

# chrome.serial

|               |   |
|---------------|---|
| Description:  | Use the <code>chrome.serial</code> API to read from and write to a device connected to a serial port. |
| Availability: | Since Chrome 23.  |
| Permissions:  | <code>"serial"</code>   |
| Learn More:   | <a href="#">Accessing Hardware Devices</a>  |

## Summary

| Types   |
|---|
| <a href="#">DataBits</a>  |
| <a href="#">ParityBit</a>   |
| <a href="#">StopBits</a>  |
| <a href="#">ConnectionOptions</a>   |
| <a href="#">ConnectionInfo</a>  |
| Methods   |
| <a href="#">getDevices</a> - <code>chrome.serial.getDevices(function callback)</code>   |
| <a href="#">connect</a> - <code>chrome.serial.connect(string path, ConnectionOptions options, function callback)</code>                   |
| <a href="#">update</a> - <code>chrome.serial.update(integer connectionId, ConnectionOptions options, function callback)</code>            |
| <a href="#">disconnect</a> - <code>chrome.serial.disconnect(integer connectionId, function callback)</code>                               |
| <a href="#">setPaused</a> - <code>chrome.serial.setPaused(integer connectionId, boolean paused, function callback)</code>                 |
| <a href="#">getInfo</a> - <code>chrome.serial.getInfo(integer connectionId, function callback)</code>                                     |
| <a href="#">getConnections</a> - <code>chrome.serial.getConnections(function callback)</code>   |
| <a href="#">send</a> - <code>chrome.serial.send(integer connectionId, ArrayBuffer data, function callback)</code>                         |
| <a href="#">flush</a> - <code>chrome.serial.flush(integer connectionId, function callback)</code>   |
| <a href="#">getControlSignals</a> - <code>chrome.serial.getControlSignals(integer connectionId, function callback)</code>                 |
| <a href="#">setControlSignals</a> - <code>chrome.serial.setControlSignals(integer connectionId, object signals, function callback)</code> |
| <a href="#">setBreak</a> - <code>chrome.serial.setBreak(integer connectionId, function callback)</code>                                   |
| <a href="#">clearBreak</a> - <code>chrome.serial.clearBreak(integer connectionId, function callback)</code>                               |
| Events  |
| <a href="#">onReceive</a>   |
| <a href="#">onReceiveError</a>  |

## Types

### DataBits

| Enum |
|------|
|------|

"seven", or "eight"

## ParityBit

Enum

"no", "odd", or "even"

## StopBits

Enum

"one", or "two"

## ConnectionOptions

Since Chrome 33.

| properties       |                              |  |
|------------------|------------------------------|--|
| boolean          | (optional)<br>persistent     | Flag indicating whether or not the connection should be left open when the application is suspended (see <a href="#">Manage App Lifecycle</a> ). The default value is "false." When the application is loaded, any serial connections previously opened with persistent=true can be fetched with <a href="#">getConnections</a> .  |
| string           | (optional)<br>name           | An application-defined string to associate with the connection.  |
| integer          | (optional)<br>bufferSize     | The size of the buffer used to receive data. The default value is 4096.  |
| integer          | (optional)<br>bitrate        | The requested bitrate of the connection to be opened. For compatibility with the widest range of hardware, this number should match one of commonly-available bitrates, such as 110, 300, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200. There is no guarantee, of course, that the device connected to the serial port will support the requested bitrate, even if the port itself supports that bitrate. <b>9600</b> will be passed by default. |
| <b>DataBits</b>  | (optional)<br>dataBits       | " <b>eight</b> " will be passed by default.  |
| <b>ParityBit</b> | (optional)<br>parityBit      | "no" will be passed by default.  |
| <b>StopBits</b>  | (optional)<br>stopBits       | "one" will be passed by default.   |
| boolean          | (optional)<br>ctsFlowControl | Flag indicating whether or not to enable RTS/CTS hardware flow control. Defaults to false.   |
| integer          | (optional)<br>receiveTimeout | The maximum amount of time (in milliseconds) to wait for new data before raising an <a href="#">onReceiveError</a> event with a "timeout" error. If zero, receive timeout errors will not be raised for the connection. Defaults to 0.   |
| integer          | (optional)<br>sendTimeout    | The maximum amount of time (in milliseconds) to wait for a <a href="#">send</a> operation to complete before calling the callback with a "timeout" error. If zero, send timeout errors will not be triggered. Defaults to 0.   |

## ConnectionInfo

Since Chrome 33.

| properties       |                           |  |
|------------------|---------------------------|--|
| integer          | connectionId              | The id of the serial port connection.  |
| boolean          | paused                    | Flag indicating whether the connection is blocked from firing onReceive events.  |
| boolean          | persistent                | See <a href="#">ConnectionOptions.persistent</a>   |
| string           | name                      | See <a href="#">ConnectionOptions.name</a>   |
| integer          | bufferSize                | See <a href="#">ConnectionOptions.bufferSize</a>   |
| integer          | receiveTimeout            | See <a href="#">ConnectionOptions.receiveTimeout</a>   |
| integer          | sendTimeout               | See <a href="#">ConnectionOptions.sendTimeout</a>  |
| integer          | (optional) bitrate        | See <a href="#">ConnectionOptions.bitrate</a> . This field may be omitted or inaccurate if a non-standard bitrate is in use, or if an error occurred while querying the underlying device. |
| <b>DataBits</b>  | (optional) dataBits       | See <a href="#">ConnectionOptions.dataBits</a> . This field may be omitted if an error occurred while querying the underlying device.  |
| <b>ParityBit</b> | (optional) parityBit      | See <a href="#">ConnectionOptions.parityBit</a> . This field may be omitted if an error occurred while querying the underlying device.   |
| <b>StopBits</b>  | (optional) stopBits       | See <a href="#">ConnectionOptions.stopBits</a> . This field may be omitted if an error occurred while querying the underlying device.  |
| boolean          | (optional) ctsFlowControl | See <a href="#">ConnectionOptions.ctsFlowControl</a> . This field may be omitted if an error occurred while querying the underlying device.  |

## Methods

### getDevices

`chrome.serial.getDevices(function callback)`

Since Chrome 33.

Returns information about available serial devices on the system. The list is regenerated each time this method is called.

| Parameters |          |   |
|------------|----------|---|
| function   | callback | Called with the list of <a href="#">DeviceInfo</a> objects. |

The callback parameter should be a function that looks like this:

```
function(array of object ports) {...};
```

| array<br>of<br>object | ports | Properties of each object |                           |   |
|-----------------------|-------|---------------------------|---------------------------|---|
|                       |       | string                    | path                      | The device's system path. This should be passed as the <b>path</b> argument to <b>chrome.serial.connect</b> in order to connect to this device. |
|                       |       | integer                   | (optional)<br>vendorId    | A PCI or USB vendor ID if one can be determined for the underlying device.  |
|                       |       | integer                   | (optional)<br>productId   | A USB product ID if one can be determined for the underlying device.  |
|                       |       | string                    | (optional)<br>displayName | A human-readable display name for the underlying device if one can be queried from the host driver.   |

## connect

```
chrome.serial.connect(string path, ConnectionOptions options, function callback)
```

Since Chrome 33.

Connects to a given serial port.

| Parameters               |                    |   |                       |                |  |
|--------------------------|--------------------|---|-----------------------|----------------|--|
| string                   | path               | The system path of the serial port to open.   |                       |                |  |
| <b>ConnectionOptions</b> | (optional) options | Port configuration options.   |                       |                |  |
| function                 | callback           | <div>Called when the connection has been opened.</div> <div>The callback parameter should be a function that looks like this:</div> <div><pre>function( ConnectionInfo connectionInfo) {...};</pre><table><tr><td><b>ConnectionInfo</b></td><td>connectionInfo</td><td></td></tr></table></div> | <b>ConnectionInfo</b> | connectionInfo |  |
| <b>ConnectionInfo</b>    | connectionInfo     |   |                       |                |  |

## update

```
chrome.serial.update(integer connectionId, ConnectionOptions options, function callback)
```

Since Chrome 33.

Update the option settings on an open serial port connection.

| Parameters               |              |  |
|--------------------------|--------------|--|
| integer                  | connectionId | The id of the opened connection.   |
| <b>ConnectionOptions</b> | options      | Port configuration options.  |
| function                 | callback     | <div>Called when the configuration has completed.</div> <div>The callback parameter should be a function that looks like this:</div> |

```
function(boolean result) {...};
```

|         |
|---------|
| boolean |
|---------|

|        |
|--------|
| result |
|--------|

## disconnect

```
chrome.serial.disconnect(integer connectionId, function callback)
```

Since Chrome 33.

Disconnects from a serial port.

### Parameters

|         |
|---------|
| integer |
|---------|

|              |
|--------------|
| connectionId |
|--------------|

The id of the opened connection.

|          |
|----------|
| function |
|----------|

|          |
|----------|
| callback |
|----------|

Called when the connection has been closed.

The callback parameter should be a function that looks like this:

```
function(boolean result) {...};
```

|         |
|---------|
| boolean |
|---------|

|        |
|--------|
| result |
|--------|

## setPaused

```
chrome.serial.setPaused(integer connectionId, boolean paused, function callback)
```

Since Chrome 33.

Pauses or unpauses an open connection.

### Parameters

|         |
|---------|
| integer |
|---------|

|              |
|--------------|
| connectionId |
|--------------|

The id of the opened connection.

|         |
|---------|
| boolean |
|---------|

|        |
|--------|
| paused |
|--------|

Flag to indicate whether to pause or unpause.

|          |
|----------|
| function |
|----------|

|          |
|----------|
| callback |
|----------|

Called when the connection has been successfully paused or unpaused.

The callback parameter should be a function that looks like this:

```
function() {...};
```

## getInfo

```
chrome.serial.getInfo(integer connectionId, function callback)
```

Since Chrome 33.

Retrieves the state of a given connection.

### Parameters

|         |
|---------|
| integer |
|---------|

|              |
|--------------|
| connectionId |
|--------------|

The id of the opened connection.

|          |
|----------|
| function |
|----------|

|          |
|----------|
| callback |
|----------|

Called with connection state information when available.

The callback parameter should be a function that looks like this:

```
function( ConnectionInfo connectionInfo) {...};
```

|                |
|----------------|
| ConnectionInfo |
|----------------|

|                |
|----------------|
| connectionInfo |
|----------------|

## getConnections

chrome.serial.getConnections(function callback)

Since Chrome 33.

Retrieves the list of currently opened serial port connections owned by the application.

### Parameters

|          |
|----------|
| function |
|----------|

|          |
|----------|
| callback |
|----------|

Called with the list of connections when available.

The callback parameter should be a function that looks like this:

```
function(array of ConnectionInfo connectionInfos) {...};
```

|                         |
|-------------------------|
| array of ConnectionInfo |
|-------------------------|

|                 |
|-----------------|
| connectionInfos |
|-----------------|

## send

chrome.serial.send(integer connectionId, ArrayBuffer data, function callback)

Since Chrome 33.

Writes data to the given connection.

### Parameters

|         |
|---------|
| integer |
|---------|

|              |
|--------------|
| connectionId |
|--------------|

The id of the connection.

|             |
|-------------|
| ArrayBuffer |
|-------------|

|      |
|------|
| data |
|------|

The data to send.

|          |
|----------|
| function |
|----------|

|          |
|----------|
| callback |
|----------|

Called when the operation has completed.

The callback parameter should be a function that looks like this:

```
function(object sendInfo) {...};
```

|        |
|--------|
| object |
|--------|

|          |
|----------|
| sendInfo |
|----------|

|         |
|---------|
| integer |
|---------|

|           |
|-----------|
| bytesSent |
|-----------|

|                           |
|---------------------------|
| The number of bytes sent. |
|---------------------------|

|   |
|---|
| enum of "disconnected", "pending", "timeout", or "system_error" |
|---|

|                  |
|------------------|
| (optional) error |
|------------------|

|   |
|---|
| An error code if an error occurred.<br><br>disconnected<br>The connection was disconnected.<br><br>pending<br>A send was already pending.<br><br>timeout<br>The send timed out. |
|---|

|  |  |  |  |  |  |  |
|--|--|--|--|--|--|--|
|  |  |  |  |  |  | system_error   |
|  |  |  |  |  |  | A system error occurred and the connection may be unrecoverable. |

**flush**

```
chrome.serial.flush(integer connectionId, function callback)
```

Flushes all bytes in the given connection's input and output buffers.

| Parameters |              |   |         |        |  |
|------------|--------------|---|---------|--------|--|
| integer    | connectionId |   |         |        |  |
| function   | callback     | <div>The callback parameter should be a function that looks like this:<br/><pre>function(boolean result) {...};</pre><table><tr><td>boolean</td><td>result</td><td></td></tr></table></div> | boolean | result |  |
| boolean    | result       |   |         |        |  |

**getControlSignals**

```
chrome.serial.getControlSignals(integer connectionId, function callback)
```

Retrieves the state of control signals on a given connection.

| Parameters |              |   |         |         |  |         |     |   |         |     |  |         |     |                       |         |    |                      |         |     |                       |
|------------|--------------|---|---------|---------|--|---------|-----|---|---------|-----|--|---------|-----|-----------------------|---------|----|----------------------|---------|-----|-----------------------|
| integer    | connectionId | The id of the connection.   |         |         |  |         |     |   |         |     |  |         |     |                       |         |    |                      |         |     |                       |
| function   | callback     | <div>Called when the control signals are available.<br/>The callback parameter should be a function that looks like this:<br/><pre>function(object signals) {...};</pre><table><tr><td>object</td><td>signals</td><td></td></tr><tr><td></td><td></td><td><table><tr><td>boolean</td><td>dcd</td><td>DCD (Data Carrier Detect) or RLSD (Receive Line Signal/ Detect).</td></tr><tr><td>boolean</td><td>cts</td><td>CTS (Clear To Send).</td></tr><tr><td>boolean</td><td>ri</td><td>RI (Ring Indicator).</td></tr><tr><td>boolean</td><td>dsr</td><td>DSR (Data Set Ready).</td></tr></table></td></tr></table></div> | object  | signals |  |         |     | <table><tr><td>boolean</td><td>dcd</td><td>DCD (Data Carrier Detect) or RLSD (Receive Line Signal/ Detect).</td></tr><tr><td>boolean</td><td>cts</td><td>CTS (Clear To Send).</td></tr><tr><td>boolean</td><td>ri</td><td>RI (Ring Indicator).</td></tr><tr><td>boolean</td><td>dsr</td><td>DSR (Data Set Ready).</td></tr></table> | boolean | dcd | DCD (Data Carrier Detect) or RLSD (Receive Line Signal/ Detect). | boolean | cts | CTS (Clear To Send).  | boolean | ri | RI (Ring Indicator). | boolean | dsr | DSR (Data Set Ready). |
| object     | signals      |   |         |         |  |         |     |   |         |     |  |         |     |                       |         |    |                      |         |     |                       |
|            |              | <table><tr><td>boolean</td><td>dcd</td><td>DCD (Data Carrier Detect) or RLSD (Receive Line Signal/ Detect).</td></tr><tr><td>boolean</td><td>cts</td><td>CTS (Clear To Send).</td></tr><tr><td>boolean</td><td>ri</td><td>RI (Ring Indicator).</td></tr><tr><td>boolean</td><td>dsr</td><td>DSR (Data Set Ready).</td></tr></table>   | boolean | dcd     | DCD (Data Carrier Detect) or RLSD (Receive Line Signal/ Detect). | boolean | cts | CTS (Clear To Send).  | boolean | ri  | RI (Ring Indicator).   | boolean | dsr | DSR (Data Set Ready). |         |    |                      |         |     |                       |
| boolean    | dcd          | DCD (Data Carrier Detect) or RLSD (Receive Line Signal/ Detect).  |         |         |  |         |     |   |         |     |  |         |     |                       |         |    |                      |         |     |                       |
| boolean    | cts          | CTS (Clear To Send).  |         |         |  |         |     |   |         |     |  |         |     |                       |         |    |                      |         |     |                       |
| boolean    | ri           | RI (Ring Indicator).  |         |         |  |         |     |   |         |     |  |         |     |                       |         |    |                      |         |     |                       |
| boolean    | dsr          | DSR (Data Set Ready).   |         |         |  |         |     |   |         |     |  |         |     |                       |         |    |                      |         |     |                       |

**setControlSignals**

```
chrome.serial.setControlSignals(integer connectionId, object signals, function callback)
```

Sets the state of control signals on a given connection.

| Parameters |  |  |
|------------|--|--|
|            |  |  |

|          |                |   |         |                |                            |         |                |                        |
|----------|----------------|---|---------|----------------|----------------------------|---------|----------------|------------------------|
| integer  | connectionId   | The id of the connection.   |         |                |                            |         |                |                        |
| object   | signals        | <p><b>Since Chrome 33.</b></p> <p>The set of signal changes to send to the device.</p> <table> <tr> <td>boolean</td><td>(optional) dtr</td><td>DTR (Data Terminal Ready).</td></tr> <tr> <td>boolean</td><td>(optional) rts</td><td>RTS (Request To Send).</td></tr> </table> | boolean | (optional) dtr | DTR (Data Terminal Ready). | boolean | (optional) rts | RTS (Request To Send). |
| boolean  | (optional) dtr | DTR (Data Terminal Ready).  |         |                |                            |         |                |                        |
| boolean  | (optional) rts | RTS (Request To Send).  |         |                |                            |         |                |                        |
| function | callback       | <p>Called once the control signals have been set.</p> <p>The callback parameter should be a function that looks like this:</p> <pre>function(boolean result) {...};</pre> <table> <tr> <td>boolean</td><td>result</td><td></td></tr> </table>                                 | boolean | result         |                            |         |                |                        |
| boolean  | result         |   |         |                |                            |         |                |                        |

## setBreak

`chrome.serial.setBreak(integer connectionId, function callback)`

**Since Chrome 45.**

Suspends character transmission on a given connection and places the transmission line in a break state until the `clearBreak` is called.

| Parameters |              |   |         |        |  |
|------------|--------------|---|---------|--------|--|
| integer    | connectionId | The id of the connection.   |         |        |  |
| function   | callback     | <p>The callback parameter should be a function that looks like this:</p> <pre>function(boolean result) {...};</pre> <table> <tr> <td>boolean</td><td>result</td><td></td></tr> </table> | boolean | result |  |
| boolean    | result       |   |         |        |  |

## clearBreak

`chrome.serial.clearBreak(integer connectionId, function callback)`

**Since Chrome 45.**

Restore character transmission on a given connection and place the transmission line in a nonbreak state.

| Parameters |              |   |         |        |  |
|------------|--------------|---|---------|--------|--|
| integer    | connectionId | The id of the connection.   |         |        |  |
| function   | callback     | <p>The callback parameter should be a function that looks like this:</p> <pre>function(boolean result) {...};</pre> <table> <tr> <td>boolean</td><td>result</td><td></td></tr> </table> | boolean | result |  |
| boolean    | result       |   |         |        |  |

## Events



onReceive

Since Chrome 33.

Event raised when data has been read from the connection.

addListener

chrome.serial.onReceive.addListener(function callback)

| Parameters |          |   |              |                            |
|------------|----------|---|--------------|----------------------------|
| function   | callback | The callback parameter should be a function that looks like this:<br><br>function(object info) {...}; |              |                            |
|            |          | object  | info         | Event data.                |
|            |          |   | integer      | connectionId               |
|            |          |   | connectionId | The connection identifier. |
|            |          |   | ArrayBuffer  | data                       |
|            |          |   | data         | The data received.         |

onReceiveError

Since Chrome 33.

Event raised when an error occurred while the runtime was waiting for data on the serial port. Once this event is raised, the connection may be set to paused. A "timeout" error does not pause the connection.

addListener

chrome.serial.onReceiveError.addListener(function callback)

| Parameters |          |   |   |  |
|------------|----------|---|---|--|
| function   | callback | The callback parameter should be a function that looks like this:<br><br>function(object info) {...}; |   |  |
|            |          | object  | info  |  |
|            |          |   | integer   | connectionId   |
|            |          |   | connectionId  | The connection identifier.                                 |
|            |          |   | enum of "disconnected", "timeout", "device_lost", "break", "frame_error", "overrun", "buffer_overflow", "parity_error", or "system_error" | error  |
|            |          |   |   | An error code indicating what went wrong.                  |
|            |          |   |   | disconnected   |
|            |          |   |   | The connection was disconnected.                           |
|            |          |   |   | timeout  |
|            |          |   |   | No data has been received for receiveTimeout milliseconds. |
|            |          |   |   | device_lost  |
|            |          |   |   | The device was most likely disconnected from the host.     |
|            |          |   |   | break  |
|            |          |   |   | The device detected a break condition.                     |

|  |  |  |  |  |  |   |
|--|--|--|--|--|--|---|
|  |  |  |  |  |  | <div>frame_error</div> <div>The device detected a framing error.</div> <div>overrun</div> <div>A character-buffer overrun has occurred. The next character is lost.</div> <div>buffer_overflow</div> <div>An input buffer overflow has occurred. There is either no room in the input buffer, or a character was received after the end-of-file (EOF) character.</div> <div>parity_error</div> <div>The device detected a parity error.</div> <div>system_error</div> <div>A system error occurred and the connection may be unrecoverable.</div> |
|--|--|--|--|--|--|---|