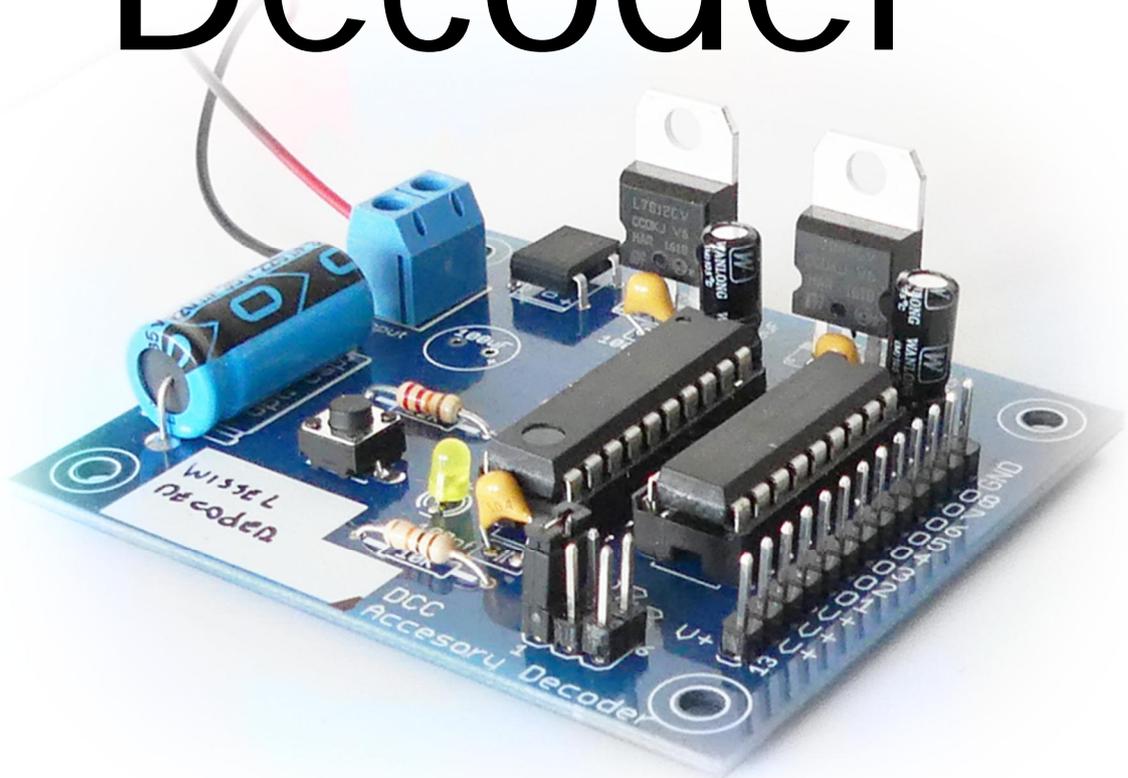


# DIY DCC Accessory Decoder



## Contents

1.	Introduction.....	3
1.1.	Credits.....	3
1.2.	Costs .....	3
2.	Hardware.....	4
2.1.	Schematic.....	4
2.2.	Component List .....	5
2.3.	PCB.....	6
3.	Building the decoder .....	7
3.1.	Testing .....	7
3.2.	Firmware.....	7
4.	How to use.....	8
4.1.	Connecting.....	8
4.2.	Programming.....	8
5.	NS Decoder .....	9
6.	NS Signal Decoder .....	11
7.	OpenDCC Firmware.....	13
7.1.	OpenDecoder .....	13
7.2.	SignalDecoder .....	15
7.3.	EffectDecoder.....	20
7.4.	NatriumDecoder.....	21
7.5.	RelayDecoder .....	22
7.6.	LightDecoder .....	23
7.7.	Dutch SignalDecoder .....	25
7.8.	Belgium SignalDecoder .....	26
8.	Other Firmware.....	27
8.1.	Switch decoder ("Wisseldecoder").....	27
8.2.	DecouplerDecoder ("Ontkoppeldecoder").....	27
8.3.	NL SignalDecoder ("Seindecoder").....	27
8.4.	NL SignalDecoder 3Bits with Speed number ("Seindecoder 3Bits") .....	27
9.	'Koploper' Settings.....	28
9.1.	NS SignalDecoder .....	28
9.2.	SignalDecoder 3Bits "Dave" .....	29

## 1. Introduction

This DCC Accessory decoder can control signals, lights, relays, switches, etc., via DCC. The functionality can be changed via firmware or settings. The PCB design is multifunctional and is designed to use the minimum amount of components.

*Building and using this schematic, board and/or firmware is entirely at your own risk. I am not responsible for any fault or damage in any shape when using this decoder.*

### 1.1. Credits

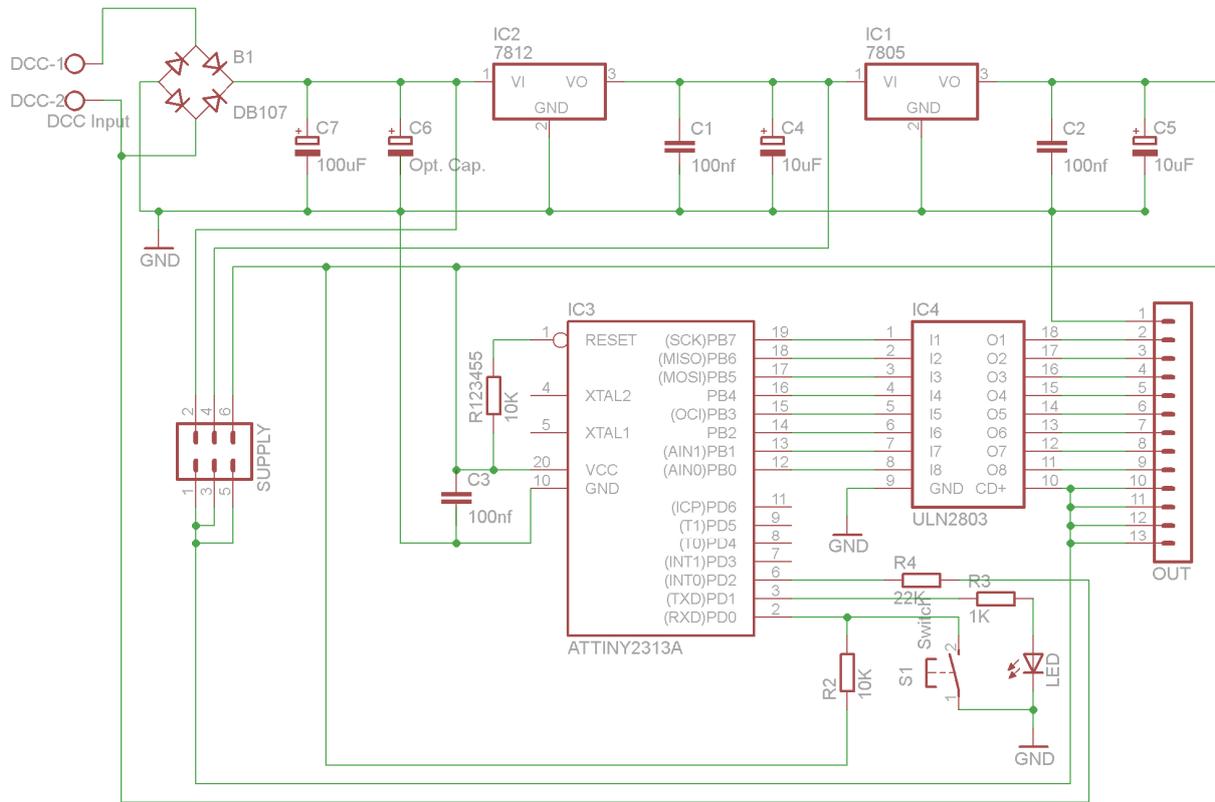
The decoder is based on the firmware of OpenDCC (<https://www.opendcc.de/>) and Dave's mini multi-function decoder (<http://www.daveshobbyhoekje.nl/dcc-mini-multi-function-decoder>). For hardware inspiration, I also looked at Paco's DCC website ([http://usuaris.tinet.cat/fmco/main\\_en.html](http://usuaris.tinet.cat/fmco/main_en.html)).

### 1.2. Costs

The components can be bought anywhere, but I found that components from China (e.g. Aliexpress) are so much cheaper. Including the PCB, the costs for one decoder can be less than 2 euro's. However, it should be noted that some components can be of less quality. For example, the 7805 and 7812 components that I bought, were current limited to 0.5A (not 1A). For most purposes, it should not matter but it would be nice to know before ordering... 😊

## 2. Hardware

### 2.1. Schematic



The power is provided by the DCC input. C7 and (optional) C6 are used as buffer capacitors. The voltage regulators plus filter capacitors will provide a steady 12V and 5V. The 12V regulator will also take some of the heat dissipation when the 5V is used for more than a power supply to the microcontroller.

The microcontroller should be loaded with the firmware, required for the functionality. The DCC signal is an input to the controller via a 22k resistor. A momentary switch can be used to program the mode/settings for the firmware. When in program mode, the LED will burn. The 1K resistor can be replaced by 470 or 560 ohm resistor when the LED is not bright enough.

An ULN2803 is a driver for the output. Normally one ULN2803 is more than enough, however, when more current is required, another ULN2803 can be placed on top of the other ("piggyback"). The power of the driver is determined by the power supply jumper settings. It can be set to the rectified power from the DCC signal (normally 17-18V), 12V or 5V. Another option is to connect the power supply externally via one of the V+ and GND pins.

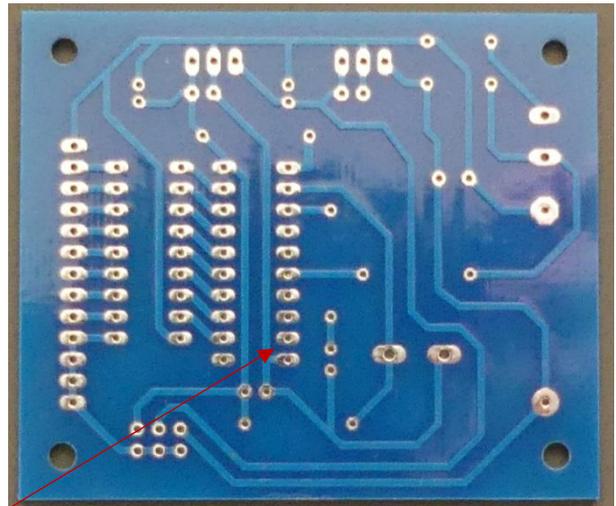
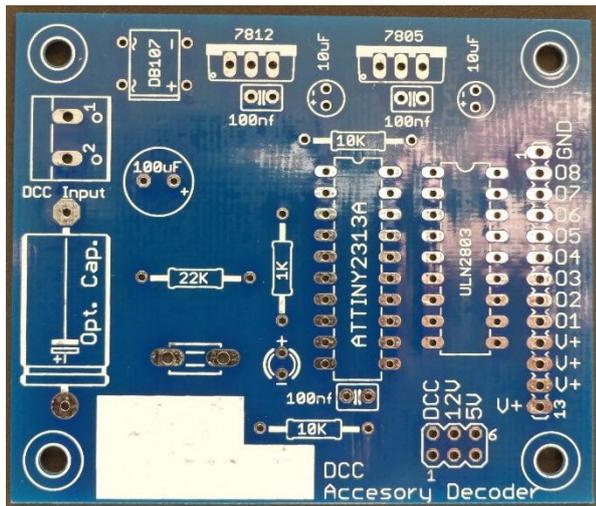
The output is done via a pin header. Screw terminals are nice but more expensive. Also replacing a decoder, in case of malfunctioning, is also easier this way.

## 2.2. Component List

Below is the list of components when using all components.

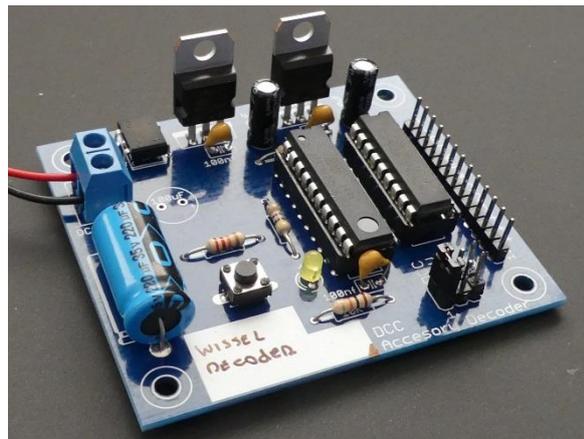
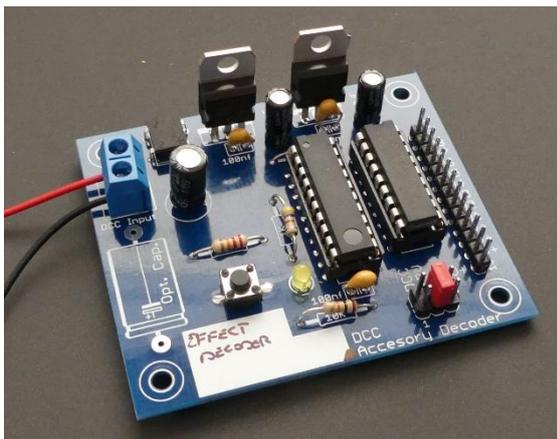
1	DB107	Rectifier
1	7812	12V power regulator
1	7805	5V power regulator
1	ATTINY2313A	Atmel microcontroller (with firmware loaded)
1	ULN2803	Driver
1	18 pin DIL socket	
1	20 pin DIL socket	
3	100 nF	Capacitor, 2.54mm
2	10uF	Capacitor, 2.54mm
1	100uF (or larger)	Capacitor
2	10k , 1/4W	Resistor
1	22k , 1/4W	Resistor
1	1k or 520 , 1/4W	Resistor
1	LED	LED, 3mm
1	2 Screw terminal, 5mm	Or just solder wires...
1	Mini switch	Two pin (not 4), 7 mm.
1	Pin header, 19 pins	13 pins for output, 2x3 pin for supply option, 2.54mm.
1	Jumper	

## 2.3. PCB



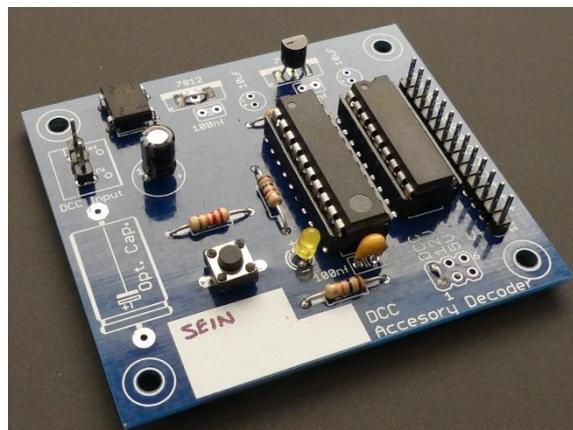
Some firmware from openDCC can use an extra option/setting. This setting can be set by soldering a connection between pins 9 and 10.

The decoder can look like this, the right decoder has a bigger buffer capacitor to use with switches:



Other options can be:

- Smaller buffer capacitor;
- Not using the 12V regulator. In that case, make a connection between the input and output, without making contact to the middle GND pin;
- Use 78L05 (because the microcontroller does not use much current) and do not use the associated filter capacitors;
- Do not use jumpers but solder the supply setting. Or use an external power supply to V+ and GND.



### 3. Building the decoder

The components can be placed according the silkscreen. Use IC sockets for the microcontroller and the driver. Build the print from small/low components to high/big components.

Before connecting the power for the first time, do a visual check to see if everything looks fine, no unwanted connections or missing components. Do not install the microcontroller and driver yet.

#### 3.1. Testing

Before placing the microcontroller and driver perform the following test:

1. Connect DCC power supply;
2. Measure between pin 10 and pin 20 of the microcontroller. This should be 5V.
3. Connect GND from multimeter to the GND connection and measure the supply jumpers:  
They should provide 16-20V for DCC (depending on your power supply), 12V and 5V.

If everything is fine, disconnect the power and place the (loaded with firmware) microcontroller and driver and test:

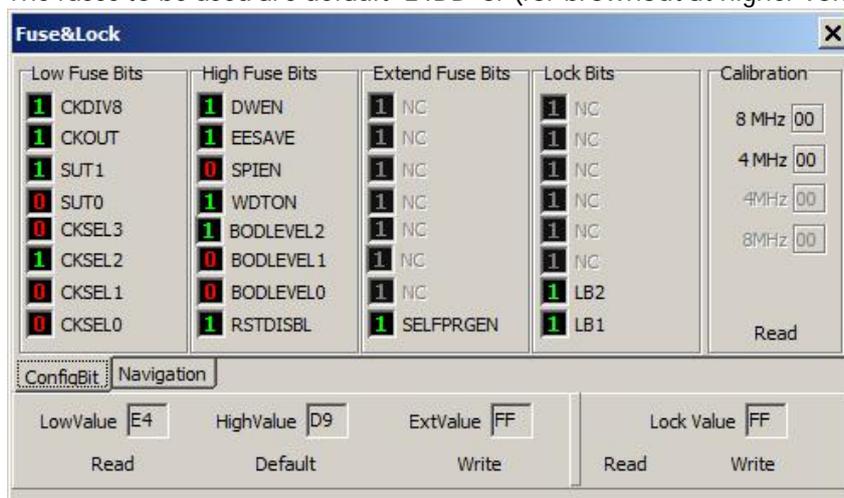
1. Connect DCC power supply;
2. Press program switch;
3. LED should light (program indication)
4. Press program switch again;
5. LED should be off.

If the LED does not light of, check if the LED is not placed in reverse. Check if the fuse-bits, eeprom and code are correctly programmed.

#### 3.2. Firmware

How to program an atiny2313A, will not discussed here.

The fuses to be used are default 'E4DB' or (for brownout at higher voltage) 'E4D9'.



The firmware works as simple accessory decoder, only standard commands are evaluated, neither DCC reset nor broadcast packets are considered.

Note that each firmware contains two files, the code (.hex) and the eeprom values (.eep). They must be loaded both! Note that the firmware from the OpenDCC site is slightly changed to match other pins!

## 4. How to use...

### 4.1. Connecting

The driver will make a connection to ground. That means that the common connection should be V+.

A track switch common will be connected to one of the V+ pins, the other two wires will be connected to O1-O2 (or O3-O4, O5-O6, O7-O8), depending on turned or straight.

The same with lights/LED, the common is V+. Use a resistor (normally 1K) to limit the current in a LED.

For 2-light signals (red/green), the red is connected to O1, green to O2 and further. So 4 signals can be controlled via one decoder.

For 3-lights signals (red/yellow/green), red is connected to O1 (O5), yellow to O2 (O6), green to O3 (O7) and O4 (O8) is optionally used for a speed indicator.

### 4.2. Programming

The setting of the operating mode and of the address is made by simply pressing the programming switch and sending an accessory command. The contents of the first received command will determine mode and address.

1. Press the program tracer for at least 100ms, the LED is turned on. This indicates that the decoder is waiting for an accessory command.
2. Now send an accessory command. The decoder determines his base address from this command, according to the following rule:

$$\text{Base address} = (\text{received address} - 1) / 4 \text{ (the remainder sets the mode)}$$

The decoder is now programmed to the following addresses:

$$\text{base address} * 4 + 1 \text{ up to } \text{base address} * 4 + 4$$

The remainder will be combined with the "coil" (red or green) is used to set the operating mode:

Selected Operation Mode			
0	2	4	6
turnout 1	turnout 2	turnout 3	turnout 4
1	3	5	7

For example, sending the switch command for turnout 18 will set the base address to 4. The decoder now reacts on turnout commands 17 to 20 and operates in mode 2.

This is because  $18/4 = 4$  with remainder 2.

## 5. NS Decoder

This firmware is an expansion of the OpenDCC, Dutch Signal decoder. It also adds modes for setting relays, simple Red/Green signals and track switch coils.

Mode	Operation
0	Two Dutch signals
1	Two Dutch signals
2	Four simple signals (red/green)
3	Max pulse output: 0.25 seconds.
4	Max pulse output: 0.50 seconds.
5	Max pulse output: 0.75 seconds.
6	Each output can be turned individually.
7	Each output can be turned individually, inverse of mode 6.

- Mode 0

Two Dutch Signals		
Address	Pattern	Light
0	Stop	Sig0 - Red
1	Drive	Sig0 - Green
2	Brake till number x 10 km/h, speed must be reached when arrived at next signal	Sig0 - Yellow & number
3	Brake till 40 km/h	Sig0 - Yellow
4	Drive with number x 10 km/h	Sig0 - Green flashing & number
5	Drive with 40 km/h	Sig0 - Green flashes
6	Brake till number x 10 km/h, with the next signal brake further until the speed is reached.	Sig0 - Yellow & number flashes
7	Drive at sight with limited speed	Sig0 - Yellow flashes
8	Stop	Sig1 - Red
9	Drive	Sig1 - Green
10	Brake till number x 10 km/h, speed must be reached when arrived at next signal	Sig1 - Yellow & number
11	Brake till 40 km/h	Sig1 - Yellow
12	Drive with number x 10 km/h	Sig1 - Green flashing & number
13	Drive with 40 km/h	Sig1 - Green flashes
14	Brake till number x 10 km/h, with the next signal brake further until the speed is reached.	Sig1 - Yellow & number flashes
15	Drive at sight with limited speed	Sig1 - Yellow flashes

- Mode 1

Two Dutch Signals		
Address	Pattern	Light
0	Stop	Sig0 - Red
1	Drive	Sig0 - Green
2	Brake till number x 10 km/h, speed must be reached when arrived at next signal	Sig0 - Yellow & number
3	Brake till 40 km/h	Sig0 - Yellow
4	Stop	Sig1 - Red
5	Drive	Sig1 - Green
6	Brake till number x 10 km/h, speed must be reached when arrived at next signal	Sig1 - Yellow & number
7	Brake till 40 km/h	Sig1 - Yellow

- Mode 2

Address	Pattern
0	O1 ON, O2 OFF
1	O1 OFF, O2 ON
2	O3 ON, O4 OFF
3	O3 OFF, O4 ON
4	O5 ON, O6 OFF
5	O5 OFF, O6 ON
6	O7 ON, O8 OFF
7	O7 OFF, O8 ON

- Mode 3-5

Address	Pattern
0	O1 pulse
1	O2 pulse
2	O3 pulse
3	O4 pulse
4	O5 pulse
5	O6 pulse
6	O7 pulse
7	O8 pulse

- Mode 6 (Mode 7 is inversed)

Address	Pattern	Address	Pattern
0	O1 ON	8	O5 ON
1	O1 OFF	9	O5 OFF
2	O2 ON	10	O6 ON
3	O2 OFF	11	O6 OFF
4	O3 ON	12	O7 ON
5	O3 OFF	13	O7 OFF
6	O4 ON	14	O8 ON
7	O4 OFF	15	O8 OFF

Default output when powering up can be set per mode via eeprom in the .eep file. The first line contains the following bytes:

<Address High><Address Low><Mode><Output mode 0>.... <Mode><Output mode 7>....

Default values are: Address=0x0000, Mode=0x00,

Output mode 0= 0x11 (2xred), Output mode 1 = 0x11 (2xred) and Output modes 2-7=0x00.

## 6. NS Signal Decoder

This firmware is an expansion of the OpenDCC, Dutch Signal decoder and contains extra functionality like dimming via a general DCC address! Also permissive signals are present. The P(ermissive) Signal will switch from green to red via yellow (and v.v.).

Mode	Operation
0	Two Dutch signals
1	Two Dutch signals (P-Signal)
2	Two Dutch signals
3	Four (simple) Dutch Signals
4	-
5	-
6	-
7	-

- Mode 0

Two Dutch Signals		
Address	Pattern	Light
0	Stop	Sig0 - Red
1	Drive	Sig0 - Green
2	Brake till number x 10 km/h, speed must be reached when arrived at next signal	Sig0 - Yellow & number
3	Brake till 40 km/h	Sig0 - Yellow
4	Drive with number x 10 km/h	Sig0 - Green flashing & number
5	Drive with 40 km/h	Sig0 - Green flashes
6	Brake till number x 10 km/h, with the next signal brake further until the speed is reached.	Sig0 - Yellow & number flashes
7	Drive at sight with limited speed	Sig0 - Yellow flashes
8	Stop	Sig1 - Red
9	Drive	Sig1 - Green
10	Brake till number x 10 km/h, speed must be reached when arrived at next signal	Sig1 - Yellow & number
11	Brake till 40 km/h	Sig1 - Yellow
12	Drive with number x 10 km/h	Sig1 - Green flashing & number
13	Drive with 40 km/h	Sig1 - Green flashes
14	Brake till number x 10 km/h, with the next signal brake further until the speed is reached.	Sig1 - Yellow & number flashes
15	Drive at sight with limited speed	Sig1 - Yellow flashes

- Mode 1, 2  
Mode 1 is for Permissive signals, Mode 2 for Controlled signals. The permissive signals will change from green to red via yellow and v.v.

Two Dutch Signals, (Controlled and Permissive)		
Address	Pattern	Light
0	Stop	Sig0 - Red
1	Drive	Sig0 - Green
2	Brake till number x 10 km/h, speed must be reached when arrived at next signal	Sig0 - Yellow & number
3	Brake till 40 km/h	Sig0 - Yellow
4	Stop	Sig1 - Red
5	Drive	Sig1 - Green
6	Brake till number x 10 km/h, speed must be reached when arrived at next signal	Sig1 - Yellow & number
7	Brake till 40 km/h	Sig1 - Yellow

- Mode 3

Four Simple Dutch Signals		
Address	Pattern	Light
0	Stop	Sig0 - Red
1	Drive	Sig0 - Green
2	Stop	Sig1 - Red
3	Drive	Sig1 - Green
4	Stop	Sig2 - Red
5	Drive	Sig2 - Green
6	Stop	Sig3 - Red
7	Drive	Sig3 - Green

Default output when powering up can be set per mode via eeprom in the .eep file. The first line contains the following bytes:

```
<Address High><Address Low><Mode><Output mode 0>... <Mode><Output mode 7>
<Dim address High><Dim Address Low><Dim Command Off><Dim Command On><Dim PWM>...
```

Default values are: Address=0x0000, Mode=0x00,

Output mode 0= 0x11 (2xred), Output mode 1 = 0x44 (2xgreen), Output mode 2= 0x11 (2xred) and Output modes 4-7=0x00.

Via the dim parameters, all signals can be dimmed for night operation. Per default, the dimming can be switched on or off via switch command 1024 (base address=0x00FF, Command Off=0x06 and Command On=0x07). The Dim PWM number determines how many cycles (of 50us) the output is off, relative to one cycle with output on. Default value is 0x05.

These values can only be changed by reprogramming the eeprom in the microcontroller.

Jumper on D5 (pin 9) to ground will turn a light off and on when the light is already on. Default, the light will not change.

## 7. OpenDCC Firmware

The information in this chapter is a summary of information already available via the original documentation and new information, but used here to have one documentation for all firmware.

### 7.1. OpenDecoder

Mode	Function
0	Turnout decoder for magnetic drives, pulse duration 0,25 sec
1	Turnout decoder for magnetic drives, pulse duration 0,5 sec
2	Turnout decoder for magnetic drives, pulse duration 1 sec
3	Turnout decoder for magnetic drives, pulse duration 2 sec
4	Decoder for light signals, 4 signals with two aspects
5	Decoder for light signals, 2 signals with three aspects, one signal with two aspects
6	Decoder for light signals, 2 signals with four aspects
7	Decoder for traffic lights

Generally, OpenDecoder doesn't need a pause between individual commands. However, if several magnetic drives are switched simultaneously, so that the power consumption of the outputs is too large, the power regulator will limit the current and switching will not take place. In this case, a short break is recommend between the execution of multiple turnout commands.

- Mode 0, 1, 2, 3

These operating modes are suitable for magnetic coils of conventional double-coil drives for switches or signals.

In mode 0, the pulse time is 0.25sec, in mode 1 the pulse time is 0.5sec, in mode 2 the pulse time is 1sec and in mode 3 the pulse time is 2sec.

If motor controlled turnout are to be operated, a bistable relay or other firmware must be used. Mode 5 is not appropriate for this.

The outputs (up to 500mA each) can be activated simultaneously, but the total should not exceed a current of 1A. Therefore, it is recommended to add a small break after switching. The decoder evaluates only the "switch on" commands. "switch off" commands are ignored and replaced with the programmed pulse duration. There are some command stations (like Intellibox®) which do not a switch off command at all.

- Mode 4

In this mode, four light signals are operated.

If a connection between pin 9 and pin 10 of the microcontroller has been made, the last state will be stored in EEPROM (for operating modes 4 to 7)

- Mode 5

In this mode, two light signals with three aspects and one signal with two aspects can be operated. Each signal aspect (Hp0, Hp1, Hp2) or (Vr0, Vr1, Vr2) is assigned to one output. An aspect can be displayed by several lamps connected to this output.

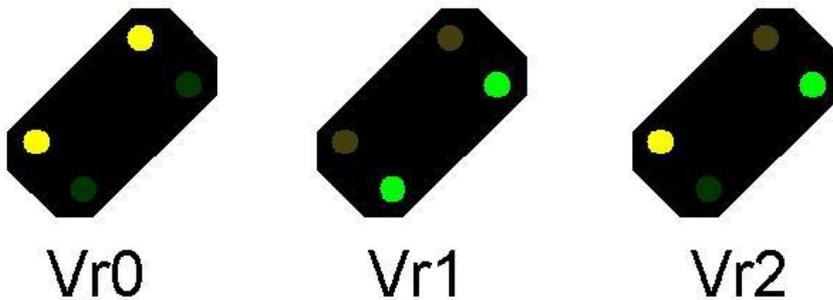
active outputs [76543210]			
xxxxx001	xxxxx100	xx010xxx	01xxxxxx
Signal1: Vr0	Signal1: Vr2	Signal2: Vr1	Signal3: Hp0
1	2	3	4
Signal1: Vr1	Signal2: Vr0	Signal2: Vr2	Signal3: Hp1
xxxxx010	xx001xxx	xx100xxx	10xxxxxx

The outputs are mapped to the commands in accordance with the above table. Each accessory command turns on exactly one output. For signals with three aspects the outputs mutually turn off each other.

If a single LED is contained in different aspects, a diode matrix must be used to connect this LED.

Example:

In the announcing signal of the DB, there are two green and two yellow lights, these are turned on accordingly to the desired aspect:



The lower yellow light is turned on with aspects Vr0 and Vr2, the upper green light is turned on with aspects Vr1 and Vr2. Therefore these LEDs are wired with a diode to both outputs. The diodes are needed for decoupling.

## 7.2. SignalDecoder

Mode	Function
0	Four signals with two aspects
1	Two signals with three aspects, whereby each LED has its own output (no diodes required)
2	station entry signal and announcement signal
3	station entry signal and announcement signal, with dark palpation
4	Flashing lights (for railroad crossing, pedestrian crossing), additional control of a stop magnet for Faller Car.
5	each output controlled individually
6	four signals with four aspects (red, green, yellow, off)
7	traffic light, with control of stop magnet

Generally, OpenDecoder doesn't need any pause between instructions. With every mode the outputs for the previous aspect are dimmed down, then the new aspect is dimmed up.

The SignalDecoder evaluates only 'coil on' commands, 'coil off' commands are ignored.

If a connection between pin 9 and pin 10 of the microcontroller has been made, the last state will be stored in EEPROM.

- Mode of operation 0

In this mode four light signals with two aspects are controlled.

Mode of operation 0: Four signals, two aspects		
Address	Aspect	Bright
0	Hp0 - Course stop	Sig0 - red
1	Hp1 - Travel	Sig0 - green
2	Hp0 - Course stop	Sig1 - red
3	Hp1 - Travel	Sig1 - green
4	Hp0 - Course stop	Sig2 - red
5	Hp1 - Travel	Sig2 - green
6	Hp0 - Course stop	Sig3 - red
7	Hp1 - Travel	Sig3 - green

- Mode of operation 1  
In this mode two light signals with three aspects each are controlled. The light signals are connected directly (without additional diodes) to the decoder.

Mode of operation 1: Two signals, three aspects			
Address	Aspect on main signal	Aspect on announcement signal	Outputs
0	Hp0 - Course stop	Vr0 - Course stop	xxxx0011
1	Hp2 - reduced speed	Vr1 - Travel	xxxx1100
2	Hp1 - Travel	Vr2 - reduced speed	xxxx0110
3	reserved	reserved	xxxx0000
5	Hp0 - Course stop	Vr0 - Course stop	0011xxxx
5	Hp2 - reduced speed	Vr1 - Travel	1100xxxx
6	Hp1 - Travel	Vr2 - reduced speed	0110xxxx
7	reserved	reserved	0000xxxx

- Modes of operation 2 and 3  
In these modes of operation a station entry signal and an announcement signal are controlled. The station entry signal has four aspects: Hp0, Hp1, Hp2 and Sh1. If the announcement signal and station entry signal are mounted at the same location, then the announcement signal is dark while the station entry signal is showing Hp0 or Sh1. The dark palpation of the announcement signal only takes place in mode 3, mode 2 is intended for two independent signals.

Mode of operation 3: controlling main signal and pilot signal			
Address	Aspect	main signal	announcement signal
0	Hp0 - Course stop	Red1 Red2	dark
1	Hp1 - Travel	Green	shines
2	Hp2 - reduced speed	Green Yellow	shines
3	Sh1 - station travel	Red1 White White	dark
4	Vr0 - expect course stop	unchanged	Yellow Yellow
5	Vr1 - expect travel	unchanged	Green Green
6	Vr2 - expect reduced speed	unchanged	Green Yellow
7	reserved	unchanged	dark

Note: driving a main signal for leaving a station together with an announcing signal with 3 aspects requires four external diodes, because the hardware only has 8 outputs, but 9 different lamps are headed for. Connection in this case is the same as with the switching decoder.

- Mode of operation 4  
 Decoder for two flashing outputs and control of a stop magnet, suitable for railroad crossing.  
 Following functions are provided:
  - St Andrews cross (2 outputs)
  - Control light at the track: there is a signal (O1, O2) showing to the engine driver that the St Andrews cross also is flashing. This signal is equipped with a white light, which begins flashing shortly after the flashing at the crossing.  
 On these signals there is another yellow point; it replaces the light for 'ready for operation' used in former times.
  - Stop magnet: This output controls the magnet for Faller Car. This output is activated approx. 2s after flashing and remains switched on 2s longer.

Mode of operation 4: St Andrews cross, blinking control display for engine driver, stop magnet

Address	Term	Effect
0	-	Group 1: turn off blinking, later turn off stop magnet
1	-	Group 1: turn on blinking, turn on stop magnet
2	-	-
3	-	-
4	-	Group 2: turn off blinking, later turn off stop magnet
5	-	Group 2: turn on blinking, turn on stop magnet
6	-	-
7	-	-

- Connection in this case:
  - 0 Group 1, St Andrews cross phase A
  - 1 Group 1, St Andrews cross phase B
  - 2 Group 1, engine driver light
  - 3 Group 1, stop magnet
  - 4 Group 2, St Andrews cross phase A
  - 5 Group 2, St Andrews cross phase B
  - 6 Group 2, engine driver light
  - 7 Group 2, stop magnet

- Mode of operation 5  
Each output is turned on-off individually; when switching on or off a smooth dimmed transition takes place. In this mode the decoder uses 8 successive addresses;

Mode of operation 5: individual controlled outputs		
Address	Term	Effect
0	-	output 0 on
1	-	output 0 off
2	-	output 1 on
3	-	output 1 off
4	-	output 2 on
5	-	output 2 off
6	-	output 3 on
7	-	output 3 off
8	-	output 4 on
9	-	output 4 off
10	-	output 5 on
11	-	output 5 off
12	-	output 6 on
13	-	output 6 off
14	-	output 7 on
15	-	output 7 off

- Mode of operation 6  
In this mode of operation the decoder uses 8 successive addresses;

Mode of operation 6: Driving R/G search Lights		
Address	Term	Bright
0	-	Sig0 - red
1	-	Sig0 - green
2	-	Sig0 - red+green
3	-	Sig0 - off
4	-	Sig1 - red
5	-	Sig1 - green
6	-	Sig1 - red+green
7	-	Sig1 - off
8	-	Sig2 - red
9	-	Sig2 - green
10	-	Sig2 - red+green
11	-	Sig2 - off
12	-	Sig3 - red
13	-	Sig3 - green
14	-	Sig3 - red+green
15	-	Sig3 - off

- Mode of operation 7  
Traffic light  
This mode controls a traffic light. The outputs are identical to the OpenDecoder, however the lights are softly dimmed on transitions.  
Links:  
[Wikipedia - Lichtzeichenanlage!](#)  
[www.info-lsa.de](http://www.info-lsa.de) many pictures and background information

### 7.3. EffectDecoder

This software is able to generate special light effects, with emphasis on running lights, light columns and the like. All modes are working with a variable timing engine, the fading of the individual bulbs occurs with an adjustable curve. When using Led's, all Led's should have a resistor (1K) connected to the output of the decoder and the cathode. The anodes of all LED's should be connected to the V+

Mode	Operation
0	Running light for construction sites, this can either go gently (smooth transitions) or jumping with flashing lights.
1	Light chain forward. One or several points of light pass hop along, with the point of light can enter at the beginning again seamlessly.
2	as 1, but backwards
3	light column
4	reserved
5	reserved
6	reserved
7	reserved

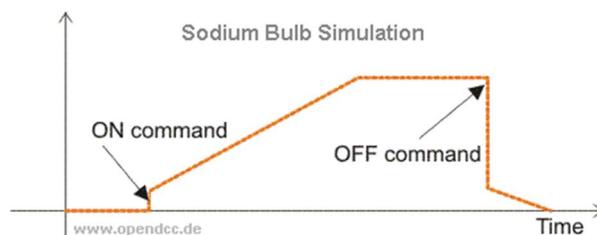
- Mode 0, construction site chase  
This mode controls the outputs to form a construction site chase. Light is running from output 1 to output 8.

Button	Effect
1	All switched off.
2	The chase is on; slow running speed and gentle up and down dimming of the lamps
3	The chase is on; fast running speed and lightning-flash of the lamps.
4	reserved
5	reserved
6	reserved
7	reserved
8	reserved

- Mode 1, Light chain forward
- Mode 1, Light chain backwards

## 7.4. NatriumDecoder

This firmware simulates the unique features of sodium vapor lamps. These are initially dim after power is first applied, and require a period of ramping up before full brightness is achieved. They also glow for a short period after being turned off. This decoder models this typical time course.



Mode	Function
0	Very fast dimming.
1	Accelerated simulation, but still observable.
2	Normal simulation, with an imperceptible rise to full brightness.
3	Very slow simulation
4	reserved
5	reserved
6	reserved
7	reserved

The Sodium Decoder only responds to "on" commands, and ignores "off" commands, as some command stations, such as Intellibox (R) do not send "off" commands. Therefore the following are used to mimic shut-off of the lamps.

Sodium decoder	
Address	Action
0	All the lamps simultaneously switch to 15% brightness and then gradually dim to zero.
1	All the lamps simultaneously switch to 15% brightness and then dim up to full brightness.
2	All the lamps randomly switch to 15% brightness and then gradually dim to zero. The switching occurs in an interval of 160 ms.
3	As in 1, but the switching-on is random within a time interval of 160ms.
4	Like 2.
5	Like 3, but also varies the initial dimming to the range 15% to 25% randomly.
6	Like 2
7	As in 5, but will also may randomly select a light to mimic a light at the end of its life. It will extinguish after a time and then restart.

## 7.5. RelayDecoder

The relay decoder is a simple decoder that just turns the output on or off, depending on the command and modes. When a Base address is programmed, output 5-8 are controlled via the next base address (base address+1).

If a connection between pin 9 and pin 10 of the microcontroller has been made, the last state will be stored in EEPROM.

Mode 1 does the inverse of mode 0, switching on instead of off and v.v.

Relay decoder mode 0		Relay decoder mode 1	
Address	Action	Address	Action
0	Output 1 off.	0	Output 1 on.
1	Output 1 on.	1	Output 1 off.
2	Output 2 off.	2	Output 2 on.
3	Output 2 on.	3	Output 2 off.
4	Output 3 off.	4	Output 3 on.
5	Output 3 on.	5	Output 3 off.
6	Output 4 off.	6	Output 4 on.
7	Output 4 on.	7	Output 4 off.
8	Output 5 off.	8	Output 5 on.
9	Output 5 on.	9	Output 5 off.
10	Output 6 off.	10	Output 6 on.
11	Output 6 on.	11	Output 6 off.
12	Output 7 off.	12	Output 7 on.
13	Output 7 on.	13	Output 7 off.
14	Output 8 off.	14	Output 8 on.
15	Output 8 on.	15	Output 8 off.

This decoder can be used with relays (mind the maximum current), but also with lights and other simple on/off functionality.

Lightdecoder, mode 7 or NSDecoder, mode 6 and 7, can also be used to control relays.

## 7.6. LightDecoder

This firmware feature special features to do an illumination on a model railway layout. It is intended to lighten individual buildings or complete streets (including the flickering of fluorescent lights). A random generator provides different sequences every transition day-night.

If a connection between pin 9 and pin 10 of the microcontroller has been made, the last state will be stored in EEPROM.

Mode	Operation
0	Combined switching of all 8 outputs, very short time interval
1	Combined switching of all 8 outputs, mean time interval
2	Combined switching of all 8 outputs, long time interval
3	Combined switching of all 8 outputs, very long time interval
4	reserve
5	reserve
6	Each output can be turned on flashing.
7	Each output can be turned individually.

In the modes 0, 1, 2 or 3, all eight outputs are controlled by one DCC command and are switching combined. However, the processing speed of the animated sequence is totally different.

- Mode 0  
All 8 outputs are controlled with one command.

Mode 0: Combined switching, fast	
Address	Action
0	All outputs off
1	All outputs turned on
2	Each output is shut down after a randomly selected time.
3	Each output is turned on after a randomly selected time.
4	All outputs off
5	Each output is turned on with a random delay and with a random flickering pattern. (fluorescent tube)
6	reserved
7	reserved

For commands 2 and 3, the processing time for each output ranges from 10 msec to 2.5 sec.

For command 5 (fluorescent tube), the initial delay ranges from 10ms to 80ms, then the lamp may flicker up to two times until it burns continuously. The flickering may take up to 400ms.

- Mode 1  
This mode is identical to the above, only here the times are extended as follows:  
  
For the commands 2 and 3 the time range is between 250ms and 63s.  
It makes no sense to call command 5, the initial delay would be 250ms and 20s, the 'flickering' would last for 100s.
- Modes 2 und 3  
This mode is identical to the above, only here the times are doubled:  
For commands 2 and 3 this results in a processing time from 500ms und 2min.
- Mode 4  
reserved
- Mode 5  
reserved
- Mode 6, flashing light (individually controlled)  
In this mode, the eight outputs can be controlled individually. However, to access these outputs, more than the usual four addresses for a decoder are required, the decoder simply uses the next four addresses.

Address	Action
1	Switching off the output 1
2	Switching on output 1; this output starts flashing.
3	Switching off the output 2
4	Switching on output 2; this output starts flashing.
5	Switching off the output 3
6	Switching on output 3; this output starts flashing.
7	Switching off the output 4
8	Switching on output 4; this output starts flashing.
9	Switching off the output 5
10	Switching on output 5; this output starts flashing.
11	Switching off the output 6
12	Switching on output 6; this output starts flashing.
13	Switching off the output 7
14	Switching on output 7; this output starts flashing.
15	Switching off the output 8
16	Switching on output 8; this output starts flashing.

- Mode 7, individually controlled outputs  
In this mode, the eight outputs can be controlled individually (permanent on). However, to access these outputs, more than the usual four addresses for a decoder are required, the decoder simply uses the next four addresses. (see mode 6)

## 7.7. Dutch SignalDecoder

This firmware can control two Dutch signals. Red is connected to O1 (O5), Yellow to O2 (O6), Green to O3 (O7) and a speed indicator to O4 (O8). Common again connected to V+ and do not forget the resistors when using LED's.

If a connection between pin 9 and pin 10 of the microcontroller has been made, the last state will be stored in EEPROM.

Two Dutch Signals		
Address	Pattern	Light
0	Stop	Sig0 - Red
1	Drive	Sig0 - Green
2	Brake till number x 10 km/h, speed must be reached when arrived at next signal	Sig0 - Yellow & number
3	Brake till 40 km/h	Sig0 - Yellow
4	Drive with number x 10 km/h	Sig0 - Green flashing & number
5	Drive with 40 km/h	Sig0 - Green flashes
6	Brake till number x 10 km/h, with the next signal brake further until the speed is reached.	Sig0 - Yellow & number flashes
7	Drive at sight with limited speed	Sig0 - Yellow flashes
8	Stop	Sig1 - Red
9	Drive	Sig1 - Green
10	Brake till number x 10 km/h, speed must be reached when arrived at next signal	Sig1 - Yellow & number
11	Brake till 40 km/h	Sig1 - Yellow
12	Drive with number x 10 km/h	Sig1 - Green flashing & number
13	Drive with 40 km/h	Sig1 - Green flashes
14	Brake till number x 10 km/h, with the next signal brake further until the speed is reached.	Sig1 - Yellow & number flashes
15	Drive at sight with limited speed	Sig1 - Yellow flashes

## 7.8. Belgium SignalDecoder

This firmware can control two Belgium signals with seven patterns (aspects) per signal. Green is connected to O1 (O5), red to O2 (O6), yellow to O3 (O7) and another yellow to O4 (O8). Common again connected to V+ and do not forget the resistors when using LED's.

If a connection between pin 9 and pin 10 of the microcontroller has been made, the last state will be stored in EEPROM.

Belgian trains are raiiling on the left site with left site signals. Flashing means you are riding on an exceptional track (and therefore right site with right sites signals). All signal patterns remain the same but are flashing.

Operation mode Belgium: 2 Signals with 7 patterns		
Address	Pattern	Light
0	Full Stop	Sig0 - Red
1	Proceed	Sig0 - Green
2	Next signal must be expected to show "Stop"	Sig0 - yellow/yellow
3	Limited approach	Sig0 - Green/Yellow
4	Full stop in opposite direction	Sig0 - Red flashing
5	Proceed in opposite direction	Sig0 - Green flashing
6	Next signal must be expected to show "Stop" in opposite direction	Sig0 - Yellow/yellow flashing
7	off	Sig0 - off
8	Full Stop	Sig1 - Red
9	Proceed	Sig1 - Green
10	Next signal must be expected to show "Stop"	Sig1 - Yellow/yellow
11	Limited approach	Sig1 - Green/Yellow
12	Full stop in opposite direction	Sig1 - Red flashing
13	Proceed in opposite direction	Sig1 - Green flashing
14	Next signal must be expected to show "Stop" in opposite direction	Sig1 - Yellow/Yellow flashing
15	off	Sig1 - off

## 8. Other Firmware

This firmware has been created by Dave van der Locht and can also be used (pins are the same). A Dutch manual and firmware can also be loaded from his site (<http://www.daveshobbyhoekje.nl/dcc-mini-multi-function-decoder/>). But a summary of the information will also be shown here.

Note that the connections for the signal decoders are Red (O1, O5), Green (O2, O6), Yellow (O3, O7) and Speed number (O4, O8).

### 8.1. Switch decoder (“Wisseldecoder”)

Mode	Operation
0	Max pulse output: 0.25 seconds.
1	Max pulse output: 0.50 seconds.
2	Max pulse output: 0.75 seconds.

### 8.2. DecouplerDecoder (“Ontkoppeldecoder”)

Mode	Operation
0	Decoupler default
1	Decoupler Koploper (easier to use in program ‘Koploper’)

In mode 1, the decoupler can be activated as soon as the loc is above the decoupler. Via a CV (timing), the activation is somewhat delayed. The decoder can control 4 decouplers via 2 addresses.

### 8.3. NL SignalDecoder (“Seindecoder”)

Mode	Operation
0	2x 3-lights signal (red, yellow, and green).
1	4x 2-lights signal (red, green)

### 8.4. NL SignalDecoder 3Bits with Speed number (“Seindecoder 3Bits”)

Mode	Operation
0	2x 3-lights signal (red, yellow, green, speed number).

## 9. 'Koploper' Settings

This paragraph describes some examples on how to use this decoder within the Koploper program.

### 9.1. NS SignalDecoder

To be used with decoder, as described in chapter 5 and 6.

A permissive signal without speed indicator should have the following settings:

The screenshot shows the 'Seineigenschappen (uitgebreid)' window. At the top, 'Aansluitnummers' are set to Adres 1: 25, Adres 2: 26, Adres 3: 0, Adres 4: 0, Adres 5: 0. The 'Verstuur altijd' checkbox is unchecked. The table below shows the signal states and their corresponding addresses.

Seinstanden	Adres 1	Adres 2	...	...	...	Volgorde
Rood	Afb	-				
Groen	Recht	-				
Geel	-	Recht				

Buttons on the right include OK, Annuleer, Help, Adressen = 0, Default, and Kopieer.

A signal with speed indicator should have the following settings.

The screenshot shows the 'Seineigenschappen (uitgebreid)' window. At the top, 'Aansluitnummers' are set to Adres 1: 29, Adres 2: 30, Adres 3: 31, Adres 4: 32, Adres 5: 0. The 'Verstuur altijd' checkbox is unchecked. The table below shows the signal states and their corresponding addresses.

Seinstanden	Adres 1	Adres 2	Adres 3	Adres 4	...	Volgorde
Rood	Afb	-	-	-		
Groen	Recht	-	-	-		
Geel + Cijfer	-	Afb	-	-		
Geel	-	Recht	-	-		
Groen knipper + cijfer	-	-	Afb	-		
Groen knipper	-	-	Recht	-		
Geel + knipperend cijfer	-	-	-	Afb		
Geel knipper	-	-	-	Recht		

Buttons on the right include OK, Annuleer, Help, Adressen = 0, Default, and Kopieer.

## 9.2. SignalDecoder 3Bits "Dave"

To be used with decoder, as described in paragraph 8.4.

The settings must be exactly the same, apart from the addresses. Note that the order is very important:

Signal 1 without speed indicator will require the following settings in mode 0:

The second signal will then be called with addresses 9, 10 and 11

Seinstanden	Adres 1	Adres 2	Adres 3	...	...	Volgorde
Rood	Afb	Afb	Afb			1,2,3
Groen	Recht	Afb	Afb			1,2,3
Geel	Afb	Recht	Afb			1,2,3
Groen knipper	Recht	Afb	Recht			1,3,2
Geel knipper	Afb	Recht	Recht			1,2,3

Signal 2 with speed indicator:

Seinstanden	Adres 1	Adres 2	Adres 3	...	...	Volgorde
Rood	Afb	Afb	Afb			1,2,3
Groen	Recht	Afb	Afb			1,2,3
Geel + Cijfer	Afb	Afb	Recht			3,1,2
Geel	Afb	Recht	Afb			1,2,3
Groen knipper + cijfer	Recht	Recht	Afb			1,2,3
Groen knipper	Recht	Afb	Recht			1,3,2
Geel + knipperend cijfer	Recht	Recht	Recht			1,2,3
Geel knipper	Afb	Recht	Recht			1,2,3