Harry's GPS LapTimer

Documentation v1.2



Harry's Technologies

Points of Interest and Triggers

Introduction

Looking into the multifaceted functionality provided by LapTimer, the probably most important feature is the concept of position based triggers for automatic timing. This concept will be introduced and discussed in this section.

The more general concept used for triggers is that of Points of Interest. A Point of Interest (POI) is a place with a defined position on the globe, defined by a polar coordinate obtained typically by GPS or GLONASS. A POI is made up from a position, a description, some display formatting, and specific behavior. Behaviors are "information providing" (named Information), "active information push" (named Alert), and finally "timing actions" (named Triggers).

Depending on the POIs included, collections of Points of Interests are named POI Sets or Tracks Sets. Predefined POI Sets can be downloaded from LapTimer's database, they can be individually defined by a user, and they can be submitted for Certification - which makes them ultimately part of LapTimer's database.



Track Set Example

Reading this Chapter

Most users will never go deep into the concepts POIs and triggers are based on. They simply will load a track set from LapTimer's database and drive. There are certain situations however, it is good to know a little more about the individual types and behavior – particularly when things don't work as expected.

We start the chapter with an overview on <u>Overview on Types of POIs</u> and what they are used for. Besides simply displaying POIs on a map, Triggers and Alerts are detected by LapTimer when approaching them. This detection is a sophisticated process taking into account speeds, distance, directions, and POI type. We will have a deep look into this process in section <u>Detection Corridor</u>. The next section <u>Using Triggers for Automatic Timing</u> will discuss the different types of triggers available and how to use them for circuits, hill climbs etc. More a background information, but an answer to a frequently asked question "How is it possible to get sub second accuracy from a 1 Hz GPS" can be found in section Interpolation.

As versatile and powerful the mechanism of triggers is, the many errors can occur when "programming" your tracks - or using prepared tracks. Other than many other apps "working solely inside a computer", LapTimer needs to interact with the

outside world. The chapter <u>Tracks</u> – following this chapter – will utilize triggers a lot. So in case you are interested in tracks, their definition, and trouble shooting them, please spend the time to fully understanding POIs and triggers.

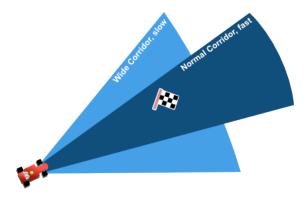
Overview on Types of POIs

There are three basic types of POIs available:

- Information POIs: this type of POI does not show any behavior, it is simply some information for a specific location. LapTimer uses Information POIs (mostly simply named POI, not Information POI) to display this information for a specific location. Examples are turns, straights, or other interesting information on, and around the track. POIs appear as simply text on maps, and they are inserted into video overlays when driving towards them. Good track sets come with lots of POIs to generate really nice overlaid videos.
- Alert POIs: this type of POI comes with some information text just like the Information POIs. Different to Information POIs, Alert POIs (mostly named Alerts) bring up an alert when you drive towards the location. There are several situations this makes sense: it can be used to warn ahead of dangerous locations, or locations that need to be approached with care (like a play school, or a speed trap). LapTimer Online features use dynamically created Alerts to bring up a yellow flag ahead of positions another driver (or a steward) has marked.
- **Trigger POIs**: this is the most sophisticated POI. On top of holding some information like all other POIs, it controls LapTimer timing, data, and video recording. Trigger POIs are typically named simply a Trigger. Triggers will be discussed in more details below. At the time of writing this document, the triggers come as Stop / Go, Stop, Go, Split, Standing Go, and Cancel triggers.

Detection Corridor

Before we dive into using triggers for automatic timing, some remarks on detection of Points of Interest. While you move around, LapTimer will continuously search for POIs nearby. Heading of movement is an important criteria here as the heading is the field of view. Points of interest "behind", are ignored, they are of no use any more. LapTimer's area of detection of POIs is defined by a corridor ahead. It's length, width, and angle to the left and right is determined by speed, receiving accuracy, and a setting that can be used to customize detection for your conditions (see Settings in chapter User Interface, the corridor can be set to Narrow, Normal, Wide).



Detection Corridors depend both on Settings and Speed

Approaching a POI is made up from two phases. The first phase is LapTimer becomes aware of the POI as it appears in the detection corridor (see picture above). While an Alert will immediately issue an alert to the user in phase one, a Trigger is simply monitored. In the seconds phase, a POI in the detection corridor has been passed – the POI is somewhere directly behind the car. For alerts, this simply means the alert dialog is removed from screen, while a trigger will be checked against some criteria finally leading to either a trigger action, or no further action. This test checks if the trigger has been within the

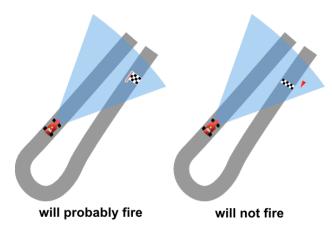
corridor width when the virtual line given by the Trigger, and the plot driven met. In case it was, LapTimer will perform the behavior given by the trigger – it will perform a timing action.

As LapTimer may detect several POIs within the corridor at the same time, additional prioritization is applied. The good thing about this is it will work for most – if not all – situations as you would expect it. Simple information POIs for example, will never disturb a trigger action.

Trigger Direction

A special type of prioritization / validation is applied for triggers with a defined direction: triggers can have a direction they are expected to be passed. As an example, a trigger with a direction of 90° (= going East) defined, will trigger only in case it is passed from West to East. It will not trigger when passed from East to West!

But why should a trigger have such a direction defined? One issue with triggers on tight tracks (e.g. kart) is they may fire on counter straights where LapTimer identifies the trigger to be passed when it does actually not belong to the current position. Have a look into the picture below: on the left side, we see a trigger with no direction defined. Such triggers are displayed as a flag (checkered in our example). On the right side, we see a trigger with a direction defined, it is displayed as a line (like a start / finish line).



Algorithm for trigger detection area

As a goody, LapTimer will set the detection corridor to Wide in case it handles triggers with a direction. Directions give a very strong criteria for validity of a trigger – it is simply not necessary to apply a probably too tight corridor if we have such a direction. Using a wide corridor will practically lead to a very low rate of triggers missed due to bad GPS accuracy – a clear bonus.

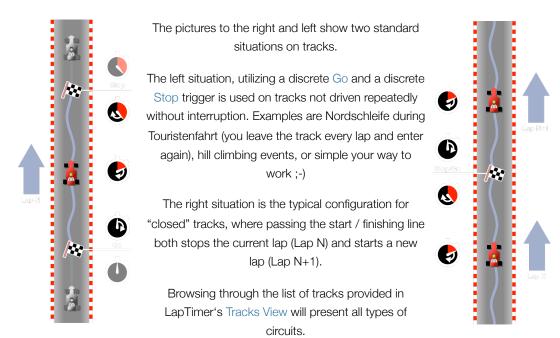
There is one important downside however: in case you have a track being driven in both directions, triggers with a direction will work in one direction only.



A Track Set (set of triggers) featuring triggers with directions will work one-way only. In case the track is driven clockwise *and* counter clockwise, two separate track sets are required.

Using Triggers for Automatic Timing

As outlined above, there are several timing and recording actions a trigger can be configured for. The basic actions are starting a new lap timing (Go) and stopping a lap timing currently active (Stop). For typical circuits laps are driven one after the other, a combined Stop/Go trigger is available. Split triggers mark the current position and time as a split time allowing sector analysis later. Finally, a Cancel trigger will cancel the current lap's recording – which is a convenient operation when coming back to the pits.



The two examples above cover flying starts used during typical track days and races. Especially in the US, there is a very popular type of competition performed getting special support by Standing Go triggers. This type of trigger is similar to Go triggers as it starts timing too. Different to Go triggers, it fires as soon as the vehicle exceeds a certain acceleration. Before Standing Go triggers are ready to fire, you need to take position near the trigger and stop your car / motorbike, and finally confirm you have staged it.



Messages when taking position at Standing Go

Now, once you accelerate, timing will start. This in turns means you measure the time from starting movement to passing the finish line (assuming you use a Stop trigger here). You need to decide which type matches your requirements best.

Finally, Cancel triggers simply cancel the current timing / the current lap. They can be used to cancel incomplete laps - mainly when you leave the track to the pits without finally passing the start / finish line. NB: the Cancel operation will cancel the last incomplete lap, not the full session.

Interpolation

As GPS does not deliver positions continuously, but as distinct fixes with a certain update rate (e. g. 1 Hz), the fix will be never 'on' the POI position. Instead, it will be slightly before or after the POI. To make things a little more complicated, it will be slightly to the left or right too.

To get timing accuracy beyond update rates, LapTimer does sophisticated interpolation to calculate the actual point and time the start / finish line / trigger has been passed. To accomplish this, LapTimer calculates the trigger's perpendicular projection to the virtual line between the GPS fix before and the GPS fix after the trigger. Having calculated this point, the time this point was passed is calculated by linear interpolation from the speed at the fix before, the speed at the fix after, and the distance between these two points. As long as the speed development is nearly linear, this interpolation yields accuracy beyond 0.01 seconds. This is the reason why even receivers with 1 Hz update rate are sufficient for timing popular track days.

Here is a list of accuracies typically achieved. Please note accuracy reported by GPS sensors are statistic values achieved 95% of the time. You need to expect deviations from the named accuracy accordingly.

- iPhone3G's and 3GS's internal GPS: better than 0.5 seconds (not recommended)
- iPhone4's, 4S's, and 5's internal GPS: better than 0.1 seconds
- Any iPhone directly integrated with a supported external sensors (see www.gps-laptimer.de/Compatibility.html): around 0.05 seconds, 5 Hz device better than 0.05 seconds