VPEB Van Perlo Elektronica & Besturingstechniek

OC32

Manual

Smart electronics for your miniature world

Author: Leon J.A. van Perlo

Version: 1.41
Date: December 15th, 2017

Release management

This manual applies to

- Print
  - OC32 Rev 00
  - OC32 Rev 01
  - OC32 Rev 02
  - OC32 Rev 03
  - OC32 Rev 04

All above mentioned versions are functionally and electrically 99% identical. If a function is applicable for a specific revision level this will be mentioned in the manual.

- Rev 01
  Replaced some through-hole components by SMD for production-efficiency.
- Rev 02
  Some SMD components have a different form factor and it has one additional interconnection that had to be made afterwards in some cases with the previous designs.
- Rev 03
  An additional 22nF capacitor has been added to the DCC interface to improve noise immunity in noisy environments.
- Rev04
  Clipterminals have been replaced by plug-in connectors for improved convenience. The option to mount clip-terminals has been retained to order on request.

With the introduction of the OC32/NG the OC32 manual is split in a hardware and software part. This manual only describes "the hardware" OC32. Since firmware and software are 100% identical to those for the OC32/NG , software functions are described in an OC32 Configuration Manual.

Preface / Reading Guide

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 "Configuration" and 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 knowledje 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.
Or course the latter does not apply to matters of more individual nature, such as warranty and orders. You can contact VPEB at dinamo@vanperlo.net

Enjoy!

Contents

(This page is intentionally left blank)

1 OC32 - Introduction

1.1 A different approach

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

The big 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 O32. 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.

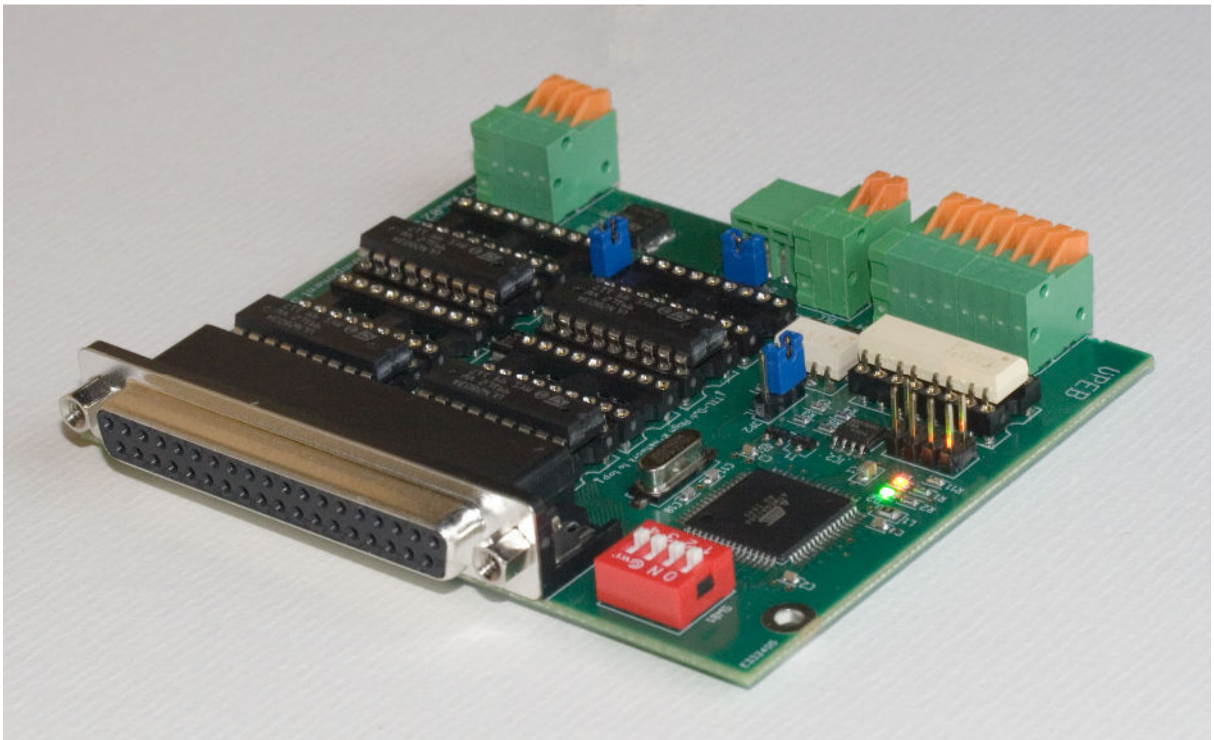

Fig 1: OC32

1.2 Functions

The OC32 offers 32 I/O Pins that are usable in a very universal way (referred to as "**Pins**" in the remainder of this manual and the software). The Pins can be electrically adapted to 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

And in combination with the DS32 connectorprint:

- Solenoids for turnouts and decouplers (up to 4A peak-current)

The 32 Pins can also be used as input. In this was a Pin can be used to connect e.g. a switch or pushbutton by which actions in the OC32 can be activated. In this way the OC32 can even be used by those who want to control their miniature world without PC.

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 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;
- ...

The OC32 can be delivered with DCC interface (option), via which the module can be controlled by DCC packets. This implies you can use the OC32 in combination with any digital control system which is capable of generating DCC accessory packets.

Also the OC32 can be delivered with 4 additional galvanically separated inputs (optocouplers). This enables the OC32 to react on external events. Which actions are taken on any external event is completey user-configurable.

Starting from version 3.0 the OC32 features an additional serial output-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 Pins. The additional serial output is available on any OC32 revision, so also the older modules, when firmware 3.0 is installed.

1.3 OC32 Pins

An OC32 Pin can be driven in PWM-mode or Servo-mode or ca be used as an input (Input-mode)

1.3.1 PWM mode

PWM mode is the standard control option and is used for any type of device except servo motors. PWM mode offers the following possibilities:

- 32 intensity levels (0% to 100% with 30 intermediate steps)
- 32 "acceleration characteristics" to simulate fading (on/off) of incandescent lamps by LEDs, slowly changing the lighting-level or accelerating and decelerating motors.

1.3.2 Servo mode

The OC32 is capable of driving servo's in both simple and dynamic mode.
Simple servo control means:

- Adapting the drive parameters to the specific properties of your servo-unit (midpoint and maximum swing);
- Setting the desired position (-64..63). The servo moves immediately to the required position.

Dynamic servo control by the OC32 is based on a physical modeling technique, comparable to a mass attached to a spring, encountering friction when it moves. By varying the parameters of this model, a large number of situations can be simulated. Because the behavior is built on a physical model by design, resulting movements appear to be very "natural" Some examples:

- Slowly moving a turnout or opening/closing a sliding fence;
- Semaphores and barriers of railway-crossings with slow-movement and self-designed ringing-effects;
- A door of a shed or garage that bounces a few times when closing and swings a few times when opening;

The possibilities are almost infinite and only limited by your imagination. To support these features there is a graphical model that predicts exactly the movement based on the parameters you choose. The simulator is very accurate, which makes your design process easy.

1.3.3 Input mode

Input mode configures the Pin as input. This mode offers the following possibilities:

- Generating a signal to the controlling PC. By this function, the controlling program can detect external events
- Activating actions on one or more of the other OC32 Pins

1.4 Communication

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

The RS485 interface is bidirectional (can receive and transmit). The RS232/TTL interface nowadays is mainly there for compatibility with the OM32 and to communicate with systems that do not support RS485.The RS232/TTL interface on the OC32 ca receive only.

Up to 96 OC32 modules can be connected to the serial bus.

The current versions of Dinamo and Dinamo/MCC offer RS485 interfaces out of the box. Most PC's however do not. For those who want to interface the OC32's directly to the PC, VPEB developed the U485. This is a very compact USB to RS485 converter.

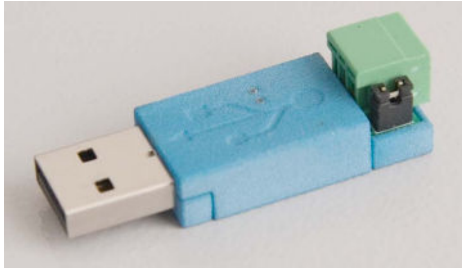

Fig 2: U485

Those who have a DCC control system and want to control the OC32's from this DCC system can buy the OC32 with a DCC interface[1]. The OC32 can be controlled as Basic DCC Accessory Decoder and as Extended DCC Accessory Decoder. Unfortunately there are still very few DCC control systems that feature Extended DCC Accessory Packets. The amount of Basic DCC and Extended DCC Accessory Decoder Addresses the OC32 module uses, can be configured by software.

The OC32 can NOT 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.

The OC32 can also drive external devices through a serial interface. If this option is used, the OC32 cannot be controlled anymore via the serial RS232/TTL interface and therefore the RS485 or DCC interface needs to be used for this.

1.5 Physical properties

The OC32 is supplied as an assembled printed circuit board without enclosere. The PCB dimensions are 100 x 80 mm. The outputs are available on a 37 pin subD connector onto which the wiring from the controlled devices can be soldered. The choice fo a solder-connection has been made to keep the module compact and economical.

In case soldering is difficult of not preferred, a special connection-module (DS32) is available with screw-type or spring-loaded clip terminals. This connection-module can also be equipped with amplification transistors that boost the maximum output current to 4A. The DS32 is described in a separate manual.

The OC32 is only supplied as an assembled and tested module, so not as "kit" or "bareboard". The reason is that many components on the OC32 are of SMD type that are not easily solderable by the average hobbyist.

For the OC32 an optional enclosure is available

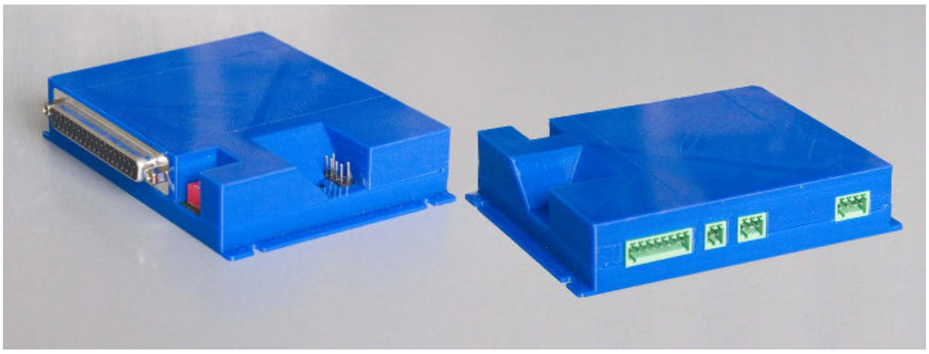

Fig 3: OC32 in enclosure

[1] The DCC interface can also be added in a later stage by the user. All needed components are "through hole" (so not SMD). You need to do some soldering to the PCB however.

2 Overview

2.1 Interfaces and functions

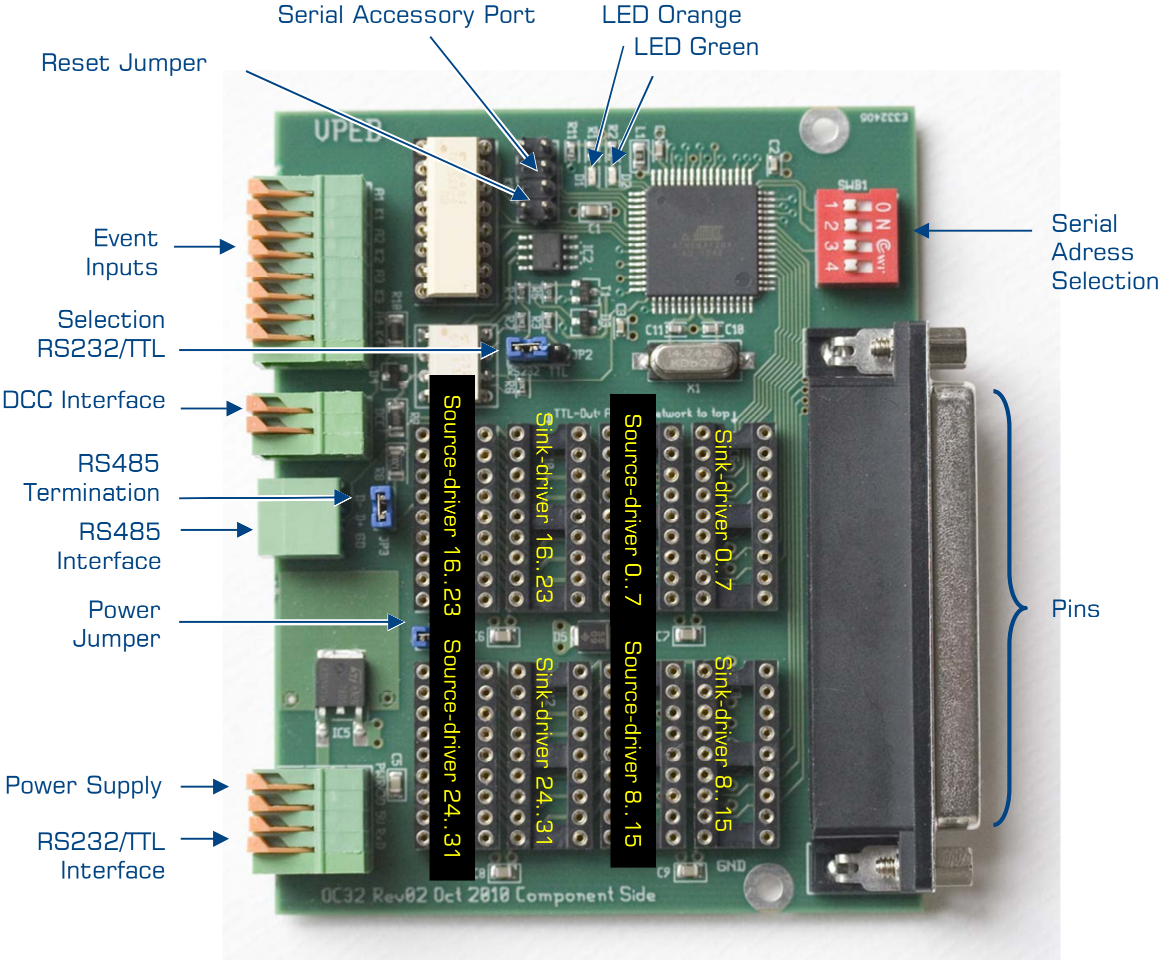

Fig 4: OC32 overview interfaces and functions

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

Interfaces:

- Pins: a 37 pole sub-D connector accommodates 32 I/O Pins and power supply;
- RS485 interface;
- DCC interface (optional);
- Event input interface (optional);
- Four pole connector for power supply and RS232/TTL interface.

On the PCB:

- 8 IC-sockets for driver-ICs. Each group of 8 outputs has one socket for a Source Driver and one for a Sink Driver;
- Indicator LEDs, orange and green;
- Address selection: a dipswitch with four positions to select the address of the module
- Reset jumper. Only used to start the bootloader;
- Serial Accessory Port: The connection to control special serial devices;
- Selection jumper RS232/TTL;
- RS485 Termination jumper;
- Power jumper.

2.2 LED indicators

The **standard** functions of the LEDs on the print (green and orange) 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).

The function of the LEDs is configurable to allow a more detailed diagnosis.

Bootloader:

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

3 Connecting power supplies

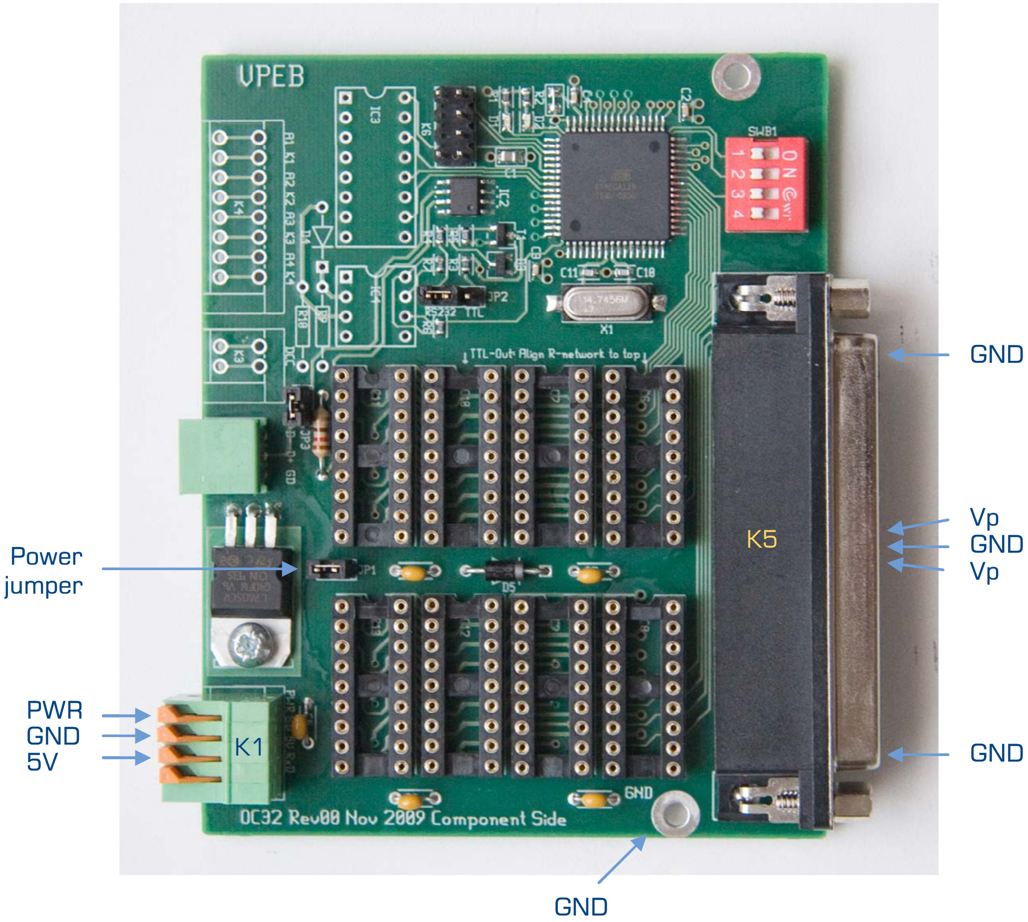

Fig 5: Connecting Power Supplies to the OC32

3.1 General

The OC32 itself and the devices that the OC32 controls (e.g. signals, motors, relay's) obviously need a power supply to operate. Depending on the actual implementation, there are several ways to provide this power supply. In all cases, this has to be a 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 repair. Be sure therefore about the kind of power supply you connect the OC32 to.

3.2 GND or reference voltage

We assume that you have a connection point in your miniature world that we can call 0V, 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 will in any case be the "GND" of the communications port that your PC uses to control the layout.
The 0V/GND (as stated above) of your miniature world has to be connected to the GND connection of the OC32. All points in figure 5 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.

Concerning power supplies:
Model-railroad transformers seldom deliver a decent Direct Current. You can use them with success, however you will need some adaptations. More on that later on. The easiest way however is to use a separate DC power supply. You can buy them at electronic stores or from the VPEB partners. For about €20,= already you can buy a properly stabilized, configurable, switching and therefore economical DC power supply that can provide 2 Amperes or more.

3.3 Connecting the power supply

The power supply for your OC32 has to be a DC power supply with a Voltage level between 7V and 20V. The power needs to be smoothed but not necessarily stabilized. We advise to keep the voltage below 15V, since over 15V, you may experience noticeable heat dissipation from the power stabilizer on the OC32. Unless in cases explicitly mentioned in this manual there is little risk this causes of failure of the module though.

3.3.1 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 5)

If you leave the "Power" jumper JP1 on the module (as delivered ex-factory) the supply voltage as provided on K1 will become available on the 37-pin connector K5 on connections Vp (plus) and GND (min), so you can feed your connected devices with this.

3.3.2 Providing power through the 37p connector (K5)

You can do it the other way around as well: The "Power" jumper JP1 connects PWR with Vp. Therefore you can also choose to provide the supply voltage via the 37 pin connector K5 (GND=min, Vp=plus). In that case you can leave K1 unconnected. You will have all OC32 connections at one side, with the exception of the connections for communication.

This setup is especially convenient when you use a DS32.
In the case of connecting to K5 (so without DS32) it is advisable to wire all 5 connection points (interconnecting 3xGND, 2xVp) on that connector for an optimal connection and minimal resistance.

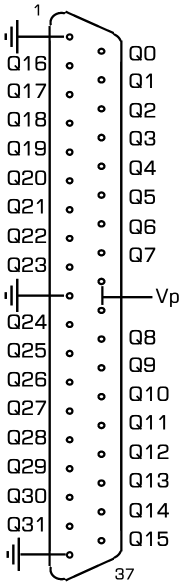

Fig 6: Pin-layout of K5
Attention: The module side of the connector is shown. This is the same as the backside of your connector. The front side (pins) of your connector is mirrored

3.3.3 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 18 V and the OC32 itself with e.g. 8V. 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 8V to 5V less energy will be dissipated than when the OC32 needs to reduce 18V to 5V

If you want this setup, you need to remove the "Power" jumper JP1, PWR and Vp will then be separated. The power supply for the OC32 itself you will deliver via K1 as described in section 3.3.1. The power source for the connected items is connected via K5 as described in section 3.3.2

3.4 Use of model railroad- and other transformers

3.4.1 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 2500µF 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!

3.4.2 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 figure6.

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 7 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 2500µF 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!

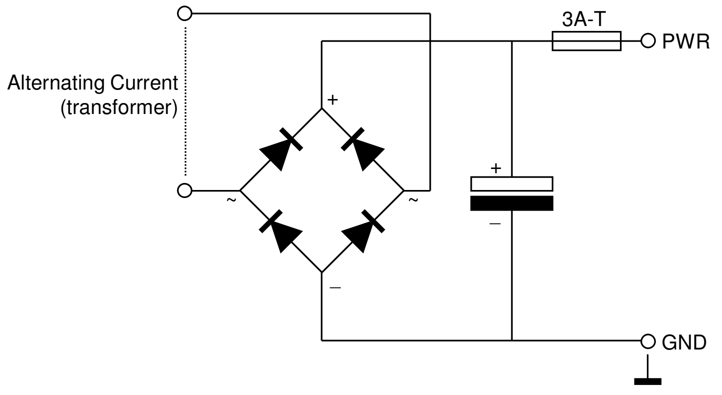

Fig 7: Rectification and smoothing AC supply

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.

3.5 Providing your own 5V

Internally, the OC32 operates on a supply voltage of 5V. The OC32 obtains this from the supply voltage delivered on K1 (PWR). The OC32 itself ensures an accurate stabilization. So you don't have to worry about that yourself.

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 K1-PWR is high (e.g. higher than 15V), then the stabilizer on the OC32 can become quite hot. **Only in these extreme cases** it can be advisable to provide an external 5V, especially if it is already present "somewhere". Basically this setup is meant for a configuration in which the OC32 is part of a Dinamo system that has an IPM as the 5V supply.

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, in figure 5 shown as "Power-jumper", must **NOT** be installed.
- The pin PWR on K1 must 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 lead to a serious and non-repairable defect in your OC32!

4 Communicating with the OC32

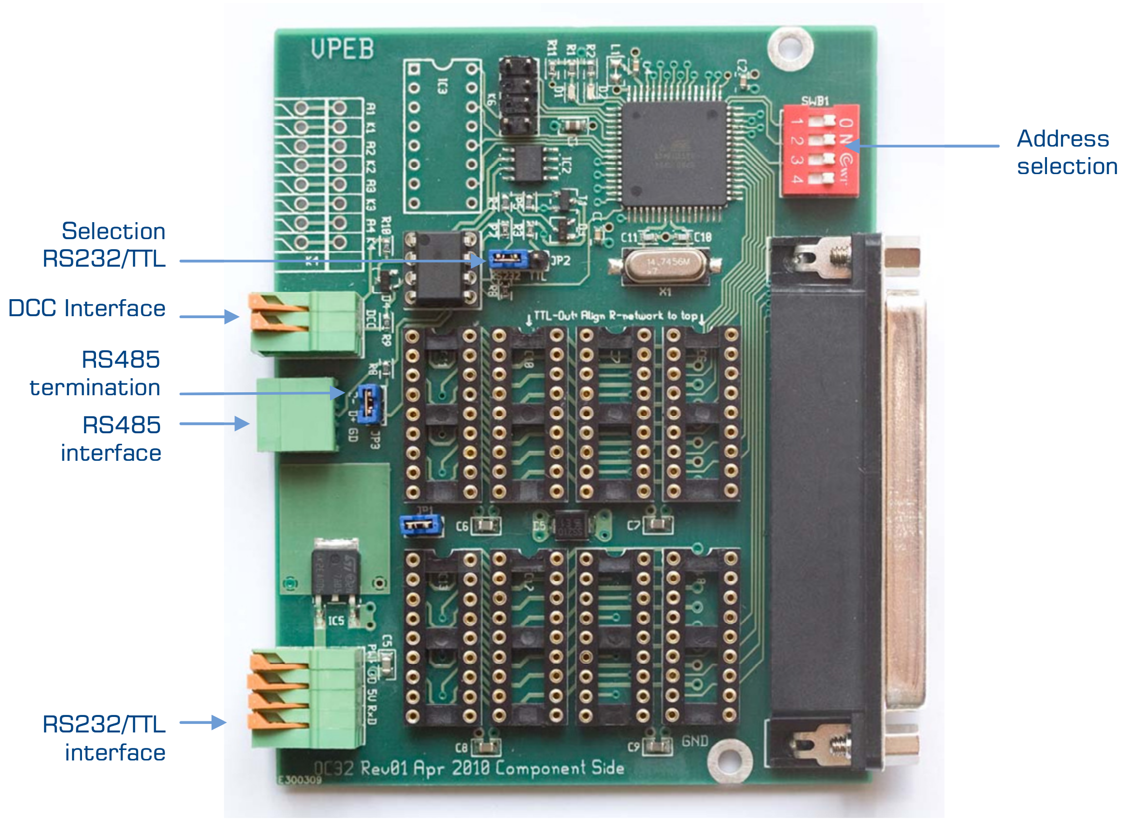

Fig 8: OC32 communication connections

4.1 Ways of communication

The OC32 has 3 communication-interfaces:

- an RS232/TTL communication channel
- an RS485 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/TTL 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/TTL 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 3 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.

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.

4.2 RS485 communication

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 is 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 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”.

In principle you require only one twisted pair, plus 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) is 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 and the UCCI/E controller. Information about this can be found in the documentation of the corresponding modules.

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.

If you have a com-port (RS232) available on your PC, then you can use that as well (see section 4.4). However, you will miss a few features.

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 K3 (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 setup will look like the one below (figure 9):

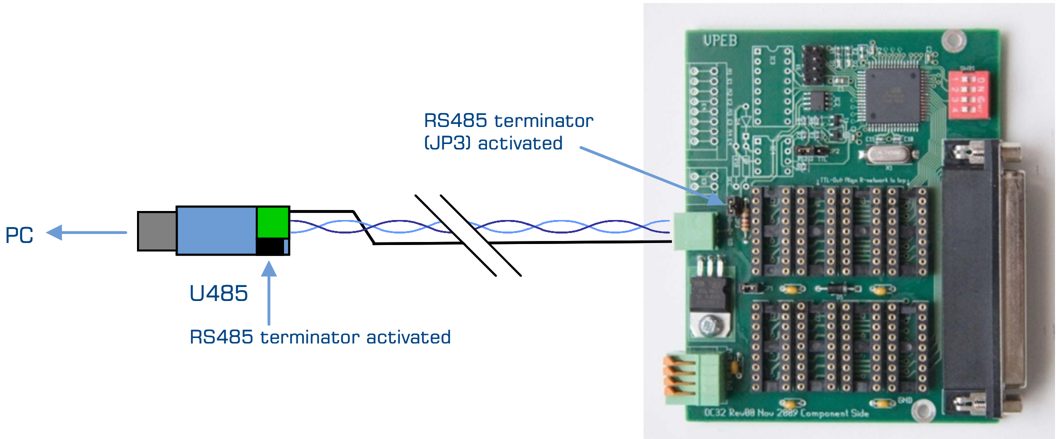

Fig 9: RS485 communication

4.3 TTL communication

TTL is only suitable for very short distances and only relevant if the OC32 module is part of a Dinamo system AND connected very close to an RM51, RM-H or RM-U controller.
Since TTL offers unidirectional communication to the OC32 only and the modern Dinamo and Dinamo/MCC systems don't offer TTL anymore, this option is considered outdated and not described any further in this manual.

4.4 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. RS232 is also provided by the older UCCI(/E) controllers and the RM-U.
NOTE: RS232 is not available on the latest UCCI/E Rev1x, the RM-U P&P or the RM-C.

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.

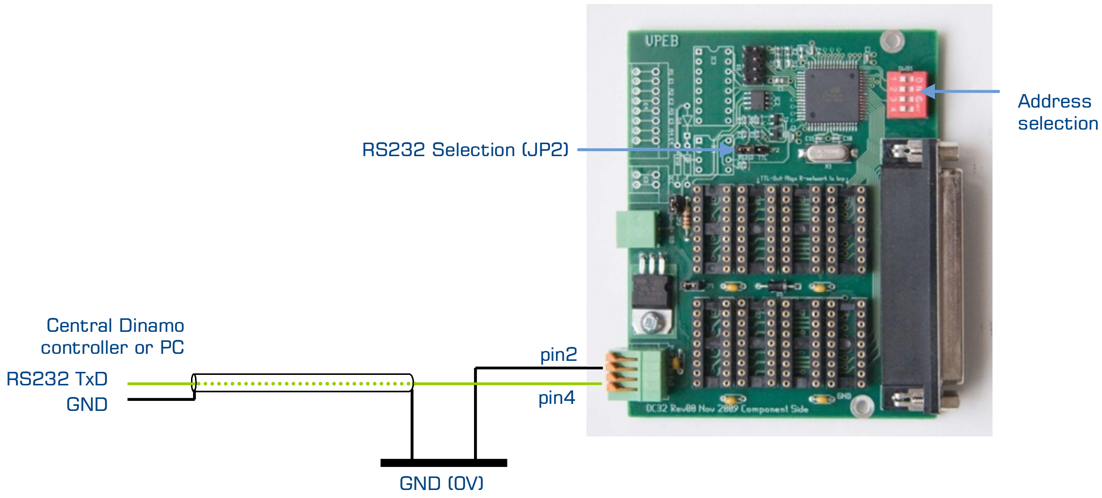

Fig 10: RS232 communication

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 10). See to it that jumper JP2 is set to "RS232".

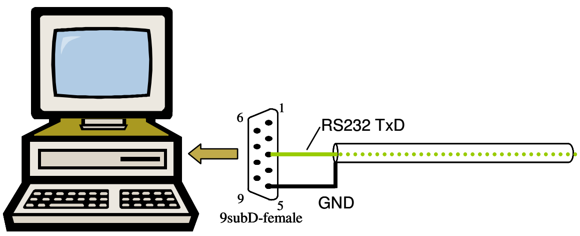

Fig 11: Connecting RS232 communication to the PC

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.

4.5 Connecting and addressing multiple OC32 modules

You can connect multiple OC32 modules to the same communication bus, with normal addressing up to 16, with extended addressing up to 96 modules.

4.5.1 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

4.5.2 Adressing (extended)

Extended addressing offers the possibility to connect in practice up to 96 modules[2] 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 configuration manual

If your control software does not support extended addressing, there are other possibilities to address more than 16 OC32 modules. However, configuration then becomes rather complex and not described in this manual.

4.5.3 RS485

The easiest way of connecting multiple modules to a single RS485 bus is to daisy-chain the modules. 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 RS485 terminator (jumper on the board) should only be activated on the first and last module on the bus;
- RS485 is a 3-wire connection. 2 wires transport the signal (D+ and D-), the third wire (GND) is for creating a reference voltage. If all your modules are on the same power supply and therefore are already interconnected via their GND, you don't need this third wire. Only if your PC and your "miniature world" don't have a connection via GND, you actually have to connect this third wire.

[2] In theory up to 1536 modules

The 3-pin connectors make daisy-chaining very 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 the first module.

It more or less will look like figure 12.

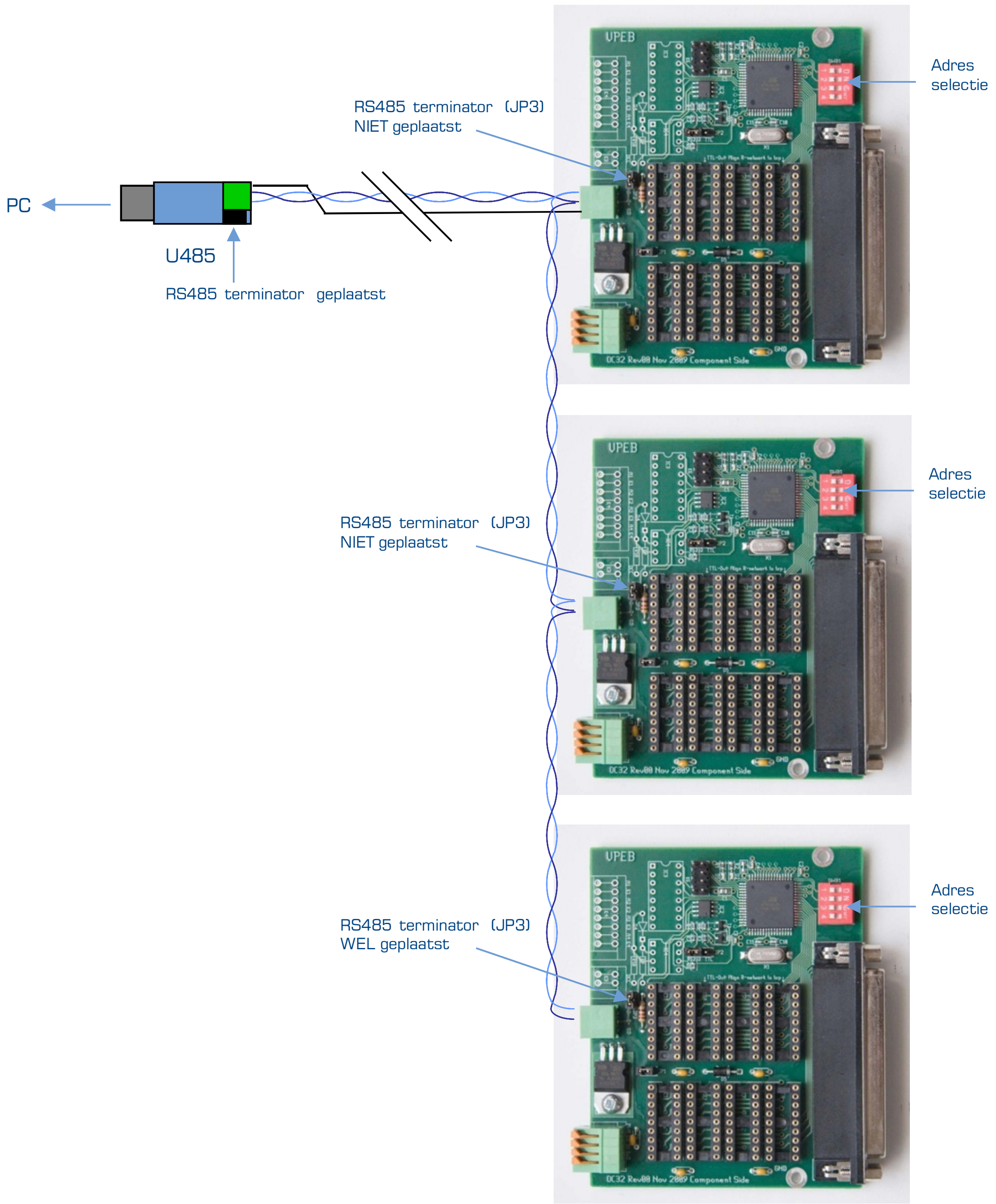

Fig 12: RS485 communication

For the record: In This description, the U485 counts as a 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 the last 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.

The power supply of all OC32 modules can be obtained from the same power source, provided that the power source can cope with the total requirement of all OC32 modules. Just interconnect the GND and PWR connections of all modules.
It is possible to use multiple power supplies, but in that case, ensure that the GND connections of all OC32 modules and all minus-poles of the power supplies are interconnected.

4.5.4 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.

4.5.5 RS232

If you have multiple OC32 modules on an RS232 bus, then you can connect them in parallel. Just interconnect the PWR (pin 1), GND (pin 2) and data (pin 4) of all modules. You can do that in a bus or star configuration.
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. If you want to connect more modules than your com-port can drive, you need to use a signal amplifier.

It is also possible to use multiple power supplies, but take care then to interconnect the GND connections of all modules and all minuses of the power supplies.

When using 5V power, you don't connect PWR, but 5V (pin3)

4.6 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 or not.

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 32 consecutive DCC accessory decoder addresses. The DCC start-address is set by means of software configuration (see configuration manual), so **NOT** with dip-switches.

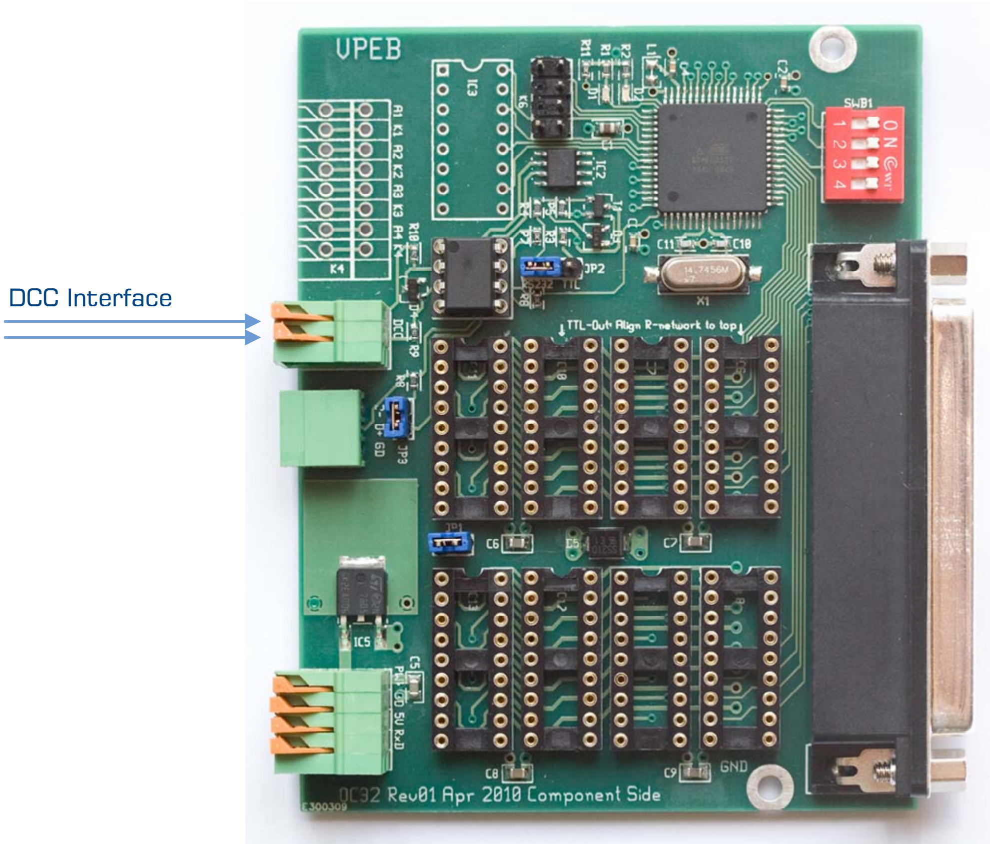

Fig 13: Connecting the DCC interface

5 Connecting the I/O Pins

5.1 Selecting the electrical characteristics

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

You have the following options:

- 500mA Sink Driver (ULN2803A): This is the default output-type;
- 350mA Source Driver (UDN2981A);
- Resistor-array (5V output with integrated resistor);
- Both Sink Driver (ULN2803A) and Source Driver (UDN2981A

The correct place for the drivers on the OC32 is depicted in fig 14.

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.

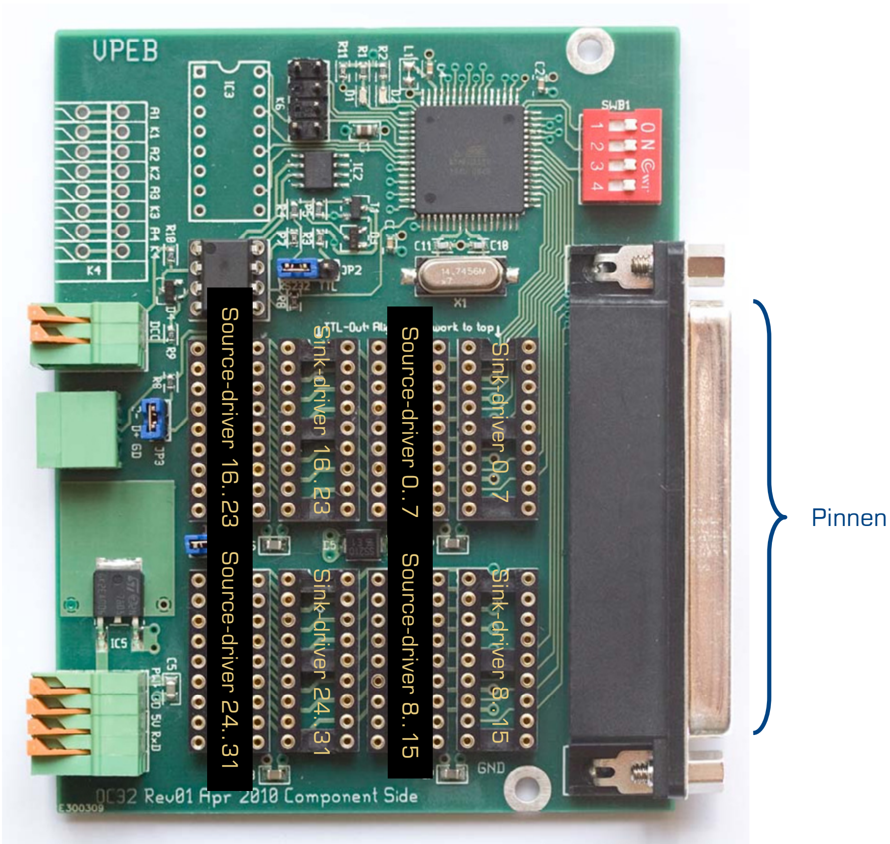

Fig 14: Locating the drivers on the OC32

Down below 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 5.4. There is explained which driver you need for specific types of devices.

5.1.1 Sink Drivers 500mA (ULN2803A)

This is the default driver delivered with the OC32 module. A load is connected between Vp and the 500mA output. The current flows from the positive voltage (Vp) through your load to the output and on the OC32 via the Sink Driver to GND/0V.

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.

5.1.2 Source Drivers 350mA (UDN2981A)

With a Source Driver the load is connected between the output and GND/0V. The current flows from positive voltage (Vp) via the Source Driver on the OC32 to the output of the OC32 and than via your load to GND/0V.

Each output can provide a peak current of 500mA for a very short period and a continuous current of 350mA 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.

5.1.3 Sink and Source Drivers (ULN2803 + UDN2981A)

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 aso the directon 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.

5.1.4 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 output of the OC32 processor is connected through a resistor to the equivalent output. 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 accidently 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. 1kΩ) 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 preferrably 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".

Should in this case a load, lets say a LED, needs to be connected between +5V and the output, the question is from where to get the 5V. If you provide 5V yourself as explained in section 3.5 it is obvious. Do you provide a raw power to pin 1 of K1 than a simple solution is available too. The internal stabilized 5V power for the OC32 is available on pin 3 of K1. This very well stabilized 5V can be used.

Take care what you do in these cases. Pin 3 of K1 is directly connected to the processor. If a higher or negative voltage is fed to this pin the processor will suffer severe damage.

5.2 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.

5.2.1 Multiple power-supplies using Sink-drivers

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 coils 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 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 by used to operate the OC32. To achieve that, PWR and Vp have to be connected separately. The power jumper JP1 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.

5.2.2 Multiple power-supplies using Source-drivers

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

When using both Source and Sink Drivers on the same OC32 it is possible to use different power supplies as described in 5.2.1 above, but **the highest voltage has to be connected to Vp, being the voltage used by the Source Driver.**

5.2.3 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.
Of course, in this case, you can also opt to power your OC32 with an external 5V, as described paragraph 3.5. Don't forget to remove Power Jumper JP1!

5.3 Connecting the I/O Pins

The loads to be driven by the OC32 are being connected to the 37 pole subD connector K5. The pin-out is given in figure 15.

Attention: shown here is the view of the connector on the OC32 module. This corresponds with the solder-side of the plug. Of course the front side view (pins) of the plug is mirrored.

Should you find it tedious to solder wires to the 37 pole connector, a connecting print with screw terminals is available. This print (DS32) can be connected to the 37 pole connector easily. A separate manual of the DS32 is available.

An additional advantage of the DS32 may be that you can use it to (optionally) can boost the peak-current of your OC32 to 4A per pin.

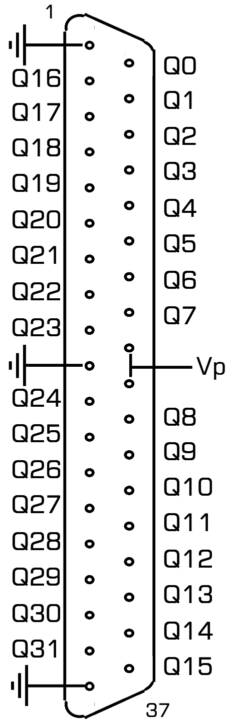

Fig 15: Pin-layout of K5
Attention: The module side of the connector is shown. This is the same as the backside of your connector. The front side (pins) of your connector is mirrored

5.4 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.

5.4.1 LED's with common anode (+)

OC32 I/O Pin: 500mA Sink Driver

Connect the common connection to the positive power terminal (Vp = 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.

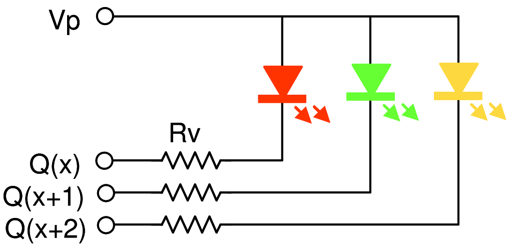

Fig 16: Connecting LEDs common.Anode

The value of the resistor depends on the voltage (Vp), 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 1,2kΩ applied by Viessmann at 14..16V results in to 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 1,8kΩ or 2,2kΩ 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.

5.4.2 LED's with common cathode (-)

OC32 I/O Pin: Resistor-array

Connect the common lead to the GND/0V. 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 220Ω. Low-current LEDs being used a higher value of the resistor array has to be applied.

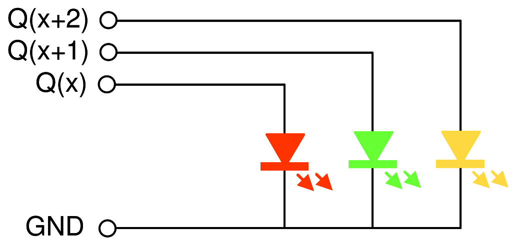

Fig 17: Connecting LEDs common Cathode

5.4.3 Bulbs

OC32 I/O Pin: 500mA Sink Driver

Bulbs have to be connected between the positive voltage (Vp) and the OC32 outputs. A resistor is not needed. The voltage (Vp) 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 Vp and the individual connections to outputs of the OC32. If bulbs have to be switched

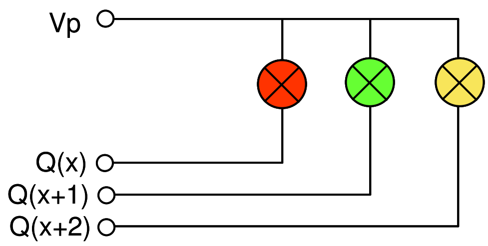

Fig 18: Connecting incand. bulbs

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

5.4.4 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 19) 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.

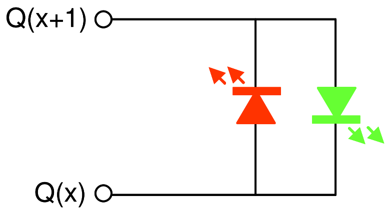

Fig 19: Connecting LEDs antiparallel in 5V mode

Should you have some spare 500mA outputs left, this will work also (figure20). 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.

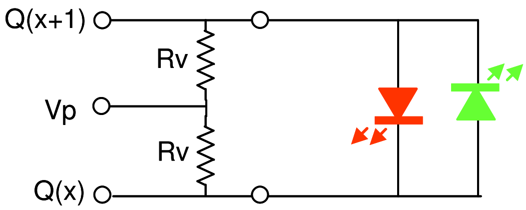

Fig 20: Connecting LEDs antiparallel in 500mA mode

5.4.5 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.

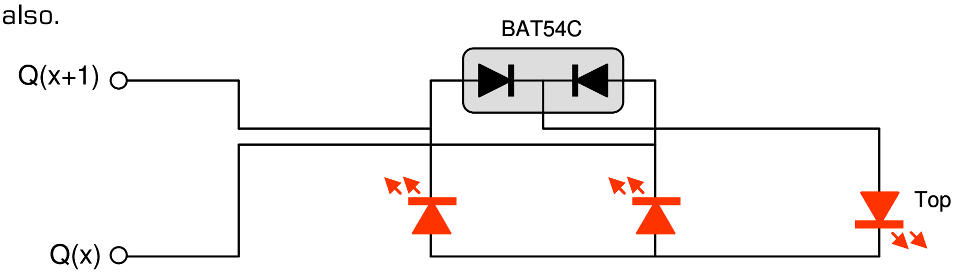

Fig 21: LEDs on the barriers (5V mode)

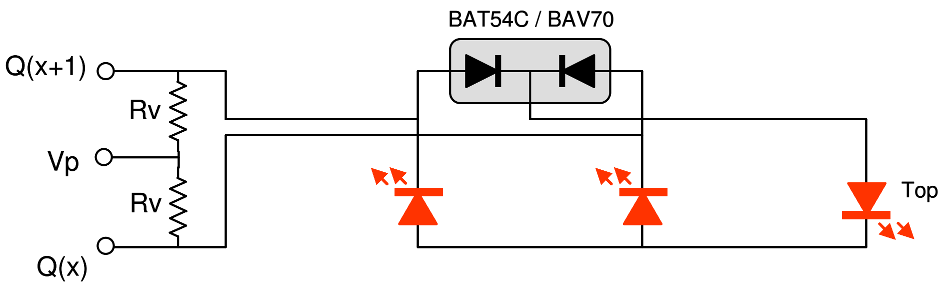

Fig 22: LEDs on the barriers (500mA mode)

5.4.6 Decouplers, turnout-coils

OC32 I/O Pin: 500mA Sink Driver

Solenoids may draw up to 1,5A (or more). That is far too much for the 500mA driver. But the current can be amplified to the required value per output with only 1 transistor. On the DS32 connection print it is possible to add amplifier transistors per output. Detailed description is to be found in the manual of the DS32.

If you prefer not to use the DS32, you can do it yourself. The base of each transistor is connected to an output of the OC32. The emitter goes to the driven component and the collectors of all transistors need to be connected to the GND/0V of your OC32/system.
Take care: the collectors handle the actual current of your loads so the wire has to be "thick" enough.
Furthermore it is important that Vp is connected to pin 28/29 of connector K5 (Vp).
The resistor of 2k2 diminishes the leaking current of the circuit when the output is switched off but is not essential.
The proposed transistors are only examples. You can select any normal PNP transistor (not darlington type) which can switch enough current.

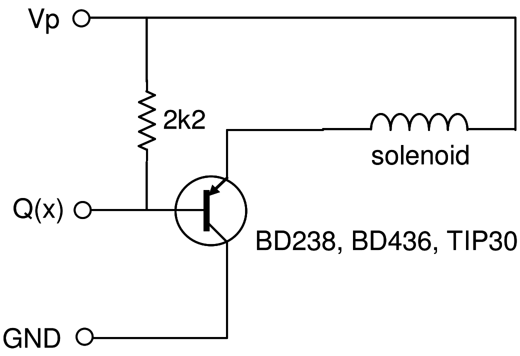

Fig 23: Increasing the output-current by a transistor

Connecting dual-solenoid turnout drive-units to the OC32 will cost you 2 outputs per drive-unit if done as described above. A very efficient way to control a large amount of these drive-units is by means of **turnout-multiplexing**. This method is described in a separate document.

5.4.7 Relays

OC32 I/O Pin: 500mA Sink Driver

Sometimes a galvanical 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

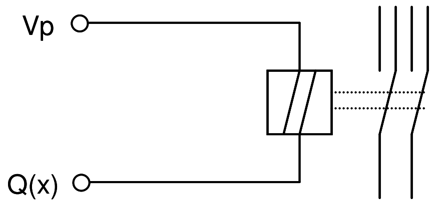

Fig 24: Connecting a relay

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.

5.4.8 Motors (unidirectional)

OC32 I/O Pin: 500mA 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 25.

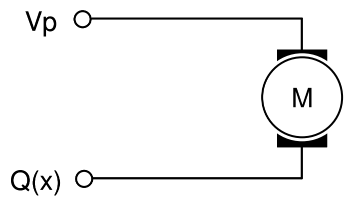

Fig 25: Conneting a motor

5.4.9 Motors (bidirectionall)

OC32 I/O Pin: Sink Driver + Source Driver

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.

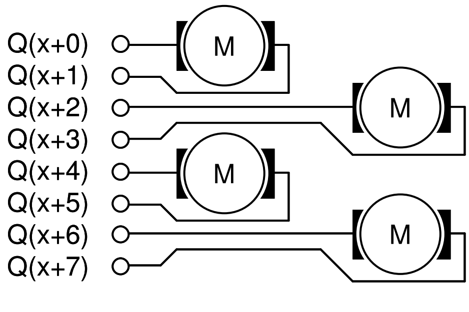

Fig 26: Connecting bidirectional motors

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.

5.4.10 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 27.

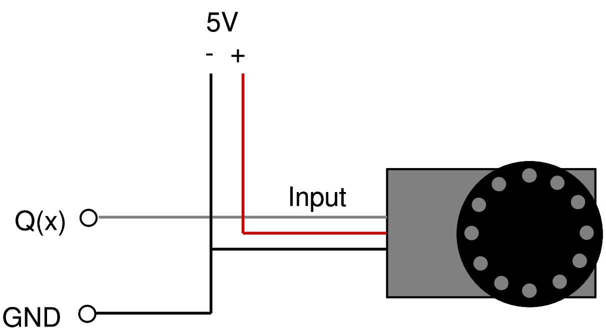

Fig 27: Connecting a servomotor

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.

To easily provide power to a servo and connect it to the OC32, a separate compact connection module has been developed, the SP04. With the SP04 up to 4 servos can be connected in a close range. When you have more servos or have servos at different locations, multiple SP04 modules can be used. A separate manual for the SP04 is available.

5.4.11 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.

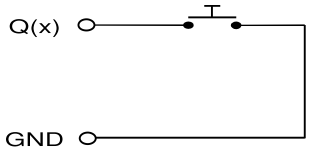

Fig 28: Connecting Pushbutton

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

5.4.12 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 5.4.11. 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 29 below. The resistor Rs depends on the output-voltage of your external system and can be found with the help of Table 2.

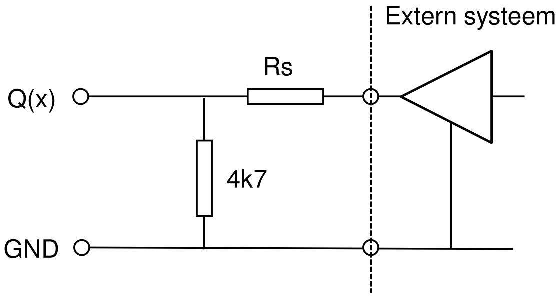

Fig 29: External system with Voltage conversion

Output Voltage (V)	Rs (kΩ)
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 5.4.11. However it is mandatory that in this case both systems share the same GND/0V. If this is not the case, you can insert an intermediate relay or an optocoupler between both systems.

6 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 wheter 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 jus tan 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/TTL interface on the OC32. The consequence is that, when using the SAP, the normal RS232/TTL 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 singal 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 to 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.

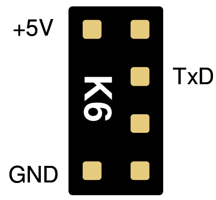

Fig 30: Pinout SAP

7 External Events

7.1 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 configuration 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 31. Each input port (In0..In3) has 2 terminals “A” and ”K”

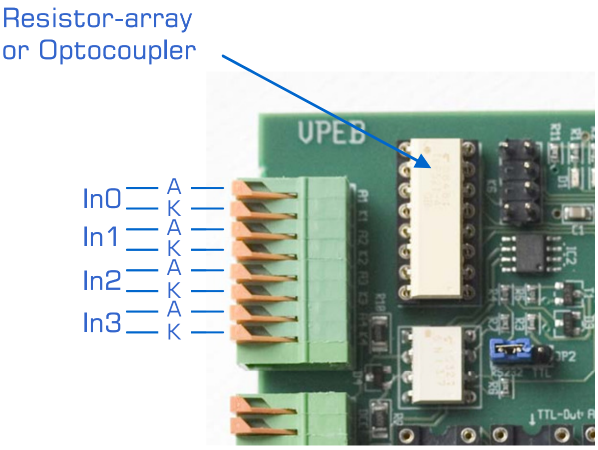

Fig 31: Optocoupler inputs

7.2 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 32 or figure 33. 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.

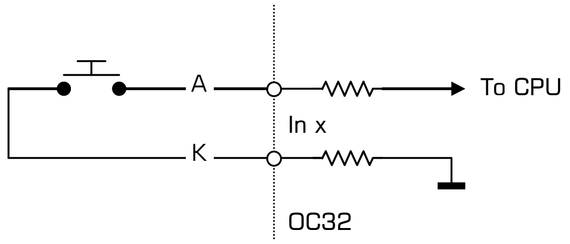

Fig 32: Pushbutton on Event Input (1)

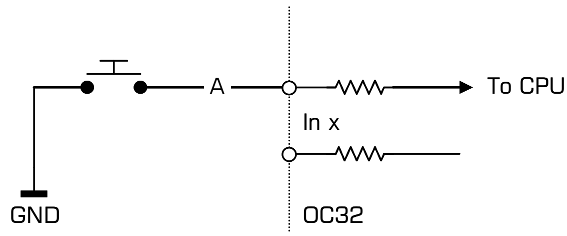

Fig 33: Pushbutton on Event Input (2)

7.3 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.

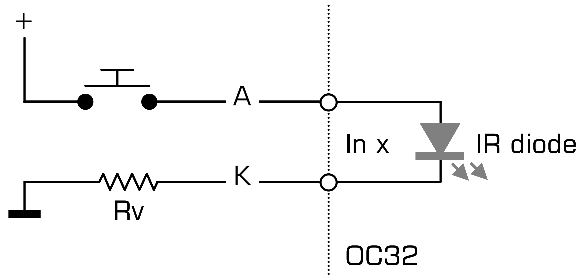

Fig 34: Connecting the Optocoupler input

If you find it difficult to calculate the series resistor Rv, please have a look in the FAQ on www.dinamousers.net

A simple calculation: Rv = (V - 2) / 5 gives the approximate value of Rv in kΩ 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 2kΩ for Rv (thus practically a 2k2 from the E12 series)

8 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)