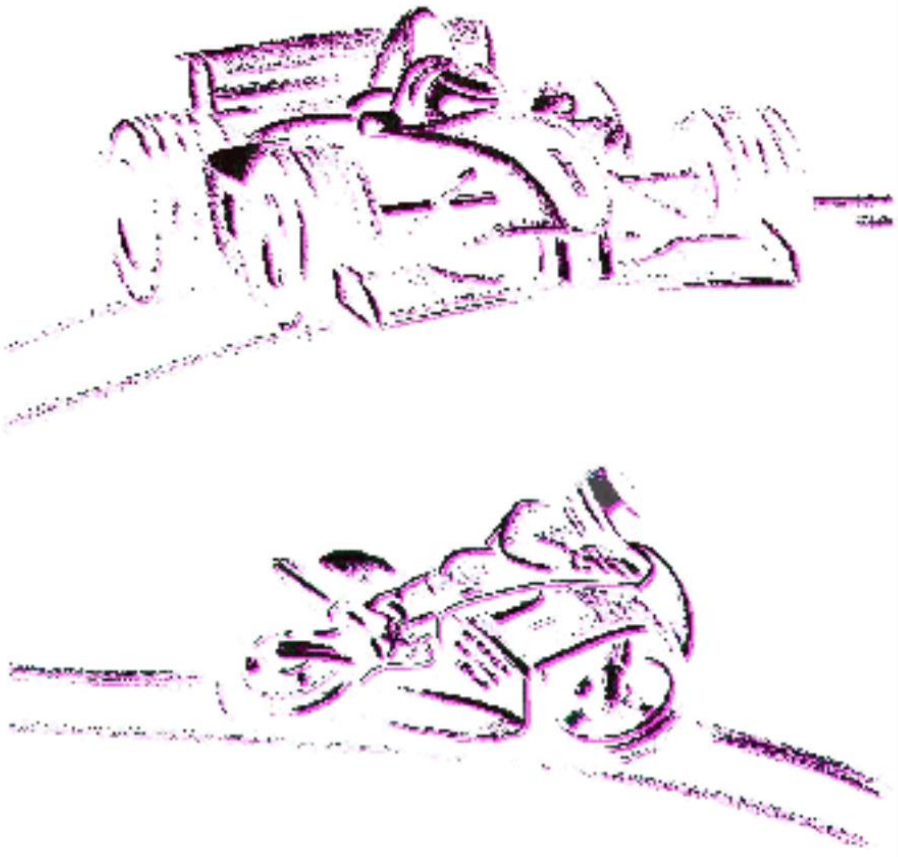
2d-datarecording.com

- English -



2D_GPSTracks

1 Table of Contents

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

2 Revision History

Revision	Description	Release Date	Author
0	Initial Release	2021-01-20	FS
1	Revision 1	2022-01-11	FS
2	Revision 2	2022-09-06	FS

Revision 0

Initial release of *2D_GPSTracks* manual

Revision 1:

Linking latest manuals in documentation references and adding use-cases (12.2 & 12.3) of 2D_GPSTracks toolchain in appendix.

Revision 2:

Adding GPSTrack-Section (5.3), Distance-to-Apex (5.4) and Slalom Skiing (5.5) examples to manual.

3 Notes and symbols used in this Manual

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

- Documentation reference to another manual

- In the paragraphs highlighted with this symbol, you will find additional information. It is particularly important that you follow the instructions given.

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

4 Introduction

During various vehicle development runs, it is often necessary to evaluate measurement data only at certain measuring points on the measurement track.

In order to be able to achieve this, in addition to the desired measurement data, GPS/GNSS positions and times can also be recorded so that the measurement data, such as engine speeds or throttle position, can be then linked with GPS/GNSS information during evaluation.

The subsequent evaluation of these tests takes place in the *Analyzer*, whereby so-called *Calculation Files* (CAL files) are used for the further processing of the recorded channels.

- For more information about *Calculation Files* and the *CalcTool* please visit the download area of our website and have a look at the *CalcTool* manual.
  https://2d-datarecording.com/en/downloads/manuals/

For the above-mentioned purposes, 2D has developed the special CAL-Files *2D_GPSTracks*, with which trigger points are generated by the GPS-*Latitude* and -*Longitude* channels during the analysis of the data, which can then be converted into various measurement tracks.

The trigger points and measurement tracks are then available as **individual channels** and can therefore be used in the *Analyzer* for linking with the actual vehicle measurement data much better than the pure *Latitude* and *Longitude* values

- The idea of trigger points and measurement tracks can be transferred to all other developmental purposes to simplify data analysis

These channels can be used in a further special 2D calculation file, called *Validate_Tracks*, in which channel-marking, e.g. evaluation of a channel only if speed > 50 km/h, is possible and also other useful channels for data analysis are created.

- For more information on *2D_ValidateTracks* please visit the download area of our website and have a look at the *2D_ValidateTracks* manual.
  https://2d-datarecording.com/en/downloads/manuals/

The individual trigger points and measurement tracks channels or the channels created by *Validate_Tracks* can then be used…

- in ***CalcTool*** for further data analysis (e.g. with RMS (Root Mean Square) or VDV (Vibration Dose Value))
- as ***Phases*** for generating a condition, which can be used for optical marking or for selecting certain data
- in ***Analyzer*** tools like *MinMax tables*, *exports,* or *XY-plots*.

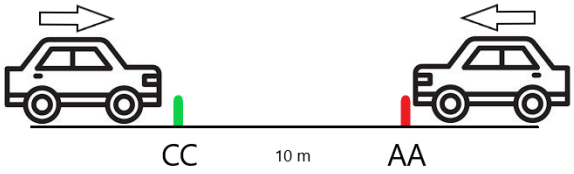
- For more information on ***CalcTool***, ***Phases***, ***Plots*** *and* ***Exports*** please visit the download area of our website and have a look at the respective manuals.
  https://2d-datarecording.com/en/downloads/manuals/

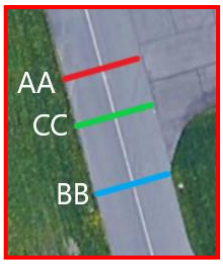
Some measurement tracks have special requirements, e.g. that they have to be driven back and forth on the same track, which can also be intercepted by the *2D_GPSTrack* calculation files, since it can be chosen that a measurement track is only generated if the corresponding two trigger points have been driven in the correct sequence by using e.g. the GPS-Course.

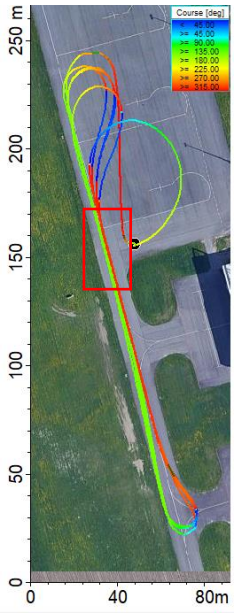
This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

5 Examples

5.1 Example 1







A vehicle is to drive on the measuring track shown above. On the test track, due to the conditions on the test track, it must be driven back to the start AA of the test track in reverse order via the trigger points over the test field.

To save valuable test time, a measurement should also be taken on the return journey.

The GPS channel *Course*, which represents the orientation of the GPS module from 0-360 degrees, is taken into account to differentiate the entry direction of the measurement tracks. In adjacent figure the whole test track is displayed with a coloured differentiation of *Course*.

- When entering the measuring track from trigger point AA, the channel *Course* supplied by the GPS module has the value 165 degrees, whereas if trigger point BB is passed first, *Course* has the value 345 degrees.

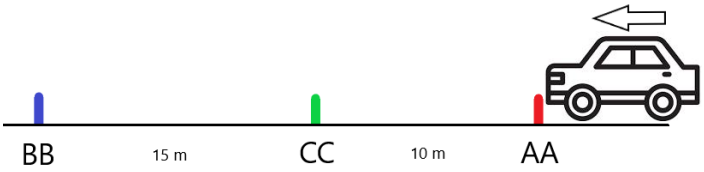
The measurement tracks were therefore previously defined by the user as follows:

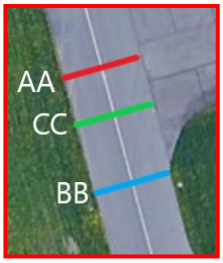
Measurement track 1	AA	→	CC
Measurement track 2	CC	→	AA

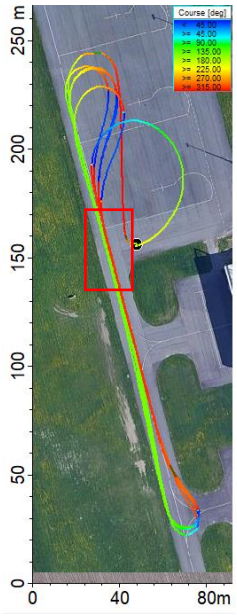
- A similar setting of the *2D_GPSTracks*-CALs can be found in chapter 8.3

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

5.2 Example 2







A vehicle is to travel along a measuring section divided into different sections. The different sections are, as shown in the upper view, measured at different distances, so the test drive is only valid if the test section was entered from point AA.

On the test track, due to the conditions on the test track, it must be driven back to the start AA of the test track in reverse order via the trigger points over the test field.

The GPS channel *Course*, which represents the orientation of the GPS module from 0-360 degrees, is taken into account to differentiate the entry direction of the measurement tracks. In adjacent figure the whole test track is displayed with a coloured differentiation of *Course*.

- When entering the measuring track from trigger point AA, the channel *Course* supplied by the GPS module has the value 165 degrees, whereas if trigger point CC is passed first, *Course* has the value 345 degrees.

In this example, individual areas as well as the complete measuring section from point AA to CC are to be analysed.

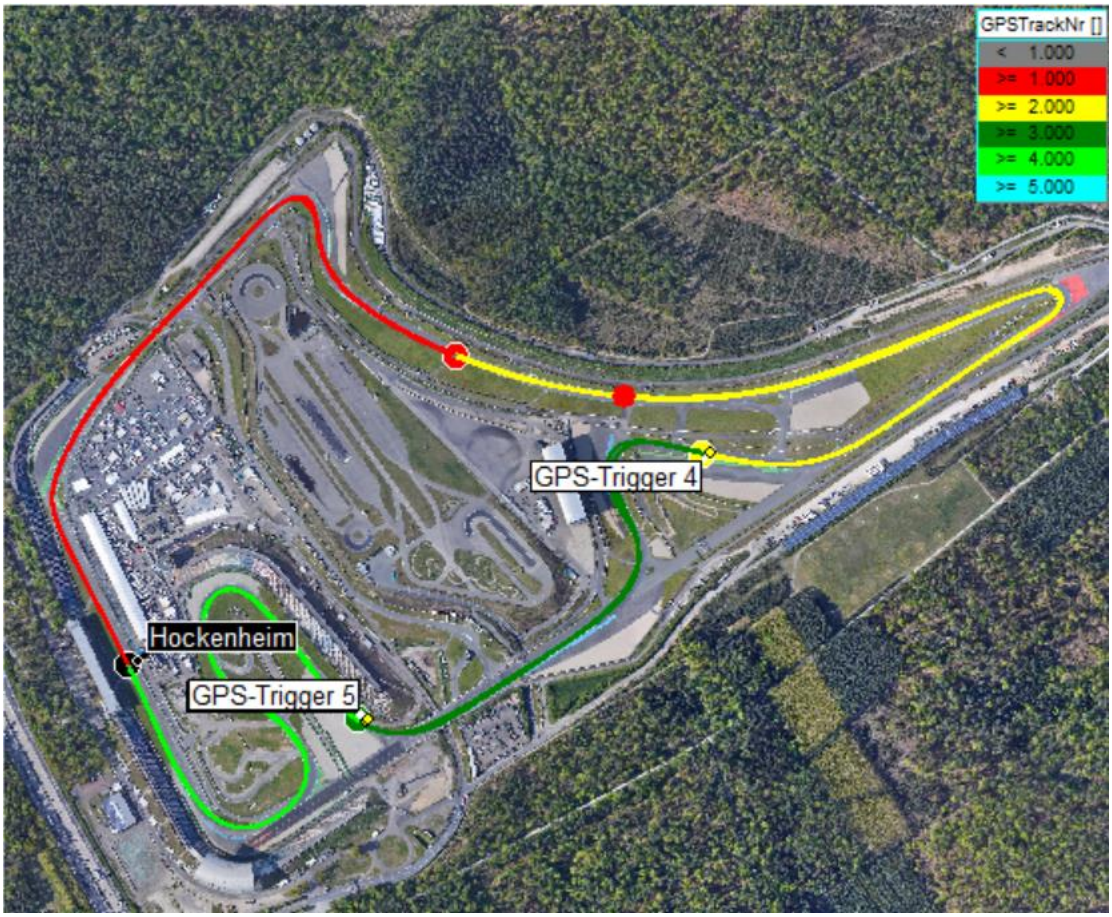
The measurement tracks were therefore previously defined by the user as follows:

Measurement track 1	AA	→	CC
Measurement track 2	CC	→	BB

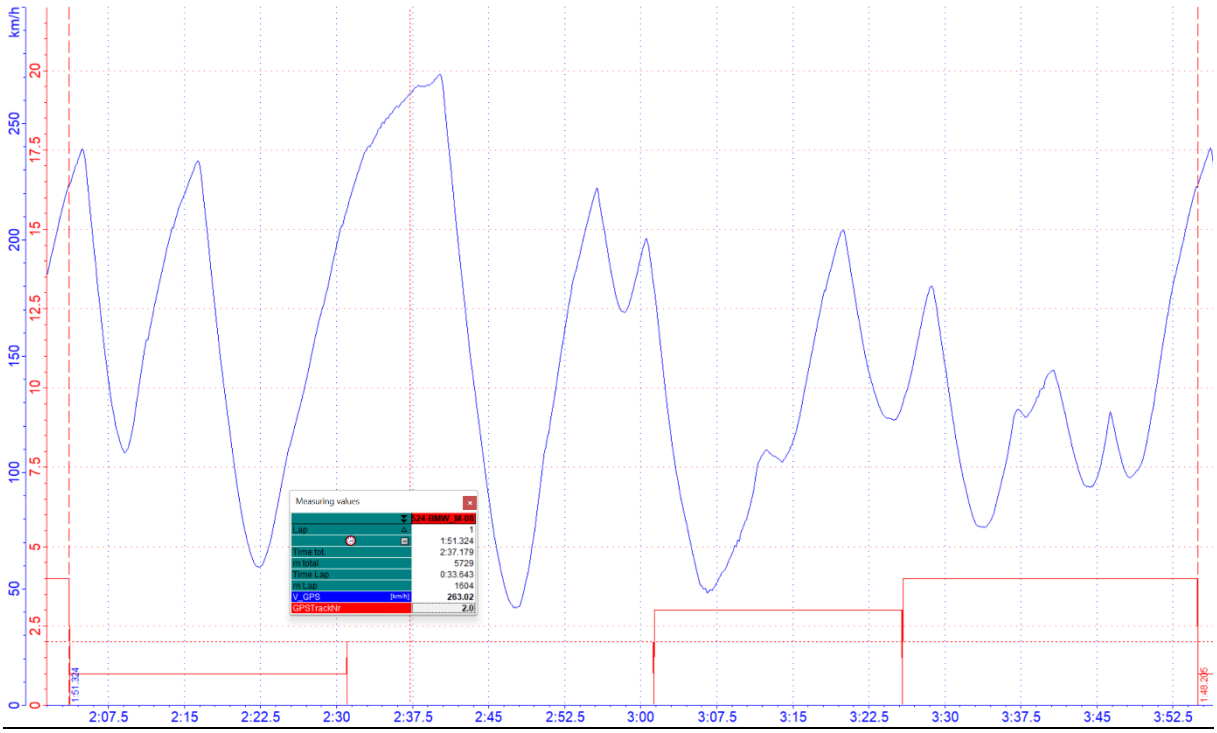
- The setting of the 2D_GPSTracks-CALs can be found in chapter 8.3

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

5.3 Example 3 – GPSTracks Sections



Beside using GPSTracks-toolchain for open-road testing and at testing-grounds, the toolchain can also be used at racetracks to separate a racetrack in different sections:



This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

5.3.1 GPSTracks-Section: Times

Furthermore, the already existing 2D-Section-Time tool, like the Section times tables, can be used with GPSTracks-Segments to e.g., compare the different section times from all laps:

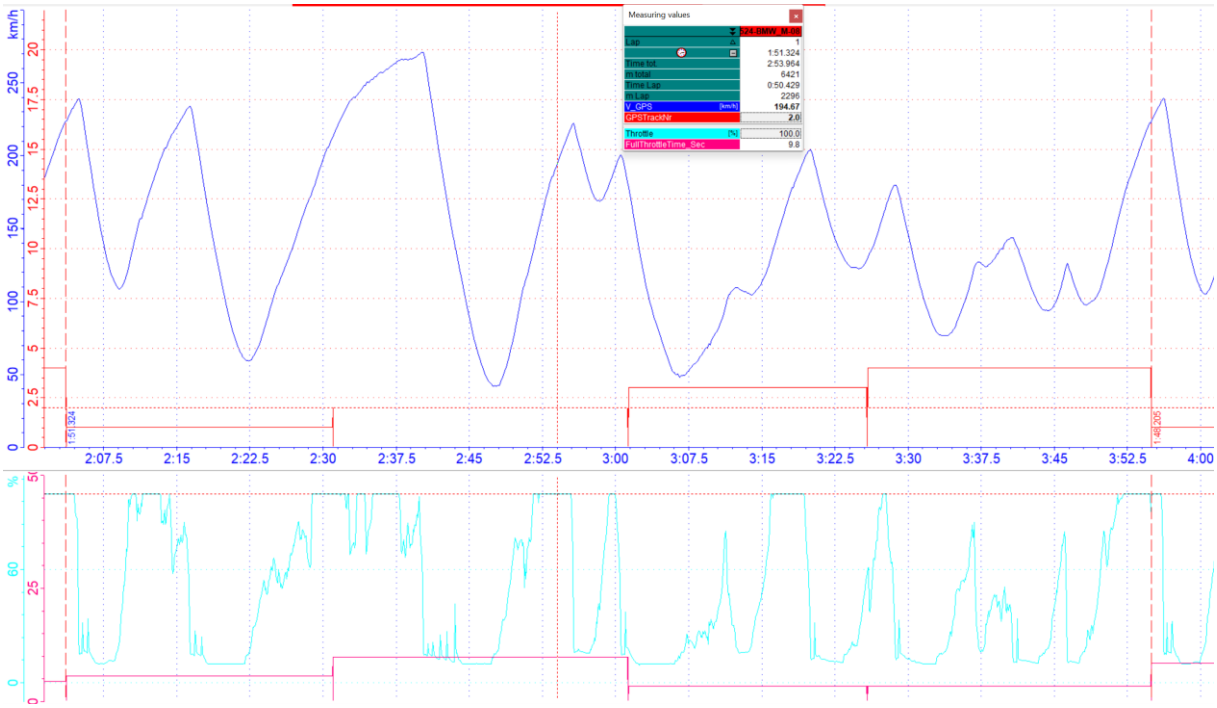
Section times (0524-BMW_M-08)

Lap	Lap time m Lap	GPS-Trigger 3 1179m		GPS-Trigger 4 2693m		GPS-Trigger 5 3498m		T4 4567m	
0	2:03.536 min							0:29.206	1036.61
1	1:51.324 min	0:27.414	1147.22	0:30.280	1470.28	0:24.520	796.58	0:29.110	1032.06
2	1:48.205 min	0:27.750	1144.11	0:29.890	1468.64	0:21.650	781.99	0:28.914	1036.61
3	6:10.456 min	0:27.696	1143.57	0:30.080	1463.35	0:21.500	780.23	4:51.180	1197.31
Ideal time	1:47.719 min	0:27.414	1147.22	0:29.890	1468.64	0:21.500	780.23	0:28.914	1036.61

5.3.2 GPSTracks-Section: Further evaluations

With GPSTracks-Sections it is possible to further analyse the different sections, what means that different parameters (V_Max, V_AVG, …) as well as any vehicle parameters can be linked and calculated by GPSTracks-Sections.

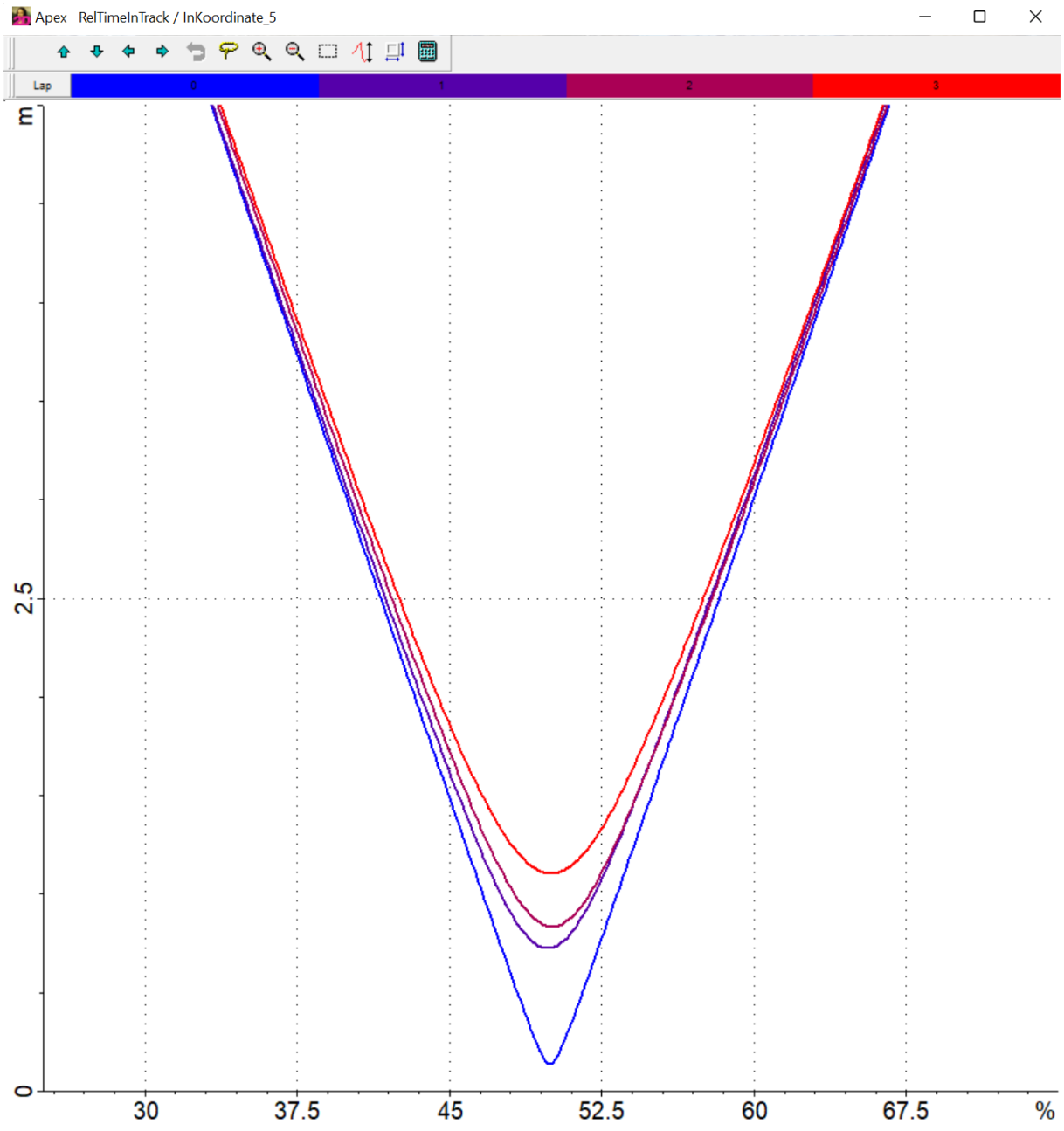
For example, the **FullThrottle-Time per GPSTracks-Section** can be calculated:



This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

5.4 Example 4 – GPSTracks Distance to Apex

With GPSTracks toolchain it is not only possible to separate a racetrack in different sections but also to calculate the distance to apex (black peak channel) in every corner of every lap.

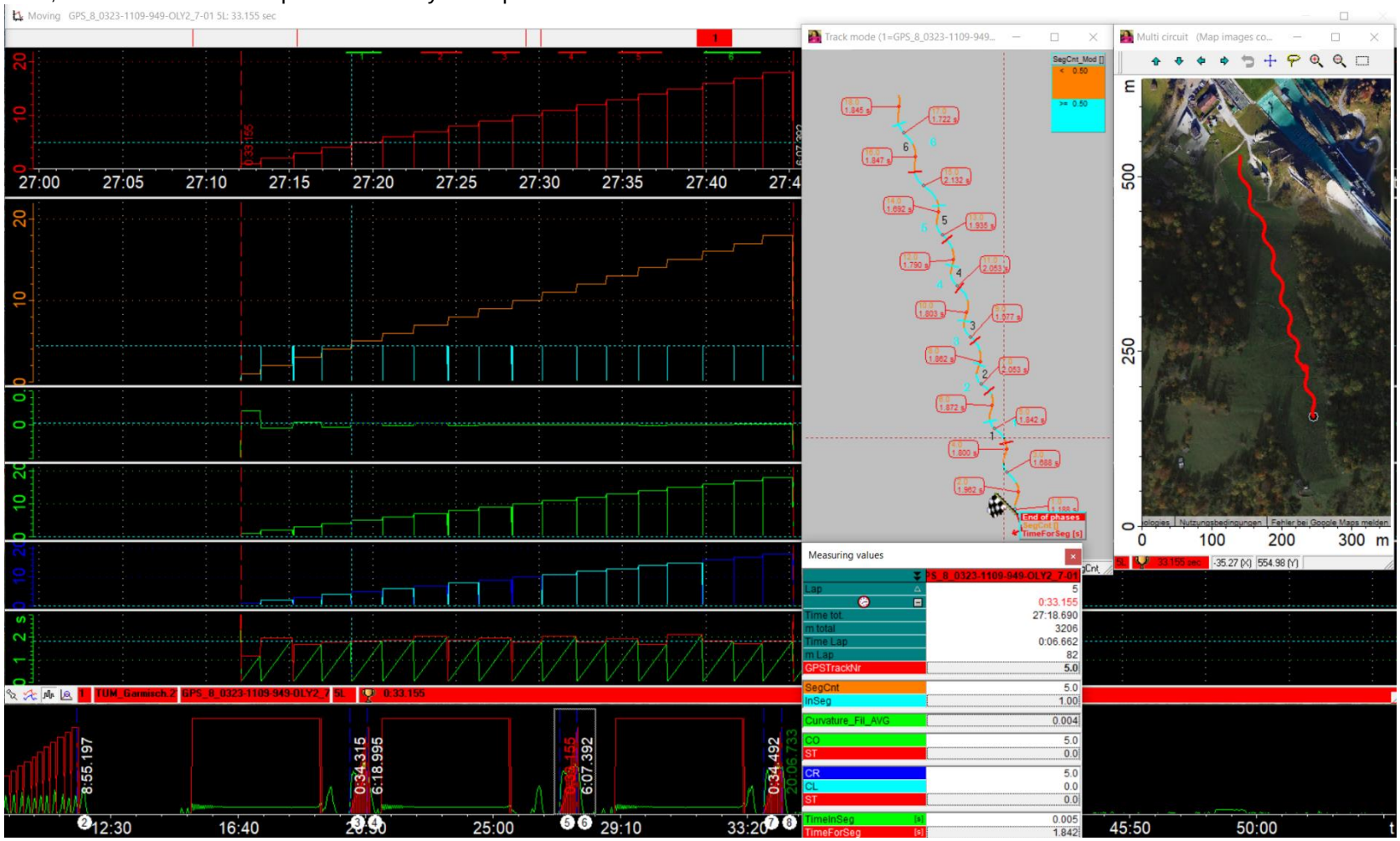


In the shown XY-Plot the distance to apex is shown for all four laps of one measurement at the same corner 5. It can be clearly seen that at first lap (blue) the distance to apex was the lowest and also the characteristic how the apex was approach was different to the other laps.

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

5.5 Example 5 – Slalom Skiing

The GPSTracks toolchain is not only suitable for racing, open road or automotive test but also for any other distance-to-task like Slalom Skiing, where the GPS-positions of the slalom poles were documented in 2D software by a manual trigger and used as trigger coordinates for GPSTracks toolchain. Thereby, the time between the slalom doors, as well as distance to poles and many other parameters could be calculated.



This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

5.5.1 Reverse positioning

At this project it was also necessary that the poles of the slalom track could be reproducibly set up again on another day.

The poles GPS-coordinates were used in combination with the 2D system and a 2D Dashboard to find the set the poles on exactly the same positions at slalom track.

5.6 Example 6 - Office2D

Since the installation of Race2020-version, a demo package is installed automatically which also contains an example to the 2D_GPS_Tracks-Cal-files.

The demo package consists, among others, of a Office2D measurement with predefined calculation files with associated templates.

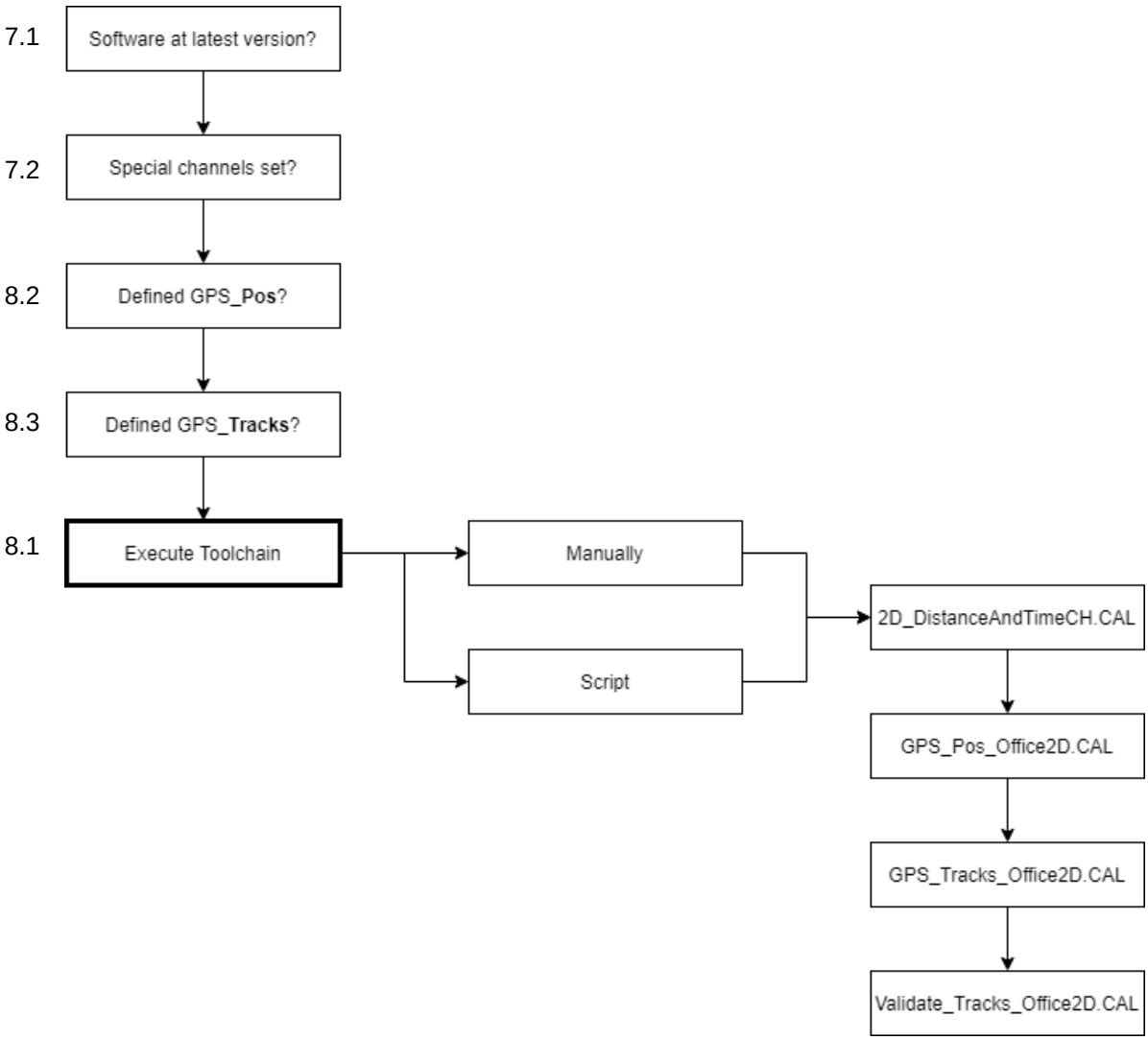
It can be accesses via:

WinARace → Button *Select an existing Event* → DemoData → CalcToolDemo → Office2D.MES.

Please open the Office2D measurement and have a look at the predefined template *GPS Tracks*.

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

6 Checklist



- If the toolchain was prepared for a use case (most of the times an event), only the last step *Execute Toolchain* must be carried out!
- All chapters after this chapter are serving as additional information.

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

7 Preparations

7.1 Update software

The *2D_GPSTracks* calculation files can only be used for Race2020 versions and newer.

Before using *2D_GPSTracks* calculation files, please ensure, that your software is updated to the latest version.

- *WinARace → Help → Search for software updates*

If the *2D_GPSTracks* is to be applied at measurements downloaded before **December 2020**, please again execute *AutoCalc-Configurator* with activated *2D_AutoGPS.CCF* for the respective measurements again.

- *WinARace → Modules → AutoCalc-Configurator*
- If multiple measurements must be recalculated, click on  in *WinARace Analysis* area and select all desired measurements → right click → *Run cal file → Run coded cal file (System\Cal folder)* → select *2D_AutoGPS.CCF* from list

Jump to checklist

7.2 Set special channels

In the Analyzer, special channels can be defined. The advantage is that flexibility is increased when working with measurements where different channel names have been assigned to channels with the same purpose. A calculation file with fixed channel names could only be applied to certain measurements! With the special channels this problem is solved because the respective special channels inside the calculation files are replaced by the channels linked in the adjoining list by the *Preprocessor*.

- Special channels: *Analyzer → Settings → Special channels*
- Also, the short-cut [SHIFT]+[s] can be used to open special channels list
- **The assignment of special channels must only be done once for each event!**

- For more information about *Special channels* please visit the download area of our website and have a look at the *CalcTool* manual.
  https://2d-datarecording.com/en/downloads/manuals/

Jump to checklist

7.2.1 2D_DistanceAndTimeCH

For the correct calculation of the distance, a speed channel, defined by the selection of special channel *Int_Speed*, is integrated. If a front or rear wheel speed detection is used, please select the respective channels at *Int_Speed* special channel. If only a GPS module is used, the GPS-speed channel *#V_GPS* should be used at *Int_Speed*. The more reliable channel should be selected at *Int_Speed*, when front or rear wheel speed detection and a GPS-module is used.

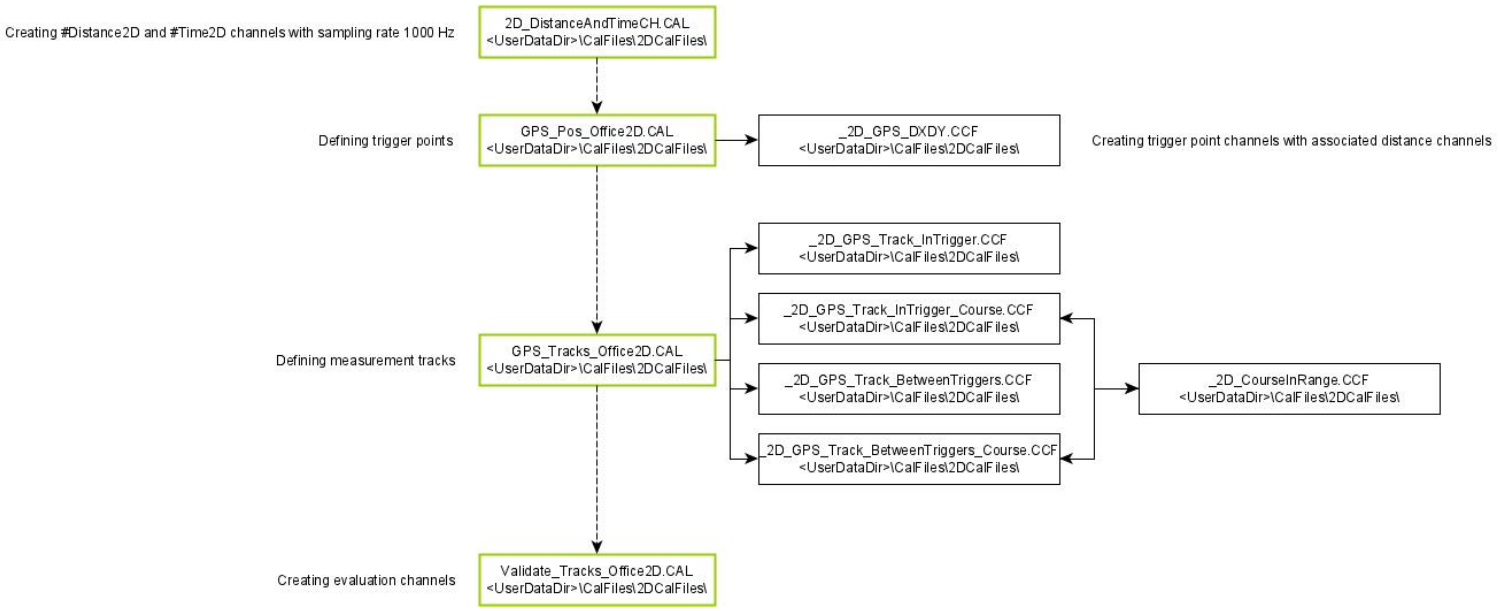
7.2.2 2D_GPSTracks

For the correct function of the GPS-trigger positions, the special channels *GPS_Latitude*, *GPS_Longitude* and *GPS_Altitude* must be set, before executing the *GPSTracks* toolchain.

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

8 CAL-Files

8.1 Structure



- For the correct function of the 2D_GPSTracks-CAL-files the shown calculation files must be executed in the shown order!
- Ensure, that 2D_DistanceAndTimeCH will be executed before GPS_Pos_Office2D-File to provide any channels needed!
- CCF files can be executed but not edited by user!

- For more information about *crypted (.CCF) and encrypted (.CAL)* files please visit the download area of our website and have a look at the *CalcTool* manual.
  https://2d-datarecording.com/en/downloads/manuals/

The calculation files which are connected by stroke lines must be executed by the user, but *Scripts* could be used for automatic execution optionally.

- If only trigger points must be analysed, only 2D_DistanceAndTimeCH and GPS_Pos_Office2D must be executed!

- For more information about *Scripts* please visit the download area of our website and have a look at the *Exports & Scripts* manual.
  https://2d-datarecording.com/en/downloads/manuals/

Jump to checklist

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

8.2 GPS_Pos_Office2D.CAL

As mentioned before, the raw *Latitude* and *Longitude* values are converted to **individual channels** for describing trigger points and measurement tracks and can therefore be used in the *Analyzer* for evaluation purposes.

Since it is usually more useful for evaluation purposes to analyse a certain area around a trigger point instead of just one point, a radius around this trigger point must also be specified in the *2D_GPSTracks*-CALs when defining each trigger point. Within this radius, the course of the signal can then be analysed, from which much more valuable developmental information can be obtained than by simply analysing a point. This radius can be freely selected by the user.

For the subsequent evaluation in *Analyzer*, it is necessary to get more information about the created trigger channels, e.g. distance from current position to trigger point centre, to be able to create plots or tables which are describing trends and courses of the desired developmental purposes.

Jump to checklist

8.2.1 [Variables]

In group [Variables] the trigger positions are defined by GPS-*Latitude* and -*Longitude* values as can be seen on the figure on the right side.

Delta_N defines the radius around trigger point *KN_Lat* and *KN_Lon*.

```
[Variables]
K1_Lat     = 49.3025125
K1_Lon     = 8.4539126
Delta_1    = 1.5

K2_Lat     = 49.3024250
K2_Lon     = 8.4539483
Delta_2    = 1.5

K3_Lat     = 49.3022951
K3_Lon     = 8.4540026
Delta_3    = 1.5
```

- A description of how these values can best be determined can be found in the Appendix 12.5.

8.2.2 [Find GPS Koordinates matching]

```
{$I <UserDataDir>\Calfiles\2DCalfiles\_2D_GPS_DXDY, P( @K1_Lat,@K1_lon ,1, @Delta_1)}  ; Create Trigger point 1
{$I <UserDataDir>\Calfiles\2DCalfiles\_2D_GPS_DXDY, P( @K2_Lat,@K2_lon ,2, @Delta_2)}  ; Create Trigger point 2
{$I <UserDataDir>\Calfiles\2DCalfiles\_2D_GPS_DXDY, P( @K3_Lat,@K3_lon ,3, @Delta_3)}  ; Create Trigger point 3
```

For each trigger point defined in [Variables], there must be an include call in the [Find GPS Coordinates matching] group to create the channels for each trigger point.

The number at the include call sets the number of the trigger point.

- In case a new trigger point is added, a new include call must be created too.

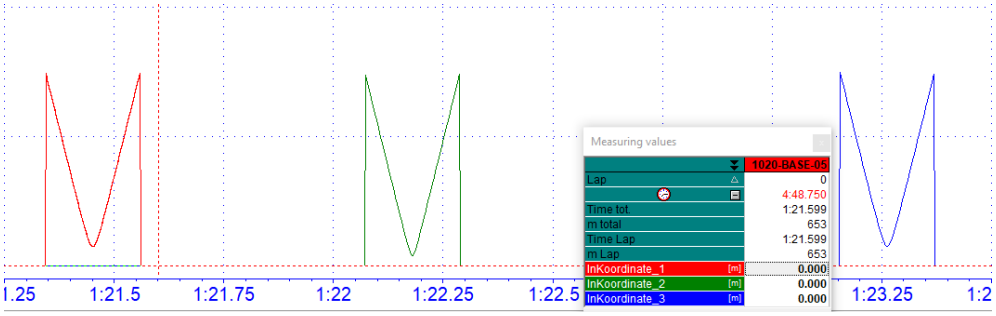
- Trigger points should never overlap each other to ensure the function of the subsequent calculation file *Validate_Tracks (see chapter 11)*!

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

8.2.3 Created channels

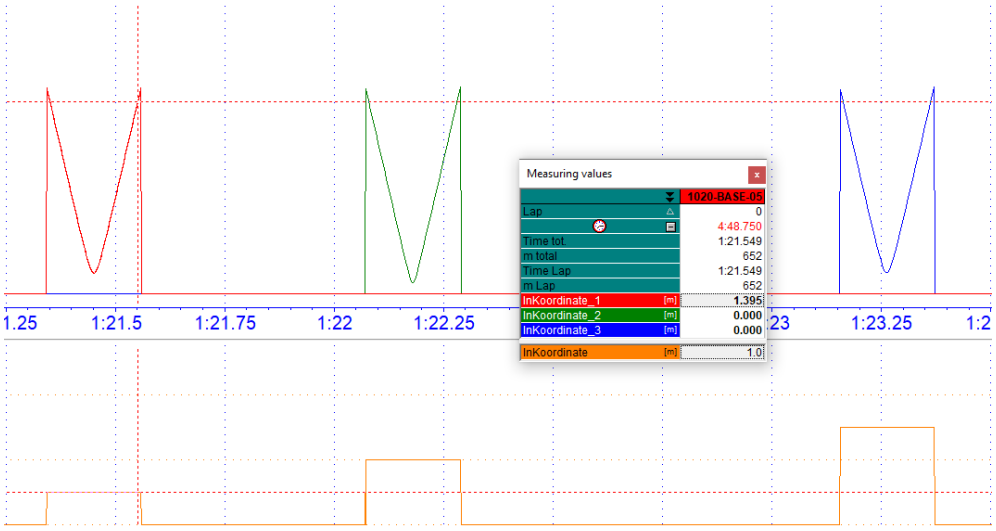
8.2.3.1 InKoordinate_N

The channels InKoordinate_*N* are marking the respectively defined trigger points and the unit is meters ([m]) because the channels contain the distance to the trigger centre as a combined information.



8.2.3.2 InKoordinate

This channel combines all InKoordinate_*N* into a single channel.

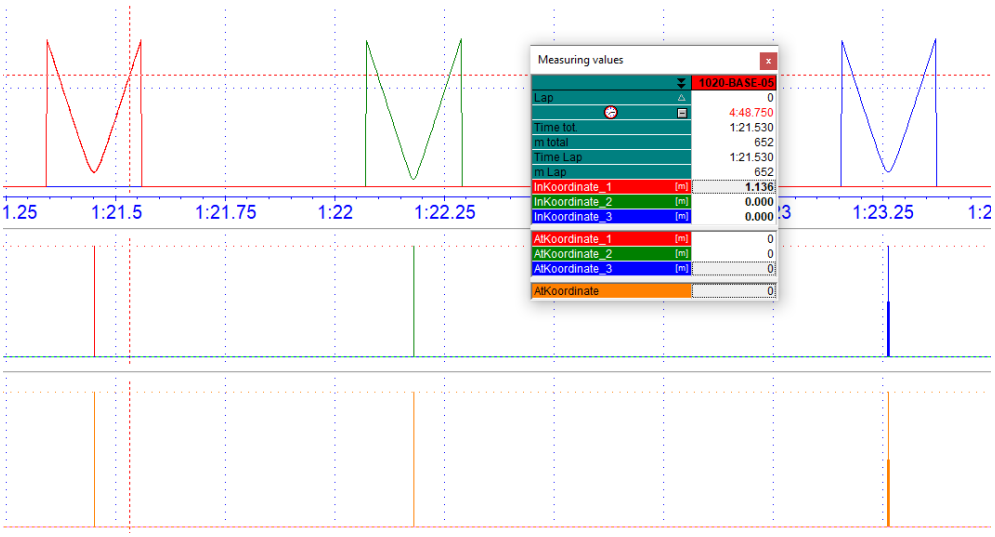


- The summary of all *InKoordinate_N* in one *InKoordinate* channel is done because of the simplified, subsequent evaluation of these *InKoordinate_N*

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

8.2.3.3 AtKoordiante_N

The channels AtKoordiante_*N* are marking the point with the minimal distance to the trigger point centre with a Boolean “1”.



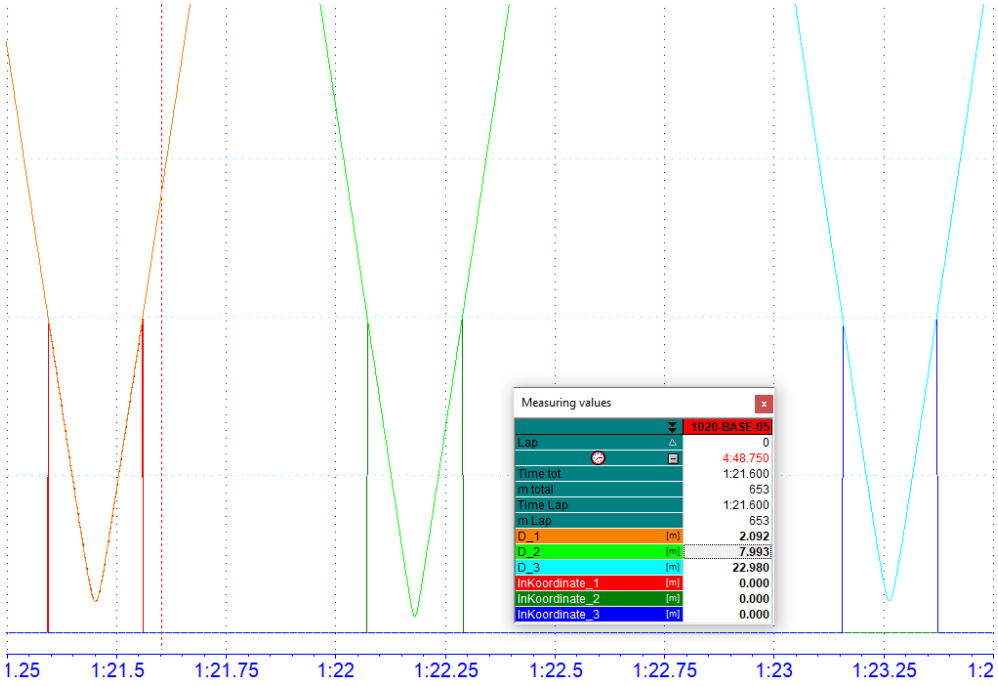
8.2.3.4 AtKoordinate

The channel AtKoordinates_*N* is marking all points with the minimal distance to the centres of respectively defined trigger points with a Boolean “1”.

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

8.2.3.5 D_N

Channel *D_N* is calculated from respective x and y distances to trigger point centre, which are also still available as channels *Dx_N* and *Dy_N*, and shows the total distance to the respective trigger point centres.



- Distance to respective trigger point centre can be evaluated **at every time** in measurement!
- The combined information with distance to trigger centre of InKoordinate_*N* can be seen here

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

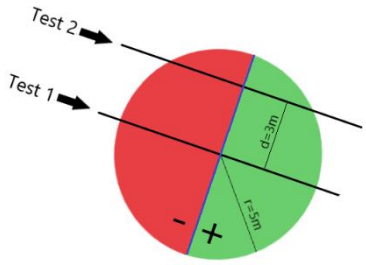
8.2.3.6 DistanceFromKoordinate

In comparison to the single *InKoordinate* channels, *DistanceFromKoordinate* provides the **distance to the centre of the currently active InKoordinate_N** combined with a plus/minus sign to identify if the current position is in the half before or after the trigger point centre.

- This channel combines information of distance of current position to the centre of currently active *InKoordinate_N* in one channel!

It is a huge difference, how a trigger point is crossed, because the distance travelled in the radius depends on the distance to the trigger centre when crossing the trigger point.

If the distance to trigger point centre gets bigger, the distance travelled through trigger point gets smaller.
In the figure shown example would mean, that at Test 2 the travelled distance in trigger point would only be 6m instead of 10m and no correct comparison could be executed.

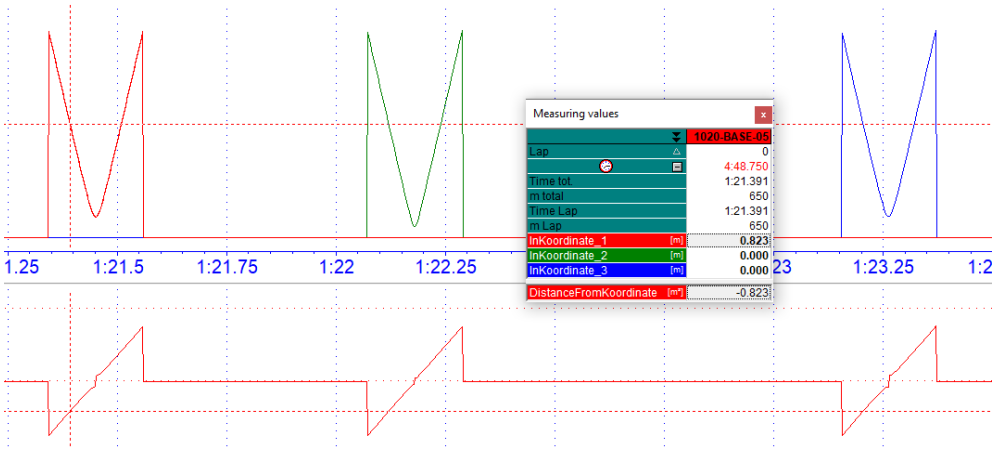


Due to the fact that no practical test can be repeated exactly the same way each time, so that, for example, each time exactly the centre of the trigger point is passed through, a channel which contains information about **the currently travelled distance** (starting at 0) **in trigger point** cannot be used for reasons of comparability.
Thus, a different parameter must be considered for the comparative analysis of the trigger points.

Thereby, channel *DistanceFromKoordinate*, gets especially important, because this channel displays the distance of current position **to the trigger point centre** with plus/minus sign of each trigger point (*InKoordinate_N*).

- A negative value (red area) of *DistanceFromKoordinate* means, the currently active position is located in the first half of the trigger position (seen from direction of heading).

By creating a channel which switches its sign from negative to positive at entering the second half of the trigger position, a comparative condition is created, because the minimal of the distance to trigger point center (*At_Koordinate*) always marks the transition of the center line (blue line).



- *DistanceFromKoordinate* can be used in *Analyzer* tools like *MinMax tables* or *XY-plots* to compare trigger points
- The summary of all distances from Koordinates in one *DistanceFromKoordinate* channel is done because of the simplified, subsequent evaluation

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

8.3 GPS_Tracks_Office2D.CAL

As mentioned before, the raw *Latitude* and *Longitude* values are converted to **individual channels** for describing trigger points and measurement tracks and can therefore be used in the *Analyzer* for evaluation purposes. Further, the *GPS_Tracks_Office2D* calculation files using the triggers in four different ways to create measurement tracks.

The four different possibilities will be described more precisely in the following chapter 8.3.1.

Since the evaluation of a measurement track is much more comprehensive, there is a specially designed calculation file *Validate_Tracks* is provided by 2D Datarecording. This calculation file will be described later in chapter 10.

Jump to checklist

8.3.1 [Call_functions]

In this group [Call_functions] the trigger points are combined to measurement tracks.

Since the use of *2D_GPSTracks* calculation files has a very wide range of applications, four different patterns of how tracks can be created were defined by 2D.

- Because often, differentiating tracks could be done by using the GPS-channel Course, 2D already provides a *_2D_CourseInRange* calculation file which is already used in *…_2D_GPS_Track_InTrigger_Course* and also in *…_2D_GPS_Track_BetweenTriggers_Course*
- Also, own track creation conditions can be created (chapter 12.1)!

Which of the four patterns must be used depends on the application and must be selected by the user himself via the following calculation file calls:

- *<UserDataDir>\CalFiles\2DCalFiles_2D_GPS_Track_InTrigger (8.3.1.1)*
- *<UserDataDir>\CalFiles\2DCalFiles_2D_GPS_Track_InTrigger_Course (8.3.1.2)*
- *<UserDataDir>\CalFiles\2DCalFiles_2D_GPS_Track_BetweenTriggers (8.3.1.3)*
- *<UserDataDir>\CalFiles\2DCalFiles_2D_GPS_Track_BetweenTriggers_Course (8.3.1.4)*

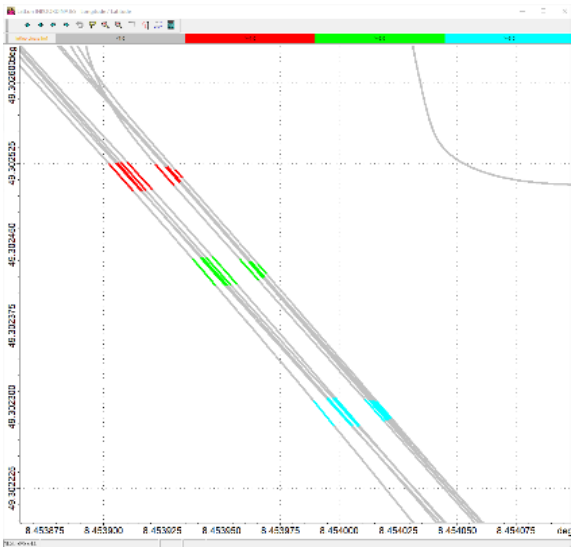
Often differentiating tracks could be successfully done by using the GPS-channel *Course*, *2D Datarecording* already provides the *…_Course* files, which are using the *_2D_CourseInRange* calculation file, which is described more in detail in chapter 10.

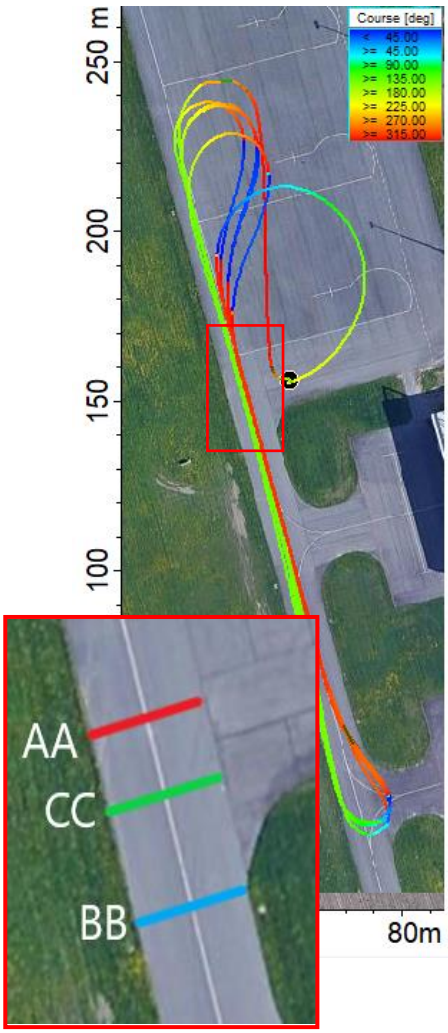
- Also, own track creation conditions can be created (chapter 12.1)!
- Using example 2 (chapter 5.2) to explain the four different track definitions.
- Measurement tracks should never overlap each other because of difficulties at subsequent evaluation (chapter 10).

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

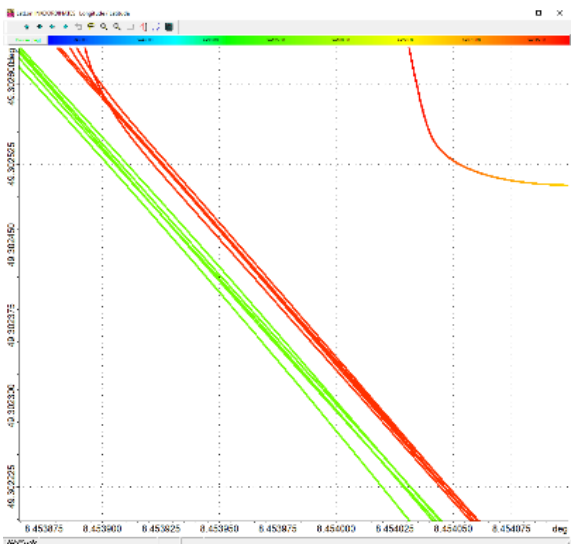
In a *XY-plot* the three defined *InKoordiante_N* are displayed over Latitude and Longitude by using Phase-condition Channel dependent (InKoordinate: 0.001 - 100) and the respective colors.

The representation of the *InKoordiante_N* in the *XY-plot* as ellipses comes from the fact that the channels *Longitude* and *Latitude* were used as X- and Y-axis assignments and these are related to a spherical earth model but being displayed an a cartesian coordinate system.





A colored differentiation with respect to *Course* can also be made. The adjacent figure shows the test runs with *Course* 165 degrees in green and the test runs with *Course* 350 degrees in red.



This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

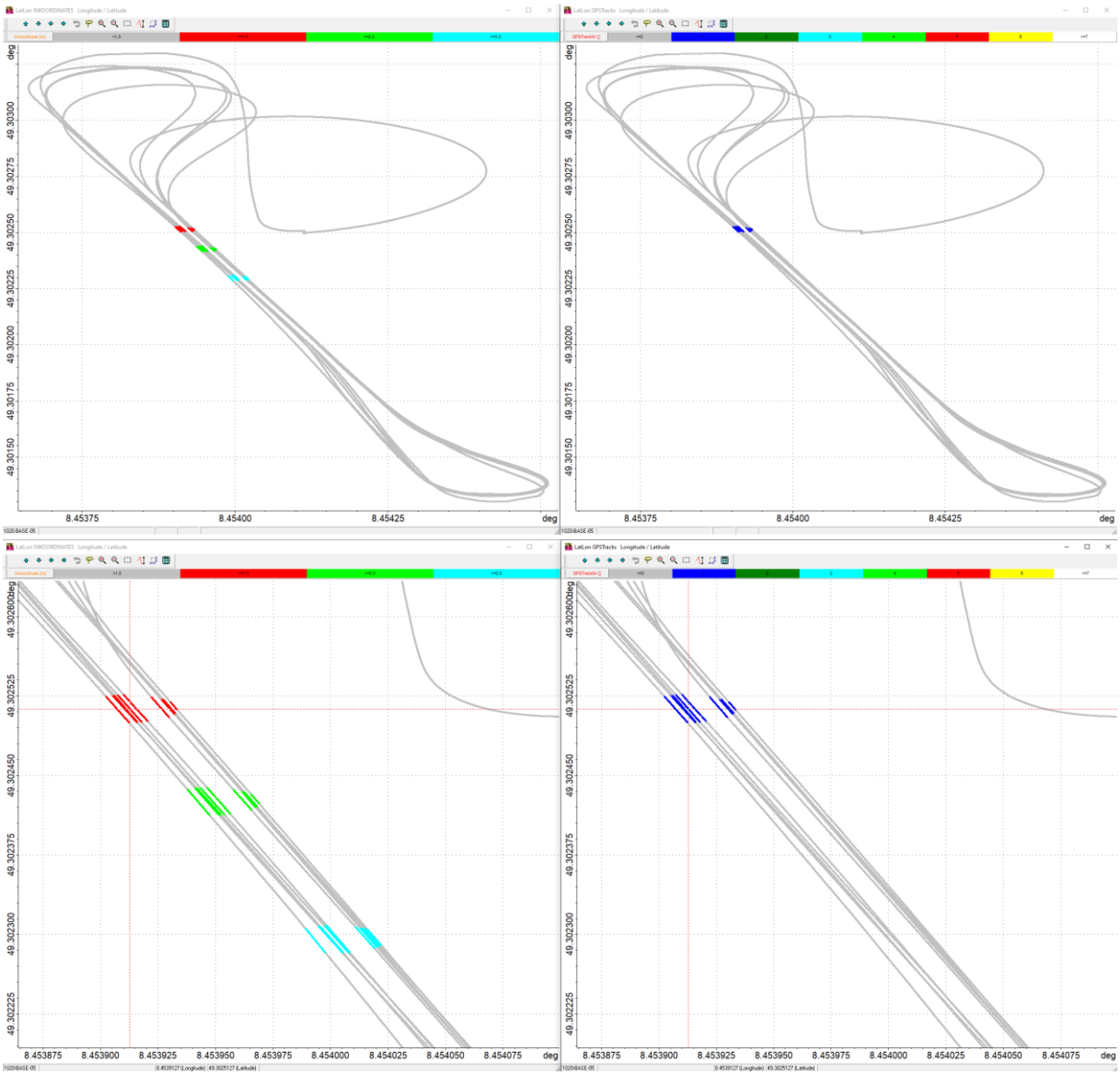
8.3.1.1 …_2D_GPS_Track_InTrigger

```
{$I <UserDataDir>\Calfiles\2DCalfiles\_2D_GPS_Track_InTrigger, P(InKoordinate_1, 1)}
```

	Parameters	Description		Chapter
P1	InKooridnate_1	Trigger	Defining which channel is used as Trigger channel	8.2.3.1
P2	1	GPSTrack_Nr	Defines GPSTrack_Nr the track is created as	8.3.2.1

When this CAL-file is called, a measurement track is created inside *InKoordinates* used.

The generation of this measurement track is **independent** on the GPS course at the entry point of the *InKoordinate* used.



All crossings of *InKoordinate_1* are recognised as GPS tracks (*GPSTrackNr*=1), **independently** from the direction the *InKoordinate_1* is entered from.

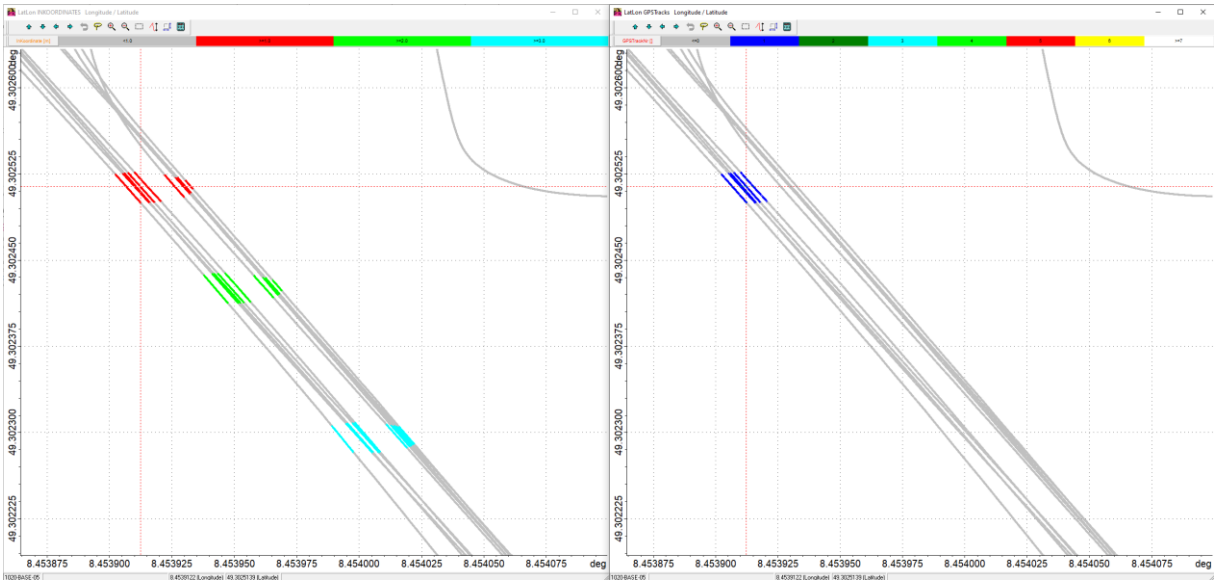
This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

8.3.1.2 …_2D_GPS_Track_InTrigger_Course

{$I <UserDataDir>\Calfiles\2DCalfiles_2D_GPS_Track_InTrigger_Course, P(InKoordinate_1, 160, 20, 1)}				
	Parameters	**Description**		**Chapter**
P1	InKooridnate_1	Trigger	Defining which channel is used as trigger channel	8.2.3.1
P2	160	Course at Trigger IN	Valid Course at Trigger IN	10
P3	20	Course-Range at Trigger IN	Valid Course Range (+/-) at Trigger IN	
P4	1	GPSTrack_Nr	Defines GPSTrack_Nr the track is created as	8.3.2.1

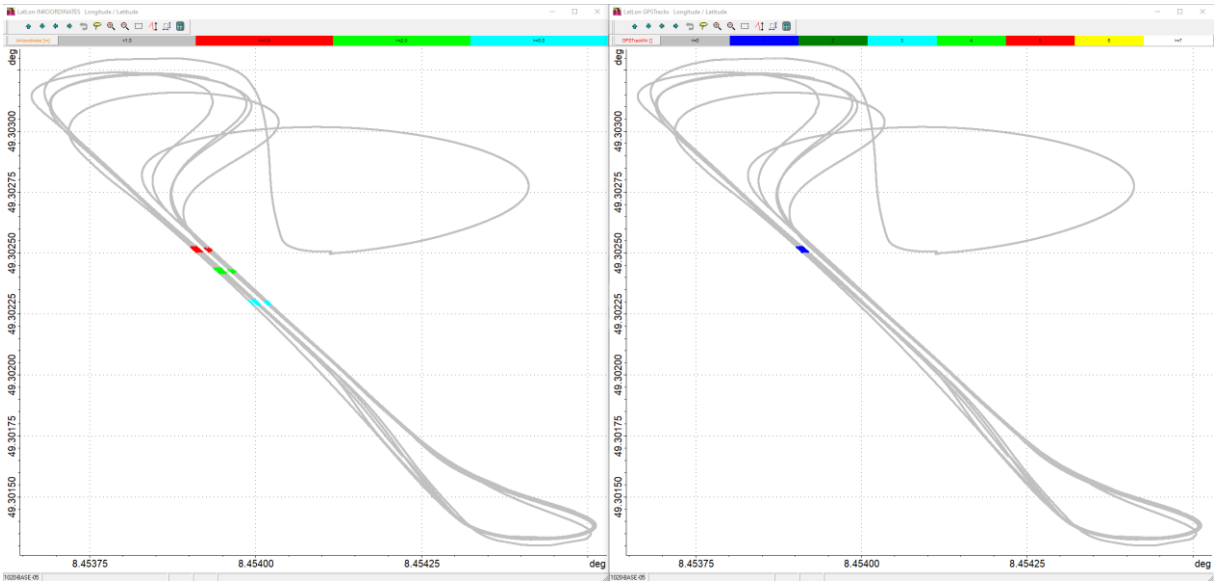
When this CAL-file is called, a measurement track is created inside *InKoordinate_N* used.

The generation of this measurement track is **dependent** on the GPS course at the entry point of the *InKoordinate_N* used only if the *InKoordinate* used are entered with the defined Course plus/minus specified Course range.



Only the crossings of *InKoordinate_1,* **headed from correct direction (#Course + Range)** (*Course* = 165 degrees), are recognised as GPS tracks (*GPSTrackNr*=1).

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.



This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

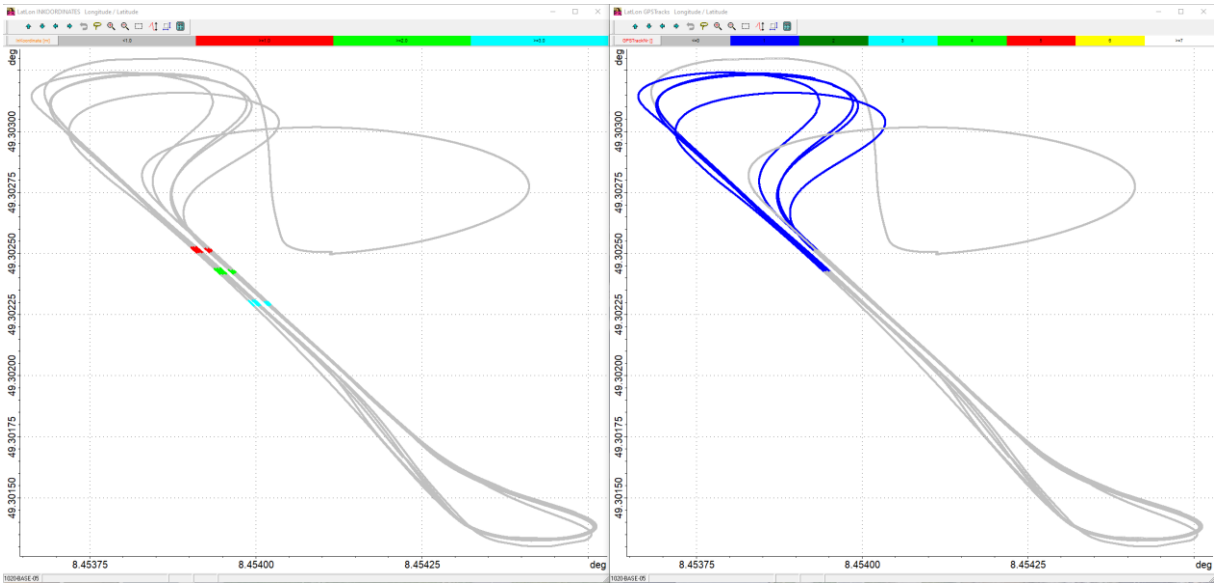
8.3.1.3 …_2D_GPS_Track_BetweenTriggers

```
{$I <UserDataDir>\Calfiles\2DCalfiles\_2D_GPS_Track_BetweenTriggers, P(AtKoordinate_1, AtKoordinate_2, 0.0, 0.0, 1)}
```

	Parameters	Description		Chapter
P1	AtKooridnate_1	Trigger IN	Defining which channel is used to create Trigger IN	8.2.3.3
P2	AtKooridnate_2	Trigger OUT	Defining which channel is used to create Trigger OUT	
P3	0.0	Trigger IN correction	Fine-adjust Trigger IN by meters	9
P4	0.0	Trigger OUT correction	Find-adjust Trigger IN by meters	
P5	1	GPSTrack_Nr	Defines GPSTrack_Nr the track is created as	8.3.2.1

When this CAL-file is called, a measurement track is created between the centres of two ***At****Koordinate_N* used.

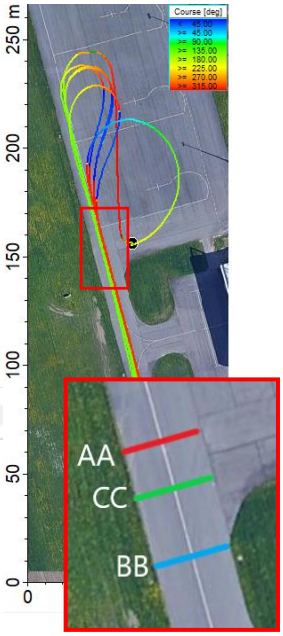
The generation of this measurement track is **independent** on the GPS course at the entry points of the two ***At****Koordinate_N* used.



When driving from *AtKoordinate_1* to *AtKoordinate_2* a GPSTrack (Course = 165 degrees) is recognised as a GPS tracks (*GPSTrackNr*=1), **independently** from the direction the *AtKoordinate_1* and *AtKoordinate_2* are entered from.

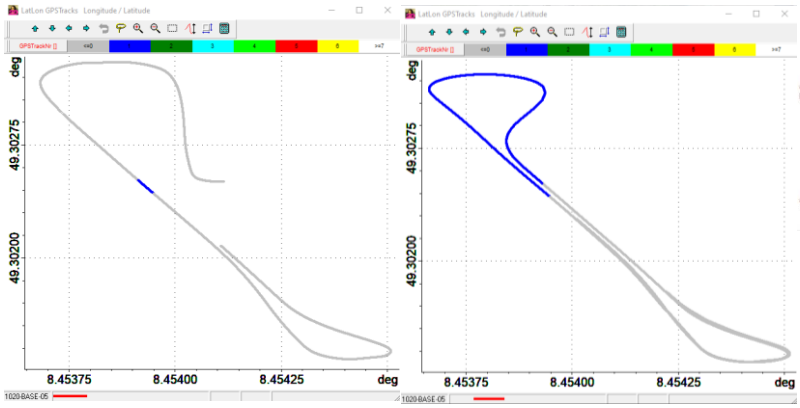
This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

The generation of the track with the *BetweenTriggers*-call is designed so that a track is only generated **when explicitly driving from *AtKoordinate_1* (AA) to *_2* (CC)**.



- It is not intended that a track with the same number from At*Koordinate_2* (CC) to *_1* (AA) is also created with the same track definition. In this case, a separate track would have to be defined.

It is Due to the test conditions in example 2 (chapter 5.2) a first, short track is generated from AA to CC while all following tracks are generated on the return journey when AA is driven over last and only after the turning loop CC is reached again.



8.3.1.4 …._2D_GPS_Track_BetweenTriggers_Course

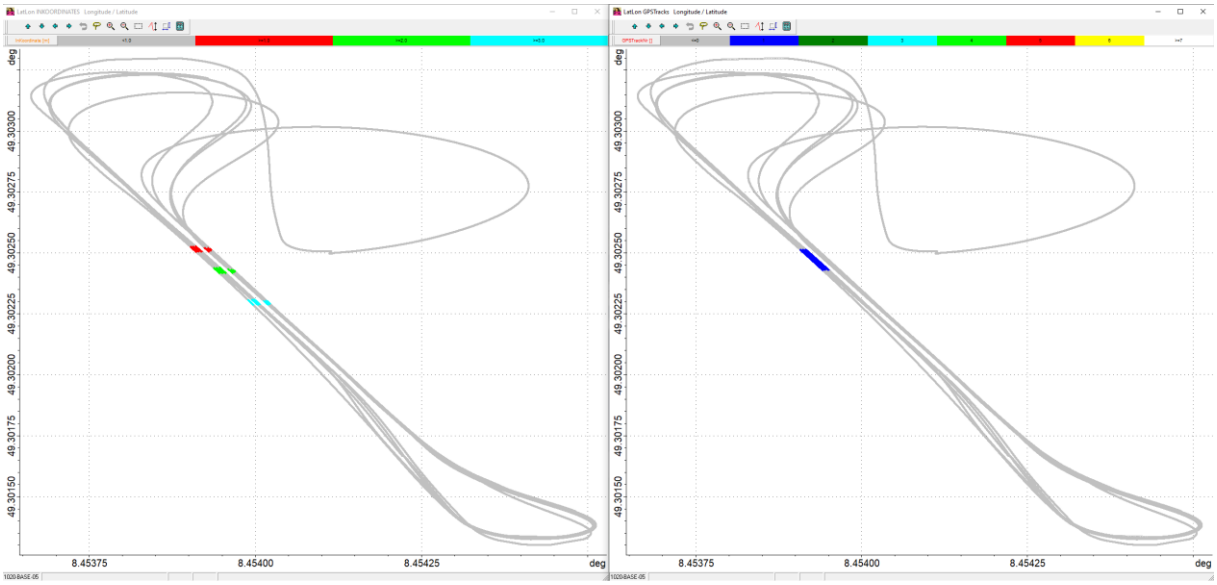
{$I <UserDataDir>\Calfiles\2DCalfiles_2D_GPS_Track_BetweenTriggers_Course, P(AtKoordinate_1, AtKoordinate_2, 0.0, 0.0, 160, 160, 1)}

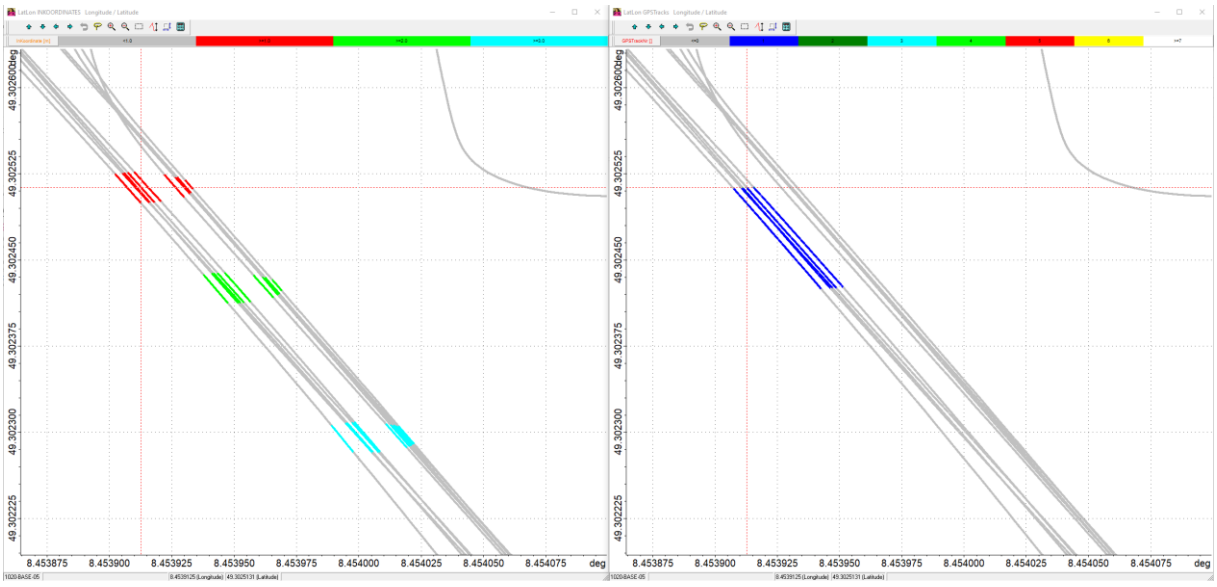
	Parameters	Description		Chapter
P1	AtKooridnate_1	Trigger IN	Defining which channel is used as Trigger IN channel	8.2.3.3
P2	AtKooridnate_2	Trigger OUT	Defining which channel is used as Trigger OUT channel	
P3	0.0	Trigger IN correction	Fine-adjust Trigger IN by meters	9
P4	0.0	Trigger OUT correction	Find-adjust Trigger IN by meters	
P5	160	Course at Trigger IN	Valid Course at Trigger IN (Course +/-20 degrees)	10
P6	160	Course at Trigger OUT	Valid Course at Trigger OUT (Course +/-20 degrees)	
P7	1	GPSTrack_Nr	Defines GPSTrack_Nr the track is created as	8.3.2.1

When this CAL-file is called, a measurement track is created between the centres of two *AtKoordinate_N* used only if the AtKoordinates used are entered with the defined Course.

The generation of this measurement track **depends** on the GPS course the two *AtKoordinate_N* used.

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.





Only when *AtKoordinate_1* and *AtKoordinate_2* are **headed from the correct direction** (*Course* = 165 degrees) it is recognised as a GPS track (*GPSTrackNr*=1).

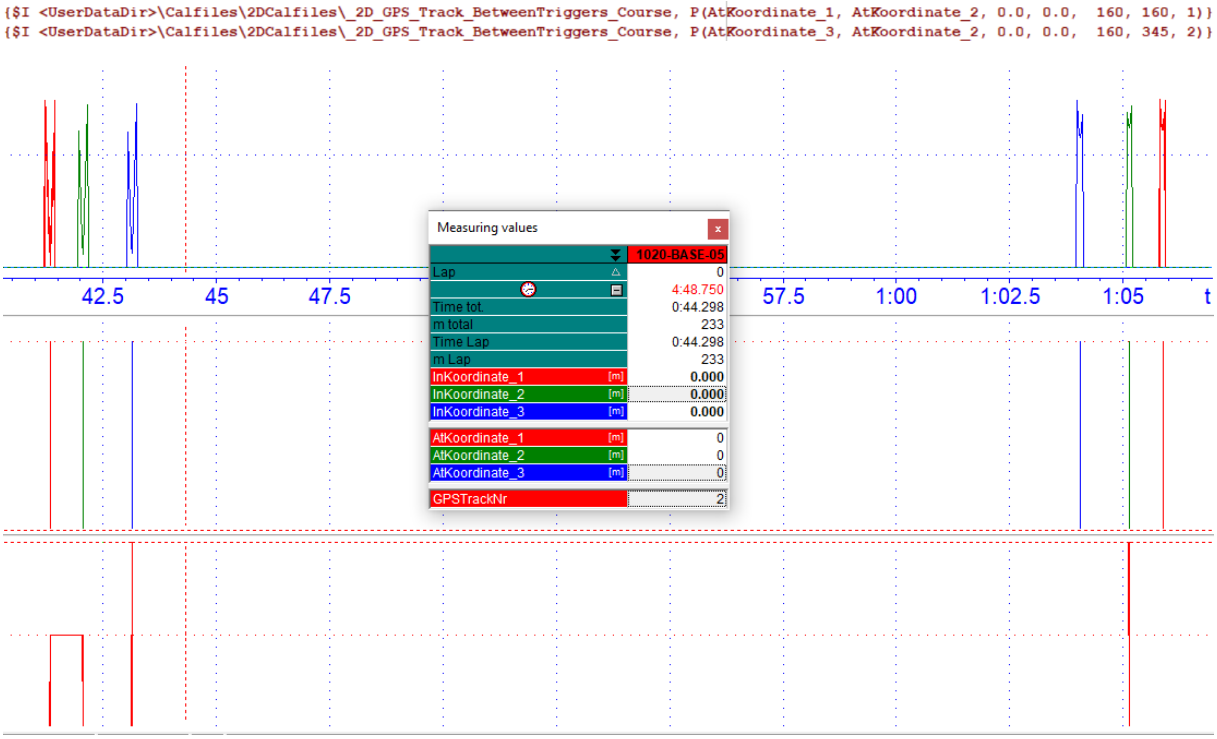
A range from +/-20 degrees is accepted when entering the *AtKoordinate_N*.

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

8.3.2 Created channels

8.3.2.1 GPSTrackNr

The channel *GPSTrackNr* is marking all respectively defined measurement tracks because all single measurement tracks are combined in this channel *GPSTrackNr*.



- The summary of all measurement tracks in only one *GPSTrackNr* channel is done because of the simplified, subsequent evaluation of these measurement tracks (chapter 10).

8.3.2.2 GPSTrackNr_BIT

This channel combines all GPSTrackNr_*N* to a single channel in a bit coded way, to indicate which *GPSTrackNr* is active if several are active at the same time.

- Measurement tracks should never overlap each other because of difficulties at subsequent evaluation (chapter 10).

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

8.4 Repositories

8.4.1 2D_DistanceAndTime

The uncrypted **2D_DistanceAndTimeCH** can be found at <*UserDataDir*>*CALFiles**2DCalFiles*\ and thus can also be added to *2D AutoCalc Configurator* for automatic execution at measurement download.

- For more information about *2D AutoCalc Configurator* please visit the download area of our website and have a look at the *CalcTool* manual.
  https://2d-datarecording.com/en/downloads/manuals/

- <UserDataDir>: Open *WinARace* → [CTRL] + [ALT] + [U]

8.4.2 GPS_Pos_Office2D & GPS_Tracks_Office2D

The CAL-files, which the user must edit to define trigger points and measurement tracks, are stored at <*UserDataDir*>*CALFiles**2DCalFiles*\, with the names **GPS_Pos_Office2D** and **GPS_Tracks_Office2D**.

- These two …Office_2D files should serve as examples of how the CAL files should be edited.

If trigger points and measurement tracks are created for a certain measurement, the user should copy the CAL files **GPS_Pos_Office2D** and **GPS_Tracks_Office2D** into the respective event folder and replace the addition *_Office2D* with a name of his choice.

- The CAL files in the event folder should then contain the GPS coordinates of the users trigger points and the measurement track definition

8.4.3 _2D_GPS_Track_... & _2D_CourseInRange

The four calculation files (see chapter 8.3), which are called from the *GPS_Tracks_Office2D* calculation file are stored at <*UserDataDir*>*CALFiles**2DCalFiles*\ and must not be edited by the user. The …_Course files serve as examples for TrackCreationConditions (see chapter 10 and 12.1).

- These calculation files must not be edited by user and are therefore created as crypted calculation files (.CCF).
- *_2D_CourseInRange* is called from _Course file for TrackCreationConditions (12.2)

8.4.4 ValidateTracks_Office2D

The CAL-file, which the user must edit to creating evaluation channels for, is stored at <*UserDataDir*>*CALFiles**2DCalFiles*\, with the names **ValidateTracks_Office2D**. For more information, please see chapter 11.

- These …Office_2D file should serve as examples of how the CAL files should be edited.

If evaluation channels must be created for a certain measurement, the user should copy the CAL file **ValidateTracks_Office2D** into the respective event folder and replace the addition *_Office2D* with a name of his choice and edit the respective calculation file in event folder.

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

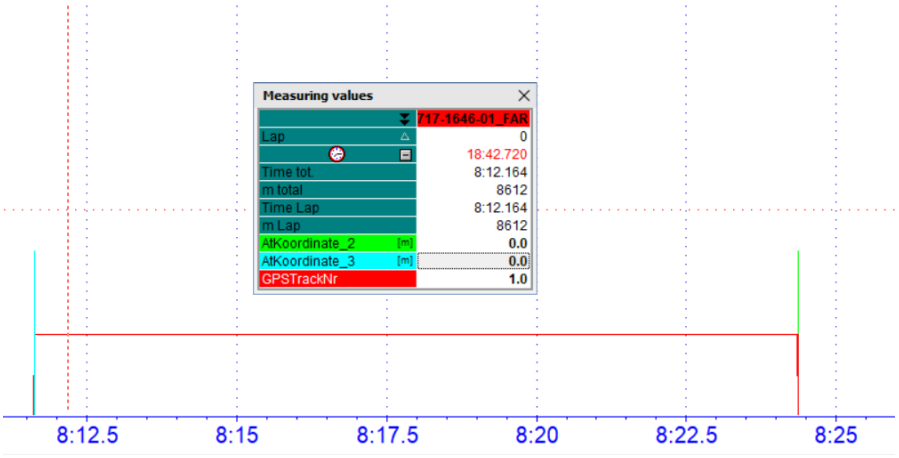
9 Trigger correction by meters

With this functionality, it is possible to fine-adjust the positions of the tracks created with two triggers, so that the track edges can be moved forwards and backwards by meters.

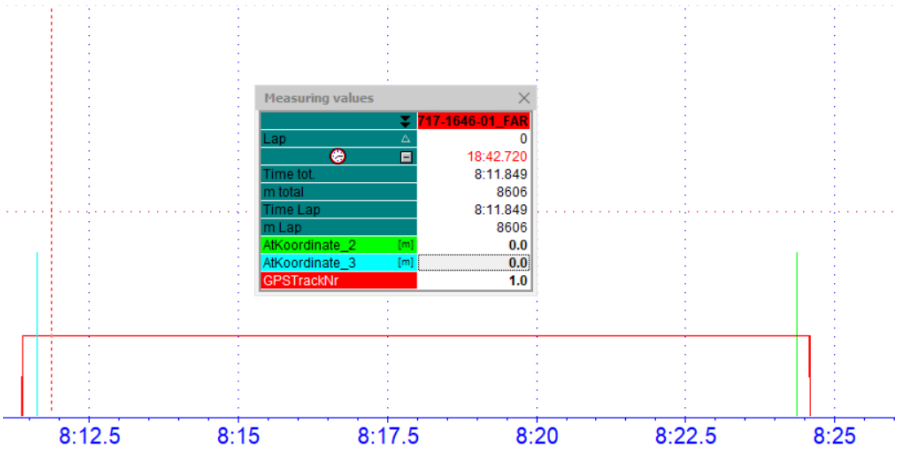
- The respective trigger coordinates (e.g. *AtKoordinate_4*) are not moved, but only the edges of the respective track!
- This function is only possible with tracks that are created via two trigger coordinates.

In the first example, the edges of *GPSTrackNr* are not shifted, whereby in the second example, the rising edge is shifted by 5 meters to the LEFT and the falling edge is shifted 5 meters to the RIGHT. Therefore, the *GPSTrackNr* is extended by 10 meters.

```
{$I <UserData>\CALFiles\2D_GPSTracks\_2D_GPS_Track_BetweenTriggers, P(AtKoordinate_3, AtKoordinate_2, 0, 0, 1)}
```




```
{$I <UserData>\CALFiles\2D_GPSTracks\_2D_GPS_Track_BetweenTriggers, P(AtKoordinate_3, AtKoordinate_2, -5, 5, 1)}
```



This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

10 _CourseInRange.CAL

Often differentiating tracks could be successfully done by using the GPS-channel *Course*, *2D Datarecording* already provides the *_2D_GPS_Track_InTrigger_Course* and *_2D_GPS_Track_BetweenTriggers_Course* calculation files (see chapter 8.3), which are using the *_CourseInRange* calculation file, which is described more in detail here.

```
{$I _2D_CourseInRange, P(125, 20)}
```

In this case, it is checked, if *#Course* has a value between 105° and 145° (125 ° +/- 20°). If the value of *#Course* is between 105° and 145°, the output-channel *#CourseInRange* is 1.

If *#Course* is not inside the specified range, output-channel *#CourseInRange* is 0.

In the following figure you can see the content of the file *_2D_GPS_Track_BetweenTrigger_Course*. In principle it is based on the file *_2D_GPS_Track_BetweenTrigger,* with the difference that the input channels are used in the *_2D_CourseInRange* file before calling the normal *_2D_GPS_Tracks_BetweenTrigger* file with the modified trigger positions *#Trigger1_CourseOK* and *#Trigger2_CourseOK*.

```
[Direction ok_Input]
{$I _2D_CourseInRange, P(@P5, 20)}
Trigger1_CourseOK = ExpandWhileTrue(#Course_InRange, #@P1)
Delete(#Trigger1_CourseOK)

[Direction ok_Output]
{$I _2D_CourseInRange, P(@P6, 20)}
Trigger2_CourseOK = ExpandWhileTrue(#Course_InRange, #@P2)
Delete(#Trigger2_CourseOK)

;=====================================================

[CALL_2020_GPS_Track_Between]
{$I _2D_GPS_Track_BetweenTriggers, P(Trigger1_CourseOK, Trigger2_CourseOK,@P3,@P4, @P7)}
```

- Where the calculation files can be found is described in chapter 8.4.
- Also, own track creation conditions can be created (chapter 12.1)!

11 Validate_Tracks.CAL

While for trigger points the evaluation channels (*DistanceFromKoordinate* → chapter 8.2.3) are, partly, already created when executing *GPS_Pos_Office2D.CAL*, creating similar channels for measurement tracks is much more comprehensive and thus outsourced to a further 2D calculation file *Validate_Tracks_Office2D*.

For the subsequent evaluation in *Analyzer*, it is necessary to get more, comparative information about the GPSTrack channels, e.g. travelled distance from start of track or expired time since start of track, to be able to create plots or tables which are describing trends and courses of the desired developmental purposes.

- For more information about *ValidateTracks* please visit the download area of our website and have a look at the *ValidateTracks* manual.
  https://2d-datarecording.com/en/downloads/manuals/

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

12 Appendix

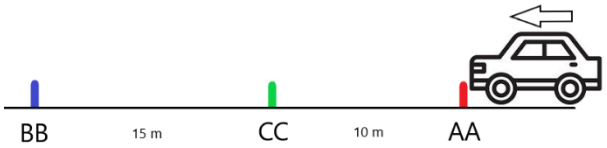
12.1 How to use own TrackCreationConditions

For most evaluation purposes it is sufficient to differentiating measurement tracks by using the *Course*-condition, that´s why 2D already provides the *…2020_GPS_Tracks_InTrigger_Course* (8.3.1.2) and *…2020_GPS_Tracks_Between_Course* (8.3.1.4) calculation files.

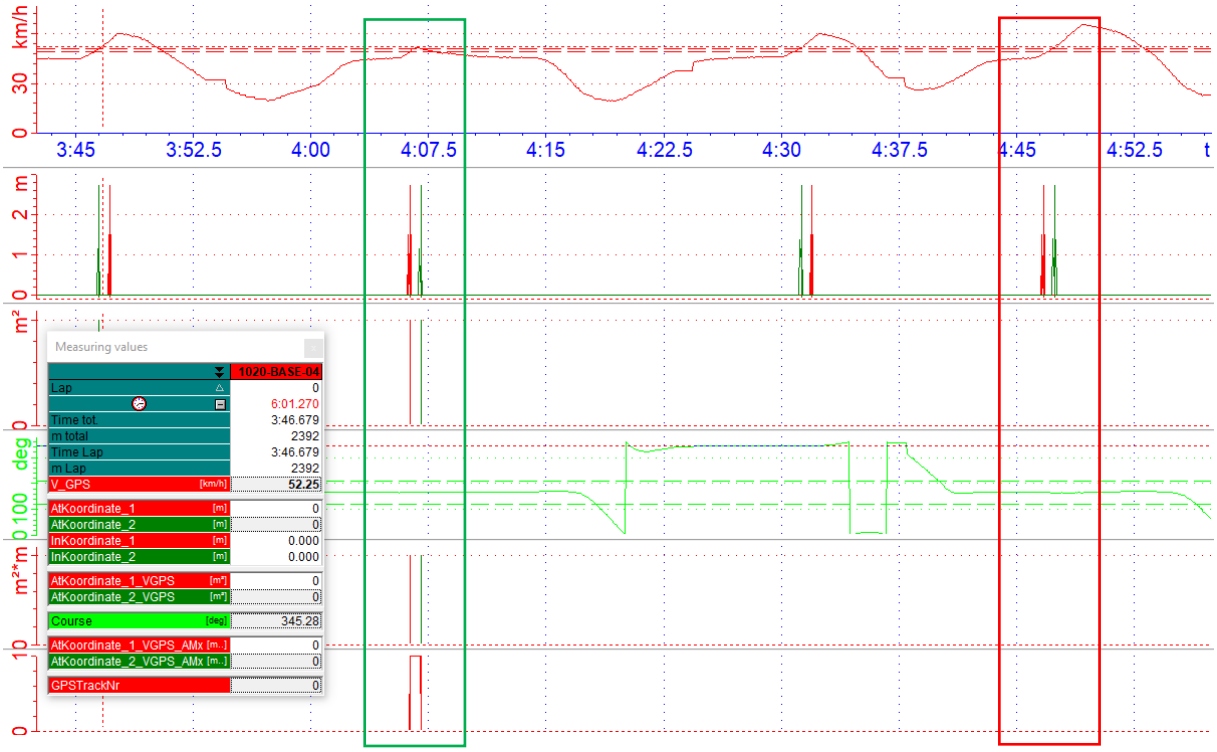
At some tests own track creation conditions must be considered too, e.g. if RPM or Speed is in a specified range at a trigger point or measurement track. The *2D_GPSTracks* calculation file provide the possibility that users can created their own track creation criteria and thus different test-driving patterns, and conditions can be specified and combined freely.

- Using example 2 (chapter 5.2) to explain how to create own trigger creation conditions
- Creating own track creation conditions should be taken place in an independent calculation file (e.g. *TrackCreationConditions.CAL* → 12.4)

In this case a track from AA to CC should only be created if the track from AA to CC is entered and left with the correct GPS-*Course* of about 165 degrees and the GPS-speed (*V_GPS*) at trigger point AA and CC is between 49 and 51 km/h. Also, it the distance of triggers AA and CC must be smaller than 12.5 m.



- Condition 1: GPS-Course at AA and CC about 165 degrees
- Condition 2: Speed at AA and CC between 49 and 51 km/h
- Condition 3: Distance between AA and CC < 12.5 meters because of multiple triggering at special test track conditions



This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

12.2 Testing Collision Avoidance Systems (static)

An application where the static trigger point definition of the *GPS_Pos_Office2D* calculation file can be used when testing the functionality of *Collision Avoidance Systems*. The created channels are extremely helpful to e.g., compare the CAN-signals of the *Collision Avoidance System* with the distance of the car to the obstacle.

Example:

A *Collision Avoidance Systems* of a car should be tested, whereby only the car to be tested must equipped with measurement equipment.

Also, the GPS-Position of the obstacle must be determined. The way to do this strongly depends on the conditions on the measurement track and can thus be done in various ways.

- Please see chapter *How to determine trigger point coordinates*
- Beside the GPS channels which are required to execute the 2D_GPSTracks toolchain, please ensure that all additional CAN-channels are to be recorded.

- The GPS-accuracy is very decisive for the quality of the results, so at this kind of test, 2D Realtime-Kinematic systems should be used
  Please see XXX or contact 2D-Datarecording for more information.

Using 2D_GPSTracks-toolchain:

To use the static trigger point functionality of 2D_GPSTracks, the trigger point definition of the obstacle must be done in *GPS_Pos_Office2D* as follows:

```
Kl_Lat     = 49.3025125
Kl_Lon     = 8.4539126
Delta 1    = 1.5
```

With the help of the channels that are generated when the *GPS_Pos_Office2D* calculation file is executed, the *Adaptive Cruise Control* can now be evaluated in the *Analyzer*.

#D_1	Contains the distance from Car 1 to Car 2 over the entire measurement period.
#InKoordiante_1	As soon as the distance from Car 1 to Car 2 is smaller than the entry at Delta_1 (15 m), this channel contains the distance to Car 2. If the distance is greater than Delta_1 (15 m), the channel value is 0.
#AtKoordinate_1	Marks the point with the closest distance from Car 1 to Car 2 if the distance is less than Delta_1 (15 m).

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

12.3 Dynamic Trigger Points

Besides defining static trigger points in *GPS_Pos_Office2D*, it is also possible to use channels as trigger coordinates for *2D_GPSTracks*-toolchain. Thereby, the channels, created by *GPS_Pos_Office2D* (see 8.2.3) can be used to e.g., determine the distance to another moving vehicle dynamically.

- **This is extremely helpful, when automotive functionalities like *Adaptive Cruise Control* or *Collision Avoidance Systems* are needed to be tested or evaluated.**

Defining static trigger point

```
Kl_Lat     = 49.3025125
Kl_Lon     = 8.4539126
Delta_l    = 1.5
```

Defining dynamic trigger point

```
Kl_Lat     = #Latitude_CAR2
Kl_Lon     = #Longitude_CAR2
Delta_l    = 15.0
```

Example:

An *Adaptive Cruise Control* functionality of a car should be tested, whereby the two cars must be used which must both be equipped with measurement equipment.

- The GPS-accuracy is very decisive for the quality of the results, so at this kind of test, 2D Realtime-Kinematic systems should be used
  Please see XXX or contact 2D-Datarecording for more information.

There are two different ways to set up an *Adaptive Cruise Control* test:

Option 1	Option 2
Equipping both cars with two individual measurement systems with GPS modules and combine measurements afterwards by using GPS-Time for synchronization	Equipping Car 1 with measurement system with GPS-module and radio telemetry receiver. Car 2 will be equipped with GPS-Module and radio telemetry transmitter which transmits GPS-data to receiver of Car 1.

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

Option 1: Merging measurements

Adaptive Cruise Control of car 1 should be tested, but both cars must be equipped with measurement systems which both including a GPS-module.

- Both systems can be expanded individually to e.g. also record CAN-channels of the respective car!

Both measurement systems record their GPS positions and all other desired CAN channels. After downloading both measurements, the individual measurements can be combined into a single measurement with the channels of both vehicles by means of 2D-Merger using the GPS times in order to then be able to apply the 2D_GPSTracks toolchain.

- In *2D-Merger and Appender* it can be selected, that at all channels of measurement 2, the extension *_CAR2* is added to differentiate the channels correctly. Please see respective manual in downloads area of our website.

Option 2: Radio telemetry

Adaptive Cruise Control of car 1 should be tested, but only car 1 must be equipped with a complete measurement system, but besides the GPS module, a radio telemetry receiver must be also added to the system of car 1.

Car 2 must be equipped with only an GPS module and a radio telemetry transmitter, which continuously sends the GPS-data from car 2 to car 1 where the respective channels are recorded and the measurement system of car 1.

Using 2D_GPSTracks-toolchain:

(*valid for option 1 and 2*)

To use the dynamic trigger point functionality of 2D_GPSTracks, the trigger point definition in *GPS_Pos_Office2D* must be done as follows:

```
Kl_Lat     = #Latitude_CAR2
Kl_Lon     = #Longitude_CAR2
Delta_1    = 15.0
```

With the help of the channels that are generated when the *GPS_Pos_Office2D* calculation file is executed, the *Adaptive Cruise Control* can now be evaluated in the *Analyzer*.

#D_1	Contains the distance from Car 1 to Car 2 over the entire measurement period.
#InKoordiante_1	As soon as the distance from Car 1 to Car 2 is smaller than the entry at Delta_1 (15 m), this channel contains the distance to Car 2. If the distance is greater than Delta_1 (15 m), the channel value is 0.
#AtKoordinate_1	Marks the point with the closest distance from Car 1 to Car 2 if the distance is less than Delta_1 (15 m).

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

12.4 TrackCreationConditions.CAL

```
[Variables]
V_GPS_MIN           = 49.0
V_GPS_MAX           = 51.0
MetersAfterTrigger1 = 12.5

[AtKoordinate_1_VGPS]
C1       = FreqNI(#V_GPS, #Atkoordinate_1.Rate)                    ; Upsample #V_GPS to Rate of #Atkoordiante_1

C2       = MinWhileTrue(#c1, #Atkoordinate_1)                      ; Detect minimum of #V_GPS in AA
C3       = MaxWhileTrue(#c1, #Atkoordinate_1)                      ; Detect maximum of #V_GPS in AA

c4       = If(#C2, >, @V_GPS_MIN,   1, 0)                          ; Check If minimum is bigger minima-treshold
c4       = If(#C3, <, @V_GPS_MAX, #C4, 0)                          ; Check If maximum is smaller maxima-treshold => AND-Condition

Result   =*(#Atkoordinate_1, #C4)                                  ; Create new InKoordinate_1_V_GPS If #V_GPS is inside limits

[AtKoordinate_2_VGPS]
C1       = FreqNI(#V_GPS, #Atkoordinate_2.Rate)                    ; Upsample #V_GPS to Rate of #Inkoordiante

C2       = MinWhileTrue(#c1, #Atkoordinate_2)                      ; Detect minimum of #V_GPS in CC
C3       = MaxWhileTrue(#c1, #Atkoordinate_2)                      ; Detect maximum of #V_GPS in CC

c4       = If(#C2, >, @V_GPS_MIN,   1, 0)                          ; Check If minimum is bigger minima-treshold
c4       = If(#C3, <, @V_GPS_MAX, #C4, 0)                          ; Check If maximum is smaller maxima-treshold => AND-Condition

Result   =*(#Atkoordinate_2, #C4)                                  ; Create new InKoordinate_2_V_GPS If #V_GPS is inside limits

[Trigger2ValidAfterMeterX]
C1                    = RisingEdge(#AtKoordinate_1_VGPS)
C1                    = FillFromBool(#Distance2d, #c1)
C1                    =-(#Distance2d, #C1)
c1                    = If(#C1, <, @MetersAfterTrigger1, 1, 0)     ; Check if 2nd trigger is valid is inside defined range in [m] (AfterMeterX--> (AMx)
C1                    = ExpandWhileTrue(#AtKoordinate_2_VGPS, #C1)
AtKoordinate_1_VGPS_AMx =*(#C1, #AtKoordinate_1)                   ; Create new InKoordinate_1, if Speed is Valid (*_VGPS*) and 2nd Trigger in Range (*_AMx)
AtKoordinate_2_VGPS_AMx =*(#C1, #AtKoordinate_2)                   ; Create new InKoordinate_2, if Speed is Valid (*_VGPS*) and 2nd Trigger in Range (*_AMx)
```

- This prepared *TrackCreationCondition* calculation file can be found at *<UserDataDir>\CalFiles\2DCalFiles* should serve as an example.
  <UserDataDir>: Open *WinARace* → [CTRL] + [ALT] + [U]

12.4.1 GPS_Tracks_...

```
{$I <UserDataDir>\Calfiles\2DCalfiles\_2D_GPS_Track_BetweenTriggers_Course, P(AtKoordinate_1_VGPS_AMx, AtKoordinate_2_VGPS_AMx, 0, 0,  165, 165, 1)}
```

- The created TrackCreationCondition-channels can be used in all four patterns of track creation patterns (8.3.1)

- TrackCreationConditions must be created very carefully and always depend on the purpose of current application!

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.

12.5 How to determine trigger point coordinates

At best, the trigger position coordinates can be determined via *Analyzer* from a measurement containing Latitude and Longitude values from test track.

- It is advantageous if measuring points on the test field match with markers visible from Google Maps, such as road markings, green spaces, trees or buildings
- It is useful to make a recorded test run before the test, with which the trigger positions can then be determined

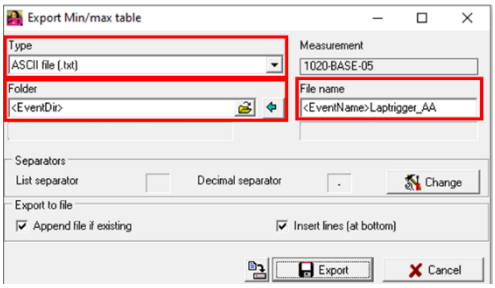
Please follow the subsequent steps if a measurement of test track is available:

1. Open measurement in *Analyzer* via *WinARace*
2. Open *MultiCircuit* from tab Functions or press <m> in Analyzer

   - *MultiCircuit* only shows Google Maps background augmentation if internet connection is available

3. Start *measurement mode* and try to determine your trigger points in *MultiCircuit*
4. If desired trigger point centre is found, set a marker by <left click>
5. <Right click> → *Add* → *Set laptrigger* and create a laptrigger
6. For other trigger point centres repeat 4 to 5
7. If all trigger point centres are marked with laptrigger open tab *Functions* → *MinMax* → *Create new* create a *MinMax*-table with name like trigger point, e.g. LatLon_Triggerpoints.
8. Select Latitude and Longitude channels and set as shown and select *All data* when confirming *MinMax*-settings with *OK.*

   Due to *LapEnd*-selection, a MinMax-table is created with information about the Latitude and Longitude of each laptriggers, and thus at the trigger points which are to defined in *GPS_Pos_Office2D.CAL* (see chapter 8.2).
9. <Right click> on *MinMax*-table and select *Export*. Set *Export* as shown and confirm with <*Export*>. Find created text file in event directory.



10. Open created text file and *GPS_Pos_Office2D.CAL* (see chapter 8.2) and copy the desired laptrigger coordinates to the calculation file to define the trigger points.

This document is subject to change at 2D decision. 2D assumes no responsibility for any claims or damages arising out of the use of this document, or from the use of modules based on this document, including but not limited to claims or damages based on infringement of patents, copyrights or other intellectual property rights.