

# OC32/NG

## Manual

## Release management

This manual applies to

- PCB/module
  - OC32 Rev10

©2017 This document, or any information contained herein, may not be copied or distributed, in whole or in parts, in whatever form, without the explicit written approval of the original author. The making of copies and prints by users of the OC32 module for their own use is allowed.

## Preface / Reading Guide

This manual describes the OC32/NG, which is an upgraded version of the original OC32. Although there are major differences between the OC32 and the OC32/NG, the /NG does not offer new functions, since all functions are made in software and the OC32 and the OC32/NG run 100% identical software. The OC32/NG however integrates the options previously offered by the OC32/DS32 combination and makes the system much easier to install and easier to adapt.

This manual describes only the “hardware” OC32 Rev10 (OC32/NG). Since the firmware and software is 100% identical to the firmware for the OC32 (Rev00-Rev04), software capabilities are described in a separate manual.

In this manual we will use both names OC32 and OC32/NG. When we write ‘OC32’, we mean ‘any version of the OC32’. Where we write ‘OC32/NG’ we specifically mean the /NG version.

The OC32 is a product with many possibilities. These extensive capabilities make the module very attractive: in fact you can use the OC32 to control (almost) any type of accessory on your miniature world (so basically everything except the trains and cars themselves). Without the need to buy other specific electronics, the OC32 can do it all. This versatility has a downside: Beginners, electronically less savvy users, face the risk of losing the overview at first. Therefore, this guide attempts to structure information with the above in mind.

Each chapter covers a specific sub-topic, eg "Power Supply", "Connecting Devices" or "Network Connection" and usually begins with information that is relevant for each user. As the chapter progresses subjects are touched which may require more specialist knowledge or more effort to understand. As a reading-aid you find a colored bar in the margin, and the black&white spectators will note that the bars have a different width:

Green	Novice: With these sections you should be able to get the basic functions working. It offers no extensive choices, clever savings or complex combinations.
Blue	Advanced level; Requires basic knowledge of electronics, some user-level experience with PC software, some logic thinking or a combination hereof. It requires you to make some choices and therefore you should be able to judge the benefits and drawbacks in your specific situation. In principle everyone should be able to practice this, however it may not be wise for everyone to start with this immediately.
Orange	Expert level: Requires reasonable to good knowledge of electronics, logic thinking capabilities, some programming skills or a combination of these. What is described in these sections can lead to damage to the electronics or other devices if it is not done correctly. So practice only if you fully understand what you are doing.

Should you consider yourself a “novice” and electronically limited skilled, or just looking for the easiest start, skip the blue and orange marked sections at first. If the basics work you can always start the more advanced levels later.

The OC32 is supported through the Dinamo Users Portal. You find the portal at <http://www.dinamousers.net>

The portal contains a “wiki” with quite some additional information, such as:

- Answers to Frequently Asked Questions
- Software and firmware updates
- A forum you can use for advice and to get your questions answered.

We urgently request you to use our support channels in the above order before personally contacting VPEB.

**Enjoy!**

## Contents

1	OC32 - Introduction.....	7
1.1	A different approach .....	7
1.2	Functions.....	7
1.3	OC32 I/O Pins.....	8
1.4	Communication.....	8
1.5	DCC.....	8
1.6	ETI.....	9
1.7	SAP.....	9
1.8	Physical properties .....	9
2	Practical Notes .....	10
2.1	Opening and closing the enclosure.....	10
2.2	Using the connectors.....	10
3	Overview .....	12
3.1	Interfaces and functions.....	12
3.2	LED indicators.....	13
4	Power Supplies.....	14
4.1	General.....	14
4.2	Some words about power supplies: .....	14
4.3	GND or reference voltage.....	15
4.4	Connecting power to the OC32.....	15
4.4.1	The standard method: through K1.....	15
4.4.2	Use of separate power supplies for OC32 and connected devices .....	15
4.4.3	Providing power to the OC32/NG via K5A or K5B.....	16
4.5	5V.....	16
4.6	Use of model railroad- and other transformers.....	17
4.6.1	Use of a locomotive transformer.....	17
4.6.2	Rectification and smoothing of an AC voltage .....	17
5	Communicating with the OC32.....	19
5.1	Ways of communication.....	19
5.2	Addressing of multiple OC32 modules (serial communication) .....	20
5.2.1	Addressing (normal).....	20
5.2.2	Addressing (extended).....	21
5.3	RS485 communication .....	21
5.3.1	RS485 general explanation.....	21
5.3.2	RS485 on the OC32/NG.....	21
5.3.3	RS485 Termination.....	22
5.3.4	Connecting to a PC via RS485.....	22
5.3.5	Connecting multiple OC32/NG's to a PC – Option 1 .....	23
5.3.6	Connecting multiple OC32/NG's to a PC – Option 2 .....	24
5.3.7	Connecting multiple OC32/NG's – Other options.....	25
5.3.8	The OC32/NG in a Dinamo(MCC) system .....	25
5.3.9	The installation of a “real” RS485 network .....	25
5.4	RS232 communication .....	25
5.5	DCC control.....	26
5.6	Additional Options.....	27
5.6.1	TTL Level serial input.....	27
5.6.2	Dual RS485 interface .....	28
6	Connecting the I/O Pins .....	30
6.1	Selecting the electrical characteristics .....	30
6.1.1	Sink Drivers 500mA (TBD62083).....	31
6.1.2	Source Drivers 500mA (TBD62783).....	31
6.1.3	Sink and Source Drivers (TBD62083 + TBD62783).....	32
6.1.4	Resistor Array (5V outputs) .....	32
6.1.5	OC32-ADM/SI (4,8A power sink-driver).....	33
6.1.6	OC32-ADM/SO (4,8A power source-driver).....	33

---

6.1.7	OC32-ADM/MX (4,8A multiplexer) .....	33
6.1.8	OC32-ADM/FH (4,8A Full H-bridge) .....	33
6.1.9	Identifying the different ADM's .....	34
6.2	Multiple power-supplies, different voltages.....	35
6.2.1	Multiple power-supplies using Sink-drivers .....	35
6.2.2	Multiple power-supplies using Source-drivers.....	35
6.2.3	High power Voltage.....	35
6.3	Connecting the I/O Pins .....	36
6.4	Example-connections.....	37
6.4.1	LED's with common anode (+) .....	37
6.4.2	LED's with common cathode (-) .....	37
6.4.3	Incandescent Bulbs.....	37
6.4.4	LED's antiparallel.....	38
6.4.5	LED's on barrier bars .....	38
6.4.6	Decouplers.....	39
6.4.7	Turnout-solenoids .....	39
6.4.8	Turnout-solenoids (multiplexed) .....	40
6.4.9	Relays.....	40
6.4.10	Motors (unidirectional) .....	41
6.4.11	Motors (bidirectionall) .....	41
6.4.12	Servo Motors.....	41
6.4.13	Pushbutton or switch (input) .....	42
6.4.14	Input from another control system (input) .....	42
7	Serial Accessory Port .....	43
8	External Events .....	44
8.1	Introduction .....	44
8.2	Resistor array.....	44
8.3	Optocoupler .....	45
9	Installation of the U485 .....	46

(This page is intentionally left blank)

## 1 OC32 - Introduction

### 1.1 A different approach

The OC32 is an electronic module to control accessories in your Miniature World. The module has 32 I/O Pins who are universally usable.

The major difference with “traditional electronics” is that in the traditional approach specific electronics is used for each device. For instance there are signal-decoders for signals, and even different types per signalling system, decoders for turnouts, illumination controllers for controlling lights in buildings and streets, etc.

With the OC32 a different approach is taken. The module is so versatile that (almost) any device you find on a miniature world, from a simple light to a complex railway crossing, can be controlled by the OC32. Therefore when applying the OC32 the necessary electronics is not calculated per function, but calculated per square meters. Depending on the number of controllable items per area the OC32's are placed at “strategic locations”. Every item can be connected to the nearest OC32. By means of configuration it is determined how each device is controlled and from what system, e.g. a train-control system, car-control, day/night simulation or even completely autonomous by the OC32.

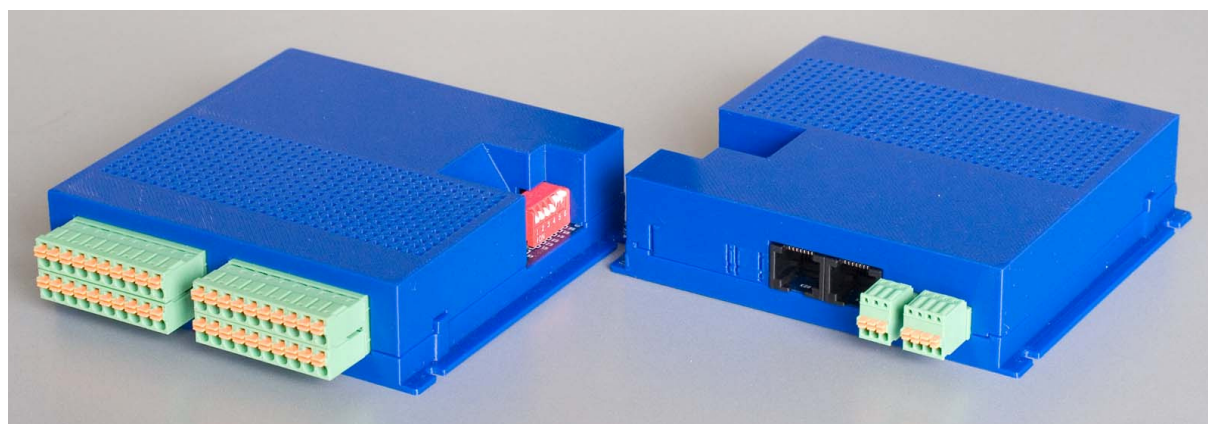


Fig 1: OC32/NG

### 1.2 Functions

The OC32 software has extensive configuration possibilities. Those who have some basic knowledge of programming techniques can adapt the OC32 completely to their personal preferences. Those who do not (yet) have this knowledge choose the predefined configurations. With a few simple mouseclicks a group of outputs can be configured for e.g. a Dutch signal, German signal, a traffic light or railway crossing.

The OC32 offers a great number of random-functions. With these functions fixed patterns can be avoided or interrupted and it is possible to generate surprising effects. Some examples are:

- Welding arc imitation, only one LED needed (e.g. to be used in your locomotive shed);
- Simulations of gas-lanterns with irregular lighting;
- Simulation of fluroscent lamps with traditional starters;
- Lighting of buildings with different or a slightly different switching pattern every night;
- Movement e.g. by means of a servomotor with random variations, e.g for a digging machine or crane;
- ...

### 1.3 OC32 I/O Pins

The OC32 I/O Pins are used to connect the devices in your miniature world. The OC32 I/O Pins need to be equipped with 'drivers' that fit the electrical characteristics of the type of device which is connected, such as:

- Incandescent lamps
- LEDs, common anode (plus) or common cathode (minus)
- Relays
- DC motors, also bidirectional (electronic reverse)
- Servomotors
- Memory wire
- Solenoids for e.g. turnouts and decouplers (up to 4.8A peak-current)
- Pushbutton, switch or other contact

When you buy the OC32, standard drivers are installed, but you may need to change the drivers, depending on your specific application. Changing drivers can easily be done by you as a user.

We will cover the drivers in much more detail later in this manual

### 1.4 Communication

The standard version of the OC32/NG is equipped with 2 serial interfaces: An RS485 interface and an RS232 interface. Via these interfaces the OC32 can be configured and controlled. It can be connected to a PC, a Dinamo or Dinamo/MCC system. A single connection supports up to 96 OC32 modules. Both serial interfaces can be used simultaneously.

The RS232 interface is unidirectional. This means you cannot use this interface to read data from the OC32, nor can you verify the OC32 configuration through this interface. The RS232 interface is mainly there for compatibility with the OM32 and for situations where you want to control OC32's with a standard PC COM-port.

The RS485 interface is bidirectional. It offers full control capabilities. Using RS485 you can bridge a distance up to 1.200 meters!  
RS485 is not what you find on a standard PC. Therefore VPEB designed the U485. This extremely compact module simply turns any USB port into an RS485 interface.



Fig 2: U485

### 1.5 DCC

The OC32 can be ordered with a DCC input<sup>1</sup>, by which the module can be controlled from any DCC compatible digital system. The OC32 can then function as a very flexible Basic DCC Accessory Decoder and as an Extended DCC Accessory Decoder. Unfortunately there are few DCC command stations today that transmit Extended ADCC Accessory Packets. The amount of basic and extended DCC addresses and the basic and extended DCC start addresses used can be configured by software.

The OC32 cannot be configured through DCC. The possibilities are extremely large and configuring through DCC would become very complex to the user. So configuration needs to be done through one of the serial interfaces.

<sup>1</sup> The DCC Interface can also be added afterwards by the user. All components needed are 'through hole' (so not SMD). You need to solder to the PCB however.



## 1.6 ETI

The OC32 can be equipped with 4 additional, galvanically separated Event Trigger Inputs (optocouplers). This enables the OC32 to react on external events. Which actions are taken on any external event is completely user-configurable. This offers the additional possibility to use the OC32 as a self-supporting “scenery controller”

## 1.7 SAP

The OC32 features an additional serial output-port (Serial Accessory Port), by which special equipment can be controlled. One can think of sound-modules to generate environmental-sound or sounds related to the devices the OC32 controls. The advantage of this serial interface is that it won't cost you any of the 32 I/O Pins. The additional serial output is available on any OC32 revision, so also the older modules, when firmware 3.0 or later is installed.

If the SAP option is used, the OC32 cannot be controlled anymore via the serial RS232 interface and therefore the RS485 or DCC interface needs to be used for this.

## 1.8 Physical properties

The OC32/NG is supplied as an assembled printed circuit board with or without enclosure.

The physical dimensions are:

- 104mm x 100mm x 20mm (l x w x h, without enclosure, without connectors)
- 136mm<sup>2</sup> x 100mm x 20mm (l x w x h, without enclosure, with connectors)
- 104mm x 113mm x 27mm (l x w x h, with enclosure, without connectors)
- 136mm<sup>1</sup> x 113mm x 27mm (l x w x h, with enclosure, with connectors)

---

<sup>2</sup> Allow for at least an additional 30mm to facilitate wiring in the connectors

## 2 Practical Notes

### 2.1 Opening and closing the enclosure

If you purchased the “boxed” version or the version with enclosure, it is good to know how to open and close the enclosure before reading the next sections.

To open the enclosure grab the upper half of the enclosure between thumb and a finger at both sides, just behind the two clamps (small bumps), as indicated in figure 3a below. Squeeze the top part of the casing slightly together and pull upwards. The upper half should flip open as in figure 3b. Once open at approx 30 degrees you can remove the top half entirely.

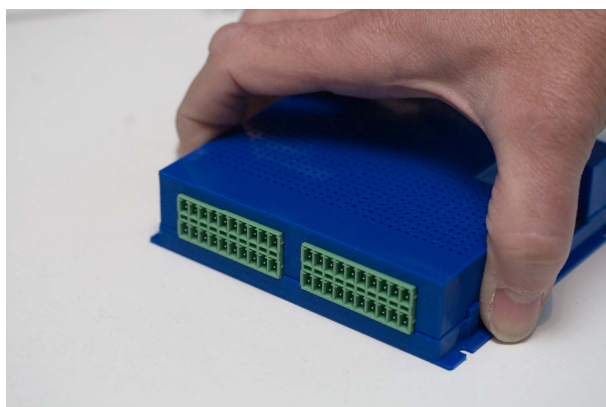


Fig 3a

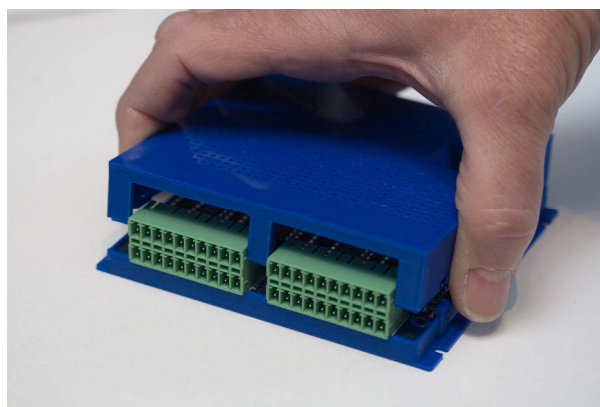


Fig 3b

Opening the enclosure

To close the enclosure, first insert the upper half in about a 30 degree angle into the triangular notches as per figure 4. Carefully check both corners are inserted correctly. Now gently close the upper half. The casing should slide just next to the DIPswitch. No substantial force should be necessary until the casing hits the two clamps at both sides near the other end. To close completely, push the upper half now between the two clamps. No excessive force should be necessary. If required, press both sides towards each other as you did when opening the case.



Fig 4: Closing the enclosure

### 2.2 Using the connectors

The connectors supplied with your OC32/NG are spring-loaded types. This means you can use solid wire, stranded wire and pre-tinned wire and in all these cases will have a solid contact, provided the wire is correctly inserted. No tooling is required except a wire-stripping tool. A small screwdriver may be handy.

K1, K2, K3 and K4 are 2,5mm spring loaded connectors.  
Accepted wire size = 0,08mm<sup>2</sup> – 0,5mm<sup>2</sup>

K5A and K5B are 3,81mm spring loaded connectors.  
Accepted wire size = 0,08mm<sup>2</sup> – 1,5mm<sup>2</sup>

**When inserting thin wire into the K5 connectors (0,20mm<sup>2</sup> or less), pay attention that the wire is centered and inserted straight in the connector. Otherwise the wire may not always be grabbed entirely by the spring.**

To insert wires in either type of connector:

- Strip insulation off the end of the wire for about 10mm.
- In case of stranded wire: twist the strands together.
- Put the connector on a solid surface with the orange clips facing upwards.
- Push-in the orange clip of the pole you want to insert the wire in. You can do this with the tip of your finger or nail, but a small screwdriver or other similar tool may be handy.
- Gently insert the wire until you feel it touches the end. Use no force.
- Release the orange clip.
- Gently pull the wire to check it's grabbed correctly

If you find it difficult to push in the orange clips without the connector flipping over, you may want to do that while the connector is inserted in the OC32/NG module. Be sure that the module is powered off when you do and don't push too hard, otherwise you may bend the PCB part of the connector. This risk is specifically there on the upper half of the 10 pole connectors that go in K5A and K5B.

When inserting the top connector in PCB part K5A or K5B when you have a module without enclosure, do not push very hard, because in extreme cases you may bend the connector on the OC32/NG. While inserting, support the back of the connector to relief excessive strain. If you have the module with enclosure the risk is minimal, since the enclosure provides mechanical support for these connectors.

## 3 Overview

### 3.1 Interfaces and functions

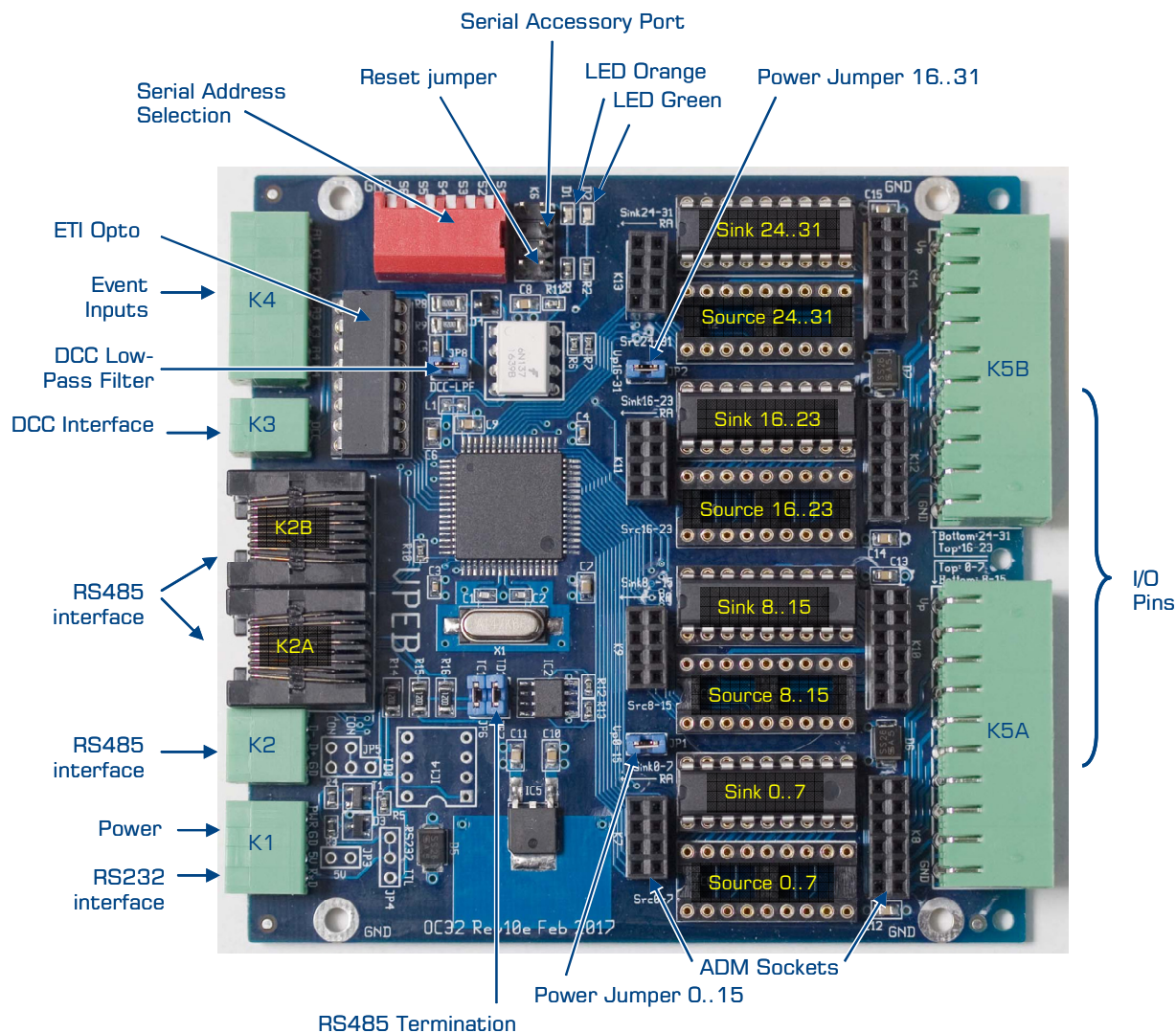


Fig 5: OC32/NG overview interfaces and functions

On the OC32/NG the following interfaces and functions can be found:

#### Interfaces:

- I/O Pins: Two 20-pole connectors (K5A, K5B) accommodate four 10-pin plugs for 32 I/O Pins and power supplies/GND;
- RS485 interface, two RJ45 sockets (K2A, K2B) and one 3-pole socket (K2);
- DCC interface (K3) (optional);
- Event input interface (K4) (optional);
- Four pole connector for power supply and RS232 interface (K1);
- Serial Accessory Port: The connection to control special serial devices;

#### On the PCB:

- 4 pairs of IC-sockets for driver-ICs. Each group of 8 outputs has one socket for a Source Driver and one for a Sink Driver;
- 4 pairs of sockets for Add-on Driver Modules (ADM's). ADM's can be used instead of driver-ICs to increase output power and for additional functions;
- Indicator LEDs, orange and green;
- DIPswitch: A dipswitch with six positions to select the address of the module;

- Reset jumper. Only used to start the bootloader;
- RS485 Termination jumpers;
- Power jumpers;
- DCC Low-pass jumper (only with DCC option)

### 3.2 LED indicators

The functions of the LEDs on the print (green and orange) depend on the firmware. Refer to the OC32 firmware manual for the latest details. Standard function can be changed by configuration by the user.

With firmware 3.0.0 the standard functions are:

#### Starting up:

- Orange: is lit during the starting sequence of the OC32 (if the power is switched on). This takes about 0.25 to 0.5 seconds

#### Normal operation:

- Green: Flashes regularly 1 second intermittent to show that the module is active and the processor is operating normally;
- Orange: Flashes shortly when the OC32 receives a correctly addressed message (Rs232, RS485 or DCC).

#### Bootloader:

When the bootloader is active (see firmware manual) both LEDs are lit continuously.



## 4 Power Supplies

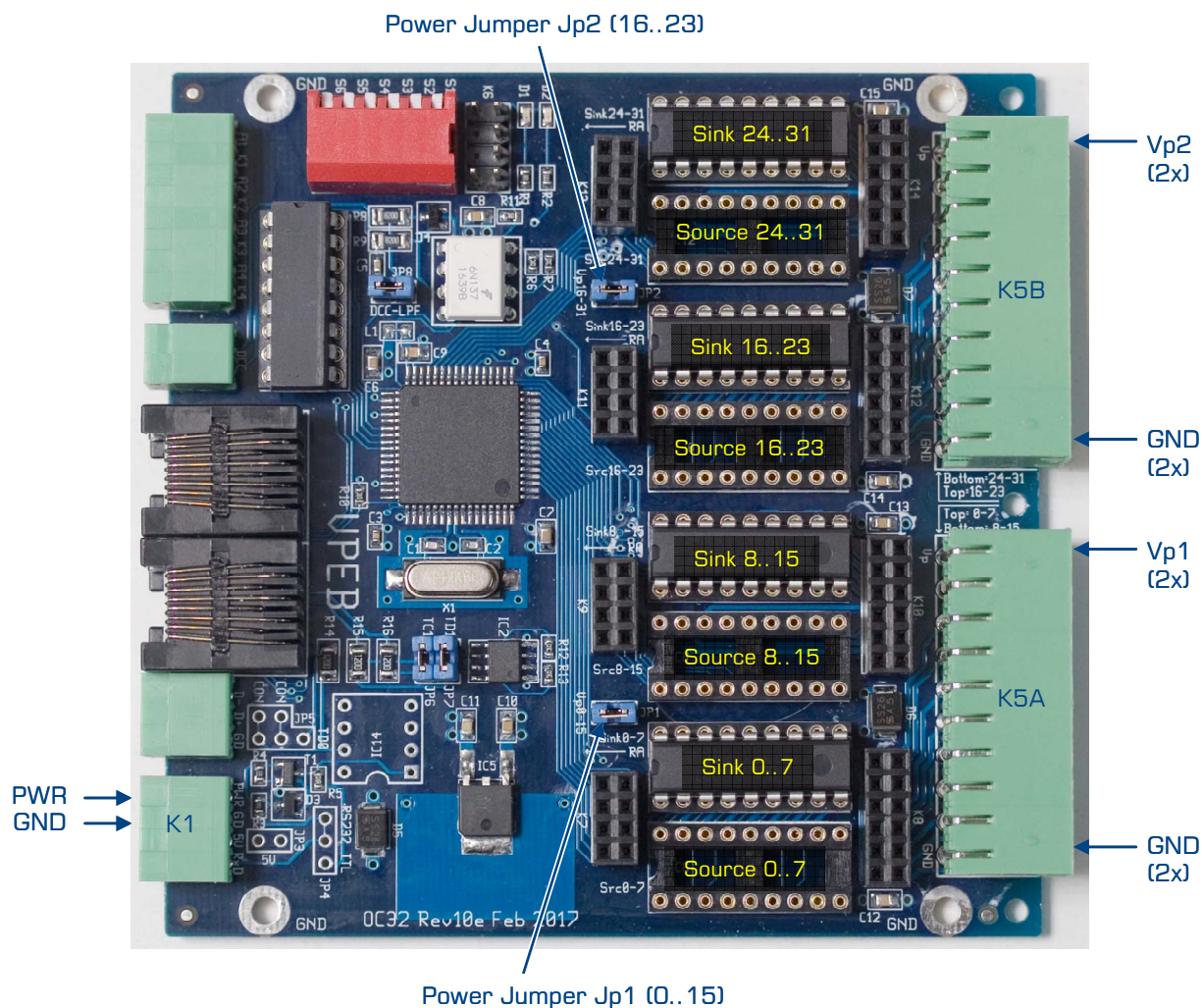


Fig 6: Connecting Power Supplies to the OC32/NG

### 4.1 General

Both the OC32 itself and the devices that the OC32 controls (e.g. signals, motors, relay's) obviously need a power supply to operate. In many cases you can use the same power supply for the OC32 and the devices it controls.

Note that **any** power supply delivered to the OC32, be it for the OC32 itself or for the controlled devices, must be a **positive DC** Voltage.



**Attention: Connecting an AC voltage or reversing the + and - connections of a DC power supply will inevitably result in a defect in the module that is beyond guarantee. Be sure therefore about the kind of power supply you connect the OC32 to.**

### 4.2 Some words about power supplies:

We often get the question if a model railroad transformer can be used for the OC32. Model railroad transformers rarely provide a proper DC voltage. You can use them conveniently, but with a some consideration and simple modification. More about that later on.

For less than €20,- already you can buy a properly stabilized, configurable, switching and therefore economical DC power supply that can provide 2-3 Amps. This is more current than an average model railroad transformer supplies, so think twice if it is worth the effort and

risk to re-use equipment designed for other purposes, especially if your electronic skills are limited.

2 Amps is a good start since the current cannot do too much harm if you make a mistake. It may not be enough to power all the devices in your miniature world, especially if your ambitions are virtually unlimited. In that case you can always buy a more powerful power supply later on and use the 2Amp supply to service your OC32's. More on multiple power supplies later on, but the message for now is: Unless you already know exactly what you need/want, a simple and standard 2Amp DC power supply is probably the best option to start.

### 4.3 GND or reference voltage

We assume that you have a connection point in your miniature world that we can call OV, Ground (GND) or reference voltage. It might be that this "reference voltage" cannot always be found easily, but in a layout controlled by a PC, this reference voltage often is the "GND" of the communications port that your PC uses to control the layout.

The OV/GND (as stated above) of your miniature world has to be connected to the GND connection of the OC32. All points in figure 6 that are labeled "GND" are interconnected on the module itself.

If your skill level on electronics is insufficient to find the reference point, then choose for a separate power supply to power the OC32('s). In that case the OC32's form a separate subsystem within your miniature world and you won't have to worry about the issue above.

### 4.4 Connecting power to the OC32

The power supply for your OC32 has to be a DC power supply with a Voltage level between 7V and 20V (preferably 15V max). The power needs to be smoothed but not necessarily stabilized.

#### 4.4.1 The standard method: through K1

Connect the minus-pole of your power supply to the connection of K1 marked GND/GD. Connect the plus-pole of your power supply to the connection of K1 labeled PWR (figure 6)

If you leave the "Power" jumpers Jp1 and Jp2 on the module (as delivered ex-factory) the supply voltage as provided on K1-PWR will become available on the 20-pin connectors K5A and K5B on connections Vp1 and Vp2 (plus) and GND (minus), so you can feed your connected devices with this. For detailed pin-out, see figure 24.

#### 4.4.2 Use of separate power supplies for OC32 and connected devices

If the item to control requires a high supply voltage or may cause a lot of interference (e.g. turnout coils with endstop), it can be wise to separate the power supply of the OC32 from the one for the devices to control. You will then power the devices with e.g. a voltage of 18V and the OC32 itself with e.g. 9V. This has 2 advantages:

- Possible interference signals and noise from connected devices will not interact that easily with the electronics. So therefore **in some cases** it can improve stability.
- The voltage conversion on the OC32 will waste less energy and therefore there will be less dissipation. While reducing the voltage from 9V to 5V less energy will be dissipated than when the OC32 needs to reduce 18V to 5V

On the OC32/NG you'll find two "Power jumpers" JP1 and JP2. These jumpers forward the PWR power you supply to K1 to the Pin connectors. JP1 connects PWR to Vp1 on K5A, JP2 connects PWR to Vp2 on K5B. If you remove JP1 or JP2 the respective Vp will be separated

from the central PWR supplied to K1, allowing you to use separate power supplies for your devices and the OC32 itself. Per OC32/NG you can even have two different power supplies for devices and a separate one for the OC32 itself.

#### 4.4.3 Providing power to the OC32/NG via K5A or K5B

The power you supply to K1 will be forwarded to Vp1 and Vp2 on the K5 connectors. You can also do it the other way around: While the power jumper JP1 is installed, power supply connected to Vp1 on K5A will be forwarded to PWR and power the internal electronics of your OC32/NG. Similar, while the power jumper JP2 is installed, power supply connected to Vp2 on K5B will be forwarded to PWR. In these cases you can leave K1 unconnected. You will have all OC32 connections at one side, with the exception of the connections for communication.

Note that, while both JP1 and JP2 are installed, Vp1 and Vp2 are automatically connected. The schematic is essentially as indicated in figure 7.

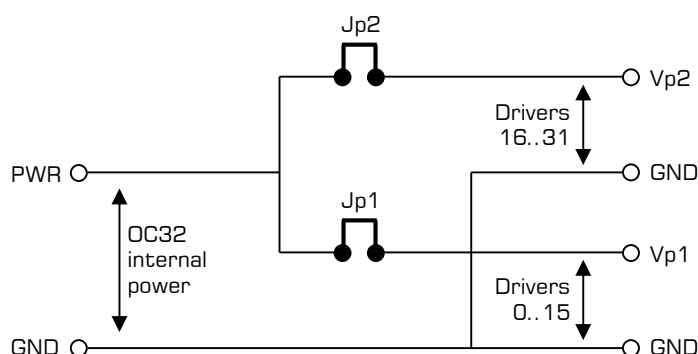


Fig 7: Power distribution schematics

## 4.5 5V

Internally, the OC32 operates on a supply voltage of 5V. The OC32 obtains this from the supply voltage PWR, delivered on K1 or delivered on Vp1/Vp2 through jumpers Jp1/Jp2. The OC32 itself ensures an accurate stabilization, so you don't have to worry about that yourself.

In rare instances, you may want to have access to the internal 5V system of the OC32.

If you make use of so-called 5V outputs (see further on), then the power for these outputs is obtained from the internal 5V. If you use the outputs to their load limit **and at the same time**, the available supply voltage on PWR is high (e.g. higher than 15V), then the stabilizer on the OC32 can become quite hot. The most obvious solution in this case is to split Vp and PWR and feed the OC32 from e.g. a 7-9V power supply. **Alternatively** you may provide an external 5V directly, especially if it is already present in close proximity of the OC32, very well stabilized and it has an efficient regulator.

Also in some cases you may need a 5V connection, e.g. if you want to drive LED's, connected with their cathode to an output configured for 5V operation (see paragraph ...). You then can make your own 5V supply, or derive it from the OC32.

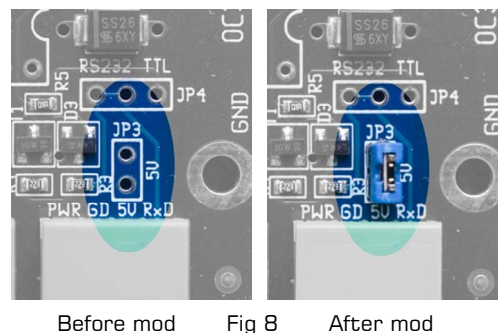
On the OC32/NG, the 5V is no longer available by default on K1, as was the case with the previous models. The reason is that this 5V is directly connected to the OC32 CPU. Any Voltage over 5.1V or under 0V applied to this interface will cause severe damage to the OC32. Since the 5V interface is hardly ever needed, it has been removed to protect the OC32 from unintended errors.



If you need the 5V interface, it can be made available by a simple modification:

- If not already present, solder a dual-pin header (pitch 2,54mm) onto position JP3
- Place a jumper on JP3 to activate the 5V interface.

If you want to provide your own 5V supply, then connect its minus to pin 2 of K1 (GND) and its plus to pin 3 of K1 (5V).



Furthermore the following requirements need to be satisfied:

- JP1 and JP2 **may NOT** be installed.
- The pin PWR on K1 **may NOT** be connected.
- Ideally the power voltage should be between 5.0V and 5.1V and be stabilized very well at a short distance of the OC32. An improperly stabilized power supply or a long wire length will lead to instability of the OC32. A negative voltage or a voltage higher than 5.5V will inevitably lead to a serious defect in your OC32 and costly repair!



## 4.6 Use of model railroad- and other transformers

### 4.6.1 Use of a locomotive transformer

If you have a locomotive transformer for a **DC system**, you can use it as a power source for your OC32 and connected items. Take into account that in most cases a locomotive transformer does not provide a real DC voltage, but a pulsing voltage. For a correct operation it is highly advisable and in most cases even mandatory to smooth the voltage pulses with a capacitor. This can be done by connecting the capacitor to the supply voltage in parallel, preferably as close to the transformer as possible. The value of the capacitor depends on the total current that you will need from the supply. As a guideline, use about 2500 $\mu$ F per Ampere. Obviously the specified voltage of the capacitor needs to be higher than the voltage you will actually provide to the OC32.



**Furthermore it is very advisable to mechanically block the direction switch on your transformer, to prevent an accidental reversal of polarity from happening!**

### 4.6.2 Rectification and smoothing of an AC voltage

If you don't have a DC voltage but an AC voltage only, then you can transform that into a DC voltage with 3 simple components. How this is achieved can be found in figure 9.



**Attention: Since the rectified voltage will be connected to your OC32('s), the OC32('s) will be connected to your digital system or to your PC and in the latter case your PC to your digital system, it is essential that the AC voltage source (the transformer itself) is not connected in any other way to your digital system. It might be feasible, but you have to know exactly what you are doing, and how your digital system is constructed. If you don't know that, then see to it that the secondary side of the transformer (the low voltage side) is not connected in any other way. Ignoring this warning can result in damage of your OC32, your digital system or both!**

So, if you have a separate AC transformer or a transformer with a separate secondary side that you can use, you can use the schematics as shown in figure 9 for rectification and smoothing. To the left you connect the output of your AC transformer. The value of the capacitor depends on the total current you need. As a guideline, use about 2500 $\mu$ F per Ampere. The diode bridge needs to be able to cope with at least the voltage and current that you are going to obtain and the voltage that the capacitor needs to cope with has to

be at least the voltage level PWR. It makes sense to incorporate a fuse to take care of the risk of fire during a short circuit!

Take into account that rectification and smoothing will increase the voltage a bit. In most cases the voltage PWR will be about 1.1 times the nominal AC voltage of the transformer.

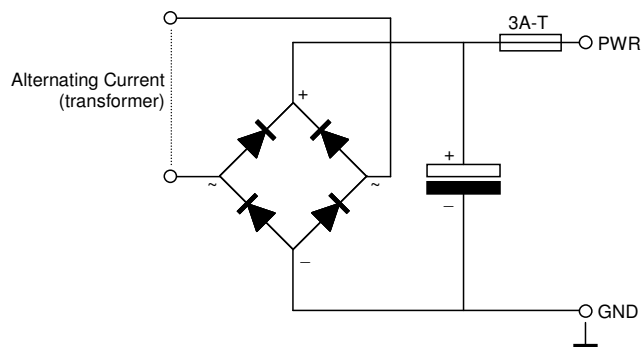


Fig 9: Rectification and smoothing AC supply

## 5 Communicating with the OC32

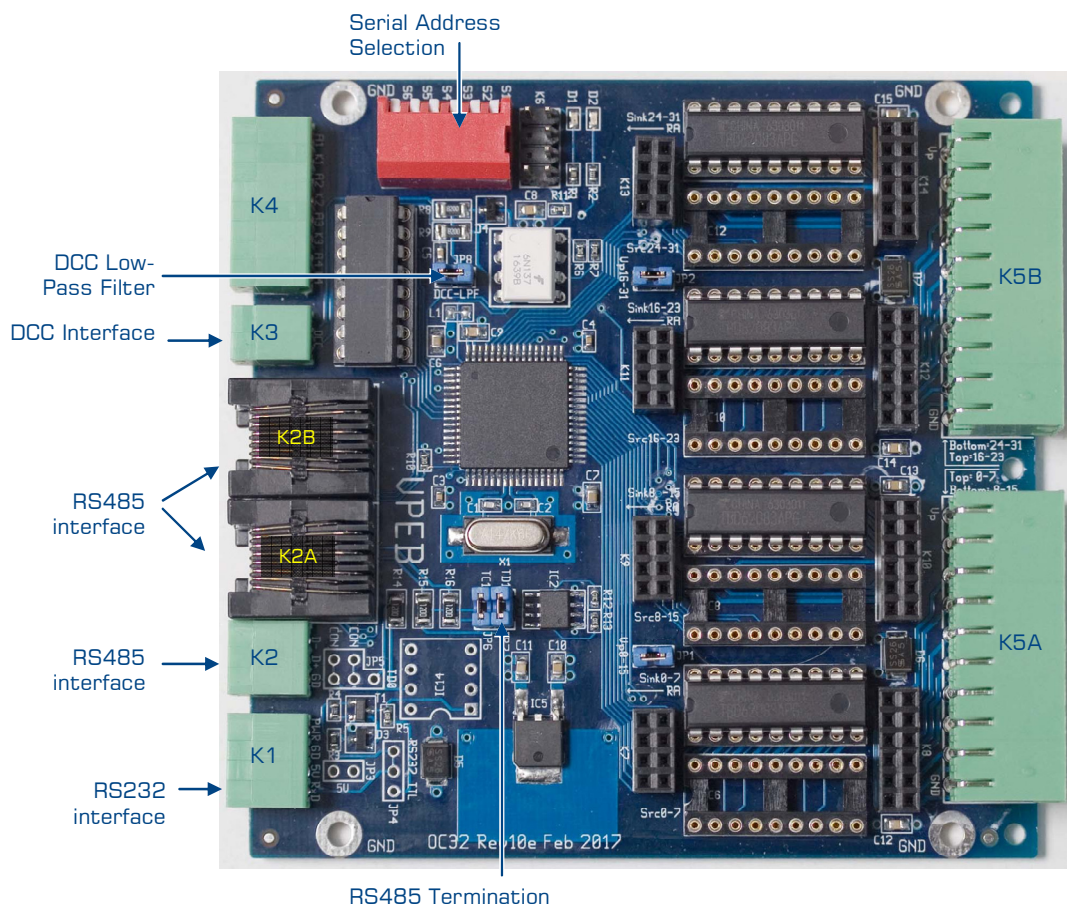


Fig 10: OC32 communication connections

### 5.1 Ways of communication

The OC32/NG has 3 communication-interfaces:

- an RS485 communication channel
- an RS232 communication channel
- a DCC interface

The three interfaces are independent and can be used simultaneously, as long as the distinct channels don't receive conflicting commands.

The RS232 channel and the DCC interface can (seen from the OC32) only receive information. The RS485 channel is bidirectional.

In the communication with the OC32, 3 levels can be distinguished:

- **Operational:** This involves commands that have to do with the situation in which your miniature world is in normal operation. Examples are commands for setting a signal or switch in a specific state. All communication interfaces can be used for this level.
- **Configuration:** This involves commands for the configuration of the OC32, for example whether an output is to be used for a servo or LED and for setting the parameters for servo control. For this level both the RS232 and the RS485 interface can be used. The RS485 interface has the advantage here because with this level it is also possible to read the settings of the OC32 and therefore gives you the possibility of verification.

- Firmware-update: With this you can provide the OC32 with new firmware. It can only be done via the RS485 interface.

The OC32 can be controlled in 4 different ways:

- Via a Dinamo or Dinamo/MCC system. In most cases you will use the RS485 interface.
- Straight from a PC with the appropriate software. In this case you preferably use the RS485 interface or, if you happen to have a COM-port and no U485, the RS232 interface.
- By a “digital system” via the DCC protocol. Note, with this method you can only send operational commands.
- Autonomously or by pushbuttons/switches.

As described above the communication channels are available simultaneously. You can use this feature for example when you control your OC32 in a Dinamo system via RS485, or in a digital system via DCC, but require an additional channel to give operational commands to the OC32 from a separate PC program to control day/night simulation.

## 5.2 Addressing of multiple OC32 modules (serial communication)

Using the serial interfaces RS485 or RS232, you can connect multiple OC32 modules to the same communication bus. In order to determine with which module you are communicating, each module has an address. With normal addressing you can address up to 16 modules, with extended addressing up to 96 modules.

Note that the ‘serial address’ is shared between the RS485 and RS232 channels, so if you use both channels simultaneously, the OC32 will have the same address on both interfaces.

### 5.2.1 Addressing (normal)

Each OC32 receives a unique address (0..15). The address to which the OC32 reacts is set by dip-switches. It doesn’t matter whether the communication is TTL, RS232, RS485 or a combination of those, nor whether your central system is Dinamo or your OC32’s connected straight to your PC.

Table 1 below shows which settings of the dip-switch correspond to which address. For the record: this is the standard numbering starting from 0. If your software starts numbering from 1 onwards, you have to add 1 to each address.

Address:	0	1	2	3	4	5	6	7
SW1	ON	OFF	ON	OFF	ON	OFF	ON	OFF
SW2	ON	ON	OFF	OFF	ON	ON	OFF	OFF
SW3	ON	ON	ON	ON	OFF	OFF	OFF	OFF
SW4	ON	ON	ON	ON	ON	ON	ON	ON

Address:	8	9	10	11	12	13	14	15
SW1	ON	OFF	ON	OFF	ON	OFF	ON	OFF
SW2	ON	ON	OFF	OFF	ON	ON	OFF	OFF
SW3	ON	ON	ON	ON	OFF	OFF	OFF	OFF
SW4	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF

Table 1: Address setting of the OC32

## 5.2.2 Addressing (extended)

Extended addressing offers the possibility to connect in practice up to 96 modules<sup>3</sup> in a serial bus. Note that extended addressing shall also be supported by your software. When extended addressing used, the address is no longer selected by the DIPswitches, but instead by a configuration setting in the OC32 flash memory. The DIPswitches on all your modules are set in identical positions and determine the channel number by which communication takes place. The actual module address is configured by OC32Config in the module. The procedure is described in the OC32 firmware manual.

## 5.3 RS485 communication

### 5.3.1 RS485 general explanation

RS485 is a serial bus for the transmission of signals over long distances. When properly installed, you can reach distances up to 1200 meters. Despite the fact that you will need some ambition to encounter this kind of distances on a model railroad at home, RS485 is a handy protocol because it makes it possible to create a reliable communication between multiple pieces of equipment.

When we talk about a “bus”, we mean that there is a single continuous cable, without branches, to which a “module” can be connected at random points. So the cable runs past every module that needs to communicate via the bus. With RS485 the cable consists of 2 wires that are twisted with each other (“twisted pair”). At both ends the cable needs to be terminated with a resistor of 120Ω.

Furthermore it is important that the modules that are communicating have a “common reference voltage”. Basically, they need to be connected to the same ground or the same common power supply. In principle, an RS485 bus contains with the twisted pair also a third wire for the reference voltage, but if your modules are already connected to the same power supply, you can skip the third wire.

Only between PC and miniature world it is advisable to install “the third wire”, if it is not already present.

In principle you require only one twisted pair, plus, if required, an extra wire for the reference voltage. Most cables that are for sale contain multiple pairs. You can for example conveniently use UTP-LAN cable (cable used for computer networks), nowadays commonly for sale in every DIY/hardware store, either with solid core or with stranded core. The latter is slightly more flexible and easier to use. UTP LAN cable contains 4 wire pairs. With RS485 you use only 1 pair (it doesn't matter which one). A wire from any other pair can be used as third wire, if necessary.

At short distances (up to about 20 meters) it hardly matters what kind of cable you use. It doesn't even have to be twisted. Therefore you can just use two insulated wires that you twist together, plus a third wire, if required. For longer distances a proper cable is not only more reliable, but also simply more convenient.

RS485 is offered by default by the Dinamo RM-U controller, the RM-C controller and the UCCI/E controller. Information about this can be found in the documentation of the corresponding modules.

### 5.3.2 RS485 on the OC32/NG

RS485 is available on the OC32/NG on 3 sockets. These sockets are all interconnected, thus from a functional perspective, it does not make any difference which one you use.

---

<sup>3</sup> In theory up to 1536 modules



The RJ45 sockets are convenient if you use standard cables. You can use any standard, straight RJ45 network cable you can find in a computer store. The 3-pole (green) wire end-connector is convenient when you need to connect to something that has not an RJ45 RS485 connector or when you want to use your own cable.

**Note:** There is no standard pin-out defined for RS485 on an RJ45 connector, so 'the standard' is only guaranteed to work between VPEB products.

### 5.3.3 RS485 Termination

We receive quite some questions about 'the RS485 terminators'. Apparently it is hard to comprehend, while in essence the idea is very simple.

RS485 is one continuous wire-pair (the 'bus'). There may, in principle, be no branches. On 'the wire' you can connect devices. So 'the wire' runs from module 1 to module 2 to module 3 to ... to module N. It is important that 'the wire' is 'terminated' at both ends, so **the first and last module only**. The OC32/NG provides terminator-jumpers so you can activate and de-activate the termination function on the module.

Note that there are two termination jumpers: TC1 and TD1. Always activate or de-activate both. See figure 11.

See the whole setup as a laundry-wire. You have to fix (terminate) the wire at both ends, otherwise it will fall, and you can clip your shirts, socks and knickers at any point to the wire.

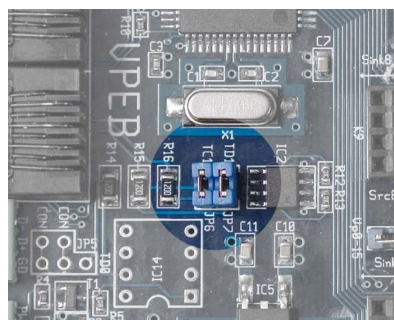


Fig 11: Termination Jumpers

### 5.3.4 Connecting to a PC via RS485

If you want to connect the OC32 to a PC, the best way to do is to use a U485 converter. That is a highly compact USB-RS485 converter especially developed for the OC32. The U485 is hardly more expensive than a standard USB-RS232 converter.

In its simplest form, the RS485 bus implementation consists of 2 modules with a cable (twisted pair) in between: at one end a U485 and at the other end an OC32. The RS485 is to be connected to pins 1,2 and 3 (labeled D+, D- en GND) of K2 (the 3-pin connector). One wire of the twisted pair is connected to D+, the other wire to D-, a third wire to GND. Because RS485 is polarity sensitive, you may not swap D+ en D-, so the wire that is connected to D- has to be connected to D- at the other side as well. The same counts for D+.

All connectors, both the one on the OC32 and on the U485 have the same pin layout, so pin 1 connects to 1, pin 2 to 2 and, if applicable, pin 3 to 3

The whole will look like the one below (figure 12):

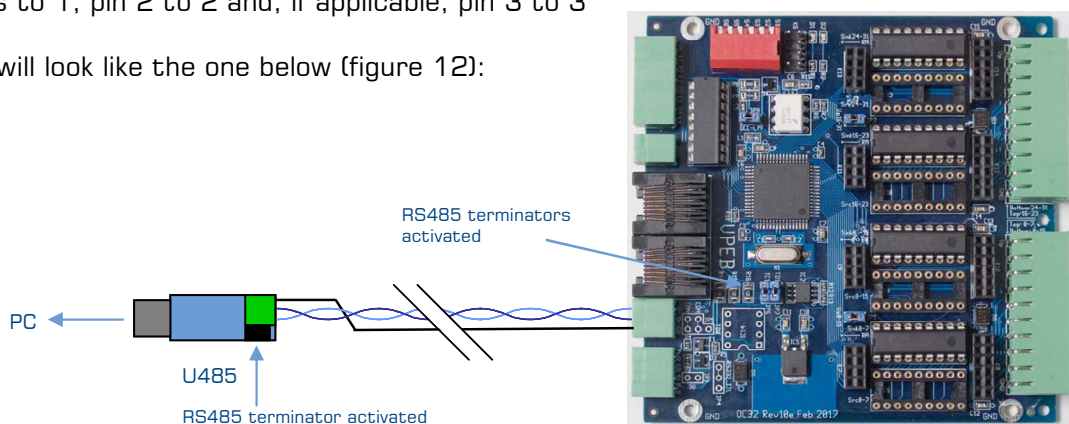


Fig 12: PC to single OC32/NG

### 5.3.5 Connecting multiple OC32/NG's to a PC – Option 1

The easiest way to connect multiple modules is to use RJ45 cables. Since the U485 has no RJ45 connector you need to connect the U485 to the first OC32/NG module as in figure 13.

Now plug an RJ45 cable in one of the RJ45 sockets of that first OC32/NG and plug the other end in an RJ45 socket of the second OC32/NG. If you have more modules, plug another cable in the free socket of the second OC32/NG and plug the other end in module 3. Continue with this process to the last module.

**Note:**

**Only the U485 and the last OC32/NG module shall have the terminator active.**

**On the OC32/NG to which you have connected the U485, only one RJ45 may be used.**

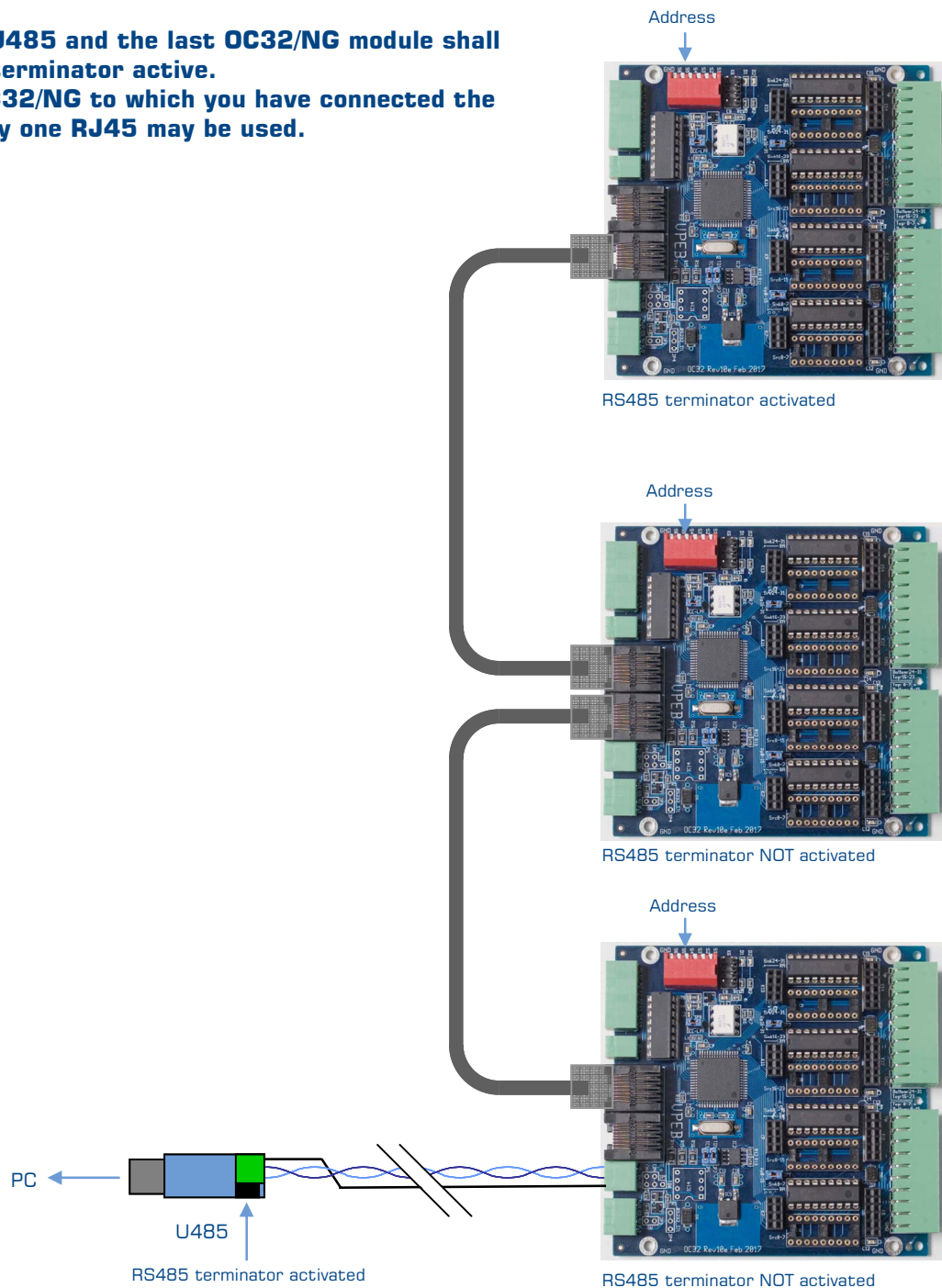


Fig 13: Connecting multiple OC32/NG's with RJ45 cables

### 5.3.6 Connecting multiple OC32/NG's to a PC – Option 2

If you cannot or don't want to use RJ45 cables, you can use the green 3-pole wire-end connector and your own cable/wires instead. Be aware of the following:

- The "cable" has to run past all modules, you must not create branches;
- All connectors have the same pin-layout, so pin 1 is connected to pin 1, pin 2 is connected to pin 2 and wherever necessary, pin 3 connected to pin 3;

The 3-pin connectors make chaining relatively easy. Strip about 2 cm from the insulation of a wire pair near the intermediate module that you want to connect. Do that for both cable ends (so the one from the previous and the one to the next module). Twist the stripped ends of the wires with the same color of the two cables firmly together. Cut the blank parts to about 1 cm and plug this in pin 1 of the connector. Perform this procedure for the other wires of the two cables and plug the result in pin 2 of the connector. We assume that all OC32 are on the same GND and that therefore the connection of pin 3 is only required between the U485 and one of the modules.

The U485 counts as a 'just another module'. It doesn't matter where in the chain the U485 is placed. Most practical is at the beginning or end but can also be placed somewhere in between. In figure 14 we deliberately put the U485 somewhere in the middle to show this. Note that in that case you may **NOT** activate the terminator on the U485 but **MUST** activate the ones on the OC32's at both ends of the chain.

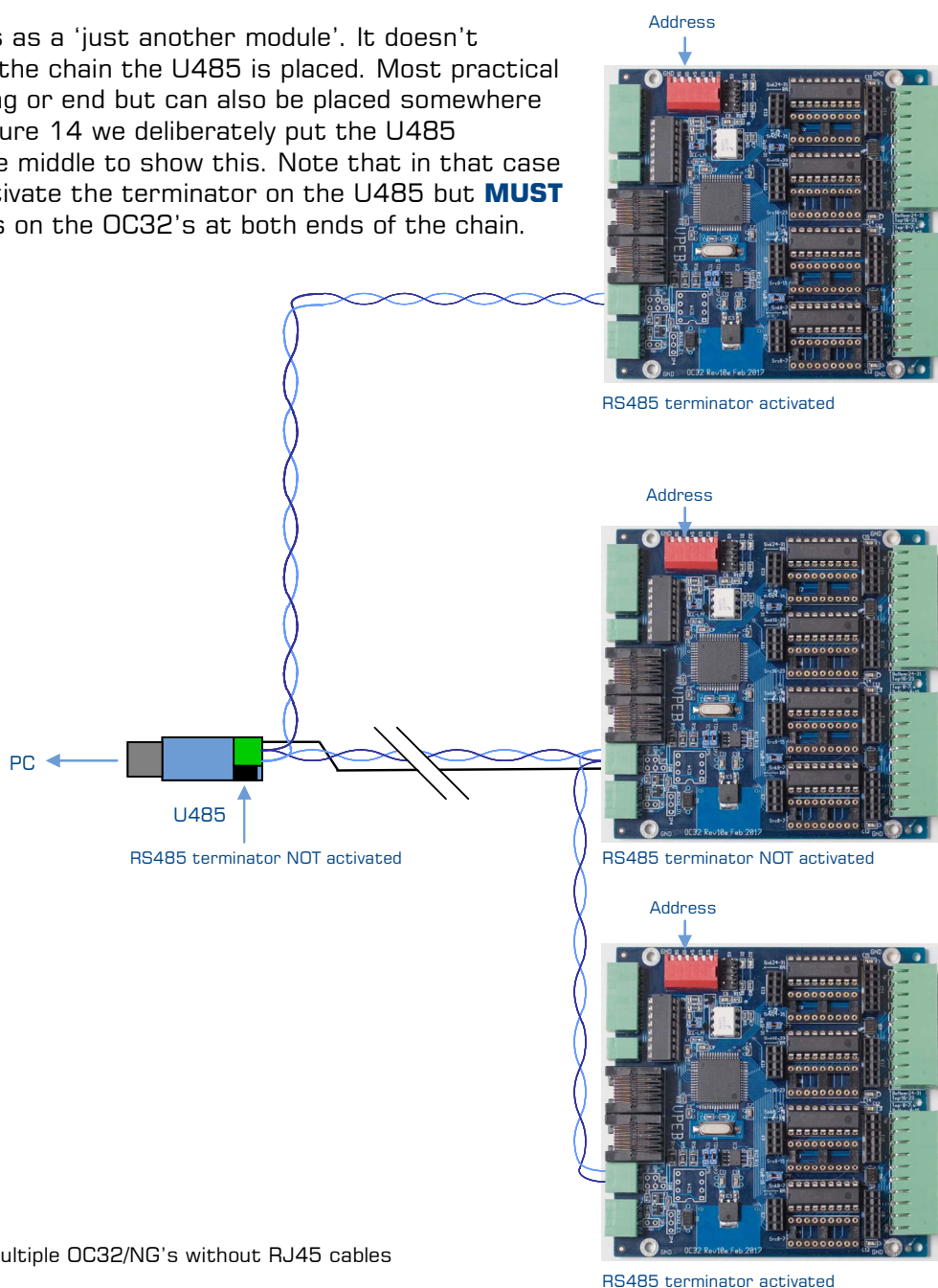


Fig 14: Connecting multiple OC32/NG's without RJ45 cables



### 5.3.7 Connecting multiple OC32/NG's – Other options

Of course you can combine options 1 and 2 as described above. Since the two RJ45's and the 3-pole connector are electrically connected, effectively every OC32/NG can be used as a 'media converter' from RJ45 to individual wires. Especially is you use a combination of OC32's and OC32/NG's this may be of value.

### 5.3.8 The OC32/NG in a Dinamo/(MCC) system

If you deploy the OC32/NG in a Dinamo or Dinamo/MCC system and you use RJ45 cabling, the OC32/NG is nothing more or less than another Dinamo module in the RS485 bus. So use one of the RJ45's to connect to the previous module and the other RJ45 to connect to the next. See the Dinamo P&P Manual for further details.

**If the OC32/NG is the first or last module in a Dinamo RS485 network, be sure that you activate both terminators (data and clock).**

### 5.3.9 The installation of a “real” RS485 network

The above describes a simple way of connecting a couple of modules to RS485. In some cases it might be handy to install a “real” RS485 network. The “bus” implementation is with respect to topology not that handy in all cases and sometimes you want to be able to connect and disconnect modules at multiple positions in a flexible way. For the record: a “real” network is not “better” or “more reliable” than a bus, but it can be more handy and flexible. There are many ways of creating such a network. To prevent this manual from becoming too bulky and because the matter also counts in a Dinamo system, we will describe it in a separate document.

## 5.4 RS232 communication

In theory RS232 is usable with distances of up to tot 15 meters, but in reality you often reach reliable communication over distances of up to 50 meters. RS232 is the standard protocol that is offered by the COM-port of a PC or a simulated COM-port by means of a USB-converter..

Today, the use of RS232 for communication with the OC32 is not recommended. The use of RS485 is preferred by far. However, should you need an additional channel from e.g. a PC to give additional commands to control functions for a day/night simulation, RS232 may be a valuable option.

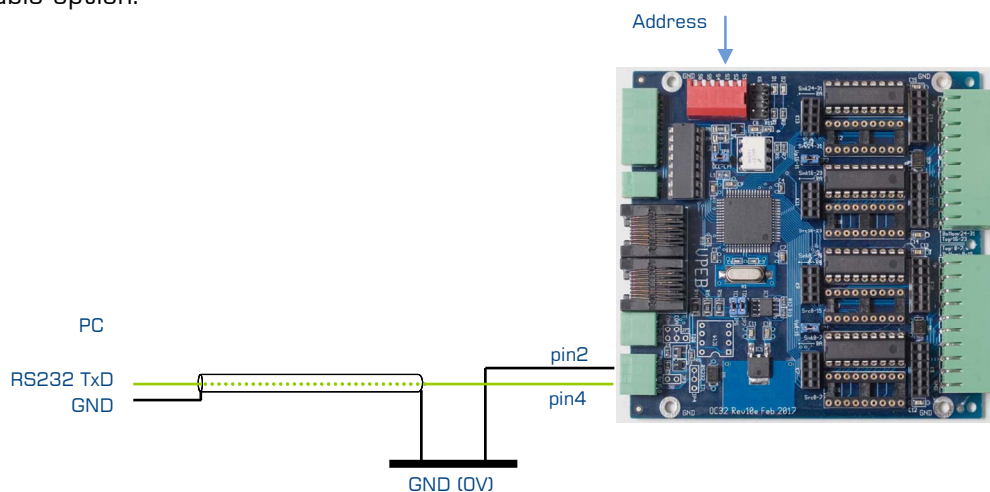


Fig 15: RS232 communication

If you connect the OC32 directly to the COM-port of a PC, then you need to obtain the RS232 TxD signal from the COM-port. You can buy a serial cable for this, but it is cheaper and most likely easier to manufacture one yourself. The case is that you only require 2 wires from the COM-port of your PC.

Buy a 9 pin subD female connector, possibly with a cover, and a piece of signal cable with at least 2 wires. You can very conveniently use a piece of cable with shielding and conducting core, a sort of very thin coax cable that is normally used for connecting audio equipment. Solder one wire to pin 5 of the subD connector (this will be the GND) and the other wire to pin 3 (this will be the TxD). If you use a shielded cable with conducting core, you solder the shield to pin 5 and the core to pin 3.

To the other side of the cable, you connect the GND wire (or shield) to the GND of your layout. The TxD signal wire you connect to pin 4 of K1 on the OC32 (green wire in fig 16).

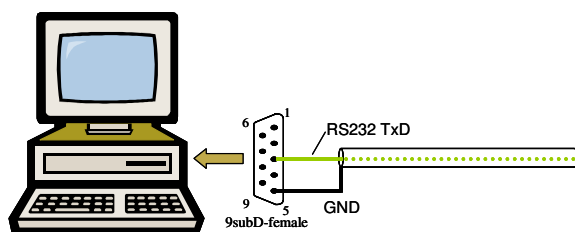


Fig 16: Connecting RS232 communication to the PC

If you are using a USB-RS232 converter, then the pin layout of its 9-pin connector should be identical to that of a normal PC COM-port and the whole game will work as described above. The only difference is that there is a USB cable between your COM-port and the PC. In most cases you have to install drivers as well to make the USB-serial converter work. Have a look at the manual of your USB-serial converter for this.

If you have multiple OC32 modules on an RS232 bus, then you can connect them in parallel. Just interconnect the GND (pin 2) and data (pin 4) of all modules. You can do that in a bus or star configuration. Of course you can also interconnect PWR (pin 1), if all modules share the same power supply.

Note that the fan-out of a standard com-port, and especially the com port on a USB-Serial converter, is limited and therefore there may be a limitation to the number of modules you can connect this way. Up to 10 units should normally work fine. If you want to connect more modules than your com-port can drive, you need to use a signal amplifier.

## 5.5 DCC control

For operational control, the OC32 can receive and process DCC signals, as long as the OC32 module is provided with a DCC interface

DCC is a 2-wire signal. Connect the DCC signal of your digital system to the DCC interface of the OC32. DCC is polarity insensitive. This means that it doesn't matter whether you swap the 2 wires of the DCC interface.

How to obtain a DCC signal from your digital system can be found in the documentation of your digital system. It might be that there are multiple DCC connections on your digital system. Just keep the following in mind:

- If possible, use a connection that is not directly connected to the track. If it does, short circuits and interference might influence the operation of accessory decoders like the OC32 is.

- Use a connection on which the data packets for Accessory decoders are generated. In principle that is any connection but there might be exceptions with your digital system.
- The load of an OC32 on the DCC interface is about 10..15 mA. It is that small that in practice you seldom have to take this extra load into account.

With DCC a large number of modules (decoders) can be connected to the digital system. To be able to control these decoders, DCC has an addressing method as well. The OC32 uses up to 128 consecutive DCC accessory decoder addresses. The DCC start-address and the amount of addresses used is set by means of software configuration (see OC32 firmware manual), so **NOT** with dip-switches.

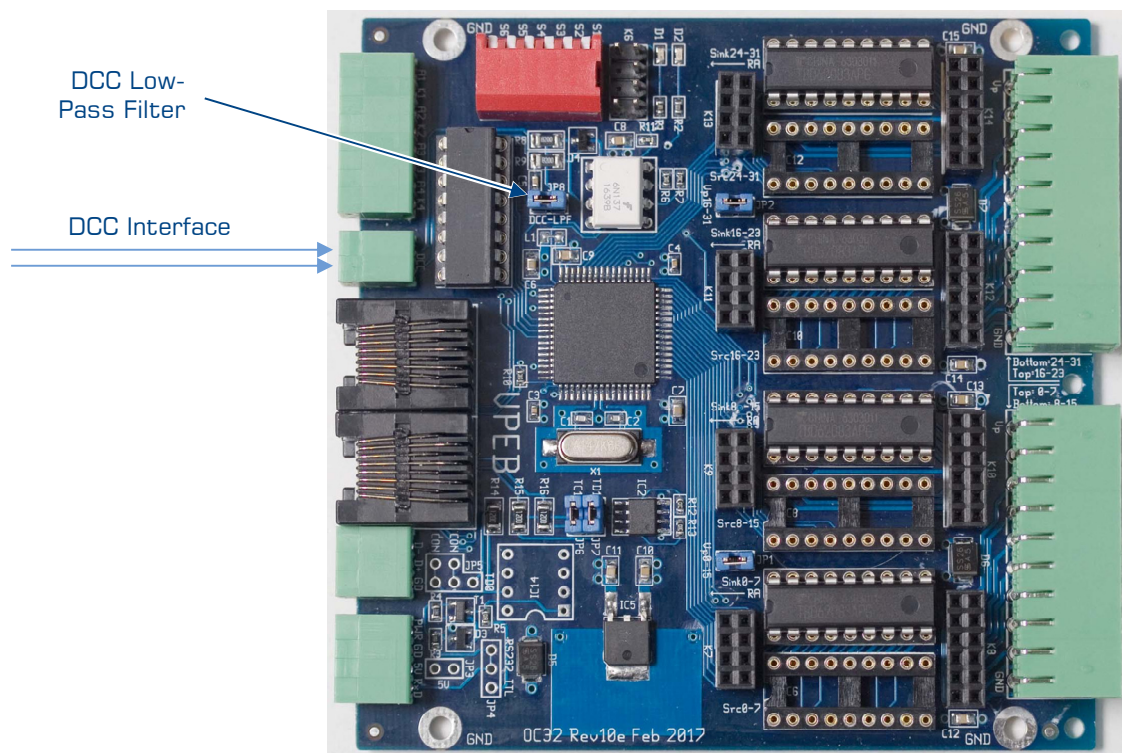


Fig 17: The DCC interface

The OC32/NG has both hardware and software filters to mitigate the effect of electrical interference, voltage spikes, etc. Some users claim that the hardware low-pass filter on the OC32 disturbs signal reception. We have found this is the case when an asymmetrical signal is applied (so actually when the signal is not DCC compliant). If these problems are encountered, the DCC Low-Pass filter on the OC32/NG can be disabled. As depicted in figure 17, you'll find a jumper JP8 (DCC-LPF). When the jumper is installed the Low-Pass filter is enabled (standard), when the jumper is removed the Low-Pass filter is disabled (not recommended unless you experience problems with the LPF enabled).

## 5.6 Additional Options

The OC32/NG has some additional options that are not available on a standard unit. If you want to use these options, you have to make a small modification to the OC32/NG. The reason for this is that these options are very rarely used and we don't want to bother the average user with yet another option to configure.

### 5.6.1 TTL Level serial input

The OC32 (original version) and the OM32 had the possibility to select TTL level input or RS232 level input for the serial interface. The OC32/NG has two serial interfaces: RS485

(channel 1) and RS232 (channel 0). You can make a modification after which you can select TTL or RS232 level for channel 0 as was the case with the 'old' OC32.

We won't explain TTL as technology in this manual. Essentially we consider it obsolete and if you don't know what to use it for, you won't need it anyway.

Required components:

- 3-pin header 2,54mm
- Jumper 2,54mm

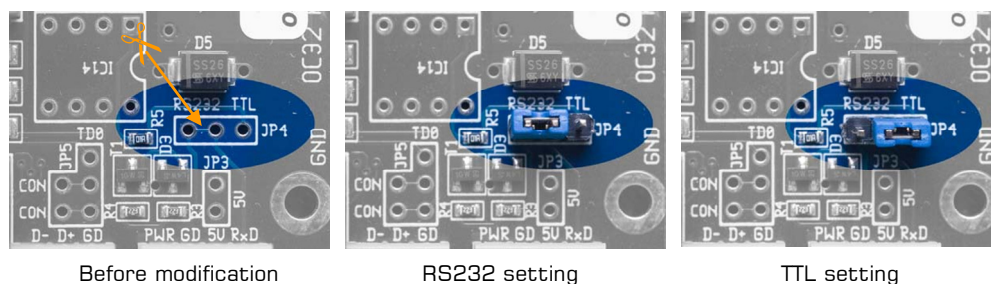


Fig 18: Adding an RS232/TTL selection option

Behind the 4-pole power connector K1 you find 3 pin-holes marked JP4. Take a very close look and you'll see that 2 holes that have the text 'RS232' next to them are shorted. First you need to cut this short circuit. You can do that by using a sharp knife or, for better control, an electric miniature milling cutter tool, such as a Proxxon drill or similar. Just remove the copper between the 2 pin holes and don't cut deeper than some 0.2mm below the surface of the PCB or you may damage the internal layers. Use a multimeter to check that both pin-holes are actually isolated.

Next solder a 3-pin 2,54mm pin-header on the position of JP4.

Now you can select RS232 or TTL by placing a jumper on 2 of the pins as indicated.

## 5.6.2 Dual RS485 interface

The OC32/NG has two serial interfaces: RS485 (channel 1) and RS232 (channel 0). In some cases it is handy to have two RS485 interfaces. For these situations you can make a modification after which you can use channel 0 in RS485 mode, RS232 mode, and if you made the TTL modification as well, in TTL mode.

Note that (at the moment) channel 0 is receive-only, so modifying it into RS485 won't make channel 0 bidirectional.

Standard the OC32/NG has 3 connectors for RS485 channel 1: K2, K2A and K3B. Since we need an additional physical RS485 interface we will separate K2 from K2A and K2B. As a result, we can use K2A and K2B for RS485 channel 1 as before and we can use K2 for RS485 channel 0.

Required components:

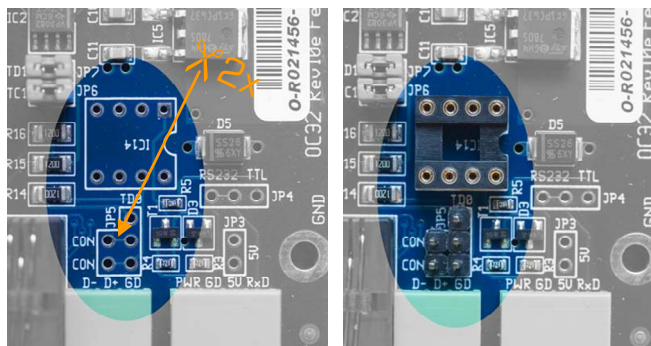
- 2-pin header 2,54mm
- 3-pin header 2,54mm
- 2 x Jumper 2,54mm
- 8-pin IC socket (preferably "turned pin", or "machined contacts", gold plated)
- MAX3082 or SN65HVD3082, DIL version

Behind the 3-pole RS485 connector K2 you'll find 5 pin-holes indicated as JP5. Take a very close look and you'll see that 2 pairs of holes that have the text 'CON' next to them are shorted. These shorts are the connection between K2 and K2A/K2B. First cut these short circuits. You can do that by using a sharp knife or, for better control, an electric miniature milling cutter tool, such as a Proxxon drill or similar. Just remove the copper between the 2



pin-hole pairs and don't cut deeper than some 0.2mm below the surface of the PCB or you may damage the internal layers. Use a multimeter to check that both pin-hole pairs are actually isolated.

Now solder a 2-pin and a 3-pin 2,54mm pin header at the position of JP5. Also solder an 8-pin IC socket at the position of IC14, observe correct polarity.



Before modification                      After modification

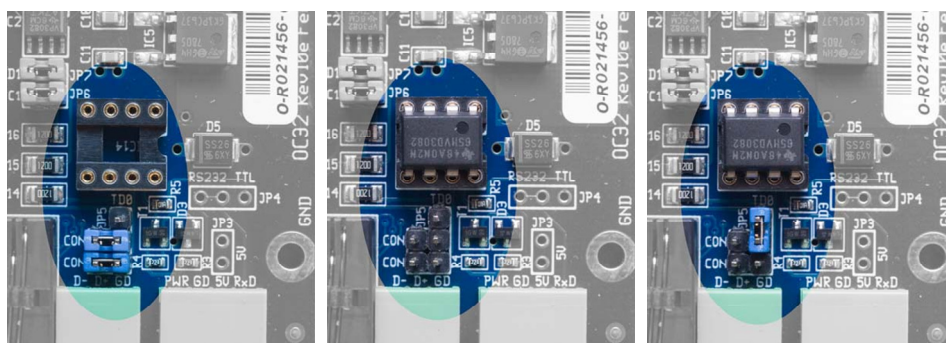
Fig 19: Adding a second RS485 bus

Installing 2 jumpers on JP5 on positions marked "CON" will restore the old situation. Inserting a Maxim MAX3082 or the cheaper Texas SN65HVD3082 (DIL version) in socket IC14 will turn K2 into an RS485 channel 0

A jumper on JP5 in position marked TDO will activate an RS485 terminator on channel 0

As a summary you'll now have the following selections:

- 2 jumpers on K5, position "CON", IC14 empty:  
Standard situation.  
Channel 0 = RS232 (or TTL if you made the mod)  
Channel 1 = RS485 available on K2, K2A, K2B
- NO jumper on K5, RS485 driver in IC14:  
Channel 0 = RS485 available on K2, no termination  
Channel 1 = RS485 available on K2A, K2B  
DO NOT CONNECT K1 pin 4!
- Jumper on K5, position TDO, RS485 driver in IC14:  
Channel 0 = RS485 available on K2, terminated  
Channel 1 = RS485 available on K2A, K2B  
DO NOT CONNECT K1 pin 4!



RS232+RS485                      2xRS485                      2xRS485  
Original situation                      No terminator on chan.0                      Terminator on Chan.0

Fig 20: Selectable options for second RS485 bus

## 6 Connecting the I/O Pins

### 6.1 Selecting the electrical characteristics

The 32 I/O Pins of the OC32 are subdivided into 4 groups of 8 I/O Pins. The 32 I/O Pins can be driven individually and independent of each other. Per group of 8 I/O Pins, the electrical properties can be chosen. This is done by placing the right "driver" depending on the required properties. A "driver" is an Electronic Module (ADM) or an Integrated Circuit (IC) which has to be inserted into a socket on the OC32. This can be done easily by yourself.

On the OC32/NG you will find 4 pairs of IC sockets and 4 pairs of ADM sockets. This may depend on the exact version of your OC32. Not all sockets are available on all models. Consult your vendor if in doubt.

Per group of 8 I/O Pins, you use either the IC sockets OR the ADM sockets (not both, unless in very special cases, not covered in this manual).

When using the IC sockets, you have the following options:

- 500mA Sink Driver (TBD62083<sup>4</sup>): This is the default output-type;
- 500mA Source Driver (TBD62783<sup>3</sup>);
- Resistor-array (5V output with integrated resistor);
- Both Sink Driver (TBD62083) and Source Driver (TBD62783)

The correct place for the drivers on the OC32/NG is depicted in fig 21.

**Attention: Source Drivers and Sink Drivers have there own receptacle. Do not interchange them because this would cause shortcutting. Resistor arrays have to be placed in de Sink Driver receptacle.**

**Attention: Using resistor arrays pin 9 and pin 10 (those pins at the largest distance from the notch) are left empty.**

Note that on the PCB there is text printed to help you indicate the right socket for your driver in case you don't have this document at hand or just to double-check that you are using the right socket for the right driver!

ADM modules are "Add-on Driver Modules" that are specifically developed for the OC32/NG. OC32-ADM's can be purchased from the VPEB partners. At the moment the following ADM's are available:

- OC32-ADM/SI: 4.8A 8-port Sink Driver;
- OC32-ADM/SO: 4.8A 8-port Source Driver;
- OC32-ADM/MX: 4.8A Multiplexer, 4-port sink + 4-port source;
- OC32-ADM/FH: 4.8A 4-channel Full H-bridge;

**When changing drivers, please observe the following:**

- **Remove and install drivers only when power is OFF**
- **Remove IC's carefully without excessively bending pins**
- **Install either regular drivers OR ADM's for the same group of 8 I/O Pins, never both. You can select per group of 8 I/O's if you use a regular driver or an ADM**
- **When installing an ADM, observe polarization. The ADM has an 8-pin and a 12-pin connector and the OC32/NG has 8-pin and 12-pin sockets, so the way the ADM shall be installed can be clearly identified. Also doublecheck that the ADM is correctly installed (without offset) before applying power.**

<sup>4</sup> The TBD series is a MOSFET driver with a very low Voltage drop. Instead you may use a 'normal' bipolar driver, being the ULN2803 or equivalent as SINK driver and the UDN2981A or equivalent as SOURCE driver, however these may introduce a higher Voltage drop.

- **When installing regular drivers, install source drivers in source-driver sockets only and install other drivers in sink-driver sockets only**

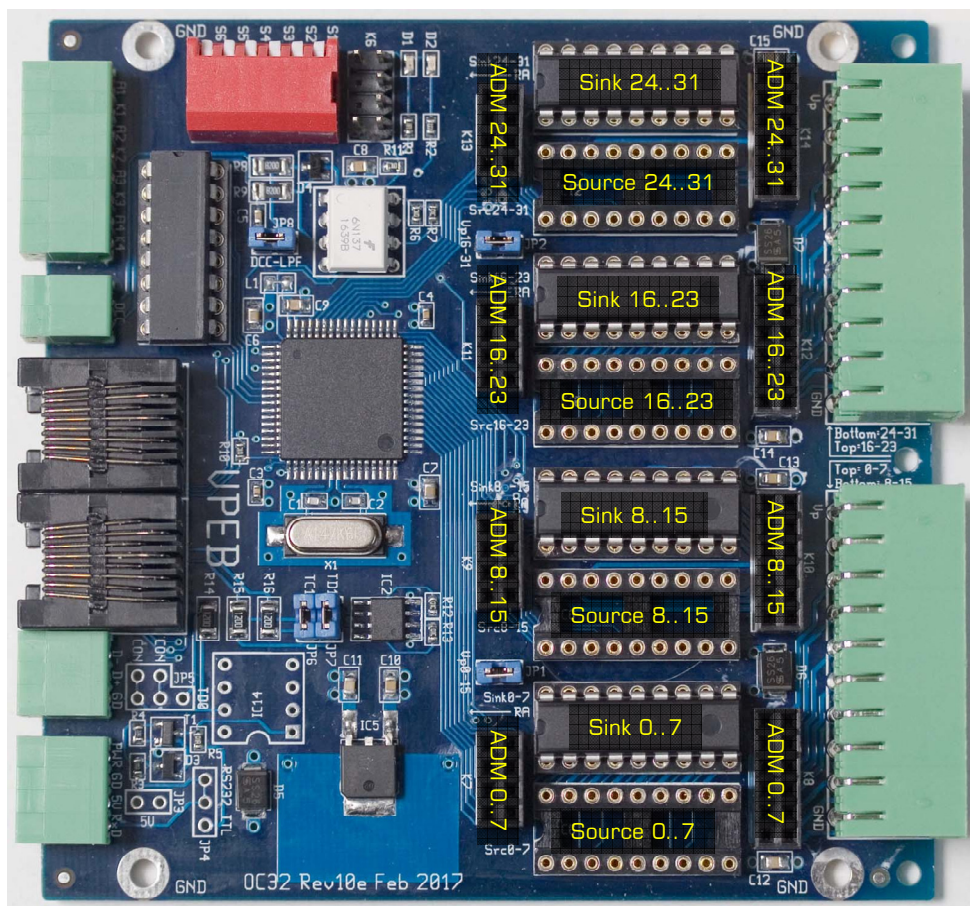


Fig 21: Locating the drivers on the OC32

In the following sections, the different sort of drivers are being explained in more detail. If it is unclear to you it is not of great concern. Just take the examples of section 6.4. There is explained which driver you need for specific types of devices.

### 6.1.1 Sink Drivers 500mA (TBD62083)

This is the default driver delivered with the OC32/NG module. A load is connected between  $V_p$  and the 500mA output. The current flows from the positive voltage ( $V_p$ ) through your load to the output and on the OC32 via the Sink Driver to GND/OV.

Each output can provide 500mA but

**Take Care: de maximum current per group of 8 outputs is 1A**

There is not a real current limitation in the OC32. You have to take care of that yourself or to be certain you have to put a fuse in between. So it is not possible to load all 8 outputs of a single group with 500mA each. In the case of more "heavy" loads they have to be spread over more groups.

### 6.1.2 Source Drivers 500mA (TBD62783)

With a Source Driver the load is connected between the output and GND/OV. The current flows from positive voltage ( $V_p$ ) via the Source Driver on the OC32 to the output of the OC32 and then via your load to GND/OV.

Each output can provide 500mA but

**Take Care: the maximum total current for the 8 outputs of each group is 1 A.**

There is not a real current limitation in the OC32. You have to take care of that yourself or to be certain you have to put a fuse in between. So it is not possible to load all 8 outputs of a single group with 350mA each. In the case of more "heavy" loads they have to be spread over more groups.

### 6.1.3 Sink and Source Drivers (TBD62083 + TBD62783)

If both a Source Driver and a Sink Driver are placed in one Group, two consecutive outputs of that group are going to act as a pair. For the electronic engineers: this means that the group is acting as an quadruple H-bridge. In most cases the load is connected between two consecutive outputs. Those two consecutive outputs can have three different states:

- Both outputs off = load off
- Output Q+0 negative, output Q+1 positive = current into one direction.
- Output Q+0 positive, output Q+1 negative = current into the other direction.
- Both outputs on = **NOT ALLOWED**

If you hook-up a DC motor, with this type of connection, not only the speed can be controlled, but also the direction of rotation. An application in practice would be a turnout motor like Tortoise or Hoffman.



**Attention: In this configuration only one output of each pair may be active simultaneously! Otherwise the result is a shortcut and a fried driver IC. The OC32 has a safety mechanism for this. To be sure that this mechanism works properly the "hardware configuration" has to be set correctly (see OC32 firmware manual)**

### 6.1.4 Resistor Array (5V outputs)

In this case a resistor network is inserted in the receptacle of the **Sink Driver**. A resistor array is nothing else than multiple resistors in an IC housing. It fits into the place where a driver would be placed otherwise.

A resistor array is not a real driver. The I/O Pin of the OC32 processor is connected through a resistor to the equivalent I/O Pin of the module. The resistor provides some protection to the processor and current limitation to protect the load. Because the processor operates with 5V internally, the output voltage of this kind of output is restricted to 5V. The electrical power is obtained from the processor directly. As a result the power is limited.

This setup can be used in the following situations:

- Driving a servomotor. A servomotor has its own power supply and needs only a digital control signal to determine the position. Sink and Source drivers would disturb the control signal and for that reason the resistor array is used.
- Driving a LED. The substantial advantage is that the series resistor for the LED is incorporated in the OC32. The LED can be connected directly. Most often the LED is connected between output and GND but it can be done between two outputs or between +5V and the output as well.
- For using the I/O Pin as input. The input signal applied to the Pin shall remain between 0V and 5V. The resistor acts as a (limited) security mechanism, so the input is not immediately destroyed when accidentally a wrong Voltage is applied to the Pin. When you use all Pins of the bank as input, you best use a somewhat higher resistance value (e.g. 1k $\Omega$ ) for improved protection.
- More complex situations like 3 LEDs on the barrier of an automatic railroad crossing (AHOB) which are driven with two outputs and two wires.



The resistor array shall be a DIL16 version with individual resistors. The value of the resistors can be selected as required but preferably do not go below 100Ω. In most connection examples a reasonable value is suggested

A 5V output in this configuration can supply/sink 40mA, however

**Take care:**

- **The total load of all 5V outputs on ports 0..15 shall not exceed 100mA;**
- **The total load of all 5V outputs on ports 16..23 shall not exceed 100mA;**
- **The total load of all 5V outputs on ports 24..31 shall not exceed 100mA.**

In this case the power for the outputs is retrieved by the processor from the electronics power supply. If a non stabilized power is connected to PWR and the OC32 has to stabilize it, the PWR should be modest. The difference in voltage multiplied by the current is converted into heat. PWR has to be at least 7V and if the 5V outputs require much power the adagio is the lower PWR the better”.

### 6.1.5 OC32-ADM/SI (4,8A power sink-driver)

Use this driver when the required current is larger than the 500mA supported by the standard driver. A load is connected between Vp and the ADM output. The current flows from the positive voltage (Vp) through your load to the output and on the OC32 via the ADM to GND/OV.

Each output can provide 4,8A as an absolute maximum

**Take Care: this is a peak current. Continuous load shall be kept below 2A per channel and below 3A total current for each driver module.**

There is no current limitation in the OC32, so you have to take care of that

### 6.1.6 OC32-ADM/SO (4,8A power source-driver)

Use this driver when the required current is larger than the 500mA supported by the standard source driver. The load is connected between the output and GND/OV. The current flows from positive voltage (Vp) via the ADM on the OC32 to the output of the OC32 and then via your load to GND/OV.

Each output can provide 4,8A as an absolute maximum

**Take Care: this is a peak current. Continuous load shall be kept below 2A per channel and below 3A total current for each driver module.**

There is no current limitation in the OC32, so you have to take care of that

### 6.1.7 OC32-ADM/MX (4,8A multiplexer)

This driver offers 4 sink-driver ports and 4 source-driver ports. It is intended as a module to drive a matrix of turnouts. One MX can drive 8 turnouts (2 solenoids each), two MX's together drive 32 turnouts (2 solenoids each).

The MX can provide 4,8A as an absolute maximum. This is a peak current, but since the /MX is a matrix driver, this will hardly be a problem in practice.

### 6.1.8 OC32-ADM/FH (4,8A Full H-bridge)

This driver is the high-current version of the sink+source driver combination as described in section 6.1.3. Same conditions apply.

Each pair can provide 4,8A as an absolute maximum

**Take Care: this is a peak current. Continuous load shall be kept below 1,5A per channel and below 2,5A total current for each driver module.**

There is no current limitation in the OC32-ADM, so you have to take care of that

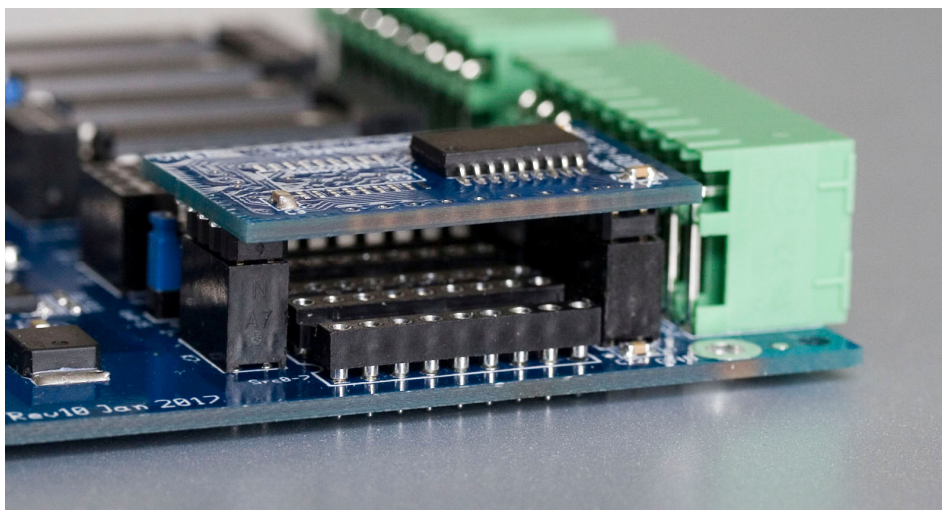


Fig 22: An installed ADM

### 6.1.9 Identifying the different ADM's

If you have an ADM and it does not have a type description on it, you can identify which version you have by the following:

- OC32-ADM/SI: At the top-side the TBD62083 is present, the TBD62783 is missing and at bottom-side the 4 IC's next to "P" are present.
- OC32-ADM/SO At the top-side the TBD62783 is present, the TBD62083 is missing and at bottom-side the 4 IC's next to "N" are present.
- OC32-ADM/MX At the top-side both TBD62783 and TBD62083 are present, at bottom-side 4 IC's are present, 2 in the "P" row, 2 in the "N" row.
- OC32-ADM/FH All components are present.

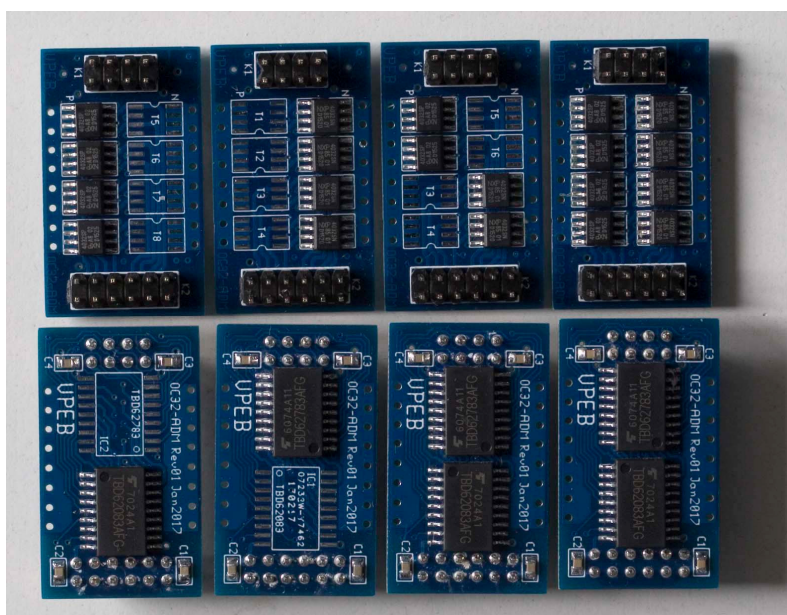


Fig 23: ADM's, top-row=bottom-side, bottom-row=top-side  
from left to right: SI, SO, MX, FH

## 6.2 Multiple power-supplies, different voltages

Even in case you have several pieces of equipment which require different power voltages, you can drive them from the same OC32. Multiple DC power supplies can be used simultaneously.

First the OC32/NG has two separate power planes for driving loads. You can separate Vp1 and Vp2 and drive each section with it's own power supply. This works for all drivers (sink, source, ADM's).

### 6.2.1 Multiple power-supplies using Sink-drivers

In addition to the fact that you have two power groups on the OC32/NG, when using Sink Drivers the number of different power supplies is virtually unlimited. The load is connected between the plus pole of the power supply and the output of the OC32 (Sink Driver). As long as the minus pole of all power supplies are connected to each other and to the GND of the OC32 the plus pole of each of those power supplies can be used to feed one or more separate accessories. The plus pole of the load is connected to the plus pole of its power supply while the minus pole of the load is connected to the output of the OC32.

There is however one additional point of interest:

In the Sink Drivers free-wheeling diodes are integrated to facilitate inductive loads to be switched-off without damage, like solenoids or relays. These diodes reside between the output (Anode) and Vp (Cathode). When using different power supplies, the consequence of this is that **always the power supply providing the highest voltage in the Vp group shall be connected to Vp.**

In most cases it is not desirable to use that highest voltage for the internal power of the OC32 also. A lower power should be used to operate the OC32. To achieve that, PWR and Vp have to be connected separately. The power jumper JP1/JP2 on the OC32 print has to be removed. The highest voltage has to be connected to Vp. The lower voltage has to be connected to PWR.

### 6.2.2 Multiple power-supplies using Source-drivers

Unlike the case when using Sink Drivers, it is not possible to connect more than one power supply to the Source Drivers in one Vp group. The Source Driver obtains the power from Vp and only one voltage can be connected to Vp.

When using both Source and Sink Drivers for different outputs but in the same Vp group it is possible to use different power supplies as described in 6.2.1 above, but **the highest voltage has to be connected to Vp, being the voltage used by the Source Driver.** All devices driven by the source driver will be driven with the same and highest Voltage.

### 6.2.3 High power Voltage

The Sink and Source drivers can deliver/withstand up to 50V. It may not be wise to use such a high voltage because this approaches the voltage levels that are harmful to the human body when touched. More over in the miniature world there are hardly any devices requiring such a high voltage.

If you need to use a voltage above 15V for your loads it is advisable to separate Vp and PWR. If the voltage is higher than 25V the separation is mandatory because the voltage stabilizer of the OC32 is not specified to handle more than 25V.

### 6.3 Connecting the I/O Pins

The loads to be driven by the OC32 are being connected to the connectors K5A and K5B. Each 20-pin socket accepts two 10-pin plugs. Each plug carries 8 signal wires plus Vp and GND

The pin-out is given in figure 16 below.

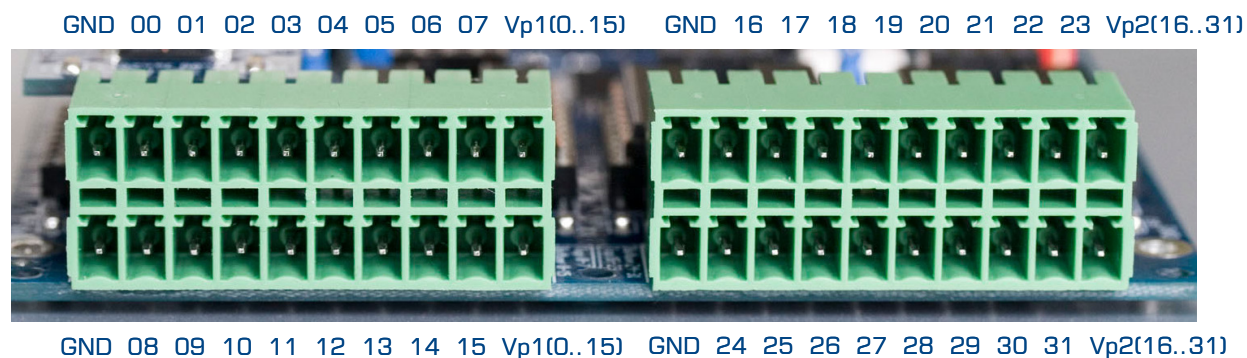


Fig 24: Pinout sockets K5A and K5B

## 6.4 Example-connections

Provided the required voltages and currents are within the limits of the OC32 it can drive all devices you may encounter. Most parts like incandescent bulbs, LEDs and relays can be connected directly. With some simple extra components other things can be driven like decouplers, turnout coils and motors. Below you find a number of examples of devices to be found in the miniature world are presented.

### 6.4.1 LED's with common anode (+)

*OC32 I/O Pin: 500mA Sink Driver*

Connect the common connection to the positive power terminal ( $V_p$  = pin 28/29 of K5). Connect the cathodes of the separate LEDs via a resistor to the output of the OC32.

When only one LED is lit simultaneously (like with NS 3 color signals or block signals), the resistor can be in the common lead and the cathodes of the LEDs are connected directly to the output of the OC32.

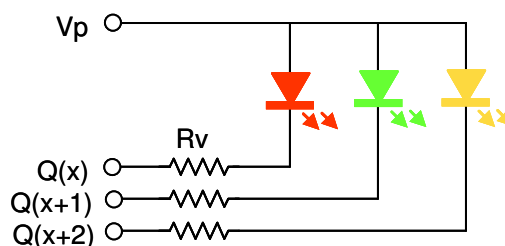


Fig 25: Connecting LEDs common.Anode

The value of the resistor depends on the voltage ( $V_p$ ), the current required by the LED and how bright you want the LED to shine. In case of ready built signals (for example Viessmann) those resistors are incorporated and the voltage is prescribed. In our opinion the 1,2k $\Omega$  applied by Viessmann at 14..16V results in too much light, especially when the surroundings are in the dark. With this value of the resistor and the built-in LEDs the result is much more attractive at a voltage level of 9..10V. With 14V a resistor of 1,8k $\Omega$  or 2,2k $\Omega$  can do the trick.

Just experiment at day and night time before installing things definitively or use an adjustable power supply

When more than one LED has to be connected to one output, connect the LEDs in series if possible. Otherwise provide each LED with its own resistor and connect several LED and resistor combinations in parallel.

### 6.4.2 LED's with common cathode (-)

*OC32 I/O Pin: Resistor-array*

Connect the common lead to the GND/OV. Connect the anode of the individual LEDs to the outputs of the OC32. A resistor is not required because it is in the OC32. Use a resistor array with a value of 220 $\Omega$ . Low-current LEDs being used a higher value of the resistor array has to be applied.

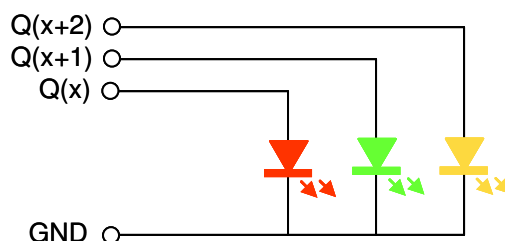


Fig 26: Connecting LEDs common Cathode

### 6.4.3 Incandescent Bulbs

*OC32 I/O Pin: 500mA Sink Driver*

Bulbs have to be connected between the positive voltage ( $V_p$ ) and the OC32 outputs. A resistor is not needed. The voltage ( $V_p$ ) has to match the voltage required by the lamp. Having more individual bulbs in the same casing (e.g. a signal) the common lead has to be connected to  $V_p$  and the individual connections to outputs of the OC32. If bulbs have to be switched

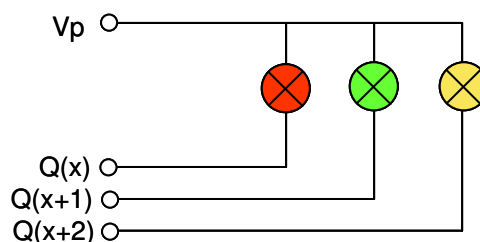


Fig 27: Connecting incand. bulbs



together they can be connected in parallel as long as the current does not exceed 500mA.

#### 6.4.4 LED's antiparallel

*OC32 I/O Pin: Resistor-array (preferred) or 500mA Sink Driver*

With some signals, the LEDs are connected anti parallel. An example would be the Märklin "Hobby" signals 74391 and 743xx. It is preferred to drive these signals with the 5V outputs of the OC32 but the 500mA driver will do the trick as well.

Using the 5V output configuration (figure 28) the signals can be connected to 2 consecutive outputs of the OC32. The required resistor is part of the OC32. For the resistor array take a value of 100Ω.

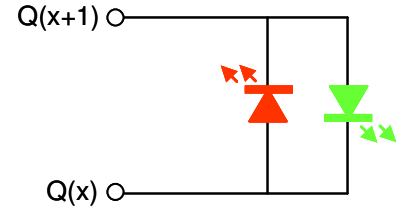


Fig 28: Connecting LEDs antiparallel in 5V mode

Should you have some spare 500mA outputs left, this will work also (figure 29). In that case you need to add 2 resistors between the 2 outputs and Vp. Be aware that the signal has to be connected the other way around because of the inverting operation of the 500mA driver. The value of the resistors depends on the required voltage Vp. At 12V a reasonable value would be 1k5, but one could experiment.

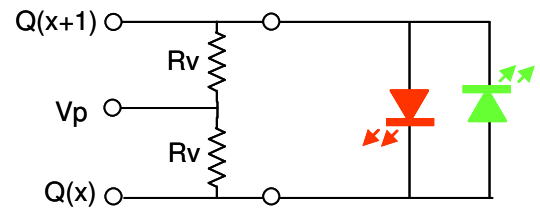


Fig 29: Connecting LEDs antiparallel in 500mA mode

#### 6.4.5 LED's on barrier bars

*OC32 I/O Pin: Resistor array or 500mA Sink Driver*

On Dutch railroad crossings 3 lights are fixed onto the barriers of the railroad crossing (AHOB). The light at the end (the top light) is lit continuously where the two other lights are blinking alternately. If you want to construct this in your miniature world it might be a challenge to attach the wires along the moving barrier. In any case the fewer wires, the better. Presented is a diagram to drive these 3 LEDs with only 2 wires and 2 outputs of the OC32. In the case of two barriers the circuit can be connected twice in parallel as long as the LEDs you use are of the same type and from the same manufacturer (if possible from the same production series). The (double) diode is put somewhere on the barrier. Take care: it is a SOT23 diode (SMD), also very tiny. To attach the LEDs to the barrier you will have to select SMD types also.

Fig 30: LEDs on the barriers (5V mode)

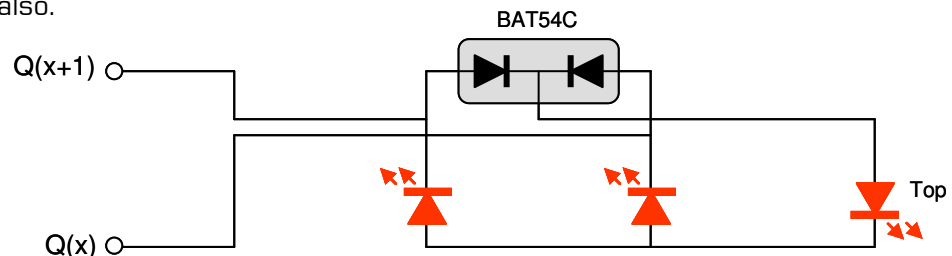
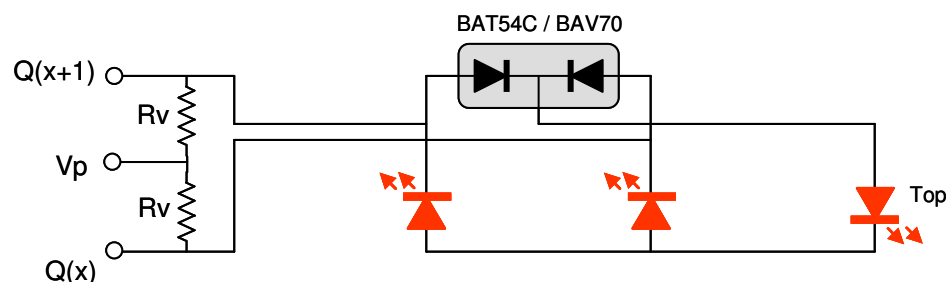


Fig 31: LEDs on the barriers (500mA mode)



### 6.4.6 Decouplers

*OC32 I/O Pin: OC32-ADM/SI (4,8A Sink Driver)*

Solenoids may draw up to 1,5A (or more). That is too much for the 500mA driver so you will need an OC32-ADM/SI to solve that. The solenoid is connected between Vp and the OC32 I/O Pin.

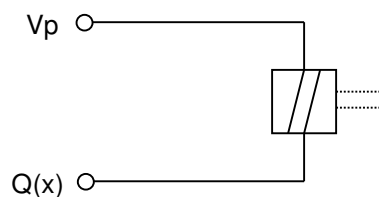


Fig 32: Decoupler solenoid

### 6.4.7 Turnout-solenoids

*OC32 I/O Pin: OC32-ADM/SI (4,8A Sink Driver)*

Solenoids may draw up to 1,5A (or more). That is far too much for the 500mA driver so you will need an OC32-ADM/SI to solve that. The turnout solenoids are connected between Vp (common wire) and two consecutive OC32 I/O Pins.

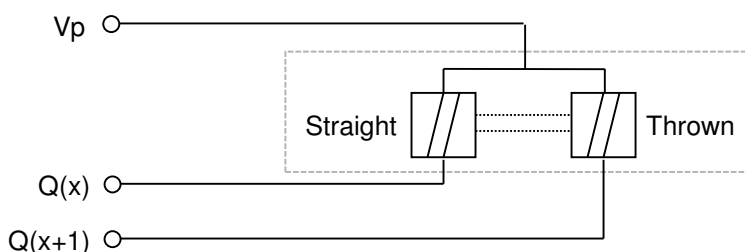


Fig 33: Solenoid driven Turnout

If you have turnouts with end-switches that switch-off current when the end-position is reached, you may, in some cases experience problems with the stability of your system. This specifically can happen when the distance between your turnouts and the OC32 is short. The reason is that end-switches can cause very high Voltage spikes when the current flowing through an inductor is suddenly interrupted. Stopping the current by electronics is a far more subtle process.

Our first advice is to remove the end-switches if reasonably possible. Many experience problems with end-switches after years of operation, since the continuous spikes are not only a pain for the electronics, but also lead to contact burn-in. At some point the turnout motor will stop working. Knowing that, removing them in the first place will solve problems now and possible problems in the future.

If you cannot or choose not to remove the end-switches, the spiking problem can be mitigated by soldering two 1nF multilayer capacitors between the connecting wires of your turnout drive-unit. If you do, place them as close to the drive unit as reasonably possible.

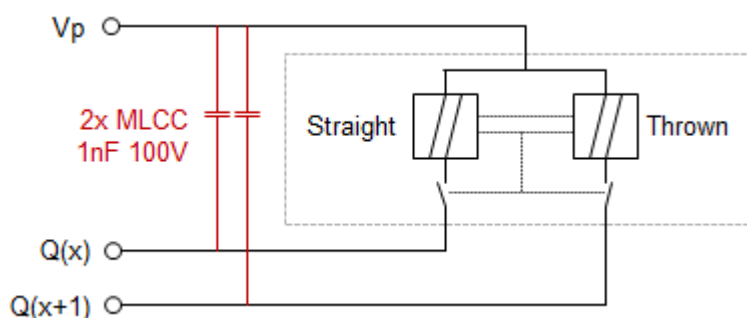


Fig 34: Solenoid driven Turnout with end-switches

## 6.4.8 Turnout-solenoids (multiplexed)

*OC32 I/O Pin: OC32-ADM/MX (4,8A Multiplexer Driver)*

If you have a large amount of turnouts to control, multiplexing is a very cost-effective method to achieve this. With one ADM/MX installed, 8 I/O pins control 8 drive units (so 16 solenoids). Two ADM/MX's control up to 32 drive units (so 64 solenoids) by just 16 I/O Pins.

Multiplexing requires some additional components, but nevertheless is it a cost-effective solution. Moreover, the multiplexing and additional components mitigate the potential spiking problem as described in paragraph 6.4.7, it saves wiring and multiplexing makes very efficient use of your power supply.

The only drawback of multiplexing is that it is slightly more work to install. Therefore there is a separate document on turnout multiplexing with the OC32. We'll just mention the basics here. Please refer to the separate document for more info and details.

## 6.4.9 Relays

*OC32 I/O Pin: 500mA Sink Driver*

Sometimes a galvanic separation is needed between output and switched load. Also when you want to switch something that requires AC (e.g. synchronous motors) this is a valid solution. Several types of simple DIL relays are available and after some searching around they can be found for prices between € 2,- and € 3,50

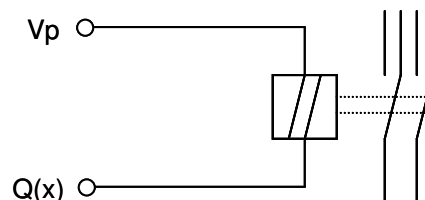


Fig 35: Connecting a relay

The diagram to connect a relay is rather simple: just connect it between Vp and the output of the OC32. We are talking about a monostable relay, meaning that the relay will always be in idle position when no voltage is applied. As soon as and as long as voltage is applied to the coil the active position is maintained. If the voltage is switched off the relay goes back into the idle position.

Take care: some relays are polarity sensitive. It is important to connect the right pole to Vp and to the OC32 output. The power supplied by Vp needs to match the voltage required by the coil of the relay. This is indicated on the relay or can be found in the documentation of the manufacturer of the relay.



### 6.4.10 Motors (unidirectional)

*OC32 I/O Pin: 500mA Sink Driver*

*or*

*OC32 I/O Pin: OC32-ADM/SI 4,8A Sink Driver*

Driving a DC motor, which needs to run in one direction only, is quite simple. Connect the motor between Vp and the output like in Fig 36.

When the motor draws more current than 350mA at maximum load (not being maximum speed) preferably use the OC32-ADM/SI.

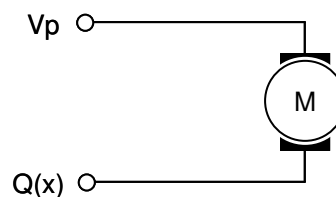


Fig 36: Connecting a motor

### 6.4.11 Motors (bidirectional)

*OC32 I/O Pin: Sink Driver + Source Driver*

*or*

*OC32 I/O Pin: OC32-ADM/FH 4,8A H-bridge*

A bidirectional motor is a motor that runs into two directions. An example is a turnout motor which moves slowly. The problem is reversing the current to have the motor run the other way around. This can be done with a so called "H-bridge" circuit. By putting both a Sink Driver and a Source Driver into the OC32 such a H-bridge is established. The motor has to be connected between two consecutive outputs.

The motor runs in one direction when output Q(x) is active. When output Q(x+1) is active the motor runs in the other direction. Also the speed can be adjusted.

**Take care: It is of vital importance that Q(x) and Q(x+1) are never active at the same time.**

This can be set-up in the OC32 configuration.

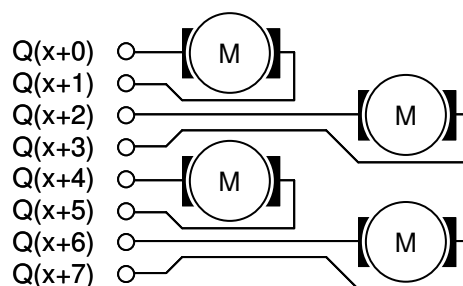


Fig 37: Connecting bidirectional motors

When the motor draws more current than 350mA at maximum load (not being maximum speed) preferably use the OC32-ADM/FH

### 6.4.12 Servo Motors

*OC32 I/O Pin: Resistor array*

A servo motor is a motor with integrated electronics, that can be moved into a desired position by means of a digital control signal. The servo motor requires a separate power supply to operate. The voltage usually has to be between 4,5V and 6V. Details can be found in the documentation of the manufacturer. Normally speaking a servo motor has three leads: GND, Power and input. These have to be connected according to figure 38.

**Take care: A servo motor can draw quite some current and especially the cheaper versions generate a lot of interference. To prevent negative impact on the Electronics, it is wise to incorporate a separate stabilizer and capacitor into the 5V power supply of the servo motor.**

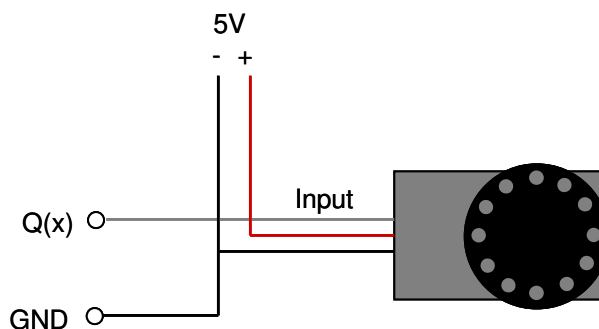


Fig 38: Connecting a servomotor

To easily provide power to a servo and connect it to the OC32, a separate compact connection module has been developed, the SPO4. With the SPO4 up to 4 servos can be connected in a close range. When you have more servos or have servos at different locations, multiple SPO4 modules can be used. A separate manual for the SPO4 is available.

### 6.4.13 Pushbutton or switch (input)

*OC32 I/O Pin: Resistor array*

An OC32 Pin can also be used as input. This opens the possibility to control a Device, connected to the OC32, such as a railroad crossing, by means of a switch, pushbuttons or e.g. reed contacts, connected to one or multiple Pins of the same OC32. Also the activation or de-activation of pushbuttons and flipping a switch can be reported to the controlling PC, if the software on that PC's supports this function.

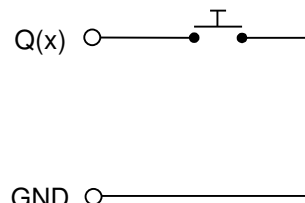


Fig 39: Connecting Pushbutton

A pushbutton or switch is connected between OC32 Pin and GND.

### 6.4.14 Input from another control system (input)

*OC32 I/O Pin: Resistor array*

Perhaps you want to control the OC32 from another system and you may do that by "hardwired interfaces", so by connecting a number of individual wires.

If 'the other control system' offers relay-outputs (potential-free contacts), the contacts of these relays are no different than the contacts of a pushbutton or switch. Therefore you can make the interface as described in paragraph 6.4.13. Replace the shown pushbutton/switch by the make/break relay-contact. Make sure that via the relay-contact no other high or negative Voltage can reach the OC32 Pin.

If your external system has electronically active outputs, then to interface successfully with the OC32, these outputs may in principle offer no higher voltage than 5V or any lower Voltage than 0V to the OC32. If the outputs supply more than 5V you can limit the Voltage delivered to the OC32 by means of a Voltage divider as shown in figure 40 below. The resistor  $R_s$  depends on the output-voltage of your external system and can be found with the help of Table 2.

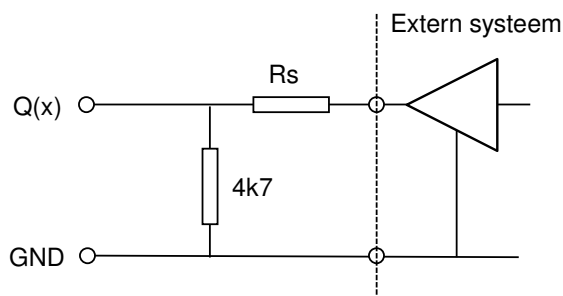


Fig 40: External system with Voltage conversion

Output Voltage (V)	$R_s$ (k $\Omega$ )
6	1,0
7,5	2,4
9	3,9
12	6,8
15	10
18	12
24	18

Table 2: Series resistor for Voltage Conversion

If your external system offers Open Collector (or Open Drain) outputs, the you can interface these directly to the OC32 Pin as per paragraph 6.4.13. However it is mandatory that in this case both systems share the same GND/OV. If this is not the case, you can insert an intermediate relay or an optocoupler between both systems.

## 7 Serial Accessory Port

Starting from firmware 3.0.0.0, the OC32 offers a Serial Accessory Port. This feature enables control of external Devices without using I/O Pins. A typical example of such an external Device could be a sound module.

If the sound module has just a single function (sound on or off, e.g. the bell of a railroad crossing) it is questionable whether serial control makes much sense. Probably in this case it is easier to use an OC32 I/O Pin to switch sound on or off. A different case is using a sound module that can play multiple sound fragments, such as background sound, depending on the time of the day and maybe specific circumstances, or playing announcements in station about arrivals and departures of trains. In these cases you want more control options and controlling a Device by serial command may be an interesting option.

Of course a sound module is just an example. In principle you can control all kind of Devices that have a serial input for control. The requirement is that the protocol is extremely simple, just a few characters for a control command, and control is unidirectional. The OC32 can send command to the external Device, but the external Device cannot send anything back. If you are handy with microcontrollers, you could develop your own control module for 'something', e.g. based on a Microchip or Atmel microcontroller, and control this serially from the OC32.

The SAP shares the serial hardware (UART) with the RS232 interface on the OC32. The consequence is that, when using the SAP, the normal RS232 port can no longer be relied on. The SAP therefore can only be used when you control your OC32 operationally by RS485, DCC or of the module functions autonomously.

The SAP has a TTL level interface (0-5V). Many microcontrollers feature a built-in serial port working on a 0-5V level and can be interfaced directly. If you need RS232 or RS485 levels, you'll need to add a signal converter after the SAP.

The SAP can be found on the 6-pin connector, next to the diagnostic LEDs. The same connector also offers 0V/GND and 5V connections. You can use this 5V to power a signal convertor, or even your external controller, if it does not consumes too much power. Be aware that the current to this port is negated by the on-board 7805 at the OC32. How much additional current can be drawn safely depends on e.g. how many outputs you already drive by resistor arrays and what unstabilised Voltage you apply to your OC32.

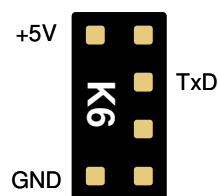


Fig 41: Pinout SAP

## 8 External Events

### 8.1 Introduction

Optionally, the OC32 can be equipped with “Event Inputs”. These are 4 additional input ports by which the OC32 can react to external events.

These extra input ports are inherited from a comparable function in LichtOrgel (LightOrgan). On this device the 4 input ports can be used to start ‘a program’. Each program generates a specific effect on your miniature world. For example, the 4 programs on a LichtOrgel could be used for

- Morning;
- Daytime;
- Evening;
- Night.

But of course it is possible to define your own implementation.

The function of the 4 “Event Inputs” on the OC32 is much more flexible than the comparable function on LichtOrgel. For every “Event Input” it can be configured if a “Pin” should react and by which “Aspect” it should react. The configuration by means of OC32config is described in the OC32 firmware manual.

Depending on your own preference or application, the interface to the 4 “Event Inputs” can be through an optocoupler or a resistor array. The position of the interface on the OC32 is shown in figure 42. Each input port (In0..In3) has 2 terminals “A” and “K”

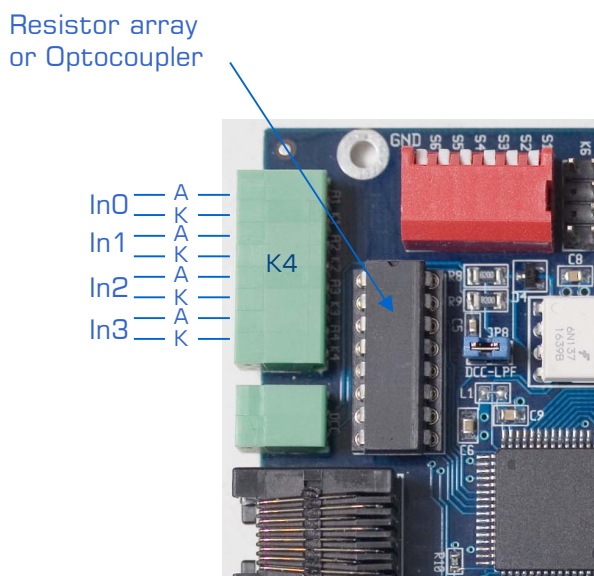


Fig 42: Optocoupler inputs

### 8.2 Resistor array

A resistor array is the preferred choice if the input port(s) are to be activated by a push-button or the contacts of a relay. A resistor array can also be used if the event input is to be activate from another OC32, OM32 or LichtOrgel and this module is on the same GND level as the OC32 to be activated. A push-button can be connected according to figure 43 or figure 44. Obviously, a push-button can be replaced by a (separately derived) contact on a relay.

If the OC32 is controlled from another OC32, OM32 or Lichtorgel, the 'A' wire should be connected to the 500mA output port of that device. That output port may not be connected to anything else and the controlling device must be connected to the same GND.

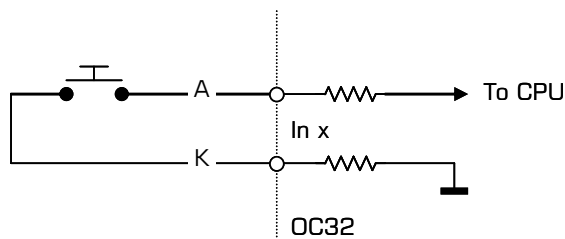


Fig 43: Pushbutton on Event Input (1)

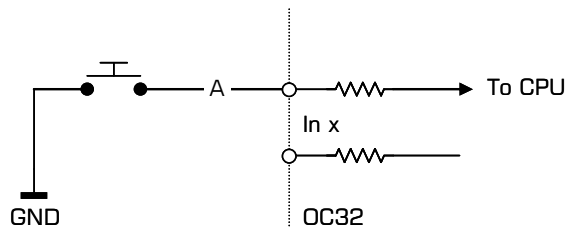


Fig 44: Pushbutton on Event Input (2)

### 8.3 Optocoupler

An optocoupler can be used if the event input ports are activated from any other electrical system. In this case, the "A" and "K" connection of each input port is respectively the Anode and Cathode of the IR diode of an optocoupler.

The maximum current through the input port optocoupler is 50mA. However, this maximum current is never needed. Therefore the advice is to stay far below this maximum. The input port is reliably triggered by a current from 1mA up. There are many ways to drive an optocoupler. It is far beyond the scope of this manual to describe all possible circuits. As an example we describe the possibility to control the input port from a pushbutton or other type of switch.

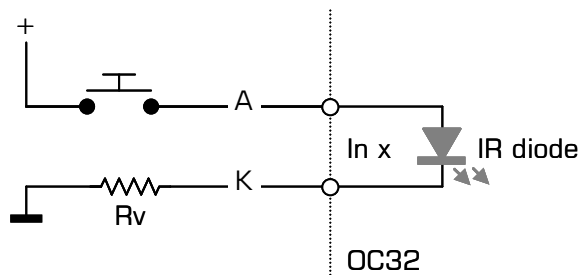


Fig 45: Connecting the Optocoupler input

If you find it difficult to calculate the series resistor  $R_v$ , please have a look in the FAQ on [www.dinamousers.net](http://www.dinamousers.net)

A simple calculation:  $R_v = (V - 2) / 5$  gives the approximate value of  $R_v$  in  $k\Omega$  that will result in a current of 5mA through the optocoupler.  $V$  is the voltage on the "+" terminal. When using a voltage of 12V the formula gives 2k $\Omega$  for  $R_v$  (thus practically a 2k2 from the E12 series)



## 9 Installation of the U485

The U485 is a USB to RS485 converter designed to be used with the OC32. The U485 connects up to 96 OC32 modules to a single USB port.

To use this interface it is of course necessary that your PC is equipped with a USB port. The U485 is compatible with the following operating systems:

- Windows 98
- Windows ME
- Windows XP
- Windows Vista
- Windows 7
- Windows 8, 8.1
- Windows 10
- Linux
- MAC OS/X

Windows from Vista onwards and Linux from kernel 2.6.31 usually recognize the U485 and automatically install the appropriate driver software if your PC is connected to the Internet. So effectively, the only thing you have to do is plug in and wait a minute for the driver to install.

For the other operating systems, the correct drivers must be installed before you plug the USB interface into the PC. In case it does not work automatically with Vista or later, unplug the U485, manually install the correct driver and try again.

The latest driver can be downloaded (for free) from the website of Future Technology Devices: <http://www.ftdichip.com/Drivers/VCP.htm>

The easiest way is to use the "setup executable for default VID and PID values". You can also find the driver on the Dinamo Users Group website.

The latest version of the driver is not guaranteed to work under Windows 2000, Windows 98 or Windows ME. However, there is an older version that can be downloaded and should work in these OS versions.

After the driver is installed you can plug the U485 into the USB port of your PC. The first time the U485 is inserted, the PC will load/configure the driver for the U485 and create a virtual COM port.

When everything went right you will now find an additional COM port in Control Panel -> System -> Hardware -> Device Manager -> Ports (the exact path depends on your OS version and language). Make a note of the COM port number. You will need this later to make the right settings in the software.

The U485 is now ready for use.

On the U485 there are two LEDs that have the following function:

- Green: The U485 receives data from the RS485 bus
- Orange: The U485 sends data to the RS485 bus

Note the U485 can be plugged directly in a PC USB port. Should this not be the case because of physical limitations, you can use a USB A-A extension cable. Should you need to do this, keep this cable short (less than 2 meters). The USB interface is susceptible to noise (which is usually extensively available in a model railroad environment), while the RS485 interface is highly immune. So bridge the distance with RS485 (after the U485) not by USB!

(This page is intentionally left blank)

(This page is intentionally left blank)