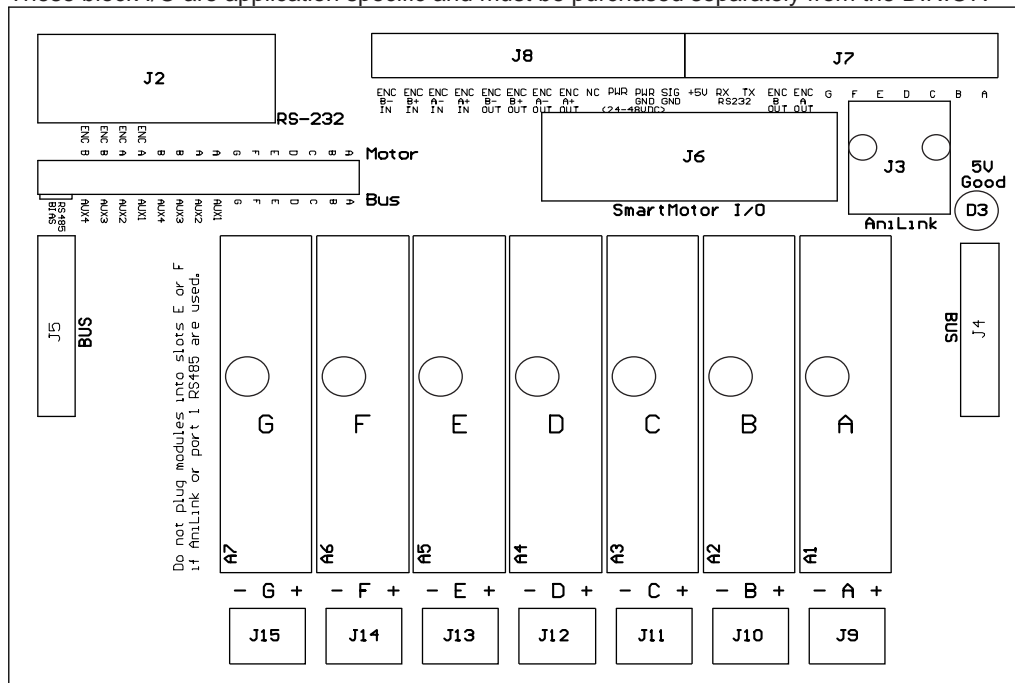


Overview

The DINIO7 is a DIN-rail mounted peripheral for the Servida Motor versions 3 and 4 product lines. Hereafter, the Servida Motor version 3 and 4 products will simply be referred to as the Servida Motor.

Wiring Input/Output up to your Servida Motor is made simple with the DINIO7 breakout board. The DINIO7 has slots for 7 Industry standard Opto Modules. This is a DIN rail mount adapter that allows the Servida Motor to easily interface to popular I/O blocks like those produced by manufacturers such as Gordos, Grayhill, Opto-22 and others. It can be used with either input or output modules.

These block I/O are application specific and must be purchased separately from the DINIO7.



The DINIO7 has four primary functions:

- Servida Motor interface connections
- Servida expansion bus
- Block I/O sockets
- Optional differential encoder interface

Mfg. I/O module Cross Ref. chart for DINIO7 Breakout

All modules listed are 5VDC logic side for Servida Motor compatability

All modules have LED indication

All Output modules have replacable fuses

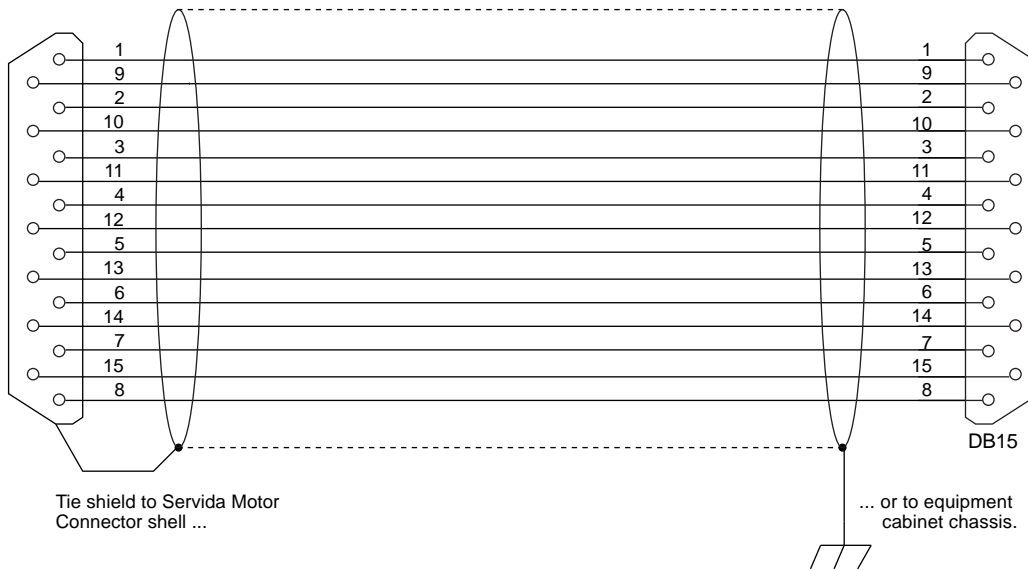
Mfg.	DC Input	DC Output	AC Input	AC Output
Opto-22	G4-IDC5	G4-ODC5	G4-IAC5	G4-OAC5A
Grayhill	70G-IDC5	70G-ODC5A	70G-IAC5A	70G-OAC5A
Crydom	X4IDC5	X4ODC5	X4IAC5	X4OAC5
Gordos/Crouzet	C4-IDC5	C4-ODC5	C4-IAC5	C4-OAC5
	(3-32VDC high side)		(120VAC high side)	

Servida Motor Interface

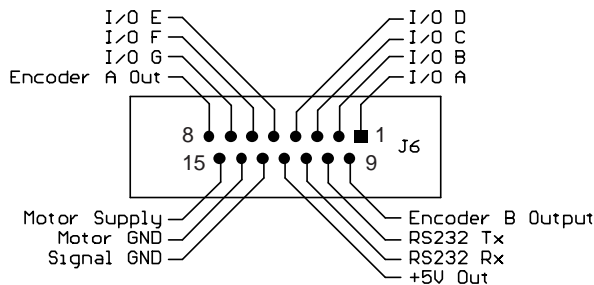
The DINIO7 provides two means of connecting to the Servida Motor – a 15 pin D subminiature connector and a bank of standard pitch screw terminal blocks. The 15 pin D subminiature connector pinout is identical to that of the I/O connector on the Servida Motor D series. Thus, you need only to use a male-male 1-to-1 pinned 15 pinned D subminiature connector to fully hook up the DINIO7 to the Servida Motor. These cables can be purchased from various companies. As the cable will be carrying 5V TTL level signals, Servida recommends that you use shielded cables. The shield is typically terminated to the metal connector shell at the Servida Motor or the equipment cabinet.

As the cable will be carrying 5V TTL level signals, Servida

The shield is typically metal connector shell at the Servida Motor or the equipment cabinet.



For reference, the pinout of the DINIO7's 15 pin D subminiature connector is

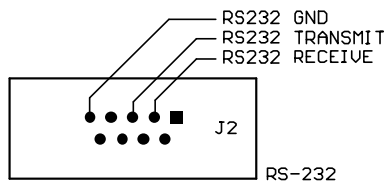


The second Servida Motor interface on the DINIO7 is the bank of standard pitch screw terminal blocks. From right to left, these connections are:

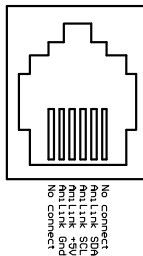
DINIO7 Terminal Block Pinout

Signal	Description
A	Digital input or output A / Encoder input A / Step input
B	Digital input or output B / Encoder input B / Direction input
C	Digital input or output C / Positive Limit input
D	Digital input or output D / Negative Limit input
E	Digital input or output E / RS-485 + / AniLink
F	Digital input or output F / RS-485 - / AniLink
G	Digital input or output G / Sync
ENC A OUT	Encoder A output from Servida Motor
ENC B OUT	Encoder B output from Servida Motor
RS232 TXD	RS-232 transmit from Servida Motor
RS232 RXD	RS-232 receive to Servida Motor
+5V	+5V, 210 mA source from Servida Motor
S16 GND	Signal ground, tied to PWR GROUND within Servida Motor
PWR GND	Power ground, tied to GROUND within Servida Motor
PWR	20 to 48VDC input power to Servida Motor
NC	No connection
ENC A+ OUT	Optional differential encoder A+ output from Servida Motor
ENC A- OUT	Optional differential encoder A- output from Servida Motor
ENC B+ OUT	Optional differential encoder B+ output from Servida Motor
ENC B- OUT	Optional differential encoder B- output from Servida Motor
ENC A+ IN	Optional differential encoder A+ input from Servida Motor
ENC A- IN	Optional differential encoder A- input from Servida Motor
ENC B+ IN	Optional differential encoder B+ input from Servida Motor
ENC B- IN	Optional differential encoder B- input from Servida Motor

When all of the connections on the 15 pin I/O connector are carried back to the Servida Motor, the user has direct access to the AniLink and channel 0 RS-232 port from the DINIO7. Serial communication is accessible through the connector labeled "RS-232":



Similarly, the AniLink network is accessible through the RJ-6 modular connector. Its pinout is given here:



The ratings for the interface to the Servida Motors are given below. Absolute maximum ratings are levels beyond which damage may occur.

Absolute maximum voltage:	5.5VDC
Absolute minimum voltage:	-0.3VDC
Absolute maximum current sourced/sunk by Servida Motor:	30 mA
Recommended operating voltage (source or sink):	4.75-5.25VDC
Valid logical high signal:	2.0VDC
Valid logical low signal to Servida Motor:	0.4VDC
Valid logical low signal from Servida Motor:	0.8VDC

Block Input/Output Sockets

The DINIO7 has 7 block I/O sockets, one for each of the 7 general purpose I/O on the Servida Motor. These sockets are compatible with both 24VDC and 110VAC I/O modules from Opto-22, Crouzet and Grayhill. When issuing commands from the Servida Motor, note that the logical interface to the I/O block is active low. That is, a port value of 0 would turn an output block, on closing its contacts. If you read the value of a port that is connected to an input block, a value of 0 would mean that there is current flowing through the I/O block.

At power up, the Servida Motor I/O ports are configured as an input pulled high. In order to close the contacts on an I/O block, you must both configure the Servida Motor port as an output and set it low.

Suppose you have a 24VDC output block in port A. If you want the I/O block contacts to close, you must first define port A as an output by using the UAO command. You then set port A to a logical low state by using the UA=0 command. Similarly, if you needed to activate the relay on port B, you would use the UB=0 command. If you want to open the contacts, you would issue the UB=1 command.

Similarly, if you had an input block, you would generally read its state by stuffing its value into a variable. For example, if you had an input block in port C, you would stuff its value into the variable g by issuing the command g=UCI.

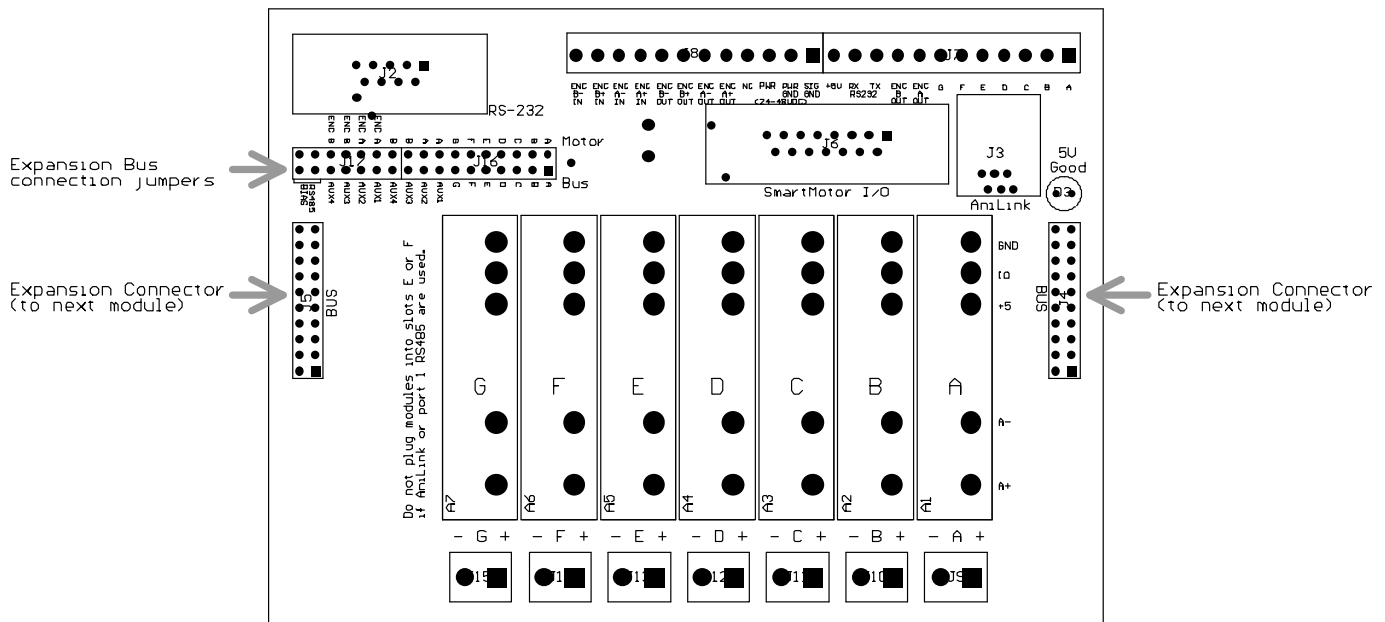
You can also use the port value directly without assigning port value into a variable, but this has limited utility and may lead to confusing timing interaction within the program - so you have been warned. It is interesting to note that you can issue this command regardless of whether the port is configured as an input or an output. For further details, please refer to the Servida Motor User's Guide.

Please note that, due to the multi-functional nature of the Servida Motor I/O, some features are lost when you use I/O blocks. For example, ports E and F can be used as digital I/O, as well as RS-485 and AniLink. If you use either the RS485 or AniLink functions, do not plug an I/O block into either of the slots corresponding to ports E and F.

When issuing commands from the Servida Motor, note that the logical interface to the I/O block is active low.

Servida Expansion Bus

The Servida Expansion Bus provides a convenient way to connect several Servida Motors and their DIN-rail mount expansion modules without using any additional cable. Every Animators DIN-rail mount expansion module has two expansion bus connectors for receiving signals from modules on either end of it. The bus passes through the module without any direct connection to any Servida Motor I/O or expansion module function. To make use of the bus, it has to be connected to some Servida Motor I/O or expansion bus card through the jumpers.



For example, let us suppose that your machine has six Servida Motors and you have a DINIO7 connected to each. If you wanted to connect port A on Servida Motors #1 and #4 together, you do not have to run a cable between the two motors. Instead, you would install the expansion bus bridge and the jumpers marked User A on their respective DINIO7s.

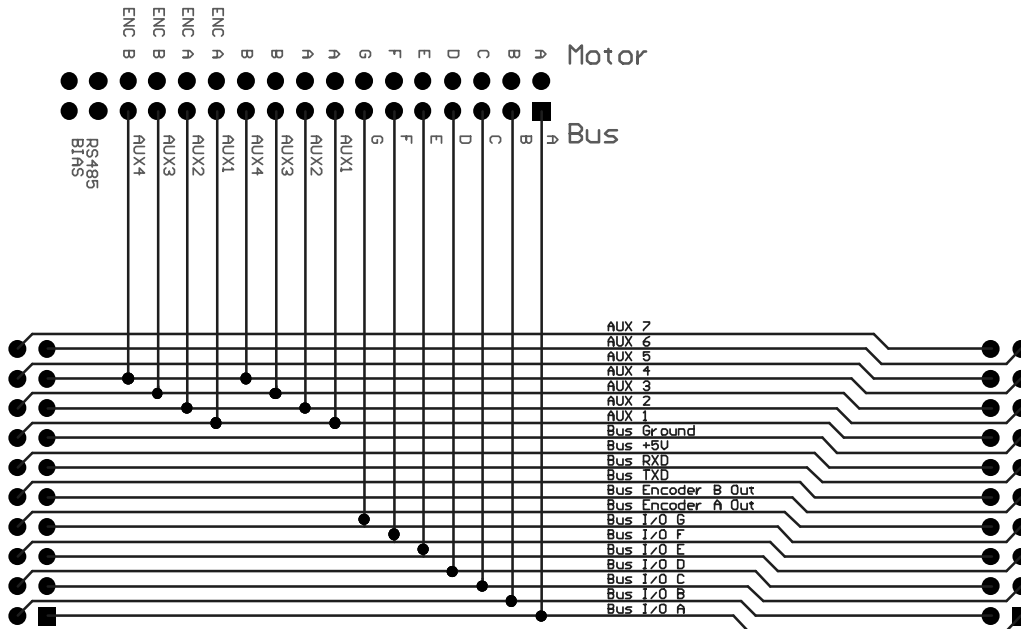
The Expansion Bus also allows for certain specialized connections. Every Expansion Bus is equipped with 7 auxiliary lines that allow you to connect the Servida Motor I/O in various ways. The DINIO7 allows you to connect the Servida Motor I/O A, B and encoder outputs to auxiliary connections on the Expansion Bus. This allows you to hook up Servida Motors in a mode follow (gearing) configuration without using any cables.

Specifically, to hook up two Servida Motors for mode follow, simply set the following jumper connections:

Master motor: Motor ENC A to Bus AUX 1
 Motor ENC B to Bus AUX 3

Slave motor: Motor A to Bus AUX1
 Motor B to Bus AUX3

A schematic of the Servida expansion bus and the DINIO7 jumpers is shown below:

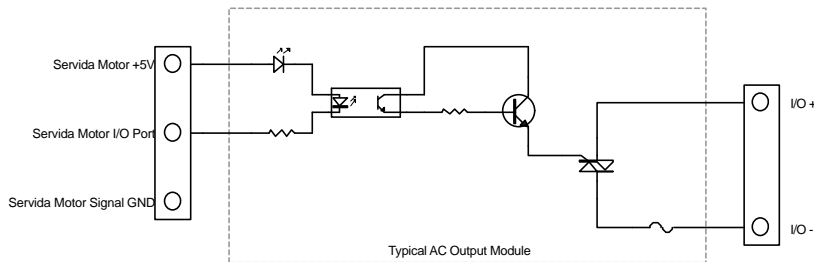
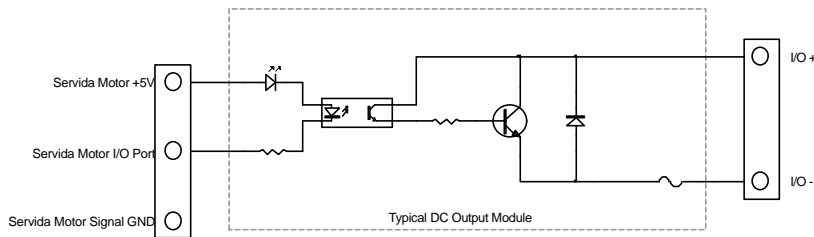
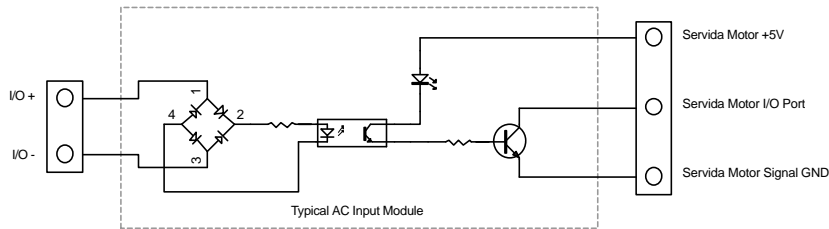
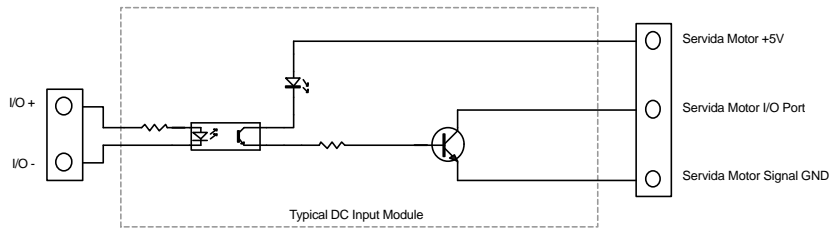


Optional Differential Encoder Interface

If your DINIO7 is equipped with the optional electronics to interface with differential encoder I/O, these signals can be interface through the Servida expansion bus, as well as the screw terminal expansion bus. These signals are not isolated from the Servida Motor, so the user is cautioned to avoid ground loops.

Application

The DINIO7 is intended to provide a convenient interface to 24VDC and 110VAC block I/O modules. These block I/O modules come in four forms: DC input, DC output, AC input and AC output. The diagrams below show the functional representation of an example of each.



These are given for reference only. Your specific I/O module may or may not function as shown in these diagrams.

You control the I/O in the same way as if you were using the local Servida Motor I/O. For example, suppose that, in our example, you want to engage relay #3 for 0.1 second. Noting that relay #3 is wired to Servida Motor Port D, you would need to set up port E as an output with default high, and turn it on by commanding the output low:

```
UD=1      `Before establishing that port E is an output,
           `predefine it to be default high
UDO       `Define port E as an output
UD=0      `Turn on relay by setting port E low
WAIT=407  `Delay 0.1 second
UD=1      `Turn relay off by setting port E high
```

To read an input, you can either stuff its value into a variable or read it directly. Suppose that you wanted to read the output #1 from the PLC. Note that output #1 is connected to Servida Motor port B.

You would typically set up port B as an input and then assign the port value to a variable – let's say we use variable bb:

```
UBI       `Setup port B as an input
bb=UBI    `stuff the port value of B into the variable bb
```

You can also read the port directly. This is generally used only when you are using it in an evaluation expression, such as IF and WHILE. For example, if want to do something if port A is low, the IF statement would be:

```
IF UAI
.
.
.
ENDIF
```

A while statement would look like:

```
WHILE UAI
.
.
.
ENDIF
```

For further details about the IF and WHILE statements, please refer to the Servida Motor Users Guide.