

# CAN Scanner

## Назначение CAN-сканера

Практически любой современный автомобиль (и другая подвижная техника) буквально напичкан множеством различных датчиков. Эти датчики непрерывно собирают огромное количество различных параметров работы узлов и агрегатов техники и передают их для обработки в электронный блок управления (ЭБУ). Причем таких ЭБУ в автомобиле может быть несколько, до 80! Это настоящий мозг автомобиля. Он сигнализирует о неполадках, облегчает управление и эксплуатацию, следит за безопасностью, развлекает водителя и пассажиров и вообще делает их нахождение внутри автомобиля комфортным. Кроме сбора данных с датчиков, ЭБУ также отдает команды различным исполнительным механизмам. Для связи ЭБУ между собой, а также со всеми датчиками и исполнительными механизмами служит CAN-шина.

Подключиться к CAN-шине достаточно просто. Все выпускаемые терминалы линейки FORT позволяют сделать это. Процесс подключения подробно описан в соответствующем [Руководстве по эксплуатации](#). Но само по себе наличие такого подключения ничего не дает, кроме того, что терминал начинает "слушать" CAN-шину. Для терминала информация из CAN-шины - это лишь бессмысленный поток нолей и единиц. И лишь при помощи специальной программы, CAN-сканера, эту ситуацию возможно изменить

Дело в том, что данные в CAN-шине передаются в виде пакетов. Формат этих пакетов определяется протоколом. Терминалы FORT могут подключаться к CAN-шинам, использующим протокол J1939. Он является наиболее распространенным и применяется большинством производителей грузовых автомобилей, автобусов, сельскохозяйственной и специальной техники. Протокол J1939 определяет формат пакета, где содержится в том числе идентификатор пакета и блок данных объемом до 8 байт. Но, во-первых, каждую минуту через CAN-шину проходят тысячи пакетов с разными идентификаторами и разными данными. А во-вторых, каждый автопроизводитель сам решает, какой байт или даже бит, и в каком пакете у него будет содержать ту или иную информацию. Единый стандарт отсутствует.

Например, Производитель 1 решил, что в 3-м байте пакета с заголовком 044A у него будет информация об уровне топлива в левом баке. А Производитель 2 решил разместить по тому же адресу значение напряжения в бортовой сети автомобиля. Мало того, к каждому значению могут применяться еще и различные множители. Например, какой-то байт содержит число 80. И только производителю известно, что на самом деле это температура окружающего воздуха умноженная на 4. Увы, и это еще не все. Адресация данных в CAN-шине для одинаковой модели автомобиля одного производителя может отличаться в зависимости от года выпуска. Есть и другие подводные камни, перечислять которые нет смысла.

Главное, что программа CAN-сканер позволит инженеру или установщику отыскать нужный параметр среди множества пакетов. А затем передать этот параметр в систему Fort Monitor для дальнейшей обработки и отображения.

## Функционал CAN-сканера

Функции	
1	Отображение принимаемых параметров и их изменение в реальном времени
2	Поиск значений с заданным отклонением и коэффициентами
3	Поиск дискретных значений
4	Отображение байтов и битов выбранного параметра
5	Наложение маски и отображение подсчитанного значения
6	Построение графика значения маски
7	Составление списка отобранных масок с отображением их значений в реальном времени
8	Отслеживание либо скрытие некоторых параметров
9	Подсвечивание изменяющихся байтов
10	Фильтрация параметров по различным критериям (отслеживаемые, скрытые, изменяющиеся, найденные)
11	Отображение графиков по каждому байту каждого параметра в реальном времени
12	Запись лога принятых данных и его воспроизведение без подключенного терминала

[CanScan.zip](#) - с примерами снятых логов

[Что еще можно сделать/изменить \(доработки\)](#) **Раздел должен быть отредактирован после внедрения доработок**

## Запуск CAN-сканера в Конфигураторе 2.0

1. Удостовериться что терминал подключен к CAN шине транспортного средства ([см. инструкцию к терминалу](#))
2. Настроить параметры подключения к CAN в конфигураторе (см. инструкцию к [конфигуратору](#))
3. Открыть CAN сканер.
4. Нажать кнопку "Запустить". Если п.п. 1 и 2 выполнены верно, то начнут отображаться принимаемые данные

**Раздел должен быть отредактирован после интеграции Конфигуратора с CAN-сканером**

## Описание интерфейса CAN-сканера

The screenshot displays the Fort CanScanner application window. At the top, there is a control bar with buttons for 'Запустить' (Start), 'Остановить' (Stop), 'Запись в файл' (Save to file), and 'Загрузить лог' (Load log). The current CAN ID is shown as 'CAN id: 0x0202'. Below this is a hex dump of the captured data: '00010111 10110011 00000000 00000000 00011011 01000000 00011001 01100110'. The main data table has columns for 'Следить' (Follow), 'Скрыть' (Hide), 'Id', and 'Data'. The data table shows several rows of CAN messages. A right-hand panel contains settings for bit selection: 'Смещение: 0', 'Размер: 16', 'Делитель: 4', and 'Отобразить'. Below this is a filter section with checkboxes for 'Показывать только отслеживаемые', 'Показывать скрытые', 'Показывать меняющиеся', and 'Показывать графики'. There is also a search section with 'Значение: 800', 'Отклонение: 50', and a 'Поиск дискретных значений' section with 'Сейчас 0', 'Сейчас 1', and 'Очистить' buttons.

- 1. Кнопки управления.** Позволяют считать и записать лог сообщений из CAN-шины. Запустить и остановить загруженный лог
- 2. Выбранный пакет данных из CAN-шины.** Выбор пакета осуществляется кликом мыши по любой строке в панели (3) (см. ниже). ID выбранного пакета отображается справа над панелью (2) жирным голубым шрифтом. Обращаем внимание, что в данной строке отображается не весь пакет, а только блок содержащихся в нем данных (8 байт). Динамично изменяющиеся в процессе "прослушивания" CAN-шины или проигрывания лога биты, выделяютяются красным цветом. Причем чем ярче оттенок красного, тем меньше времени прошло с момента изменения значения. Предусмотрена возможность выделения для анализа нескольких последовательных битов (фрагмента данных). Это можно сделать мышью, или вручную задать в панели (4) количество нужных битов и смещение относительно начала пакета данных. Выбранные биты выделяются голубым фоном. Каждый байт представлен в двоичном формате, но под каждым байтом отображается его HEX-представление
- 3. Панель отображения пакетов данных их CAN-шины.** По умолчанию в панели отображаются все пакеты из CAN-шины в HEX-формате. Пакеты, в которых зафиксировано изменение данных, выделены голубым фоном. В столбце ID отображен уникальный идентификатор пакета. Есть возможность скрыть ненужные пакеты или следить за нужными, кликнув соответствующие чекбоксы в строке. Выбранный для отображения в панели (2) пакет выделяется синим фоном.
- 4. Панель выбора части последовательных битов (фрагмента данных) в пакете.** В поле "Размер" указывается количество нужных вам битов. В поле "Смещение" задается смещение первого из нужных вам битов относительно первого бита в пакете. При этом есть возможность задать как прямой порядок байтов от старшего к младшему (BigEndian или BE) так и обратный порядок (LittleEndian или LE). Задав все значения и кликнув кнопку "Отобразить", нужные вам байты в панели (2) будут выделены голубым фоном. Назначение поля "Делитель" будет описано ниже (5).
- 5. Панель анализа фрагмента данных.** Для удобства анализа выбранный фрагмент данных здесь представлен в виде графика. Также в панели представлено исходное и тарированное значение фрагмента данных в десятичном виде. *Например, вы предполагаете, что некий фрагмент данных содержит значение количества оборотов двигателя. Терминал подключен к CAN-шине автомобиля, и вы можете отслеживать пакеты данных в режиме онлайн. Выбрав в поле (4) нужный фрагмент данных вы нажимаете на педаль газа и видите, как график взлетает вверх. Отпускаете - график идет вниз. Все видно наглядно. По тахометру вы держите обороты на уровне 1500. Значение выбранного фрагмента данных - около 6000. Но, применив делитель 4 в панели (4), вы получите искомое тарированное значение 1500. О том, как подобрать правильный делитель написано [здесь](#)*
- 6. Панель отображения выбранных фрагментов данных.** Часто бывает, что при поиске какого-то параметра, например тех же оборотов двигателя, сразу несколько фрагментов данных в разных пакетах имеют примерно одинаковое значение. В таких случаях требуются дополнительные проверки. Поэтому все отобранные фрагменты данных можно сохранить в отдельной панели. Для этого нужно выбрать фрагмент, как это описано в разделе (4) и кликнуть кнопку "Отобразить". Значение выбранного фрагмента данных появится в панели (6) вместе с другой необходимой информацией
- 7. Панель фильтрации и поиска данных.** Это основная рабочая панель, содержащая большое количество элементов. Поэтому ей посвящен весь следующий раздел

## Описание панели фильтрации и поиска данных

### 1. Фильтры (можно активировать один или несколько):

- "Показывать только отслеживаемые". При активации данного фильтра в панели (3) будут показаны только те пакеты, которые отмечены галочками в столбце "Следить"
- "Показывать скрытые". При активации данного фильтра в панели (3) будут показаны только те пакеты, которые отмечены галочками в столбце "Скрыть"
- "Показывать меняющиеся". При активации данного фильтра в панели (3) будут показаны пакеты, в которых с момента последнего обновления данных изменился хотя бы один бит информации
- "Показывать графики". При выборе данной функции, фоном для каждого байта во всех отображаемых пакетах в панели (3) будет динамически меняющийся график, аналогичный графику в панели (5). Выглядит это следующим образом:



Данная функция является чрезвычайно полезной. Беглого взгляда на любой пакет достаточно, чтобы понять - какой байт в нем представляет практический интерес, а какой нет. *Например, вы пытаетесь отыскать в CAN-шине какое-то фиксированное значение. Очевидно, что байты с текущими значениями "7C" и "41" вам не подходят, так как по графику видно, что они содержат циклично меняющиеся значения. Также, как и байты "FE" и "02", графики которых походят на "белый шум". Последний байт состоит из одних единиц, в средних байтах всего по одному биту имеют значение "1", а значит они тоже скорее всего не подходят. Соответственно интерес может представлять только первый байт со значением 68*

- "Окно подсветки изменений". Этот фильтр позволяет выделять красным цветом в поле (3) фрагменты данных, которые меняются чаще, чем заданная в фильтре величина (в секундах).

2. **Цветовая легенда** справа от фильтров является простой подсказкой, каким цветом фона будут выделяться те или иные отфильтрованные пакеты в панели (3)

3. **Поиск значений** позволяет найти в массиве данных параметр, значение которого вам примерно известно. Как это сделать - описано [ниже](#)

4. **Поиск дискретных значений.** Процесс описан в следующем разделе

## Поиск дискретных значений

### Поиск дискретных значений

Показывать найденные бинарные значения

Такой метод поиска подходит для определения факта открытия или закрытия дверей, состояния ремней безопасности (пристегнуты/отстегнуты) или габаритных огней и прочих дискретных параметров. Алгоритм простой: кликаем кнопку "Сейчас 0" когда искомое состояние неактивно (например, дверь закрыта), либо кнопку "Сейчас 1" когда активно. В результате в поле (3) отфильтровываются пакеты, где содержатся биты, не меняющие своего значения до клика кнопки. Лучше всего показать работу метода на примере.

Нас интересует факт открытия водительской двери. Предполагаем, что при закрытой двери некий бит во всем массиве данных имеет значение "0", а при открытой - "1"

- В исходном состоянии дверь закрыта. Нажимаем на "Сейчас 0". Лучше сделать это несколько раз подряд. В результате в поле (3) отфильтруются пакеты, где содержится хотя бы один бит, ни разу не меняющий своего значения "0" между нажатиями кнопки. Но этих пакетов будет много.
- Открываем дверь и нажимаем на "Сейчас 1". Также повторяем это несколько раз. В поле (3) останутся только те пакеты, где хотя бы один бит поменял свое значения с "0" на "1". Таких останется всего несколько штук, и их уже можно анализировать визуально.
- Включаем фильтрацию "Показывать найденные бинарные значения" для отображения подходящих параметров.
- Несколько раз открываем и закрываем дверь, и визуально ищем изменяющийся байт. Все такие байты будут выделяться красным цветом. Если визуально определить нужный байт не удалось, можно повторить пп. 1 и 2, и еще сократить количество отфильтрованных пакетов.
- Выделяем подходящий пакет кликом мыши и контролируем в поле (2) какой бит меняет своё состояние при открывании/закрывании двери.
- Выделяем в поле (2) нужный бит. Это можно сделать кликом по нему мыши, либо в поле (4) вручную задать смещение и размер фрагмента данных.
- Нажимаем в поле (4) кнопку "Отобразить" для добавления в список отображенных значений в поле (5)
- В списке отображенных значений в поле (5) можно задать Id (номер) параметра для передачи на сервер и нажать на кнопку в ячейке "mask" для записи конфигурации в терминал.

## Поиск параметров с известными значениями

Такой метод поиска применяется для определения параметров, чьи значения точно или приблизительно известны. Например уровень топлива в баке, пробег, обороты двигателя и т.п. Задать можно как саму величину так и допустимое отклонение от нее. Например, по индикатору на панели приборов вы видите, что бак заполнен топливом на 3/4. Объем бака - 100 литров. Указатель уровня топлива может иметь погрешность до 10%. Значите имеет смысл задать значение - 75 и отклонение 10.

Также очень часто встречается ситуация, когда искомое значение умножается на какой-то коэффициент (множитель) и представлено в пакете именно в таком виде. Соответствующий пример приведен в описании Панели анализа фрагмента данных.

В данном методе поиске подразумевается, что искомая маска имеет размер и смещение кратные байту. Размер может быть от одного до четырех байт.

Например, ищем уровень топлива в баке. Предположим, что на панели приборов автомобиля отображается точное целое значение уровня.

### Поиск значений

Значение:

Отклонение:

Показывать найденные значения

#### Коэффициенты:

<input checked="" type="checkbox"/> 1	<input checked="" type="checkbox"/> 10
<input checked="" type="checkbox"/> 2	<input checked="" type="checkbox"/> 20
<input checked="" type="checkbox"/> 4	<input checked="" type="checkbox"/> 100
<input checked="" type="checkbox"/> 8	<input checked="" type="checkbox"/> 200
<input checked="" type="checkbox"/> 16	<input checked="" type="checkbox"/> 2.5
<input checked="" type="checkbox"/> 32	<input checked="" type="checkbox"/> 255/100

- Исходное состояние - зажигание включено, двигатель заглушен. Вводим в поле "Значение" текущее значения объема топлива в баке в соответствии с показаниями на приборной панели, например 45. В поле "Отклонение" вводим величину, на которую по нашему мнению индикатор может ошибаться, например 1 литр. Таким образом мы задали диапазон поиска от 44 до 46 литров.
- Выбираем коэффициенты, на которые может быть умножен объем топлива в CAN шине. Какие именно коэффициенты выбрать - конкретные рекомендации дать сложно. На первом этапе можно выбрать все коэффициенты, а затем последовательно уменьшать этот список.
- Нажимаем на кнопку "Задать". В поле (3) после этого будут подсвечиваться и отображаться найденные значения.

Следить	Скрыть	Id	Data
<input type="checkbox"/>	<input type="checkbox"/>	0075	45,6 44,7058823529412
<input type="checkbox"/>	<input type="checkbox"/>	0079	DF 0A 00 01 FC DC 89 96
<input type="checkbox"/>	<input type="checkbox"/>	009E	45
<input type="checkbox"/>	<input type="checkbox"/>	009F	45,8823529411765 46
<input type="checkbox"/>	<input type="checkbox"/>	0165	40 80 19 45 E0 00 0D 4F
<input type="checkbox"/>	<input type="checkbox"/>	0202	44,75
<input type="checkbox"/>	<input type="checkbox"/>	021F	44 44
<input type="checkbox"/>	<input type="checkbox"/>	0415	00 14 FA 00 FF 00 50 00
<input type="checkbox"/>	<input type="checkbox"/>	042B	46
<input type="checkbox"/>	<input type="checkbox"/>	0454	44
<input type="checkbox"/>	<input type="checkbox"/>	04FA	00 00 00 21 3F 1F 48 C0

- Выбираем в списке подходящий параметр и задаем маску в соответствии с подсвечиваемыми байтами. Начинаем видеть график. Можно задать коэффициент для значения маски. Например, для топлива это может быть 4.
- Заводим двигатель и ждем, когда выработается часть топлива. Либо вручную заливаем или сливаем часть топлива в/из бака. По графику и изменению значения определяем, подходит ли найденная маска.

Поиск по приведенному выше примеру займет достаточно много времени. А обороты двигателя или напряжение бортовой сети, к примеру, можно отыскать буквально за несколько минут.

## Запись отобранного параметра в конфигуратор

Этот раздел будет дописан после интеграции с конфигуратором

## Заключение

Описанными выше двумя методами можно отыскать в CAN-шине большинство параметров. Но нужно быть готовым к ситуации, когда для поиска нужного параметра придется помучиться. Например, один и тот же байт одного и того же пакета в одно время может содержать абсолютный пробег, а в другое - относительный. Все зависит от состояния предыдущего байта. И подобные ситуации не редкость. Но в конце концов все получится. Удачи!