LINlogger

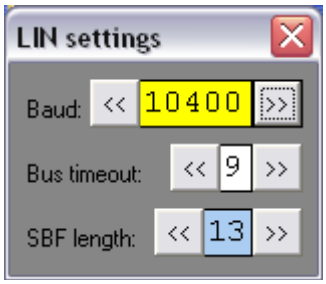
LIN Logger,
LIN Master,
LIN Slave

4800 – 20000 baud

In this short manual we will not touch questions about LIN network itself because all necessary documentation is freely available from official sources, it isn't a great secret. Our task is to explain how **LINlogger** works. Just some nuances and descriptions of functionality.

1. LOGGER

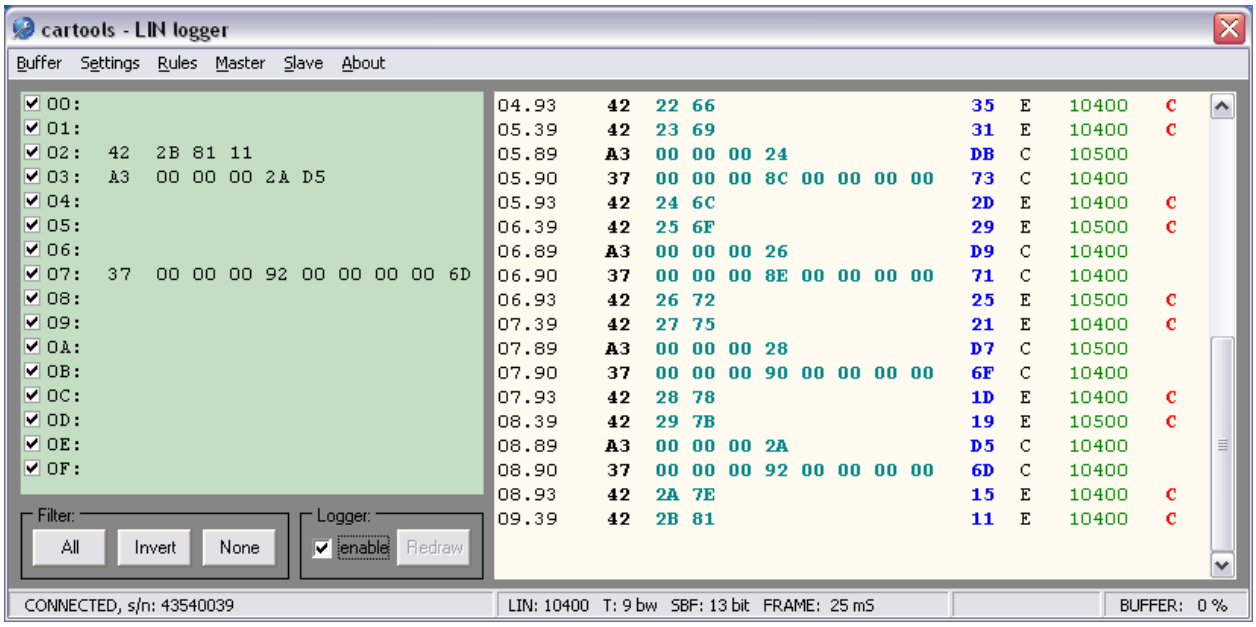
At first it is necessary to set correct LIN baud rate. To do that open **menu** → **settings**. Here you can choose baud rate, timeout, SBF length. It isn't necessary to close this menu item, just move it to any free space on desktop. It is possible to change parameters "on the fly" in case if you don't know exact values.



1. **Baud rate:** any valid baud rate value between 4800 and 20000. You can choose right one by pressing buttons (presetted values) or by entering it directly (it is validated and becomes active only after **enter** key is pressed).
2. **Bus timeout:** wait time, 1 unit = 1 byte length according to selected baud rate. How it works: if valid header already received (SBF, SYNC BYTE and at least 1 byte of data – ID) and nothing else happens (no answer from slave etc.), logger waits until timeout reached – if no more data on LIN. Then it assumes that there will be no more data and transmit data already existing to PC. This is in case of incomplete messages or messages where length is not full 8 bytes. Of course, if next transmission starts or 8 data bytes and checksum already received, frame is sent to PC immediately.
3. **SBF length** – according to LIN specification. It is significant only if LINlogger is working as master node, otherwise it doesn't matter which value used.

Now about **menu** → **Rules**. You can choose what rules are correct and applied, or you can choose to ignore them all. These rules are used to prepare messages when working as LIN

master or LIN slave mode (described later in this manual). Rules in case of logger affect only how messages are filtered and displayed. Message is received and displayed anyway.



If everything is set correctly and LINlogger already attached to working LIN network, LIN frames are displayed on screen.

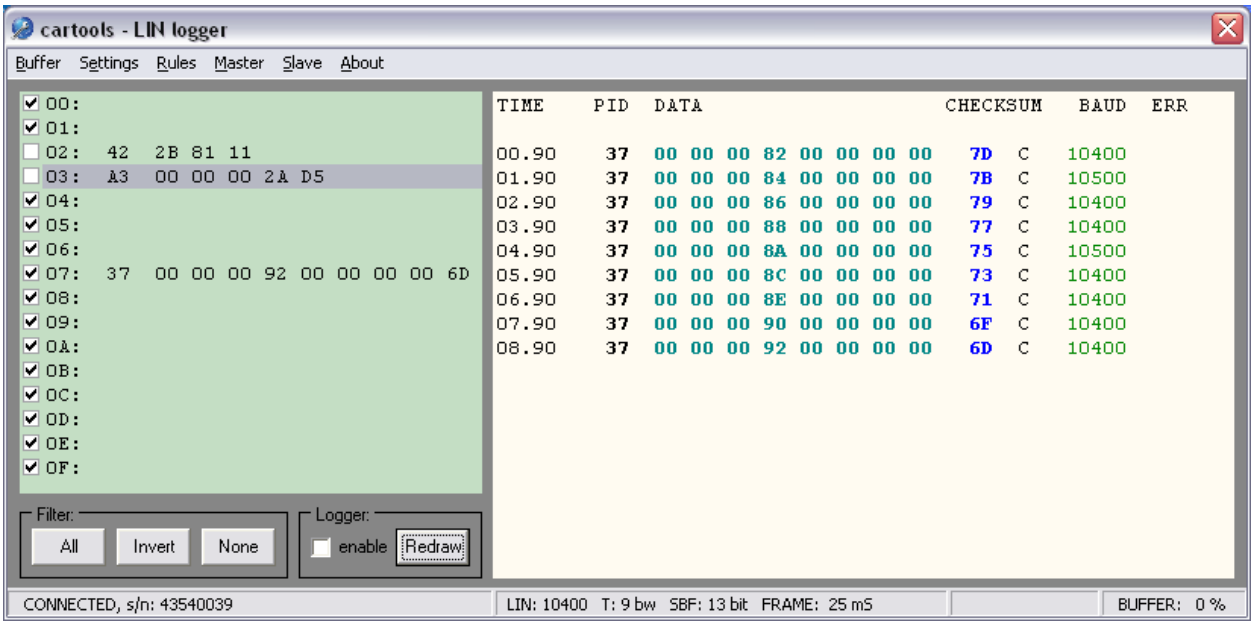
At the left side of main screen: acceptance filter. You can accept or decline messages with selected ID. Last message received with appropriate ID is displayed here (just for illustrative purposes to simplify selection). If check box near ID is unchecked, message is declined or filtered out. If LIN 1.3 rules set, selection from 16 possible ID values available. If LIN 2.x you will see 64, if ID rules ignored at all you can choose from 256 possible ID values. Example: LIN 1.3 selected, PID 37 (hex) is equal to ID 7 according to rules how LIN 1.3 PID is made. If check box is in unchecked state, all LIN messages with ID = 7 are rejected (for example, PID= 87 (hex), 47 (hex) etc).

On the right side – log window itself. Displayed: time stamp in seconds, message PID, up to 8 bytes of data, checksum, type of checksum (classic or enhanced, if match criteria, no matter which one is currently selected), baud rate (last 2 digits rounded down to 00) and errors if any (according to selected rules). Some notes and explanations:

1. Baud rate: LIN network can work even if it's clock significally differs from nominal value (deviation up to 15% allowed). Sometimes it is useful to know real baud rate on LIN bus.
2. Time stamp: set by PC when message is received from LINlogger. It isn't set by hardware at the start of each frame, so take care on accuracy. Value is provided for illustrative purposes only, in most cases it is more than enough.
3. Errors – checksum and PID both are verified according to rules. In addition for LIN 1.3 messages their length is checked for validity too. Of course, if PID is set to **any** and checksum to **ignore**, there will be no any error warnings displayed at all.
4. Received message is always displayed in black and white. Colors are applied by pushing Redraw button (enabled when logging is stopped). **Menu → buffer → use highlights** must be checked to show log screen in colors. Data received in buffer is completely redrawn, for larger data amounts it takes some time to complete.
5. If buffer is completely filled (100%) it is erased and filling start again from 0. Be careful, sometimes observe buffer fill in percents (status bar, right side). Buffer size is 65535 records, more than enough in most situations.

If logger is enabled messages are accepted and stored into buffer according to acceptance filter rules.

When logger is stopped, you can apply filter rules to already received messages. Choose ones you wish to see and press **Redraw** button. Filtered messages are not deleted from buffer, just hidden. Example:



It is possible to save job results in 3 different ways:

1. ***.rtf** – rich text format – text file with colors and highlights,
2. ***.txt** – simple text file,
3. ***.bin** – binary buffer.

It is possible to load *.bin files only (previously saved message buffer).

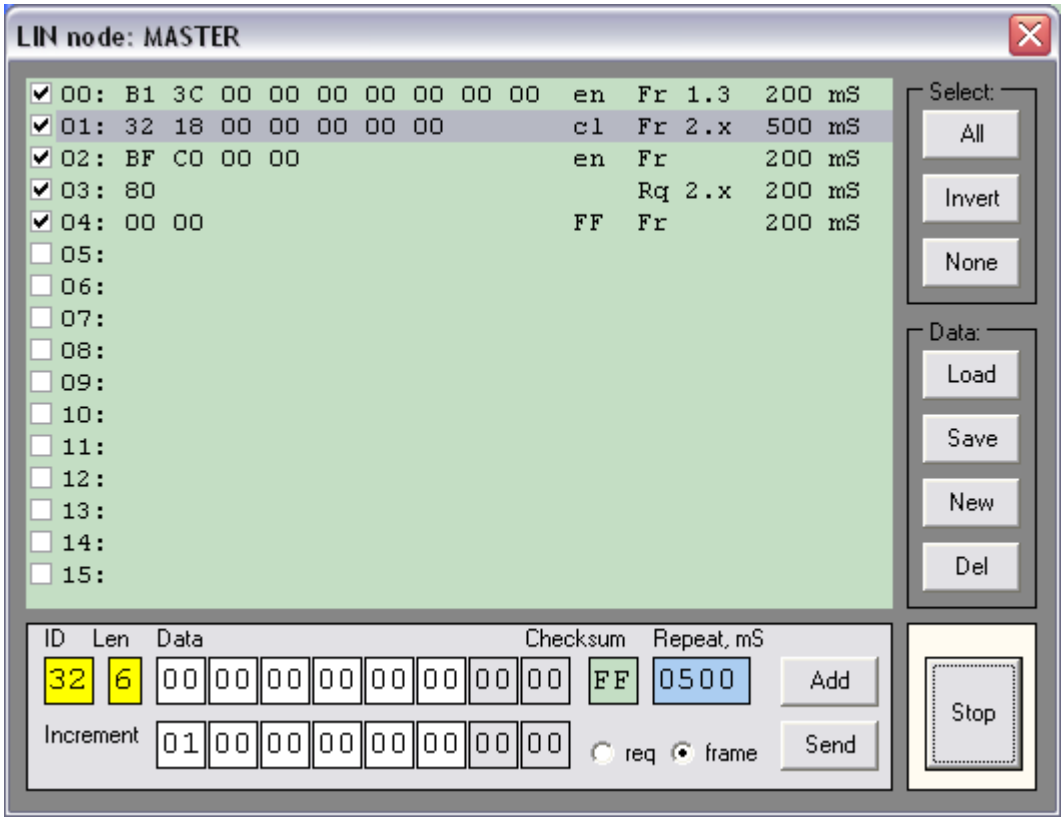
2. LIN MASTER

It is possible to send a single message to the LIN and / or periodic messages in a predetermined time intervals and using different individual message rules if necessary. Rules are defined in the main menu of LINlogger, currently active rules are applied when message is created / edited and added to list. In the example below messages were created with the following rules:

00: PID: LIN 1.3, checksum: enhanced,
01: PID: LIN 2.x, checksum: classic,
02: PID: any, checksum: enhanced,
03: PID: LIN 2.x, data request from slave,
04: PID: LIN 2.x, checksum: any (manual / forced checksum used, 0xFF here).

Each message is validated according to rules selected, invalid PIDs and data lenghts are corrected automatically. In case of LIN 1.3 or LIN 2.x PID is corrected, if LIN 1.3 selected message length is corrected too.

Increment – data bytes are incremented by increment value after each successful transmission. For example, if increment of byte is set to 0x02, start value is 0x00, this selected byte in 1st message will be 0x00, 0x02 in 2nd, 0x04 in 3rd etc.



Repeat – message is periodic if repeat value is not 0. Otherwise message is sent only once.

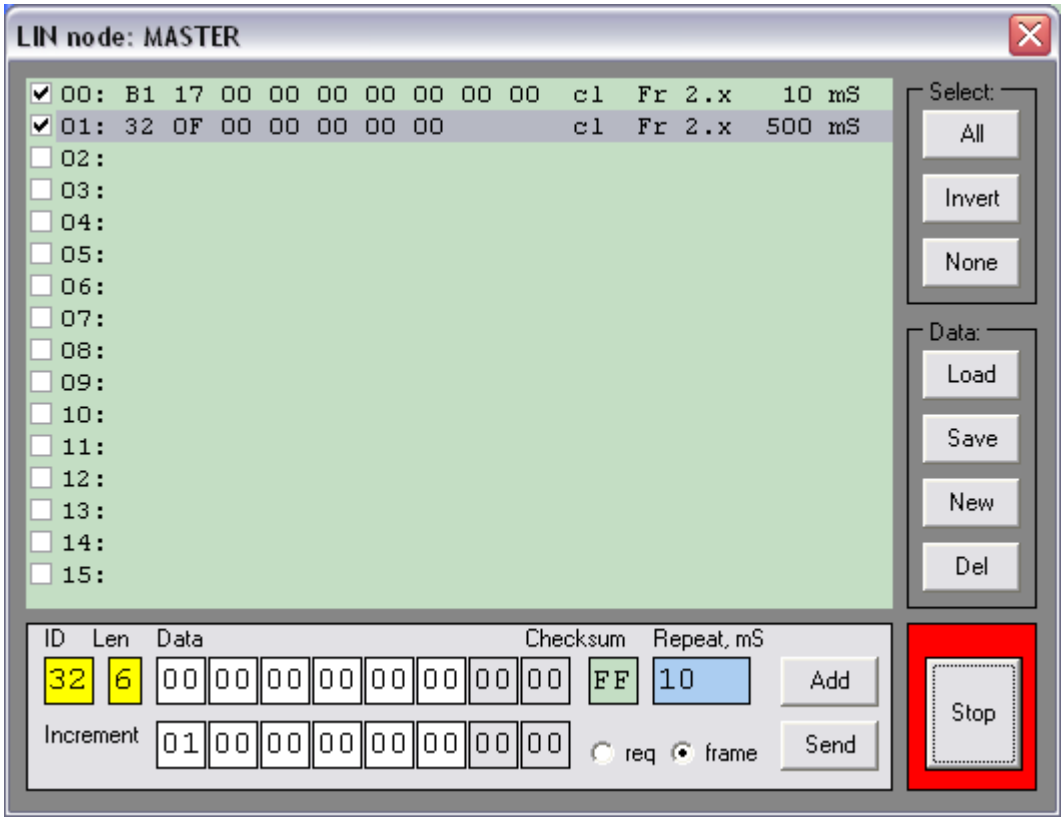
Messages are added to list by pushing button **Add**. It is possible to edit already created messages too. When valid message is selected from list, message data is transferred to editor fields. If empty space is selected data isn't updated, it is possible to transfer messages to empty spaces by selecting them before **Add** is pushed. Be careful because currently active rules applied when transferring edited message back to list or to empty space. No matter what rules was in use when message initially was created.

Transfer starts immediately when button **Run** is pushed. Panel around button changes its color to white (like in screenshot above, **Run** becomes **Stop**). It is possible to transfer single message from editor fields to LIN too (button **Send**). In this case data are put on to queue and transferred at first available time window (listed messages have higher priority).

Listed messages are transferred in order starting from 00 up to 15. Only checked messages are transferred. Broadcast message is updated according to increment values if any.

It is important not to overload message queue – if too much messages selected with small repeat intervals, overrun is possible. Messages are transferred with increased time intervals. If overrun occurs panel around Run/Stop button lights red. Frame window interval is displayed in main form status bar - FRAME xx mS. It depends on currently active LIN baud rate setting and is equal to time necessary for transmit of SBF + SYNC + ID + 8 bytes of data + checksum +

some extra time for interframe space and delay between master request and slave response if any.

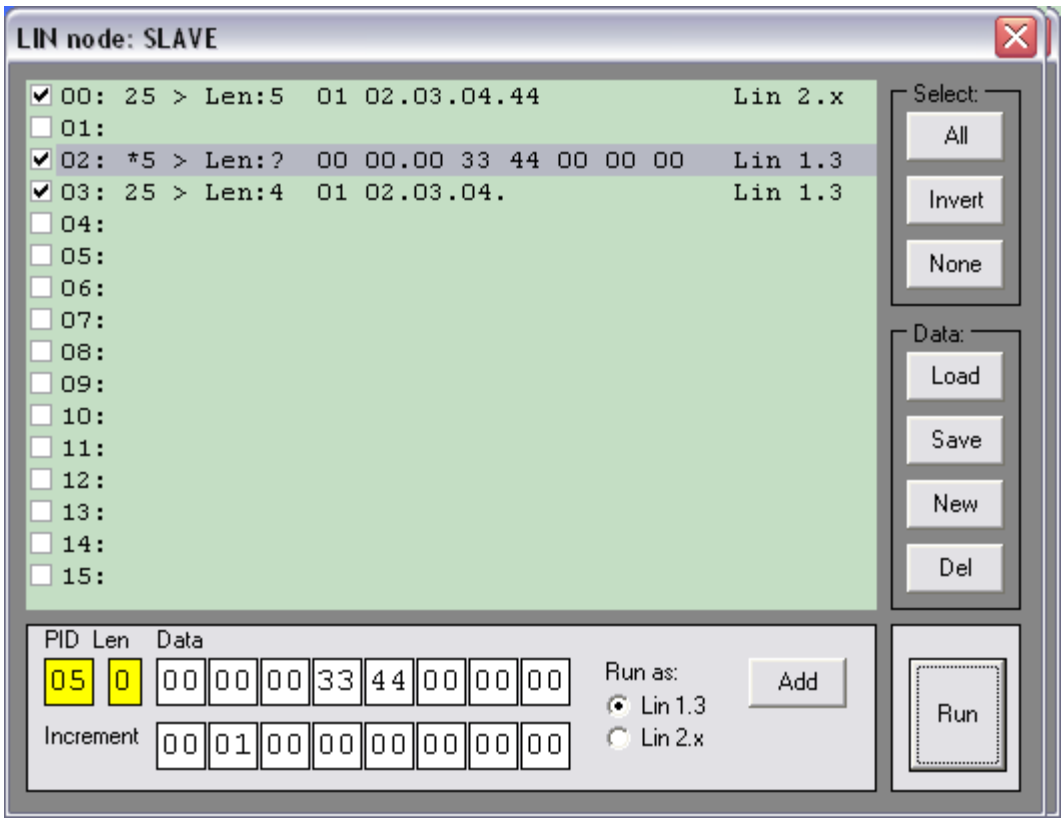


Logger screen shows real situation on LIN bus anyway.

3. LIN SLAVE

Slave responses are configured in similar manner like master messages. Some differences and specifics: checksum is always applied to message according to message type. If message is configured as LIN 1.3 message, classic checksum takes place. If LIN 2.0 enhanced checksum used. In addition:

1. If LIN 1.3 and message length is set to 0, data field length is calculated according to PID received from master (message listed in line 02, example below).
2. Point after data means that increment exist for this byte. Data shown in list is not updated to currently actual values according to increment, you can observe what happens on LIN bus by examining logger screen only. If message is disabled and then enabled again, data bytes are reset to their initial values.
3. If more than one rule match criteria, one listed first (from line 00 up to 15) is executed. In example below all slave responses are valid if ID 25(hex) received from master. But only 00 executed.
4. When creating messages select message type first (LIN 1.3 or LIN 2.x) just to avoid mistakes. That's because PID and length both are validated according to message type.



4. Hardware

There are 2 status LED's:

RED: activity on LIN bus

GREEN: pullup to +12v is switched on

Pullup is turned on automatically when transferring first message to LIN (LIN master mode) and remains on until LIN master window is active.

It is important to understand that LIN bus supports multiple slaves, but only **one master**!

Device drivers for PC: standard D2XX drivers from FTDI, version 2.08.xx or above.

cartools