BOUNDARY CONDITION

Chrysler CCD/SCI Scanner User's Guide

V1.44

Last update: 2020.09.01

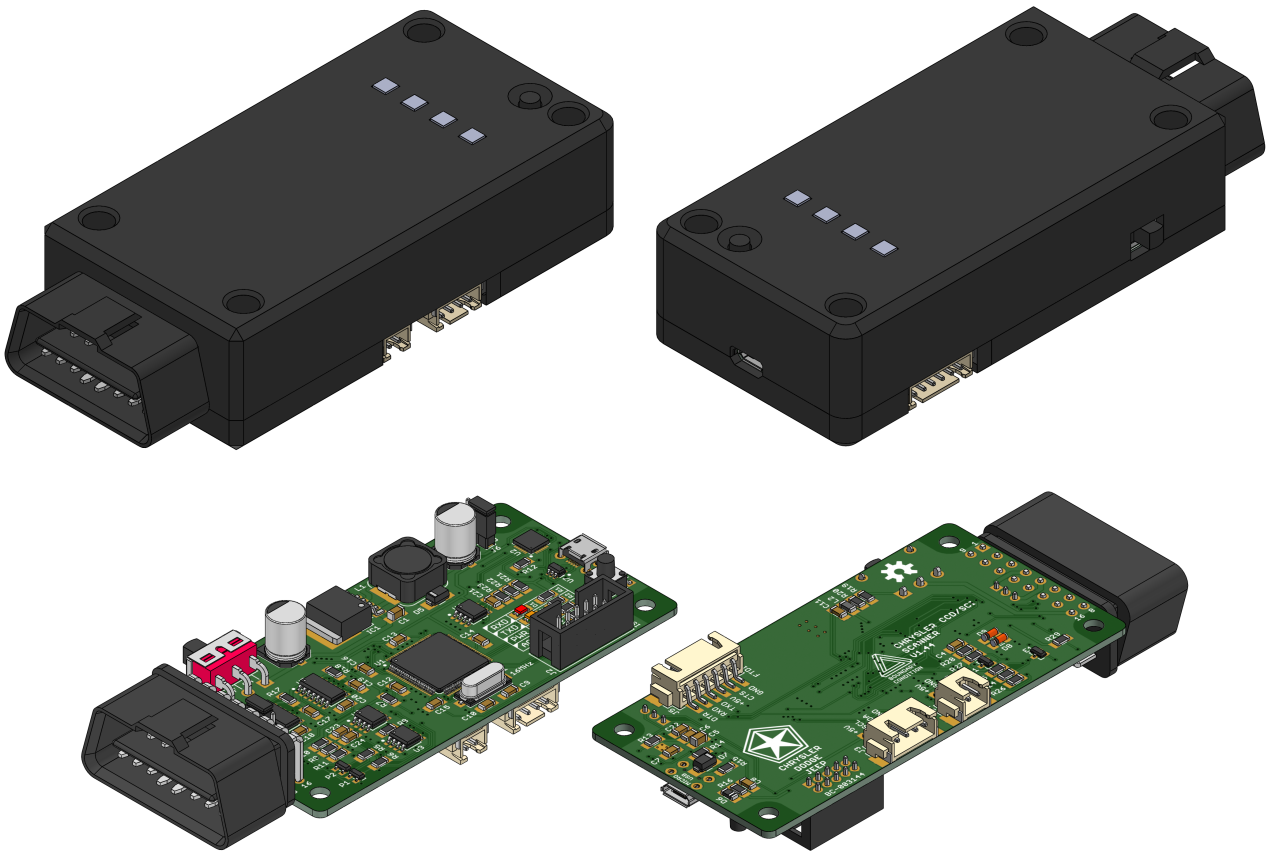


Figure 1. CAD drawing of the diagnostic scanner.

1. Features

- Send and receive diagnostic messages to and from CCD-bus and SCI-bus.
- Measure battery voltage directly on the OBD-16 pin.
- I2C LCD connectivity for 16x2 or 20x4 character LCD.
- Micro USB, FTDI and ISP-10 connectivity for communication and programming.
- Vehicle coverage: 1983-2004 equipped with CCD/SCI-bus.

2. Applications

- Read diagnostic data, read/erase fault codes, perform actuator tests.

3. Description

This device is a DIY diagnostic tool for older Chrysler/Dodge/Jeep passenger vehicles and other vehicles that use Chrysler's proprietary CCD-bus and/or SCI-bus to communicate with the outside world.

The scanner can be directly connected to an OBD2 female connector or 2 kinds of OBD1 connectors using adapter cables.

At its heart the scanner is a custom Arduino Mega board with circuits to interface with said buses. It is fully compatible and programmable with Arduino IDE via Micro USB, FTDI or ISP-10 AVR programmer.

The Windows GUI makes it easy to interface with the scanner and thus your vehicle. You can listen passively to all communications between modules or send request messages.

4. Peripherals

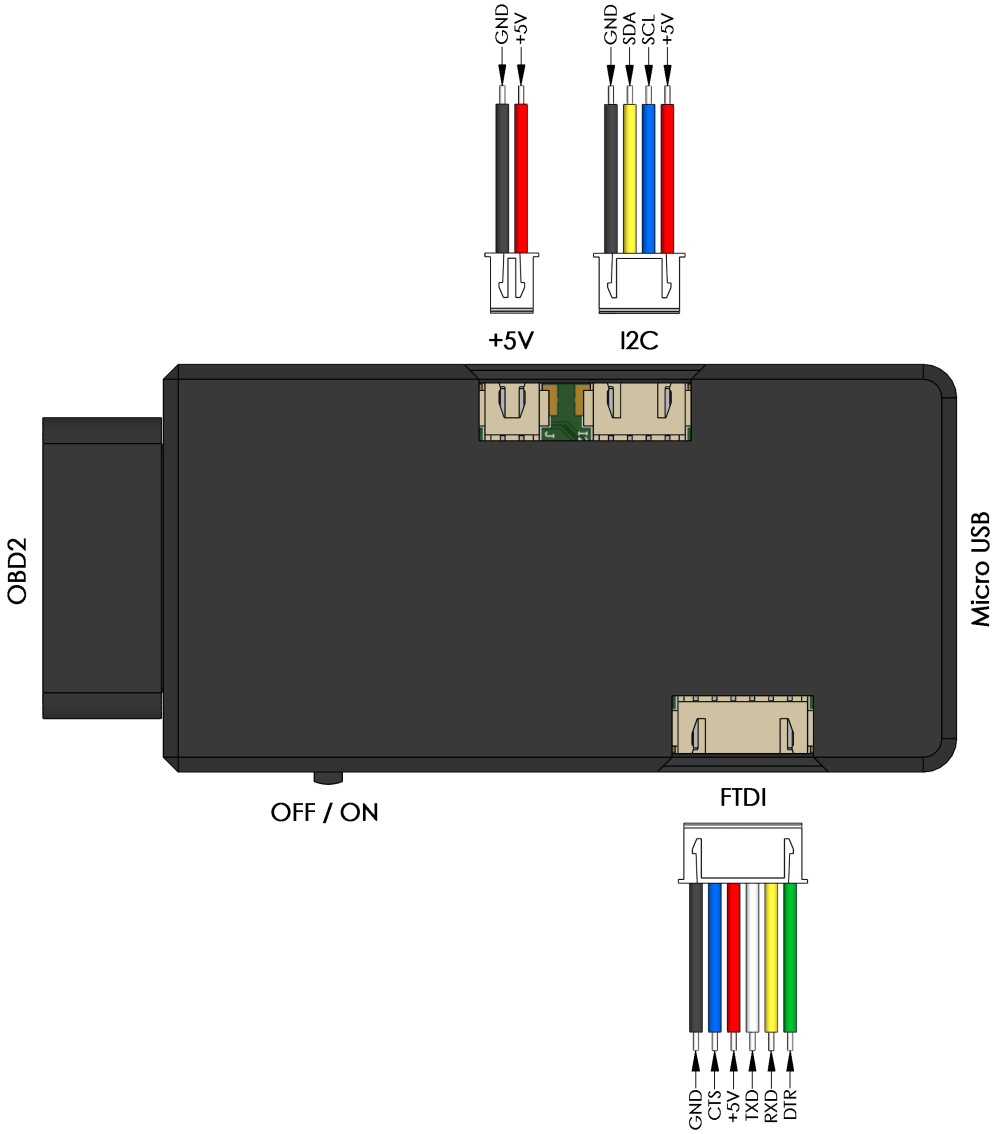


Figure 2. Connector description.

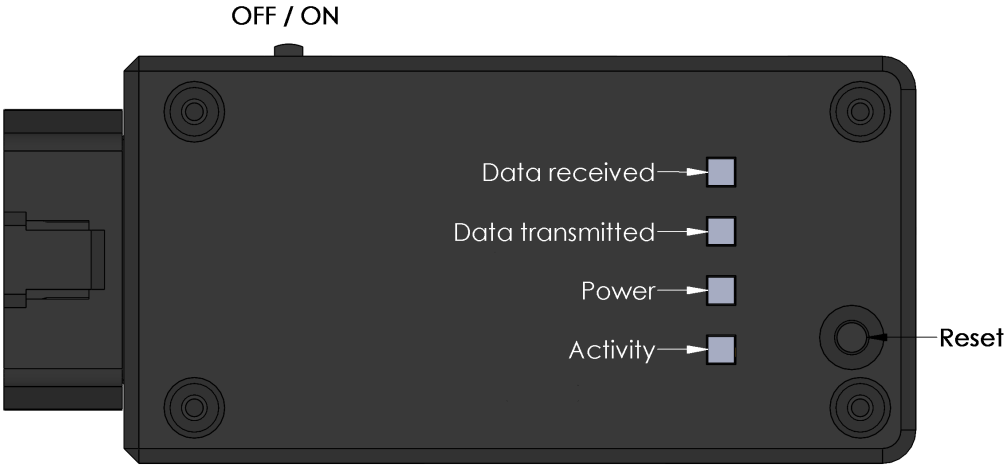


Figure 3. Status LED description and reset button location.

4.1 OBD1 compatibility

Pre-1996 vehicles have non-standard diagnostic connectors. There is one for CCD-bus located inside the vehicle around the driver's side, and another one for SCI-bus in the engine compartment.

4.1.1 Body connector (CCD-bus)

Figure 4. Two types of body connectors.

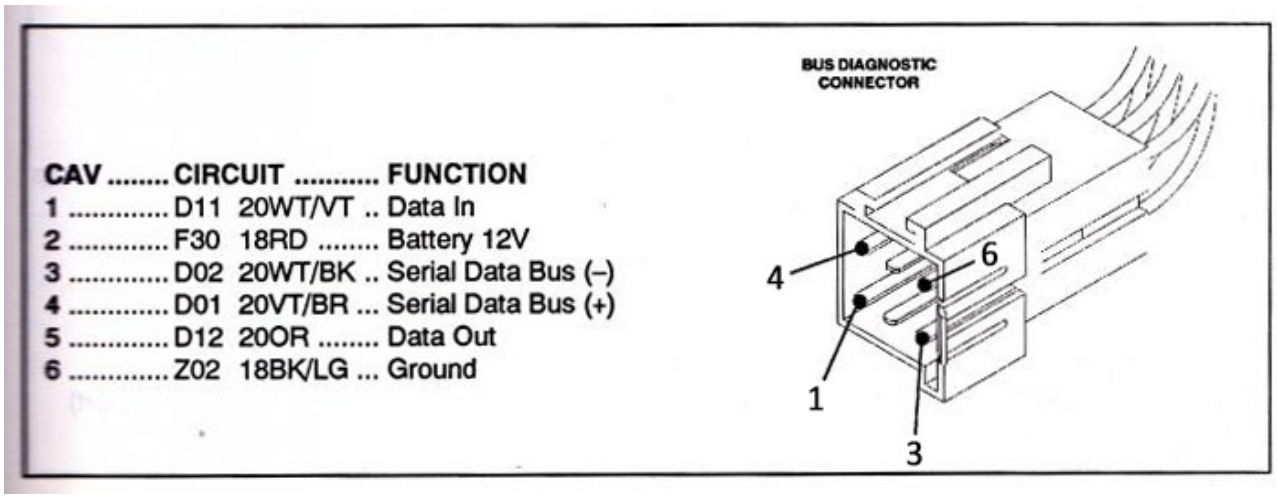


Figure 5. Body connector pinout.

Data In (1) and Data Out (5) pins are SCI-bus pins not related to the engine controller.

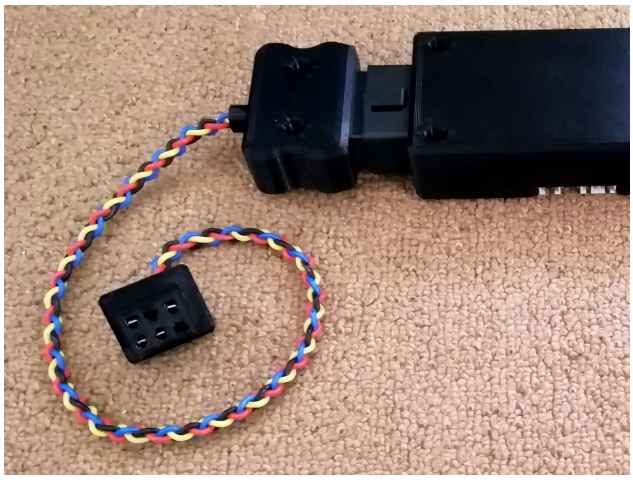


Figure 6. Body connector adapter cable connected to the scanner.

4.1.2 Engine connector (SCI-bus)

Figure 7. Engine connector (SCI-bus).

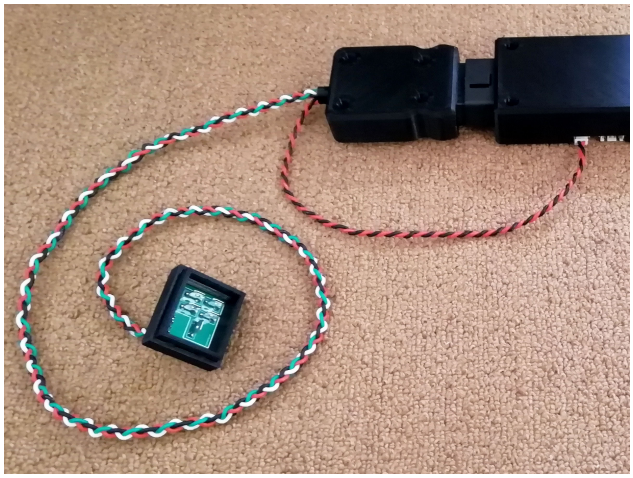


Figure 8. Engine connector adapter cable connected to the scanner.

The adapter cable has an internal circuitry for UART logic inversion which needs 5V power supply. This power comes from the scanner through the 2-pin connector.

The UART logic inverter can be disabled inside the connector casing but it's rarely necessary.

5. Windows GUI description

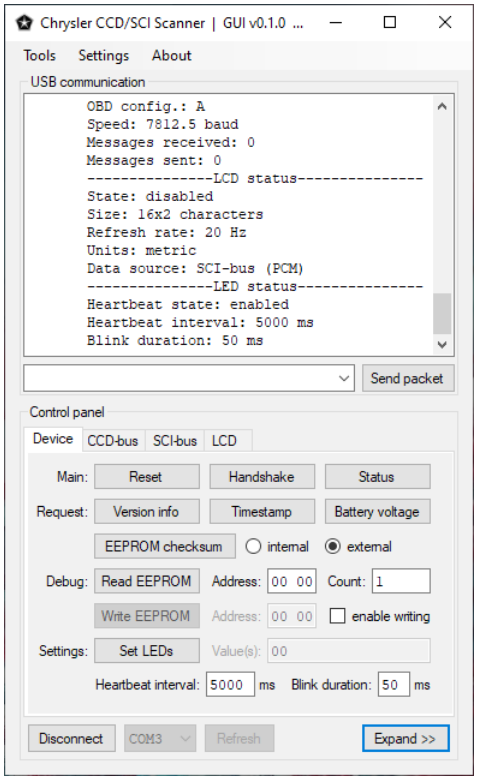


Figure 9. Windows GUI collapsed view.

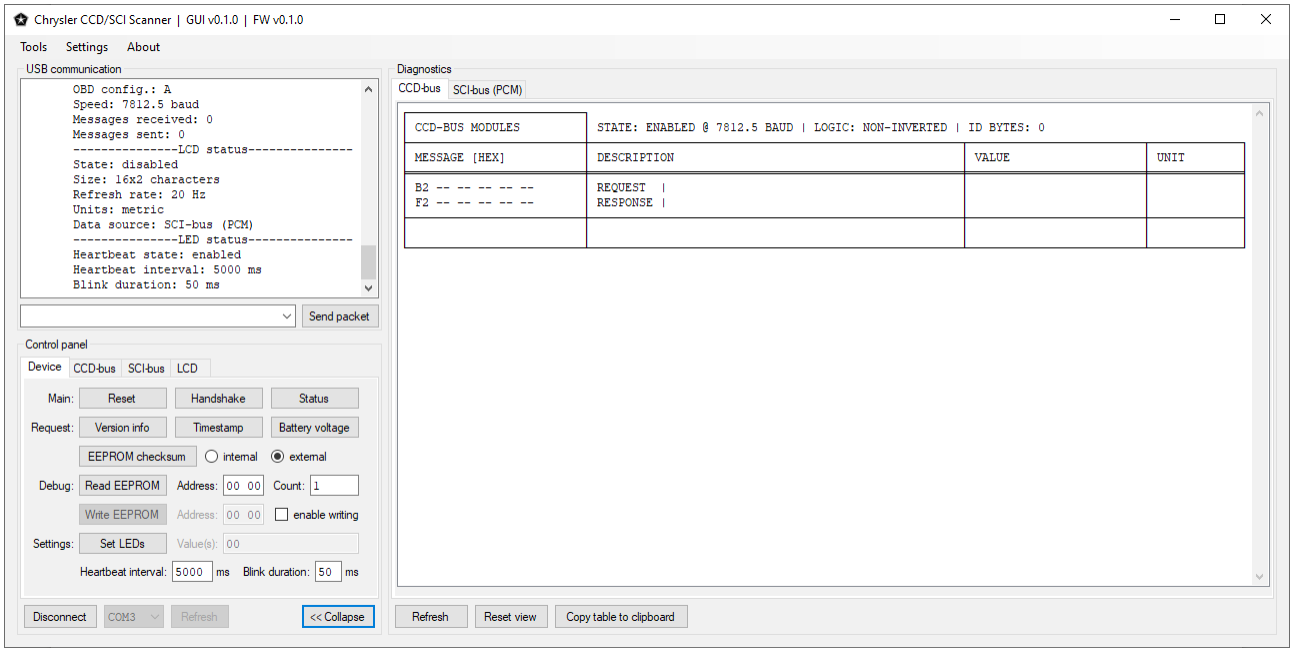


Figure 10. Windows GUI expanded view.

Menu strip:
- Tools: update GUI and scanner firmware.
- Settings: units (imperial / metric), include timestamp in log files.
- About.

The user interface is separated into 3 main groups:
- USB communication: all raw and interpreted USB data is displayed here.
- Control panel: user controls for interacting with the scanner.
- Diagnostics: diagnostics tables are displayed here.

5.1 USB communication group

The USB communication between a host computer and the scanner is displayed in the text box and it is followed with interpretations when applicable. All messages appearing here are saved in separate log files.

Using the combo box below custom communication packets can be sent to the scanner, although not recommended to use it without prior knowledge. For further information about frame formats refer to the UART Protocol PDF.

5.2 Control panel group

Separate tabs contain controls to interact with various parts of the scanner.

In disconnected state select the correct COM-port and click on the Connect button. The GUI tries 5 times to request handshake from the device on a given COM-port before giving up. Click on the Refresh button when no COM-ports are visible.

If the user hits the Enter key while the cursor is in a text/combo box, then the appropriate command is executed, as if the associated button had been clicked.

5.2.1 Device tab

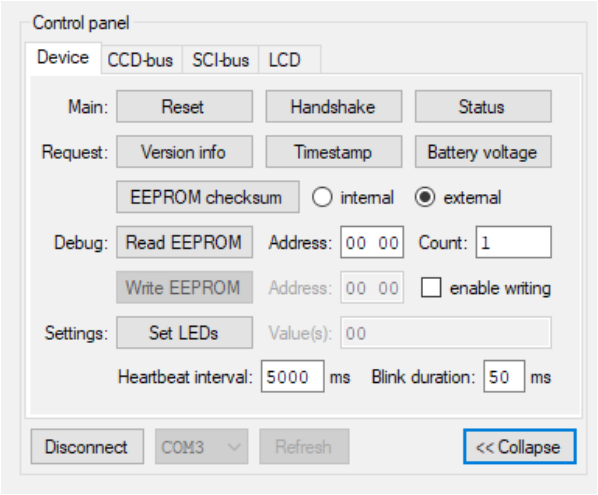


Figure 11. Controls on the device tab.

Reset: reset scanner.

Handshake: get scanner handshake.

Status: get scanner status.

Version info: request hardware/firmware information.

Timestamp: request timestamp.

Battery voltage: request battery voltage measurement.

EEPROM checksum: check if the internal/external EEPROM checksum is correct.

Read/write EEPROM: read/write EEPROM content at various addresses.

Set LEDs: configure LED heartbeat interval (blue) and blink duration (all LEDs).

5.2.2 CCD-bus tab

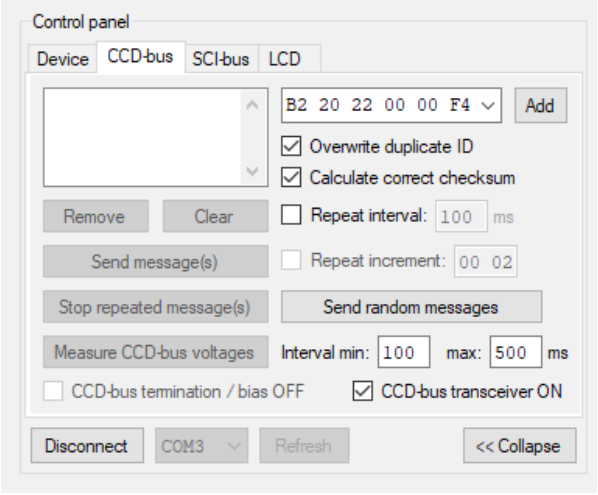


Figure 12. Controls on the CCD-bus tab.

Add: insert message(s) from the combo box to the message list box. Multiple messages can be inserted in this list box by separating them with commas. By double clicking on a message it is copied back to the combo box where it can be edited. To finish editing click on the Edit button. With a single message in the combo box when the enter key is pressed, it gets sent on the CCD-bus without considering the content of the message list box (quick send).

Remove: remove selected message(s) from the message list box.

Clear: remove all messages from the message list box.

Send message(s): broadcast all messages in the list box on the CCD-bus.

Overwrite duplicate ID: when a message with the same first byte is already present in the message list box then it gets replaced with the one being added.

Calculate correct checksum: all CCD-bus messages are error-checked by other modules. It is important that the last byte of a message is correctly calculated. With this option the GUI replaces the last placeholder byte of the message with the correct checksum. The scanner itself also does a checksum calculation before broadcasting a message on the bus.

Repeat interval: the scanner is capable of repeating a single or set of messages. With this option the delay between messages can be configured. Uncheck this option if there's no need to repeat message(s). In this case the message list is sent once.

Repeat increment: the scanner is capable of repeating a single message with the condition that for every repeat this value is added to the message. This advanced option can be used to sweep through RAM/ROM addresses. For further information refer to the UART Protocol PDF. Uncheck this option if there's no need for repeated message increments.

Stop repeated message(s): repeated message transmission can be stopped with this button.

Send random messages: with this debug option random 3-6 bytes long CCD-bus messages can be generated for internal testing. Minimum and maximum delays between random messages can be configured in the two text boxes below the

button. It is not recommended to use this feature when the scanner is connected to a live CCD-bus network.

Measure CCD-bus voltages: the V1.44+ hardware can measure CCD-bus voltages for basic fault diagnosis purposes. This button is inactive below V1.44 hardware version. Nominal CCD-bus voltage is 2.50V with a few hundreds of millivolts swings when 0 data-bits are written on the bus.

CCD-bus termination and bias ON/OFF: the V1.44+ hardware has a programmable CCD-bus termination and bias resistor network switch. This button is inactive below V1.44 hardware version, the resistor network has to be manually jumpered inside the box.

CCD-bus transceiver ON/OFF: message transmission and reception can be temporarily enabled or disabled with this option.

5.2.3 SCI-bus tab

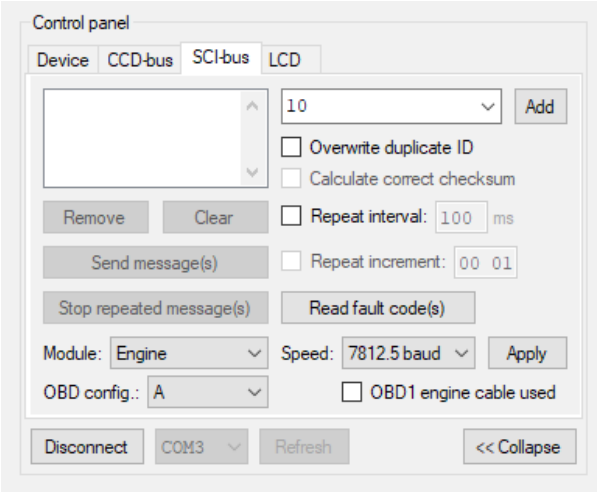


Figure 13. Controls on the SCI-bus tab.

Add: insert message(s) from the combo box to the message list box. Multiple messages can be inserted in this list box by separating them with commas. By double clicking on a message it is copied back to the combo box where it can be edited. To finish editing click on the Edit button. With a single message in the combo box when the enter key is pressed, it gets sent on the SCI-bus without considering the content of the message list box (quick send).

Remove: remove selected message(s) from the message list box.

Clear: remove all messages from the message list box.

Send message(s): broadcast all messages in the list box on the SCI-bus.

Overwrite duplicate ID: when a message with the same first byte is already present in the message list box then it gets replaced with the one being added.

Calculate correct checksum: transmitted SCI-bus messages are not checksum protected. When fault code list is received a checksum byte is appended to it by the engine controller which can be either verified or ignored. The GUI verifies this checksum byte automatically.

Repeat interval: the scanner is capable of repeating a single or set of messages. With this option the delay between messages can be configured. Uncheck

this option if there's no need to repeat message(s). In this case the message list is sent once.

Repeat increment: the scanner is capable of repeating a single message with the condition that for every repeat this value is added to the message. This advanced option can be used to sweep through RAM/ROM addresses. For further information refer to the UART Protocol PDF. Uncheck this option if there's no need for repeated message increments.

Stop repeated message(s): repeated message transmission can be stopped with this button.

Read fault code(s): shortcut for sending the fault code request byte (10) in low speed mode.

Module: select which module on the SCI-bus the scanner should talk to. Only the engine controller is supported for now by software. Transmission controllers use CCD-bus for diagnostics purposes and keep SCI-bus for re-flashing/programming, which function this scanner does not support.

OBD configuration: OBD2 connectors have different pinouts for SCI-bus throughout the years. "A" and "B" settings route communication internally to different OBD2 pins. The OBD1 engine adapter cable routes the wires to the default A-configuration (6:RX, 7:TX).

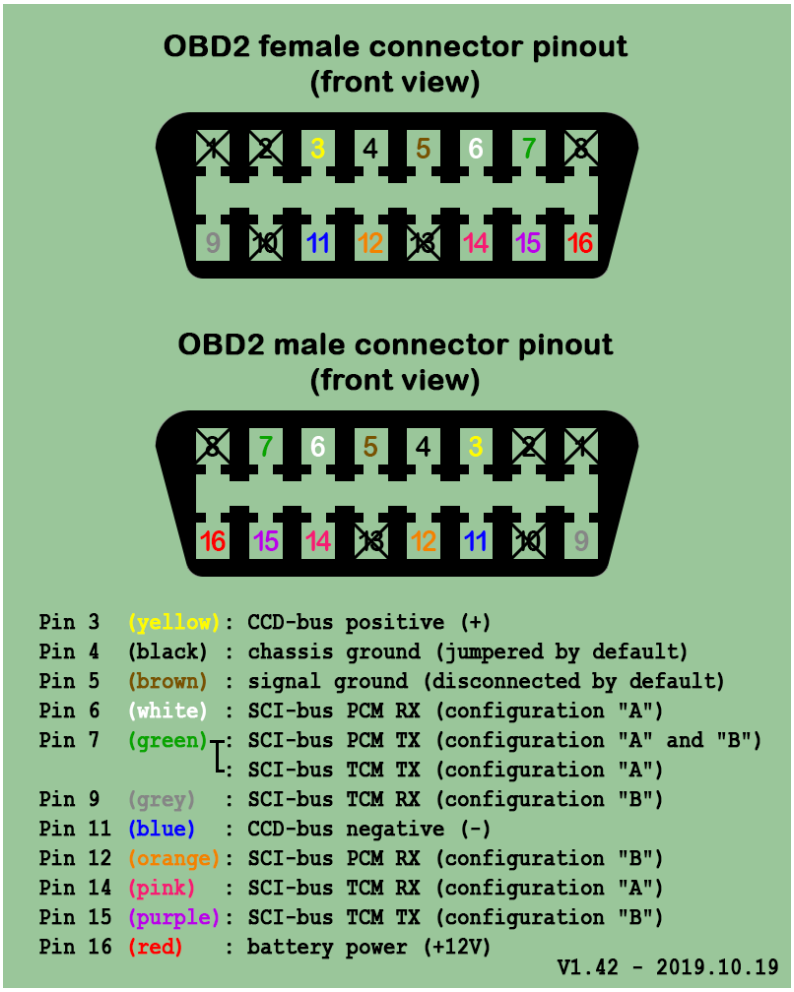


Figure 14. OBD2 connector pinout.

Speed: baudrate at which the engine controller communicates. The default speed is 7812.5 baud after reset or when the key is cycled in the ignition switch. This mode is called low-speed diagnostic mode.
When the speed is set to "off" the SCI-bus transceiver circuit is temporarily disabled.

OBD1 engine cable used: this option makes the scanner compatible with the inverted SCI-bus logic used by OBD1 vehicles.

Apply: update module, OBD configuration, speed and logic settings in the scanner.

5.2.4 LCD tab

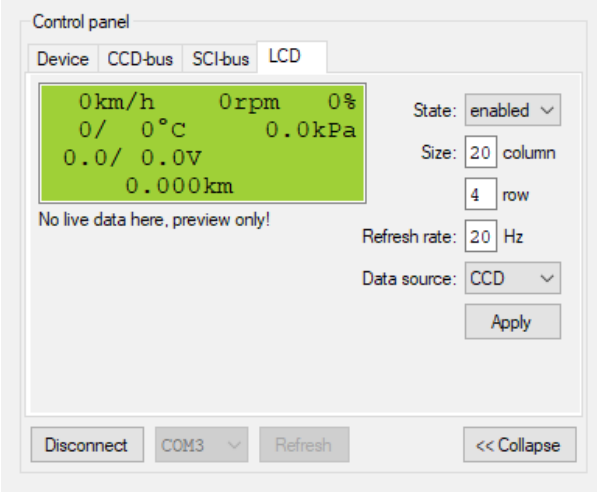


Figure 15. Controls on the LCD tab.

State: enable or disable data printing on the LCD. When disabled only the splash screen is printed. Otherwise a basic parameter layout appears on the LCD containing:
- vehicle speed,
- engine speed,
- throttle position,
- engine coolant and ambient temperature,
- intake manifold absolute pressure,
- battery and charging voltage,
- odometer value.

Size: arbitrary LCD size can be configured here but only 16x2 and 20x4 LCDs with I2C backpack (Arduino style) are supported for now.

Refresh rate: ~~number of prints per seconds. Data coming in between refreshes are not buffered, therefore lost. Character LCDs are slow, 20 refresh per second is an acceptable rate.~~ Disabled for the time being.

Data source: select CCD-bus or SCI-bus PCM for data source.

Units can be set in the Settings / Unit menu.

Apply: update LCD settings in the scanner. These settings are saved in the external EEPROM and they are retained between resets.

5.3 Diagnostics group

Refresh: update visible table to its latest state. This function is rarely necessary because tables are automatically updated when a new message arrives.

Reset view: clear all messages from the visible table.

Copy to clipboard: copy table to the clipboard for the purpose of saving it in a text editor.

5.3.1 CCD-bus tab

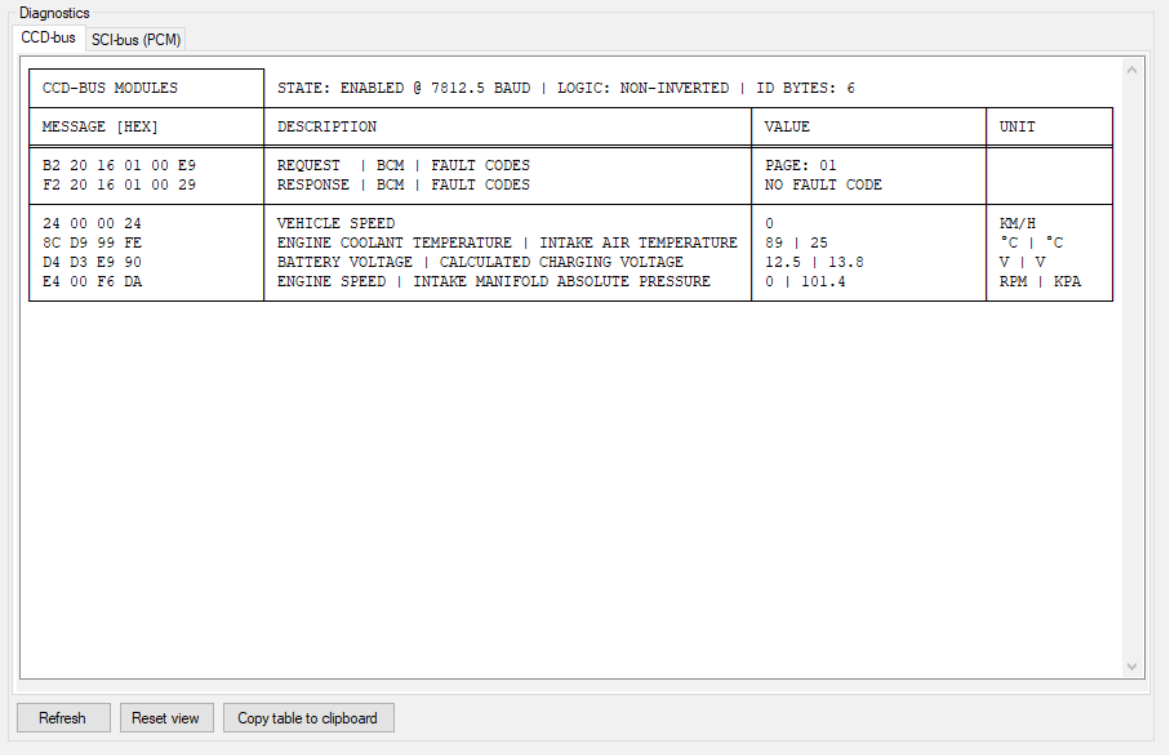


Figure 16. CCD-bus diagnostics table.

CCD-bus messages are displayed in a text table in ascending order. When a known message is received the description, value and unit columns are also filled with interpreted information. Otherwise the raw message bytes are displayed only.

Diagnostic request and response messages (B2 and F2 ID bytes) are separately highlighted at the top of the table.

5.3.2 SCI-bus tab

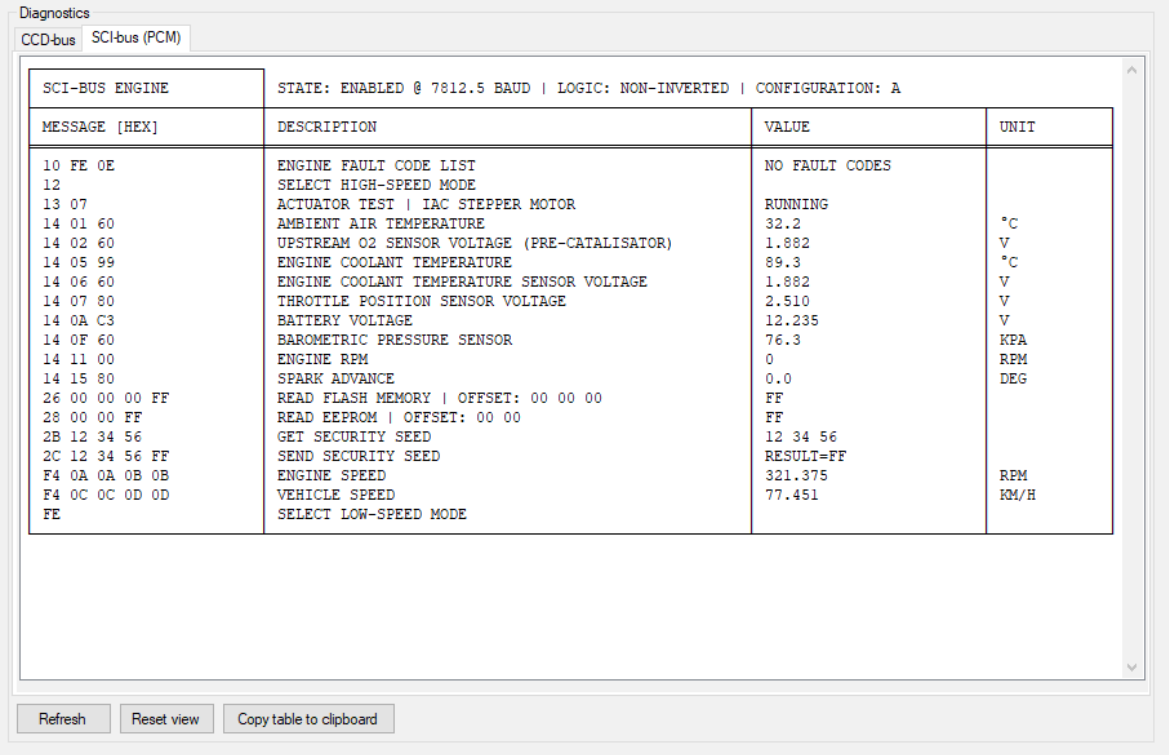


Figure 17. SCI-bus diagnostics table.

SCI-bus messages are displayed in a text table in ascending order. When a known message is received the description, value and unit columns are also filled with interpreted information. Otherwise the raw message bytes are displayed only.

Low-speed and high-speed messages are displayed in the same table. They are distinguishable by the ID byte values: high-speed messages always start with FX ID bytes.

6. CCD-bus diagnostics

One of the CCD-bus' main characteristics is that messages are continuously flowing on it when the bus is in wake-up state. All modules are connected in parallel with the bus and talk to each other. The diagnostic scanner acts as another module. Without doing anything the scanner can observe bus-traffic passively and the user can learn a great deal of what is going on with the vehicle.

However, certain diagnostic data need to be queried with B2-messages either targeted for a specific module or all of them. In response the target module sends an F2-message containing valuable information.

6.1 Diagnostic message formats

6.1.1 Request

B2 AA BB CC DD CS

B2: diagnostic request ID
AA: target module address
BB: command
CC: parameter #1
DD: parameter #2
CS: checksum

Module addresses:

10: VIC – Vehicle Info Center
18,1B: VTS – Vehicle Theft Security
19: CMT – Compass Mini-Trip
1E: ACM – Airbag Control Module
20: BCM – Body Control Module
22,60: MIC – Mechanical Instrument Cluster
41,42: TCM – Transmission Control Module
43: ABS – Antilock Brake System
50: HVAC – Heat Vent Air Conditioning
80: DDM – Driver Door Module
81: PDM – Passenger Door Module
82: MSM – Memory Seat Module
96: ASM – Audio System Module
C0: SKIM – Sentry Key Immobilizer Module
FF: ALL modules present on the CCD-bus (for group request)

If the target module does not exist on the CCD-bus then there is simply no F2-response received.

Commands:

00: reset module
10: actuator test
12: read digital parameter (switch position)
14: read analog parameter (voltage)
16: read fault codes
22: read ROM
24: read module ID / software version
2A: read vehicle identification number (VIN)
2C: write vehicle identification number (VIN)
40: erase fault codes
B0: write settings
B1: read settings
E0: self test
FF: command error response

Not all commands are available for all modules. The above example is valid for specific BCMs. When a command is not recognized by the target module, or parameters are out of range, then an F2-response is sent with an FF command.

6.1.2 Response

F2 AA BB CC DD CS

F2: diagnostic response ID
AA: responding module address
BB: command
CC: parameter #1
DD: parameter #2
CS: checksum

The responding module always repeats its address and command. When a command is not supported it is replaced by the FF byte. The 2 parameter bytes contain information regarding the request. Usually the request message has information stored in the Parameter #1 byte which is echoed in the response and the response is stored in the Parameter #2 byte.

The GUI has interpretations for many of these requests and responses but not all of them is recognized.

6.1.3 BCM example

Reset module:
B2 20 00 00 00 D2
F2 20 00 00 00 12

Actuator test:
Chime:
B2 20 10 00 08 EA
F2 20 10 00 00 22

Read digital parameter:
B2 20 12 00 00 E4
F2 20 12 04 66 8E

B2 20 12 01 00 E5
F2 20 12 11 01 36

B2 20 12 02 00 E6
F2 20 12 30 00 54

B2 20 12 03 00 E7
F2 20 12 02 15 3B

B2 20 12 04 00 E8
F2 20 12 F2 04 1A

B2 Parameter #1 contains the request (Parameter #2 is ignored), F2 Parameter #1 and #2 contains the response.

Read analog parameter:
B2 20 14 00 00 E6
F2 20 14 00 00 26

B2 20 14 01 00 E7
F2 20 14 00 01 27

B2 20 14 02 00 E8
F2 20 14 00 00 26

B2 20 14 03 00 E9
F2 20 14 00 76 9C

B2 Parameter #1 contains the request (Parameter #2 is ignored), F2 Parameter #2 contains the response.

Read fault codes:
B2 20 16 01 00 E9
F2 20 16 01 A4 CD

B2 20 16 02 00 EA
F2 20 16 02 48 72

B2 20 16 03 00 EB
F2 20 16 03 00 2B

Page number is repeated in the F2 response and F2 Parameter #2 (A4,48,00) stores a bit-encoded fault code list, meaning every page contains 8 fault bits.

Read VIN (Vehicle Identification Number):
B2 20 2A 01 00 FD
F2 20 2A 01 XX CS -> XX=VIN#1
B2 20 2A 02 00 FE
F2 20 2A 02 YY CS -> YY=VIN#2
...
B2 20 2A 11 00 0D
F2 20 2A 11 ZZ CS -> ZZ=VIN#17

7. SCI-bus diagnostics

Unlike CCD-bus, SCI-bus is completely silent by default. It cannot communicate on its own because it needs external voltage biasing. Enter the diagnostic scanner that provides the required voltage biasing. Still diagnostic information is purely request-based.

The engine controller operates in two modes: low-speed diagnostic mode and high-speed parameter interrogation mode. By default low-speed mode is selected when the engine controller wakes up or the key is cycled in the ignition switch.

In low-speed mode the engine controller and scanner communicates using half-duplex approach at 7812.5 baud, that is every request byte sent by the scanner is first echoed back by the engine controller, meaning that communication is only active at 1 direction at any given time. The response is expected within some delay after the last echoed byte. Responses are usually 1 byte long.

In high-speed mode full-duplex approach is used which means no byte echoes are expected and communication can be 2-way at any given time. In this advanced mode communication takes place at 62500 baud and the user gains access to the engine controllers RAM content where all the raw and high precision calculations are stored.

Switching between these two baudrates in the scanner is automatic when speed-selecting bytes are received. The GUI updates speed controls too.

7.1 Low-speed diagnostic mode commands

Fault code list
Request: 10
Response: 10 XX YY ZZ FE CS
XX YY ZZ: fault code list.
FE: end of list, sometimes preceded with FD.
CS: checksum.
If no fault code is present: 10 FE 0E or 10 FD FE 0B.

Fault bit list
Request: 11
Response: 11 XX YY
XX: fault bits high byte
YY: fault bits low byte

Select high-speed mode
Request: 12
Response: 12
The echo of 12 comes in low-speed mode, following bytes are expected to be sent in high-speed mode.

Actuator test
Request: 13 XX
Response: continuous stream of 13 XX while actuator test is running.
Stop test: send 13 00 until 13 00 00 is received.
Modes (XX):
00: STOP ALL TESTS
01: IGNITION COIL BANK #1
02: IGNITION COIL BANK #2

03: IGNITION COIL BANK #3
04: FUEL INJECTOR BANK #1
05: FUEL INJECTOR BANK #2
06: FUEL INJECTOR BANK #3
07: IDLE AIR CONTROL STEPPER MOTOR
... for full list see SCI.cs file.
Not all tests are available for all ECUs.

Diagnostic data

Request: 14 XX
Response: 14 XX YY
Parameters (XX):
01: AMBIENT AIR TEMPERATURE
02: UPSTREAM O2 SENSOR VOLTAGE
05: ENGINE COOLANT TEMPERATURE
06: ENGINE COOLANT TEMPERATURE SENSOR VOLTAGE
07: THROTTLE POSITION SENSOR VOLTAGE
08: MINIMUM THROTTLE POSITION SENSOR VOLTAGE
09: KNOCK SENSOR VOLTAGE
0A: BATTERY VOLTAGE
0B: INTAKE MANIFOLD ABSOLUTE PRESSURE (MAP)
0C: DESIRED IAC STEPPER MOTOR POSITION
0E: ADAPTIVE FUEL FACTOR
0F: BAROMETRIC PRESSURE
10: MINIMUM ENGINE SPEED
11: ENGINE SPEED
... for full list see SCI.cs file.
Not all parameters are available for all ECUs.

16-bit ROM value

Request: 15 XX YY
Response: 15 XX YY ZZ
XX YY: 16-bit ROM offset.
ZZ: ROM value at given offset.

ROM constant value

Request: 16 XX
Response: 16 XX YY
XX: 80 00 + XX offset.
YY: ROM constant at given offset.

Erase fault codes

Request: 17
Response: 17 XX
XX: E0 or E0 E0 E0 -> erased, otherwise failed.
Stop engine before erasing codes.

Control ASD relay

Request: 18 XX
Response: 18 XX YY
Unknown diagnostic mode.

Minimum idle speed

Request: 19 XX
Response: 19 XX
Temporarily set engine speed to 7.85×XX RPM. Timeouts after a few seconds and the engine returns to normal idle speed. Request needs to be continuously sent at regular intervals. Do not enter XX<42 (518 RPM) because it might stall the engine.

Switch test

Request: 1A XX
Response: 1A XX YY
XX: 01-04. YY: bit-encoded values.

Reset EMR
Request: 1C XX
Response: 1C XX YY
Modes (XX):
10: RESET EMR 1
11: RESET EMR 2

Reset adaptive memory
Request: 23 XX
Response: 23 XX YY
Modes (XX):
01: ERASE ENGINE FAULT CODES / EEPROM
02: ADAPTIVE FUEL FACTOR
03: IAC COUNTER
04: MINIMUM TPS
05: FLEX FUEL PERCENT
... for full list see SCI.cs file.

24-bit ROM value
Request: 26 XX YY ZZ
Response: 26 XX YY ZZ AA
XX YY ZZ: 24-bit ROM offset.
AA: ROM value at given offset.

16-bit EEPROM value
Request: 28 XX YY
Response: 28 XX YY ZZ
XX YY: 16-bit EEPROM offset.
ZZ: EEPROM value at given offset.

Select low-speed mode
Request: FE
Response: FE
In low-speed mode nothing happens.

... for full list see SCI.cs file.

7.2 High-speed parameter interrogation mode commands

Select RAM table
Request: F0-FF, except FE
Response: F0-FF, except FE
Table number is echoed back on success.

Single RAM value
Request: FX XX (XX=00-EF)
Response: FX XX AA.
FX: RAM table has to be selected for every request.
XX: RAM offset in the given table.
AA: RAM value at given offset.

Multiple RAM values
Request: FX XX YY
Response: FX XX AA YY BB.
FX: RAM table has to be selected for every request.
XX: RAM offset.
AA: RAM value at XX.
YY: RAM offset.
BB: RAM value at YY.

RAM table dump
Request: FX FF
Response: FX 00 AA 01 BB 02 CC ... EE DD EF EE
FX: RAM table has to be selected for every request.
FF: read all RAM values in the given table.

Select low-speed mode Request: FE
Response: FE
The echo of FE comes in high-speed mode, following bytes are expected to be sent in low-speed mode.

Example RAM table dumps from a 1995 Chrysler Stratus JA 2.0L:

F2	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	00	00	00	00	00	00	A8	57	12	29	00	3B	6B	F4	00	E3
10	00	00	89	20	1C	1F	24	38	03	2D	CD	55	04	C4	B4	00
20	33	B9	EB	A8	86	03	93	10	35	0E	17	FA	12	00	0A	07
30	0B	00	00	FB	00	02	C4	18	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	45	00	FF	00	00	00	00
50	00	00	00	FC	00	00	00	00	00	00	00	03	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
80	00	00	04	00	60	00	00	00	00	00	00	00	00	00	00	00
90	00	80	80	80	80	80	80	80	80	80	80	00	00	00	00	00
A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	E5
C0	00	00	32	00	04	69	00	00	00	00	00	00	10	02	00	10
D0	02	10	00	C0	38	01	10	01	02	00	0C	00	00	00	00	00
E0	00	00	20	C0	00	24	00	00	00	00	00	00	00	00	00	00
F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

F4	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	F0	00	00	04	00	00	00	00	00	00	00	00	00	00	24	C8
10	67	93	23	29	23	8C	95	EA	F4	F4	00	18	18	8D	95	00
20	00	EA	F0	34	00	2F	29	19	80	19	80	FB	00	F0	0F	00
30	00	0F	0F	0F	0F	1C	40	3B	0F	0F	E5	00	FF	FB	3B	80
40	10	01	00	0C	00	00	0F	0F	18	17	07	18	82	C0	00	10
50	00	80	80	80	0F	80	80	80	0F	0F	0F	0F	0F	0F	0F	0F
60	0F	0F	10	35	0E	17	FA	12	00	0A	07	0B	00	00	FB	00
70	02	C4	18	00	00	00	00	00	00	00	19	18	00	00	00	00
80	00	00	00	00	00	00	00	00	00	00	00	00	00	0F	0F	0F
90	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F
A0	0F	0F	0F	0F	0F	0F	00	00	0F	0F	0F	0F	0F	0F	0F	0F
B0	0F	0F	0F	0F	0F	0F	0F	00	00	00	80	0F	0F	0F	0F	0F
C0	00	00	00	00	0F	0F	0F	0F	00	00	00	00	03	2D	CD	55
D0	0F	0F	00	00	00	00	00	00	00	00	00	00	00	FF	8E	4F
E0	0F	0A	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F
F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

F5	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	0F	80	80	80	0F	80	80	80	80	80	00	80	80	00	0F	0F
10	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F
20	FB	1C	40	FB	FF	24	00	0F	0F	0F	0F	0F	0F	0F	0F	0F
30	0F	0F	0F	00	00	00	00	00	00	00	0F	0F	0F	0F	0F	0F
40	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
50	00	00	0F	0F	0F	0F	0F	0F	00	00	00	00	00	00	0F	0F
60	00	0F	0F	0F	00	00	00	00	00	00	80	60	FC	0F	0F	0F
70	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	00	0F	00	00	00	0F
80	00	00	00	00	03	2D	CD	55	00	00	00	00	00	00	00	00
90	00	00	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F
A0	0F	00	00	00	95	C0	01	8E	89	89	20	00	02	00	00	2B
B0	00	00	00	00	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F
C0	00	00	00	00	00	00	00	00	00	00	00	0F	0F	0F	0F	0F
D0	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F
E0	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F	0F
F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

F6	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	00	00	00	00	00	00	A8	57	12	29	00	3B	6B	F4	00	E3
10	00	00	89	20	1C	1F	24	38	03	2D	CD	55	04	C4	B4	00
20	33	B9	EB	A8	86	03	93	10	35	0E	17	FA	12	00	0A	07
30	0B	00	00	FB	00	02	C4	18	00	00	00	00	00	00	00	00
40	00	00	00	00	00	00	00	00	00	45	00	FF	00	00	00	00
50	00	00	00	FC	00	00	00	00	00	00	00	06	00	00	00	00
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
80	00	00	04	00	60	00	00	00	00	00	00	00	00	00	00	00
90	00	80	80	80	80	80	80	80	80	80	00	00	00	00	00	00
A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	E5
C0	00	00	32	00	3A	04	00	00	00	00	00	00	10	02	00	10
D0	02	10	00	C0	38	01	10	01	02	00	0C	00	00	00	00	00
E0	00	00	20	C0	00	24	00	00	00	00	00	00	00	00	00	00
F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

Offset	Parameter	Scaling	Unit
F4 01	DTC		
F4 02	DTC Freeze Frame		
F4 03	Fuel System Status (Bank 1)	Table 1	
F4 04	Fuel System Status (Bank 2)	N/A	
F4 0A 0B	Engine Speed	0.125	rpm
F4 0C 0D	Vehicle Speed	0.025105766	km/h

Table 1	
0x01	Open Loop
0x02	Closed Loop
0x04	Open Loop / Drive
0x08	Open Loop / DTC
0x10	Closed Loop / DTC

Lots of undiscovered parameters remain here.

7.3 Organizing a custom SCI-bus table layout

Since SCI-bus is a request based bus the user should compile a list of request messages and do a repeated message transmission to keep the SCI-bus table updated as if it were a CCD-bus table.

7.3.1 Low-speed mode SCI-bus example

Copy the following example line into the combo box of the SCI-bus tab and click on the Add button or press the Enter key.

```
10,14 01,14 02,14 05,14 07,14 09,14 0A,14 0B,14 0E,14 11
```

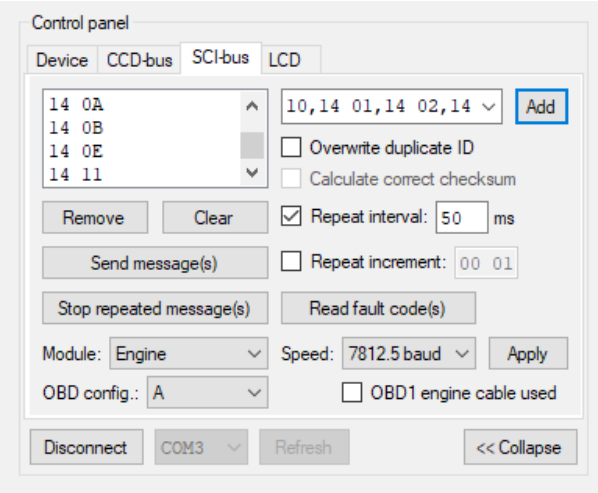


Figure 18. Custom low speed SCI-bus request messages.

Enable the Repeat interval option and enter 50 ms into the text box next to it. The fastest refresh interval in low-speed mode is limited around 25 ms.

Make sure that the engine controller is in low-speed mode (7812.5 baud) and click on the Send messages(s). See results in the SCI-bus table.

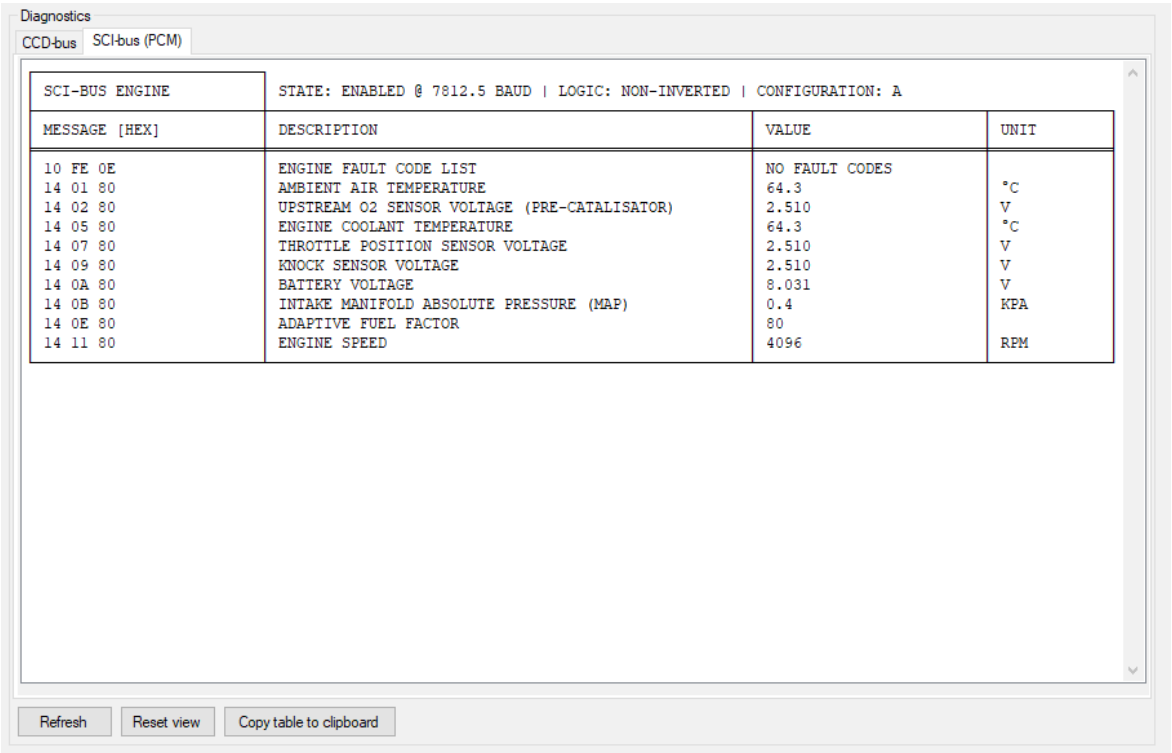


Figure 19. Custom low-speed SCI-bus request messages displayed in the table.

To stop this diagnostic session click on the Stop repeated message(s) button.

Many more parameters can be added or removed from the list, it is up to the user's preference. Keep in mind that the more parameters there are on the list, the more slow the refresh rate of individual lines are going to be.

7.3.2 High-speed mode SCI-bus example

In order for this example to work the engine controller needs to be in high-speed mode. To do this, in low-speed mode, type 12 into the combo box and press the Enter key for quick sending.

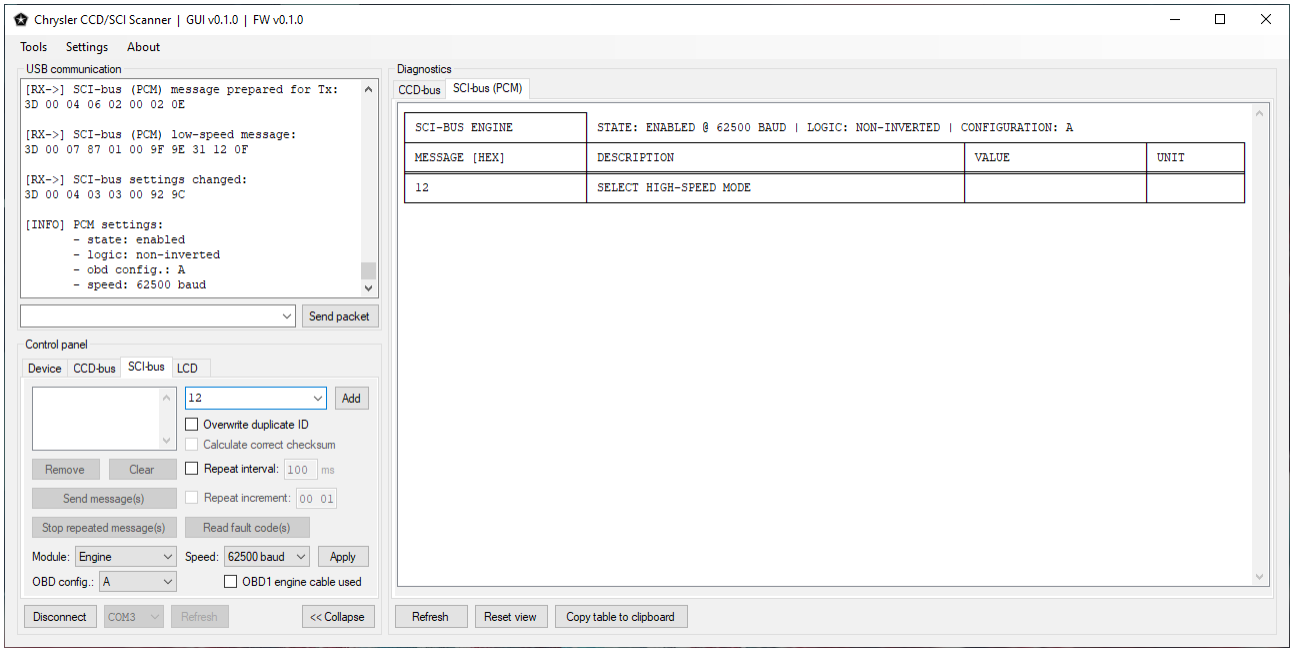


Figure 20. Switch SCI-bus to high-speed mode.

The engine controller echoes back 12, the scanner configures itself to high-speed mode too and sends a settings packet back to the user.

Now copy the following example line into the combo box of the SCI-bus tab and click on the Add button or press the Enter key.

```
F4 0A 0B,F4 0C 0D
```

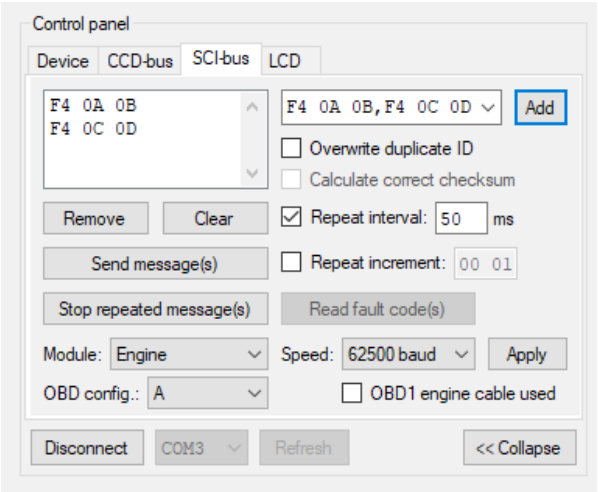


Figure 21. Custom high-speed SCI-bus request messages.

Enable the Repeat interval option and enter 50 ms into the text box next to it. The fastest refresh interval in high-speed mode is limited around 1-5 ms per RAM address. Carefully choose this value because it too little interval can freeze the scanner and the GUI.

Click on the Send messages(s) button to begin message transmission. See results in the SCI-bus table.

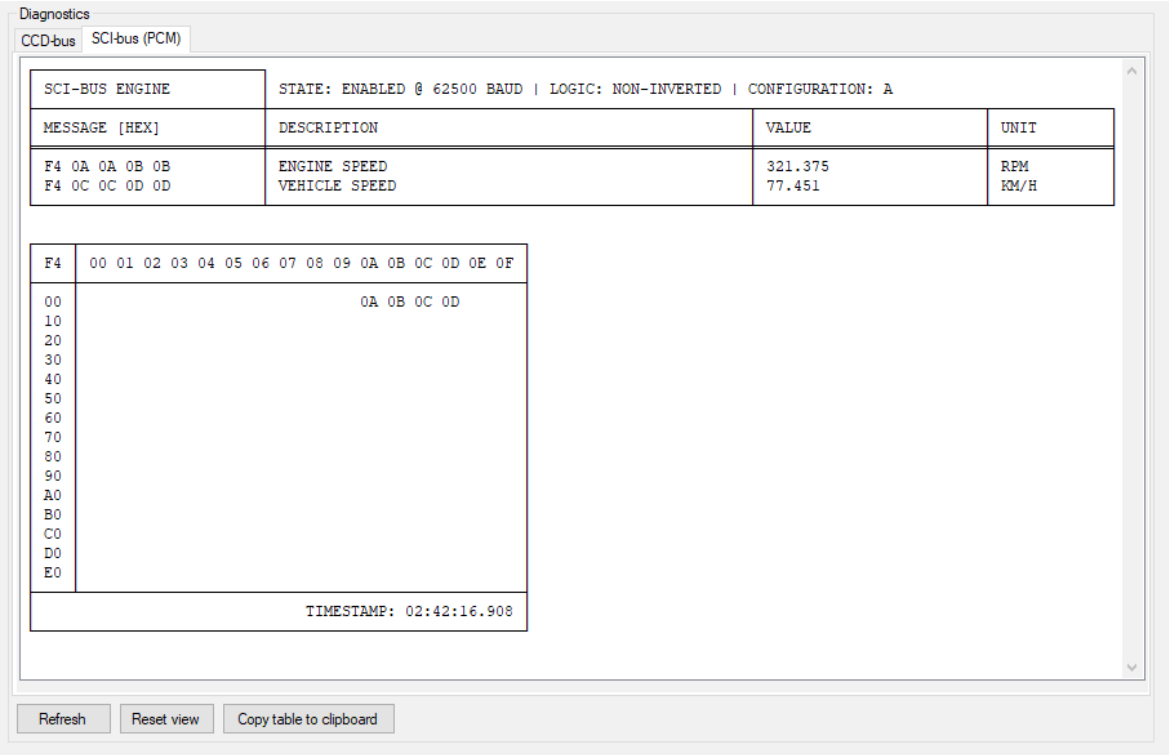


Figure 22. Custom high-speed SCI-bus request messages displayed in the table.

High precision parameters of interest come usually in 16-bit resolution, meaning that most of the time at least 2 RAM values need to be read one after another. RAM values are also displayed in another summary table.

To stop this diagnostic session click on the Stop repeated message(s) button.

To return to low-speed mode, type FE into the combo box and press the Enter key for quick sending.

Again, many more parameters can be added or removed from the list, it is up to the user's preference. Keep in mind that the more parameters there are on the list, the more slow the refresh rate of individual lines/table values are going to be.

7.3.3 Mixed low-speed and high-speed mode SCI-bus example

This method might be unstable because switching communication speeds may require multiple attempts.

Copy the following example line into the combo box of the SCI-bus tab and click on the Add button or press the Enter key.

```
FE,10,14 01,14 02,14 05,14 07,14 09,14 0A,14 0B,14 0E,14 11,12,F4 0A 0B,F4 0C 0D
```

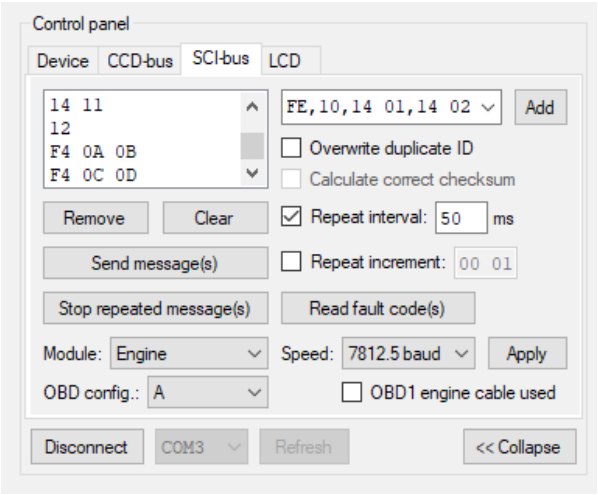


Figure 23. Custom mixed-speed SCI-bus request messages.

Notice that the first byte is the return to low-speed mode command. This command is safe to use in both speeds and makes this method not care about the initial baudrate.

Enable the Repeat interval option and enter 50 ms into the text box next to it.

Click on the Send messages(s) button to begin message transmission. See results in the SCI-bus table.

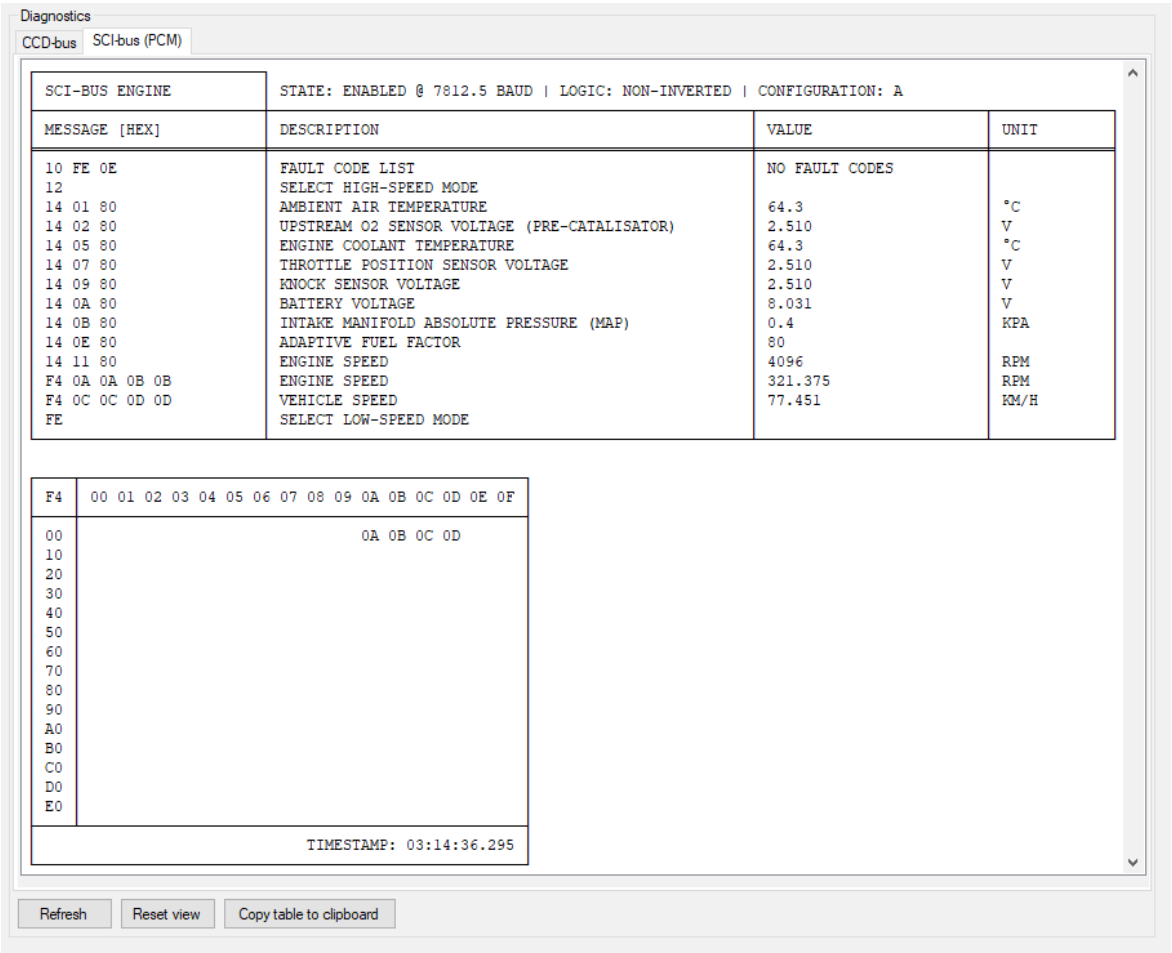


Figure 24. Custom mixed-speed SCI-bus request messages displayed in the table.

To stop this diagnostic session click on the Stop repeated message(s) button.

7.3.4 High-speed mode SCI-bus RAM table dump example

Switch the engine controller to high-speed mode (12), then quick-send the FX FF command to dump every value from a RAM table once. The FF shortcut makes the scanner ask for all available RAM values, as if 00 01 02 ... EE EF were written.

Be very careful when repeating this command. It is especially slow to read 240 bytes one after another. Start from 500 ms interval and go down from there, until the scanner/firmware freezes, to establish a working minimum interval.

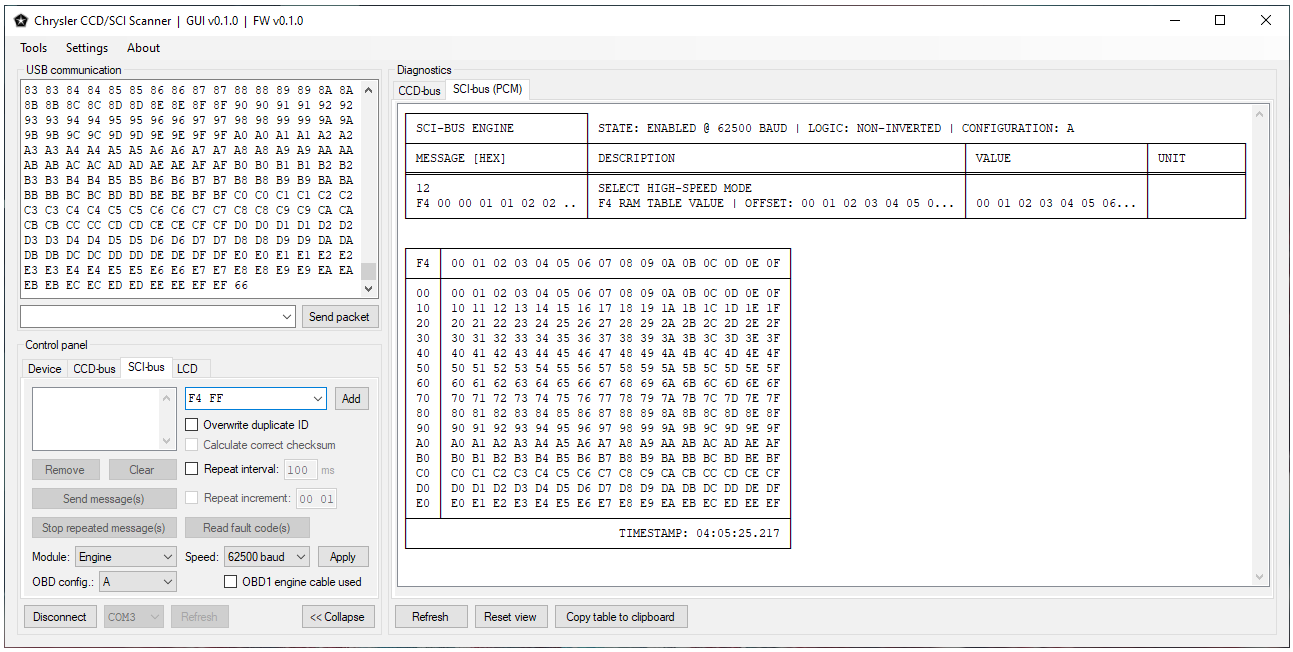


Figure 25. High-speed SCI-bus RAM table dump.

It is recommended to reset tables (Reset view button) when individual RAM values are read after a table dump. This way old values are cleared and only new ones appear.

7.3.5 Engine fault code list request example

In order for this feature to work the engine controller needs to be in low-speed mode. Fault codes can be requested by either sending the command 10, or by clicking on to the Read fault code(s) button.

When no fault codes are detected the USB text box says the following:

```
[RX->] SCI-bus (PCM) fault code list:
3D 00 09 87 01 01 26 DC A2 10 FE 0E 52

[INFO] No PCM fault code found.
```

If there are stored fault codes in the engine controller:

```
[RX->] SCI-bus (PCM) fault code list:
3D 00 0A 87 01 01 29 35 FB 10 3E FE 4C 84

[INFO] PCM fault code(s) found:
3E: UPSTREAM O2 SENSOR SHORTED TO VOLTAGE
```

Present fault codes and interpretations are listed in the same text box.
For all possible fault codes see SCI.cs file.
To erase fault codes stop the engine and send the command 17 in low-speed mode.

8. Advanced diagnostics

8.1 CCD-bus repeated messages with parameter incrementing

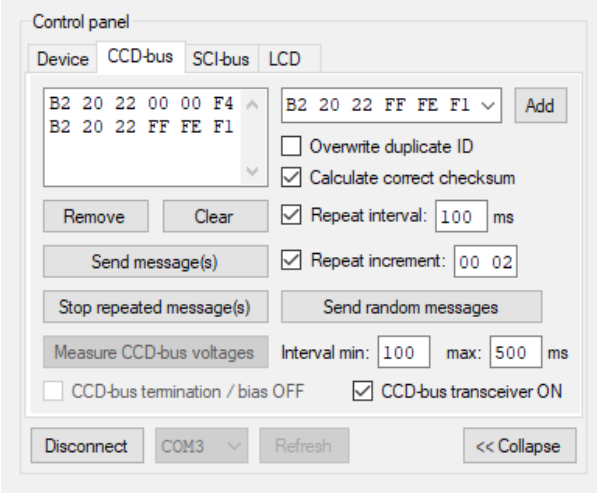


Figure 26. CCD-bus ROM reading example.

This advanced feature lets the user sweep through many parameters in one go. In this example a Body Control Module (BCM) ROM-reading session is prepared. Only B2-messages are supported.

Two messages need to be present in the message list box: start and end message. The 4th and 5th parameter bytes are the variables here. Because of the properties of F2-responses every second ROM address is queried thus the increment is set to 2.

```
B2 20 22 00 00 F4
F2 20 22 AA BB CS -> ROM:0000=AA | ROM:0001=BB
B2 20 22 00 02 F6
F2 20 22 CC DD CS -> ROM:0002=CC | ROM:0003=DD
...
B2 20 22 FF FE F1
F2 20 22 EE FF CS -> ROM:FFFE=EE | ROM:FFFF=FF
```

Once every ROM address is read a binary file can be parsed from the log files. This binary file contains the software which runs in the BCM and can be disassembled to better understand it.

Currently this feature is unstable and may fail shortly after the session is started. The binary file parser is under development.

8.2 SCI-bus repeated messages with parameter incrementing

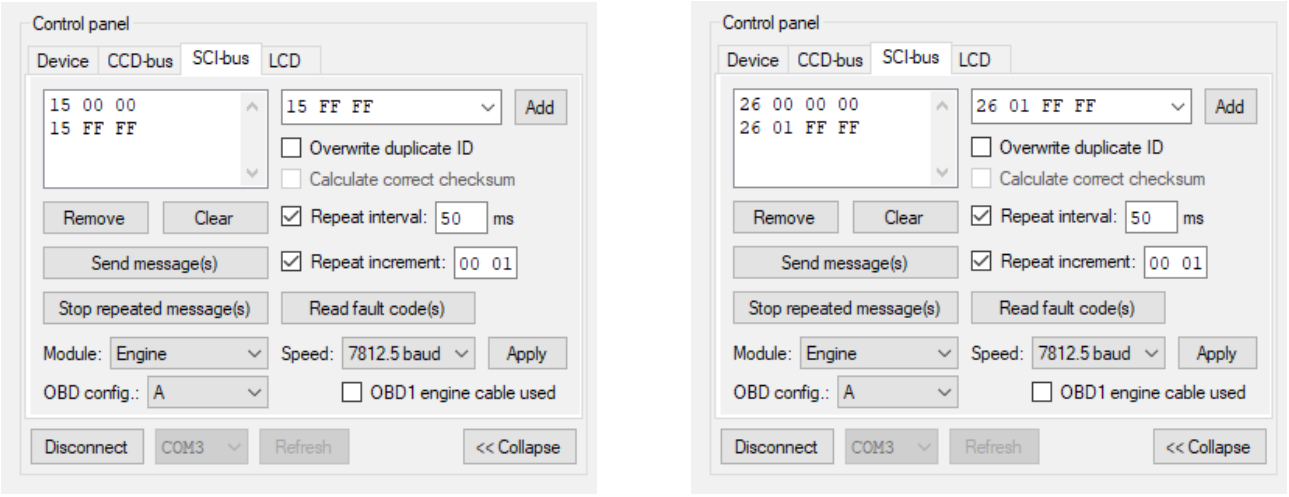


Figure 27. SCI-bus ROM reading example (left: 16-bit, right: 24 bit).

This advanced feature lets the user sweep through many parameters in one go. In this example an engine controller ROM-reading session is prepared. 1-4 byte long messages are supported.

Two messages need to be present in the message list box: start and end message. When the message length is 2-4 bytes, then the first ID byte is kept constant and the rest of the bytes get incremented. When the message is only 1 byte long then the ID byte itself gets incremented.

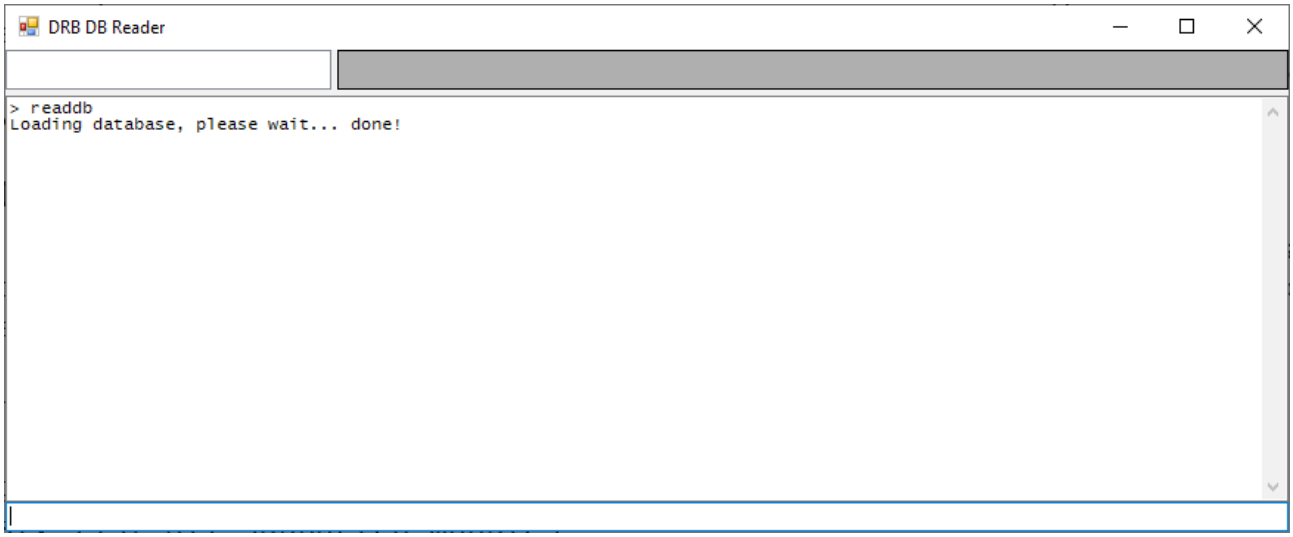
```
TX: 15 00 00
RX: 15 00 00 AA -> ROM:0000=AA
TX: 15 00 01
RX: 15 00 01 BB -> ROM:0001=BB
...
TX: 15 FF FE
RX: 15 FF FE CC -> ROM:FFFE=CC
TX: 15 FF FF
RX: 15 FF FF DD -> ROM:FFFF=DD
```

Once every ROM address is read a binary file can be parsed from the log files. This binary file contains the software which runs in the engine controller and can be disassembled to better understand it.

Currently this feature is unstable and may fail during the session. The binary file parser is under development.

8.3 Introduction to the DRBDBReader software

This is a wonderful and one-of-a-kind software to decode dealer level (DRB3 and StarScan) diagnostic databases. A compiled executable with a DRB3 database can be found in this application's DRBDBReader folder.



Commands:

readdb: load "database.mem" file into RAM

modlist: list all supported modules

modsearch <name>: search module by <name>

modtxlist <id>: list all commands of a module <id>

dumpconverter <id>: list properties of a command <id>

txrunconverter <id> <value>: imperial conversion of <value> using <id> properties

txrunconvertermetric <id> <value>: metric conversion of <value> using <id> properties

txsearch <str1> && <str2> && ... && <strn>: search commands among all modules

Example:

```
> readdb
Loading database, please wait... done!

> modlist
[...]
BCM JA BASE; sc: Body; 0x1010
[...]

> modtxlist 0x1010
[...]
ENGINE RPM: CCD; xmit: E4-00-FF; sc: Body; 0x80002bd9
[...]
```

```
> dumpconverter 0x80002bd9
TYPE:  NUMERIC
REC:   11-11-4E-54-02-08
DSREC: 4E-54-00-00-3F-80-00-00-3E-29-00-00-00-00-00-05-3E-29-20-44

UNIT: RPM
SLOPE:  32
OFFSET: 0
SLCONV: 1
OFCONV: 0
```

Manual calculation: multiply the raw hexadeximal number by the SLOPE value then add the OFFSET value. When metric conversion is available then this imperial result has to be multiplied by SLCONV again then OFCONV added.
Pay attention to what number formatting you use. The software makes a difference between hexadecimal (0x16) and decimal (22) number formats.

Example CCD-bus message: E4 4A 00 2E (hexadecimal format implied).
Here 4A is the raw engine speed.

```
> txrunconverter 0x80002bd9 0x4A
2368 RPM
```

Manual calculation: 0x4A = 74 -> (74 × SLOPE) + OFFSET = (74 × 32) + 0 = 2368 RPM

To search a specific ID-byte on the CCD-bus you can do the following:

```
> txsearch ccd && xmit: 24
VEHICLE SPEED: CCD; xmit: 24-00-FF; sc: Air Temp Control; 0x80003883
VEHICLE SPEED: CCD; xmit: 24-00-FF; sc: Vehicle Theft Security; 0x80003b61
VEHICLE SPEED: CCD; xmit: 24-00-FF; sc: Transmission; 0x80004c19
VEHICLE SPEED: CCD; xmit: 24-00-FF; sc: Transmission; 0x80004c1a
VEHICLE SPEED SENSOR: CCD; xmit: 24-00-FF; sc: Body; 0x80004ca1
VEHICLE SPEED: CCD; xmit: 24-00-FF; sc: MIC; 0x80004d64
VEHICLE SPEED SENSOR: CCD; xmit: 24-00-FF; sc: Compass Mini-Trip; 0x80004e81
```

Now it's a trial and error to find out which record to use but usually they contain the same scaling rules among different modules.

```
> dumpconverter 0x80004ca1
TYPE:  NUMERIC
REC:   11-11-0C-22-01-03
DSREC: 0C-22-00-00-3F-CD-FE-FC-3E-30-00-00-00-00-00-03-3E-31-20-44

UNIT (DFLT/MTRC): MPH/KPH
SLOPE:  1
OFFSET: 0
SLCONV: 1.609344
OFCONV: 0
```

Notice that the SLOPE value is 1 and OFFSET is 0. This means that the raw byte value is meant to be used as is and you only need to append the appropriate UNIT.

Example CCD-bus message: 24 16 24 5E (again hexadecimal format implied).
This format is interesting because it makes use of the two payload bytes to avoid conversion between imperial and metric units.

See the first payload byte is 0x16 = 22 MPH and the second byte 0x24 = 36 KM/H happens to be equal to 22 × 1.609344 (= 35.4 ~ 36) as per the metric conversion rule.

9. Log files

The diagnostic scanner saves detailed log files for USB/CCD/SCI-communications separately in the application's LOG folder.

Settings / Include timestamp in log files.

This menu makes the GUI save the timestamp before CCD/SCI-bus message bytes.

10. GUI/Firmware update

Tools / Update.

Firmware update is fully automatic.
GUI update just downloads latest .zip-file and user has to unpack and overwrite the application's .exe-file.

11. Disclaimer

The hardware and software is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

The GUI contains no vehicle profiling capabilities and relies on non-variable diagnostic parameters that are same for every vehicle. Under no circumstances will this device replace a dealer level diagnostic scanner.