

OLR Serial Protocol (Board ← → Host)

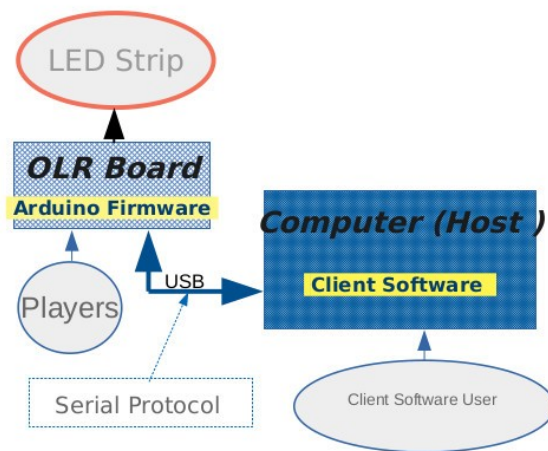
Table of Contents

Terminology.....	2
Implementation notes.....	2
Message format.....	2
Commands List.....	3
Commands Description.....	4
# protocol handshaking.....	4
@ Enter Configuration mode.....	4
* Leave Configuration mode.....	5
: set board id.....	5
Unique Board Id (UID) string format:.....	5
\$ get board id.....	6
? get software type ID.....	7
Software Id String format.....	7
% get software version.....	8
Software Version String format.....	8
Version Number Guidelines.....	8
! send log/error message.....	9
C set race Configuration.....	10
T Track configuration - Total LEDs Number.....	12
Note:.....	12
P Track configuration - Players number.....	13
Note:.....	13
A Track configuration - Ramp configuration.....	14
B Track configuration - Pitlane length (Boxes).....	15
E Track configuration - Battery configuration.....	16
G Track configuration - AutoStart [Y N].....	17
K Track configuration - Physics configuration.....	17
M Track configuration - Demo Mode [Y N].....	18
H out tunnel distance notification (OLR Network only).....	19
D reset basic params to Default values.....	20
W Writes current parameter set to EEPROM.....	20
Q Query current parameters set.....	21
R Race phase.....	23
n Set "Network / Relay race" mode (Network mode only).....	24
p Telemetry: current car parameters in race (position, battery, etc).....	25
r Car leaving (Network mode only).....	26
s Car left the circuit - last position reached (Network mode only).....	26
t Car coming (Network mode only).....	27
u Car enter the circuit (Network mode only).....	27
w Car win the race.....	28
Document revisions:.....	29

Terminology

In the present doc the terms **Board**, **Host**, **Standalone mode** and **Network mode** indicate:

- **Board**: OLR Board - The microcontroller running the firmware (*managing Buttons, LED strip, etc*)
- **Host**: A Software running on a computer connected to the Board via Serial interface (USB).
The "*client software*" can be *OLR Update/Configuration Utility, OpenLEDRace Network Client*, or simply a Serial Console used to manually send commands to the firmware (*ex: Arduino IDE serial console*).
- **Standalone mode**: The Board plays OLR games that starts and ends in the '*Circuit*' (LED Strips) directly connected to the Board.
- **Network mode (Relay Race)**: The Board is part of a Network and participates to Relay Races where Cars *moves* from one Board to another.



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
%	Get Software Version	Get Board software version
!	Send log/error msg	Send a log/error message to peer
C	Race Configuration	Set main race configuration parameters
T	Track length configuration	Command used to configure the Total Number of LEDs in the track.
P	Players number configuration	Command used to configure number of players (2,3,4)
A	Ramp (Slope) configuration	Command used to configure the Ramp (Hill, Slope)
B	Box length configuration	Command used to configure the PitLane (Box)
E	Battery configuration	Command used to configure the Battery
G	Auto start configuration	Command used to configure how to start a new race
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)
M	Demo Mode	<i>Simulated race</i> where cars run without any real player (Useful to get people attention in Fairs, yest the code and to, etc)
D	Load Racetrack defaults	Reset to defaults the Racetrack configurable parameters (Track length, Box length, etc)
W	Save current configuration	Tell the Board to permanently 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
n	Network Race	Command used to set the firmware in "Network mode" (Relay race)
p	Car current position	Telemetry: Car current position and status parameters (battery, etc)
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.

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

Commands Description

In the following sections the columns "**Initiate**", "**From**" and "**Response**" contain the id of the board sending the message.

- **B → Board**
- **H → Host**

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

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 (currently only Network client of the "Open LED Race Network" game set/read the Board unique ID).

The Board writes the ID to EEPROM immediately (*no "Write" command needed after ":"*)

:	Set board Unique Id	
initiate	Syntax	Description
H	:id[EOC]	Sent from Host to set the Board's Unique Id
Parameter		
Id	<i>See "UID_format" below</i>	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}$`

Length: 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	<code>\$[EOC]</code>	Sent from Host to get Board's Unique Id
Response	From	Notes
<code>\$Id[EOC]</code>	B	Send the UID strings

Examples		
Origin	Command	Description
H	<code>\$[EOC]</code>	Host send a get BoardId request
B	<code>\$3179c3ec6e28ah64[EOC]</code>	The Board send back the UID (3179c3ec6e28ah64)
Origin	Command	Description
H	<code>\$[EOC]</code>	Host send a get info request
B	<code>\$????????[EOC]</code>	<p>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).</p> <p>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.</p>

? get software type ID

Used by Host 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 first char represents the main category:

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

The next three 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 software will log the message and decide what to do according to the "Type" parameter (*may be an error*)

!	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	Cstart,nlap,repeat,finish[EOC]	Host Send Race configuration parameters to Board
Parameter	Format	Description
start	[0-1]	Standalone mode: always start=1 Network mode: Start Line of the race is in this Board ? (Y/N) (0=No, 1=Yes)
nlap	[1-9][0-9]? max 2 chars (range 1-99)	Standalone mode: Number of laps of the Race Network mode: 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)
repeat	[1-9][0-9]? max 2 chars (range 1-99)	Standalone mode: always repeat=1 Network mode: Number of times to repeat the configured section of 'nlap' laps
finish	[0-1]	Standalone mode: always finish=1 Network mode: Finish Line of the race is in this Board ? (Y/N) (0=No, 1=Yes)
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	Network mode	
Origin	Command	Description
H	C0,5,2,1[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 repeat 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 Finish Line Procedure.
B	COK[EOC]	Response from the Board. Values for Position,Laps,Repeat,Finish has been set as requested by the host.

Example 2	Network mode	
Origin	Command	Description
H	C1,2,3,0[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	COK[EOC]	Response from the Board. Values for Position,Laps,Repeat,Finish has been set as requested by the host.

Example 2	Standalone mode	
Origin	Command	Description
H	C1,3,1,1[EOC]	laps=3: Each car will need to complete 3 laps before it can cross the Finish Line
B	COK[EOC]	Response from the Board – Race Configured OK

T Track configuration – Total LEDs Number

T	Racetrack Length configuration	
initiate	Syntax	Description
H	Tnled[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	T600[EOC]	Total Length is 600 (2 x 300 LED Strip connected).
B	TOK[EOC]	Response from the Board: <i>the value for "Racetrack Length" has been set</i>

Note:

After receiving 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)

P Track configuration – Players number

P	Players number configuration	
initiate	Syntax	Description
H	Pn[EOC]	Host Set Players number for the races
Parameter	Format	Description
n	[2,3,4]	Number of Players/Controllers used in the Racetrack
Response	From	Notes
TOK[LF]	B	Board sends "OK" string
TNOK[LF]	B	Board indicates that something went wrong

Example		
Origin	Command	Description
H	P3[EOC]	3 Players (Red, Green, Blue)
B	POK[EOC]	Response from the Board: <i>the value for "Players number" has been set</i>

Note:

After receiving this command, the board have to be hard-reset to re-initate the firmware

A Track configuration – Ramp configuration

A	Ramp (Slope) configuration	
initiate	Syntax	Description
H	Astart,center,end,height,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 - 254]	Ramp elevation
perm	[0-1]	Ramp always on? (Y/N) <ul style="list-style-type: none"> [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	A140,150,160,12,0[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	AOK[EOC]	Response from the Board: <i>values for "Slope" has been set</i>

B Track configuration – Pitlane length (Boxes)

B	Pit Lane (Box) configuration	
initiate	Syntax	Description
H	Bnled,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) <ul style="list-style-type: none"> [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	B120,0[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	BOK[EOC]	Response from the Board: <i>the value for "Pitlane Length" has been set</i>

E Track configuration – Battery configuration

E	Battery configuration	
initiate	Syntax	Description
H	Edelta,min,boost,active[EOC]	Host set the Battery configuration
Parameter	Format	Description
delta	[1-254]	Value of a single step for Battery consumption: On each controller activation the battery "discharge" delta/100 (firmware divides by 100: Battery usage for a single "click":[0.01% – 2.54 %])
min	[1-100]	Battery does not discharge below this "min" percentage
boost	[1-254]	Speed boost when a car gets fully recharged
active	[0-1]	Battery mode on? <ul style="list-style-type: none"> [0] off [1] the board will activate Battery mode on startup.
Response	From	Notes
EOK[EOC]	B	Board sends "OK" string
ENOK[EOC]	B	Board indicates that something went wrong

Example		
Origin	Command	Description
H	E6,40,20,1[EOC]	Set the Battery delta to 6 (0.06), battery charge will not drop below 40%, the speed boost when a car stops for a fully recharge is 20. active=1 → board will activate Battery Mode on startup
B	EOK[EOC]	Response from the Board: <i>values for "Battery" has been set</i>

G Track configuration – AutoStart [Y|N]

G	Race start mode configuration	
initiate	Syntax	Description
H	Gautostart[EOC]	Host Set AutoStart mode [y n] for races
Parameter	Format	Description
autostart	[0 1]	<ul style="list-style-type: none"> 0: To start countdown, players will need to press every active controller (confirming everybody is ready) 1: Race countdown starts automatically soon after the previous race ends
Response	From	Notes
GOK[LF]	B	Board sends "OK" string
GNOK[LF]	B	Board indicates that something went wrong

Example		
Origin	Command	Description
H	G0[EOC]	AutoStart=OFF: To start countdown, players will need to press every active controller (confirming everybody is ready)
B	GOK[EOC]	Response from the Board: <i>the value for "AutoStart" has been set</i>

K Track configuration – Physics configuration

K	Weight, Friction configuration	
initiate	Syntax	Description
H	Kkg,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	K0.006,0.015[EOC]	Set weight constant to 0.006 and Friction to 0.015
B	KOK[EOC]	Response from the Board: <i>values for kg, kf has been set</i>

M Track configuration – Demo Mode [Y|N]

G	Demo mode Standard mode	
initiate	Syntax	Description
H,B	Mdemo[EOC]	H→B: Host Set demo_mode [on off] B→H: Board communicates to Host changes of state for demo_mode option (*)
Parameter	Format	Description
demo	[0 1]	<ul style="list-style-type: none"> 0: Standard mode: Race managed by Players using Controllers (Buttons) (**) 1: Demo mode: <i>Simulated race</i> where cars run without any real player (<i>Useful to test the code and to get people attention in Fairs, etc</i>)
Response	From	Notes
MOK[LF]	B	Board sends "OK" string
MNOK[LF]	B	Board indicates that something went wrong
M[0 1][LF]	B	Board communicates DemoMode parameter status [0 1]

Example	Host → Board	
Origin	Command	Description
H	M0[EOC]	Demo mode= off : Game need real players.
B	MOK[EOC]	Command OK
B	MOK[EOC]	Response from the Board: <i>"Demo mode" has been deactivated</i>

Example	Board → Host	
Origin	Command	Description
B	M1[EOC]	Demo mode= on : Board plays a <i>demo game</i> . Board send this command to Host to comunicate the DEMO mode has been activated. <i>It happens when there is no user interaction for longer than INATIVITY_TIMEOUT→ board gets automatically in DEMO mode and comunicates it to Host)</i>

(*) If the Board has M1 saved in the EEPROM configuration, it will start in demo_mode. On user activity (somebody uses the controllers) the boards jumps automatically to demo=off mode and sends M0 to the host. After *inactivity timeout (nobody using the controllers)* it jumps back to demo=on mode and sends M1 to the Host.

(**) When Host sends a M0 command (demo=off), the board does not gets back to Demo mode after the inactivity timeout !

H out tunnel distance notification (OLR Network only)

N	Set 'car is Reaching the OutTunnel' notification distance	
initiate	Syntax	Description
H	Hnum[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]+ <i>Range: 0-254</i>	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	H8[EOC]	Host request the board to set the Reaching tunnel notification distance to '8'
B	HOK[EOC]	Response from the Board: <i>the value for "Notification distance" has been set</i>

D reset basic params to Default values

Reset configurable parameter values to the program-defined defaults.

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 parameters values set (<i>runtimes values / they may be different from the values stored in EEPROM</i>)	
initiate	Syntax	Description
H	Q[EOC]	Host request the current configurable parameters values
Response	From	Notes
QTK :a,b,c,d,e,f,g,h,i[EOC] QRP :a,b,c,d,e[EOC] QBT :a,b,c,d[EOC] QRC :a,b,c,d,e,f,g,h[EOC]	B	Board issue 4 messages (4 x [EOC]) containing the whole set of configurable values
Returned parameters description		
QTK	Track params	
a	nled_total	Total number of LEDs in the Racetrack (configurable with "T" command)
b	nled_main	debug parameter: <i>when Pitlane is active: number of LEDs currently in the Main Path</i>
c	nled_aux	debug parameter: <i>when Pitlane is active: number of LEDs currently in the Pitlane Path</i>
d	nled_init_aux	debug parameter: <i>position of the Pitlane entrance</i>
e	box_len	Total number of LEDs, at the end of the Racetrack, reserved for the Pitlane
f	box_alwaysOn	the board always activate pitlane on startup [0 1]
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
i	auto_start	Countdown for a New race starts automatically after a race ends [0 1]
QRP	Ramp params	
a	start	LED number where the ramp Starts
b	center	LED Number where ramp is centered.
c	end	LED number where the ramp ends
d	height	Ramp elevation
e	alwaysOn	the board always activate Ramp/Slope on startup [0 1]
QBT	Batt params	
a	delta	Step (increment/decrement) for battery level
b	min	Minimun value (%) for battery (it will not discharge belowe this value)
c	boost_scaler	X-tra speed boost on fully recharge
d	active	the board activate Battery mode on startup [0 1]
QRC	Race params	
a	start	OLRNetwork Only - Always 1 in Standalone mode
b	nlap	Number of laps of a Race
c	repeat	OLRNetwork Only - Always 1 in Standalone mode
d	finish	OLRNetwork Only - Always 1 in Standalone mode
e	player3_on	Player 3 active on startup
f	player4_on	Player 4 active on startup
g	demo_mode	Software running in DEMO mode [0 1]
h	network_race	Is a Relay Race [0 1] (<i>not stored in EEPROM</i>)

Example		
Origin	Command	Description
H	Q [EOC]	Host send a get paremeters Set request
B	<i>QTK:300,300,0,-1,60,0,0.006,0.015,1</i> [EOC] <i>QRP:80,90,100,6,0</i> [EOC] <i>QBT:3,60,10,0</i> [EOC] <i>QRC:1,5,1,1,0,0,0,0</i> [EOC]	Messages from the Board with the current cfg values

R Race phase

In **Standalone mode** 'R' commands are always originated by the Board (*the board manage internally Config, Ready, Countdown, etc status*). The 'R' messages sent by the Board may be used to implement external display, debug via serial console, connect a robot managing the start flag....

In **Network mode (Relay Race)** the Host send 'R' commands to the Board to change the Board's internal status. *For example, when in a Relay Race every participant has been correctly configured, Host send 'R3' (Race ready) to the Board.*

R	Current Race Phase	
initiate	Syntax	Description
B,H	Rnum[EOC]	Current Race phase Board and Host send this command to notify changes in Race Status
Parameter	Format	Description
num	[0-9]	single numeric char
Initiate	Value	Status description
B	0	Idle (<i>Network mode only</i>)
B	1	Configuration
B	2	Configuration complete Board send this after receiving the last needed configuration parameter (<i>Network mode only</i>)
B,H	3	Ready to race Sent from Board when reach the Ready status (Standalone mode, Network mode) Sent from Host when every participant Board reach the [Configuration Complete] status (<i>Network mode only</i>)
B,H	4	Countdown started Sent from Board when Countdown starts (Standalone mode, Network mode) Host transmit the command to others OLR in the same Relay Race so any Board receives it (<i>Network mode only</i>)
B,H	5	Racing - Race Started (<i>Countdown finished</i>) Same as <i>Countdown started phase</i> above
B,H	6	Race Paused (Safety car) - <i>not implemented</i>
B,H	7	Resume Race (Safety car leave) - <i>not implemented</i>
B,H	8	Race Complete Same as (<i>Countdown started phase</i>) above
Response	From	Notes
ROK[LF]	B	Board sends "OK" string
RNOK[LF]	B	Board indicates that something went wrong

Example		
Origin	Command	Description
B	R4[EOC]	Board send this when the Countdown phase starts.

n Set "Network / Relay race" mode (Network mode only)

Tells the firmware to set the *Network Race mode* (Relay race)

n	Request the firmware to set the Network Race mode (Relay Race)	
initiate	Syntax	Description
H	n[EOC]	Host request activation of the Network Race mode (Relay Race)
Response	From	Notes
nOK[EOC]	B	Board sends "OK" string (ACK)
nNOK[EOC]	B	Board indicates that something went wrong

This setting will **not** survive to a Board reset/reboot !!!

When a "W" (*write params to EEPROM*) command is received, the value of this parameter will not be saved !!!

The Network Client program controlling the Relay Race sends the "n" (*enter "relay race" mode*) command in the handshake phase.

When the Board receives this command it goes from "Standalone" mode to "Network/Relay Race" mode.

On reset/reboot the board always starts in "Standalone" mode

p Telemetry: current car parameters in race (position, battery, etc)

p	Status for each car in the race	
initiate	Syntax	Description
B	pCnumStrackNlap,Rpos,Bat[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 (<i>percentage</i>)
Bat	[00-100]	Battery charge status (<i>percentage</i>)
Response	From	Notes
	H	No response from host

Example		
Origin	Command	Description
B	p1B8,70,85[EOC] p2M9,62,59[EOC]	<ul style="list-style-type: none">Two cars are currently "running" in the Board:<ul style="list-style-type: none">Car "1" is in SubTrack "B" (boxes) in Lap number "8", relative Lap Position 70%, Battery at 85%Car "2" is in Track "M" (Main) in Lap number "9", relative Lap Position 62%, Battery at 59%

r Car leaving (Network mode only)

r	Car is about to leave the current Racetrack	
Initiate	Syntax	Description
B	rNum[EOC]	Car 'Num' is 'n' position away 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 Player will see 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	r1[EOC]	Car "1" reached the distance from tunnel specified with the 'N' config param

s Car left the circuit – last position reached (Network mode only)

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

t Car coming (Network mode only)

r	Car is about to enter into the Racetrack	
---	--	--

r	Car is about to enter into the Racetrack	
H	tNum[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	t1[EOC]	Car "1" is 'n' step away from here....

u Car enter the circuit (Network mode only)

u	Car arrived to this Racetrack	
initiate	Syntax	Description
H	uData[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	wNum[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	w1[EOC]	Car "1" won the race

Document revisions:

- 2023_07_20: Luca – **Ver 1.2**
 - Added command:
 - M – Demo mode
 - n – Network mode
 - Modified command:
 - Q (added output for [M],[n] parameters status)
- 2022_06_10: Luca – **Ver 1.1**
 - Modified Leave configuration mode (~ → *)
- 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