Visualeyez

REAL TIME 3D MOTION TRACKER

VZ10K/10K5

LOW LEVEL CONTROL DEVELOPMENT MANUAL

(For Release With NDA)

(Feb 24 2020)

by

C. C. H. Ma, Ph.D.

PhoeniX Technologies Incorporated

Vancouver, British Columbia, CANADA
info@ptiphoenix.com **http://www.ptiphoenix.com**

TABLE OF CONTENTS

1. Overview

1.1 The *Visualeyez* System

Visualeyez is a high accuracy optical 3D position sensing system. It senses the lights from a point-light source, such as a Light Emitting Diode (LED), to derive essentially three angles of incidence with three optical sensing elements. The sensing elements are referred to as Left (near the power switch), Center and Right Sensing Eyes. Assuming the sensing unit is mounted horizontally, the Left and Right Eyes both sense the yaw angles to the target while the Center Eye senses the pitch angle to the target. Then, together with knowledge of the exact distances between the three Eyes, the system computes by triangulation the x, y, and z coordinates of the light source with respect to its own position and orientation.

The system needs time to acquire sufficient photon energy from the point-light source to generate high-quality signals for accurate high-resolution computation of the 3D coordinates of the target. For higher utility, the system must operate fast with short output latency, so that more target coordinates can be generated to capture complex motions of fast moving objects in real-time. These two requirements oppose each other, the solution is obviously higher efficacy in every aspect of the system development. However, the system is also required to be compact and highly portable without sacrificing operating volume, which entails wide-angle operation, massive physical miniaturization and compact mechanical designs. After years of research and development with the above mutually opposing objectives, the *Visualeyez* motion tracking system is finally born.

The latest *Visualeyez* trackers, VZ10K and its shorter cousin VZ10K5, are capable of detecting and generating more than 10,000 3D target coordinates per second. The data output latency is no more than three target sampling periods. Each of them can operate over 90° in both horizontal and vertical directions. This yields a nearly square-horn shaped sensing volume which is much larger than a round-horn shaped volume of similar sensing angles. This allows a tracker to be neatly mounted in a corner to sense all targets within a rectangular room. High-speed high-density multi-layered multi-processor electronics carry out all computations in pipelined fashion within each sensing unit. Outputs from each tracker are already the 3D (x, y, z) target position coordinates. No external signal processing is required. This makes each tracker a 'practically-transparent optical 3D coordinate measurement machine'.

1.2 Automatic Exposure Adjustment

The optical operation principle of the Sensing Eyes of a *Visualeyez* tracker is similar to that of the sensor in a digital camera. To collect good signal from a target for coordinate computation, each Eye must be exposed to the target light for a well-controlled amount of time, and this time is light-intensity dependent. Should any Eye be over-exposed or under-exposed the computed coordinates may not be accurate or computation may not even be possible. Since every operating environment has a different lighting condition, a target may not point in the same direction or stay in the same position over time, and many different targets may be used within a single capture session, the optimal exposure times required for the

three sensing eyes to sense a target cannot remain fixed normally. To help make the trackers easier to use, an automatic exposure algorithm was developed and built into every *Visualeyez* tracker. The user should leave this automatic- exposure algorithm on whenever possible, and especially when the lighting in the operating environment is powered by alternating current (i.e., AC- powered). Nevertheless, the user can still set the exposure values according to his/her own preferences in very special applications.

1.3 Sample Operation Time

To ensure a desired motion capture frame rate can be achieved, a *Visualeyez* system consisting of the VZ10K and/or VZ10K5 trackers allows the user to specify the maximum amount of time that a tracker can spend to sample the signals from a target. This time is referred to as the Sample Operation Time ('SOT'). The maximum exposure time for sensing the target lights is naturally limited by this SOT setting.

The SOT must range between 2 and 15. The following table shows relationships between the SOT, the actual amount of time that the tracker will spend to sample signals from a target, the maximum possible Eye exposure time, and the maximum possible target sampling rate for a VZ10K or VZ10K5 tracker:

SOT Setting	Sample Operation Time (µs)	Maximum Exposure Time (µs)	Maximum Sampling Frequency (Hz)
2	76.8	64.8	13,020
3	102.4	90.4	9,765
4	128.0	116.0	7,812
5	153.6	141.6	6,510
6	179.2	167.2	5,580
7	204.8	192.8	4,882
8	230.4	218.4	4,340
9	256.0	244.0	3,906
10	281.6	269.6	3,551
11	307.2	295.2	3,255
12	332.8	320.8	3,004
13	358.4	346.4	2,790
14	384.0	372.0	2,604
15	409.6	397.6	2,441

1.4 Targets and Target Control Modules (TCMs)

The VZ10K and VZ10K5 ('VZ10K') series trackers can sense a point-light-source ‘target’ (or interchangeably

'marker') such as a LED with approximately 730~810nm wavelength over a long distance, and a LED outside the preceding range but still within approximately 720~850nm wavelength over a shorter (~60%) distance. To sense the position of a target, the target must be the 'brightest' light source within the tracker's operating space at the time. Each *Visualeyez* system can sense the positions of up to 512 individually identifiable targets/markers one at a time. These are divided into eight groups of 64 markers each. Each 64-marker group is assigned a 'TCMID', and the 64 markers under each TCMID are distinguished by 64 LEDIDs. For a VZ10K system, LED markers can be powered by the PTI K-series Target Control Modules (TCMKs), or operated as the PTI self-identified SIKMarkers (please inquire with PTI on availability), or as the latest semi-wireless OctKMarkers ('Octopus K-series Markers'). The markers powered by TCMKs are referred to as 'Standard Markers'.

To sense the position of a target, that target must flash on command by the tracker. A TCMK can flash up to either 8, 16 or 32 LED markers connected to it, depending on which TCMK model (TCMK8, TCMK16 or TCMK32 respectively) is used. The connectors on a TCMK are assigned certain LEDIDs by jumper or switch settings. When a Standard Marker is connected to a connector on a TCMK, it adopts the LEDID of that connector. Multiple TCMKs can be applied at the same time to control as many Standard Markers as needed.

Each TCMK can flash the Standard Markers connected to it in any sequence as desired by the user. To do this the sequence must first be programmed into each system tracker and then to each TCMK too. Then the exact moment to flash ('fire') a particular marker, or to repeat the sequence, must be communicated to the TCMK by the Master tracker in real-time. These information can be passed to each TCMK by either tetherless or tethered communication means between the Master tracker and each TCMK.

A SIKMarker operates identically to a OctKMarker, and they both operate differently from a Standard Marker controlled by a TCMK. An OctKMarker will flash in a factory-preset fixed order and only when its turn comes up. This 'turn' number is referred to as the 'LED_COUNT' of a SIKMarker or OctKMarker.

1.5 Target Flashing Sequence (TFS)

Only the position of one single target can be sensed at any one time by a *Visualeyez* system. To sense the positions of more than one target, the targets must be flashed one at a time, sequentially. The sequence in which to flash the targets is referred to as the Target Flashing Sequence (TFS). The TFS must be programmed into the tracker(s) as well as into the TCM(s) before the actual position sensing actions can take place.

The TFS consists either of the (TCMID, LEDID, #flash) sets of numbers for the Standard Markers controlled by TCMK(s), or of the (TCMID, LED_COUNT) pairs of numbers for the SIKMarkers and/or OctKMarkers, in the sequence the targets are to be flashed for position sensing ('capture'). '#flash' represents the number of times the same Standard Marker, labeled '(TCMID, LEDID)', is to be captured consecutively before advancing to capture the next target in the sequence. The consecutively captured data can be averaged to increase the marker position sensing accuracy if desired.

Note that 'LEDID' and '#flash' are only meaningful for a Standard Marker controlled by a TCMK. They are meaningless (i.e., #flash must always be 1) for both the SIKMarkers and the OctKMarkers. In fact, the same (TCMID, LEDID) set of a Standard Marker may appear multiple number of times within the TFS and the system would still operate properly! This is for highly specialized applications.

The amount of memory buffer reserved for keeping the TFS is limited. Please see the command details in later sections for additional information.

Normally a *Visualeyez* system simply repeatedly capture the target positions according to the TFS. Going through the TFS once is referred to as one capture 'frame' or 'cycle'.

1.6 Multi-Rate Sampling

A TCMK can be set by software to operate in a 'Multi-Rate Sampling' mode. A LED marker connected to a TCMK set in a Multi-Rate Sampling mode can be flashed more than once (not necessarily consecutively) per capture frame. This will increase the capture frequency and spatial resolution of the captured position of that particular marker by several times, without wasting any system resource capturing the other markers in the same frequency unnecessarily. For example, when swinging a golf club the golfer's body motion is much slower than the club-head motion. To capture all motions in roughly the same spatial resolution, the club-head should be captured at much higher frequency than the body.

Three Multi-Rate Sampling modes are available, in addition to the ordinary (non-multi-rate) sampling mode. SM0 is the ordinary sampling mode. A marker connected to a TCMK set in the default 'SM0' mode will flash at most once per capture cycle.

SM1 is a multi-rate sampling mode in which a standard marker may flash up to two times per capture cycle. A TCMK set in this mode will assign two LEDIDs to one of its connectors. If both LEDIDs are enrolled in the TFS, then the marker connected to that connector will flash twice per capture cycle.

SM2 is a multi-rate sampling mode in which a standard marker may flash up to four times per capture cycle. A TCMK set in this mode will assign four LEDIDs to a connector. Whenever one of these LEDIDs is enrolled in the TFS, the marker connected to it will flash. However, note that a TCMK32 contains 32 connectors, yet only up to 64 LEDIDs can be assigned to any TCMK. So only the lowest 16 of the connectors are each assigned 4 LEDIDs when a TCMK32 is set to operate in the SM2 sampling mode. The remaining connectors will be assigned no LEDID at all, hence any marker connected to these connectors will never get flashed.

SM3 is a multi-rate sampling mode in which a standard marker may flash up to eight times per capture cycle. A TCMK set in this mode will assign eight LEDIDs to each of its lowest eight connectors. Whenever one of the eight LEDIDs assigned to a connector is enrolled in the TFS, the marker connected to that connector will flash. Once again, note that a TCMK32 contains 32 connectors, and a TCMK16 contains 16 connectors, yet only up to 64 LEDIDs can be assigned to any TCMK. So only the lowest 8 connectors are each assigned 8 LEDIDs when a

TCMK is set in the SM3 sampling mode. The remaining connectors will be assigned no LEDID at all, hence any marker connected to any of these connectors will never get flashed.

1.7 Vibrator Activation Sequence (VAS)

A *Visualeyez* 10K system can also activate vibrator-equipped semi-wireless OctKMarkers and the wireless SIKMarkers (please check with PTI for availability) to vibrate upon command by a user application. To activate the vibrator associated with a marker, the user application must first send a command to the tracker with the marker ID, (the (TCMID, LED_COUNT) pair for an OctKMarker or SIKMarker) of the marker associated with the vibrator, to queue it in a Vibrator Activation Sequence (VAS). Thereafter the vibrators in the VAS will be activated, one at a time, in the first-come first-served order. The first vibrator in the VAS will be activated to vibrate when its associated marker is flashed for position sensing (motion capture).

(Henceforth the marker ID of the marker associated with a vibrator shall also be referred to as the ID of the vibrator, which may also interchangeably be referred to as the 'vibrator ID' of the vibrator.)

1.8 Vibrator Activation Failure Recovery

Note that due to the 'first-come first-served' manner of vibrator activation operation, if the marker ID of the first vibrator in the VAS is not also present within the TFS, then the first vibrator will never get activated since its associated LED target will never get flashed! Moreover, if the first vibrator does not get activated, then none of the remaining vibrators behind it in the VAS would ever get activated!

To resolve the above problem, the entire VAS will be wiped clean (cleared) whenever the first vibrator is found to remain un-activated for more than one frame (one capture cycle) period. Needless to say, this means all vibrators associated with the marker IDs behind the first one in the VAS will not get activated either.

To re-activate a vibrator after that, the user must re-enroll its ID into the VAS during a subsequent capture cycle.

1.9 Vibrator Activation Duration

Once a vibrator is activated, it will last for a duration determined by the vibrator-equipped marker implementation. For the latest markers, this is approximately 0.1~0.12 second (100 to 120ms). If another ID is entered into the VAS for the same vibrator, then the vibration duration will re-start from that moment.

2. Operation Requirements

The following requirements must be satisfied in order for a latest *Visualeyez* VZ10K/10K5 tracker to function properly:

1. A RS-422 serial port is properly installed in the host computer.

2. The tracker is connected to the above port through a properly wired cable. (Note that the connector pin assignments of a RS422 serial communication card may be different for different card manufacturers. Hence the pin connections of this cable is not standardized!)

3. The tracker uses a standard 'DB9' 9-pin male connector for communication with the host. The pin names and definitions of this 'Data Connector' on the tracker side are as follows:

PIN	Name	Description
1	TD(B)	Transmit Data B
2	RESET\	Hardware Reset (Pull low >= 1sec to initiate a reset. Wait for >= 2sec after release, for the tracker to reboot)
3	RTS	Request to Send
4	RD(A)	Receive Data A
5	n/c	No connection
6	TD(A)	Transmit Data A
7	RD(B)	Receive Data B
8	reserved	reserved
9	GND	Ground

4. The host computer's serial port must be set to these settings for communication:

Computer Serial Port Settings	
Baud rate	2.5 MBaud
Start bits	1
Data bits	8
Stop bits	1
RTS/CTS handshaking	Disabled

3. Host Commands - Overview

A host can send instruction commands through a high speed RS-422 serial port to a VZ10k series of the *Visualeyez* 3D motion capture trackers. Note that a *Visualeyez* tracker does not contain any command buffer. Each command received from the host is processed immediately. Any command that is received before the preceding command is completely processed may cause the system to behave unexpectedly! Therefore, no command should be sent before the previous one is completed. Should any unexpected behavior occurs, reset the tracker hardware (pull down the RESET\ pin of the tracker connector for >= 1 second) and start all over again.

The tracker instruction commands can be grouped into the following three categories:

Reset Command

Software command to reset part of the tracker hardware.

Settings Commands

Commands to set different system operating options.

Marker Control Programming Commands

Commands to program the target control modules and the target flashing sequence. (These will be referred to as simply “Programming commands” from now on.)

Capture Action Commands

Commands to make the tracker ready for, start or stop the marker position capture operation.

All groups of commands have the same command format, which is described in detail in Section 4.2. Individual commands within each group are described later in Sections 4.6, 4.7 and 4.8.

4. *Visualeyez* System Outputs

Although the host can send many different commands to a *Visualeyez* tracker, each output from the tracker to the host is always 19 bytes long. A 19 byte output set is referred to as either a DATA set or a MESSAGE set. Except for the Action commands, the tracker will send an ACK ('Acknowledgement') or ERR message set back to the host in response to each command it receives. An Action command will of course return one or more data set(s). The 19 bytes are sent least significant byte first (LSB first) by the tracker. Each byte is also sent least significant bit first (LSb first) in accordance to serial communication standards.

4.1 Command Responses

ACK or ERR?

When a *Visualeyez* tracker receives a command from the host, it will first validate (but not fool-proof-check) the command.

- If the command is invalid, the tracker will output an error message (ERR) set to the host, with at least the error code (on the 15th byte), to signal that an invalid command has been received and ignored. At this time, the tracker is ready for receiving another command.

- If the command is valid, the tracker will immediately execute the command. At this time, there is no output yet.

- When the command process is complete, there may or may not be any data generated. In case the command does not generate any data, the tracker will output a message set with at least the ASCII character ACK (ASCII 06h) (on the 15th byte), to the host signaling a valid command has been received and executed. If the command yields any data set, the tracker will not send out an ACK, but will output the data set instead. After outputting either an ACK message set, or a data set, the tracker is ready to receive another command.

- NOTE: To save processing time, not all operation errors are handled, nor are garbage cleaned, if it is useless to handle them or they do not jeopardize system operation.

Figure 4.1 below shows a flowchart of this process, where 'BEL' is equivalent to 'ERR'.

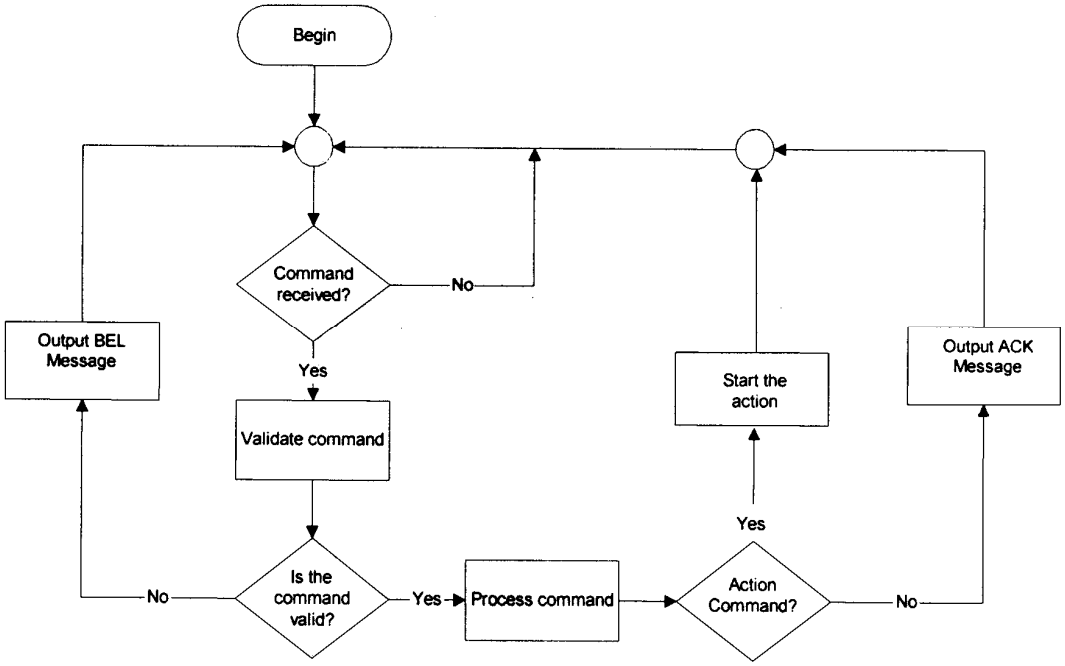


Figure 4.1 _*Visualeyez* Command Process Flowchart

The Data Set

The 19 bytes of a data set consists of the following variables. For each variable, the tracker sends the MSB first.

- **Timestamp** (4 bytes)

  The relative time (in microseconds) at which the target position was captured.

- **Target Coordinates** (3x3 bytes -)

  The X, Y, and Z values (3 bytes each, in l0um unit) of the marker position relative to the center of the center eye of the *Visualeyez* tracker.

- **Status Word** (4 bytes)

  Conditions of the above target position coordinates.

- **Target Identifiers** (2 bytes)

  The LEDID followed by the TCMID (Target Control Module ID) of the marker. (Note that the LED_COUNT of a SIKMarker or OctKMarker is not explicitly outputted. User must correlate a LEDID to a LED_COUNT by him/herself. This can be easily done if the LEDID is always set equal

to the LED_COUNT of a SIKMarker or OctKMarker when designing the TFS (target flashing sequence).)

The Message Set

Except for the first message set following a hardware reset, the 19 bytes of a message set contains much less information because they are primarily for acknowledging completion of a command process. *Visualeyez* does not contain a command buffer. Each command is processed immediately. Any command that is received before the preceding one is completely processed may cause unexpected results! Therefore, no command should be sent before the previous one is completed.

The 19 bytes of a message set consists of the following:

- **Command Code** (1 byte)
  The ASCII character denoting the command just processed.
- **Command Index** (1 byte)
  The ASCII character denoting the 'index' of the command just processed.
- **Reserved** (11 bytes)
- **Message Parameter** (1 byte)
  A binary number associated with the particular Message ID.
- **Message ID** (1 byte)
  Identifies the nature (name) of this message.
- **Reserved** (4 bytes)
  The precise bit or byte definitions are presented in the Data Set Definitions section.

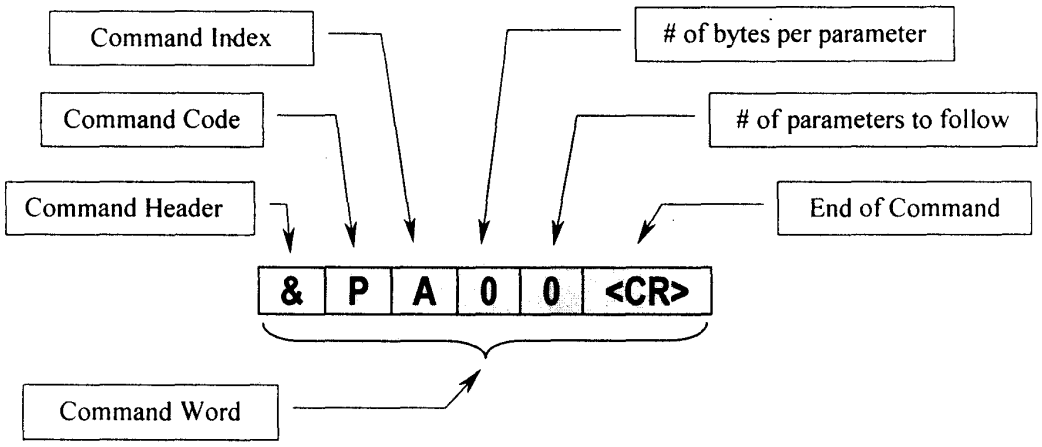
4.2 The Command Format

Every instruction command for a *Visualeyez* tracker is constructed with a 6 byte word, followed by a variable length (possibly none) of parameter(s). Most of the action commands are not followed by any parameter. Most of the setting and programming commands carry parameters consisting of the new setting values to be passed to the tracker.

The format of each command is as follows:

Byte		Description
1	"&"	The "&" (ASCII 26h) character serves as a command header, to indicate the start of a command.
2	Command code	Name of the command, to distinguish one command from another.
3	Command index	Indicates which object(s) is the command meant for.
4	#bytes (/parameter)	Indicates how many bytes will each parameter following the command (if any) consist of. This byte is ignored if no parameter will follow.
5	#parameters	Number of parameters to follow after the command. This byte must be 0 if a command requires no parameter.
6	"<CR>"	A 'carriage return' character (ASCII 0Dh). Indicates the end of the command and the beginning of the parameters if any.
If any	Binary number(s)	Parameter(s), most-significant byte (MSB) of each parameter first if consisting of multiple bytes, as specified in command bytes 4 and 5 above.

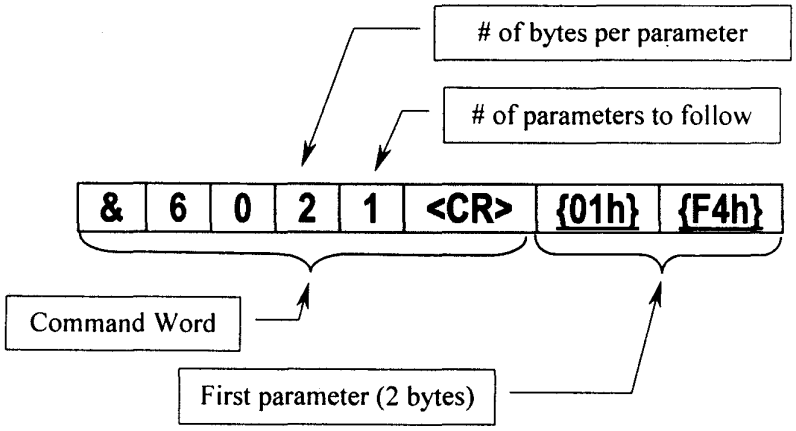
For example, **&PA00<CR>** is a command that enables the "Double Sampling" process for all Eyes of the tracker. It starts with a command header "&", then the command name "P", with command index "A" (for "All"). This command does not require any parameter, so bytes 4 and 5 are both "0". Byte 6 ("<CR>") is a carriage-return character (ASCII 0Dh) to indicate the end of the command word.



Another example, **&p312<CR>{0Ah}{05h}**, is a command carrying two 1-byte parameters. It appends the identification (LEDID) number of a target (marker) connected to a particular target control module (TCM) to the end of the existing target flashing sequence (TFS), and specifies the number of times that this particular marker is to be flashed (and sensed) before flashing the next one. In this example, the command will append marker #10 (LEDID = 0Ah) of TCM #3 (TCMID = 03h) to the end of the existing flashing sequence, and programs it to flash for *5* times consecutively. This command starts with the command header "&", then the name of the command "p", with command index "3" (for TCMID = 03h). In order to specify that marker #10 is to flash 5 times consecutively, two 1-byte parameters must follow the command without any break in between. To indicate that the following two parameters are 1-byte long each, the fourth byte of the command word is "1", and the fifth byte is "2". Following the command, the first parameter specifies the marker's LEDID, 0Ah, and the second specifies the number of consecutive flashes, 05h, to make.

To complete a command, the binary parameter(s) must be sent right after the command word is sent, in the order of the command format. If there is more than 1 parameter to send, the second parameter must be sent right after the first one.

If a parameter's value is greater than 255, then set the number of bytes per parameter to 2 or more, and send the most-significant byte of each parameter first. For example, the command to specify the Number of Cycles to loop through a target flashing sequence is **&60*n*1<CR>{xxh}**, where *n* specifies the number of bytes per parameter, and xx specifies the number (in hexadecimal) of cycles. If value of the parameter requires two bytes to specify, then *n* would be 2. Note that **such setting will also specify that all other parameters, if any, must also be 2 or more bytes long each!** In this example, if the Number of Cycles is 500 (lF4h), for example, then the command should be written as follow:



4.3 Data Set Definitions

The 19 byte data set output by a *Visualeyez* tracker for each target sensed contains the timestamp, X, Y, Z coordinates, status, marker identifier, and target control module identifier as follows.

Byte	Description
(MSB) 1 ~ 4 (LSB)	Timestamp
(MSB) 5 ~ 7 (LSB)	X
(MSB) 8 ~ 10 (LSB)	Y
(MSB) 11 ~ 13 (LSB)	Z
(MSB) 14 ~ 17 (LSB)	Status Word
18	LEDID
19	TCMID

Timestamp

Timestamp is an unsigned 32bit long integer. Its value specifies the precise relative time (in microseconds, μs) at which the position of the target was sensed. For example,

$$T = 000E238Ah = 926{,}602 \quad \Rightarrow \quad 926{,}602\mu s = 926.602ms = 0.926602 \text{ second.}$$

X, Y, Z

The X, Y and Z coordinates of the sensed marker position are each a 24bit long 'signed integer'. Their units are all in ten micrometers (10μm). For example,

$$X = FE29E7h \Rightarrow \quad -120345 = -1203450\mu m = -1203.45mm = -1\ .20345m.$$

Status Word

The Status Word is a 32bit number that contains various information regarding the quality of the signals acquired to compute the **X,** Y and **Z** coordinates, as well as status of the coordinates in the data set. Its bit definitions are as follows: L_a

MSB(14)			(15)			(16)			LSB(17)		
E	HHH	mmmm	111	L_a	AAAA	TTT	L_b	BBBB	TTT	L_c	CCCC

where

E: End-of-frame indicator bit

1 → this is the last data set of the flash (capture) sequence

HHH: Coordinate computation status bits

0 → no error

x → reserved (not used, always 0 for now)

mmmm: Maximum ambient light level sensed by the three eyes.

L_a/L_b/L_c: Right/Center/Left Eye signal level status bit

0 ⇒ no anomaly

1 ⇒ Warning: Data quality low, though signal height good (COR_CORR_LOW)
(=> subjective based, observed captured marker positions mostly good)

AAAA/BBBB/CCCC: Right/Center/Left Eye data processing status bits

0 ⇒ no anomaly

2 ⇒ raw signal weak (peak value low - NUC_PEAK_LOW)
(=> captured marker position maybe noisy)

3 ⇒ processed signal too weak (peak value too low - COR_HUMP_LOW)

4 ⇒ raw signal too high (peak value saturated - NUC_PEAK_HIGH)
(=> captured marker position inaccurate)

5 ⇒ processed signal out of range (cause unknown - COR_SPACING_RANGE)

6 ⇒ signal noisy (not enough info - NUC_HUMPS_FEW)

9 ⇒ signal not type-K (LR indeterminate - COR_ID_INDETERM_LR)

10 ⇒ signal not type-K (UD indeterminate - COR_ID_INDETERM_UD)

12 ⇒ no signal (no distinguishable signal found - NUC_ NOISE_ONLY)

14 ⇒ signal abnormal (center out of range - COR_CENT_OUT_RANGE)

TTT TTT: A 6-bit dumb rotating number ('trigger index') that starts from 1 for the first data set captured. It increments from 1 to 63, rotates back to 0, then repeats 1, 2, 3, ... 63, 0 so on forever, for every data set captured.

LEDID

The LED (Light Emitting Diode) target/marker identifier consists of 7 bits of the 18th byte as shown below.

1	I I I I I I I

where

l l l l l l l: 1 to 64 → the LED identifier ('LEDID').

The constant '1' bit can be used for 19-byte Data/Message set alignment purpose.

TCMID

The TCM (Target Control Module) identifier consists of 4 bits in the 19th byte as follows.

111	**0**	**t t t t**

where

t t t t: 1 to 8 → the TCM identifier ('TCMID').

The constant '1' and '0' bits can be used for 19-byte Data/Message set alignment purpose.

4.4 Message Set Definitions

Except for the very first set (the Initial Message Set) following a system hardware reset, the 19 byte Message Set output by a *Visualeyez* tracker in response to a command or an operating condition contains the command code, command index, message parameter and message identifier as follows:

Byte	Definition
1	Command Code
2	Command Index
3 ~ 13	(Reserved)
14	Message Parameter
15	Message ID
16 ~19	Check Bits

Command Code

The name of the command (see later sections) that resulted in this message set.

Command Index

The index of the command (see later sections) that resulted in this message set.

Message Parameter

The parameter, if any, associated with this message (to facilitate debugging).

Message ID

Additional identifier, if implemented, that distinguishes this message set from another for internal development purposes.

Check Bits

The Status Word is a 32bit number that contains various information regarding the quality of the signals acquired to compute the X**,** Y and Z coordinates, as well as status of the coordinates in the data set. Its bit definitions are as follows:

(16)		(17)		(18)		(19)	
1 1 1	r r r r r	1 1 1	r r r r r	1	r r r r r r r	1 1 1 0	r r r r

where

r: reserved

4.5 Initial Message Set Definitions

Immediately after a hardware reset (including the power-up reset), a *Visualeyez* tracker will output a message set. This Initial Message Set will carry the following information:

Byte.	Content Description
1	'1h'
2	'2h'
3	'3h'
4	'4h'
5	MSB of Tracker Serial Number (if any)
6	7th byte of Tracker Serial Number (if any)
7	6th byte of Tracker Serial Number (if any)
8	5th byte of Tracker Serial Number (if any)
9	4th byte of Tracker Serial Number (if any)
10'	3rd byte of Tracker Serial Number (if any)
11	2nd byte of Tracker Serial Number (if any)
12	LSB of Tracker Serial Number (if any)
13	Reserved
14	Reserved
15	'01h' (Code for 'Initialized')
16	'10h'
17	'11h'
18	'12h'
19	'13h'

The above term 'Tracker Serial Number' (if embedded) consists of the numerical part of the actual tracker serial number only.

5. The *Visualeyez* Commands

5.1 Command Shortlists

This section shortlists the *Visualeyez* tracker command sets for easy reference. The next sections will present them in details.

Reset Command:

Code	Description
&`	Make core processor reset the rest of the tracker hardware

Settings Commands:

Code	Description
&L	Set the Signal Quality Requirement level
&O	Set the Minimum Signal Requirement level
&P	Turn on the 'Double Sampling' process
&Q	Turn on the 'Single Sampling' process (default)
&S	Enable sampling by 'Internal Triggering'
&U	Specify the 'Sample Operation Time' limit
&V	Set exposure to user-specified values
&W	Turn on the 'Automatic Exposure' process (default)
&X	Set multi-rate sampling option for the TCMKs
&Y	Set the auto-exposure feedback control gain
&6	Set the number of cycles (frames) to repeat sampling the markers in the target flashing sequence before stopping automatically (default = 0FFFFFFFFh)
&7	Do nothing, just return an ACK message immediately
&u	To toggle on/off a specified marker for display
&v	To set Sampling Period and Sequence Intermission Period
&^	Enable tether-mode control of the TCMs
&_	Enable tetherless-mode control of the TCMs (default)

Marker Control Programming Commands:

Code	Description
&n	Make all TCMs synchronize on detection of the first TCMID of the target flashing sequence (TFS)
&o	Make all TCMs synchronize on detection of the End Of Frame signal (default)
&p	To clear the existing target flashing sequence OR Append a specific LED marker of a specific TCM, together with a specified consecutive flashing number, to the end of the existing target flashing sequence
&q	To make all TCMs ready to flash their respective first LED targets
&r	To make tracker program the target flashing sequence into all TCMs
&]	To reset all TCMs to their default states and indicate so by flashing once

Capture Action Commands:

Code	Description
&3	Initiate periodic sampling according to the preset Sampling Period and Sequence Intermission Period
&5	Stop the periodic sampling process
&G	Enroll ID of a vibrator for activation when the associated marker flashes
&N	Make tracker wait for an external pulse before starting periodic sampling
&R	Arm the tracker for sampling by 'External Triggering' (default)

Internal Settings Commands:

Code	Description
&^	Enable tether-mode control of TCMs, with specified delay time
&_	Enable tetherless-mode control of TCMs, with specified delay time
&=	Return raw sensor data as results (for factory calibration purposes)
&<	Return 3D coordinates as results (default)
&;	Return both raw sensor data and 3D coordinates as results
&9	Enable refraction compensation (default)
&:	Disable refraction compensation
&Z	Set desired signal peak value for Automatic Exposure to achieve

Internal Action Commands:

Code	Description
&K	Make tracker wait for an external pulse before starting periodic sampling by external triggering (Note: Requires >=1ms between the standby-start pulse and the first external trigger pulse)
&J	Fetch the specified misalignment parameter
&M	Temporarily change the specified misalignment parameter
&x	Burn the current misalignment parameters into the ROM

5.2 Reset Command

Name: **“&`”**

Function:

Make the tracker core processor reset the rest of the tracker hardware.

Command Index:

0 Index is ignored.

Parameters:

0 No parameter is needed.

Comments:

This command will cause the tracker core processor to generate a signal to reset the rest of the tracker system hardware.

Notes:

1. The 'core processor' will reset the rest of the tracker system hardware for 0.1 second, then stays idle for ~1.5 seconds to wait for the hardware to reboot. For continuous proper operation of the tracker core processor, be sure to wait for the tracker to return an ACK message before issuing another command to the core!
2. The tracker core processor itself does not get reset by this command. To reset the entire tracker including the core processor, electronically pull down the RESET\ pin of the Data Connector (see Operation Requirements section).

Example:

To make the core processor reset the rest of the tracker hardware, issue:

&`000<CR>

See Also:

The DB9 Data Connector in the Operation Requirements section.

5.3 Settings Commands

Name: "&L"

Function:

Specify the Signal-Quality-Requirement (SQR) level. This level ('Min_Corr') is for comparison with the signal quality of the sensed marker light samples to be used to derive the 3D marker coordinates, and generate the appropriate status information for each data set.

Command Index:

0 Index is ignored

Parameters:

1 (0~15) One 1-byte parameter specifies the desired minimum SQR value.

Comments:

This command specifies the minimum required signal quality of the marker light samples in order to set the ' L_a ', ' L_b ' and ' L_c ' bits of the Status Word. By checking these bits the user can determine whether a data set was derived from high enough quality signal samples or not.

Notes:

1. This is a coarse test parameter.

Example:

To set the SQR to be 5, issue the command:

&L011<CR>{05h}

See Also:

&O

Name: "&O"

Function:

Specify the Minimum-Signal-Requirement (MSR) level. The peak signal values of the sensed marker light samples are compared with this level before the samples are used to compute the 3D marker coordinates.

Command Index:

0 Index is ignored

Parameters:

1 (0~960) One parameter specifies the MSR value.

Comments:

This command specifies the minimum required level that the peak-signal values of the marker light samples must exceed before the 3D marker position is computed. The results of the comparison for the three Eyes are also reflected in the 'data processing status bits' of the Status Word of a data set.

Notes:

1. A peak-signal value of ~300 is normally good enough for computing a marker's position. So the MSR should be set lower than this. The factory default MSR value is 102.

Example:

To set the MSR to be 280, issue the command:

&O021<CR>{01h}{18h}

See Also:

&L

Name: "&P"

Function:

Turn on 'Double Sampling' (turn off 'Single Sampling') process for all Eyes.

Command Index:

0 Index is ignored

Parameters:

0 No parameter is needed.

Comments:

This command turns on the 'Double Sampling' process for all Eyes to sample the marker lights. With Double Sampling, two samples of the marker lights are acquired consecutively for each marker, one when the marker is still off, another when the marker is on. Then the off-sample is subtracted from the on-sample to obtain the signal for computing the marker position.

Notes:

1. This function is not yet available.
2. Twice as much Sample Operation Time ('SOT') is required for this function. Hence the maximum sampling rate is reduced accordingly when this function is on.
3. A continuous light source will not be sensed due to the subtraction operation.

Example:

To turn on Double Sampling for all Eyes, issue the command:

&P000<CR>

See Also:

&Q, &U

Name: "&Q"

Function:

Turn on 'Single Sampling' (turn off 'Double Sampling') process for all sensing Eyes.

Command Index:

0 Index is ignored

Parameters:

0 No parameter is needed.

Comments:

This command turns on the normal 'Single Sampling' process for all Eyes to sample the marker lights. With Single Sampling, only one sample of the marker lights is acquired by each Eye. Hence the tracker may be more sensitive to ambient lights. However, a continuous point light source, e.g., a flash light, maybe sensed and its position computed.

Notes:

1. This is the default tracker sampling process.

Example:

To turn on the Single Sampling process, issue:

&Q000<CR>

See Also:

&P

Name: "&S"

Function:

Turn on the 'Internal Triggering' (turn off the 'External Triggering') process.

Command Index:

0 Index is ignored.

Parameters:

0 No parameter is needed.

Comments:

Enable the marker sampling process to be triggered by an internal signal generated per the user-specified Sampling Period.

Notes:

1. System Default: External Triggering (Internal Triggering turned off)
2. The internally generated trigger pulses are output to the EXT TRIG jack on the tracker rear panel.

Example:

To turn on the Internal Triggering process, simply issue:

&S000<CR>

See Also:

&R

Name: "&U"

Function:

Set the Sample Operation Time ('SOT') that the tracker can spend to sample the light signals of a marker before computing its 3D coordinates.

Command Index:

0 Index is ignored

Parameters:

1 (2~15) Specifies the SOT value.

Comments:

This command specifies the amount of time that the tracker shall spend to sample the light signals of each marker. The exact physical amount of time that the tracker will spend is:

Sample Operation Time = (SOT+1) * 256 * 0.1μs

Notes:

1. The Eye exposure times can never exceed this amount. Hence in case an Eye needs to be exposed to a longer time duration than the SOT allows, the SOT value must be increased first.
2. If the exposure times are all shorter than the SOT allows, the tracker will simply idle until the SOT is exhausted.

Example:

To set the SOT to be 3, issue the command:

&U011<CR>{03h}

See Also:

&V, &W

Name: "&V"

Function:

Enable manual exposure of the three sensing Eyes using the specified Exposure time(s) ('Manual Exposure'), and disable the 'Automatic Exposure' function.

Command Index:

0 Index is ignored

Parameters:

3 Specify the desired exposure times in one-tenth of a microsecond (0.1µs) unit:
Parameter 1: (1 ~ (256*SOT + 136)) Exposure time (in 0.1µs unit) for the Right Eye
Parameter 2: (1 ~ (256*SOT + 136)) Exposure time (in 0.1µs unit) for the Center Eye
Parameter 3: (1 ~ (256*SOT + 136)) Exposure time (in 0.1µs unit) for the Left Eye

Comments:

This command specifies the individual exposure values (in 0.1µs unit) that the three sensing Eyes must use to sample signals from a marker.

Notes:

1. The maximum applicable exposure value is limited by the 'Sample Operation Time' (SOT), which should be set (using the &U command) prior to issuing this command.
2. The best values to use are dependent on the instantaneous ambient lighting conditions as well as the position and orientation of the marker(s) to be sampled. Experiments are required for each different application.
3. Time-varying ambient lights (e.g., 60Hz indoor lighting) may cause time-varying performance.
4. This function is meant for highly controlled lighting environments only, not recommended for use by general users.

Example:

To set the SOT to be 3, followed by enabling manual exposure of the Right Eye for 70µs, Center Eye for 55.5µs, and the Left Eye for 38µs, issue:

&U011<CR>{03h}
:
&V023<CR>{02h}{BCh}{02h}{2Bh}{01h}{7Ch}

See Also:

&U, &W

Name: "&W"

Function:

Turn on the 'Automatic Exposure' function to independently adjust the most appropriate exposure time for each Eye to sample light signals of a marker automatically.

Command Index:

0 Index is ignored

Parameters:

0 No parameter is needed

Comments:

Turns on the dynamic feedback controlled Automatic Exposure algorithm. The tracker will determine the optimum Exposure value to apply to each sensing Eye independently to sample each different marker.

Notes:

1. Tracker Default: Automatic Exposure enabled.
2. The maximum amount of exposure time is limited by the SOT setting. Please see the table in the Sample Operation Time section.

Example:

To turn on Automatic Exposure (and disable Manual Exposure) function, simply issue:

&W000<CR>

See Also:

&U, &V

Name: "&X"

Function:

Make all TCMKs of the specified TCMID operate in the specified multi-rate sampling mode.

Command Index:

0 Index is ignored

Parameters:

1~8 1-byte (0~3) parameter(s) each specifies either the ordinary sampling mode (SM0) or a multi-rate sampling mode (SM1~SM3) for the TCMK(s) with TCMIDs 1 to 8 respectively.

Comments:

For a VZ10K *Visualeyez* system, the target control modules (TCMKs) can be software-programmed to flash the connected standard markers for multi-rate sampling purposes. All TCMKs with the same TCMID will be set to the same sampling mode at once. Hence up to eight parameters are possible for this command.

Notes:

SM0 - (Default) the ordinary (non-multi-rate) sampling mode. A standard marker connected to a TCMK set in this sampling mode will flash at most once per capture cycle.

SM1 - a multi-rate sampling mode. A marker will flash up to two times per capture cycle in this mode.

SM2 - a multi-rate sampling mode. A marker will flash up to four times per capture cycle in this mode.

SM3 - a multi-rate sampling mode. A marker will flash up to eight times per capture cycle in this mode.

Example:

To set all TCMK(s) with TCMID = 1 to operate in the SM2 mode, all TCMK(s) with TCMID = 2 to operate in the SM0 mode, and all TCMK(s) with TCMID = 3 to operate in the SM3 mode, issue:

&X013<CR>{02h}{00h}{03h}

See Also:

See the Multi-Rate Sampling section.

Name: "&Y"

Function:

Set the dynamic feedback control gain for the Automatic Exposure algorithm.

Command Index:

1 for the Left Eye only
2 for the Center Eye only
3 for the Right Eye only
A for all Eyes

Parameters:

1 (1~15) Specifies the desired gain value.

Comments:

This command specifies the desired auto-exposure feedback control gain value for one Eye or for all three Eyes at once. The higher the 'gain' value, the faster the automatically adjusted exposure value will try to approach the reference value required to attain optimal signal strength for computation of the marker position. However, in case a marker's motion causes an abrupt change in its light signal conditions, the higher gain may cause the auto-exposure adjustment to overshoot and yield a signal strength that is worse for position computation than without the adjustment. Hence a compromise should normally be adopted in setting this gain.

Notes:

1. Default Gain Setting: 8
2. Only the lower 4 bits of the specified value will be used. In case of 0, it will be changed to become 1.

Example:

To set the auto-exposure feedback gain to 11 for all three Eyes, issue the command:

&YA11<CR>{0Bh}

See Also:

&W

Name: "&6"

Function:

Set the Number of Cycles (frames) to repeat sampling the markers in the target flashing sequence (TFS) before stopping automatically.

Command Index:

0 Index is ignored.

Parameters:

1 (l ~ 0FFFFFFFh) Specifies the cycle number.

Comments:

This command sets the number of cycles to loop through the target flashing sequence and sense the marker positions, before stopping automatically. To initiate the actual sampling, invoke either **&2**, **&3**, **&N** or **&R**.

Notes:

The system default is -1 (0FFFFFFFh).

Example:

1. To set the Number of Cycles to sample targets in the target flashing sequence to 256, the command is:

   &6021<CR> {01h} {00h}

2. To set the Number of Cycles to be 120,000 (01D4C0h), the command is:

   &6031<CR>{01h}{D4h}{C0h}

See Also:

&2, &3

Name: "&7"

Function:

Do nothing, just return an ACK message set.

Command Index:

0 Index is ignored.

Parameters:

0 No parameter is needed.

Comments:

This command is for prompting a *Visualeyez* tracker for a response to indicate proper connection and the core system firmware operation.

Example:

To test whether a tracker is connected properly and can communicate with the host computer, simply issue the command:

&7000<CR>

Name: "&u"

Function:

Toggle a specified marker ON or OFF for visual spotting of its whereabouts.

Command Index:

1~8 Specifies the TCM_ID of the marker to toggle.

Parameters:

1 1-byte (1~64) parameter to specify the LED_ID of the marker to toggle.

Comments:

This command toggles a specified marker on or off. This allows the user to visually verify its location. It eliminates the need to manually keep track of the whereabouts of a marker before connecting the marker cable to a TCM. Hence a marker can be arbitrarily placed on a capture subject, and its cable simply plugged into any connector on a TCM for operation. Then when necessary, use this command to visually find out the location of a particular marker.

Notes:

1. This command does not cause the tracker to sample the marker's position.
2. The specified marker will remain ON (or OFF) until another **&u** command is invoked.

Example:

Assume target *#5* of TCM #8 is OFF initially. To turn it on, the command is:

&u811<CR>{05h}

Name: "&v"

Function:

Set the Sampling Period and Sequence Intermission Period for a periodic marker sampling process.

Command Index:

0 Index is ignored.

Parameters:

2 Parameter 1: (the minimum ~ 0FFFFFFFFh) Sampling Period (in µs)
Parameter 2: (0 ~ 0FFFFFFFFh) Sequence Intermission Period (in µs)

Comments:

Sampling Period

- The uniform time interval between two consecutive marker flashes/captures.

Sequence Intermission Period

- The inactive time interval between two complete cycles of flashing/capture of all the targets in the target flashing sequence.

Notes:

System Default: Indeterminate

Example:

Assume the desired target flashing sequence (together with the number of times that each is to be flashed consecutively before advancing to the next target) has been programmed (see **&p**). To sample the sequence of targets at 1.0ms (=03E8h µs) interval, and recess for 2.5s (= 2625A0h µs) after completing the sequence before repeating, the command and its two parameters are:

&v032<CR>{00h}{03h}{E8h}{26h}{25h}{A0h}

See Also:

&3, &5

Name: "&^"

Function:

Setup the system for tether-mode operation.

Command Index:

0 Index is ignored.

Parameters:

0 No parameter is needed.

Comments:

This command will merely change the timing in sending out the marker flashing signals to the TCMs.

Notes:

1. System Default: Tetherless mode enabled (tether-mode disabled).
2. A control cable (tether) must be connected from the CTRL OUT jack on the rear panel of the master tracker to every TCM (in parallel) in order to control them.

Example:

To control the TCMs in tether-mode, simply issue the command:

&^000<CR>

See Also:

&_

Name: "&_"

Function:

Setup the system for tetherless-mode control of the TCMs.

Command Index:

0 Index is ignored.

Parameters:

0 No parameter is needed.

Comments:

This command will merely change the timing in sending out the marker flashing signals to the TCMs.

Notes:

1 System Default: Tetherless mode enabled (tether-mode disabled).

2 A PTI supplied radio transmitter must be connected to the CTRL OUT jack on the rear panel of the master tracker, and a matching receiver must be connected to every TCM (in parallel) in order to control them.

3 If a tetherless operation is disrupted by radio noise interference, the target flashing sequence may go out of step with the tracker sensing sequence for as long as the interference persists. Should this happen, one or more frames (flashing sequence cycles) of results may appear to be ruined when in fact only the LED_ID(s) and TCM_ID(s) may be scrambled due to the interference. The computed marker coordinates should still be accurate and useful. Most of them should be salvageable by correcting the LED_ID(s) and TCM_ID(s).

Example:

To setup the system for tetherless operation, simply issue the command:

&_000<CR>

See Also:

&^

5.4 Marker Control Programming Commands

Name: "&n"

Function:

Make all TCMKs (for the standard markers) re-synchronize with the tracker on detection of the first-TCMID of the target flashing sequence (TFS), instead of the default End-Of-Frame signal.

Command Index:

0 Index is ignored.

Parameters:

0 No parameter is needed.

Comments:

This function allows the smart hot-swappable SIKMarkers and/or OctKMarkers to operate properly within the first-TCMID group. Previously a different TCMID group must precede the group(s) the SIKMarkers and/or OctKMarkers belong to, or else the '864' button must be pressed to create the first-TCMID group with only a virtual marker '864'. Now these smart markers can belong to the first-TCMID group and still operate well, if this command is issued prior to programming the TCMKs with the &r command.

Notes:

1. This command must be issued before the &r command in order to be effective.
2. The first-TCMID should appear uniquely, no more than once, within the TFS.
3. Those markers specified in the TFS, following the second appearance of the 'first-TCMID' (if any), will never light up and hence never be captured!

Example:

To make TCMKs synchronize with the tracker on detection of the first-TCMID, issue:

&n000<CR>

See Also:

&o, &r, Sec. 1.4

Name: "&o"

Function:

Make all TCMKs (for the standard markers) re-synchronize with the tracker on detection of the End-Of-Frame signal (default), instead of the first-TCMID of the target flashing sequence (TFS).

Command Index:

0 Index is ignored.

Parameters:

0 No parameter is needed.

Comments:

This function will return all TCMKs back to the default operation mode in which they will re-synchronize with the tracker on detection of the End-Of-Frame signal. In this operation mode, the hot-swappable SIKMarkers and/or OctKMarkers belonging to the first-TCMID group will not function properly. To avoid this problem, a virtual marker with a TCMID not associated with any of the problem markers may be added as the first-TCMID group of the TFS (as the '864' button for the VZ4000 systems did). However, this operation mode allows all TCMIDs in the TFS to be repeated multiple number of times if desired.

Notes:

1. This command must be issued before the &r command in order to be effective.
2. A hot-swappable SIKMarker or OctKMarker belonging to the first-TCMID group of the TFS will not work properly in this operation mode.

Example:

To return all TCMKs back to re-synchronize with the tracker on detection of the default End-Of-Frame signal, simply issue:

&o000<CR>

See Also:

&n, &r, Sec. 1.4

Name: "&p"

Function:

Clear the existing target flashing sequence (TFS)

OR

Append a (LEDID, #flash) pair to the existing TFS, so that the physical marker with the specified ID will later be flashed and sampled by the tracker, for the #flash number of times consecutively, before advancing to flash and sample the next marker in sequence.

Command Index:

1~8 Specifies the TCMID of the specified marker being appended.

Parameters:

0 Clear the existing target flashing sequence when no parameter is specified.

OR

2 Two 1-byte (LEDID, #flash) pair specify the marker and the desired number of consecutive flashes (and samplings) to make for the marker, where

LEDID: (1~64) This, together with the TCMID, specify the marker being appended to the existing flashing sequence.

#flash: (1~255) The number of consecutive times to flash the specified marker before advancing to flash the next marker.

Comments:

When Parameters = 0, this command will cause the entire existing TFS to be 'cleared'.

When Parameters > 0, the tracker assumes that the parameters following the command word will come in a pair defined as (LEDID, #flash) and will append them to the existing TFS.

Notes:

3. Invoke this command for as many times as desired to create a target flashing sequence for sampling multiple markers.
4. A standard marker can be specified more than once within the TFS. A SIMarker or OctMarker can be specified more than once too, but there must be a TCMID change between the repetition.
5. Consecutive invocations with both the TCMID and LEDID being identical (for standard markers only) will result in the #flash counts being combined and treated as one single invocation.
6. Non-consecutive invocations with both the TCMID and LEDID being identical will not be combined and will be treated as different markers by the sampling process. This is applicable

to standard markers only. See note 2 above.

7. For a given TCMID, the maximum number of (LEDID, #flash) pairs that can be present in the target flashing sequence is limited to 64.
8. The maximum number of TCMID changes that can be counted going through the target flashing sequence is limited to 64. The start or re-start of the sequence is considered to have made a TCMID change.

Example:

The following commands

&p000<CR>
&p512<CR>{08h}{3}
&p712<CR>{3Bh}{1}
&p512<CR>{0Ah}{9}
&p512<CR>{0Ah}{6}
&p512<CR>{06h}{2}
&p712<CR>{3Bh}{2}

will first clear the previous flashing sequence, then reconstruct it to be

{{08h}{3}{3Bh}{1}{0Ah}{0Fh}{06h}{2}{3Bh}{2}}.

In this flashing sequence, the first-TCMID is 5. The total number of (LEDID, #flash) pairs under TCM5 is 3, that under TCM7 is 2. The total number of TCMID changes is 4.

See Also:

&r, &q, &v

Name: “&q”

Function:

Command the tracker to make all TCM(s) ready to flash their respective first markers per the target flashing sequence.

Command Index:

0 Index is ignored.

Parameters:

0 No parameter is needed.

Comments:

This command readies all TCM(s) to flash their respective first targets according to the programmed target flashing sequence.

Notes:

This command should be invoked only AFTER the target flashing sequence has been programmed into all TCM(s) with the **&r** command, and before issuing an action command. Immediately after sending this command, the TCM(s) will be ready to flash their respective targets according to the target flashing sequence (pending only the tracker receiving an action command).

Example:

To make the tracker program all the TCM(s) and then synchronize them to the start of the target flashing sequence, invoke the commands:

&r000<CR>
&q000<CR>
.
.
&3000<CR>

See Also:

&p, &r, &]

Name: "&r"

Function:

Command the tracker to program the target flashing sequence into all TCM(s).

Command Index:

0 Index is ignored.

Parameters:

0 No parameter is needed.

Comments:

This command will order the tracker to program all the TCM(s) with the target flashing sequence previously constructed with the **&p** commands.

Notes:

This will replace any previous target flashing sequence stored in each TCM.

Example:

To make the tracker program all the TCM(s) and synchronize them to the start of the target flashing sequence, invoke the commands:

&r000<CR>
&q000<CR>

See Also:

&p, &q, &]

Name: "&]"

Function:

Command the tracker to make all TCM(s) reset to their default states, and indicate this by flashing all the markers one time.

Command Index:

1 Index is ignored.

Parameters:

1 No parameter is needed.

Comments:

This command will order the tracker to make all TCM(s) reset to their initial states as if the TCM(s) have just been powered up.

Notes:

This command will finish with the TCMs flashing all of their respective markers for one flashing-period.

Example:

To make the TCM(s) reset to their initial states, invoke the command:

&]000<CR>

See Also:

&p, &q, &r

5.5 Capture Action Commands

Name: "&3" ('Periodic Sampling')

Function:

Initiate a periodic (pipelined) sensing process with the SOT, Sampling Period and Sequence Intermission Period previously specified by the **&v** command.

Command Index:

0 Index is ignored.

Parameters:

0 No parameter is needed.

Comments:

This command initiates periodic sampling of the targets in the target flashing sequence.

Notes:

1. No acknowledgement will be generated by the tracker upon receiving this command.
2. It takes a certain amount of time for the tracker to switch from flashing markers with one TCMID to flashing markers with a different TCMID. Therefore, when the sampling period is set too short, a sampling process may be skipped automatically (without sensing any target) to allow time for making the TCM switch. This skip will happen for sure if the sampling period (see **&v**) is shorter than a certain length. This length is a function of the SOT (see **&U**) and the largest TCMID (see **&p**) involved in the target flashing sequence.

Example:

To initiate a periodic sensing process:

&3000<CR>

See Also:

&5, &6, &N

Name: "&5" ('Stop Sampling')

Function:

Stop the periodic sensing process.

Command Index:

0 Index is ignored.

Parameters:

0 No parameter is needed.

Comments:

This command will stop the periodic target sensing process initiated by **&3** or **&N**.

Notes:

The process will also stop automatically if the previously specified Number of Cycles (frames; see **&6**) is reached.

Example:

Assuming the system is currently periodically sensing positions of the targets according to the programmed target flashing sequence, to stop the process immediately, issue the following command:

&5000<CR>

See Also:

&3, &6, &N

Name: **"&G"** ('Activate Vibrator')

Function:

Appends in real-time the ID of a vibrator (TCMID, VIBID pair) to the existing vibrator activation sequence (VAS) to cause the vibrator to be activated the next time its associated marker is flashed. VIBID is the same as the LEDID of the marker. Note that the duration for which the vibrator will stay activated depends on the vibrator implementation.

Command Index:

1~8 Specifies the TCMID of the vibrator being appended.

Parameters:

1 1-byte (1~64) parameter to specify the VIBID of the vibrator to be activated.

Comments:

This instruction command is normally issued by a user application software in real-time during a capture. To avoid interfering with the more important marker position sensing (motion capture) process, this command should be issued only when the tracker is not very busy capturing the marker motions. This prompts the following notes.

Notes:

1. No acknowledgement will be generated by the tracker upon receiving this command.
2. The maximum number of vibrator IDs (vibrators) that can be accumulated in the VAS is limited to 64. As soon as a vibrator is activated, its ID is erased from the VAS.
3. The vibrators in the VAS are activated one at a time in the first-come first-served order. The first vibrator in the VAS is activated when its associated LED marker is flash for position sensing.
4. When activated, a vibrator will vibrate for at least one frame (capture cycle) period, but the exact duration depends on the vibrator implementation, then stop. To make it continue to vibrate, the user must re-enroll its ID with the VAS using this command, before the vibrator stops.
5. If a vibrator ID in the VAS is not also present in the TFS, that vibrator and all those behind it will never get activated. To resolve this stuck problem, the entire VAS will be cleared after one frame period. To activate a vibrator after that, its ID must be re-enrolled with the VAS afresh.
6. This command should be issued only when the tracker is not busy capturing the marker motions. Therefore it is recommended that it be issued only when the motion capture frequency is below maximum or during the Intermission Time between capture cycles.

Example 1:

The following commands

&G311<CR>{08h}
&G211<CR>{40h}
&G711<CR>{15h}

will construct a VAS as

{{308}{264}{721}}

and cause the vibrators associated with markers 208, 264 and 721 to be activated one by one, in exactly the same order. Assuming these three IDs are also present in the TFS, then the activations may take place over one, two or up to three frame periods, depending on how they are ordered within the TFS and when the tracker received the three commands.

Example 2:

Assuming the TFS consists eight vibrator-equipped markers as follows:

{{201}{202}{203}{204}{205}{206}{207}{208}}

Issuing the following commands during an intermission period during motion capture ...

&G211<CR>{06h}
&G211<CR>{03h}

will activate vibrator 206 within the first capture frame period following commands issuance, then vibrator 203 will be activated in the next capture frame period.

Example 3:

Assuming the same TFS as in Example 2, issuing the following commands during an intermission period during motion capture ...

&G211<CR>{03h}
&G211<CR>{06h}

will activate both vibrators 203 and 206 within the same first capture frame period following commands issuance.

See Also:

&p, and sections on 'Vibrator'

Name: **"&N"** ('External Start Periodic Sampling')

Function:

To make the tracker wait for an external pulse before starting a periodic sampling process

Command Index:

0 Index is ignored

Parameters:

0 No parameter is needed.

Comments:

After invoking this command, the user must apply one, and only one, pulse ('Start Pulse') to the EXT TRIG jack on the rear-panel of the tracker to make it start sampling periodically according to the pre-specified Sampling Period and Sequence Intermission Period.

Notes:

1. No acknowledgement will be generated by the tracker upon receiving this command.
2. Until receiving the Start Pulse, the tracker will idle and does not generate any response to the host.
3. The Start Pulse must be applied through an 'Ext Start Input' adaptor available from PTI. The pulse must be normally-high, negative-going, TTL, and 50ns~1μs in width (see the tracker Specifications).

CAUTION – PERMANENT DAMAGE RISK !!!

Applying improper signals of any kind, or not applying signal(s) through an Ext Start Input adaptor, into a *Visualeyez* tracker may cause PERMANENT DAMAGES to the tracker and will VOID THE WARRANTY! A professional electronic engineer must be employed to apply any signal into a tracker for any purpose whatsoever.

Example:

To put a tracker in 'external-start periodic-sampling' mode, invoke the following command then apply one single pulse to the EXT TRIG jack according to the above comments and cautions:

&N000<CR>

See Also:

&3, &5, &v, &R

Name: **"&R"** ('External Triggering')

Function:

Turn on the 'External Triggering' (turn off the Internal Triggering) process, to enable the tracker to sample markers in synchronism with an externally generated pulse train.

Command Index:

0 Index is ignored.

Parameters:

0 No parameter is needed.

Comments:

This function enables sampling marker positions in synchronism with electronic pulses applied to the EXT TRIG jack ('External Triggers') on the rear-panel of the tracker. Normally this is for synchronizing several PTI trackers to operate as a 'Multi-Tracker System' (MTS). However, the user may use this function to synchronize the sampling operation with a user-supplied external pulse train. Should this be the case, NOTE that the pulses MUST BE applied through an 'Ext Start Input' (ESI) adaptor available from PTI. The pulses must be normally-high, negative-going, TTL pulses. The sampling is synchronized with the trailing rising edge of each pulse.

Notes:

1. No acknowledgement will be generated by the tracker upon receiving this command.
2. This should be the last command to apply before the tracker receives the External Triggers.
3. System Default: External Triggering enabled (Internal Triggering disabled).
4. System operation is still subject to all constraints applicable to an Internal Triggering process. For example, the time interval between two pulses must still be greater than or equal to the minimum allowed internal sampling period.
5. After invoking this function, the tracker will idle to wait for the External Triggers to initiate a capture process. Until then, the tracker will not generate any response to the host.

CAUTION – PERMANENT DAMAGE RISK !!!

Applying improper signals of any kind, or not applying signal(s) through an Ext Start Input adaptor, into a *Visualeyez* tracker may cause PERMANENT DAMAGE to the tracker and will VOID THE WARRANTY! A professional electronic engineer must be employed to apply any user-supplied signal into a tracker for any purpose whatsoever.

Example:

To enable External Triggering, simply issue:

&R000<CR>

See Also:

&S