Tech Tips for Data Logging
Author "motolefty" from the TuneECU-Forum of https://tuneecu.boards.net/

"Start logging and stop guessing!"

Data logging is an important feature of TuneECU.

It lets you record what your engine is doing WHILE you're actually riding the bike.

This is a powerful feature, on par with the mapping changes and ECU manipulations that can be performed with TuneECU.

However, data logging in TuneECU can be "quirky" to use and challenging to learn.

Here are some tips to help you **get the most out of data logging with TuneECU**.

DISCLAIMER: I had no role in developing the TuneECU app and have no access to its "internals" (source code).
These tips are just the things I learned trying to troubleshoot my own bike (2016 Street Triple 675 ABS).
All of these findings are based on "a sample size of one". As such, YMMV!
If you discover something that goes beyond this writeup, please post a reply to share your findings.
Lastly, I will use the words "point" and "sensor" interchangeably.

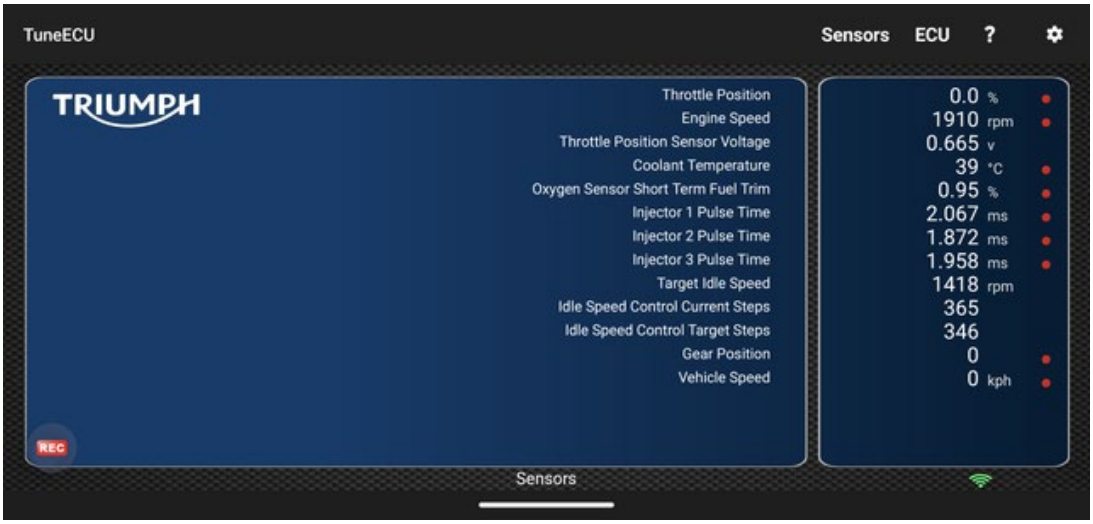
A big thanks to Alain for reviewing the first draft and providing some clarifications.

***** Getting Started *****

Obviously, the first step is to get TuneECU connected to your bike. I have no advice for that. Go check the rest of this forum...

Then read the basic instructions here: Data Logging

Then give it a try. Here's what it looks while recording some data points for my bike:



Hint: **Click on the pictures to enlarge them** (in a new tab).

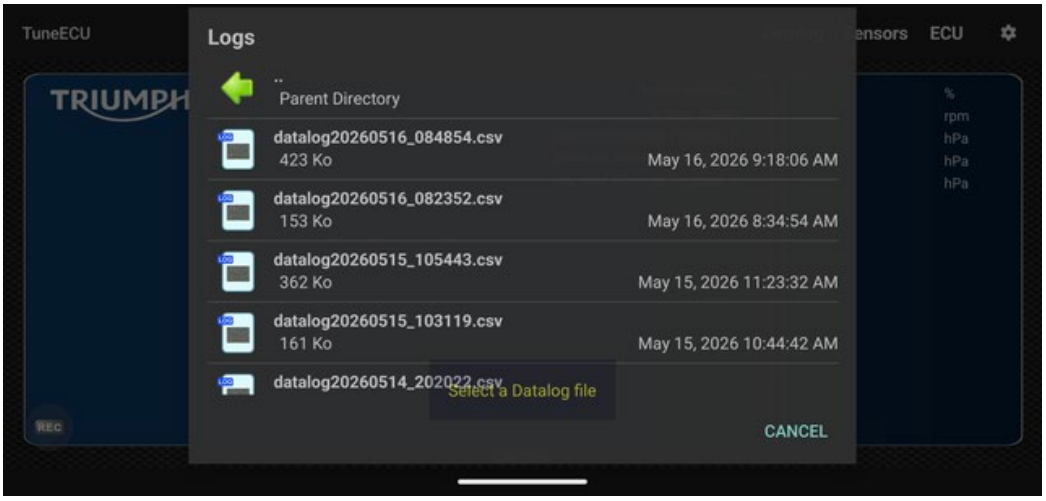
Notice that the Record Button ("REC") turns red when TuneECU is logging values.

Notice also that only some of the selected points are being recorded, as indicated by the red dots on the right. (More on this later...)

***** Using the Log Viewer *****

Once you've recorded a log file, TuneECU has a sophisticated and easy-to-use "log viewer" that can be used to graph the recorded sensor values.

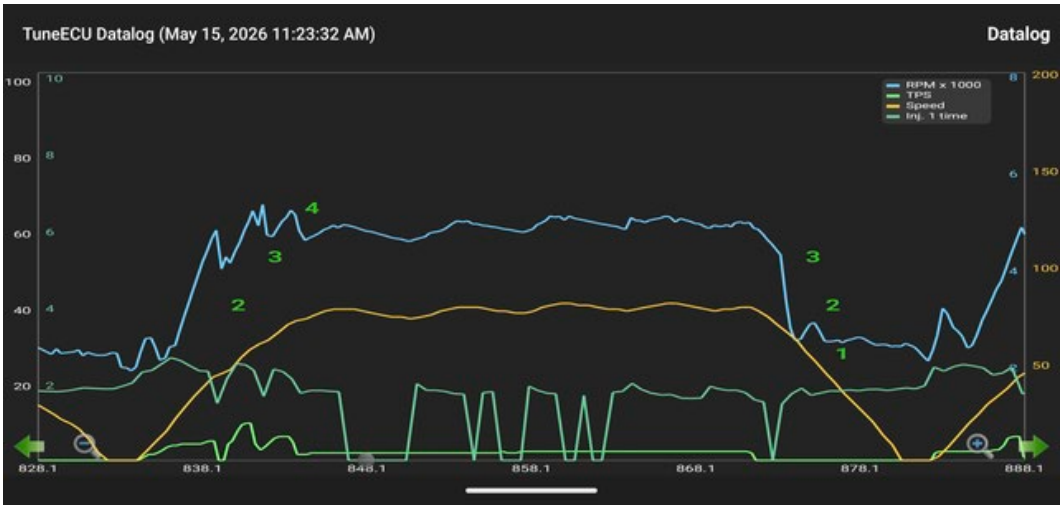
You open the Log Viewer by choosing "Open" from the "Datalog" menu and then selecting the desired "datalog*xxx*.csv" file:



Shortcut: Choosing "Display" from the "Datalog" menu will automatically open your most recent log file, bypassing the file selection dialog.

Note: The "Datalog" menu will be hidden until you've recorded at least one log file.

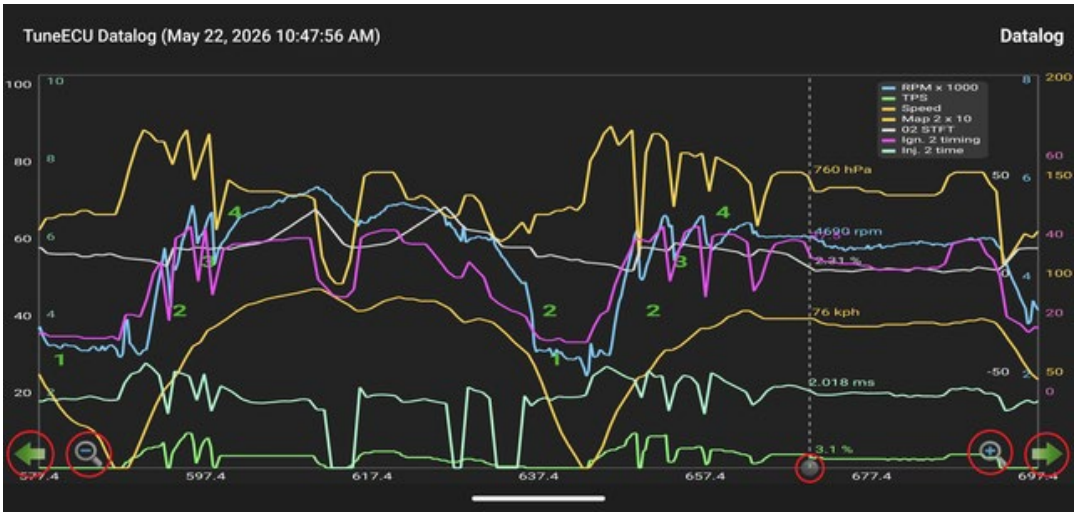
You can then scroll back and forth along the X-axis (which represents time in seconds) to examine your bike's performance during the test ride:



Let's look closer at the screenshot above to get some practice reading the graph.

This is a 60-second interval covering a steep downhill stretch where the bike repeatedly yanks me back, then surges forward.
Start by looking at the orange trace for "Vehicle Speed". Notice how it starts at zero, rises to 80 kph (50 mph), then falls back to zero.
Notice how vehicle speed correlates with "Engine RPM" and "Gear Position". You can easily see the upshifts, the steady middle section, and then the downshifts at the end.
Now look at "Throttle Position" (the green trace at the very bottom). Notice that it's perfectly steady in the middle section. (It's a steep hill: 2% throttle still gives 50 mph.)
Lastly notice how the ECU "helpfully" cuts the fuel delivery to ZERO multiple times during this descent (the blue-green trace for "Injector Pulse Time 1").
This proves that the "bucking" is being commanded by the ECU ("Decel Fuel Cut Off"), not by the knucklehead on the throttle. (PM me if you know how to fix this!)

Here's a second Log Viewer example that lets us dive a little deeper (a different downhill stretch followed by uphill twisties):

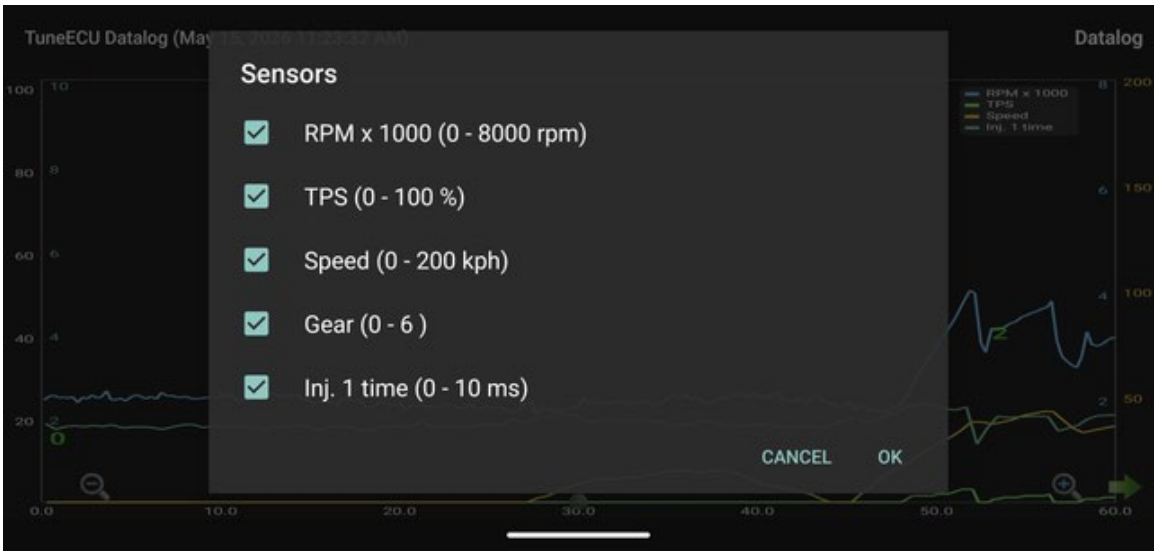


This example has more sensors, covers more time (2 minutes), and shows the numeric values at a given time.
The zoom buttons (highlighted) allow you to expand or compress the time span of the display.
You can set the time span to 15, 30, 60, 120, or 300 seconds. The default is 60 seconds (as in the first example).
The scroll buttons (highlighted) allow you to move the display forward or back by one "screenful". Or just swipe it.

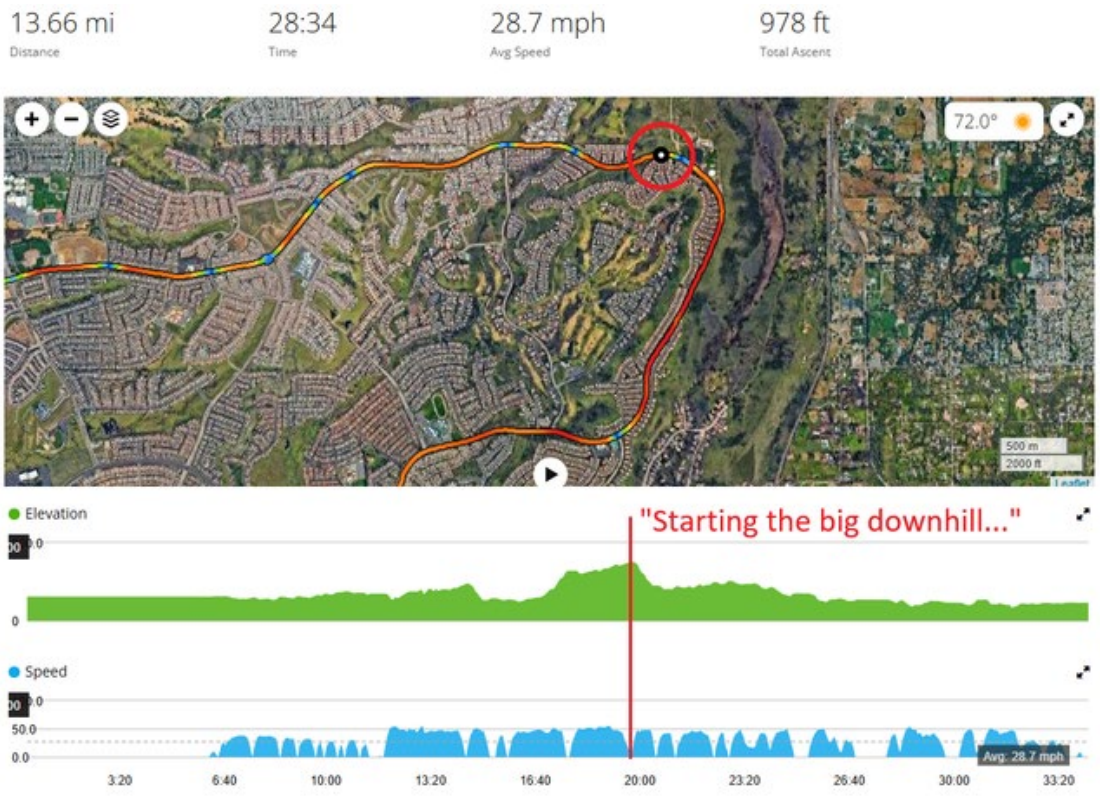
If you press on the X-axis (highlighted), the Log Viewer will show you the numeric value for each sensor at that point in time.
This is a very handy feature that lets you "get quantitative" without opening the actual log file.
By sliding your finger from left to right you can see how the values change over time. "Nice!"

Next, notice that each trace on the graph has its own color-coded scale (displayed on the left or right Y-axis).
If you tap on the Legend box, a checkbox list of the graphed sensors will be displayed.
This shows you the "unit of measure" for each sensor and its value range.

By unselecting a sensor on this list you can (temporarily) hide it to simplify the graph.
This can be helpful if you're drowning in too much information.



Lastly, if your test ride lasts for more than a few minutes it's easy to get "lost" trying to read the graph. You won't be able to match the log to your riding experience.
I found it helpful to use my Garmin watch to track location vs time. (UWAYL - "Use Whatever App You Like")
I then matched the Garmin timestamp against the TuneECU traces for "Vehicle Speed" and "Gear Position" to identify each segment within my test ride:



OK, that's it for the basics.

Now let's unpack the quirks in TuneECU logging:

***** Not All Sensors Can Be Recorded *****

The biggest limitation to data logging with TuneECU is that **only about half** of the sensors that can be read and displayed (on the Sensors screen)
can also be "recorded" in the log file (and subsequently graphed in the Log Viewer).
Futhermore, the choice of which points can and cannot be logged seems quite arbitrary.
For example, none of the "binary" (On-Off) sensors can be logged, which seems to be a limitation based on their data type.
However, all the sensors that ARE logged are "numeric" types, but there are still many numeric sensors that cannot be logged.
Hence, the only way to know if a particular sensor can be recorded is to try recording it (subject to the additional limitations below).

***** Maximum Number of Displayed Sensors *****

The maximum number of sensors that can be READ is limited to the number of lines that can be displayed on the Sensors screen.
The Sensors screen does not support scrolling or continuation pages. (IMHO this is a completely reasonable limitation.)
On my large-screen Android phone the limit is 16 simultaneous points. (On devices with larger or smaller screens this limit might be different.)

If you select too many points, the TuneECU app will silently turn some of them off.

***** Maximum Number of Recorded Sensors *****

The maximum number of sensors that can be RECORDED (logged) is LESS than the number that can be read and displayed (above).
In my tests, this limit was usually 10 sensors, but sometimes 9 or 12 (as shown in the screenshot above).
The reason for this limitation is unknown.
The bottom line is this: Just because a sensor is being read and displayed doesn't guarantee that it's being recorded.
Always pay attention to the red dot beside each sensor line. The point is being recorded only if it shows a red dot.

As above, if you select too many points, the TuneECU app will silently turn some of them off.

***** Maximum Sampling Rate is ~70 ms per Sensor *****

Sensors are read sequentially ("polled") at a rate that is "kinda slow".
These are just the limitations of (decades-old) OBD2 technology.
On my bike it takes 70 milliseconds to read each point. YMMV!
Note: The polling rate can be determined by observing the gap between timestamps in the log file (see below).

Since it takes the same amount of time to read each point,
the total time needed to update the entire list is proportional to the number of sensors selected.
This is an **important tradeoff: The more sensors you select, the slower they go.**
Note: You can't SEE this tradeoff anywhere in the UI (including the Log Viewer). You can only discover it by examining the log file.

If you select just one sensor, it can be updated 14 times per second (i.e, every 70 ms).
If you select 2 sensors, they can be updated 7 times per second (i.e., every 140 ms).
If you select 7 sensors, they can be updated twice per second.
If you select 16 sensors (the maximum), they can only be updated every 1.2 seconds.

Use these "rules" to adjust your list of selected sensors for best results.

Note: Because of the next "quirk" the actual update rate is usually LESS than this.

***** The Sampling Rate is Not Identical for All Sensors *****

The math in the previous section assumes that all sensors are sampled at the same rate. However, this is NOT TRUE.
In reality, TuneECU gives priority to the "Engine Speed" and "Throttle Position" sensors and samples them from 2X-6X more often than "ordinary" sensors.
The rationale for this behavior is unknown.
The only way to determine the actual sampling rate (and the order in which sensors are read) is to

examine the entries in the datalog.csv file (see below).
Just be aware that actual sampling rates will generally be slower than those calculated in the previous section.
Note: You could avoid this quirk by excluding "Engine Speed" and "Throttle Position" from your selected sensors, but they're usually essential, so this solution isn't widely applicable.

***** Using the "Log" Checklist *****

CAUTION: Using the "Log" checklist can lead to unexpected logging behavior. **Most users should avoid this workflow**. "KISS me, Baby!"

The "Log" item on the "Sensors" menu opens a checklist of sensors that looks similar to the "List", but differs in two important ways:
(a) The "Log" dialog shows shows only the loggable sensors that were previously selected using the "List" dialog, and
(b) Unchecking a sensor in the "Log" dialog means that it will be displayed on the Sensors screen but NOT recorded in the log file.
In other words, the "Log" dialog allows you to selectively disable the writing of an otherwise loggable sensor.
Obviously, this will also disable the graphing of that sensor using the Log Viewer.

However, there is no advantage in doing this.
It won't improve scanning performance -- because you've already spent the time needed to read that point (see above).
And it won't usefully simplify your graph -- because the Log Viewer has an even easier way to hide an unwanted sensor trace (see above).
Furthermore, the workflow to enable and use this option is "seriously quirky". You've been warned...

The first step in calling up the "Log" checklist is to "Disconnect" from your ECU. "Okay, if you say so..."
Then go to the Sensors page and long-press (or double-tap) on the Record button. The Record button will change from gray to green.
At this point, the "Log" item on the "Sensors" menu will be enabled and you can use it to open the "Log" dialog.
Uncheck the sensors that you do not want recorded and press "OK" to save your changes.
Inspect the displayed sensors to confirm that the red dots disappeared from the sensors you do not want recorded.
Now either: (a) Reconnect to your ECU to automatically start recording (without toggling the Record button), or
(b) Long-press the Record button to make it gray again, then reconnect to your ECU and proceed as usual. "Bleh!"

The real pitfall of using the "Log" dialog to make a point unwritable is that the change is **silently persistent**.
If you uncheck a sensor in the "Log" dialog, TuneECU will remember this even if you remove that

sensor from the selection "List".
If you later restore that sensor to the "List" it will still be disabled for logging. Yikes! The setting is sticky and invisible.
You can only re-enable the logging of that sensor by re-checking its box in the "Log" dialog.
Most users will be better off without these complications.

***** Datalog Timestamps *****

The timestamp displayed in the Log Viewer title bar is the ENDING time for the log, whereas the timestamp in the filename uses the STARTING time.
This can lead to confusion, especially if you made back-to-back recordings (where the end time of the first looks like the start time of the second).

***** Deleting Old Log Files *****

The TuneECU app doesn't provide a mechanism to delete old datalog.csv files (stored in your "TuneECU/Logs" folder).
If you're a frequent user of data logging, you'll need to find some other "File Manager" to do this.

Now let's examine the recorded log file itself:

Hint: Skip this section unless you're a masochist or have a problem that can't be solved using the Log Viewer combined with the live Sensors screen.

***** Using the datalog.csv File *****

A **big benefit** of data logging with TuneECU is that you have access to the raw sensor log.
This allows you to use any tool imaginable to analyze and graph your log data: Excel, Pandas, ChapGPT, whatever!

The TuneECU log file is stored as a simple .csv file (Comma-Separated-Values) that can be opened by a wide variety of computer programs.
The files are stored on your Android device in the "TuneECU/Logs" folder.
The file names follow the pattern: "datalog*YYYYMMDD_HHmmSS*.csv" (where the timestamp is the STARTING time for the log)

There are four "header" lines at the top plus a blank line followed by a list of sensor readings.
Each sensor reading is written on its own line and has three fields: "Sensor ID", "Timestamp", and "Value" [my names]
The sensor ID is an integer that MIGHT vary according to your bike or ECU model. (I'm guessing they vary, but my sample size is one.)
The timestamp is an integer field that represents the number of milliseconds since the start of logging.
The reading's value is stored as a decimal number with one or more digits after the decimal place.
The "unit of measure" for each point is NOT recorded in the log file. You'll have to get that from the

Sensors display screen (see above).
The delimiter between data fields is a semicolon (';').

There are no headers for the field names used in the sensor readings. You just have to know what they are.
There are no headers to define the sensor ID numbers. You'll have to figure them out on your own (see below).

Here's an excerpt from the top of a TuneECU log file:

TuneECU build 6429
API=33
PARAM=3:0:0
Level=4

022;0;670.0
000;53;2030.0
024;124;660.0
001;190;0.0
091;250;1.833
000;326;2100.0
092;392;1.892
001;477;0.0
093;571;1.797
000;641;2070.0
005;704;0.0
001;769;0.0
004;830;0.0
000;891;2100.0
022;972;660.0
001;1026;0.0
022;1083;670.0
000;1149;2040.0
024;1209;670.0
001;1306;0.0
091;1400;1.869
000;1460;2020.0
092;1537;1.855

Note that the sensors aren't read in numerical order (5 comes before 4) and that sensors 0 and 1 (Engine RPM and Throttle Position) are interleaved (repeated) between the other sensors (as discussed above).

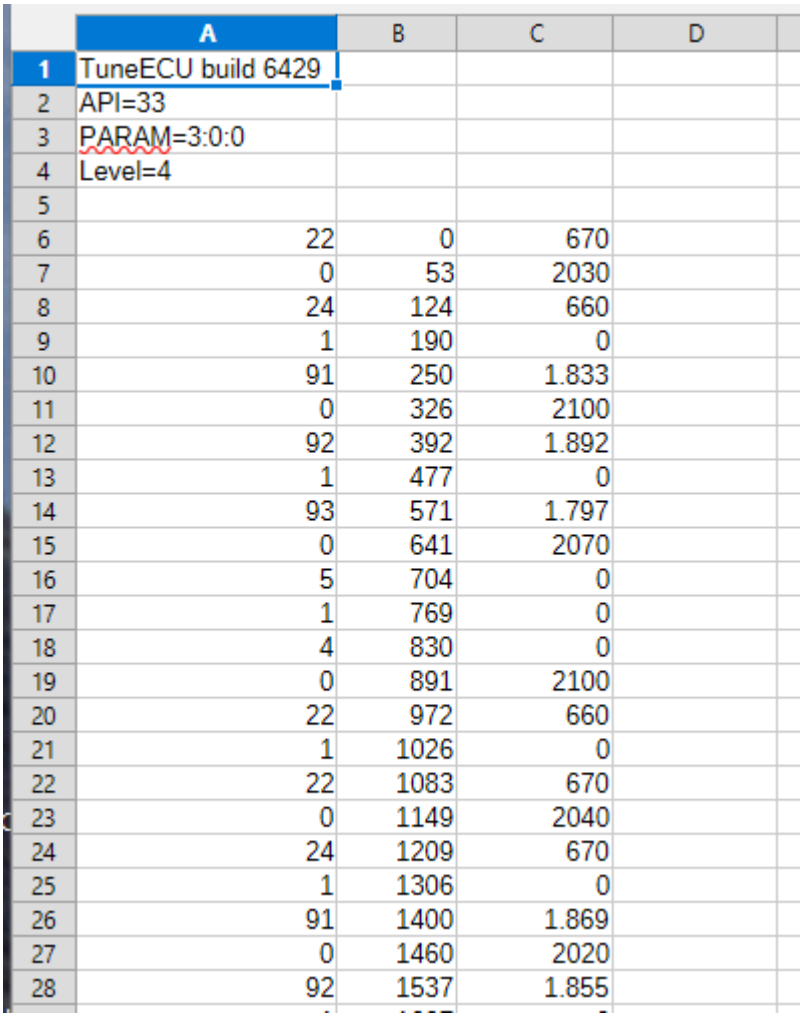
Hint: The interval between timestamps is the polling rate for your combination of ECU, OBD2 adapter, and Android device. On my bike it averages about 70 ms between readings, but fluctuates significantly.

***** Opening the datalog.csv File in Excel *****

I'm a dinosaur, so I'll use Excel (actually LibreOffice Calc) to read and graph the log file.
Feel free to use any tool you want. This is **just an example**.

The first step is to transfer the log file from your Android device to your computer. "Good luck!" (I can't help you with that.)
Then open the log file in Excel. On most PCs it should work to just double-click on the file listing in Windows Explorer.
Excel knows how to read .csv files and can automatically figure out that the delimiter used here is a semi-colon. So, just click "OK" to import the file.

Your file should now be displayed in Excel:



	A	B	C	D
1	TuneECU build 6429			
2	API=33			
3	PARAM=3:0:0			
4	Level=4			
5				
6	22	0	670	
7	0	53	2030	
8	24	124	660	
9	1	190	0	
10	91	250	1.833	
11	0	326	2100	
12	92	392	1.892	
13	1	477	0	
14	93	571	1.797	
15	0	641	2070	
16	5	704	0	
17	1	769	0	
18	4	830	0	
19	0	891	2100	
20	22	972	660	
21	1	1026	0	
22	22	1083	670	
23	0	1149	2040	
24	24	1209	670	
25	1	1306	0	
26	91	1400	1.869	
27	0	1460	2020	
28	92	1537	1.855	

The next step is to get rid of the TuneECU header lines. Just select lines 1-4 and delete them.
You should now have one blank line at the top. Type in some field names for the data columns. I use: "Sensor", "Timestamp", and "Value".
Now go to the Excel "Data" menu and choose "AutoFilter". You should get a drop-down filter on each data column.

Now use "Save As" to store your worksheet in a format that supports (and remembers) data analysis and graphing. I use ".xlsx".
If you like to keep things neat and tidy, you can now delete the original .csv file.

Your worksheet should now look like this:

	A	B	C	D
1	Sensor	Timestamp	Value	
2	22	0	670	
3	0	53	2030	
4	24	124	660	
5	1	190	0	
6	91	250	1.833	
7	0	326	2100	
8	92	392	1.892	
9	1	477	0	
10	93	571	1.797	
11	0	641	2070	
12	5	704	0	
13	1	769	0	
14	4	830	0	
15	0	891	2100	
16	22	972	660	
17	1	1026	0	
18	22	1083	670	
19	0	1149	2040	
20	24	1209	670	
21	1	1306	0	
22	91	1400	1.869	
23	0	1460	2020	
24	92	1537	1.855	

As this point, you can start analyzing the data without even creating a graph.
Just click on the drop-down filter for the Sensor column and choose whichever sensor you think will be most informative.
Then scroll down to a time range of interest and look at the numbers.
For example, on my bike, Sensor 91 is "Injector Pulse Time 1":

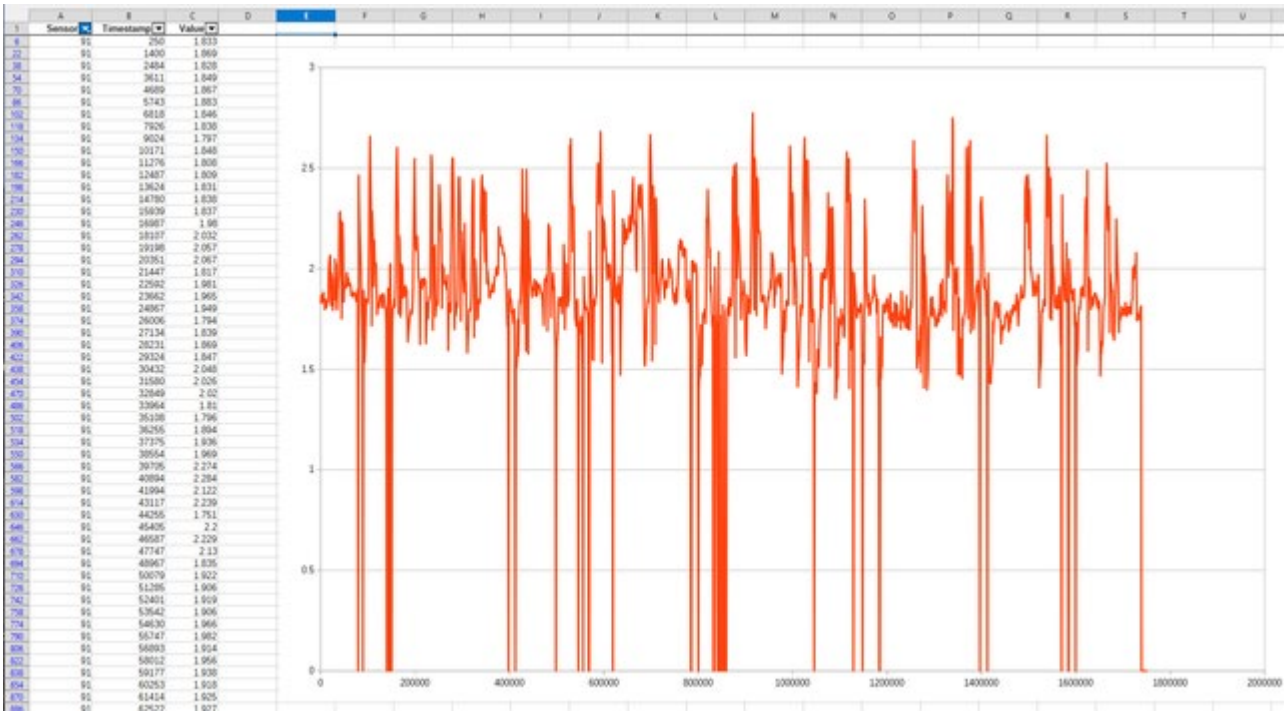
	A	B	C	D
1	Sensor	Timestamp	Value	
11558	91	827104	1.915	
11574	91	828404	1.831	
11590	91	829538	1.715	
11606	91	830742	1.764	
11622	91	832095	0	
11638	91	833894	0	
11654	91	835495	2.006	
11670	91	837239	1.786	
11686	91	838976	1.718	
11702	91	840825	0	
11718	91	842317	0	
11734	91	843486	2.083	
11750	91	844617	1.785	
11766	91	845711	0	
11782	91	846836	1.767	
11798	91	847963	0	
11814	91	849190	0	
11830	91	850360	0	
11846	91	851480	0	
11862	91	852618	1.82	
11878	91	853682	1.798	
11894	91	854791	1.781	
11910	91	855909	0	
11926	91	857118	0	
11942	91	858234	0	
11958	91	859358	0	
11974	91	860509	1.658	
11990	91	861642	1.805	
12006	91	862797	1.789	

Once you've got a theory of where to look, this simple technique lets you drill down to the raw sensor values with only a small amout of effort.

Remember: If you only need a few readings, you can also use the "Tap-for-Values" technique in the Log Viewer to display numeric sensor values.

***** Graphing the Data in Excel *****

The next level of sophistication (and effort) is to graph your data using Excel (or some other tool).
I'm a lightweight at this part of Excel, so this example is quite primitive. Here's the gist:
First, clear your filters (if any) from the previous step.
Then insert a new chart of type "XY (Scatter)" using the "Lines Only" display option. Then click "Next" twice (not "Finish").
Remove the data series for "Timestamp", leaving only the data series for "Value".
Edit the "X-Values" data range to change the referenced column from $A to $B (for both start and end of the range). Then click "Finish".
Resize the chart to make it as big as possible and ensure that it is "anchored" to the "page".
Now use the drop-down filter on the Sensor column to select and display one sensor at a time.
For example:



Excel lets you see the entire test ride all at once (~30 minutes), instead of viewing it chunk-by-chunk using the Log Viewer.
This makes it easier to see the big picture. Notice here that the ECU is repeatedly cutting my fuel delivery to zero. ("Who thought this was a good idea??")

***** List of Sensor IDs Needed for Each Bike *****

To interpret the readings in the TuneECU log file, you'll need a list of the sensor ID numbers for your bike.
At present, there's no easy way to get this list. You'll have to build it by hand, point by point.
(I *assume* that it's different for every bike.)
Start by recording a handful of "well-known" points, where you can easily recognize each point by its data value(s).
Then just keep expanding the list until you've covered all the points (sensors) that you care about.
Remember: Not all sensors can be logged (see above).

As an example, here's the (partial) Sensor ID list for my bike (2016 Street Triple 675 ABS):

	A	B	C
1			
2	**Code**	**Sensor**	**Units**
3	0	Engine Speed	rpm
4	1	Throttle Position	%
5	3	Coolant Temperature	C
6	4	Vehicle Speed	kph
7	5	Gear Position	(int)
8	6	Calculated Load	%
9	20	Manifold Absolute Pressure 1	hPa
10	22	Manifold Absolute Pressure 2	" "
11	24	Manifold Absolute Pressure 3	" "
12	26	Barometric Pressure	hPa
13	31	Air Temperature	C
14	40	Oxygen Sensor Short Term Fuel Trim	%
15	50	Battery Voltage	V
16	60	Oxygen Sensor Voltage	V
17	71	Ignition Timing 1	deg
18	72	Ignition Timing 2	" "
19	73	Ignition Timing 3	" "
20	91	Injector 1 Pulse Timing	ms
21	92	Injector 2 Pulse Timing	" "
22	93	Injector 3 Pulse Timing	" "

To get the "units of measure", you'll need to scrape it from the Sensors screen or from the Log Viewer's "Sensors" list (tap on the legend).

If you build a sensor ID list for your bike, please consider sharing it here to make it easier for the next guy. 🙂

OK. That's it for this tutorial. "Go forth and do great things." TuneECU can help.

P.S. **Let's keep this thread on point.**
Only post a reply if you have additional information to contribute, such as an example of how data logging pinpointed your problem or the sensor ID list for your bike.
Otherwise, just hit the "Thumbs Up" button.

Thanks!