

# RTC:HokuyoAist User Guide

Geoffrey Biggs  
geoffrey.biggs@aist.go.jp

November 8, 2010

## 1 Introduction

RTC:HokuyoAist is an RT Component for the OpenRTM-aist middleware. It provides a driver for Hokuyo laser range sensors. It wraps the hokuyo\_aist library from Gearbox<sup>1</sup>. It functions with all current models of laser scanner, including the URG-04LX (Classic-URG), UHG-08LX (Hi-URG), UTM-30LX (Top-URG) and UXM-30LX/UXM-30LX-E (Tough-URG).

The laser remains off until the component is activated. The laser is then turned on, unless the component is configured to be in pull mode. On start-up, the laser's internal clock is calibrated to the computer's clock, including the communications delay. Time stamps in the range data are calculated using this calibrated time, so no calibration is necessary by users of the data.

This software is developed at the National Institute of Advanced Industrial Science and Technology. Approval number H22PRO-1194. The development was financially supported by the New Energy and Industrial Technology Development Organisation Project for Strategic Development of Advanced Robotics Elemental Technologies. This software is licensed under the Eclipse Public License -v 1.0 (EPL). See LICENSE.TXT.

## 2 Requirements

RTC:HokuyoAist uses the Gearbox “hokuyo\_aist” library<sup>2</sup> for its functionality. At least version 2.0.0 is required, which can be found in Gearbox release 10.11. A compiled copy of this library is included in the installer for users of Windows.

RTC:HokuyoAist requires the C++ version of OpenRTM-aist-1.0.0.

RTC:HokuyoAist uses the CMake build system<sup>3</sup>. You will need at least version 2.6 to be able to build the component.

RTC:HokuyoAist works on Windows, Linux and Mac OS X. It uses the Gearbox “flexiport” library for communications with the laser. This must be installed. A compiled copy of this library is included in the installer for users of Windows.

## 3 Installation

### 3.1 Binary

Users of Windows can install the component using the binary installer. This will install the component and all its necessary dependencies. It is the recommended method of installation in Windows.

1. Download the installer from the website.

---

<sup>1</sup><http://gearbox.sourceforge.net/>

<sup>2</sup>[http://gearbox.sourceforge.net/group\\_\\_gbx\\_\\_library\\_\\_hokuyo\\_\\_aist.html](http://gearbox.sourceforge.net/group__gbx__library__hokuyo__aist.html)

<sup>3</sup><http://www.cmake.org/>

2. Double-click the executable file to begin installation.
3. Follow the instructions to install the component.
4. You may need to restart your computer for environment variable changes to take effect before using the component.

The component can be launched by double-clicking the `rtc_hokuyoaist_standalone` executable. The `rtc_hokuyoaist` library is available for loading into a manager, using the initialisation function `rtc_init`.

### 3.2 From source

Follow these steps to install RTC:HokuyoAist from source in any operating system:

1. Download the source, either from the repository or a source archive, and extract it somewhere.

```
tar -xvzf rtc_hokuyoaist-2.0.0.tar.gz
```

2. Change to the directory containing the extracted source.

```
cd rtc_hokuyoaist-2.0.0
```

3. Create a directory called “build”:

```
mkdir build
```

4. Change to that directory.

```
cd build
```

5. Run `cmake` or `cmake-gui`.

```
cmake ../
```

6. If no errors occurred, run `make`.

```
make
```

7. Finally, install the component. Ensure the necessary permissions to install into the chosen prefix are available.

```
make install
```

8. The install destination can be changed by executing `ccmake` and changing the variable `CMAKE_INSTALL_PREFIX`.

```
ccmake ../
```

The component is now ready for use. See the next section for instructions on configuring the component.

RTC:HokuyoAist can be launched in stand-alone mode by executing the `rtc_standalone` executable (installed into `${prefix}/bin`). Alternatively, `librtc.so` can be loaded into a manager, using the initialisation function `rtc_init`. This shared object can be found in `${prefix}/lib` or `${prefix}/lib64`.

## 4 Configuration

The available configuration parameters are described in Table 1.

Parameter	Effect
port_opts	Change the options used to open the port to the laser. See flexiport for details.
start_angle	The angle to begin scanning at, in radians. Set this to 0 to begin at the default for a full scan.
end_angle	The angle to stop scanning at, in radians. Set this to 0 to end at the default for a full scan.
cluster_count	The number of readings to group into each cluster. The default is 1.
enable_intensity	When set to true, intensity data will also be sent.
high_sensitivity	Some models feature a high sensitivity mode. Use this to enable it.
pull_mode	Switch to pull mode. You will need to request each scan through the HokuyoAist service interface.
new_data_mode	When set to true, new data will be requested each time. Otherwise, the most recent data will be sent.
error_time	The minimum time between errors, in seconds. If errors occur closer together than this value, the component will transition to the error state. Otherwise, it will attempt to reset the laser and continue.
x, y, z	The laser's position in 3D space.
roll, pitch, yaw	The laser's orientation in 3D space.

Table 1: Available configuration parameters.

Name	Type	Data type	Purpose
ranges	OutPort	RTC::RangeData	Range data scanned by the laser.
intensities	OutPort	RTC::IntensityData	Intensity data scanned by the laser. Only available when intensity data is enabled.
ranger	Service	Ranger	Service port providing the generic RTC::Ranger interface.
		HokuyoAist	Service port providing the HokuyoAist interface, for specialised features not supported by the generic RTC::Ranger interface.

Table 2: Available ports.

## 5 Ports

The ports provided by the component are described in Table 2.

The communications protocol used by the Hokuyo lasers reports error values within the range data using values less than 20. This component filters out these values and replaces them with 0m. You should consider all zero values as unknown when processing sensor data.

## 6 Examples

An example configuration file is provided in the `${prefix}/share/rtc_hokuyo_aist/examples/conf/` directory.

## 7 Changelog

### 7.1 2.0

- Support hokuyo\_aist library v2 API.