

(This page is intentionally blank)

Control your Miniature World

OC32

**Device Definitions
(Generic)**

Author: Leon J.A. van Perlo
Version: 2013/03/25
Date: March 27, 2013

Release management

This manual applies to

- Software
 - OC32Config Rel 0.0.2.3d (or later)
- Definitions file
 - OC32Devices Generic 2013/03/25

Reading Guide

This manual contains the description of definitions for devices, relevant to a **French (CH)** theme on your Miniature World. For a full understanding it is necessary to read the OC32 Manual as well.

To be clear: A "device" in this respect means: A part that is connected to and controlled by the OC32, so for example a railway signal, traffic light or turnout-drive.

Device Definitions include:

- The order in which the different connections of your devices have to be connected to the OC32. The First pin of the OC32, used to control the device is [N+0], the next are [N+1], [N+2], etc. In the diagrams usually just [0], [1] is shown to save some space. It is important that the right connection-order is maintained for the Device Definition to work correctly on the device;
- The characteristics by which each pin, used by the device, is driven;
- The "aspect definitions" belonging to the device. The definitions for pin [N+0] form the complete set to control the device by the program Koploper.
When controlling the OC32 by DCC, usually (depending on your digital control system and software) you can address aspects 0 and 1 for each DCC address only. In order to use all aspects of the device, the "aspects" 2 to 11 defined at [N+0] can be addressed indirectly through aspects 0 and 1 of subsequent pins [N+1], [N+2], etc

Device Definitions do **NOT** include:

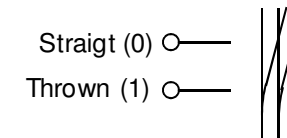
- The type of output (sink driver, source driver, resistor-array) to be used on the OC32. This depends on the electrical properties of the device you are using/connecting. Please consult the manual of the "device" and the OC32 manual. So the Device Definitions only define the order in which outputs have to be connected and the way the device is controlled from software, not the electrical properties.

Should you run into unsolvable situations, please use the forum at <http://www.dinamousers.net>

5 Turnouts

5.1 Pulse drive (coils and motors with or without end-switch)

[2]Gen: Turnout (pulse)



This Device Definition may require finetuning. Activate "Show Details"

Aspect definitions:

Pin [N+0]

Asp 0 = Straight

Asp 1 = Thrown

Finetuning:

Pin = [N+0]

Aspect Configuration section:

Aspect = 0 ("Straight")

Instruction 1 ("On")

Below "T(/60)" enter the time for the desired pulse duration in 1/60s units. 9 (example) therefore results in a pulse of 9s/60 = 150ms.

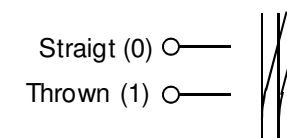
Aspect = 1 ("Thrown")

Instruction 1 ("On")

Below "T(/60)" enter the time for the desired pulse duration in 1/60s units. 9 (example) therefore results in a pulse of 9s/60 = 150ms.

5.2 Permanent drive (coils and motors with end-switch)

[2]Gen: Turnout (permanent)



Pin [N+0]

Asp 0 = Straight

Asp 1 = Thrown

4 Miscellaneous

4.1 On/Off

{1}Gen: On/Off

Pin [N+0]
 Asp 0 = Off
 Asp 1 = On

4.2 Blink

{1}Gen: Blink

This Device Definition may require finetuning. Activate "Show Details"

Aspect definitions:

Pin [N+0]
 Asp 0 = Off
 Asp 1 = Blink

Finetuning:

Pin = [N+0]
 Aspect Configuration section:
 Aspect = 1 ("Blink")
 Instruction 0 ("Invert")
 Below "T(/60)" enter the time for the half period in 1/60s units. 30 (example) therefore is a half-period of 30s/60 = 0,5s and so a blink-frequency of 1x per second.

4.3 Welding Arc (intermittend)

{1}Gen: Welding Arc

Pin [N+0]
 Asp 0 = Off
 Asp 1 = On

Contents

1	Handling Device Definitions by OC32 Config 0.0.2.3d	4
1.1	Several Definition Files	4
1.2	Standard Definition File	4
1.3	(re)Loading Definition Files	4
1.4	Names and Descriptions.....	4
2	Signals.....	5
3	Traffic Lights	5
4	Miscellaneous	6
4.1	On/Off	6
4.2	Blink.....	6
4.3	Welding Arc (intermittend).....	6
5	Turnouts	7
5.1	Pulse drive (coils and motors with or without end-switch)	7
5.2	Permanent drive (coils and motors with end-switch)	7

1 Handling Device Definitions by OC32 Config 0.0.2.3d

1.1 Several Definition Files

Device Definitions for the OC32 are split in several definition-files:

- OC32Devices Generic <datestamp>.def
This file contains a limited set of definitions for general use.
- OC32Devices <CC> <datestamp>.def
Contain contry-specific definitions. <CC> is the ISO Country Code in capitals, so "NL" for the Netherlands and "DE" for Germany. Names for devices also contain the Country Code.
- OC32Devices <Special> <datestamp>.def
Contains definitions for sepcial devices

1.2 Standard Definition File

The definition file you want to use as "default" when starting OC32Config (in many cases OC32Devices Generic ...def will be a logical choice) must be copied to the folder in which OC32Config.exe is installed and (there) renamed to "OC32Devices.def".

If the startup-directory of OC32Config does not contain "OC32Devices.def", OC32Config asks for the initial definition file everytime it starts.

1.3 (re)Loading Definition Files

The user is advised to save the OC32DeviceDefinitions of interest "somewhere" on the PC's harddisk. Save these files on a location where you can find them easily.

In OC32Config (0.0.2.3. or later) a button "Reload DD" can be found, meaning "Reload Device Definitions". By this button a new Device Definition File can be loaded while OC32Config remains active. The first time this function is used you'll need to browse to the location where the definition files are stored. After the first time this happens automatically..

If you need the definitions from more than one dinfinition file and should you find dit inconvenient to switch between definition files, you can (from OC32Config 0.0.2.3d or later) by pressing the Shift" key while clicking "Reload DD" append a definition file to the definitions already loaded.

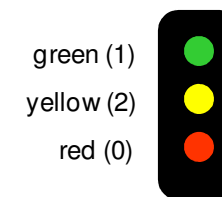
1.4 Names and Descriptions

Names of Devices, Pins and Aspects are in principle in the language of the country they refer to. For Germany e.g. "(5)DE:Hauptsperrsignal". The reason for this choice is that the OC32 is used by customers in many different countries and this is the most original way.

Names in generic and special definition files in principle are in English.

2 Signals

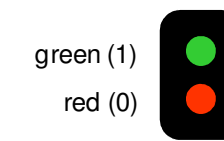
[3]Gen: Red/Green/Yellow



Pin [N+0]
Asp 0 = Red
Asp 1 = Green
Asp 2 = Yellow

Pin [N+1]
Asp 0 = (R) Yellow

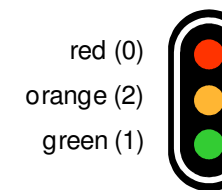
[2]Gen: Red/Green



Pin [N+0]
Asp 0 = Red
Asp 1 = Green

3 Traffic Lights

[3]Gen: Traffic Light

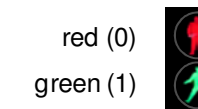


Pin [N+0]
Asp 0 = Red
Asp 1 = Green
Asp 2 = Orange
Asp 3 = Orange Blink
Asp 4 = Red+Orange

Pin [N+1]
Asp 0 = (R) Orange
Asp 1 = (R) Orange Blink

Pin [N+2]
Asp 0 = (R) Red + Orange

[3]Gen: Pedestrian Light



Pin [N+0]
Asp 0 = Red
Asp 1 = Green
Asp 2 = Green Blink
Asp 3 = Dark

Pin [N+1]
Asp 0 = (R) Green Blink
Asp 1 = (R) Dark