the last error encountered. It returns: • **`E0`** if No Error has occurred. • **`E1`** if Unrecognized command. • **`E2`** if Invalid Command Parameter. • **`E3`** if Command Conflict Error. • **`E4`** if Calibration switch not closed. • **`E5`** if Non-Volatile RAM error.
		Notes:	Upon reading error status, error is cleared.
`F`	**Buffer Definition** (Port Command)	**`Fstart,size`**	Specifies starting location and number of locations used in the internal buffer for the selected port. Range of both values is 0 to 8191. See Notes.
		`F?`	Returns the starting location and number of locations used in the defined buffer.
		Notes:	The sum of **`start+size`** must be less than 8192. The default values are 0,1024; 1024,1024; 2048,1024; and 3072,1024 for ports 1 through 4 respectively.

Command		Syntax	Description
G	**GET Trigger Mask** (System Command)	Gmask	Sets bits in the GET Trigger Mask which specifies which ports will be triggered upon receipt of a bus **GET** command.
		G-mask	Clears specified bits in the GET Trigger Mask.
		G0	Clears all bits in the GET Trigger Mask.
		G?	Returns current GET Trigger Mask.
		Mask bits:	
		Bit 1	Enable trigger to DAC port 1. (decimal 1)
		Bit 2	Enable trigger to DAC port 2. (decimal 2)
		Bit 3	Enable trigger to DAC port 3. (decimal 4) (DAC488/4 only)
		Bit 4	Enable trigger to DAC port 4. (decimal 8) (DAC488/4 only)
		Notes:	The default mask value is 0.
H	**Offset Calibration** (Port Command)	Hval	Specifies the Offset Calibration Constant for selected range and port. Range is ±255.
		H?	Returns Offset Calibration Constant for selected range and port.
		Notes:	Calibration Constants must be programmed while using the Direct Control mode (**C0**). Autorange must be disabled (**A0**) to program Calibration Constants. The current voltage output is retriggered upon execution of this command. The default value is 0.
I	**Interval** (Port Command)	Ival	Specify time interval (in milliseconds) used with the Waveform control mode. Range for **val** is 1 to 65535.
		I?	Returns the current interval.
		Notes:	The default value is 1000, or 1 second.
J	**Gain Calibration** (Port Command)	Jpos,neg	Specifies the Gain Calibration Constant for both polarities of the selected range and port. **pos** = 0 to 255; **neg** = 0 to 255
		J?	Returns Gain Calibration Constants for selected range and port.
		Notes:	Calibration Constants must be programmed while using the Direct Control mode (**C0**). Autorange must be disabled (**A0**) to program Calibration Constants. The current voltage output is retriggered upon execution of this command. The default values are 128,128.
K	**End-Or-Identify Control** (System Command)	Kn	Where **n = 0** or **1**.
		K0	Assert **EOI** on last bus terminator.
		K1	(Default) Disable **EOI**.
		K?	Returns the current **EOI** Control setting.
L	**Buffer Location** (Port Command)	Lval	Specifies current buffer location. Range for **val** is 0 to 8191.
		L?	Returns the current buffer location in the format specified by the Output Format (**On**) command. The value returned after an **L?** is the first location that will be used in Stepped and Waveform modes.
M	**Service Request Mask** (System Command)	Mmask	Sets bits in the Service Request Mask which specifies which DAC488/4 events will generate a bus service request.
		M-mask	Clears the specified bits in the Service Request Mask.
		M0	Clears all bits in the Service Request Mask.
		M?	Returns current Service Request Mask.
		Mask bits:	
		Bit 1	Enable **SRQ** on DAC port 1 trigger ready. (decimal value 1)
		Bit 2	Enable **SRQ** on DAC port 2 trigger ready. (decimal value 2)
		Bit 3	Enable **SRQ** on DAC port 3 trigger ready. (decimal value 4) (DAC488/4 only)
		Bit 4	Enable **SRQ** on DAC port 4 trigger ready. (decimal value 8) (DAC488/4 only)
		Bit 5	Enable **SRQ** on Trigger Overrun. (decimal value 16)
		Bit 6	Enable **SRQ** on error. (decimal value 32)
		Bit 8	Enable **SRQ** on External input transition. (decimal value 128)

Command		Syntax	Description
N	**Number of Cycles** (Port Command)	Nval	Specifies the number of cycles through the buffer in the waveform mode. val is 0 to 65535, where 0 is continuous.
		N?	Returns the number of repetitions specified for use in the waveform mode.
		Notes:	The default value is 1.
O	**Output Format** (System Command)	On	Where n = 0, 1 or 2.
		O0	(Default) Sets output format to volts in ±10.0000 (fixed).
		O1	Sets output format to volts in decimal bits.
		O2	Sets output format to volts in hexadecimal bits.
		O?	Returns current output format selected.
P	**Port Select** (System Command)	Pn	Where n = 1, 2, 3, or 4.
		P1	(Default) Select DAC port 1.
		P2	Select DAC port 2.
		P3	Select DAC port 3. (DAC488/4 only)
		P4	Select DAC port 4. (DAC488/4 only)
		P?	Returns currently selected port.
Q	**External Trigger Mask** (System Command)	Qmask	Sets bits in the External Trigger Mask which specifies which the active edge of the external input line and which ports will be triggered upon a transition of that line.
		Q-mask	Clears the specified bits in the external trigger mask.
		Q0	Clears all bits in the External Trigger Mask.
		Q?	Returns current External Trigger Mask.
		Mask bits:	
		Bit 1	Enable trigger to DAC port 1. (decimal 1)
		Bit 2	Enable trigger to DAC port 2. (decimal 2)
		Bit 3	Enable trigger to DAC port 3. (decimal 4) (DAC488/4 only)
		Bit 4	Enable trigger to DAC port 4. (decimal 8) (DAC488/4 only)
		Bit 8	External input line edge sense: 1 = negative edge triggered (decimal 128); 0 = positive edge triggered.
		Notes:	The Bit 8 default value is 0.
R	**Range Select** (Port Command)	Rn	Where n = 0, 1, 2, or 3.
		R0	Select DAC Ground range.
		R1	(Default) Select DAC ±1 volt range.
		R2	Select DAC ±5 volt range.
		R3	Select DAC ±10 volt range.
		R?	Returns DAC volt range.
		Notes:	This command should not be used when Autorange is enabled.
S	**System Defaults** (System Command)	Sn	Where n = 0, 1, 2, or 3.
		S0	Restores the factory default values to NV-RAM.
		S1	Saves current settings as default values to NV-RAM.
		S2	Restore Gain and Offset Calibration Constants.
		S3	Saves current Calibration Constants to Calibration NV-RAM.
		S?	Returns the last Sn command executed.
T	**Command Trigger Mask** (System Command)	Tmask	Sets bits in the Command Trigger Mask that specify which ports will be triggered upon receipt of a Trigger command (@).
		T-mask	Clears specified bits in the Command Trigger Mask.
		T0	Clears all bits in the Command Trigger Mask.
		T?	Returns current Command Trigger Mask.
		Mask bits:	
		Bit 1	Enable trigger to DAC port 1. (decimal 1)
		Bit 2	Enable trigger to DAC port 2. (decimal 2)
		Bit 3	Enable trigger to DAC port 3. (decimal 4) (DAC488/4 only)
		Bit 4	Enable trigger to DAC port 4. (decimal 8) (DAC488/4 only)
		Notes:	The default value is 0.

Command		Syntax	Description
U	**User Status** (System Command)	Un	Where n = 0, 1, 2, 3, 4, 5, 6, 7, or 8.
		U0	Send system status on talk.
		U1	Send DAC port 1 status on talk.
		U2	Send DAC port 2 status on talk.
		U3	Send DAC port 3 status on talk. (DAC488/4 only)
		U4	Send DAC port 4 status on talk. (DAC488/4 only)
		U5	Send Digital input port status on talk.
		U6	Send Overrun status on talk.
		U7	Returns actual Output voltage and range.
		U8 (def)	Returns programmed Output voltage and range.
		U?	Returns current status select setting.
		Notes:	All status commands are one shot.
V	**Value Output** (Port Command)	Vvolts	Write volts value for selected DAC port.
		V#bits	Write decimal bit value for selected DAC port.
		V#$hexZ	Write hex bit value for selected DAC port
		V?	Returns current range and value for selected port in the format specified by the Output Format (On) command.
		Notes:	The current Autorange and Range affect the value. When programming in bits Autorange must be disabled. The digital port must be programmed in bits.
W	**Test** (System Command)	Wn	Where n = 0 or 1.
		W0	(Default) Turn off TEST LED.
		W1	Turn on TEST LED.
		W?	Return state of TEST LED.
X	**Execute** (System Command)	X	Execute Command String.
Y	**Bus Terminator** (System Command)	Yn	Where n = 0, 1, 2, or 3.
		Y0	(Default) Bus terminator is carriage-return line-feed.
		Y1	Bus terminator is line-feed carriage-return.
		Y2	Bus terminator is carriage return only.
		Y3	Bus terminator is line feed only.
		Y?	Returns current bus terminator setting.
?	**Query** (System Command; Command Support)	?	Returns present configuration or mode of the command preceding the **?**.

Command Reference

Like the command summary, the following detailed command reference presents all 26 of the DAC488 system and port commands in alphabetical order according to their command syntax. Support for these 26 commands, including the *Query* (**?**) command extension, and the use of the Serial Poll Status Byte, is also presented. All examples are given using GW-BASIC or BASICA.

@ - Trigger On Command

TYPE	System Command	
SYNTAX	`@`	Command trigger. Triggers the ports and channels specified in the *Command Trigger Mask* (**T**). This command does NOT require the *Execute* (**X**) command to operate.

DESCRIPTION

The Trigger command triggers the ports specified in the Command Trigger Mask (**Tmask**). This command does not require the Execute (**X**) command and is executed immediately upon receipt. It may be used in any control mode except the Direct Control (**C0**) mode (which does not require a trigger) to trigger one or more DAC ports simultaneously.

EXAMPLE 1: Triggering a single port.

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488
`PRINT#1,"OUTPUT09; A0 C1 T1 P1 R2 V3 X"`	**Line 2:** Disable Autoranging, select Indirect Control mode, Enable Port 1 for triggering on @, select Port 1, select 5 volt range, output 3 volts. (The DAC488 will not output the programmed voltage until it receives the @.)
`PRINT#1,"OUTPUT09;@"`	**Line 3:** Trigger the DAC. The DAC will output 3 volts on port 1.

EXAMPLE 2: Triggering multiple ports.

`PRINT#1,"OUTPUT09;T3 P1 R2 V3 X"`	**Line 1:** Enable Ports 1 and 2 for triggering on @, select Port 1, select 5 volt range, program port 1 for 3 volts.
`PRINT#1,"OUTPUT09;P2 R3 V8 X"`	**Line 2:** Select Port 2, select 10 volt range, program port 2 for 8 volts.
`PRINT#1,"OUTPUT09;@"`	**Line 3:** Trigger the DAC488. The DAC488/4 will output 3 volts on port 1 and output 8 volts on port 2.

A - Autorange

TYPE	Port Command	
SYNTAX	**An**	Where **n = 0** or **1**.
	A0	Disable autoranging for selected DAC port.
	A1	(Default) Enable autoranging for selected DAC port.
	A?	Returns current autorange setting.

DESCRIPTION

The Autorange command will enable or disable autoranging for the selected DAC port. If Autoranging is enabled, the DAC488 will select the range for the best resolution for the specified output voltage. The DAC488 will select a range for the specified voltage (V) based on the table shown. If Autoranging is disabled, the range is set using the Range Select (**Rn**) command.

Voltage	Range
0 < V < ±1	R1 (±1 V range)
1 < V < ±5	R2 (±5 V range)
5 < V	R3 (±10 V range)

Note: If Autoranging is enabled, the Range Select (**Rn**) command should not be used. A Conflict Error (**E3**) will result if the Range command is issued while the Port is in Autorange.

EXAMPLE 1: Triggering a single port.

```
PRINT#1,"CLEAR09"
PRINT#1,"OUTPUT09;C0 P1 A1 V3 X"
```

Line 1: Reset the DAC488

Line 2: Select Direct Control mode, select Port 1, enable Autorange, output 3 volts. DAC488 will output 3 volts on port 1.

B - Buffer Data

TYPE	Port Command	
SYNTAX	**`Brng, volts`**	Writes a voltage value in the buffer for the selected DAC port where **`rng`** selects the range and is one of the following: • When rng = 0, then Range = Ground • When rng = 1, then Range = ± 1 volt • When rng = 2, then Range = ± 5 volt • When rng = 3, then Range = ±10 volt and where **`volts`** is any voltage value specified as a number without a V prefix.
	`Brng, #bits`	Writes voltage value in bits to the buffer for selected DAC port
	`Brng, #$hexZ`	Writes voltage value in hexadecimal bits to the buffer for selected DAC port. Hexadecimal values must be followed by a **`Z`**.
	`B?`	Returns the range and voltage value at the location pointer for the selected DAC port in the format specified by the Output Format (**`On`**) command.

DESCRIPTION

The Buffer Data command is used to write a voltage value to the internal buffer. The buffer data is written to the location pointed to by the location pointer. The Location Pointer can be set to any location in the buffer by using the Buffer Location (**Lval**) command. The range (**rng**) and voltage value (**volts**) must be specified when using the Buffer Data command.

When creating a buffer of voltage values, the buffer is first defined using the Buffer Definition (**Fstart,size**) command, the location pointer is set to the first location in the newly defined buffer by using the Buffer Location (**Lval**) command, and the Buffer Data command is then used to fill the buffer.

The Execute (**X**) command must be used after each Buffer Data command. The location pointer will automatically increment after each Buffer Data command is executed. Buffer data may be written to locations outside of the defined buffer area for the selected port if so desired.

Note: (1) Each use of this command will cause the location counter to increment after the value has been written to or read from the buffer. (2) Factory default buffer values are range 0, zero volts. (3) If the Buffer Data command is used more than once within a command string, each Buffer command must be followed by the Execute (**X**) command.

Note: The DAC488 may be stopped while in the waveform mode by sending the Waveform Control (**C3**) command. This will halt the DAC488 but rearm it for another trigger from the bus.

EXAMPLE

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488
`PRINT#1,"OUTPUT09;A0 C3 P1 F0,2 G1 N3 L0 I2000 X"`	**Line 2:** Disables autorange, select waveform control mode, select port 1, define a buffer with the first location=0 and the number of locations=2, select trigger on **GET**, number of cycles=3, set location pointer to 0 (first location in the defined buffer), set time interval between points to 2 seconds.
`PRINT#1,"OUTPUT09;B2,3 X B2,4 X"`	**Line 3:** Sets buffer location 0 to 3 volts, sets buffer location 1 to 4 volts. ***Note:*** The location pointer will increment after the **B2,3** command is executed.
`PRINT#1,"OUTPUT09;L0X"`	**Line 4:** Sets location pointer back to 0 (first location). This is done so that when the DAC488 is triggered, it will start the sequence with the value in location 0.
`PRINT#1,"TRIGGER"`	**Line 5:** Trigger the DAC488. The DAC488 will output 3 volts, pause 2 seconds, output 4 volts, pause 2 seconds, then repeat the sequence 3 times.

C - Control Mode

TYPE	Port Command	
SYNTAX	**Cn**	Where **n = 0, 1, 2,** or **3**
	C0	(Default) Direct mode trigger on Execute (**X**) command.
	C1	Indirect mode, trigger on **GET**, External trigger or @.
	C2	Stepped mode, trigger on **GET**, External trigger or @.
	C3	Waveform mode, trigger on **GET**, External trigger or @.
	C?	Returns current control mode for selected port.

DESCRIPTION

The Control Mode command is used to select the operational mode of the DAC488. The modes and their operation are described in the following four examples.

Note: Issuing any control mode command stops the prior mode activity and rearms the port for the selected mode. For example, if the DAC488 is generating a waveform, it may be halted and rearmed by issuing the **C3** command.

EXAMPLE 1: Using the Direct Control mode.

```
PRINT#1,"CLEAR09"
```

Line 1: Reset the DAC488.

```
PRINT#1,"OUTPUT09;C0 P1
  A0 R2 V4 X"
```

Line 2: Select Direct Control mode, select Port 1, disable autoranging, select ±5 volt range, output 4 volts. The DAC488 will output 4 volts on Port 1 after executing this command string.

EXAMPLE 2: Using the Indirect Control mode.

```
PRINT#1,"CLEAR09"
```

Line 1: Reset the DAC488.

```
PRINT#1,"OUTPUT09;C1 P1
  T1 A0 R2 V4 X"
```

Line 2: Select Indirect Control mode, select Port 1, enable port 1 to trigger on @, disable autoranging, select ±5 volt range, output 4 volts.

```
PRINT#1,"OUTPUT09;@"
```

Line 3: The DAC488 will output 4 volts on Port 1 upon receiving the @ command.

EXAMPLE 3: Using the Stepped Control mode.

```
PRINT#1,"CLEAR09"
```

Line 1: Reset the DAC488.

```
PRINT#1,"OUTPUT09;C2 P1
  F0,3 L0 Q1 X"
```

Line 2: **C2** selects Stepped Control mode, **P1** selects port 1, **F0,3** defines a buffer with the first location as location 0 and the number of values in the sequence to 3, **L0** sets the location pointer back to 0 (first location) and **Q1** enables port 1 to trigger on an external trigger.

```
PRINT#1,"OUTPUT09;B1,1
  X B2,3 X B2,4 X"
```

Line 3: **B1,1X** selects the ±1 volt range, 1 volt for first point, **B2,3X** selects the ±5 volt range, 3 volts for second point and **B2,4X** selects the ±5 volt range, 4 volts for third point.

```
PRINT#1,"OUTPUT09;L0 X"
```

Line 4: **L0** sets the location pointer is set back to location 0 before the port is triggered. If this is not done, the location pointer will start at location 3, not location 0. The output of the sequence is controlled by the rate at which external triggers occur.

EXAMPLE 4: Using the Waveform Control mode. (See next page.)

EXAMPLE 4: Using the Waveform Control mode.

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.
`PRINT#1,"OUTPUT09;C3 P1 F0,2 G1 N0 L0 I2000 X"`	**Line 2:** Select Waveform Control mode, select port 1, define a buffer with the first location=0 and the size=2, enable port 1 to trigger on **GET**, set number of cycles to continuous, set location pointer to 0 (first location), set time interval between points to 2 seconds.
`PRINT#1,"OUTPUT09;B1,1 X B2,4 X"`	**Line 3:** Select ±1 volt range, 1 volt for first point, select ±5 volt range, 4 volts for second point.
`PRINT#1,"OUTPUT09;L0X"`	**Line 4:** Set the location pointer back to 0 (first location). This is done so that when the DAC488 is triggered, it will start the sequence with the value in location 0. The location pointer is automatically incremented whenever a voltage value is sent to the DAC488 internal buffer using the Buffer Data (**Brng,volts**) command.
`PRINT#1,"TRIGGER"`	**Line 5:** Trigger the DAC488. The DAC488 will output 1 volt, pause for 2 seconds, output 4 volts, pause for 2 seconds, then repeat this sequence indefinitely or until it receives another Control Mode (**Cn**) command.

D - Digital Output

TYPE	System Command	
SYNTAX	**`Dval`**	Outputs the value on the digital output port, where **`val`** is a value between 0 and 255. The value is output in binary (sending a **`D1`** command will set line 1 to a logic high level).
	`D?`	Returns the current value of the digital output port in the format specified by the Output Format (**`On`**) command.

DESCRIPTION

The Digital Output command is used to output up to 8 bits of data to the digital output port. The data will appear at the digital output port as soon as the command is executed. There are no triggering or buffer options available when using the digital output port. The digital input port may be read using the **U5** Status command.

Note: (1) The default output value is 0 (all digital output lines low, 0 volts). (2) All bits on the digital output port will be affected when using this command. For example, if **D1X** is sent, line 1 will be set to a logic high level (+5 volts) and the other lines set to a logic low level (0 volts).

EXAMPLE

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.
`PRINT#1,"OUTPUT09;D6 X"`	**Line 2:** Set lines 2 and 3 high on the digital output port (6 = 0000 0110 binary)
`PRINT#1,"OUTPUT09;D?"`	**Line 3:** Read the programmed state of the digital output port
`PRINT#1,"ENTER09"`	
`INPUT#2,A$`	
`PRINT A$`	**Line 6:** Display shows: **6**.

E? - Query Error Status

TYPE	System Command	
SYNTAX	**E?**	Return and clear present error condition. After execution of Error Query command, DAC488 will return one of the following error codes: • **E0** if No Error has occurred. • **E1** if Unrecognized command (e.g. **Z4X**). • **E2** if Invalid Command Parameter (e.g. **A62X, C10X**). • **E3** if Command Conflict Error (e.g. **A1 R2**). • **E4** if Calibration switch not closed. • **E5** if Non-Volatile RAM error, system defaults or Calibration Constants may be lost.

DESCRIPTION

The Error Query command is used to determine the present error condition on the selected port of the DAC488. After execution of the Error Query command, the present error condition is cleared. The error condition is also cleared by executing the Status (**U0**) command. When an error has occurred, the ERROR LED on the DAC488 front panel will turn on. The ERROR LED will stay on until an Error Query (**E?**) or Status (**U0**) command is executed.

If a Calibration Write Protected Failure (**E4**) occurs, this error condition may also be cleared by executing the Error Query (**E?**) command. This type of error occurs when an attempt is made to save Calibration Constants without first setting the calibration switch on the back panel to the depressed position.

If a Non-Volatile RAM error (**E5**) occurs on power up, the **S?** command may be used to check the system defaults. One of the following responses will be returned:

- **S0** = Factory defaults are in use, user programmed defaults were lost
- **S1** = User programmed defaults are still valid

The Calibration Constants may be tested by issuing the following command string: **A0 R0 X J?**. One of the responses below will be returned:

- **J128,128** = Factory Calibration Constants are in use, Calibration Constants which were saved from a calibration procedure have been lost. The DAC488 should be recalibrated if this response is returned.
- **J127,127** = Calibration Constants which were saved from a calibration procedure are still valid.

Note: After execution of the Error Query command, the present error condition is cleared.

EXAMPLE 1: No error.

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.
`PRINT#1,"OUTPUT09;E?"`	**Line 2:** Request error status.
`PRINT#1,"ENTER09"`	
`INPUT#2,A$`	
`PRINT A$`	**Line 5:** Display shows **E0**, no errors have occurred.

EXAMPLE 2: Invalid Command Parameter.

`PRINT#1,"OUTPUT09;C0 P1 A0 R1 V3 X"`	**Line 1:** Select Direct Control mode, select Port 1, disable autoranging, select ±1 volt range, output 3 volts. ERROR LED on DAC488 should turn on after this command string is executed.
`PRINT#1,"OUTPUT09;E?"`	**Line 2:** Request error status.
`PRINT#1,"ENTER09"`	
`INPUT#2,A$`	
`PRINT A$`	**Line 5:** Display shows **E2** (invalid command parameter) since an attempt was made to output 3 volts on the 1 volt range. The **V3** command caused the error since valid voltage values for the 1 volt range must be between -1 and 1 volt.

EXAMPLE 3: Unrecognized Command. (See next page.)

EXAMPLE 3: Unrecognized Command.

```
PRINT#1,"OUTPUT09;Z4X"
PRINT#1,"OUTPUT09;E?"
PRINT#1,"ENTER09"
INPUT#2,A$
PRINT A$
PRINT#1,"OUTPUT09;E?"
PRINT#1,"ENTER09"
INPUT#2,A$
PRINT A$
```

Line 1: Send an invalid command to the DAC488.

Line 2: Request error status. ***Note:*** ERROR LED should be on.

Line 5: Display shows **E1** (unrecognized command). ***Note:*** ERROR LED should be off.

Line 6: Request error status.

Line 9: Display shows **E0** since error condition has been cleared after reading error status.

F - Buffer Definition

TYPE	Port Command	
SYNTAX	`Fstart, size`	Defines the starting location and size of a buffer, where **`start`** and **`size`** are each a value between 0 and 8191, such that their sum is less than 8192.
	`F?`	Returns starting location and buffer size of most recently defined buffer

DESCRIPTION

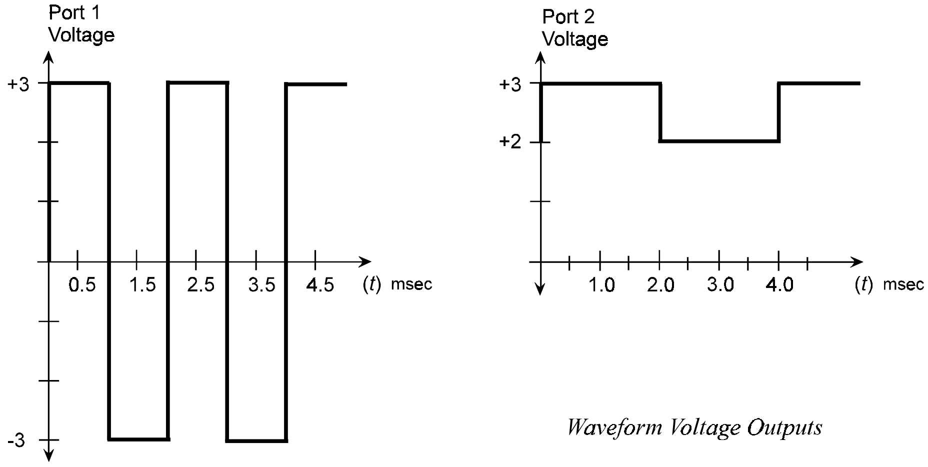
The Buffer Definition command is used to create a buffer of voltage values for use in the Stepped (**C2**) or Waveform (**C3**) Control modes. A buffer is defined by specifying the starting location and the buffer size. Any size buffer may be defined, but the last location must be at a numerically higher location than the first location. More than one port may share the same buffer if it is desired to output the same values at different ports simultaneously.

Note: (1) The sum of **start** and **size** must be less than 8192. (2) The default start and size values for each port are as follows:

Port	Def. Start	Def. Size	DAC488
1	0	1024	Both
2	1024	1024	Both
3	2048	1024	DAC488/4 only
4	3072	1024	DAC488/4 only

EXAMPLE

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.
`PRINT#1,"OUTPUT09;C3 P1 F0,2 G1 N0 L0 I1 X"`	**Line 2:** Select Waveform Control mode, select port 1, define a buffer with the starting location=0 and size=2, enable port 1 to trigger on **GET**, set number of cycles to continuous, set location pointer to 0 (first buffer location for the port 2 buffer), set time interval between points to 1 millisecond.
`PRINT#1,"OUTPUT09;B2,3 X B2,-3 X"`	**Line 3:** Set buffer location 0 to 3 volts, set buffer location 1 to -3 volts.
`PRINT#1,"OUTPUT09;L0 X"`	**Line 4:** Set location pointer back to 0 (first location for the port 1 buffer).
`PRINT#1,"OUTPUT09;C3 P2 F10,2 G2 N0 L10 I2 X"`	**Line 5:** Select Waveform Control mode, select port 2, define a buffer with the starting location=10 and size=2, enable port 2 to trigger on **GET**, set number of cycles to continuous, set location pointer to 10 (first location for the port 2 buffer), set time interval between points to 2 milliseconds.
`PRINT#1,"OUTPUT09;B2,3 X B2,2 X"`	**Line 6:** Set buffer location 10 to 3 volts set buffer location 11 to 2 volts.
`PRINT#1,"OUTPUT09;L10 X"`	**Line 7:** Set location pointer to 10 (first location for the port 2 buffer).
`PRINT#1,"TRIGGER"`	**Line 8:** Trigger the DAC488. The DAC488 will output the following waveforms on ports 1 and 2.



Waveform Voltage Outputs

G - GET Trigger Mask

TYPE	System Command	
SYNTAX	**`Gmask`**	Sets bits in the GET Trigger Mask that specify which ports will be triggered upon receipt of a bus **GET** command. The mask bits are as follows (decimal values are shown in parenthesis): • Bit 1 = Enable **GET** trigger on DAC port 1. (decimal 1) • Bit 2 = Enable **GET** trigger on DAC port 2. (decimal 2) • Bit 3 = Enable **GET** trigger on DAC port 3. (decimal 4) (DAC488/4 only) • Bit 4 = Enable **GET** trigger on DAC port 4. (decimal 8) (DAC488/4 only)
	`G-mask`	Clears the specified bits in the GET Trigger Mask.
	`G0`	Clears all bits in the GET Trigger Mask.
	`G?`	Returns current GET Trigger Mask.

DESCRIPTION

The GET Trigger Mask command is used to mask or unmask the **GET** trigger to allow DAC port triggering upon receipt of an IEEE Group Execute Trigger (**GET**) from the IEEE bus. One or more DAC ports may be masked or unmasked using this command.

Note: The default mask value is 0.

EXAMPLE

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.
`PRINT#1,"OUTPUT09;C1 G8 P4 A0 R2 V3 X"`	**Line 2:** Select Indirect Control mode, enable port 4 to trigger on a **GET**, select Port 4, disable autoranging, select 5 volt range, output 3 volts.
`PRINT#1,"TRIGGER"`	**Line 3:** Trigger the DAC. DAC488 will output 3 volts on port 4.

H - Offset Calibration

TYPE	Port Command	
SYNTAX	**`Hval`**	Specifies Offset Calibration Constant for selected range and port, where **`val`** is a value between -255 and +255.
	`H?`	Returns the Offset Calibration Constant for selected range and port.

DESCRIPTION

The Offset Calibration command programs the Offset Calibration Constant for the selected range and port. The Direct Control (**C0**) mode must be selected when using this command. The value may be any number in the range ± 255.

Note: (1) Calibration Constants must be programmed while using the direct control mode (**C0**). (2) Autorange must be disabled (**A0**) to program Calibration Constants. (3) The current programmed output voltage will appear at the analog output upon receipt of this command.

EXAMPLE

`PRINT#1,"OUTPUT09;C0 P1 A0 R2 H125 X"`	**Line 1:** Select Direct Control mode, select Port 1, disable autoranging, select ±5 volt range, set offset for ±5 volt range to 125.
`PRINT#1,"OUTPUT09;H?"`	**Line 2:** Request current Offset Calibration Constant for port 1.
`PRINT#1,"ENTER09"`	
`INPUT#2,A$`	
`PRINT A$`	**Line 5:** Display shows: **H+00125**

I - Interval

TYPE	Port Command	
SYNTAX	**Ival**	Specifies time interval used with the Waveform Control mode, where **val** is the time value between 1 and 65535 where 1 count = 1 msec.
	I?	Returns the current interval.

DESCRIPTION

The Interval command specifies the time interval between each voltage value that is output when in the Waveform Control mode (**C3**). The frequency of the output waveform may be controlled by selecting the proper interval based on the number of voltage points per cycle.

Note: The default value for **val** is 1000 milliseconds, or 1 second.

EXAMPLE

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.

`PRINT#1,"OUTPUT09;C3`
`  P1 F0,100 G1 N0 L0`
`  I40 X"`
	Line 2: Select Waveform Control mode, select port 1, define a buffer with the first location=0 and the number of locations=100, enable port 1 to trigger on **GET**, set number of cycles to continuous, set location pointer back to 0 (first location in the buffer), set the time interval between points to 40 milliseconds.

`PRINT#1,"TRIGGER"`	**Line 3:** Trigger the DAC488. A voltage value will be output at port 1 every 40 milliseconds.

J - Gain Calibration

TYPE	Port Command	
SYNTAX	`Jpos,neg`	Specifies the Gain Calibration Constant for both polarities of the selected range and port, where **pos** and **neg** are each a value between 0 and 255.
	`J?`	Returns the positive and negative Gain Calibration Constants for the selected range and port.

DESCRIPTION

The Gain Calibration command is used to set the positive and negative Gain Calibration Constants for the selected range and port. The range for each constant must be between 0 and 255. The larger the gain constant, the larger the absolute output voltage.

Note: (1) The DAC port must be in Direct Control mode (**C0**) and Autorange must be disabled (**A0**) to program the Calibration Constants. (2) The current voltage output is retriggered upon execution of this command. (3) The default value for **pos** and **neg** is 128.

EXAMPLE

`PRINT#1,"OUTPUT09;C0 P1 A0 R2 J50,60 X"`	**Line 1:** Select Direct Control mode, select Port 1, disable autoranging, select ±5 volt range, set gain for the positive polarity to 50, set gain for negative polarity to 60.
`PRINT#1,"OUTPUT09;J?"`	**Line 2:** Request current Gain Calibration Constants for port 1.
`PRINT#1,"ENTER09"`	
`INPUT#2,A$`	
`PRINT A$`	**Line 5:** Display shows: **J050,J060**.

K - End-Or-Identify Control

TYPE	System Command	
SYNTAX	**Kn**	Where **n = 0** or **1**.
	K0	Assert **EOI** on last bus terminator.
	K1	(Default) Do not assert **EOI** on last bus terminator.
	K?	Returns current **EOI** selection.

DESCRIPTION

The **EOI** line is one of five bus management lines on the IEEE 488 interface. It is used by a Talker to indicate the end of a multiple byte transfer sequence. The controller can change the **EOI** mode by programming the DAC488 from the bus. In the **K0** mode, the **EOI** line is asserted by the DAC488 on the last byte of every bus output string. In the **K1** mode, the **EOI** function is disabled.

EXAMPLE

```
PRINT#1,"OUTPUT09;K1 X"          Line 1:  Disables EOI assertion on last byte.
PRINT#1,"OUTPUT09;K?"            Line 2:  Request EOI selection from DAC488.
PRINT#1,"ENTER09"
INPUT#2,A$
PRINT A$                         Line 5:  Display shows:  K1.
```

L - Buffer Location

TYPE	Port Command	
SYNTAX	`Lval`	Sets the location pointer to location **`val`**, where **`val`** is a value between 0 and 8191.
	`L?`	Returns current buffer location. The **`L?`** command may be used even when the DAC488 is running in Stepped or Waveform Control mode.

DESCRIPTION

The Buffer Location command is used to set the location pointer to a specified location in the internal buffer. This command is used in conjunction with the Buffer Data (**Brng,volts**) command. After a data value is put in the buffer by using the Buffer Data command, the location pointer is automatically incremented.

The location pointer points to the first voltage value which will be output when the Stepped (**C2**) or Waveform (**C3**) Control modes are used. Once a DAC port is triggered in the Stepped or Waveform Control modes, the location pointer will always point to the next location containing the voltage value to be output. Each port has its own location pointer, therefore the location pointer may be set to a different location for each port. Before triggering a DAC port, the location pointer should be set to the desired starting point in the internal buffer.

CAUTION

DO NOT USE the Buffer Location (`Lval`) command to change the location pointer while the DAC488 is generating a waveform in Waveform Control mode (`C3`). Otherwise unpredictable operation may occur.

EXAMPLE

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.
`PRINT#1,"OUTPUT09;C3 P1 L0 F200,400 G1 N30 I2000 X"`	**Line 2:** Select Waveform Control mode, select port 1, define a buffer with the first location=200 and the size=400, enable port 1 to trigger on **GET**, set the number of cycles to 30, set location pointer back to 0 (first location), set time interval between points to 2 seconds. **Note:** Assume that we are using a buffer that has already been loaded with voltage values. Using the Buffer Definition (**Fstart,size**) command above defines what segment of the buffer we will be using when in Waveform mode.
`PRINT#1,"OUTPUT09;L200X"`	**Line 3:** Set location pointer to 200 (start of the buffer which was previously defined).
`PRINT#1,"TRIGGER"`	**Line 4:** Trigger the DAC488. The DAC488 will output the values in the buffer starting at location 200. (The buffer we have defined starts at location 200 and ends at location 600.)

M - Service Request Mask

TYPE	System Command	
SYNTAX	`Mmask`	Sets bit(s) in the Service Request Mask to specify which events will generate a bus service request (**SRQ**). The mask bits are as follows (decimal values are shown in parenthesis): • Bit 1 = Enable **SRQ** when DAC port 1 is ready for trigger. (decimal 1) • Bit 2 = Enable **SRQ** when DAC port 2 is ready for trigger. (decimal 2) • Bit 3 = Enable **SRQ** when DAC port 3 is ready for trigger. (decimal 4) (DAC488/4 only) • Bit 4 = Enable **SRQ** when DAC port 4 is ready for trigger. (decimal 8) (DAC488/4 only) • Bit 5 = Enable **SRQ** on trigger overrun. (decimal 16) • Bit 6 = Enable **SRQ** on error. (decimal 32) • Bit 8 = Enable **SRQ** on external input transition. (decimal 128)
	`M-mask`	Clears the specified bits in the Service Request Mask.
	`M0`	Clears all bits in the Service Request Mask.
	`M?`	Returns current Service Request Mask.

DESCRIPTION

The Service Request Mask command is used by the DAC488 to alert the controller to one of several conditions described below. Multiple Service Request (**SRQ**) conditions can be enabled simultaneously by issuing them separately or by combining them in one command string. If multiple Service Request Mask commands are contained within the same command string, each **Mmask** command must be followed by an Execute (**X**) command. The programmed **SRQ** modes will remain enabled until the **M0** (**SRQ** disabled) command is sent, or the controller sends a Device Clear (**DCL**) or Selected Device Clear (**SDC**).

Note: Like all mask commands, the Service Request Mask bits are logically linked together with OR, as received. To delete any one bit, it is necessary to use the **M-mask** option.

EXAMPLE 1: Invalid Command.

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.
`PRINT#1,"OUTPUT09;M32 X"`	**Line 2:** Select **SRQ** on error.
`PRINT#1,"OUTPUT09;Z6 X"`	**Line 3:** Send an invalid command. **Note:** ERROR and SRQ LEDs should illuminate.

EXAMPLE 2: Enable and Disable Commands.

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.
`PRINT#1,"OUTPUT09;M2 X M4 X"`	**Line 2:** Select **SRQ** on DAC port 2 ready, and **SRQ** on DAC port 3 ready.
`PRINT#1,"OUTPUT09;M6 X"`	**Line 3:** This command string would have the same effect as the command string sent in the line above where **M2X** plus **M4X** equals **M6X**.
`PRINT#1,"OUTPUT09;M-1 X"`	**Line 4:** The above command string will disable port 1 from generating an **SRQ** when ready.

N - Number of Cycles

TYPE	Port Command	
SYNTAX	**`Nval`**	Specifies the number of cycles that will be output when a DAC port is in the waveform mode, where **`val`** is a value between 0 and 65535; **`val`** = 0 will cause the DAC port to output the buffer values continuously.
	`N?`	Returns the number of cycles specified for use in the Waveform mode.

DESCRIPTION

The Number of Cycles command specifies how many times the values in a specified buffer will be output. For example, if it is desired to output a waveform 3 times, the number of cycles would be 3.

Note: The default value for n is 1, or 1 cycle.

EXAMPLE

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.
`PRINT#1,"OUTPUT09;C3 P1 F0,2 G1 N30 L0 I2000 X"`	**Line 2:** Select Waveform Control mode, select port 1, define a buffer with the first location=0 and a size=2, enable port 1 to trigger on **GET**, set number of cycles=30, set location pointer to 0 (first location), set time interval between point to 2 seconds.
`PRINT#1,"TRIGGER"`	**Line 3:** Trigger the DAC488. The DAC488 will output the waveform 30 times then halt.

O - Output Format

TYPE	System Command	
SYNTAX	**On**	Where **n = 0, 1** or **2**.
	O0	(Default) Sets output format to volts in ±10.0000 (fixed).
	O1	Sets output format to volts in decimal bits.
	O2	Sets output format to volts in hexadecimal bits.
	O?	Returns current output format selected.

DESCRIPTION

The Output Format command selects the output format that the DAC488 uses when sending a voltage value to the controller. The available output formats are: 0 to ±10.0000 volts (fixed point), 0 to ±4000 (decimal bits), or 0 to $0FFF (positive hexadecimal values) and $F001 to $FFFF (negative hexadecimal values). When using the hexadecimal format, negative voltages are specified using two's compliment form.

The bit resolution is: 250 µV/bit, 1.25 mV/bit and 2.5mV/bit for the ±1 volt, ±5 volt and ±10 volt ranges respectively.

EXAMPLE 1: Volts in ±10.0000 (fixed).

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.
`PRINT#1,"OUTPUT09;C0 P1 A0 R2 X"`	**Line 2:** Select Direct Control mode, Port 1, disable autoranging, select ±5 volt range.
`PRINT#1,"OUTPUT09;O0 V4 X"`	**Line 3:** Select output format in ±10.0000 (fixed), output 4 volts.
`PRINT#1,"OUTPUT09;V?"`	**Line 4:** Request the present voltage value.
`PRINT#1,"ENTER09"`	
`INPUT#2,A$`	
`PRINT A$`	**Line 7:** Display shows: **V+04.00000**.

EXAMPLE 2: Volts in Decimal Bits.

`PRINT#1,"OUTPUT09;O1"`	**Line 1:** Set output format to decimal bits.
`PRINT#1,"OUTPUT09;V?"`	**Line 2:** Request the present voltage value.
`PRINT#1,"ENTER09"`	
`INPUT#2,A$`	
`PRINT A$`	**Line 5:** Display shows: **V#+03200** (3200 x 1.25 mV = 4 V).

EXAMPLE 3: Volts in Hexadecimal Bits.

`PRINT#1,"OUTPUT09;O2X"`	**Line 1:** Set output format in hexadecimal bits.
`PRINT#1,"OUTPUT09;V?"`	**Line 2:** Request the present voltage value.
`PRINT#1,"ENTER09"`	
`INPUT#2,A$`	
`PRINT A$`	**Line 5:** Display shows: **V#$0C80** ($C80 = 3200).

P - Port Select

TYPE	System Command	
SYNTAX	**Pn**	Where **n = 1, 2, 3,** or **4**.
	P1	(Default) Select Port 1.
	P2	Select Port 2.
	P3	Select Port 3. (DAC488/4 only)
	P4	Select Port 4. (DAC488/4 only)
	P?	Return present port selection.

DESCRIPTION

The Port command determines which port is selected for use with subsequent commands. Only one port at a time may be selected. However, commands may be sent to all ports within one command string if each set of port commands are followed by the Execute (**X**) command.

EXAMPLE 1: One Command String for One Port.

```
PRINT#1,"CLEAR09"
PRINT#1,"OUTPUT09;C0 P1 A1
  V7 X"
```

Line 1: Reset the DAC488.

Line 2: Select Direct Control mode, select Port 1, enable autoranging, output 7 volts. The DAC488 will output 7 volts on port 1.

EXAMPLE 2: One Command String for Multiple Ports.

```
PRINT#1,"CLEAR09"
PRINT#1,"OUTPUT09; P1 C0 A0
  R3 V7 X P2 C0 A0 R3 V6 X"
```

Line 1: Reset the DAC488.

Line 2: Select Port 1, select Direct Control mode, disable autoranging, select ±10 volt range, output 7 volts. Select Port 2, select Direct Control mode, disable autoranging, select ±10 volt range, output 6 volts. Port 1 will output 7 volts and Port 2 will output 6 volts.

Q - External Trigger Mask

TYPE	System Command	
SYNTAX	**`Qmask`**	Set bit(s) in the External Trigger Mask to specify which ports will be triggered by an external trigger event. The mask bits are as follows (bit values are shown in parenthesis): • Bit 1 = Enable trigger to DAC port 1. (decimal value 1) • Bit 2 = Enable trigger to DAC port 2. (decimal value 2) • Bit 3 = Enable trigger to DAC port 3. (decimal value 4) (DAC488/4 only) • Bit 4 = Enable trigger to DAC port 4. (decimal value 8) (DAC488/4 only) • Bit 8 = External input edge sense: 1 = negative edge triggered; 0 = positive edge triggered. (decimal value 128)
	`Q-mask`	Clears the specified bits in the External Trigger Mask.
	`Q0`	Clears all bits in the External Trigger Mask.
	`Q?`	Returns current External Trigger Mask.

DESCRIPTION

The External Trigger Mask command enables or disables the specified DAC port for triggering by an external trigger signal. The external trigger signal is applied to the External Trigger/SRQ input on the digital I/O port.

Note: The default value for all bits is 0.

EXAMPLE

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.
`PRINT#1,"OUTPUT09;C1 Q129 P1 A0 R2 V2  X"`	**Line 2:** Select Indirect Control mode, enable port 1 for triggering by an external trigger signal, trigger on negative edge of external trigger signal, select port 1, disable autoranging, select 5 volt range, output 2 volts. Port 1 will output 2 volts when an external trigger pulse is applied. (Port 1 will trigger on the negative edge of the external trigger pulse).
`PRINT#1,"OUTPUT09;Q-12 X"`	**Line 3:** Clear bits 3 and 4 of External trigger mask to disable triggering of ports 3 and 4 when an external trigger pulse is received.

R - Range Select

TYPE	Port Command	
SYNTAX	**Rn**	Where **n = 0, 1, 2,** or **3**.
	R0	Select DAC Ground range. (DAC488 will output 0 volts only)
	R1	(Default) Select DAC ±1 volt range.
	R2	Select DAC ±5 volt range.
	R3	Select DAC ±10 volt range.
	R?	Returns DAC voltage range selection for selected port.

DESCRIPTION

The Range command specifies which output voltage range is used on the selected port. Autoranging must be disabled (**A0**) when using the Range command, otherwise a command Conflict Error (**E3**) will occur. The bit resolution is: 250 µV/bit, 1.25 mV/bit and 2.5 mV/bit for the ±1 volt, ± 5volt and ±10 volt ranges respectively.

Note: This command may not be used when Autoranging is enabled (**A1**) for the selected port.

EXAMPLE

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.
`PRINT#1,"OUTPUT09;C0 P1 A0 R2 V4 X"`	**Line 2:** Select Direct Control mode, select Port 1, disable autoranging, select ±5 volt range, output 4 volts. The DAC488 will output 4 volts on Port 1.

S - System Defaults

TYPE	System Command	
SYNTAX	**Sn**	Where **n = 0, 1, 2,** or **3**.
	S0	Restore the factory default command settings as the power on defaults.
	S1	Save the current command settings as the power on defaults.
	S2	Restore factory default Calibration Constants to NV-RAM.
	S3	Save current Calibration Constants in NV-RAM.
	S?	Return the last System Defaults command executed.

DESCRIPTION

The System Defaults (**Sn**) command may be used to save or restore a default setting for each command. The System Defaults command is also used to save or restore the Calibration Constants.

All Calibration Constants and a power on default setting for each command are saved in an internal Non-Volatile RAM (NV-RAM). Once Calibration Constants have been saved in the NV-RAM, they will be used whenever the DAC is commanded to output a voltage. The command options saved in NV-RAM will be the default settings when the unit is powered on.

Note: Issuing an **S2** or **S3** command without the calibration enable switch in the depressed position will cause a Calibration Write Protected (**E4**) error. After calibration, the **S3** command would be used to save the new Calibration Constants in the NV-RAM.

EXAMPLE

PRINT#1,"CLEAR09"	**Line 1:** Reset the DAC488.
PRINT#1,"OUTPUT09;A0 C1 O1 P2 Y1 X"	**Line 2:** Configure port 2 as follows: Autorange disabled, Indirect Control mode, Output Format in decimal bits, line feed (LF) terminator.
PRINT#1,"OUTPUT09;S1X"	**Line 3:** Save this configuration as the power on default configuration for port 2.

T - Command Trigger Mask

TYPE	System Command	
SYNTAX	**`Tmask`**	Set bit(s) in the Command Trigger Mask to specify which ports will be triggered by the Trigger command (@). The mask bits are as follows (bit values are shown in parenthesis): • Bit 1 = Enable DAC port 1 to trigger on @. (decimal 1) • Bit 2 = Enable DAC port 2 to trigger on @. (decimal 2) • Bit 3 = Enable DAC port 3 to trigger on @. (decimal 4) (DAC488/4 only) • Bit 4 = Enable DAC port 4 to trigger on @. (decimal 8) (DAC488/4 only)
	`T-mask`	Clears the specified bits in the Command Trigger Mask.
	`T0`	Clears all bits in the Command Trigger Mask.
	`T?`	Returns current Command Trigger Mask.
DESCRIPTION		
The Command Trigger Mask enables or disables the specified DAC port for triggering by the Trigger command (@). **Note:** The default bit value for each port is 0.		
EXAMPLE		
`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.	
`PRINT#1,"OUTPUT09;C1 T1 P1 A0 R2 V4 X"`	**Line 2:** Select Indirect Control mode, enable port 1 to trigger on @, select port 1, disable autoranging, select ±5 volt range, output 4 volts.	
`PRINT#1,"OUTPUT09;@"`	**Line 3:** Trigger DAC port 2. The DAC488 will output 4 volts on port 1.	

U - User Status

TYPE	System Command	
SYNTAX	**Un**	Where **n = 0, 1, 2, 3, 4, 5, 6, 7,** or **8**.
	U0	Send system status on talk.
	U1	Send DAC port 1 status on talk.
	U2	Send DAC port 2 status on talk.
	U3	Send DAC port 3 status on talk. (DAC488/4 only)
	U4	Send DAC port 4 status on talk. (DAC488/4 only)
	U5	Send digital input port status on talk.
	U6	Send Overrun status on talk.
	U7	Send actual output voltage and range on talk.
	U8	(Default) Send programmed output voltage and range on talk.
	U?	Returns current status select setting.

DESCRIPTION

The User Status (**Un**) command will cause the DAC488 to send a status message when next addressed to Talk. **U5** is used to read the digital input port. The **Dval** command is used to write to the digital output port. The status of the DAC488 may be read at any time without interfering with normal operation. Any error conditions are cleared after the status string is read by the controller.

The format of the status string returned by the DAC488 after receiving a Status command is as follows:

Command(s)	Returned Status String Format
U0	***.*D####E#G###K#M###O#P#Q###S#T###U#W#Y#** where the leading information ***.*** is the revision number of the DAC488 firmware.
U1,U2,U3,U4	**A#C#F#####,#####I#####L#####N#####P#R#V#.#####**
U5	**###**, where **###** is the decimal value of the input lines.
U6	**###**, where **###** specifies the port which was overrun: 001 = port 1, 002 = port 2, 004 = port 3, 008 = port 4, 006 = ports 2 and 3, etc.
U7	**C#P#R#V+##.#####,** where **V** is the actual output voltage
U8	**A#C#P#R#V+##.#####,** where **V** is the programmed output voltage.

Note: Each **#** equals the parameter(s) corresponding to that command.

EXAMPLE

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.
`PRINT#1,"OUTPUT09;U0 X"`	**Line 2:** Request the system status of the DAC488.
`PRINT#1,"ENTER09"`	
`INPUT#2,A$`	**Line 4:** Read the status string.
`PRINT A$`	**Line 5:** Display: ***.*D####E#G###K#M###O#P#Q###S#T###U#W#Y#** where each # would show the current option in use for each command.
`PRINT#1,"OUTPUT09;S0 X"`	**Line 6:** Restore the factory defaults as the power on defaults
`PRINT#1,"CLEAR09"`	**Line 7:** Reset the DAC488. Issuing an IEEE Interface Clear (**IFC**) will set the DAC488 to its power on state.
`PRINT#1,"OUTPUT09;U2 X"`	**Line 8:** Request the status of Port 2.
`PRINT#1,"ENTER09"`	
`INPUT#2,A$`	**Line 10:** Read the status.
`PRINT A$`	**Line 11:** Status string returned will be: **A1C0F01024,01024I01000L01024N00001P2R0V+00.00000** with the following data: Autorange on (**A1**), Direct Control mode (**C0**), First Buffer Location is 1024, Buffer size is 1024 locations (**F01024,01024**), Interval is 1000 milliseconds (**I01000**), Buffer Location Pointer is 1024 (**L01024**), Number of Cycles is 1 (**N00001**), Port 2 is the selected port (**P2**), Range is Ground Range (**R0**) (the range will be Ground Range, if Autoranging is disabled), and programmed output voltage is 0 volts (**V+00.00000**).

System Status (U0) Return Codes

A#	Autorange
A0	Autorange disabled for selected DAC port.
A1	Autorange enabled for selected DAC port.

C#	Control Mode
C0	Direct Mode, trigger on Value Output (**Vval**) command.
C1	Indirect Mode, trigger on **GET**, External trigger or **@**.
C2	Stepped Mode, trigger on **GET**, External trigger or **@**.
C3	Waveform Mode, trigger on **GET**, External trigger or **@**.

D###	Digital Output
###	A number between 0 and 255 showing the programmed state of the Digital Output port.

E#	Error Query
E0	No error.
E1	Unrecognized command.
E2	Invalid command parameter.
E3	Command conflict (e.g. attempting to select a range when in autorange).
E4	Calibration write protected.
E5	Non-Volatile RAM Error.
Note:	Upon reading the error status the error condition is cleared.

F####,xxxx	Buffer Definition
####	Starting location of the defined buffer (0 to 8191).
xxxx	Size of the defined buffer (0 to 8191).

G##	GET Trigger Mask
G0	Clears all bits in the GET Trigger Mask.
G1	Enable DAC port 1 to trigger on **GET**.
G2	Enable DAC port 2 to trigger on **GET**.
G4	Enable DAC port 3 to trigger on **GET**. (DAC488/4 only)
G8	Enable DAC port 4 to trigger on **GET**. (DAC488/4 only)
Note:	The status indication reflects the sum of all received GET Trigger Mask commands.

I#####	Interval
#####	Time interval **val** in milliseconds (**val** = 1 to 65,535 millisec).
Note:	The default Interval is **val** = 1000 (or 1 second).

K#	End-Or-Identify Control
K0	**EOI** asserted on last bus terminator.
K1	**EOI** disabled.

L####	Buffer Location
####	The present buffer pointer location (0 to 8191).

M###	Service Request Mask
M0	**SRQ** is disabled.
M1	**SRQ** on port 1 trigger ready.
M2	**SRQ** on port 2 trigger ready.
M4	**SRQ** on port 3 trigger ready. (DAC488/4 only)
M8	**SRQ** on port 4 trigger ready. (DAC488/4 only)
M16	**SRQ** on trigger overrun.
M32	**SRQ** on error.
M128	**SRQ** on external input transition.
Note:	The status indication reflects the sum of all received Service Request Mask commands.

N#####	Number of Cycles
#####	Number from 0 to 65,535 where 0 = Continuous output.

O#	Output Format
O0	Output format in Volts (±10.0000).
O1	Output format in decimal bits.
O2	Output format in hexadecimal bits.

P#	Port Select
P1	Port 1 selected.
P2	Port 2 selected.
P3	Port 3 selected. (DAC488/4 only)
P4	Port 4 selected. (DAC488/4 only)

Q###	External Trigger Mask
Q0	All bits are cleared in External Trigger Mask.
Q1	Enable external trigger on port 1.
Q2	Enable external trigger on port 2.
Q4	Enable external trigger on port 3. (DAC488/4 only)
Q8	Enable external trigger on port 4. (DAC488/4 only)
Q128	External input is negative edge triggered.
Note:	The status indication reflects the sum of all received External Trigger Mask commands.

R#	Range Select
R0	DAC Ground Range selected (output 0 volts only).
R1	DAC ±1 volt Range selected.
R2	DAC ±5 volt Range selected.
R3	DAC ±10 volt Range selected.

S#	System Defaults
S0	Restores the factory default values to NV-RAM.
S1	Saves the current settings as default values to NV-RAM.
S2	Restore Calibration Constants to Calibration NV-RAM.
S3	Save Calibration Constants to Calibration NV-RAM.

T##	Command Trigger Mask
T0	All bits are cleared in Command Trigger Mask.
T1	Enable Command Trigger on port 1.
T2	Enable Command Trigger on port 2.
T4	Enable Command Trigger on port 3. (DAC488/4 only)
T8	Enable Command Trigger on port 4. (DAC488/4 only)
Note:	The status indication reflects the sum of all received Command Trigger Mask commands

U#	User Status
U0	System Status sent on talk.

W#	Test
W0	TEST LED is off.
W1	TEST LED is on.

Y#	Bus Terminator
Y0	**CR LF** (carriage-return line-feed).
Y1	**LF CR** (line-feed carriage-return).
Y2	**CR** only (carriage return only).
Y3	**LF** only (line feed only).

V - Value Output

TYPE	Port Command	
SYNTAX	`Vval`	Writes volts value for selected DAC port, where `val` is a value between -10 and +10.
	`V#bits`	Writes bit value for selected DAC port, where `val` is a value between -4095 and +4095.
	`V#$hexZ`	Writes hexadecimal bit value for selected DAC port, where `val` is a value between F001 and +0FFF (-4095 and +4095).
	`V?`	Returns current range and value for selected port in the format specified by the Output Format (`On`) command.

DESCRIPTION

The Value Output command is used to program a voltage value to the selected DAC port. A voltage may be specified in any of three formats:

- **Volts: V7.5** or **V0.75E+1** (7.5 volts)
- **Decimal bits: V#3000** (3000 x 2.5 mV = 7.5 volts)
- **Hexadecimal bits: V#$BB8Z** (**$BB8** = 3000, which is also 7.5 volts)

The bit resolution is: 250 µV/bit, 1.25 mV/bit and 2.5mV/bit for the ±1 volt, ±5 volt and ±10 volt ranges respectively.

Notes: (1) When programming in bits, Autorange must be disabled (**A0**). The voltage output when programming in bits will depend on the range selected. Factory default value is zero volts (**V+00.00000**), Ground range (**R0**).
(2) When programming in hexadecimal bits, the hexadecimal number must be preceded by a **$** and followed by the letter **Z**.

EXAMPLE

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.
`PRINT#1,"OUTPUT09;C0 P1 A0 R2 V4 X"`	**Line 2:** Select Direct Control mode, select port 1, disable autoranging, select ±5 volt range, output 4 volts. The DAC488 will output 4 volts on Port 1.
`PRINT#1,"OUTPUT09;C0 P1 A0 R1 V#3200 X"`	**Line 3:** Select Direct Control mode, port 1, disable autoranging, select ±1 volt range, output 0.8 volts (voltage specified in decimal bits). The DAC488 will output 0.8 volts on Port 1. (3200 x 250 µV = 0.8 volts)
`PRINT#1,"OUTPUT09;C0 P1 A0 R2 V#3200 X"`	**Line 4:** Select Direct Control mode, port 1, disable autoranging, select ±5 volt range, output 4 volts (voltage specified in decimal bits). The DAC488 will output 4 volts on Port 1. (3200 x 1.25 mV = 4 volts)
`PRINT#1,"OUTPUT09;C0 P1 A0 R3 V#3200 X"`	**Line 5:** Select Direct Control mode, port 1, disable autoranging, select ±10 volt range, output 8 volts (voltage specified in decimal bits). The DAC488 will output 8 volts on Port 1. (3200 x 2.5 mV = 8 volts)

W - Test

TYPE	System Command	
SYNTAX	**Wn**	Where **n = 0** or **1**.
	W0	(Default) Turn off TEST LED.
	W1	Turn on TEST LED.
	W?	Return the state of the TEST LED.

DESCRIPTION

The Test command is used to verify that communication has been established with the DAC488. If the **W1** command is executed, the front panel TEST LED should light. If the **W0** command is then executed, the TEST LED should turn off. This command can be used to test the connection between the IEEE Controller and the DAC488.

EXAMPLE

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.
`PRINT#1,"OUTPUT09;W1X"`	**Line 2:** Turn on the TEST LED. The TEST LED on the front panel should light.
`PRINT#1,"OUTPUT09;W0X"`	**Line 3:** Turn off the TEST LED. The TEST LED on the front panel should turn off.

X - Execute

TYPE	System Command	
SYNTAX	**X**	Execute command string.

DESCRIPTION

Commands sent to the DAC488 will result in no action until the unit is instructed to execute these commands. This is done by issuing the Execute (**X**) command, usually as the last character of a command string. Commands sent without an **X** are stored in an internal buffer and are not executed until an **X** is received.

Any number of Execute commands may be inserted into the same command string. If a command is used more than once in a command string, the Execute command must be used before that command is used again.

EXAMPLE

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.
`PRINT#1,"OUTPUT09;V4"`	**Line 2:** Send **V4** to the DAC488 command input buffer.
`PRINT#1,"OUTPUT09;X"`	**Line 3:** Instruct the DAC488 to execute its command input buffer. The DAC488 will output 4 volts on the selected port upon receipt of the **X**.

`PRINT#1,"OUTPUT09;P1 V4 X`
`P2 V3 X"`
	Line 4: Two Port Select (**Pn**) commands issued within the same string require an **X** after each usage.

Y - Bus Terminator

TYPE	System Command	
SYNTAX	**Yn**	Where **n = 0, 1, 2,** or **3**.
	Y0	(Default) **CR LF** (carriage-return line-feed).
	Y1	**LF CR** (line-feed carriage-return).
	Y2	**CR** only (carriage return only).
	Y3	**LF** only (line feed only).
	Y?	Return present Bus Terminator selection.

DESCRIPTION

The IEEE 488 bus terminator(s) at power on, default to the Terminator setting saved in the Non-Volatile RAM. The terminators may be programmed for any combination of carriage return (**CR**) and line feed (**LF**). The **Y0** setting is the most commonly accepted terminator, **CR LF**. **Y1** reverses the sequence to send **LF CR**. **Y2** sends **CR** only, and **Y3** sends **LF** only. The Terminator selection may be saved as the power on default setting by using the System Defaults (**S1**) command.

EXAMPLE

`PRINT#1,"OUTPUT09;Y3 X"`	Select line feed terminator.

? - Query

TYPE	System Command (Command Support)	
SYNTAX	**?**	Returns present configuration or mode of the command preceding the **?**.

DESCRIPTION

All DAC488 commands offer a Query (**?**) option which may be used to determine the present configuration or mode of a command previously executed. To use this option, the first letter of the command is used, followed by a question mark (**?**). Any number of these Query commands may be combined into one string to allow the user to construct a specialized status command requesting the DAC488 to return only that information which is of interest for a given application. The Execute (**X**) command is not needed when using the Query option of a command.

EXAMPLE 1: SRQ Status.

`PRINT#1,"CLEAR09"`	**Line 1:** Reset the DAC488.
`PRINT#1,"OUTPUT09;M32 X"`	**Line 2:** Select **SRQ** on error.
`PRINT#1,"OUTPUT09;M?"`	**Line 3:** Determine last **Mmask** command executed.
`PRINT#1,"ENTER09"`	**Line 4:** Read data from the DAC488.
`INPUT#2,A$`	
`PRINT A$`	**Line 6:** Display shows: **M032**.

EXAMPLE 2: Port 1 Status Using Status Command (U).

`PRINT#1,"OUTPUT09;A0 C0 P1 R3 V8.12345 X"`	**Line 1:** Autoranging off, Direct Control mode, Select port 1, ±10 volt range, output 8.12345 volts.
`PRINT#1,"OUTPUT09;U8 X"`	**Line 2:** Request actual output voltage and range of port 1.
`PRINT#1,"ENTER09"`	**Line 3:** Read data from the DAC488.
`INPUT#2,A$`	
`PRINT A$`	**Line 5:** Display shows the status message: **A0C0P1R3V+08.12250** **Note:** The programmed voltage was rounded to the nearest value.

EXAMPLE 3: Port 1 Status Using Query Command (?).

`PRINT#1,"OUTPUT09;A0 C0 P1 R3 V8.12345 X"`	**Line 1:** Autoranging off, Direct Control mode, Select port 1, ±10 volt range, output 8.12345 volts.
`PRINT#1,"OUTPUT09;A?C?P?R?V?"`	**Line 2:** Request the status of port 1.
`PRINT#1,"ENTER09"`	**Line 3:** Read data from the DAC488.
`INPUT#2,A$`	
`PRINT A$`	**Line 5:** Display shows the same status message: **A0C0P1R3V+08.12250**

Serial Poll Status Byte

TYPE	Command Support		
SUMMARY	**Bit Location**	**Decimal Value**	**Description**
	DIO1	1	(LSB) DAC port 1 ready for trigger.
	DIO2	2	DAC port 2 ready for trigger.
	DIO3	4	DAC port 3 ready for trigger. (DAC488/4 only)
	DIO4	8	DAC port 4 ready for trigger. (DAC488/4 only)
	DIO5	16	Trigger Overrun.
	DIO6	32	Error.
	DIO7	64	Service Request bit.
	DIO8	128	(MSB) External Trigger Input Transition.

DESCRIPTION

The Serial Poll Status byte is sent upon receiving the Serial Poll (**SPOLL**) command from the controller. Refer to the Service Request Mask (**Mmask**) command description for details on how the Serial Poll byte is affected. To enable each bit to reflect the true status of the device, the appropriate **Mmask** command must be executed. The significance of each bit in the Serial Poll Status byte is shown below:

- **DIO1:** When enabled by the **M1** command, **DIO1** is set when DAC port 1 can accept a trigger. **DIO1** is cleared if the port is not ready or is generating a waveform in the Waveform (**C3**) mode.

- **DIO2:** When enabled by the **M2** command, **DIO2** is set when DAC port 2 can accept a trigger. **DIO2** is cleared if the port is not ready or is generating a waveform in the Waveform (**C3**) mode.

- **DIO3:** When enabled by the **M3** command, **DIO3** is set when DAC port 3 can accept a trigger. **DIO3** is cleared if the port is not ready or is generating a waveform in the Waveform (**C3**) mode. If the DAC488/2 is being Serial Polled, a zero will always be returned in this bit position.

- **DIO4:** When enabled by the **M4** command, **DIO4** is set when DAC port 4 can accept a trigger. **DIO4** is cleared if the port is not ready or is generating a waveform in the Waveform (**C3**) mode. If the DAC488/2 is being Serial Polled, a zero will always be returned in this bit position.

- **DIO5:** The **DIO5** bit is set when the DAC488 receives a trigger while processing a previous trigger. The bit is cleared by executing the Overrun Status (**U6**) or the Error Query (**E?**) commands and then reading the response.

- **DIO6:** The **DIO6** bit is set when any one of the error conditions (listed under the Error Query command) occurs. This bit is cleared by executing an Error Query (**E?**) or Status (**U0**) command.

- **DIO7:** The **DIO7** bit will be set when the DAC488 generates a Service Request (**SRQ)**. This is used by the controller to determine that the Service Request was generated by the DAC488. This bit is cleared when the DAC488 is Serial Polled.

- **DIO8:** The **DIO8** bit is set when an external trigger pulse is received on the external trigger/SRQ input and the DAC488 is properly armed. This bit is cleared when the DAC488 is Serial Polled.

EXAMPLE (Using a DAC488/4)

`PRINT#1,"OUTPUT09;S0 X"`	**Line 1:** Restore the factory power on defaults.
`PRINT#1,"CLEAR09"`	**Line 2:** Reset the DAC488.
`PRINT#1,"OUTPUT09;M32 X"`	**Line 3:** Select **SRQ** on error.
`PRINT#1,"OUTPUT09;P7 X"`	**Line 4:** Send an invalid bus command. ERROR and SRQ LEDs should illuminate.
`PRINT#1,"SPOLL09"`	**Line 5:** Serial Poll Status byte returned will be 111 (64 + 32 + 8 + 4 + 2 + 1). The interpretation of this Serial Poll Status byte is as follows: • 64: The DAC488 was the source of the **SRQ**. • 32: There was an error. • 08: Port 4 is ready for a trigger. • 04: Port 3 is ready for a trigger. • 02: Port 2 is ready for a trigger. • 01: Port 1 is ready for a trigger. When the DAC488 is Serial Polled, the SRQ LED will turn off.

Notes

Troubleshooting 8

Radio Interference Problems

DAC488 hardware systems generate, use and can radiate radio frequency energy, and if not installed and not used correctly, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, you the user are encouraged to try to correct the interference by one or more of the following measures:

- ***Antenna Adjustment:*** Reorient or relocate the receiving antenna.
- ***Spatial Separation:*** Increase the separation between the equipment and receiver.
- ***Circuit Separation:*** Connect the equipment to an outlet on a circuit different from that to which the receiver is connected.

Otherwise, consult the dealer of an experienced radio/television technician for help.

Electrostatic Discharge (ESD) Damage

The discharge of static electricity can damage some electronic components. Semiconductor devices are especially susceptible to ESD damage. You should always handle components carefully, and you should never touch connector pins or circuit components unless you are following ESD guidelines in an appropriate ESD-controlled area. Such guidelines include the use of properly-grounded mats and wrist straps, ESD bags and cartons, and related procedures.

Other Issues

Power-Up Activation

CAUTION

⚠ **Line Voltage: The protective conductor terminal on the AC line connector must be connected to an external protective earthing system. Failure to make such a connection will impair protection from shock.**

LED Patterns for Error Conditions

Condition	**LED Indicators**					
	TALK	**LISTEN**	**SRQ**	**ERROR**	**TEST**	**POWER**
Normal (No Errors)	(Off)	(Off)	(Off)	(Off)	(Off)	ON
ROM Error	ON	ON	ON	ON	ON	ON
RAM Error	FLASHING	FLASHING	FLASHING	FLASHING	FLASHING	ON
NV-RAM Checksum Error	(Off)	(Off)	(Off)	ON	(Off)	ON
No Power	(Off)	(Off)	(Off)	(Off)	(Off)	(Off)

With the power cord plugged in and connected to the DAC488, turn on the unit by depressing the rear-panel power switch. All of the front-panel LED indicators should light up for approximately one second while the DAC488 performs an internal ROM and RAM self-check. At the end of this self-check all of the LED indicators should turn off except for *POWER*.

Otherwise, the DAC488 may be in an error condition according to the one of the four following LED patterns:

- ***If all of the LED indicators remain on*:** Then a ROM error has occurred.
- ***If the POWER LED indicator remains on while the rest of the LED indicators flash continuously:*** Then a RAM error has occurred. Try cycling the power to the DAC488 to determine that the error is repeatable.
- ***If the POWER and ERROR LED indicator remain on while the rest of the LED indicators are off:*** Then a checksum error on Non-Volatile RAM occurred, and the saved configurations may be lost. The checksum error condition may be cleared by doing a save of a configuration using the *System Defaults* (**S**) command. The error condition may also be cleared by using the *Query Error Status* (**E?**) command.
- ***If all of the LED indicators are off:*** Then there may not be any power being supplied to the DAC488. In this event, check to make sure that the AC power cable is securely connected at both ends. Otherwise, there may be a problem with the fuse. For more information, see section *Power Line & Fuse Configuration* in Chapter 3.

Trigger Processing & Trigger Overrun

Operation of the DAC488 is controlled by an internal 1 millisecond timer. Each DAC port is updated at a maximum rate of once every millisecond. When a trigger event occurs, the DAC will output the programmed voltage within 1 millisecond of receiving the trigger. If a second trigger event occurs before the DAC has finished processing the first trigger, the second trigger will be held pending and a trigger overrun error will occur. The error condition may be cleared by issuing the Error Query command (E?) and then reading the response from the DAC488.   After the first trigger is processed, the pending trigger will then be processed. Any additional triggers which may occur while a trigger is held pending will be ignored.

Query Error Status

The *Query Error Status* (**E?**) command is used to determine the present error condition on the selected port of the DAC488. After execution of the *Query Error Status* command, the present error condition is cleared. The error condition is also cleared by executing the *Status* (**U0**) command. When an error has occurred, the *ERROR* LED on the DAC488 front panel will turn on. The *ERROR* LED will stay on until a *Query Error Status* (**E?**) or *Status* (**U0**) command is executed.

If a Calibration Write Protected Failure (**E4**) occurs, this error condition may also be cleared by executing the *Query Error Status* (**E?**) command. This type of error occurs when an attempt is made to save Calibration Constants without first setting the calibration switch on the back panel to the depressed position.

If a Non-Volatile RAM error (**E5**) occurs on power up, the **S?** command may be used to check the system defaults. One of the following responses will be returned:

- **S0** = Factory defaults are in use, user programmed defaults were lost
- **S1** = User programmed defaults are still valid

The Calibration Constants may be tested by issuing the following command string: **A0 R0 X J?**. One of the responses below will be returned:

- **J128,128** = Factory Calibration Constants are in use, Calibration Constants which were saved from a calibration procedure have been lost. The DAC488 should be recalibrated if this response is returned.
- **J127,127** = Calibration Constants which were saved from a calibration procedure are still valid.

For more information, see the *Query Error Status* (**E?**) command in the Chapter 7 command reference.

Appendix A

IEEE 488 Bus & Serial Bus Lines

Bus State	Bus Line	Data Transfer (DIO) Lines							
		8	7	6	5	4	3	2	1
Bus Management Lines									
IFC	Interface Clear								
REN	Remote Enable								
IEEE 488 Interface: Bus Management Lines									
ATN	Attention ($04)	0	0	0	0	0	1	0	0
EOI	End-Or-Identify ($80)	1	0	0	0	0	0	0	0
SRQ	Service Request ($40)	0	1	0	0	0	0	0	0
IEEE 488 Interface: Handshake Lines									
DAV	Data Valid ($08)	0	0	0	0	1	0	0	0
NDAC	Not Data Accepted ($10)	0	0	0	1	0	0	0	0
NRFD	Not Ready For Data ($20)	0	0	1	0	0	0	0	0
Serial Interface: Bus Management Lines									
DTR	Data Terminal Ready ($02)	0	0	0	0	0	0	1	0
RI	Ring Indicator ($10)	0	0	0	1	0	0	0	0
RTS	Request To Send ($01)	0	0	0	0	0	0	0	1
Serial Interface: Handshake Lines									
CTS	Clear To Send ($04)	0	0	0	0	0	1	0	0
DCD	Data Carrier Detect ($08)	0	0	0	0	1	0	0	0
DSR	Data Set Ready ($20)	0	0	1	0	0	0	0	0
Hexadecimal & Decimal Values									
	Hexadecimal Value	$80	$40	$20	$10	$08	$04	$02	$01
	Decimal Value	128	064	032	016	008	004	002	001

IEEE 488 Bus Commands

Bus State	IEEE 488 Bus Command	Data Transfer (DIO) Lines							
	(ATN is asserted “1”)	8	7	6	5	4	3	2	1
DCL	Device Clear	0	0	0	1	0	1	0	0
GET	Group Execute Trigger ($08)	0	0	0	0	1	0	0	0
GTL	Go To Local ($01)	0	0	0	0	0	0	0	1
LAG	Listen Address Group ($20-3F)	0	0	1	a	d	d	r	n
LLO	Local Lock Out ($11)	0	0	0	1	0	0	0	1
MLA	My Listen Address	0	0	1	a	d	d	r	n
MTA	My Talk Address	0	1	0	a	d	d	r	n
PPC	Parallel Poll Config	0	1	1	0	S	P2	P1	P0
PPD	Parallel Poll Disable ($07)	0	0	0	0	0	1	1	1
PPU	Parallel Poll Unconfig ($15)	0	0	0	1	0	1	0	1
SCG	Second. Cmd. Group ($60-7F)	0	1	1	c	o	m	m	d
SDC	Selected Device Clear ($04)	0	0	0	0	0	1	0	0
SPD	Serial Poll Disable ($19)	0	0	0	1	1	0	0	1
SPE	Serial Poll Enable ($18)	0	0	0	1	1	0	0	0
TAG	Talker Address Group ($40-5F)	0	1	0	a	d	d	r	n
TCT	Take Control ($09)	0	0	0	0	1	0	0	1
UNL	Unlisten ($3F)	0	0	1	1	1	1	1	1
UNT	Untalk ($5F)	0	1	0	1	1	1	1	1
Hexadecimal & Decimal Values									
	Hexadecimal Value	$80	$40	$20	$10	$08	$04	$02	$01
	Decimal Value	128	064	032	016	008	004	002	001

ASCII Codes

ASCII Code Summary

Decimal Values 00 to 63 – ACG, UCG & LAG

Box Items

Hexadecimal Value	$41 65	Decimal Value
	A	
Bus Message	01	(in center) ASCII Character

Addressed Command Group (ACG)

$00 00 NUL	$01 01 SOH GTL	$02 02 STX	$03 03 ETX	$04 04 EOT SDC	$05 05 ENQ	$06 06 ACK	$07 07 BEL PPD
$08 08 BS GET	$09 09 HT TCT	$0A 10 LF	$0B 11 VT	$0C 12 FF	$0D 13 CR	$0E 14 SO	$0F 15 SI

Universal Command Group (UCG)

$10 16 DLE	$11 17 DC1 LLO	$12 18 DC2	$13 19 DC3	$14 20 DC4 DCL	$15 21 NAK PPU	$16 22 SYN	$17 23 ETB
$18 24 CAN SPE	$19 25 EM SPD	$1A 26 SUB	$1B 27 ESC	$1C 28 FS	$1D 29 GS	$1E 30 RS	$1F 31 US

Listen Address Group (LAG)

$20 32 SP 00	$21 33 ! 01	$22 34 ” 02	$23 35 # 03	$24 36 $ 04	$25 37 % 05	$26 38 & 06	$27 39 ’ 07
$28 40 (08	$29 41) 09	$2A 42 * 10	$2B 43 + 11	$2C 44 , 12	$2D 45 - 13	$2E 46 . 14	$2F 47 / 15
$30 48 0 16	$31 49 1 17	$32 50 2 18	$33 51 3 19	$34 52 4 20	$35 53 5 21	$36 54 6 22	$37 55 7 23
$38 56 8 24	$39 57 9 25	$3A 58 : 26	$3B 59 ; 27	$3C 60 < 28	$3D 61 = 29	$3E 62 > 30	$3F 63 ? UNL

Decimal Values 64 to 127 – TAG & SCG

Box Items

Hexadecimal Value	$41 65	Decimal Value
	A	
Bus Message	01	(in center) ASCII Character

Talk Address Group (TAG)							
$40 64 @ 00	$41 65 A 01	$42 66 B 02	$43 67 C 03	$44 68 D 04	$45 69 E 05	$46 70 F 06	$47 71 G 07
$48 72 H 08	$49 73 I 09	$4A 74 J 10	$4B 75 K 11	$4C 76 L 12	$4D 77 M 13	$4E 78 N 14	$4F 79 O 15
$50 80 P 16	$51 81 Q 17	$52 82 R 18	$53 83 S 19	$54 84 T 20	$55 85 U 21	$56 86 V 22	$57 87 W 23
$58 88 X 24	$59 89 Y 25	$5A 90 Z 26	$5B 91 [27	$5C 92 \ 28	$5D 93] 29	$5E 94 ^ 30	$5F 95 _ UNT
Secondary Command Group (SCG)							
$60 96 ‘ 00	$61 97 a 01	$62 98 b 02	$63 99 c 03	$64 100 d 04	$65 101 e 05	$66 102 f 06	$67 103 g 07
$68 104 h 08	$69 105 i 09	$6A 106 j 10	$6B 107 k 11	$6C 108 l 12	$6D 109 m 13	$6E 110 n 14	$6F 111 o 15
$70 112 p 16	$71 113 q 17	$72 114 r 18	$73 115 s 19	$74 116 t 20	$75 117 u 21	$76 118 v 22	$77 119 w 23
$78 120 x 24	$79 121 y 25	$7A 122 z 26	$7B 123 { 27	$7C 124 | 28	$7D 125 } 29	$7E 126 ~ 30	$7F 127 DEL 31

ASCII Code Details

Decimal Values 00 to 31 – ACG & UCG Characteristics

ASCII Control Codes (Decimal 00 to 31)				
Dec Value	**Hex Value ($)**	**Character & Abbreviation**	**Name**	**Bus Message**
Addressed Command Group (ACG)				
00	**$00**	**None / NUL**	Null	None
01	**$01**	**^A / SOH**	Start of Header	Go To Local (**GTL**)
02	**$02**	**^B / STX**	Start of Text	None
03	**$03**	**^C / ETX**	End of Text	None
04	**$04**	**^D / EOT**	End of Transmission	Selected Device Clear (**SDC**)
05	**$05**	**^E / ENQ**	Inquiry	None
06	**$06**	**^F / ACK**	Acknowledgement	None
07	**$07**	**^G / BEL**	Bell	Parallel Poll Disable (**PPD**)
08	**$08**	**^H / BS**	Backspace	Group Execute Trigger (**GET**)
09	**$09**	**^I / HT**	Horizontal Tab	Take Control (**TCT**)
10	**$0A**	**^J / LF**	Line Feed	None
11	**$0B**	**^K / VT**	Vertical Tab	None
12	**$0C**	**^L / FF**	Form Feed	None
13	**$0D**	**^M / CR**	Carriage Return	None
14	**$0E**	**^N / SO**	Shift Out	None
15	**$0F**	**^O / SI**	Shift In	None
Universal Command Group (UCG)				
16	**$10**	**^P / DLE**	Data Link Escape	None
17	**$11**	**^Q / DC1**	Device Control 1	Local Lockout (**LLO**)
18	**$12**	**^R / DC2**	Device Control 2	None
19	**$13**	**^S / DC3**	Device Control 3	None
20	**$14**	**^T / DC4**	Device Control 4	Device Clear (**DCL**)
21	**$15**	**^U / NAK**	Negative Acknowledgement	Parallel Poll Unconfig (**PPU**)
22	**$16**	**^V / SYN**	Synchronous Idle	None
23	**$17**	**^W / ETB**	End of Transmission Block	None
24	**$18**	**^X / CAN**	Cancel	Serial Poll Enable (**SPE**)
25	**$19**	**^Y / EM**	End of Medium	Serial Poll Disable (**SPD**)
26	**$1A**	**^Z / SUB**	Substitute	None
27	**$1B**	**^[/ ESC**	Escape	None
28	**$1C**	**^\ / FS**	File Separator	None
29	**$1D**	**^] / GS**	Group Separator	None
30	**$1E**	**^^ / RS**	Record Separator	None
31	**$1F**	**^_ / US**	Unit Separator	None

Note: (1) ASCII control codes are sometimes used to "formalize" a communications session between communication devices. (2) **DC1, DC2, DC3, DC4, FS, GS, RS,** and **US** all have user-defined meanings, and may vary in use between sessions or devices. (3) **DC4** is often used as a general "stop transmission character." (4) Codes used to control cursor position may be used to control print devices, and move the print head accordingly. However, not all devices support the full set of positioning codes.

Decimal Values 00 to 31 – ACG & UCG Descriptions

ASCII Control Codes (00 to 31)		
Dec	**Name**	**Description**
Addressed Command Group (ACG)		
00	Null (**NUL**)	Space filler character. Used in output timing for some device drivers.
01	Start of Header (**SOH**)	Marks beginning of message header.
02	Start of Text (**STX**)	Marks beginning of data block (text).
03	End of Text (**ETX**)	Marks end of data block (text).
04	End of Transmission (**EOT**)	Marks end of transmission session.
05	Inquiry (**ENQ**)	Request for identification or information.
06	Acknowledgement (**ACK**)	"Yes" answer to questions or "ready for next transmission." Used in asynchronous protocols for timing.
07	Bell (**BEL**)	Rings bell or audible alarm on terminal.
08	Backspace (**BS**)	Moves cursor position back one character.
09	Horizontal Tab (**HT**)	Moves cursor position to next tab stop on line.
10	Line Feed (**LF**)	Moves cursor position down one line.
11	Vertical Tab (**VT**)	Moves cursor position down to next "tab line."
12	Form Feed (**FF**)	Moves cursor position to top of next page.
13	Carriage Return (**CR**)	Moves cursor to left margin.
14	Shift Out (**SO**)	Next characters do not follow ASCII definitions.
15	Shift In (**SI**)	Next characters revert to ASCII meaning.
Universal Command Group (UCG)		
16	Data Link Escape (**DLE**)	Used to control transmissions using "escape sequences."
17	Device Control 1 (**DC1**)	Not defined. Normally used for ON controls.
18	Device Control 2 (**DC2**)	Usually user-defined.
19	Device Control 3 (**DC3**)	Not defined. Normally used for OFF controls.
20	Device Control 4 (**DC4**)	Usually user-defined.
21	Negative Acknowledgement (**NAK**)	"No" answer to questions or "errors found, re-transmit." Used in asynchronous protocols for timing.
22	Synchronous Idle (**SYN**)	Sent by asynchronous devices when idle to insure sync.
23	End of Transmission Block (**ETB**)	Marks block boundaries in transmission.
24	Cancel (**CAN**)	Indicates previous transmission should be disregarded.
25	End of Medium (**EM**)	Marks end of physical media, as in paper tape.
26	Substitute (**SUB**)	Used to replace a character known to be wrong.
27	Escape (**ESC**)	Marks beginning of an Escape control sequence.
28	File Separator (**FS**)	Marker for major portion of transmission.
29	Group Separator (**GS**)	Marker for submajor portion of transmission.
30	Record Separator (**RS**)	Marker for minor portion of transmission.
31	Unit Separator (**US**)	Marker for most minor portion of transmission.

Note: (1) ASCII control codes are sometimes used to "formalize" a communications session between communication devices. (2) **DC1, DC2, DC3, DC4, FS, GS, RS,** and **US** all have user-defined meanings, and may vary in use between sessions or devices. (3) **DC4** is often used as a general "stop transmission character." (4) Codes used to control cursor position may be used to control print devices, and move the print head accordingly. However, not all devices support the full set of positioning codes.

Decimal Values 32 to 63 – LAG

ASCII Character Set (Decimal 32 to 63)				
Dec	**Hex**	**Character**	**Name**	**Bus Message**
Listen Address Group (LAG)				
32	**$20**	**<space>**	Space	Bus address 00
33	**$21**	**!**	Exclamation Point	Bus address 01
34	**$22**	**“**	Quotation Mark	Bus address 02
35	**$23**	**#**	Number Sign	Bus address 03
36	**$24**	**$**	Dollar Sign	Bus address 04
37	**$25**	**%**	Percent Sign	Bus address 05
38	**$26**	**&**	Ampersand	Bus address 06
39	**$27**	**‘**	Apostrophe	Bus address 07
40	**$28**	**(**	Opening Parenthesis	Bus address 08
41	**$29**	**)**	Closing Parenthesis	Bus address 09
42	**$2A**	*****	Asterisk	Bus address 10
43	**$2B**	**+**	Plus Sign	Bus address 11
44	**$2C**	**,**	Comma	Bus address 12
45	**$2D**	**-**	Hyphen or Minus Sign	Bus address 13
46	**$2E**	**.**	Period	Bus address 14
47	**$2F**	**/**	Slash	Bus address 15
Listen Address Group (LAG)				
48	**$30**	**0**	Zero	Bus address 16
49	**$31**	**1**	One	Bus address 17
50	**$32**	**2**	Two	Bus address 18
51	**$33**	**3**	Three	Bus address 19
52	**$34**	**4**	Four	Bus address 20
53	**$35**	**5**	Five	Bus address 21
54	**$36**	**6**	Six	Bus address 22
55	**$37**	**7**	Seven	Bus address 23
56	**$38**	**8**	Eight	Bus address 24
57	**$39**	**9**	Nine	Bus address 25
58	**$3A**	**:**	Colon	Bus address 26
59	**$3B**	**;**	Semicolon	Bus address 27
60	**$3C**	**<**	Less Than Sign	Bus address 28
61	**$3D**	**=**	Equal Sign	Bus address 29
62	**$3E**	**>**	Greater Than Sign	Bus address 30
63	**$3F**	**?**	Question Mark	Unlisten (**UNL**)

Decimal Values 64 to 95 – TAG

ASCII Character Set (Decimal 64 to 95)				
Dec	**Hex**	**Character**	**Name**	**Bus Message**
Talk Address Group (TAG)				
64	**$40**	**@**	At Sign	Bus address 00
65	**$41**	**A**	Capital A	Bus address 01
66	**$42**	**B**	Capital B	Bus address 02
67	**$43**	**C**	Capital C	Bus address 03
68	**$44**	**D**	Capital D	Bus address 04
69	**$45**	**E**	Capital E	Bus address 05
70	**$46**	**F**	Capital F	Bus address 06
71	**$47**	**G**	Capital G	Bus address 07
72	**$48**	**H**	Capital H	Bus address 08
73	**$49**	**I**	Capital I	Bus address 09
74	**$4A**	**J**	Capital J	Bus address 10
75	**$4B**	**K**	Capital K	Bus address 11
76	**$4C**	**L**	Capital L	Bus address 12
77	**$4D**	**M**	Capital M	Bus address 13
78	**$4E**	**N**	Capital N	Bus address 14
79	**$4F**	**O**	Capital O	Bus address 15
Talk Address Group (TAG)				
80	**$50**	**P**	Capital P	Bus address 16
81	**$51**	**Q**	Capital Q	Bus address 17
82	**$52**	**R**	Capital R	Bus address 18
83	**$53**	**S**	Capital S	Bus address 19
84	**$54**	**T**	Capital T	Bus address 20
85	**$55**	**U**	Capital U	Bus address 21
86	**$56**	**V**	Capital V	Bus address 22
87	**$57**	**W**	Capital W	Bus address 23
88	**$58**	**X**	Capital X	Bus address 24
89	**$59**	**Y**	Capital Y	Bus address 25
90	**$5A**	**Z**	Capital Z	Bus address 26
91	**$5B**	**[**	Opening Bracket	Bus address 27
92	**$5C**	\	Backward Slash	Bus address 28
93	**$5D**	**]**	Closing Bracket	Bus address 29
94	**$5E**	**^**	Caret	Bus address 30
95	**$5F**	**_**	Underscore	Untalk (**UNT**)

Decimal Values 96 to 127 – SCG

ASCII Character Set (96 to 127)				
Dec	**Hex**	**Character**	**Name**	**Bus Message**
Secondary Command Group (SCG)				
96	**$60**	**′**	Grave	Command 00
97	**$61**	**a**	Lowercase A	Command 01
98	**$62**	**b**	Lowercase B	Command 02
99	**$63**	**c**	Lowercase C	Command 03
100	**$64**	**d**	Lowercase D	Command 04
101	**$65**	**e**	Lowercase E	Command 05
102	**$66**	**f**	Lowercase F	Command 06
103	**$67**	**g**	Lowercase G	Command 07
104	**$68**	**h**	Lowercase H	Command 08
105	**$69**	**i**	Lowercase I	Command 09
106	**$6A**	**j**	Lowercase J	Command 10
107	**$6B**	**k**	Lowercase K	Command 11
108	**$6C**	**l**	Lowercase L	Command 12
109	**$6D**	**m**	Lowercase M	Command 13
110	**$6E**	**n**	Lowercase N	Command 14
111	**$6F**	**o**	Lowercase O	Command 15
Secondary Command Group (SCG)				
112	**$70**	**p**	Lowercase P	Command 16
113	**$71**	**q**	Lowercase Q	Command 17
114	**$72**	**r**	Lowercase R	Command 18
115	**$73**	**s**	Lowercase S	Command 19
116	**$74**	**t**	Lowercase T	Command 20
117	**$75**	**u**	Lowercase U	Command 21
118	**$76**	**v**	Lowercase V	Command 22
119	**$77**	**w**	Lowercase W	Command 23
120	**$78**	**x**	Lowercase X	Command 24
121	**$79**	**y**	Lowercase Y	Command 25
122	**$7A**	**z**	Lowercase Z	Command 26
123	**$7B**	**{**	Opening Brace	Command 27
124	**$7C**	**|**	Vertical Line	Command 28
125	**$7D**	**}**	Closing Brace	Command 29
126	**$7E**	**~**	Tilde	Command 30
127	**$7F**	**DEL**	Delete	Command 31

Notes

Index

Notes

Abbreviations

•	(bullet symbol) “and” (e.g. *SC•CA)	FCC	Federal Communications Commission
*	(asterisk symbol) “unasserted” (e.g. *SC)	FF	Form Feed (ASCII Code)
*CA	Not Controller Active mode	FS	File Separator (ASCII Code)
*SC	Not System Controller mode	GET	Group Execute Trigger bus command
A/D	Analog-to-Digital	GPIB	General Purpose Interface Bus
ACG	Addressed Command Group	GS	Group Separator (ASCII Code)
ACK	Acknowledgement (ASCII Code)	GTL	Go To Local bus command
ADC	Analog-to-Digital Converter	GUI	Graphical User Interface
API	Application Program Interface	H/W	Hardware
ASCII	American Standard Code for Info. Interchange	HT	Horizontal Tab (ASCII Code)
ATN	Attention line	IDDC	Invalid Device Dependent Command
BEL	Bell (ASCII Code)	IDDCO	Invalid Device Dependent Command Option
BS	Backspace (ASCII Code)	IEEE	Institute of Electrical & Electronic Engineers
CA	Controller Active mode	IFC	Interface Clear line
CAN	Cancel (ASCII Code)	IOCTL	Input/Output Control
CCL	Character Command Language	ISA	Industry Standard Architecture bus
CJC	Cold Junction Compensation	ISR	Interrupt Service Routine
CMD	Bus Command interpretation	ist	Bus Device Individual Status
CR	Carriage Return (ASCII Code)	LAG	Listen Address Group bus command
CSR	Calibration Status Register	LED	Light-Emitting Diode
CTS	Clear To Send line	LF	Line Feed (ASCII Code)
DAV	Data Valid line	LLO	Local Lock Out bus command
DC1	Device Control 1 (ASCII Code)	LSB	Least Significant Bit
DC2	Device Control 2 (ASCII Code)	MAV	Message Available bit
DC3	Device Control 3 (ASCII Code)	MLA	My Listen Address
DC4	Device Control 4 (ASCII Code)	MSB	Most Significant Bit
DCD	Data Carrier Detect line	MSS	Master Summary Status bit
DCL	Device Clear bus command	MTA	My Talk Address
DDE	Dynamic Data Exchange	N/U	Not Used
DEL	Delete (ASCII Code)	NAK	Negative Acknowledgement (ASCII Code)
DIO	Data Transfer (I/O) line	NDAC	Not Data Accepted line
DLE	Data Link Escape (ASCII Code)	NRFD	Not Ready For Data line
DLL	Dynamic Link Library	NUL	Null (ASCII Code)
DMA	Direct Memory Access	NV-RAM	Non-Volatile RAM
DMM	Digital Multimeter	PCI	Peripheral Component Interconnect bus
DSR	Data Set Ready line	PPC	Parallel Poll Configure bus command
DTR	Data Terminal Ready line	PPD	Parallel Poll Disable bus command
EEPROM	Electronically Erasable Programmable ROM	PPU	Parallel Poll Unconfig bus command
EM	End of Medium (ASCII Code)	RAM	Random-Access Memory
ENQ	Inquiry (ASCII Code)	REN	Remote Enable line
EOI	End-Or-Identify line	RI	Ring Indicator line
EOL	End-Of-Line terminator	RMS	Root Mean Square
EOT	End of Transmission (ASCII Code)	ROM	Read-Only Memory
EPROM	Erasable Programmable ROM	RQS	Request for Service bit
ESB	Event Status Register bit	RS	Record Separator (ASCII Code)
ESC	Error Source Register	RS-	Revised Standard (e.g. RS-232, RS-422)
ESC	Escape (ASCII Code)	rsv	Request for Service bit
ESE	Event Status Enable Register	RTD	Resistance Temperature Device
ESR	Event Status Register	RTS	Request To Send line
ETB	End of Transmission Block (ASCII Code)	SC	System Controller mode
ETX	End of Text (ASCII Code)	SCG	Secondary Command Group

SCPI	Standard Cmds. for Programmable Instruments	**SUB**	Substitute (ASCII Code)
SCSI	Small Computer System Interface bus	**SYN**	Synchronous Idle (ASCII Code)
SDC	Selected Device Clear bus command	**T/C**	Thermocouple
SI	Shift In (ASCII Code)	**TAG**	Talk Address Group bus command
SO	Shift Out (ASCII Code)	**TCT**	Take Control bus command
SOH	Start of Header (ASCII Code)	**TTL**	Transistor-Transistor Logic
SPD	Serial Poll Disable bus command	**UCG**	Universal Command Group
SPE	Serial Poll Enable bus command	**UNL**	Unlisten bus command
SRE	Service Request Enable Register	**UNT**	Untalk bus command
SRQ	Service Request line	**US**	Unit Separator (ASCII Code)
STB	Status Byte Register	**VDM**	Virtual DOS Machine
STX	Start of Text (ASCII Code)	**VT**	Vertical Tab (ASCII Code)

Notes