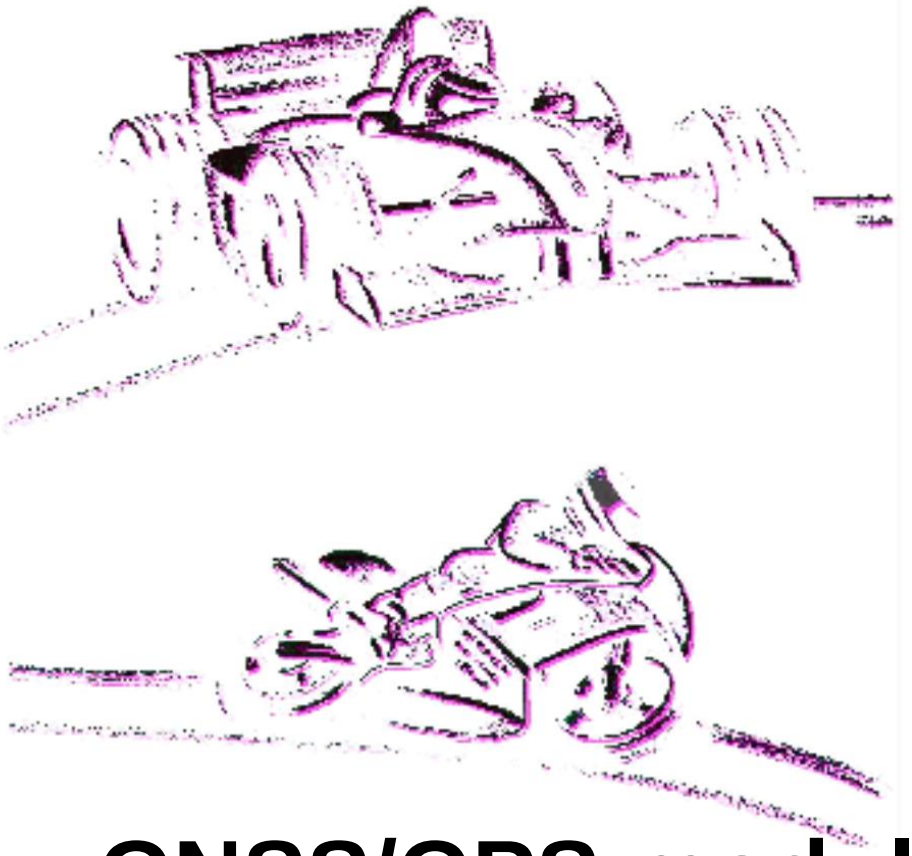
2d-datarecording.com

- English -



GNSS/GPS modules

Creating Lap- and Section-Times

Revision History

Revision	Description	Release Date	Author
0	Initial Release	2014-09-16	
1	Revision 1	2021-07-27	FS
2	Revision 2	2022-03-18	FS

Revision 0:

Initial release of manual

Revision 1:

General revision of the complete manual with regard to all points and improvement of the individual points through the addition of illustrations

Revision 2:

Improvement of description of LapGPS and setting examples.

Content

1 Notes and symbols used in this Manual

Further Information
In the paragraphs highlighted with this symbol, you will find tips and practical advice to work with the 2D-Software.

Documentation reference
Documentation reference to another manual or handout

Important information
It is very important to follow the instructions given

2 Preface

Before setting the laptime function please see the general GPS/GNSS manual, which contains the setting steps of GPS/GNSS module with further, important information, and ensure that everything is set correctly.

Documentation reference
The general *GPS/GNSS Manual* can be found at download area of the 2D website:

www.2D-Datarecording.com/manuals/

At 2D laptimes can be online created in many different ways, e.g., using GPS coordinates or TransponderX2 messages as laptrigger signals.

Documentation reference
Respective manual for creating laptimes via TransponderX2 messages can be found at our website:
www.2D-Datarecording.com/manuals/

For teams using TransponderX2 and Laptiming via GPS alternately, the GPS/GNSS2CAN module is able to create laptrigger signals as normally via GPS coordinates but sends laptrigger signal on TransponderX2-CAN-Identifier to simulate the TransponderX2 module.

This means that when switching between the two laptime generation options, only the modules need to be changed and no settings need to be changed!

Further Information
For easy switching between the two laptime generation options, an adapter cable can be purchased from 2D to connect the GPS/GNSS2CAN module to the system's designated TransponderX2 connector.

Documentation reference
Please see the manual *GPS/GNSS Modules* at download area of the 2D website for setting example: www.2D-Datarecording.com/manuals/

Creating and analysing laptimes is also possible in post-processing via Analyzer.

Documentation reference
Please see the manual *Laptiming Analyzer* at download area of the 2D website:

www.2D-Datarecording.com/manuals/

3 Preparations

Various 2D modules are capable of calculating a laptime and since for most applications it is not sufficient to only calculate the laptime but also to display it, a sensible structure, i.e., where the Lap- and SecTime is calculated or displayed, should be set up, which strongly depends on the later use.

It must be distinguished between calculating, recording, and displaying Lap- or SecTimes.

Calculating:

Basically, the calculation of Lap- and SecTimes must only be done in **one** module of the system.

The following modules are able to calculate laptimes:

- Logger
- Dashboards
- GPS/GNSS2CAN

With serial GPS/GNSS modules it is only possible to calculate laptimes in logger (with serial GPS input).

The Lap- and SecTimes can then be send via CAN to other devices for recording or displaying, but by calculating it only in one module ensures that calcuted, recorded and displayed Lap- and SecTimes matching.

Recording:

If the Lap- and SecTimes are calculated in the **logger**, the laptime is also recorded.

However, if the Lap- and SecTimes are calculated in a **GPS/GNSS2CAN** module or in a **dashboard**, the calculated Lap- and SecTimes must be sent via CAN to the **logger**, where the transmitted Lap- and SecTime channels are linked to the LapTime and SecTime events of the **logger** for recording.

Displaying:

However, if the Lap- and SecTimes are calculated in a GPS/**GNSS2CAN** module or in a **logger**, the calculated Lap- and SecTimes must be sent via CAN to the **dashboard**, where the transmitted Lap- and SecTime channels are linked to the LapTime and SecTime events of the **dashboard** for displaying.

The via CAN received Lap- and SecTime should **not** be used directly on dashboards output channels but should be linked to the dashboards LapTime und SecTime event to ensure that also other display-internal functions like count channel Laps or GAP time are working correctly.

⇨ Only if all steps of this manual are followed correctly, the calculated Lap- and SecTime will correspond to the Lap- and SecTime shown on a display!

⇨ Please see the laptimes setting example in appendix (6.1) for choosing the correct way to set up your system!

4 Laptimes

Further Information
Valid for GPS/GNSS2CAN and serial GPS/GNSS modules!

This document is describing how to set up lap and section timing functionality via **GPS coordinates**.

Basically, GPS laptimes are created from a trigger channel *#LapGPS*, which reduces its value from 65535 the closer it gets to a defined position, where also the radius and current speed (4.2) are having an impact on the trigger channel *#LapGPS*.

For laptimes, a finish line predefined by 2D is used as the trigger position (4.1.1) or the user can enter coordinates manually (0).

The trigger signal is then used to calculate the laptimes via the Laptimes event (0)!

Important information
#LapGPS is the only valid trigger channel for GPS laptime creation!

Further laptime settings must be done in laptime event (0)!

Important information
Loggers, Dashboards and GPS/GNSS2CAN-modules are able to create a trigger signal!

It is important that laptimes should always be calculated in just one module of a system only and, if available, is only used in other modules!

Which module is used for laptime calculation depends on the application.

Please see the setting examples in appendix (6.1)!

- Further information to *Laptimes* event be found at *Events* manual: http://2d-datarecording.com/downloads/manuals/

4.1 Laptrigger position

There are two different possibilities, the laptrigger position can be defined:

Automatically - using the table *2DTrkPos* (already loaded at modules 2D factory setting)
Manually – manual input of finish line coordinates

No matter if the laptrigger GPS position is entered manually or automatically, it is important that only the *LapGPS* channel should be used of the module which calculates laptime.

The LapGPS channel must therefore never be sent via CAN for GPS lap time generation but only the laptime event channel of the module which has calculated the laptime via the *LapGPS* channel!

The following modules are able to calculate laptimes:

- Logger
- Dashboards
- GPS/GNSS2CAN

Please see the laptimes setting example in appendix (6.1) for different possibilities to set up system.

Important information
This is also valid when laptimes created with serial GPS/GNSS module because the LapGPS-Coordiantes are then entered in Loggers-*LapGPS*-Channel!

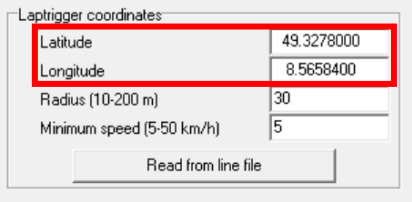
4.1.1 Automatic detection of laptrigger position

To enable the automatic detection of laptriggers, the current GPS position is permanently compared with a table called *2DTrkPos* which contains the coordinates of the finish lines of over 300 circuits all over the world *(see chapter* 6.3*)*. At every power-on the modules calculates the distance between its current GPS position and the nearest entry of 2DTrkPos table.

If the module detects at power on that it is within a radius of 5 km from a finish line stored in 2DTrkPos, the corresponding finish line coordinates are automatically entered in the *#LapGPS* channel.

Thereby, no laptrigger coordinates must be typed in!

Example Hockenheim:



Important information
An automatic entry only takes place if the currently entered LapGPS coordinates are not within a 5 km radius around the GPS coordinates of a 2DTrkPos entry or are 0 for both Latitude and Longitude.
If already entered LapGPS coordinates are within 5 km of a radius around a 2DTrkPos entry, these entries are considered fine adjustments and are therefore not overwritten by automatic detection of laptrigger position.
Within these 5 km only the 2DTrkPos GPS coordiantes are only entered automatically again, if the LapGPS coordiantes are 0 or more than 5 km away from current position at the at a power on.
If the system detects during a power-on that it is now within a 5 km radius of another entry in the 2DTrkPos table, which is the case, for example, when changing to a different race track, the (manually) entered LapGPS coordians of the last race track will be overwritten, since this track is no longer within the 5 km radius of the currently recognized 2DTrkPos entry.

Important information
Radius and minimum speed entries must also be set correctly at automatically laptrigger detection

If the distance from current position to a finish line in table is smaller than the manually defined radius and the speed is higher than the minimum speed, a laptrigger signal is automatically set by #LapGPS.

Important information
The 2D module is now just able to create laptrigger signal but no laptime itself! Therefore, the trigger channel *LapGPS* must be linked to LapTime-Event (see 0)

Further Information
Please see appendix 6.2 how to add laptriggers to *2D_TrkPos.tbl* for automatic laptrigger detection.

4.1.1.1 Example Almeria/Andalucía circuit

The Almeria (green) and Andalucía (red) circuit are right next to each other, where only the coordinates of the Almeria circuit are in the 2DTrkPos table (it is not possible to have two entries in 2DTrkPos within a 5 km radius).

Since both circuits are within a radius of 5 km, the GPS laptrigger coordinates are automatically entered as LapGPS coordinates when the system is turned-on on one of the two circuits.

At the beginning of the event, the GPS laptrigger coordinates of the Andalucía circuit must be entered manually as LapGPS coordinates.

These coordinates will not be overwritten by the coordinates of the Almeria circuit, even after the system is switched off, because the manually entered coordinates are also within the 5 km radius around the Almeria GPS laptrigger position of the 2DTrkPos entry.

If the racetrack is then changed, e.g. to Jerez, the coordinates entered in LapGPS are automatically overwritten with the Jerez coordinates when the system is switched on.

Andalucía circuit laptrigger position:

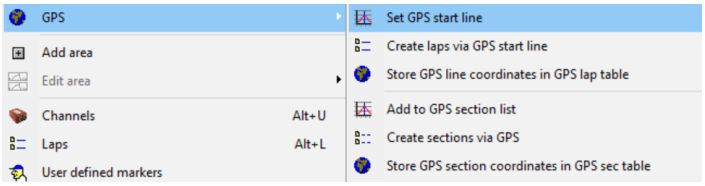
Latitude: 37.08900139
Longitude: -2.27276268

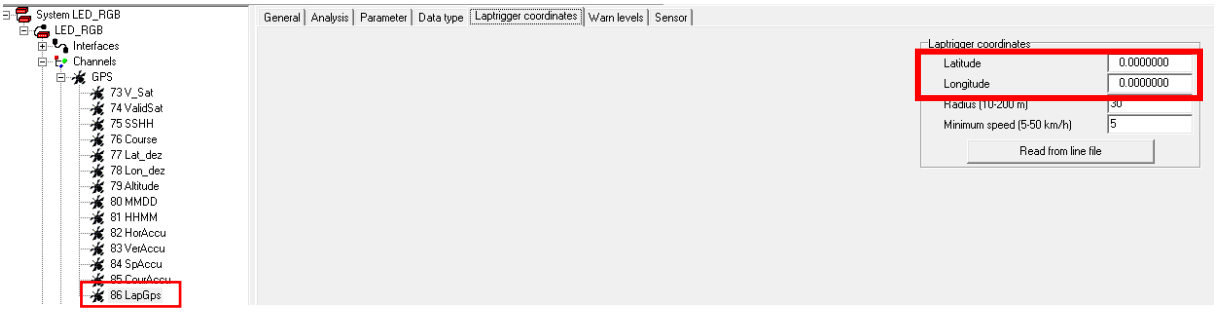
4.1.2 Manual detection of laptrigger position

If there is a need to force a module to use a certain finish line or manually enter GPS coordinates, there is the possibility to do this in the channel settings of channel *#LapGps*.

Either the Latitude and Longitude values can be entered directly, or it is also possible to load a line file which contains the Latitude and Longitude values created by Analyzer.

Further Information
Make an installation lap to get an accurate GPS measurement of the circuit layout and open respective measurement in Analyzer. Set marker by left click on desired finish line position. Open submenu by right click and navigate to *Set GPS start line* and save line file to respective event. The line file can now be loaded in WinIt via button *Read from line file.*





Important information
Radius and minimum speed entries must also be set correctly at manually laptrigger detection

Confirm your changes with <**Apply**>.

If the distance from current position to the manually defined *laptrigger coordinates* is smaller than the manually defined radius and the speed is higher than the minimum speed, a laptrigger signal is set by #LapGPS.

Important information
The 2D module is now just able to create laptrigger signal but no laptimes itself! Therefore, the trigger channel *LapGPS* must be linked to LapTime-Event (see 0)!

4.1.3 LapDetect event GPS/GNSS2CAN

- The LapDetect events are only available in GPS/GNSS2CAN modules!

The *LapDetect* event can be used to manually set the current Latitude and Longitude coordinates as laptrigger GPS coordinates of channels #LapGPS **online on track**.

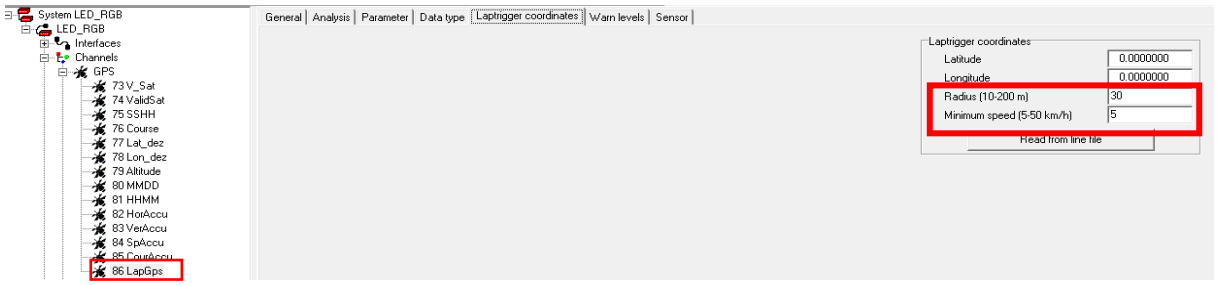
As laptrigger-set-condition any channel, e.g. analogue channel connected to light switch or horn button, can be used.

Laptrigger coordinates are then used for Laptimes event to create Laptimes via GPS coordinates.

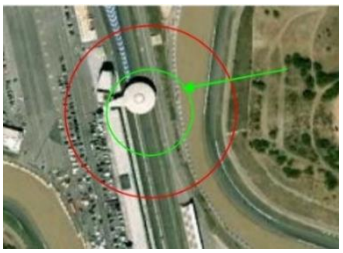
- Detailed instructions on how to use the *LapDetect* event be found at *Events* manual: http://2d-datarecording.com/downloads/manuals/

4.2 LapGPS - Defining radius and minimum speed at trigger point

To avoid invalid laptrigger signals at narrow circuit layouts, the tab *Laptrigger coordinates* settings of GPS-channel *#LapGPS* can be used to adjust generation of laptriggers:



The defined search radius must be large enough to accommodate different driving lines as well as GPS drift. However, if the search radius is too large it may be possible to generate invalid lap times due to the lap signal occurring more than once in a lap, see figure below!



Speed at lap trigger point	Recommended radius
<180 km/h	25 m
180 – 240 km/h	35 m
>240 km/h	45 m

4.3 Selecting the trigger channel for LAPTIME event

Important information
#LapGPS is the only valid trigger channel for GPS laptime creation!

Further laptrigger settings of *#LapGPS* must be done in previously!

The following modules are able to calculate laptimes:

- Logger
- Dashboards
- GPS/GNSS2CAN

This event function can be found inside the system tree via ***<Channels>*** → ***<Event>*** → ***<Laptime>*** and calculates the laptimes from the trigger signal of *#LapGPS*.

To turn on this function, open the tab ***<General>,*** then select the *Turned on* checkbox.

To define the triggering channel, open the tab ***<Parameter>***, select the correct trigger channel (*#LapGPS*) from the *Channel-number* dropdown-box.

The ***<Timeout>*** option can be used to prevent repeated laptime activation (false lap triggers or shortcuts) by entering a time value (in milliseconds), ensuring no laptimes are generated until the time elapses.

The frequency at which the triggering channel is checked for changes is defined by the ***<Samplingrate>.***

If a GPS-channel (e.g. *#LapGPS*) is selected at channel-number, the trigger threshold area inputs are ineffectual and only the laptrigger coordinates settings of #LapGPS are important for calculating the laptime.

Confirm your changes with <**Apply**>.

The Laptime is now calculated in the Laptime Event channel of this module!

Further Information
If the Lap- and SecTimes are calculated in the **logger**, the laptime is also recorded.

If the Lap- and SecTimes are calculated in **GPS/GNSS2CAN** or **Dashboard**, the respective Event channel must be send to **logger** for recording and to **dashboard** for displaying.

4.3.1 Sending *LAPTIME* event via CAN

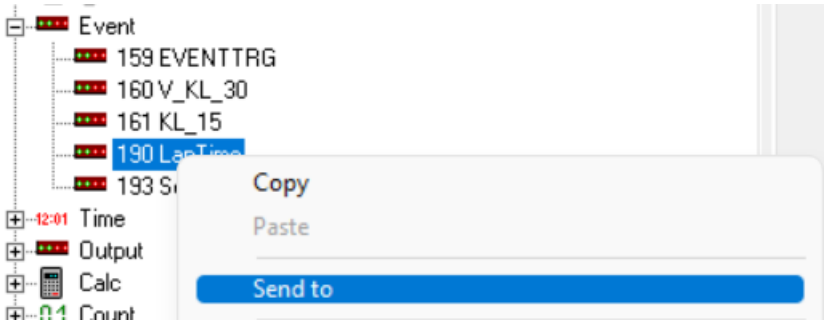
If the laptime calculation is **not** performed in the logger, the laptime **event** channel of the module that calculated the laptime must be transmitted via CAN, so that the laptime can be recorded in a logger or displayed in a display.

The Laptime-Event-channel can be sent via CAN to other 2D CAN modules (e.g. Dashboards) or recorded like other channels!

The LapGPS channel must therefore never be sent via CAN for GPS lap time generation but only the laptime event channel of the module which has calculated the laptime via the *LapGPS* channel!

Important information

To send the laptime event channel via CAN, please use the Send-To function from the submenu and use 2D-Laptime-CAN-Identifier 0x1A0 (if not already used).



5 Section times

Further Information
Valid for GPS2CAN and serial GPS modules!

Basically, GPS section times are also created from trigger channel *#LapGPS*, which reduces its value from 65535 the closer it gets to a defined position. In case of GPS section times the position is only defined by table entries. The trigger signal is then used at the Sectimes-Event to calculate section times!

Further Information
Loggers, Dashboards and GPS/GNSS2CAN-modules are able to create a trigger signal!

It is important that section times should always be calculated in just **one module of a system only** and, if available, is only used in other modules!

- Further information to *Sectimes* event be found at *Events* manual: http://2d-datarecording.com/downloads/manuals/

5.1 Section trigger positions

Compared to laptriggers, there are no predefined sections for section triggers. However, the section trigger can be loaded to a 2D-module via section table *2DSecPos*, so that these sections are available online on 2D modules for further use in dashboards on track.

Important information
For the creation of section triggers, a complete round with GPS data must be available as a measurement in the Analyzer!

KIT-version users are not able to use section triggers online on the 2D module!

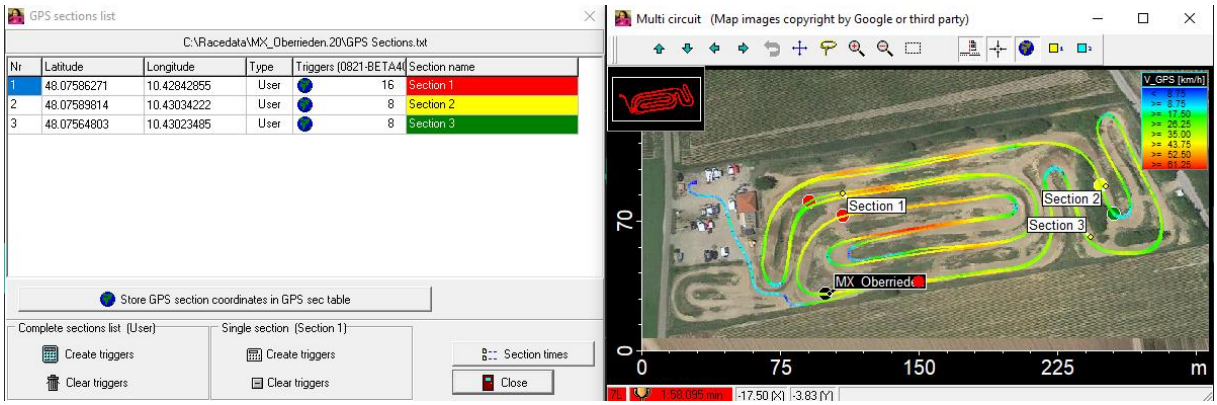
In case that section triggers are not already part of the table *2DSecPos*, it can be added manually.

If a new section trigger should be added to *2DSecPos*, please execute the following steps:

- Make an installation lap to get an accurate GPS measurement of the circuit layout
- Open the measurement in Analyzer, press <**space**> to enter measure-mode and put the cursor on your preferred section trigger and left-click to set a marker on the preferred position.

Further Information
It is helpful to open Multi Circuit (<**m**>) or Circuit (<**c**>) plots

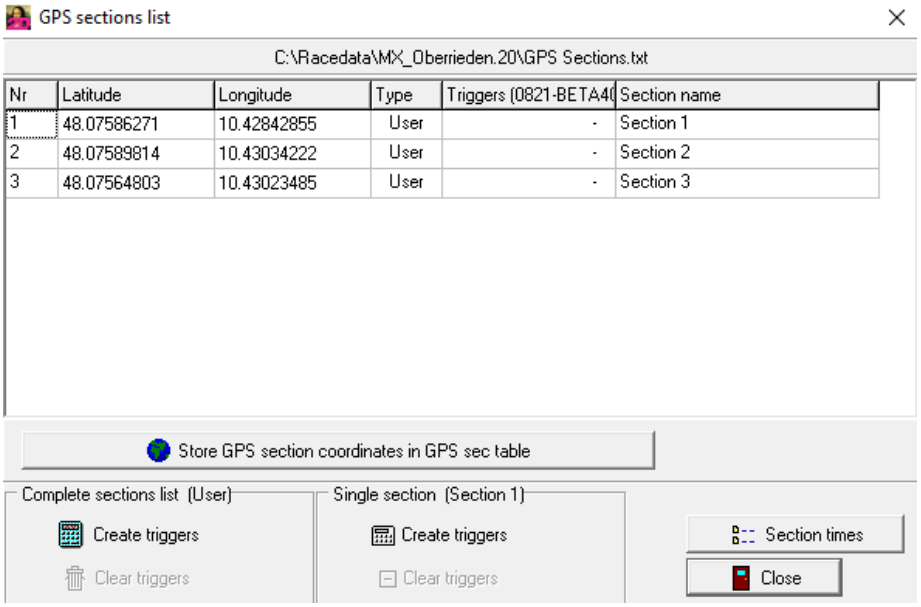
- To add the marker position to section list, right-click with mouse on an arbitrary position in Analyzer, select *GPS* → *Add to GPS section list* and chose a name of section trigger



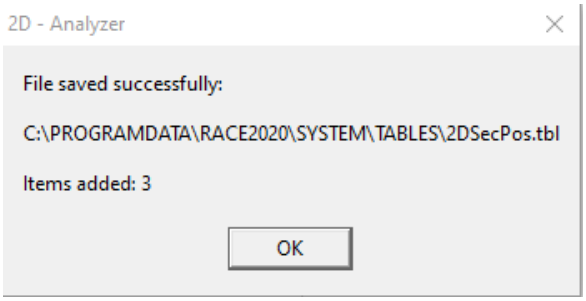
Further Information
Fill the GPS section list with as desired.

The GPS section list shows all available GPS sections triggers of an event, but only sections triggers which are matching to current measurement are written in black!

- Click button <**Create triggers**> from *Complete sections list* area to create sections in current measurement



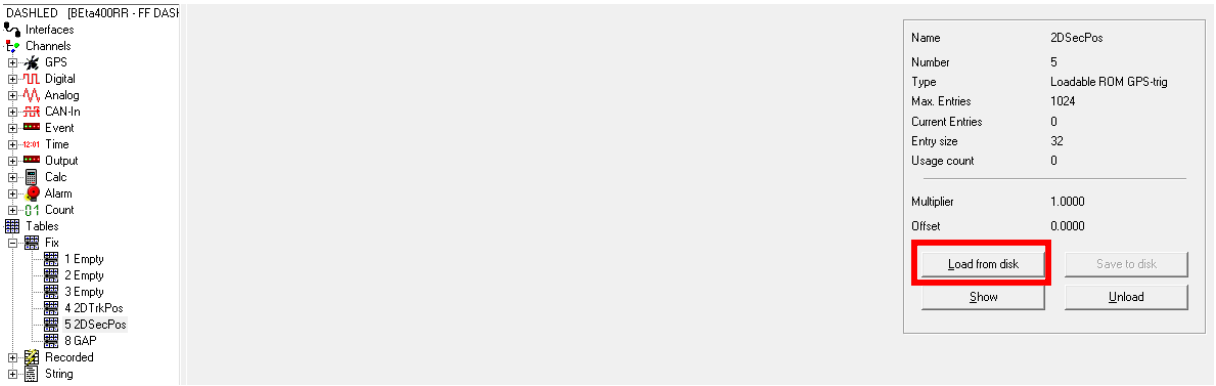
- Click button <**Store GPS section coordinates in GPS sec table**> to store defined sections in *2DSecPos*



- Open WinIt and connect the desired 2D module

Further Information
With this step, the *2DSecPos* table is updated/loaded in the desired 2D module, to enable automatic laptrigger detection for the desired 2D module.

- Select table *2DSecPos,* click on button <**Load from disk, go to the directory**> where the tables *2DSecPos* was saved to in the previous step and load *2DSecPos* to 2D module.



- Press <**Apply**> to save new *2DSecPos* table on 2D module

Important information
The 2D module is now just able to create section trigger signal but no section time itself!

Therefore, the trigger channel #LapGPS must be linked to SecTime-Event (see 5.2)!

5.2 Selecting the trigger channel for SECTIME event

It is important that only the *LapGPS* channel should be used of the module which calculates SecTime.

The LapGPS channel must therefore never be sent via CAN for GPS SecTime generation but only the SecTime event channel of the module which has calculated the SecTime via the *LapGPS* channel!

Please see the laptimes setting example in appendix (6.1) for different possibilities to set up system. (these examples are also valid for SecTime creation via GPS.)

Important information
#LapGPS is the only valid trigger channel for GPS section time creation!

Further section trigger settings of LapGPS must be done in laptimes chapter!

The following modules are able to calculate laptimes:

- Logger
- Dashboards
- GPS/GNSS2CAN

This event function can be found inside the system tree via ***<Channels>*** → ***<Event>*** → ***<Sectime>*** and calculates the section times from the trigger signal of *#LapGPS*.

To turn on this function, open the tab ***<General>,*** then select the *Turned on* checkbox.

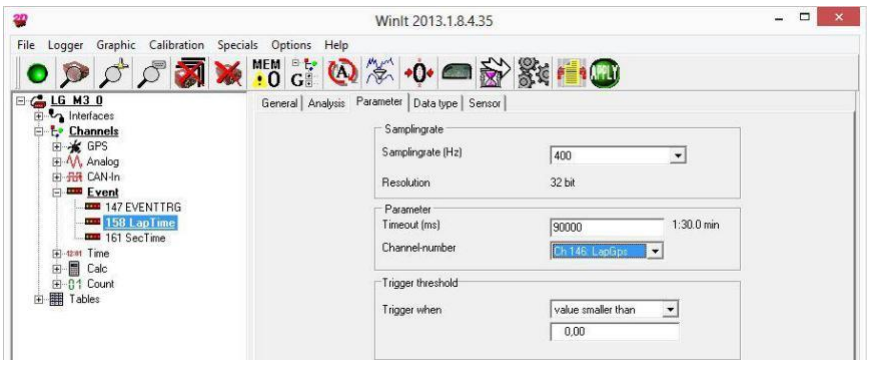
To define the triggering channel, open the tab ***<Parameter>***, select the correct trigger channel (*#LapGPS*) from the *Channel-number* dropdown-box.

The ***<Timeout>*** option can be used to prevent repeated section time activation (false lap triggers or shortcuts) by entering a time value (in milliseconds), ensuring no section times are generated until the time elapses.

The frequency at which the triggering channel is checked for changes is defined by the ***<Samplingrate>.***

If a GPS-channel (e.g. *#LapGPS*) is selected at channel-number, the trigger threshold area inputs are ineffectual and only the laptrigger coordinates settings of #LapGPS are important for calculating the Sectime.

Confirm your changes with <**Apply**>.



The Sectime is now calculated in the Sectime Event channel of this module!

Further Information
If the Lap- and SecTimes are calculated in the **logger**, the laptime is also recorded.

If the Lap- and SecTimes are calculated in **GPS/GNSS2CAN** or **Dashboard**, the respective Event channel must be send to **logger** for recording and to **dashboard** for displaying.

Documentation reference
Respective Dashboard manual for further Section times display settings can be found at 2D website: www.2D-Datarecording.com/manuals/

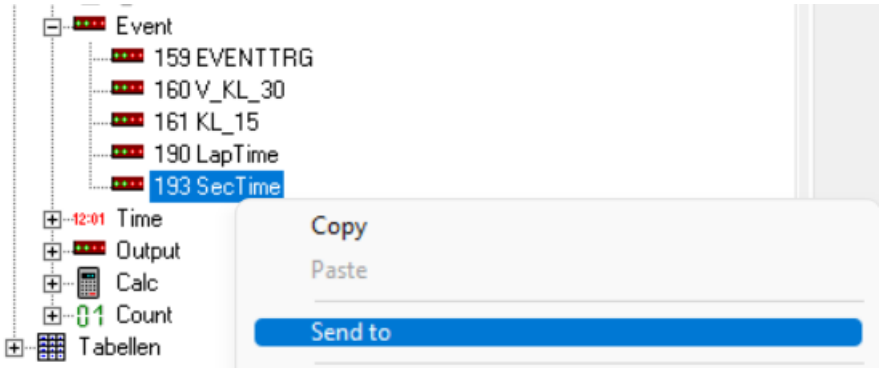
5.2.1 Sending *SecTime* event via CAN

If the SecTime calculation is **not** performed in the logger, the SecTime **event** channel of the module that calculated the SecTime must be transmitted via CAN, so that the SecTime can be recorded in a logger or displayed in a display.

The SecTime -Event-channel can be sent via CAN to other 2D CAN modules (e.g. Dashboards) or recorded like other channels!

The LapGPS channel must therefore never be sent via CAN for GPS SecTime generation but only the SecTime event channel of the module which has calculated the SecTime via the *LapGPS* channel!

Important information
To send the SecTime event channel via CAN, please use the Send-To function from the submenu and use 2D- SecTime -CAN-Identifier 0x1B0 (if not already used).



6 Appendix

6.1 Laptime setting examples

As already mentioned, various 2D modules are capable of calculating a laptime and since for most applications it is not sufficient to only calculate the laptime but also to display it, a sensible structure, i.e. where the laptime is calculated or displayed, should be set up, which strongly depends on the later use.

It must be distinguished between calculating, recording, and displaying Lap- or SecTimes.

Calculating:

Basically, the calculation of Lap- and SecTimes must only be done in **one** module of the system.

The following modules are able to calculate laptimes:

- Logger
- Dashboards
- GPS/GNSS2CAN

With serial GPS/GNSS modules it is only possible to calculate laptimes in logger (with serial GPS input).

The Lap- and SecTimes can then be send via CAN to other devices for recording or displaying, but by calculating it only in one module ensures that calcuted, recorded and displayed Lap- and SecTimes matching.

Recording:

If the Lap- and SecTimes are calculated in the **logger**, the laptime is also recorded.

However, if the Lap- and SecTimes are calculated in a **GPS/GNSS2CAN** module or in a **dashboard**, the calculated Lap- and SecTimes must be sent via CAN to the **logger**, where the transmitted Lap- and SecTime channels are linked to the LapTime and SecTime events of the **logger** for recording.

Displaying:

However, if the Lap- and SecTimes are calculated in a GPS/**GNSS2CAN** module or in a **logger**, the calculated Lap- and SecTimes must be sent via CAN to the **dashboard**, where the transmitted Lap- and SecTime channels are linked to the LapTime and SecTime events of the **dashboard** for displaying.

The via CAN received Lap- and SecTime should **not** be used directly on dashboards output channels but should be linked to the dashboards LapTime und SecTime event to ensure that also other display-internal functions like count channel Laps or GAP time are working correctly.

When linking a via CAN received laptime event channel to a laptime event for recording or displaying, the trigger threshold option **Value @ change** must be used!

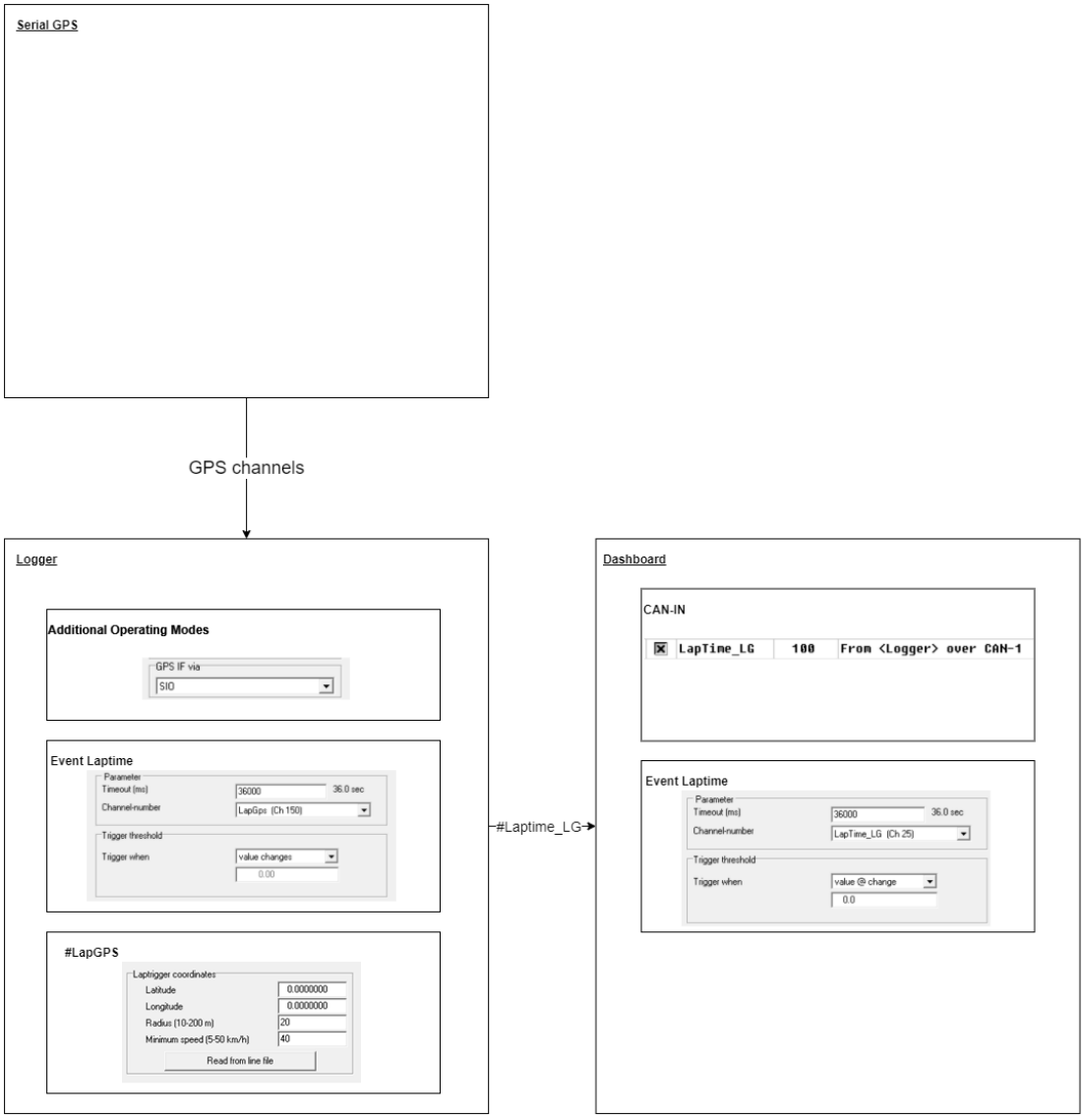
⇨ Only if all steps of this manual are followed correctly, the calculated Lap- and SecTime will correspond to the Lap- and SecTime shown on a display!

⇨ Please see the laptimes setting example in appendix (6.1) for choosing the correct way to set up your system!

6.1.1 Example 1 – Creating laptimes by GPS-Positions in Logger – Serial GPS/GNSS

The setup is explained in this example by the fact that the laptrigger signals are generally acquired via a GPS-positions, received by **serial GPS/GNSS**.

In this example, the **Logger** calculates, records, and transmits the laptime via CAN to the Dashboard (for displaying). Transferring the in logger calculated laptime to dashboard ensures that the recorded lap time and the displayed lap time match.



Value @ change: Laptime is not calculated but received laptime can used for displaying or recording!

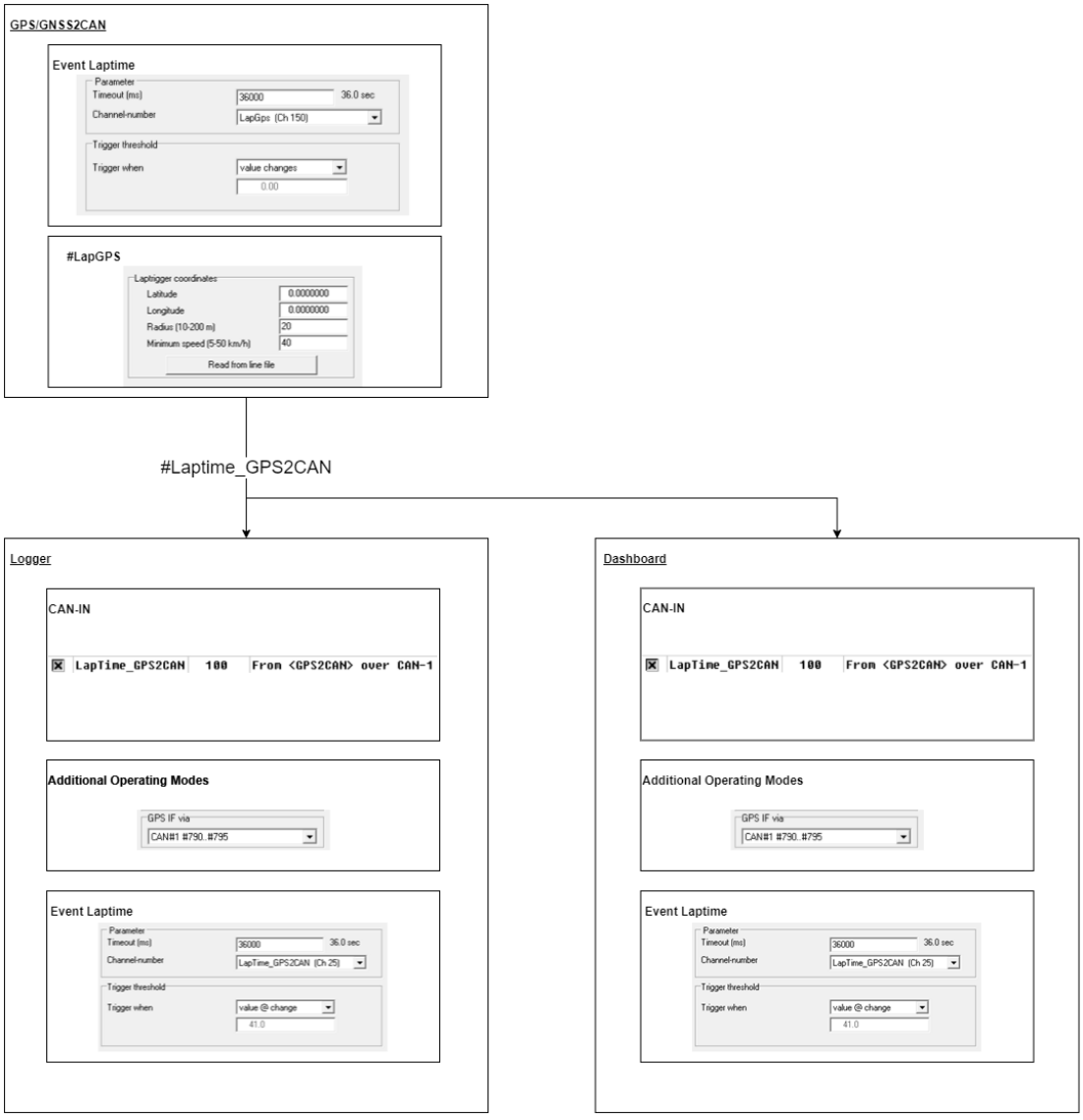
With calculating laptimes via LapGPS (in logger) the trigger treshold options are insufficient!

6.1.2 Example 2 – Creating laptimes by GPS-Positions in GPS/GNSS2CAN

The setup is explained in this example by the fact that the laptrigger signals are generally acquired via a GPS-positions, received by **GPS/GNSS2CAN**.

In this example, the **GPS/GNSS2CAN** calculates and transmits the laptime via CAN to the logger (for recording) and Dashboard (for displaying). Transferring the in GPS/GNSS2CAN calculated laptime to logger and dashboard ensures that the recorded lap time and the displayed lap time match.

Further Information
In contrast to the serial GPS, the GPS2CAN can already calculate laptimes itself and pass them on to other CAN bus participants.



Value @ change: Laptime is not calculated but received laptime can used for displaying or recording!

Value changes: Laptime is calculated, recorded, and also can be used for displaying!

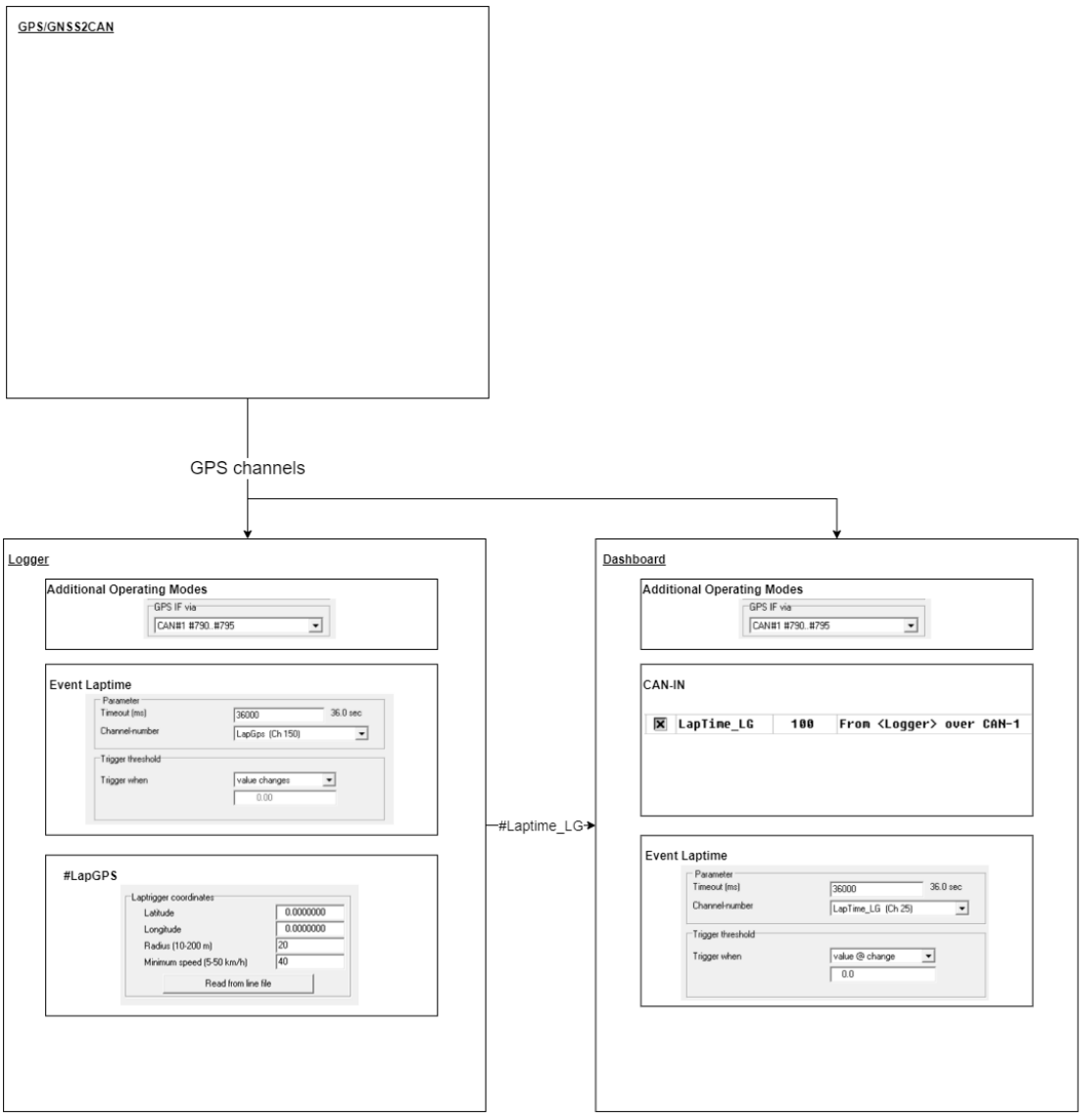
6.1.3 Example 3 – Creating laptimes by GPS-Positions in Logger – GPS/GNSS2CAN

The setup is explained in this example by the fact that the laptrigger signals are generally acquired via a GPS-positions, received by **GPS/GNSS2CAN**.

In this example, the **Logger** calculates records, and transmits the laptime via CAN to the Dashboard (for displaying). Transferring the in logger calculated laptime to dashboard ensures that the recorded lap time and the displayed lap time match.

Further information

The laptime is also recorded in logger.



Value @ change: Laptime is not calculated but received laptime can used for displaying or recording!

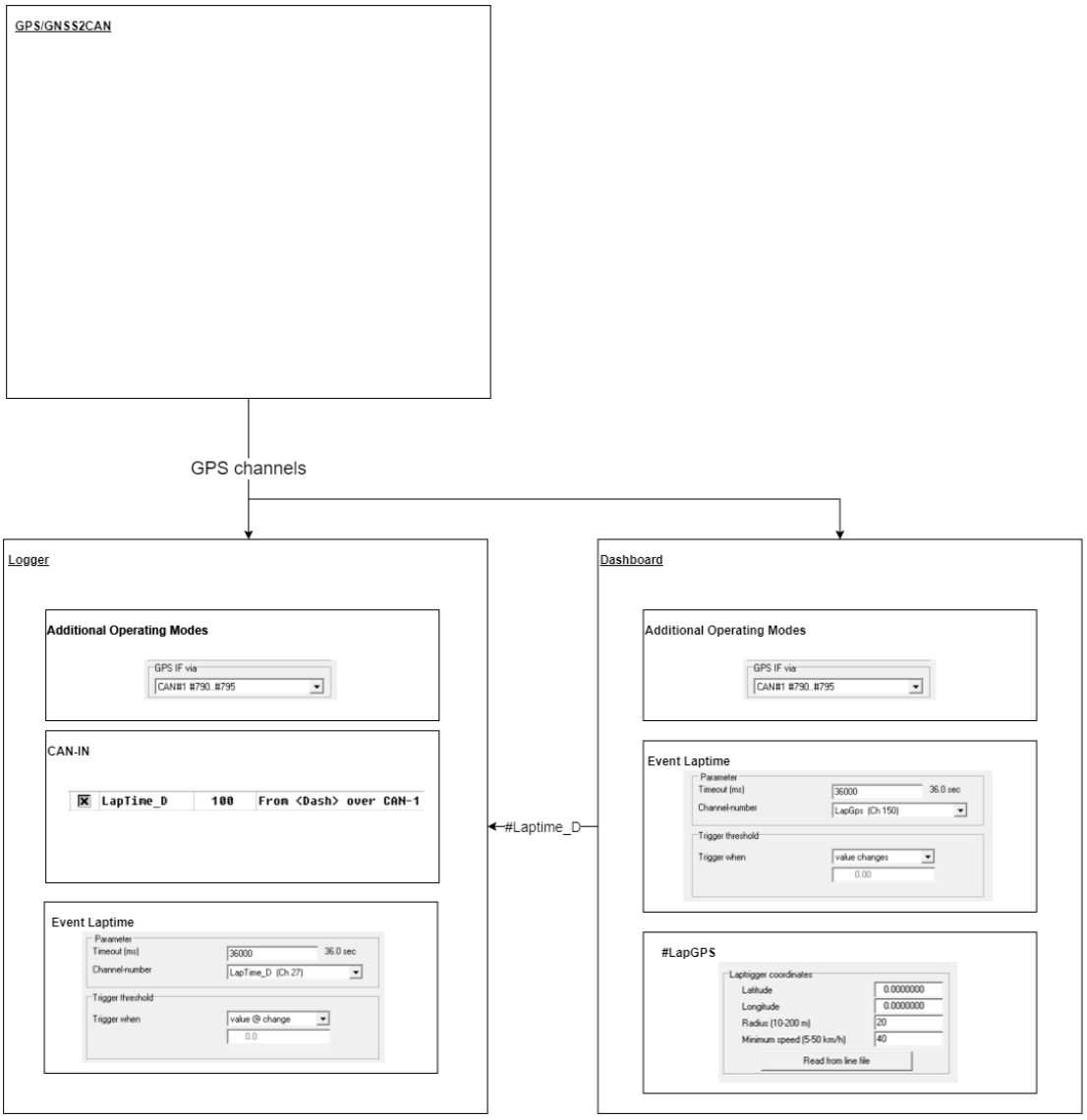
Value changes: Laptime is calculated, recorded, and also can be used for displaying!

6.1.4 Example 4 – Creating laptimes by GPS-Positions in Dash – GPS/GNSS2CAN

The setup is explained in this example by the fact that the laptrigger signals are generally acquired via a GPS-positions, received by **GPS/GNSS2CAN**.

In this example, the **Dashboard** calculates and transmits the laptime via CAN to the logger (for recording). Transferring the in dashboard calculated laptime to logger ensures that the recorded lap time and the displayed lap time match.

Further Information
The laptime created by Dashboard is recorded in Logger



Value @ change: Laptime is not calculated but received laptime can used for displaying or recording!

Value changes: Laptime is calculated, recorded, and also can be used for displaying!

6.2 Creating new entry in 2DTrkPos -table

Further Information
If circuit is not part of the table *2DTrkPos*, first the 2D software should be searched for updates to ensure the latest version of table is checked.

In case that a start line coordinates are not already part of the table *2DTrkPos*, it can be added manually.

If a new finish line should be added to *2DTrkPos*, please execute the following steps:

- Make an installation lap to get an accurate GPS measurement of the circuit layout
- Open the measurement in Analyzer, press <**space**> to enter measure-mode and put the cursor on your preferred start-line position and left-click to set a marker on the preferred position.

Further Information
It is helpful to open Multi Circuit (<**m**>) or Circuit (<**c**>) plots

- To manually create laptriggers at the installation lap measurement, right-click with mouse on your preferred start line position, select *GPS → Set GPS start line* and save line-file with name of the track in desired directory (e.g. event directory).

- To load the previously set laptrigger coordinates to *2DTrkPos* and *2DTrkNme* tables, right-click with mouse on arbitrary position in measurement, select *GPS →Store GPS line coordinates in GPS lap table*, open the previously saved line-file and type in the name of the track in automatically opened track-window.

- By clicking <**Ok**>, the line coordinates and the name of the track are added to the respective tables!

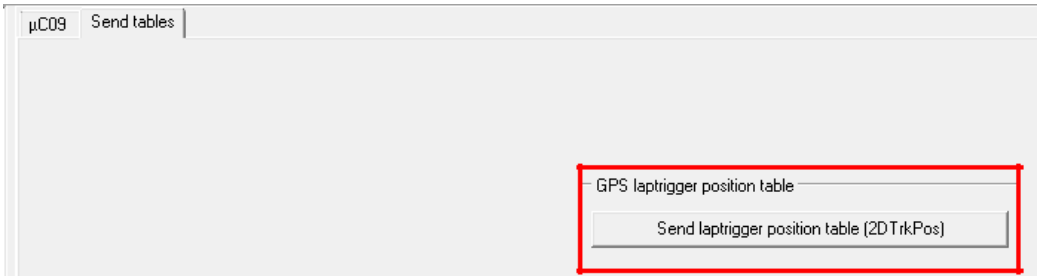
- Open WinIt and connect the desired 2D module

Further Information
There are differences between standard Winit- and KIT-WinIt-versions!

Standard version:

Select table *2DTrkPos,* click on button <**Load from disk**>, go to the directory where the tables *2DTrkPos* and *2DTrkNme* are saved to in the previous step and load *2DTrkPos*.

KIT version:



Select tab *Send tables,* click on button <**Send laptrigger position table (2DTrkPos)**>, go to the directory where the tables *2DTrkPos* and *2DTrkNme* are saved to in the previous step and load *2DTrkPos*.

Further Information
With this step, the *2DTrkPos* table is updated in the desired 2D module, to enable automatic laptrigger detection for the desired 2D module.

- Press <**Apply**> to save new *2DTrkPos* table on 2D module. Now the finish line is also available for automatic finish line detection on this 2D module!

6.3 Testing Laptime in box

Because it is not always possible to test that Laptimes are created and displayed correctly, the following example can be used to test the Laptime creation in the box without going out on track.

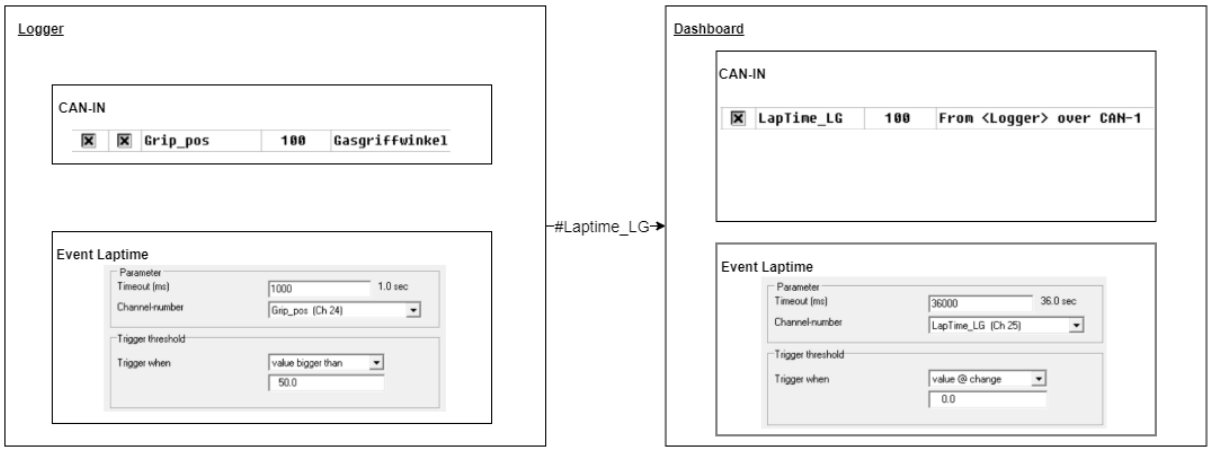
Further Information
It is assumed that the laptime is generated in a logger, which is connected to a bike's CAN bus to access the vehicles data, and the laptime is only displayed at a dashboard (Example 2)

Please note down the correct settings of creation of laptrigger via GPS coordinates

In this example the channel *#Grip_pos* is received via CAN bus and recorded in the loggers CAN-IN channels.

This channel is used as Trigger channel for Laptime event with the condition that a trigger is only created when the value of #Grip_pos is bigger than 50.

Important information
Timeout values must also be adjusted to test creation of laptimes in box!



Now the throttle grip of the bike can be used to trigger the Laptimes event, and the dashboard can be used to check if laptime is created correctly.

Important information
Please reset the parameters for creation of laptimes via GPS coordinates!

6.4 List of race tracks in 2DTrkPos-table

Abbeville	Gingerman Raceway	Norisring
Achna Speedway	Goodwood	Nürburgring
Adelaide	Gotland Ring	Okayama
Adria	Grattan	Oran Park
Ahvenisto	Grobnik	Oschersleben
Aiginio	Guadix	Oulton Park
Aintree	Guapore	Padborg Park
Alastaro	Hampton Downs	Palm Beach
Albacete	Hastings Motorsport Park	Pannoniaring
Albert Park	Hawai Raceway	Park Algar
Alcarrás	Heartland Park of Topeka	Pau Arnos
Ales	High Plains Raceway	Paul Ricard
Almeria (Circuit De Almeria)	Hockenheim	Paw
Anglesey	Homebush	Penbry International
Anhembi - Sao Paulo	Homestead	Phillip Island
Anneau du Rhin	Honjo	Pinarbasi
Ansan	Hungaroring	Pittsburgh
Artic Circle Raceway	Imola	Pocono Raceway
Ascari	Inde Motorsports Ranch	Pomposa
Aschheim	Indianapolis	Port Elizabeth
Assen	Infineon Raceway	Portland
Atlanta Motor Speedway	Interlagos	Poznan
Autodromo da Salta	Istanbul Park	Prestwold Hall
Autodromo Goianina	Jacarepagua	Pukekohe
Autodromo Hermanos Rodrigues	Jarama	Pukekohe Park Raceway
Autopolis	Jennings	Putnam Park
Bahrain	Jerez	Quensland
Barbagallo	Johor	Ramenskoe
Barber Motorsports Park	Jyllandsring	Red Bull Ring
Baskerville	Kansas Speedway	Reno Fernley
Bathurst	Kartodromo Fatima	Riberao Preto
Bay Meadows	Katar- Losail	Ring Djursland
Beave Run Motorsports Complex	Kemora	Rioveggio
Bedford Autodrome	Kentucky Speedway	Road Atlanta
Belle Isle	Kinnekulle	Roberto Muras Circuit
Bilster Berg	KIP-Palmela	Rockingham
Blackhawk Farms	Knockhill	Ruapuna
Blyton Park	Knutstorp	Rudskogen

Bombarral	Korea Intertional Circuit	Sachsenring
Botniaring	Korfez	Saellandsringen
Braga	Kyalami	Salzburg
Brainerd	LaFerteGaucher	Samara-Ring
Brands Hatch	Laguna Seca	San Luis Potosi
Brasilia	Lakeside Queensland	Sandia
Bresse	Las Vegas Motor Speedway	Sandown
Bristol	Ledenon	Santa Cruz do Sul
Brno	LeMans	Sauga Circuit Auduring
Broadford	Lime Rock	Schleizer Dreieck
Bruntingthorpe	Ljungbyhed Park	Sebring
Buddh International Circuit	Llandow	Seinajoki
Bukernieki	Loheac	Sendai Highland
Buttonwillow Raceway	Lonato	Sentul
Cadwell Park	Londrina	Sepang
Calabogie	Loudon	Serres
Calafat	Lydden	Shanghai
Calder	Magione	Siena
California Speedway	Magny Cours	Silverstone
Campo Grande	Mallala	Slovakia Ring
Cape Town	Mallorca-Rennarena	Snetterton
Carolina Motorsport Park	Mallory Park	Spa Francorchamps
Cartagena	Manfeild	Spa Nishiura Motor Park
Cascavel	Mantorp	Spring Mountain
Castelloli	Martinsville Speedway	St Petersburg Street Circuit
Castle Combe	Mas du Clos	Sturup
Catalunya	Maze	Sugo
Charlotte	Megara	Summit Point Motorsports Park
Chayka	Memphis	Surfers Paradise
Chelsea Handling Track	Mendig	Sviestad
Chennai	Miachkovo	Symmons Plains
Chenviers	Michelin Laurens Proving Grounds	Talladega Superspeedway
Chicago	Mid America	Taruma
Chicagoland	Mid Ohio	Taupo
Chuckwalla Valley Raceway	Midvaal	Teretonga
Circuit of the Americas	Miller Motorsports Park	Texas Motor Speedway
Croft	Milwaukee	Three Sisters
Croix-en-Ternois	Misano	Thruxton
Curborough	Mittsverigebanan	Thunderhill Raceway Park
Curitiba	Modena	Timaru
Darlington Speedway	Mont Tremblant	Tocancipa
Darwin	Monteblanco	Tokachi

Daytona	Montreal	Track
Dijon	Monza	Toronto Motorsports Park
Donington Park	Morgan Park	Tsukuba
Dover	Mornay	Valencia
Dreux	Moroso	Valerbanen
Dubai	Mosport	Valkenswaard
Dunsfold	Most	Vallelunga
East London	Motegi	Varano
Eastern Creek	Motopark	Velo Citta - Mogi Guacu
Ebisu South	Motorland Aragon	Velopark
Elvington	Motorland Suzuka	Virginia International Raceway
Enna Pergusa	Mugello	Wakefield Park
Estoril	Myrtle Beach	Waneroo
Eurospeedway Lausitzring	Nakhonchaisri	Waterford Hills
Eusebio	Nardo	Watkins Glen
Fay de Bretagne	Nashville	Welkom
Fiorano	Navarra	Wesbank Raceway
Folembray	Nazareth	Willow Springs
Fort Devens	Nelson Ledges	Winton
Franciacorta	New Jersey Lightning	Yas Marina
Fuente Alamo	New Jersey Thunderbolt	Zandvoort
FujiSpeedWay	Nikko	Zolder
Gateway International	Nogaro	Zuhai
Gellerasen Circuit	Nola Motorsports Park	Zwartkops Raceway