

Integration with Home Assistant

See the Home Assistant documentation of the [LIRC component](#) to have a more details. If you want that Home Assistant recognize the press of a specific button of your remote, you have to add the button and the remote name in the `.lircrc` file located in `/home/homeassistant/`. The `.lircrc` file is accessible [via samba](#).

See this example of entry in the `.lircrc` file:

[/home/homeassistant/.lircrc](#)

```
begin
  remote = SONY
  button = KEY_1
  prog   = home-assistant
  config = see-this
end
```

Add an entry for every button that you want to be recognized. The button can be associated later to an automation.

The values you set for the field “button” must be the same as in the `lircd.conf` file as well as the “remote” name.

The values you put in the “config” field will be the sensor value in Home Assistant when you press the button.

Restart `lirc` to load the changes.

```
pi@d-diot:~ $ sudo systemctl restart lircd
```

Test your LIRC installation before proceeding by running:

```
pi@d-diot:~ $ sudo -u homeassistant -H -s
homeassistant@d-diot:/home/pi $ ircat home-assistant
```

At every press of `KEY_1` in the SONY remote a line should appear. Switch back to user “pi” and restart Home Assistant

```
homeassistant@d-diot:/home/pi $ exit
pi@d-diot:~ $ sudo systemctl restart home-assistant@homeassistant
```

Now every time you press a button defined in the `.lircrc`, the LIRC component fires `ir_command_received` events on the bus. You can capture the events and respond to them in an automation script. See the [LIRC component](#) documentation to learn more.

Use the RESTful API of lirc_web

If you want to send an IR command directly from Home Assistant, that is useful when you are on a remote connection where the lirc_web interface is not accessible, you can use the built in [RESTful API of lirc_web](#).

To do this you have to configure the [RESTful Command](#) component of Home Assistant. For every remote key add an entry in the configuration.yaml (or in a different file with a split configuration) as in the example below:

</home/homeassistant/.homeassistant/configuration.yaml>

```
# Example configuration.yaml entry
rest_command:
  lirc_web_YOUR_REMOTE_YOUR_KEY:
    url: 'http://localhost:3000/remotes/YOUR_REMOTE/YOUR_KEY'
    method: POST
```

Substitute YOUR_REMOTE and YOUR_KEY according to your needs. To see the list of remote and buttons as configured in lirc_web, go to:

- <http://d-diot.local:3000/remotes.json>

Now you can create a momentary switch for the button with the [template switch component](#), adding the lines under the oled switch to the switches.yaml file:

</home/homeassistant/.homeassistant/switches.yaml>

```
# Template switches
- platform: template
  switches:
    oled:
      friendly_name: "Status"
      value_template: "{{ is_state('binary_sensor.oled_status', 'on') }}"
    turn_on:
      service: shell_command.start_oled
      data:
        entity_id: binary_sensor.oled_status
    turn_off:
      service: shell_command.stop_oled
      data:
        entity_id: binary_sensor.oled_status
  lirc_web_YOUR_REMOTE_YOUR_KEY:
    friendly_name: "Arrow up"
    value_template: "false"
    turn_on:
```

```
service: rest_command.lirc_web_YOUR_REMOTE_YOUR_KEY  
turn_off:  
service: rest_command.lirc_web_YOUR_REMOTE_YOUR_KEY
```

From:
<https://wiki.d-diot.com/> - **d-diot wiki**

Permanent link:
https://wiki.d-diot.com/how_to/lirc/integration_with_home_assistant?rev=1553975093

Last update: **2019/03/30 19:44**

