

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 to send serial commands is required to utilize all of the controller's abilities. There is a separate page with an [overview of serial commands](#).

Physical interface

MS2000 and RM2000 controllers have a physical RS-232 serial port (a 9-pin D-sub connector). Use with a null modem RS-232 cable. Since 2008, controllers also have a USB Type B connector. Use this with a standard USB Type-B to Type-A cable to connect to ubiquitous Type-A USB receptacle on the host PC. New controllers ship with both a RS-232 and USB cable, and the customer can choose which one to use.

TG-1000 "Tiger" controllers have USB Type B connector in the required TGCOM communication card. An optional TG232 card can be added to provide a physical RS-232 port if required.

If your controller can accept more than one physical connector it does not make a difference which you use.

Serial Link Info

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

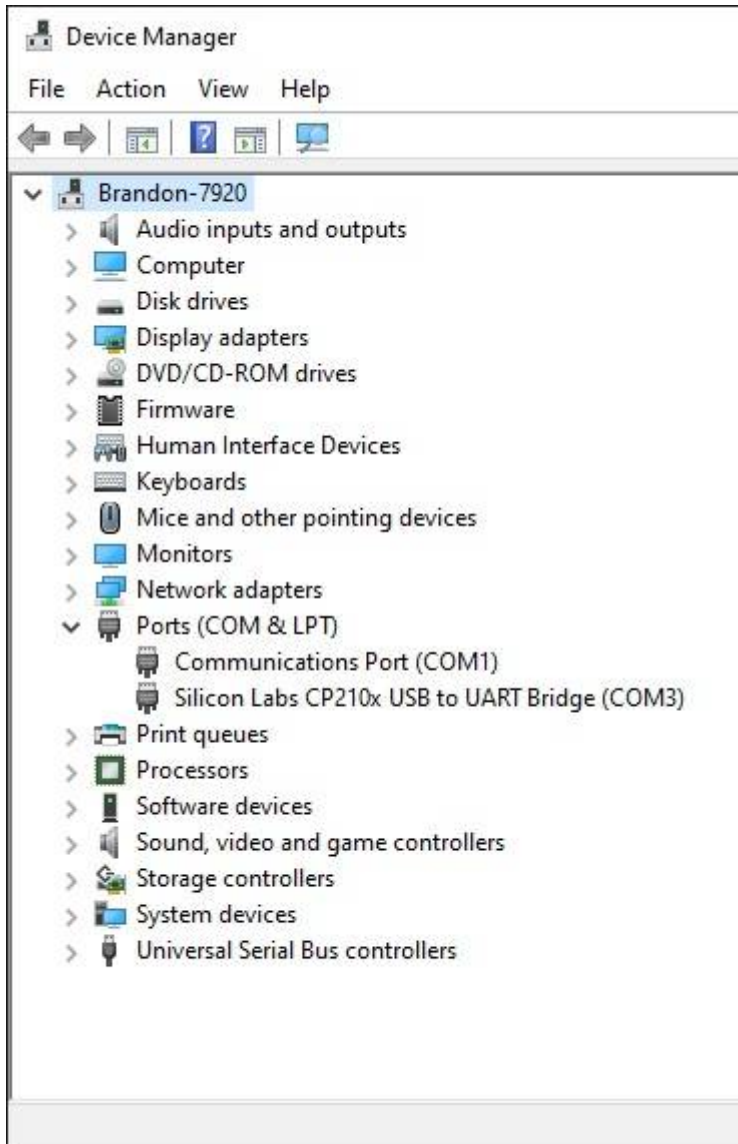
Serial Port Drivers

For USB connections, the controller includes a serial COM port emulator chip that requires drivers on the host PC. Usually the operating system will automatically install the drivers, but if not the drivers can be found at the Silicon Labs [CP210x USB to UART Bridge VCP Drivers](#) downloads page.

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). If you run into trouble with the automatically-installed drivers then uninstall and use the Windows 7/8/10 version 6.x instead.

For troubleshooting connectivity issues in Windows with a USB cable: open Windows Device Manager and expand the section "Ports (COM & LPT)". There should be an entry with name beginning "Silicon Labs CP210x" that appears when you plug the cable into the controller and disappears when you unplug the cable. If this happens then the Windows driver is working correctly; note the port name (e.g. "COM3") for use with software. If you don't see such an entry appearing and disappearing with

the cable connection then there is a problem unrelated to software, likely in the Windows drivers but perhaps in the cable or controller.



Windows Device Manager

Software

All software control is using the serial commands sent over the serial port (or virtual serial port with USB connection). There are many software choices available.

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

ASI Console, ASI Tiger Console, and ASI Tiger Control Panel

ASI Console: for MS2000, RM2000 and FW1000 controllers. Info and download at [here](#). This is the only software that can be used to update firmware on MS2000, RM2000, and FW1000 controllers. It has some control buttons and ability to send serial commands.

ASI Tiger Console: for TG-1000/Tiger controllers. Info and download at [here](#). You can send serial commands using the input box at the bottom of the scripting tab once you have it set up. This is the only software that can be used to update firmware on TG-1000/Tiger controllers.

Tiger Control Panel: for TG-1000/Tiger controllers. Like the Tiger Console it lets user send serial commands to the controller, but it is generally easier to use and has a few additional utilities like displaying live axis position and states. Info and download [here](#).

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. Using the ASI device adapters is recommended, and most common operations can be done without even knowing the underlying serial commands because of functionality exposed via device properties and the device-level API.

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.

However, when us the device adapters there is rarely a need to send direct serial commands. This is because the device adapter provides properties and an API for common tasks such as moving the stage.

Python

Many users want to interact with ASI devices through Python. There are at least 3 viable options:

1. Send serial commands directly using the pyserial library
2. Use pymmcore package to access the MMCore layer of Micro-Manager where ASI devices have excellent support
3. Use pycromanager to access the full Micro-Manager API including MMCore

See the [Python](#) page for details.

Julia

You can send serial commands using the “LibSerialPort” package from Julia programming language package manager.

See the [Julia](#) page for details.

LabView

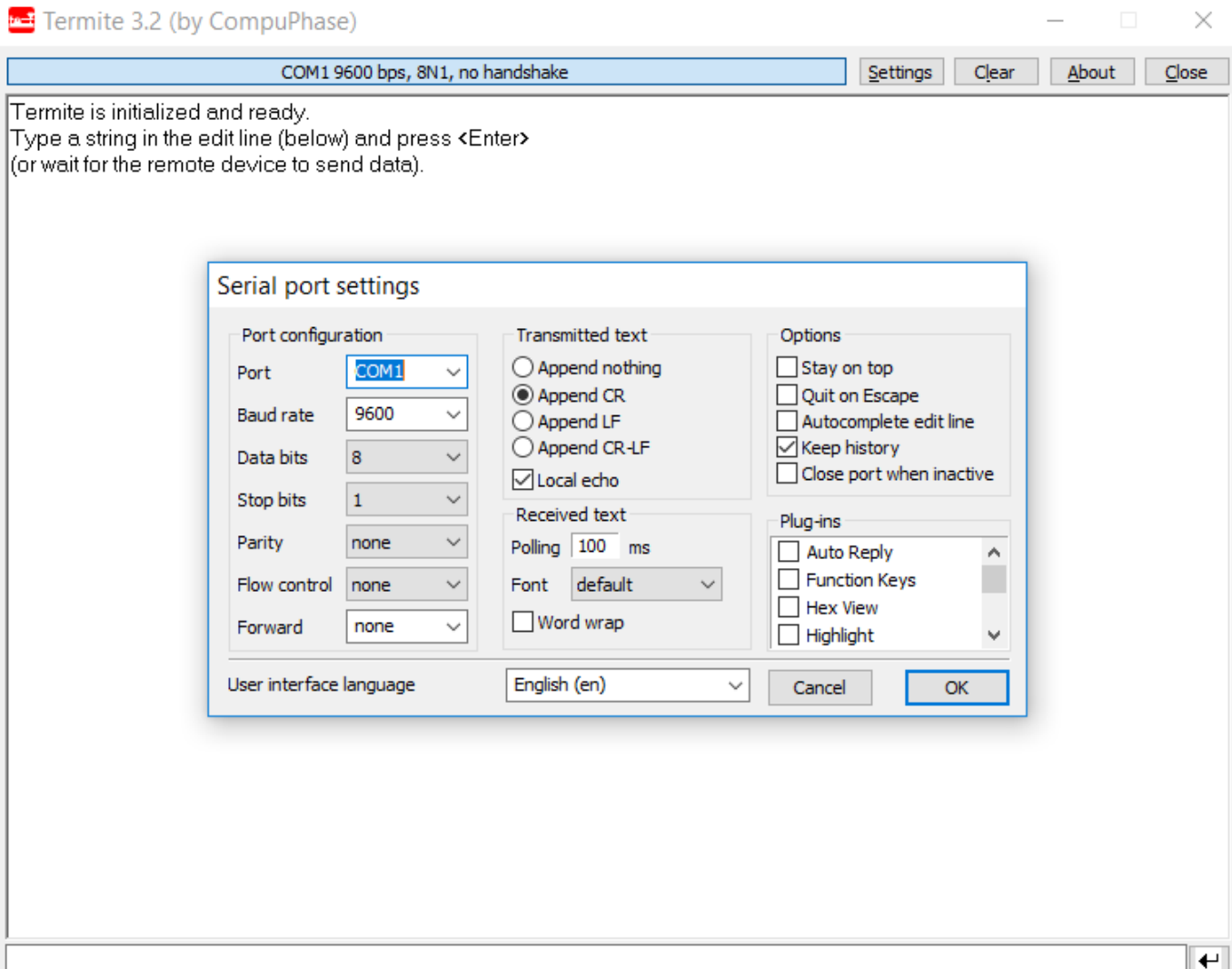
LabView drivers are available from ASI, see the ASI main website under Support → Downloads ([direct link](#)).

Terminal Programs

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

Termiter

A free and easy to use terminal program is [Termiter](#). Below is a screenshot of the setup dialog and how it should be configured to communicate with our controllers using either an RS232 cable or USB cable:



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

Other Third Party Applications

Various 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.

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.

Low Level Commands

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: **2025/03/20 17:33**

