VPEB Van Perlo Elektronica & Besturingstechniek

UCCI/E

R10/R11

Manual

Smart electronics for your miniature world

Author: Leon J.A. van Perlo
Version: 2.1
Date: June 8, 2021

Release Management

This manual is applicable for:

Module:

- UCCI/E Rev11
- UCCI/E-u Rev10
- UCCI/E-s Rev10

Firmware:

- UCCI/E 2.01

Software:

- DinamoConfig 1.40C

Preface

This manual describes the use of the UCCI/E as a stand-alone unit or as part of an RM-U or RM-C based Dinamo system. This manual is not intended to explain the operation of an RM-U, RM-C or the construction of a Dinamo system. For that information, we refer to the relevant manuals.

UCCI(/E) is a module intended for the control of digital cars according to the Dinamo/MCC protocol. The original version (without /E in the name) has 4 large resistors to regulate the current through the transmit loops. Those resistors get quite hot and UCCI was never supplied in a housing for that reason. The later versions (with /E) are recognizable by 4 coils arranged in a kind of trapezoidal shape. This version does not produce extreme heat and is usually supplied in a housing.

UCCI/E has been delivered in various versions. This manual only describes the UCCI/E Rev10 and Rev11. These versions are recognizable by the presence of two RJ45 sockets for RS485 network communication. If you have an UCCI (without /E), consult the UCCI manual V1.0. If you have a UCCI/E Rev00, Rev01 or Rev02, please consult the UCCI/E manual V1.0. (available in English)

UCCI/E can be found in 3 different types:

- **UCCI/E.** This is the full version, which can be connected to a PC as a stand-alone unit, or can be integrated into an RM C or RM-U based Dinamo network.
- **UCCI/E-s.** This is the version without USB interface, which can **only** be used in a Dinamo network, controlled by an RM-U or RM-C.
- **UCCI/E-u.** This is the full version, which is primarily intended to be connected to a PC as a stand-alone unit. Like UCCI/E and UCCI/E-s, this module can also be included in a Dinamo network, but lacks the DIP switches for the required address setting. The address of the module must then be set through software configuration.

UCCI/E Rev10 is produced in a –u and –s version. UCCI/E Rev11 is only supplied in a complete version (so without addition).

Contents

1 Dinamo/MCC

1.1 Introduction

On a "model railway" nowadays one finds more often an extension with moving model-cars. For that reason, among other things, VPEB prefers not to speak of a "model railway", but of a "miniature world". For some hobbyists, this even means that there are no longer any trains in the miniature world, but only cars and possibly trams.

The moving cars are often from a German manufacturer that was originally best known for making plastic buildings and related accessories[1]. The possibilities to drive cars of this system in original condition are very limited.

VPEB developed the Dinamo/MCC system to control these types of cars[2] in an advanced way. Cars are then equipped with a decoder that receives commands via a wireless communication system embedded in the road surface. The road surface also contains sensors, which transmit the positions of the cars to the control system. In addition, there can be turnouts, crossings and traffic lights to make the layout realistic and complete.

The Dinamo/MCC system is essentially made up of three different modules:

- UCCI/E(-s): This module generates the signals to control the decoders in the cars and reads the sensors in the road surface to measure and report the position of vehicles.
- MCCdec: The decoder(s) in the vehicles, which receive commands from UCCI/E(-s) and which control speed and functions in the car.
- OC32(/NG): The module that controls, among other things, traffic lights, road-exits ("turnouts") and other scenery.

In addition, we have two modules to facilitate the connection of position sensors:

- SWdec: For connecting 8 reed contacts per module
- SHdec: For connecting 8 Hall sensors per module

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:

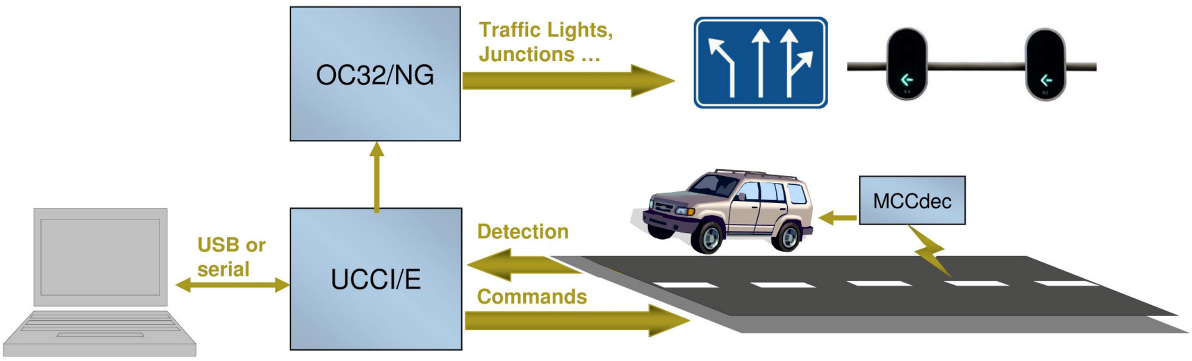


Figure 1: Schematic setup of a small/medium Dinamo/MCC system

[1] We prefer not to mention the name of this manufacturer here because a VPEB partner has been sued by this party in the past for showing the name and logo in advertising. We would like to prevent a repetition of such matters, both for VPEB and its partners.

[2] Use of the Dinamo/MCC system is certainly not limited to the vehicles of the above-mentioned manufacturer. In principle, any vehicle in scale Z to 00 that drives independently and follows a guide wire in the road can be controlled with Dinamo/MCC.

The UCCI/E is connected to a PC with adequate control software through a USB connection. UCCI/E generates the signals for the decoders and reads the feedbacks. UCCI/E (optionally) controls one or more OC32(/NG) 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-C (or RM-U) central control unit, connected to a PC with adequate control. An RM-C/1+ 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. An RM-C/2 can even control up to 32 UCCI/E controllers and up to 32 OC32 controllers, good for some 1.250 meters.

Schematically this large-scale setup looks as follows:

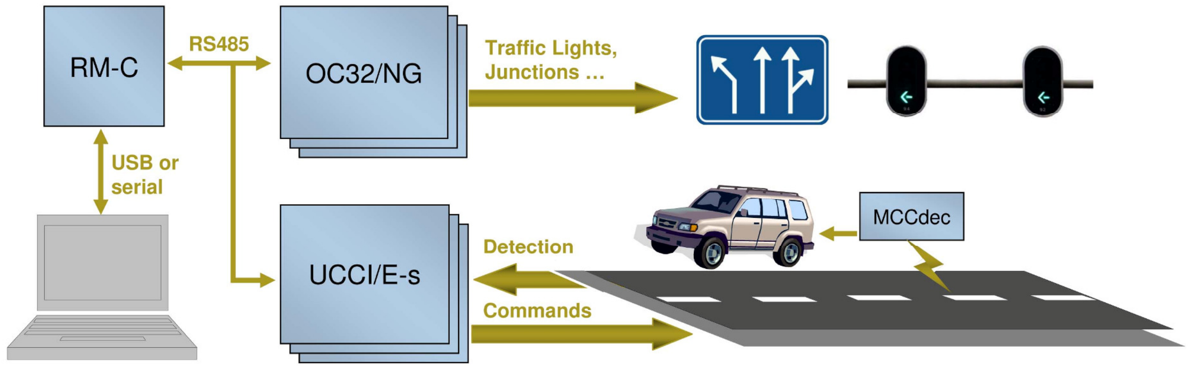


Figure 2: Schematic setup of a large Dinamo/MCC system

Note, in the above diagram, UCCI/E is replaced by an UCCI/E-s. The '-s' stands for 'slave unit', meaning this unit requires an RM-C or RM-U central controller to connect to the controlling PC. However, an UCCI/E can be used as an UCCI/E-s, so if you start with the setup per figure 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-C controller and as many UCCI/E-s and OC32 controllers as required. The same applies if you start with a basic set-up and at some point realize that it would be nice to have a tram ride along. Then you can still upgrade to a Dinamo system and include your existing UCCI/E in it.

The UCCI/E-s is cheaper than the UCCI/E since it lacks the USB interface.

Note that in case an UCCI/E is directly connected to the PC (figure 1), the OC32 modules are controlled via UCCI/E. However, when an RM-C or RM-U is used (figure 2), both UCCI/E and OC32 modules (as well as any other Dinamo modules, such as TM44) are included in one RS485 network controlled by the RM-C or RM-U.

2 UCCI/E(-s)

2.1 Overview

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

- Communication with a PC via USB **or**
  Communication with an RM-C or RM-U central controller via RS485
- Driving 2 transmission loops for cars
- Reading128 position sensors
- When connected to PC: Controlling up to 16 OC32 modules via RS485

There have been a number of different versions over time. As already written in the preface, there is even a UCCI (without /E). In this manual we only describe the versions Rev10 and Rev11, recognizable by the RJ45 connections.

UCCI/E (firmware 2.01) is fully compatible with the Dinamo 3.0 protocol, but only supports those commands that are 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 interface and is therefore cheaper than the normal UCCI/E. As indicated in section 1.3, the UCCI/E-s is typically used to build a large system based on the RM-U or RM-C as the central module.

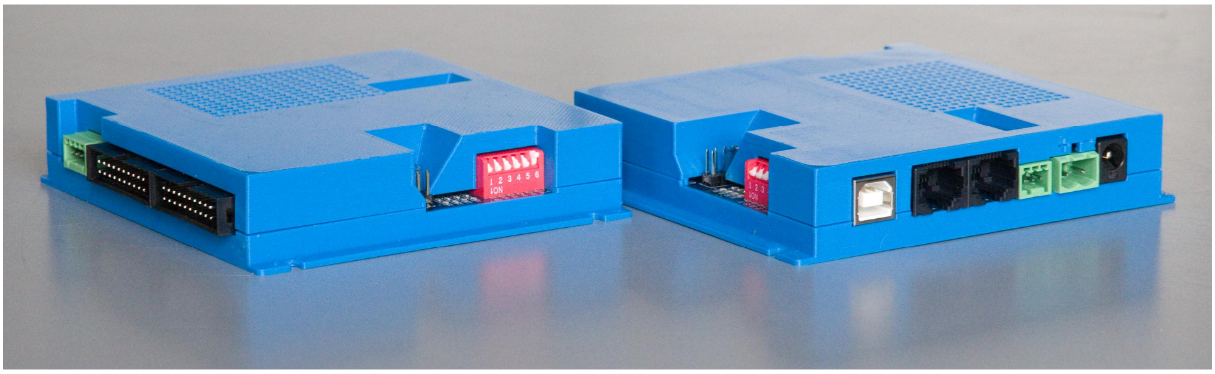


Figure 3: UCCI/E-Rev11 in housing (2x)

2.2 Mounting

The UCCI/E(-s) is meant to be mounted in the direct vicinity of or below your miniature world. If you are going to use the USB interface for connection to the PC, it makes sense to keep the distance between your UCCI/E and your PC as short as possible, since USB is rather susceptible to electronic interference. Unlike older models, UCCI/E Rev10 and Rev11 do have an electrically isolated USB interface, which has significantly reduced this "problem".

Mounting UCCI/E in housing is almost child's play. Screw the housing by the flanges with the 4 supplied (PZ1) screws on a flat surface. Make sure that you leave enough space to insert the plugs and make sure that you can still reach the DIP switches on the side (if applicable).

If there is little space to mount the modules, it is possible to “stack” Dinamo P&P modules. An UCCI/E without housing has 4 mounting holes (3mm). These are in the same place at UCCI/E, TM44 and OC32/NG. A mounting frame is available to make mounting and optionally stacking modules without housing easy. This frame is first screwed onto a flat surface and then you can mount UCCI/E on it with (supplied) M3 bolts. By using M3x20 spacers you can mount one or more other modules on a UCCI/E, such as other UCCI/E modules, OC32/NGs or TM44s. When stacking modules on a TM44, longer spacers are required. See the Dinamo P&P manual 1.3

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 6.5.4)*
R_N *is the resistance (Ohm) of loop N (see paragraph 4.2)*

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).

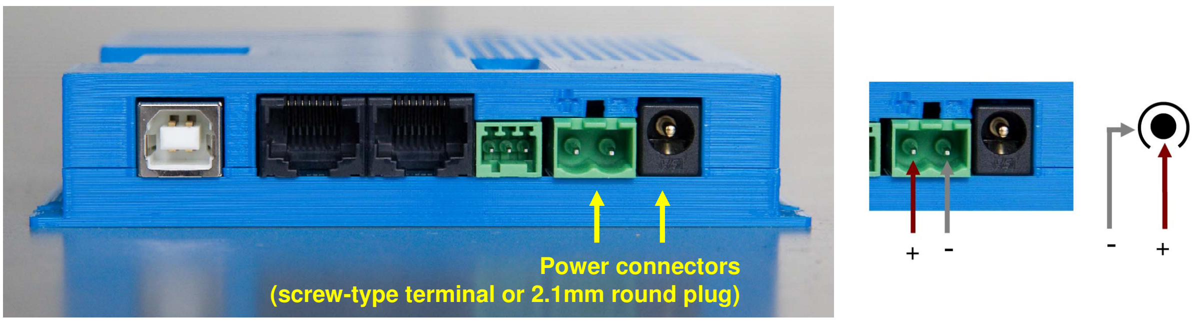


Figure 4: Power supply sockets on the UCCI/E(-s)

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 figure 4.

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 USB interface.

Figure 5: PC interfaces on UCCI/E

To use USB, your PC must of course be equipped with a USB connection. Your PC must also be equipped with a driver to communicate with UCCI/E. From Windows Vista, UCCI/E is normally automatically recognized and the correct drivers are loaded, **provided** your PC is connected to the Internet.

If your PC is not connected to the Internet or if you are using an older version of the Windows operating system, you must install the correct drivers **before** connecting the UCCI/E USB interface. You can download it for free from the Future Technology Devices website: www.ftdichip.com/Drivers/VCP.htm

Drivers can also be found at https://www.dinamousers.net, although they may not be the latest versions. It works easiest if you use the "setup executable for default VID and PID values.

The FTDI drivers are suitable from Windows98. Correct operation under Windows95 is not guaranteed. In addition to Windows, drivers are available for Linux and Mac OS X.

After you have provided your PC with the correct driver (if necessary) or have made sure that your PC has a working internet connection, you can connect UCCI/E with a standard USB A-B cable. Preferably use one that is suitable for USB2.0, as these are usually better shielded. If all is well, your UCCI/E will be recognized automatically and a "virtual com-port" will be created for UCCI/E. When the USB port of the PC is connected to UCCI/E, the blue LED on UCCI/E will turn on. During installation, this LED will blink a few times.

The USB interface is electrically isolated with UCCI/E Rev10 and Rev11. This means that the USB electronics are powered from the USB connection of the PC and UCCI/E is powered from the power supply supplied in paragraph 2.3. The above procedure, setting up the USB connection to the PC, will therefore work, even if UCCI/E is not supplied with power. You cannot communicate meaningfully in that situation, because the processor of UCCI/E is then off, but the blue LED will still light up.

3 RS485

3.1 Het netwerk

The RS485 network connection on UCCI/E can be used in two ways:

- To control OC32(/NG) modules from PC via UCCI/E when UCCI/E is connected to the PC via USB.
  OR
- To insert UCCI/E as a “slave unit” in a Dinamo network.

For the record: RS485 is **not Ethernet** (the kind of network you connect computers to, among other things), but we do use the same connectors and cables.

The easiest way to create an RS485 network is to use standard RJ45-UTP network cables (unshielded twisted pair). You can buy these cables in any store and webshop where computers and/or network components are sold. The VPEB partners also offer them. The "quality" of the cable is unimportant, certainly for the usual distances on a model railway layout. So Cat3, Cat5, Cat5e, Cat6 or no Cat at all, in principle it all works, as long as the RJ45 connectors are properly mounted and at least the inner 6 pins connected 1 to 1.

The total length of the RS485 network may (theoretically) be 1200 meters. Since you do need some ambition to reach this length on your layout, the length of the cables you use between the modules is not crucial, but do not buy or make them much longer than reasonably necessary, if only to keep the installation tidy.

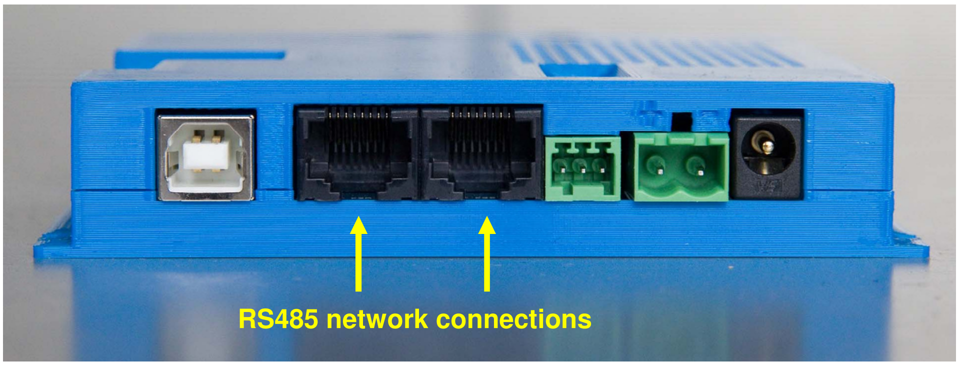


Figure 6: RS485 network connections on UCCI/E

3.2 Connecting OC32/NG modules to UCCI/E

In addition to controlling and detecting vehicles, which UCCI/E does itself and about which more later, you will probably also come across intersections (switches) on a serious street map where you want to give cars the choice to choose a certain route and you may want those intersections also equipped with traffic lights. Controlling switches, traffic lights and an almost endless list of other accessories can be done with the OC32/NG. The OC32/NG has simply too much capabilities to describe here and that is why we refer you to the OC32 manuals that have been written for this.

We assume that you are using an OC32/NG. It is also possible with the older (non -/NG) version, about which more in paragraph 3.3.

If you want to connect one or more OC32/NG's to UCCI/E, plug one end of an RJ45 cable into one of the RS485 network connections of UCCI/E (figure 6) and plug the other end into one of the RS485 network connections of the OC32/NG (see OC32/NG manual). If you have more than one OC32/NG, plug another cable into the second RS485 connector of the first OC32/NG you connected and plug the other end into the next OC32/NG. You can repeat this game until you have connected 16 OC32/NGs. The result, with 3 OC32/NGs as an example, then looks like in figure 7.

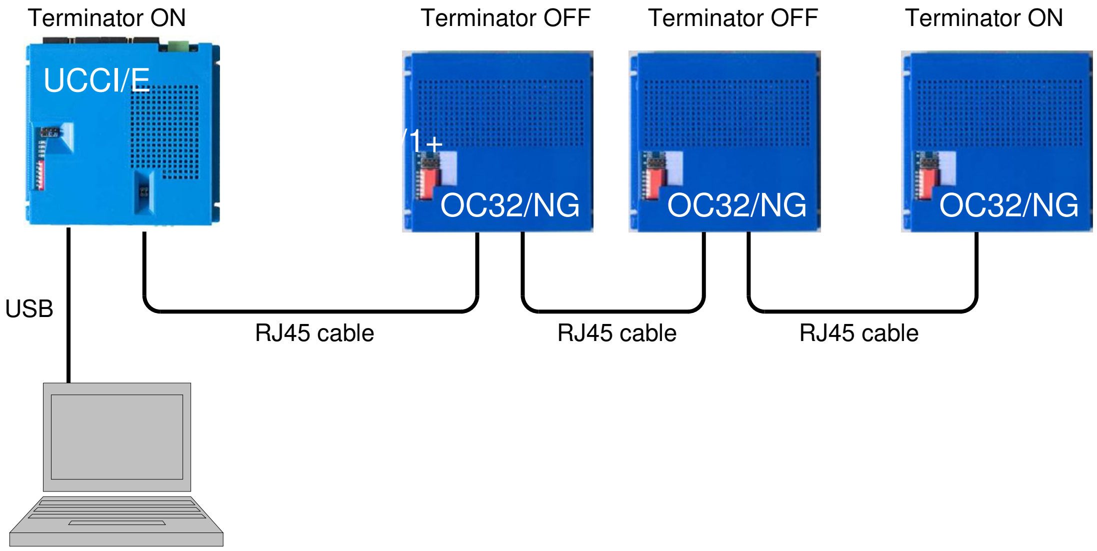


Figure 7: Connect OC32's to UCCI/E

The principle of an RS485 network is that it is one long, continuous cable, to which "stations" are connected at various places that can transmit and receive. We call such a continuous cable a "bus". It is essential that this bus is "terminated" with a terminating resistor at both the beginning and the end, and therefore only at the beginning and end. Such a resistor is officially called a **Terminator**. You can visually imagine it as a plug in both ends of the bus to prevent the electrons from falling out.

The first module in your RS485 bus is UCCI/E. In the standard configuration at UCCI/E the terminator is already activated. The last module in your bus is the OC32/NG module with only one occupied RJ45 connection. This last module must therefore also have a terminator.

The terminator is also activated on the OC32/NG at the factory. So if you connect one OC32/NG to a UCCI/E, it's a matter of connecting the two modules with an RJ45 cable and you're done. If you have multiple OC32/NG modules, you will have to disable the terminators of the intermediate modules. How to do that can be found in the OC32/NG manual.

Multiple stations can be connected to an RS485 network and they all communicate via the same cable. To ensure that it is clear who is talking to whom, each station must have a **unique address**. But even if you only connect one OC32/NG to a UCCI/E, this OC32 must have an address. The factory address of the OC32/NG is set to 0. If you only connect one OC32/NG to a UCCI/E, that's fine too and you don't have to worry about it. If you connect multiple OC32/NG modules, you will have to set the addresses of the other OC32s. How to do that is in the OC32/NG manual.

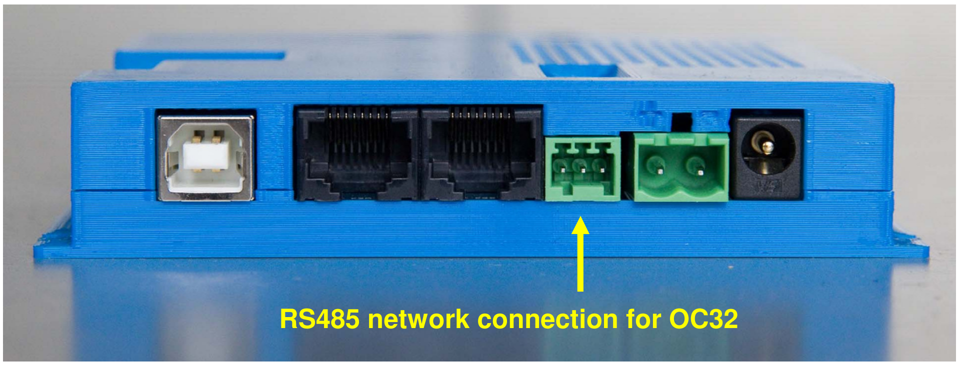
For the record: Addresses of modules have no relation whatsoever with the order in which modules are connected. The order can be completely arbitrary. However, it is essential that the terminator is **only** activated on the first (UCCI/E) and last OC32/NG module.

3.3 Connecting OC32 (not-/NG) modules to UCCI/E

If you don't have the OC32/NG but the older OC32 (non-/NG), you can also use it. The difference (at this point) is that the OC32 does not have RJ45 sockets. So you have to make your own cable.

In addition to the two RJ45 sockets on UCCI/E, there is also a 3-pin connection. The same type of connection is on the OC32.

Make a cable between the UCCI/E and OC32. The supplied connectors are provided with spring clamps. Strip the conductor about 10 mm, press the orange catch, insert the wire, release the catch. Usually it is easiest if you put the plug in the socket of the module, then you have something to press against.



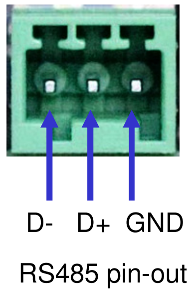


Figure 8: RS485 connection for OC32 (old model)

Use a twisted pair cable if possible. Use one wire pair for D-/D+ and any other wire for GND. The connections on UCCI/E and OC32 are identical, so pin 1 goes to pin 1, pin 2 to pin 2 and pin 3 to pin 3. Make sure that the terminating resistor on the OC32 is activated and that you set the address.

If you have multiple OC32 (non -/NG) modules, you can easily daisy-chain them. Note that you go from UCCI/E to the first OC32 and from this first OC32 on to the second, from the second on to the third, etc. So loop through and don't split off. Make sure that only the last OC32 has the terminating resistor activated and that all OC32s have a unique address.

3.4 Connecting UCCI/E-s modules to an RM-C or RM-U controller

If you are building a “large system” in accordance with figure 2, or if you want to include a UCCI/E in a Dinamo system that you also use to control trains, then UCCI/E is inserted in the RS485 bus that is controlled by the central RM-C or RM-U controller. In that case, also consult the Dinamo Plug & Play Manual 1.3, as it explains how to build the RS485 network from an RM-C. If you use an RM-U P&P with RJ45 converter it works exactly the same way[3].

The aforementioned Dinamo Plug & Play Manual 1.3 explains how to build a Dinamo RS485 network based on RJ45 cables. UCCI/E is included in that network in exactly the same way as a TM44 or OC32/NG module.

If UCCI/E is the last module in the network, the network terminator on UCCI/E must be enabled, if UCCI/E is not the last module, the terminator must be disabled. UCCI/E terminators are easily accessible, even when the housing is mounted. In figure 9 you see the top view of the module with two jumpers in the recess.

- Jumpers in = terminator enabled
- Jumpers out = terminator disabled

Make sure you place the jumpers as shown in figure 9 and not rotated by 90 degrees. If you can't reach them with your fingers, use tweezers or pliers to grab them.

[3] If you have an RM-U (Classic), it's also possible, but we won't discuss details in this manual because we expect it to be irrelevant for 99% of the readers and just confusing.

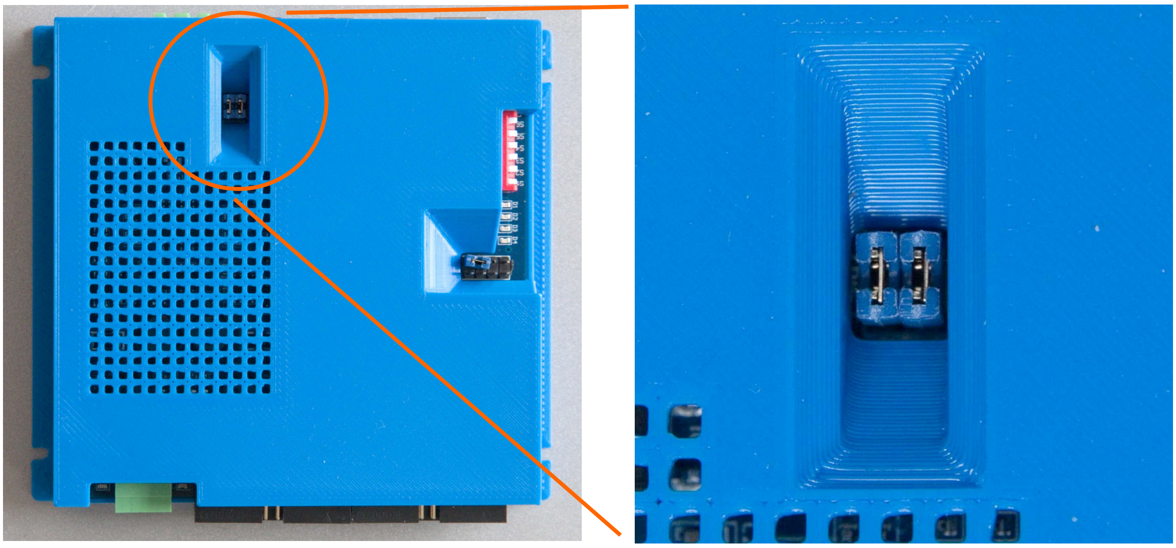


Figure 9: UCCI/E Terminator

Each UCCI/E you include in an RS485 network must also have a unique address. UCCI/E shares the address space with TM44 and TM-H modules. This means that addresses of TM44, TM-H and UCCI/E modules may not overlap. In concrete terms: If you have a TM44 (pair) with address 5/0 and 5/1, you may not have a UCCI/E with address 5.

You set the RS485 address of UCCI/E and UCCI/E-s (easiest) with the DIP switches on the side. Switch 5 must be set to ON for this. Also keep in mind that the address you set will only become active after you restart the module!

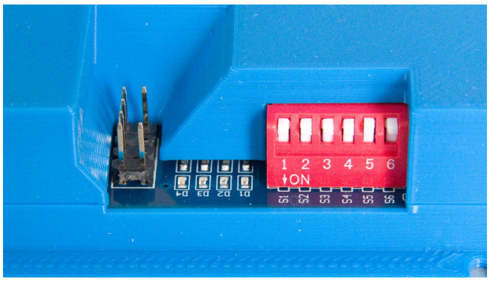


Figure 10: Dipswitch

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

Table1: UCCI/E-s addresses

You will not find a DIP switch on a UCCI/E-u. If you want to use that module in an RS485 network, you must set the address by means of software configuration. Configuring the address is easiest if you connect the module with USB to your PC. For details, see section 6.5.3.

4 Transmission loops

4.1 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 guide strip. The “guide strip”, as meant in this chapter, can be steel wire, but nowadays we advise to use magnetic strip.

The advantage of magnetic strip over steel wire is that the distance between the strip and the steering magnet of the car can be much greater. For example, you can use all kinds of road surfaces, including surfaces with structure, such as cobblestones. You can (with some care) even run agricultural vehicles on grass! Another advantage is that the vehicles can drive with a "floating steering magnet", so that the magnet no longer wears and you no longer get stripes on your road (except from the tires).

The disadvantage of magnetic strip is that you have to make sure that you put the right side up. Furthermore, it is very important that the magnetic strip is “contiguous”. Gaps between strips of more than 0.5 mm lead to local disturbance of the magnetic field and incorrect steering behavior.

Make sure you have the right type of magnetic strip that is magnetized over the entire length N-Z. A width of 3mm and a thickness of 1mm is the most common size.

Because the magnetism of the magnetic strip is permanent and the signal transmission relies only on changes in electromagnetic fields, the permanent magnetism of the strip has no influence on the communication with the vehicles.

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 horizontal plane formed by the wires, so it is important to keep the distance as small as possible.

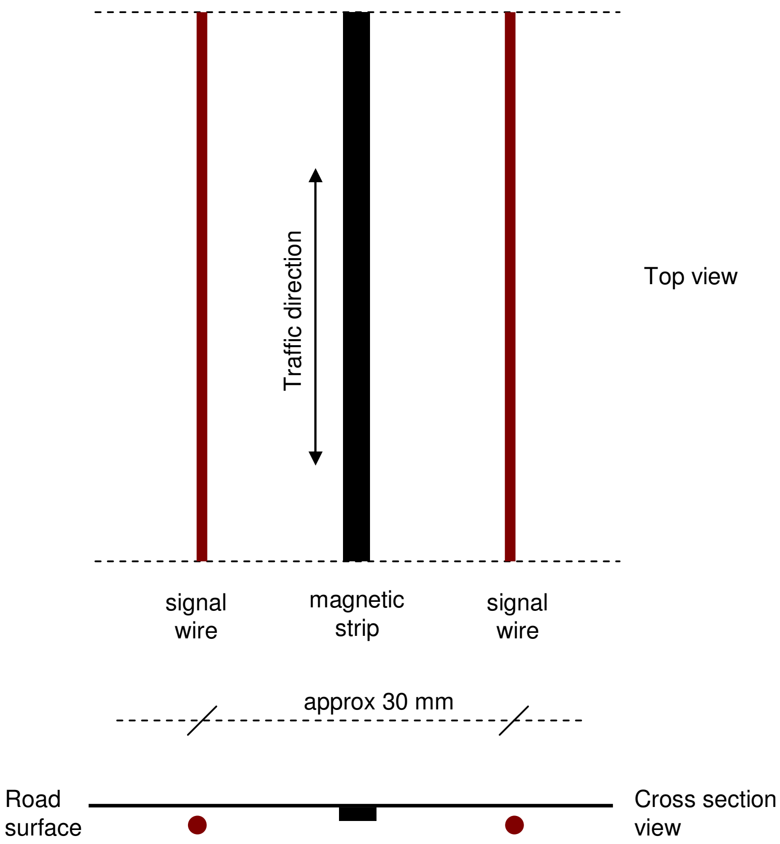


Figure 11: Building the communication system

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:

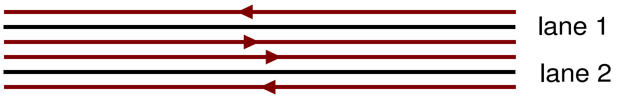


Figure 12: Current direction in a multi-lane road

As a rule of thumb, rule nr 2. applies when the distance between the two wires of different lanes is less than 5 cm.

So before starting to wire your layout it is of importance to think ahead. In order not to make this manual too bulky, we describe the principles here, but there is a separate document[4] in which we describe the "Best Practices" of MCC in more detail.

Below you see an oval with 2 antenna wires in which the currents are neatly opposite. If we just lay these two wires and do nothing else we would have two separate wires and not one continuous wire. Therefore, choose an arbitrary point where you "cut" both wires. Short the two wires coming from the left and short the wires coming from the right. We call this a "Link". The result of a "Link" is that it turns two loose wires into one long continuous wire. Then we still have to provide the continuous antenna wire with signal. To do this, "cut" the wire at any location and connect the wires of a UCCI/E transmitting loop to it. We call this point a "Feed". It is best to use a twisted pair as the connection cable between the "Feed" and UCCI, e.g. a pair of a UTP LAN cable.

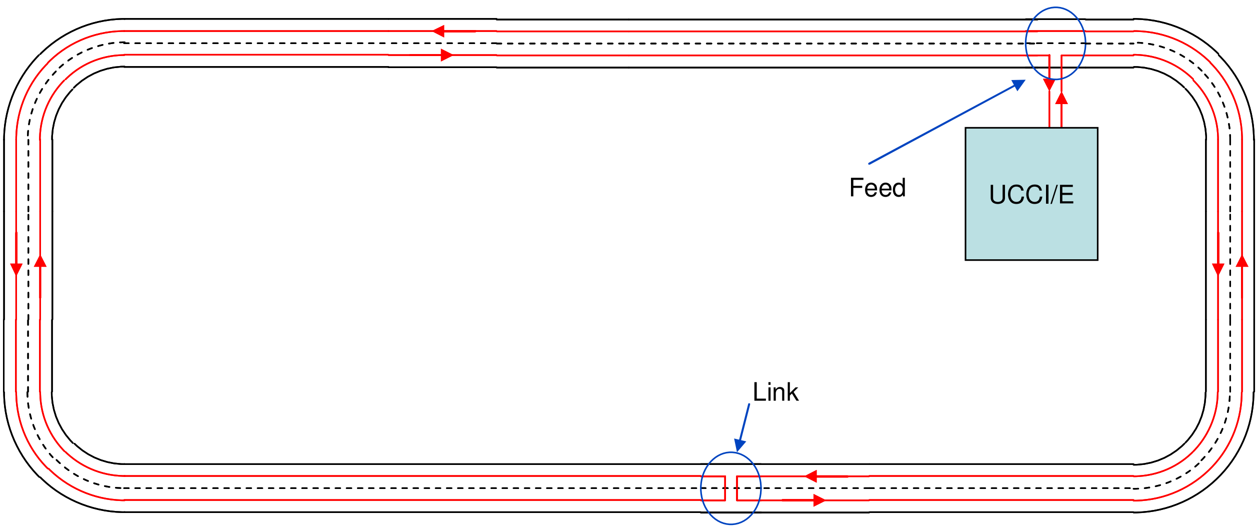


Fig 13: Wiring an 'oval'

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. Figure 14 shows an example how this problem can be solved.

[4] "in preparation" when this manual was issued.

Here we see two ovals, each with a "Link". The inner oval has a "Feed" and thus a signal. To connect both ovals and eventually get one long wire fed by UCCI we use a "Bridge". A "Bridge" links two parallel lanes so that they are connected in such a way that the currents of the wires lying next to each other are equal.

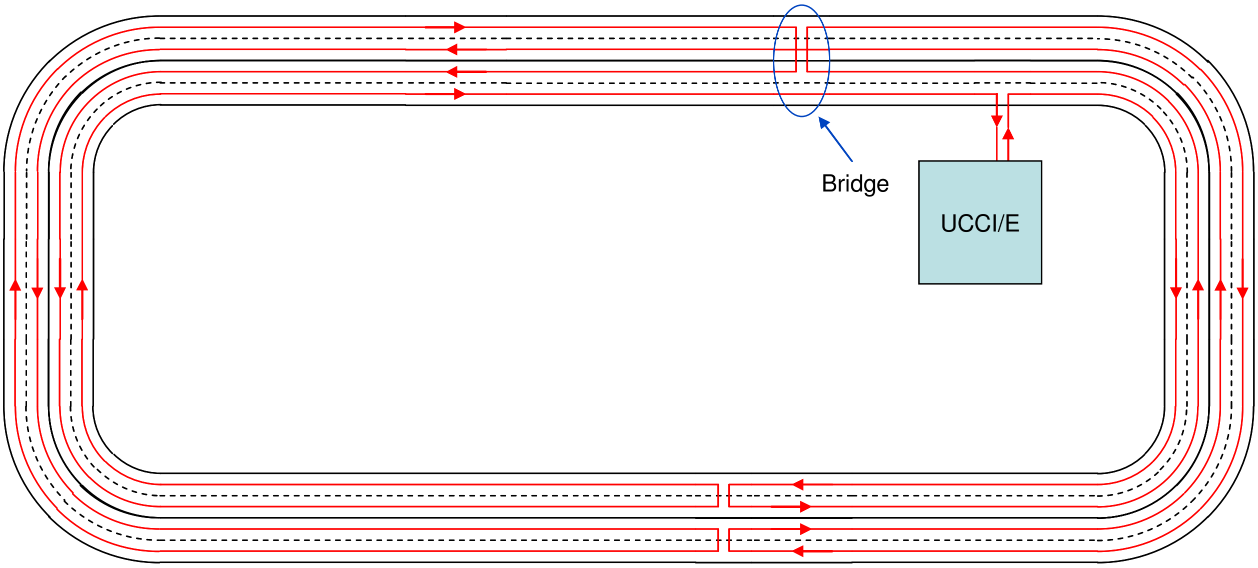


Figure 14: Wiring a double oval

If you have "turnouts" in your layout the puzzle gets somewhat more complicated, but hopefully the example if figure 15 gives a clue how to cope with these situations. The nice thing is that with the elements "Feed", "Link" and "Bridge" you can solve literally any situation. Here we see 6 of these elements. Rules of thumb:

- Each street plan of up to 40 meters of lane has exactly one Feed
- Every stretch of road where two lanes are parallel has exactly one Bridge
- Each "loop" in your plan has exactly one Link

A "Link" element comes in two guises. One of them is that in which the current direction in the wires before and after the link continues in the same direction (circled in green in figure 15). We call this a "Transparent Link". The other form is that in which the currents before and after the link are opposite, as if the link forms a mirror (magenta in figure 15). We call this a "Mirror Link".

With a Feed, Bridge and Transparent Link, if properly installed, there is **no disturbance** of the signal transmitted by the antenna. Electromagnetically, these elements are "invisible". The exception is the Mirror Link. This element introduces a small disturbance. At the location of a Mirror Link you will have a worse signal over a distance of a few cm.

The latter also applies in most cases at intersections and exits. since the antenna wires cannot run parallel there. If the distance over which you have poor reception is less than 10 cm, this will not cause any problems in practice[5]. The decoder has a "timeout" so that the vehicle will continue to drive for a certain distance, even without reception. If necessary, you can adjust the timeout in the decoder. However, remember that in places with poor reception it is better not to plan stops. The vehicles will most likely stop neatly as desired, but you will never be able to command them to move again.

[5] And 10 cm is exceptionally much in practice. The actual distance over which there is no reception will rather be in the order of 1 to 2 cm.

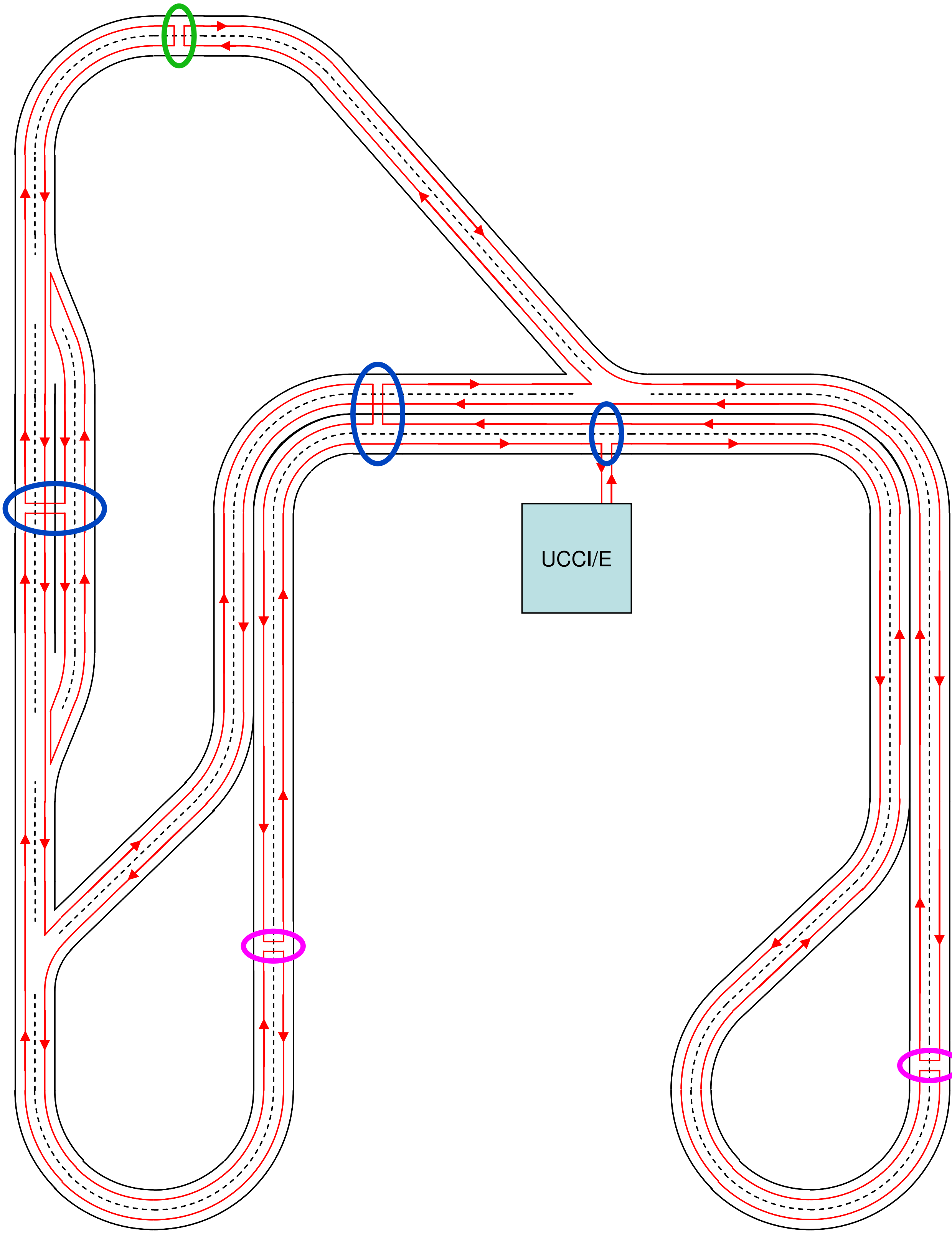


Figure 15: Wiring a more complex situation

If you have designed your street plan with antenna and you want to know if you did it right, there is an effective way to check that. Print out your plan or make a copy. Take a thick pencil or marker, put it on paper at the Feed and follow the wire in the direction of the arrows without taking your pencil/marker from the paper until you are back at the Feed. Touched everything?

4.2 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.

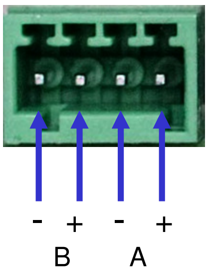
The wire used can actually be any (copper) wire. Preferably use insulated wire, to avoid leaking-currents and unintended short-circuits. Also the insulation protects your wire from physical damage.

The cross-section of the communication wire should be around 0,25mm2 (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 0.14mm2 and 0.35mm2, 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 2: 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.



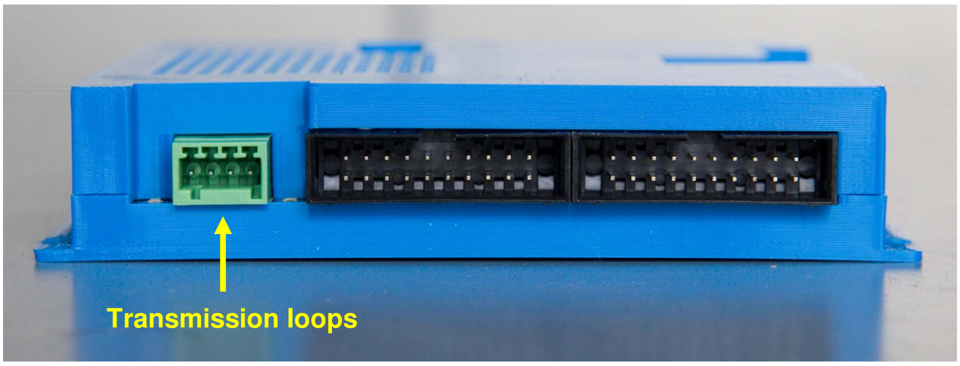


Fig 16: Transmission-loop outputs

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 4.1), so mind this when choosing the + and -.

You'll probably need to cross some distance between the UCCI/E(-s) output and the point (Feed) 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 (a bad connection somewhere or wire to thin)
- Quick flash = UCCI/E(-s) supply voltage is to low in relation to the loop-resistance

4.3 Small scales

The basics as described in paragraph 4.1 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 figure 17 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.

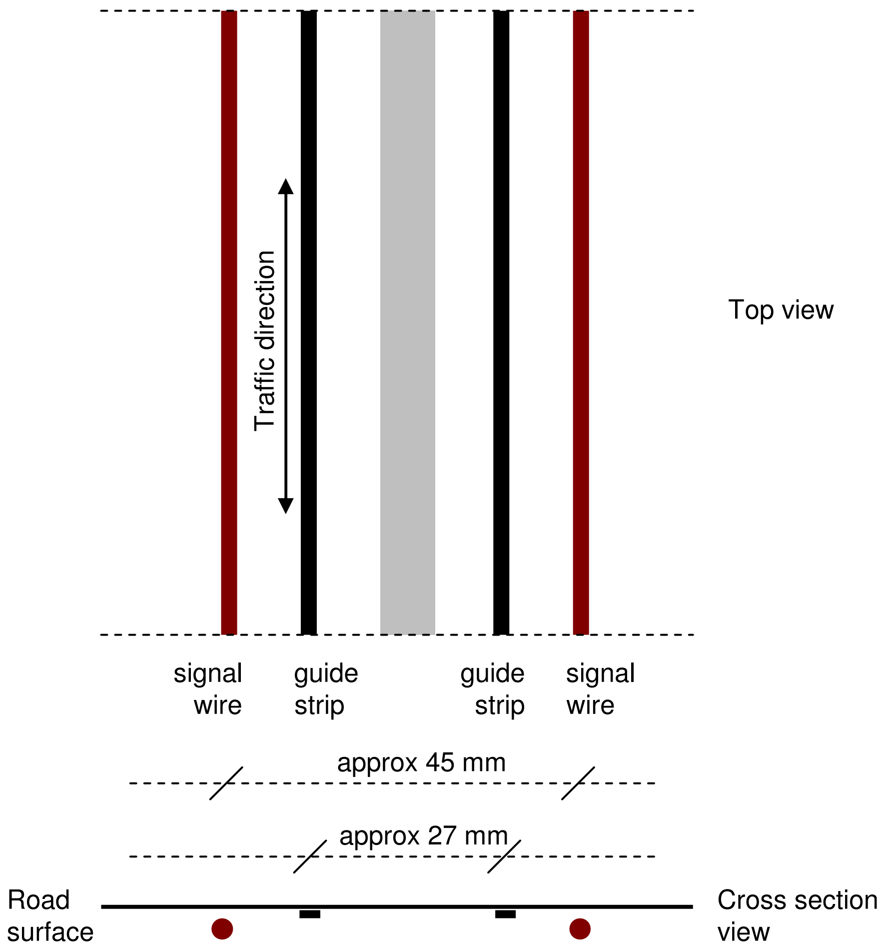


Figure 17: Signal wires in small scales

5 Feedbacks

5.1 Choice

To enable correct control from the software, the position of the vehicles must be reported back to this control software. UCCI/E can read in and process 128 feedbacks. Two techniques are available for detecting positions:

- Reed-contacts
- Hall-sensors

You can use both techniques intermingled. You can judge for yourself whether that makes sense. To help you make a choice, we have the following tips for you:

Reed-contacts	Hall-sensors
Low-cost	Relatively expensive
Somewhat vulnerable during installation	Robust during installation
Reliable and very durable after correct installation	Reliable, however electronic component, so it can fail
Use (almost) no power	Use power
Moderately sensitive used with steel wire Rather sensitive when used with magnetic strip, if correctly adjusted.	Extremely sensitive, automatic adjustment
Can report permanent feedback	Cannot report permanent feedback (pulse only)

Table 3: Differences reed-contacts versus Hall-sensors

The essential difference is actually sensitivity and ease of installation versus price.

If you use reed contacts, they must be of the "normally open" type (i.e. close when a magnetic field is applied), in principle these are the "standard" versions and the cheapest. Preferably choose one with a high sensitivity. A good usable size is approximately 14 x 2.1 mm. A connection module (SWdec) is available for connecting reed contacts to UCCI/E, with which you can easily connect 8 reed contacts per module. The SWdec is ready to buy or to build yourself.

The SHdec connection module is available for connecting Hall sensors. With the SHdec you can connect 8 Hall sensors per module to UCCI/E. The SHdec is only available ready-made and contains a microcontroller and firmware that automatically adjusts the Hall sensors. The Hall sensors to be used are of the linear type.

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

Although steel wire as a guide is cheap, we do not recommend its use in a new installation. Magnetic strip is readily available, very affordable and leads to much better steering behavior. However, because there are also existing layouts where steel wire is used and it may be undesirable to have to break it open, we mention here the possibility.

If you want the vehicle's steering magnet to trigger the reed contact, accurate installation of the reed contacts is required. Make sure that the reed contact is placed parallel to and directly next to the steel wire and directly against the surface of the road surface (see figure 18).

This way of activating reed contacts works well with the steering magnet of H0 trucks and buses. The smaller cars have less powerful magnets that are too weak to activate the reed contacts. In those cases, an extra magnet must be mounted under the car.

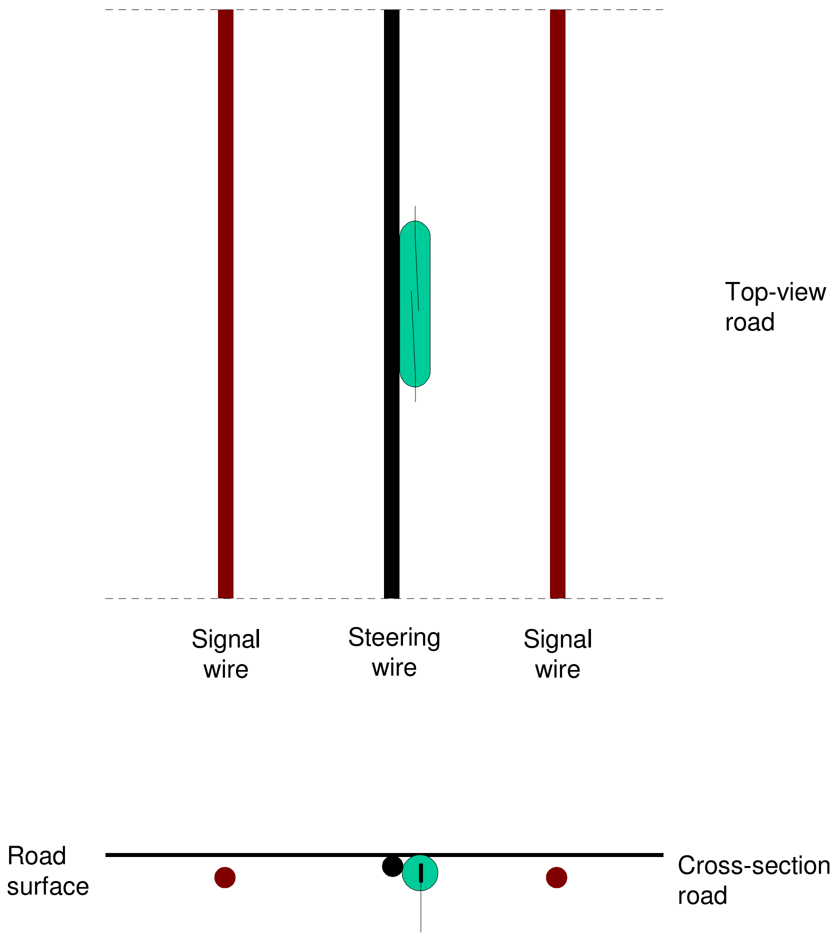


Figure 18: Placing of reed-contacts when using steel-wire for guidance

5.3 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 risky 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 figure 19 below.

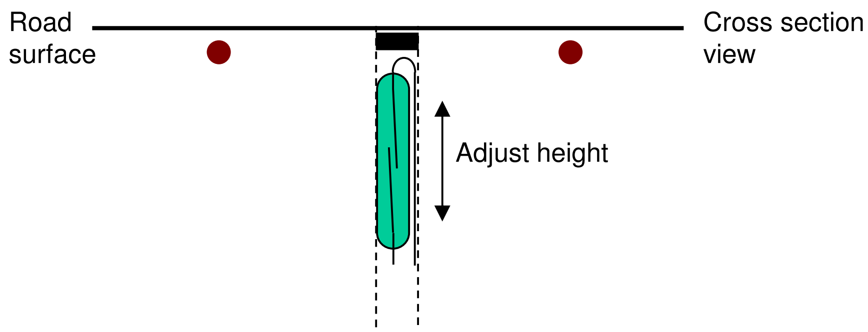


Figure 19: Placing of reed-contacts when using magnetic strip for guidance

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.

5.4 Hall-sensors

We only describe the use of Hall sensors in combination with. magnetic strip. They probably also work well with steel wire and installation and operation are approximately the same. We just never tested it, so we can't make a good statement about it. Since steel wire is hardly used in new installations these days, it also seems less relevant to us.

We assume that you are using the SHdec to connect the Hall sensors and connect them to UCCI/E. A separate manual is available for the SHdec, which also describes the installation and connection of the sensor in detail. Please refer to this. We limit ourselves here to the essentials.

The SHdec is designed for a linear Hall sensor and tested with the Honeywell SS49E. The SS49E has a flat side and a trapezoidal side. The SHdec is designed so that approaching the flat side of the sensor by the south pole of a magnet activates the sensor. The magnetic field of the steering magnets of model cars is standardized with the north pole up. This means that the magnetic strip is also placed with the north pole up, otherwise the steering will not work. As a result, the sensor should be placed with the flat side up. Place the sensor under the magnetic strip, and tight against the magnetic strip. The magnetism of the strip is eliminated anyway and the closer the sensor is to the road surface, the better it works.

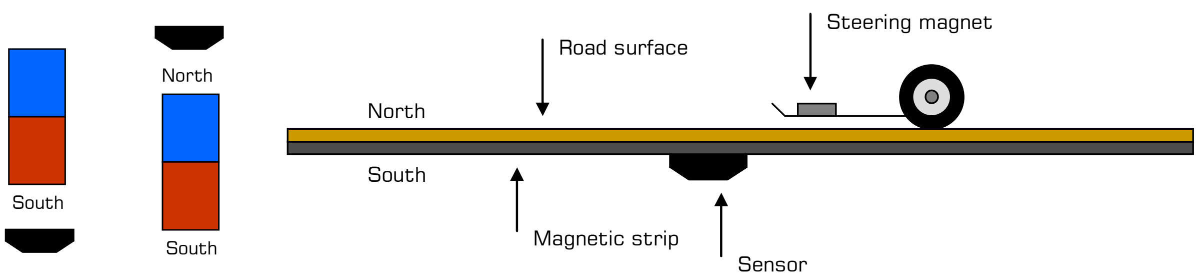


Figure 20: Installation of a Hall-sensor

One might think that the sensor can be placed even closer to the steering magnet by cutting the magnetic strip and placing the sensor in the opening. **Please don't!!** The sensor works fine, but the magnetism of the strip wraps around the edge of the strip at the opening, resulting in strange steering behavior. The car could even lose track in that spot.

5.5 Connecting sensors to UCCI/E

As already indicated, UCCI/E can read and process 128 sensors. We assume that you use the SWdec for connecting reed contacts and the SHdec for connecting Hall sensors. How to do this exactly and what you should pay attention to can be found in the manual of the relevant module. So read that too.

On the back of UCCI/E you will find two flat cable connectors. There is a difference between Rev10 and Rev11, about which more later. You can connect 64 sensors to each connector of UCCI/E by means of a ribbon cable. SWdecs and SHdecs can each accommodate 8 sensors. You can therefore connect 8 SWdecs or SHdecs or, if desired, a combination thereof to one flat cable. In order to recognize and address the individual sensors, each SWdec and SHdec therefore has an address that you set with a jumper.

SWdecs are connected with a 16-wire flat cable. SHdecs are connected with a 20-wire flat cable. The difference is that the SHdec needs power supply which is supplied by the 4 extra wires. Furthermore, the core assignment of the flat cables is identical.

5.5.1 Connecting SHdecs and SWdecs to UCCI/E Rev11

UCCI/E Rev11 has two 20-pin flat cable connections. The right connection in figure 21 is intended for sensors 0..63, the left connection is for sensors 64..127. A 20-core ribbon cable transmits the information for 64 sensors and also provides the power supply for SHdecs.

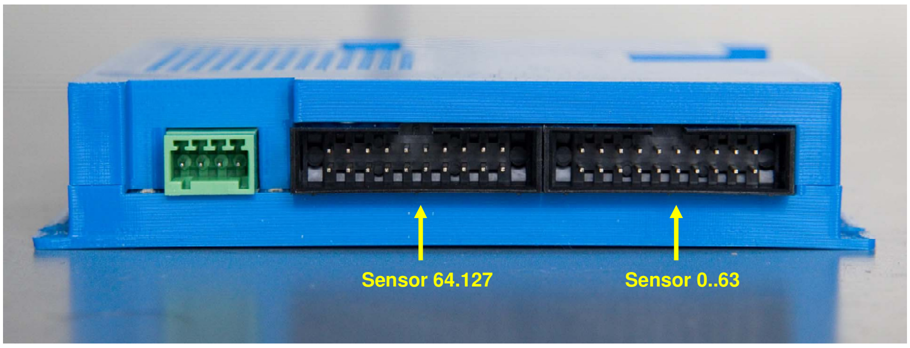


Figure 21: Connections on UCCI/E Rev11 for position sensors

Connecting SHdecs to UCCI/E rev11 is pretty straightforward. Take a 20-core flat cable of (more than) sufficient length to lead from UCCI/E along the first 8 SHdecs. Press a 20-pin ribbon cable connector onto the end of the cable. Note that wire 1, usually marked with a red stripe, is connected to pin 1 of the connector. Pin 1 is marked with a triangle on the connector. Plug the connector into the right flat cable connector of UCCI/E.

Route the cable to the first target SHdec and fit a second connector there. Also pay attention to wire 1 and the triangle on the connector. Plug the connector into the SHdec and repeat until the last SHdec.

Tip: If you do not connect 8 SHdecs, but if you intend to add more SHdecs later, leave enough excess length on the cable to make this possible.

If you have more than 8 SHdecs, repeat the above with a second flat cable that goes into the left flat cable connection of UCCI/E.

In the end, your setup should look like graphically shown in figure 22

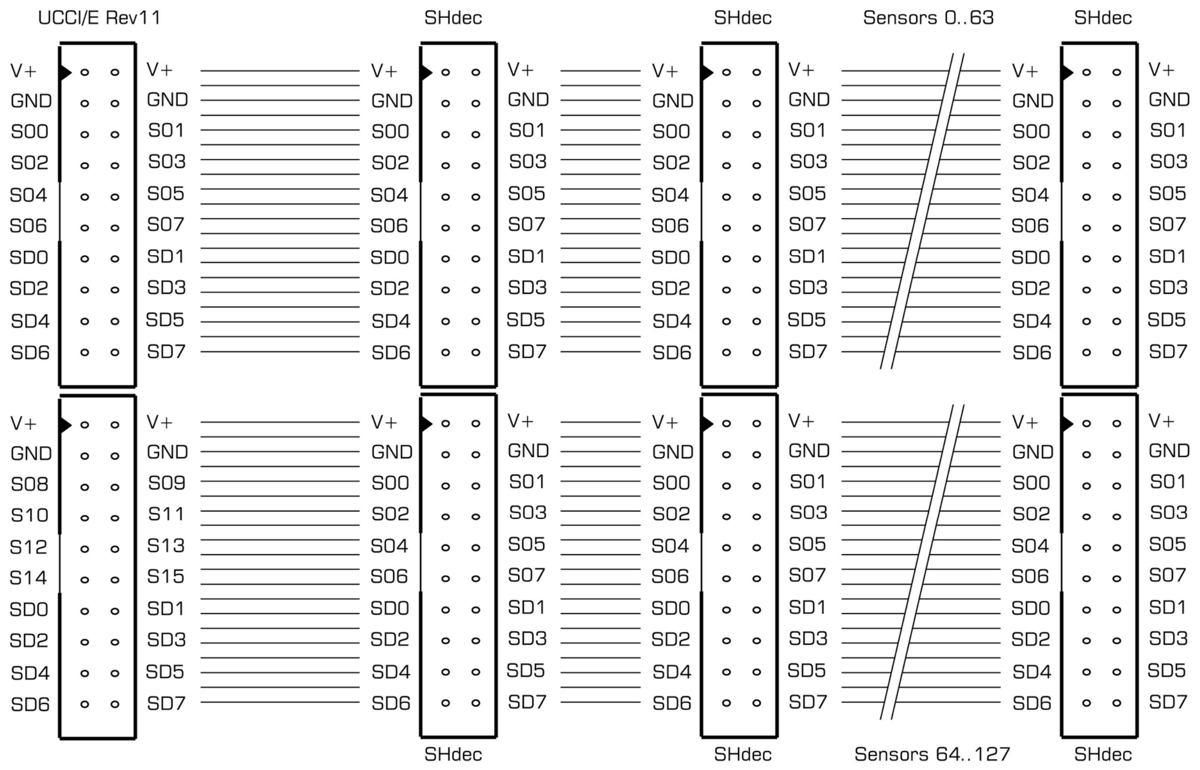


Figure 22: SHdecs connected to UCCI/E Rev11

Connecting SWdecs to UCCI/E Rev11 is almost as easy. The only difference is that a SWdec has a 16-pin flat cable connection. The first 4 wires of the 20-pole connection to UCCI/E are intended for power supply, which the SWdec does not need. The function of the remaining 16 pins is completely identical.

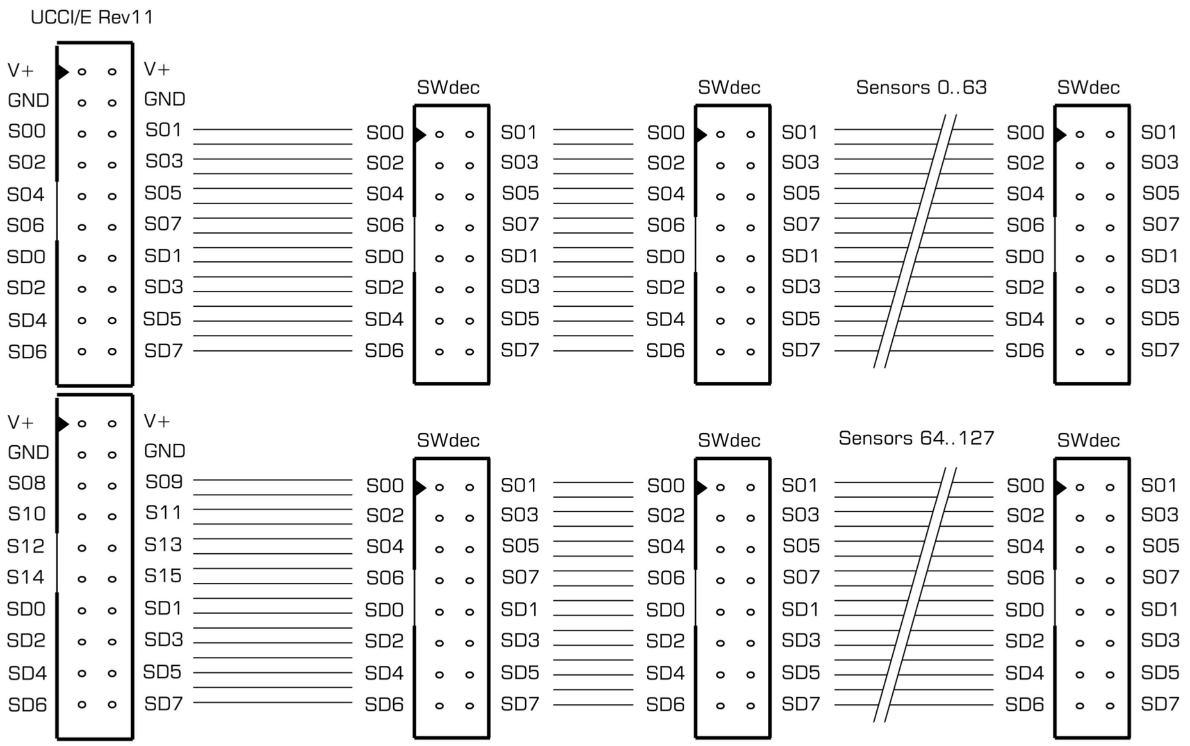


Figure 23: SWdecs connected to UCCI/E Rev11

So to connect SWdecs take a 16-core flat cable of (more than) sufficient length to lead from UCCI/E along the first 8 SWdecs. To connect this to UCCI/E, a 20-pin flat cable connector must be fitted to this cable. Note that pins 1..4 are not used now. Place the 16-wire cable in the connector in such a way that the red wire ends up on pin 4. Wire 16 will therefore lie against the edge of the connector that is furthest from the triangle. There are 4 free blades on the side of the triangle on the connector. Pinch the cable and insert the connector into the right flat cable connector of UCCI/E.

Route the cable to the first intended SWdec and press a 16-wire flat cable connector there. Make sure that wire 1 is now connected to the pin with the triangle on the connector. Plug the connector into the SWdec and repeat until the last SWdec.

If you have more than 8 SWdecs, repeat the above with a second flat cable that goes into the left flat cable connection of UCCI/E.

In the end, your setup should look like graphically shown in figure 23

5.5.2 Connecting SHdecs and SWdecs to UCCI/E Rev10

UCCI/E Rev10 has a 26-pin flat cable connection and a 16-pin flat cable connection. The 26-pole connector is intended for sensors 0..63, the 16-pole is for sensors 64..127.

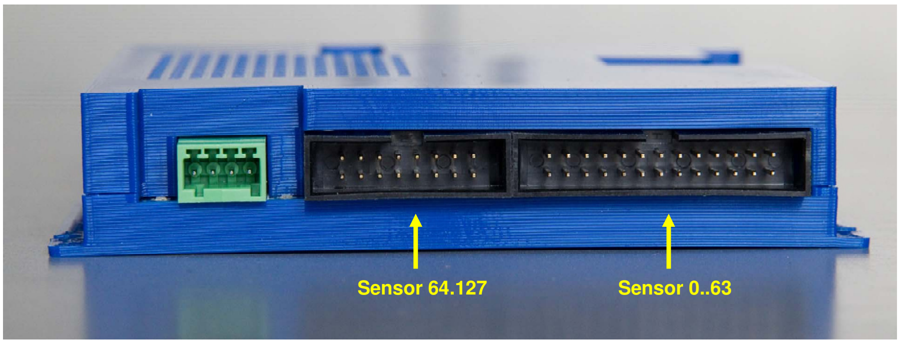


Figure 24: Connections on UCCI/E Rev10 for position sensors

Connecting SWdecs to UCCI/E rev10 is pretty straightforward. Take a 16-core flat cable of (more than) sufficient length to lead from UCCI/E along the first 8 SWdecs. In order to be able to connect this cable to the right-hand connector of UCCI/E, a 26-pole flat cable connector must be fitted. Lay the 16-wire cable in the connector in such a way that wire 1, usually marked with a red stripe, comes to pin 1 of the connector. Pin 1 is marked with a triangle on the connector. On the other side of the connector, 10 connections remain empty. Plug the connector into the UCCI/E 26-pin flat cable connector.

Route the cable to the first intended SWdec and fit a 16-pin connector there. Make sure that wire 1 is near the triangle on the connector. Plug the connector into the SWdec and repeat until the last SWdec.

Tip: If you do not connect 8 SWdecs, but if you intend to add SWdecs later, leave enough excess length on the cable to make this possible.

If you have more than 8 SWdecs, repeat the above with a second flat cable that goes into the left-hand flat cable connection of UCCI/E. Only that second connector is not a 26-pin but a 16-pin. Wire 1 joins the triangle, just like the SWdecs.

In the end, your setup should look like graphically shown in figure 25

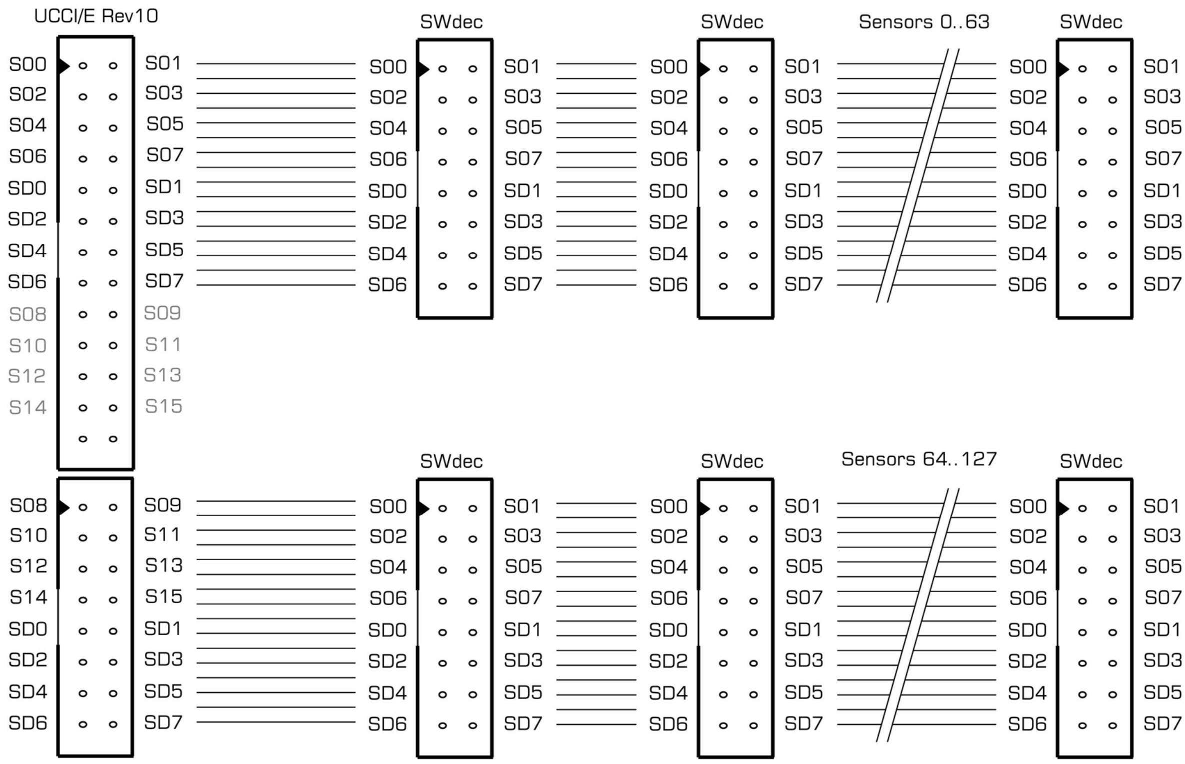


Figure 25: SWdecs connected to UCCI/E Rev10

You can also connect SHdecs to UCCI/E-Rev10. The only problem is that UCCI/E-Rev10 does not provide power for the SHdecs via the ribbon cable connections. You then have to connect this power supply yourself at UCCI/E.

Take a 20-wire flat cable of sufficient length. At UCCI/E, split the first 4 wires (counting from wire 1, normally marked with a red stripe) off the cable over a length of 5 to 10 cm. Place the remaining 16 wires in a 26-pole flat cable connector, such that wire 4, the first wire counting from the red stripe that is still attached to the rest, ends up on pin 1 of the connector (marked with the triangle). So 10 connections remain empty. Press the connector onto the cable.

Now split the 4 remaining wires into individual wires. Strip the insulation over a length of approximately 1 cm. Twist wires 1 and 2 together and twist wires 3 and 4 together. Solder a piece of red wire to wires 1 and 2 and a black wire to wires 3 and 4. Insulate the soldering point, preferably with heat shrink tubing. Now connect the red and black wire to a power supply of between 8 and 12V (red = plus). That could just be the UCCI/E power supply. So you can just screw the appropriate wires into the rectangular power supply connector on the other side of UCCI/E. If you already use that connector to supply power to UCCI/E, simply add the wires. If you use the round connector to supply power, the supplied power will come out on the rectangular connector and you can use it as usual.

Route the flat cable to the first target SHdec and press a 20-wire flat cable connector on the cable at that location. Make sure that wire 1 is now connected to the pin with the triangle on the connector. Plug the connector into the SHdec and repeat until the last SHdec.

If you have more than 8 SHdecs, repeat the above with a second flat cable that comes with cores 4..20 into the 16-pole flat cable connection of UCCI/E.

In the end, your setup should look like graphically shown in figure 26.

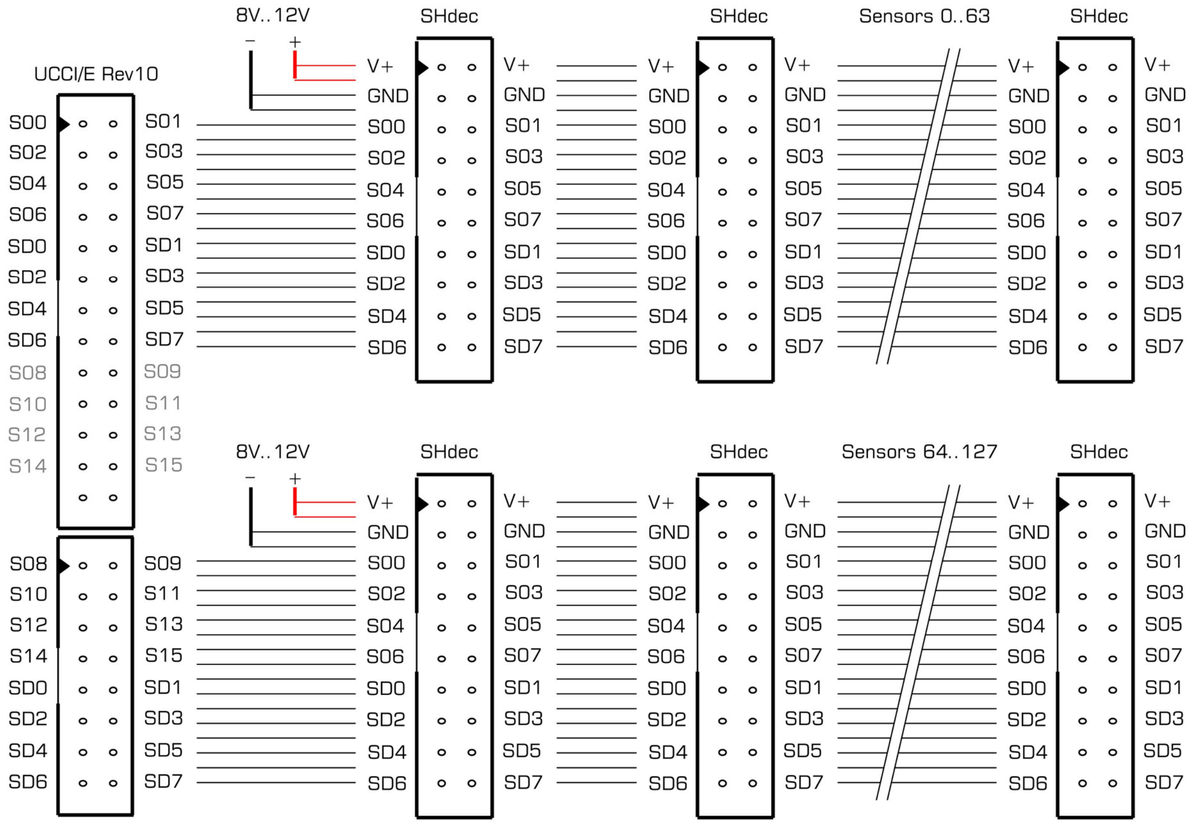


Figure 26: SHdecs connected to UCCI/E Rev10

5.5.3 Connecting both SHdecs and SWdecs

If you have a mix of SHdecs and SWdecs, the first option is to use the connection for sensor 0..63 for the SHdecs and the connection for sensor 64..127 for SWdecs, or vice versa, as you wish.

If that doesn't work and you have to connect a mix of SHdecs and SWdecs to one ribbon cable, this is no problem. Please note that wires 1-4 provide the power supply and are not connected to the SWdecs. If all SWdecs are at the end of the cable, you can simply cut wires 1-4 after the last SHdec and continue with a 16-wire flat cable, leaving the original wire 5 becoming wire 1. If SWdecs are between the SHdecs, split the wires 1-4 from the other wires where the connector for the SWdec should be, over a length of eg 10cm. **Pay attention** that you cut **between** the cores and not **through**. Wires 5-20 now go into the 16-pin connector for the SWdec, wire 5 on pin 1 (triangle). Wires 1-4 continue over the connector to the remaining part of the cable (see figure 27).

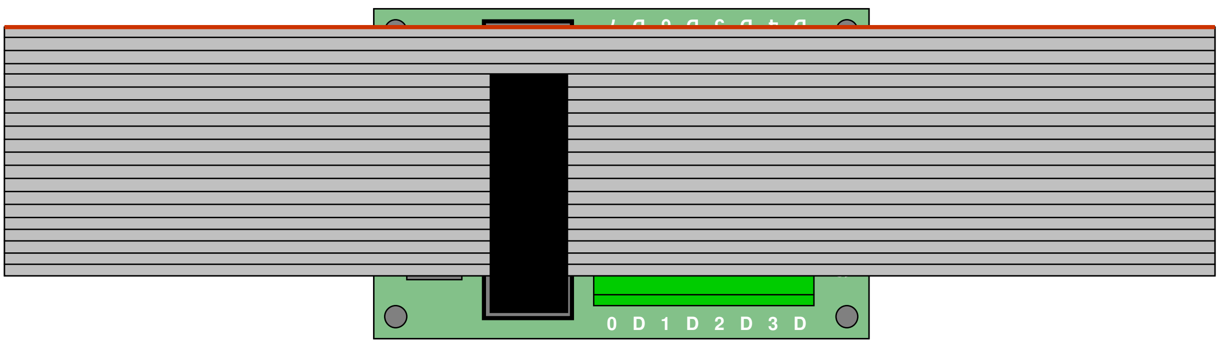


Figure 27: Connecting an SWdec to a 20-core SHdec cable

6 Configuratie en testen

6.1 LEDs

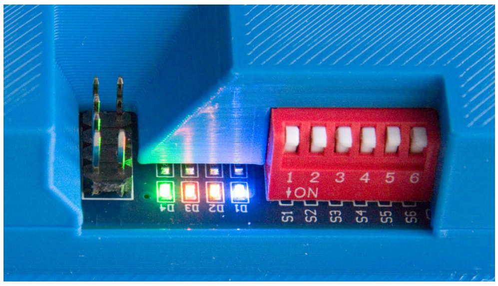


Figure 28: LEDs on UCCI/E

There are 4 general indication LEDs on UCCI/E which give some idea of what is happening. LEDs are located on the side next to the DIP switch (Figure 28)[6]. From right to left:

- Blue: USB interface active
- Yellow: UCCI/E(-s) sends a message on the RS485 network-interface
- Orange: When connected to a PC: UCCI/E sends a message to the PC
  When connected as UCCI/E-s to an RM-x7 central unit: UCCI/E sends a message to the RM-x.
- Green: UCCI/E received a command and is processing it.

6.2 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 1.40B or later

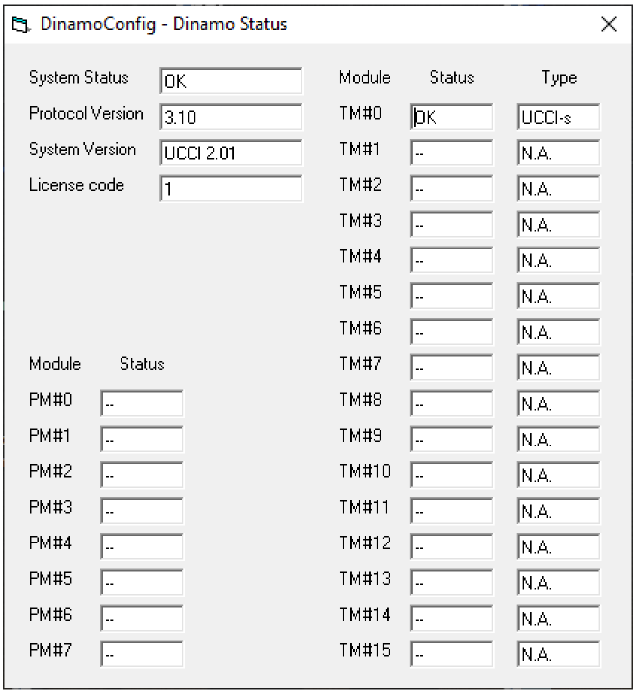
An extensive manual with a lot of extra information is available for DinamoConfig 1.40. Please refer to this. We only mention the essentials in this manual!

Start DinamoConfig and select the com-port to which your UCCI/E is connected. When using USB, this will be the virtual com-port assigned to your 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 you need!

Select the right com-port and click the "Status" button. You should now see the pop-up window of figure 29.

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-x module. But, we don't have an RM-C or RM-U, do we? Well actually the UCCI/E simulates an RM-x with limited functionality plus one virtual UCCI/E-s at address 0. The limited-



Figuur 29: DinamoConfig Status Window

[6] Note that a UCCI/E-u does not have a DIP switch and that UCCI/E-s does not have a USB interface and therefore the blue LED is also missing on that module..

[7] An RM-x does not exist, but we use it as a collective name for RM-U or RM-C since these two modules are functionally identical.

function RM-x can send commands to connected OC32 modules and it can address one integrated UCCI/E-s module. The "connected" UCCI/E-s is shown on the right-hand side as TM#0, Status=OK, Type=UCCI-s. Actually, the system behaves as it were a real RM-x plus UCCI/E-s.

Now close the status window and select the RM-x tab in the main window (figure 30). You are viewing the RM-x functional part of the system and the top-right hand corner tells you that it is an UCCI simulating the RM-x function. There are a number of configuration-settings you can make that we will describe later.

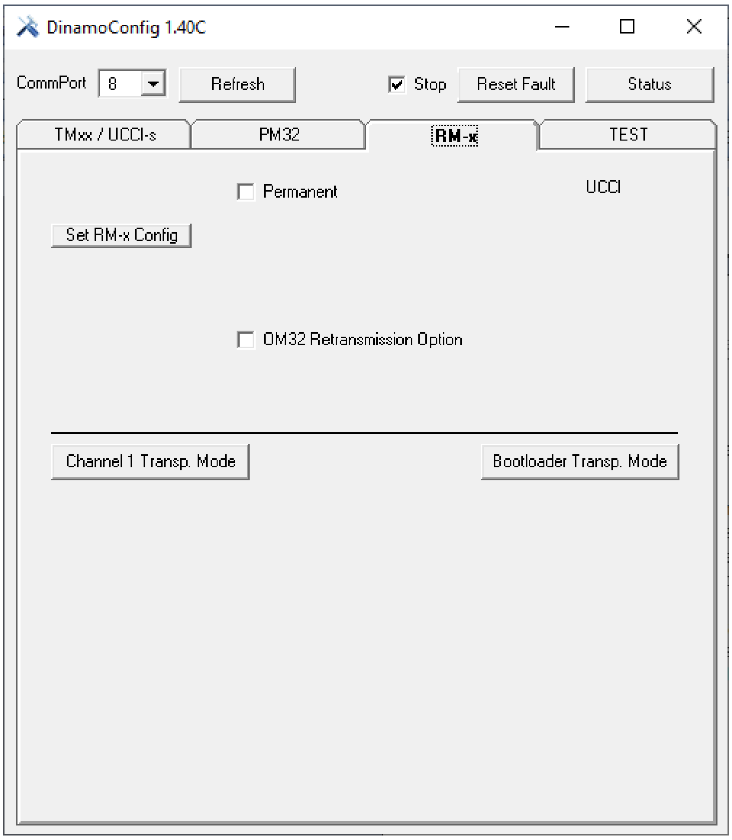


Figure 30: DinamoConfig RM-x tab

Select the TMxx/UCCI-s tab (figure 31). 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.

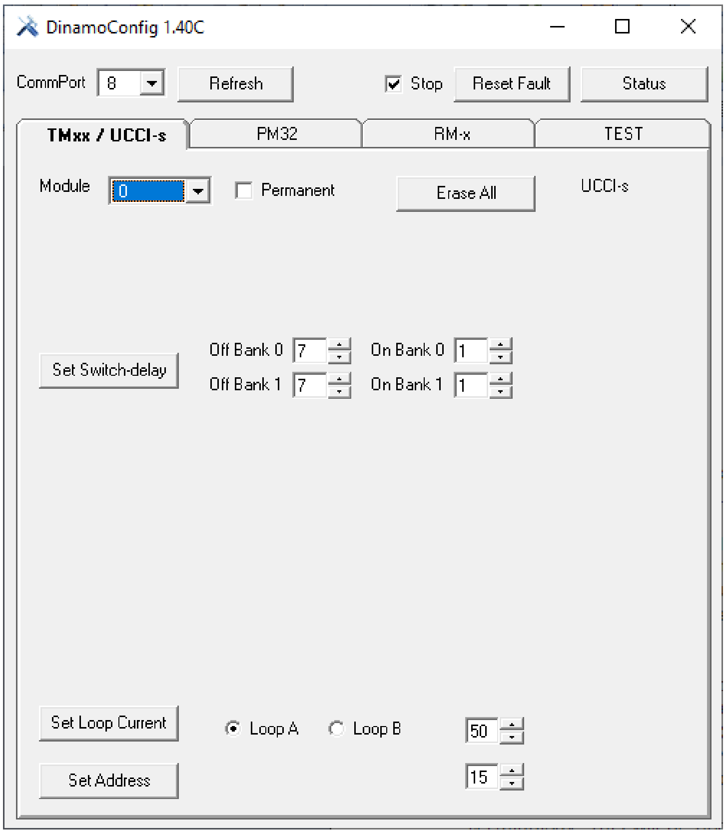


Figure 31: DinamoConfig TMxx/UCCI-s tab

6.3 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.

6.4 RM-x configuration

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

Note: If you have connected an UCCI-E as UCCI/E-s to a real RM-U or RM-C, then you will set up your real RM-U or RM-C with this part of DinamoConfig. After all, the virtual RM-x of UCCI/E is disabled in that case.

6.4.1 OM32 Retransmission

In an autonomous 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 the RM-X in the UCCI/E can send every OM32 message twice. Of course this will slow down communication somewhat, but since the network interface has nothing else to do except sending messages to OC32's this will be hardly be noticeable. You achieve this by setting the OM32retr flag.

6.4.2 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 RM-x part of UCCI/E in transparent mode. From that moment until a "reset" of the UCCI/E, the module behaves as plain USB-RS485 converter.

Transparent mode can be activated by clicking "Channel 1 Transp Mode". You can also set Transparent Mode directly from the OC32Config program.

While Transparent Mode is active, the orange LED on UCCI/E flashes When sending data from PC to the RS485 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 (only) be done by rebooting your UCCI/E.

6.4.3 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 converter.

Bootloader Transparent Mode can be activated by clicking "Bootloader Transp. Mode".

While Bootloader 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 (only) be done by rebooting your UCCI/E.

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

6.5 UCCI-s configuration settings

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

6.5.1 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.

6.5.2 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.

6.5.3 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 or RM-C. 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 (easier) set the address by DIPswitch (see paragraph 3.4).

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.

Note (3): Setting the address of an UCCI/E that is linked as UCCI/E-s to an RM-x can be tricky because the module already has an (active) address. Since only modules with a USB interface do not have a DIP switch, setting the address is therefore easiest if you connect UCCI/E directly to the PC with USB. You can then set the address, while it has no effect at that moment.

6.5.4 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.

6.6 Testing

Starting from version 1.40, DinamoConfig offers testing features to test your system. It is even possible to drive (trains and) cars this way with, but with more than one or two vehicles it is almost impossible to control. So to really run your layout you need software that is intended for that.

The right tab is called TEST and looks like shown below.

At the top right of the tab you will find “System Status”.

NOTE: This status is important if you want to drive something!

- Disconnected: There is no connection to Dinamo. Of course nothing will drive.
- Fault: There is an error status. There are no vehicles driving until you clear the error status by using “Reset Fault”
- Stop: All vehicles have been stopped manually. You can turn "Stop" on or off with the check box at the top.
- Run: Vehicles can drive.

In the middle you will see three sub-tabs for analog trains, digital trains and cars.

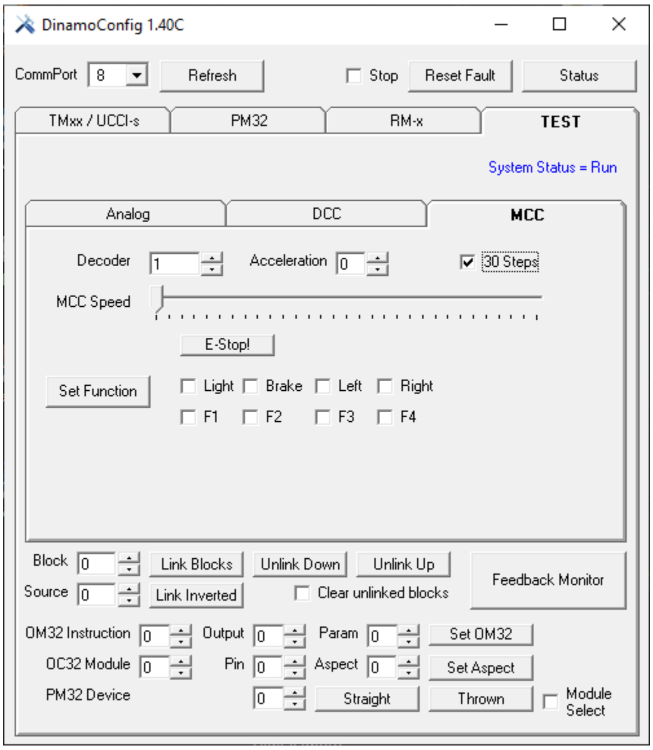


Figure 32: DinamoConfig MCC tab

6.6.1 MCC

The MCC tab is for controlling cars. To drive a car you have to choose a decoder number.

NOTE: You can select decoder number 0. This decoder number is a “broadcast”. This means that **all** decoders will respond to this.

“Acceleration” selects the Acceleration Index that you can configure in the car's decoder. Normally 0 is fast, or immediately and a high number delayed, but you can adjust that yourself in the decoder of the car.

With “30 steps” you indicate whether you want to control the decoder with 15 or 30 speed steps. Controlling a 15-step decoder with 30 steps is not a problem in itself. The decoder will only see the odd steps as the next even step.

“E-stop” generates speed 0 with Acceleration Index 0 for the car.

In the lower half you can control the functions Light, Brake, Left Right and F1..F4.

6.6.2 OC32 control

In the lower part of the TEST tab you can control a number of other functions. Besides the Feedback Monitor (see 6.6.3) the only two relevant to MCC are the OC32 SetAspect instruction and the OM32 Instruction.

NOTE: With UCCI/E as central module, the OC32 instruction does not work! The reason is that UCCI/E controls the OC32 modules as OM32 modules. If you use an RM-x as a central module and use UCCI/E as UCCI/E-s, OC32 control will work.

To issue an OC32 command via a UCCI/E as a central module, use the OM32 instruction. If you want to give an OC32 SetAspect instruction, choose OM32Instruction = 1. Output is the Pin you want to control, 32 Pins per OC32 module. Param is the Aspect in which you want to place the relevant Pin.

You can also give (other) OM32 instructions. For details, please refer to the OC32 manual and specifications.

6.6.3 Feedback Monitor

The Feedback Monitor is opened with the button of the same name on the right. You will then get approximately the following window:

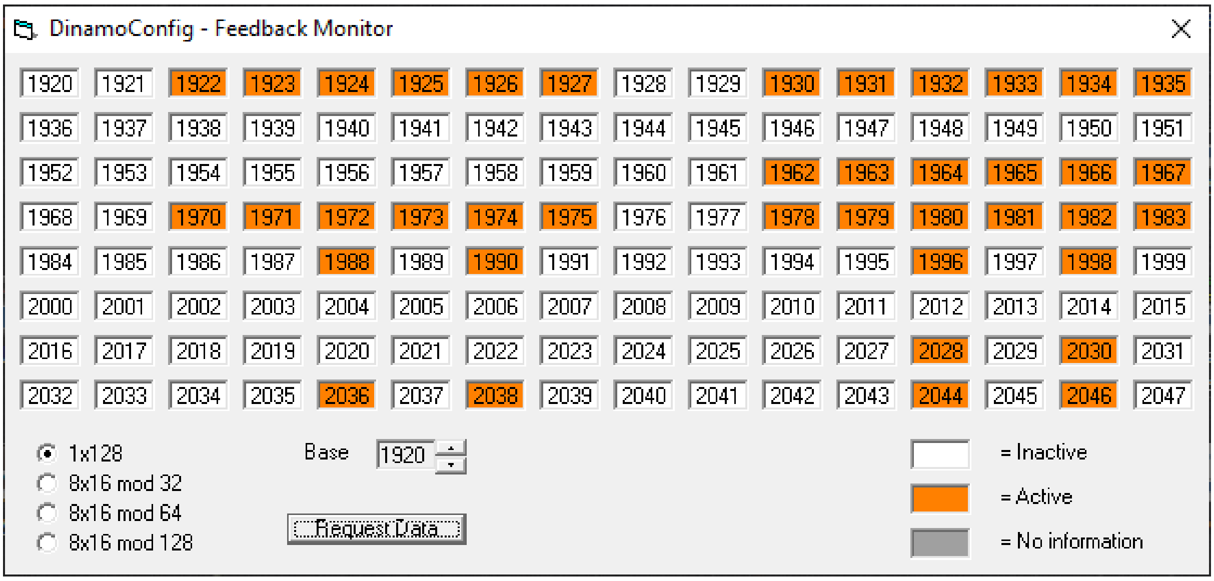


Figure 33: DinamoConfig Feedback Monitor

The window shows the status of 128 feedbacks. There are several “views” available. For UCCI/E-s you preferably choose 1x128. At “Base” you choose which feedback you want to see. UCCI/E-s#0 starts at 0, UCCI/E-s#1 starts at 128, etc

The button “Request Data” requests the feedback status of the feedbacks that are currently “in view”.

7 Firmware Update

7.1 UCCI/E Bootloader

UCCI/E is delivered with Bootloader facility. A Bootloader is a small piece of extra firmware (software) in the UCCI/E processor, which makes it possible to provide UCCI/E with new software when it is made available by VPEB. UCCI/E can be updated via the USB interface or the RS485 network interface. To be able to perform an update/upgrade of UCCI/E, you must have the following items:

a) A PC with the Windows operating system
b) If UCCI/E is connected to the PC by USB: a USB interface on the PC, or
If UCCI/E is part of a network controlled by an RM-C or RM-U: An RM-C or RM-U connected to the PC (so when using an RM-U, RS232 is also an option).
c) VPEB Bootloader software (on your PC).
d) The latest version UCCI/E firmware to install.

Regarding a) and b):

So, in principle this can be “the system” with which you control your railway or street plan.

Regarding c):

Bootloader software is a program on your PC that allows you to load the firmware for UCCI/E into the UCCI/E processor.
NOTE: The VPEB Bootloader software is universal for all VPEB modules that support a Bootloader. So if you have already installed the Bootloader software, e.g. for an OC32, you do not have to do this again and you can skip step 1 of paragraph 7.2.

Regarding d):

Firmware is the software that has to go **into** UCCI/E itself and that ensures the operation of the UCCI/E. The Bootloader software on the PC and the Bootloader in the UCCI/E-CPU together ensure that you can install the UCCI/E firmware.

Regarding c) and d):

The VPEB Bootloader software and new firmware for UCCI/E can be found on the DinamoUsers portal (www.dinamousers.net). The prerequisite for accessing this software is that you have registered on the above portal and that you have Dinamo Customer Status. Registration is free and possible for anyone who agrees to the terms of use and you get the Dinamo Customer Status for free, or you can request it if you have purchased an UCCI/E or any of the other VPEB products.

7.2 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 your UCCI/E is connected via an RM-C or RM-U and you haven't done this before: Install DinamoConfig on your PC, preferably version 1.40B or newer
3. Download the UCCI/E 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. Make sure the UCCI/E is connected to the PC, directly or via an RM-C/RM-U;
5. Power up your Dinamo/MCC system;
6. If you have an UCCI/E directly connected to your PC by USB: go to step 7
   If you have a system with RM-C/RM-U and UCCI/E(-s): Start DinamoConfig. Select the com-port by which your RM-U is connected to the PC.
   If you have an RM-U: 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-x. Switch the RM-C/RM-U to Bootloader Transparent Mode by cicking "Bootloader Transp. Mode". At the RM-U/RM-C now the red/orange LED shall be lit (continuously) and the blue LED if you use USB. Close DinamoConfig.
7. Start AVRootloader.exe. You'll see the window similar to fig 34;
   The baudrate is at 38400 and "Sign" shows "VPEBbootloader". **Do Not Change This!**, otherwise it won't work!

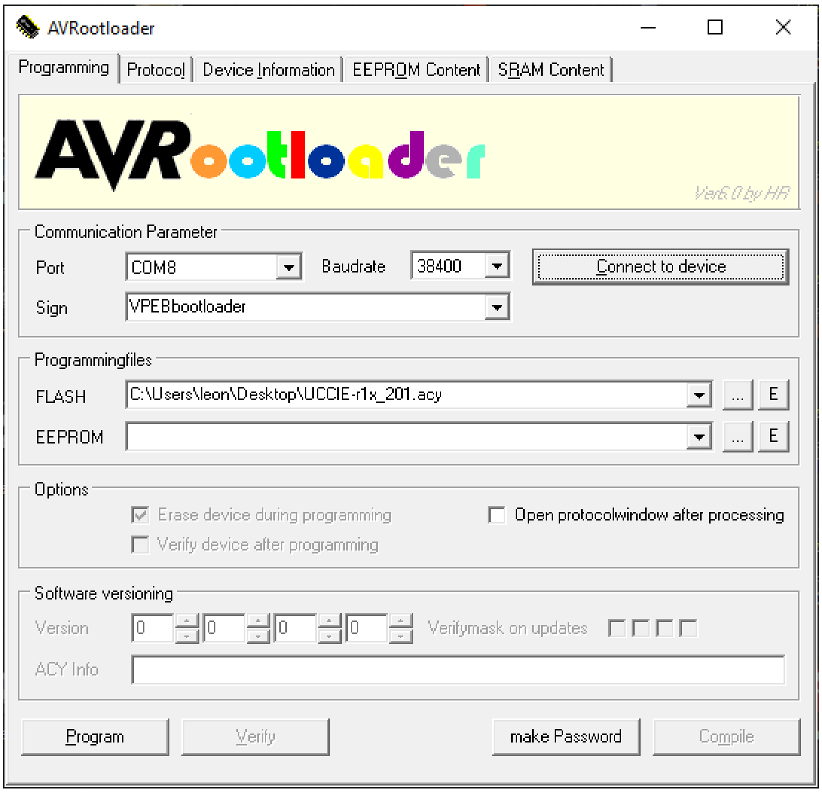


Figure 28: AVRootloader

8. Set "Port" to the com-poort by which your PC communicates to your system. 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. So you will manually need to select the right com-port.
9. 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. Mind

that you select file type "Encrypted Programming File (*.acy)", while searching for your file, otherwise you won't see it.

10. Reset the UCCI/E(-s) you wish to update. You do this by placing a jumper on the Reset pins (fig 35) and leave it there. Usually there is a spare jumper you can use, parked on one of the pins of the jumper-block.

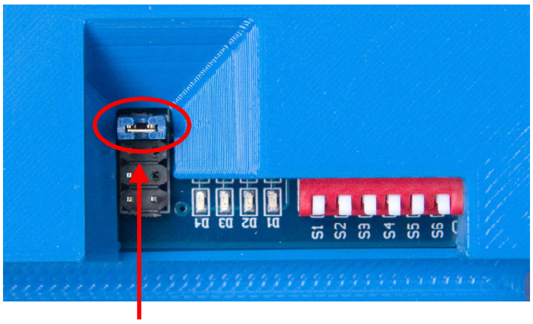


Figure 35: Reset

11. 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-C/RM-U, the green LED flashes, while the red one remains permanently on;
12. Now take the reset-jumper from the target UCCI/E "in a fluent motion" and put it aside. The green, orange and yellow LED on the (target) UCCI/E should be lit.
    If you communicate through an RM-C/RM-U, the green and one of the yellow LEDs on the RM-C/RM-U flash, while the red one remains permanently on. The top of the AVRootloader window now shows "connected".
13. Now (in the tab "Programming") click the button "Program". The button that was marked "Disconnect device" now shows "working" and after a few seconds "Disconnect device" again.
    If you don't see any errors reported, your new firmware is installed in your UCCI/E. If you want to see what happened, you can check the "Protocol" tab.
14. Click "Diconnect device" (in the "Programming" tab). UCCI/E now starts with the new firmware.
15. Store the reset-jumper on the parking-position where you found it;
16. If you have a system with multiple UCCI/E modules and you wish to update these as well (recommended), repeat the above steps from step 10;
17. Close AVRootloader
    If you had to modify connections, restore the original configuration. If you communicate through an RM-C/RM-U, reboot or reset this module. Your system is ready for use again.

This page is intentionally left blank