

# ioctl CEC\_RECEIVE and CEC\_TRANSMIT

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 2)**

Unknown directive type "c:namespace".

```
.. c:namespace:: CEC
```

## Name

CEC\_RECEIVE, CEC\_TRANSMIT - Receive or transmit a CEC message

## Synopsis

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 19)**

Unknown directive type "c:macro".

```
.. c:macro:: CEC_RECEIVE
```

```
int ioctl(int fd, CEC_RECEIVE, struct cec_msg *argp)
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 23)**

Unknown directive type "c:macro".

```
.. c:macro:: CEC_TRANSMIT
```

```
int ioctl(int fd, CEC_TRANSMIT, struct cec_msg *argp)
```

## Arguments

fd

File descriptor returned by `c:func:'open()'`.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 31); [backlink](#)**

Unknown interpreted text role "c:func".

argp

Pointer to struct `cec_msg`.

## Description

To receive a CEC message the application has to fill in the `timeout` field of struct `c:type:'cec_msg'` and pass it to `ref:'ioctl CEC_RECEIVE <CEC_RECEIVE>'`. If the file descriptor is in non-blocking mode and there are no received messages pending, then it will return -1 and set `errno` to the `EAGAIN` error code. If the file descriptor is in blocking mode and `timeout` is non-zero and no message arrived within `timeout` milliseconds, then it will return -1 and set `errno` to the `ETIMEDOUT` error code.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 39); [backlink](#)**

Unknown interpreted text role "c:type".

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-**

master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 39); [backlink](#)

Unknown interpreted text role "ref".

A received message can be:

1. a message received from another CEC device (the `sequence` field will be 0, `tx_status` will be 0 and `rx_status` will be non-zero).
2. the transmit result of an earlier non-blocking transmit (the `sequence` field will be non-zero, `tx_status` will be non-zero and `rx_status` will be 0).
3. the reply to an earlier non-blocking transmit (the `sequence` field will be non-zero, `tx_status` will be 0 and `rx_status` will be non-zero).

To send a CEC message the application has to fill in the struct `:c:type:'cec_msg'` and pass it to `ref:'ioctl CEC_TRANSMIT <CEC_TRANSMIT>'`. The `ref:'ioctl CEC_TRANSMIT <CEC_TRANSMIT>'` is only available if `CEC_CAP_TRANSMIT` is set. If there is no more room in the transmit queue, then it will return -1 and set `errno` to the `EBUSY` error code. The transmit queue has enough room for 18 messages (about 1 second worth of 2-byte messages). Note that the CEC kernel framework will also reply to core messages (see `ref:'cec-core-processing'`), so it is not a good idea to fully fill up the transmit queue.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 58); [backlink](#)

Unknown interpreted text role "c:type".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 58); [backlink](#)

Unknown interpreted text role "ref".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 58); [backlink](#)

Unknown interpreted text role "ref".

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 58); [backlink](#)

Unknown interpreted text role "ref".

If the file descriptor is in non-blocking mode then the transmit will return 0 and the result of the transmit will be available via `ref:'ioctl CEC_RECEIVE <CEC_RECEIVE>'` once the transmit has finished. If a non-blocking transmit also specified waiting for a reply, then the reply will arrive in a later message. The `sequence` field can be used to associate both transmit results and replies with the original transmit.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 68); [backlink](#)

Unknown interpreted text role "ref".

Normally calling `ref:'ioctl CEC_TRANSMIT <CEC_TRANSMIT>'` when the physical address is invalid (due to e.g. a disconnect) will return `ENONET`.

**System Message: ERROR/3** (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 76); [backlink](#)

Unknown interpreted text role "ref".

However, the CEC specification allows sending messages from 'Unregistered' to 'TV' when the physical address is invalid since some TVs pull the hotplug detect pin of the HDMI connector low when they go into standby, or when switching to another input.

When the hotplug detect pin goes low the EDID disappears, and thus the physical address, but the cable is still connected and CEC still works. In order to detect/wake up the device it is allowed to send poll and 'Image/Text View On' messages from initiator 0xf ('Unregistered') to destination 0 ('TV').

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 89)**

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{1.0cm}|p{3.5cm}|p{12.8cm}|
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 91)**

Unknown directive type "c:type".

```
.. c:type:: cec_msg
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 93)**

Unknown directive type "cssclass".

```
.. cssclass:: longtable
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 95)**

Unknown directive type "flat-table".

```
.. flat-table:: struct cec_msg
   :header-rows: 0
   :stub-columns: 0
   :widths:      1 1 16

* - u64
  - ``tx_ts``
  - Timestamp in ns of when the last byte of the message was transmitted.
    The timestamp has been taken from the ``CLOCK_MONOTONIC`` clock. To access
    the same clock from userspace use :c:func:`clock_gettime`.
* - u64
  - ``rx_ts``
  - Timestamp in ns of when the last byte of the message was received.
    The timestamp has been taken from the ``CLOCK_MONOTONIC`` clock. To access
    the same clock from userspace use :c:func:`clock_gettime`.
* - u32
  - ``len``
  - The length of the message. For :ref:`ioctl CEC_TRANSMIT <CEC_TRANSMIT>` this is filled in
    by the application. The driver will fill this in for
    :ref:`ioctl CEC_RECEIVE <CEC_RECEIVE>`. For :ref:`ioctl CEC_TRANSMIT <CEC_TRANSMIT>` it will
    filled in by the driver with the length of the reply message if ``reply`` was set.
* - u32
  - ``timeout``
  - The timeout in milliseconds. This is the time the device will wait
    for a message to be received before timing out. If it is set to 0,
    then it will wait indefinitely when it is called by :ref:`ioctl CEC_RECEIVE <CEC_RECEIVE>`.
    If it is 0 and it is called by :ref:`ioctl CEC_TRANSMIT <CEC_TRANSMIT>`,
    then it will be replaced by 1000 if the ``reply`` is non-zero or
    ignored if ``reply`` is 0.
* - u32
  - ``sequence``
  - A non-zero sequence number is automatically assigned by the CEC framework
    for all transmitted messages. It is used by the CEC framework when it queues
    the transmit result for a non-blocking transmit. This allows the application
    to associate the received message with the original transmit.

    In addition, if a non-blocking transmit will wait for a reply (i.e. ``timeout``
    was not 0), then the ``sequence`` field of the reply will be set to the sequence
    value of the original transmit. This allows the application to associate the
    received message with the original transmit.
* - u32
  - ``flags``
  - Flags. See :ref:`cec-msg-flags` for a list of available flags.
* - u8
  - ``msg[16]``
  - The message payload. For :ref:`ioctl CEC_TRANSMIT <CEC_TRANSMIT>` this is filled in by the
    application. The driver will fill this in for :ref:`ioctl CEC_RECEIVE <CEC_RECEIVE>`.
    For :ref:`ioctl CEC_TRANSMIT <CEC_TRANSMIT>` it will be filled in by the driver with
    the payload of the reply message if ``timeout`` was set.
```

```

* - _u8
  - ``reply``
- Wait until this message is replied. If ``reply`` is 0 and the
  ``timeout`` is 0, then don't wait for a reply but return after
  transmitting the message. Ignored by :ref:`ioctl CEC_RECEIVE <CEC_RECEIVE>`.
  The case where ``reply`` is 0 (this is the opcode for the Feature Abort
  message) and ``timeout`` is non-zero is specifically allowed to make it
  possible to send a message and wait up to ``timeout`` milliseconds for a
  Feature Abort reply. In this case ``rx_status`` will either be set
  to :ref:`CEC_RX_STATUS_TIMEOUT <CEC-RX-STATUS-TIMEOUT>` or
  :ref:`CEC_RX_STATUS_FEATURE_ABORT <CEC-RX-STATUS-FEATURE-ABORT>`.

  If the transmitter message is ``CEC_MSG_INITIATE_ARC`` then the ``reply``
  values ``CEC_MSG_REPORT_ARC_INITIATED`` and ``CEC_MSG_REPORT_ARC_TERMINATED``
  are processed differently: either value will match both possible replies.
  The reason is that the ``CEC_MSG_INITIATE_ARC`` message is the only CEC
  message that has two possible replies other than Feature Abort. The
  ``reply`` field will be updated with the actual reply so that it is
  synchronized with the contents of the received message.

* - _u8
  - ``rx_status``
- The status bits of the received message. See
  :ref:`cec-rx-status` for the possible status values.

* - _u8
  - ``tx_status``
- The status bits of the transmitted message. See
  :ref:`cec-tx-status` for the possible status values.
  When calling :ref:`ioctl CEC_TRANSMIT <CEC_TRANSMIT>` in non-blocking mode,
  this field will be 0 if the transmit started, or non-0 if the transmit
  result is known immediately. The latter would be the case when attempting
  to transmit a Poll message to yourself. That results in a
  :ref:`CEC_TX_STATUS_NACK <CEC-TX-STATUS-NACK>` without ever actually
  transmitting the Poll message.

* - _u8
  - ``tx_arb_lost_cnt``
- A counter of the number of transmit attempts that resulted in the
  Arbitration Lost error. This is only set if the hardware supports
  this, otherwise it is always 0. This counter is only valid if the
  :ref:`CEC_TX_STATUS_ARB_LOST <CEC-TX-STATUS-ARB-LOST>` status bit is set.

* - _u8
  - ``tx_nack_cnt``
- A counter of the number of transmit attempts that resulted in the
  Not Acknowledged error. This is only set if the hardware supports
  this, otherwise it is always 0. This counter is only valid if the
  :ref:`CEC_TX_STATUS_NACK <CEC-TX-STATUS-NACK>` status bit is set.

* - _u8
  - ``tx_low_drive_cnt``
- A counter of the number of transmit attempts that resulted in the
  Arbitration Lost error. This is only set if the hardware supports
  this, otherwise it is always 0. This counter is only valid if the
  :ref:`CEC_TX_STATUS_LOW_DRIVE <CEC-TX-STATUS-LOW-DRIVE>` status bit is set.

* - _u8
  - ``tx_error_cnt``
- A counter of the number of transmit errors other than Arbitration
  Lost or Not Acknowledged. This is only set if the hardware
  supports this, otherwise it is always 0. This counter is only
  valid if the :ref:`CEC_TX_STATUS_ERROR <CEC-TX-STATUS-ERROR>` status bit is set.

```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 202)**

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{6.2cm}|p{1.0cm}|p{10.1cm}|
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 206)**

Unknown directive type "flat-table".

```

.. flat-table:: Flags for struct cec_msg
   :header-rows: 0
   :stub-columns: 0
   :widths:      3 1 4

   * .. _`CEC-MSG-FL-REPLY-TO-FOLLOWERS`:

      - ``CEC_MSG_FL_REPLY_TO_FOLLOWERS``
      - 1

```

- If a CEC transmit expects a reply, then by default that reply is only sent to the filehandle that called :ref:`ioctl CEC\_TRANSMIT <CEC\_TRANSMIT>`. If this flag is set, then the reply is also sent to all followers, if any. If the filehandle that called :ref:`ioctl CEC\_TRANSMIT <CEC\_TRANSMIT>` is also a follower, then that filehandle will receive the reply twice: once as the result of the :ref:`ioctl CEC\_TRANSMIT <CEC\_TRANSMIT>`, and once via :ref:`ioctl CEC\_RECEIVE <CEC\_RECEIVE>`.

```
* .. _`CEC-MSG-FL-RAW`:
```

- ``CEC\_MSG\_FL\_RAW``
- 2
- Normally CEC messages are validated before transmitting them. If this flag is set when :ref:`ioctl CEC\_TRANSMIT <CEC\_TRANSMIT>` is called, then no validation takes place and the message is transmitted as-is. This is useful when debugging CEC issues. This flag is only allowed if the process has the ``CAP\_SYS\_RAWIO`` capability. If that is not set, then the ``EPERM`` error code is returned.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 235)**

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{5.6cm}|p{0.9cm}|p{10.8cm}|
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 239)**

Unknown directive type "flat-table".

```
.. flat-table:: CEC Transmit Status
   :header-rows: 0
   :stub-columns: 0
   :widths:      3 1 16

* .. _`CEC-TX-STATUS-OK`:

- ``CEC_TX_STATUS_OK``
- 0x01
- The message was transmitted successfully. This is mutually exclusive with :ref:`CEC_TX_STATUS_MAX_RETRIES <CEC-TX-STATUS-MAX-RETRIES>`. Other bits can still be set if earlier attempts met with failure before the transmit was eventually successful.

* .. _`CEC-TX-STATUS-ARB-LOST`:

- ``CEC_TX_STATUS_ARB_LOST``
- 0x02
- CEC line arbitration was lost, i.e. another transmit started at the same time with a higher priority. Optional status, not all hardware can detect this error condition.

* .. _`CEC-TX-STATUS-NACK`:

- ``CEC_TX_STATUS_NACK``
- 0x04
- Message was not acknowledged. Note that some hardware cannot tell apart a 'Not Acknowledged' status from other error conditions, i.e. the result of a transmit is just OK or FAIL. In that case this status will be returned when the transmit failed.

* .. _`CEC-TX-STATUS-LOW-DRIVE`:

- ``CEC_TX_STATUS_LOW_DRIVE``
- 0x08
- Low drive was detected on the CEC bus. This indicates that a follower detected an error on the bus and requests a retransmission. Optional status, not all hardware can detect this error condition.

* .. _`CEC-TX-STATUS-ERROR`:

- ``CEC_TX_STATUS_ERROR``
- 0x10
- Some error occurred. This is used for any errors that do not fit ``CEC_TX_STATUS_ARB_LOST`` or ``CEC_TX_STATUS_LOW_DRIVE``, either because the hardware could not tell which error occurred, or because the hardware tested for other conditions besides those two. Optional status.

* .. _`CEC-TX-STATUS-MAX-RETRIES`:

- ``CEC_TX_STATUS_MAX_RETRIES``
```

```

- 0x20
- The transmit failed after one or more retries. This status bit is
  mutually exclusive with :ref:`CEC_TX_STATUS_OK <CEC-TX-STATUS-OK>`.
  Other bits can still be set to explain which failures were seen.
* .. _`CEC-TX-STATUS-ABORTED`:

- ``CEC_TX_STATUS_ABORTED``
- 0x40
- The transmit was aborted due to an HDMI disconnect, or the adapter
  was unconfigured, or a transmit was interrupted, or the driver
  returned an error when attempting to start a transmit.
* .. _`CEC-TX-STATUS-TIMEOUT`:

- ``CEC_TX_STATUS_TIMEOUT``
- 0x80
- The transmit timed out. This should not normally happen and this
  indicates a driver problem.

```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 304)**

Unknown directive type "tabularcolumns".

```
.. tabularcolumns:: |p{5.6cm}|p{0.9cm}|p{10.8cm}|
```

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 308)**

Unknown directive type "flat-table".

```

.. flat-table:: CEC Receive Status
   :header-rows: 0
   :stub-columns: 0
   :widths:      3 1 16

* .. _`CEC-RX-STATUS-OK`:

- ``CEC_RX_STATUS_OK``
- 0x01
- The message was received successfully.
* .. _`CEC-RX-STATUS-TIMEOUT`:

- ``CEC_RX_STATUS_TIMEOUT``
- 0x02
- The reply to an earlier transmitted message timed out.
* .. _`CEC-RX-STATUS-FEATURE-ABORT`:

- ``CEC_RX_STATUS_FEATURE_ABORT``
- 0x04
- The message was received successfully but the reply was
  ``CEC_MSG_FEATURE_ABORT``. This status is only set if this message
  was the reply to an earlier transmitted message.
* .. _`CEC-RX-STATUS-ABORTED`:

- ``CEC_RX_STATUS_ABORTED``
- 0x08
- The wait for a reply to an earlier transmitted message was aborted
  because the HDMI cable was disconnected, the adapter was unconfigured
  or the :ref:`CEC_TRANSMIT <CEC_RECEIVE>` that waited for a
  reply was interrupted.

```

## Return Value

On success 0 is returned, on error -1 and the `errno` variable is set appropriately. The generic error codes are described at the [ref:Generic Error Codes <gen-errors>](#) chapter.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 343); [backlink](#)**

Unknown interpreted text role "ref".

The `ioctlCEC_RECEIVE <CEC_RECEIVE>` can return the following error codes:

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 347); [backlink](#)**

Unknown interpreted text role "ref".

#### EAGAIN

No messages are in the receive queue, and the filehandle is in non-blocking mode.

#### ETIMEDOUT

The timeout was reached while waiting for a message.

#### ERESTARTSYS

The wait for a message was interrupted (e.g. by Ctrl-C).

The `ref`ioctl CEC_TRANSMIT <CEC_TRANSMIT>`` can return the following error codes:

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 359); [backlink](#)**

Unknown interpreted text role "ref".

#### ENOTTY

The `CEC_CAP_TRANSMIT` capability wasn't set, so this `ioctl` is not supported.

#### EPERM

The CEC adapter is not configured, i.e. `ref`ioctl CEC_ADAP_S_LOG_ADDRS <CEC_ADAP_S_LOG_ADDRS>`` has never been called, or `CEC_MSG_FL_RAW` was used from a process that did not have the `CAP_SYS_RAWIO` capability.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 366); [backlink](#)**

Unknown interpreted text role "ref".

#### ENONET

The CEC adapter is not configured, i.e. `ref`ioctl CEC_ADAP_S_LOG_ADDRS <CEC_ADAP_S_LOG_ADDRS>`` was called, but the physical address is invalid so no logical address was claimed. An exception is made in this case for transmits from initiator 0xf ('Unregistered') to destination 0 ('TV'). In that case the transmit will proceed as usual.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 371); [backlink](#)**

Unknown interpreted text role "ref".

#### EBUSY

Another filehandle is in exclusive follower or initiator mode, or the filehandle is in mode `CEC_MODE_NO_INITIATOR`. This is also returned if the transmit queue is full.

#### EINVAL

The contents of struct `c.type`cec_msg`` is invalid.

**System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\userspace-api\media\cec\[linux-master] [Documentation] [userspace-api] [media] [cec] cec-ioc-receive.rst, line 382); [backlink](#)**

Unknown interpreted text role "c.type".

#### ERESTARTSYS

The wait for a successful transmit was interrupted (e.g. by Ctrl-C).