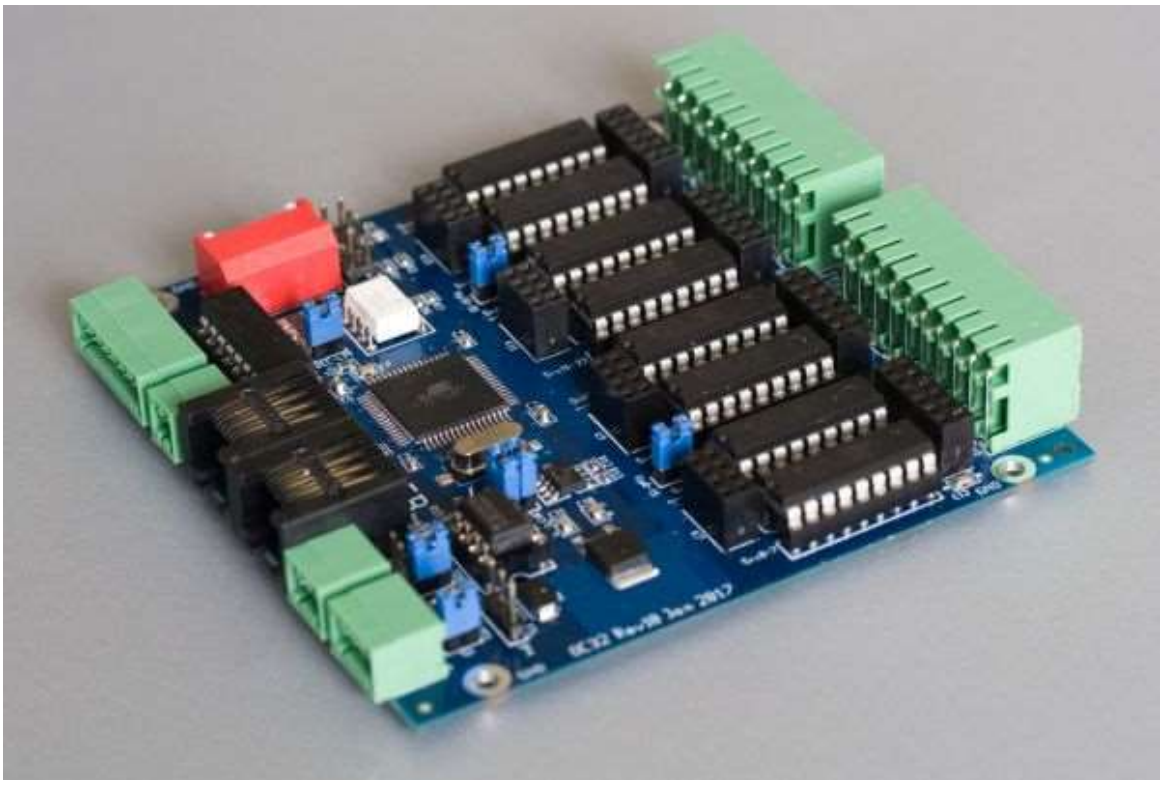
DTS Tutorial:
Adjusting servo control on OC32



OC32 is a product of VPEB

Content

Introduction

Servos are a very popular way to control switches, signals and other moving parts on the model railroad. They are not only cheap to purchase but also many times more reliable as, for example, magnetic coils. In addition, various moving effects and speeds can also be depicted with servos. Now the servo does have one major disadvantage, they need electronics to move. These electronics give the servo a middle position, the various positions and speed. Of all available electronics available in the model railroad world, the OC32 from VPEB is by far the most versatile. Without any restriction on the freedom of movement of the servo, the OC32 provides extremely stable control of almost all brands and types of servos. It does not matter whether the servo is analogue or digital.

I will explain how to connect the Servo to the OC32 in the OC32 Tutorial “Connecting the Servo to the OC32” which you can download on our website:
https://domburgtrainsupport.nl/informatie/handleidingen

In this Tutorial we are going to talk about setting a servo in OCConfig, the program with which you can configure the pins of the OC32.

If you have any questions or comments, you can post them by sending an email to info@domburgtrainsupport.nl

Best regards,
Martin Domburg

Analogue or digital servo's

I get that question more often; it does not matter for the OC32 whether you connect an analogue servo or a digital servo to the pin. But you do notice a lot physically, I have listed some advantages and disadvantages below:

Analogue Servo's

Pros

- ✓ Cheap and efficient
- ✓ Price between 1 and 5 euros
- ✓ You hear them moving

Cons:

- × More sensitive to malfunctions
- × Gears are poorer in quality
- × They can loose position with a mechanical resistance
- × Coarse movement, still nice and slow, but compared to digital, rather coarse

This does not mean that they are bad, with correct mounting of the servo and transmission, they will continue to function well for years. In our test set-up there are more than 160 analogue servos, in more than 6 years we have had to adjust a total of only 12 servos and 3 have been replaced because they became defective. The defects were all the Towerpro brand.

The best-functioning micro servos are the Turnigy TG9e and the Hextronics HXT900. The HXT900 is somewhat stronger than the TG9e and can better handle unexpected situations.

Digital servo's

Pros

- ✓ Very stable
- ✓ Less sensitive to malfunctions
- ✓ Very quiet
- ✓ Finer movements
- ✓ Very suitable for special effects

Cons

- × More expensive than analogue servos starting at just under 8 euros.
- × You do not hear them which is difficult during the adjustment
- × Less universally applicable as analogue servos that often have the same housing shape.

Digital servos have been overpriced for years for simple applications such as changing switches. However, since the beginning of 2019, servo manufacturer Turnigy has launched the TG9d. A Microservo in the same housing as its analogue brothers. And with an affordable price tag of € 7.95. From that moment the digital servos started gaining ground from the analogue servos. We have tested the Tg9d against several good-looking digital servos ranging between 12 and 25 euros. The Tg9d performed very well and often even better than the more expensive servos.

You can find the various servos in our webshop www.dtswebshop.nl

Terminology

The OC32 has quite a few terms that do not sound obvious to everyone. Here I try to create some clarity in the meaning of the terms. This can be useful to know when you start setting the servo.

Pin

A pin is a physical output from the OC32. This has 32 pins divided into four groups of eight pins. Each pin has its own address.

Aspect

Each Pin has several positions, within the OC32 we speak of aspects instead of positions. Each pin can have 4 or 12 aspects (positions).

Instruction

The software that operates the OC32 gives an instruction to an address (each pin has its own address) and thereby indicates which aspect needs to be switched. It is then useful if the OC32 then also knows what to do about the aspect of the pin. These are the instruction rules. The moment the aspect of that pin is controlled, the OC32 completes the instructions one by one from top to bottom.

Device Definition

Within the OC32 software we talk about an apparatus, which we call a "Device". In fact, a device is a collection of pins that together form something. For example, think of a 3-color signal. These are 3 pins that together form a device. The instructions in the Device Pin (the first and the master pin) control all pins associated with that device. The standard devices are collected in a Device Definition file which you can load with the “Reload DD” button. In the drop-down menu you can then load and use the desired Device.

Device

A device is a collection of pins that are required to control a device.

Device Pin

This is the main pin of a device; this pin is often also the first pin. This contains the instructions with which the corresponding pins are controlled.

Pin Offset

In the first frame of each instruction you can specify the P.O. The Pin Offset is nothing more than a transmission from the instruction in question to another pin on the OC32. A P.O. of 16 means, for example, that that instruction applies to the pin that is 16 outputs away. Suppose you control an aspect on pin 1, with an instruction with a P.O. with the value 10. Then the instruction on pin 11 is executed.

Range

Each servo has a range, an area in which it can move. It is not always necessary or desirable for the servo to use its complete range for the purpose required. This is highly dependent on the transmission from the servo to the part to be moved. In the OC32 it is possible to limit the range from S (small) to XL (Extra Large)).

Midpoint

A servo is driven with pulses. The OC32 can control these pulses in a value from -63 to +63. In the middle of this lies of course the value 0. This value 0 is physically not guaranteed to be the midpoint (midpoint). Every servo will deviate from this, but that can be corrected by shifting the midpoint. If you did not do this, the centre of control could be outside the range of a minus and plus value. For a good set value between the two positions, shifting the midpoint is essential, this is the position of the servo in which the object to be controlled is in the middle, not the servo itself.

Time & Base

This is a time setting to delay that instruction. Time displays the set time, and Base is the multiplication factor of this time. A handy tool is to double click on either the Time box or the Base box, a pop-up will follow with which you can reach a specific time with Time and Base for that instruction.

Init

This is the initial position of the pin when applying power to the OC32. That is, the value in that box represents the aspect that will be driven as soon as the OC32 is energized. A value of -1 tells the OC32 that it does not have to do anything until the pin is controlled. A value of 0 will adress Aspect 0 and so on…

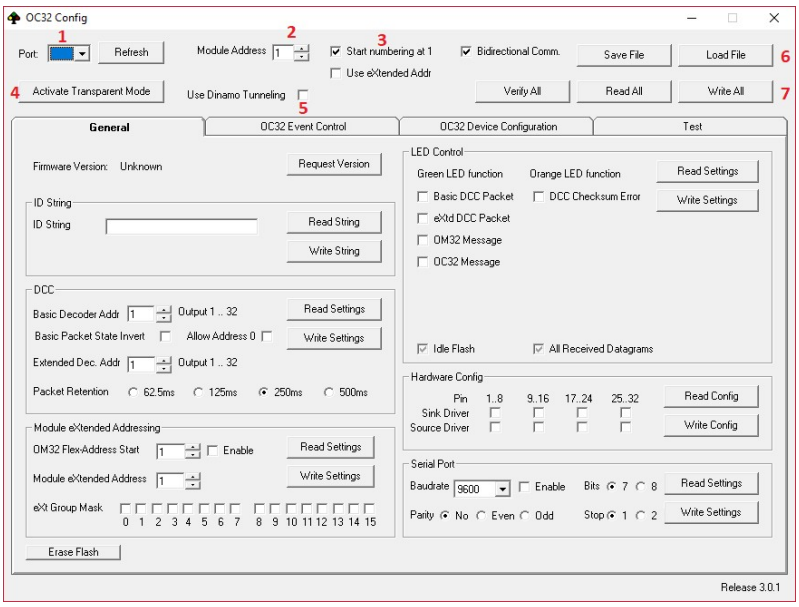
Starting up OCConfig

In this tutorial we assume that you have already installed OC Config on your computer and that you have connected the servo via SP04r in accordance with OC32 Tutorial “Connecting servo to the OC32” which you can download on our website: https://domburgtrainsupport.nl/information/manuals

To ensure proper operation and stability of the servo, it is important that you follow the instructions in the above-mentioned tutorial. Otherwise you may suffer from servo convulsions and malfunctions.

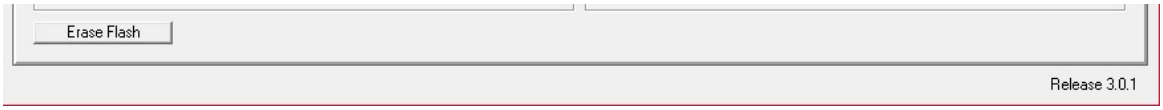
During the start-up of OCConfig you may get a well-known message: “Device Definition not found”, after you have clicked OK, the program opens the browser so that you can load a device Definition file. You can find these files on DinamoUsers.net where you can access the file gallery with customer status. Here you will find zip files with the devices. If you have already done this, select the "general" Definition file. It contains the settings for a servo.

After you have done this, the program opens in the General Tab



1. Comport number
2. Module address
3. Numbering according to software
4. Put Dinamo RM in TM
5. Use Dinamo Tunneling
6. Save / load settings
7. Flash options to the OC32

Now select the Comport to which your OC32 is connected. This can be a U485 usb converter or a Dinamo RM-U or RM-C. In the latter case, after selecting the Comport, click on Transparent mode to make the RM module invisible. Otherwise the PC cannot communicate with the OC32. Tunneling is also possible, but that is considerably slower than the TM mode. The red LED lights up on the RM after inserting it into TM mode.



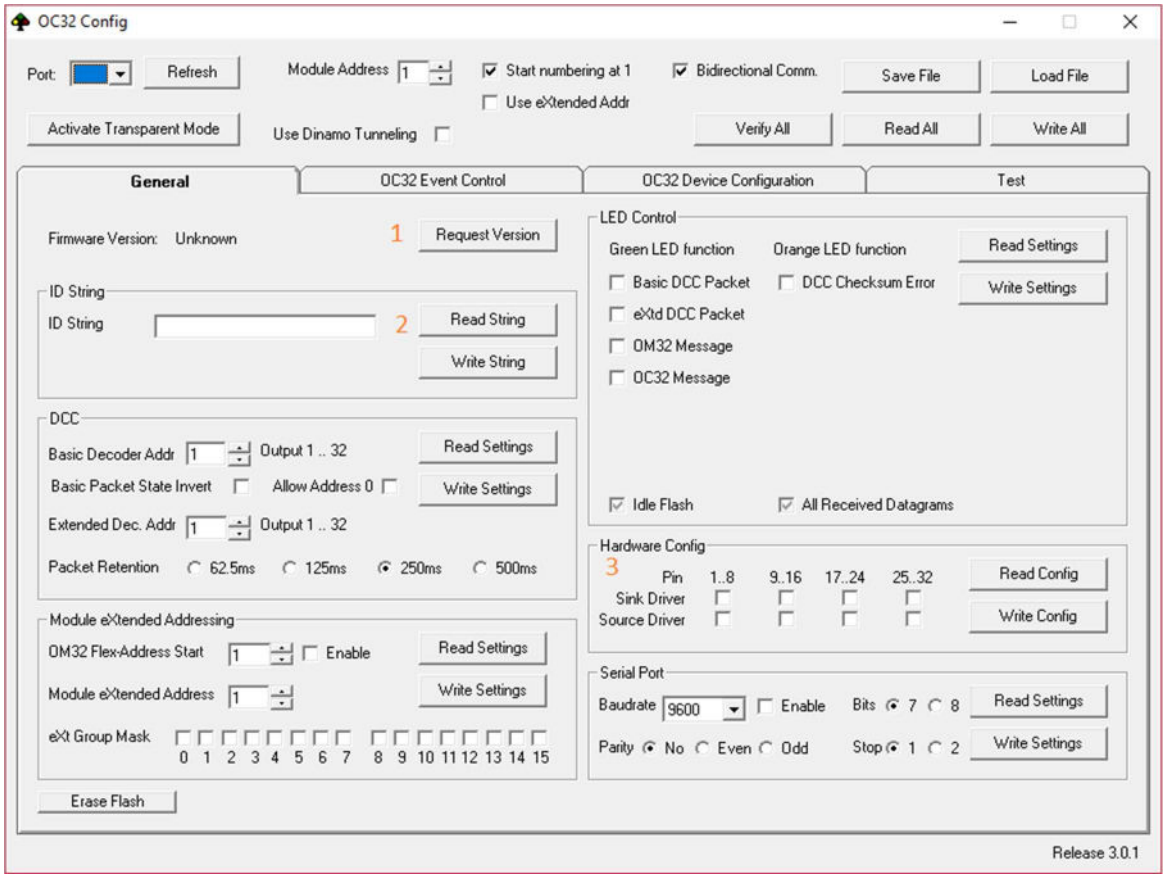
Below you will find two important points of attention:

Erase Flash: this will completely empty the OC32

Release: Always check that you are using the latest firmware and software for best performance. You can find the firmware and software on DinamoUsers.net, or you can contact us to help you with this.

General Settings

To begin with, we need to deal with this tab to ensure that you have set all important settings correctly. We only deal with the items that are important for this tutorial. All other settings can be found in the OC32 manual. This can be downloaded at www.vpeb.nl



The first point (1) of attention can be found at 1, the firmware version. Clicking on this will display the current firmware for the module that you have selected. If so then you have a good connection with the OC32. Check if you have the latest firmware in the OC32.

You can find the latest firmware and release notes on DinamoUsers.net. If you have problems updating the firmware, you can contact us for our DTS Update Service at www.domburgtrainsupport.nl

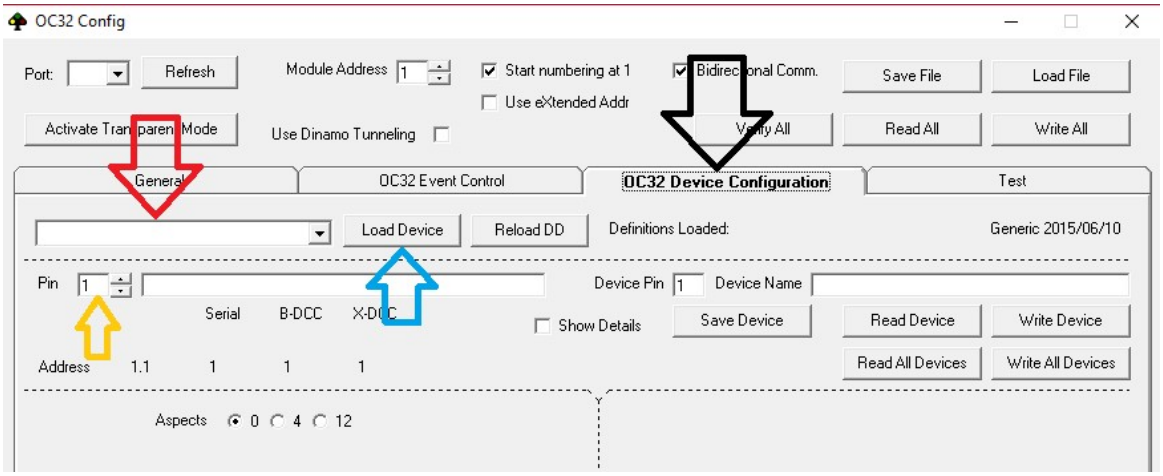
As a second point of interest (2) you can give the OC32 a name, this can be extremely useful if you want to see if you have connected the correct address to the correct OC.

The latter (3) is very important for correct operation. Here you indicate which drivers you have placed in the 4 benches. A resistance bank is used for servos. The indicated range indicates which of the four banks it concerns. Below each range you will see two check boxes, one for sink driver and one for source driver. With a resistance bank you leave both check boxes for that range empty, the OC32 then understands that no sink and no source driver has been placed, then there is only 1 option left: resistance bank.

For relays we usually use sink drivers, most relay modules such as the HPP4 use a joint plus, so we switch the minus via the OC32 pins. Unless you have relay modules with a common minus, then you use a source driver to switch the pluses via the OC32 pins.

Load device

Before we can start setting a servo, we first must load the servo at the correct pin. We assume that you have connected the servo to pin 1 (orange arrow).



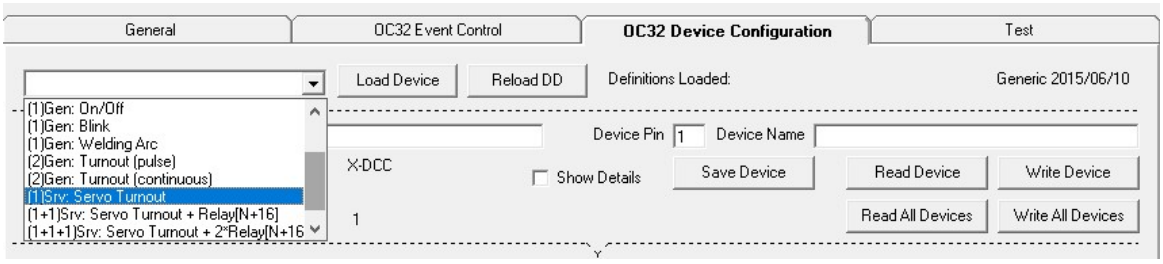
Step 1

Click on the “OC32 Device Configuration” tab according to the black arrow. You will see the screen above.

Step 2

You can load the desired device in the selection field at the red arrow. At the far right of this line you can see which file you have loaded. In this case the file “Generic 2015/06/10”.

If you click on the selection window you will get this drop down:



In this selection screen you will see all the Devices that have been collected in the Generic file that you loaded from your explorer during the start-up. You select the Device “Servo Turnout” selected above.

Explanation about the names:

You will see a number in parentheses for each device. This indicates how many pins that device takes up. In this case 1 pin, but below it you see the device “servo Turnout + Relay” with in brackets 1 + 1 which means that it concerns two pins that are distributed over the OC32. In this case, the second pin is 16 pins away and is controlled via a Pin Offset. If there is another number in brackets on a device without the "+" sign, the pins are grouped together.

If you have selected the device “Servo Turnout”, it will appear in the drop down window.

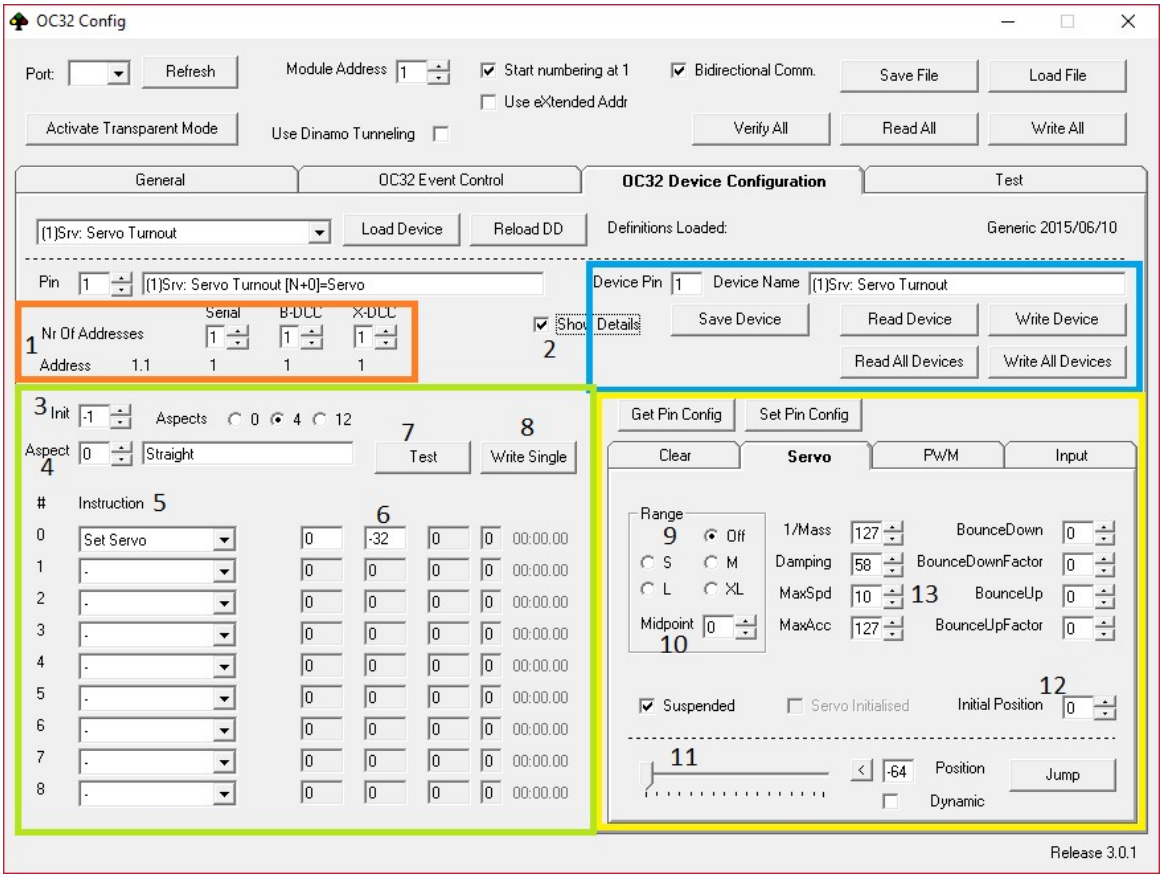
Step 3

Click on "load Device" (blue arrow), the device will then be loaded in pin 1. If you then tick the check box “Show Details” you will see various settings appear at the bottom. We will discuss this further in the next chapter.

The button next to "Load Device" is the "Reload DD" button. With this button you can select another existing or custom Device Definition file in your browser.

A tour around the main screen

What you see now will come across as magic at the beginning. But there is good logic in the layout. With this chapter we walk you through the different options of the Device.



If you look at the screen, the first thing that catches the eye is the red frame. In that part, the OC32 indicates the address of the pin. You can use this information to specify in software such as iTrain.

The Device displayed in the blue frame. These are the options in this context:

Device Pin is the master Pin, so the first pin that is used to control a device. All corresponding pins are addressed from this pin. Next to it you see the name of the device. It is possible to create a device yourself, give it a name and save it as a DD file with the save device button.

Read Device
Read the settings of this device, so it also reads the corresponding pins.

Write Device
Writes all settings from this device to the OC32, including the corresponding pins.

The “Read all devices” and “write all devices” buttons are self-explanatory, it then takes all the devices present at the same time. When you have finished setting, always use "Write Device". If you later adjust the OC32, always click on “Read all Devices” so that you are sure that you are editing the settings of the device that is stored in the OC32. So not with the options Load and Save File, the chance of a mistake is then too large.

Then we come to the two large fields, these are the most important fields. The green frame on the left shows the settings of the positions and its instructions. You will see the following:

3. The Init position for that Pin
4. The current aspect (status). To the right of it you can see that there are 4 aspects available for this pin.
5. The instructions for this aspect. Here you see the instruction “Set Servo”, with aspect 1 you will see that it has already been entered. The device Servo Turnout assumes a two-position switch with aspect 0 representing the position straight on, aspect 1 bend off the position.
6. The position to which the servo will be sent if this aspect of pin 1 is addressed.
7. After writing the device, the aspect can be tested with this button
8. This button can just as quickly save the changes in this aspect, is often faster than the entire device.

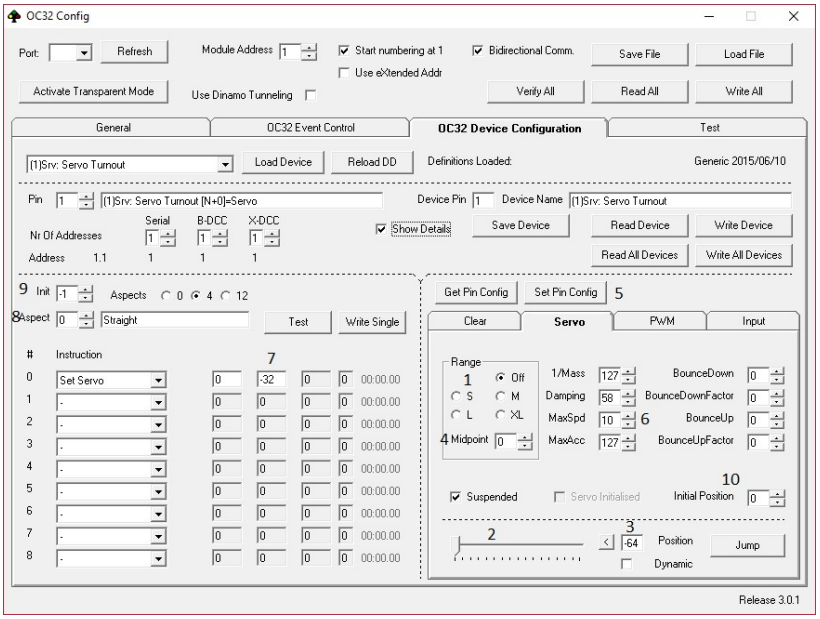
Finally, we see the yellow frame on the right, this is the PIN CONFIG. This specifies the behaviour of the pin. I have only marked the important items that we will need to handle a switch with a servo.

9. The Range in which the servo may be set
10. The middle position of the switch, or another device
11. The controller with which we can determine the position of the servo
12. The position that the pin takes when the Init is at -1. 0 stands for middle position (midpoint)
13. Speed with which the servo handles.

Configuring the Servo

It took a while, but the time has come, we are going to set servos. To get started, we first need to configure the pin behaviour. With the image below we show you what you need to do in several steps.

We assume a switch, but the steps are the same for an arm signal or a door, for example



1. Range
2. Controller to determine the position
3. Value of the position
4. Midpoint
5. Put the pin live for testing
6. Speed at which the servo moves
7. Position of the servo in that aspect
8. The selected mode (aspect)
9. Initial aspect
10. Value of Initial position at -1

Note, if a servo starts to hum then something is not right in the transmission, read the frequently asked questions later in the manual.

Roadmap:

1. Select a range
2. Click on “Set Pin Config”
3. Move the control to -63 and +63 and check that the servo gives enough change for the switch to change positions. Start with S
4. If the range is too small, repeat steps 1 to 3 until a range is found that is enough.

5. Set the controller to 0
6. Click Set Pin Config
7. Check whether the points with the tongues are free and, in the middle,
8. Adjust the Midpoint by changing the value positive or negative
9. Click Set Pin Config
10. Click on the controller in position 0
11. Repeat steps 8 to 10 until the midpoint has a value with the tongues at the centre of the switch.

12. Click Set Pin Config again
13. Now move the controller to a position that the switch will be in the straight-ahead position.
14. Take over the value of the position next to the controller
15. Enter this value in the Instruction 0 of Aspect 0 (the default value is -32)

16. Click Set Pin Config again
17. Now move the controller to a position that the turnout will be deflected.
18. Enter this value in the Instruction 0 of Aspect 1 (default is 32)

19. Click on “Write Device”

20. Click on the test button in aspect 0 and aspect 1 and check whether the servo correctly moves the switch as you had planned. If the servo does not, repeat steps 5 to 19.
21. Adjust the transfer speed by raising or lowering the MaxSpd.
22. Your servo is set correctly.

After each change you must click on “Write device” before you can use the test button. Later in this manual I will tell you how to apply the Suspend function to the servo, which we recommend to always use!

Addition 1:

The Init function is not important if you have switches with isolated frogs.

Assess whether you need to use the init function. This is not so important when using Dinamo. With DCC use, incorrectly setting the midpoint can result in a short-circuit if you polarize the frogs with a relay.

Addition 2:

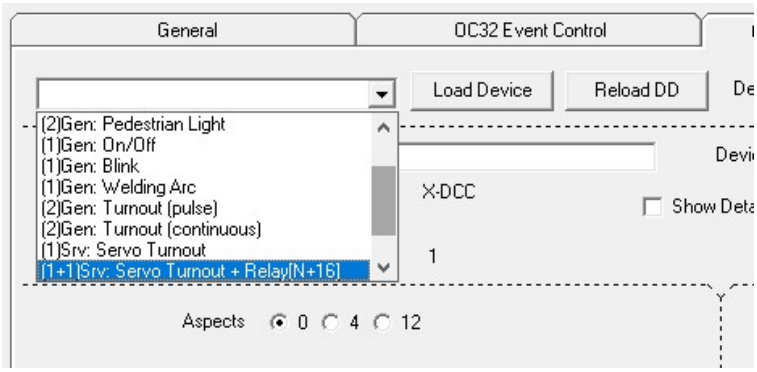
Within iTrain, enter a state delay of 2000 ms at the switch in addition to the address of the pin. Else it may occur that the train will start running before all changes have been made. A servo goes slower than a magnetic coil. Play with this value to get a nice synchronization between the switch in iTrain and the switch physically. In general, 2000 ms is simply enough.

Configurating a servo with relay for polarising

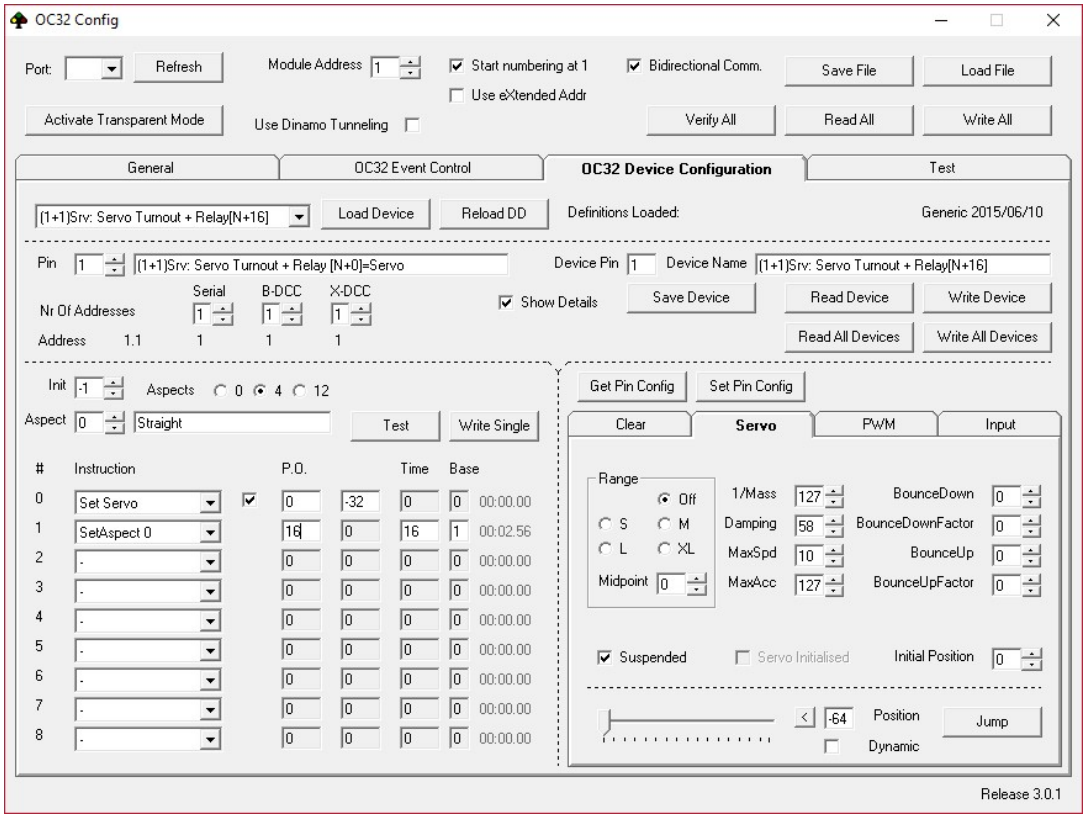
It is possible to connect a relay to the servo to set the polarity of the frogs to the correct polarity. This is done with a timing in controlling the relay. The HPP4 is designed as a relay for this purpose. Check our website for more information about the HPP4.

Because the OC32 offers the possibility to control 16 servos with 16 relays, the layout of the Device Definition is arranged so that you use the first 16 pins for a servo (resistance bank) and the last 16 pins for the relay (sink driver). This means that in the Device the pin of the servo is the master pin, and the pin to which the relay is connected has been shifted by 16 pins. (P.O. is standard at 16). In short, servo on pin 1, relay on pin 17 and so on…. This can be changed by the value at the P.O. (Pin Offset).

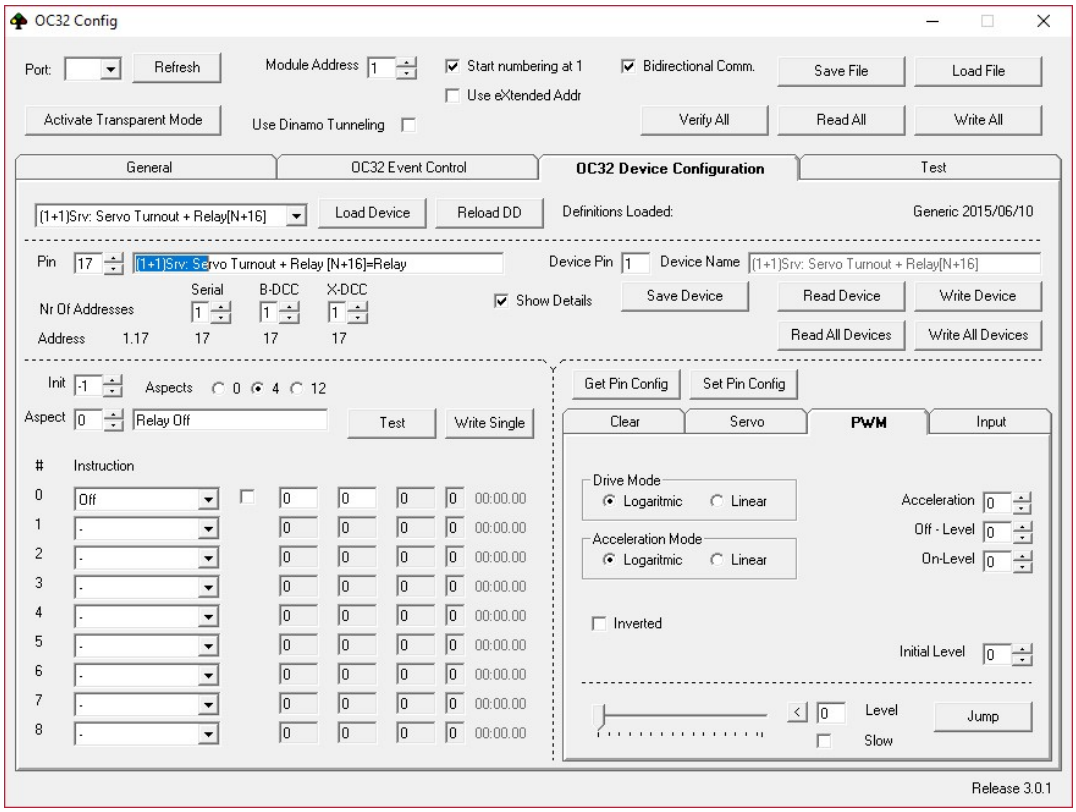
To start, we select a different device:



After selecting and loading the Device you will see these details appear in the pin that you used:



You can see that an instruction has been added with both aspects with a Pin Offset of 16 and a Time / Base delay. Now we will look 16 pins further on:



Here we see the relay; the Pin Config is also on PWM. This is because this pin does not control a servo but a pulse width modulation. You will also see that the Device Pin is set to "1". This indicates that this

pin component is part of a device that is controlled from pin 1. The program now writes the two pins together when you click on “write device”.

This also has two aspects On (Aspect 0) and OFF (Aspect 1))

The servo with relay is set as follows:

First, repeat all steps from the chapter "Setting the servo". After you have adjusted the servo completely and it handles nicely, we can time the relay. We have a step-by-step plan for this.

We assume the use of the HPP4, with each relay having a status indication LED. You can also tell by the relay whether it "clicks". You perform all steps in the master pin where the servo is located, in this case pin 1. We don't have to do anything at pin 17.

The steps after you have adjusted the servo:

1. Click on test in aspect 1 and check whether the correct relay is connected to which you have connected the polarization.
2. Click on test at aspect 0 and check whether the correct relay switches off again to which you have connected the polarization.
3. If this is not the case, you can adjust the P.O for each aspect. adjust, It is best to connect the relay to the correct pin (master pin + 16)

4. After this check, we click on test in aspect 0
5. Listen and see if the relay switches when both tongues are widely free from the rails.
6. Adjust the Time and Base for aspect 0, instruction 1 (Set Aspect 0).
    a. Double click on Time or Base gives a tool to determine the time
    b. Right next to Base you can see the set time in milliseconds
    c. On average, the value of Time is between 6 and 10 for a Base of 1
    d. the duration is partly determined by the speed of the servo
7. Click on “Write single”
8. Click on test at Aspect 1, if it is ready move on to step 9
9. Click on test at Aspect 0 and check whether the timing is correct.
10. Repeat steps 6 to 9 until the relay switches off when the tongues are properly clear of the rails.

11. Now it's time for the bending off position aspect 1
12. Listen and see if the relay switches when both tongues are widely free from the rails.
13. Adjust the Time and Base for aspect 1, instruction 1 (Set Aspect 1).
    e. Double click on Time or Base gives a tool to determine the time
    f. Right next to Base you can see the set time in milliseconds
    g. On average, the value of Time is between 6 and 10 for a Base of 1
    h. the duration is partly determined by the speed of the servo
14. Click on “Write single”
15. Click on test at Aspect 0, if it is ready move on to step 16
16. Click on test in Aspect 1 and check whether the timing is correct.
17. Repeat steps 13 to 16 until the relay switches on when the tongues are properly clear of the rails.

Your servo is now properly set up with a timed frog polarization relay.

For DCC use, we do need to check the Init position. If you do not do this, a closure may occur when the control panel is switched on because the tongues are not yet free. Certainly, if the power supply of the

OC32 switches on more slowly than that of the power station itself. If you have the tongues 100% free after the initialization has been completed, you will also receive a short-circuit and the control panel will no longer go into green until the malfunction has been resolved.

The safest way is to always set the INIT to aspect 0 when using DCC. Then the OC32 immediately sets the switch to the correct position when it starts up.

You can also choose to adjust the Init position in the Pin Config. You can correct your midpoint there during start-up. However, setting the Init to aspect 0 is still the neatest.

Adding Servo Suspend function

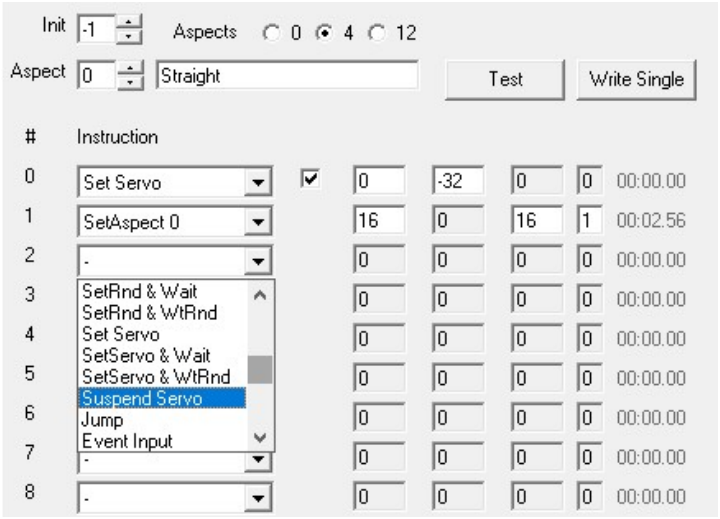
When a servo is ready and there is little to no mechanical resistance on the transmission, controlling the servo is no longer useful. The OC32 will continue to repeat the control out standard. The disadvantage of this is that, in addition to a constant power consumption, faster wear can occur on the cogs and the transmission. Especially if there is a slight mechanical resistance because the spring steel pushes against something. The servo also grumbles slightly. An additional advantage is that the servo is less sensitive to interference pulses if the control is not there, it reduces the chance of "convulsions".

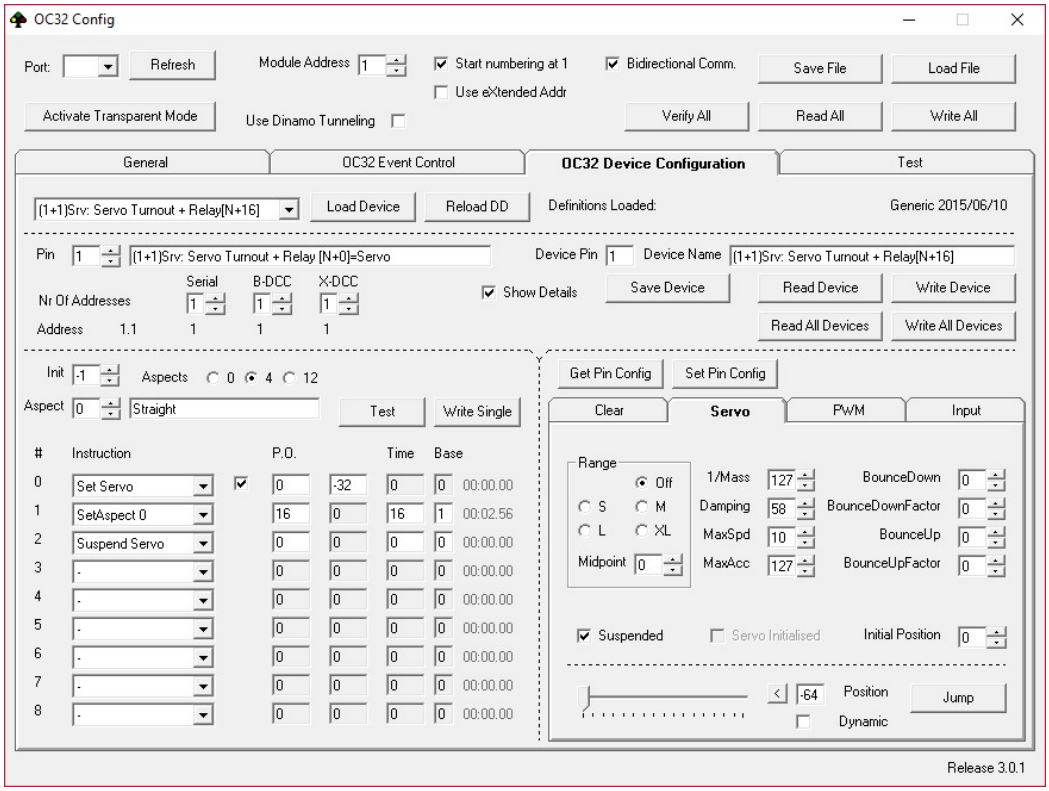
We always recommend adding the Suspend Servo instruction to all aspects of the Servo. It is a simple instruction that switches off the control of the servo based on timing.

These are the steps:

- Go to the instructions in aspect 0
- If you only control a servo, click on the drop-down window at instruction 1

If you also only control a relay, click on the drop-down window under instruction 2





1. Click on the Suspend Servo instruction
2. P.O. remain 0, we want to disable this pin, not another pin
3. Time and Base need a time. I select a time in the base which is double the timing of the relay. With only a servo I start with a time of 3.20 seconds.
4. Time = 20
5. Base = 1
6. Click on aspect 1
7. Repeat steps 2 to 5
8. Click Write Device

Now we only must test whether the Suspend function does not intervene too early or too late.

1. Select an aspect
2. Click Test
3. See if the switch goes all the way, if not (the tongues stay halfway) then increase the timing (Time / Base) of that aspect until it does
4. If the servo hum continues after the switch is completely moved, then lower the timing (time / Base)

You can easily experience it yourself afterwards. Without the Suspend function you cannot move the servo arm by hand after the servo has been activated. With the Suspend Function you feel that it switches off the control and you can move the arm again by hand.

Saving the Device as a pre-set

If you are ready, and you have the same switches and servos, it is smart to make a new Device out of this when you are done with the servo. You can then reload this device with every switch. The only thing you need to correct:

- ✓ Range
- ✓ Midpoint
- ✓ Position for aspect 0 and 1
- ✓ Timing of the relay in aspects 0 and 1

This is how you do this:

1. Change Device Name if desired
2. Click Save Device
3. Explorer opens, now give the definition a recognizable name
4. Click on save
5. Then click on “Reload DD” in the OC Config
6. Select your new Definition file
7. Choose the device by selecting it in the drop-down menu
8. Select the correct pin
9. Click on Load Device

On www.dtsportal.nl you will find several devices made by us for Peco code 55 switches and with the addition of the suspend functions.

FAQ

The midpoint value is not in the middle of the range.

This is because you visually determine the midpoint by looking at the switch. This does not automatically mean that the value of the midpoint is exactly between the value of aspect 0 and aspect 1. The most important thing is that the value of the midpoint is between these values and not outside it. If the switch is physically in the middle, that is enough.

The servo is humming

The servo experiences resistance in the transmission. This can be caused by the servo arm being unable to move freely, the spring steel towards the switch or the switch tongues themselves.

You will overcome this if it cannot be solved mechanically by applying the Suspend function.

The servo is not able to move the switch completely

This can be caused because the spring steel is impeded to complete the turn of the switch. For example, due to an obstacle or dirt between the switch tongues. Also, if the movement of the spring steel is not in line with the direction of the sleeper, the servo can twist the tug so that the tongues do not end up in their recesses.

When the power is switched on, my servos crash

A switch-on peak occurs when a power supply is switched on. This peak is such that the servos respond to this. On the SP04r there are components that reduce this effect, but it cannot prevent the servo from giving a little blow.

The convulsion should not cause any harm, the convulsion is so short that the movement is at most 1 mm. If it does cause damage, check the settings of the servo, you probably did not enter the midpoint correctly. Also check whether you have correctly applied the power supply and control (see our manual for this). You can also choose to use the Init position, then the servos are directly controlled to a position.

The servo is stressing randomly

A servo is sensitive, so we prescribe to carry out the cabling according to our manual. Also, do not place the cables near strong magnets such as contact wires or magnet coils.

The servo's stress extremely when I switch my power on

Then there is a problem with the Ground (V-) somewhere in the cabling. If it is loose somewhere between the power supply and the OC32, or between the OC32 and the SP04r, a potential difference can occur between the V + from the servo to the control signal. Turn everything off and check the cabling. If necessary, first disconnect the K5 plugs and check whether it is in the power supply or in one of the cables to the SP04r modules by plugging in the k5 plugs one by one.

What is the best way to mount a servo?

That does not really matter, if the movement of the servo arm results in a neat movement of the switch tongues with minimal resistance, that is just right. The distance of the spring steel between the servo and the switch is also not a condition.

Is there a minimum distance between servo and track?

Yes, there is, preferably a minimum of 2 centimetres of space between the switch and the servo. To guarantee this, DTS has developed an mdf servo bracket. You can also create the brackets to mount the servo yourself from metal or with a 3D printer. If you keep a space of 2 cm between the wood plate and the servo, you will not be bothered by the magnetism of the rails.

Epilogue

I wrote this tutorial for general personal use. You do not have to pay for this manual and it can be downloaded free of charge on our website. If you want to copy the text for private or club use, please contact us.

Domburg Train Support is an official partner of VPEB and an official reseller of the products. You can also contact Domburg Train Support for advice, support and help at home or via TeamViewer. If this manual does not work with the OC32, please contact us via our website.

I hope this tutorial will help you set up the OC32 in combination with servos. If you have any comments or remarks, please let me know. I can then process this in a new version. You can report this by sending an email to info@domburgtrainsupport.nl

Thank you for reading and using this manual.

Sincerely,
Martin Domburg