 kvova 22 июня 2015 в 10:36

# Сетевой интерфейс для BMW

Разработка систем передачи данных

Из песочницы

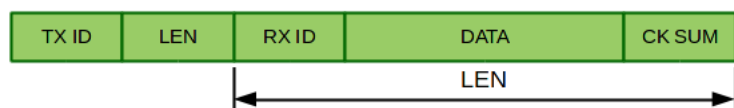
В данной статье речь будет о локальной низкоскоростной сети взаимодействия блоков управления автомобиля BMW — I/K-bus. А точнее том, как с ней могут взаимодействовать приложения из под Linux. На картинках проиллюстрирую созданный мною вариант.

Итак, передо мной встала задача расширить функциональность моего автомобиля в области информационно-развлекательной системы. Просто мне этого очень захотелось. Автомобиль хороший, но пожилой. Его создавали во времена, когда даже mp3 не был в обширном употреблении. Поэтому многих современных удобств он лишён. К тому же есть в голове дополнительные идеи, воплотив которые, я смогу подчеркнуть свою индивидуальность.

Информационно-развлекательная система выполняется на устройствах, в основе которых контроллеры с заложенными программами. Здесь называть эти устройства блоками управления. Каждый такой блок управления несёт свою функциональную нагрузку, будь то поддержание температуры салона, регулировка положения сидений, воспроизведение музыки и видео, навигация и прочее. Весь этот набор блоков управления должен взаимодействовать друг с другом, управляться с места водителя и пассажиров, передавать диагностические данные. Для этой цели и была разработана сеть I-bus. Впоследствии появилась технически идентичная сеть K-bus и их объединение I/K-bus.

Архитектура сети I-bus выполнена по схеме «общая шина», т. е. импульсы данных от узлов (блоков управления) передаются по обычному медному проводу соединённых в одной точке. Поэтому узлы должны делить общую среду передачи и передавать данные по очереди. Как определяется эта очередь или приоритетность я не знаю, полагаю просто прослушивается шина на наличие занятости и с учётом защит интервала и незанятости шины принимается решение о передаче. В «молчаливом» состоянии уровень потенциала на шине относительно корпуса составляет от 7 В до напряжения питания автомобиля. При подаче доминантного бита в шину потенциал снижается до 2 В и ниже. Битовая скорость взаимодействия узлов постоянная и составляет 9600 бит/с. Цифра знакомая из UART. Но не только в скорости передачи имеются сходства, также формат символов в I-bus соответствует одной из вариаций доступных в UART. Символ состоит из 11 бит: стартовый бит, 8 бит данных, бит чётности, стоповый бит. Эти особенности позволяют физически подключаться к шине через интерфейсы UART и SPI. Только необходимо позаботиться о преобразовании уровней сигнала с помощью простенькой схемы или готового преобразователя. Для этой цели вполне сойдёт k-line адаптер известный в диагностических интерфейсах. На физическом уровне они полностью совместимы. Кстати им и пользуюсь.

Про физический уровень сети I-bus я в общем рассказал, теперь опишу кратко канальный уровень, это если следовать последовательной многоуровневой модели OSI. Так будет логичнее. Как я упомянул ранее, для передачи используется общая среда, которую надо делить во времени и передавать данные различным адресатам. Тут можно провести аналогию с Ethernet — данные передаются кадрами в которых содержится адрес отправителя, адрес получателя, полезная нагрузка (данные) и контрольная сумма. Кадр не имеет фиксированного размера и лежит в пределах 5 — 37 символов. Формат кадра я нарисовал ниже:



Здесь TX ID — адрес отправителя, 1 символ;

LEN — размер кадра с вычетом двух первых символов, 1 символ;

RX ID — адрес получателя, 1 символ;

DATA — полезная нагрузка, 1 — 33 символа;

CK SUM — контрольная сумма, 1 символ.

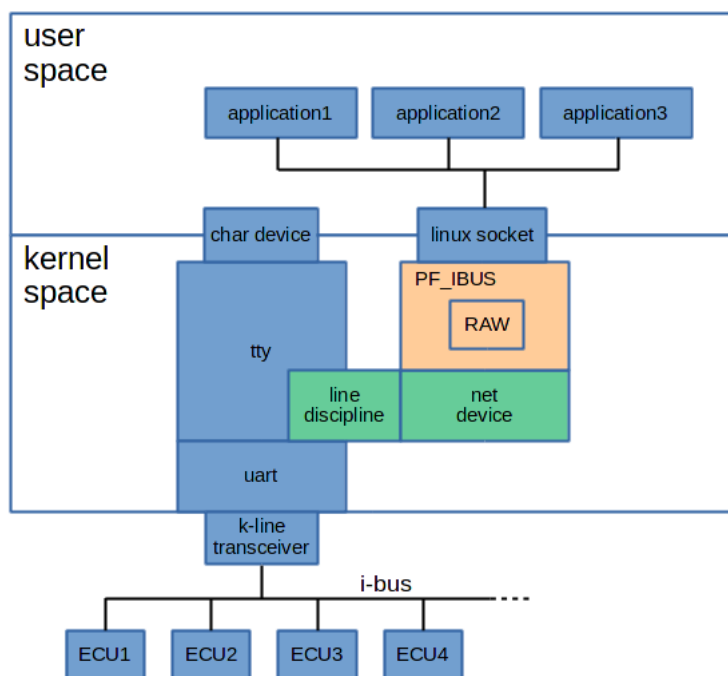
Корректность принятого кадра определяется символом контрольной суммы. Отправитель не проверяет корректность принятого кадра получателем. Возможно это делается на уровне приёма собственного кадра. Если не было коллизий и других сбоев на шине, то отправитель корректно примет собственный кадр и не предпримет попыток повторной отправки. По информации найденной мною во всемирной паузе сказано, что отправитель ждет 100 мс положительной квитанции от получателя. Увы, на практике я этого не встречал. Возможно это применяется для особенно важных сообщений и в протоколах более высокого уровня.

То что входит в полезную нагрузку кадра, относится к протоколу следующего уровня. Работу с этим протоколом лучше возложить на прикладной уровень и работать с ним в приложениях. Повторюсь ещё раз, я ищу применение взаимодействия сети I/K-bus в операционной системе на ядре Linux. Функции блоков управления будут выполнять приложения в пользовательском пространстве. Внутри операционной

системы организовать взаимодействия между приложениями не сложно, есть механизмы межпроцессного взаимодействия (IPC). Но главная задача связать их с процессами блоков управления автомобиля. За примером обратимся к более известному варианту сети взаимодействия контроллеров — CAN. В Linux эта технология развивается в двух проектах: SocketCAN и can4linux. Первый проект основывается на драйвере сетевого устройства и протоколов, выполняющих предварительную обработку кадров и связывающих сетевое устройство с интерфейсом сокетов. Второй вариант основан на символьном устройстве, подробности реализации этого проекта не знаю, так как не работал с ним. полагаю аналог can4linux для I/K-bus является драйвер tty. Достаточно настроить скорость, формат символов последовательного порта и путем чтения/записи ttyS файла будет выполняться прием/отправка данных в шину. Конечно если она подключена к последовательному порту адаптером.

Мне показался более привлекательным вариант SocketCAN, и я пошёл этим же путём. Объясню почему. Драйвер выполнен из двух независимых частей: сетевого устройства и сетевого протокола. В сетевом устройстве решаются вопросы взаимодействия с аппаратной частью и множественного доступа, а в модуле сетевого протокола — фильтрация сообщений и взаимодействия с процессами пользователя. Приложения подключаются к сети I/K-bus посредством сокетов и задачи множественного доступа и фильтрации снимаются. В принципе можно возложить фильтрацию и множественный доступ на какой-нибудь сервер, подключенный к tty устройству и не лезть в ядро. Вобщем да, но все равно не обойтись без межпроцессорного взаимодействия, а это как вариант тот же сокет. К тому же в сетевом стеке Linux решены многие задачи с очередями сообщений, которые придётся реализовывать в сервере. И ко всему этому использование сетевого устройства гармонично вписывается в философию администрирования операционной системы. Например командой `ifconfig` можно посмотреть состояние интерфейса, остановить его или запустить.

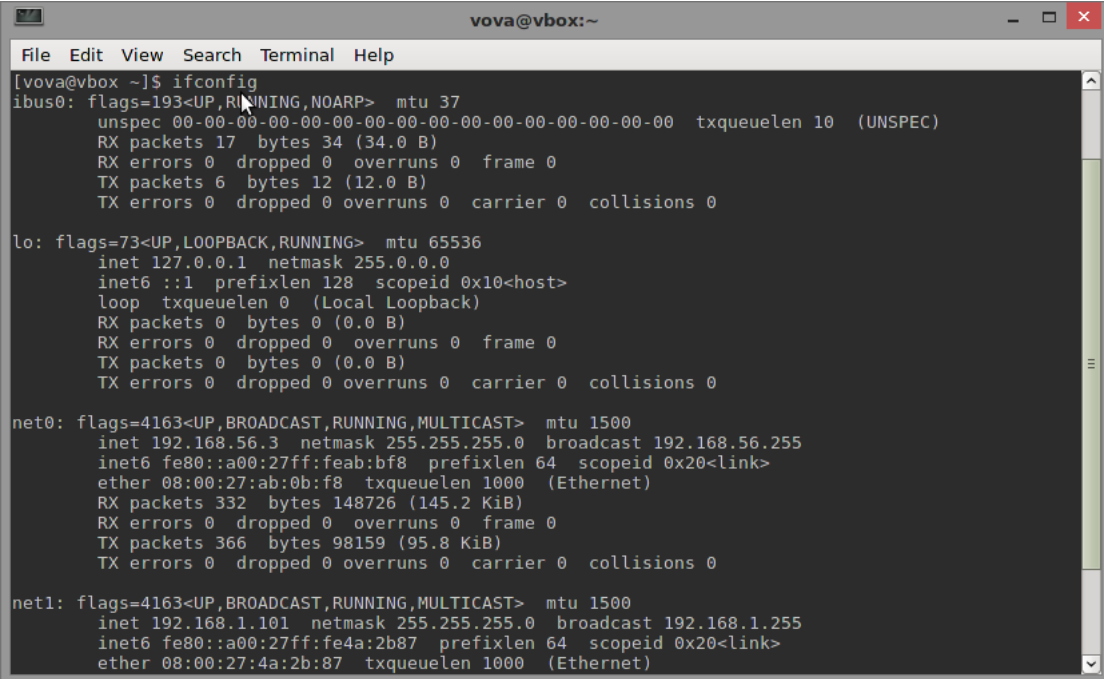
Драйвер сетевого устройства для I/K-bus я выполнил на основе `slcan`, который выполнен на основе SLIP. Я не застал те времена, когда IP пакеты передавались между компьютерами по последовательному порту. Их и сейчас можно передавать таким способом, но это не актуально. А вот передавать I/K-bus кадры таким способом хороший вариант. Драйвер tty сложен и доступ к низкоуровневой его части получить через дисциплину линии. Так делается в SLIP и `slcan`, так поступил и я, написав драйвер `slibus`. Когда через tty устройство активизируется созданная дисциплина линии, в системе появится сетевое устройство `ibusN`, где N — 0,1,2... Что бы было понятнее приведу схему. В ней зелёным отмечены квадраты, функции которых выполняет модуль ядра `slibus`.



Оранжевым цветом отмечен функционал модуля сетевого протокола. Опять же не стал изобретать велосипед и по аналогии модулей `can_raw` создал модуль `af_ibus_raw`. При загрузке этот модуль регистрирует новое семейство протоколов `PF_IBUS` и в нем же реализован сокет для полного доступа к кадру. Посредством вызова `setsockopt` можно включить фильтр принимаемых сообщений по идентификатору отправителя и получателя. По умолчанию сокет принимает все сообщения из шины.

Должен сказать, что есть одна неприятность, и заключается она в том, что для возможности загрузки этих модулей необходимо патчить

А теперь давайте посмотрим как это все работает. Загрузим модули ядра, инициализируем дисциплину линии под номером, соответствующем `slibus`, и поднимем сетевой интерфейс `ibus0`. Команда `ifconfig` в терминале покажет нам что то подобное:

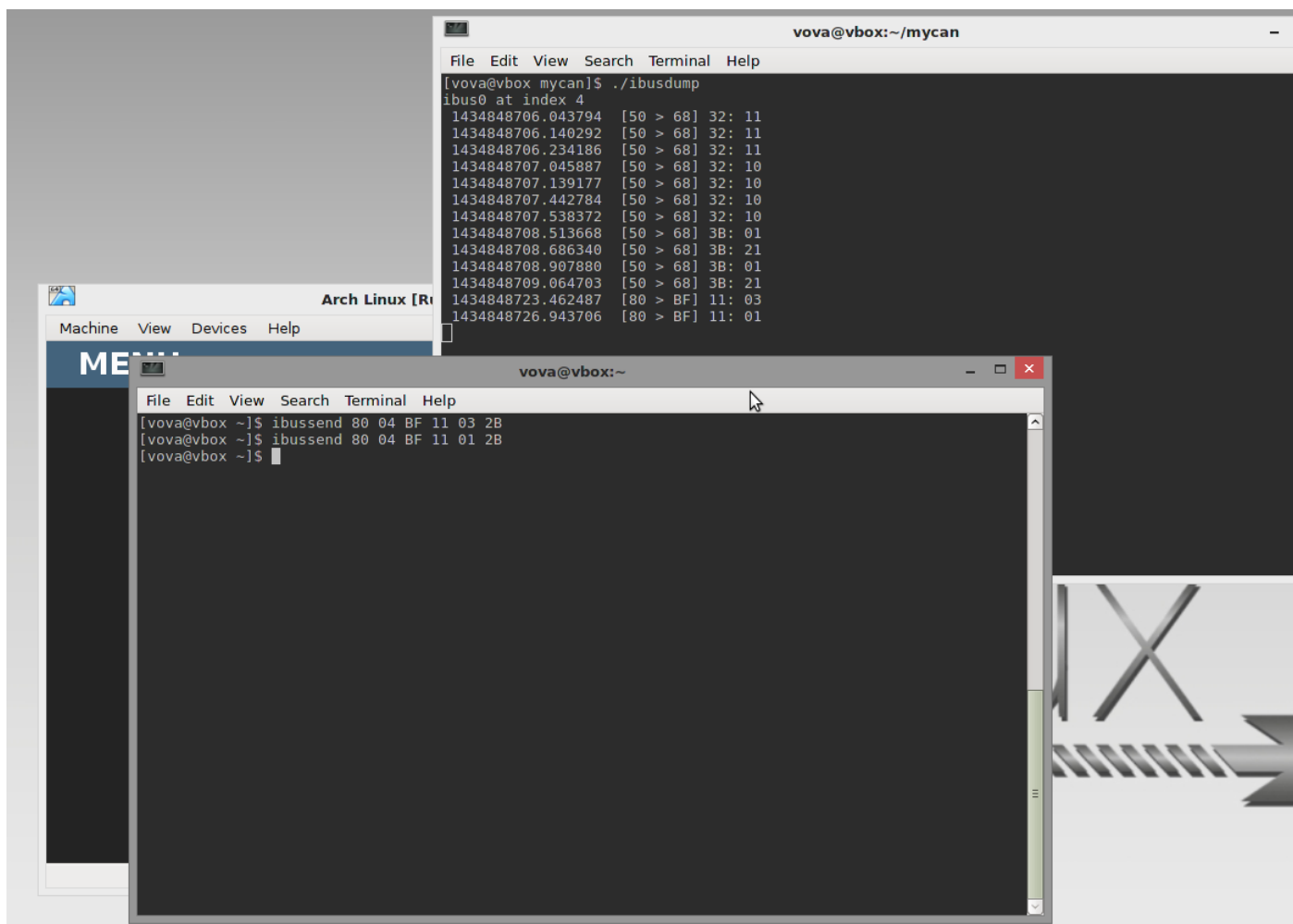


```
vova@vbox:~  
File Edit View Search Terminal Help  
[vova@vbox ~]$ ifconfig  
ibus0: flags=193<UP,RUNNING,NOARP> mtu 37  
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 10 (UNSPEC)  
    RX packets 17 bytes 34 (34.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 6 bytes 12 (12.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 0 (Local Loopback)  
    RX packets 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 0 bytes 0 (0.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
net0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.56.3 netmask 255.255.255.0 broadcast 192.168.56.255  
    inet6 fe80::a00:27ff:feab:bf8 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:ab:0b:f8 txqueuelen 1000 (Ethernet)  
    RX packets 332 bytes 148726 (145.2 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 366 bytes 98159 (95.8 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
net1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.1.101 netmask 255.255.255.0 broadcast 192.168.1.255  
    inet6 fe80::a00:27ff:fe4a:2b87 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:4a:2b:87 txqueuelen 1000 (Ethernet)
```

Как видим сетевой интерфейс успешно запущен и имеет статистику по трафику. Прежде чем сделать скриншот, я специально погонял д по интерфейсу. Пользуюсь сделанными драйверами несколько месяцев, сбоев не наблюдалось. Но есть недоработки, которые еще не устранил. К ним вернусь позже, а пока работаю над приложениями по мере свободного времени.

Сильные стороны данного подхода заключаются в ряде аспектов. Приложения запущенные в операционной системе имеют полноценнь доступ к шине и через неё же взаимодействуют друг с другом. Допустим в комплектации моего автомобиля отсутствует CD-чейнджер. И достаточно написать приложение эмулирующее это устройство. При этом оно будет воспроизводить различные форматы файлов и онл радио. Затем я захочу доукомплектовать телефонным модулем, которого у меня нет или не нравится штатный. Я просто напишу ещё программку, которая по блютуз подключится к смартфону и выполнит ввод-вывод голоса и информации в штатные места. Или создать чнибудь свое, например игра световыми приборами e-light. Таким образом приложения разрабатываются независимо друг от друга, мог устанавливаться и удаляться по желанию владельца.

На рисунке ниже показана работа программ ibusdump и ibussend. Что делают эти команды, думаю понятно из названия. Последние две строчки ibusdump показывают, что по шине передавались сообщения, которые я отправил через ibussend.



На этом остановлюсь, пожалуй. О том, что передаётся в полезной нагрузке кадра и для каких блоков управления я расскажу в другой раз.

Метки: i-bus, can, bmw

↑ +26 ↓ 106 21,3k 16



17,0

Карма

0,0

Рейтинг

24

Подписчики

Владимир Король @kvova

Инженер-программист

Github

Поделиться публикацией

#### ПОХОЖИЕ ПУБЛИКАЦИИ

17 июля 2014 в 14:47

Установка авторадара с интерфейсом SAE J1939 на автомобиль без CAN шины

↑ +15

16,5k

50

28

16 июня 2014 в 21:44

Nissan и BMW присоединятся к планам Tesla по развитию электрозаправок

↑ +49

27,9k

31

123

24 апреля 2014 в 15:15

Пользовательские типы в Qt по D-Bus

21.02.2018

Сетевой интерфейс для BMW / Хабрахабр

↑ +6

👁 6,7k

📌 32

💬 2

|                                                                                         |                                                          |
|-----------------------------------------------------------------------------------------|----------------------------------------------------------|
| РАБОТА В IT-ИНДУСТРИИ                                                                   | Мой к                                                    |
| Ведущий инженер linux<br>АйТиЭм Холдинг • Екатеринбург • Полный рабочий день            | от 80 000                                                |
| Программист SQL<br>АО Группа "СВЭЛ" • Екатеринбург • Полный рабочий день                | До 70 000                                                |
| Системный аналитик BPM системы<br>АО Группа "СВЭЛ" • Екатеринбург • Полный рабочий день | До 70 000                                                |
| Все вакансии                                                                            | <div>Создать резюме</div> <div>Разместить вакансию</div> |



Реклама

Комментарии 16

- fsmorygo 22.06.15 в 11:34

# 📌

↑

А BMW не предупреждает автовладельца о том, что его машину взламывают?
- kvova 22.06.15 в 11:54

# 📌 ↻ 🔍

↑

О таких случаях я не слышал. Возможно в новых моделях такое есть.
- GAS\_85 22.06.15 в 13:35

# 📌

↑

Мы об «F» серии говорим, или это еще про «E»?
- kvova 22.06.15 в 13:40

# 📌 ↻ 🔍

↑

Про «E». Конкретно у меня E39
- GAS\_85 22.06.15 в 13:43

# 📌 ↻ 🔍

↑

Крики «ДРУГ МОЙ!» и пошел перечитывать статью.
- Hamakev 23.06.15 в 09:29

# 📌 ↻ 🔍

↑

Подскажите, а для e36 это реализуемо? (я имею ввиду — I/K-bus там тоже?). Есть на опыты e36 1994 года.
- kvova 23.06.15 в 10:11

# 📌 ↻ 🔍

↑

Как пишут некоторые источники, I-bus появился начиная с e30. Но думаю блоков управления подключенных к этой шине очень мало на моделях ниже e38.
- en1gma 22.06.15 в 15:46

# 📌 ↻ 🔍


↑

на f-ках most. да уже на e6x most.  
а это совсем другая история...
- Ktulhy 22.06.15 в 14:11





# 📌

↑


А как на скорости 9600 бит/с передавать музыку? Или там всё хитрее устроено?

-  kvova



22.06.15 в 15:21




↑

По этой сети переаются простейшие данные. О медийном контенте и речи быть не может. Разве чтотекст. Видео и аудио передаются отдельн линиями. Как правило в аналоговой форме.
-  mafet





22.06.15 в 15:20



↑

Bay! Какая интересная тематика затронута! А можно поподробней, что хочется сделать?
-  kvova

22.06.15 в 15:41




↑





Хочется много, но сделано мало. На данный момент есть не доделанный, но работоспособный графический интерфейс. Сделан на базе Matc и Gtk2.

Есть эмулятор cd-чейнджера на базе mpd.


Здесь я опробовал e-light. Музыку накладывал уже на видеоредакторе.

Буду показывать по мере готовности.
-  velezh



23.06.15 в 18:12



↑


Пишите еще. Обо всем, что с этим связано.
-  Armanio

23.06.15 в 03:38







↑


Читал статью и не мог придумать примера того, что нужно реализовать в 39ом. Потом наткнулся на ваш пример с телефоном и понял, что это о

Очень крутая затея. Слежу!
-  DAT540





13.06.16 в 23:03



↑

Я в свое время планировал кучу фичь реализовать отсутствующих на 39том, например автоматическое складывание зеркал при закрывании машины, и соответственно раскладывание при включении зажигания. Да и тройное моргание поворотником при трогании рычага переключа поворотов тоже приятная штука отсутствующая на нем.
-  kvova

14.06.16 в 06:48



↑

Самая большая проблема — это отсутствие свободного времени, увы.

Только полноправные пользователи могут оставлять комментарии. Войдите, пожалуйста.

ИНТЕРЕСНЫЕ ПУБЛИКАЦИИ

Что это такое – BPM, и как компании его строить

↑ +6    👁 144    📖 6    💬 0

Внедрение IdM. Часть 3.2. Как построить модель доступа?

↑ +7    👁 127    📖 2    💬 0

Флаги в аргументах функций

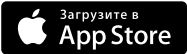
↑ +8    👁 1,2k    📖 15    💬 1

Корпорация Samsung начала массовое производство корпоративных SSD объемом в 30 ТБ GT

↑ +9    👁 4,5k    📖 6    💬 21

Иерархия IT-систем и выбор программного обеспечения для организации труда

↑ +13    👁 1,5k    📖 6    💬 5



Войти

Публикации

О сайте

Реклама

Регистрация

Хабы

Правила

Тарифы

Компании

Помощь

Контент

Пользователи

Соглашение

Семинары

Песочница

Конфиденциальность



© 2006 – 2018 «ТМ»

Служба поддержки

Мобильная версия

