

RS-232 Communication

All ASI Controllers (TG-1000/Tiger, MS-2000 and RM-2000) have an RS-232 serial link. While the controller can be used in stand-alone mode completely separate from a computer, having a connection so send serial commands is required to utilize all of the controller's abilities.

MS2000 and RM2000 controllers normally have a physical RS-232 serial port (a 9-pin D-sub connector). There is a USB Type A connector on controllers built since ~2008 and on TG-1000/Tiger controllers, which is these days more common to use because USB ports are ubiquitous and more plentiful. It does not make a difference which physical connector is used. For USB connection, the controller includes a chip that emulates a serial com port which requires drivers. Sometimes the operating system will automatically install the drivers for this chip, but if not they can be found at <http://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx>. As of early 2018 there was a bug affecting the "Universal" version of the SiLabs driver when used with the ASI Console programs (depending on .NET) so use the Windows 7/8/10 version 6.x instead.

Table 1: Serial Link Info

	MS2000 and RM2000	TG-1000/Tiger
Baud Rate	9600 (default), 19200, 28800, or 115200 (set by DIP switch)	115200
Data Bits	8	
Parity	None	
Stop Bits	1	
Flow Control	None	

Software

All software control is using the serial commands sent over the serial port or virtual serial port.

Due to the nature of serial ports, only one program can use the serial port at a time. So only one of these softwares can be used at a given time to send commands and receive replies from the ASI controller.

ASI Console and ASI Tiger Console

For MS2000, RM2000 and FW1000 controllers ASI Console can be used to communication via serial port or virtual serial port (USB-serial adapter) to control, configure, and update firmware. More info on Downloading and Setting up ASI console is [here](#).

For TG-1000/Tiger controller, ASI Tiger Console can be used to communication via serial port or virtual serial port (USB-serial adapter) to control, configure, and update firmware. More info on downloading and setting up Tiger console is [here](#). You can send serial commands using the input box at the bottom of the scripting tab once you have it set up.

Another program for interacting with TG-1000 controller is [Tiger Control Panel](#), like the Tiger Console it lets user send serial commands to the controller, however it has a few additional utilities like displaying live axis position and states.

LabView

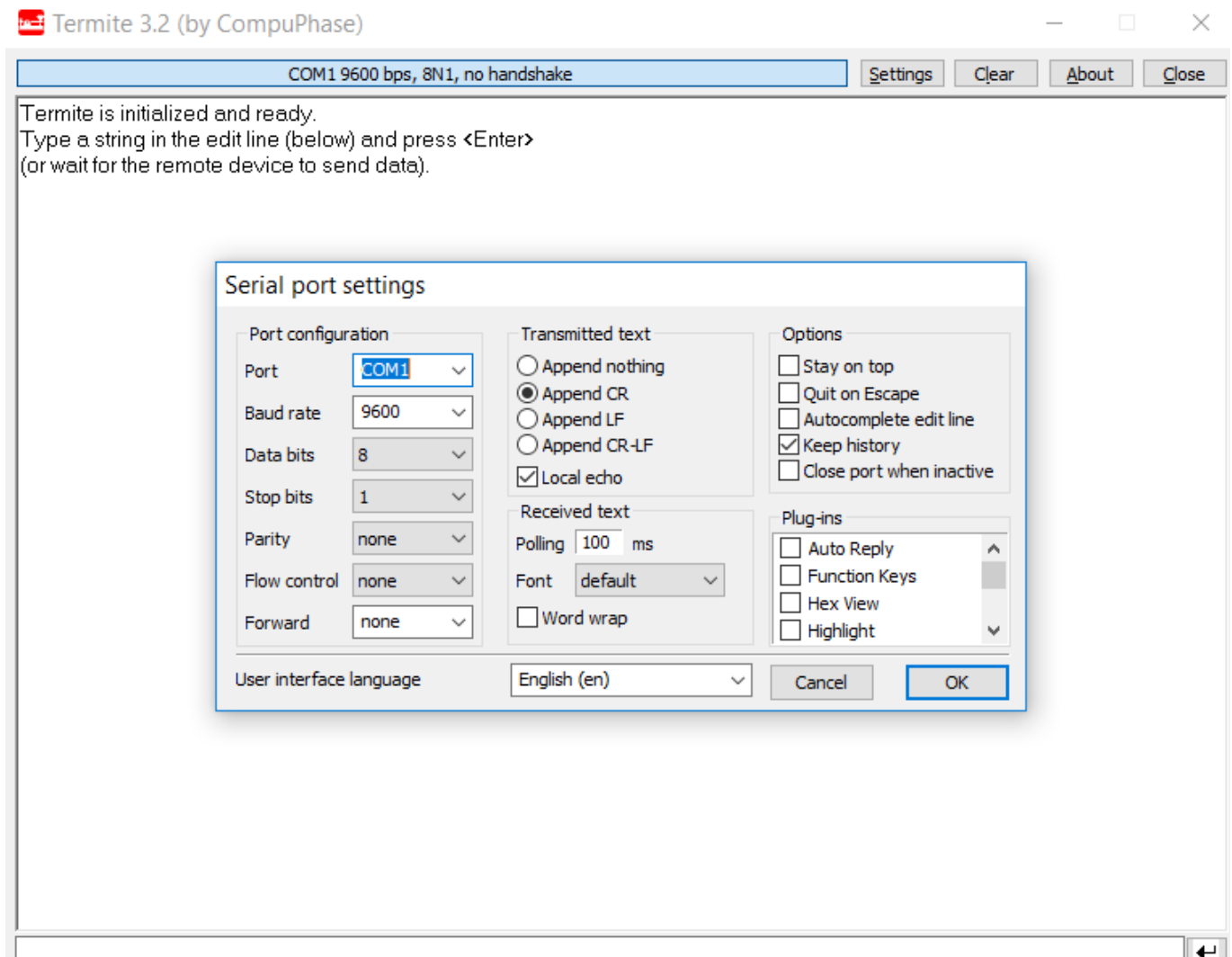
LabView drivers are available from ASI, see the ASI main website under Support → Downloads ([direct link](#)).

Terminal Programs

You can use terminal programs such as Advanced Serial Port Monitor, Termit, TeraTerm, PuTTY, and HyperTerminal to send and receive commands directly from the controller.

Termit

An free and easy to use terminal program is Termit (https://www.compuphase.com/software_termit.htm). Below is a screenshot of the setup dialog and how it should be configured for talking with our controllers (through RS232 or USB):



Note: your *Port* and *Baud rate* setting requirements may be different than shown, e.g. baud rate is 112500 for Tiger controllers.

Advanced Serial Port Monitor

ASI uses [Advanced Serial Port Monitor](#) in-house for serial communications even though it is a paid program after a brief trial period. The main thing it offers that others don't is a "Spy Mode" for monitoring communication done via other programs.

Micro-manager

Micro-Manager is a free open-source microscope control software, and ASI contributes and supports device adapters to utilize ASI hardware.

There are two ways to send serial commands in Micro-Manager, the first is generic for any serial device using FreeSerialPort device adapter and the second is specific for ASI hardware using the ASI device adapter, either ASIStage for MS-2000 and RM-2000 controllers or ASITiger for TG-1000 controllers.

Micro-Manager FreeSerialPort

Create a device using the FreeSerialPort adapter in the Hardware Configuration Wizard. Assign the same com port as your ASI product uses. In the Device/Property browser (MM Tools menu) set the CommandTerminator property to be \r and the ResponseTerminator property to be \n. Then modify the Command property to be whatever command you like and look for the reply in the Response property. You can do this from the Property Browser and/or from your script using the `mmc.SetProperty()` and `mmc.GetProperty()` methods. If the Response property is too long to fully display then try to copy it and paste into another program; \r indicates the start of a new line.

Micro-Manager ASI Device Adapters

If you haven't already, add the ASI controller to your configuration using the Hardware Configuration Wizard. Then use the property SerialCommand and SerialResponse to send commands and view the controller's reply. You can do this from the Property Browser and/or from your script using the `mmc.SetProperty()` and `mmc.GetProperty()` methods. For TG-1000 "Tiger" controllers, this is under the TigerCommHub device. If the SerialResponse property is too long to fully display then try to copy it and paste into another program; \r indicates the start of a new line.

Other Third Party Applications

Finally, proprietary high-level microscope control software which support ASI controllers (e.g. Molecular Devices' Metamorph, Nikon Elements) uses serial commands to communicate with the controller. The communication details are generally hidden from the user.

Command

The controller's instruction set is implemented using the following format:

```
COMMAND X=?????? Y=?????? Z=?????? <Carriage Return>
```

The COMMAND is a string of ASCII characters such as MOVE or HOME, which must be followed by a space. All commands are case insensitive. Many commands have abbreviated versions that help cut down on typing time and serial bus traffic.

Next are the axis parameters. (Bracketed [] parameters are optional.) The axis name is given, followed immediately by an equal sign and the axis parameter value. Each axis must be separated from the one before by one blank space. One or more axes may be specified on a single command line. An axis symbol typed without an = assignment is often assumed to mean =0, but that behavior isn't guaranteed in general (it does, however, work for the commonly-used MOVE and MOVREL commands). Sometimes the command format may not require a parameter value (e.g., INFO X). Commands will accept integer or decimal numbers; internal truncation or rounding will occur if fractional decimals are of no meaning to the command.

axis or [axis] is the placeholder for the axis name, which is a single character. All 26 alphabet characters A-Z can be used as axis names, and the special character * means "all axes" (not including filterwheels). For example, X and Y are typically used for a sample translation stage, Z and F are commonly used for focus axes, and A-D are the default letters for scanner axes. Filter Wheels are designated by numbers, TG-1000 can accommodate up to 10 wheels numbered 0-9.

For Tiger: When [Addr#] appears in the format, then the intended card address must be prepended to the serial command, as the command is Card-Addressed. ... indicates more arguments can be sent with the same command.

All commands are completed with a Carriage Return (ASCII hex code: 0D). The controllers receive ASCII characters one at a time and place them into their memory buffer. With the exception of single hex code commands like the tilde ~, the controller will not process a command in the memory buffer until the Carriage Return <CR> has been received.

Reply

Upon receiving a Carriage Return <CR>, the Controller will process the command stored in its command buffer, clear the command buffer, and return a reply.

MS-2000 Reply Syntax

When a command is recognized, the controller will send back a colon : (hex code: 3A) to show that it is processing the command. When processing of the command is complete, an answer is returned with any requested information, typically beginning with the letter A. In some cases, the answer part of the reply is delayed until the completion of the command. The reply is terminated by a carriage return and a linefeed character <CR><LF>. In the examples below, the <CR> and <CR> <LF> are implied. This programming manual gives examples in the MS-2000 reply syntax unless otherwise

specified.

Examples

Typed commands are in THIS TYPEFACE
Controller replies are in THIS TYPEFACE

```
MOVE X=1234 Z=1234.5 <CR>
:A <CR><LF>
MOVE X Y Z <CR>
:A <CR><LF>
WHERE X <CR>
:A 0 <CR><LF>
MOVE X=4 Y=3 Z=1.5 <CR>
:A <CR><LF>
WHERE X Y Z <CR>
:A 4 3 1.5 <CR><LF>
WHERE Z Y X <CR>
:A 4 3 1.5 <CR><LF>
```

Tiger Reply Syntax

The TG-1000 has two reply syntaxes; the active one is set using the VB F command. The default syntax is backwards compatible with the MS-2000 controller, including all the quirks and inconsistencies between commands. The Tiger syntax is more self-consistent and in some cases more explanatory (e.g. with WHERE), but is not backwards compatible. Choice of reply syntax is completely arbitrary and does not affect operation.

In the Tiger syntax no :A is sent back. Furthermore, whenever an axis position or command value is returned (i.e. whenever the command is a query), the axis letter is always specified. Consequently, when no information needs to be sent back and there is no error the controller simply replies with <CR><LF> only.

The above examples in Tiger reply syntax are as follows:

```
MOVE X=1234 Z=1234.5 <CR>
:A <CR><LF>
MOVE X Y Z <CR>
:A <CR><LF>
WHERE X <CR>
:A 0 <CR><LF>
MOVE X=4 Y=3 Z=1.5 <CR>
:A <CR><LF>
WHERE X Y Z <CR>
:A 4 3 1.5 <CR><LF>
WHERE Z Y X <CR>
:A 4 3 1.5 <CR><LF>
```

Query of Parameters

Most commands used to set parameter values can be queried for the current values using the question-mark syntax: CMND X? Y? Z? F?

The controller will respond with CMND's current settings, e.g. :A X=0 Y=1 Z=10 F=2

This feature is most useful when using a terminal program to change controller parameters to verify that you have made the changes that you think you did, or to check present settings.

Error Codes

When a command is received that the controller cannot interpret, for one reason or another, an error message is returned in the following format:

```
:N-<error code>
```

The error codes are as follows:

:N-1	Unknown Command (Not Issued in TG-1000)
:N-2	Unrecognized Axis Parameter (valid axes are dependent on the controller)
:N-3	Missing parameters (command received requires an axis parameter such as x=1234)
:N-4	Parameter Out of Range
:N-5	Operation failed
:N-6	Undefined Error (command is incorrect, but for none of the above reasons)
:N-7	Invalid card address
:N-8..:N-10	Reserved
:N-11..:N-20	Reserved for filterwheel
:N-21	Serial Command halted by the HALT command
:N-30..:N-39	Reserved

Optimizing Communication Speed

For certain applications the speed of the communication can become a limiting factor, e.g. when keeping a live view of positions and statuses of all axes on a complicated system. Some ideas and notes about improving communication speed follow.

Obviously, to increase communication speed the first step is to increase the baud rate as much as possible. Tiger uses 115200 baud always, but for the MS2000 family of controllers (including RM2000, etc.) the speed is configured using DIP switches as [detailed elsewhere](#); normally 115200 baud can be used without any problem. To compute the raw transfer time, take 10 seconds and divide by the baud rate (10 because there are 8 data bits plus a start and stop bit sent for every byte), so for every millisecond only 1 byte can be sent at 9600 baud compared with more than 11 bytes at 115200 baud.

Note that the computer, drivers, and high-level software can play a strong role in determining the communication speed. As an example, we noticed that the round-trip time for position queries on a 2015-era Xeon-based Windows machine was either 16ms or 31ms using Advanced Serial Port Monitor,

but connecting the same controller to a very similar computer with identical drivers, operating system, and serial software the round-trip time is almost always 11ms. Using Micro-Manager on the latter computer reduces the round trip time to 8ms but in general Micro-Manager seems to check serial traffic in 10ms intervals.

Because the TG-1000/Tiger controllers are modular, servicing most commands require that the communication card parse them, relay the message to the relevant card (e.g. for motorized axes or micro-mirror or PLC), and then relay the response. This adds some extra time compared with the MS2000 family of controllers. For instance, querying a position using home-built serial software takes 7ms on Tiger and about half that on MS2000. The intra-controller communication happens at 57600 baud, though it could be extended to 115200 if needed. For use with Micro-Manager, for example, this extra intra-controller communication time is irrelevant because it happens within the 10ms polling period that Micro-Manager appears to use.

If communication speed is of the utmost importance, it is possible to use binary commands in which case the controller doesn't need to parse the human-readable commands. For MS2000 these are called low level commands and [documentation is here](#). For TG-1000 these are called W commands and the [documentation is here](#). Avoiding the parsing will save about a millisecond or so. Depending on the exact nature of the command the number of bytes transmitted can be more or less than with the usual high-level commands.

[serial](#), [tech note](#), [ms2000](#), [tiger](#)

From:

<http://asiimaging.com/docs/> - **Applied Scientific Instrumentation**

Permanent link:

http://asiimaging.com/docs/tech_note_rs232_comm

Last update: **2019/04/18 23:35**

