

# Command:TTL

TTL functionality differs based on whether the controller is a Tiger (TG-1000) or MS2000/RM2000 controller, due to hardware differences. Some TTL modes are only available with certain firmware modules.

The Tiger (TG-1000) and MS2000 controller electronics have a buffered TTL input (IN0) and output (OUT0) port that are usually connected to the IN and OUT BNC connectors on the back of the controller. These ports allow voltages in the range of 0V to 5V as an input, where any voltage below 0.95v(+/-0.3v) is a LOGIC LOW signal. Any signal above 1.6 V (+/-0.3 V) is considered a LOGIC HIGH state. Any signals in between 0.95 to 1.6 V will maintain the same logic state that was registered from the last *known* state (Schmitt Triggered inputs). The TTL input has a 10K Ohm resistor to ground, and connecting to the input of a Schmitt Trigger 5W TTL gate. The output is CMOS-compatible 5v TTL directly from a single CMOS gate. The behavior of these connectors are determined by the IN0\_mode and OUT0\_mode parameters set by the TTL X and TTL Y commands respectively. There are also has several unbuffered I/O ports on the motherboard that are occasionally exposed for special purposes.



**Warning!** Absolute maximum voltage to be applied to ASI's controller: -0.5 V to 5.5 V. Any voltage applied that is greater than 5.5 V or less than -0.5 V will void the warranty and may cause damage to the controller!

On Tiger TG-1000 controllers, some cards have buffered TTL input (IN0) and output (OUT0) ports exposed, in which case the behavior is determined by the IN0\_mode and OUT0\_mode parameters set by the TTL X and TTL Y commands respectively. The TTL command is Card-Addressed, meaning that on Tiger it applies to each card separately. A few Tiger cards have extended functionality using the TTL\_AUXILIARY firmware module affected by the TTL R and TTL T commands.

On older versions of the MS2000 and TG1000 firmware, you may need to enable the axes with the [RM Y=# command](#) for TTL-triggered ring buffer moves if the default value is incorrect. After MS2000 version 9.20, the default is 3.

Sending the command RM without any other arguments sets the TTL input interrupt flag and performs the same operation that a single TTL IN0 input pulse would control as determined by the current IN0\_mode. See [RBMODE \(RM\)](#)

MS2000 or RM2000 syntax

<b>Format</b>	TTL [X=IN0_mode] [Y=OUT0_mode] [F=OUT0_polarity] [T=report_mode] TTL Firmware v9.2k+ required.
<b>Remembered</b>	Using SS Z

Unless otherwise specified, the TTL commands used for Tiger apply to all MS-2000 based systems as well.

Tiger syntax

Format	[Addr#]TTL [X=IN0_mode] [Y=OUT0_mode] [Z=aux_IO_state] [F=OUT0_polarity] [R=aux_IO_mask] [T=aux_IO_mode] [Addr#]TTL Firmware v3.16+ required.
Type	Card-Addressed
Remembered	Using [Addr#]SS Z

## TTL Input Mode

### IN0\_mode

**0** - Turns off TTL IN0 controlled functions; TTL interrupt DISABLED.

**1** - TTL IN0 initiates a Move-to-Next-Position of the stored positions in the ring buffer pointed to by the `buffer_pointer`. When the `buffer_pointer` reaches a value equal to the number of saved positions, it resets to the first position, allowing cyclic repetitions to the saved locations. See [RBMODE](#) and [LOAD](#) commands.

**2** - TTL IN0 repeats most recent relative move (see [MOVREL command](#)). For example, begin a session by issuing the command `RM Y=7`, then `MOVREL X=0 Y=0 Z=10`, and each subsequent TTL trigger will cause the Z axis to move 1 micron<sup>1</sup>. This function can be used for repetitive relative moves of any axis or combination of axes on the controller (for MS-2000) or on the card (for Tiger). For the “focus axis” only you may directly set the dZ value with the [ZS](#) command's X parameter.

As of MS2000 9.2o and Tiger v3.38 only the axes that are enabled with `RM Y=#` will be moved. The default axis mask is 3, for X and Y only to match with legacy firmware. Also, the [BACKLASH](#) setting is ignored during TTL-triggered relative moves, on earlier versions of the firmware you will see incorrect results with fast TTL pulses if the backlash is not set to 0.

**3** - TTL IN0 initiates an autofocus operation on systems with autofocus installed.

**4** - Enables TTL IN0 controlled Z-stacks (see the [ZS command](#)).

**5** - Enables TTL IN0-started [serial position reporting](#). Information is asynchronously sent out to the serial interface every `report_time` interval, where `report_time` is set with the [RT command](#). Data returned in the serial stream are the elapsed time in milliseconds since the TTL trigger, followed by the position of each axis enabled by the `axis_byte` (set by the [RM command](#)). On TRACKING systems, the PMT sum signal is also reported. Reporting is toggled on and off by triggering the TTL

input.

---

**6** - On the rising edge of a TTL pulse, initiate a stage scan if the scan state is IDLE. This is equivalent to sending the [SCAN command](#) without arguments to initiate a scan. Note: unlike the scan command, subsequent triggers will not stop the stage scan.

---

**7** - TTL commanded [ARRAY](#) move to next position.

---

MS-2000 v9.54 or Tiger v3.54 required

**8** - TTL commanded [ARRAY](#) move to next position, that does a Z-stack at each array position (see the [ZS command](#)).

---

**9** - Used with [CRISP](#) focus lock. TTL IN0 HIGH engages lock if the system is in the READY state. TTL IN0 LOW will cause the system to UNLOCK if it is locked already.

---

**10** - Toggle TTL OUT0. If TTL OUT0 is set either LOW or HIGH, an input pulse on the TTL IN0 will cause the output to toggle to the other state.

---

Tiger v3.31 required

**11** - SERVOLOCK\_TTL mode. Requires the [SERVOLOCK\\_TTL](#) firmware module. To engage the SERVOLOCK\_TTL mode use the [LK](#) command which will automatically change the TTL IN0 mode and normally will restore it afterwards. When the SERVOLOCK\_TTL mode is engaged do not change the TTL X setting.

Not yet implemented for MS2000.

---

MS-2000 v9.52 or Tiger v3.24 required

**12** - Behavior is exactly the same as mode 1 above except the moves are relative rather than absolute.

**CAUTION:** If you are using TTL X mode 12, the values entered into the ring buffer using the [LOAD](#) command represent RELATIVE coordinates, not ABSOLUTE coordinates. You must drive the stage to the appropriate starting position before triggering a ring buffer sequence.

---

MS-2000 v9.21 or Tiger v3.14 required

**20** - The TTL IN0 pulse turns on the TTL OUT0 for a fixed duration set by the ["RT Y"](#) command.

---

Can be used with a LED illumination to act as a flash of defined duration. For the TGLED card and MS2000/RM2000 with a Dual LED card (DLED), all LED channels remain off until a TTL pulse is received, and the TTL rising edge turns all LED channels ON and they remain ON for the duration set by the **RT Y** command.<sup>2)</sup> LED intensity is set by the **LED** command.

---

MS-2000 v9.2l or Tiger v3.14 required

**21** - Similar to mode 20 except TTL pulses cycle through the channels of the TGLED card or Dual LED card. On each TTL pulse, only ONE LED channel turns ON for a fixed time. The next TTL pulse turns on the next LED channel for a fixed time, and so on.

For Tiger v3.24 and later, when **RB F=3** the TGLED card cycles through all channels without waiting for TTL pulses. If an intensity is set to 0 using the LED command then that channel is automatically skipped.

---

MS-2000 v9.2n or Tiger v3.30 required

**22** - Similar as mode 20 except waits for TTL pulse to go low before turning off the LED. Setting RT Y isn't required.

---

MS-2000 v9.2n or Tiger v3.30 required

**23** - Same as mode 21 except waits for TTL pulse to go low before turning off the LED. Setting RT Y isn't required.

---

Tiger v3.30 required

**30** - Used with Single Axis Modes 2 and 4, see the **SAM** command.

Mode 2: On the rising edge of a TTL pulse, the routine is performed once.

Mode 4: On the rising edge of a TTL pulse, the routine runs continuously.

Not yet implemented for MS2000.

---

MS-2000 v9.2n required

**103** - Repeating Autofocus. This mode does not use TTL IN. When set, the controller will automatically attempt an autofocus routine every *K* milliseconds indefinitely until TTL X is changed. The repetition time, *K*, is set using **RTIME X**. Example: **RT X=5000** will set the repetition rate to 5 seconds. This is effectively the same as using **TTL X=3** with an oscillator connected to TTL IN.

---

## TTL Output Mode

**OUT0\_mode**

**0** - TTL OUT0 unconditionally set LOW.

---

**1** - TTL OUT0 unconditionally set HIGH.

---

**2** - generates TTL pulse at end of a commanded move (MOVE, MOVREL, move via ring buffer, or via array module). The pulse duration is set with command **RT Y**. Note that any move that is initiated before the pulse duration is complete, will reset the duration timer and make the TTL output low immediately. For sequenced (automatic) array moves, use **RT Z** to set the delay before the next array element is sequenced.

---

**3** - ~~output TTL OUT0 gated HIGH during axis index 0 (X) constant speed move~~ **not implemented but hardware-level TTL signal is available.**

---

**4** - ~~output TTL OUT0 gated HIGH during axis index 1 (Y) constant speed move~~ **not implemented but hardware-level TTL signal is available.**

---

**5** - ~~output TTL OUT0 gated HIGH during axis index 2 (Z) constant speed move~~ **not implemented but hardware-level TTL signal is available.**

---

**8** - TTL OUT0 timed arrival pre-pulse output. See RT command. Requires PREPULSE firmware module.

---

**9** - TTL OUT0 PWM and MicroServo or Solenoid output. See the LED or the SECURE command. Requires LED\_DIMMER or USERVO or SOL\_LK firmware module. On WK, with LED\_DIMMER, the PWM frequency is 1KHz.

---

**10** - Output TTL OUT0 gated HIGH upon completion of video AUTOFOCUS function. AUTOFOCUS hardware and firmware required.

---

**11** - Generates TTL OUT0 pulse at end of commanded move providing CRISP is in 'F' state (active and within tolerance). Waits for CRISP 'F' state after move completion to send a pulse. Generally not useful with TG-1000 controllers because the XY and focus axes are on different cards. ON MS-2000

---

the behavior of the STATUS command is modified when this TTL mode is set: 'N' is returned when CRISP is in the In-Focus 'F' state and otherwise 'B' is returned. The pulse duration is set with command [RT Y](#). Note that any move that is initiated before the pulse duration is complete, will reset the duration timer and make the TTL output low immediately. For sequenced (automatic) array moves, use [RT Z](#) to set the delay before the next array element is sequenced.

---

**12** - TTL OUT0 high when CRISP is in the F state, low otherwise.

---

**20** - TTL OUT0 set during SPIM state machine operation. Requires MM\_SPIM firmware module (TG-1000 only). Was code 10 until v3.12.

---

**21** - TTL1 backplane signal (PLC address 42) set high at the end of a ring buffer move or AIJ-initiated move (for laser trigger). Requires MM\_TARGET firmware module (TG-1000 only). Was code 11 until v3.12. As of firmware v3.36 outputs to the TTL1 backplane signal instead of to the usual TTL output.

---

Tiger v3.17 required

**22** - In this mode, TTL OUT0 is controlled by the [single-axis function](#) module. With the [SAP](#) command the user can generate a TTL pulse that is synchronized with [single-axis function](#) actuator motion. The pulse duration is set with command [RT Y](#).

---

Tiger v3.38 required

**30** - TTL OUT1 high when the position of first axis is more than the PC setting away from its target position.

---

Tiger v3.38 required

**31** - TTL OUT1 high when the position of second axis is more than the PC setting away from its target position.

---

Tiger v3.38 required

**32** - TTL OUT1 high when the position of third axis is more than the PC setting away from its target position.

---

Tiger v3.38 required

**33** - TTL OUT1 high when the position of fourth axis is more than the PC setting away from its target position.

---

## TTL Output Polarity

### OUT0\_polarity

**1** - Default polarity

**-1** - Invert the polarity of TTL OUT0

## TTL Auxiliary (Tiger Only)

### aux\_IO\_state

**Z** - Requires TTL\_AUXILIARY firmware module; behavior depends on the firmware build and hardware present. Sets the state of the auxiliary TTL output according to the aux\_IO\_mask. Input and output as a decimal number representing the binary pattern desired. The following uses have been defined so far:

For MM\_SPIM firmware with SPIM TTL card: Bit0 = Side0/Laser0 output, Bit1 = Side1/Laser1 output

### aux\_IO\_mask

**R** - Requires TTL\_AUXILIARY firmware module; behavior depends on the firmware build and hardware present. Controls how the aux\_IO\_state bits are used, or how the backplane is used when aux\_IO\_mode is set to 2. Input and output as a decimal number representing a binary mask. If the corresponding mask bit is set to 1 then the state bit will be reflected at the output, but if the mask bit is 0 then the state bit has no effect. The following uses have been defined so far:

For MM\_SPIM firmware: Defaults to 3 = 0b00000011.

### aux\_IO\_mode

**T** - Requires TTL\_AUXILIARY firmware module; behavior depends on the firmware build and hardware present. The SPIM state machine overrides these setting during its operation.

**Mode 0** - TTL outputs determined by aux\_IO\_state/mask.

**Mode 1** - TTL outputs determined by the LED command (requires MM\_LASER\_TTL module). The default setting for MM\_SPIM firmware.

**Mode 2** - Simulates a TTL input from the backplane. The backplane value is masked by aux\_IO\_mask and the binary value is considered. If a 0-1 transition occurs then a TTL input pulse is simulated and action will be taken depending on the setting of IN0\_mode. The default setting for TTL\_AUXILIARY on piezo firmware.

## Report Mode (MS-2000 Only)

report\_mode - This mode requires the TTL\_OUT\_REPORT firmware module. MS-2000 v9.52 required.

**T** - Enables serial reporting on every output TTL pulse when set to be non-zero. Will output on separate serial port if enabled with compile-time flag.

**Mode 51** - Outputs the following bytes in order:

Mode 51 Byte Order

If ARRAY\_MODULE is defined

- X array index as 16-bit unsigned int (AIJ X? equivalent)
- Y array index as 16-bit unsigned int (AIJ Y? equivalent)

Next the encoder value for all axes in order as 32-bit signed integer in twos-complement, 4 bytes per axis (can be converted to reported position using multiplier)

If CRISP is enabled

- CRISP error value as signed int
- CRISP sum value as signed int

Footer:

- carriage return '\r'
- 16-bit checksum calculated on bytes 1-20 according to IP header algorithm
- carriage return '\r'

IP header algorithm for checksum computation

- Treat every 2 bytes as 16-bit unsigned integer. Compute the running sum all 2-byte chunks in a 32-bit register.
- Sum the two halves of the 32-bit register, sum the resulting two halves again in the case of overflow, and subsequently perform a bitwise inversion: this is the 16-bit checksum value.
- To validate the checksum repeat the same steps but include the 16-bit checksum value in the running sum. The computed checksum should be 0000. If not there was an error in checksum generation or transmission of the data.

Example: firmware includes ARRAY\_MODULE, CRISP and has X Y and Z axes. TTL Y=2 sets a TTL pulse at the end of the move. TTL T=51 sets this report format. The report will be a string of numbers like this (spaces for readability only)

```
0002 0001 0003 1E00 FFFF FFFF FFFF FFEE FC81 0679 0D DF0F 0D
```

where coordinate is (2,1) in the array, the X encoder value is 3268608 in decimal (corresponding to almost exactly 9mm position with 2 TPI rotary stage), Y encoder value is -1 in decimal, Z encoder value is -18 in decimal, CRISP error value is -895 in decimal, CRISP sum value is 1657 decimal, and the checksum is hex DF0F. To validate the checksum add each of these 16-bit unsigned numbers (excluding the 0D delimiters) which results in 5FFFA in hex. FFFA plus 5 is FFFF, and upon bitwise inversion the value becomes 0 as expected.



## Without Any Argument

In firmware version 3.16 and above on TG-1000 and version 9.2k and above on MS2000/RM2000, when the TTL command is issued without any arguments, like [Card Addr#]TTL , the controller reports the state of TTL IN.

For Tiger/TG-1000, the controller replies with :A 0 when signal is low. and :A 1 when TTL IN sees a high signal.

For MS2000/RM2000, the controller replies with the inverse polarity, :A 1 when the signal is low and :A 0 when the signal is high. This is a logical bug in the firmware, but because it was in the field so long before being noticed we made the intentional decision to keep the behavior unchanged.

[commands](#), [led](#), [tiger](#), [ms2000](#), [tgled](#), [crisp](#), [ttl](#), [spim](#), [array](#), [servolock ttl](#)

<sup>1)</sup>

strictly speaking, it will be the closest multiple of the specified distance in encoder units because that is how the move distance is stored internally

<sup>2)</sup>

Old text, unknown meaning: TTL Y must be in 0 mode (TTL Y=0) or it might cause an issue.

From:

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

Permanent link:

<https://asiimaging.com/docs/commands/ttl>

Last update: **2026/01/27 18:28**

