Quad Channel Inductive Loop Vehicle Detector

PC Configurator Software

https://github.com/elektronika-ba/quad-loop-detector-configurator

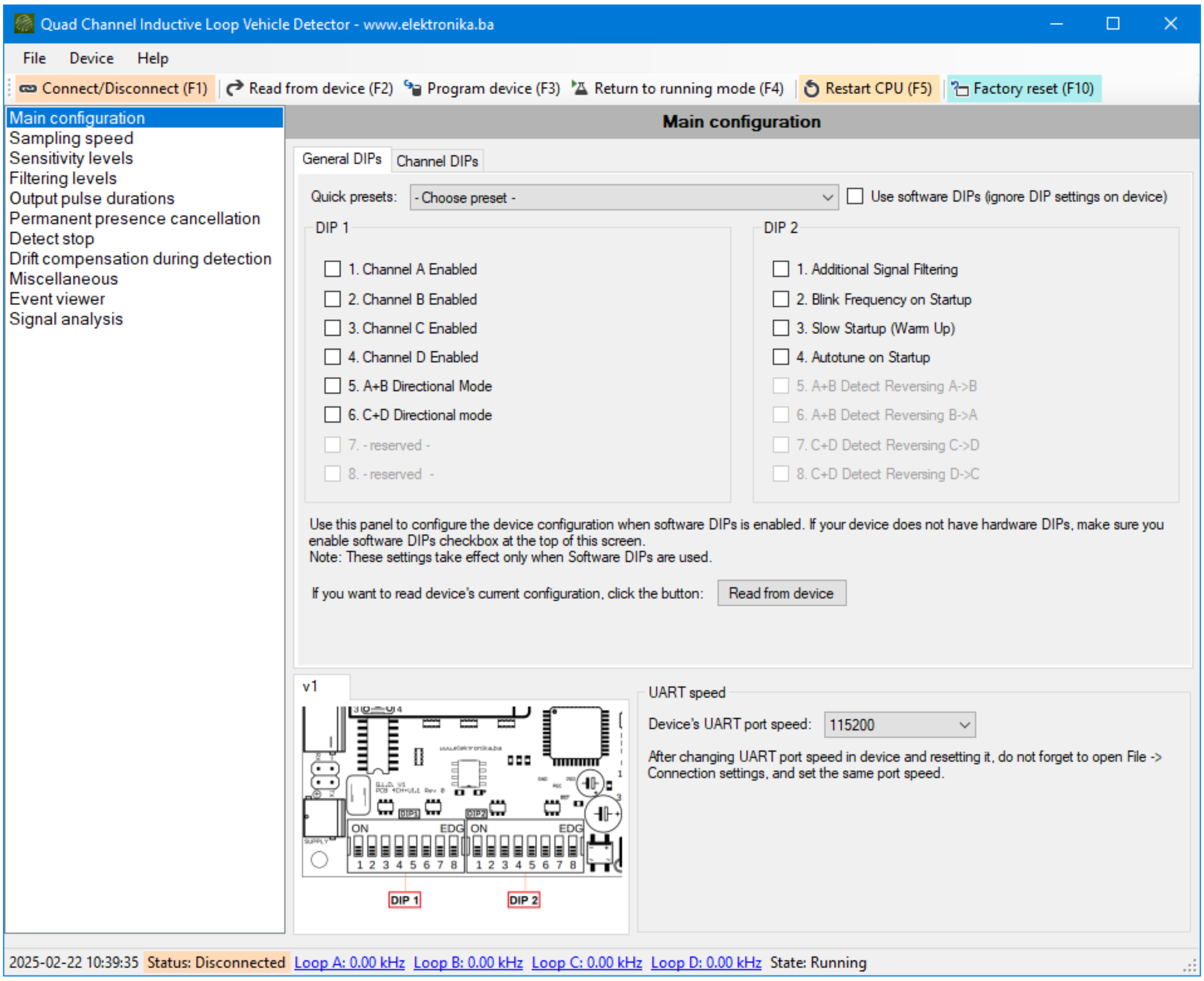

Figure 1 - PC Configurator Software

This software is open source. You can download source code at:

https://github.com/elektronika-ba/quad-loop-detector-configurator

Contents

Introduction

This software is used to configure and fine-tune the Quad Channel Inductive Loop Vehicle Detector. Communication with the device is done via UART at 4 possible speeds 8N1, no flow control. Baud rate speeds that device support are: 9600, 19200, 57600 and 115200 (default).

Initiating the communication

In order to start the communication, device needs to exit "running mode" (normal operating mode) and enter "communication mode". This is done by sending "start character" to **device (slave)** from **PC (master)**. Start character (byte) is CTRL+Z (0x1A hexadecimal). After device receives this character it will enter communication mode and answer with a reply:

```
READY>v2
END>
```

In case device does not receive another command within 25 seconds, it will automatically return to "running mode".

In this mode, device can accept a number of commands:

Command byte (ASCII)	Description
G	Get **config packet** from device
S	Send **config packet** to device
L	Toggle event logging mode
E	Read errors (if any) from device
F	Get current frequencies of all loops
A	Toggle real-time frequency reporting mode
C	Toggle real-time frequency counter reporting mode
W	Restore factory settings in device
T	Read hardware DIPs as read at start-up (power-up)
X	Soft-reset the device and return to communication mode
Y	Soft-reset the device and return to running mode
Q	Exit communication mode and return to running mode

Table 1 - List of available commands

Notes: Each command that device sends is terminated with: "END>" following carriage return and line feed characters (0x0D and 0x0A respectively). Master device **must not** terminate their own commands with carriage return and line feed characters!

Get config packet

Reading current device's settings can be done by issuing "G" command. Upon reception, device will reply with current config packet:

```
GET>82
AA04050607043C3C081E1E003C00320032002A00280022001E001A00140010000A00
080008000600040002C80332C803040001050A141932193219321932 84D000F00010
10101010101010003232323 20F01
END>
```

Note: Reply contains number of bytes that follow (82). Bytes are encoded as hexadecimal ASCII.

Set config packet

In order to send new config packet to device, command “S” should be issued. The device will answer with:

```
SET>82
END>
SET><
END>
```

Number 82 is the number of bytes that device now expects. Right after that, device will send “<” which is a flow-control character. This means that device is ready to receive next byte (actually next two hexadecimal ASCII characters). Device will keep sending this character after every byte it receives until the last one.

In case timeout occurs, device will reply with:

```
ERR>
END>
```

After successful programming of new config packet, device will reply with:

```
OK>
END>
```

Note: Device needs to be reset (by issuing “X” or “Y” command, or by pressing the hardware reset button) for new configuration packet to activate.

Toggle event logging mode

In order to enable or disable event logging, device must receive command “L”. Upon reception, device will reply with:

```
LOG>1
END>
```

or

```
LOG>0
END>
```

…depending on whether event logging has been enabled or disabled. When enabled, it is required to exit the “communication mode” and return to “running mode” with command “Q” or by waiting for internal communication timeout to happen, in order to start receiving events. When an event happens, device will report it in format (example):

```
EVENT[1]>08
END>
```

Some events have an additional parameter after the event code and it is always separated by comma:

```
EVENT[0]>01,041.129
END>
```

Brackets “[ ]” identify which loop produced an event. Loop A will be marked as [0] and loop B with [1]. For some “joined” events such as directional logic reporting, these brackets contain the value of the channel group, instead of the channel index.

Example for the joined event of channel group 0 (A+B):

```
EVENT[0]>15
END>
```

Example for the joined event of channel group 1 (C+D):

```
EVENT[1]>15
END>
```

List of all events that device can send is in the following table:

Event Code	Joined event	Parameter format (where applicable)	Description	Example
01	No	XXX.YYY	**Un-detect** Parameter shows the strength of detection in percentages in fixed format 000.000	EVENT[1]>01,012.333
02	No	XXX.YYY	**Rollaway, did not stop** Parameter shows the strength of detection in percentages in fixed format 000.000	EVENT[0]>02,022.236
03	No	XXX.YYY	**Un-detect, because of PPC** Parameter shows the strength of detection in percentages in fixed format 000.000	EVENT[1]>03,052.236
04	No	XXXXX	**Movement while stopped** Parameter shows the strength of movement while vehicle was stopped, in same units as sensitivity. Parameter is in 5 digits fixed format.	EVENT[0]>04,00023
05	No	XXXXX	**Movement before stopped** Parameter shows the strength of movement before vehicle has possibly stopped, in same units as sensitivity. Parameter is in 5 digits fixed format.	EVENT[1]>05,00009
06	No	XXX.YYY	**Repeated stop detected** Parameter shows the strength of detection in percentages in fixed format 000.000	EVENT[0]>06,014.122
07	No	XXX.YYY	**Initial stop detected**	EVENT[0]>07,015.122

			Parameter shows the strength of detection in percentages in fixed format 000.000	
08	No		**Detect** (Detection strength cannot be calculated at this stage)	EVENT[1]>08
09	No		**Movement detected after PPC event**	EVENT[0]>09
11	Yes		**Cancelled 1->2 direction**	EVENT[0]>11
12	Yes		**Going back 2->1 direction**	EVENT[1]>12
13	Yes		**Passed 2->1**	EVENT[1]>13
14	Yes		**Cancelled 2->1 direction**	EVENT[0]>14
15	Yes		**Going back 1->2**	EVENT[0]>15
16	Yes		**Passed 1->2**	EVENT[1]>16

Table 2 - Event list

Note: Percentage of signal strength is calculated as frequency change in same units of value as sensitivity setting, over maximum possible sensitivity that can be set.

$$\% = (frequency\_change * 100) / maximum\_possible\_sensitivity$$

…where:
frequency_change is the change in oscillator frequency but in same units of value as sensitivity setting;
maximum_possible_sensitivity = 1000.

Read errors (if any) from device

At any time during operation, device can be polled in order to determine whether there is (or was) an oscillator error by issuing the "E" command. Device will reply with error codes for all loops. Loop A is marked as "[0]", loop B is marked as "[1]", loop C is marked as "[2]" and loop D is marked as "[3]":

```
ERROR[0]>0
ERROR[1]>2
ERROR[2]>0
ERROR[3]>5
END>
```

Error code	Description	Example
0	No error in history	ERROR[0]>0
2	Oscillator stopped working	ERROR[0]>2
3	Frequency too low for operation	ERROR[1]>3
4	Frequency too high for operation	ERROR[0]>4
5	Oscillator unstable, cannot perform (re)calibration	ERROR[1]>5

Table 3 - Error codes

Get current frequencies of all loops

In order to read current frequency for all loops, command "F" should be issued. It will reply with frequencies in kHz for all loops. Loop A is marked as "[0]", loop B is marked as "[1]", loop C is marked as "[2]" and loop D is marked as "[3]":

```
FREQ[0]>80.60352
FREQ[1]>75.98752
FREQ[2]>78.87520
FREQ[3]>81.75233
END>
```

Toggle real-time frequency reporting mode

In order to enable or disable real-time frequency reporting, device must receive command “A”. Upon reception, device will reply with:

```
ANA>1
END>
```

or

```
ANA>0
END>
```

…depending on whether frequency reporting has been enabled or disabled. When enabled, it is required to exit the “communication mode” and return to “running mode” with command “Q” or by waiting for internal communication timeout to happen, in order to start receiving signal frequency information. Device will report frequency (for every active loop) on every 8 collected samples from loop oscillator.

```
ANA[1]>50.2122,50.2122,50.2122,-48.3212,-48.3213,-48.3213,-48.3213,-48.3213
END>
ANA[1]>-48.3212,-48.3213,-48.3212,-48.3213,-48.3213,-48.3213,-48.3212,-48.3213,-48.3213
END>
ANA[1]>-48.3212,-48.3213,-48.3212,-48.3213,-48.3213,50.2122,50.2122,50.2122
END>
```

Note: Unit of value is kHz. Values are always in 0.0000 format. Negative value means that there is currently no detection, and positive values mean that there is a detection present.

Important note: Using this feature is recommended only when UART baud rate is set to maximum speed: 115200 bps. Slower UART speed will slow down device response!

Toggle real-time frequency counter reporting mode

In order to enable or disable real-time frequency counter reporting, device must receive command “C”. Upon reception, device will reply with:

```
CNT>1
END>
```

or

```
CNT>0
END>
```

…depending on whether frequency counter reporting has been enabled or disabled. When enabled, it is required to exit the “communication mode” and return to “running mode” with command “Q” or by waiting for internal communication timeout to happen, in order to start receiving signal frequency counter information. Device will report every frequency counter value (for every active loop) after sampling each loop channel.

```
C[0]>34171
C[1]>34007
C[2]>34094
C[3]>34070
C[0]>34171
C[1]>34007
C[2]>34094
C[3]>34070
...
```

Important note: Using this feature is recommended only when UART baud rate is set to maximum speed: 115200 bps. Slower UART speed will slow down device response!

Restore factory settings in device

Resetting device to factory settings is useful after programming wrong config packet. This will also configure device to use default baud rate: 115200 bps. This can be done by issuing "W" command. Device will confirm resetting with a reply:

```
FACTORY>OK
END>
```

To activate default config packet, device must be reset.

Read hardware DIPs as read at start-up (power-up)

When it is required to read device's hardware DIPs, command "T" can be issued. It will reply with:

```
DIP[0]>4F
DIP[1]>0C
END>
```

Values are encoded in hexadecimal ASCII. "[0]" marks DIP 1 (red) and "[1]" marks DIP 2 (blue).

Soft-reset the device and return to communication mode

Resetting device and automatically returning it to "communication mode" can be done by issuing "X" command. Device will answer with:

```
RESET>
END>
RESUME>
END>
READY>v2
END>
```

At this moment, device is in the same state as it was before issuing the reset command.

Soft-reset the device and return to running mode

Resetting device and returning to "running mode" can be done by issuing "Y" command. Device will answer with:

```
RESET>
END>
RESUME>
END>
```

At this moment, device is in the “running mode” and operational.

Exit communication mode and return to running mode

Exiting “communication mode” is done by sending the “Q” command. Device answers with:

```
QUIT>
END>
RESUME>
END>
```

At this moment, device is in the “running mode” and operational.

Format of config packet

Config packet format version “v2” consists of 82 bytes encoded as hexadecimal ASCII characters that are transferred over UART to and from the device. This means that config packet consists of 190 ASCII characters which are transferred over UART. Config packet contains all configurable parameters for device’s operation.

Config packet example:

```
AA0405060743C3C081E1E003C00320032002A00280022001E001A00140010000A00080008000
6000400002C80332C80304000010050A14193219321932193284D000F00010101010101010100032
3232320F01
```

Table 4 - Example of config packet

Value example (ASCII encoded hexadecimal)	Description	Recommended value
AA	Device configuration validity flag. If any other value is sent to device, upon restart (power cycle) it will load factory values.	AA
04 05 06 07	Device sensitivity levels. • First byte: sensitivity level for loop A • Second byte: sensitivity level for loop B • Third byte: sensitivity level for loop C • Fourth byte: sensitivity level for loop D	01-08
04C8C8 085050	Filtering parameters (3 bytes for normal filtering and 3 bytes for additional filtering). • First byte: Signal averaging size • Second byte: Negative drift compensation rate (frequency drift towards detection) • Third byte: Positive drift compensation rate (frequency drift away from detection) Note: Bigger signal averaging size equals better signal filtering but slower response time. Smaller drift compensation rate equals better signal filtering.	First byte: 01-08 Second byte: 28-FF Third byte: 28-FF

003C0032 0032002A 00280022 001E001A 00140010 000A0008 00080006 00040002	Sensitivity parameters for all 8 sensitivity banks. Each sensitivity bank/level consists of 4 bytes. Order of levels is 8, 7, …, 2, 1. For each sensitivity level bytes are as follows: • First and second byte (word): Detect threshold • Third and fourth byte (word): Un-detect threshold Example: Value 003C0032 for level 8 in this example means: 0x003C = 60 for detect threshold, 0x0032 = 50 for un-detect threshold. Note: Smaller number equals more sensitivity. Units of value is Hz (Hertz) and value of 1 corresponds to device's current sensitivity for current operating frequency. For actual values, refer to configuration of *Sampling speed*.	Detect threshold: minimum recommended value is 0004 Un-detect threshold: should always be a little bit smaller than detect threshold (by 15%)
C8	Detect stop timer. This is the time that device must not detect more change than defined with *Detect stop threshold*. The detection of change is done by second timer called *Detect stop slow check timer*. Note: Unit of value is 1 sampling speed period.	Around 1500 milliseconds
03	Detect stop threshold. This is the value of change that device must not detect within the *Detect stop timer* time. It is in same units of value as sensitivity parameters.	03
32	Detect stop slow check timer is a timer that device uses to sample the signal in order to detect the change. Note: Unit of value is 10ms	32
C8 03	Detection of loop activity absence. This is required for PPC option in order to start recalibration process once previously stopped vehicle leaves. • First byte: time of no-activity • Second byte: threshold of no-activity These two parameters have the same meaning as in *Detect stop* option (timer and threshold). Note: Unit of value for first byte is 1 sampling speed period. Unit of value for second byte is same as in sensitivity setting.	First byte: around 2 seconds Second byte: around 3
0400	Drift compensation during detection start timer. This is part of the no-movement detector in which device detects that vehicle has completely stopped over the loop so it can start compensating for change. Example: 0x0400 = 1024 sampling periods. Note: Unit of value is 1 sampling speed period.	Around 1100 so that timer starts after ~10 seconds.
01	Drift compensation during detection threshold. This is the value that must not change within *Drift compensation during detection start timer* period.	Minimum and maximum value recommended is 01
05 0A 14	Permanent presence canceller (PPC) timers. • First byte: PPC level 1 (short) • Second byte: PPC level 2 (medium) • Third byte: PPC level 3 (long)	01-FF

	Note: Unit of value is 1 minute. Example: 05 equals to 5 minutes.	
1932 1932 1932 1932	Output pulse durations for outputs: A, B, C and D. • First byte: normal pulse duration • Second byte: extended pulse duration Note: Unit of measure is 10ms. Example: 0x19 = 25 which equals to 250ms	
84D0	Signal sampling speed. Very important parameter that should be kept in recommended value range. Many device parameters depend on chosen sampling speed value. Example: 0x84D0 = 34000. Sampling speed = ((1/4000000)*34000)*1000 = 8.5 ms	639C- C350
00 F0 00	Software defined **General** configuration DIPs. • First byte: software DIPs enabled or disabled • Second byte: DIP 1 value • Third byte: DIP 2 value Note: First byte determines whether software DIPs are used. Value 00 here disables software DIPs and enables hardware DIPs, and value FF enables software DIPs and disables hardware DIPs.	First byte: 00 or FF
10 10 10 10 10 10 10 10	**Channel specific** configuration DIPs, in 4 groups. • First byte in group: DIP 1 • Second byte in group DIP 2	
00	Baud rate for UART communication. 0x00 = 115200 bps 0x01 = 57600 bps 0x02 = 19200 bps 0x03 = 9600 bps	00-03
32 32 32 32	Output activation delay for outputs A, B, C and D. Note: Unit of measure is 50ms. Example: 0x32 = 50 which equals to 2500ms	
0F	Slow startup timer Note: Unit of measure is 1 second.	
01	Runtime re-calibration delay timer Note: Unit of measure is 1 minute.	

Table 5 - Explanation of config packet fields

Note: The device does not perform validation of received config packet so in case invalid values are programmed it is possible to reset the device to factory values by using sensitivity-change buttons at power-up or by using command "W" in communication mode.

Event logging and real-time signal reporting

While device is in "running mode" it can send various event codes to master (PC) over UART. This way it is possible to receive all events such as detect, un-detect, speed information, direction information and so on. It is also possible to receive signal frequency information in order to feed it to processing software for further analysis and vehicle classification.

These options are explained in *Toggle event logging mode*, *Toggle real-time frequency reporting mode* and *Toggle real-time frequency counter reporting mode*.

Using the configurator software

Please connect your device to PC via provided USB<->UART(TTL) programming cable. This configurator software requires Microsoft .NET 4.0 to be installed on PC.

Connecting to device

To connect to device, click “Connect/Disconnect” button or press F1 on keyboard. This will open COM port where your device is connected. In case there is an error opening COM port, a popup will appear where you can select your COM port. On this screen, you can also set the baud rate **which must match** the one programmed in the device. If you cannot remember which baud rate your device is set to, you can try communicating on every baud rate that device supports (9600, 19200, 57600, 115200).

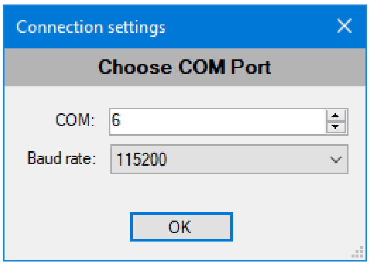

Figure 2 - Connection settings

After selecting COM port, click OK and try connecting again.

This dialog can also be accessed from the menu bar, “File” -> “Connection settings”.

Menu bar

Menu bar contains list of functions required for setting up the connection, creating, opening and saving profiles, connecting to device and performing programming.

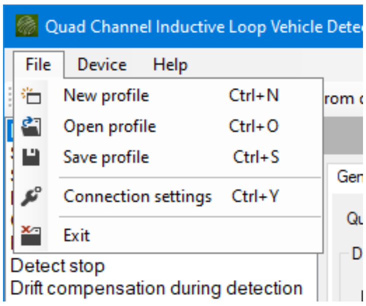

Figure 3 - Item “File” in menu bar

Item “File” contains option to create blank profile with default values for the device, opening previously saved profile and saving profile to a new file. By using profiles, it is easy to save settings to file and transfer them to a different location in order to program other remote devices.

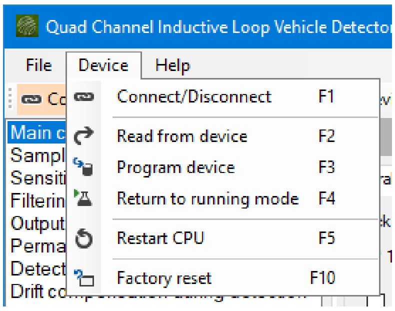

Figure 4 - Item “Device” in menu bar

Item “Device” contains options for communication with the device, such as reading current configuration, programming and restarting.

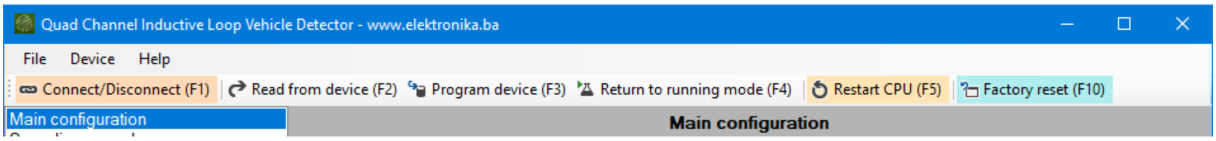

Figure 5 - Additional menu bar

There is also an additional menu bar with shortcuts for most frequently used options.

Status bar

Status bar contains information about current connection status, current loop frequency (refresh on click), current operating mode (as set by dropdown list in section “Operating mode”), and current device state (“communication mode” or “running mode”).

Figure 6 - Status bar

Parameters (sections)

Device parameters are divided in sections which can be accessed by clicking on each item in the left-hand list. Each section is separately described.

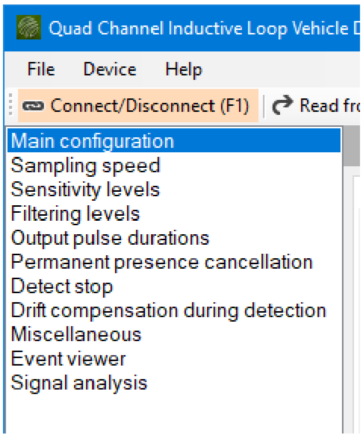

Figure 7 - Parameters (sections)

Main configuration

In this section, it is possible to configure device **general** and **channel specific** configuration by a method called "software DIPs". This means that it is possible to override hardware DIPs on device and remotely re-configure its operating parameters.

In order for device to accept software DIP settings, it is required to select the checkbox next to dropdown list ("Use software DIPs").

To read device's hardware DIP settings, button "Read from device" can be clicked. It is important to note that device's hardware DIP settings are loaded only upon device reset. Changing hardware DIPs during device operation is ignored.

Please refer to external document ***Quad Channel Inductive Loop Vehicle Detector (Q.L.D. V1)*** for detailed explanation of **general** and **channel specific** configuration.

This screen is also used to set device's UART port speed!

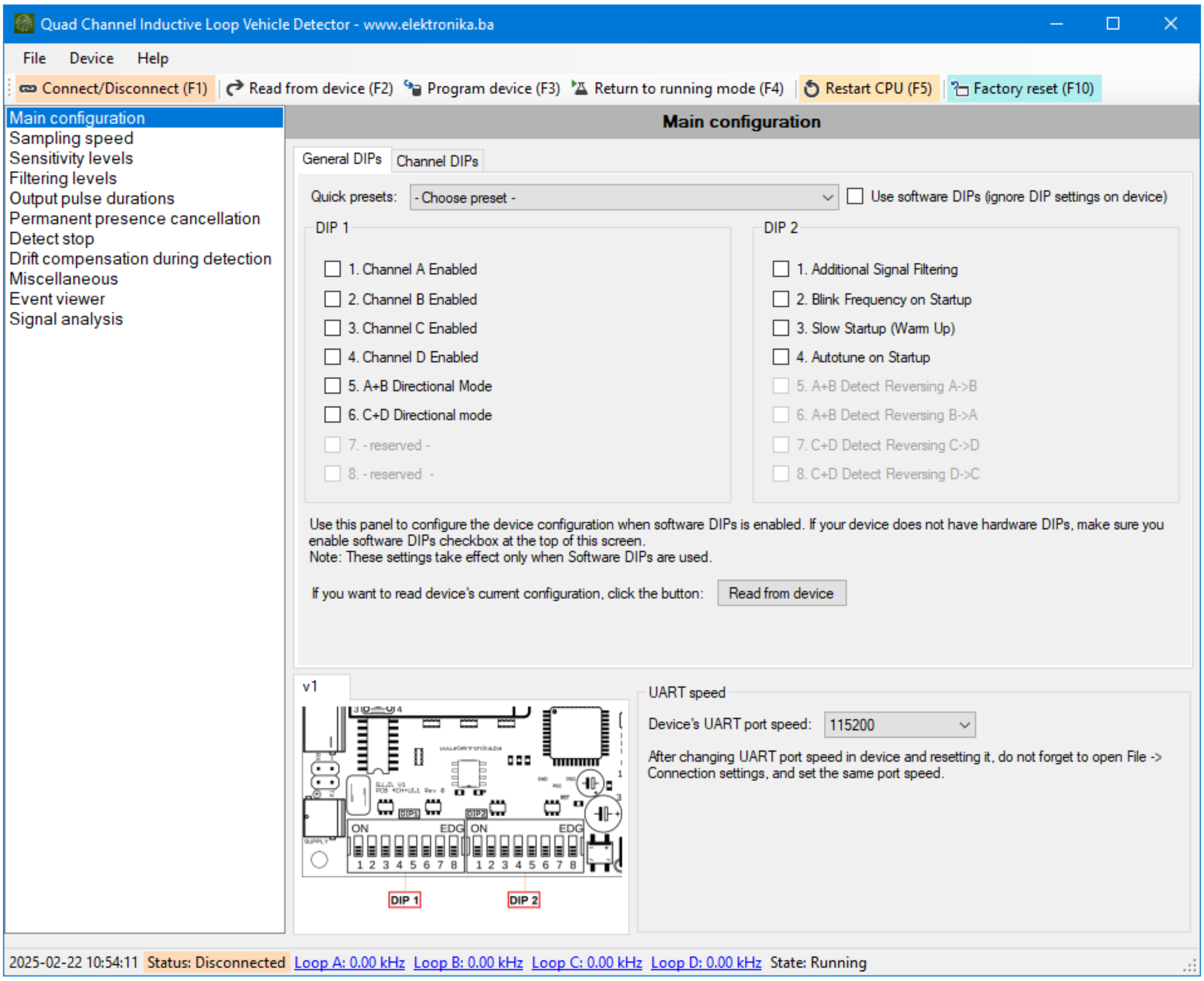

Figure 8 - Main configuration

Sampling speed

Sampling speed is the most important parameter that can be configured. Sampling speed has an effect on device sensitivity and response time. Device has a fixed sampling speed across its operating frequency range therefore it is possible to know how sensitive device actually is in Hz values at current operating frequency.

Response time is affected by sampling speed because device is “blind” during sampling. Only after sampling has completed, it can perform calculations and decide whether metallic object has come into range (or moved away).

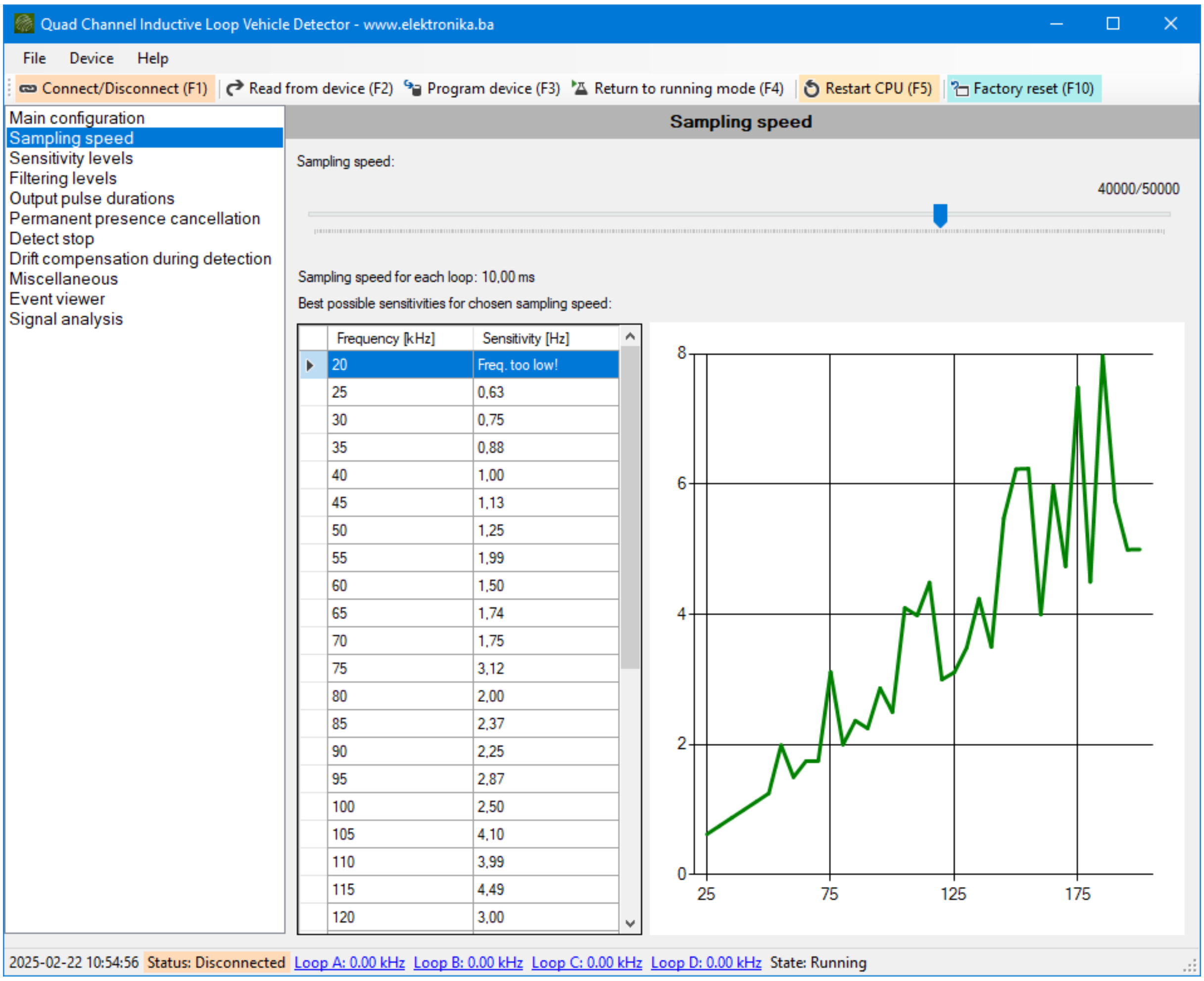

Figure 9 - Sampling speed

Sensitivity levels

One “sensitivity value” corresponds to amount of frequency change as shown in the table in picture above. This means that when device’s sensitivity threshold is set to value “4”, it will detect a metallic object when frequency changes from (for example) 30,000 Hz to 29,996.6 Hz. Value of 29,996.6 Hz corresponds to base frequency of 30,000 Hz *minus* the sensitivity value of 4 (which is 4*0.85 Hz at 30,000 Hz for sampling speed shown in the picture below). This yields sensitivity of 0.002834 % of change in frequency.

Sensitivity can be adjusted for each sensitivity bank (1-8). Currently selected sensitivity bank from the dropdown will be chosen as current device’s sensitivity for that channel upon device reset.

In order to achieve maximum device reliability of operation, it is not advisable to set “Detect threshold” to less than “4 units of frequency change”. It is also advisable to set the “Undetect threshold” at value less than “Detect threshold” by at least 15% - but this value can be experimented with.

Note: It is important to set maximum sensitivity for bank 8 and least sensitivity for bank 1, because ABS option will automatically switch to bank 8 when enabled and detection performed.

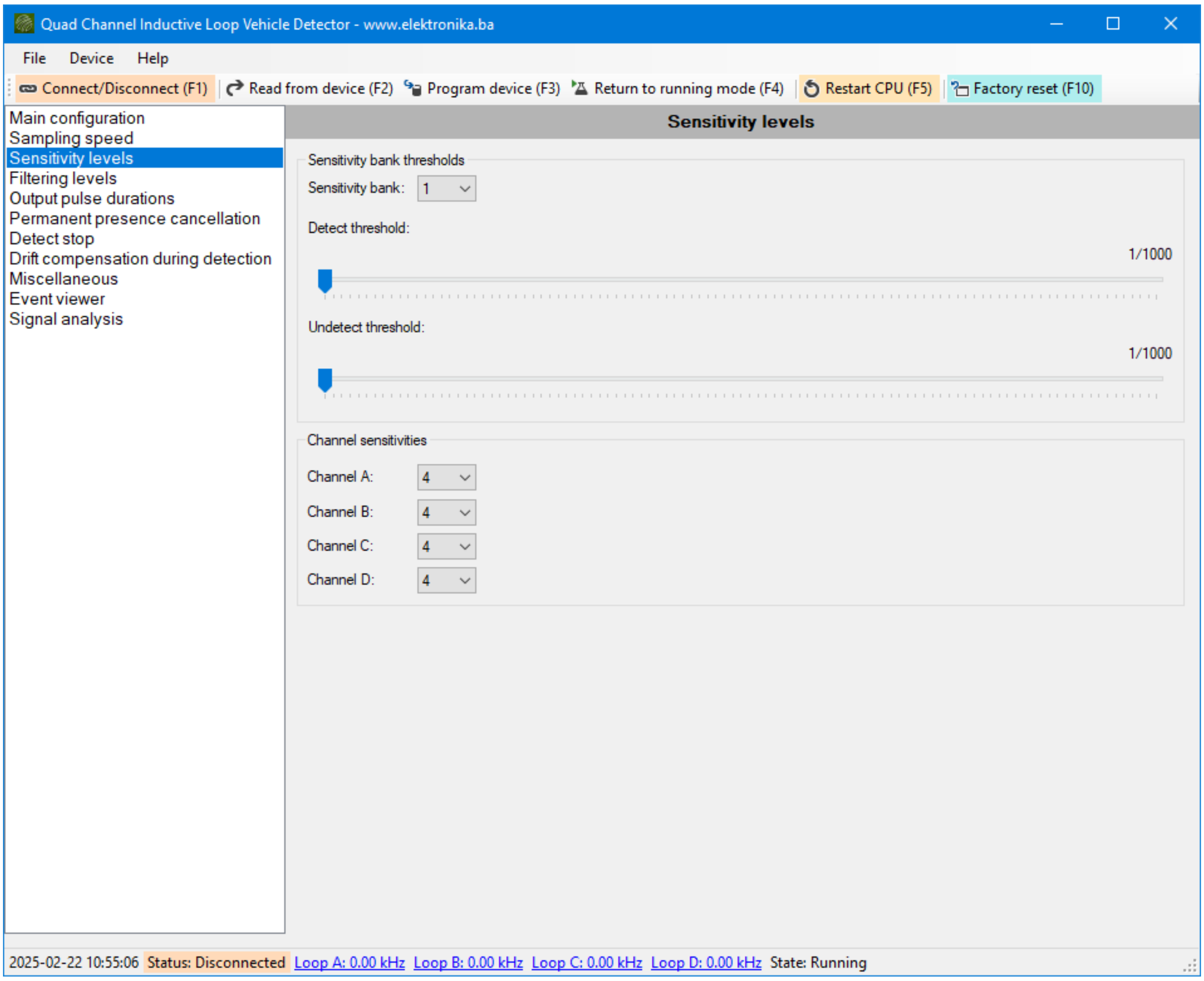

Figure 10 - Sensitivity levels

Filtering levels

Filtering is very important parameter for device's reliable operation. There are two filtering levels: "Normal" and "Additional" filtering.

There are two timers with adjustable period of execution. When timer executes, internal baseline frequency will be re-adjusted to currently measured frequency. This means that device will "track" the baseline loop frequency and re-adjust it as device operates. This will prevent device from false-triggering when ambient temperature slowly changes or loop slowly moves in ground. This re-adjustment is called drift compensation. Drift is compensated while device is not in "detect state" as well when it is in "detect state". When device is in "detect state" there are special additional timers and algorithms which are explained in *Drift compensation during detection* section. When baseline frequency changes slowly, the device will compensate after the negative or positive timer executes, depending on whether frequency change is negative or positive. Negative frequency change might result in a detection therefore this timer is always a bit slower than positive drift timer.

Negative and positive drift timer values follow principle: lower timer value means better signal filtering. Lower timer value also means that vehicle should move faster over the loop area because we would not want the device to quickly compensate for vehicle movement. We only want it to compensate for ambient frequency change which (hopefully) happens slowly.

Signal averaging is actually calculation of mean value of collected samples from the loop oscillator. Value in this parameter can be selected from 1-8 where bigger value means more signal filtering but a little bit slower response time.

Note: It is advisable to set better filtering values for "Additional filtering" dropdown item.

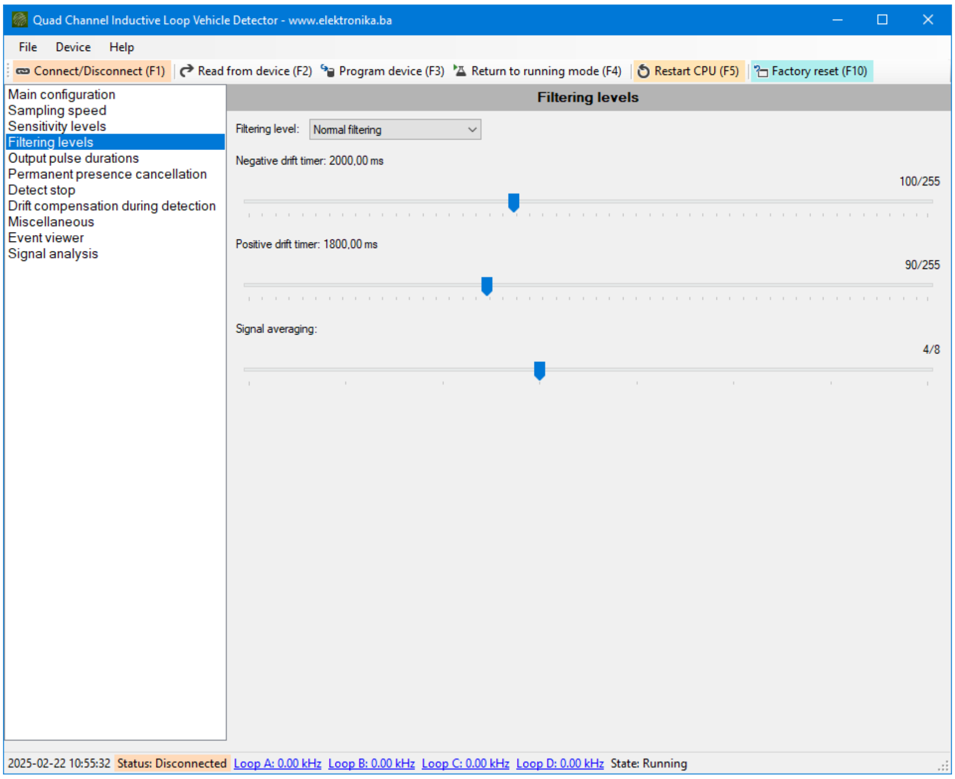

Figure 11 - Filtering levels

Output pulse durations

When outputs are configured to "pulse mode" instead of "presence mode", durations of these pulses can be adjusted in this section for all relays individually.

Note: It is advisable to set longer durations to "extended pulse duration" slider.

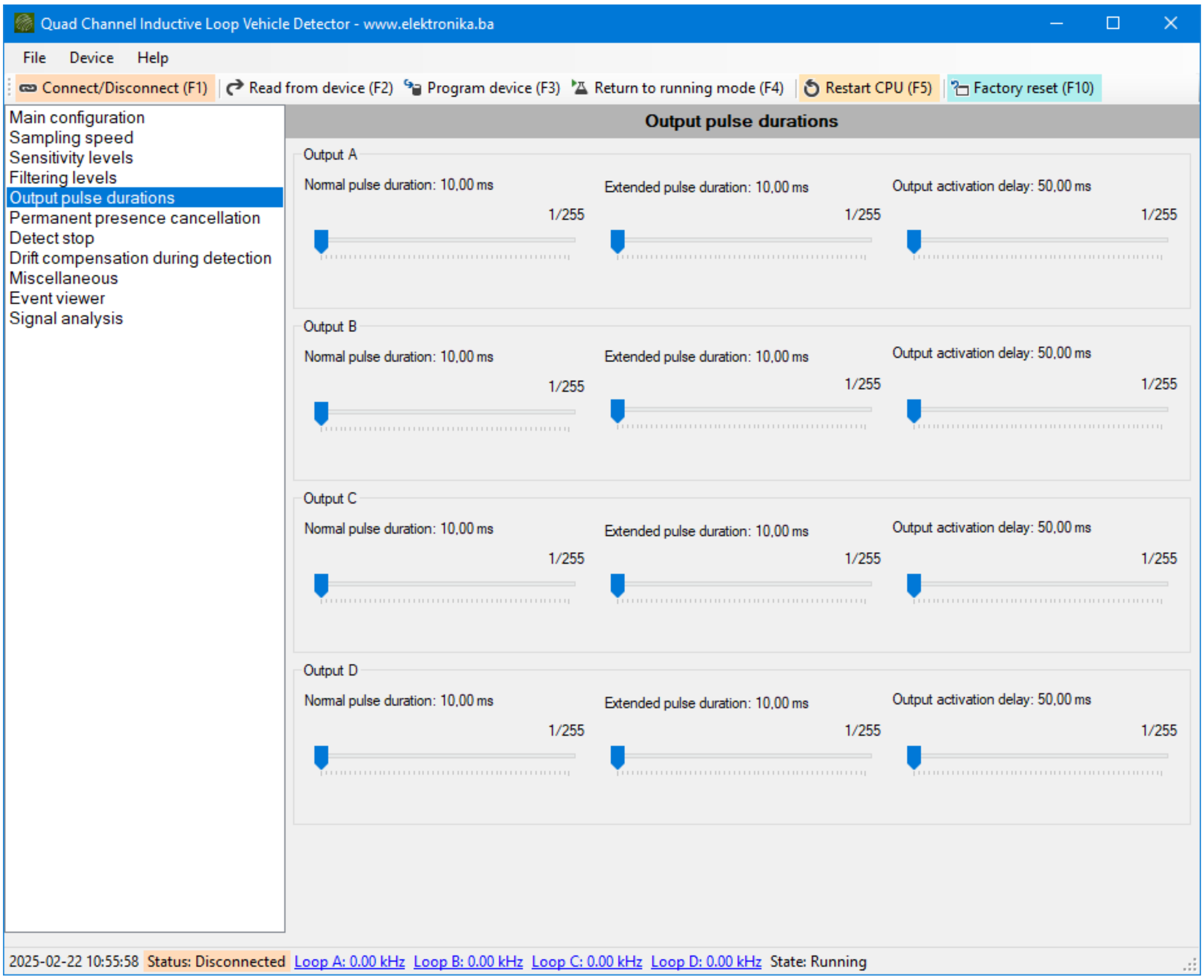

Figure 12 - Relay pulse durations

It is possible to delay the activation of each output with the "Output activation delay" slider. Selected values of these delays are utilized only in case "Presence Output" is enabled for that channel.

Important: Do not enable the option "Delay Output Activation" while "Presence Output" is disabled in *Main configuration* setting. In case this options is enabled while outputs are configured to pulse upon detection, the output might never be triggered.

Permanent presence cancellation

Sometimes it is required for device to cancel-out current detection in case vehicle parks over loop (or very close to it when sensitivity is set to maximum). The time after which device will cancel-out the detection can be fine-tuned in this section. After the device cancels-out detection of parked vehicle, it can now perform detections of other vehicles even though first vehicle has already occupied the loop (only in cases when loop is laid out so that it is possible additional vehicle to occupy it).

In order for device to cancel-out current detection, it uses parameters described in *Detect stop* section. Vehicle must be completely stopped over loop and only then the PPC timer starts the countdown. Each passing vehicle will reset this timer.

When originally cancelled-out vehicle leaves the loop, the device will recalibrate once again in order to start detecting new vehicles. For this to happen, there must be no passing vehicles over the loop for a short period of time. This time is set with "Detect leave parameters" in the same section as shown in the image bellow. This time is usually 1-2 seconds, and threshold is usually set to "3". Threshold has the same unit of value as sensitivity unit.

Note: As level of PPC can be selected from DIP switches, it is advisable to set shortest duration to first slider and maximum time to last slider.

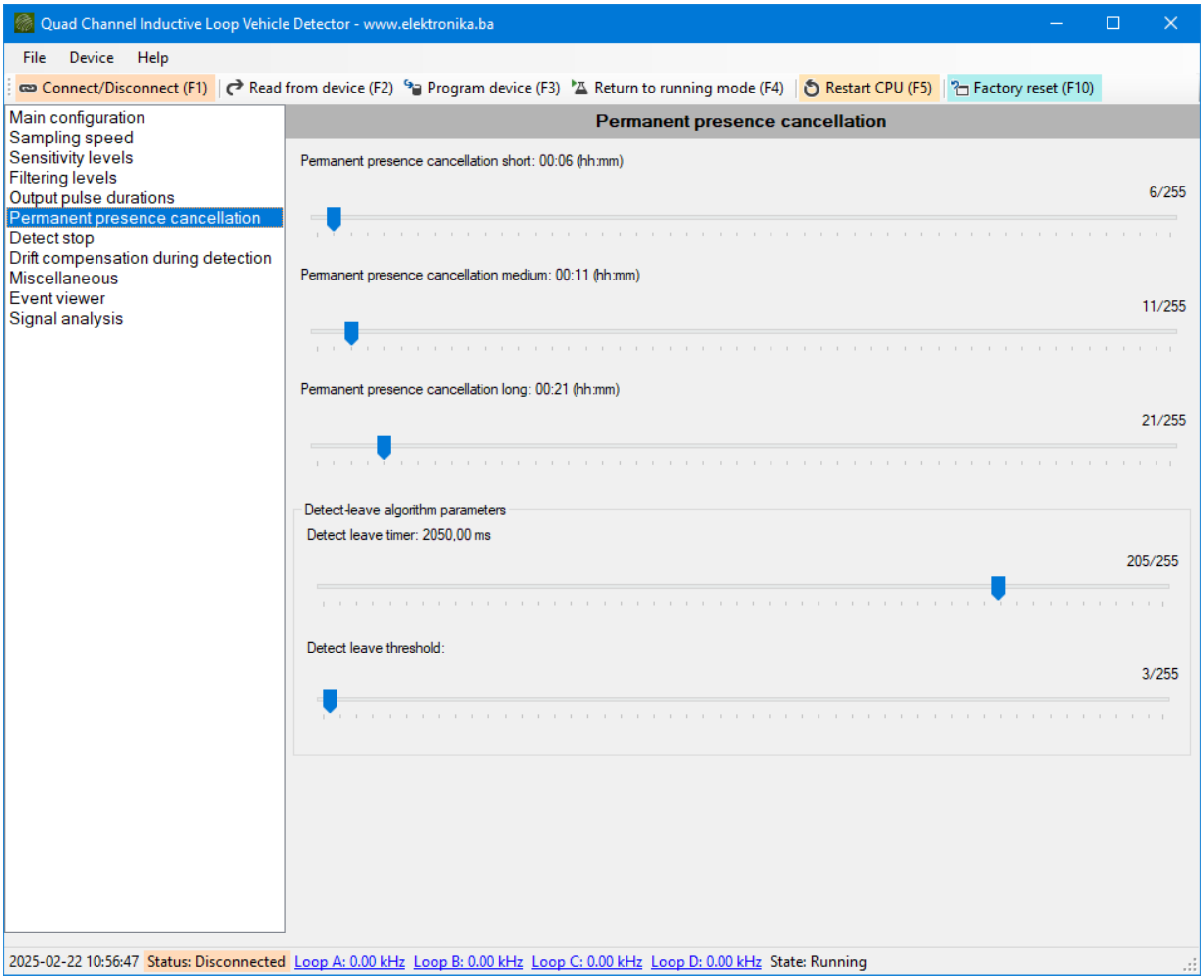

Figure 13 - Permanent presence cancellation

Detect stop

Stopped vehicle is a vehicle that did not produce the change in frequency of set threshold in some amount of time.

There parameters can be fine-tuned, but it is usually enough to have threshold set to “3” and timer to 1.5 seconds.

The “Slow check” timer should not be less than 0.5 s and it is used to slow down the calculations in order to better detect the change. This parameter can be experimented with, or left to default value.

Units of threshold parameter is the same as units of sensitivity.

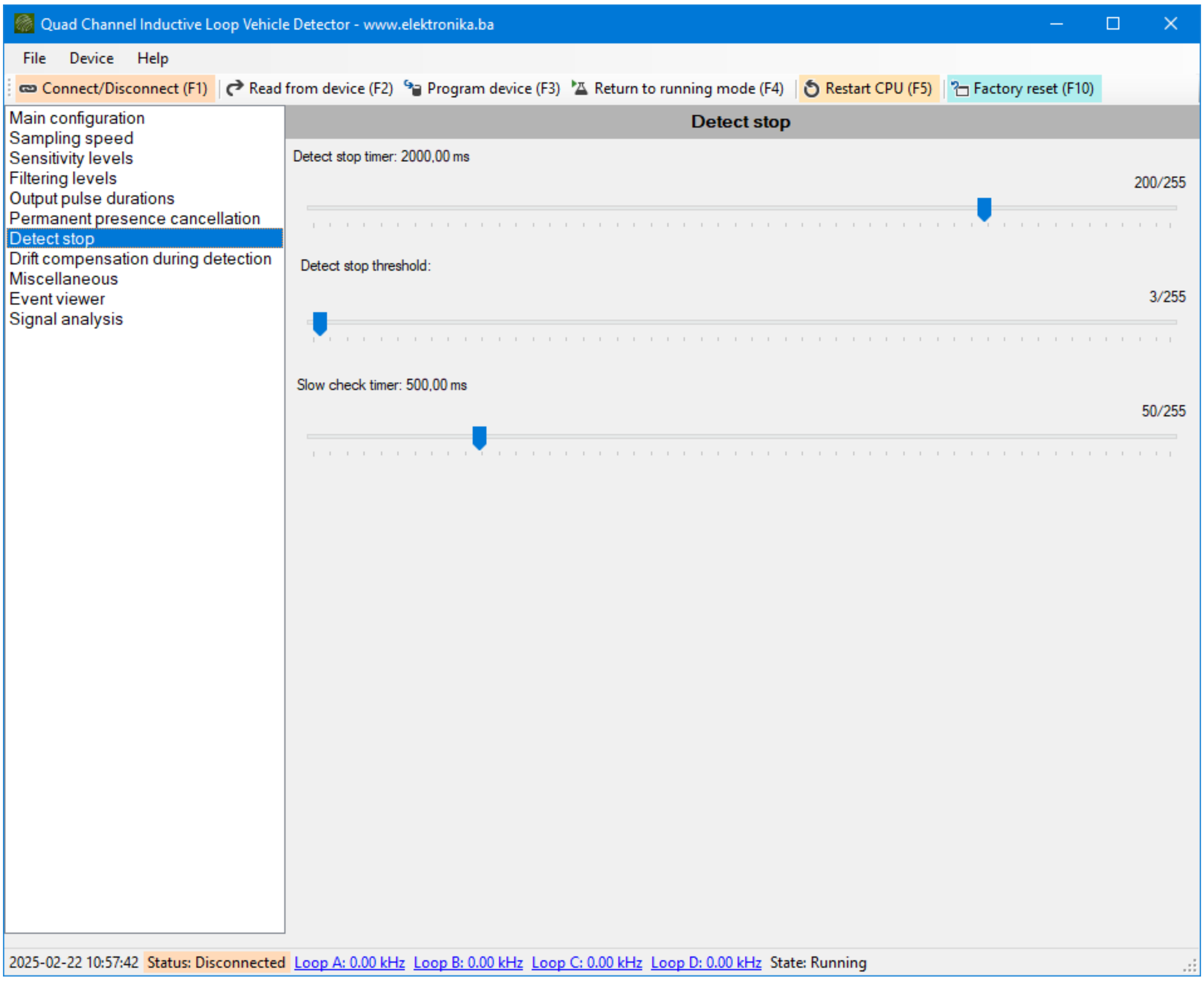

Figure 14 - Detect stop

Drift compensation during detection

As mentioned earlier in section *Filtering levels*, drift compensation is done also when vehicle is present over the loop. In order to start the tracking of baseline frequency during detection it is crucial to do this when vehicle has completely stopped over the loop to prevent compensation of vehicle movement. Parameters for detecting when vehicle has stopped over the loop should be separated from “Detect stop” parameters and threshold should be kept to a minimum, usually “2” with timer kept to larger value, usually 6 seconds.

Drift compensation during detection will start when device does not sense change of set threshold within set timer period. If device senses the change in frequency larger than set threshold, it will pause the drift compensation and restart it again after it detects that vehicle is not moving again.

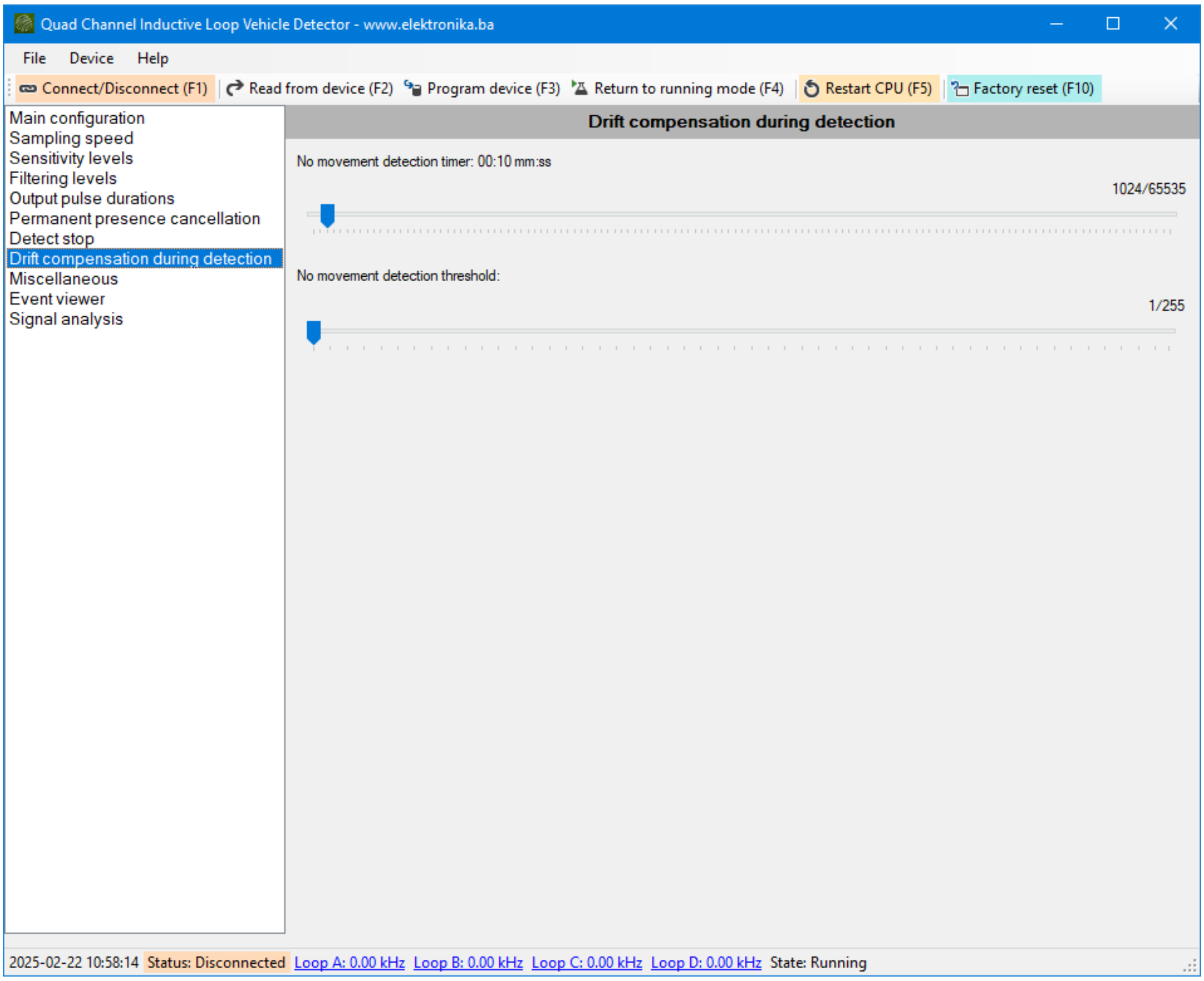

Figure 15 - Drift compensation during detection

Miscellaneous

Slow startup (warm up)

If the device has been installed where maximum detection sensitivity is required but filtering is at the lowest setting, it is recommended to activate this option which will upon startup set the sensitivity to the lowest bank (1) and then gradually (over time) increase the sensitivity to the required (wanted) bank on every timeout of this timer.

Runtime Re-Calibration delayer

Re-calibration of channels upon runtime error is constant until problem goes away but it also takes time for calibration to finish making other channels seem non-responsive. In case there is a loop failure on some channel but other channels should remain as responsive as possible, the recalibration of problematic channels can be delayed by this pre-set time.

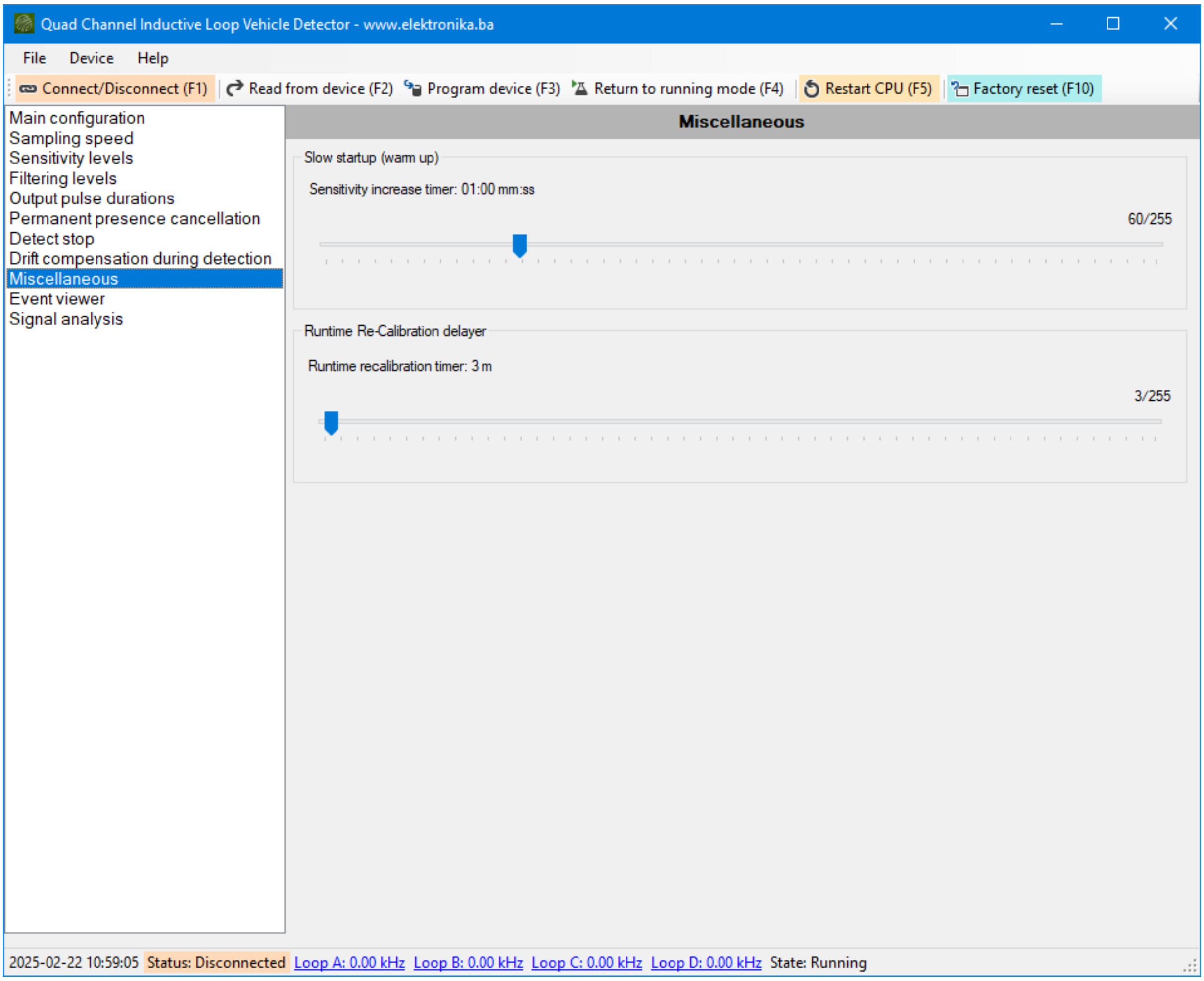

Figure 16 - Speed trap

Event viewer

In this section, device can be switched to “Event logging mode” where it will send various events during operation. List of events it can send is described in detail in *Table 2 - Event list*.

This screen parses those parameters and displays them in more human-readable format with timestamps where it is possible to save entire log into a file.

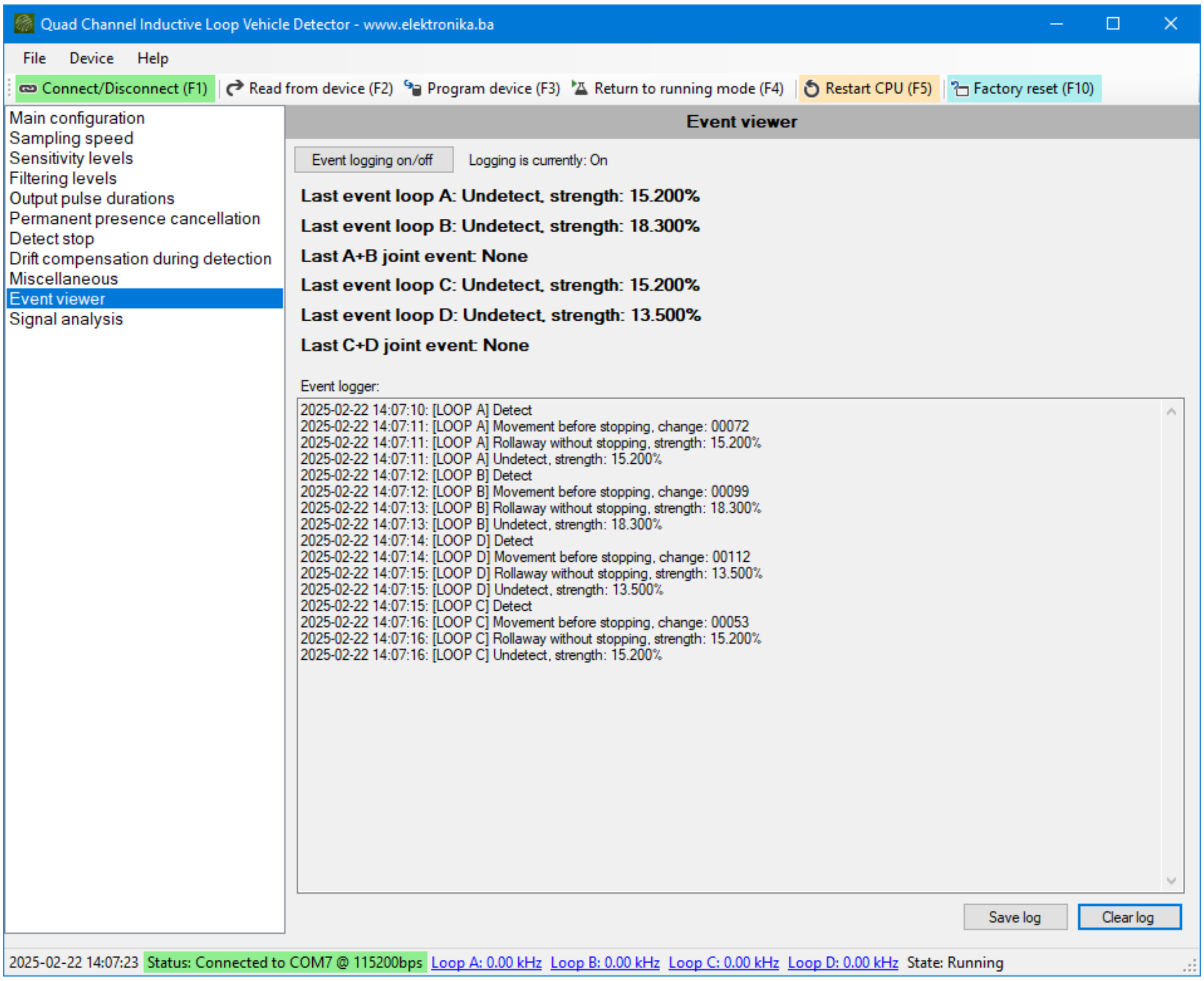

Figure 17 - Event viewer

Real-time frequency reporting

In this section, it is possible to enable "Real-time frequency reporting" (signal analysis) mode and view vehicle's magnetic signature when it passes over the loop. It is possible to save the plotted graph into file as image by clicking on it, and it is also possible to save the signal value dataset into text file for further processing. When "Auto save" checkbox is ticked, each detection (each new dataset) will be automatically saved into the last directory used for saving.

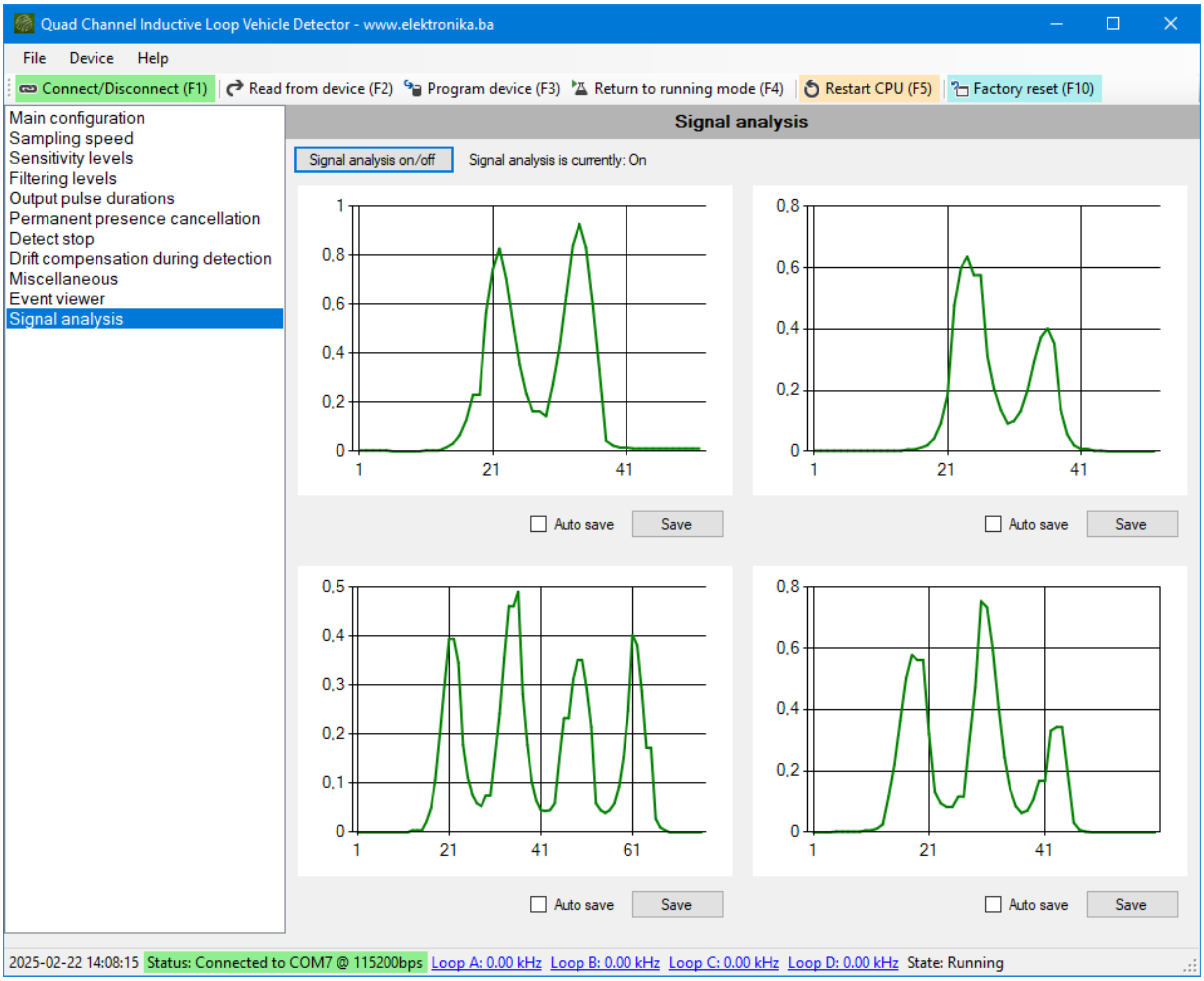

Figure 18 - Real-time frequency reporting

Document History

Date	Document version	Description
2025-02-22	2025-02-22-InitialRelease	Initial version created from the *Dual Channel PC Configurator* document.