Command list for TLC v24x FWv1.63

This document describes all commands and settings for the TLC firmware version 1.63.

This version supports the TLC hardware versions 2.40 to 2.45.

To be able to set the laser current or heater voltages the TLC must first be switched on using "SYST:STAT 1".

1.1 Serial port

The TLC uses an FTDI USB to serial converter. The FTDI drivers are typically supported by windows but can also be downloaded from their website: https://ftdichip.com/drivers/vcp-drivers/

The driver will add a virtual com port for the TLC. To communicate with the TLC the com port settings are as follows:

Serial port settings	
Baudrate	**115200**
Data bits	**8**
Parity	**No parity**
Stop bits	**1**
Handshaking	**None**
Command string termination	**CR+LF**

The baudrate can be set to a different value using the command **"**COMM:BAUD baudrate".
E.g. COMM:BAUD 57600 will set the baudrate to 57600.
Note that the TLC will return to the default baudrate on reset or power cycling.

1.2 Command syntax and returned values

All commands return a 0 if the command was executed successfully or 1 if there was an error.
E.g. SYST:STAT 1 will activate the TLC but will also return 0.
Query commands return a 0 as well as the result of the query.
E.g. LSR:ILEV? Might return 0 200 where 200 is the current output of the laser driver.
This prefix is on by default but can be disabled with the command COMM:PFX 0

After a command is given for the first time the next command, if it is the same command apart from the parameters, can be abbreviated using ";"
E.g. After first sending the command DRV:D 0 3.5 to set the first driver to a voltage of 3.5V
The same (or in this case second) driver can be set using: ;1 4.3
This will set driver 1 to 4.3V.

1.3 Root level commands

Returned Value	Command	First Operand	Second Operand	User Mode	Description
Root					
0 **string**	CMDL?	-	-	always	Returns all possible commands on this level.
0 **string**	*IDN?	-	-	always	Returns the device identifier string
0	*RST	-	-	always	Resets system

1.4 System commands

Returned Value	Command	First Operand	Second Operand	User Mode	Description
System					
0 **string**	SYST:CMDL?	-	-	always	Returns all possible commands on this level.
0	SYST:STAT	**bool** [0..1]	-	always	(De)Activates the system.
0 **bool**	SYST:STAT?	-	-	always	Returns the current system state
0	SYST:SRN	**string**		admin	Sets the serial number.
0 **string**	SYST:SRN?	-	-	always	Returns the serial number
0	SYST:PWD	**string**	-	always	Used to set admin mode
0 **bool**	SYST:PWD?	-	-	always	Returns the admin mode state
0 **string**	SYST:TTM?	-	-	admin	Debug Command. Returns the system overview.
0	SYST:HWV	**int** [240,241,242]	-	admin	Sets the TLC hardware version. Can be either 240, 241 or 242.
0 **int**	SYST:HWV?	-	-	always	Returns the TLC hardware version.

1.5 Fan commands

The fan commands are only supported for hardware versions 2.40, 2.41. Note that the hardware does not check if the fan is actually present.

Returned Value	Command	First Operand	Second Operand	User Mode	Description
Fan					
0 **string**	FAN:CMDL?	-	-	always	Returns all possible commands on this level.
0	FAN:STAT	**bool** [0..1]	-	always	Enables or disables the fan.
0 **bool**	FAN:STAT?	-	-	always	Returns the current fan state

1.6 Communication commands

Returned Value	Command	First Operand	Second Operand	User Mode	Description
Communication					
0 **string**	COMM:CMDL?	-	-	always	Returns all possible commands on this level.
0	COMM:BAUD	**int**	-	always	Sets the baud rate. Valid options are: 57600, 38400, 28800, 19200, 14400 and 9600. Note that this value defaults to 155200 at powerup.
0 **int**	COMM:BAUD?	-	-	always	Returns the baud rate
0	COMM:ECHO	**bool** [0..1]	-	always	Sets the debug mode that echoes the command that was sent.
0 **bool**	COMM:ECHO?	-	-	always	Returns the echo command debug mode state
0	COMM:PFX	**bool** [0..1]	-	always	Sets the prefix state
0 **bool**	COMM:PFX?	-	-	always	Returns the prefix state
0	COMM:TC	**bool** [0..1]	-	always	Sets the debug mode to measure command execution time
0 **bool**	COMM:TC?	-	-	always	Returns the measure command execution time debug mode state

1.7 Laser current source commands

Returned Value	Command	First Operand	Second Operand	User Mode	Description
LASER					
0 **string**	LSR:CMDL?	-	-	always	Returns all possible commands on this level.
0	LSR:STAT	**bool**	-	admin, system active	Switches the laser driver on or off
0 **bool**	LSR:STAT?	-	-	always	Returns the laser driver state
	LSR:ILEV	**float**	-	admin, system active	Sets the laser current in mA (Should be <250mA)
0 **float**	LSR:ILEV?	-	-	always	Returns laser current in mA
0 **float**	LSR:IMAX?	-	-	always	Returns the maximum allowable current in mA.
0 **float, float**	LSR:CAL?	-		admin	Returns the laser current calibration.
0	LSR:CAL	**float**	**float**	admin	Sets the laser current calibration. Output laser current is updated immediately!
0	LSR:IMAX	**float**		admin	Sets the maximum allowable output current in mA.

1.8 TEC driver commands

Returned Value	Command	First Operand	Second Operand	User Mode	Description
TEC					
0 **string**	TEC:CMDL?	-	-	always	Returns all possible commands on this level.
0	TEC:STAT	**bool**	-	admin	Switches the TEC driver on or off. Do not switch off when laser is active!
0 **bool**	TEC:STAT?	-	-	always	Returns TEC driver state
0 **float**	TEC:TEMP?	-	-	always	Returns the actual TEC temperature (°C)
0	TEC:TTGT	**float**	-	always	Set the target temperature of the TEC (°C)
0 **float**	TEC:TTGT?	-	-	always	Returns the target temperature of the TEC (°C).
0 **float**	TEC:ITEC?	-	-	always	Returns the actual current applied to the TEC (A).
0 **float**	TEC:VTEC?	-	-	always	Returns the actual voltage applied to the TEC (V).

1.9 TEC configuration commands

Returned Value	Command	First Operand	Second Operand	User Mode	Description
TEC					
0 **string**	TEC:CFG:CMDL?	-	-	always	Returns all possible commands on this level.
0	TEC:CFG:TMIN	**float**	-	admin	Sets the minimum allowed target temp (°C).
0 **float**	TEC:CFG:TMIN?	-	-	always	Returns the minimum allowed target temp.
0	TEC:CFG:TMAX	**float**	-	admin	Sets the maximum allowed target temp (°C).
0 **float**	TEC:CFG:TMAX?	-	-	always	Returns the maximum allowed target temp.
0	TEC:CFG:ITGT	**float**	-	admin	Sets the target temp on start-up (°C).
0 **float**	TEC:CFG:ITGT?	-	-	always	Returns the target temp on start-up (°C).
0	TEC:CFG:ILIM	**float**	-	admin	Returns the maximum applied current.
0 **float**	TEC:CFG:ILIM?	**string**	-	always	Sets the maximum applied current.

1.10 TEC calibration commands

Returned Value	Command	First Operand	Second Operand	User Mode	Description
TEC					
0 **string**	TEC:CAL:CMDL?	-	-	always	Returns all possible commands on this level.
0	TEC:CAL:STAT	**bool**	-	admin	Sets whether temperature calibration is enabled.
0 **bool**	TEC:CAL:STAT?	-	-	always	Returns whether temperature calibration is enabled.
0	TEC:CAL:VAL	**float**	**float**	admin	Sets the coefficient (first operand) and offset (second operand) of the linear temperature calibration.
0 **float float**	TEC:CAL:VAL?	-	-	always	Prints the coefficient and offset of the linear temperature calibration.
0 **string**	TEC:CAL:INF?	-	-	always	Prints temperature calibration function.

1.11 TEC controller commands

Returned Value	Command	First Operand	Second Operand	User Mode	Description
TEC					
0 **string**	TEC:CTRL:CMDL?	-	-	always	Returns all possible commands on this level.
0 **int**	TEC:CTRL:PSHR	**int**	-	admin	Sets the P-share of the control loop.
0 **int**	TEC:CTRL:PSHR?	-	-	always	Returns the P-share of the control loop.
0 **int**	TEC:CTRL:ISHR	**int**	-	admin	Sets the I-share of the control loop.
0 **int**	TEC:CTRL:ISHR?	-	-	always	Returns the I-share of the control loop.
0 **int**	TEC:CTRL:DSHR	**int**	-	admin	Sets the D-share of the control loop.
0 **int**	TEC:CTRL:DSHR?	-	-	always	Returns the D-share of the control loop.
0 0	TEC:CTRL:SAVE	-	-	admin	Save changed settings to TEC non-volatile storage.
0 **string**	TEC:CTRL:CMD	**string**	-	admin	Sends a command directly to the TEC controller, and prints the response.

1.12 Actuator (e.g. heater) driver commands

Returned Value	Command	First Operand	Second Operand	User Mode	Description
ACTUATOR					
0 **string**	DRV:CMDL?	-	-	always	Returns all possible commands on this level.
0	DRV:STAT	**bool**	-	admin, system active	Switches the actuator driver power supply on or off.
0 **bool**	DRV:STAT?	-	-	always	Returns the actuator driver power supply state
0	DRV:D	**int [0..5]**	**float/word**	admin, system active	Sets actuator <operand1> to value <operand2>.
0	DRV:DP	**int [0..5]**	**float**	admin, system active	Presets actuator <operand1> to value <operand2>. The driver will only output this value after an update command is given.
0 **float/word**	DRV:D?	**int [0..5]**	-	always	Returns the current or preset value for actuator <operand1>
0	DRV:U	**-**	-	Always	Updates the driver outputs based on the preset values
0	DRV:CLR	-	-	always	Sets all actuators to zero volt.
0	DRV:TEST	**-**	-	admin	Actuator test function.
0	DRV:SPT	**int [0..39]**		admin, system active	Saves the current heater settings under preset <operand1>
0	DRV:LPT	**int [0..39]**		admin, system active	Sets the heaters to the preset values stored under <operand1>
0	DRV:CPT			always	Clears all presets (to 0) but does not update the current heater settings.

1.13 Actuator configuration commands

Returned Value	Command	First Operand	Second Operand	User Mode	Description
ACTUATOR					
0 **string**	DRV:CFG:CMDL?	-	-	always	Returns all possible commands on this level.
0 **float**	DRV:CFG:DM?	**int [0..5]**	-	always	Returns the maximum actuator voltage for actuator <operand1>
0	DRV:CFG:DL	**int [0..5]**	**float**	always	Sets actuator <operand1> output limit to value <operand2>. This value is stored in eeprom and limits the output value of the driver.
0 **float**	DRV:CFG:DL?	**int [0..5]**	-	always	Returns the limit value for actuator <operand1>
0	DRV:CFG:LLD	**-**	**-**	always	Load Limit Defaults. Resets the drive limits to their maximum possible values.
0 **int**	DRV:CFG:ST?	-	-	always	Returns the supported SID type
0	DRV:CFG:ST	**int [0..2]**	-	admin	Sets the supported SID type.
0 **int**	DRV:CFG:DN?	**-**	-	always	Returns the total number of actuators (typically 6) available in the system.
0	DRV:CFG:SBM	**bool**	-	always	Set the drive commands to accept integer values
0 **bool**	DRV:CFG:SBM?	-	-	always	Returns the state of SBM
0 **float**	DRV:CFG:CFR?	**int [0..5]**	-	always	Returns the conversion factor for actuator <operand1>

DRV:D parameter type

The commands DRV:D and DRV:D? set or return the output value of the driver. By default this value is set or returned using a float where the float is an actual representation of the output value of the driver, in this case voltage. It is also possible, however, to set the driver output to use an integers instead by sending the command DRV:CFG:SBM 1 to the TLC.

This has the advantage that it speeds up the processing of the command but is does comes at the price of having to scale the input value to an integer on the controlling program on the pc.

If this mode is used then first retrieve the conversion factor for each driver using DRV:CFG:CFR? And store these in an array. E.g.:

```
float cfr[6];
cfr[0]= GetSerialReturnedValue("DRV:CFG:CFR? 0");     //For all 6 drivers
```

Then, instead of sending DRV:D 0 5.1
The voltage of 5.1V first needs to be scaled and converted to an int using:
uint16_t IntValue=static_cast<uint16_t>(ValueInVoltage*cfr[0]);
and then sent to the TLC using DRV:D 0 IntValue

1.14 Actuator cycler commands

The actuator cycler only operates on actuators 0, 3, 4, 5.

Returned Value	Command	First Operand	Second Operand	User Mode	Description
ACTUATOR					
0 **string**	DRV:CYC:CMDL?	-	-	always	Returns all possible commands on this level.
0 **int**	DRV:CYC:COUN?	-	-	always	Returns the number of entries in the cycler table.
0	DRV:CYC:COUN	**int [1..7743]**	-	always	Sets the number of entries in the cycler table (persists after power cycle).
0 **int**	DRV:CYC:INT?	-	-	always	Returns the interval of the cycler, in microseconds.
0	DRV:CYC:INT	**int [40..35000]**	-	always	Sets the interval of the cycler, in microseconds.
0	DRV:CYC:LOAD	**int [0..7742]**	-	system active	Sets the actuators to the values saved in entry <operand1> in the cycler table.
0	DRV:CYC:SAVE	**int [0..7742]**	-	system active	Saves the current values of the actuators to entry <operand1> in the cycler table.
0	DRV:CYC:RUN	-	-	system active	Start the cycler.
0	DRV:CYC:FRUN	-	-	system active	Start the cycler in fast mode (see note below).
0	DRV:CYC:ABRT	-	-	system active	Stop the cycler.

Cycler fast mode

When the driver cycler is started in fast mode, the TLC does not accept the regular commands anymore. Instead, it only accepts a single character to stop the cycler: 'x' (hexadecimal 0x78, without the regular CR+LF terminator). When the cycler is stopped, regular commands are accepted once again.

If the interval is set to less than 200 us, the cycler can only be run in fast mode.

Improving performance

In some cases it might be important to increase the update rate of the actuator drivers, I.e. to speed up the tuning of the laser.

The fastest possible sequential update rates can by achieved by doing the following:
COMM:PFX 0 //Switch off the prefix. It takes time to send this character.
DRV:CFG:SBM 1 //Switch the DRV:D command to use integers.
Then, make sure to use ";" for all identical commands following the first command.

The fastest possible simultaneous update is possible using either the built in eeprom memory to store preset values or use several DRV:DP commands followed by a DRV:U command.

For instance, the sequence:
DRV:DP 0 2.3
;1 8.7
;2 12.5
DRV:U
Will update the three driver outputs to the given values nearly simultaneously.

Of course the commands COMM:PFX 0 and DRV:CFG:SBM can be used to further improve performance here as well.