

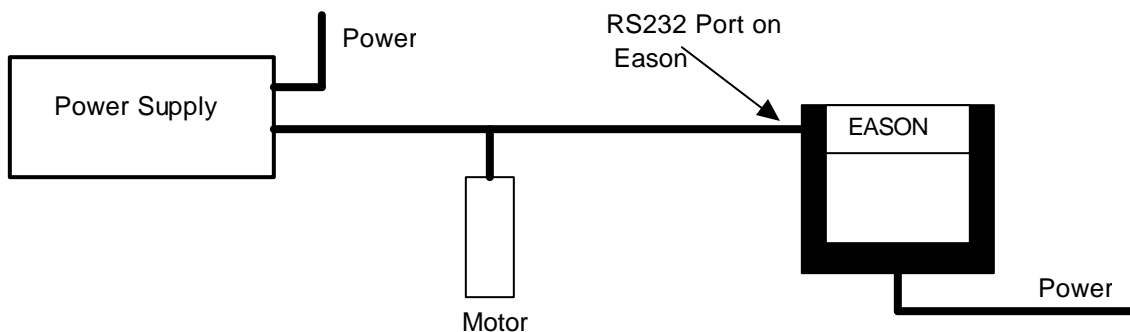
**Suggested Application Development Process:**

1. Use the included demo program with WinBuild 2000: in the c:\program files\winbuild 2000\examples directory.
2. Compile & download the demo program into the Eason.
3. Connect Eason to Quicksilver to verify basic communications.
4. Tune your system using the Quicksilver development software, and create your motion control program. Most aspects of motion critical tasks should be written and debugged here- feedback, control, I/O, etc.
5. Identify variables in the Quicksilver program you would like the Eason HMI to display/monitor/modify.
6. Now expand your initial simple Eason 2000 program- use the following guide to communicate these variables to/from the Quicksilver.
7. Create the rest of your HMI application with WinBuild 2000. Use the initial program as a base, and add your HMI and control tasks.

Hardware Setup:

The Quicksilver motion controllers communicate to the Eason 2000 Family products via serial connection. All 2000 Family products have at least two ports available for RS232 communication, and those with expansion options can be fitted with more serial connections.

Hardware connections between the Eason & Quicksilver motion controller should consist of just a simple RS232 cable. Use the RS232 cable included with the Quicksilver motor for easy hookup. Either communication port on the Eason is available for RS232 use.



Software Setup:

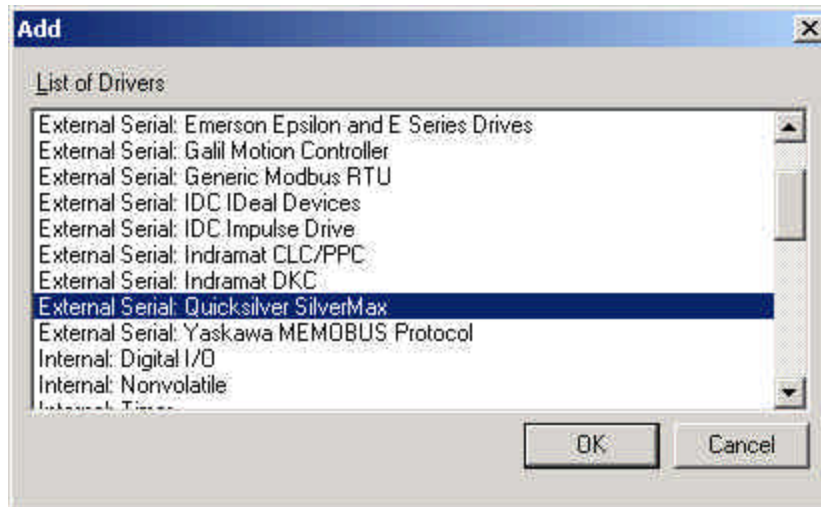
See the example program “quicksilver.oip” listed in the ‘c:\programfiles\winbuild2000\examples’ directory.

Or,

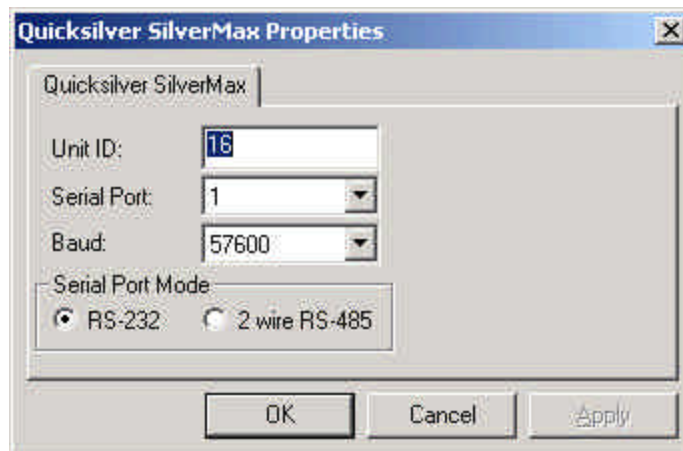
Start a new Winbuild 2000 project, and add the Quicksilver driver.

SETUP ▶ PROJECT ▶ DRIVERS

Click “ADD” and scroll down and select the “Quicksilver Silvermax” driver.



Once the driver has been added, click on the Silvermax driver listing, and click on the "Properties" button.



Here you can set up the Eason to communicate to the Quicksilver motor. Set the Motor ID, serial port, baud rate and RS232/485 mode.

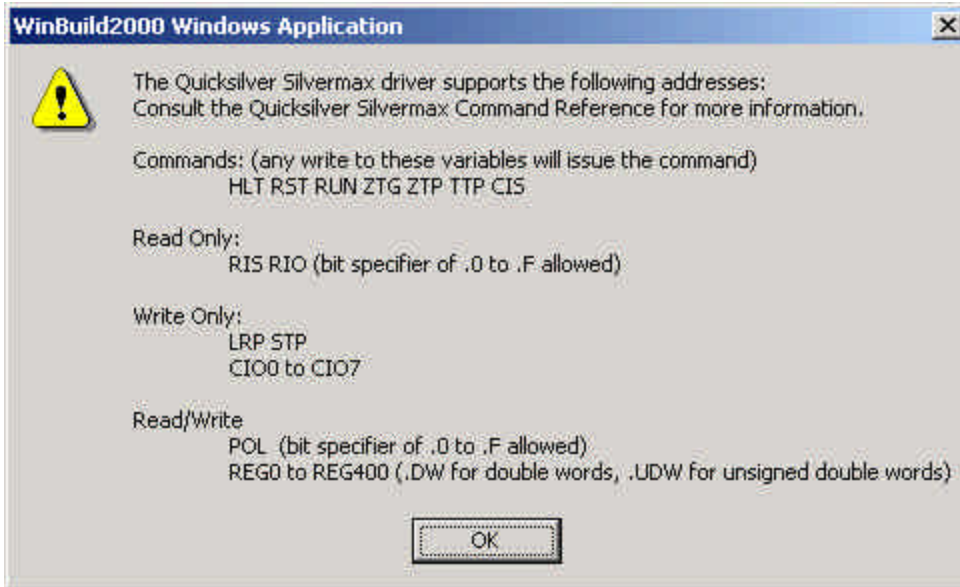
Creating Variables:

Now set up variables from the Quicksilver you would like access to.

In Winbuild 2000, go to the tag table by selecting SETUP ► TAGS.

Click the "New Tag" button to insert a new tag into the project. Give it a name, and source it to the Quicksilver driver.

In the ADDRESS column type one of the following addresses/parameters in the Quicksilver you can access easily with the driver:

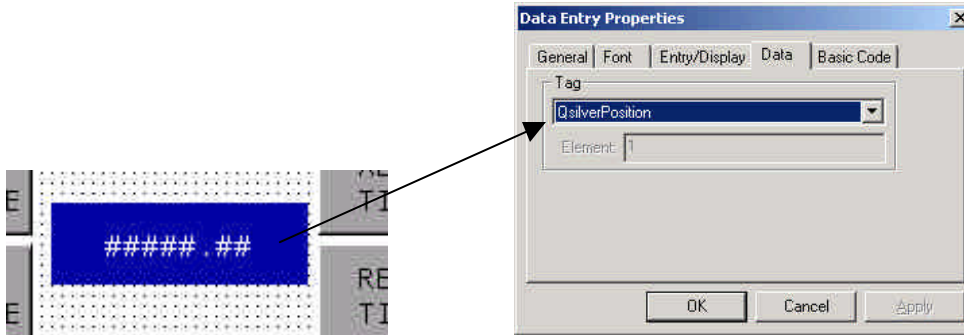


For example, below are sample tags of how to access the POL or STP command in the Quicksilver.

Tag Table					
	Tag Name	Source	Address	Type	Default
1	QSLVRRUN	Quicksilver SilverMax [16] on 1	RUN	Byte	0
2	QsilverPoll	Quicksilver SilverMax [16] on 1	POL	Unsigned Word	0
3	QsilverIO	Quicksilver SilverMax [16] on 1	RIO	Unsigned Word	0
4	QsilverRestart	Quicksilver SilverMax [16] on 1	RST	Byte	0
5	QsilverStop	Quicksilver SilverMax [16] on 1	STP	Double Word	0
6	QsilverHalt	Quicksilver SilverMax [16] on 1	HLT	Byte	0
7	QsilverStatus	Quicksilver SilverMax [16] on 1	RIS	Unsigned Word	0
8	QsilverPosition	Quicksilver SilverMax [16] on 1	REG1.DW	Double Word	0
9	QsilverCIS	Quicksilver SilverMax [16] on 1	CIS	Byte	0
10	JUNK	Internal Volatile	S:	String	0

To access these values, just use the Eason Tag Name wherever you need to in your project. To start a program, just set QSLVRRUN=1. To Halt, just set the variable QsilverHalt=1.

To put the value of position up on the screen, just create a data display box, sourced to the variable QsilverPosition.



For instances where the driver can't access a variable, or to send a specific command to the Quicksilver, you can use the COM macros to send/receive information on the communication port.

Sending commands:

Sending commands to the Quicksilver is easy- just use the PUT COM pseudocodes to send out the ASCII command. For instance, the following command sets I/O point 1 as an output in the Quicksilver: (of address #16) (All commands referenced from actual examples in the Quicksilver documentation.)

```
COM: PUT COM 1 VALUE "@16 188 1 1"
```

Or, this command tells the motor to move 80000 counts relative in the positive direction.

```
COM: PUT COM 1 VALUE "@16 135 80000 10000 53687100 0 0"
```

This is all that is required to send out commands to the motor. The PUT COM pseudocodes automatically add a carriage return <CR> to the end of the transmission. To read responses however requires a bit of work. See the "Reading Responses" section for more.

Reading Responses:

Responses come back from the Quicksilver in slightly different formats, depending on the command that you sent.

"* 10"

The "* 10" indicates that the SilverMax accepted the command. The "*" is the basic response indicator, and the "10" is the Hexadecimal number for the address of the SilverMax motor.

"# 10"

If the command requires a data response from the motor, the response will include a "# 10" followed by a confirmation code of "000C" and then two more higher & lower hex words to include the data.

So for instance when requesting position from the Quicksilver, the command would be:

```
COM: PUT COM 1 VALUE "@16 12 1 "
```

And the response from the motor would be: "# 10 000C 0005 06A3 <CR>"

The carriage return is viewed by the Eason a termination character, and is discarded. When sending a command to the Quicksilver motor, you will need to immediately read in the response from the motor.

To read in a response from the Quicksilver, use the GET COM pseudocode:

```
GET COM "JUNK$"
```

The Eason "Quicksilver" demo program uses a subroutine called "EATCHR" to read in the communication port, and parse out the response characters.

For the example of reading back position information, the following BASIC code takes the response string (JUNK\$), and pulls out the relevant high & low hex words, and converts them to decimal for display onto the screen (variable BRIAN) using the HEX function:

```
IF MID$(JUNK$,6,4)="000C" THEN
  HIGH$=MID$(JUNK$,11,4)
  LOW$=MID$(JUNK$,16,4)
  BRIAN=HEX(HIGH$)*65535+HEX(LOW$)
ENDIF
```

"! 10"

A response of "! 10" from the quicksilver indicates an error in sending a command. This response is usually accompanied by an error value. See the Quicksilver manual for the NAK response codes. (In the example Eason-Quicksilver program, the tag "MESSAGE" has "messaging" set up to display the NAK codes.)

The following part of the example subroutine "EATCHR" checks to see if the return string is a "! 10" and if it is, pulls the error code from the response (JUNK\$). The error code is put into the tag MESSAGE.

```
VALID$=LEFT$(JUNK$,4)

IF VALID$="! 10" THEN
  STATUS=3
  MESSAGE=VAL(MID$(JUNK$,11,4))
ENDIF
```

Troubleshooting:

If you cant communicate:

1. Check Eason COM INIT statement to make sure it matches the settings you have for the Quicksilver.
2. Use Quicksilver software to establish communication to motor to verify port is working.
3. Use Hyperterminal to examine the outgoing strings from the Eason to ensure that they are of proper format.