Laser Communication Protocol

V1.2

Content

A. Communication mode

Supports TCP and RS232 modes（User port & Service port）
Set communication parameters based on site requirements

B. Protocol format

Laser Receive	Start code	Device ID	Content	End mark
Laser Feedback	Start code	Device ID	Feedback	End mark

The start and end characters can be set on the operation interface. Default Settings: start code " ", end mark ";;"
Hexadecimal is supported，Example：the start code is 02 in ASCII，can be set to 0x02

If the data does not meet the protocol format requirements, the system will respond: “Error, Invalid data;;”
All data must be in utf8 format.

Note：The following protocol descriptions use the default start and end characters as examples

C. Protocol specification

1． Get the board connection status

Function description：Check the connection status between the upper and lower computers
Instruction：GetLinkStatus

```
Send：GetLinkStatus;;
Receive：1;;
```

Note： 1- means connection is OK
0- means board not connected

2． Get marking status

Function description：Check system working status
Instruction：GetMarkStatus

Send：GetMarkStatus;;
Receive：0;;

Note：
Receive:0 system idle
1 system in simulated state
2 system in marking state
3 system in preview state
4 system in laser correction state
5 system in red light correction mode
6 laser forced out
7 system in rotation marking mode

3. Get marking count

Function description：Get the marking count of the current system, including total marking, current marking, missed marking count value， single marking time.
Instruction：GetCount

Send：GetCount;;
Receive：2,1,0,100;;

Note：In the received content
2-total marking count
1- current marking count
2- 0-number of missed marking
3- 100-100ms for single marking time

4. Get the total marking count（Function as above）

Function description：Get the total marking count
Instruction：GetMarkedCount

Send：GetMarkedCount;;
Receive：2;;

Note：2-total making count

5. Get the missed marking count（Function as above）

Function description：Get the count of missed marking
Instruction：GetMissedCount

```
Send：GetMissedCount;;
Receive：0;;
```

Note：Receive 0-missed count

6. Clear count

Function description：clear current count value
Instruction：ClearCount

```
Send：ClearCount;;
Receive：Ok;;
```

Note：OK means clear success
Failed,ErrorInfo means clear failed，ErrorInfo-log description
example“is marking;” means marking is in progress， can not clear the count

7. Get current template data

Function description：Get the current template data
Instruction：GetMarkData

```
Send：GetMarkData;;
Receive：DATA1DATA2;;
```

Note： DATA1- content of field1
DATA2- content of field2...
DATA(take effect only after it has been marked)

8. Get direction of the product

Function description：Get current product direction Settings
Instruction：GetAssemblyLineDir

```
Send：GetAssemblyLineDir;;
Receive：0;;
```

Note： 0-R to L
1-L to R

9. Set production line direction

Function description：Set the current production line direction
Instruction：SetAssemblyLineDir

```
Send：SetAssemblyLineDir,1;;
Receive：Ok;;
```

Note：Send1-L to R；0-R to L
Receive：Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description，take effect only after it has been marked

10. Get moving type

Function description：Get current moving type setup , including： static、 encoder、 fixed speed.
Instruction：GetAssemblyLineType

```
Send：GetAssemblyLineType;;
Receive：0;;
```

Note： 0-encoder mode
1-fixed speed
2-static

11. Set moving type

Function description：Set current moving type , including： static、 encoder、 fixed speed.
Instruction：SetAssemblyLineType

```
Send：SetAssemblyLineType,0;;
Receive：Ok;;
```

Note：Send0-encode mode；1-fixed speed；2-static
Receive：Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description，take effect only after it has been marked

12. Get encoder distance/pulse value

Function description：Get the current encoder pulse distance value, (the value is valid only when the moving type is encoder)

Instruction：GetEncodeParam

```
Send：GetEncodeParam;;
Receive：21.0;;
```

Note：Receive21.0- The value means the distance of each pulse, measured in um/p

13. Set encoder distance/pulse value

Function description：Set the current encoder pulse distance value, (the value is valid only when the moving type is encoder)
Instruction：SetEncodeParam

```
Send：SetEncodeParam,21.0;;
Receive：21.0;;
```

Note：Receive：Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description，take effect only after it has been marked

14. Get fixed speed value

Function description：Get the current fixed speed value, (the value is valid only when the moving type is fixed speed)
Instruction：GetFixSpeed

```
Send：GetFixSpeed;;
Receive：21.0;;
```

Note：Receive21.0- The value means the speed of the product, measured in m/min

15. Set fixed speed value

Function description：Set the current fixed speed value, (the value is valid only when the moving type is fixed speed)
Instruction：SetFixSpeed

```
Send：SetFixSpeed,21.0;;
Receive：Ok;;
```

Note：Receive：Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description，take effect only after it has been marked

16. Get trigger delay

Function description：get the current trigger delay information：delay mode & delay value
Instruction：GetTriggerDelay

```
Send：GetTriggerDelay;;
Receive：1,100;;
```

Note：Receive1-delay mode(0 close，1 distance，2 time)
100-delay value，-/mm/ms

17. Set trigger delay

Function description：get the current trigger delay parameter(delay mode & delay value)
Instruction：SetTriggerDelay

```
Send：SetTriggerDelay,1,100;;
Receive：Ok;;
```

Note：Send1-delay mode(0 close，1 distance，2 time)
100-delay value，-/mm/ms
The delay value can be omitted, after which only the delay mode is set

Receive：Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description，take effect only after it has been marked

18. Get continuous marking interval parameters

Function description：Get the current continuous trigger interval value
Instruction：GetPipIntveralDistance

```
Send：GetPipIntveralDistance;;
Receive：100;;
```

Note：Receive100-interval distance，measured in mm

19. Set continuous marking interval

Function description：Set the current continuous trigger interval value
Instruction：SetPipIntveralDistance

```
Send：SetPipIntveralDistance,100;;
Receive：Ok;;
```

Note：Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description，take effect only after it has been marked

20．Get template list

Function description：Get the list of templates saved in the local directory
Instruction：GetDocList

```
Send：GetDocList;;
Receive：1.bpd;2.bpd;;
```

Note：Receive1.bpd-name of template

21．Open the template

Function description：Open the specified template. This command valid only in edit mode
Instruction：OpenDoc

```
Send：OpenDoc,file;;
Receive：Ok;;
```

Note：file-template name，example：1.bpd
Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description

22．Change the marking template

Function description：change the specified template for marking，This command takes effect only in the marking state. After command is running, if the marking is in progress, it takes effect after the marking is complete. If it is in waiting state, it will takes effect when marking.
Instruction：SwitchDoc

```
Send：SwitchDoc,file;;
Receive：Ok;;
```

Note：file-template name，example：1.bpd
Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description

23. Save current template

Function description：Save the current template. This command takes effect only when the template is not in marking state
Instruction：SaveCurrentDoc

```
Send：SaveCurrentDoc;;
Receive：Ok;;
```

Note：Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description

24. Save the current template as a new file

Function description：Save the current template as a new file，The file is saved in local directory
Instruction：SaveDocAs

```
Send：SaveDocAs,file;;
Receive：Ok;;
```

Note：file-new file name，example：1.bpd
Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description

25. Get the list of current template content

Function description：Gets list of fields for the current template, receiving field names in order
Instruction：GetShapeList

```
Send：GetShapeList;;
Receive：Shape1;Shape2;;
```

Note：Shape1-name of the 1st field，Shape2-name of the 2nd field...

26. Get the field text content

Function description：Get the text content of the specified field,it can get multiple field content at the same time； This instruction is only valid for text, 2D code and Barcode； In the edit state, get the current content, in the marking state, get the last marked content；
Instruction：GetShapeData

Send：GetShapeData,Shape1,Shape2;;
Receive：text1;text2;;

Note：Shape1-the 1st field name；Shape2-the 2nd field name..
text1-content of the 1st field；text2-content of the 2nd field..
Failed,ErrorInfo-failed，ErrorInfo- log description

27．Get field position

Function description：Get the location of a specified field. Multiple field information can be get at the same time. Rect (x,y, W, H) will received, which is the upper-left coordinate of the field and the width and height of the field；
Instruction：GetShapePos

Send：GetShapePos,Shape1,Shape2;;
Receive：Shape1,x1,y1,w1,h1;Shape2,x2,y2,w2,h2;;

Note：Shape1-name of the 1st field，Shape2-name of the 2nd field..
x1,y1,w1,h1-position information of the 1st field；x2,y2,w2,h2- position information of the 2nd field..
Failed,ErrorInfo-failed，ErrorInfo- log description

28．Set the text content, position, and Angle

Function description：Set the text content, position, and Angle parameters of the specified field；
Set text content is only valid for text, 2D code and Barcode；
Set Angle means to rotate the specified Angle value at the current Angle；
This command takes effect immediately after being sent during editing；
If the command is sent during marking, the marking takes effect after the current marking is complete. If it is in the waiting state, the next marking takes effect immediately；
This command takes effect only when the external communication cache mode is disabled
Instruction：SetShapeData

Send：
SetShapeData,shapeName1,text1,posx1,posy1,postype1,angle1,cx1,cy1,0;shapeName2,text2,posx2,posy2,postype2,angle2,cx2,cy2,0;;
Receive：Ok;;

Note： shapeName-field name，When it null, operations are performed on all the fields of the template, but the text content of the field cannot be set
text1- The 1st external data content of the field
posx1- distance traveled in X direction

posy1- distance traveled in Y direction
postype1-0(relative move)，1(absolute move)
angle1-Rotation Angle of the field
cx1- X coordinate of the central point, when it null (comma cannot be omitted)
cy1- Y coordinate of the central point, when it null (comma cannot be omitted)
0- Relative rotation, fixed value, can be omitted

This command can be sent in parts, for example, if only want to change the text content, all data after Text1 can be omitted

Field name can be omitted when there is only one field in the template (comma cannot be omitted)

Example：caution: Fields must contain external data variable text to allow content modification

To change the contents of field 1 to text1 and field 2 to text2, use the following instruction:

SetShapeData,1,text1;2,text2;;

To move all the fields as a whole to the center point, use the following instruction:

SetShapeData,,,0,0,1;;

Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description

29. Send cached data to change field text, position, Angle (visual marking communication application)

Function description：Send cached data to modify field text, position, Angle

This command takes effect only in the marking state and invalid in editing state

This command takes effect only when the external communication cache mode is enabled

The system automatically clears the cache after exit the marking state

Set text content is only valid for text, 2D code and Barcode；

The position movement and Angle are superimposed under the basic state of the field, the offset and rotation amount made last time will not affect the basic state of the field, and the offset and rotation amount made next time will still be in the basic state, please read the Note for specific instructions；

Please perform offset first, then rotation

Instruction：PushShapeData

Send：

PushShapeData,shapeName1,text1,posx1,posy1,postype1,angle1,cx1,cy1,0;shapeName2,text2,posx2,posy2,postype2,angle2,cx2,cy2,0;;

Receive：Ok;;

Note：shapeName-field name，When it null, operations are performed on all the fields of the template, but the text content of the field cannot be set

text1- The 1st external data content of the field
posx1- distance traveled in X direction

posy1- distance traveled in Y direction
postype1-0(relative move)，1(absolute move)
angle1-Rotation Angle of the field
cx1- X coordinate of the central point, when it null (comma cannot be omitted)
cy1- Y coordinate of the central point, when it null (comma cannot be omitted)
0- Relative rotation, fixed value, can be omitted

This command can be sent in parts, for example, if only want to change the text content, all data after Text1 can be omitted

Field name can be omitted when there is only one field in the template (comma cannot be omitted）

Example: When marking integrated with vision system, want move relative (10,20) and then rotate relative to the center point of the whole template (90°). The following instructions can be used

```
PushShapeData,,,10,20,0,90;;
```

Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description

30．Send marking content in real time

Function description：Send the text content of the external data variable；
Only valid for external data variables；
This command takes effect immediately after being sent during editing；
If the command is sent during marking, the marking takes effect after the current marking is complete. If it is in the waiting state, the next marking takes effect immediately；
Instruction：SetVD

```
Send：SetVD,text1,text2,text3...textN;;
Receive：Ok;;
```

Note：text1- External variable text for channel1，text2- External variable text for channel2..
Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description

31．Send marking content in cache mode

Function description：Send the text content of the external data variable；
Only valid for external data variables；
This command takes effect only during marking state；
The sent data will be running in sequence；
Instruction：PushVD

Send：PushVD,text1,text2,text3...textN;;
Receive：Ok;;

Note：text1- External variable text for channel1，text2- External variable text for channel2..
Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description

32．Clear cache（Clear the data of “PushShapeData”）

Function description：Clears all data sent by PushShapeData. This command is used in conjunction with PushShapeData
When the cache is cleared and the data is sent again, the next marking will take effect immediately
Instruction：ClearCache

Send：ClearCache;;
Receive：Ok;;

Note：Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description

33．Modify vector graphics file

Function description：Modify vector graphics file
This command takes effect only in editing state
Instruction：SetVectorgraphData

Send：SetVectorgraphData,Shape1,plt,data;;
Receive：Ok;;

Note：Shape1- Vector graphics field name
Plt-Vector file format, system can support three types: PLT, DXF, AI
Data-Vector file data stream
Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description

34．Get the details of the specified pen number

Function description：Get the details of the specified pen number
Available parameters:MarkSpeed； JumpSpeed； Power； PW； Freq； DotTime
Range of pen numbers:[0,15]
Instruction：GetPen

Send：GetPen,0,MarkSpeed,Power;;
Receive：4000,80;;

Note：0-pen number 0
MarkSpeed,Power-Marking speed and power under the pen number，Multiple items can be entered, separated by commas
Receive:4000-marking speed value，80-power value

35. Set parameter of pen number

Function description：Set parameter of the specified pen number；
Items that can be set:MarkSpeed； JumpSpeed； Power； PW； Freq； DotTime
Range of pen numbers: [0,15]
The setting takes effect when the system enter the marking state
Instruction：SetPen

Send：SetPen,ID,0;MarkSpeed,4000;Power,80;;
Receive：Ok;;

Note：ID- Fixed character, cannot be omitted
,- Separator
0-Pen number
;- Field separator
MarkSpeed,4000-Speed value
Power,80-Power value

Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description

36. Specifies the marking field

Function description：During marking, specify the field to be marked；
This command is valid only in the marking state
After running, it will take effect at the next marking
Instruction：MarkShape

Send：MarkShape,Shape1,Shape2;;
Receive：Ok;;

Note：Shape1，Shape2- The name of the field to be marked. If want to mark the entire template, send “ MarkShape;;”
Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description

37. Start

Function description：Start marking, System will enter marking state
Instruction：StartMark

```
Send：StartMark;;
Receive：Ok;;
```

Note：Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description

38. Stop

Function description：Exit marking state, System will be in Standby mode
Instruction：StopMark

```
Send：StopMark;;
Receive：Ok;;
```

Note：Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description

39. Turn on the red pilot

Function description：run the red light pilot，System will be in position preview mode
Instruction：StartRedMark

```
Send：StartRedMark;;
Receive：Ok;;
```

Note：Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description

40. Manual trigger（simulate the sensor trigger）

Function description：In waiting for trigger state, this command will trigger the system to mark
This command is valid only in marking state(Waiting for trigger)
Instruction：ManualTrgger

```
Send：ManualTrgger;;
Receive：Ok;;
```

Note：Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description

41. Serial number reset

Function description：Resets all SN in the template to their original values
Reset in marking state, takes effect when next marking
Instruction：ClearSN

```
Send：ClearSN;;
Receive：Ok;;
```

Note：Ok-set success；Failed,ErrorInfo-failed，ErrorInfo- log description

42. Feedback the system status

Function description：After the system status changes, the command actively feedback the status change. This command takes effect only when the function is enabled on the system interface
Instruction：MarkStatus

```
Receive：MarkStatus:2;;
```

Note：2-marking state
0- system idle
1- simulated state
2-marking state
3- preview state
4- laser correction state
5- red light correction mode
6- laser forced out
7- rotation marking mode

43. Feedback the number of marking

Function description：After the system completes a marking, the system automatically feedback the number of completed marking. This command must be enabled on the system interface
Instruction：MarkCount

```
Receive：MarkCount:2;;
```

Note：2- Total number of system marking completed

44. Feedback the marking content

Function description：After the marking is complete, System actively feedback the current marking content. This command takes effect only when the function is enabled on the system interface
This command returns only the contents of the current template
Instruction：MarkData

```
Receive：MarkData:text1text2;;
```

Note：text1text2- The text content of the field

45. Feedback the number of missed marking

Function description：When the system detects a missed marking, will actively feedback the number of missed marking. This command takes effect only when the function is enabled on the system interface
Instruction：MissCount

```
Receive：MissCount:1;;
```

Note：1- Total number of missed marking