

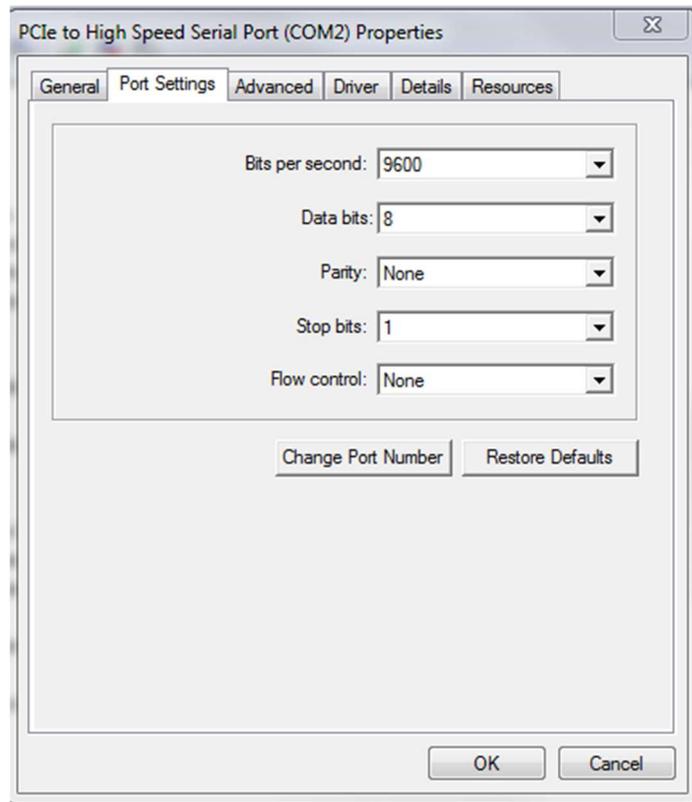
Optics Focus Instruments Co., Ltd.

[**COMMANDS AND PROTOCOLS**]

2024-10-31 updated

1. Port Parameters Settings:

- Bits per second: 9600
- Data bits: 8
- Parity: None
- Stop bits: 1
- Flow control: None



2. Command Format:

- There's no space in the command line.
- It's not available to run multi-commands at one time. You have to run next command after the first command has been performed and finished.
- If you're not familiar with serial port programming, we suggest you to program by the ActiveX control file (.ocx) or DLL file we provide. Please download them from our website.
- The commands are Visual Basic programming language format. If you use other programming language, please change to corresponding format.

For example:

For connection command, VB format is `"?R"&CHR$(13)`, VC format is `{0x3f,0x52,0x0d}`; or `{'?','R',0x0d}`;

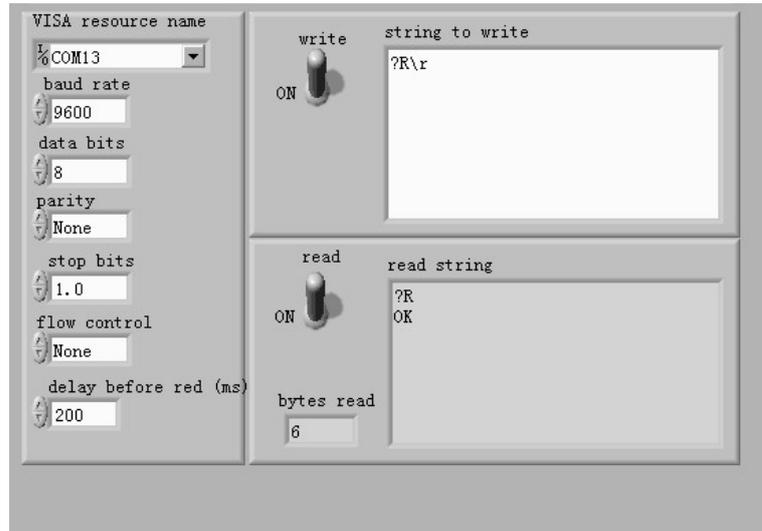
- If you use the commands in LabVIEW, please add `'\r'` at the end of commands.

For example:

`?R\r`

We provide a .vi file (Basic Serial Write and Read.vi) for programming via serial port directly.

You can input and test the commands by using it in LabVIEW.



- If you use the commands in Matlab, please set "LF" as the read terminator and "CR" as the write terminator.

For example:

```

15 % com4 is the communication port.
16 SP = serialport("COM4",9600);
17 configureTerminator(SP,"LF","CR") % "LF" = read terminator,
18 % "CR" = write terminator
19 %% Do while difference from center is greater than 5 microns
20
21 %%
22 Y_move = SP.writeread("Y+1000")
23 Y_pos = SP.writeread("?Y")
24 r_pos = SP.writeread("?r")
25

```

Command Window

New to MATLAB? See resources for [Getting Started](#).

```

Y_move =

    "Y+1000"
    "OK"

Y_pos =

    "?Y"
    "Y+10692"

r_pos =

    "?r"
    "r+39083"

```

3. Commands:

(1) Connection

Command: "?R"&CHR\$(13)

Return: "?R"&CHR\$(13)&"OK"&CHR\$(10) // successful connection

Explanation: If the controller has been connected successfully, the controller will return "OK"&CHR\$(10) within 200ms. **Please send this command before sending any other commands.**

(2) Speed inquiry

Command: "?V"&CHR\$(13)

Return: "?V"&CHR\$(13) "V number"&CHR\$(10)

Explanation: The "number" in Return is the current speed value in form of ASCII. The range of the number is from 0 to 255.

(3) Position inquiry

Command:

"?X"&CHR\$(13) //inquire the position of X axis

"?Y"&CHR\$(13) //inquire the position of Y axis

"?Z"&CHR\$(13) //inquire the position of Z axis

"?r"&CHR\$(13) //inquire the position of R axis

"?t"&CHR\$(13) //inquire the position of T1 axis

"?T"&CHR\$(13) //inquire the position of T2 axis

Return:

"?X"&CHR\$(13)&"X+number"&CHR\$(10) OR

"?X"&CHR\$(13)&"X-number"&CHR\$(10)

Explanation: The "number" in Return is the current position value of the corresponding axis in form of ASCII.

"+" means the position is at positive direction of the zero position.

"-" means the position is at negative direction of the zero position.

(4) Speed setting

Command: "V"&number&CHR\$(13)

Return: "V"&number&CHR\$(13)&"OK"&CHR\$(10) //set the speed successfully

Explanation: The "number" in command is the speed value in form of ASCII. The range of the number is from 0 to 255.

If the speed has been set up successfully, the controller will return "OK"&CHR\$(10).

After a speed value is set, it will be applied to the axis that will be set to move. Then after a new speed value is set, it will be applied to the new axis that will be set to move. E.g. You can set a speed value then move X axis, then the speed value will be applied to X axis. Then you set a new speed value and then move Y axis, the new speed value will be applied to Y axis.

(5) Return to origin

Command:

"HX"&mode&CHR\$(13) //set X axis to return to the origin

"HY"&mode&CHR\$(13) //set Y axis to return to the origin

"HZ"&mode&CHR\$(13) //set Z axis to return to the origin

`"Hr"&mode&CHR$(13) //set R axis to return to the origin`

`"Ht"&mode&CHR$(13) //set T1 axis to return to the origin`

`"HT"&mode&CHR$(13) //set T2 axis to return to the origin`

`Return: "HX"&mode&CHR$(13)&"OK"&CHR$(10) //return to the origin successfully`

Explanation: The "mode" in command has two values: "0" or "1" in the form of ASCII. "0" means the axis will stop right away after it returned to the origin. "1" means the axis will be back to the last position after it returned to the origin.

After this command has been performed and finished, the controller will return "OK" &CHR\$(10).

(6) Return to origin inquiry

Command: `"?H"&CHR$(13)`

Return: `"?H"&CHR$(13)&"H000000"&CHR$(10)`

Explanation: The number "0" in each position of "000000" can be "1" or "0". The six numbers "000000" in turn are on behalf of X, Y, Z, R, T1 and T2 axis. "1" means the corresponding axis has returned to the origin and "0" means the axis has not returned to the origin.

(7) Motion control

Command:

`"X direction number"&CHR$(13)`

`"Y direction number"&CHR$(13)`

`"Z direction number"&CHR$(13)`

`"r direction number"&CHR$(13)`

`"t direction number"&CHR$(13)`

`"T direction number"&CHR$(13)`

Return:

`"X direction number"&CHR$(13)&"OK"&CHR$(10) //the axis has moved the displacement`

Explanation: The "direction" in command is the motion direction. Its value can be "+" or "-". "+" means the axis will move at the positive direction and "-" means the axis will move at the negative direction.

The "number" in command is the displacement value (pulse number) in form of ASCII.

The final position = current position + number (when direction is "+")

The final position = current position – number (when direction is "-")

After this command has been performed and finished, the controller will return "OK" &CHR\$(10).

(8) Stop command

Command: `"S"&CHR$(13)`

Return: `"ERR4"&CHR$(10), "OK"&CHR$(10)`

Explanation: This command is valid when the motion controller runs the "Return to origin" or "Motion control" command. The Return means the axis is stopped, but it doesn't move to the appointed position. So you need to get the current the position of the axis firstly via the "Position inquiry" command in some cases.

4. Formulas:

4.1 Definition:

- Step value: The steps stepper motor runs when the stepper motor rotates one circle.
- Stepper angle: The angles stepper motor rotates by single pulse caused.
- Pulse equivalent: The displacement (mm) of translation stage or rotation degree (degree) of rotary stage by single pulse caused, namely control resolution.
- Subdivision: Default subdivision is 2, please refer to the table on the rear panel of controller.
- Pulse number: The corresponding pulse of the displacement translation stage moves or the degree rotary stage rotates.
- Speed value range: 0-255. It depends on the value you set up.
- The relationship between the speed value and pulse period: $30 / (\text{speed value} + 1)$. When the speed value is 255, the driver will send 0.1171875 pulses / ms.

4.2 Formulas:

Integral step value of per rotation of the stepper motor = $360 / \text{stepper angle of the stepper motor}$

Stepper angle of stepper motor is 0.9 degree or 1.8 degree. Please refer to the specifications of specific motorized stage.

(1) For translation stage:

■ Pulse equivalent = $\text{pitch of the lead screw (mm)} / (\text{integer step value of per rotation of the step motor} * \text{subdivision}) = \text{pitch of the lead screw (mm)} * \text{stepper angle} / (360 * \text{subdivision})$

■ Actual displacement (mm) = $\text{pulse number} * \text{pulse equivalent}$

If you set the pulse number of displacement to N, then the actual displacement equals to $N * \text{pulse equivalent}$.

■ Pulse number = $\text{Actual displacement (mm)} / \text{pulse equivalent}$

If you want to move the translation stage M mm, then the corresponding pulse number equals to $M / \text{pulse equivalent}$.

■ Actual speed(mm/s) = $(\text{speed value}+1) * 22000 * \text{pulse equivalent} / 720$

The speed value is the value you set up. The range of the speed is from 0 to 255.

(2) For rotation stage or goniometer stage:

- Pulse equivalent = $360 / (\text{integer step of per rotation of the step motor} * \text{subdivision} * \text{transmission ratio}) = \text{stepper angle} / (\text{subdivision} * \text{transmission ratio})$
- Actual rotation degree (degree) = pulse number * pulse equivalent
If you set the pulse number of rotation degree to N, then the actual rotation degree equals to N * pulse equivalent.
- Pulse number = Actual rotation degree (degree) / pulse equivalent
If you want to move the rotary stage M degree, then the corresponding pulse number equals to M / pulse equivalent.
- Actual speed (degree/s) = $(\text{speed value} + 1) * 22000 * \text{pulse equivalent} / 720$
The speed value is the value you set up. The range of the speed is from 0 to 255.

5. Errors Definitions:

- ERR1 communication error/sent invalid commands/communication is a timeout
- ERR2 communication not established
- ERR3 invalid command
- ERR4 stop command
- ERR5 limit switch is valid