



# Pc Lap Counter protocol for Arduino or others.

---

## Table of Contents

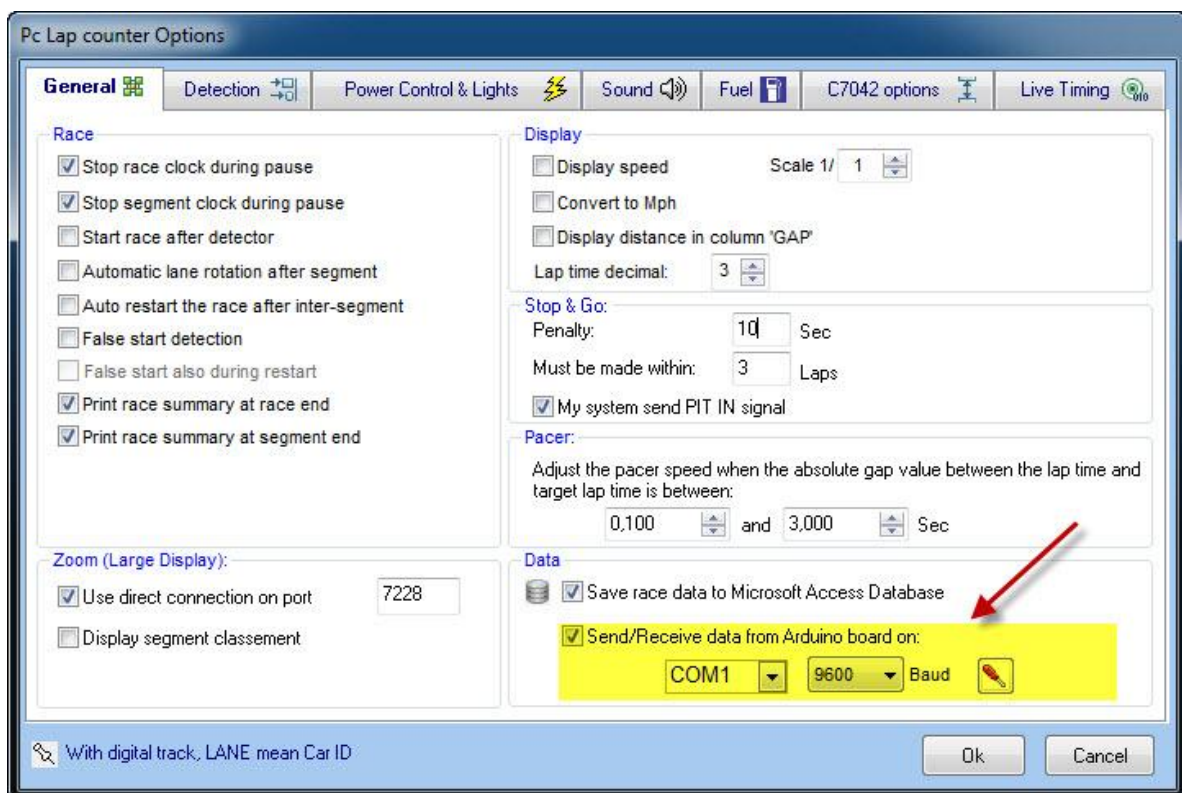
1. Protocol version history: .....	1
2. General consideration .....	2
3. Output protocol (Pc Lap Counter write commands to the COM port) .....	5
4. Input protocol (Pc Lap Counter read commands from the COM port) .....	7
5. Examples .....	8

## 1. Protocol version history:

- Version 1.00 (Pc Lap Counter version 5.26)
- Version 1.01 (Pc Lap Counter version 5.26c):
  - add Stop & Go penalty in the input protocol
- Version 1.02 (Pc Lap Counter version 5.31) :
  - Add Car cross S/F line in the output protocol
  - Add race clock in the output protocol
- Version 1.03 (Pc Lap Counter version 5.34):
  - add fuel level in the output protocol
- Version 1.04 (Pc Lap Counter version 5.35):
  - add transponder ID in the input protocol
  - Modification of the lap time format for message [SF]
- Version 1.05 (Pc Lap Counter version 5.40):
  - You can change the baud rate to use.
- Version 1.06 (Pc Lap Counter version 5.43):
  - You can select “Arduino” as detector type to discharge the main screen of Pc Lap Counter from the lap and the pit lane detection.

## 2. General consideration

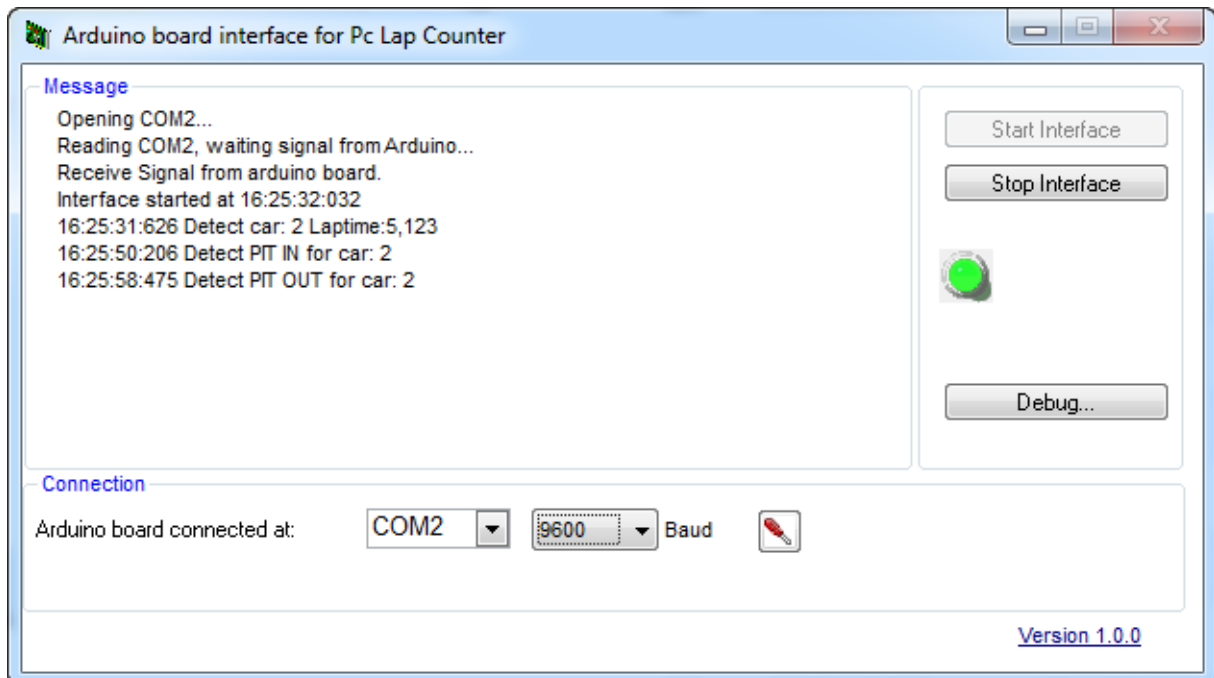
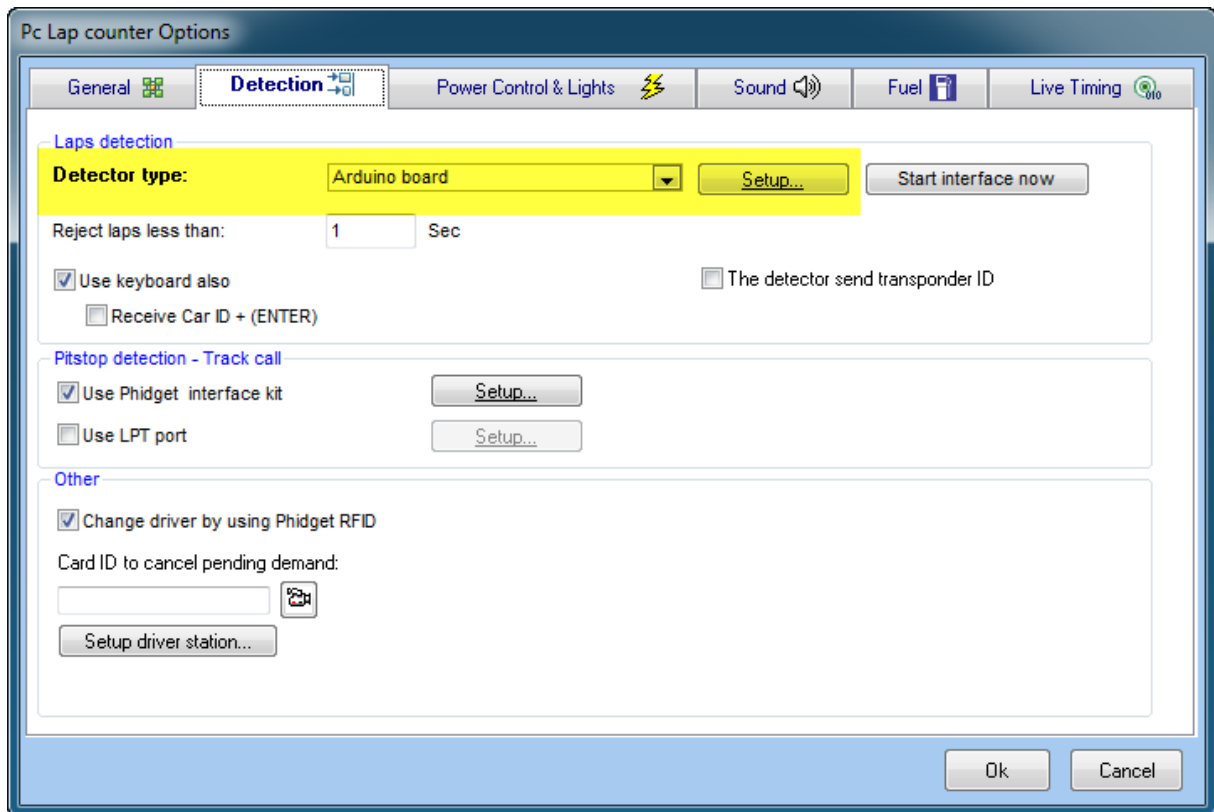
Since version 5.26 Pc Lap Counter can send and receive messages from a COM port and so it's possible to use a Arduino board with Pc Lap Counter. To use Arduino board you need to program a sketch base on the following protocol description ('sketch' = 'program' in Arduino terminology). To activate the « send/receive messages from COM port » feature just tick the following option in Pc Lap Counter :



As soon this option is activated Pc Lap Counter will send messages for starting lights, power control, etc... and can also receive messages for S/F line, Pit Stop, track call and external buttons, etc...

*REM:*

*Since Pc Lap Counter version 5.43 you can also select "Arduino" as detector type, this is useful to discharge the lap and pit lane detection from the Pc Lap Counter main screen and have better display performance:*



*Of course when you select "Arduino" as detector type it means that you dedicate a Arduino board just to receive the lap and the pit lane messages (just to receive the message [SFnn], [PInn],[POnn] and [RF]) and you will have to use another Arduino board to process all the other messages.*

**Protocol description:**

Simple read/write of strings, a Pc Lap Counter message start with the characters « [« and end with the characters « ] ».

nn = lane/car number

ln = lights number

Port setting: 9600 bauds, 8 bit, 1 stop bit, Parity none (since version Pc Lap Counter 5.40 you can change the baud rate to use, by default it use 9600 bauds).

### 3. Output protocol (Pc Lap Counter write commands to the COM port)

Purpose	Command	Remark
<b>Starting lights</b>	[SLIn1] -> on [SLIn0] -> off	In : 01 to 05 = lights 1 to 5 06 = GO light 07 = STOP light 08 = Yellow flag lights
<b>Power control</b>	[PWnn1] -> on [PWnn0] -> off	nn 00 = all lanes/cars
<b>False start lights</b>	[FSnn1] -> on [FSnn0] -> off	
<b>Stop &amp; go lights</b>	[SGnn1] -> on [SGnn0] -> off	
<b>First place lights</b>	[FPnn1] -> on [FPnn0] -> off	
<b>Out of fuel lights</b>	[OFnn1] -> on [OFnn0] -> off	
<b>Low fuel lights</b>	[LFnn1] -> on [LFnn0] -> off	
<b>Max fuel lights</b>	[MFnn1] -> on [MFnn0] -> off	
<b>Pit stop lights</b>	[PTnn1] -> on [PTnn0] -> off	
<b>Car cross the S/F line</b>	[CFnnLxxxPxxTHH:SS.mmm]	<ul style="list-style-type: none"> <li>• nn = car number / lane number</li> <li>• xxxx = number of laps</li> <li>• xx = position in the race (standing)</li> <li>• HH:MM:SS.mmm = lap time</li> </ul> <p><u>Examples :</u>  <b>[CF02L0125P01T00:00:08.556]</b>            Car 2 cross the SF line and get 125 laps, car is first , the lap time is 8.556  <b>[CF04L0123P02T00:00:10.557]</b>            Car 4 cross the SF line and get 123 laps, car position is second , the lap time is 10.557</p>
<b>Race clock</b>	[RCstHH:MM:SS]	<ul style="list-style-type: none"> <li>• s = race status               <ul style="list-style-type: none"> <li>○ 0 --&gt; Race setup</li> <li>1 --&gt; Race started</li> <li>2 --&gt; Race finished</li> <li>3 --&gt; Race paused</li> </ul> </li> <li>• t = clock type               <ul style="list-style-type: none"> <li>○ R = Remaining time (available only with time limited race)</li> <li>○ E = Elapsed time</li> <li>○ S = Segment remaining time (available only with time limited race)</li> </ul> </li> </ul>

- L = Lap remaining (available only with lap limited race and time is replaced by number of laps remaining)

Examples :

**[RC0R03:00:00]** --> Setup a race, remaining time is 3 hours.

**[RC1R02:59:59]** --> Race started and race remaining time is 2 hour, 59 minutes and 59 seconds.

**[RC1E00:00:01]** --> Race started and elapsed time is 1 second.

**[RC3R02:50:00]** --> Race paused and race remaining time is 2 hours and 50 minutes.

**[RC1S01:00:00]** --> Race started and it remain 1 hour in the segment.

**[RC1L20]** --> race started and it remain 20 laps (lap limited race)

**Fuel Level**

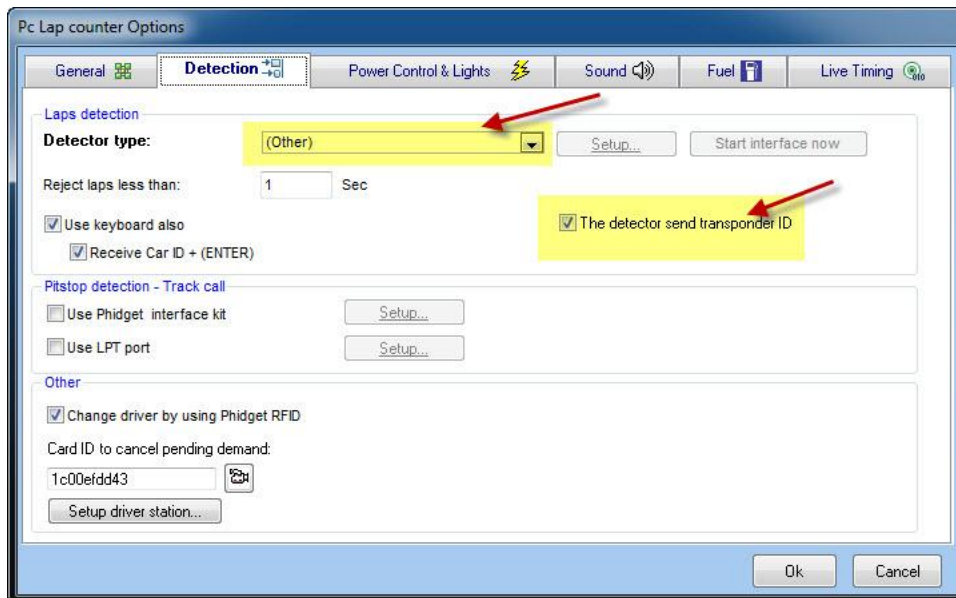
[FLnrrxxx]

- r = Refuel Status
  - 0 = not refueling
  - 1 = Refueling
- xxx = fuel level out of 100

## 4. Input protocol (Pc Lap Counter read commands from the COM port)

Purpose	Command	Remark
Car Cross the S/F line	[SFnn] or [SFnn?HH:MM:SS.mmm] or [SFnn\$laptime]	<ul style="list-style-type: none"> <li>• HH:MM:SS.mmm is the time stamp of the lap</li> <li>• Laptime is the lap time of the lap in milliseconds</li> </ul> <p><u>Examples:</u> The message <b>[SF02]</b> will count a lap for lane 2 and PCLC will calculate the lap time. The message <b>[SF02\$65287]</b> Will count a lap for lane 2 and the lap time will be 1 minute, 5 seconds and 287 milliseconds (00:01:05.287 sec)</p>
Car enter into the pit lane	[PInn]	
Car leave the pit lane	[POnn]	
Start the race	[BT01]	
Restart the race	[BT02]	
Pause the race	[BT03]	
Start/Pause/Restart the race	[BT04]	
Power off	[BT05]	
Power on	[BT06]	
End the race	[BT07]	
Power on/off	[BT08]	
Yellow Flag on/off	[BT09]	
Give a stop & go penalty	[SGnn]	
RFID	[RFtagid]	Where tagid is the tag number
Car cross the S/F line for transponder system (*)	[TRtransponderid] or [TRtransponderid?HH:MM:SS.mmm] or [TRtransponderid\$laptime]	<ul style="list-style-type: none"> <li>• Transponderid is the transponder ID of the car.</li> <li>• HH:MM:SS.mmm is the time stamp of the lap</li> <li>• Laptime is the lap time of the lap in milliseconds</li> </ul> <p><u>Examples:</u> The message <b>[TR02]</b> will count a lap for car with transponder id 02 and PCLC will calculate the lap time. The message <b>[TR02\$65287]</b> will count a lap for car with transponder id 02 and the lap time is 1 minute, 5 seconds and 287 milliseconds (00:01:05.287 sec).</p>

(\*) you can send the transponder ID to Pc Lap Counter, to enable this feature in Pc Lap Counter select "Others" as Detector type and tick the option "The detector send transponder ID" like here:



Also, when you use Pc Lap Counter with a transponder system, you need to link the transponder ID to a car in Pc Lap Counter , look here [http://www.pclapcounter.be/amb20\\_decoder.html](http://www.pclapcounter.be/amb20_decoder.html) for more explanation and specially the section “How to set a race with a transponder system”

## 5. Examples

# Output examples :

Hex Value	ASCII Value
5B 53 4C 30 31 30 5D	[SL010]
5B 53 4C 30 32 30 5D	[SL020]
5B 50 54 30 31 30 5D	[PT010]
5B 46 50 30 31 30 5D	[FP010]
5B 4C 46 30 31 30 5D	[LF010]
5B 4D 46 30 31 30 5D	[MF010]
5B 4F 46 30 31 30 5D	[OF010]
5B 46 53 30 32 30 5D	[FS020]



5B 53 47 30 32 30 5D [SG020]

5B 50 54 30 32 30 5D [PT020]

5B 46 50 30 32 30 5D [FP020]

## Input examples :

---

Hex Value	ASCII Value
5B 42 54 30 31 5D	[BT01]
5B 53 46 30 31 5D	[SF01]
5B 53 46 30 32 5D	[SF02]
5B 53 46 30 31 5D	[SF01]
5B 42 54 30 34 5D	[BT04]
5B 42 54 30 34 5D	[BT04]
5B 52 46 31 61 62 63 31 65 31 31 31 5D	[RF1abc1e111]
5B 54 52 31 32 33 34 35 36 5D	[TR123456]
5B 54 52 31 32 33 34 34 3F 31 33 3A 32 30 3A 31 35 2E 31 32 33 5D	[TR12344?13:20:15.123]