Infrared Remote Controls

Appendix Overview

An Infrared (IR) Cisco Remote Control can be connected to the Cisco IIEC 4650 device so that the end user can control applications and remote playback without touching the screen or using a mouse. This appendix explains how to configure applications to allow use of either a Cisco and non-Cisco remote control.

Topics in this appendix include:

- “Cisco Remote Control IEP-IR-K9”
  - “Battery”
  - “Infrared Sensor”
  - “Remote Control Buttons”
  - “Remote Control Implementation”
  - “global.ir Object”
  - “Usage of global.ir Object”
- “Non-Cisco Remote Controls”
  - “Using the IEM’s IR Configuration Property”
  - “Using the API”

Cisco Remote Control IEP-IR-K9

The Cisco Remote Control IEP-IR-K9 is the remote control model that is supported with the IEC.

Battery

Warning

Lithium Batteries are used in this module. Do not try to charge, discharge, or replace these batteries.
The Cisco remote control is battery-powered. It uses a standard, 3V CR2025 lithium battery, manufactured by a well-known supplier such as Panasonic or Toshiba. The name of the actual manufacturer is etched into the face of the battery.

When the battery loses its charge or when you remove the battery, the remote control will not work until the battery is replaced.

You should always recycle or dispose of the battery in accordance with:

- Its manufacturer guidelines.
- Regulations in your locale for disposal and recycling.

**Note** Remote control settings are not lost when you remove or replace the battery.

To replace the battery in the remote control, follow these steps:

1. Place the remote control on a flat surface, button-side down.
2. To unlock the battery clip and remove it, pinch the locking mechanism and slide the clip out of the remote control.
3. To remove the old battery from the clip, pivot the battery so that it touches only the opening of the clip.
4. Insert a new battery so that positive charge (+) symbols are visible simultaneously on the battery and the clip.
5. Slide the clip back into the remote control.

### Infrared Sensor

The IEC is equipped with an infrared (IR) sensor that receives, recognizes, and reacts to the signals from this Cisco remote control.

The way that you mount an IEC can limit how well it responds to these signals. The mounting method might block the IR sensor.

It is recommended to use an IR extender with the remote control. With the IR extender, the range of the remote control is approximately up to 15 feet (4.57m).

The IR sensor of the extender (or the IR sensor of the IEC if you are not using the extender) must be in the line of sight of the end user. The remote control will not work if the IR sensor is behind the touchscreen or blocked by another peripheral.

**Note** Please ensure that you are using a suitable IR cable that was suggested by your Cisco distributor or importer source.

### Remote Control Buttons

All the buttons can be programmed. The functions listed in the tables below are standard functions for the keys.
### Table E-1  System Control Buttons

<table>
<thead>
<tr>
<th>Function</th>
<th>IR Signal Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>0xff 00 0090</td>
</tr>
<tr>
<td>Input</td>
<td>0xef 10 0090</td>
</tr>
<tr>
<td>Information</td>
<td>0xe0 1f 0090</td>
</tr>
<tr>
<td>Help</td>
<td>0xf6 09 0090</td>
</tr>
</tbody>
</table>

### Table E-2  Playback Control Buttons

<table>
<thead>
<tr>
<th>Function</th>
<th>IR Signal Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Play</td>
<td>0xf9 06 0090</td>
</tr>
<tr>
<td>Pause</td>
<td>0xfd 02 0090</td>
</tr>
<tr>
<td>Stop</td>
<td>0xfc 03 0090</td>
</tr>
<tr>
<td>Rewind</td>
<td>0xfb 04 0090</td>
</tr>
<tr>
<td>Fast forward</td>
<td>0xfa 05 0090</td>
</tr>
<tr>
<td>Live</td>
<td>0xf7 08 0090</td>
</tr>
</tbody>
</table>

**Note**

The Live button is usually used to jump to the next channel or track.

### Table E-3  Audio Control Buttons

<table>
<thead>
<tr>
<th>Function</th>
<th>IR Signal Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume increase</td>
<td>0xbf 40 0090</td>
</tr>
<tr>
<td>Volume decrease</td>
<td>0xbe 41 0090</td>
</tr>
<tr>
<td>Mute</td>
<td>0xfe 01 0090</td>
</tr>
</tbody>
</table>

### Table E-4  Channel Control Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>IR Signal Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Channel down (-)</td>
<td>0xbc 43 0090</td>
</tr>
<tr>
<td>Channel up (+)</td>
<td>0xbd 42 0090</td>
</tr>
<tr>
<td>Last</td>
<td>0xe3 1e 0090</td>
</tr>
<tr>
<td>1</td>
<td>0xf2 0d 0090</td>
</tr>
<tr>
<td>2</td>
<td>0xf1 0e 0090</td>
</tr>
<tr>
<td>3</td>
<td>0xf0 0f 0090</td>
</tr>
<tr>
<td>4</td>
<td>0xe1 1e 0090</td>
</tr>
<tr>
<td>5</td>
<td>0xe8 17 0090</td>
</tr>
</tbody>
</table>
The following is an example of programming for the remote control in an application:

```javascript
if(!filterIR){
    filterIR = true;
    document.getElementById('trace').innerHTML = 'IR pressed key code: "+key+
" +skey+');
    switch(skey){
        case 'play':
            // Code for play
            break;
        case 'up':
            // Code for up
            break;
        case 'left':
            // Code for left
            break;
        case 'ok':
            // Code for ok
            break;
        case 'right':
            // Code for right
            break;
        case 'down':
            // Code for down
            break;
        case 'menu':
            // Code for menu
            break;
        case 'guide':
            // Code for guide
            break;
        case 'page up':
            // Code for page up
            break;
        case 'page down':
            // Code for page down
            break;
        case 'exit':
            // Code for exit
            break;
    }
}
```
Remote Control Implementation

The IR port is active by default. No additional configuration is required.

You will need to embed the global.ir object into your application code in order for your applications to perform the expected action when the end user presses a button on the remote control.

global.ir Object

The global.ir object implements the IR interface. It allows an application to receive signals from the infrared remote control.

The global.ir object code is:

```javascript
interface Ir {
    readonly attribute string lastError;
    List <String> availableControls() const;
    bool setCurrentControl(in string device);
}
```

```javascript
HUD("play");
playVideo();
break;
case "pause":
    HUD("pause");
togglePause();
    break;
case "fastforward":
    HUD("fwd");
    playNextTrack();
    break;
case "rewind":
    HUD("rewind");
    playPrevTrack();
    break;
case "stop":
    HUD("stop");
    stopVideo();
    break;
case "right":
    HUD("right");
    selectNext();
    break;
case "left":
    HUD("left");
    selectPrevious();
    break;
case "okay":
    HUD("play");
    playSelected();
    break;
case "power":
    window.location = "index.html";
    default:
    break;
}
irFilterTimeout = setTimeout(function(){filterIR=false}, 750);
```
signals:
    event(in uint key, in string skey, in string configName) const;
    error(in string err) const;
}

Table E-8  global.ir Object Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lastError</td>
<td>Last error occurred</td>
</tr>
<tr>
<td>availableControls()</td>
<td>Returns the list of the supported remote controls</td>
</tr>
</tbody>
</table>
| setCurrentControl(in string device) | Sets the current remote control to use.  
                                    | The device name must be obtained from availableControls() list.             |
|                            | Leave the device name empty to use browser.ir.configuration. In this case    |
|                            | you should set browser.ir.configuration.enabled to true and browser.ir.configuration to the valid LIRC configuration. |
| event(in uint key, in string skey, in string configName) | Remote control event  
                                    | • The event control code is set to key.                                    |
|                            | • The control name (such as "poweroff", "ch1", "up", etc.) is set to skey. |
|                            | • The configuration name, which is rarely needed, is set to configName.      |

Usage of global.ir Object

The following HTML code contains an example of global.ir usage.

```html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <title>...: global.ir test :..</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  </head>
  <body>
    {  
      margin: 20px;
      background-color: #111111;
      color: #eeeeee;
      font-weight: bold;
      font-family: Arial;
      font-size: 18px;
      color: #eeeeee;
    }
```
var errorId;
var eventId;
var controlId;
var timer;
var remoteInfo;

function init()
{
    errorId = document.getElementById("error");
    eventId = document.getElementById("event");
    controlId = document.getElementById("control");
    remoteInfo = document.getElementById("remoteInfo");

global.ir.error.connect(onError); writeLog("onError() connected to signal 
global.ir.error");
global.ir.event.connect(onEvent); writeLog("onEvent() connected to signal 
global.ir.event");

    try{
        var irconf = global.registry.value("browser.ir.configuration");

        if(irconf=='<binary mimeType="text/plain"><! [CDATA[]]> </binary> ' ||
        irconf==''){
            // set default remote as current.
            //global.ir.setCurrentControl(defaultRemote);
            remoteInfo.innerHTML = "Default (Cisco remote control)");
            writeLog("Default remote control configuration applied.");
        } else{
            global.ir.setCurrentControl();
            remoteInfo.innerHTML = "* user defined remote control *");
            writeLog("User defined remote control configuration applied.");
        }
    } catch(ex){
        writeLog("Exception: "+ex);
    }
}

function deinit()
{
    global.ir.error.disconnect(onError);
    global.ir.event.disconnect(onEvent);
}

function onError(err)
{
    eventId.innerHTML = err;
    writeLog("onError(): "+err);
}

function onEvent(key, skey, config)
{
    eventId.innerHTML = "* + key + ' ' + skey + ' ' + config;
    writeLog("onEvent(): "+key + " " + skey + " " + config);
    clearTimeout(timer);
    timer = setTimeout(function() { eventId.innerHTML = ""; }, 750); 
}
Non-Cisco Remote Controls

This section explains how to customize remote control programming so that non-Cisco remote controls can be used. You can use either the IEM’s IR Configuration property or the global.ir.setCurrentControl() API.

Using the IEM’s IR Configuration Property

To customize remote control programming, you will need to perform the following tasks:

1. Create a new group if the devices that will use the customized programming are not already in a group
2. Add devices to the new group
3. Create an IR configuration policy and upload the remote’s Linux Infrared Remote Control (LIRC) config file from the manufacturer
4. Apply policy to the device group

Create a New Group

If the devices that will use the customized programming are not already in a group, create a new group by following these steps:
Step 1  Click Devices in the left pane.
Step 2  In the Edit menu, click New Group.
Step 3  In the Create New Group dialog box, enter a group name in the Group Name field such as “IRConfig_IECs”.
Step 4  Click Create.
A folder labeled with the group’s name appears in the left and center panes.

Add Devices to the New Group

Add all the devices that will use the customized programming to this new group:

Step 1  Click Devices in the left pane.
Step 2  In the center pane, double-click the new group’s icon.
Step 3  In the Edit menu, click Properties.
Step 4  Click the Member Of tab in the center pane to view a list of devices in the group.
Step 5  Click +.
Step 6  In the Add to Group dialog box, check the devices’ check boxes.
Step 7  Scroll to the bottom of the Add to Group dialog box and click Add.
Step 8  Click Apply.
Step 9  In the Predefined actions menu, click Reboot.
Step 10  Click OK.

Create an IRConfig Policy

You will need a remote configuration file for this task. Contact the manufacturer of your remote control to obtain it.

Follow these steps to create a policy for the IR configuration:

Step 1  Click Policies in the left pane.
Step 2  In the Edit menu, click New Policy.
Step 3  Enter a policy name (for example, “IRConfig”) in the Policy Name field of the Create New Policy dialog box.
Step 4  Click Create. A new policy is created and its icon appears in the center pane.
Step 5  Click the Policy tab within the new policy.
Step 6  Click the browser > ir > configuration property.
Step 7  Click the IR configuration button under the Value column.
Step 8  Click Load file in the IR configuration editor and upload the LIRC config file for your remote control.
Step 10  Click Apply.

Apply the IRConfig Policy to the Device Group

Now apply the policy to the device group:

Step 1  Click Devices in the left pane.
Step 2  In the center pane, click on the new groups icon.
Step 3  In the Edit menu, click Properties.
Step 4  Click the Policies tab.
Step 5  In the Available policies list, choose the IRConfig policy.
Step 6  Click the green arrow.  
        The policy now appears in the Applied policies list.
Step 7  Click Apply.
Step 8  In the Predefined actions menu, click Reboot.
Step 9  Click OK.

Using the API

The IR can also be configured using the global.ir.setCurrentControl() API.

Step 1  Create and upload the LIRC config file for the remote.
Step 2  To enable and use the uploaded config file, call global.ir.setCurrentControl() API from your application. See the “global.ir Object” section above for information on how to call the API.
        The user-defined remote controller should now work.