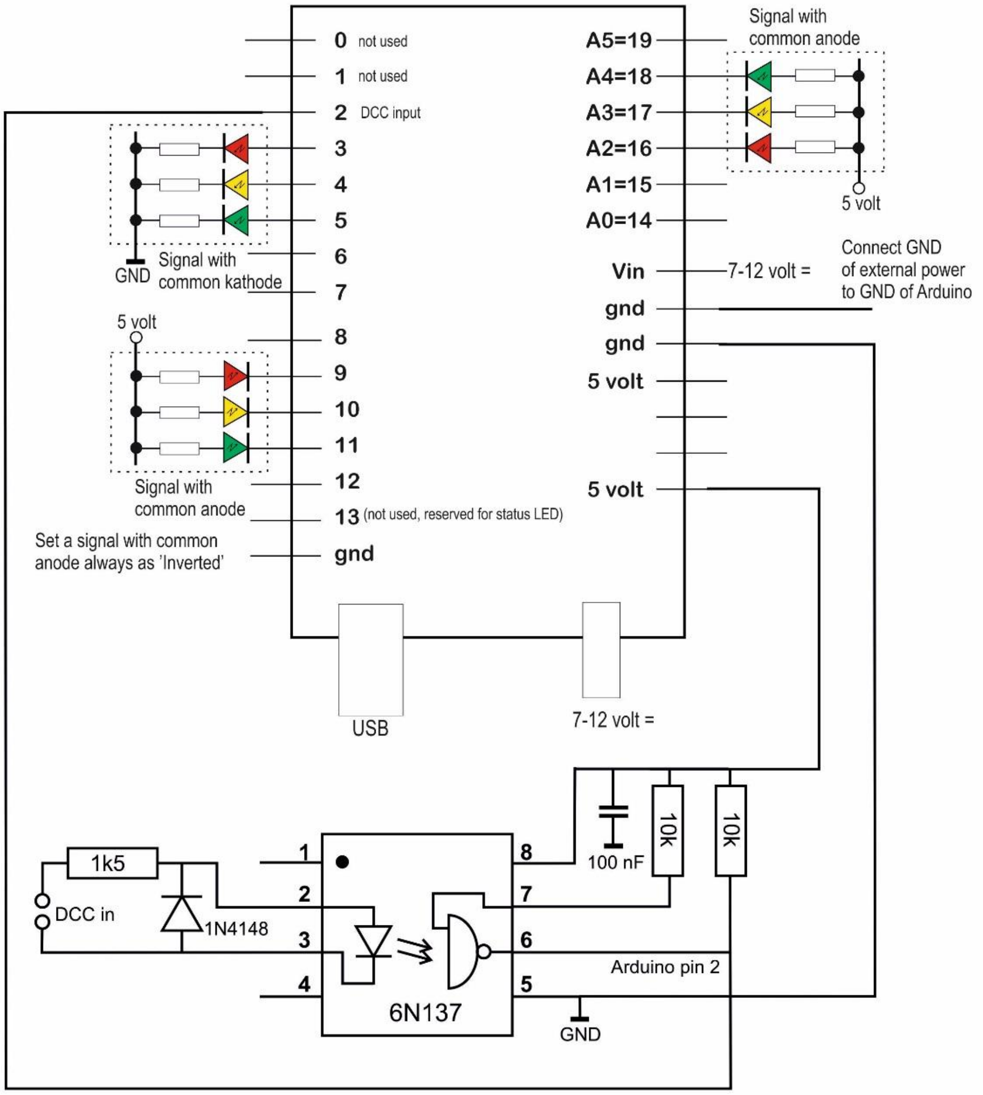
Schema DCCNext

The DCCNext has 16 outputs, which are connected to different pins of the Arduino.
A normal Arduino board contains an Atmel ATMega386 with pins of the chip, which are connected to pins on the board.
The drawing of the DCCNext is not complete and is just an example:



This picture shows three examples of connecting signals to the Arduino.
Power the Arduino via 7-12 V dc on the powerjack (advised) or Vin.
Or use the DCC/Power shield
Please note that each pin can source/sink a maximum of 40 mAmp!

Here are the pins of the DCCNext and pin numbers, the Arduino and the Atmel chip in case you might want to load your own program into the DCCNext::

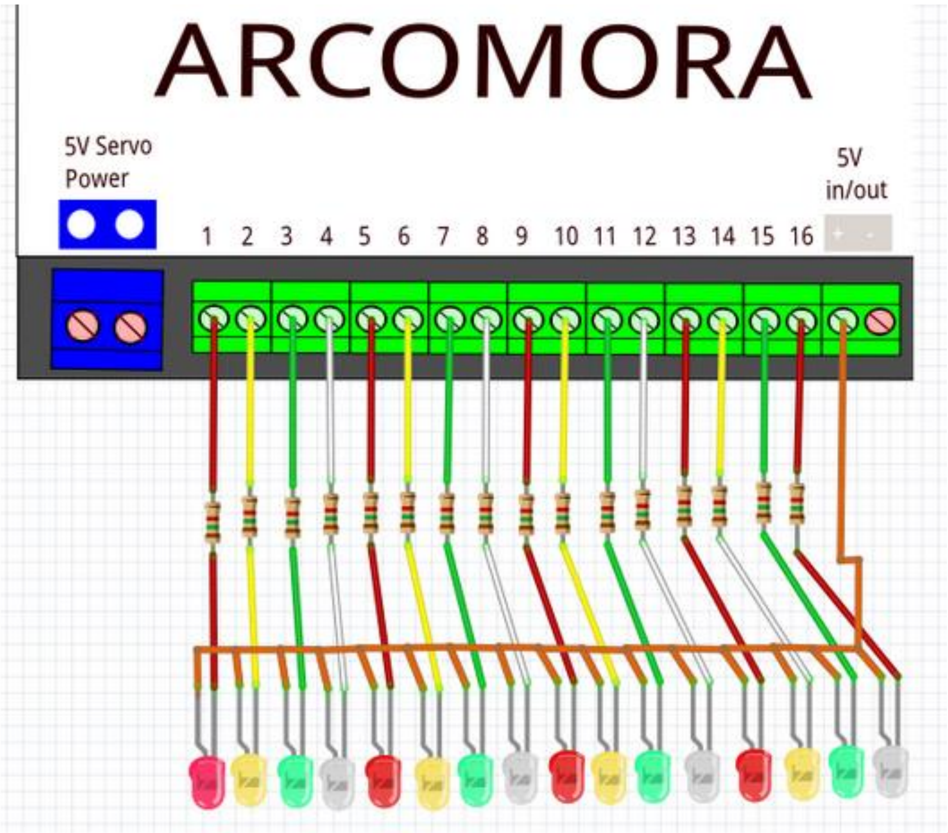
DCCNext	Arduino pin
	0 RX
	1 TX
	2 DCC in
1	3
2	4
3	5
4	6
5	7
6	8
7	9
8	10
9	11
10	12
	13
11	14/A0
12	15/A1
13	16/A2
14	17/A3
15	18/A4
16	19/A5

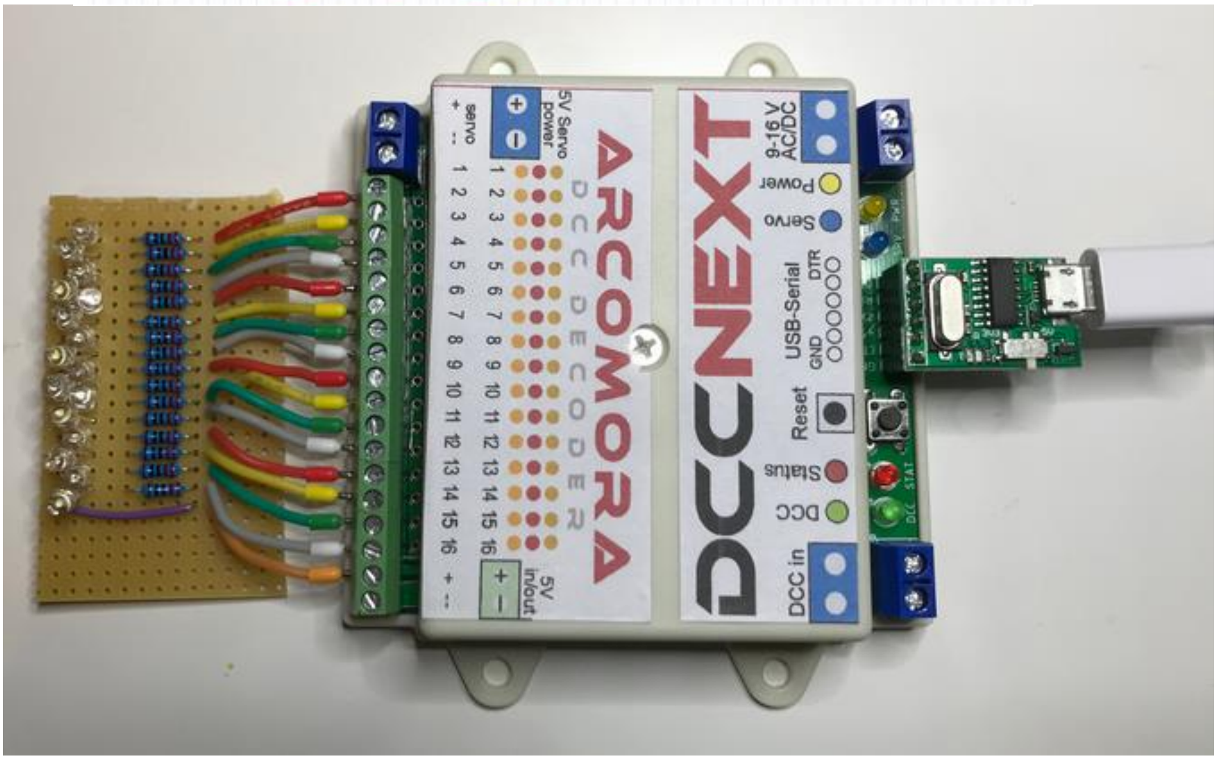
Assemblage

Assemble and solder neatly according to the manual.
Tip: stick a piece of transparent adhesive plastic over the stickers at the top.

Tester

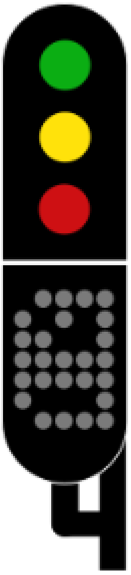
To test the DCCNext 4 red, 4 yellow, 4 green and 4 white LEDs were connected with an experimental PCB with corresponding wire and insulation terminal. Orange = 5V. The AC/DC of 16V was connected to DCC during this test.



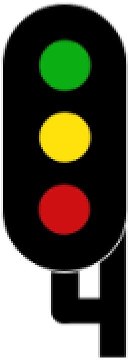


Configuring

With the ArSigDec manual of Arcomora a Dutch P signal with speed indicator and a P signal without speed indicator were configured



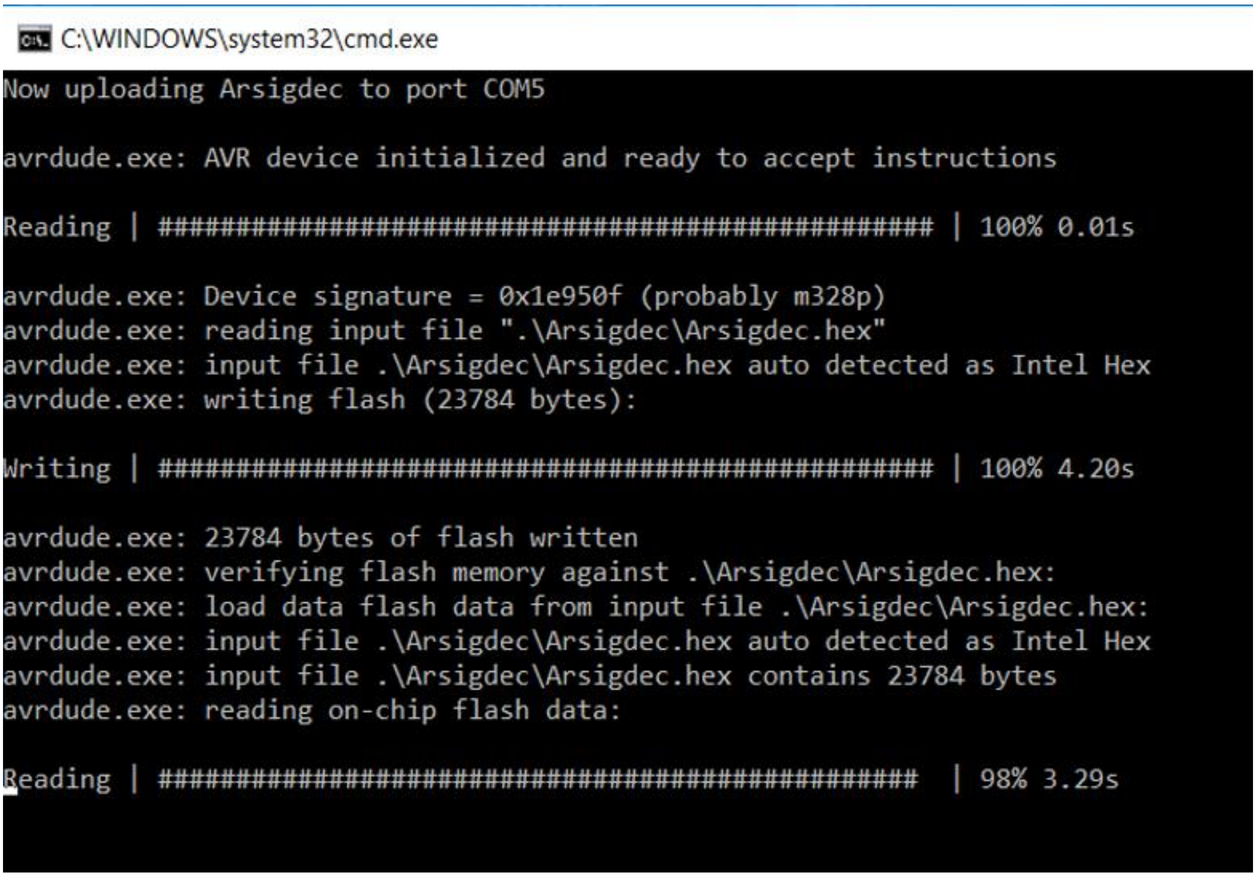
Adres 150 P signal with speed indicator. Adresses 150 and 151 each possible with a RED and GREEN command.



Adres 152 P signal without speed indicator. Adresses 152 and 153 each possible with a RED and GREEN command.

Upload

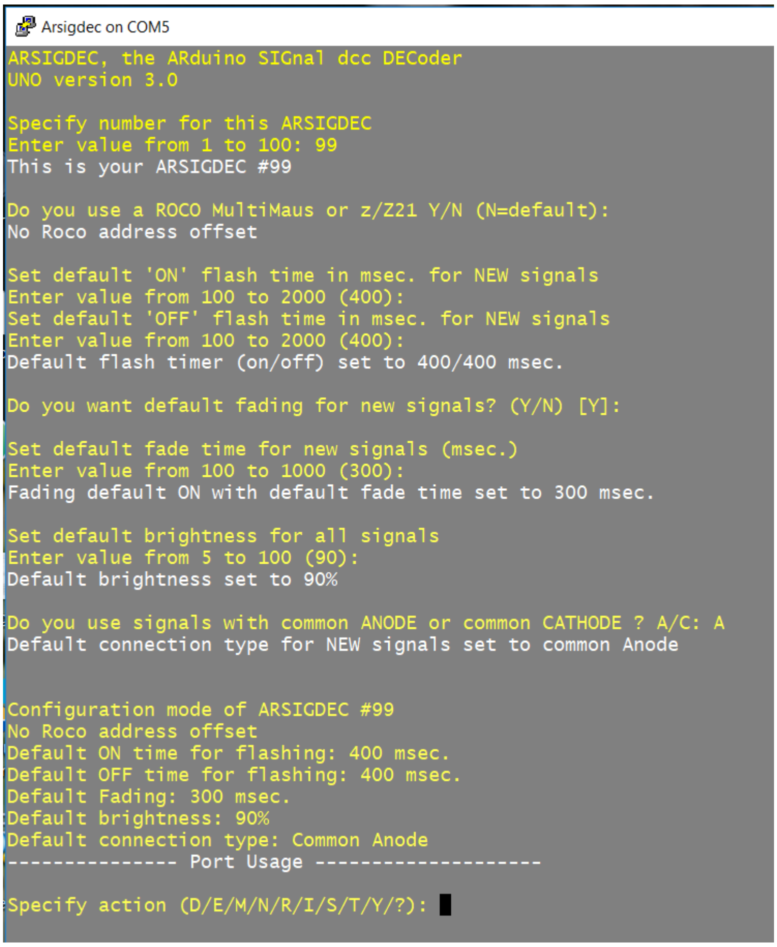
The first time a DCCNext is connected to a laptop the user can click shortcut Arsigdec and the program searches for the USB port. When succeeded the ArSigDec software is loaded into the DCCNext



```
Now uploading Arsigdec to port COM5

avrdude.exe: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.01s

avrdude.exe: Device signature = 0x1e950f (probably m328p)
avrdude.exe: reading input file ".\Arsigdec\Arsigdec.hex"
avrdude.exe: input file .\Arsigdec\Arsigdec.hex auto detected as Intel Hex
avrdude.exe: writing flash (23784 bytes):

Writing | ################################################## | 100% 4.20s

avrdude.exe: 23784 bytes of flash written
avrdude.exe: verifying flash memory against .\Arsigdec\Arsigdec.hex:
avrdude.exe: load data flash data from input file .\Arsigdec\Arsigdec.hex:
avrdude.exe: input file .\Arsigdec\Arsigdec.hex auto detected as Intel Hex
avrdude.exe: input file .\Arsigdec\Arsigdec.hex contains 23784 bytes
avrdude.exe: reading on-chip flash data:

Reading | #################################################  | 98% 3.29s
```

First configuration

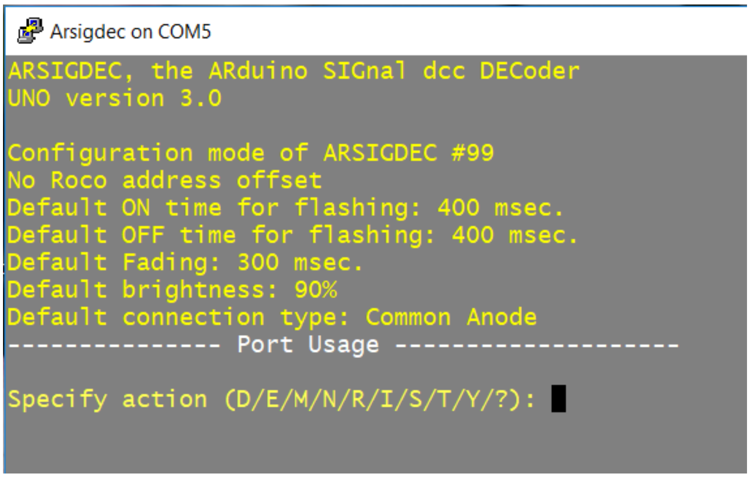
The first time after reinstall some parameters must be set.



```
ARSIGDEC, the ARduino SIGnal dcc DECoder
UNO version 3.0

Specify number for this ARSIGDEC
Enter value from 1 to 100: 99
This is your ARSIGDEC #99

Do you use a ROCO MultiMaus or z/Z21 Y/N (N=default):
No Roco address offset

Set default 'ON' flash time in msec. for NEW signals
Enter value from 100 to 2000 (400):
Set default 'OFF' flash time in msec. for NEW signals
Enter value from 100 to 2000 (400):
Default flash timer (on/off) set to 400/400 msec.

Do you want default fading for new signals? (Y/N) [Y]:

Set default fade time for new signals (msec.)
Enter value from 100 to 1000 (300):
Fading default ON with default fade time set to 300 msec.

Set default brightness for all signals
Enter value from 5 to 100 (90):
Default brightness set to 90%

Do you use signals with common ANODE or common CATHODE ? A/C: A
Default connection type for NEW signals set to common Anode


Configuration mode of ARSIGDEC #99
No Roco address offset
Default ON time for flashing: 400 msec.
Default OFF time for flashing: 400 msec.
Default Fading: 300 msec.
Default brightness: 90%
Default connection type: Common Anode
--------------- Port Usage --------------------

Specify action (D/E/M/N/R/I/S/T/Y/?):
```

Reropen

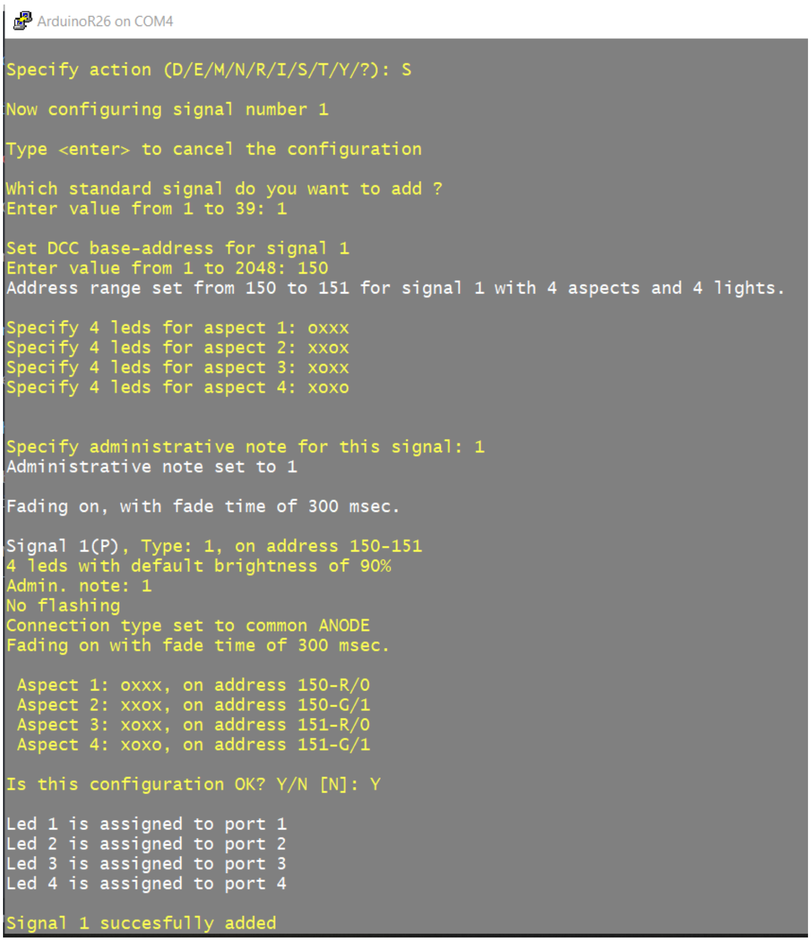
A short welcome is shown when connecting the DCCNext to a laptop after first install:



```
ARSIGDEC, the ARduino SIGnal dcc DECoder
UNO version 3.0

Configuration mode of ARSIGDEC #99
No Roco address offset
Default ON time for flashing: 400 msec.
Default OFF time for flashing: 400 msec.
Default Fading: 300 msec.
Default brightness: 90%
Default connection type: Common Anode
--------------- Port Usage --------------------

Specify action (D/E/M/N/R/I/S/T/Y/?):
```

Configure standard signal

First a Dutch P-signal with number box (database number 1) is selected at DCC start address 150:



```
Specify action (D/E/M/N/R/I/S/T/Y/?): S

Now configuring signal number 1

Type <enter> to cancel the configuration

Which standard signal do you want to add ?
Enter value from 1 to 39: 1

Set DCC base-address for signal 1
Enter value from 1 to 2048: 150
Address range set from 150 to 151 for signal 1 with 4 aspects and 4 lights.

Specify 4 leds for aspect 1: oxxx
Specify 4 leds for aspect 2: xxox
Specify 4 leds for aspect 3: xoxx
Specify 4 leds for aspect 4: xoxo


Specify administrative note for this signal: 1
Administrative note set to 1

Fading on, with fade time of 300 msec.

Signal 1(P), Type: 1, on address 150-151
4 leds with default brightness of 90%
Admin. note: 1
No flashing
Connection type set to common ANODE
Fading on with fade time of 300 msec.

 Aspect 1: oxxx, on address 150-R/0
 Aspect 2: xxox, on address 150-G/1
 Aspect 3: xoxx, on address 151-R/0
 Aspect 4: xoxo, on address 151-G/1

Is this configuration OK? Y/N [N]: Y

Led 1 is assigned to port 1
Led 2 is assigned to port 2
Led 3 is assigned to port 3
Led 4 is assigned to port 4

Signal 1 succesfully added
```

Likewise, a Dutch P-signal without number box (database number 2) was selected at DCC start address 152.

With a T command the LEDs can be tested from the laptop. A small disadvantage (probably because everything in the test setup is common anode) is that all LED's will light up immediately. With 16 test LEDs on one Atmel386 in the board, this is too much current. First disconnect a few LED connections or immediately configure the maximum number of signals within the DCCNext. In the latter case the DCCNext will extinguish almost all signals after typing an E (Exit to operation mode)..

DCC commands to aspect

To test the configuration, the DCC Command Centre (e.g. DR5000) and/or MultiMaus RED and GREEN commands are first sent to the configured addresses to check which commands are required for which signal aspect.

Signal 1, 4 aspects, database number 1

Command	**aspect**
150-R	To red, if before green then via yellow
150-G	To green, if before red then via yellow
151-R	To Yellow
151-G	To yellow + White

Signal 2, 3 aspects, database number 2

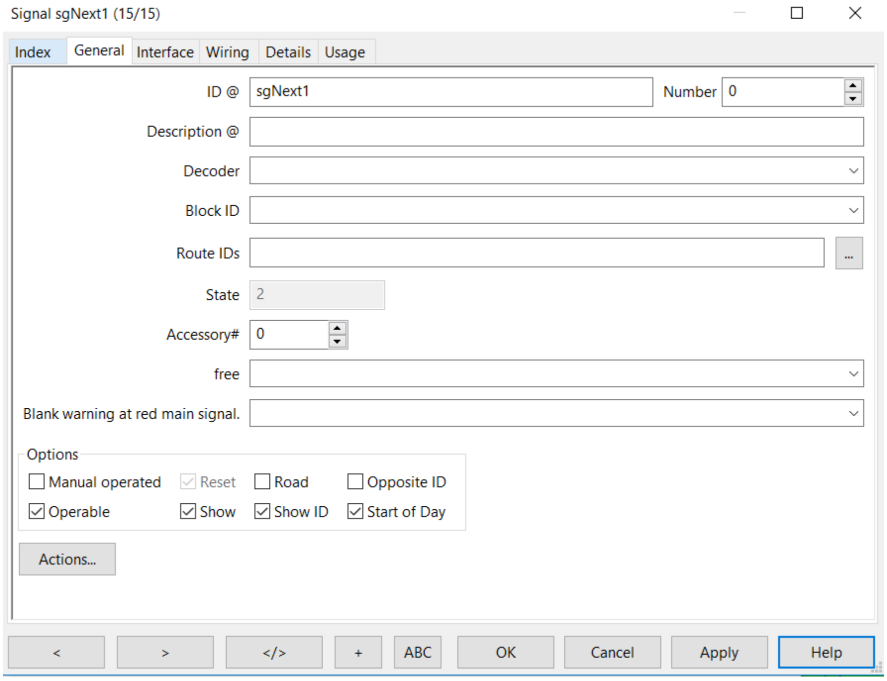
Command	**aspect**
152-R	To green, if before red then via yellow
152-G	To green, if before red then via yellow
153-R	To Yellow
153-G	Nothing

This is all in accordance with the manual.

RocRail configuration P-sein

The RED and GREEN commands seen above must now come from RocRail due to a correct configuration. On the Arcomora site, reference is made to a video explaining that for the DCCNext the address method 'Linear' must be chosen. The changes with respect to the default settings are circled.

Create a signal in RocRail with the name sgNext1, for example:



Signal sgNext1 (15/15)

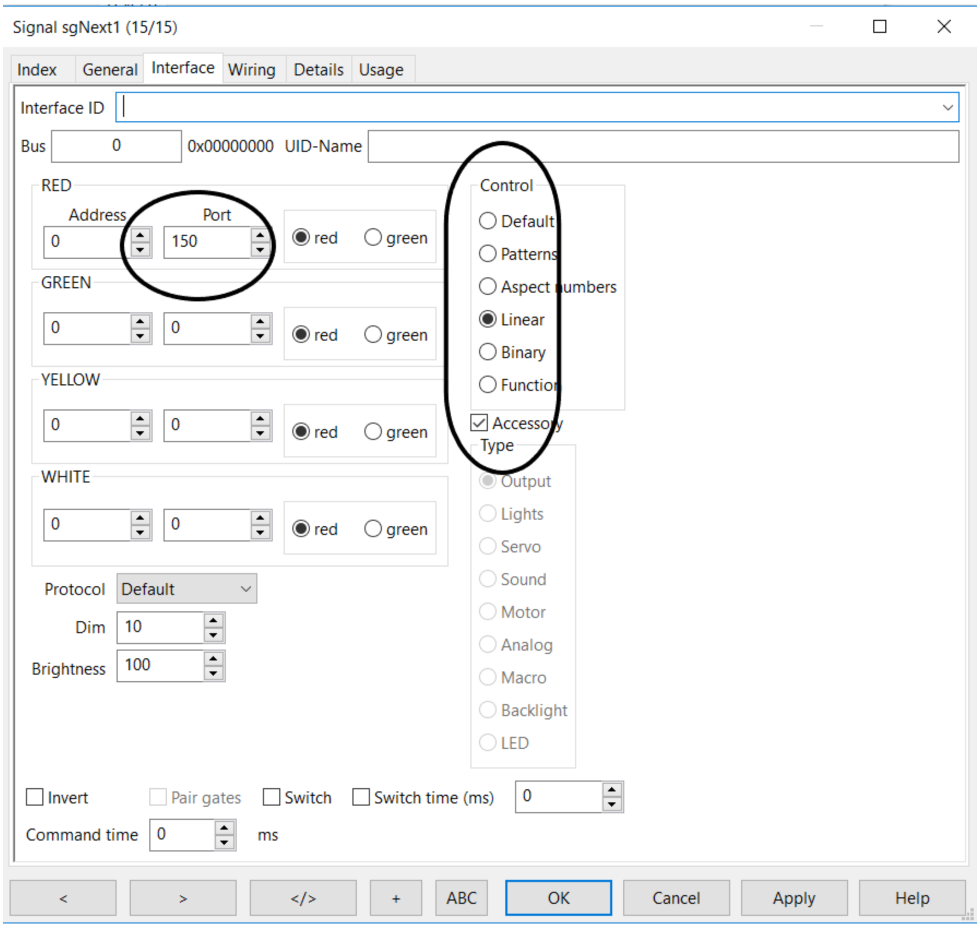
ID @	sgNext1
Number	0
Description @	
Decoder	
Block ID	
Route IDs	
State	2
Accessory#	0
free	
Blank warning at red main signal.	

Options

☐ Manual operated ☑ Reset ☐ Road ☐ Opposite ID

☑ Operable ☑ Show ☑ Show ID ☑ Start of Day

In the Interface tab, set the port address to 150 (PADA) and Control to Linear:



Interface ID

Bus 0 0x00000000 UID-Name

RED
Address 0 Port 150 ● red ○ green

GREEN
0 0 ● red ○ green

YELLOW
0 0 ● red ○ green

WHITE
0 0 ● red ○ green

Protocol Default

Dim 10

Brightness 100

Control
○ Default
○ Patterns
○ Aspect numbers
● Linear
○ Binary
○ Function

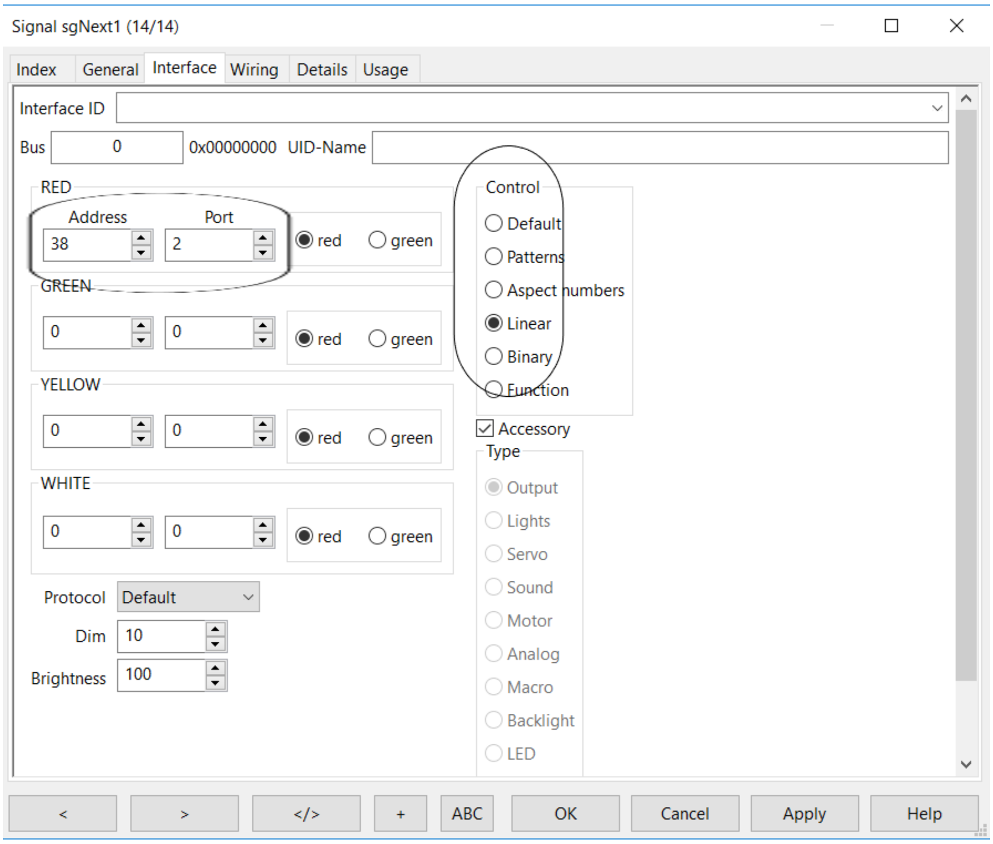
☑ Accessory

Type
● Output
○ Lights
○ Servo
○ Sound
○ Motor
○ Analog
○ Macro
○ Backlight
○ LED

□ Invert □ Pair gates □ Switch □ Switch time (ms) 0

Command time 0 ms

If the user necessarily wants to work with MADA (which does not matter to the DCCNext):



Interface ID

Bus 0 0x00000000 UID-Name

RED
Address 38 Port 2 ● red ○ green

GREEN
0 0 ● red ○ green

YELLOW
0 0 ● red ○ green

WHITE
0 0 ● red ○ green

Protocol Default

Dim 10

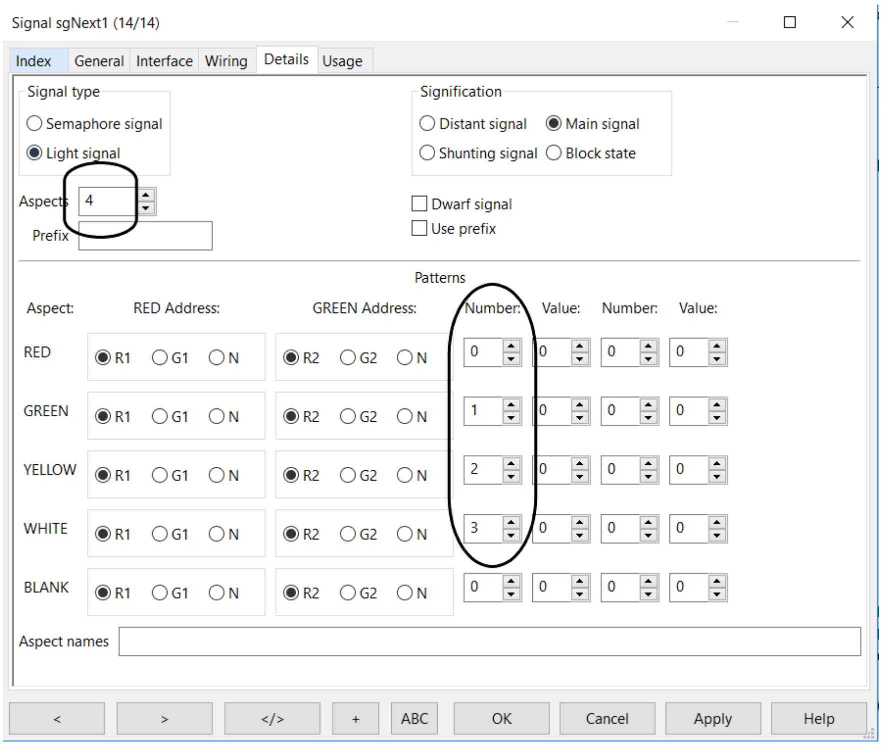
Brightness 100

Control
○ Default
○ Patterns
○ Aspect numbers
● Linear
○ Binary
○ Function

☑ Accessory

Type
● Output
○ Lights
○ Servo
○ Sound
○ Motor
○ Analog
○ Macro
○ Backlight
○ LED

In the Details tab, enter the correct sequence numbers for RED 0, for GREEN 1, YELLOW 2 and WHITE 3. If you have a P signal without a number box, only fill in RED, GREEN and YELLOW and, of course, 3 signal images.:



Signal type: ○ Semaphore signal ● Light signal

Aspects: 4

Prefix:

Signification: ○ Distant signal ● Main signal ○ Shunting signal ○ Block state

□ Dwarf signal

□ Use prefix

Patterns

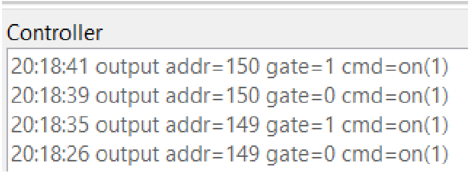
Aspect:	RED Address:	GREEN Address:	Number:	Value:	Number:	Value:
RED	● R1 ○ G1 ○ N	● R2 ○ G2 ○ N	0	0	0	0
GREEN	● R1 ○ G1 ○ N	● R2 ○ G2 ○ N	1	0	0	0
YELLOW	● R1 ○ G1 ○ N	● R2 ○ G2 ○ N	2	0	0	0
WHITE	● R1 ○ G1 ○ N	● R2 ○ G2 ○ N	3	0	0	0
BLANK	● R1 ○ G1 ○ N	● R2 ○ G2 ○ N	0	0	0	0

Aspect names:

- Always click on the signal sgNext1 in the RocRail interface then the signal image RED, YELLOW, GREEN and YELLOW+WHITE lights up alternately (with a neat transition from RED to GREEN via a short intermediate position YELLOW). Another option is to right-click on the signal and choose the desired signal aspect.

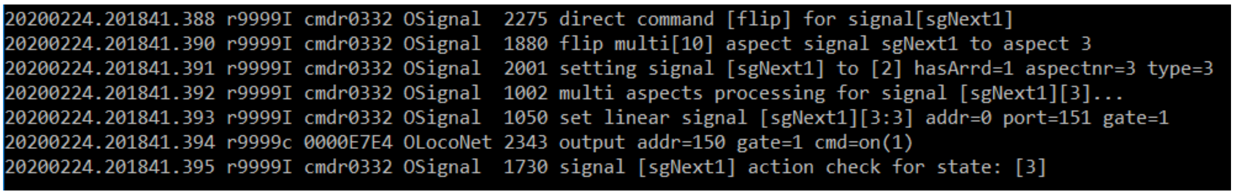
- Take a look at the test DCCNext and the logging in the RocRail interface. If everything is set correctly you will see that RocRail sends the same commands as tried by hand before.

In the logging of RocView you can see the last 4 commands for 4 aspects of signal sgNext1. The addresses in the RocView log are always 1 lower than in the user interface. The latest command is at the top:



```
Controller
20:18:41 output addr=150 gate=1 cmd=on(1)
20:18:39 output addr=150 gate=0 cmd=on(1)
20:18:35 output addr=149 gate=1 cmd=on(1)
20:18:26 output addr=149 gate=0 cmd=on(1)
```

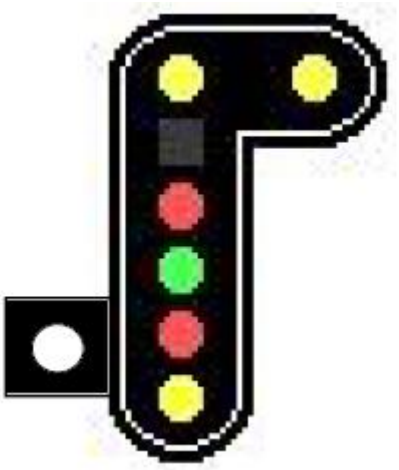
In the RocRail log you can see the last aspect for signal sgNext1. The addresses in the RocRail log are also always 1 lower than in the user interface. The latest command is at the bottom:

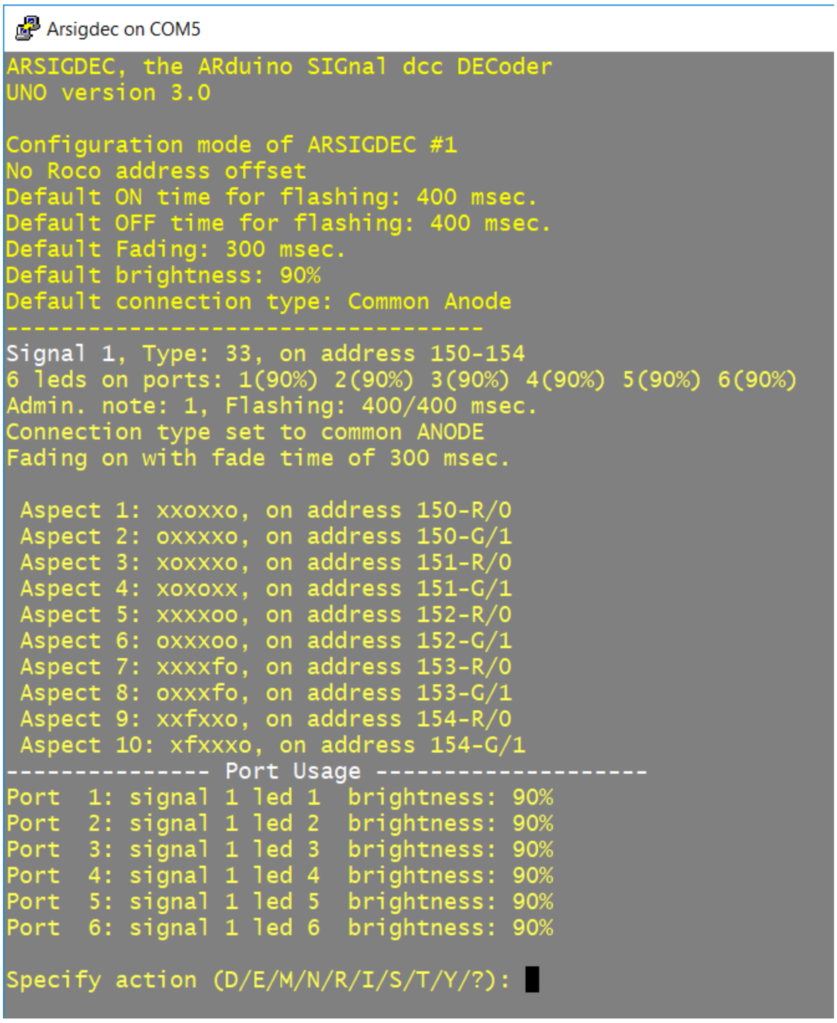


```
20200224.201841.388 r9999I cmdr0332 OSignal  2275 direct command [flip] for signal[sgNext1]
20200224.201841.390 r9999I cmdr0332 OSignal  1880 flip multi[10] aspect signal sgNext1 to aspect 3
20200224.201841.391 r9999I cmdr0332 OSignal  2001 setting signal [sgNext1] to [2] hasArrd=1 aspectnr=3 type=3
20200224.201841.392 r9999I cmdr0332 OSignal  1002 multi aspects processing for signal [sgNext1][3]...
20200224.201841.393 r9999I cmdr0332 OSignal  1050 set linear signal [sgNext1][3:3] addr=0 port=151 gate=1
20200224.201841.394 r9999c 0000E7E4 OLocoNet 2343 output addr=150 gate=1 cmd=on(1)
20200224.201841.395 r9999I cmdr0332 OSignal  1730 signal [sgNext1] action check for state: [3]
```

RocRail configuration French signal

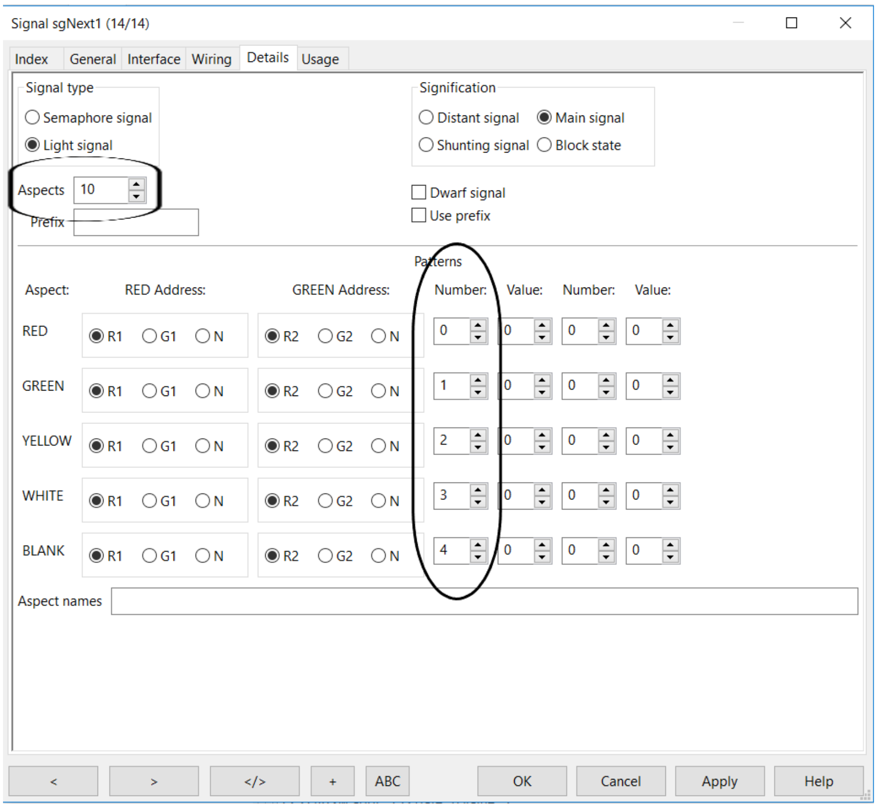
The ArSigDec can also be configured for a signal with number 33 from the database 'Cible-E avec oeilleton' from France. The number of aspects in RocRail is then set to 10.





```
ARSIGDEC, the ARduino SIGnal dcc DECoder
UNO version 3.0

Configuration mode of ARSIGDEC #1
No Roco address offset
Default ON time for flashing: 400 msec.
Default OFF time for flashing: 400 msec.
Default Fading: 300 msec.
Default brightness: 90%
Default connection type: Common Anode
-----------------------------------
Signal 1, Type: 33, on address 150-154
6 leds on ports: 1(90%) 2(90%) 3(90%) 4(90%) 5(90%) 6(90%)
Admin. note: 1, Flashing: 400/400 msec.
Connection type set to common ANODE
Fading on with fade time of 300 msec.

 Aspect 1: xxoxxo, on address 150-R/0
 Aspect 2: oxxxxo, on address 150-G/1
 Aspect 3: xoxxxo, on address 151-R/0
 Aspect 4: xoxoxx, on address 151-G/1
 Aspect 5: xxxxoo, on address 152-R/0
 Aspect 6: oxxxoo, on address 152-G/1
 Aspect 7: xxxxfo, on address 153-R/0
 Aspect 8: oxxxfo, on address 153-G/1
 Aspect 9: xxfxxo, on address 154-R/0
 Aspect 10: xfxxxo, on address 154-G/1
--------------- Port Usage --------------------
Port  1: signal 1 led 1  brightness: 90%
Port  2: signal 1 led 2  brightness: 90%
Port  3: signal 1 led 3  brightness: 90%
Port  4: signal 1 led 4  brightness: 90%
Port  5: signal 1 led 5  brightness: 90%
Port  6: signal 1 led 6  brightness: 90%

Specify action (D/E/M/N/R/I/S/T/Y/?): █
```

And RocRail also:


Signal sgNext1 (14/14)

Index | General | Interface | Wiring | Details | Usage

Signal type
○ Semaphore signal
● Light signal

Aspects 10

Prefix

Signification
○ Distant signal ● Main signal
○ Shunting signal ○ Block state

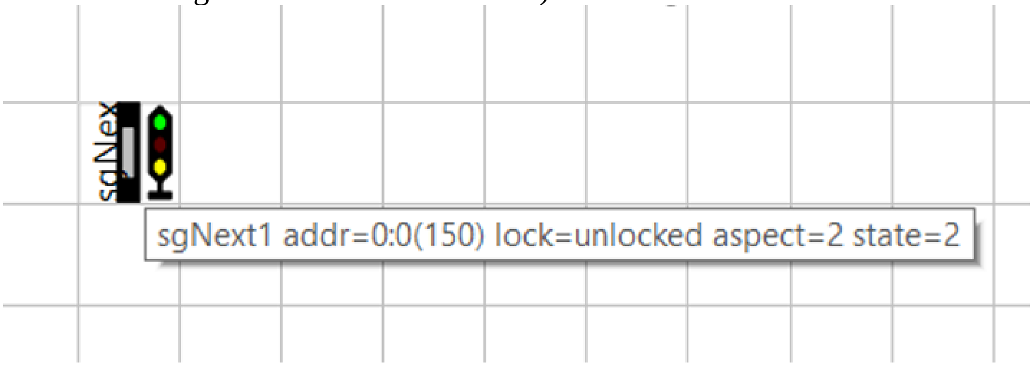
□ Dwarf signal
□ Use prefix

Patterns

Aspect:	RED Address:	GREEN Address:	Number:	Value:	Number:	Value:
RED	● R1 ○ G1 ○ N	● R2 ○ G2 ○ N	0	0	0	0
GREEN	● R1 ○ G1 ○ N	● R2 ○ G2 ○ N	1	0	0	0
YELLOW	● R1 ○ G1 ○ N	● R2 ○ G2 ○ N	2	0	0	0
WHITE	● R1 ○ G1 ○ N	● R2 ○ G2 ○ N	3	0	0	0
BLANK	● R1 ○ G1 ○ N	● R2 ○ G2 ○ N	4	0	0	0

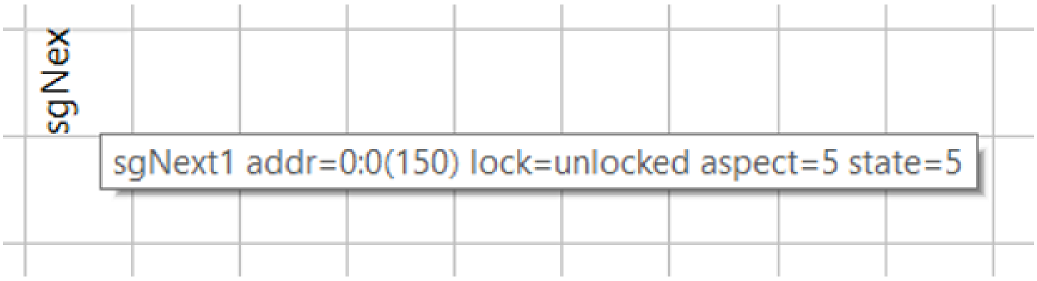
Aspect names

However, the aspects after BLANK have not been entered. Nevertheless, RocRail knows the other signal aspects by clicking on the signal each time in the RocRail interface. All signal aspects are scrolled through neatly (Please note: the technical number of the signal is 1 lower, so in this configuration it runs from 0 to 9).



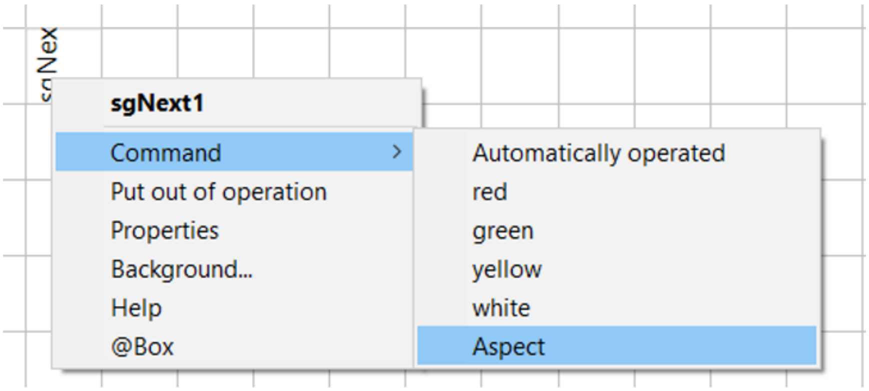
sgNext1 addr=0:0(150) lock=unlocked aspect=2 state=2

However, as soon as the sixth signal aspect and higher is required, the signal symbol disappears from the RocRail interface, but returns to signal aspect 1 to 5.

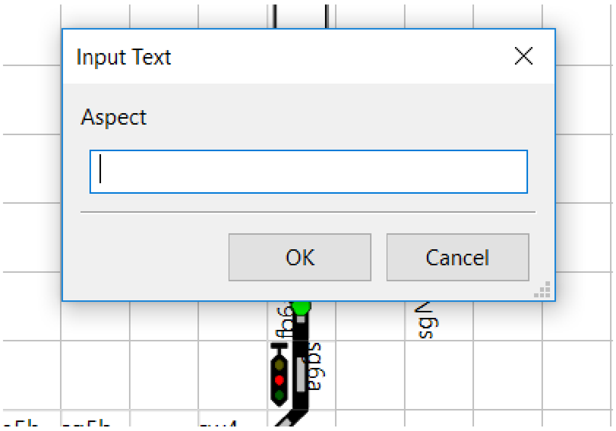


sgNext1 addr=0:0(150) lock=unlocked aspect=5 state=5

You can click the right mouse button on the signal and then select aspect:



Then you can directly enter the desired aspect number:


Input Text
Aspect
OK
Cancel
sg6
sg6a
sgN