OLR Serial Protocol (Board ← → Host)

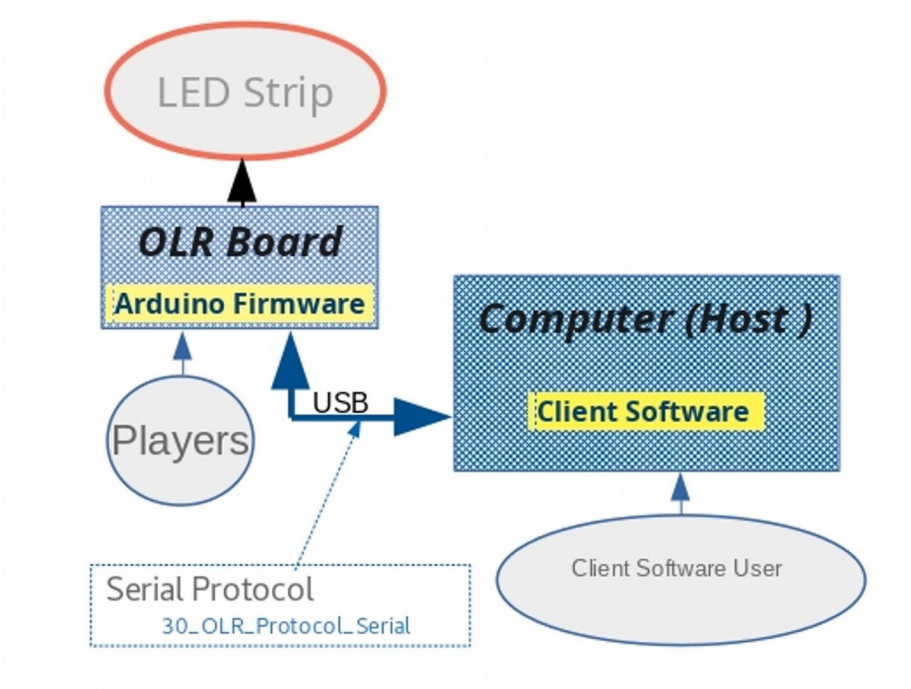
Table of Contents

Terminology

In the present doc the terms ***Board*** and ***Host*** indicate:

- ***Board***: OLR Board - The microcontroller managing the led strip (Arduino)
- ***Host***: A Software running on a computer connected to the Board via Serial interface (Currently only USB).

  The software can be "*OLR Update/Configuration Utility*", "*OpenLEDRace Network Client*", etc.



The software running on Host and the Firmware on Board communicate using the Protocol described in the present document.

Implementation notes

Message format

Messages are "arrays of chars" composed by:

- **Command_id**: one char identifying the command, case sensitive
  - Optional **Parameters** (not every command have parameters)
- If a command have parameters, the first one comes immediatly after the command_id (no separators).
- In case command have more than one parameter, the following ones are comma-separated
  - **EOC** - End Of Command char:
  - **ASCII 10/0x0A** = Line Feed = new line = '\n'
- Example: **C1,3,1,1**\n
- Command "**C**" (*race configuration*) with parameters, ended with the ***EOC*** character

Commands List

Cmd	Description	Notes
#	Protocol Handshaking	Host-Board handshake on startup
@	Enter Configuration mode	Host request the Board to **enter in configuration mode**
~	Leave Configuration mode	Host request the Board to **leave configuration mode**
:	Set Unique ID	Set Board Unique ID
$	Get UID	Get Board Unique Id
?	Get Software Id (Type)	Get Board Software Type (OLR, OLRNet,...) *--see specific doc with defined SoftwareType IDs*
%	Get Software Version	Get Board software version
!	Send log/error msg	Send a log/error message to peer
C	Race **C**onfiguration	Set main race configuration parameters
T	Track length configuration	Command used to configure the Total Number of LEDs in the track.
B	Box length configuration	Command used to configure the PitLane (Box)
A	Ramp (Slope) configuration	Command used to configure the Ramp (Hill, Slope)
K	Physics parameters configuration	Configure “Weight” and “Friction” constants
H	OutTunnel distance notification	Set the positions where Board will send the notification for a Car reaching the OutTunnel (The car is ‘n’ position away from the OutTunnel)
D	Load Racetrack defaults	Reset to defaults the Racetrack configurable parameters (Track length, Box length, etc)
W	Save current configuration	Tell the Board to permamently store current configuration
Q	Query board cfg	Host request the current situation of the Config Parameters Set
R	Race phase	Command used to notify current Race phase
p	Car current position	Telemetry: Car current position in the Racetrack
r	Car Leaving	Car is ‘n’ position away from the OutTunnel. The next OLR in the race (the one receiving the car) will receive a ‘Car Coming’ command.
s	Car Left	Car is in the last position before OutTunnel. The next OLR in the race will receive a ‘Car Arrived’ command.
t	Car Coming	A car is arriving in the OLR (The circuit where it’s now sent a ‘Car Leaving’ command). The Players will see the InTunnel turned on with the color of the arriving car
u	Car Enter	A car enter the OLR (The circuit where the car was until now sent a ‘Car Left’ command). The car will ‘come out’ here from the inTunnel.

Cmd	Description	Notes
w	Car Win the Race	A car just won the current race

Commands Description

In the following sections the column "**Initiate**" contains the id of the board sending the message.

- ***B*** - ***Board***: OLR Board (Arduino Nano)
- ***H*** - ***Host***: Host where the OpenLedRace Network Client is running (Computer).

Same rule applies to the "***From***" column in "***Response***"

Some commands may be originated by both peers (ex: Handshake, racephase, etc)

The string ***[EOC]*** indicates the EOC (End of Command) char = "line feed" = ASCII 10 (0A)

protocol handshaking

#	Protocol Handshaking	
Initiate	**Syntax**	**Description**
B, H	**#***[EOC]*	**Sent to initialize a connection (Board and Host)**
Response	**From**	**Notes**
#*[EOC]*	H, B	The connection opens succesfully when a "**#**" is received 'back' from the peer

---

@ Enter Configuration mode

@	Enter Configuration mode request	
initiate	**Syntax**	**Description**
H	**@***[EOC]*	Sent from Host to put the board in configuration mode. The OLR Board wil stop managing the current status and wait for configuration commands
Response	**From**	**Notes**
@OK*[EOC]*	B	Board sends "OK" string
@NOK*[EOC]*	B	Board indicates something went wrong

~ Leave Configuration mode

~	Leave Configuration mode request	
initiate	**Syntax**	**Description**
H	*~[EOC]*	Sent from Host to tell Board to leave the "Configuration mode" and goes to the "running" status (firmware-dependent)
Response	**From**	**Notes**
~OK*[EOC]*	B	Board sends "OK" string
~NOK*[EOC]*	B	Board indicates something went wrong

: set board id

Not every OLR Board have a unique ID stored in EEPROM.
Currentrly only "Open LED Race Network" game (Network client and firmware) uses the UID.

The Host uses the **Set Board Unique Id** command to permanently store and UID into the board's EEPROM.
The Board writes the ID to EEPROM immediately (*no "B" command needed after ":"*)

:	Set board Unique Id	
initiate	**Syntax**	**Description**
H	**:***id[EOC]*	Set Board Unique Id request Sent from Host to Set Board's Unique Id
Parameter		
Id	*See "UID_format" below*	String representing the Unique Id.
Response	**From**	**Notes**
:OK*[EOC]*	B	Board sends "OK" string
:NOK*[EOC]*	B	Board indicates something went wrong

Unique Board Id (UID) string format:

```
^[\x33-\x7E]{16}$
```

Lenght: 16 chars
Valid Chars: Ascii 7-bit Printable Chars excluding 'space'=ASCII 32 (this means ASCII chars between 33 (0x21) and 126 (0x7E) inclusive

$ get board id

$	Get Board Id	
initiate	**Syntax**	**Description**
H	**$***[EOC]*	Get Board Id request Sent from Host to get Board's Unique Id
Response	**From**	**Notes**
$Id*[EOC]*	B	Send the UID strings

Examples		
Origin	Command	Description
H	**$***[EOC]*	Host send a **get BoardId** request
B	**$**3179c3ec6e28ah64*[EOC]*	The Board send back the UID (**3179c3ec6e28ah64**)
Origin	Command	Description
H	**$***[EOC]*	Host send a **get info** request
B	**$***?????????[EOC]*	The Board send back an invalid UID (**if you are looking at it in a Serial Console, you usually see a bunch of question marks or other chars / non-printable ASCII**). This usually happens when the UID is not set yet, so the Board send back the contents of the area of the EEPROM where the UID is supposed to be stored.

? get software type ID

Used to identify the software installed on the Board

?	Get Software Type Id	
initiate	**Syntax**	**Description**
H	*?[EOC]*	OLR Board software Id request
Response	**From**	**Notes**
?ver*[LF]*	B	Where “ver” is the string representing the Software Id

Software Id String format

```
[0-9a-zA-Z]+.[0-9a-zA-Z]+.[0-9a-zA-Z]+.[0-9a-zA-Z]+
```

Four numbers and/or letters.

Example		
Origin	Command	Description
H	*?[EOC]*	Host send a **get software id** request
B	**?**A4P0*[EOC]*	The message from the Board indicates ID="A4P0"

Guidelines to Assign a software Id to the Arduino Software:

The fist char represents the main category:

- "A" - Open LED Race, developed by Open LED Race Team

The next theree chars identify the software itself:

- 4P0 : 4 Players with Pitlane and Slope (our standard 4 players software)
- 2N0 : Open LED Race Network 2 players

% get software version

Used by Host to check software compatibility with Board's software version

%	Get Software Version	
initiate	**Syntax**	**Description**
H	**%** *[EOC]*	OLR Board software version request
Response	**From**	**Notes**
%ver*[LF]*	B	Where "ver" is the string representing the Software Version

Software Version String format

```
[0-9]+\.[0-9]+\.[0-9]+
```

Three dot-separated **decimal numbers**.

Example		
Origin	Command	Description
H	**%***[EOC]*	Host send a **get software version** request
B	**%**0.8.1*[EOC]*	The message from the Board indicates Version="0.8.1"

Version Number Guidelines

The three numbers represents the "Major.Minor.Patch" version.

Guidelines to Assign a version number to the Arduino Software:

- Major version zero (0.y.z) is for initial development. Anything MAY change at any time.
- Version 1.0.0 defines first 'Stable' version
- Increment:
  - MAJOR version when you make incompatible changes
  - MINOR version when you add functionality in a backwards compatible manner
  - PATCH version when you make backwards compatible bug fixes.

! send log/error message

The software running on the Board use this command to send messages to be written into the Host logfile.
The Host will log the message and decide what to do with the relay race according to the “Type” parameter (do nothing, stop the race, etc.)

!	**Send log/error message**	
initiate	**Syntax**	**Description**
B	**!**Type,*Message[EOC]*	Board sends an error/log message to Host
Parameters		
Type	[0-3]	single char
	1	**Log only** - Board want to write a log a message into the Host’s LogFile, Sent usually in development/debug phase to trace the dialog between Board and Host
	2	**Warning** - Board send back a “warning” message Sent by board on ‘not blocking’ errors like, for example, unknown commands or parameters
	3	**Blocking Error** - The boards have a Severe error condition and cannot proceed. The Host will log the message into the Host Message LogFile and decide what to do (if the Host is running a RelayRace it will Stop the Race)
Message	String	Message Board want to write into the Host’s LogFile
Response	**From**	**Notes**
	H	No answer sent from Host

Example		
Origin	Command	Description
B	!1,invalid Car=[3] in [t] command *[EOC]*	Board send a warning message about a previously received command

C set race Configuration

C	Set Race Configuration Parameters	
initiate	**Syntax**	**Description**
H	**C**start,nlap,repeat,finish*[EOC]*	Host Send Race configuration parameters to Board
Parameter	**Format**	**Description**
start	[0-1]	**OLR Network: Start Line** of the race is in this Board ? (Y/N) (0=No, 1=Yes) ***OLR Standalone: Always 1 for standalone OLR***
nlap	[1-9][0-9]? max 2 chars (range 1-99)	**OLR Network**: Number of consecutive laps in each **section** of the Relay Race (consecutive laps the cars will "run" before race finish **or** car get trough the OutTunnel) ***OLR Standalone: Number of laps of a Race***
repeat	[1-9][0-9]? max 2 chars (range 1-99)	**OLR Network**: Number of times to **repeat the configured section** of '**nlap**' laps ***OLR Standalone: Always 1 for standalone OLR***
finish	[0-1]	**OLR Network: Finish Line** of the race is in this Board ? (Y/N) (0=No, 1=Yes) ***OLR Standalone: Always 1 for standalone OLR***
Response	**From**	
COK*[EOC]*	B	Board sends "OK" string
CNOK*[EOC]*	B	Board indicates that something went wrong (ex: wrong parameter value or format)

Set Race Configuration Examples

Example 1	OLR Network	
Origin	**Command**	**Description**
H	**C**0,5,2,1*[EOC]*	**start=0**: The Race starts in another OLR – The Board will be waiting for messages like “Race Started”, “Car 1 Leaving”, Car 1 Left”, etc...
		laps=5: Each car will need to complete 5 laps before it can cross the Finish Line **or** get to the next OLR (see ‘repeat’ param)
		repeat=2: Each car will need to repoeat 2 times the section of ‘nlap’ laps. This means we’ll expect each car will be sent back here after we previously sent it out to another Racetrack.
		finish=1: The Race ends here.This OLR will manage the Finsh Line Procedure.
B	**C**OK*[EOC]*	Response from the Board. Values for Position,Laps,Repeat,Finish has been set as requested by the host.

Example 2	OLR Network	
Origin	**Command**	**Description**
H	**C**1,2,3,0*[EOC]*	**start=1**: The Race starts here (This Board will be managing the Start Race phase – Semaphore countdown, etc.)
		laps=2: Each car will need to complete 2 laps before can get to the next Racetrack
		Repeat=3: Each car will need to repeat 3 times the section of ‘**nlap**’ laps.
		finish=0: The Race ends in another OLR.
B	**C**OK*[EOC]*	Response from the Board. Values for Position,Laps,Repeat,Finish has been set as requested by the host.

Example 2	OLR Standalone	
Origin	**Command**	**Description**
H	**C**1,3,1,1*[EOC]*	laps=3: Each car will need to complete 3 laps before it can cross the Finish Line
B	**C**OK*[EOC]*	Response from the Board – Race Configured OK

T Track configuration – Total LEDs Number

T	Pit Lane (Box) configuration	
initiate	**Syntax**	**Description**
H	**T**nled*[EOC]*	Host Set Racetrack Length Configuration
Parameter	**Format**	**Description**
nled	Total number of LEDs in the Track	Ex: 300 for a single 5mt - 60 LED/mt LED Strip
Response	**From**	**Notes**
TOK*[LF]*	B	Board sends “OK” string
TNOK*[LF]*	B	Board indicates that something went wrong

Example		
Origin	Command	Description
H	**T**600*[EOC]*	Total Length is 600 (2 x 300 LED Strip connected).
B	**T**OK*[EOC]*	Response from the Board: *the value for “Racetrack Length” has been set*

Note:

Usually, after sending this command, the board have to be hard-reset to re-initate the library managing the LED strip (firmware allocates space for the LED array in the setup() function)

B Track configuration – Pitlane length (Boxes)

B	Pit Lane (Box) configuration	
initiate	**Syntax**	**Description**
H	**B**nled,perm*[EOC]*	Host request the Board to Set the Pitlane length to a specific value
Parameter	**Format**	**Description**
nled	[0-MAXLED]	Total number of LEDs, at the end of the Racetrack, reserved for the Pitlane
perm	[0-1]	**Box always on?** (Y/N) • [0] to activate Pitlane players have to push the Activate Pitlane button on Startup (default behaviour) • [1] the board will always activate pitlane on startup.
Response	**From**	**Notes**
BOK*[LF]*	B	Board sends “OK” string
BNOK*[LF]*	B	Board indicates that something went wrong

Example		
Origin	Command	Description
H	**B**120,0*[EOC]*	Total Length for Pitlane is **120** LEDs, **perm=0** → default behaviour, the board will do not activate pitlane unless user push “Activate Pitlane” button on startup
B	**B**OK*[EOC]*	Response from the Board: *the value for “Pitlane Length” has been set*

A Track configuration – Ramp configuration

A	Ramp (Slope) configuration	
initiate	**Syntax**	**Description**
H	**A**start,center,end, high,perm*[EOC]*	Host set the **Ramp** configuration
Parameter	**Format**	**Description**
start		LED number where the ramp Starts
center		LED Number where ramp is centered.
end		LED number where the ramp ends
height	[0 - 1023]	Ramp elevation
perm	[0-1]	**Ramp always on?** (Y/N) • [0] to activate Ramp/Slope players have to push the Activate Ramp button on Startup (default behaviour) • [1] the board will always activate Ramp/Slope on startup.
Response	**From**	**Notes**
AOK*[EOC]*	B	Board sends “OK” string
ANOK*[EOC]*	B	Board indicates that something went wrong

Example		
Origin	**Command**	**Description**
H	**A**140,150,160,12,0*[EOC]*	Set the ramp centered in led 150 with an elevation of 12, **perm=0** → default behaviour, the board will do not activate slope unless user push “Activate Slope” button on startup
B	**A**OK*[EOC]*	Response from the Board: *values for “Slope” has been set*

K Track configuration – Physics configuration

K	Weight, Friction configuration	
initiate	**Syntax**	**Description**
H	**K**kg,kf*[EOC]*	Weight and Friction constants configuration
Parameter	**Format**	**Description**
kg	`[0-1]+\.\d{3}` *Range: 0.0001-1.999*	Weight constant: Used to calculate speed loss/gain on slopes
kf	`[0-1]+\.\d{3}` *Range: 0.0001-1.999*	Frictions constant: Used to calculate speed loss when player does not push the controller
Response	**From**	**Notes**
KOK*[EOC]*	B	Board sends “OK” string
KNOK*[EOC]*	B	Board indicates that something went wrong

Example		
Origin	Command	Description
H	**K**0.006,0.015*[EOC]*	Set weight constant to 0.006 and Friction to 0.015
B	**K**OK*[EOC]*	Response from the Board: *values for kg, kf has been set*

H out tunnel distance notification (OLR Network only)

N	Set 'car is Reaching the OutTunnel' notification distance	
initiate	**Syntax**	**Description**
H	**H**num*[EOC]*	Set the positions where the Board will send the notification message for a Car reaching the OutTunnel (the car is **num** position away from the OutTunnel). It will be used by the "next" board in the relay race to 'light up' its input tunnel visual effect
Parameter	**Format**	**Description**
num	[0-9]+	One or more char representing a decimal number
Response	**From**	**Notes**
NOK*[LF]*	B	Board sends "OK" string
NNOK*[LF]*	B	Board indicates that something went wrong

Example		
Origin	Command	Description
H	**H**8*[EOC]*	Host request the board to set the Reaching tunnel nofication distance to '8'
B	**H**OK*[EOC]*	Response from the Board: *the value for "Notification distance" has been set*

D reset basic params to Default values

Reset to the program-defined default values any configurable parameter.

- Ramp (Slope)
- Pit Lane (Box)
- LED number (strip length)
- Race laps number
- Friction/Gravity Constant

D	Reset params to default values as defined in the source file	
initiate	**Syntax**	**Description**
H	**D***[EOC]*	Host request a Reset to Default configuration
Response	**From**	**Notes**
DOK*[EOC]*	B	Board sends “OK” string (ACK)
DNOK*[EOC]*	B	Board indicates that something went wrong

W Writes current parameter set to EEPROM

After changing values in the parameter set the Host calls this commad to store permanently the new configuration.

D	Reset params to default values as defined in the source file	
initiate	**Syntax**	**Description**
H	**W***[EOC]*	Host request to save current parameter set to EEPROM
Response	**From**	**Notes**
WOK*[EOC]*	B	Board sends “OK” string (ACK)
WNOK*[EOC]*	B	Board indicates that something went wrong

Q Query current parameters set

Q	Current paremeters values stored on the Board	
initiate	**Syntax**	**Description**
H	**Q***[EOC]*	Host request the current configurable parameters values
Response	**From**	**Notes**
TRACK:a,b,c,d,e,f,g,h*[EOC]* ***RAMP***:a,b,c,d*[EOC]* ***RACE***:a,b,c,d*[EOC]*	B	Board issue 3 messages (3 x *[EOC]*) containing the whole set of configurable values

Returned parameters description		
Track params		
a	**nled_total**	Total number of LEDs in the Racetrack ***(configurable with "T" command)***
b	nled_main	debug parameter: *when Pitlane is active: number of LEDs currently in the Main Path*
c	nled_aux	debug parameter: *when Pitlane is active: number of LEDs currently in the Pitlane Path*
d	nled_init_aux	debug parameter: *position of the Pitlane entrance*
e	**box_len**	Total number of LEDs, at the end of the Racetrack, reserved for the Pitlane ***(configurable with "B" command)***
f	**box_alwaysOn**	the board always activate pitlane on startup [0|1] ***(configurable with "B" command)***
g	weight_const	Weight constant: Used to calculate speed loss/gain on slopes
h	friction_const	Frictions constant: Used to calculate speed loss when player does not push the controller
Ramp params		
	start	LED number where the ramp Starts
	center	LED Number where ramp is centered.
	end	LED number where the ramp ends
	height	Ramp elevation
	alwaysOn	the board always activate Ramp/Slope on startup [0|1]
Race params		
	start	OLRNetwork Only - **Always 1 for standalone OLR**
	nlap	Number of laps of a Race ***(Configurable with 2nd parameter of "C" Command)***
	repeat	OLRNetwork Only - **Always 1 for standalone OLR**
	finish	OLRNetwork Only - **Always 1 for standalone OLR**

Example		
Origin	**Command**	**Description**
H	**Q***[EOC]*	Host send a get current paremeters Set request
B	TRACK:1200,1200,0,-1,60,0,0.006,0.015*[EOC]* RAMP:180,190,200,15[*[EOC]* RACE:1,2,1,1*[EOC]*	Messages from the Board with the current cfg values

R Race phase

R	Current Race Phase	
initiate	**Syntax**	**Description**
B,H	**R**num*[EOC]*	Current Race phase Board and Host send this command to notyfy changes in Race Status
Parameter	**Format**	**Description**
num	[0-9]	single numeric char
Initiate	**Value**	**Description**
B	0	**Idle**
H	1	Host request the Board to **enter in configuration mode** ***--DEPRECATED-- Use the "@" command***
B	2	**Configuration Complete.** Board send this after receiving the last needed configuration parameter
H	3	**Race Ready** Sent from Host when every participant Board reach the [Configuration Complete] status When the OLR where Race starts receive this, it starts the Countdown
B,H	4	**Countdown started** (*Red light on*) The OLR where race starts **send** this message - any other OLR in the same Race will **receive** it
B,H	5	**Racing** - Race Started (*Countdown finished*) The OLR where race starts **send** this message and any other participant OLR will **receive** it
B,H	6	Race **Paused** (Safety car) *not implemented*
B,H	7	**Resume** Race (Safety car leave) *not implemented*
B,H	8	Race **Complete** The OLR managing the **Finish line** send this message when the winner cross it. Any other participant OLR will **receive** the message
Response	**From**	**Notes**
ROK*[LF]*	B	Board sends "OK" string
RNOK*[LF]*	B	Board indicates that something went wrong

Example		
Origin	**Command**	**Description**
B	**R**4*[EOC]*	Board send this when the Countdown phase starts.

p Telemetry: current car Position in race

p	Position for each car in the race	
initiate	**Syntax**	**Description**
B	**p**CnumStrackNlap,Rpos*[EOC]*	Position for each car in the race Sent during race for each car currently in this Board.
Parameter	**Format**	**Description**
Cnum	[1-9]	One char representing Car Number
Strack	[A-Z]	One char representing the SubTrack where the car is
	M	Main Track
	B	Box Track (Pit Lane)
	U	Not a Track
Nlap	[1-99]	Number of the Current Lap.
Rpos	[00-99]	Relative position in a track (*percentage*)
Response	**From**	**Notes**
	H	No response from host

Example		
Origin	Command	Description
B	**p**1B1.95p2M5.45*[EOC]*	• Two cars are currentry "running" in the Board: ◦ Car "1" is in Track "B" in Lap number "1", relative Lap Position 95% ◦ Car "2" is in Track "M" in Lap number "5", relative Lap Position 45%

r Car leaving (OLR Network only)

r	**Car is about to leave the current Racetrack**	
Initiate	**Syntax**	**Description**
B	**r**Num*[EOC]*	Car '**Num**' is 'n' position avay from OutTunnel Sent during race so the next OLR in the race (the one receiving the car) will turn on the InTunnel light effects.The Playere will sees the InTunnel turned on with the color of the arriving car
Parameter	**Format**	**Description**
Num	[1-9]	One char representing Car Number
Response	**From**	**Notes**
	H	No response from host

Example		
Origin	Command	Description
B	**r**1*[EOC]*	Car "1" reached the distance from tunnel specified with the '**N**' config param

s Car left the circuit - last position reached (OLR Network only)

s	**Car left**	
initiate	**Syntax**	**Description**
B	**s**Data*[EOC]*	Car is at the last valid position of the path '**Data**' byte contains car id and speed
Parameter	**Format**	**Description**
Data	Byte (char)	One byte representing Car Number and speed
	Speed → Bits:[0:4]	5 bits representing speed
	Car Num → Bits:[5:7]	3 bits representing the car number
Response	**From**	**Notes**
	H	No response from host

Example		
Origin	Command	Description
B	**s00100100***[EOC]*	**Please note**: 00100100 **is not** the "00100100" string !!! It represents the binary value of **one byte:** Bits:[0:4] = 00100 → Bin representation Dec "4" Bits:[5:7] = 001 → Bin representation of Dec "1" Car "1" Laved the Racetrack with speed '4' If you look at this command in a Serial console, you will see "s$" (ASCII table: dec:36→Binary:00100100 →Char:$)

t Car coming (OLR Network only)

r	Car is about to enter into the Racetrack	
Initiate	**Syntax**	**Description**
H	**t**Num*[EOC]*	Car 'Num' is arriving. Received during the race when a car is arriving from another OLR. This Board will turn the InTunnel On
Parameter	**Format**	**Description**
Num	[1-9]	One char representing Car Number
Response	**From**	**Notes**
	B	No response from Board

Example		
Origin	Command	Description
H	**t**1*[EOC]*	Car "1" is 'n' step away from here....

u Car enter the circuit (OLR Network only)

u	Car arrived to this Racetrack	
initiate	**Syntax**	**Description**
H	**u**Data*[EOC]*	**Car 'Num' with 'Speed'** enters the circuit '**Data**' byte contains car id and speed Received during the race when a car arrives from another OLR.
Parameter	**Format**	**Description**
Data	Byte (char)	One byte representing Car Number and speed
	Speed → Bits:[0:4]	5 bits representing speed
	Car Num → Bits:[5:7]	3 bits representing the car number
Response	**From**	**Notes**
	B	No response from Board

Example		
Origin	Command	Description
B	**u00100100***[EOC]*	Car "1" Laved the Racetrack with speed '4' *– see description in command "**s**"–**Car Left** example*

w Car win the race

w	Car win the current race	
Initiate	**Syntax**	**Description**
B	**w**Num*[EOC]*	Car 'Num' just win the race Sent by the Board when a car cross the Finish Line
Parameter	**Format**	**Description**
Num	[1-9]	One char representing Car Number
Response	**From**	**Notes**
	H	No response from Host

Example		
Origin	Command	Description
B	**w**1*[EOC]*	Car "1" won the race

Document revisions:

- 2020_12_07: Luca – **Ver 1.0**
    - Changed ID for command “Boxlength (from “T” to “B”)
    - Added command:
        - @ - Enter Configuration mode (***R1*** *is now deprecated*)
        - ~ - Leave Configuration mode
        - K – Physics constants configuration (kf, kg)
        - T - Track length configuration (number of LED in racetrack)
        - ? - Get Software Type
        - W – Write / store permanently current parameter (ex: EEPROM)
- 2020_04_24: Luca
    - Modified “Get Software Version→Version StringFormat”
    - Doc cleanup
- 2019_09_15: Angel
    - Added T,A and D commands
- 2019_08_31: Angel
    - Modified *u* and *s* command to include car speed
- 2019_08_04: Angel
    - Modified [Get UID] command
- 2019_08_03: Angel
    - Changed field separator character to “,”
    - Review command [Race Phase]
- 2019_07_30: Luca
    - added [get Software Version] command
    - modified [get board info] -> [get Board UID]
- 2019_07_29: Angel
    - added commands [Configuration Race]
    - deleted commands [Race Starts here] [Laps number] [Repeat Section number] [Race Finish Line]
- 2019_07_27: Luca
    - added commands [Car Arriving] [Car Arrived] with ‘Special’ command format
    - added commands [Send Log/Error] [Car Win] [Query Board cfg]
    - added parameter values to [Race Phase] command
    - modified parameter **[P]** from Position[0-9]+ to **StartsHere[0-1]**
    - deleted [Car current lap] parameter (not used)
- 2019_07_15: Luca
    - added commands [Reset][Get Board Info] [Set Unique Id]
- 2019_05_10: Angel
    - added commands [Car Current Position]
- 2019 march
    - Doc created