OLRN 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***: The Host running the OpenLedRace Network software (Computer)

The Board is currently connected to the Host via Serial interface (USB)

Implementation notes

- Local Communication between Board and Host use plain ASCII (not binary).
- Messages are kept very short (lightweight protocol for “speed” and low resources.)

Message formats

- Messages are composed by 2 parts: **Command**, **Parameters**.
- Messages sending back command confirmation uses “***command*OK**“ and “***command*NOK**“
    - Example: **CNOK** is the ‘error’ answer sent for a C command
- Message ends with the “EOC” (End Of Command) char:
    - [LF] - Line Feed = ASCII 10/0x0A = new line = ‘\n’

Example: A Board (OLR Device / Arduino) sending the “Configuration Complete” message
R2[LF]

Commands List

Cmd	Description	Notes
#	Protocol Handshaking	Host-Board handshake on startup
@	Reset	Host send a Reset to the Board
:	Set Unique ID	Set Board Unique ID
$	Get UID	Get Board Unique Id
%	Get Version	Get Board software version
!	Send log/error msg	Send a log/error message to peer
C	Race **C**onfiguration	Set race configuration
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)
Q	Query board cfg	Host request the current situation of the Config Parameters Set
R	Race phase	Command used to notify current Race phase
T	subTrack configuration	Command used to configure the PitLane (Box)
A	r**A**mp configuration	Command used to configure a Ramp (Hill, Slope)
D	Load SubTrack and Ramp default	Command used to Reset to Default “T” (Sub-tRack) and “A” (rAmp) parameters
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.
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 command)

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

protocol handshaking

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

---

@ reset

@	Reset (To be implemented)	
initiate	**Syntax**	**Description**
H	**@***[LF]*	**OLR Board Reset request** Sent from Host to Reset the OLR Board to the initial state (before handshake)
Response	**From**	**Notes**
		No response expected from Board

: set board id

The first time a Board is connected to a OLRNetwork the Unique Id may be empty (not every Board comes with the ID preloaded in EEPROM).

The Host get an Unique Board Id *(usually from the Network)* and send it to the Board with the **Set Board Unique Id** command.
The software running on the Board write the UID to EEPROM.
From now on, this is the ID the board will send back on receining a **Get Unique Id** command.

:	**Set board Unique Id**	
initiate	**Syntax**	**Description**
H	**:***id[LF]*	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*[LF]*	B	Board sends "OK" string
NOK[LF]	B	Board indicates something went wrong

Unique Board Id (UID) string format:

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

Lenght: 16 chars
Valid Char: 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	**$***[LF]*	Get Board Id request Sent from Host to get Board's Unique Id
Response	**From**	**Notes**
$Id[LF]	B	Send the UID strings

Examples		
Origin	**Command**	**Description**
H	**$***[LF]*	Host send a **get BoardId** request
B	**$**3179c3ec6e28ah64*[LF]*	The Board send back the UID (**3179c3ec6e28ah64**)
Origin	**Command**	**Description**
H	**$***[LF]*	Host send a **get info** request
B	**$***????????[LF]*	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 version

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

%	Get Software Version	
initiate	**Syntax**	**Description**
H	**%** *[LF]*	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	**%***[LF]*	Host send a **get software version** request
B	**%**0.8.1*[LF]*	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[LF]*	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 *[LF]*	Board send a warning message about a previously received command

C set race Configuration

The Host send this command when a user confirm the 'Race Configuration'.

C	**Set Race Configuration Parameters**	
initiate	**Syntax**	**Description**
H	**C**start,nlap,repeat,finish*[LF]*	Host Send Race configuration parameters to Board
Parameter	**Format**	**Description**
start	[0-1]	**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)	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)	Number of times to **repeat the configured section** of '**nlap**' laps
finish	[0-1]	**Finish Line** of the race is in this Board (Y/N) (0=No, 1=Yes)
Response	**From**	
COK[LF]	B	Board sends "OK" string
CNOK[LF]	B	Board indicates that something went wrong (ex: wrong parameter value or format)

Set Race Configuration Examples

Example 1		
Origin	Command	Description
H	**C**0,5,2,1*[LF]*	**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*[LF]*	Response from the Board. Values for Position,Laps,Repeat,Finish has been set as requested by the host.

Example 2		
Origin	Command	Description
H	**C**1,2,3,0*[LF]*	**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*[LF]*	Response from the Board. Values for Position,Laps,Repeat,Finish has been set as requested by the host.

H out tunnel distance notification

N	Set 'car is Reaching the OutTunnel' notification distance	
initiate	**Syntax**	**Description**
H	**H**num*[LF]*	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
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*[LF]*	Host request the board to set the Reaching tunnel nofication distance to '8'
B	**H**OK*[LF]*	Response from the Board: *the value for "Notification distance" has been set*

R Race phase

R	Current Race Phase	
initiate	**Syntax**	**Description**
B,H	**R**num*[LF]*	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**
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**2*[LF]*	Board previously received the whole set of cfg params (Position, Laps, Repeat, FinishLine) and automatically send a **configuration complete** command

T subTrack configuration

This configuration is stored in non-volatile memory.

T	Pit Lane (Box) configuration	
initiate	**Syntax**	**Description**
H	**T**box.tbd*[LF]*	Host request the Board to Set the Pitlane length to a specific value
Parameter	**Format**	**Description**
box	[0-MAXLED]	Number of the led where the PitLane starts. **Set 0 to remove box**
tbd	[TBD]	Not used yet, set to 0.
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**260,0*[LF]*	Set the box line in led number 260.

A rAmp configuration

This configuration is stored in non-volatile memory.

A	Ramp (Slope) configuration	
initiate	**Syntax**	**Description**
H	**A**center.high*[LF]*	Host set the **Ramp** centered at ‘**center**’ with a altidude of ‘**high**’
Parameter	**Format**	**Description**
center	[0-MAXLED]	Number of the led where ramp is centered. Set 0 to remove Ramp.
height	[0 - 1023]	Ramp elevation
Response	**From**	**Notes**
AOK*[LF]*	B	Board sends “OK” string
ANOK*[LF]*	B	Board indicates that something went wrong

Example		
Origin	Command	Description
H	**A**150,12*[LF]*	Set the ramp centered in led 150 with an elevation of 12

D reset basic params to Default values

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

- Ramp (Slope)
- Pit Lane (Box)
- LED number (MAXLED, strip length)
- Friction Constnt
- Gravity Constant

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

p telemetry: current car Position in race

p	**Position for each car in the race**	
initiate	**Syntax**	**Description**
B	**p**CnumStrackNlap,Rpos*[LF]*	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**1M1,46*[LF]*	Car “1” is in SubTrack “M” in Lap number “1” Relative Lap Position 46%

r car leaving

r	Car is about to leave the current Racetrack	
Initiate	**Syntax**	**Description**
B	**r**Num*[LF]*	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*[LF]*	Car "1" reached the distance from tunnel specified with the '**N**' config param

s car left the circuit (last position reached)

s	Car left	
initiate	**Syntax**	**Description**
B	**s**Data*[LF]*	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***[LF]*	**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

r	Car is about to enter into the Racetrack	
Initiate	**Syntax**	**Description**
H	**t**Num*[LF]*	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*[LF]*	Car "1" is 'n' step away from here….

u - car enter the circuit

u	Car arrived to this Racetrack	
initiate	**Syntax**	**Description**
H	**u**Data*[LF]*	**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***[LF]*	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*[LF]*	Car 'Num' just win the race Sent by the Board managing the Finish Line when a car cross it
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*[LF]*	Car "1" won the race

Document revisions:

- 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 charactero 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