SMART OCCUPANCY DETECTOR FOR MODEL RAILROAD

Jan Horáček[1], Jiří Rybička[2]

[1]Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 13 00 Brno, Czech Republic
[2]Department of Informatics, Faculty of Business and Economics, Mendel University in Brno, Zemědělská 1, 13 00 Brno, Czech Republic

ABSTRACT

For control of a model railway, simple track occupancy detectors are in use to detect the presence and absence of a vehicle on the track. However, these detectors do not use the full potential of a digital command control (DCC) system. A RailCom technology allows information to be obtained from the ve- hicle's DCC decoder, thus the detector can receive operationally important data from the rolling stock vehicle. The subject of the paper is the description of requirements and design of an own RailCom detector MTB-RC, which presents an alternative to commercially available model railway RailCom detectors. MTB-RC is an open-source and open-hardware project, which is also compatible with the rest of the trackside hardware used in the Track Vehicle Control Laboratory FBE MENDELU. MTB-RC can read the addresses of the DCC decoders present on the track and transmit them to the railway control software via MTBbus.

Keywords: railroad, model, interlocking, RailCom, occupancy detector, Digital Command Control, DCC, economic sustainability

JEL Code: L920

1 INTRODUCTION

The model railway in the Track Vehicle Control Laboratory is a complex computer-controlled electronic system. One of the key components required for the safe operation of a (model) rail- way is a track occupancy detector. The detector detects the presence of a vehicle – an engine or a car – in a section. The railway is then divided into multiple sections, each section is connected to one detector. A small model station with 3 tracks can contain circa 10 sections.

In the model, the track occupancy detector detects whether the current flows through the rails – from one rail to another. In Digital Command Control (DCC) (National Model Railroad Association, 2024b) system, each engine contains a decoder (a small electronic module) which draws some small current continuously. Each axle of a car contains an embedded resistor (15 kOhm) to draw the current as well. Thus any present vehicle in the model track draws current and this current is detected by the track occupancy detector as a voltage drop on a diode (other methods to detect current exist too).

https://doi.org/10.11118/978-80-7509-990-7-0075

In some situations, however, a different approach to occupancy detection is beneficial. An extension of a DCC protocol called *RailCom* allows the information to be read from the vehicle's decoder by the detector (DCCWiki, 2024a). The decoder must support RailCom, but most of the decoders nowadays have the support implemented and even enabled by default. Also, the RailCom detector only reads information from the decoders, so cars are not detected by the detector. Thus, the RailCom detector cannot simply replace the current-based detector mentioned previously. The RailCom detector must be perceived as an extension of the current-based detection. A technology with different usages, different implementation, and different price.

2 DCC

The Digital Command Control (DCC) protocol was created in 1992 by National Model Railroad Association (NMRA) (DCCWiki, 2024b). The association in cooperation with companies producing model rail- road hardware created a DCC Working group, which then released standards describing DCC from electrical and protocol point of view.

The DCC was designed as a one-way protocol. The data flows from the *command station* to the vehicle's decoders via tracks. The command station is controlled by throttles or computers. The main purpose of the command station is to generate a DCC signal. The signal is sometimes amplified by a *DCC booster*. The decoder is usually a small electronic module (e.g. 1×3 cm) with a powerful microcontroller that decodes DCC and controls the vehicle. Each decoder has its address, so different engines can be distinguished. The decoder controls mainly (1) motor, (2) lights, (3) sounds.

DCC transmits data in packets using voltage between rails (National Model Railroad Association, 2024a). We call wires from the com- mand station or a booster to track lines J and K. At each instant of a time, one rail compared to another has either positive or negative voltage. Length of pulses distinguishes logical 0 (100 us pulse) and 1 (58 us pulse). Each packet can contain a different number of bytes (see Figure 1).

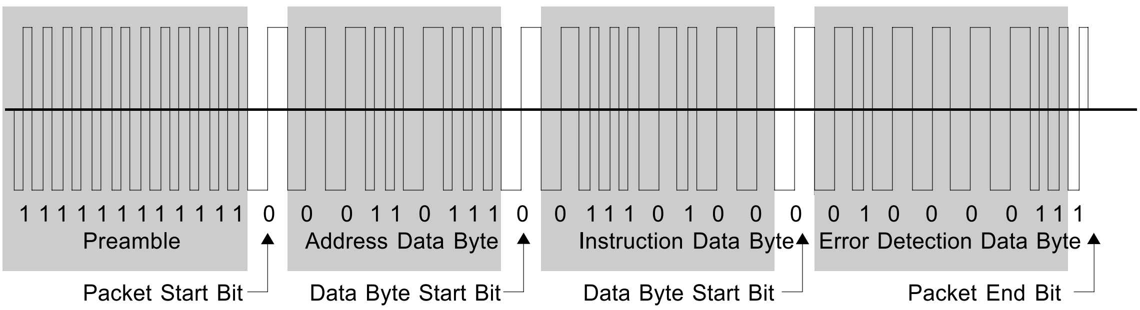

Fig. 1: An example of a DCC packet (National Model Railroad Association, 2024a)

The command station transmits packets one by another, cyclically addressing all decoders it wants to command. It is important to send the data (e.g. speed command) cyclically, because the command station does not know whether the decoder received the command successfully. The decoder could be temporarily busy, the engine could temporarily lose electrical contact with rails, etc.

Detailed specification of a DCC protocol is available in National Model Railroad Association (https://www.nmra.org/sites/default/files/s-92-2004-07.pdf).

3 RAILCOM

In 2006, an NMRA standard (NMRA, 2024) describing an extension of a DCC protocol was released. The standard describes so-called *RailCom protocol*, which allows the mobile decoders to transmit data back to the command station while normally operational.

The principle is as follows: in the space between DCC packets (right side of a Figure 1), the decoder transmits data. The space is called a *cutout*. The decoder contains a capacitor which gives it enough power for transmission. An overview of components participating in a RailCom transmission is shown in Figure 2.

RailCom transmission is current-based. To transmit a 0, the decoder must supply a current of 30+4/-6 mA with a voltage drop at the track of up to 2.2 V. To transmit a 1, the current must be at the most ±0.1 mA (NMRA, 2024). Compared to a voltage-based design, current-based transmission is much more resistant to interferences. However, a special switch in the command station or a booster is required. The switch just connects output J and K wires, so the current loop is closed. There are command stations and boosters with and without RailCom support. Thus, RailCom-supported booster or command station is required.

The last part in 2 is a *detector*. The detector is a device, which reads data from the decoder. The detector senses the current generated by the decoder and reads the data. Note there may be multiple detectors in the current loop, which is beneficial in some situations.

From the data point of view, RailCom transmission is a 250 kbis/s single-direction UART (NMRA, 2024). This is a large gap from DCC communication, which has an average speed of 6 kbit/s.

The decoder transmits data in 2 channels serially. Channel 1 usually contains an address of a decoder transmitting the data, channel 2 contains a response to a DCC command requesting the decoder to respond (e.g. read configuration from the decoder or acknowledgment).

4 REQUIREMENTS

Based on the possibilities of the RailCom, we infer use-cases in which the protocol is beneficial and the requirements for the solution.

We will focus on the address-reading capability of RailCom. There are other important usages – e.g. the command station does not have to resend data to the decoder cyclically, if the decoder confirmed (via RailCom) that it received the DCC packet correctly. However, this paper aims to focus on a decoder-address-reading capability.

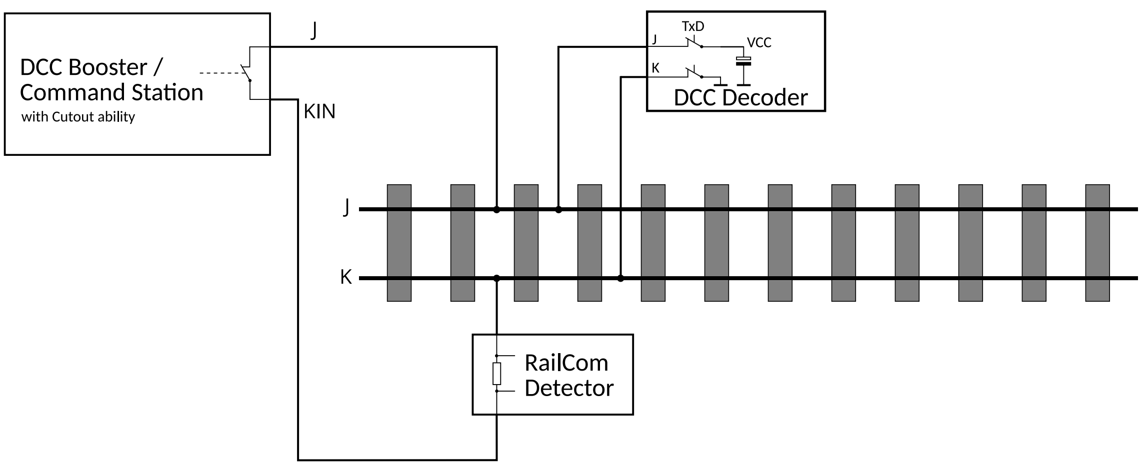

Fig. 2: Components participating in a RailCom transmission

Reading a decoder's address is beneficial in situations when the address is unknown. Usually, when the train is in movement on a computer-controlled railroad like the one in Track Vehicle Control Laboratory, the software already knows its address, because it must have com- manded it to move. We identified the following situations, in which RailCom technology is useful.

1. **Shunting in a train depot.** In the depot, engines are usually changed between rails fast and often. They are even sometimes packed or unpacked by the owner. Moving the responsibility to track vehicle addresses from the operator to a device seems to be beneficial.
2. **Track for new engines.** There is usually a track or tracks at the model railroad specifically designed for mounting new vehicles. The detector may automate the creation of a new train by automatically detecting the new engine's DCC address.
3. **Handover track.** When part of the track is managed by another railroad control system, the detector may be used on the entering track from a foreign system to detect incoming vehicle addresses.

Based on the use-cases, the detector should support several track circuits. The detector should be able to read the decoder's address. If more vehicles are present (e.g. multitraction, multiple engines on a track in a depot, etc.) the detector should be able to read the addresses of all of them. The detector should transmit detected addresses to the *train control system*. The detector should be financially available, customizable, and fixable/replaceable for long-term sustainability.

5 STATE OF THE ART

There are several industrial solutions for obtaining the address of a vehicle on the track.

5.1 LRC120 by Lenz Elektronik

Lenz Elektronik manufactures the *LRC120* RailCom Display module. The module can display the address of the decoder in the connected track section on the four-digit display. The module is available from many vendors, for example, AMB Models (AM Modely, 2024).

The main disadvantage of the module is the absence of an interface for reporting the address.

5.2 10808 detector by Roco

One of the largest companies producing vehicles and equipment for model railways offers the *Z21 10808 detector*. The detector can be connected to the Roco Z21 command station using R- Bus or CAN. It supports 8 sections and allows various settings. To use RailCom, the CAN bus must be used together with Roco Z21 command station (Roco, 2024).

From our point of view, a disadvantage of this solution is a vendor-lock requiring the use of the Roco command station. Deployment would also require adding cabling.

5.3 YD6016LN-RC by YaMoRC (Digikeijs)

The company Digikeijs, which offered a widespread and popular DCC hardware recently announced the end of its activities. The successor company is called YaMoRC, the products are similar.

YaMoRC offers a YD6016LN-RC RailCom track occupancy detector (similar to the former Digikeijs DR5088RC) with 16 track sections. The detector supports per-track channel-1 address reporting and one global detector for reading other data from vehicles. All information from the YD6016LN-RC is transported via the *LocoNet* bus, with firmware updates possible over a USB connection. The module is configurable over USB (YaMoRC, 2024).

Detector	N.o. tracks	Buses	Price
LRC120 *	1	N/A	1482 CZK
Roco 10808 **	8	R-Bus, CAN	2980 CZK*****
YD6016LN-RC ***	16	LocoNet	3200 CZK
RailComDisplay ****	1	N/A	DYI

Tab. 1 Comparison of commercially-available RailCom detectors.
* AMB Modely (2024), ** Roco (2024), *** YaMoRC (2024), **** F. M. Cañada (2024), ***** tvlaky.cz (2024)

However, the module is not available now and its deployment would require additional cabling to connect it to the command station.

5.4 RailComDisplay by F. M. Cañada

A well-known enthusiast for DCC automation F. M. Cañada designed a RailCom module with display, which shows the address of a DCC decoder similar to LRC120 (F. M. Cañada, 2024).

This module is local-only (without any bus), however the big advantage of the RailComDisplay is the availability of schematics. The module is a source of useful inspiration.

5.5 Conclusion

Several commercial RailCom feedback modules exist, and each has its advantages and disadvantages. Overall features are summarized in a Table 1.

For our requirements, we decided to design our own RailCom detector MTB-RC. The detector will be compatible with existing track hardware (no need to add more cabling), and it will be fully customizable as the whole design is upon us. Moreover, we will provide an open--source and open-hardware alternative to the existing commercial products.

6 MTB-RC

An own MTB-RC module was designed. Schematic and PCB design were created in an open software KiCad and are available online[1]. MTB-RC supports the following features.

1. Support for 8 track circuits. This number is a good compromise between scalability and price.
2. Connection to the MTBbus (Jan Horáček, 2024). MTBbus is currently used for stationary hardware control in Track Vehicle Control Laboratory.
3. Open-source and open-hardware design.

6.1 Principles of design

Now we describe important principles of design of the MTB-RC RailCom detector.

The detector is connected to the MTBbus and DCC. These two buses are generally galvanic separated, so the design is composed of two top-level galvanic separated parts.

1. DCC part,
2. MTB part.

https://github.com/kmzbrnoI/mtb-rc

The detector contains a microcontroller (MCU) to process the data. It is not desired for the MCU to be powered off when the DCC to the track is disconnected (e.g. because of a short-circuit on the track). So the MCU lays in the MTB part powered by MTBbus power supply.

The MTB part of the detector contains standard MTB module components like on e.g. MTB-UNI module (Jan Horáček, 2024).

- Jumpers to select the address
- Indication LEDs
- RS485 driver (an integrated circuit)
- Button

An important question is how the data shall be transferred from the DCC part to the MTB part (and vice versa if required). To answer the question we need to understand how RailCom detection works in general.

6.2 RailCom detection principle

The RailCom detection principle is described in NMRA (2024) in detail. Here, we briefly describe key principles.

As we discussed in chapter 3, one of the J and K wires from the command station (or the booster) to the track is simply cut and both ends are connected to the detector. Let's assume the K wire is cut. A Figure 3 shows how these two ends are connected inside the detector.

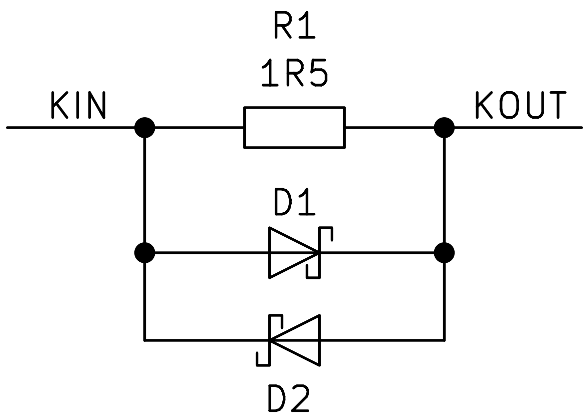

Fig. 3: RailCom current detection principle.

The current flow causes a voltage drop $U_{KIN-KOUT}$ on R1, which is then measured. In addi- tion, protective Schottky diodes D1 and D2 are added to avoid high temperatures of R1 while the maximum continuous current of 3 A flows. The detector uses adequate parts and adequate resistor size to dissipate the heat. Schottky diodes are used so the maximum voltage between KIN and KOUT is never more than cca 0.5 V.

The voltage $U_{KIN-KOUT}$ is then compared using window comparator circuit. See Figure 4.

The comparators are powered from e.g. -5 V to +5 V with 0 = KIN. Voltage references ±18.5 mV are defined in (NMRA, 2024).

The comparators have an open-collector output. If $U_{KIN-KOUT}$ (-18.5, +18.5) mV, (log- ical 0 of RailCom), DATA is connected to -5 V, otherwise (logical 1 of RailCom), DATA is in high-impedance state. Output data are directly in UART format readable by MCU.

MTB-RC contains 8 instances of circuits presented in Figures 3 and 4 to support 8 separate track circuits.

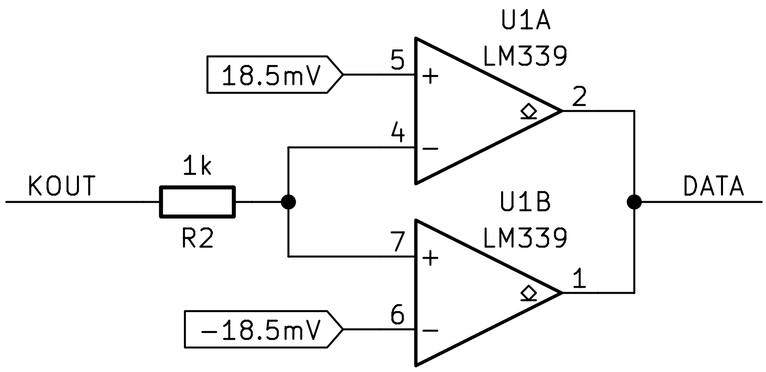

Fig. 4: A window comparator circuit

6.3 MTB-DCC interface

As small MCUs with 8 separate UARTs basically do not exist, multiplexing is used. STM32F103 ARM Cortex-M3 MCU was selected as a main MCU, because of its availability, relatively low price, powerful core, and advanced peripherals (STMicroelectronics, 2024). STM32F103 has 3 UARTs, out of which one must be used for MTBbus communication. The remaining two UARTs shall be used for RailCom reading. This implies 4-way multiplexing. The proposed solution is not ideal, as some data are always lost by design, but the loss should not break address detection.

The multiplexer is present in the DCC part. This implies no MCU in the DCC part is required. The interface between the DCC part and the MTB part consists of:

- 2 UARTs (DCC → MTB)
- 2 multiplexer-control wires (MTB → DCC)
- DCC signal (DCC → MTB)
- Cutout signal (DCC → MTB)

Cutout signal is used to inform MCU that RailCom cutout occurred, so the MCU starts reading RailCom data (via UARTs). DCC signal is transmitted so the processor can read the DCC from the command station, so it knows to which command the decoder answers.

6.4 Multiple decoders detection problem

One of the key problems of RailCom detection is multiple decoders in a section. As discussed in cahpter 3, each decoder sends its address in channel 1, which results in unreadable data when multiple decoders are present in a single track section.

The idea of multiple address detection is to read DCC from the command station and check for RailCom data in channel 2. If the command station sends any command to a decoder present in the track, the decoder (and only the decoder!) responds in channel 2. This allows us to detect decoders, even if there are more of them on the track. A necessary condition is the decoder being addressed by the command station. This issue must be solved separately and is out of the scope of this paper.

6.5 Firmware

The firmware for STM32F103 MCU is implemented in C language and is available online[2]. STM32 HAL library was used. The firmware consists of RailCom decoding, DCC decoding, and MTB communication.

7 CONCLUSION

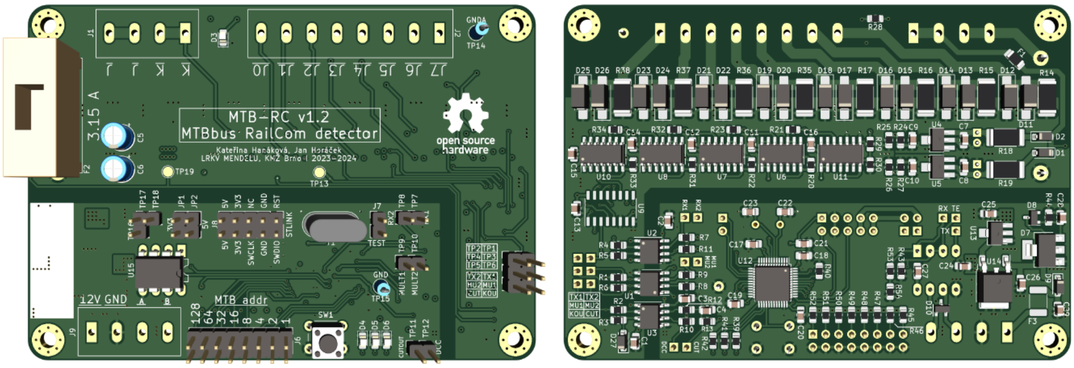

Fig. 5: MTB-RC PCB

Design of the MTB-RC RailCom detector has been described, the module has been manufactured, firmware was implemented and the module passed basic laboratory tests.

Overall parameters of the MTB-RC are summarized in a Table 2. Compare the table to the Table 1.

Detector	N.o. tracks	Buses	Price
MTB-RC	8	MTBbus	~ 600 CZK

Tab. 2 Overall features of MTB-RC.

The detector makes it possible to use additional functionalities of the track operation. One of the functions that we expect to make operations more efficient is the automation of deploying another vehicle in a designated section (most often a locomotive depot). This minimizes the time spent on engine changing and avoids errors caused by entering the wrong address by an operator.

Putting the RailCom detector in operation is the first step to a more reliable and efficient track control system equipped with the ability to read acknowledgments of DCC commands sent to vehicles.

Acknowledgments

This paper was supported by the project CZ.02.1.01/0.0/0.0/16_017/0002334 Research Infrastructure for Young Scientists, this is co-financed from Operational Programme Research, Development and Education.

REFERENCES

NATIONAL MODEL RAILROAD ASSOCIATION. 2024a. Communications Standards For Digital Command Control. *NMRA* [online]. https://www.nmra.org/sites/default/files/s-92-2004-07.pdf [visited on 01/11/2024].

NATIONAL MODEL RAILROAD ASSOCIATION. 2024b. DCC Working Group – DCC Home Page. *NMRA* [online]. https://www.nmra.org/dcc-working-group [visited on 01/11/2024].

CAÑADA, F. M. 2024. RailComDisplay – Local display of RailCom data. *RailCom* [online]. https://usuaris.tinet.cat/fmco/railcom_en.html [visited on 01/11/2024].

DCCWIKI. 2024a. History of Digital Command Control. *DCCWiki* [online]. https://dccwiki.com/DCC _ History [visited on 01/11/2024].

DCCWIKIB. 2024b. RailCom. *DCCWiki* [online]. https://dccwiki.com/RailCom [visited on 01/06/2024].

Jan Horáček. MTB – IO bus for model railroad control. https://mtb.kmz-brno.cz/ [visited on 01/13/2024].

ITVLAKY.CZ. 2024 Z21 detector Roco 10808. *itvlaky* [online]. https://www.itvlaky.cz/dcc-vazby/10808/ [visited on 06/24/2024].

AMB MODELY. 2024 LE15120 LRC120 zobrazovací modul adresy RailCom. *AMB Modely* [online]. https://www. ambmodely.cz/le15120-lrc120-zobrazovaci-modul-adresy-railcom/d-8149 [visited on 01/06/2024].

NMRA. 2024. Communications Standard for Digital Command Control Basic Decoder Transmission. *NMRA* [online]. https://www.nmra.org/sites/default/files/s-9.3.2_2012_12_10.pdf [visited on 01/06/2024].

ROCO. 2024. Z21 detector. *Z21* [online]. https://www.z21.eu/en/products/z21-detector-railcom [visited on 01/06/2024].

STMICROELECTRONICS. 2024. *STM32F103x8 Datasheet*. https://www.st.com/resource/en/datasheet/stm32f103c8.pdf [visited on 01/13/2024].

YAMORC. 2024. YD6016LN-RC RailCom enabled feedback module. *YaMoRC* [online]. http://yamorc.de/products/?singleproduct=1246 [visited on 01/11/2024].

Contact information
Jan Horáček: e-mail: jan.horacek@mendelu.cz
Jiří Rybička: e-mail: jiri.rybicka@mendelu.cz