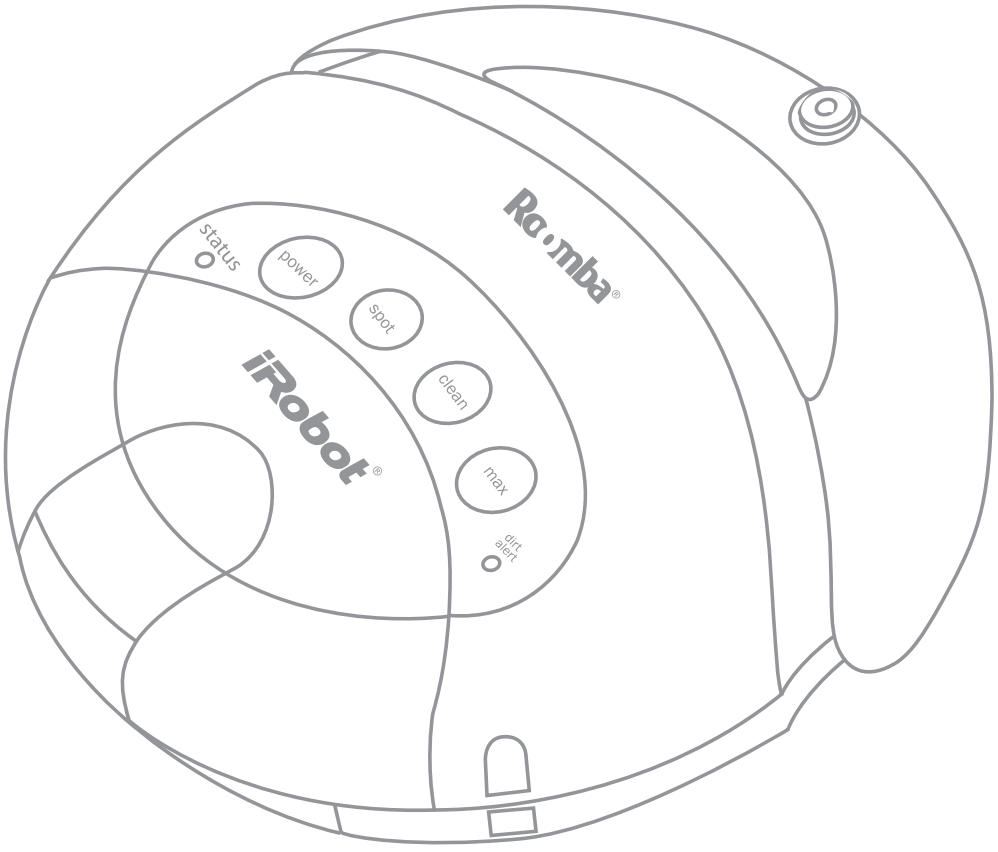
iRobot® Roomba® Serial Command Interface (SCI) Specification



iRobot®

www.irobot.com

SCI Overview

Versions of iRobot® Roomba® Vacuuming Robot manufactured after October, 2005 contain an electronic and software interface that allows you to control or modify Roomba's behavior and remotely monitor its sensors. This interface is called the iRobot Roomba Serial Command Interface or Roomba SCI.

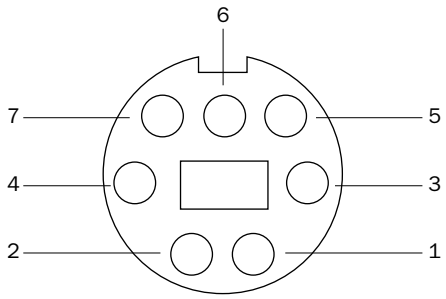
Roomba SCI is a serial protocol that allows users to control a Roomba through its external serial port (Mini-DIN connector). The SCI includes commands to control all of Roomba's actuators (motors, LEDs, and speaker) and also to request sensor data from all of Roomba's sensors. Using the SCI, users can add functionality to the normal Roomba behavior or they can create completely new operating instructions for Roomba.

Physical Connections

To use the SCI, a processor capable of generating serial commands such as a PC or a microcontroller must be connected to the external Mini-DIN connector on Roomba. The Mini-DIN connector provides two way serial communication at TTL Levels as well as a Device Detect input line that can be used to wake Roomba from sleep. The connector also provides an unregulated direct connection to Roomba's battery which users can use to power their SCI applications. The connector is located in the rear right side of Roomba beneath a snap-away plastic guard.

ROOMBA'S EXTERNAL SERIAL PORT MINI-DIN CONNECTOR PINOUT

This diagram shows the pin-out of the top view of the female connector in Roomba. Note that pins 5, 6, and 7 are towards the outside circumference of Roomba.



Pin	Name	Description
1	Vpwr	Roomba battery + (unregulated)
2	Vpwr	Roomba battery + (unregulated)
3	RXD	0 – 5V Serial input to Roomba
4	TXD	0 – 5V Serial output from Roomba
5	DD	Device Detect input (active low) – used to wake up Roomba from sleep
6	GND	Roomba battery ground
7	GND	Roomba battery ground

The RXD, TXD, and Device Detect pins use 0 – 5V logic, so a level shifter such as a MAX232 chip will be needed to communicate with a Roomba from a PC, which uses rs232 levels.

Serial Port Settings

Baud: 57600 or 19200 (see below)
Data bits: 8
Parity: None
Stop bits: 1
Flow control: None

By default, Roomba communicates at 57600 baud. If you are using a microcontroller that does not support 57600 baud, there are two ways to force Roomba to switch to 19200:

METHOD 1:

When manually powering on Roomba, hold down the power button. After 5 seconds, Roomba will start beeping. After 10 seconds, Roomba will play a tune of descending pitches. Roomba will now communicate at 19200 baud until the battery is removed and reinserted (or the battery voltage falls below the minimum required for processor operation) or the baud rate is explicitly changed via the SCI.

METHOD 2:

You can use the Device Detect to change Roomba's baud rate. After you have awakened Roomba (using Device Detect or by some other method) wait 2 seconds and then pulse the Device Detect low three times. Each pulse should last between 50 and 500 milliseconds. Roomba will now communicate at 19200 baud until the battery is removed and reinserted (or the battery voltage falls below the minimum required for processor operation) or the baud rate is explicitly changed via the SCI.

Here is a Python code fragment that illustrates this method (Device Detect is connected to the PC's RTS line via a level shifter):

```
ser = serial.Serial(0, baudrate=19200,
timeout=0.1)

ser.open()

# wake up robot
ser.setRTS (0)
time.sleep (0.1)
ser.setRTS (1)
time.sleep (2)

# pulse device-detect three times
for i in range (3):
       ser.setRTS (0)
       time.sleep (0.25)
       ser.setRTS (1)
       time.sleep (0.25)
```

SCI Modes

The Roomba SCI has four operating modes: off, passive, safe, and full. On a battery change or other loss of power, the SCI will be turned off. When it is off, the SCI will listen at the default baud bps for an SCI Start command. Once it receives the Start command, the SCI will be enabled in passive mode. In passive mode, users can do the following:

- Request and receive sensor data using the Sensors command
- Execute virtual button pushes to start and stop cleaning cycles (Power, Spot, Clean, and Max commands)
- Define a song (but not play one)
- Set force-seeking-dock mode

Users cannot control any of Roomba's actuators when in passive mode, but Roomba will continue to behave normally, including performing cleaning cycles, charging, etc. When in passive mode, users can then send the Control command to put the robot into safe mode.

In safe mode, the users have full control of the robot, except for the following safety-related conditions:

- Detection of a cliff while moving forward (or moving backward with a small turning radius)
- Detection of wheel drop (on any wheel)
- Charger plugged in and powered

When one of the conditions listed above occurs, the robot stops all motors and reverts to passive mode.

For complete control of the robot, users must send the Full command while in safe mode to put the SCI into full mode. Full mode shuts off the cliff and wheel-drop safety features. (The robot will still not run with a powered charger plugged in.) This mode gives users unrestricted control of the robot's actuators. To put the SCI back into safe mode, users can send the Safe command.

If no commands are sent to the SCI when it is in safe or full mode, Roomba will wait with all motors off and will not respond to button presses or other sensor input.

To go back to passive mode from safe or full mode, users can send any one of the four virtual button commands (Power, Spot, Clean, or Max). These button commands are equivalent to the corresponding button press in normal Roomba behavior. For instance, the Spot command will start a spot cleaning cycle.

Allow 20 milliseconds between sending commands that change the SCI mode.

Roomba SCI Commands

Listed below are the commands that users send to the SCI over to the serial port in order to control Roomba. Each command is specified by a one-byte opcode. Some commands must also be followed by data bytes. The meaning of the data bytes for each command are specified with the commands below. The serial byte sequence for each command is also shown with each separate byte enclosed in brackets. Roomba will not respond to any SCI commands when it is asleep. Users can wake up Roomba by setting the state of the Device Detect pin low for 500ms. The Device Detect line is on Roomba external Mini-DIN connector.

Start Command opcode: 128 Number of data bytes: 0

Starts the SCI. The Start command must be sent before any other SCI commands. This command puts the SCI in passive mode.

Serial sequence: [128]

Baud Command opcode: 129 Number of data bytes: 1

Sets the baud rate in bits per second (bps) at which SCI commands and data are sent according to the baud code sent in the data byte. The default baud rate at power up is 57600 bps. (See Serial Port Settings, above.) Once the baud rate is changed, it will persist until Roomba is power cycled by removing the battery (or until the battery voltage falls below the minimum required for processor operation). You must wait 100ms after sending this command before sending additional commands at the new baud rate. The SCI must be in passive, safe, or full mode to accept this command. This command puts the SCI in passive mode.

Serial sequence: [129] [Baud Code]

Baud data byte 1: Baud Code (0 – 11)

Baud code	Baud rate in bps
0	300
1	600
2	1200
3	2400
4	4800
5	9600
6	14400
7	19200
8	28800
9	38400
10	57600
11	115200

Control Command opcode: 130 Number of data bytes: 0

Enables user control of Roomba. This command must be sent after the start command and before any control commands are sent to the SCI. The SCI must be in passive mode to accept this command. This command puts the SCI in safe mode.

Serial sequence: [130]

Safe Command opcode: 131 Number of data bytes: 0

This command puts the SCI in safe mode. The SCI must be in full mode to accept this command.

Note: In order to go from passive mode to safe mode, use the Control command.

Serial sequence: [131]

Full Command opcode: 132 Number of data bytes: 0

Enables unrestricted control of Roomba through the SCI and turns off the safety features. The SCI must be in safe mode to accept this command. This command puts the SCI in full mode.

Serial sequence: [132]

Power Command opcode: 133 Number of data bytes: 0

Puts Roomba to sleep, the same as a normal "power" button press. The Device Detect line must be held low for 500 ms to wake up Roomba from sleep. The SCI must be in safe or full mode to accept this command. This command puts the SCI in passive mode.

Serial sequence: [133]

Spot Command opcode: 134 Number of data bytes: 0

Starts a spot cleaning cycle, the same as a normal "spot" button press. The SCI must be in safe or full mode to accept this command. This command puts the SCI in passive mode.

Serial sequence: [134]

Clean Command opcode: 135 Number of data bytes: 0

Starts a normal cleaning cycle, the same as a normal "clean" button press. The SCI must be in safe or full mode to accept this command. This command puts the SCI in passive mode.

Serial sequence: [135]

Max Command opcode: 136 Number of data bytes: 0

Starts a maximum time cleaning cycle, the same as a normal "max" button press. The SCI must be in safe or full mode to accept this command. This command puts the SCI in passive mode.

Serial sequence: [136]

Drive Command opcode: 137 Number of data bytes: 4

Controls Roomba's drive wheels. The command takes four data bytes, which are interpreted as two 16 bit signed values using twos-complement. The first two bytes specify the average velocity of the drive wheels in millimeters per second (mm/s), with the high byte sent first. The next two bytes specify the radius, in millimeters, at which Roomba should turn. The longer radii make Roomba drive straighter; shorter radii make it turn more. A Drive command with a positive velocity and a positive radius will make Roomba drive forward while turning toward the left. A negative radius will make it turn toward the right. Special cases for the radius make Roomba turn in place or drive straight, as specified below. The SCI must be in safe or full mode to accept this command. This command does change the mode.

Note: The robot system and its environment impose restrictions that may prevent the robot from accurately carrying out some drive commands. For example, it may not be possible to drive at full speed in an arc with a large radius of curvature.

Serial sequence: [137] [Velocity high byte] [Velocity low byte] [Radius high byte] [Radius low byte]

Drive data bytes 1 and 2: Velocity (-500 – 500 mm/s)

Drive data bytes 3 and 4: Radius (-2000 – 2000 mm)
Special cases: Straight = 32768 = hex 8000
Turn in place clockwise = -1
Turn in place counter-clockwise = 1

Example:

To drive in reverse at a velocity of -200 mm/s while turning at a radius of 500mm, you would send the serial byte sequence [137] [255] [56] [1] [244].

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

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

Motors Command opcode: 138 Number of data bytes: 1

Controls Roomba's cleaning motors. The state of each motor is specified by one bit in the data byte. The SCI must be in safe or full mode to accept this command. This command does not change the mode.

Serial sequence: [138] [Motor Bits]

Motors data byte 1: Motor Bits (0 – 7)

0 = off, 1 = on

Bit	7	6	5	4	3	2	1	0
Motor	n/a	n/a	n/a	n/a	n/a	Main Brush	Vacuum	Side Brush

Example:

To turn on only the vacuum motor, send the serial byte sequence [138] [2].

Leds Command opcode: 139 Number of data bytes: 3

Controls Roomba's LEDs. The state of each of the spot, clean, max, and dirt detect LEDs is specified by one bit in the first data byte. The color of the status LED is specified by two bits in the first data byte. The power LED is specified by two data bytes, one for the color and one for the intensity. The SCI must be in safe or full mode to accept this command. This command does not change the mode.

Serial sequence: [139] [Led Bits] [Power Color] [Power Intensity]

Leds data byte 1: Led Bits (0 – 63)

Dirt Detect uses a blue LED: 0 = off, 1 = on

Spot, Clean, and Max use green LEDs: 0 = off, 1 = on

Status uses a bicolor (red/green) LED: 00 = off, 01 = red, 10 = green, 11 = amber

Bit	7	6	5	4	3	2	1	0
LED	n/a	n/a	Status (2 bits)		Spot	Clean	Max	Dirt Detect

Power uses a bicolor (red/green) LED whose intensity and color can be controlled with 8-bit resolution.

Leds data byte 2: Power Color (0 – 255)

0 = green, 255 = red. Intermediate values are intermediate colors.

Leds data byte 3: Power Intensity (0 – 255)

0 = off, 255 = full intensity. Intermediate values are intermediate intensities.

Example:

To turn on the dirt detect and spot LEDs, make the status LED red, and to light the power LED green at half intensity, send the serial byte sequence [139] [25] [0] [128]

Song Command opcode: 140 Number of data bytes: 2N + 2, where N is the number of notes in the song

Specifies a song to the SCI to be played later. Each song is associated with a song number which the Play command uses to select the song to play. Users can specify up to 16 songs with up to 16 notes per song. Each note is specified by a note number using MIDI note definitions and a duration specified in fractions of a second. The number of data bytes varies depending on the length of the song specified. A one note song is specified by four data bytes. For each additional note, two data bytes must be added. The SCI must be in passive, safe, or full mode to accept this command. This command does not change the mode.

Serial sequence: [140] [Song Number] [Song Length] [Note Number 1] [Note Duration 1] [Note Number 2] [Note Duration 2] etc.

Song data byte 1: Song Number (0 – 15)
Specifies the number of the song being specified. If you send a second Song command with the same song number, the old song will be overwritten.

Song data byte 2: Song Length (1 – 16)
Specifies the length of the song in terms of the number of notes.

Song data bytes 3, 5, 7, etc.: Note Number (31 – 127)
Specifies the pitch of the note to be played in terms of the MIDI note numbering scheme. The lowest note that Roomba can play is note number 31. See the note number table for specific notes. Any note number outside of the range of 31 to 127 will be interpreted as a rest note and no sound will be played during this note duration.

Song data bytes 4, 6, 8, etc.: Note Duration (0 – 255)
Specifies the duration of the note in increments of 1/64 of a second. Therefore, half-second long note will have a duration value of 32.

Note Number Table for Song Command (with Frequency in Hz)

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

Play Command opcode: 141 Number of data bytes: 1

Plays one of 16 songs, as specified by an earlier Song command. If the requested song has not been specified yet, the Play command does nothing. The SCI must be in safe or full mode to accept this command. This command does not change the mode.

Serial sequence: [141] [Song Number]

Play data byte 1: Song Number (0 – 15)
Specifies the number of the song to be played. This must match the song number of a song previously specified by a Song command.

Sensors Command opcode: 142 Number of data bytes: 1

Requests the SCI to send a packet of sensor data bytes. The user can select one of four different sensor packets. The sensor data packets are explained in more detail in the next section. The SCI must be in passive, safe, or full mode to accept this command. This command does not change the mode.

Serial sequence: [142] [Packet Code]

Sensors data byte 1: Packet Code (0 – 3)
Specifies which of the four sensor data packets should be sent back by the SCI. A value of 0 specifies a packet with all of the sensor data. Values of 1 through 3 specify specific subsets of the sensor data.

Force-Seeking-Dock Command opcode: 143 Number of data bytes: 0

Turns on force-seeking-dock mode, which causes the robot to immediately attempt to dock during its cleaning cycle if it encounters the docking beams from the Home Base. (Note, however, that if the robot was not active in a clean, spot or max cycle it will not attempt to execute the docking.) Normally the robot attempts to dock only if the cleaning cycle has completed or the battery is nearing depletion. This command can be sent anytime, but the mode will be cancelled if the robot turns off, begins charging, or is commanded into SCI safe or full modes.

Serial sequence: [143]

Roomba SCI Sensor Packets

The robot will send back one of four different sensor data packets in response to a Sensor command, depending on the value of the packet code data byte. The data bytes are specified below in the order in which they will be sent. A packet code value of 0 sends all of the data bytes. A value of 1 through 3 sends a subset of the sensor data. Some of the sensor data values are 16 bit values. These values are sent as two bytes, high byte first.

Sensor Packet Sizes

Packet code	Packet Size
0	26 bytes
1	10 bytes
2	6 bytes
3	10 bytes

Bumps Wheeldrops

Packet subset: 1

Range: 0 - 31

Data type: 1 byte, unsigned

The state of the bump (0 = no bump, 1 = bump) and wheeldrop sensors (0 = wheel up, 1 = wheel dropped) are sent as individual bits.

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

Note: Some robots do not report the three wheel drops separately. Instead, if any of the three wheels drops, all three wheel-drop bits will be set. You can tell which kind of robot you have by examining the serial number inside the battery compartment. Wheel drops are separate only if there is an "E" in the serial number.

Wall

Packet subset: 1

Range: 0 – 1

Data type: 1 byte, unsigned

The state of the wall sensor is sent as a 1 bit value (0 = no wall, 1 = wall seen).

Cliff Left

Packet subset: 1

Range: 0 – 1

Data type: 1 byte, unsigned

The state of the cliff sensor on the left side of Roomba is sent as a 1 bit value (0 = no cliff, 1 = cliff).

Cliff Front Left

Packet subset: 1

Range: 0 – 1

Data type: 1 byte, unsigned

The state of the cliff sensor on the front left side of Roomba is sent as a 1 bit value (0 = no cliff, 1 = cliff).

Cliff Front Right

Packet subset: 1

Range: 0 – 1

Data type: 1 byte, unsigned

The state of the cliff sensor on the front right side of Roomba is sent as a 1 bit value (0 = no cliff, 1 = cliff)

Cliff Right

Packet subset: 1

Range: 0 – 1

Data type: 1 byte, unsigned

The state of the cliff sensor on the right side of Roomba is sent as a 1 bit value (0 = no cliff, 1 = cliff)

Virtual Wall

Packet subset: 1

Range: 0 – 1

Data type: 1 byte, unsigned

The state of the virtual wall detector is sent as a 1 bit value (0 = no virtual wall detected, 1 = virtual wall detected)

Motor Overcurrents

Packet subset: 1

Range: 0 – 31

The state of the five motors' overcurrent sensors are sent as individual bits (0 = no overcurrent, 1 = overcurrent).

Bit	7	6	5	4	3	2	1	0
Motor	n/a	n/a	n/a	Drive Left	Drive Right	Main Brush	Vacuum	Side Brush

Dirt Detector Left

Packet subset: 1

Range: 0 - 255

Data type: 1 byte, unsigned

The current dirt detection level of the left side dirt detector is sent as a one byte value. A value of 0 indicates no dirt is detected. Higher values indicate higher levels of dirt detected.

Dirt Detector Right

Packet subset: 1

Range: 0 – 255

Data type: 1 byte, unsigned

The current dirt detection level of the right side dirt detector is sent as a one byte value. A value of 0 indicates no dirt is detected. Higher values indicate higher levels of dirt detected.

Note: Some robots don't have a right dirt detector. You can tell by removing the brushes. The dirt detectors are metallic disks. For robots with no right dirt detector this byte is always 0.

Remote Control Command

Packet subset: 2

Range: 0 – 255 (with some values unused)

Data type: 1 byte, unsigned

The command number of the remote control command currently being received by Roomba. A value of 255 indicates that no remote control command is being received. See Roomba remote control documentation for a description of the command values.

Buttons

Packet subset: 2

Range: 0 – 15

Data type: 1 byte, unsigned

The state of the four Roomba buttons are sent as individual bits (0 = button not pressed, 1 = button pressed).

Bit	7	6	5	4	3	2	1	0
Button	n/a	n/a	n/a	n/a	Power	Spot	Clean	Max

Distance

Packet subset: 2

Range: -32768 – 32767

Data type: 2 bytes, signed

The distance that Roomba has traveled in millimeters since the distance it was last requested. This is the same as the sum of the distance traveled by both wheels divided by two. Positive values indicate travel in the forward direction; negative in the reverse direction. If the value is not polled frequently enough, it will be capped at its minimum or maximum.

Angle

Packet subset: 2

Range: -32768 – 32767

Data type: 2 bytes, signed

The angle that Roomba has turned through since the angle was last requested. The angle is expressed as the difference in the distance traveled by Roomba's two wheels in millimeters, specifically the right wheel distance minus the left wheel distance, divided by two. This makes counter-clockwise angles positive and clockwise angles negative. This can be used to directly calculate the angle that Roomba has turned through since the last request. Since the distance between Roomba's wheels is 258mm, the equations for calculating the angles in familiar units are:

Angle in radians = (2 * difference) / 258

Angle in degrees = (360 * difference) / (258 * Pi).

If the value is not polled frequently enough, it will be capped at its minimum or maximum.

Note: Reported angle and distance may not be accurate. Roomba measures these by detecting its wheel revolutions. If for example, the wheels slip on the floor, the reported angle of distance will be greater than the actual angle or distance.

Charging State

Packet subset: 3

Range: 0 – 5

Data type: 1 byte, unsigned

A code indicating the current charging state of Roomba.

Code	Charging State
0	Not Charging
1	Charging Recovery
2	Charging
3	Trickle Charging
4	Waiting
5	Charging Error

Voltage

Packet subset: 3

Range: 0 – 65535

Data type: 2 bytes, unsigned

The voltage of Roomba's battery in millivolts (mV).

Current

Packet subset: 3

Range: -32768 – 32767

Data type: 2 bytes, signed

The current in milliamps (mA) flowing into or out of Roomba's battery. Negative currents indicate current is flowing out of the battery, as during normal running. Positive currents indicate current is flowing into the battery, as during charging.

Temperature

Packet subset: 3

Range: -128 – 127

Data type: 1 byte, signed

The temperature of Roomba's battery in degrees Celsius.

Charge

Packet subset: 3

Range: 0 – 65535

Data type: 2 bytes, unsigned

The current charge of Roomba's battery in milliamp-hours (mAh). The charge value decreases as the battery is depleted during running and increases when the battery is charged.

Capacity

Packet subset: 3

Range: 0 – 65535

Data type: 2 bytes, unsigned

The estimated charge capacity of Roomba's battery. When the Charge value reaches the Capacity value, the battery is fully charged.

Roomba SCI Commands Quick Reference

Command	Opcode	Data Byte 1	Data Byte 2	Data Byte 3	Data Byte 4	Etc.
Start	128					
Baud	129	Baud Code (0 – 11)				
Control	130					
Safe	131					
Full	132					
Power	133					
Spot	134					
Clean	135					
Max	136					
Drive	137	Velocity (-500 – 500)		Radius (-2000 – 2000)		
Motors	138	Motor Bits (0 – 7)				
Leds	139	Led Bits (0 – 63)	Power Color (0 – 255)	Power Intensity (0 – 255)		
Song	140	Song Number (0 – 15)	Song Length (0 – 15)	Note Number 1 (31 – 127)	Note Duration 1 (0 – 255)	Note Number 2, etc.
Play	141	Song Number (0 – 15)				
Sensors	142	Packet Code (0 – 3)				
Force-Seeking-Dock	143					

Baud data byte 1: Baud Code (0 – 9)

Baud code	Baud rate in bps
0	300
1	600
2	1200
3	2400
4	4800
5	9600
6	14400
7	19200
8	28800
9	38400
10	57600
11	115200

Motors data byte 1: Motor Bits

0 = off, 1 = on

Bit	7	6	5	4	3	2	1	0
Motor	n/a	n/a	n/a	n/a	n/a	Main Brush	Vacuum	Side Brush

Leds data byte 1: Led Bits (0 – 63)

Dirt Detect uses a blue LED: 0 = off, 1 = on

Spot, Clean, and Max use green LEDs: 0 = off, 1 = on

Status uses a bicolor (red/green) LED: 00 = off, 01 = red, 10 = green, 11 = amber

Bit	7	6	5	4	3	2	1	0
LED	n/a	n/a	Status (2 bits)		Spot	Clean	Max	Dirt Detect

Power uses a bicolor (red/green) LED whose intensity and color can be controlled with 8-bit resolution.

Leds data byte 2: Power Color (0 – 255)

0 = green, 255 = red. Intermediate values are intermediate colors.

Leds data byte 3: Power Intensity (0 – 255)

0 = off, 255 = full intensity. Intermediate values are intermediate intensities.

Roomba SCI Sensors Quick Reference

Packet Code	Packet Size
0	26 bytes
1	10 bytes
2	6 bytes
3	10 bytes

Name	Groups	Bytes	Value Range	Units
Bumps Wheeldrops	0, 1	1	0 – 31	
Wall	0, 1	1	0 – 1	
Cliff Left	0, 1	1	0 – 1	
Cliff Front Left	0, 1	1	0 – 1	
Cliff Front Right	0, 1	1	0 – 1	
Cliff Right	0, 1	1	0 – 1	
Virtual Wall	0, 1	1	0 – 1	
Motor Overcurrents	0, 1	1	0 – 31	
Dirt Detector - Left	0, 1	1	0 – 255	
Dirt Detector - Right	0, 1	1	0 – 255	
Remote Opcode	0, 2	1	0 – 255	
Buttons	0, 2	1	0 – 15	
Distance	0, 2	2*	-32768 – 32767	mm
Angle	0, 2	2*	-32768 – 32767	mm
Charging State	0, 3	1	0 – 5	
Voltage	0, 3	2*	0 – 65535	mV
Current	0, 3	2*	-32768 – 32767	mA
Temperature	0, 3	1	-128 – 127	degrees C
Charge	0, 3	2*	0 – 65535	mAh
Capacity	0, 3	2*	0 – 65535	mAh

** For 2 byte sensor values, high byte is sent first, followed by low byte.*

Bumps Wheeldrops

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

Motor Overcurrents

Bit	7	6	5	4	3	2	1	0
Motor	n/a	n/a	n/a	Drive Left	Drive Right	Main Brush	Vacuum	Side Brush

Buttons

Bit	7	6	5	4	3	2	1	0
Button	n/a	n/a	n/a	n/a	Power	Spot	Clean	Max

Charging State Codes

Code	Charging State
0	Not Charging
1	Charging Recovery
2	Charging
3	Trickle Charging
4	Waiting
5	Charging Error