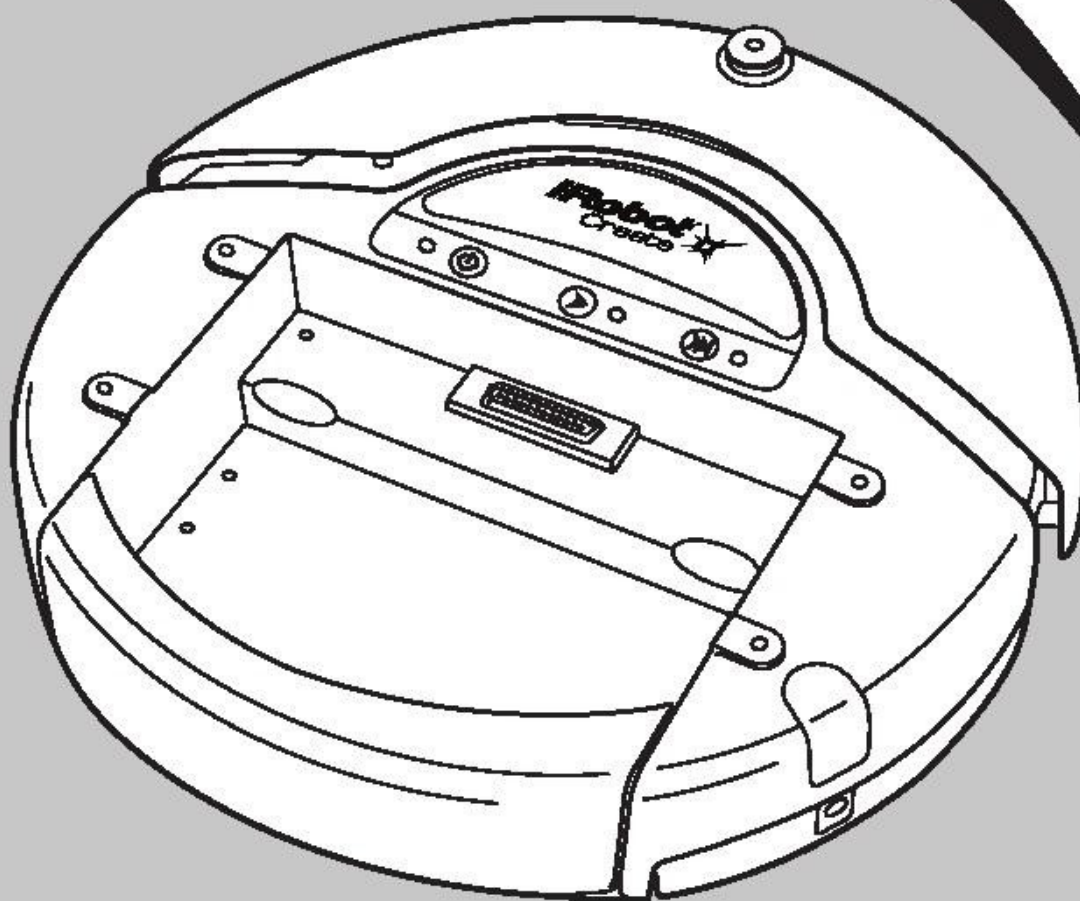


# iRobot конструктор ОТКРЫТЫЙ ИНТЕРФЕЙС



i♥Robots

Перевод на русский язык :  
Компания **РобоПарк** [www.robopark.ru](http://www.robopark.ru)

**iRobot**

[www.irobot.com](http://www.irobot.com)

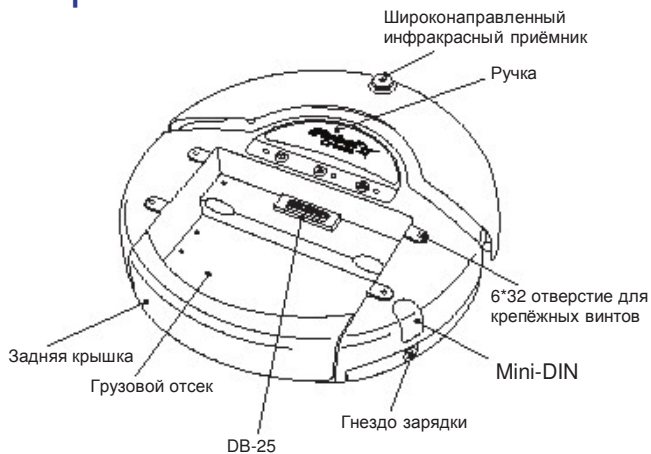
## Содержание

iRobot конструктор открытый интерфейс. Обзор.....	3
Физические соединения.....	4
Mini-DIN коннектор .....	4
Разъем грузового отсека .....	4
Настройки последовательного порта.....	5
iRobot Create режимы открытого интерфейса .....	6
Команды открытого интерфейса .....	7
Команды запуска .....	7
Режимы команд .....	7
Демонстрационные команды.....	8
Команды управления приводами и устройствами .....	9
Команды ввода .....	13
Команды сценариев .....	15
Команды ожидания .....	15
iRobot конструктор открытый интерфейс, пакеты сенсоров .....	17
iRobot конструктор открытый интерфейс, список команд .....	22
iRobot конструктор открытый интерфейс, список пакетов сенсоров .....	24

## iRobot конструктор открытый интерфейс Обзор

Открытый интерфейс (ОИ) конструктора состоит из электронного и программного интерфейсов. Данные интерфейсы контролируют поведение и считывают информацию с сенсоров. Электронный интерфейс включает 7 pin Mini-DIN и DB-25 разъемы для обмена данными, электронные датчики, приводы, такие как, двигатели и световые индикаторы. Программный интерфейс позволяет Вам манипулировать поведением конструктора. Он считывает информацию с сенсоров при помощи серии команд, включает команды режима, команды привода, звуковые команды, демонстрационные команды и команды опроса сенсоров. Эти команды посылаются на последовательный порт конструктора с компьютера или микроконтроллера посредством Mini-DIN разъема или DB-25 разъема грузового отсека.

### Устройство

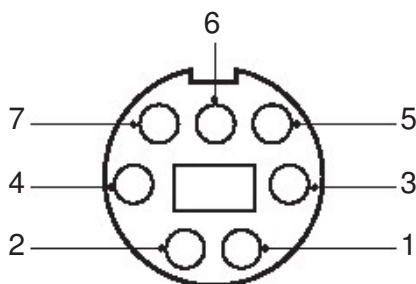


## Физические соединения

Для использования ОИ, компьютер или микроконтроллер, способный выработать последовательные команды должен быть подключен к внешнему разъёму Mini-DIN или DB-25 грузового отсека. Эти разъёмы обеспечивают двухстороннее, последовательное соединение на TTL (0 – 5V) уровнях. Коннекторы также обеспечивают нерегулируемое прямое соединение с аккумулятором конструктора, которое Вы можете использовать для сопрягаемых устройств. Разъём грузового отсека, также обеспечивает регулируемое 5V напряжение на нескольких входящих и выходящих контактах (смотрите таблицу ниже). Mini-DIN коннектор расположен на правой стороне конструктора, под пластиковой защитой, коннектор DB-25 расположен по центру, спереди грузового отсека.

### Mini-DIN коннектор

Эта схема показывает назначение (вид сверху) выводов розетки в конструкторе. Имейте в виду, что разъёмы 5,6 и 7 расположены ближе к краю корпуса.



Pin	Имя	Описание
1	Vpwr	+ аккумуляторной батареи
2	Vpwr	+ аккумуляторной батареи
3	RXD	0 – 5В последовательный вход
4	TXD	0 – 5В последовательный выход
5	BRC	Установка скорости передачи данных
6	GND	- аккумуляторной батареи
7	GND	- аккумуляторной батареи

Так как, RXD и TXD контакты используют логическое напряжение 0 – 5В, а последовательные порты ПК используют напряжения уровня rs232 (+/-12В), необходима коррекция уровней напряжения. Поэтому используйте iRobot кабель последовательного порта для конструктора, а не обычный кабель последовательного порта. iRobot кабель последовательного порта для конструктора содержит необходимое устройство для согласования напряжений, в то время как обычный кабель производит непосредственное соединение.

### Разъём грузового отсека

Разъём грузового отсека, расположенный в середине на его передней части, содержит 25 контактов, которые Вы можете использовать для подключения периферийных устройств, таких как дополнительные датчики. Разъём грузового отсека имеет 4 цифровых входа, аналоговый вход, 3 цифровых выхода, 3 выхода с большим током (например для управления двигателями), индикатор зарядки, вход для управления питанием, последовательный вход (Tx) и последовательный выход (Rx), опорное напряжение 5В, + аккумуляторной батареи и – аккумуляторной батареи.



Pin	Имя	Описание
1	RXD	0 – 5В последовательный вход
2	TXD	0 – 5В последовательный выход
3	Кнопка вкл./выкл.	Включает и выключает конструктор при изменении напряжения от низкого к высокому
4	Аналоговый вход	0 – 5В аналоговый вход
5	Цифровой вход 1	0 – 5В цифровой вход
6	Цифровой вход 3	0 – 5В цифровой вход
7	Цифровой выход 1	0 – 5В, 20 mA цифровой выход
8	5В	Стабилизированное напряжение 5В (100 mA) - опорное напряжение, когда конструктор включен
9	Vpwr	Напряжение батареи, ток до 0.5А
10	Vpwr	Обеспечивает питание от аккумулятора (1.5 А) при включённом конструкторе.
11	Vpwr	Обеспечивает питание от аккумулятора (1.5 А) при включённом конструкторе.
12	Vpwr	Обеспечивает питание от аккумулятора (1.5 А) при включённом конструкторе.
13	Зарядка робота	При зарядке - высокий уровень (5V)
14	GND	- питания
15	Детектор устройства/ установка скорости	0-5В цифровой вход, который может быть использован для изменения скорости передачи данных (подробности ниже)
16	GND	- питания
17	Цифровой вход 0	0 – 5В цифровой вход
18	Цифровой вход 2	0 – 5В цифровой вход
19	Цифровой выход 0	0 – 5В, 20 mA цифровой выход
20	Цифровой выход 2	0 – 5В, 20 mA цифровой выход
21	GND	- питания
22	Управление 0	Управление устройством 0 (0.5А)
23	Управление 1	Управление устройством 1 (0.5А)
24	Управление 2	Управление устройством 2 (1.5А)
25	GND	- питания

## Настройки последовательного порта

Скорости обмена данными: 57600 или 19200 бод (смотрите ниже)

Количество бит данных: 8

Стоповые биты: 1

По умолчанию, iRobot конструктор осуществляет передачу данных на скорости 57600 бод.

Если, Вы используете микроконтроллер, который не поддерживает скорость 57600, то переключение на скорость 19200 можно осуществить двумя способами:

Способ 1:

При включении конструктора, удерживайте нажатой клавишу Play. Примерно, через 4 секунды, конструктор проиграет мелодию с нисходящей тональностью. Конструктор будет производить обмен данными на скорости 19200 до тех пор, пока питание не будет отключено; пока не будет извлечён и заново установлен аккумулятор; пока напряжение аккумулятора не упадёт до минимума, не позволяющего нормально работать; или пока скорость не будет изменена посредством открытого интерфейса.

Способ 2:

Используйте контакты разъемов для изменения скорости: контакт 15 на разъёме грузового отсека; контакт 5 на Mini-DIN разъёме. После включения конструктора, подождите 2 секунды и подайте на контакт импульсы изменения скорости (серия из 3 импульсов низкого уровня). Каждый импульс должен быть более 50, но менее 500 миллисекунд. Конструктор будет производить обмен данными на скорости 19200 до тех пор, пока питание не будет отключено; пока не будет извлечён и заново установлен аккумулятор; пока напряжение аккумулятора не упадёт до минимума, не позволяющего нормально работать; или пока скорость не будет изменена посредством открытого интерфейса.

## iRobot конструктор режимы открытого интерфейса

ОИ конструктора имеет четыре рабочих режима: выключено (**off**), пассивный, безопасный и полный. После смены батареи или при установке элемента питания в первый раз, ОИ находится в режиме "**off**". Когда он выключен, то скорость обмена задана по умолчанию (57600, смотрите установки последовательного порта выше) для получения команды **Start**. При прохождении команды **Start**, Вы можете войти в любой из четырёх рабочих режимов, посредством пересылки команды режима на ОИ. Вы также можете переключаться между рабочими режимами в любое время, посредством отправки команды с режимом, который Вы хотите использовать на ОИ.

### Пассивный режим (**Passive**)

При отправке команды **Start** или любой другой команды из демо-программ (которые запускают демонстрацию, например, **Spot Cover**, **Cover**, **Cover** и **Dock** или **Demo**), ОИ входит в пассивный режим. При нахождении ОИ в пассивном режиме, Вы можете опросить и получить данные о состоянии датчиков, используя любую из команд для сенсоров, но не сможете изменить текущие настройки командами для приводов (двигателей, громкоговорителя, индикаторов, управляющих двигателей, цифровых выводов). Для изменения режимов работы приводов, Вы должны переключиться из пассивного режима на полный или безопасный.

Находясь в пассивном режиме, Вы можете опросить сенсоры конструктора, наблюдать за исполнением одной из десяти встроенных демонстраций и осуществлять зарядку аккумулятора.

### Безопасный режим (**Safe**)

Когда Вы отправляете команду **Safe** на ОИ, конструктор входит в безопасный режим. Безопасный режим предоставляет Вам полный контроль над конструктором, за исключением следующих функций, отвечающих за безопасность:

- Обнаружение препятствий в виде уступов или ступеней во время движения вперёд (или движения назад с маленьким радиусом вращения, меньшим, чем радиус самого робота).
- Обнаружение блокировки колеса (любого колеса).
- Зарядка (подключение к сети).

Если, происходит, одна из вышеописанных ситуаций в безопасном режиме ОИ, то конструктор останавливает все двигатели и возвращает прибор в пассивный режим.

Если, в безопасном режиме не было отправлено никаких команд, то конструктор ожидает с выключенными двигателями и индикаторами и не реагирует на нажатие клавиш Play или Advance. Он также не реагирует на информацию от сенсоров.

**Имейте в виду**, что при входе в безопасный режим, зарядка отключается.

### Полный режим

При отправке команды **Full** на ОИ, конструктор входит в полный режим. Полный режим предоставляет Вам полный контроль как над конструктором, так и над его приводами и всеми функциями, отвечающими за безопасность, которые не отключаются в безопасном режиме. Для перевода ОИ назад в безопасный режим, Вы должны отправить команду **Safe**.

Если, в полном режиме не было послано никаких команд на ОИ, то конструктор ожидает с выключенными двигателями и индикаторами не реагирует на нажатие клавиш Play или Advance. Он также не реагирует на информацию от сенсоров.

**Имейте в виду**, что при входе в безопасный режим, зарядка отключается.

## Команды открытого интерфейса

Здесь приведён список команд ОИ конструктора. Каждая команда начинается с однобайтового кода (**opcode**). После некоторых команд должны следовать байты данных. Все команды ОИ конструктора, включая необходимые байты данных, описаны ниже.

**Внимание.** Всегда отправляйте необходимое количество байтов данных для команды, иначе процессор останется в режиме ожидания до тех пор, пока не пройдут все необходимые байты данных.

### Команды запуска

Следующие команды запускают открытый интерфейс и подготавливают его к использованию.

Старт (Start) Opcode:128 Байт данных: 0

Эта команда запускает ОИ. Перед отправкой какой-либо команды на ОИ, Вы всегда предварительно должны послать команду Start.

- Последовательный код команды: [128].
- Вводится из режимов: пассивный, безопасный или полный.
- Меняет режим на: Пассивный. При начале работы оповещает однократным звуковым сигналом о выходе из режима **off**.

Установка скорости (Baud) Opcode: 129 Байт данных: 1

Эта команда устанавливает скорость в битах в секунду (бод или bps), при которой команды ОИ и данные передаются в устройство. По умолчанию, установка скорости при включённом электропитании равна 57600 bps, но стартовая установка скорости может быть снижена до 19200 путём нажатия и удерживания клавиши Play во время запуска конструктора, до тех пор, пока Вы не услышите мелодию с снисходящим тоном. После того, как установка скорости была изменена, она сохраняется до тех пор, пока конструктор не вышел из рабочего режима при помощи нажатия клавиши вкл/выкл или пока напряжение аккумулятора не упадёт до минимума. Перед отправкой команд на новой скорости, Вам необходимо подождать в течение 100 миллисекунд после отправки команды изменения скорости.

**Внимание.** На скорости 115200, должно быть по крайней мере 200µs между установкой каждого из параметров. В противном случае параметры могут быть не приняты.

- Последовательный код: [129] [код скорости]
- Вводится из режимов: пассивный, безопасный или полный.
- Не переводит в другой режим
- Байт данных скорости: код (0 - 11)

Код	Диапазон скорости
0	300
1	600
2	1200
3	2400
4	4800
5	9600
6	14400
7	19200
8	28800
9	38400
10	57600
11	115200

### Команды режима

Конструктор имеет четыре рабочих режима: выключено, пассивный, безопасный и полный. Конструктор включается в пассивном режиме. Следующие команды меняют режим ОИ.

Безопасный (Safe) Opcode: 131 Байт данных: 0

Эта команда устанавливает ОИ в безопасный режим, предоставляя пользователю контроль над конструктором. Она отключает все световые индикаторы. ОИ может находиться в пассивном, безопасном, или полном режиме для принятия этой команды.

- Последовательный код команды: [131]
- Вводится из режимов: пассивный, безопасный или полный
- Меняет режим на: безопасный

**Имейте в виду:** команда **Control** (130) идентична команде **Safe**. Команда **Control** не оказывает воздействия, она присутствует для совместимости с открытым интерфейсом Roomba. Вместо неё используйте команду **Safe**.

Полный (Full) Opcode: 132 Байт данных: 0

Эта команда предоставляет Вам полный контроль над конструктором, путём установки ОИ в полный режим, при этом отключается защита от опрокидывания, защита блокировки колёс и внутренние функции безопасной зарядки. Таким образом, в полном режиме конструктор исполняет любую команду, которую, Вы ему отправляете, даже в тех случаях, когда внутренняя зарядка подключена к сети электропитания или робот обнаружил уступ, или колёса были заблокированы.

- Последовательный код команды: [132]
- Вводится из режимов: пассивный, безопасный или полный
- Меняет режим на: полный

## Демонстрационные команды

Следующие команды предназначены для запуска встроенных демонстрационных функций конструктора.

**Demo** Opcode: 136 Байт данных: 1

Эта команда запускает демонстрационную функцию.

- Последовательный код: [136] [вариант демонстрации]
- Вводится из режимов: пассивный, безопасный или полный.
- Изменяет на режим: пассивный
- Байт данных демонстрации : (-1 - 9)

### Название демонстрации, описания и номера

Номер	Демонстрация	Описание
-1 (255)	Abort current demo	Отменяет демонстрацию на конструкторе
0	Cover	Передвижение по всей комнате, с помощью комбинации режимов, таких как, хаотичное движение, движение вдоль стен, движение по спирали.
1	Cover and Dock	Идентично передвижению по всей комнате с одним исключением. Если робот видит инфракрасный сигнал от базовой станции, то он использует этот сигнал для стыковки с базовой станцией и самостоятельной зарядки.
2	Spot Cover	Конструктор передвигается по околостартовой зоне, двигаясь по спирали от центра и к центру.
3	Mouse	Конструктор двигается в поисках стены. Найдя стену, конструктор двигается вдоль стены по периметру комнаты.
4	Drive Figure Eight	Конструктор продолжительно движется «по восьмёрке».
5	Wimp	Конструктор двигается вперед, если его подтолкнули сзади. При движении, объезжает возникшие препятствия.
6	Home	<p>Конструктор направляется к виртуальной стене, если задняя и боковые стороны приёмника ИК-излучения закрыты (например, изоляцией).</p> <p>Виртуальная стена излучает ИК-сигналы, которые конструктор воспринимает своим всенаправленным приёмником, расположенным сверху бампера.</p> <p>Если, Вы хотите, чтобы конструктор двигался к виртуальной стене, а не от нее, то заблокируйте всю поверхность инфракрасного датчика, оставив спереди небольшую щель.</p> <p>Конструктор вращается для обнаружения виртуальной стены, затем движется по направлению к ней. При ударе о стену или другое препятствие, конструктор останавливается.</p>

Номер	Демонстрация	Описание
7	Tag	Идентична предыдущей демонстрации. Добавлена функция многократного приближения конструктора к виртуальным стенам, сталкиваясь с одной, затем с другой.
8	Pachelbel	Конструктор проигрывает мелодии, когда активированы датчики антипрокидывания (обнаружен уступ).
9	Banjo	<p>Конструктор проигрывает набор звуков для каждого из его четырёх датчиков антипрокидывания. Выберите аккорд, используя бампер следующим образом:</p> <ul style="list-style-type: none"> <li>• Нет бампера: соль мажор</li> <li>• Правый/левый бампер: ре мажор 7</li> <li>• Оба бампера (центр): до мажор</li> </ul>

Вы также можете вызвать демонстрационные функции Cover, Cover and Dock, Spot, используя коды, описанные ниже. Эта функция присутствует для показа совместимости с OI Roomba.

**Cover** Opcode: 135 Байта данных нет

Эта команда запускает демонстрацию **Cover**.

- Последовательный код: [135]
- Вводится из режимов: пассивный, безопасный и полный.
- Изменяет режим на: пассивный

**Cover and Dock** Opcode: 143 Байта данных нет

Эта команда запускает режимы **Cover and Dock**.

- Последовательный код: [143]
- Вводится из режимов: пассивный, безопасный или полный
- Изменяет режим на: пассивный

**Spot** Opcode: 134 Байта данных нет

Эта команда запускает режим **Spot Cover**.

- Последовательный код: [134]
- Вводится из режимов: пассивный, безопасный или полный.
- Изменяет на режим: пассивный

## Команды управления приводами и устройствами.

Следующие команды контролируют приводы и устройства конструктора: колёса, динамик, светодиоды, цифровые выходы и выходы для дополнительных двигателей.

### Drive Opcode: 137 Байты данных: 4

Эта команда контролирует ведущие колёса конструктора. Она использует четыре байта данных, интерпретируемые как два 16 битных числа. Первые два байта определяют среднюю скорость ведущих колёс в миллиметрах в секунду, старший байт посылается первым. Следующие два байта определяют радиус, по которому конструктор совершает разворот, в миллиметрах. Большой радиус заставляет конструктор двигаться прямее, при меньшем радиусе конструктор будет осуществлять более крутые повороты. Радиус измеряется от центра круга вращения до центра конструктора. Команда Drive с положительными скоростью и радиусом заставляет конструктор двигаться вперёд с отклонением налево. Отрицательный радиус заставляет конструктор двигаться вперёд с отклонением направо. В особых случаях, конструктор вращается на месте или двигается прямо, как описано ниже. Отрицательная скорость заставляет конструктор двигаться назад.

**Имейте в виду:** внутренние и внешние ограничения могут препятствовать выполнению конструктором некоторых команд движения. Например, конструктор при движении с большим радиусом не сможет развивать полную скорость.

- Последовательный код: [137] [стар. байт скорости] [млад. байт скорости] [стар. байт радиуса] [млад. байт радиуса]
- Вводится из режимов: безопасный или полный
- Не переводит в другой режим
- Байт данных скорости (-500 – 500 mm/s)
- Байт данных радиуса (-2000 – 2000 mm)

#### Особые случаи:

Прямо = 32768 или 32767 = hex 8000 или 7FFF

Вращение на месте по часовой стрелке = hex FFFF

Вращение на месте против часовой стрелки = hex 0001

#### Пример:

Для реверсивного движения со скоростью -200 mm/s при вращении с радиусом 500mm, отправьте следующую последовательность байтов:

[137] [255] [56] [1] [244]

Скорость = -200 = hex FF38 = [hex FF] [hex 38] = [255] [56]

Радиус = 500 = hex 01F4 = [hex 01] [hex F4] = [1] [244]

### Drive Direct Opcode: 145 Байты данных: 4

Эта команда позволяет Вам контролировать независимое движение колёс конструктора вперёд и назад. Она использует четыре байта данных, интерпретируемые как два 16 битных числа. Первые два байта определяют скорость правого колеса в миллиметрах в секунду, старший байт посылается первым. Следующие два байта определяют скорость левого колеса в том же формате. Положительная скорость заставляет робот двигаться вперёд, в то время как отрицательная скорость заставляет двигаться назад.

- Последовательный код: [145] [старший байт двигателя справа] [младший байт двигателя справа] [старший байт двигателя слева] [младший байт двигателя слева]
- Вводится из режимов: безопасный или полный
- Не переводит в другой режим
- Байт данных правого колеса: скорость правого колеса (-500 – 500 mm/s)
- Байт данных левого колеса: скорость левого колеса (-500 – 500 mm/s)

### LEDs Opcode: 139 Байты данных: 3

Эта команда контролирует светодиоды конструктора. Состояние светодиодов **Play** и **Advance** определяется двумя битами в первом байте данных. Состояние светодиода электропитания определяется следующими двумя байтами данных: один для цвета, другой для яркости.

- Последовательный код: [139] [биты светодиодов] [цвет] [интенсивность]
- Вводится из режимов: безопасный или полный
- Не переводит в другой режим
- Байт данных 1: биты светодиодов (0 – 10)

**Advance** и **Play** используют зелёные светодиоды. 0 = выкл, 1 = вкл

Бит	7	6	5	4	3	2	1	0
LEDs	н/д	н/д	н/д	н/д	Advance	н/д	Play	н/д

Для индикации электропитания используется двухцветный (красный/зелёный) светодиод. Интенсивность и цвет этого светодиода устанавливаются с 8-битным разрешением.

- Байт данных 2 : цвет светодиода (0 – 255) 0 = зелёный, 255 = красный. Средние значения обеспечивают промежуточные цвета (оранжевый, жёлтый и т.д.).
- Байт данных 3 : яркость светодиода (0 – 255) 0 = выкл., 255 = полная интенсивность. Средние значения дают среднюю яркость.

#### Пример:

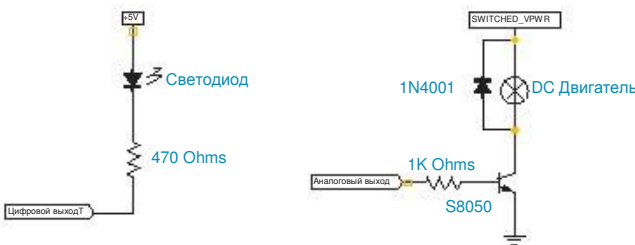
Для включения светодиода Advance и зелёного светодиода электропитания с половинной яркостью, отправьте последовательность байтов [139] [8] [0] [128].

**Digital Outputs** Opcode: 147 Байт данных: 1

Эта команда контролирует состояние 3 цифровых выходов на 25 контактном разъёме грузового отсека. Цифровые выходы рассчитаны на ток до 20 мА.

- Последовательный код: [147] [выходные биты]
- Вводится из режимов: безопасный или полный
- Не переводит в другой режим
- Цифровые выходы байт данных 1: выходные биты (0 –7); 0 = низкий (0V); 1 = высокий (5V).

**Примерная схема**



**Предупреждение:** при включении конструктора, цифровые выходы находятся в высоком уровне в течение 3 секунд до окончания загрузки процессора.

Бит	7	6	5	4	3	2	1	0
Выход						digital-out-2 (pin 20)	digital-out-1 (pin 7)	digital-out-0 (pin 19)

**PWM low side drivers** Opcode: 144 Байты данных: 3

Эта команда позволяет Вам изменять мощность дополнительных двигателей. Каждым байтом данных Вы определяете мощность двигателя (max 128). Например, если Вы хотите управлять двигателем напряжением 25% от напряжения аккумулятора, то вычислите значения байта данных по следующей формуле  $128(\text{max}) * 25\% = 32$ .

- Последовательный код [144] [рабочий цикл двигателя 2] [рабочий цикл двигателя 1] [рабочий цикл двигателя 0]
- Вводится из режимов: безопасный или полный
- Не переводит в другой режим
- Байт данных 1: рабочий цикл двигателя 2 (0 - 128)
- Байт данных 2: рабочий цикл двигателя 1 (0 - 128)
- Байт данных 3: рабочий цикл двигателя 0 (0 - 128)

**Пример:**

Для того, чтобы включить двигатель 2 на 25% и двигатель 0 на 100%, отправьте последовательный код [144] [32] [0] [128]

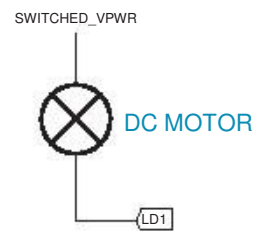
**Low side drivers** Opcode: 138 Байты данных: 1

Эта команда позволяет Вам контролировать (вкл/выкл) 3 дополнительных двигателя. Состояние каждого из двигателей определяется одним битом в байте данных.

Питание дополнительных двигателей 0 и 1 обеспечивается током до 0.5А., двигателя 2 током до 1.5 А. Если нагрузка превышает допустимое значение, то ток ограничивается и выставляется сообщение в пакете сенсоров номер 14.

- Последовательный код: [138] [биты двигателя]
- Вводится из режимов: безопасный или полный
- Не переводит в другой режим
- Байт данных 1: биты двигателя (0 – 7)

**Примерная схема**



0 = выкл, 1 = вкл , мощность 100%

Бит	7	6	5	4	3	2	1	0
Выход						двигатель2 (pin 24)	двигатель 1 (pin 22)	двигатель 0 (pin 23)

**Пример:**

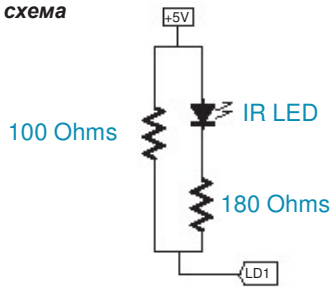
Для включения дополнительного двигателя 1, отправьте последовательный код [138] [2].

**Имейте в виду:** контроль скорости дополнительных двигателей осуществляется командой **PWM**. Эта команда совместима с ОИ Roomba.

**Send IR** Opcode: 151 Байты данных: 1

Эта команда отсылает *передаваемый байт* на контакт №23 разъёма грузового отсека, используя формат, необходимый для инфракрасного приёмника конструктора. Используйте шунтирующий резистор около 100 Ом, включив его параллельно инфракрасному светодиоду и токоограничивающему резистору 180 Ом (смотрите схему ниже).

- Последовательный код: [151] [*передаваемый байт*]
- Вводится из режимов: безопасный или полный
- Не переводит в другой режим
- Байт данных IR: возможные значения отправляемого байта (0 - 255)

**Примерная схема****Song** Opcode: 140 Байты данных: 2N+2,

где N является количеством нот в мелодии

Эта команда позволяет Вам записать до 16 песен, которые Вы сможете проиграть позже. Каждой мелодии задан свой номер. Команда **Play** использует этот номер для выбора мелодии. Каждая мелодия может содержать до 16 нот. Каждая нота имеет свой номер в формате MIDI. Её продолжительность задаётся долями секунды. Количество байтов данных варьируется в зависимости от длины песни. Мелодия из одной ноты равняется 4 байтам данных. Для каждой дополнительной ноты в мелодии добавляется 2 байта данных.

- Последовательный код: [140] [номер песни] [длина песни] [номер ноты 1] [продолжительность ноты 1] [номер ноты 2] [продолжительность ноты 2], и т.д.

- Вводится из режимов: пассивный, безопасный или полный.
- Не переводит в другой режим

- Первый байт данных песни: номер песни (0 – 15)

Каждая песня имеет свой номер. Если, Вы посылаете новую песню на этот номер, то старая песня удаляется.

- Второй байт данных песни: длина песни (1 – 16)  
Длина песни должна соответствовать количеству нот.

- Нечётные байты данных кроме первого: номер ноты (31 – 127)

Конструктор проигрывает мелодию, согласно номерам формата MIDI. Самая низкая нота, проигранная конструктором, имеет номер 31. Конструктор воспринимает и остальные ноты кроме 31 – 127, но не способен их воспроизвести.

- Чётные байты данных кроме второго: длительность ноты (0 – 255)

Длительность музыкальной ноты можно изменять с шагом 1/64 секунды. Пример: музыкальная нота длиной в пол секунды имеет значение 32, т.к.  $1/64 * 32 = 1/2$ .

Номера нот и их частоты в Герцах

Номер	Нота	Частота	Номер	Нота	Частота	Номер	Нота	Частота
			60	C	261.6	96	C	2093.0
			61	C#	277.2	97	C#	2217.5
			62	D	293.7	98	D	2349.3
			63	D#	311.1	99	D#	2489.0
			64	E	329.6	100	E	2637.0
			65	F	349.2	101	F	2793.8
			66	F#	370.0	102	F#	2960.0
31	G	49.0	67	G	392.0	103	G	3136.0
32	G#	51.0	68	G#	415.3	104	G#	3322.4
33	A	55.0	69	A	440.0	105	A	3520.0
34	A#	58.3	70	A#	466.2	106	A#	3729.3
35	B	61.7	71	B	493.9	107	B	3951.1
36	C	65.4	72	C	523.3	108	C	4186.0
37	C#	69.3	73	C#	554.4	109	C#	4434.9
38	D	73.4	74	D	587.3	110	D	4698.6
39	D#	77.8	75	D#	622.3	111	D#	4978.0
40	E	82.4	76	E	659.3	112	E	5274.0
41	F	87.3	77	F	698.5	113	F	5587.7
42	F#	92.5	78	F#	740.0	114	F#	5919.9
43	G	98.0	79	G	784.0	115	G	6271.9
44	G#	103.8	80	G#	830.6	116	G#	6644.9
45	A	110.0	81	A	880.0	117	A	7040.0
46	A#	116.5	82	A#	932.3	118	A#	7458.6
47	B	123.5	83	B	987.8	119	B	7902.1
48	C	130.8	84	C	1046.5	120	C	8372.0
49	C#	138.6	85	C#	1108.7	121	C#	8869.8
50	D	146.8	86	D	1174.7	122	D	9397.3
51	D#	155.6	87	D#	1244.5	123	D#	9956.1
52	E	164.8	88	E	1318.5	124	E	10548.1
53	F	174.6	89	F	1396.9	125	F	11175.3
54	F#	185.0	90	F#	1480.0	126	F#	11839.8
55	G	196.0	91	G	1568.0	127	G	12543.9
56	G#	207.7	92	G#	1661.2			
57	A	220.0	93	A	1760.0			
58	A#	233.1	94	A#	1864.7			
59	B	246.9	95	B	1975.5			

---

**Play Song** Opcode: 141 Байты данных: 1

Эта команда позволяет Вам выбрать песню из списка записанных песен на конструкторе. Вы должны добавить одну или более песен на конструктор, используя команду **Song**, чтобы команда **Play** работала. Эта команда не исполняется, если мелодия уже проигрывается. Подождите, пока запущенная мелодия закончится, только после этого отправляйте команду. Имейте в виду, что при помощи пакетов сенсоров можно проверить готовность конструктора принять эту команду.

- Последовательный код: [141] [номер песни]
- Вводится из режимов: безопасный или полный
- Не переводит в другой режим
- Байт данных проигрывания песни : номер песни (0 – 15)  
Номер песни, выбранной для воспроизведения.

**Входящие команды (Input Commands)**

Следующие команды позволят Вам считывать состояние встроенных датчиков конструктора, цифровых и аналоговых входов и некоторых других меняющихся внутренних функций. Конструктор обновляет эти характеристики каждые 15 ms. Не отправляйте эти команды чаще указанного интервала.

---

**Sensors** Opcode: 142 Байты данных: 1

Для отправки пакета байтов данных, этой команде необходим ОИ. Имеется 43 разных пакета данных. Каждый представляет характеристику определённого сенсора или группы сенсоров.

За более подробной информацией о пакетах сенсоров обратитесь к разделу "Пакеты сенсоров открытого интерфейса конструктора" (Create Open Interface Sensor Packets).

- Последовательный код: [142] [идентификатор пакета]
- Вводится из режимов: пассивный, безопасный или полный
- Не переводит в другой режим
- Байт данных сенсоров: номер (идентификатор) пакета (0 - 42)  
Определяет, какой из 43-х пакетов данных с датчиков следует отправить, используя ОИ. При значении 6 будет выведен пакет с данными обо всех датчиках. Показания от 0 до 5 указывают на определенные подгруппы данных с датчиков (смотрите таблицу на страницах 24, 25).

---

**Query List** Opcode: 149 Байты данных: N + 1,

где N номер необходимых пакетов.

Эта команда позволяет Вам вывести список пакетов данных с сенсоров. Результат выводится сразу, как и после команды **Sensors**. Робот выдает пакеты в том порядке, который был определён Вами.

- Последовательный код: [149] [количество пакетов] [пакет 1] [пакет 2]...[пакет N]
- Вводится из режимов: пассивный, безопасный или полный
- Не переводит в другой режим
- первый байт данных листа опроса: количество запрошенных пакетов (0 - 255)
- Следующие байты данных: номера запрошенных пакетов (0 - 42)

**Пример:**

Для получения состояния левого датчика антипрокидывания (пакет 9) и датчика виртуальной стены (пакет13), отправьте следующий последовательный код: [149] [2] [9] [13]

**Stream** Opcode: 148 Байты данных: N + 1,  
где N номер запрошенных пакетов.

Эта команда запускает непрерывный поток пакетов данных. Список запрошенных пакетов отправляется каждые 15 ms, данный интервал используется конструктором для обновления данных.

Этот метод является оптимальным для опроса датчиков, если Вы контролируете конструктор через беспроводную сеть (которая обладает низкими скоростными характеристиками) с программным обеспечением, установленным на компьютере.

- Последовательный код: [148] [количество пакетов] [номер пакета 1] [номер пакета 2] [номер пакета 3] и т.д.
- Вводится из режимов: пассивный, безопасный или полный.
- Не переводится в другой режим
- Байт данных 1: количество запрошенных пакетов (0 - 43)
- Байты данных с 2 по N: номера пакетов (0 - 42)

Формат возвращаемых данных:

[19] [количество байтов] [пакет номер 1]  
[данные пакета номер 1] [пакет номер 2]  
[данные пакета номер 2]...[контрольная сумма]

Контрольная сумма состоит из одного байта. Это 8 битовое число суммы байтов от первого байта до байта контрольной суммы. Если Вы сложите все байты от первого до контрольной суммы и контрольную сумму, то младший байт результата будет равен 0.

**Пример:**

Для получения данных состояния левого переднего датчика антипрокидывания (пакет 29) и датчика виртуальной стены (пакет 13), отправьте следующую серию команд:

[148] [2] [29] [13]

**Имейте в виду:** пакет левого датчика антипрокидывания состоит из 2 байтов, а пакет инфракрасного датчика состоит из 1 байта.

Возвращаемые данные с конструктора будут иметь следующий вид:

19	5	29	2 25	13	0	182
Начало	Ко-во байт	Пакет №1	Данные пакета №1	Пакет №2	Данные пакета №2	Контрольн. сумма

**На заметку:** расчёт контрольной суммы: (5 + 29 + 2 + 25 + 13 + 0 + 182) = 256 и (256 & 0xFF) = 0.

В первой части потока данных, состояние левого датчика антипрокидывания конструктора было 549 (0x0225), а сигнал с датчика виртуальной стены отсутствовал.

Не следует запрашивать большее количество данных, чем может быть отправлено в текущем скоростном диапазоне в промежутке времени 15 ms. Например, при скорости 57600 максимум 86 байтов может быть отправлено за 15 ms:

$$15 \text{ ms} / 10 \text{ bits} (8 \text{ data} + \text{start} + \text{stop}) * 57600 = 86.4$$

Если, запрашивается больше данных, то произойдёт сбой операции. Это даст неверную контрольную сумму.

Первый байт и контрольная сумма могут быть использованы для коррекции программой, принимающей данные. Все блоки данных начинаются с 19 и заканчиваются однобайтовой контрольной суммой.

**Pause/Resume Stream** Opcode: 150 Data Bytes: 1

Эта команда служит для остановки и перезапуска потока данных без аннулирования номеров запрошенных пакетов.

- Последовательный код: [150] [состояние потока]
- Вводится из режимов: пассивный, безопасный или полный.
- Не переводится в другой режим
- Байт состояния потока данных: действие над потоком данных - пауза/возобновление (0 / 1)

0 останавливает поток, при этом список запрошенных пакетов остаётся прежним. 1 запускает поток, используя список пакетов, запрошенных в последний раз.

## Команды последовательности действий

Следующие команды позволят Вам определить последовательность выполняемых действий, состоящих из простых команд.

<b>Script</b>	Opcode: 152	Data Bytes: N + 1
---------------	-------------	-------------------

Где N является количеством байтов в последовательности действий.

Последовательность действий состоит из команд ОИ, которые могут иметь длину не более 100 байт. Контроль потоков данных отсутствует, но команды ожидания (смотрите ниже) сохраняют конструктор в его текущем состоянии, до тех пор, пока не наступит определённое событие.

- Последовательный код: [152] [длина последовательности] [Opcode 1] [Opcode 2] [Opcode 3] и т.д.
- Вводится из режимов: пассивный, безопасный или полный.
- Не переводит в другой режим
- 1 байт данных: длина последовательности (0 – 100)  
Определяется количеством команд в последовательности действий.  
Значение 0 обнуляет текущую последовательность.
- Байты данных 2 и последующие: команды и байты данных ОИ

**Это полезно знать:** для повторения последовательности используйте команду **Play Script** в качестве последней команды.

### Примеры последовательностей команд :

Продвижение на 40 см и остановка:  
152 13 137 1 44 128 0 156 1 144 137 0 0 0 0

Выключение светодиода Play при воздействии на бампер:  
152 17 158 5 158 251 139 2 0 0 158 5 158 251 139 0  
0 0 153

Продвижение по квадрату:  
152 17 137 1 44 128 0 156 1 144 137 1 44 0 1 157  
0 90 153

<b>Play Script</b>	Opcode: 153	Data Bytes: 0
--------------------	-------------	---------------

Эта команда загружает предыдущую заданную последовательность команд ОИ.

- Последовательный код: [153]
- Вводится в режимах: пассивный, безопасный или полный.
- Не переводит в другой режим.

<b>Show Script</b>	Opcode: 154	Data Bytes: 0
--------------------	-------------	---------------

Эта команда возвращает для просмотра предыдущий заданный код, начиная с байта количества и завершая байтами команд и данных. Если, были запущены команды **Stream** или **Pause/Resume Stream**, то поток данных с датчиков останавливается. Для перезапуска потока, отправьте команду **Pause/Resume Stream** (opcode 150).

- Последовательный код: [154]
- Вводится из режимов: пассивный, безопасный или полный
- Не переводит в другой режим

## Команды ожидания

Следующие команды переводят конструктор в режим ожидания до достижения определённого времени, или расстояния, или угла вращения, или события. В режиме ожидания, конструктор не меняет своё состояние и не реагирует на входящие сигналы. Команды ожидания используются только в команде **Script**.

<b>Wait Time</b>	Opcode: 155	Data Bytes: 1
------------------	-------------	---------------

Эта команда переводит конструктор в режим ожидания на определённое время. В этом состоянии конструктор не реагирует на входящие сигналы.

- Последовательный код: [155] [время]
- Вводится из режимов: пассивный, безопасный или полный.
- Не переводит в другой режим.
- Байт времени ожидания: время (0 - 255)  
Определяет время ожидания в десятых долях секунды с точностью 15 ms.

**Wait Distance** Opcode: 156 Data Bytes: 2

Эта команда переводит конструктор в режим ожидания до тех пор, пока он не пройдёт определённую дистанцию в мм. При движении конструктора вперёд, его дистанция увеличивается. При движении конструктора назад, дистанция уменьшается. При пассивном вращении колёс в любом из направлений, дистанция увеличивается. До тех пор, пока конструктор не пройдет заданную дистанцию, его состояние не изменится, он не будет реагировать на какие-либо внешние сигналы.

**Имейте в виду:** эта команда аннулирует показания дистанции, которые приходят и сохраняются в пакетах сенсоров 19, 2 и 6.

- Последовательный код: [156] [старший байт дистанции] [младший байт дистанции]
- Вводится из режимов: пассивный, безопасный или полный
- Не переводит в другой режим
- 1-2 байты данных дистанции: 16-битовая дистанция в мм, старший байт первый (-32767 -32768)

**Wait Angle** Opcode: 157 Data Bytes: 2

Эта команда переводит конструктор в режим ожидания до тех пор, пока он не произведет поворот на определённый угол в градусах. Когда, конструктор вращается против часовой стрелки, угол увеличивается. Когда, конструктор вращается по часовой стрелке, угол уменьшается. До тех пор, пока конструктор не произведёт поворот на определённый угол, его состояние не изменится, он также не будет реагировать на внешние сигналы.

**Имейте в виду:** эта команда аннулирует показания угла, которые приходят и сохраняются в пакетах сенсоров 20, 2 и 6.

- Последовательный код: [157] [старший байт угла] [младший байт угла]
- Вводится из режимов: пассивный, безопасный или полный
- Не переводит в другой режим
- 1-2 байты данных угла: 16- битовый угол в градусах, старший байт первый (-32767 -32768)

**Wait Event** Opcode: 158 Data Bytes: 1

Эта команда переводит конструктор в режим ожидания до тех пор, пока он не распознает определённое событие. Пока это событие не свершится, он не изменяет своего состояния и не реагирует на внешние сигналы.

- Последовательный код: [158] [номер события]
- Вводится из режимов: пассивный, безопасный или полный.
- Не переводит в другой режим.
- Байт данных события : номер события (1 to 20 and -1 to -20)

Для ожидания обратного (противоположного) события, отправьте дополнительное значение этого события, используя таблицу (смотрите ниже). Например, для события «столкновение бампером с препятствием», обратным будет следующий последовательный код [158] [251], который соответствует [158] [-5].

Ожидаемое событие

Событие	Номер	Дополнительное значение
Wheel Drop	1	255
Front Wheel Drop	2	254
Left Wheel Drop	3	253
Right Wheel Drop	4	252
Bump	5	251
Left Bump	6	250
Right Bump	7	249
Virtual Wall	8	248
Wall	9	247
Cliff	10	246
Left Cliff	11	245
Front Left Cliff	12	244
Front Right Cliff	13	243
Right Cliff	14	242
Home Base	15	241
Advance Button	16	240
Play Button	17	239
Digital Input 0	18	238
Digital Input 1	19	237
Digital Input 2	20	236
Digital Input 3	21	235
OI Mode = Passive	22	234

## Открытый интерфейс конструктора пакеты сенсоров

Отвечая на команду запроса состояния датчиков **Sensors**, на команду запроса списка состояния датчиков **Query list**, на команду запроса потока данных с датчиков, конструктор возвращает один из 43 различных *пакетов данных* датчиков в зависимости от запрашиваемого *пакета данных*. Некоторые пакеты включают в себя группы других пакетов. Некоторые данные с датчиков могут быть 16 битовыми.

Большинство из пакетов (номера 7 – 42) содержат данные об одном сенсоре, которые могут быть 1 или 2 байтовыми. Двухбайтовые пакеты состоят из 16-бит, старший байт передается первым.

Некоторые из пакетов (0-6) содержат группы из одиночных пакетов.

### Размер и содержание пакетов

ID группы пакета	Размер пакета	Содержит пакеты
0	26 bytes	7 - 26
1	10 bytes	7 - 16
2	6 bytes	17 - 20
3	10 bytes	21 - 26
4	14 bytes	27 - 34
5	12 bytes	35 - 42
6	52 bytes	7 - 42

### Bumps and Wheel Drops Packet ID: 7 Data Bytes: 1

Состояние бампера (0 = бампер не нажат, 1 = бампер нажат) и датчиков полного выпуска колёс (0 = колесо поднято, 1 = колесо выпущено) передаются как отдельные биты.

Диапазон: 0 - 31

Бит	7	6	5	4	3	2	1	0
Сенсор	н/д	н/д	н/д	Выпуск Переднего	Выпуск левого	Выпуск правого	Сталкивание слева	справ

### Wall Packet ID: 8 Data Bytes: 1

Состояние датчика стены передаётся как 1 бит (0 = стена не видна, 1 = стена видна).

### Cliff Left Packet ID: 9 Data Bytes: 1

Состояние датчика антипрокидывания на левой стороне конструктора передаётся, как 1 бит (0 = видит пол, 1 = не видит пол).

### Cliff Front Left Packet ID: 10 Data Bytes: 1

Состояние датчика антипрокидывания на передней стороне конструктора слева передаётся как 1 бит (0 = видит пол, 1 = не видит пол).

### Cliff Front Right Packet ID: 11 Data Bytes: 1

Состояние датчика антипрокидывания на передней стороне конструктора справа передаётся как 1 бит (0 = видит пол, 1 = не видит пол).

### Cliff Right Packet ID: 1\_ Data Bytes: 1

Состояние датчика антипрокидывания на правой стороне конструктора передаётся как 1 бит (0 = видит пол, 1 = не видит пол).

### Virtual Wall Packet ID: 13 Data Bytes: 1

Состояние датчика виртуальной стены передаётся как 1 бит (0 = виртуальная стена не определена, 1 = виртуальная стена определена). Имейте в виду, что силовое поле в верхней части базовой станции, также влияет на состояние этого датчика.

### Low Side Driver and Wheel Overcurrents Packet ID: 14 Data Bytes: 1

Состояние датчиков тока трёх дополнительных и двух ведущих двигателей передаются отдельными битами (0 = уровень тока нормальный, 1 = уровень тока превышен).

Двигатель	Максимальный ток
LD0	0.5A
LD1	0.5A
LD2	1.5A
Ведущие Колёса	1.0A

Бит	7	6	5	4	3	2	1	0
Motor	н/д	н/д	н/д	Левый	Правый	LD-2	LD-0	LD-1

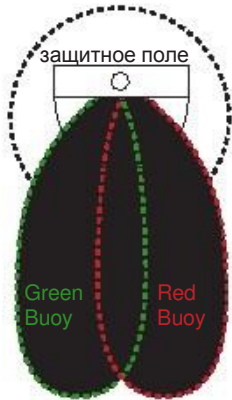
**Unused Bytes** Packet IDs: 15 - 16 Data Bytes: 1

Неиспользованные байты: два неиспользованных байта отправляются после байта превышения уровня тока при запрошенных пакетах 0, 1, или 6. Значения двух неиспользованных байтов всегда равно 0.

**Infrared Byte** Packet ID: 17 Data Bytes: 1

Значение этого байта соответствует информации, полученной инфракрасным приёмником. Значение 255 указывает на то, что ИК приёмник не принимает байт. Этот байт отражает информацию, принятую от дистанционного пульта, или базовой станции, или конструктора (при использовании команды Send IR) или устройств, созданных пользователем.  
 Диапазон: 0 – 255

Конфигурация лучей базовой станции



Тип устройства	Значение	Наименование
Пульт ДУ	129	Left
	130	Forward
	131	Right
	132	Spot
	133	Max
	134	Small
	135	Medium
	136	Large / Clean
	137	PAUSE
	138	Power
	139	arc-forward-left
	140	arc-forward-right
	141	drive-stop

Тип устройства	Значение	Наименование
Программируемый Пульт ДУ	142	Send All
	143	Seek Dock
Базовая станция	240	Reserved
	248	Red Buoy
	244	Green Buoy
	242	Force Field
	252	Red Buoy and Green Buoy
	250	Red Buoy and Force Field
	246	Green Buoy and Force Field
	254	Red Buoy, Green Buoy and Force Field

**Buttons** Packet ID: 18 Data Bytes: 1

Состояние клавиш **Play** и **Advance** передается отдельными битами (0 = клавиша не нажата, 1 = клавиша нажата).

Диапазон: 0 - 5

Бит	7	6	5	4	3	2	1	0
Клавиша	н/д	н/д	н/д	н/д	н/д	Advance	н/д	Play

**Distance** Packet ID: 19 Data Bytes: 2

Это расстояние, которое конструктор прошёл в миллиметрах, после последнего запроса, выраженное 16-битами (старший байт первый). Равняется сумме дистанций, пройденных обоими колёсами делённых на два. Положительные показания указывают на движение вперёд; отрицательные показания указывают на движение в обратном направлении. Если, опрос происходит редко, то значения фиксируются на их минимуме или максимуме.

Диапазон: -32768 – 32767

**Angle** Packet ID: \_0 Data Bytes: 2

Это угол в градусах, на который повернулся робот после последнего запроса, выраженный 16-битами (старший байт первый). Углы против часовой стрелки являются положительными, по часовой стрелке отрицательными. Если, опрос происходит редко, то значения фиксируются на их минимуме или максимуме.

Диапазон: -32768 – 32767

**Имейте в виду:** конструктор использует энкодеры колёс для измерения расстояния и угла. Если колёса проскальзывают, то реальные расстояние и угол могут отличаться от показаний конструктора.

**Charging State** Packet ID: 21 Data Bytes: 1

Этот код показывает состояние зарядки.

Диапазон: 0 - 5

Код	Состояние зарядки
0	Не заряжается
1	Возобновление зарядки
2	Полная зарядка
3	Зарядка малым током
4	Ожидание
5	Ошибка зарядки

**Voltage** Packet ID: 22 Data Bytes: 2

Этот код показывает напряжение аккумулятора конструктора в милливольтках (mV).

Диапазон: 0 – 65535

**Current** Packet ID: 23 Data Bytes: 2

Входящий и выходящий ток аккумулятора конструктора в миллиамперах (mA). Отрицательные токи указывают, что ток выходит из аккумулятора, например, во время работы. Положительные токи указывают, что ток течёт в аккумулятор, например во время зарядки.

Диапазон: -32768 – 32767

**Battery Temperature** Packet ID: 24 Data Bytes: 1

Температура аккумулятора в градусах Цельсия.

Диапазон: -128 – 127

**Battery Charge** Packet ID: 25 Data Bytes: 2

Заряд аккумулятора конструктора в миллиамперах в час (mAh). Показатель заряда уменьшается во время разрядки аккумулятора и увеличивается, когда аккумулятор заряжается. Имейте в виду, что этот показатель не будет точным, если Вы используете щелочные (алкалиновые) батареи.

Диапазон: 0 – 65535

**Battery Capacity** Packet ID: 26 Data Bytes: 2

Предполагаемая емкость аккумулятора конструктора в миллиамперах в час (mAh). Имейте в виду, что этот показатель не будет точным, если Вы используете щелочные (алкалиновые) батареи.

Диапазон: 0 – 65535

**Wall Signal** Packet ID: 27 Data Bytes: 2

Уровень сигнала датчика стены, выраженный 16 битовым числом, старший байт первый.

Диапазон: 0-4095

**Cliff Left Signal** Packet ID: 28 Data Bytes: 2

Уровень сигнала левого датчика антипрокидывания, выраженный 16 битовым числом, старший байт первый.

Диапазон: 0-4095

**Cliff Front Left Signal** Packet ID: 29 Data Bytes: 2

Уровень сигнала левого переднего датчика антипрокидывания, выраженный 16 битовым числом, старший байт первый.

Диапазон: 0-4095

**Cliff Front Right Signal** Packet ID: 30 Data Bytes: 2

Уровень сигнала переднего правого датчика антипрокидывания, выраженный 16 битовым числом, старший байт первый.

Диапазон: 0-4095

**Cliff Right Signal** Packet ID: 31 Data Bytes: 2

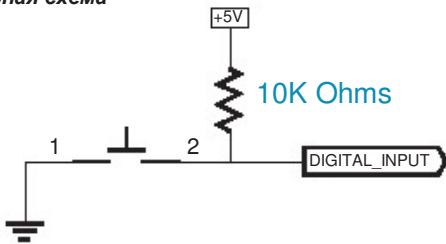
Уровень сигнала правого датчика антипрокидывания, выраженный 16 битовым числом, старший байт первый.

Диапазон: 0-4095

**Cargo Bay Digital Inputs** Packet ID: 32 Data Bytes: 1

Состояние цифровых входов на 25-и контактном разъёме грузового отсека передаётся отдельными битами. (0 = низкий, 1 = высокий (5V)). Имейте в виду, если, на контакте *изменения скорости передачи данных* высокий уровень, то скорость установлена по умолчанию 57600.

**Примерная схема**



Диапазон: 0 - 31

Бит	7	6	5	4	3	2	1	0
	н/д	н/д	н/д	Device Detect /Baud Rate Change (pin 15)	Digital Input 3 (pin 6)	Digital Input 2 (pin 18)	Digital Input 1 (pin 5)	Digital Input 0 (pin 17)

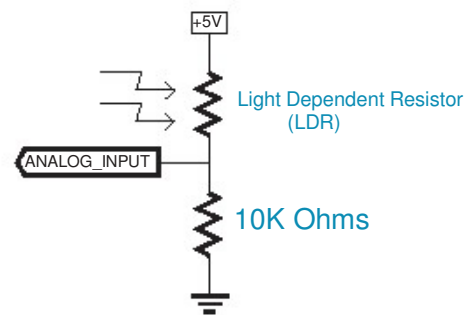
Контакт *определения устройства* может быть использован для изменения скорости передачи. Низкий уровень на этом контакте обеспечивает скорость 19200, высокий уровень – 57600.

**Cargo Bay Analog Signal** Packet ID: 33 Data Bytes: 2

Состояние аналогового входа 25-и контактного разъёма грузового отсека отображается при помощи 10 бит, старший байт первый. 0 = 0 вольт; 1023 = 5 вольт. Аналоговый вход на контакте №4.

Range: 0 - 1023

**Примерная схема**



**Charging Sources Available** Packet ID: 34 Data Bytes: 1

Соединение конструктора с базовой станцией, зарядным устройством отображается отдельными битами (смотрите ниже).

Диапазон: 0 - 3

1 = источник зарядки присутствует, подключено питание; 0 = источник зарядки отсутствует, питание отключено.

Bit	7	6	5	4	3	2	1	0
Charging Source	n/a	n/a	n/a	n/a	n/a	n/a	Home Base	Internal Charger

**OI Mode** Packet ID: 35 Data Bytes: 1

Состояние открытого интерфейса выводится отдельными битами (смотрите ниже).

Диапазон: 0 - 3

Number	Mode
0	Off
1	Passive
2	Safe
3	Full

**Song Number** Packet ID: 36 Data Bytes: 1

Выводится номер, последней выбранной песни.

Диапазон: 0-15

**Song Playing** Packet ID: 37 Data Bytes: 1

Выводится состояние проигрывания песни. 1 = песня проигрывается; 0 = песня не проигрывается.

Диапазон: 0-1

**Number of Stream Packets** Packet ID: 38 Data Bytes: 1

Выводится номер пакета потока данных.

Диапазон: 0-43

**Requested Velocity** Packet ID: 39 Data Bytes: 2

Выводятся последние показания скорости при выполнении последней команды Drive, выраженные 16 битовым числом, старший байт первый.

Диапазон: -500 - 500 mm/s

**Requested Radius** Packet ID: 40 Data Bytes: 2

Выводится последний радиус при выполнении последней команды Drive, выраженный 16 битовым числом, старший байт первый.

Диапазон: -32768 - 32767 mm

**Requested Right Velocity** Packet ID: 41 Data Bytes: 2

Выводится последняя скорость правого колеса при выполнении последней команды Drive Direct, выраженная 16 битовым числом, старший байт первый.

Диапазон: -500 - 500 mm/s

**Requested Left Velocity** Packet ID: 4\_ Data Bytes: 2

Выводится последняя скорость левого колеса при выполнении последней команды Drive Direct, выраженная 16 битовым числом, старший байт первый.

Диапазон: -500 - 500 mm/s

## Открытый интерфейс конструктора СПИСОК КОМАНД

Command	Opcode	Data Bytes: 1	Data Bytes: 2	Data Bytes: 3	Data Bytes: 4	Etc.
Start	128					
Baud	129	Baud Code: (0 – 11)				
Control	130					
Safe	131					
Full	132					
Spot	134					
Cover	135					
Demo	136	Demos (-1 - 9)				
Drive	137	Velocity (-500 – 500 mm/s)		Radius (-2000 – 2000 mm)		
Low Side Drivers	138	Output Bits (0 – 7)				
LEDs	139	LED Bits (0 – 10)	Power LED Color (0 – 255)	Power LED Intensity (0 – 255)		
Song	140	Song Number (0 - 15)	Song Length (1 - 16)	Note Number 1 (31 – 27)	Note Duration 1 (0 - 255)	Note Number 2, etc.
Play	141	Song Number: (0 – 15)				
Sensors	142	Packet ID: (0 – 42)				
Cover and Dock	143					
PWM Low Side Drivers	144	Low Side Driver 2 Duty Cycle (0 - 128)	Low Side Driver 1 Duty Cycle (0 - 128)	Low Side Driver 0 Duty Cycle (0 - 128)		
Drive Direct	145	Right wheel velocity (-500 – 500 mm/s)		Left wheel velocity (-500 – 500 mm/s)		
Digital Outputs	147	Output Bits (0–7)				
Stream	148	Number of Packets	Packet ID 1 (0 – 42)	Packet ID 2, etc.		
Query List	149	Packet ID 1 (0 – 42)	Packet ID 2, etc.			
Pause/Resume Stream	150	Range: 0-1				
Send IR	151	Byte (0 - 255)				
Script	152	Script Length: (1 – 100)	Command Opcode 1	Command Data Byte 1, etc.	Command Opcode 2	Etc.
Play Script	153					
Show Script	154					
Wait Time	155	Time (0 – 255 seconds/10)				
Wait Distance	156	Distance (-32767 - 32768 mm)				
Wait Angle	157	Angle (-32767 - 32768 degrees)				
Wait Event	158	Event ID (1 to 20 and -1 to -20)				

**Код скорости обмена данными (0 – 11)**

Код	Скорость (бод в секунду)
0	300
1	600
2	1200
3	2400
4	4800
5	9600
6	14400
7	19200
8	28800
9	38400
10	57600
11	115200

**1 байт данных светодиодов : биты (0 – 10)**

**Advance** и **Play** используют зелёные светодиоды: 0 = выкл, 1 = вкл

Бит	7	6	5	4	3	2	1	0
LED	n/a	n/a	n/a	n/a	Advance	n/a	Play	n/a

**2 и 3 байты данных светодиодов: яркость, цвет (0 - 255)**

Для индикации питания используется двухцветный (красный/зелёный) светодиод. Яркость и цвет этого светодиода можно менять с 8-и битной точностью.

2-ой байт данных светодиодов: цвет индикации (0 – 255)

0 = зелёный, 255 = красный. Средние показатели дадут промежуточные цвета (оранжевый, жёлтый и т.д.).

3-ий байт данных светодиодов: яркость индикации (0 – 255)

0 = выкл., 255 = полная интенсивность. Средние показатели обеспечат среднюю яркость.

**Байт данных цифровых выходов : биты (0 – 7)**

0 = низкий уровень (0В); 1 = высокий уровень (5В).

Цифровые выходы рассчитаны на ток до 20 mA.

Bit	7	6	5	4	3	2	1	0
Output						digital output 2 (pin 20)	digital output 1 (pin 7)	digital output 0 (pin 19)

**Байт данных дополнительных двигателей: биты (0 – 7)**

0 = выкл., 1 = вкл. (100% мощность в рабочем режиме)

Bit	7	6	5	4	3	2	1	0
Output						Low Side Driver 2	Low Side Driver 1	Low Side Driver 0

## Открытый интерфейс конструктора список пакетов датчиков

Отвечая на команду запроса состояния датчиков **Sensors**, конструктор выводит один из 43 различных пакетов данных датчиков в зависимости от запрашиваемого пакета данных. Некоторые пакеты содержат группы других пакетов, которые выводятся последовательно. Некоторые из показателей данных сенсоров имеют 16 битовые значения. Эти значения выводятся как два байта, старший байт первый.

### Содержание и размеры групп пакетов

Group Packet ID	Packet Size	Contains Packets
0	26 bytes	7 - 26
1	10 bytes	7 - 16
2	6 bytes	17 - 20
3	10 bytes	21 - 26
4	14 bytes	27 - 34
5	12 bytes	35 - 42
6	52 bytes	7 - 42

Packet Membership	Name	Bytes	Value Range	Units		
4	27	Wall Signal	2	0 - 4095		
	28	Cliff Left Signal	2	0 - 4095		
	29	Cliff Front Left Signal	2	0 - 4095		
	30	Cliff Front Right Signal 2		0 - 4095		
	31	Cliff Right Signal	2	0 - 4095		
	32	User Digital Inputs	1	0 - 31		
	33	User Analog Input	2	0 - 1023		
	34	Charging Sources Available	1	0-3		
	5	35	OI Mode	1	0-3	
		36	Song Number	1	0 - 15	
		37	Song Playing	1	0-1	
		38	Number of Stream Packets	1	0 - 42	
		39	Velocity	2	-500 - 500	mm/s
		40	Radius	2	-32768 - 32767	mm
41		Right Velocity	2	-500 - 500	mm/s	
42		Left Velocity	2	-500 - 500	mm/s	

### Содержание и размеры групп пакетов

Packet Membership	Name	Bytes	Value Range	Units			
01	67	Bumps and Wheel Drops	1	0 - 31			
		8	Wall	1	0-1		
		9	Cliff Left	1	0-1		
		10	Cliff Front Left	1	0-1		
		11	Cliff Front Right	1	0-1		
		12	Cliff Right	1	0-1		
		13	Virtual Wall	1	0-1		
		14	Overcurrents	1	0 - 31		
		15	Unused	1	0		
		16	Unused	1	0		
		2	17	IR Byte	1	0 - 255	
			18	Buttons	1	0 - 15	
			19	Distance	2	-32768 - 32767	mm
			20	Angle	2	-32768 - 32767	mm
		3	21	Charging State	1	0-5	
			22	Voltage	2	0 - 65535	mV
	23		Current	2	-32768 - 32767	mA	
	24		Battery Temperature	1	-128 - 127	degrees Celsius	
	25		Battery Charge	2	0 - 65535	mAh	
	26		Battery Capacity	2	0 - 65535	mAh	

### Коды состояния зарядки

Code	Charging State
0	Not charging
1	Reconditioning Charging
2	Full Charging
3	Trickle Charging
4	Waiting
5	Charging Fault Condition

### Свешивание колёс и воздействие на бампер

Bit	7	6	5	4	3	2	1	0
Sensor	n/a	n/a	n/a	Wheel-drop Caster	Wheel-drop Left	Wheel-drop Right	Bump Left	Bump Right

### Превышение силы тока ведущими и дополнительными двигателями

Bit	7	6	5	4	3	2	1	0
Motor	n/a	n/a	n/a	Left Wheel	Right Wheel	Low Side Driver 2	Low Side Driver 0	Low Side Driver 1

**Кнопки**

Bit	7	6	5	4	3	2	1	0
Button	n/a	n/a	n/a	n/a	n/a	Advance	n/a	Play

**Цифровые входы грузового отсека**

Bit	7	6	5	4	3	2	1	0
Button	n/a	n/a	n/a	Baud Rate Change (pin 15)	Digital Input 3 (pin 6)	Digital Input 2 (pin 18)	Digital Input 1 (pin 5)	Digital Input 0 (pin 17)

**Доступные источники зарядки**

Bit	7	6	5	4	3	2	1	0
Charging Source	n/a	n/a	n/a	n/a	n/a	n/a	Home Base	Internal Charger

**Режимы ОИ**

Number	Mode
0	Off
1	Passive
2	Safe
3	Full

