

# Connecting UniOP to Galil/Yaskawa controllers

This Technical Note contains the information needed to connect UniOP operator panels to Galil's DMC series controllers and Yaskawa's SCM series controllers, using the serial communication protocol.

The Galil/Yaskawa driver is associated to the Designer file D32UPLC082.DLL.

## Contents

1	Introduction .....	3
2	Controller Setup .....	3
3	Data Fields and Command Macros .....	3
4	RDA, Alarms and Mailbox .....	4

---

**Tn084**

**Ver. 1.00**

**© 2004 Sitek S.p.A. – Verona, Italy**

Subject to change without notice

The information contained in this document is provided for informational purposes only. While efforts were made to verify the accuracy of the information contained in this documentation, it is provided “as is” without warranty of any kind.

[www.exor-rd.com](http://www.exor-rd.com)

## 1 Introduction

To create a Designer project for connection to one of Galil or Yaskawa controllers, select the driver “Galil/Yaskawa” from the list of available communication drivers in the ‘Configure Controller’ dialog box.

---

*Note: In the case the Galil/Yaskawa driver does not appear in the list, make sure the D32UPLC082.DLL file is present in the Designer installation folder and then press the Refresh button.*

---

Connection to the controller is made via either a CA37 cable on the main communication port, or cable CA36 that includes hardware handshaking, necessary with the newer controllers with a USB or Ethernet port.

## 2 Controller Setup

Communications should be set to 19200 baud, no parity, one stop bit, and echo on. To turn echo on in the controller, send the command EO1 then BN to save the controller parameters.

In the controller setup window, a selection is available for the use of hardware handshaking in the driver. CA36 has to be used when the “use RTS/CTS” is selected. Older project files and connections to a controller using the CA37 with no handshaking will work without any modification as the default controller setup is not using the RTS/CTS control signal. Please see figure below:

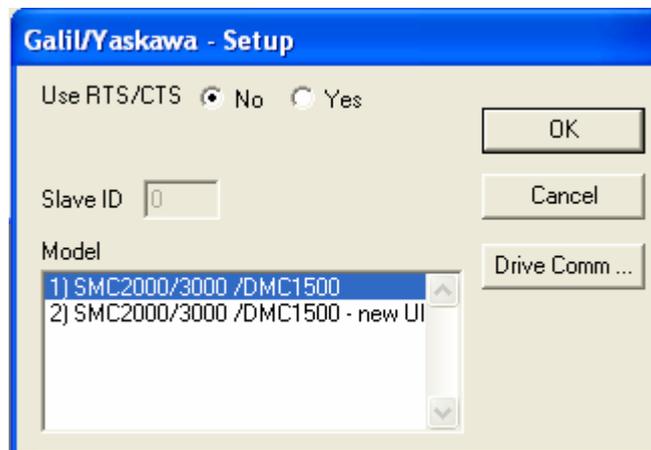


Figure 1

## 3 Data Fields and Command Macros

Using the Galil/Yaskawa driver, the user is able to monitor and modify variables and arrays, as well as send commands to the controller.

However, there are some restrictions placed on variables and arrays. For the panel to be able to monitor and modify variables, **the name of the variable must be defined as Vnnn** (where nnn is a three digit number). For example, V001 is a valid variable name, whereas V1 is not valid. The Vnnn variable can be used in the controller program to change parameters, as well as to change program flow. Arrays are used in a similar manner and can be dimensioned using a *DM* statement in the motion control program. They must be defined as **UNIOP[zzzz]**, where zzzz can be any number ranging from 0 to 4999.

Variables and arrays in the controller can have data in the form x.y (where x is ten digits and y is four digits). **UniOP will only operate on the integer range and disregard the fractional part of the form.**

The *Send Command* macro in the *Keyboard Macro Editor* can be used to program keys or touchcells to issue commands to the motion controller. Refer to the controller manuals for valid command strings. Please see picture below:



Figure 2

## 4 RDA, Alarms and Mailbox

Variables and UniOP array are the only available data types for the RDA, alarms and the Interlock mailbox.

The interlock mailbox when used with with 32bit based controllers like the DMC/SMC models needs to be referenced to a Word. The structure of the mailbox is word oriented and will require some special data manipulation in the controller to work correctly. Basically the mailbox structure is as below:

Offset	
0	- Status word
1	- Command/Response word
2	- Parameter 0
3	- Parameter 1
4	- Parameter 2 ....

It will be necessary in the controller to combine the Status word and the Command/Response word, then Parameter 0 and Parameter 1 ...

For further information on using the interlock mailbox and a description of the Function codes and parameters, please refer to Designer help.

To enable the Mailbox, in Designer, go to Projects, Interlock Mailbox and click on reference. Please see figure below:

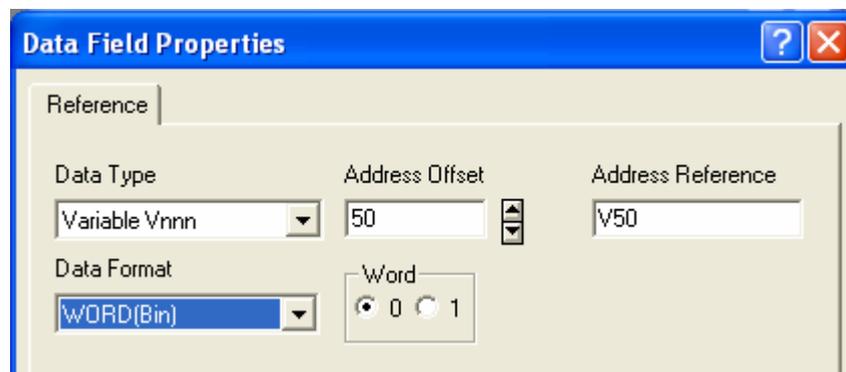


Figure 3

Make sure to select Word as the data format and Word 0 for the reference.

Since the Variables in the controller are defined as 32bit Integers, the Command/Status word of the mailbox will be the first 16 bits of **V050**, the function/response code being the last or most significant 16 bits.

Up to twenty 16-bit words can be used by the mailbox, so in the controller, the user should pre-define and set aside 10 variables, such as "**V050**" through "**V059**" or **UniOP[0]** through **UniOP[9]**.

As an example, let use the mailbox to download a recipe, with the mailbox reference being **V050**. For a recipe download, the Command code is 2, with Parameter 0 as the page number for the recipe and Parameter 1 the recipe set number. To execute the download, a value of 1 needs to be placed in the Status word.

To download recipe set 4 on page 7, we need to input a value of 262151 (hex **40007**) in **V051** (4 shifted 16 bits to the left, plus 7. In binary: 100 0000 0000 0000 0111)

And the value of 131073 in **V050** (hex **20001**)

(2 shifted 16 bits to the left, plus 1. In binary: 10 0000 0000 0000 0001)

*Note: To set a hex value in the SCM3010, you would use the "\$" symbol. Example: "**V050=\$40007**"*

If the mailbox executes correctly, a value of 4 will be returned in **V050**. If not, a list of error codes is available in the manual for each command.

## Appendix A. Communication Error Codes

Current communication status is displayed in the System Menu of the UniOP.

A message and a numeric error code describe the error status.

The message reports the current communication status. The number shows the code of the current communication error or, if the communication is correct, the code of the last error encountered. When the error code 0 is shown, it means there have been no communication errors since this system start-up.

<b>Code</b>	<b>Description</b>	<b>Notes</b>
<b>00</b>	No error	There are no communication errors and there have been no errors since start-up.
<b>04</b>	Negative ACK	NAK returned by the controller.
<b>05</b>	Time out	No response received from the controller within the timeout period.
<b>06</b>	Response error	Unexpected/unrecognized response from the controller
<b>07</b>	General communication error	General unknown communication error.
<b>11</b>	Line Error	Parity error or similar