

DINAMO Model Railroad Control

UCCI

Manual

Author: Leon J.A. van Perlo
Version: 1.0
Date: August 2nd, 2008

Release management

This manual is applicable for modules and kits consisting of:

- Print
 - UCCI Rev00 Oct 2007
- Firmware
 - UCCI Rel β1 2007/09/13

©2007-2008 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 module for their own use is allowed.

This document is written using OpenOffice.org Writer and generated with CutePDF

Contents

| | |
|--|----|
| 1 Introduction..... | 4 |
| 2 Functional description..... | 5 |
| 2.1 Generic..... | 5 |
| 2.2 Transmission-loops..... | 6 |
| 2.3 Feedbacks..... | 9 |
| 3 Mounting and Connecting..... | 10 |
| 3.1 Mounting UCCI..... | 10 |
| 3.2 Power Supply..... | 10 |
| 3.3 PC communication..... | 10 |
| 3.4 Transmission Loops..... | 11 |
| 3.5 Feedbacks..... | 11 |
| 3.6 Turnouts, Traffic Lights, etc..... | 13 |
| 4 Operation and Configuration..... | 14 |
| 4.1 Operation..... | 14 |
| 4.2 Configuration..... | 14 |
| 4.2.1 Switch activation delay..... | 14 |
| 4.2.2 Switch release delay..... | 14 |

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. Cars are stopped by means of a magnetic field generated by a coil underneath the road surface, actuating a reed-contact that interrupts the current to the motor, forcing a sudden stop. Other control-possibilities are almost completely absent.

On behalf of Railz Miniworld in Rotterdam, a method has been developed to control these kind of cars in a sophisticated way. Cars are equipped with a decoder that receives commands through a wireless communication system. The signals and commands for the decoders can be generated by the Dinamo control system.

To meet the wishes of hobbyists to apply this system on smaller layouts the UCCI module has been developed. UCCI is an integrated RM/TM-CC module with the following functions:

- Communication with a PC via RS232 or USB
- Driving 2 communication loops
- Reading 128 position sensors
- Controlling OM32 serial modules

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

UCCI is delivered as an assembled module only.



Fig 1: UCCI (2x)

2 Functional description

2.1 Generic

UCCI is an integrated module for controlling speed and functions of model-cars, guided by a steering-wire. An example of this is the Faller™ Car System.

UCCI needs to be connected to a PC. The interface can be USB or a serial port (RS232). A suitable control-application on the PC sends commands to UCCI to control cars, 'turnouts', traffic lights, etc. and receives event messages from UCCI to determine the positions of the cars.

The (almost real-time) control of speed and function of cars occurs via an MCC (Model Car Control) decoder placed in the vehicle. The MCC decoder receives its information via a receiving-coil mounted in the bottom of the car.

The road-surface over which the car moves acts as a transmitter. As long as the receiver-coils is below 3 to 4 cm over the road-surface, the MCC decoder receives the information transmitted by the road. In order to transform the road surface into a transmitter, 2 copper-wires have to be applied. Details can be found in paragraph 2.2. Together the 2 copper-wires form a 'transmission-loop' of which 2 can be driven by UCCI.

Position feed-back of the cars can be done by means of reed-contacts in the road. The reed-contacts are connected to UCCI by means of a matrix-wiring. UCCI can read 128 reed-contacts

'Turnouts' and traffic lights can be controlled using the OM32 controller. These controllers can be connected to UCCI so that the PC can control these devices through the UCCI controller

Schematically the setup looks as follows:

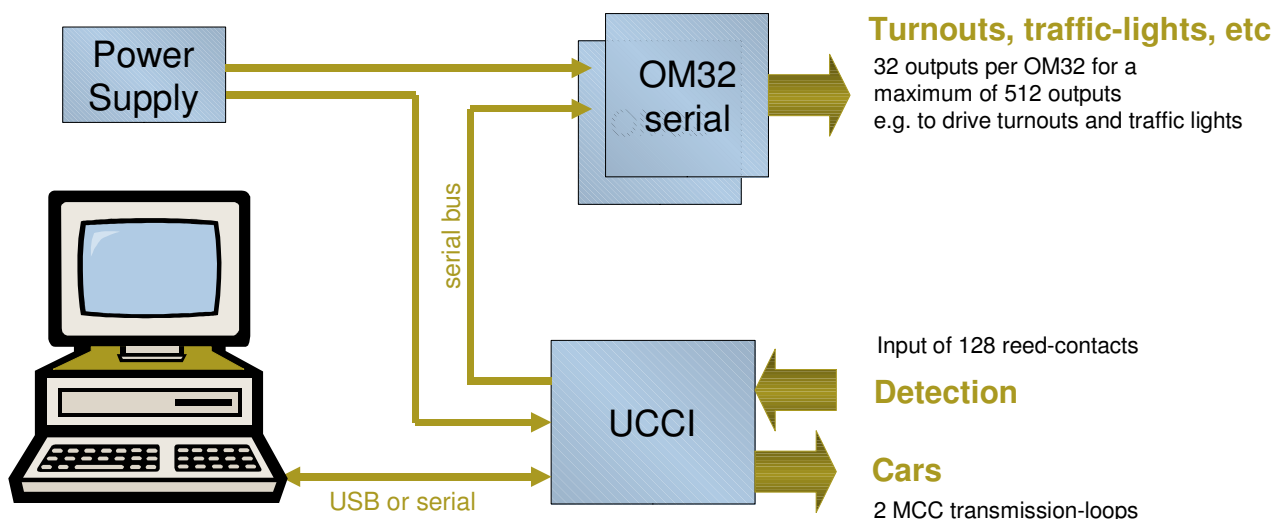


Fig 2: Schematic setup of the system

2.2 Transmission-loops

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. 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.

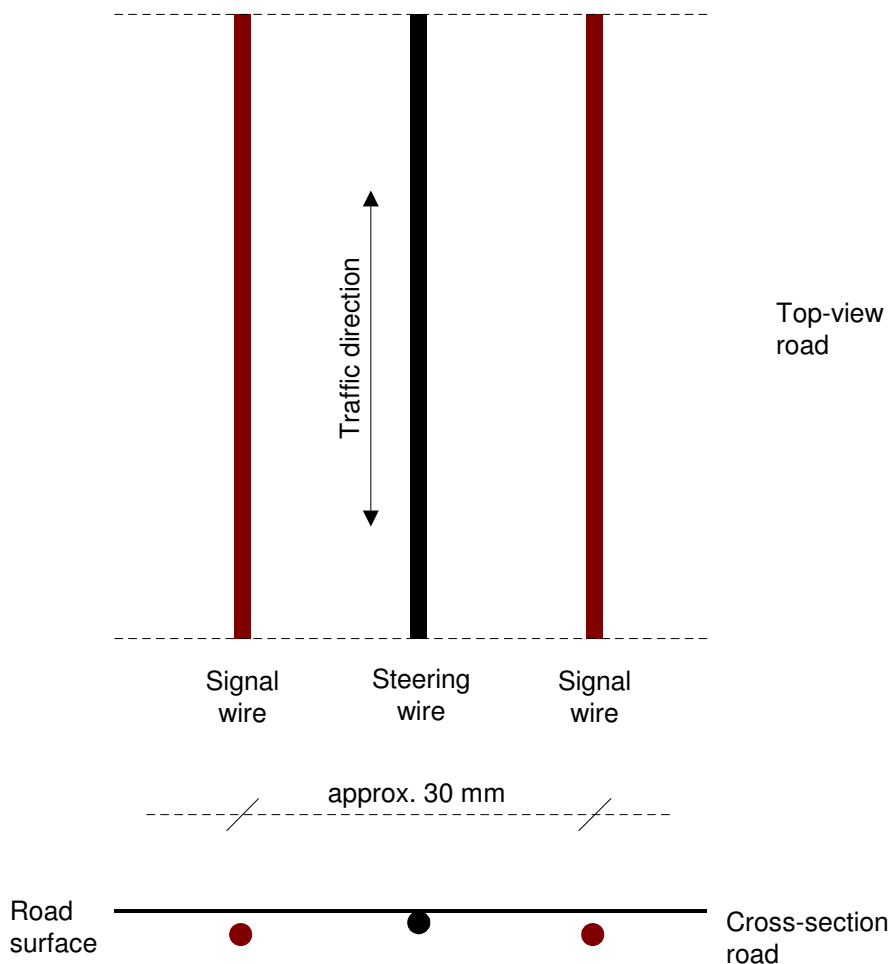


Fig 3: Building the communication system

The copper wires will conduct electrical current. All wires in the road are electrically wired in 'series'. It is essential that the direction of the current in each pair of wires is opposite, so it is of importance to think ahead how you're going to wire the layout.

A simple 'oval' is easy. You choose an arbitrary point where the wires are fed below the layout. The 2 wires 'to the left' 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.

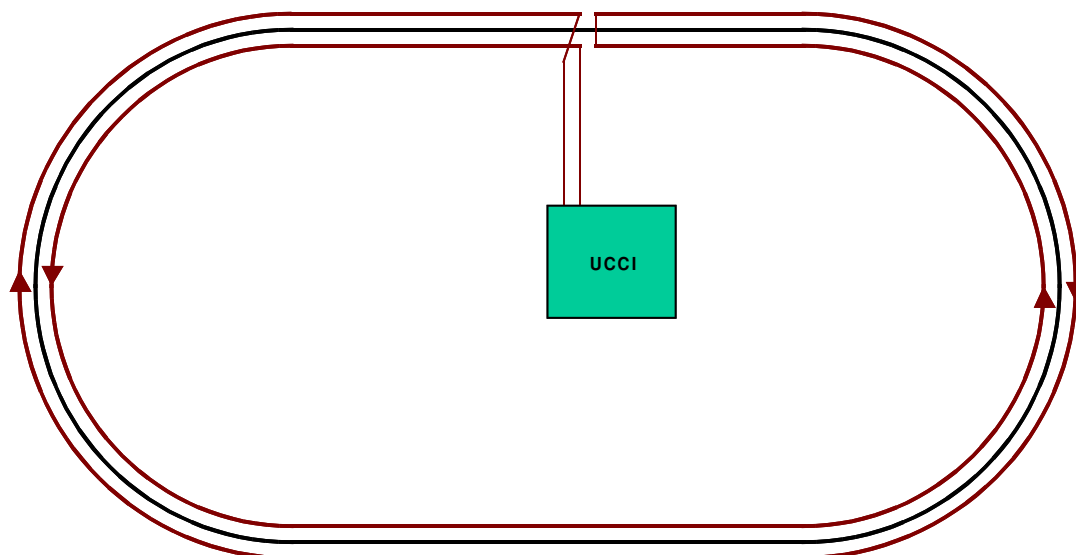


Fig 4: connecting an 'oval'

In case of a multi-lane road it's best to make sure that the current in the wires of different lanes that are closest together is of equal direction. It's irrelevant whether the direction the cars are going is equal or opposite.



Fig 5: Current direction in a multi-lane road

In case of a double oval you could make 2 independent transmission-loops. You can also simply put the 2 loops in series as in fig 6. While the loop does not become too long this 'serial arrangement' is preferred to limit the necessary power.

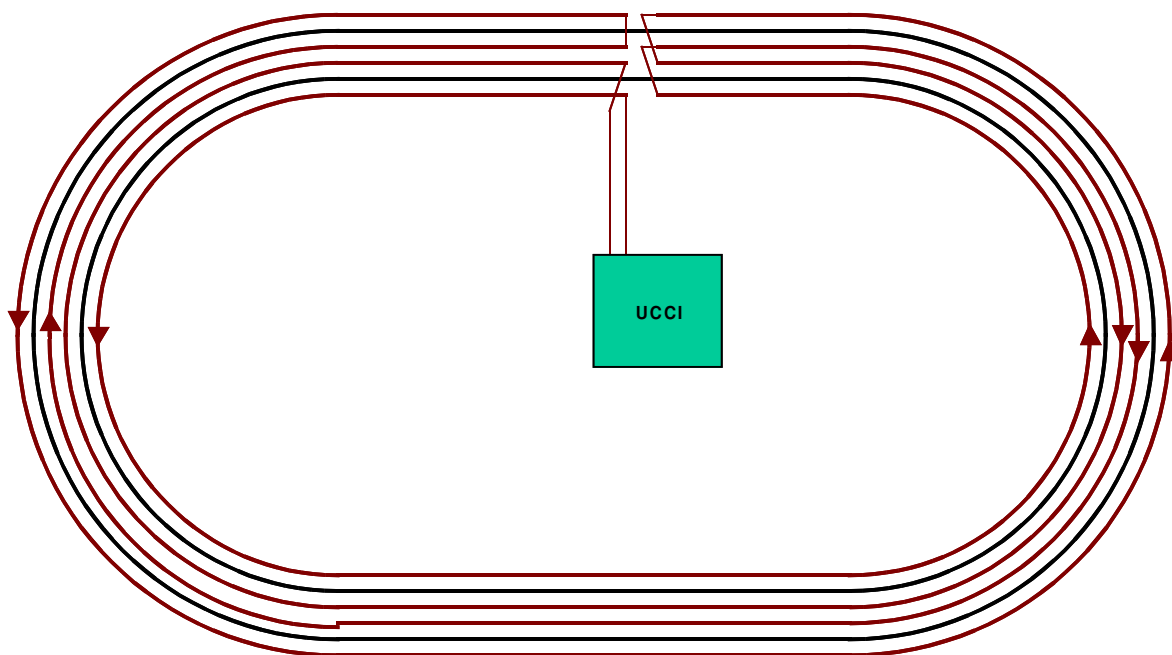


Fig 6: Connecting a double oval

If you have 'turnouts' in your layout the puzzle gets somewhat more complicated, but hopefully the example in fig 7 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.

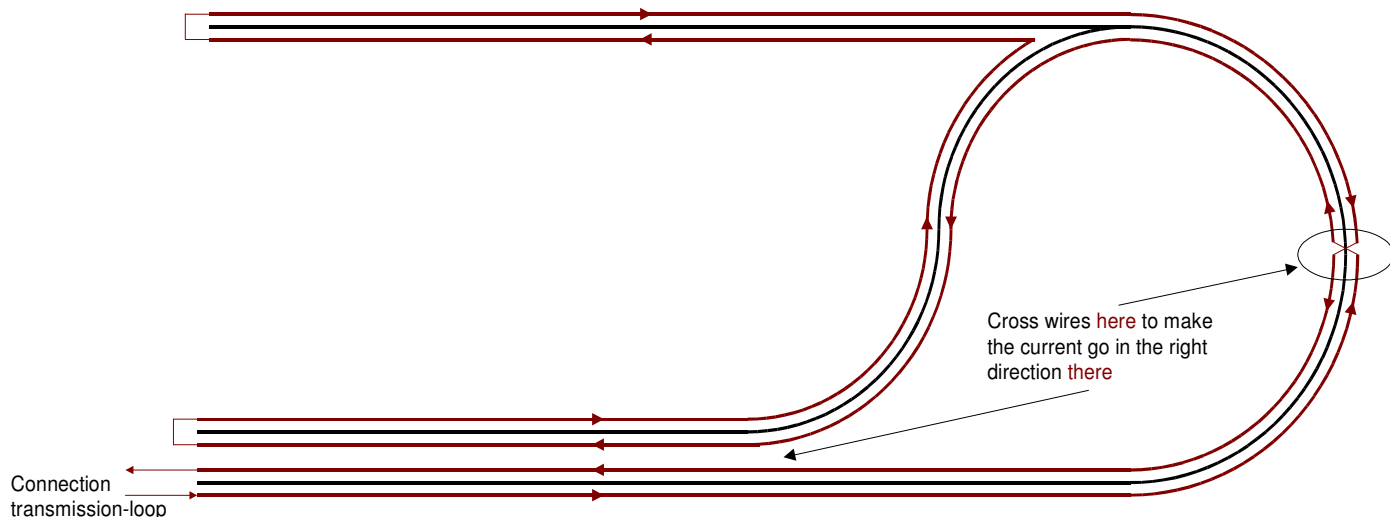


Fig 7: aansluiting bij een complexere situatie

The cross-section (diameter) of the communication wire should be around 0,22mm² (or AWG24 in US sizes). Wire from UTP LAN cable usually is 24 AWG and well suited for this purpose. Thinner or thicker wire may be used as well. You only must take care of the total resistance of the transmission loop.

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

Tabel 1: Resistance of wire (copper)

The transmission-loop resistance has a relation to the transmission-power and the power supply Voltage used. This relation can be found in table 2.

| Supply(V) | Rmin(Ω) | Rmax(Ω) |
|-----------|---------|---------|
| 7,5 | 0 | 3,6 |
| 8,0 | 0,9 | 5,1 |
| 9,0 | 3,2 | 7,9 |

Tabel 2: Relation between power supply and transmission-loop resistance

The above info should be used as indication only: Things don't fail immediately if it's not 100% right, but keep it as a guideline.

Example: Suppose you use an 8V power supply and 0,14mm² wire. The total road-length is 10 meters, so approximately 20 meters of wire. The total resistance thereof is 20 * 0,13Ω = 2,6Ω. This fits very well between the Rmin (0,9) en Rmax (5,1) values.

Suppose you want to use a 9V power supply in this case. The transmission-loop resistance actually is too low. You could insert a series-resistor of e.g. 2,2Ω in the loop. Keep in mind this resistor gets hot: You'll 'burn' around 0,3W per Ohm, so for a value of 2,2Ω you should use a type of at least 1 Watt (0,3W/Ω * 2,2Ω = 0,66W)

When your loop becomes too long for the supply voltage used, use the second transmission loop or raise your supply voltage. Practice has shown that loop-lengths up to 50 meters are no problem when the above conditions are met.

2.3 Feedbacks

The position of cars on the layout can be tracked by means of reed-contacts. A suitable size is approx. 14 x 2,1mm (e.g. Conrad 503770, January 2008). If you use Faller cars, these have a magnet in the steering mechanism that follows the steel wire in the road.

The 'steering magnet' can be used to activate the reed contacts as well. If you want to use this method, it requires a precise mounting of your reed-contacts. 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. 8).

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 cantmount the reed-contacts differently, but in that case you'll need an extra magnet under the car to activate them.

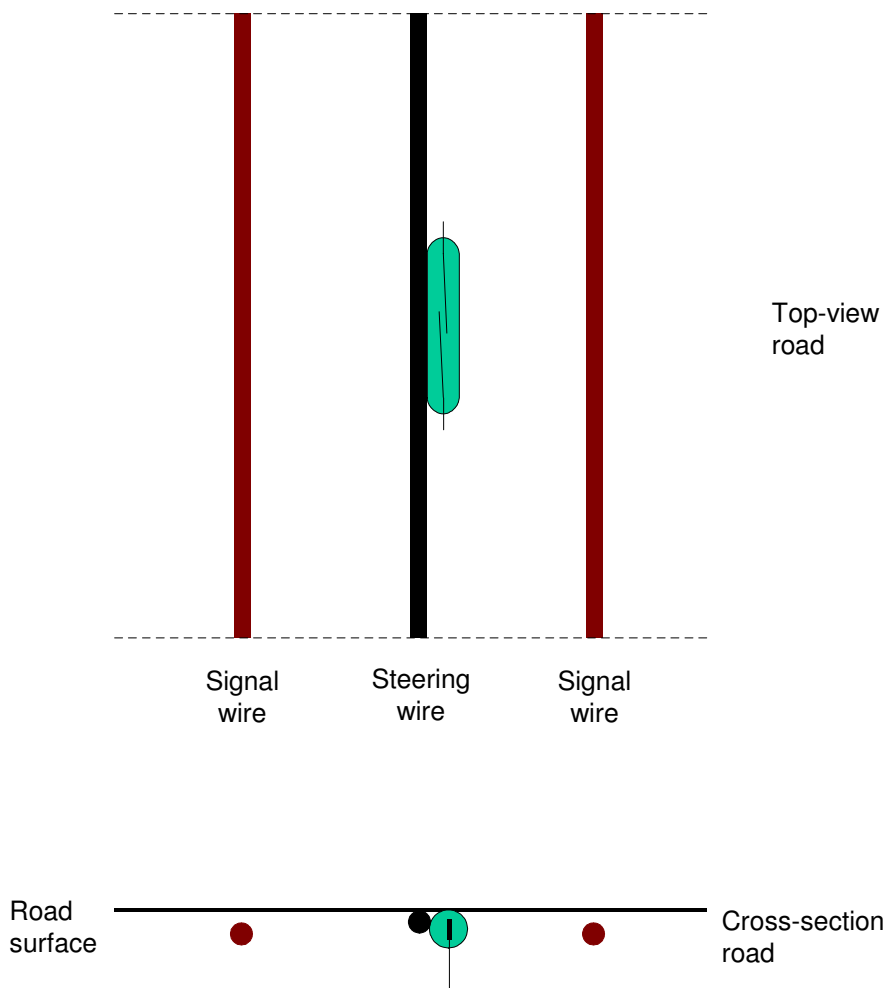


Fig 8: Placing of reed-contacts

3 Mounting and Connecting

3.1 Mounting UCCI

On UCCI you'll find an IC with a heat sink on top of it, next to 4 larger resistors. These components drive the transmission loops. The IC and the 4 resistors will get quite hot. This is normal. However you need to make sure there is some airflow possible to make the heat disappear.



WARNING: The print at solder side around the chip with the heat sink functions as heat sink as well. Therefore don't mount the module directly on a surface, but use spacers of at least 10mm. If you use both transmission-loops it is advisable to mount the module in a vertical position with the heat sink and resistors in the upper position. This maximizes natural air-flow.

3.2 Power Supply

UCCI needs a DC power supply of 7,5..9V and a current capacity of 1,2A or more. It does not have to be well stabilized, but it needs to be DC.

You'll find 2 connectors on UCCI: A 2-pole (green) connector with screw-terminal and a round plug. You may use one of them, depending what is mounted on your power supply unit. With the round plug, the centre pin is +.

Make sure the polarity of your power supply is correct!



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

3.3 PC communication

PC Communication can be established by means of a serial RS232 connection or a USB interface. When using the serial 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. 9 below. The gray connections are not used by UCCI, so 3 wires will do.

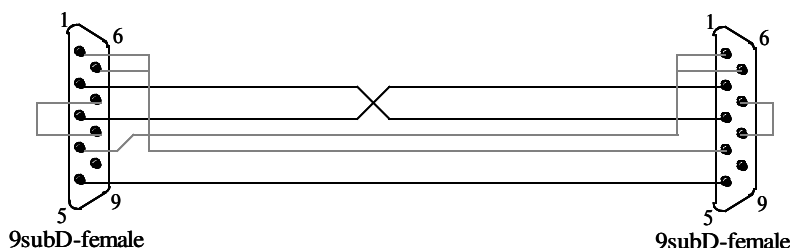


Fig 9: Nulmodemkabel

The second possibility is using the USB interface. You can use a standard USB interface cable.

ATTENTION: UCCI is not powered from the USB port, so also when using USB you'll need to apply a separate power supply.

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. Make sure UCCI has power supply and **next** plug in the USB cable. The interface now shall install automatically. When the driver is active and has established connection with UCCI, the blue LED on the module will light. (During installation of the driver this LED will flash a few times.)

In Control Panel -> System -> Hardware -> Device Manager -> Ports (the exact path will depend on your OS version and- language) you'll find an extra COM-port. Open "properties" of this com port, go to "Port Settings" and click "Advanced".

It is NOT wise to change the COM port number here, usually you'll experience failures!

Change the following values:

- Receive (Bytes): 64
- Transmit (Bytes): 64
- Latency Timer (msec): 2

Click OK and close the opened windows. Disconnect UCCI from the PC and reconnect to load the new settings.

Your interface is ready for use.

3.4 Transmission Loops

If you installed the transmission loops as described in paragraph 2.2 you can now connect them to UCCI. Use the green 4-pole connector next to the 4 large resistors. Pin 1 and 2 form one loop and pin 3 en 4 form the other loop. From which side you count is irrelevant. You can insert the stripped wire in the back of the supplied connector after depressing the orange levers. The connector accepts both solid and stranded wire.

3.5 Feedbacks

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.

Every switch shall be connected between a unique combination of Switch Sense and Switch Drive pins. This way, every switch will get a unique address.

Every switch needs to be equipped with a diode (1N4148) in series with the switch. These diodes are necessary to avoid ghost detection in case multiple switches are activated simultaneously. Otherwise simultaneous activation of e.g. switch 0, 1 en 8 would cause a short circuit between sense-1 and drive-1 and incorrectly report switch 9 active.

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!

How the switch-matrix is built can be found in fig. 10 below.

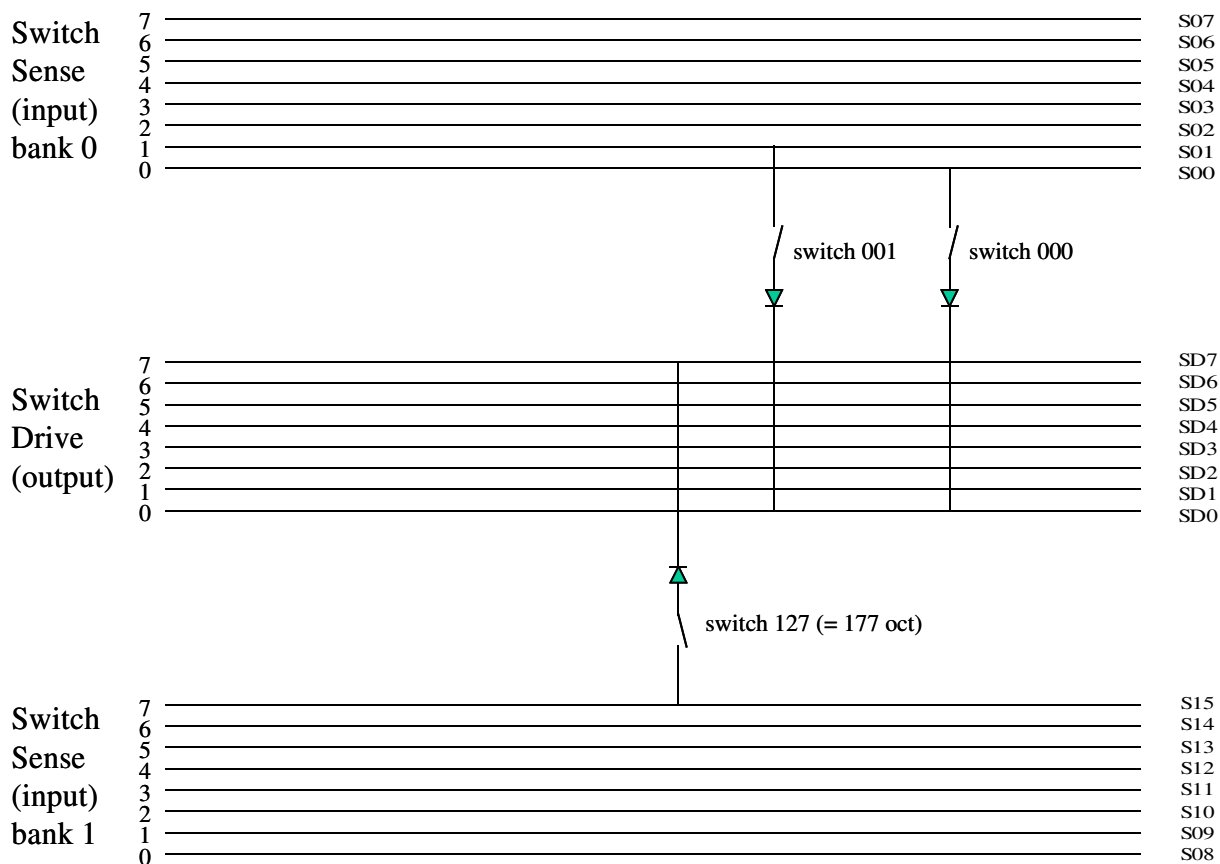


Fig 10: Switch Matrix

To connect the switches 2 flatcable connectors are available. You'll find a 26 pin and a 16 pin connector on UCCI. The pin-out is as follows:

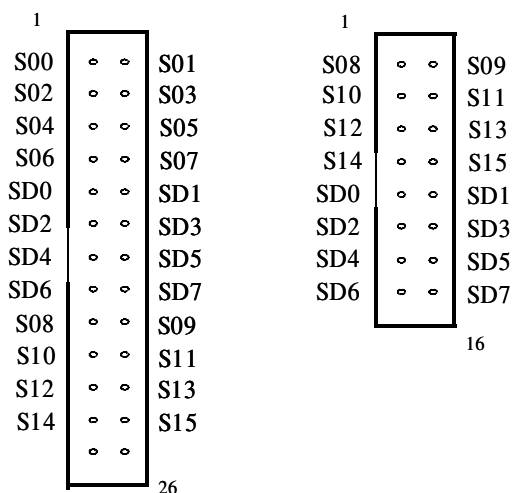


Fig 11: Connectors for switches

De 26 pin connector contains all necessary signals. The 16 pin is there in case you want to connect the switches in bank 1 by means of a separate flatcable.

If you don't want to make the wiring for the switch matrix yourself, there is a special mini-PCB (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.

3.6 Turnouts, Traffic Lights, etc.

'Turnouts' and traffic lights can be controlled by means of an OM32 serial module. Details on the OM32 can be found in the OM32 manual.

UCCI can control the OM32 via RS232 or RS485. The RS485 chip is not mounted on UCCI by default and the OM32 serial is by default delivered in RS232 configuration, so in this manual we'll limit ourselves to RS232 modus.

The power supplies of the OM32 and UCCI modules shall be connected to the same GND (0V) potential. Maybe you want to power the OM32 from a different power supply. This is no problem as long as the 0V potential of both power supplies are interconnected.

The signal for the OM32 comes from the white 3-pin connector on UCCI. You only need to connect pin 2 of this connector to pin 4 of K1 on the OM32 module(s). De OM32 shall be configured as "OM32 serial for use with Koploper", so RS232 modus and a "fixed" bitrate of 19.200bps.

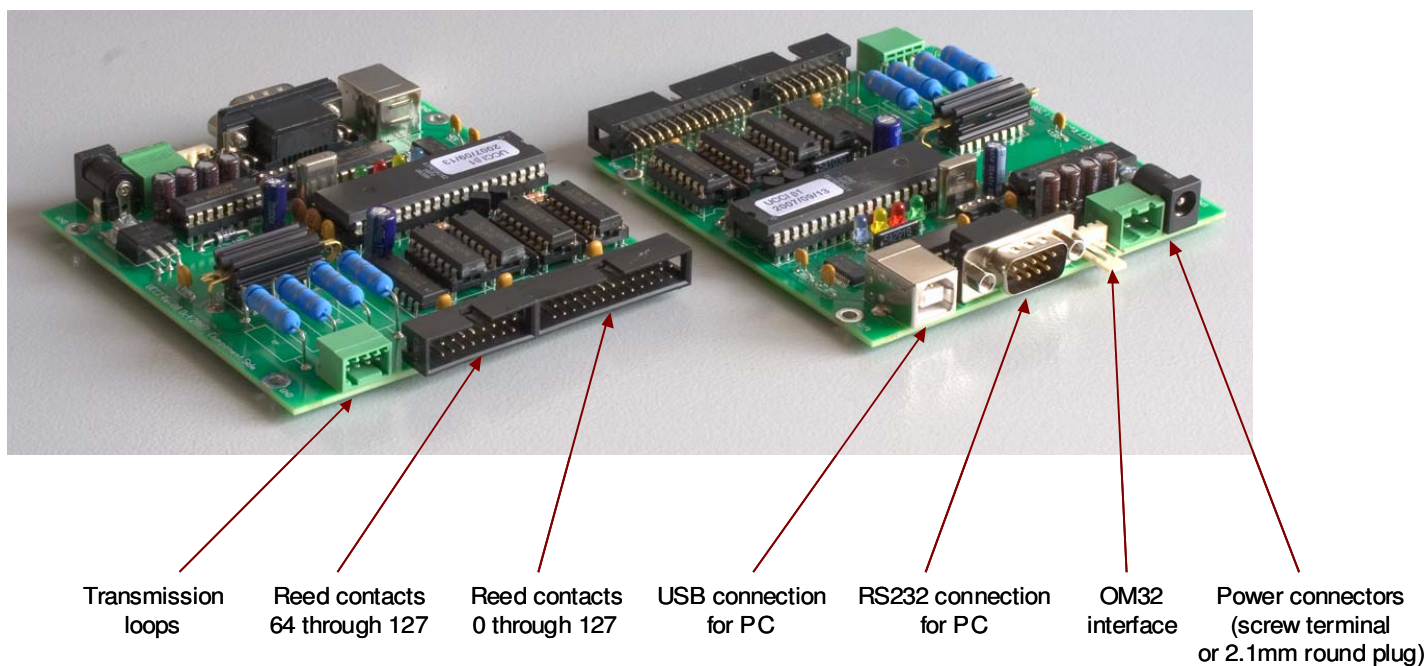


Fig 12: Overview of connections

4 Operation and Configuration

4.1 Operation

There's little to control on UCCI. Operation 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
- Red: UCCI sends a datagram to the PC, red LED on = communication with PC active
- Green: Command-processing: UCCI received a command from the PC and is processing this.
- Yellow: UCCI sends a message to an OM32 module

4.2 Configuration

UCCI 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 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 happens in a similar way as configuration of Dinamo 3.x modules that support this feature. For configuration the DinamoConfig configuration-tool is available.

The following configurations can be made:

4.2.1 Switch activation delay

Before UCCI reports a closed switch (reed-contact) as active, UCCI 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 can be set per bank of 64 switches between 0 and 7 (so N=1..8). Default Switch Activation Delay = 1.

UCCI behaves software-wise as a TM-H/TM-CC controller in a Dinamo system. When using DinamoConfig you need to act as if you are configuring TM-H module 0.

4.2.2 Switch release delay

As with switch closure, before UCCI reports a switch inactive, UCCI 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 can be set per bank of 64 switches between 0 and 7 (so N=1..8). Default Switch Release Delay = 7.

UCCI behaves software-wise as a TM-H/TM-CC controller in a Dinamo system. When using DinamoConfig you need to act as if you are configuring TM-H module 0.

This page is intentionally left blank