

# Ceylan-Oceanic: Enocean facilities in Erlang



**Organisation:** Copyright (C) 2022-2022 Olivier Boudeville

**Contact:** about (dash) oceanic (at) esperide (dot) com

**Creation date:** Wednesday, September 7, 2022

**Lastly updated:** Saturday, September 10, 2022

**Version:** 0.0.1

**Status:** In development

**Dedication:** Users and maintainers of the `Ceylan-Oceanic` library.

**Abstract:** The role of the `Ceylan-Oceanic` library is to provide Erlang-based facilities for the support of the Enocean building automation system.

The latest version of this documentation is to be found at the [official Ceylan-Oceanic website](http://oceanic.esperide.org) (<http://oceanic.esperide.org>).

This documentation is also mirrored [here](#).

## Table of Contents

<b>Overview</b>	<b>3</b>
<b>Testing Ceylan-Oceanic in Two Steps</b>	<b>3</b>
<b>Hardware Prerequisites</b>	<b>3</b>
<b>Operating System Support</b>	<b>3</b>
<b>Software Prerequisites</b>	<b>4</b>
<b>Testing</b>	<b>5</b>
Basic, Direct Testing . . . . .	5
Oceanic Testing . . . . .	5
<b>Enocean Documentation</b>	<b>6</b>
<b>Support</b>	<b>6</b>
<b>Related Projects</b>	<b>6</b>
<b>Please React!</b>	<b>6</b>
<b>Ending Word</b>	<b>6</b>

## Overview

The Ceylan-Oceanic library provides [Erlang](#)-based facilities for the support of the [Enocean](#) building automation system.

Ceylan-Oceanic is a rather autonomous part of the [Ceylan](#) project, and relies on [Myriad](#). Ceylan-Oceanic can be readily built and run on most Unices, including of course GNU/Linux.

The project repository is located [here](#).

At least a basic knowledge of Erlang is expected in order to use Ceylan-Oceanic.

## Testing Ceylan-Oceanic in Two Steps

Now, let's discuss these subjects a bit more in-depth.

## Hardware Prerequisites

In terms of Enocean devices, one needs typically:

- any kind of emitter/sensor, for example a rocker button like [these ones](#)
- a receiver, typically a USB gateway

Popular USB dongles, which often relies on the TCM 310 chip, include the [USB300](#) one (around 37 Euros in France) or the USB310 one (around 50 Euros in France) that we prefer as it features a [SMA connector](#) allowing to connect an external antenna in order to boost emission/reception ranges.

We will rely here on such a configuration.

## Operating System Support

Once the USB dongle is connected (here on an Arch Linux box), `lsusb` tells us that it is detected as:

```
Bus 003 Device 009: ID 0403:6001 Future Technology Devices International, Ltd FT232 S
```

(which applies both to USB300 and USB310)

We will interact with this USB gateway as if it was a serial port.

Rather than having it designated by an obscure, potentially changing name (like `/dev/ttyUSB0`, `/dev/ttyUSB1`, etc.), we prefer assigning it a fixed, well-chosen path, like `/dev/ttyUSBEnOcean`.

For that one may define a suitable udev rule, typically stored in `/etc/udev/rules.d/99-enocean.rules`, whose content can simply be:

```
SUBSYSTEM=="tty", ATTRS{idVendor}=="0403", ATTRS{idProduct}=="6001", SYMLINK+="ttyUSB
```

Following option could be added as well, to set the group of this TTY: `GROUP="dialout"` ou `GROUP="uucp"`, in which case your user shall be in that group (rather than running `sudo chmod 777 /dev/ttyUSB0` each time the device is inserted for example).

One may run `sudo udevadm control --reload-rules && sudo udevadm trigger` to ensure it is taken into account.

Then inserting said dongle should generate log entries that `journalctl -xe` can show, like (timestamps and hostname edited):

```
kernel: usb 3-11: new full-speed USB device number 9 using xhci_hcd
kernel: usb 3-11: New USB device found, idVendor=0403, idProduct=6001, bcdDevice= 6.00
kernel: usb 3-11: New USB device strings: Mfr=1, Product=2, SerialNumber=3
kernel: usb 3-11: Product: FT232R USB UART
kernel: usb 3-11: Manufacturer: FTDI
kernel: usb 3-11: SerialNumber: A600AVJD
mtp-probe[74533]: checking bus 3, device 9: "/sys/devices/pci0000:00/0000:00:14.0/usb3"
kernel: ftdi_sio 3-11:1.0: FTDI USB Serial Device converter detected
kernel: usb 3-11: Detected FT232RL
kernel: usb 3-11: FTDI USB Serial Device converter now attached to ttyUSB0
mtp-probe[74533]: bus: 3, device: 9 was not an MTP device
mtp-probe[74548]: checking bus 3, device 9: "/sys/devices/pci0000:00/0000:00:14.0/usb3"
mtp-probe[74548]: bus: 3, device: 9 was not an MTP device
```

## Software Prerequisites

Ceylan-Oceanic relies on general-purpose services offered by [Ceylan-Myriad](#) (implying of course [Erlang itself](#)), and on an Erlang driver for serial communication.

We chose [erlang-serial](#) for that, and we prefer installing it in user space that way:

```
$ mkdir ~/Software && cd ~/Software
$ git clone https://github.com/tonyg/erlang-serial.git
$ cd erlang-serial
$ make && DESTDIR=. make install
```

Then using `erlang-serial` will be just a matter of adding it to the code path.  
To test it:

```
$ erl -pa $HOME/Software/erlang-serial/erlang/lib/serial-1.1/ebin
Erlang/OTP 25 [erts-13.0] [source] [64-bit] [smp:8:8] [ds:8:8:10] [async-threads:1] [JVM:17.0.10]

Eshell V13.0 (abort with ^G)
1> serial:start().
<0.82.0>
```

Perfect!

One may update the *Erlang-serial* section in our [GNUmakevars.inc](#) to take into account any other convention.

Run, from the root of Oceanic, `make info-serial` to check that `ERLANG_SERIAL_BASE` points indeed to a directory containing serial's `ebin` directory.

## Testing

### Basic, Direct Testing

It is as simple as executing from the command-line:

```
$ od -x < /dev/ttyUSBEnOcean
00000000 0055 0707 7a01 10f6 2e00 96e1 0130 ffff
00000020 ffff 0039 554b 0700 0107 f67a 0000 e12e
```

`stty` can be also used for that.

Incoming data can also be recorded and "replayed" (but this will not activate an EnOcean receiver):

```
$ cat < /dev/ttyUSBEnOcean > my_record.bin
$ cat my_record.bin > /dev/ttyUSBEnOcean
```

### Oceanic Testing

From the root of the Ceylan-Oceanic clone, supposing Myriad and erlang-serial are already available and built:

```
# Ensure that Ceylan-Oceanic is built:
$ make all

$ cd test
$ make oceanic_run
```

```
Running unitary test oceanic_run (third form) from oceanic_test
```

```
--> Testing module oceanic_test.
```

```
Starting the EnOcean test based on the gateway TTY '/dev/ttyUSBEnOcean'.
```

```
[debug] Using TTY '/dev/ttyUSBEnOcean' to connect to EnOcean gateway, corresponding to
```

```
[debug] Stopping serial server <0.84.0>.
```

```
--> Successful end of test.
```

```
(test finished, interpreter halted)
```

```
(command success reported)
```

## Enocean Documentation

- [ETS]: [Enocean Technical Specifications](#)
- [ESP3]: [Enocean Serial Protocol \(ESP3\) - SPECIFICATION](#)

## Support

Bugs, questions, remarks, patches, requests for enhancements, etc. are to be reported to the [project interface](#) (typically [issues](#)) or directly at the email address mentioned at the beginning of this longer document.

## Related Projects

They may be used as sources of inspiration:

- the [Python EnOcean](#) library
- the (Java) [OpenEnocean openHAB binding](#)
- a first [Rust implementation](#)

## Please React!

If you have information more detailed or more recent than those presented in this document, if you noticed errors, neglects or points insufficiently discussed, drop us a line! (for that, follow the [Support](#) guidelines).

## Ending Word

Have fun with Ceylan-Oceanic!

