VPEB Van Perlo Elektronica & Besturingstechniek

UCCI/E

Manual

Author: Leon J.A. van Perlo
Version: 1.0
Date: February 18, 2014

Smart electronics for your miniature world

Release Management

This manual is applicable for modules and kits consisting of:

Module:

- UCCI/E Rev00
- UCCI/E-s Rev00
- UCCI/E Rev01
- UCCI/E-s Rev01

Firmware:

- UCCI/E Bootloader 1.0 / Firmware UCCI/E 2.00 (or later)

Software:

- DinamoConfig 1.11

Preface

This manual describes the use of the UCCI/E as part of the Dinamo/MCC system. To complete your understanding of the system you may also need to read the manuals for the OC32, RM-U and MCCdec.

Note there are 2 versions of the UCCI/E. In this manual they will be referred to as follows:

- The normal UCCI/E, that can be used as an autonomous system, is referred to as **UCCI/E**
- The UCCI/E-s without USB and RS232 interface, is referred to as **UCCI/E-s**
- The UCCI/E in general, which can be either the UCCI/E or the UCCI/E-s is referred to as **UCCI/E(-s)**

©2014 This document, or any information contained herein, may not be copied or distributed, in whole or in parts, in whatever form, without the explicit written approval of the original author. The making of copies and prints by users of the Dinamo system or the UCCI/E module for their own use is allowed.

Contents

1 Dinamo/MCC 4
1.1 Introduction 4
1.2 Small/medium sized layout 4
1.3 Large size layout 4
2 UCCI/E(-s) 6
2.1 Overview 6
2.2 Mounting 6
2.3 Power supply 7
2.4 Connecting UCCI/E to your PC 8
2.5 Controlling OC32 or OM32 Serial modules from the UCCI/E 9
2.6 Connecting UCCI/E-s modules to the RM-U controller 10
2.6.1 RS485 10
2.6.2 Connecting RM-U, UCCI/E-s and OC32 modules 11
2.6.3 Addressing 12
3 Transmission loops 14
3.1 The basics 14
3.2 Loop length, wire size and connection to UCCI/E(-s) 17
3.3 Small scales 18
4 Feedbacks 19
4.1 Installing reed-switches when using steel-wire for steering 19
4.2 Installing reed-switches when using magnetic strip for steering 20
4.3 Connecting feedbacks to UCCI/E(-s) 21
5 Operation and Configuration 23
5.1 Operation 23
5.2 UCCI/E: Two logical units 23
5.3 Configuration 25
5.4 RM-U configuration settings 25
5.4.1 OM32 Retransmission 25
5.4.2 Transparent Mode 25
5.4.3 Bootloader Transparent Mode 26
5.5 UCCI-s configuration settings 26
5.5.1 Switch activation delay 26
5.5.2 Switch release delay 26
5.5.3 Set Address 26
5.5.4 Set Loop Current 27
6 Firmware Update 28
6.1 Requirements 28
6.2 Firmware Update 29

1 Dinamo/MCC

1.1 Introduction

On a model rail road layout nowadays one finds more often an extension with moving model-cars, for example on the basis of the Faller Car System. The possibilities to control cars of this system in the original state are very limited.

VPEB developed the Dinamo/MCC system to control these kind of cars in a sophisticated way. Cars are equipped with a decoder that receives commands through a wireless communication system embedded in the road-surface. The road-surface also contains feedback-sensors that report the positions of the cars to the control system. In addition, there can be turnouts and traffic lights to make the layout realistic and complete.

The Dinamo/MCC system is built using three different modules:

- UCCI/E(-s): This module generates the signals to control the decoders in the cars and reads the position sensors in the road;
- MCCdec: The decoder(s) in the car(s) that receive commands from UCCI/E(-s) and control all functions of the car;
- OC32: This module controls traffic lights, turnouts and any other scenery on your layout.

1.2 Small/medium sized layout

A small to medium sized layout, consisting of a maximum of 40 meters of road (the sum of all lanes), can be controlled by a single UCCI/E and optionally one or more OC32 controllers.

Schematically this setup looks as follows:

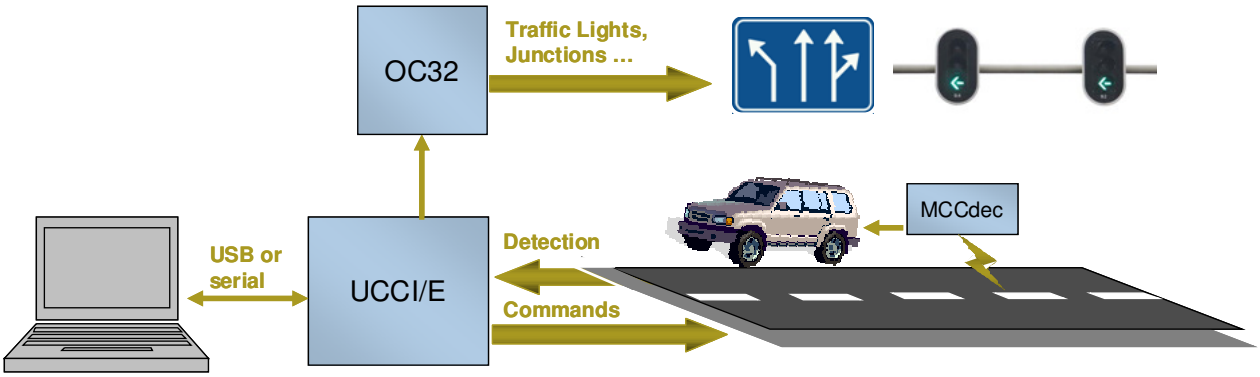

Fig 1: Schematic setup of a small/medium Dinamo/MCC system

The UCCI/E is connected to a PC with adequate control software through a USB or RS232 (com-port) connection. UCCI/E generates the signals for the decoders and reads the feedbacks. UCCI/E (optionally) controls one or more OC32 controllers as required to drive turnouts, traffic lights and other accessories.

1.3 Large size layout

When the layout extends beyond the limit of 40 meters of road (the sum of all lanes) that can be controlled by a single UCCI/E, a slightly different setup is required. In this case more than one UCCI/E is needed to control the total length of lanes on your layout. The large-scale Dinamo/MCC system consists of an RM-U central control unit, connected to a PC with

adequate control software through a USB or RS232 (com-port) connection. The RM-U controls up to 16 UCCI/E controllers and up to 16 OC32 controllers. In total therefore this setup is able to control up to 640 meters of lanes.
Schematically this large-scale setup looks as follows:

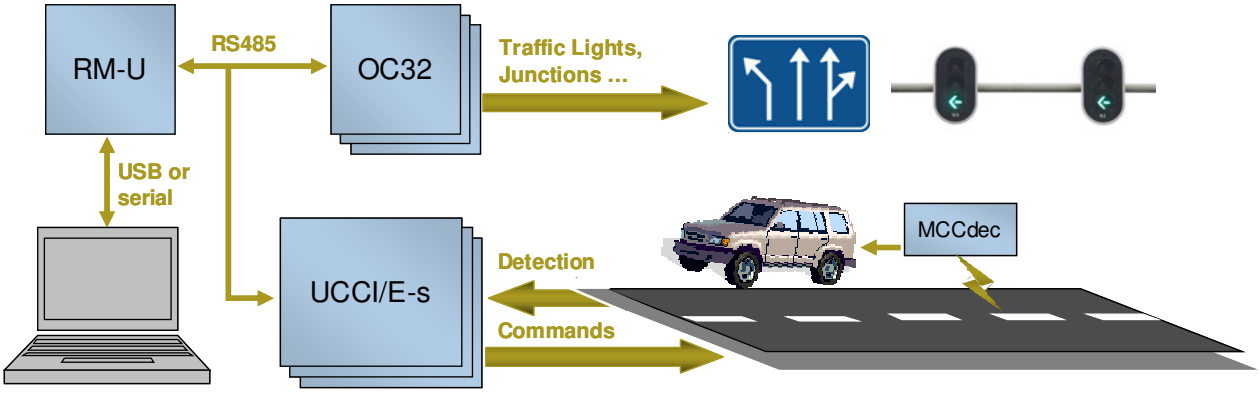

Fig 2: Schematic setup of a large Dinamo/MCC system

Note, in the above diagram, UCCI/E is replaced by an UCCI/E-s. The '-s' stands for 'slave unit', meaning this unit requires a RM-U central controller to connect to the controlling PC. An UCCI/E can be used as an UCCI/E-s, so if you start with the setup per fig 1 and your ambition outgrows the capacity of the small-medium system, you can migrate to the large system without disinvestment by adding an RM-U controller and as many UCCI/E-s and OC32 controllers as required.
The UCCI/E-s is cheaper than the UCCI/E since it lacks the UBS and RS232 interfaces.

If you already own a Dinamo/MCC system, based on the UCCI or UCCI-s controllers (so without /E), you can use these modules intermingled with the new UCCI/E-s in a single system, provided you have installed the latest firmware on these modules. Please contact our support pages on http://www.dinamousers.net for details.

2 UCCI/E(-s)

2.1 Overview

UCCI/E is a single-board central controller providing the following functions:

- Communication with a PC via RS232 or USB
- Driving 2 transmission loops
- Reading128 position sensors
- Controlling up to 16 OC32 modules

UCCI/E is fully compatible with the Dinamo 3.0 protocol, but only supports those commands relevant for car-control.

UCCI/E-s is a stripped-down version that can only be used as a slave-unit. It lacks the USB and RS232 interfaces and therefore is slightly cheaper than the normal UCCI/E. As pointed out in paragraph 1.3 the UCCI/E-s is typically used to build a large system based on the RM-U as a central module.

UCCI/E and UCCI/E-s is delivered as an assembled module only.

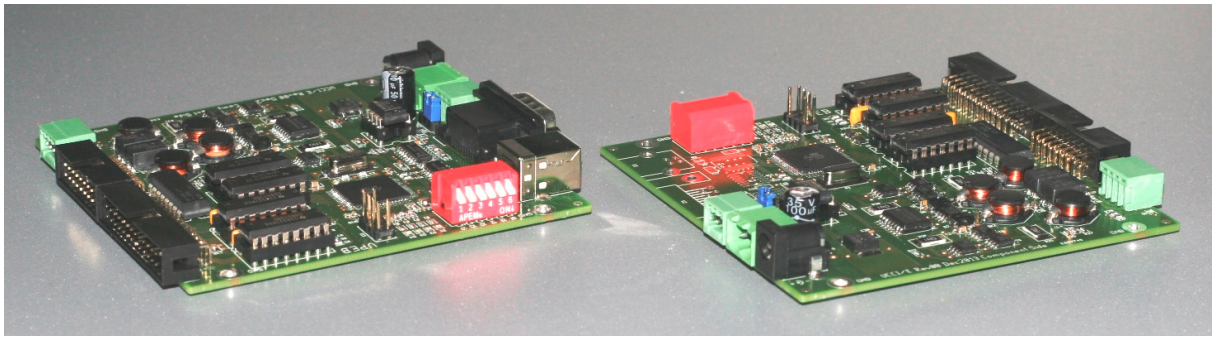

Fig 3: UCCI/E (left) + UCCI/E-s (right)

Note: Normally on UCCI/E you will not find the DIPswitch (red in the above picture), since it only has a purpose on the UCCI/E-s. In case you eventually want to use the UCCI/E as an UCCI/E-s the function of the DIPswitch can also be provided by a software-setting.

2.2 Mounting

The UCCI/E(-s) is meant to be mounted in the direct vicinity of or below your miniature world. Especially if you are going to use the USB interface for connection to the PC, make sure to keep the distance between your UCCI/E and your PC as short as possible, since USB is rather susceptible to electronic interference.

UCCI/E(-s) has 4 mounting holes (3mm). Preferably use spacers (approx 10mm length) to mount the UCCI/E(-s). Especially when you mount UCCI/E(-s) on a metal surface, make sure the rear of your UCCI/E(-s) does not get in touch with the mounting surface.
When mounting UCCI/E(-s) on a metal plane, mind that the mounting holes are connected to the 0V/GND potential of the module en thus that any metal surface will carry the same electrical potential.

UCCI/E(-s) modules are "stackable" by using M3 x 20mm spacers between the modules. You can also stack UCCI/E(-s) and TM44 modules in case you want to control both your cars and trains by a single Dinamo system.

When mounting the modules, make sure that you have enough space around the module to plug the connectors and in case of an UCCI/E-s make sure you can reach the DIPswitch on the side.

2.3 Power supply

The UCCI/E(-s) shall be powered by a **DC power supply**. The Voltage needs to be between 7.5V and 12V DC. The power supply does not necessarily have to be stabilized, but it needs to be DC and the ripple needs to be less than 0.5V.
When the supplied voltage is too low, your UCCI/E(-s) may not function correctly. A voltage higher than 13V may irreversibly destroy your UCCI/E(-s), so in practice, do not apply more than 12V DC.

The supply current required by UCCI/E(-s) depends on the supply voltage, A higher supply voltage will require less current to be drawn from the power supply. At 8V UCCI/E(-s) will rarely use more than 600mA. At 12V the consumption usually will be less than 450mA.

You should not, but if you really care to calculate the power consumption of UCCI/E(-s), the formula is

$$I_S = 50mA + 0{,}4 * (I_A * (R_A + 4) + * I_B * (R_B + 4)) / V_S$$

Where:

V_S is the power supply voltage (V)
I_S is the required power supply current (mA)
I_N is the current setpoint (mA) in loop N (see paragraph 5.5.4)
R_N is the resistance (Ohm) of loop N (see paragraph 3.2)

UCCI/E(-s) has 2 connectors for power supply. A 2-pole (green) connector with screw-terminal and a round socket. You may use either one, depending what is attached to your power supply unit. Use the one that is most practical in your situation. The 2.1 mm round socket is meant for standard wall-outlet power supplies. Usually one of the plugs provided with these units fits. Make sure the center pin is plus and the ring is minus. If in doubt, check with a multimeter before applying the power supply to UCCI/E(-s).

If you have a power supply without connector you can attach the wires to the rectangular power-supply socket. A matching plug with screw-terminals is included in your UCCI/E(-s) package. Note the correct polarity as indicated in fig 4 below.

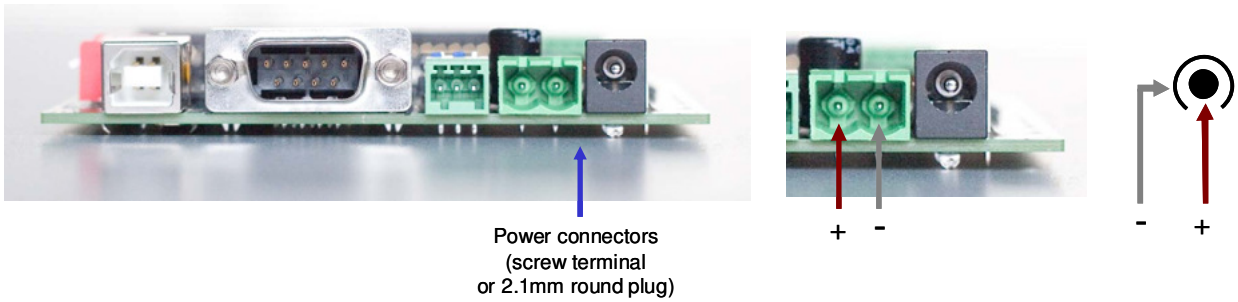

Fig 4: Power supply sockets on the UCCI/E(-s)

WARNING: Connecting an AC or an incorrectly polarized DC power source may lead to irreversible damage to your UCCI/E(-s), your power-supply or both. So make sure you know what you're doing!

2.4 Connecting UCCI/E to your PC

PC Communication can be established by means of a serial RS232 connection or a USB interface. The picture below indicates the position of the connectors to be used.

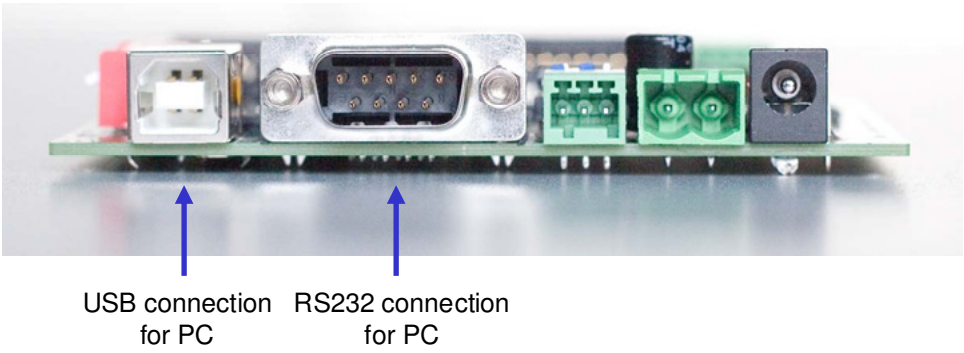

Fig 5: PC interfaces on UCCI/E

When using the RS232 interface you need to use a so called null-modem cable. If you don't have one you can simply make one yourself according to fig. 6 below. The gray connections are not used by UCCI, so just 3 wires will do.

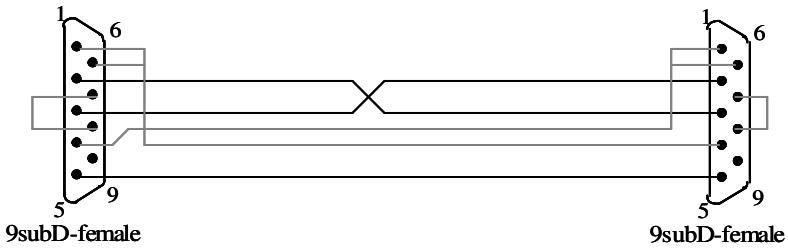

Fig 6: Nullmodem cable

The second possibility is using the USB interface. You can use a standard USB A-B interface cable. For stability reasons we highly recommend to use a USB cable shorter than 3 meters.

Note: UCCI/E is not powered from the USB port, so also when using USB you'll need to apply a power supply as described in paragraph 2.3.

Before connecting the USB interface you need to install the correct drivers on your PC. These can be downloaded from the website of Future Technology Devices or the Dinamo User Group site:

http://www.ftdichip.com/Drivers/VCP.htm

The easiest way is to use the "setup executable for default VID and PID values".

First install the driver package from the above links. Make sure UCCI has power supply and **next** plug in the USB cable. The interface now shall install automatically. During installation of the driver the blue LED on UCCI/E will flash a few times. When the driver is active and has established connection with UCCI, the blue LED on the module will light permanently. Your interface is ready for use.

2.5 Controlling OC32 or OM32 Serial modules from the UCCI/E

For communication with other modules, UCCI/E is equipped with a network interface. The network interface supports either RS485 or RS232, but not simultaneously.

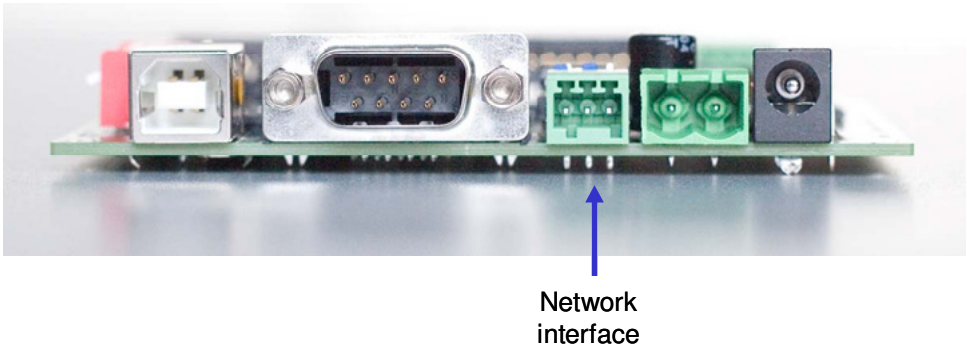

Fig 7: Network interface

On an UCCI/E, the network interface can be used to connect OC32 or OM32-Serial modules in order to control turnouts, traffic lights and other accessories from your Dinamo/MCC system.[1]

The network interface can be operated in RS485 or RS232 mode. By default, the interface

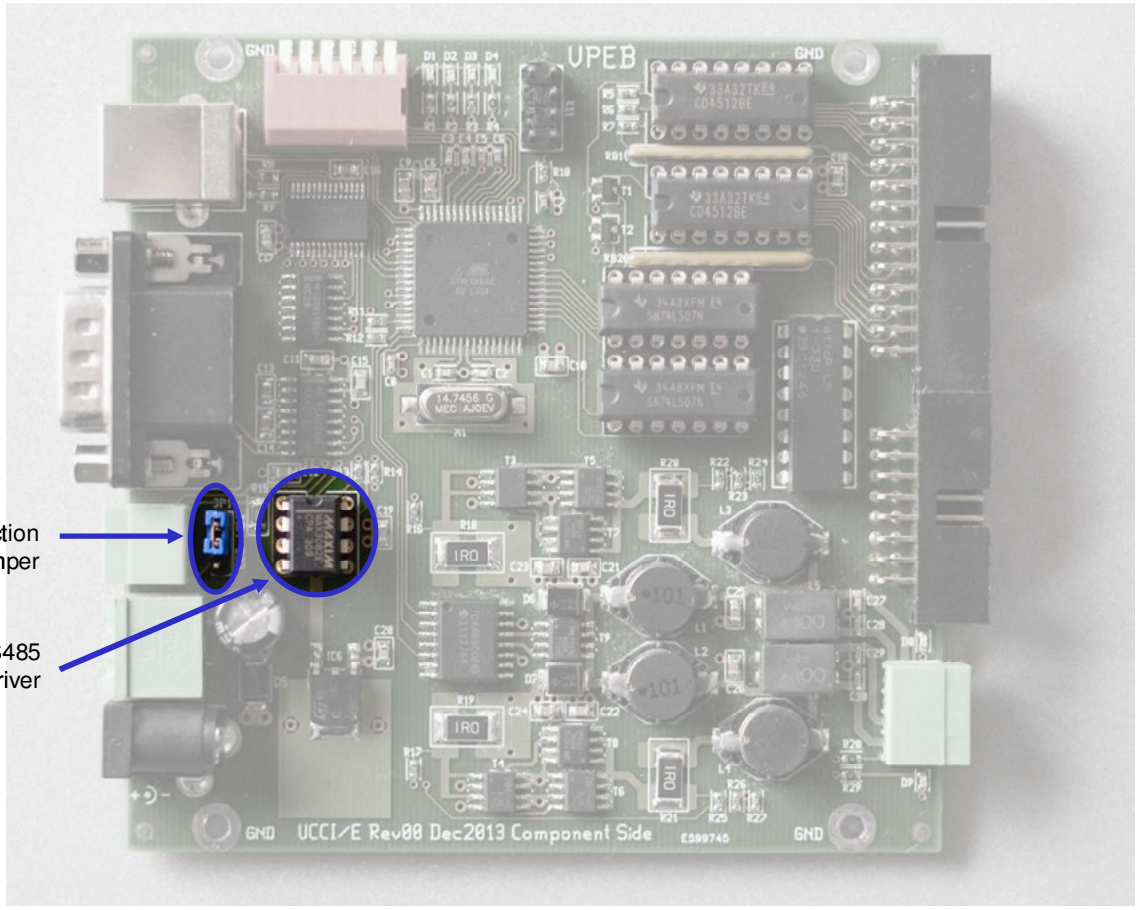

Fig 8: Network interface configuration

[1] Note the OM32 Serial is not produced anymore and replaced by the OC32. But if you own an OM32 Serial already, you can use it with the Dinamo/MCC system

Is in RS485 mode and normally, you will connect OC32 modules using RS485. In that case the MAX3082 chip (indicated in fig 8 as RS485 driver) shall be in place.

Adjacent to the driver chip you'll find a jumper (selection jumper). This jumper can be in 3 postions:

- RS232
- TER (termination)
- Not present. In this case you can "park" the jumper on one pin only. And leave the other free.

For RS485 operation, the jumper shall **NOT** be in the RS232 position.

RS485 is a 2-wire bus-structure (please refer to paragraph 2.6 or the OC32 manual for more detail). The "bus" starts at one station (module), passes all stations that need to be on the network and ends at the last station. The first and last module on the bus shall "terminate" the RS485 bus. Termination is done by putting the selection jumper in position "TER". So if UCCI/E is on either end of the bus, "TER" shall be activated, when UCCI/E is somewhere in between, the jumper shall be "not present".

In case the RS485 interface on your OC32 modules is already used for other purposes or in case you want to connect OM32 serial modules that have no RS485, you can switch the network interface to RS232 mode. Note that in RS232 mode the UCCI/E can only transmit data to the attached modules and thus communication will be unidirectional.

To switch the interface into RS232 mode, remove the MAX3082 chip from its socket (store it in a safe place, should you need it again in the future) and put the selection jumper in position RS232.

The pin-outs of the network-connector is shown in fig 9 to the right. Note that in case of RS485, all 3-pole connectors on Dinamo modules (UCCI, UCCI/E(-s), OC32, U485) are identical, so all cables shall be straight (pin 1 to pin 1, pin 2 to pin 2, pin 3 to pin 3). The use of pin 3 is optional and not necessary when all modules have their GND potential interconnected by the power supplies (which is highly recommended !).

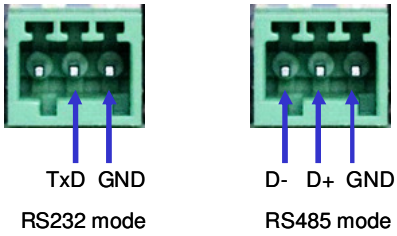

Fig 9: Network interface pin-out

2.6 Connecting UCCI/E-s modules to the RM-U controller

An UCCI/E-s module shall be controlled from an RM-U central unit as indicated in fig 2. The connection is made by RS485 via the network-interface.

If you want to use an UCCI/E as an UCCI/E-s, make sure the network interface is configured for RS485 as described in paragraph 2.5 above. On an UCCI/E-s there is no other option than RS485 and in most cases, on an UCCI/E-s, the RS485 driver is soldered to the board and cannot be removed.

2.6.1 RS485

RS485 is a serial bus for the transmission of signals over (relatively) 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 very useful 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 between the two wires.

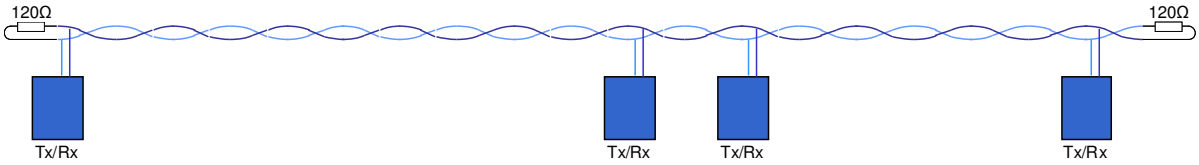

Fig 10: RS485 bus

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 addition to the twisted pair, RS485 also should include 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.

Most cables that can be purchased ready-made, 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 the third wire, if necessary.

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

2.6.2 Connecting RM-U, UCCI/E-s and OC32 modules

The most practical implementation is to mount the RM-U somewhere near your PC and the UCCI/E-s and OC32 modules below or behind your layout. We assume all your UCCI/E-s and OC32 modules are on the same GND, so (at least) the minus-poles of all power supplies are interconnected. The RM-U is powered by the USB bus from your PC and is not necessarily on the same GND level.

Run a twisted-pair cable + a third (GND) wire from the RM-U to the first module at your layout. Next, continue with the twisted-pair only (so without 'the third wire') across all other modules until the last. Activate the termination-jumper on the RM-U, activate the termination jumper on the last (UCCI/E-s or OC32) module and de-activate all other termination jumpers.

Note that the polarity of the wires in the twisted pair does matter, so connect D+ to D+ and D- to D- on all modules. If you interchange them, nothing gets damaged, but it does not work.. When connecting to the RM-U, the pin-out depends whether you have the "standard" or "Plug&Play" version of the RM-U. Please refer to the RM-U or Dinamo P&P manual for details.

Function	OC32/UCCI 3 pole	RM-U (std) 9-subD	RM-U (P&P) 9-subD
D-	1	7	3
D+	2	8	2
GND	3	5	5

Table 1: Connector Pinouts

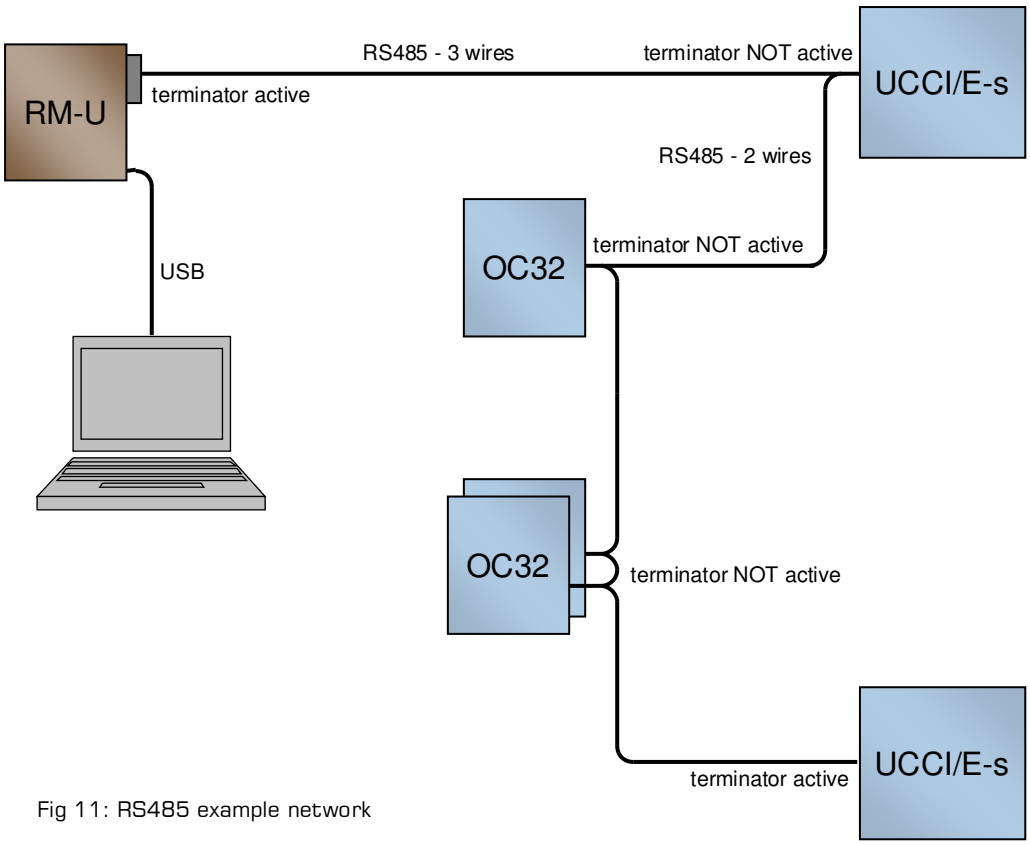

Fig 11: RS485 example network

2.6.3 Addressing

Since all modules are on the same pair of wires, when sending information between two stations, there needs to be a way to determine which station should receive the data. Therefore each module on the bus has a unique address. There can be 16 UCCI/E-s modules and 16 OC32 modules on the same bus. UCCI/E-s modules and OC32 modules are in a different "address space", so addresses can overlap (so you can e.g. have UCCI/E-s address 0 and 1 and OC32 address 0, 1 and 2 on the same bus [2]).

The address on an UCCI/E-s can be set in two ways. The easiest way is setting by DIPswitch. Switch 5 shall be in the ON position and switches 1 through 4 will determine the module address as per table 2 below. Mind that the changing the DIPswitches requires a restart of the module before the new settings become active!

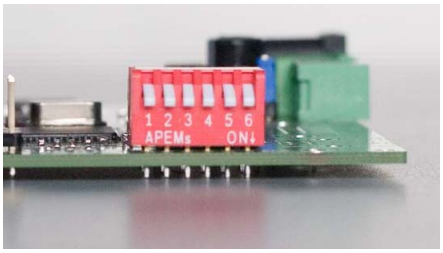

Fig 12: Dipswitch

Address	S1	S2	S3	S4	S5
0	On	On	On	On	On
1	Off	On	On	On	On
2	On	Off	On	On	On
3	Off	Off	On	On	On
4	On	On	Off	On	On
5	Off	On	Off	On	On
6	On	Off	Off	On	On
7	Off	Off	Off	On	On
8	On	On	On	Off	On
9	Off	On	On	Off	On
10	On	Off	On	Off	On
11	Off	Off	On	Off	On
12	On	On	Off	Off	On
13	Off	On	Off	Off	On
14	On	Off	Off	Off	On
15	Off	Off	Off	Off	On
Software	X	X	X	X	Off

Table 2: UCCI/E-s addresses

[2] If you build a system with both UCCI/E-s and TM44 modules, these modules share the same address space. Please refer to the TM44 or Dinamo Plug&Play manual for details.

On an UCCI/E you will normally not find the DIPswitch. If there is no DIPswitch or when SW5 is left in the OFF postion, the modules address is set by software. Please refer to paragraph 5.5.3 for details. In an UCCI/E(-s) the default module address is 0[3]. So if you migrate from an UCCI/E based system to an RM-U + UCCI/E-s based system, the UCCI/E that now becomes an UCCI/E-s automatically gets address 0, which is probably what you want.

[3] Note this differs from the UCCI-s, where the default address is 15.

3 Transmission loops

3.1 The basics

To allow the road to transmit commands for the cars using it, the road-surface needs to be equipped with an 'antenna'. This antenna consists of 2 parallel copper wires in the road-surface on both sides of the steering wire. To be clear: “Steering wire”, as meant in this chapter, can be steel wire as originally deployed by the Faller Car System or magnetic strip, as it is nowadays more becoming a standard. Since the magnetism of the magnetic strip is permanent and the signal transmission relies on changes in electromagnetic fields only, the permanent magnetism of the strip has no influence at all.

The distance between the wires must be around 3 cm (it does not have to be very precise, everything between 2,5 and 4 cm works fine. The exact distance of the wires to the road-surface is not crucial, however the reception is reliable until 3 to 4 cm above the plane formed by the wires, so it is important to keep the distance as small as possible.

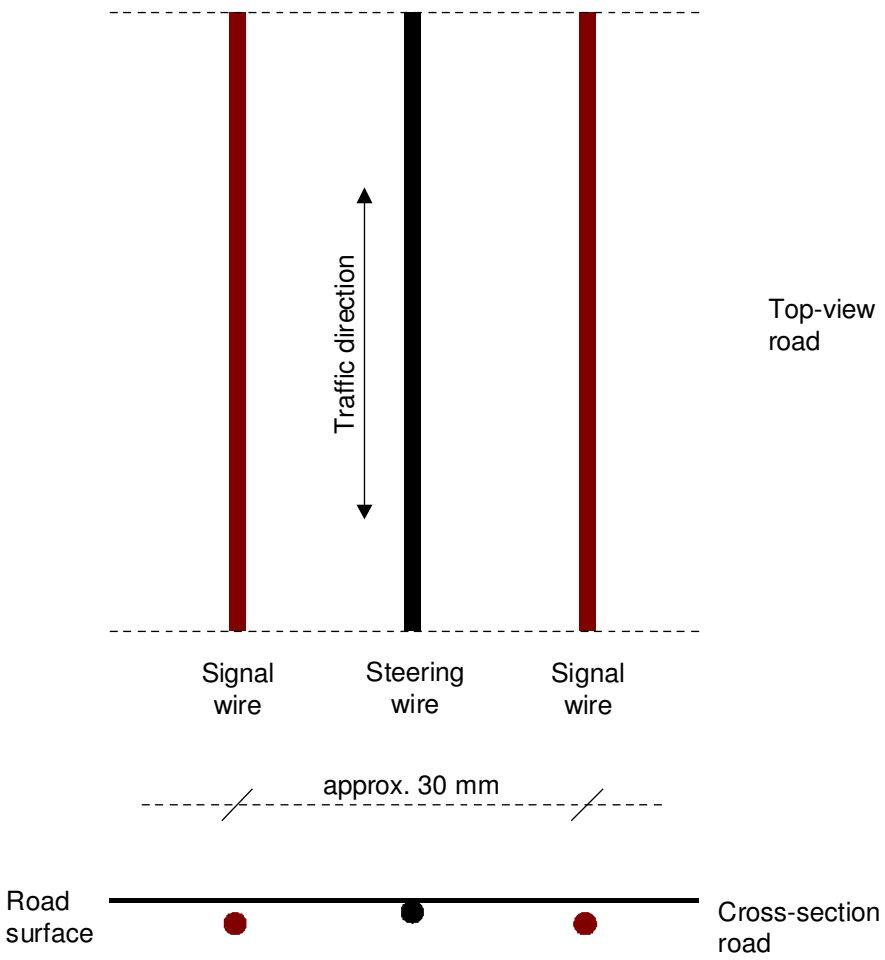

Fig 13: Building the communication system

The copper wires will conduct electrical current. All wires in the road are electrically wired in 'series', so in the end you can look upon it as one very long wire, beginning and ending at UCCI/E(-s). When laying the wires, there are two basic rules to follow:

1. The direction of the current in each pair of wires of the same lane shall be opposite
2. When two lanes are near eachother, the current in both wires of different lanes that are closest together shall flow in the same direction.

Graphically the above rules can be expressed as follows:

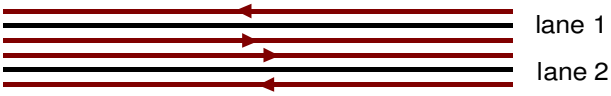

Fig 14: Current direction in a multi-lane road

As a rule of thumb, rule nr 2. applies when the distance between the two wires of different lanes is less than 6 cm

So before starting to wire your layout it is of importance to think ahead.

A simple 'oval' is easy. You choose an arbitrary point where the wires are fed below the layout. The 2 wires 'going to the right' are simply interconnected underneath the road. The remaining 2 are connected to one of the transmission-loop outputs of UCCI. As a connection cable between the connection point and UCCI it's best to use a twisted-pair wire, e.g. a pair from a UTP LAN cable.

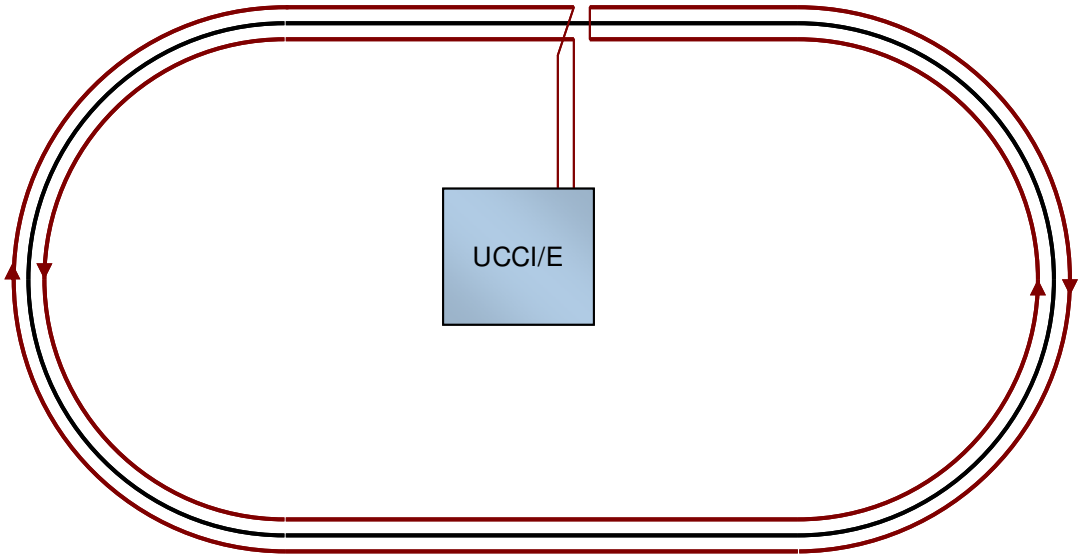

Fig 15: Wiring an 'oval'

In case of a double oval rule nr 2 comes into play. The current in the 2 wires in the middle needs to flow in equal direction. At the same time the entire wiring needs to be one continuous wire. Fig 16 shows an example how this problem can be solved.

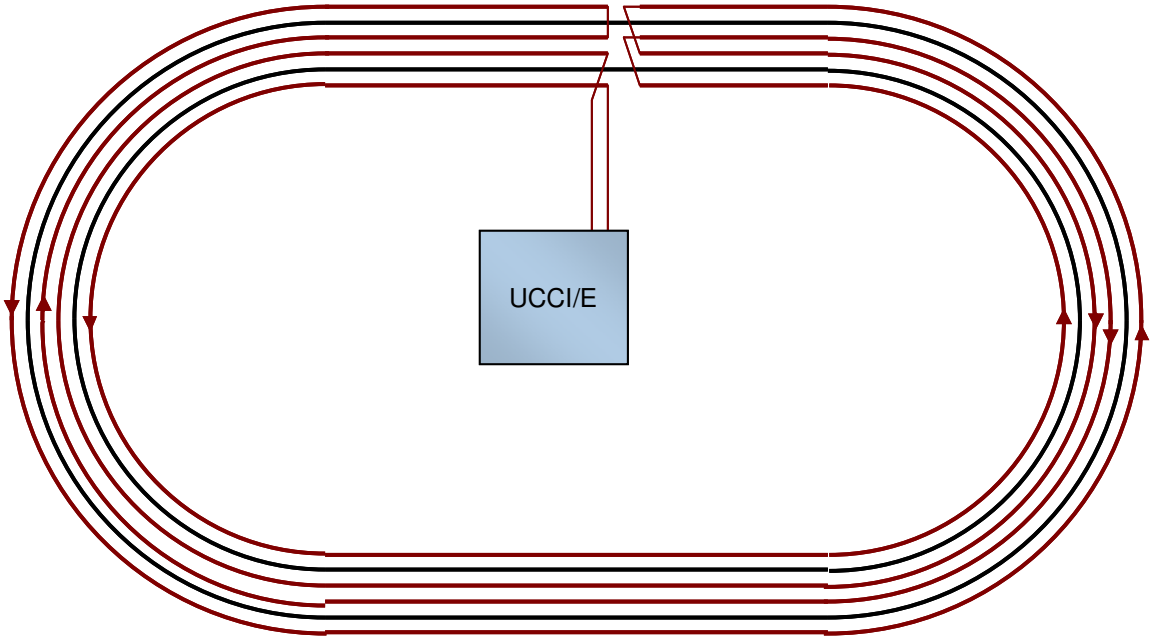

Fig 16: Wiring a double oval

If you have 'turnouts' in your layout the puzzle gets somewhat more complicated, but hopefully the example if fig 17 gives a clue how to cope with these situations. Keep in mind that at 'turnouts' and crossings you'll probably have less than optimal reception since the transmission wires cannot run in parallel over there. However if the distance over which reception is 'bad' is less than 10 cm or so, this will not cause any problem. If necessary you can adapt the 'timeout'in the decoder. Just remind not to plan stopping points at these spots. Your cars will very likely stop as desired, but you'll never be able to give the command to make them move again.

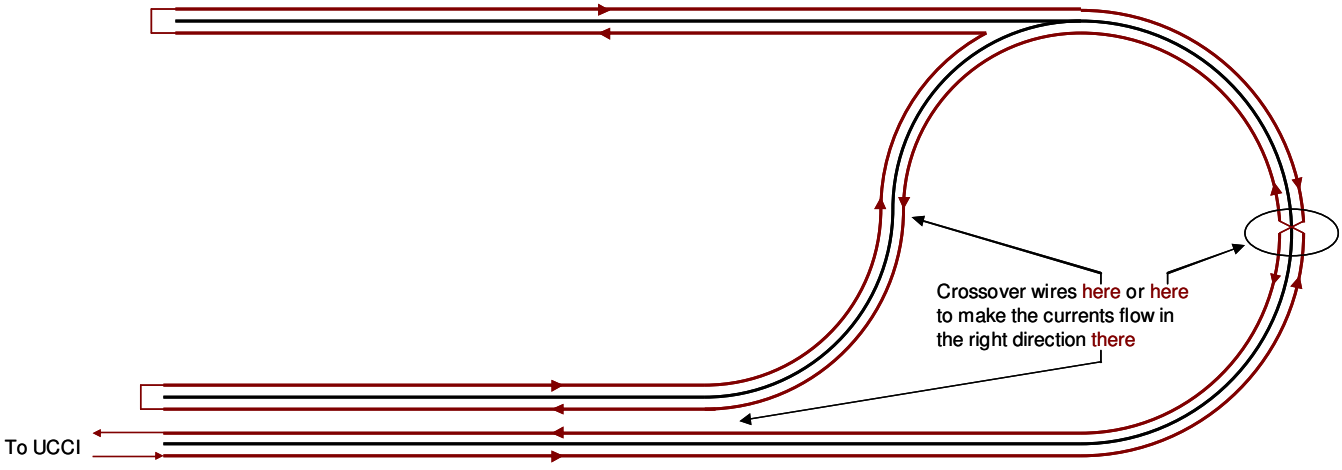

Fig 17: Wiring a more complex situation

If you have to make wire crossings, like in fig 17 above, make the crossing at an arbitrary point somewhere in a lane. Preferably, do not make cossings at junctions. At those spots the situation is already a little challenging, so do not add complexity to those spots.

3.2 Loop length, wire size and connection to UCCI/E(-s)

The maximum length of a single loop is 20 meters of road (lane), so roughly 40 meters of wire. UCCI/E(-s) has 2 outputs for transmission loops, so you can connect 2 loops of 20 meters (road) each, giving a total of 40 meters per UCCI/E(-s) (80 meters of wire).

The 20 meters per loop is no absolute number. It has however been found in practice that the probability to have a good communications system drops rapidly on (road) lengths over 20-25 meters.

The wire used can actually be any (copper) wire. Preferably use insulated wire, to avoid leaking-currents and unintended short-circuits. Also the insulation protects your wire from physical damage.

The cross-section (diameter) of the communication wire should be around 0,25mm^2 (or AWG24 in US sizes). Wire from UTP LAN cable usually is 24 AWG and well suited for this purpose. Thinner or thicker wire, anything between 0.14mm^2 and 0.35mm^2, may be used as well. Thinner wire has a higher resistance per meter and will require more energy from the UCCI/E(-s) to 'push' the current through the loop. So the thinner the wire is, the more current UCCI/E(-s) will draw from your power supply.

Size (mm^2)	Size (AWG)	Resistance (Ω/m)
0,35	22	0,05
0,22	24	0,08
0,14	26	0,13

Table 3: Resistance of wire (copper)

On UCCI/E(-s) you find two interfaces for transmission loops, named Loop A and Loop B, both are on the same connector. Each loop has a + and – connection.

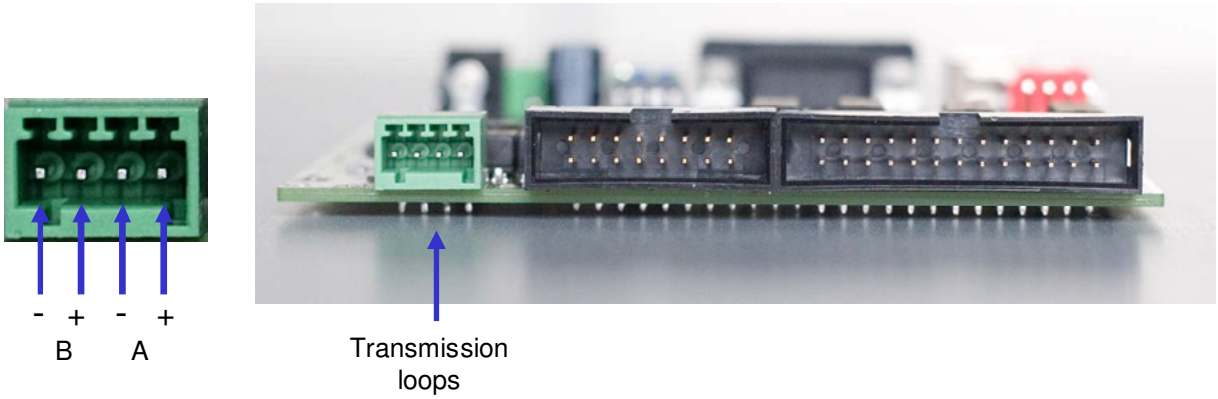

Fig 18: Transmission-loop outputs

It is very important that both loops are not intermingled. So connect one loop to A+ and A- and connect the other loop, if present, to B+ and B-. Connecting a single loop to either A+ and B- or to B+ and A- may lead to severe damage to your UCCI/E(-s). This is considered to be a user error and not covered by warranty.

Which wire is + and – is irrelevant, however, if different loops feed two adjacent lanes it is important that the currents in the wires closest together flow in equal direction (rule 2 from paragraph 3.1), so mind this when choosing the + and -.

You'll probably need to cross some distance between the UCCI/E(-s) output and the point where your loop starts/ends on your layout. You can bridge the distance with a twisted pair of wires. Several meters is no problem and the length of that cable will not decrease the maximum length of your loop. As long as you keep both wires twisted, the cable won't transmit any substantial signal, so hardly any energy is lost.

On both sides of the Loop-output connector you will find a greed LED. For each loop this LED indicates the following:

- Off = Loop disconnected (or UCCI/E(-s) not powered)
- On = Loop active and OK
- Flash = Loop resistance is to high (you probably have a bad connection somewhere)
- Quick flash = UCCI/E(-s) supply voltage is to low in relation to the loop-resistance

3.3 Small scales

The basics as described in paragraph 3.1 are applicable for scale H0 (1:87) or similar. When you apply the Dinamo/MCC system to scale N (1:160) or a scale close to N, you will not be able to meet the required distances on a multilane road. In scale H0 a lane is around 5 cm wide, so two transmission wires, 30mm apart, will fit nicely within a lane. In scale N a lane is around 2.7 cm wide. On a single lane road there is no problem. You can put the transmission wires just besides the road, so they will be around 25 mm apart, which is fine. However, on a double-lane road it does not fit!

The most common solution is found in fig 19 below. On a multilane road, plan two steering wires and thus two lanes between a single pair of transmission wires. To achieve this, the transmission wires have to be placed further apart than the recommended 30 mm. The result will be that you will experience sub-optimal reception in the shaded area, however, in that shaded area there will be no traffic, unless a vehicle passes from one lane to the next. In those rare instances you should make sure that the passage does not keep the car too long in the shaded area, so the car is in the new lane before the decoder-timeout triggers.

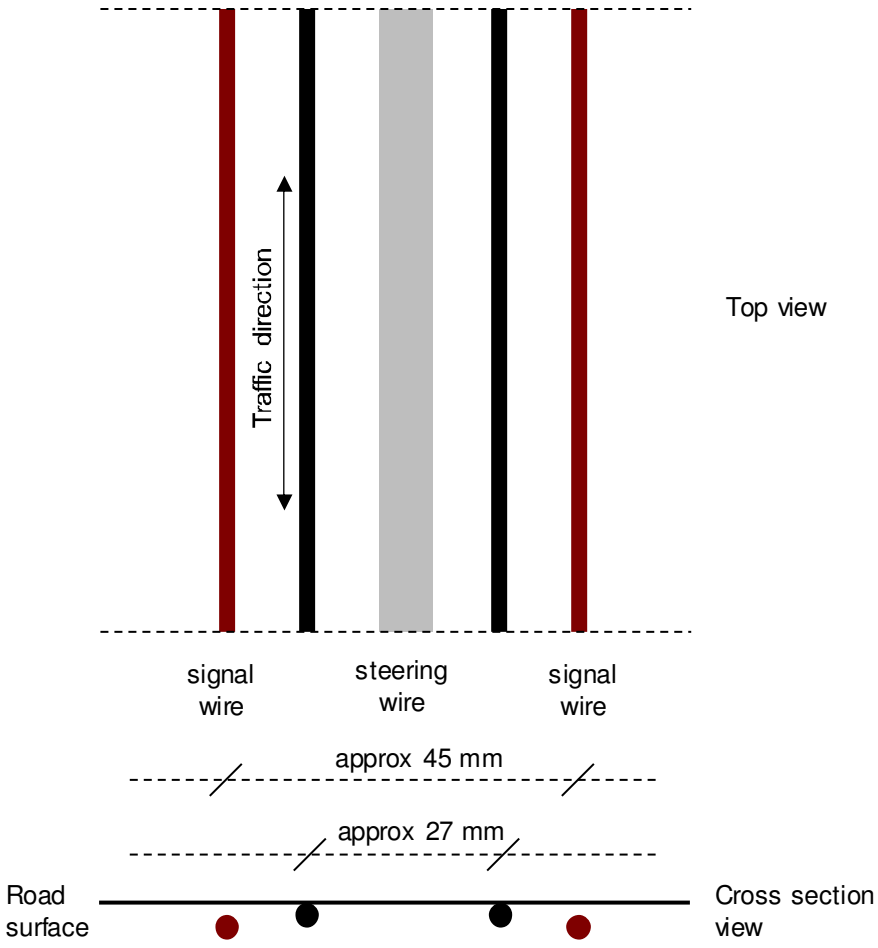

Fig 19: Signal wires in small scales

4 Feedbacks

The position of cars on the layout can be tracked by means of reed-contacts. In most cases the 'steering magnet' of the car can be used to activate the reed contacts when the car passes.

A suitable size is approx. 14 x 2,1mm or smaller. Beware that smaller reed-contacts can be quite fragile and the glass can break if you bend the leads without care. Use small pliers or tweezers to hold the lead between the glass and the point where you bend the lead to avoid breaking the glass.

UCCI/E(-s) can read 128 switches (reed-contacts) for feedbacks. Switches are connected by multiplexing, which means not every switch has its own wire, but wires are shared to save electronics and wire.

4.1 Installing reed-switches when using steel-wire for steering

If you use steel-wire as a guide-wire and want the steering magnet to trigger the feedbacks, a precise mounting of your reed-contacts is required. You must make sure that the reed-contacts are laid directly next to the steel-wire and very close to the road-surface (see fig. 20).

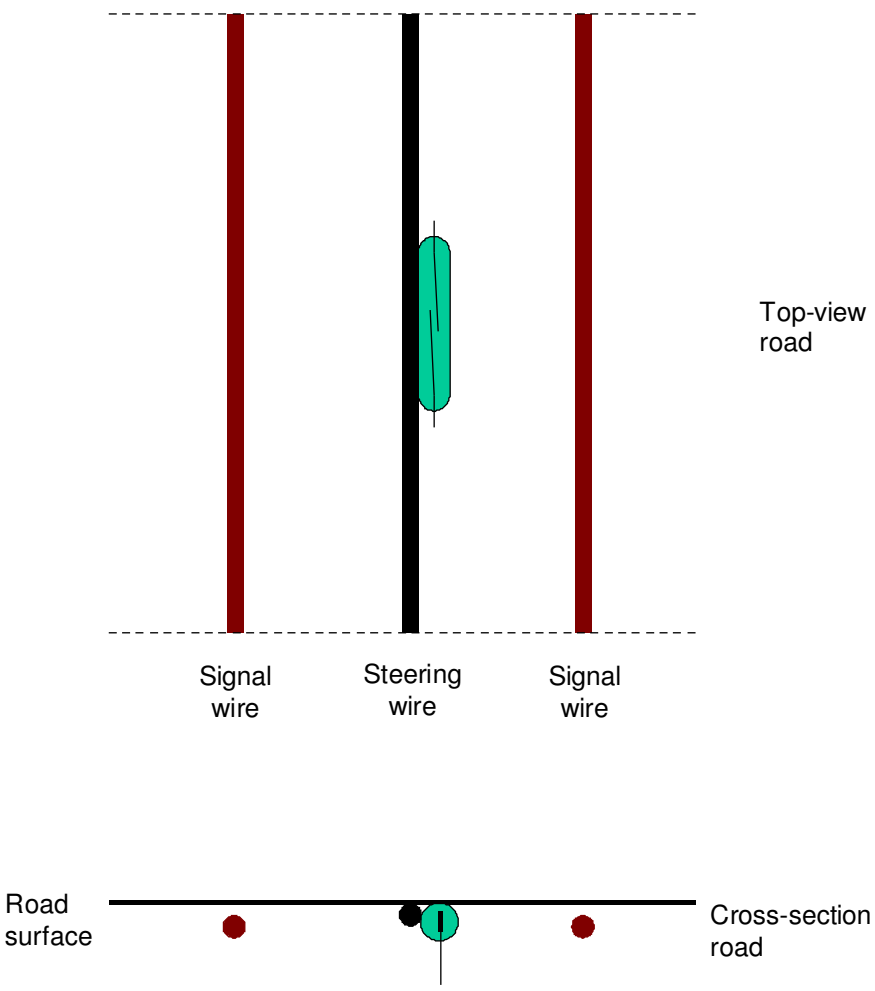

Fig 20: Placing of reed-contacts when using steel-wire for guidance

This way of activating reed-contacts works well with the steering magnet of Faller trucks and busses. The smaller cars have less powerful magnets that are too weak to activate the reed-contacts. In these cases you'll have to mount an additional magnet underneath the car.

Of course you can mount the reed-contacts differently, but in most cases you'll need an extra magnet under the car to activate them.

4.2 Installing reed-switches when using magnetic strip for steering

If you use magnetic strip as a guide-wire, the advised positioning of reed-contacts is different. Since the strip is a magnet by itself, it already introduces a magnetic field in the reed contact. This pre-magnetisation can be used to your advantage to increase the sensibility of the reed-contacts. If positioned well, even the smallest steering magnets can activate the feedbacks.

Start by bending one leg of the reed-contact and flip it back along the side of the switch. Solder wires (insulated) of sufficient length to both legs of the contact. Any wire size will do, smaller diameter will be less risk to physically damage your reed-contacts. When building your layout, it will be practical to prepare your reed-contacts this way before you start, but first try a few to get used to the methodology..

Before laying your magnetic strip, drill a vertical hole exactly at the place where your magnetic strip will be. The drill diameter shall be just large enough so you can push your prepared reed-contact in from below right to the surface without breaking the glass. Something between 2.5 and 3.5 mm usually will do. Do not insert the reed-contact yet.

Next lay your magnetic strip and fix it.

Temporarily connect the wires of the reed-contact to an instrument that can detect the opening/closing of the contact. A multimeter will do. A practical tool is a battery + suitable resistor + LED, you can simply make one yourself. A battery + buzzer may work too. Use what you have available and find practical.

Now insert the reed-contact from below and push it in until the contact closes by the magnetism of the strip. Pull back the contact just to the position when it opens again. Fix the position of the reed-switch. You can do this by e.q. a woodstick (be careful not to break the switch), hot-glue, cleaning-gum (also sold as “poster buddies”). Actually anything that keeps the contact in place, but allows repositioning if required.

The result should be similar to fig 21 below.

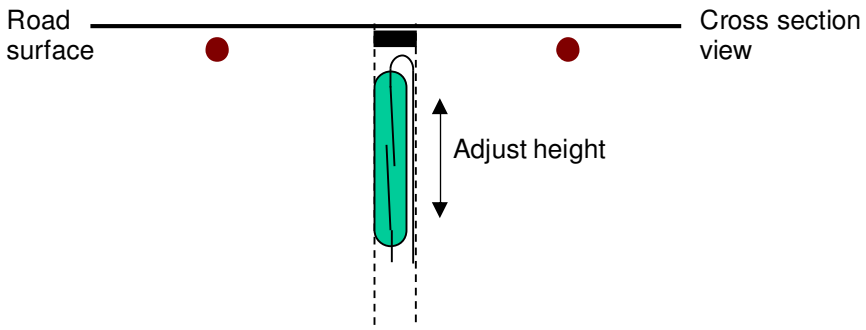

Fig 21: Placing of reed-contacts when using steel-wire for guidance

The magnetism of the magnetic strip sets the reed-contact at the edge or triggering. Now when any magnet passes over the strip this will increase the magnetic field and trigger the switch.

4.3 Connecting feedbacks to UCCI/E(-s)

UCCI can read 128 switches for feedbacks. Switches (e.g. reed-contacts) are connected by multiplexing, which means not every switch has its own wire, but wires are shared to save electronics and wire. When 128 switches are connected individually this would require at least 128+1 wires. When multiplexed as by the way UCCI/E(-s) does, 128 contacts only require 24 wires.

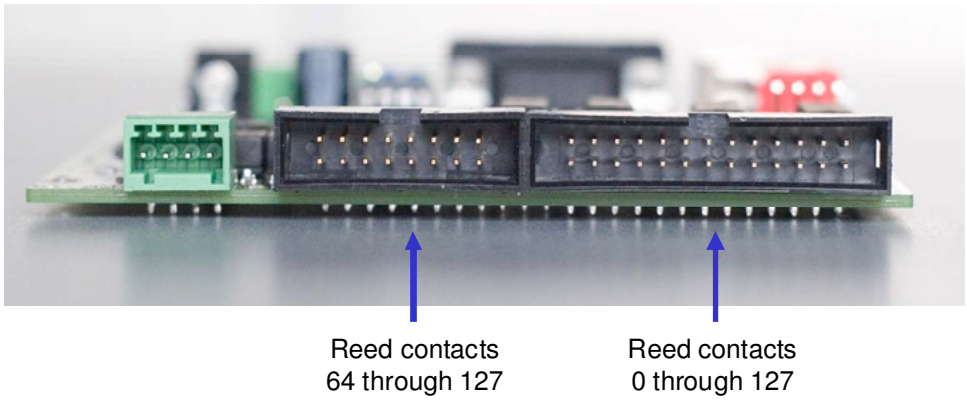

Fig 22: Feedback-Interfaces on UCCI/E(-s)

On UCCI/E(-s) you will find 2 (flatcable) connectors for feedbacks: A 26 pole connector and a 16 pole connector. The 26 pole connector holds the connections for all feedbacks. The 16 pole connector holds the connections for the upper 64 contacts. You can connect all feedbacks to the 26 pole connector only, which requires the least number of wires, however for practical reasons most users choose to connect the lower 64 feedbacks to the 26 pole connector and the upper 64 feedbacks to the 16 pole connector.

The pin-out of the connectors is as follows:

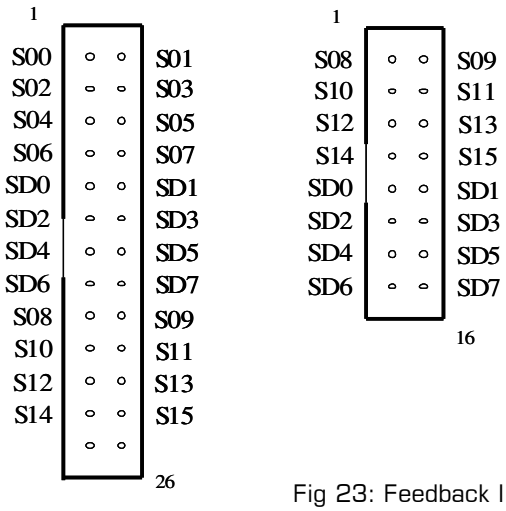

Fig 23: Feedback Interface connector pin-outs

There are 3 groups of connections to be distinguished:

- S0..S7: The SENSE wires for feedbacks 0..63
- S8..S15: The SENSE wires for feedbacks 64..127
- SD0..SD7: The DRIVE wires for feedbacks 0..127

As you can see in fig 23, the sense-wires for feedbacks 64..127 and the drive-wires are available on both connectors. The sense-wires for feedbacks 0..63 are only available on the 26-pole. If you choose to use both connectors, typically on the 26-pole connector only pins 1 through 16 are used.

Every feedback (switch) shall be connected between a unique pair of wires, one being a sense-wire, the other being a drive wire. In series with every switch a diode (type 1N4148) has to be placed, with the cathode directing to the drive-wire. Since there are 16 sense-wires and 8 drive wires, this gives 16 * 8 = 128 unique combinations, thus 128 possible feedbacks.

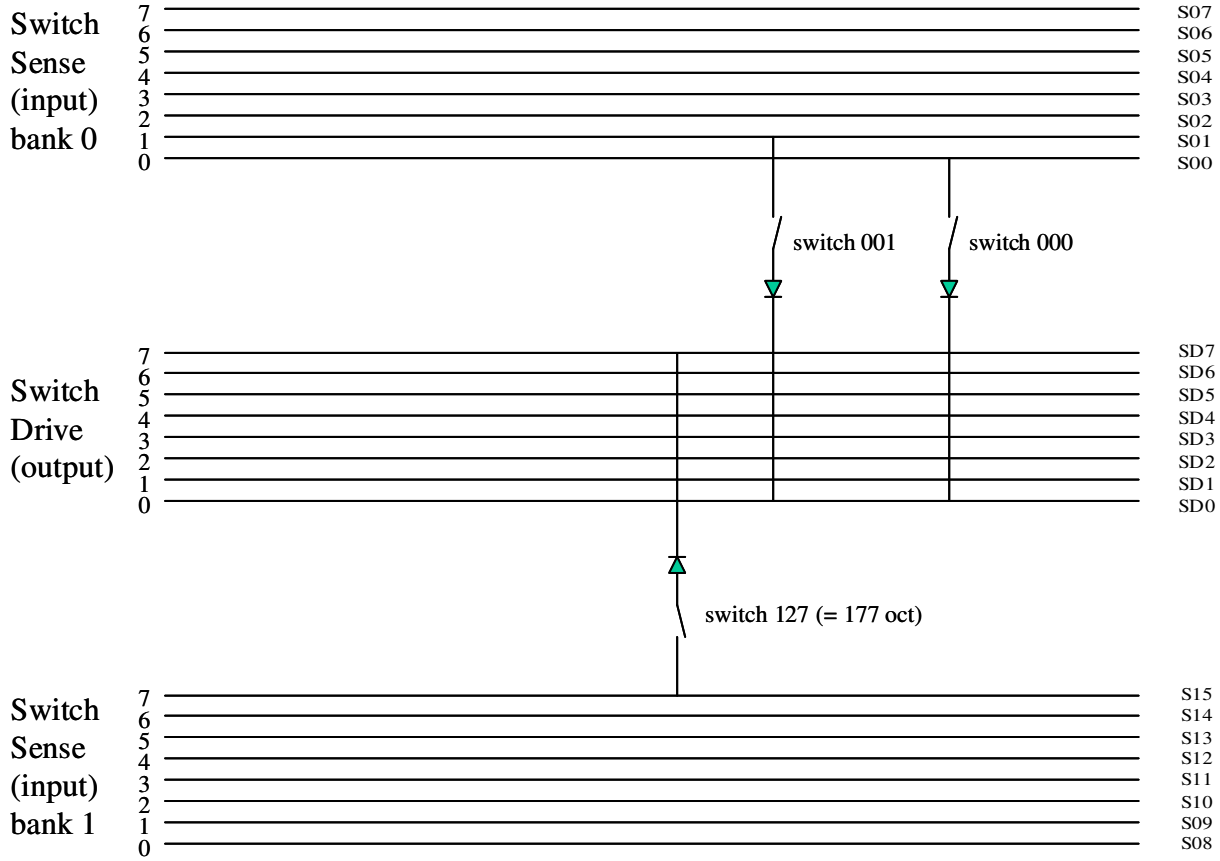

Fig 24: Switch Matrix

The switch-address is determined by the pins (wires) between which the switch is connected. The address in octal notation is <bank>.<switch_drive>.<switch_sense>.

Example: Switch between sense 3/bank 1 (S11) and drive 4 (SD4) = 143 oct = 99 decimal
Tip: If you have trouble with octal-decimal conversion you'll find a calculator in your Windows. Put the calculator in scientific mode et voila!

Of course you can calculate the address, but in practice it is much simpler to just connect all feedbacks and make sure the wire-pairs are unique. Then power-up your system, start-up the testing software and simply test all feedbacks and mark the address reported by the software on your layout-drawing.

If you want, you can make the wiring for the switch matrix yourself. If you rather go the easy-way, there is a break-out module (SWdec) available to which you can connect 8 switches. This PCB also contains the 8 diodes and a 16 pin flatcable connector.
You can connect 8 of these SWdecs onto 1 flatcable for a total of 64 switches. In this setup the separate 16 pin connector on UCCI is handy, since you can use it to connect a second string of 8 SWDECs. The different connector sizes make sure you can't accidentally interchange them.

For SWdec a separate manual is available.

5 Operation and Configuration

5.1 Operation

UCCI/E(-s) is fully controlled by the PC application. The LEDs give some indication of what's happening and may be helpful during troubleshooting:

- Blue: USB active
- Orange: When connected to a PC: UCCI/E sends a datagram to the PC
  When connected as UCCI/E-s to an RM-U central unit: UCCI/E-s sends a datagram to the RM-U.
- Green: Command-processing: UCCI/E(-s) received a command from the host (PC or RM-U) and is processing it.
- Yellow: UCCI/E(-s) sends a message on the network-interface

5.2 UCCI/E: Two logical units

Logically UCCI/E is a combination of two control systems integrated in one physical unit. You will start to notice this when you test and configure the module with DinamoConfig. Download and install the DinamoConfig tool from the VPEB website or the DinamoUser group. Make sure you have version 1.11 or later

Start DinamoConfig and select the com-port to which your UCCI/E is connected. This will either be the real com-poort if using RS232 or it will be the virtual com-port assigned to your USB driver when using the USB interface. When using USB it can sometimes be unclear which com-port is assigned to your UCCI/E. In that case, unplug your USB interface and click the "Refresh" button in DinamoConfig. Now look in the drop-down list next to the com-port box which com-ports are available. Remind them or write it down. Now plug in the USB interface to UCCI/E. Count to 10, click "Refresh" and open the drop-down list. There should be one additional com-poort. That's the one!

Next click the "Status" button. You should now see the pop-up window of fig 25.

System Status in the top-left corner can be "OK" or "Fault". If it shows "Fault" nothing is wrong. It just means that there has been no communication for 2 seconds or more with your UCCI/E and the system has entered an emergency stop. This will solve automatically when you start using your system.

The actual state the status window shows is the state of your RM-U module. But wait, we don't have an RM-U! Well actually the UCCI/E simulates an RM-U with limited functionality + one virtual UCCI/E-s at address 0. The limited-function RM-U can send commands to connected OC32 modules and it can address one UCCI/E-s module. The "connected" UCCI/E-s is shown on the right-hand side as TM#0, Status=OK, Type=TMCC. Actually, the

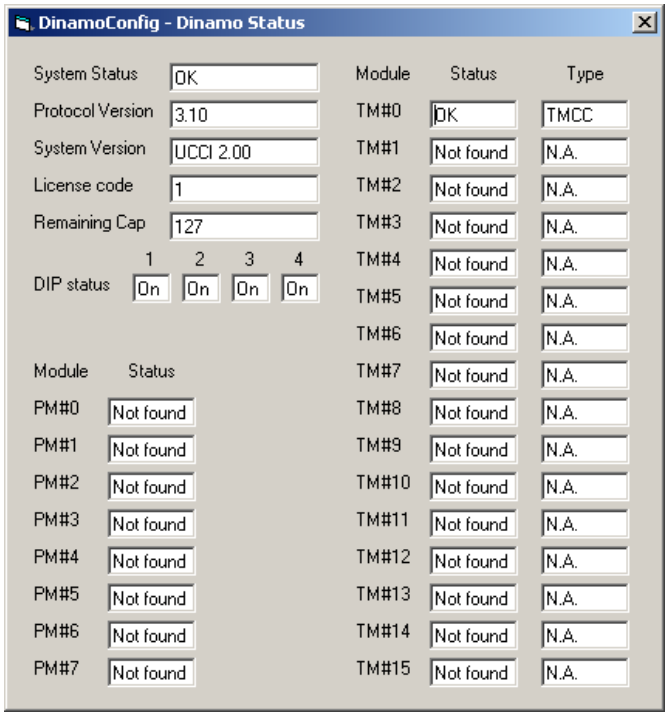

Fig 25: DinamoConfig Status Window

system behaves as it were a true RM-U + UCCI/E-s.

Now close the status window and select the RM-U/UCCI tab in the main window (fig 26). You are viewing the RM-U functional part of the system and the top-right hand corner tells you that it is an UCCI simulating the RM-U function. There are a number of configuration-settings you can make that we will describe later.

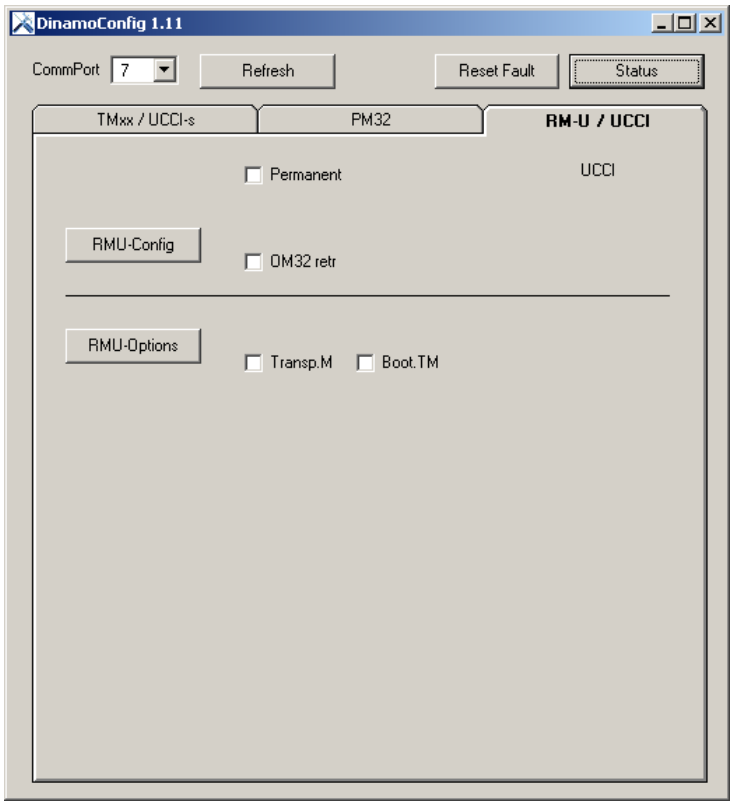

Fig 26: DinamoConfig RM-U/UCCI tab

Select the TMxx/UCCI-s tab (fig 27). In the top-left corner you can select either Module 0 or "All UCCI". Since you have only one virtual UCCI/E-s there's nothing else to choose from. The top-right hand corner tells you that you are viewing the UCCI-s functional part of the system. Again there are a number of configuration-settings you can make that we will describe later.

Note that UCCI, UCCI-s and the TMCC are actually the names of the previous versions of UCCI/E(-s). The reason is that the current protocol (3.10) does not correctly translate the new UCCI/E terminology. This will be corrected in the near future, but is of no influence to the functionality.

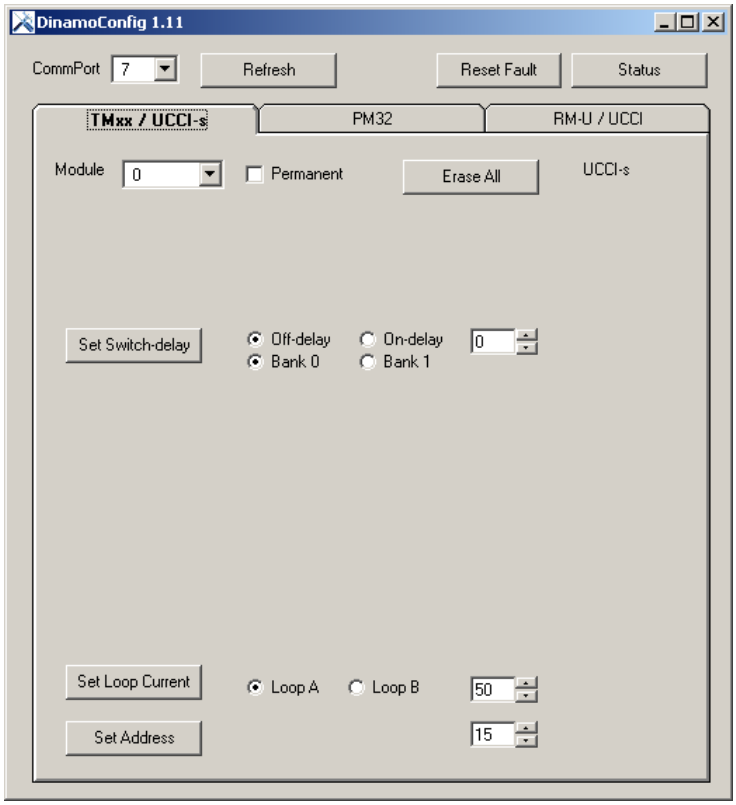

Fig 27: DinamoConfig TMxx/UCCI-s tab

5.3 Configuration

UCCI/E(-s) has a number of configuration-options that can be altered by software. Configurations can be temporary (active until the module is restarted) or (semi-)permanent. With permanent configuration, settings are written in flash memory and reloaded when UCCI/E(-s) is powered up. Permanent settings can be rewritten with new settings.

The flash-memory is specified for approx. 100.000 write-cycles. In normal circumstances this will never be reached in a regular lifetime, but since it cannot be determined how often you 'flash' (you could even write a program doing that) there is no warranty on the user-flash memory of the CPU.

Configuration of UCCI/E happens in a similar way as configuration of Dinamo 3.x modules that support this feature. For configuration the DinamoConfig configuration-tool is available. Since UCCI/E simulates 2 modules (see paragraph 5.2) configuration settings are in two differ area's: The RM-U area and the UCCI-s area

5.4 RM-U configuration settings

These configuration settings apply to the RM-U functional part of UCCI/E and therefore are only applicable when an UCCI/E is connected to the PC as an autonomous system.

5.4.1 OM32 Retransmission

In a Dinamo/MCC system, in the current release, OC32 modules are addressed as if they were an OM32. Since the OC32 is backwards compatible, this works fine. However, the OM32, and thus the OC32 in OM32 compatibility mode, can only receive data and does not send any response. To increase the reliability of communication UCCI/E can send every Ox32 message twice. Of course this will slow down communication somewhat, but since the network interface has nothing else to do except sending messages to Ox32's this will be hardly noticeable. You achieve this by setting the OM32retr flag.

5.4.2 Transparent Mode

If you want to configure and test OC32 modules, connected to the UCCI/E, the OC32Config program needs a "transparent" connection with the OC32 modules. You achieve this by putting the UCCI/E in transparent mode. From that moment until a "reset" of the UCCI/E, the module behaves as plain USB-RS485, USB-RS232 or RS232-RS485 converter.

Transparent mode can be activated by checking Transp.M. and clicking "RM-U options". You can also set Transparent Mode directly from the OC32Config program.

Transparent Mode functions both when using RS232 and USB.
Important notice: When using RS232 the connection speed is fixed at 19.200bps. So when you use OC32Config, use only this speed setting. When using USB the set speed is irrelevant.

Transparent Mode works best when you have connected your OC32's to UCCI/E via RS485 In that case you are offered the full configuration-functionality. If you connect OC32('s) to your UCCI/E by RS232, it works too, but you'll lack the possibility to verify any settings.

While Transparent Mode is active, the orange LED on UCCI/E flashes When sending data from PC to the RS485/RS232 netwerk after UCCI/E, the green LED lights, when sending data from the RS85 network after UCCI/E to the PC the yellow LED lights.

Ending Transparent Mode can be done by rebooting your UCCI/E.

5.4.3 Bootloader Transparent Mode

If you want update OC32 modules, connected to the UCCI/E, with new firmware, the Bootloader program needs a “transparent” connection with the OC32 modules. You achieve this by putting the UCCI/E in Bootloader Transparent Mode. From that moment until a “reset” of the UCCI/E, the module behaves as plain USB-RS485 or RS232-RS485 converter. Note firmware updates only work on an RS485 connection to the OC32’s. So when you have connected your OC32’s to UCCI/E by RS232, you’ll need a separate RS485 to update.

Bootloader Transparent Mode can be activated by checking **both** Transp.M. and Boot.TM and clicking “RM-U options”.

Bootloader Transparent Mode functions both when using RS232 and USB.
Important notice: When using RS232 the connection speed is fixed at 38.400bps. So when you use AVRootloader, use only this speed setting. When using USB the set speed is irrelevant.

While Transparent Mode is active, the orange LED on UCCI/E is continuously lit. When sending data from PC to the RS485 network after UCCI/E, the green LED lights, when sending data from the RS85 network after UCCI/E to the PC the yellow LED lights.

Ending Bootloader Transparent Mode can be done by rebooting your UCCI/E.

To be clear: Bootloader Transparent Mode is **not** meant to update UCCI/E(-s) itself, but to update components sitting behind UCCI/E with new firmware.

5.5 UCCI-s configuration settings

These configuration settings apply to the UCCI/E-s functional part of UCCI/E(-s).

5.5.1 Switch activation delay

Before UCCI/E(-s) reports a closed switch (reed-contact) as active, UCCI/E(-s) needs to have 'seen' the switch closed during N consecutive scans. The 'delay' is primarily intended to avoid noise generating false events. Switches are scanned approx. 180 times per second.
If N=2 (delay=1) a switch must be scanned 2x closed (consecutive).
The Activation Delay of switches kan be set per bank of 64 switches between 0 and 7 (so N=1..8). Default Switch Activation Delay = 1.

5.5.2 Switch release delay

As with switch closure, before UCCI/E(-s) reports a switch inactive, UCCI/E(-s) must have 'seen' the corresponding input inactive for N consecutive times. (almost) Every switch ‘bounces’ during opening or closure. The Switch Release Delay is primarily used as debounce mechanism.
Switches are scanned approx. 180 times per second.
If N=8 (delay=7) a switch must be scanned 8x open (consecutive).
The Release Delay of switches kan be set per bank of 64 switches between 0 and 7 (so N=1..8). Default Switch Release Delay = 7.

5.5.3 Set Address

This parameter sets the address of the UCCI/E-s. You can always set the address but it will only be applicable when you use the system as a real UCCI/E-s, so connected to an RM-U. In case of an autonomous system, the virtual UCCI/E-s always has address 0.

Note(1) If the DIPswitch is installed, you can also set the address by DIPswitch (see paragraph 2.6.3).
Note(2) The address change has only effect after a reboot of the module. So in case of a software address setting only a permanent setting is useful.

5.5.4 Set Loop Current

The current introduced in each transmission loop can be set per loop. The current (mA) is 10x the parameter entered here. So setting 50 results in approximately 500mA. The current can be set between 250mA and 700mA. Default is 400mA, which usually is a good setting. Long loops extending to the maximum length have to spread their transmission energy over a larger area, so may need some more power. If you have loops with different lengths you may set the loop currents differently to have more or less equal reception on both loops. Just try and experiment.

6 Firmware Update

UCCI/E(-s) has a Bootloader provisioning, which means that you can update the module with the latest UCCI/E(-s) firmware when released by VPEB. This chapter describes how to use the Bootloader and how to update your UCCI/E(-s) firmware.

6.1 Requirements

UCCI/E(-s) can be updated via the USB interface or the RS485 network interface.

UCCI/E, when connected directly to the PC, therefore can be updated via the USB interface only. Note in this case, the RS232 interface will not work for the update. If you normally operate your system by RS232, you'll have to connect the UCCI/E to a PC by USB before you continue with the update procedure below. Please make sure that the correct USB drivers are installed and the UCCI/E is recognized by your PC as described in paragraph 2.4

An UCCI/E-s will normally be connected to an RM-U central controller by RS485 and can be updated via this interface. Also an UCCI/E used as UCCI/E-s can be updated this way. The RM-U can be connected to your PC by USB or RS232. Both interfaces will work for the update. Note the RM-U needs to have firmware 1.02 or later (preferably firmware 1.11 or later). If you do not have an RM-U with the required firmware, please update the RM-U first using the procedures described in the RM-U manuals.

To perform the firmware-update the following is required:

a) A PC with the Windows operating system.
   In general, this can be the system you normally use to operate your layout.
b) DinamoConfig1.11 or later (installed on your PC).
   DinamoConfig is a test and configuration tool for your Dinamo and/or Dinamo/MCC system.
c) VPEB Bootloader software (installed on your PC).
   Bootloader software is a program on your PC, by which you can write the UCCI/E(-s) firmware into the UCCI/E(-s) processor.
   NOTE: The VPEB Bootloader software is generic for all VPEB modules that have Bootloader support. If you already installed the Bootloader-software e.g. for your OC32, you don't have to repeat this and you can skip step 1 of paragraph 6.2.
d) The latest version UCCI/E(-s) firmware (to install).
   Firmware is the software that shall actually be installed **in** UCCI/E(-s) and makes your system work.

Regarding b), c) and d):

DinamoConfig, the VPEB Bootloader software and new firmware for UCCI/E(-s) can be found on the DinamoUsers portal (http://www.dinamousers.net). In order to access this software, it is required that you are registrered on the above mentioned portal and that you have Dinamo Customer Status. Registration is free of charge for everyone that accepts our conditions of use and the Dinamo Customer Status will be assigned to you, also free of charge, of you purchased VPEB products, either directly or through one of our partners. If you don't have the right status but are entitled for it, just ask. The procedure can be found on the DinamoUsers portal.

6.2 Firmware Update

Please take the following steps:

1. If you haven't done this before: Install the VPEB Bootloader software on your PC. You can do this by downloading the .zip file and extract it to a folder of your choice. It is advisable to do that somewhere in "Program Files". If desired you can make a shortcut to the extracted AVRootloader.exe. This step has to be taken only once;
2. If you haven't done this before: Install DinamoConfig on your PC. When doing a "fresh install", preferably use DinamoConfig1_11Setup (or later). If you already have an older version of DinamoConfig installed, use DinamoConfig1_11Update or later to update the installed version;
3. Download the UCCI/E(-s) firmware to be installed. Unpack the .zip file. The file you need has extension *.acy. Store that file somewhere on your PC where you can locate it later;
4. Power up your Dinamo/MCC system;
5. If you have an UCCI/E directly connected to your PC by USB: go to step 6
   If you have a system with RM-U and UCCI/E-s: Start DinamoConfig. Select the com-port by which your RM-U is connected to the PC. Klick on "Status". Check that the RM-U version is 1.02 or later (if not, update your RM-U first). Close the "status" window. Select the tab RM-U/UCCI. Switch the RM-U to Bootloader Transparent Mode by selecting options "Transp.M" **and** "BootTM" (both) and click "Options". You'll probably see a warning saying "No response from Dinamo". Ignore this. At the RM-U now the red LED shall be lit (continuously) and the blue LED if you use USB. Close DinamoConfig.
   Note: If you use a more recent version DinamoConfig it may work slightly different. In that case, check the DinamoConfig release notes;
6. Start AVRootloader.exe. You'll see the window similar to fig 28;
   The baudrate is at 38400 and "Sign" shows "VPEBbootloader". **Do Not Change This!**, otherwise it won't work!
   Optionally you may tick "Open protocol-window after processing".

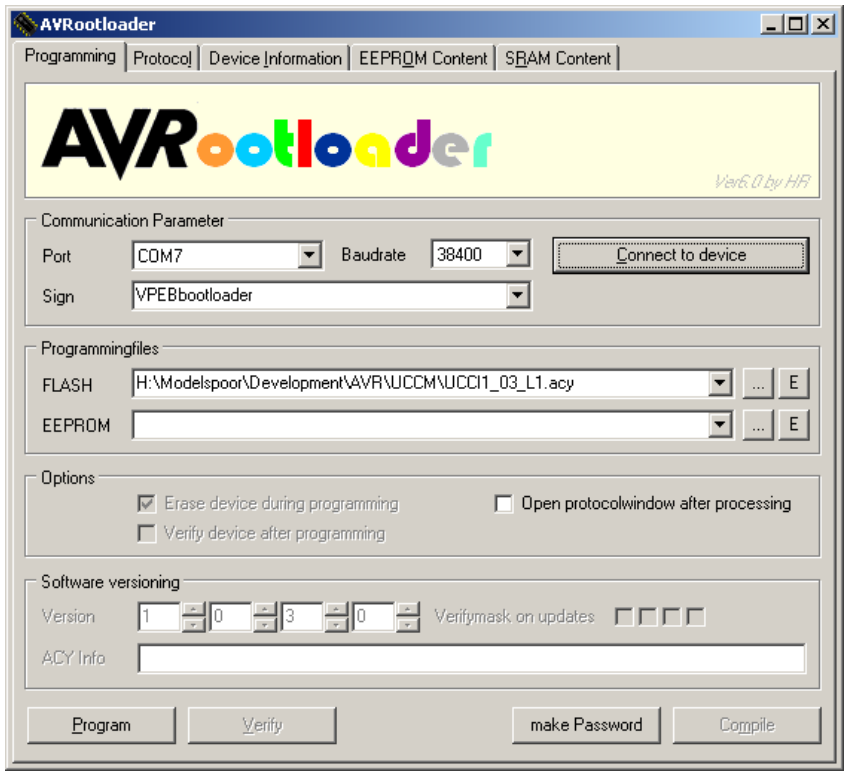

Fig 28: AVRootloader

7. Set "Port" to the com-poort by which your PC communicates to your system. When using RS232 (RM-U only), that will be the real com-port of your RS232. When using

USB, it will be the virtual com-port the USB driver has been assigned to. Normally it will be the same port as the one by which you control your system from your regular software..
NOTE: The selection "AUTO" will not work with the type of Bootloader in your UCCI/E(-s). So you will manually need to select the right com-port.

8. In the box next to "FLASH", select the *.acy file you saved at step 3 above. You can do this by clicking "…" at the right side of the text box and selecting the right file.
9. Reset the UCCI/E(-s) you wish to update. Do this by placing a jumper on the Reset pins (fig 29) and leave it there. Usually there is a spare jumper you can use, parked on one of the pins of the jumper-block.

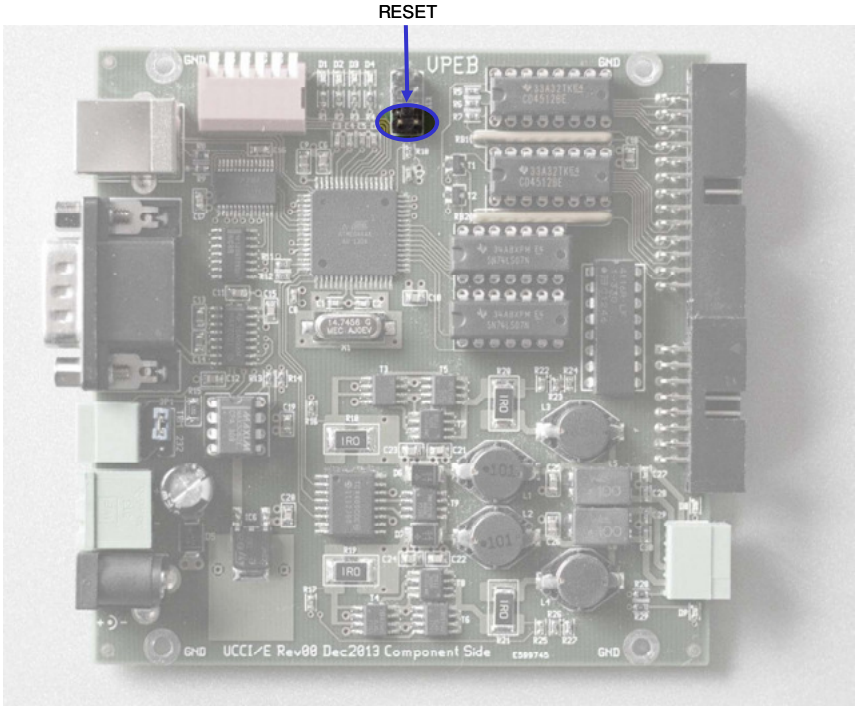

Fig 29: Reset

10. Click in AVRootloader on "Connect to device". At the top of the window it says "Connecting…, please press RESET on the Device".
    If you communicate through an RM-U, the green LED on the RM-U flashes, while the red one remains permanently on;
11. Now take the reset-jumper from the UCCI/E(-s) "in a fluent motion" and put it aside. The green, orange and yellow LED on the (target) UCCI/E(-s) should be lit.
    If you communicate through an RM-U, the green and one of the yellow LEDs on the RM-U flash, while the red one remains permanently on;
12. The top of the AVRootloader window now shows "connected". In the tab "Device Information" you'll find some information on the processor-type, the current software version and some other details, all not being very important;
13. Now (in the tab "Programming") click the button "Program". If you activated "Open protocol-window after processing" in step 6, after a few seconds the "Protocol" tab will open and show the results. Your new firmware is installed in your UCCI/E(-s).
14. Click the button "Disconnect device" in the "Programming" tab. UCCI/E(-s) now restarts with the new firmware.
15. Store the reset-jumper on the parking-position where you found it;
16. If you have a system with multipleUCCI/E-s modules and you wish to update other UCCI/E-s modules, repeat the above steps from step 9;
17. Close AVRootloader
    If you had to modify connections, restore the original configuration. If you communicate through an RM-U, reboot or reset your RM-U. Your system is ready for use again.

This page is intentionally left blank

This page is intentionally left blank