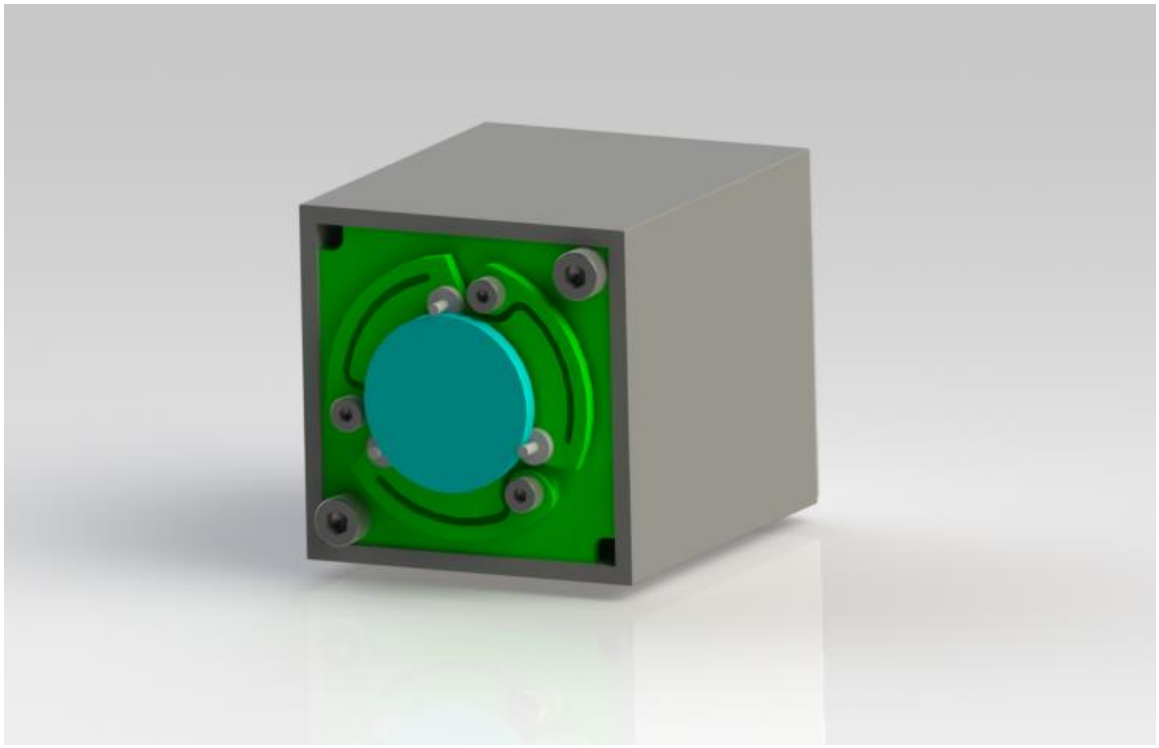
www.newson.be

NEW : CYCLOPS™ CYA-S20
Fully Integrated Laser Beam Deflection Head

Product Brief

The CYA-S20 is a fully integrated laser beam deflection head designed for precision and ease of use. It combines actuator, driver electronics, and control logic in a compact unit, eliminating the need for external steering electronics.



Content

1. Features of CYA-S20

- Compact, high-speed, and accurate 2D laser beam deflection tool
- Fully integrated system — no additional external steering hardware required !
- Complete two-axis actuator with single mirror (CYCLOPS^{TM} design) and driver electronics
- Setpoint steering unit supporting speed-controlled linear interpolation for smooth motion
- Full-field calibration using setpoint mapping tables
- Laser gate output with adjustable delay
- Standard 20 mm round mirror mount
- Simple control via standard OS tools (Explorer, HyperTerminal,…)
- Acts as a flash drive interface for plug-and-play configuration
- DOS-style UART command interface
- Built-in applications for data streaming and standalone operation
- Easy calibration and tuning
- Truly embraces environment: typical low power consumption <3 Watt (250mA at 12 V)!

2. Overview

The actuator includes USB-C and 9-pin SUB-D socket connector.

When connected via USB, CYA-S20 appears as a flash drive. Any standard file browsing/managing application can be used to exchange files with the actuator. Files allow the user to configure and provide data to the system. While the user can decide to create sub directories, the actuator itself can only process files stored in the root. Some (system) files should be on the flash for proper operation.

2.1. Files

1. **CFG.txt**
    - JSON formatted file
    - Defines UART baud rate and X/Y axis orientation
2. **CAL.txt**
    - Text file (CSV)
    - Comprises field mapping data
    - Can be uploaded
    - Can be edited using `CALEDIT`

2.2. Command Line Interface Mode

When the CYA-S20 Flash drive is "ejected" by the host, or when there is no USB connection, the CYA-S20 goes into command line interface mode.

Commands received via the serial link are parsed and executed. All data should be ASCII-encoded and are not case-sensitive. Incoming characters are converted to uppercase and echoed back to the host, allowing any standard terminal program to function as the user I/O device.

The interfaces baudrate, parity and stop bits are set, using the configuration file CFG.TXT (default "57600,n,1").

3. Command Description

3.1. DOS-inspired command syntax

- CHKDSK
- DIR
- TYPE
- VER

3.2. Extended commands

- CALEDIT
- CONFIG
- LOAD
- PATCH
- RESET
- RUN
- TUNE

3.3. CONFIG command

The configuration file (**CFG.txt**, JSON formatted) should be edited on a host computer and copied thereafter into the CYA-S20's flash memory. At power up or when the USB connection with the host is dropped, the system scans the flash for the configuration file. The command CONFIG provides means to verify the configuration.

Configuration File Setup

- The configuration file must be in **JSON format**.
- Edit the file on your **host computer**.
- Copy the file to the **CYA-S20's flash memory**.

When CYA-S20 Loads the Configuration?

- At **startup**, or when the **USB connection to the host is lost**, the system scans the flash memory for the configuration file.

Verify the Configuration

- Use the CONFIG command to check the current configuration

Command Description

- UART sets baud rate, parity and number of stop bits
- DELAY time shift on the Gate output (Pin 3 on SUB D connector)
- MIRRX sets X axis orientation (0 or 1)

- `MIRRY` sets Y axis orientation (0 or 1)
- `SWAPXY` exchanges the X and the Y axes (0 or 1)

Example

```
{
"UART": "57600,n,1",
"DELAY":1000,
"MIRRX":0,
"MIRRY":0,
"SWAPXY":1
}
```

3.4. `CALEDIT` command

To counter image distortions, the applied setpoints must be adjusted to achieve a proper square coordinate system. These adjustment values depend on the beam position and are slightly system-specific. To generate this transfer function, CYA-S20 uses lookup tables defined by mapping data stored in the file **CAL.txt**. All mapping data is presented as ASCII text in 16-bit offset binary hexadecimal format, starting from the top-left point (0000,FFFF) down to the bottom-right (FFFF,0000).

Because the system uses 25 calibration points, the `CAL.txt` file must be exactly 230 bytes. The `CALEDIT` command provides a method to manually edit this file. When executed, the command-line interface is paused, and the UART connection is used for jogging the 25 calibration points.

When started, the gate signal is activated and the deflector moves to mapping point 12 (center position). Calibration points can be selected using arrow-style keystrokes. More details on these keystrokes are provided in the description of the `RUN` command.

Once the desired mapping point is reached, editing begins by sending the character `E` (Edit). In this edit mode, arrow keystrokes no longer select points but shift the current calibration point. Since the laser is active, the live position can be observed. When the correct position is achieved, exit edit mode by sending the character `A`. Arrow style keystrokes are then used again to scroll between calibration points.

After adjusting all calibration points, send the character `S` (Save and Exit) to stop the laser and update the `CAL.TXT` file. Sending the character `X` (Exit) aborts the calibration process without

updating CAL.TXT.

The device is shipped with an 1-to-1 (empty) mapping file. Note the offset binary hexadecimal notation: the first mapping point 0000FFFF represents the top-left position. “Left” corresponds to the minimum X value (0), while “top” corresponds to the maximum Y value (FFFF). At the center (calibration point 12), both X and Y show the value 8000.

Example

```
0000FFFF,4000FFFF,8000FFFF,C000FFFF,FFFFFFFF<r><n>

0000C000,4000C000,8000C000,C000C000,FFFFC000<r><n>

0000C000,40008000,80008000,C0008000,FFFF8000<r><n>

00004000,40004000,80004000,C0004000,FFFF4000<r><n>

00000000,40000000,80000000,C0000000,FFFF0000<r><n>
```

3.5. LOAD command

In RUN mode, vector commands are sent by the host for processing by the actuator CYA-S20. To repeat an image, the same data would normally need to be resent repeatedly. CYA-S20 includes a mechanism to simplify such stamping functions.

Before invoking RUN, tool memory can be preloaded using the LOAD command. This preloaded data can then be fetched in RUN mode by sending the character F followed by the start index. The LOAD command returns the start index and the number of characters added to the tool list. Loading begins at location 1 of the tool table, and tool data from successive LOAD calls is appended.

To copy raw vector commands from the CYA-S20’s flash into the tool list, use LOAD filename.txt.

To load a wobble style tool use LOAD WOBBLE(Amplitude X, Frequency X, Phase X, Amplitude Y, Frequency Y Phase Y)

Example

```
LOAD WOBBLE(10000,50,0,10000,100,90)
```

3.6. RUN command

The RUN command stalls the command-line interface and activates beam-steering mode. CYA-S20 is switched on, and the UART connection is used as the source for deflection commands. All commands and parameters must be sent as plain ASCII text. The data stream is case-insensitive, and any characters not listed below will be ignored.

Character L and M

The vector commands LXXXXYYYY and MXXXXYYYY move the deflector's setpoint toward a target position.

- During the ramp, the gate signal is active when executing a line-to L command
- and inactive when executing a move-to M command.
- XXXX and YYYY represent the X and Y coordinates of the target point, expressed in 16-bit offset binary format.
- Each vector command is always 9 bytes in size.
- Commands do not require separators and may be concatenated.

Character Q

CYA-S20 uses a queue to synchronize received commands with actual mirror movement. If a previous command is still in progress, newly received commands are queued. The queue can hold up to 256 vector commands. To prevent overflow, the host should query the queue state before sending additional vector commands.

- Use the Q command to check the queue status:
  - Returns character b (busy) when the queue contains more than 128 commands.
  - Returns character i (idle) when no commands remain to execute.
  - Returns character r (ready) when the queue has at least 128 empty spaces.

Character Tx

TX Time command:

- This command sets the execution time (in microseconds) for subsequent L (line-to) and M (move-to) commands.
- It does not affect the execution time of commands already queued.
- After receiving the character T, the controller begins reading the time value.
- The command terminates upon receiving the first non-digit character.

Character O

OXXXXYYYY offset command sets the reference point for vector commands.

- XXXX and YYYY are the zero positions
- and have to be presented in a 16 bit offset binary presentation.

Character RA

RA rotation command rotates vector commands:

- After receiving the character A, the controller begins reading the rotation angle in degrees.

- The valid range is **0–359°**.
- Vector commands are rotated around the reference point.
- Additional rotation controls:

  < - Incremental counterclockwise rotation
  > - Incremental clockwise rotation

Character [A, [B, [C, [D

Relative offset commands begin with [followed by a character.

[A relative offset command increments Y offset with 256. This sequence is also generated when the up arrow is pressed on the computers keyboard. By sending the keystroke for the arrow up, the host can shift the reference point upwards.

[B relative offset command decrements Y offset with 256. This is the keystroke for arrow down.

[C relative offset command increments X offset with 256. This is the keystroke for arrow right.

[D relative offset command decrements X offset with 256. This is the keystroke for arrow left.

Character Fx<r>

FX<r> tool fetch command. The system has a 256 KByte tool memory for storing tools. Tools are images that are loaded using the command line function LOAD prior to invoking RUN. After receiving the address for the fetch, the controller starts fetching commands from the tool list. The F command should be terminated by a <r> character.

Character A

A abort command resets offset and image rotation. It also aborts a running tool.

Character X

X exits the runmode and reenables the command line interface

Example

Example for marking a square:

Start and stop at the center position
Size = 50% of the field size
Execution time for 1 circumference = 40,000 µs

```
T10000M40004000L4000C000LC000C000LC0004000L40004000M80008000
```

3.7. VER and PATCH command

VER and PATCH command: CYA-S20 supports on site firmware updating.

- To update, copy the latest firmware (CYSA.BIN) from our website (**www.newson.be**) onto the CYA-S20's flash memory.
- Activate the command line interface (eject the flash drive)
- and invoke the PATCH command to program the deflectors processor.
- The command VER can be used to read out the installed firmware version.

4. Pin Assignments

9pin SUB-D socket connector pin assignment

Pin #	Pin description
Pin 1	0V
Pin 2	Reset input, pull down (open collector) to reset the device
Pin 3	Gate output to control the laser. (3.3V logic, output resistance 50 ohm)
Pin 4	UART TXD output (3.3V logic, pin is high when UART is idle)
Pin 5	0V (power supply input)
Pin 6	5V output to power small electronics (Iout < 200 mA)
Pin 7	3.3V general purpose IO
Pin 8	UART RXD input (3.3V logic)
Pin 9	12V (power supply input, Iin < 1A)

5. LED indicator

Color	Status
Orange	Flash Drive Mode
Red	Terminal Mode
Green	Deflector Running (amplifiers On)
Off	Deflector Running (amplifiers Off)

6. Dimensions