Peter Norberg Consulting, Inc.
Professional Solutions to Professional Problems
Ferguson, MO 63135 (314) 521-8808

Information and Instruction Manual for the **ISODAC Series of** Isolated Digital-to-Analog Boards

By
Peter Norberg Consulting, Inc.

Authored in the United States of America. Manual published February 25, 2016 5:55 AM

Table of Contents

Disclaimer and Revision History

All of our products are constantly undergoing upgrades and enhancements. Therefore, while this manual is accurate to the best of our knowledge as of its date of publication, it cannot be construed as a commitment that future releases will operate identically to this description. Errors may appear in the documentation; we will correct any mistakes as soon as they are discovered, and we will post the corrections on the web site in a timely manner. Please refer to the specific manual for the version of the hardware and firmware that you have for the most accurate information for your product.

This manual describes the ISODAC Isolated Digital-to-Analog conversion board, artwork version 5.

Product Warnings

Note that the product is not fully protected against static electricity. Its components can be damaged simply by touching the board when you have a "static charge" built up on your body. Such damage is not covered under either the satisfaction guarantee or the product warranty. Please be certain to safely "discharge" yourself before handling any of the boards or components.

LIFE SUPPORT POLICY

Due to the components used in the products (such as those from National Semiconductor Corporation and others), Peter Norberg Consulting, Inc.'s products are not authorized for use in life support devices or systems or in devices that can cause any form of personal injury if a failure occurs.

Note that National Semiconductor states "Life support devices or systems are devices which (a) are intended for surgical implant within the body, or (b) support or sustain life, and in whose failure to perform when properly used in accordance with instructions or use provided in the labeling, can be reasonably expected to result in a significant injury to the user". For a more detailed set of such policies, please contact National Semiconductor Corporation.

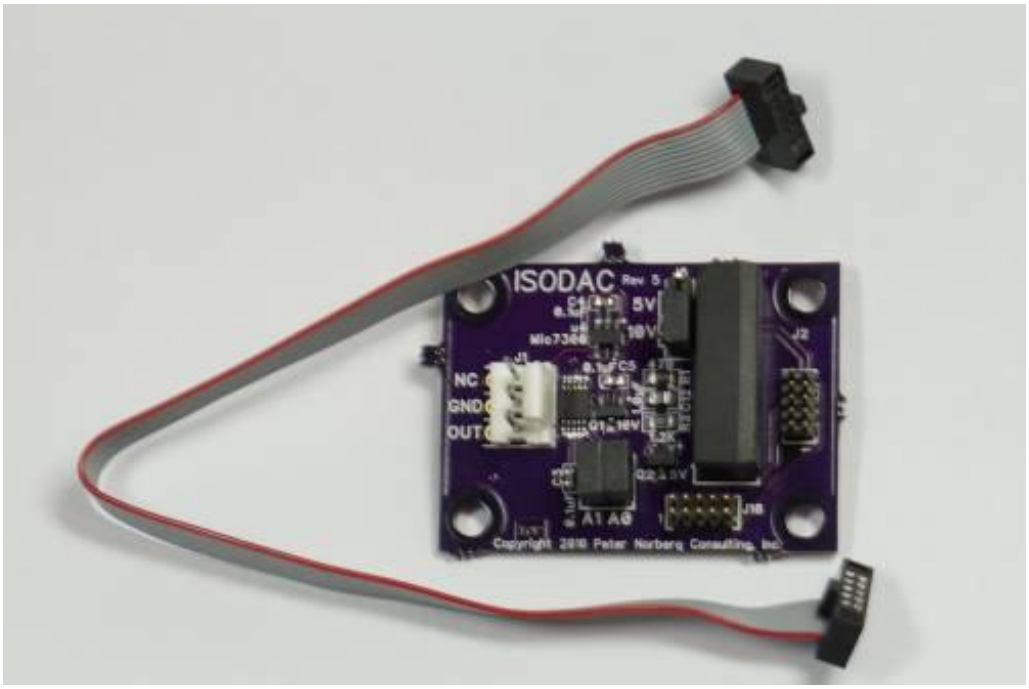
Introduction and Product Summary

The Peter Norberg Consulting, Inc. IOSDAC series of isolated digital-to-analog adapter boards are tiny (1.1" x 1.5") boards that allow generation of isolated 0-5 or 0-10 volt voltages by any of our TI-ARM-based motor controller products (such as our BC6D20, BC6D25 and SD6DX series). The product uses the tiny 10-pin JTAG programming header on those boards for its power and control input, and provides for at least 100 volts of isolation.

The ISODAC adapter is powered off of the hosting board – no additional power supply is needed in order for it to operate. It uses a digital isolator to isolate the board logic from the hosting board, and it uses an isolated DC to DC converter to generate all of the on-board voltages needed for its operation. It includes its own memory which can be used to set the power-on DAC voltage that is present as soon as power is applied.

From a connections point of view, there are 3 connectors on the board. Two are identical 10-pin JTAG headers, which are used to connect to the host board and to another ISODAC unit. The remaining connector is used to provide the isolated 0-5 or 0-10 volt output signal.

The two JTAG connecters are identical; either may be connected to the hosting ARM-based controller, while the other is available for daisy-chaining up to four of the ISODAC products. This allows up to 4 independent output DACs to be used by the hosting motor controller.



The voltage range is selected by an on-board jumper (the one that is close to the ISODAC text at the top of the above image). When connecting the top 2 pins together, the unit is configured for 0 to 5 volt operation. When connecting the bottom 2 pins together, the unit is configured for 0 to 10 volt operation. In either case, the board is normally accurate to about +/1% of the requested voltage.

The output of the board is present on the remaining 3-pin connector (to the left in the above image). Connect the GND pin to your ground reference (the point at which you want the 0-5/10 volt signal to be based from), and connect the OUT pin to the point where you want the 0-5/10 volt control signal to be present. The signal is driven through a standard analog buffer amplifier, and can supply up to 3 mA of current.

Via daisy-chaining, up to 4 ISODAC boards may be connected to one hosting motor controller. The two jumpers labeled A0 and A1 define the board address as required for the motor controller to identify the board.

Connecting the product

The ISODAC board appears as follows:

The image on the left shows the revision 5 board, the one in the center shows board with a JTAG cable attached to the rightmost connecter, and the one on the right shows the board with the JTAG cable attached to the bottom connector. The white connector on the left is the Amp MTA-100 connector (which may be replaced with a screw terminal connector as an order option) that provides the output signals from the board.

The board as shown is 1.5 inches wide by 1.1 inches tall. The mounting holes are 0.125" in diameter, positioned in 0.125" from each corner. This size hole can accept up to a number 5 screw; this means that the standard #4 mounting posts work quite well. The screw hole positions are therefore:

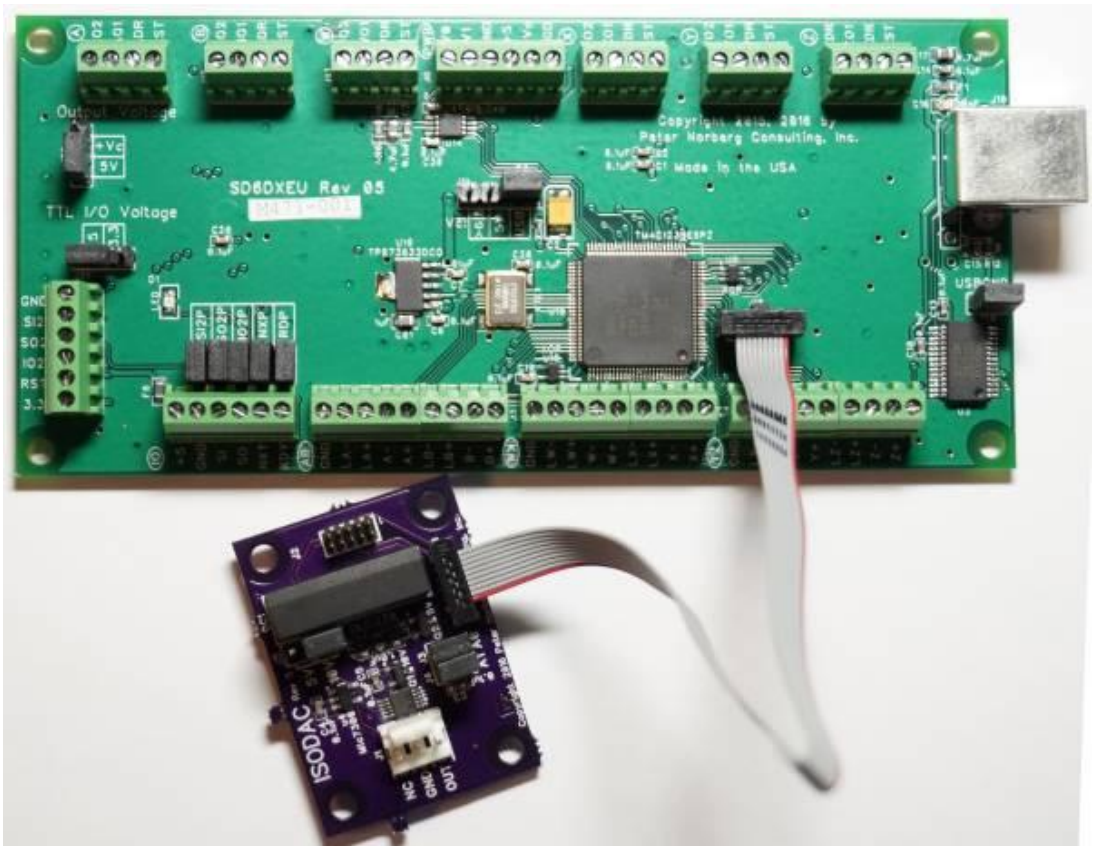
0.125, 0.125 1.375, 0.125
0.125, 0.975 1.375, 0.975

The Amp (or screw-terminal) connector provides the 0-5 or 0-10 volt signal. The signals are defined as:

Signal	Description
NC	<no connection>
GND	Ground reference. This is the reference from which the voltage is generated; it does NOT need to be ground, but it is the base from which the 0-10 volt signal is generated
OUT	0-5 or 0-10 volt output signal (as defined by the board jumper), relative to the GND signal.

As noted in the above table, you need to have a good ground reference for the system to work correctly. The "GND" signal on the ISODAC board is to be connected to the appropriate point in your device that is to be used as a base reference for the 0-5/10 volt output signal. The output is strictly relative to this value.

The OUT line is the actual 0-5 or 0-10 volt output as generated by the board and your software. It is relative to the GND signal on the connector.



The JTAG cable from the ISODAC board needs to be plugged into the matching connector on the BC6D20/BC6D20/SD6DX board. On the ISODAC board, the cable needs to be installed as shown in the above two images: there are two rows of connectors involved, and you must make certain that both rows are used by the connector and that the red part of the cable (pin 1) is positioned as shown in the above images. Note that the cable connector may have a tab on it (some cables have them, some do not): if there is a tab, position the cable such that the tab points towards the nearest edge of the board on the ISODAC unit.

When connected to the BC6D20/BC6D25/SD6DX board, the red part of the cable needs to be to the right (labeled as pin 1 on the connector) when the USB connector is also on the right, and the tab part of the cable should be on the connector side that is towards the center of the board (as shown above). As with the ISODAC, make certain that all 10 pins are correctly inserted into the connector! It is easy to shift the connection by 1 row on the current artworks, which will result in the ISDAC system not working. Observe that the connector tab (if present) is directed towards the center of the BC6D20/BC6D25/SD6DX board.

Although the daisy-chain system can accept up to 4 ISODAC boards, if more than one board is connected to a given BC6D20 or BC6D25 controller then you will probably need to make certain that the BC6D20/BC6D25 board logic is powered off of an external power supply instead of just being powered off of the USB system.

Using the product with the BC6D20/BC6D25 and SD6DX controllers

The ISODAC is supported by several commands in firmware version 5.33 and later for the BC6D20/BC6D25/SD6DX products. Configuration of the voltage scales to be used for each board is done via 4 bits in the 't' command (SetIOConfig), while setting the voltage itself is performed via the $J (SetExtDac, SetExtDacInMillivolts, SetExtDacInVolts) command.

The A0 and A1 jumpers are used to identify the board to the motor controller. In the following table, "IN" means the jumper is installed, "OUT" means that it is removed:

A1	A0	Board
IN	IN	0: addressed as 0, 1 in $J command
IN	OUT	1: addressed as 2, 3 in $J command
OUT	IN	2: addressed as 4, 5 in $J command
OUT	OUT	3: addressed as 6, 7 in $J command

Each board has three "addresses" associated with it:

- The board number (0-3) is used to identify the board in the 't' command
- The board address defined by the board number * 2 is used to set the DAC without memorizing the value for the next power cycle event
- The board address defined by ((the board number * 2) + 1) is used to set the DAC and to memorize the setting for use the next time the board is powered up

The following sections are lifted and edited from the BC6D20NCRouter manual.

The two commands "$J" and "t" are direct serial commands that are part of the normal command set for the firmwares for the BC6D20, BC6D20 and SD6DX boards.

$J – Set external ISODAC voltages (SetExtDac, SetExtDacInMillivolts and SetExtDacInVolts)

On all of the six axis products, an external board called the "ISODAC" may be connected via the 10 pin header in the middle of the board. This board provides for a 0-5 or 0-10 volt isolated DAC output, suitable for use in such items as spindle motor control.

As with the "J" command, the firmware looks for a period (".") in the value sent with the 'J' command. If there is no period, then the units are millivolts. If there is a period, then the units are volts.

The command is comma separated based: the values terminated by commas specify which dac to set (and whether to memorize the dac setting), while the value just before the $J specifies the voltage.

The mapping of the dac address value is:

Address	**ISODAC Board #**	**Use**
0	0	Just set the dac
1	0	set dac and memorize value for the next power cycle
2	1	Just set the dac
3	1	set dac and memorize value for the next power cycle
4	2	Just set the dac
5	2	set dac and memorize value for the next power cycle
6	3	Just set the dac
7	3	set dac and memorize value for the next power cycle

For example,

```
0,3,2000$J
```

would set the dacs on boards 0 and 1 to 2 volts, and would memorize that setting on board 1.

NOTE: Please see the 't' command for information on how to configure the scale used in this function (whether the board is set up as 0-5 or 0-10 volt output).

The "set dac and memorize value for the next power cycle" setting means that the DAC actually memorizes the value for use the next time that it is powered on. Additionally, the value gets saved in an array that will be saved to the board's flash memory if you then tell the board to "memorize settings" (the "-123456789e" command). If a value is memorized at the DAC level without the board also being told to memorize settings, the DAC will only get the new memorized value when it is first powered on (not whenever the board resets).

t – Configure Special Features (SetIOConfig)

… <remaining bits not documented here; please refer to the BC6D20NCRouter manual>

Bits 11-14: Define the voltage range for optional ISODAC boards

These four bits are used to inform the firmware as to the jumper setting on your external ISODAC boards. The firmware needs to know whether your boards are set up as 0-5 volt (the default) or 0-10 volt systems, in order to correctly scale the output request values in the $J command. If a given bit is CLEAR, then the board jumper is set for 0-5 volt operation. If it is SET, then the board jumper is set for 0-10 volt operation.

If you do NOT correctly configure this feature to match your ISODAC board setting, then your DAC voltage will be a factor of 2 off of what it should be!

Bit	Value if set	ISODAC board
11	(+2048)	ISODAC board 0 jumper set to 10 volts if set, set to 5 volts if clear
12	(+4096)	ISODAC board 1 jumper set to 10 volts if set, set to 5 volts if clear
13	(+8192)	ISODAC board 2 jumper set to 10 volts if set, set to 5 volts if clear
14	(+16384)	ISODAC board 3 jumper set to 10 volts if set, set to 5 volts if clear

The following section defines the calls as used in the on-board scripting language to access the ISODAC product.

The BC6D20, BC6D25 and Sd6DX products share a scripting language that allows the board to operate using high level commands. This section documents the portion of that language that supports the ISODAC system.

SETEXTDAC(<idDac>, <Value>) sets selected external ISODAC to the requested value

SetExtDac is used to set the requested external ISODAC to the indicated value on the any of our products.

<idDac> is mapped as:

0	ISODAC Board 0, set value, do not memorize
1	ISODAC Board 0, set value, memorize it for next power cycle
2	ISODAC Board 1, set value, do not memorize
3	ISODAC Board 1, set value, memorize it for next power cycle
4	ISODAC Board 2, set value, do not memorize
5	ISODAC Board 2, set value, memorize it for next power cycle
6	ISODAC Board 3, set value, do not memorize
7	ISODAC Board 3, set value, memorize it for next power cycle

The dac value units are either millivolts or volts, depending on the parameter passed. If the parameter is integer, the dac value is expressed in millivolts. If it is floating point, then the value is expressed in volts.

For example,

SetExtDac(0, 400)

SetExtDac(2, 0.4)

Would actually set both dacs to 0.4 volts.

The "set dac and memorize value for the next power cycle" setting means that the DAC actually memorizes the value for use the next time that it is powered on. Additionally, the value gets saved in an array that will be saved to the board's flash memory if you then tell the board to "memorize settings" (the "EEPromSaveCurrentSettings(-123456789)" command). If a value is memorized at the DAC level without the board also being told to memorize settings, the DAC will only get the new memorized value when it is first powered on (not whenever the board resets).

SETEXTDACINMILLIVOLTS(<idDac>, <Value>) sets selected external ISODAC dac(s) to the requested value

SetExtDacInMillivolts is used to set the requested external ISODAC to the indicated value on the any of our products.

<idDac> is bit mapped as:

0	ISODAC Board 0, set value, do not memorize
1	ISODAC Board 0, set value, memorize it for next power cycle
2	ISODAC Board 1, set value, do not memorize
3	ISODAC Board 1, set value, memorize it for next power cycle
4	ISODAC Board 2, set value, do not memorize
5	ISODAC Board 2, set value, memorize it for next power cycle
6	ISODAC Board 3, set value, do not memorize
7	ISODAC Board 3, set value, memorize it for next power cycle

The dac value units are millivolts for this call.

For example,

SetExtDacInMillivolts(3, 400)

Would set ISODAC board 1 to 400 millivolts, and memorize the value.

The "set dac and memorize value for the next power cycle" setting means that the DAC actually memorizes the value for use the next time that it is powered on. Additionally, the value gets saved in an array that will be saved to the board's flash memory if you then tell the board to "memorize settings" (the "EEPromSaveCurrentSettings(-123456789)" command). If a value is memorized at the DAC level without the board also being told to memorize settings, the DAC will only get the new memorized value when it is first powered on (not whenever the board resets).

SETEXTDACINVOLTS(<idDac>, <Value>) sets selected external ISODAC dac(s) to the requested value

SetExtDacInVolts is used to set the requested external ISODAC to the indicated value on the any of our products.

<idDac> is bit mapped as:

0	ISODAC Board 0, set value, do not memorize
1	ISODAC Board 0, set value, memorize it for next power cycle
2	ISODAC Board 1, set value, do not memorize
3	ISODAC Board 1, set value, memorize it for next power cycle
4	ISODAC Board 2, set value, do not memorize
5	ISODAC Board 2, set value, memorize it for next power cycle
6	ISODAC Board 3, set value, do not memorize
7	ISODAC Board 3, set value, memorize it for next power cycle

The dac value units are volts for this call.

For example,

SetExtDacInVolts(0, 2.5)

Would set ISODAC board 0 to 2.5 volts.

The "set dac and memorize value for the next power cycle" setting means that the DAC actually memorizes the value for use the next time that it is powered on. Additionally, the value gets saved in an array that will be saved to the board's flash memory if you then tell the board to "memorize settings" (the "EEPromSaveCurrentSettings(-123456789)" command). If a value is memorized at the DAC level without the board also being told to memorize settings, the DAC will only get the new memorized value when it is first powered on (not whenever the board resets).

SETIOCONFIG(<Config>) configures special board features

The SetIOConfig function is used to configure the RDY and NXT lines for special operations, SI2/SO2 for TTL-serial, and encoder enables for the two on board encoder systems.

Please see the "t" command for details of the format of the <Config> parameter to this function.

For example, to set ISODAC board 0 as a 5 volt output, with ISODAC board 1 as a 10 volt product (with all other features of the board at their power-on defaults) issue the command

SetIOConfig(4096)